2017

Adaptive Neural Architectures for Intuitive Robot Control

Melidis, Christos

http://hdl.handle.net/10026.1/9998

http://dx.doi.org/10.24382/670

University of Plymouth

ADAPTIVE NEURAL ARCHITECTURES FOR INTUITIVE ROBOT CONTROL

by

CHRISTOS MELIDIS

A thesis submitted to Plymouth University

in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Computing, Electronics and Mathematics

Faculty of Science and Technology

May 2017

ADAPTIVE NEURAL ARCHITECTURES FOR INTUITIVE ROBOT CONTROL

by

CHRISTOS MELIDIS

Abstract

THIS thesis puts forward a novel way of control for robotic morphologies. Taking inspiration from Behaviour Based robotics and self-organisation principles, we present an interfacing mechanism, capable of adapting both to the user and the robot, while enabling a paradigm of intuitive control for the user. A transparent mechanism is presented, allowing for a seamless integration of control signals and robot behaviours. Instead of the user adapting to the interface and control paradigm, the proposed architecture allows the user to shape the control motifs in their way of preference, moving away from the cases where the user has to read and understand operation manuals or has to learn to operate a specific device. The seminal idea behind the work presented is the coupling of intuitive human behaviours with the dynamics of a machine in order to control and direct the machine dynamics. Starting from a tabula rasa basis, the architectures presented are able to identify control patterns (behaviours) for any given robotic morphology and successfully merge them with control signals from the user, regardless of the input device used. We provide a deep insight in the advantages of behaviour coupling, investigating the proposed system in detail, providing evidence for and quantifying emergent properties of the models proposed. The structural components of the interface are presented and assessed both individually and as a whole, as are inherent properties of the architectures. The proposed system is examined and tested both in vitro and in vivo, and is shown to work even in cases of complicated environments, as well as, complicated robotic morphologies. As a whole, this paradigm of control is found to highlight the potential for a change in the paradigm of robotic control, and a new level in the taxonomy of human in the loop systems.

Contents

List of Figures

Acknowledgements

First off all, I would like to thank Prof. Davide Marocco for his supervision in the last three years; for allowing me to explore and enabling me to pursue new ideas, in shaping me as a researcher and as a young academic, my supervisory team, Dr. Martin Stoelen, for his support and input, Dr. Jeremy Goslin for the thought provoking conversations, and Prof. Mike Phillips for sharing new perspectives.

I am grateful for being part of CogNovo, a team of excellent people, kind in heart and strong in mind, an interdisciplinary group of researchers that were able to work as academic peers as well as friends. I would like to thank Prof. Sue Denham and all the rest involved in the formation and organisation of what was a one of a kind Doctoral Training Centre.

A very special thanks goes to a great teacher Ioannis A. Tsoukalas, the person who foresaw the potential in my younger self, and showed me an attitude towards education which I since aspire after, for without him I doubt I could get to this point.

Special thanks should go to my family and friends for always being there, supportive and encouraging in the easy and hard times. For the joy and laughter, the headaches and the cheers.

Authors declaration

AT no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award.

Work submitted for this research degree at Plymouth University has not formed part of any other degree either at Plymouth University or at another establishment.

Relevant scientific seminars and conferences were regularly attended at which work was often presented, external institutions were visited and several papers have been published.

Publications:

Melidis C., Marocco D. An Exploration on Intuitive Interfaces for Robot Control Based on Self Organisation. In: Headleand C., Teahan W., Ap Cenydd L. (eds) Artificial Life and Intelligent Agents. ALIA 2014. Communications in Computer and Information Science, vol 519. Springer, Cham, 2015

C. Melidis and D. Marocco A Human Centric Approach to Robotic Control, IEEE International Conference on Systems, Man, and Cybernetics, 2015

C. Melidis and D. Marocco KURE: a Two-Way Adaptive System for Intuitive Robot Control, Workshop on Behavior Adaptation, Interaction and Learning for Assistive Robotics, IEEE RO-MAN, 2016

C. Melidis and D. Marocco KURE: Kinematic universal remote interface a human centred remote robot control paradigm, IEEE International Conference on Systems, Man, and Cybernetics, 2016

C. Melidis, H. Ilzuka, and D. Marocco Intuitive control of mobile robots: An architecture for autonomous adaptive dynamic behaviour integration, Springer Cognitive Processing, 2017 (*in press*)

Conference presentations:

Artificial Life and Intelligent Agents. ALIA 2014 , Bangor, Wales, UK. An Exploration on Intuitive Interfaces for Robot Control Based on Self Organisation

IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015 . Kowloon, Hong Kong. A Human Centric Approach to Robotic Control

IEEE RO-MAN 2016 , Workshop on Behavior Adaptation, Interaction and Learning for Assistive Robotics, NYC, USA.KURE: a Two-Way Adaptive System for Intuitive Robot Control

IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 , Budapest, Hungary. KURE: Kinematic universal remote interface a human centred remote robot control paradigm

Word count for the main body of this thesis: **55528**

Signed: ______________________________

Date: ______________________________

Introduction

Motivation

Sensing and acting machines had always attracted attention, both as a research field and in the commercial market. Nowadays, the idea of commercially available robots is farther more realizable than ever. User interaction has been a surging field for both communities and markets, and human robot interaction and control have long been active research fields. Whether robotic morphologies are built for entertainment or not, remote control holds a significance as it allows the user control over the robot. Usage of robotic morphologies as tools has found applications in industry, rescue missions as well as military. Under this perspective the task defines the constrains of the control mechanisms and dictates the interfacing approach. Most systems, being specifically constructed for a given task, are realized with the restrictions embedded in the operational mechanisms. Most of the complications in control arrive from this design pattern. One usually needs to train oneself before using and interface, in order to familiarize with the control commands. Also, the control of most systems becomes laborious and difficult, when a sequence of commands has to be communicated to control the robot. These observations signify a need for change in the control paradigms and interfaces deployed. Hence, our approach is deemed worthy; changing the way we understand robotic morphologies and their behavioural aspects, while, at the same time, the interfacing mechanisms can be realized closer to the human operator, easing and assisting the usability of the system, by incorporating intelligent aspects.

One of the best examples of the complexity in controlling robotic morphologies can be found in the 'WowWee Robosapien X'[1] which is a humanoid robot toy, which features a controller with 23 keys each one having two functions mapped on it. Other examples

[1] http://www.wowwee.com/robosapien-x/, Robosapien X website

of commercial robotic toys, with similar controlling complexity include the ‘Roboraptor’ featuring a PlayStation like controller, the ‘Robome’ a three wheeled robot controlled though a smartphone, and the Parrot ‘Jumping Sumo’, a two wheeled robot also controlled through a smartphone. Besides the simplicity of the actuators and the low degrees of freedom of the two latter examples, and the complexity of the first ones, the idea that the user has to familiarise oneself with the control commands and paradigm remains. The possibility of a one-to-one mapping of the input device to the robot makes the problem less visible, but still, without providing an intuitive way of control for all the users. All the existing applications follow the assumption that the user has to be trained, and that all the users have the same competencies and are equally enabled in using the provided input device.

The remote control of mechanical devices equipped with a large number of actuators, such as humanoid robots, is a challenging task. When dealing with the resulting large number of degrees-of-freedom, the nature of the interface provided to the human operator plays a fundamental role in the success of tele-robotic performance. A wide range of tele-robotic interfaces have been explored so far; some are very rigid devices that require a great deal of cognitive and manual effort, while other more intuitive systems, based on one-to-one body mapping, are in contrast very complex and expensive devices, often specifically tailored to a single robotic platform [1].

Providing a mapping between simple control mechanisms and complex robots is not easy. Taking into account the current state of technology and the most recent advances in machine learning, artificial intelligence, artificial life, human machine interaction and control theory, a framework is proposed capable of coping with the task. At the same time and in order to make such an interface we consider human intuition and ones personal understanding of the affordances of the controlled morphology and mechanics used for control. In the studies described in this thesis, most of the paradigms are applied to robotic morphologies. The talk is about robotic morphologies and not robots, in order to disentangle the reader from the perception of a prototypical robot. This as one of the pillars of the work undertaken regards the self organisation of robotic behaviours in sensing and acting machines of various shapes and sizes. Even more, when in cases, we consider as robots mechanical parts with actuating capabilities 'living' in simulated environments. The theories applied are taking into account the nature of the data the interface should work on, without making any assumptions - explicit or implicit-, neither for the controlled robotic morphology nor for the way of communication with the interface operator. Thus we can see how the principles investigated here can be of use in any type of machinery that has sensors and actuators.

The framework described and analysed in this thesis is accompanied and supported by the implementation of an agile interface able to control every possible robotic morphology, a universal interface, based on the principles stated by the framework. This implementation provides an evaluation method of the principal properties of the proposed framework. At the same time, it accesses the viability and reliability of the algorithms, technologies and methods proposed. Finally, it reveals the constraints and possibilities of such a system. The author's belief is that a system of such complexity must be performed to be fully understood and only through the realisation of such control and communication processes can we truly govern such creation.

The core idea guiding the evolution of the interface is motor control. Based on building block hypothesis of behaviours from actions, we pursuit the exploration of the simple actions of robotic morphologies that can serve the construction of behaviours. The majority of cases for robot control are biased in the way they envision the behavioural repertoire of the robotic morphology at hand. Predefined robot behaviours or configurations for control, may not match the possible behaviours of the robot in the environment it inhibits. At the same time an autonomous way of exploring the behavioural potential of the robot allows for a universal framework for the control of arbitrary robotic morphologies without being explicitly designed for it. Techniques exploring these building blocks of robotic behaviours inside their environment provide more useful behaviours for the user to operate on. Having an implicit assumption of how a robotic morphology is bound to behave or act, prejudges the way that the operator communicates their intention for control. Co-evolving behaviours of both the robot and the operator changes the paradigm of control, allowing for a more meaningful communication, taking into account not only constrains of the robotic morphology, but also societal, preferential and other aspects coupled to the operator and the operating environment that should otherwise be explicitly defined.

The framework is described modularized as it is essential to account for changes and different techniques to be used later, by the author or others. In the chapters to follow the framework will be explained in detail and the constrains will be outlined, taking into account the interface implementation accompanying the given thesis, the particulars of the modules implemented, as well as the possible alternative mechanisms supporting partial or full functional equivalence of the modules at hand.

Objectives, Scope and Contribution of the Thesis

Our goal is the implementation of an agile interface able to control every possible robotic morphology, a universal interface. To do so we need an automated mechanism that can examine and explore the robotic morphology connected to the interface and extract interesting features, with respect to the desired control pattern. We identify interesting features as behaviours that can be produced by the robot and are meaningful to the user, according to the task in hand. The purpose of the interface is to map the behaviours of the operator to those produced by the robots, resulting in the association between the robots and operator behaviours. To achieve this, we reverse the informational flow of the interface, as suggested in [2]. The robot acts first and the operator responds to the exhibited behaviours with their own, through the input device. The input device thus, plays a critical part on the behaviours the operator can implement. Multidimensional input devices, i.e. Kinect sensor, could enable a whole body mapping, whereas simpler ones, i.e. on-off switches or joysticks, are more restrictive [3], [4] and [5].

The proposed interface is able to explore the capabilities of the robotic morphology based on the homeokinesis principle [6]. As described by Martius and colleagues in [7], self organisation of the sensorimotor loop can explore the behavioural repertoire of a robot. Based on this research we formulate the principles for the interaction between the interface and the robot. For the interaction between the interface and the human operator we propose a framework for a behaviour based interaction.

Putting the ideas of user interfaces in the context of intuitive control, provides us with a way of dealing with the interfacing complexity, merging ideas from User Interface Management Systems and Ecological Interface Design (EID). Enabling user interaction with complex systems, according to EID, need a systems of equal complexity. Given though, a suitable representation of the robot, and an intelligent interface, we have the potential of reducing the complexity that the user has to face in the interaction with the system. At the same time, the idea of intention extraction from the user shows a way forward in defining the principle components of the framework and interface, in order to assist the remote and intuitive control of robotic morphologies of arbitrary complexity.

Contributions in Robot Self Organisation. Self Organisation of behaviours has been a goal of numerous researchers in the past. Driving inspiration from their work, the goal here is to extend their methods by incorporating human intervention. Dynamical Systems, provide robust controlling mechanisms for robots, while Machine Learning can help adapting these methods, according to the particulars of each robot. Putting everything together, a greater vision for a collaboration between human and robot is put forward. Enabling the robot to self explore its potential behavioural variety and at the same time refining these behaviours according to the preferences of the operator.

Contributions in Time Series Control. Having as a starting point, established models and architectures for time series manipulation, the methods proposed provide a new vision in the way we can understand Human - Machine, or Robot, interaction. The aim is to develop a robust algorithm and architecture, capable of supporting the time variant configurations of the potential input devices used. In this extend, the research is extended towards time aware Artificial Neural Network architectures, capable of adapting to the timed inputs. To this extend, the proposed mechanisms not only removes a great load from the designing process of input devices, but also personalise them according to the operator's needs.

The vision behind this work is to make a unified approach in embodied communication, as afforded by the input device. Provide a high-level translation of behaviours, regardless of the morphology in hand. Entangle the time and spatial domain of sensing and acting machines, with human behaviours. Support high level, efficient, communication with minimal effort. In the area of Dynamical Systems, our work can be realised as a mapping from the input space to the sensorimotor space of the robot. Making this mapping time varying and adjustable towards both ends, results to a problem of two way optimisation, towards the robot and the user. Given the nature of the input, this research can scale from a translation of embodiment, to abstraction and dimension reduction. That being said *Motor Schemata Theory*, as put forward by *Michael A. Arbib*, provides a proof and an investigation drive in developing a framework capable of identifying, extracting and reusing them, not only autonomously but also in collaboration with a human operator.

On a more theoretical level, based on the ideas of situated, embodied, and enacted cognition, we investigate the way we communicate our movements to another morphology; the way that we understand and use our body. We can observe how the material agency of the input device affects and affords the user's control patterns. An investigation on how the mediated experience of another body -through the input device and interface- can result to a kinesthetic experience, enhancing the way understand the morphology and its environment. As a parallel to Boden's 'conceptual spaces', this interface aims to provide the constrains and allowances for the range of possible mappings between user and robotic morphology.

The original contributions of this thesis can be summarised as follows:

- investigation on the potentials of autonomously adapted architectures for the intuitive control of robotic morphologies by humans ;

- provide a novel architecture for the formation of intuitive robot control paradigms;
- provide a methodology for approaching the autonomous adaptation of control strategies, bringing together methods from autonomous robotics, human-robot and human-machine interaction, together with user interface design and theoretical perspectives on intuition;
- experimental evidence demonstrating the usability of Recurrent Neural Architectures on user behaviour acquisition and specifically Recurrent Neural Networks with Parametric Bias architectures;
- extensive experimental evidence on the usability, properties and effectiveness of Echo State Networks and Reservoir Architectures on human behaviour recognition and mapping;
- experimental evidence from the application of the proposed paradigm, method, and architecture in the intuitive control of two wheeled mobile robots;
- experimental evidence from the application of the proposed paradigm, method, and architecture in the intuitive control of multiple Degree of Freedom robotic arm;

Structure of the Thesis

The structure of the thesis follows the experimental investigation undertaken and its publication record. Starting from the investigation of randomly robotic morphologies and an exploration of possible methods, the system is iteratively improved reaching the points of user testing and control of robots in real world scenarios.

Chapter 1 provides the theoretical background and perspectives surrounding our work. Divided in three sections, first (*Section* 1.1) provides the general theoretical surrounding of our methods and work and given the reasoning behind our methodological approach. Secondly, in *Section* 1.2 it provides the background and up to date research examples on the methods for robot behaviour organisation. Third, in *Section* 1.3 it provides the background and up to date research methods in dealing with human input in the field of robotics. Finally, the chapter is concluded with a *Synopsis* 1.4, where everything is brought together to highlight our approach and main pillars of research.

Chapter 2 provides the *Methodological Background* of the work in this thesis. It describes the methods for the *Interface the Human Operator* and the methods for *Interfacing the Robotic Morphology*.

In *Chapter 3* we discuss the experimental results of the first iteration of our proposed method. Working with Recurrent Neural Networks with Parametric Bias and randomly constructed robotic morphologies our idea of intuitive robot control is put forward as a *Human Centric Approach to Robotic Control*.

In *Chapter 4* we proceed with an exploration on efficiency and efficacy of Reservoir Architectures - and Echo State Networks in particular - in the on-time adaptation of an interface suitable for Intuitive Robot Control. Working with new (i.e. compared to *Chapter 3*) randomly constructed robotic morphologies we show the applicability and generality of the methods for robot control. Our proposed architecture is put forward as a *Two-way Adaptive Interface for Intuitive Robot Control*, as we demonstrate the adaptation procedure in both ends - the human and the robot.

In *Chapter 5* we proceed to a detailed and extensive experimentation on the properties of the Echo State Network as hinted in the previous chapters. The validity of the approach is benchmarked, tested in vitro and also with human participants.

In *Chapter 6* describes experiments of our proposed -novel- system in a scenario of control of an 'e-puck' robot. The system is tested in whole, using a simulated version of the robot.

Chapter 7 describes experiments of our proposed system with an 8 Degree of Freedom robotic hand in vivo. The setup is put to test in the behaviour exploration in a real-complicated robot and tested in the remote control of a 'door opening' scenario.

Finally, this thesis concludes with an overview of the results of the thesis, future perspectives and applications of the research undertaken in other fields.

Chapter 1

Theoretical Background

Feeling the Body, Human-Machines Interaction and Cybernetics

In this chapter the concepts and theory are elaborated in detail providing the theoretical introduction of the work presented in the thesis.

Our work lies in the intersection of cybernetic control and human in the loop systems. Our approach is meant to provide a novel autonomous way of coupling human and robot dynamic behaviours as to enable human control over the robot. For the work of this thesis we work on a two-fold, dealing with the autonomous generation of robot behaviours on one hand and with the extraction of dynamic behaviours from the human operator on the other.

In this chapter we start from theoretical aspects of embodied cognition and approaches to the creation of artificial cognitive systems, continue with the field of robot control and investigate the current state of the art and approaches found in literature for the organisation of behaviours in robotic platforms. We proceed with a similar investigation in the field of Human-Machine Interaction followed with a more specific view of Human-Robot Interaction. Through the investigation the differences between the two fields are highlighted as well as the commonalities. Finally, everything is brought together providing the theoretical description of the system.

1.1 Philosophical Perspectives to Embodied Cognitive Science

While control and interaction systems research has little impact on the formation of robotic behaviours, there lies an interest in the entanglement of human and robot behaviours. Even more in the case of a robot aware of its surroundings situated in an environment, aware of the possibilities it has to offer. Situating the human in the robot loop, in a transparent manner, presents a unique possibility for control, as the controller and controlled must co-exist and work in harmony for the accomplishment of any goal to be successful. In our research we take up this challenge of providing a first person experience of the robot's realm to the human, coupling the sensorimotor loops of human and robot, creating a cybernetic control paradigm. Looking from the human's perspective new kinds of experience can emerge; existing through a different embodiment, in the same environment. Looking through the robot's view, its sensory apparatus now extended with the capacity and experience of the human allows for an existence beyond the reach of any artificial approach.

1.1.1 Embodied Cognitive Science

Moving away from the classical paradigm of symbolic AI, the embodiment thesis provides an alternative for the explanation and creation of intelligent systems. The novelty lies with the idea that sensorimotor dynamics can only be perceived and analysed in coupled manner. This, as the experience is actively altered through the actions of oneself and their effects in their body as well as the environment. Under this paradigm, the system's controller is placed in the body (embodied) and the environment (situated), guiding movements (enacting) that acquire information. Thus, the paradigm shifts from a computer processing information to a more wholistic approach where the body and the environment are taken into account as well as the actions of the system as they relate to the changes perceived in the environment.

Embodiment: Many features of cognition are embodied in that they are deeply dependent upon characteristics of the physical body of an agent, such that the agent's beyond-the-brain body plays a significant causal role, or a physically constitutive role, in that agent's cognitive processing [8].

Enactivism Seminal is the work *The Embodied Mind* by Varela et al. [9], in the field of embodied cognition. In their work they call for a change in the direction of cognitive sciences and the creation of cognitive systems. Stressing out that symbolic representations together with an a priori knowledge cannot accommodate the feedback from the dynamic changes in the environment resulting from the actions of an embodied agent. The main idea being that cognition is a dynamic sensorimotor activity, and that the environment is experienced as the outcome of the actions of the embodiment. This idea of inseparable perception and action gives rise to their new 'enactive' approach.

Sensorimotor Contingencies In their work O'Regan and Noe [10] propose a new approach in dealing with representations. Opposing the usual idea that there are internal representations stored in the brain, they propose that the world serves as an external memory of the brain. Under this paradigm perception arises from what they call 'sensorimotor contingencies' (SMC), and make for the action, physical properties and characteristics based on sensory information.

Under this paradigm the idea would be that an cognitive system could be steered based on modulations of its sensory systems. In extension, taking this idea one step ahead behaviours could be formed from the re-occurrence of sensory inputs and their relative states. Extending this theory one can arrive to the formation of an internal notion of space, by means of sensory offsets, consistencies and co-occurancies [11].

1.1.2 Ecological Perception - Affordances

The concept of *affordances* was first introduced by J.J. Gibson [12]. It described the potential action enabled by an environment or a given object, especially one that is easily discoverable. These 'action possibilities' latent in the surroundings of an agent, need be discovered by the agent, providing them with a unique view.

Affordances, or clues in the environment that indicate possibilities for action, are perceived in a direct, immediate way with no sensory processing. Examples include:

buttons for pushing, knobs for turning, handles for pulling, levers for sliding, etc. Based upon Gestalt theories, Affordance Theory has various implications for design, human-computer interaction, and ergonomics amongst others. Some believe that good design makes affordances explicit.

Extending the idea of affordances to bodies introduces the term of bodily affordances [13, 14, 15, 16]. Bodily affordances describe how one's body is arranged, while indicate possibilities for its movement. Extending the concept of body schema, 'bodily affordances' include both postural and structural elements of the body.

Two are the main implications of these perceptual possibilities. One, that the robots morphology has a build in potential for movement. According to Gibson such a potential is recognised by an observer as is. It is part of the material agency of the body and the arrangement of its parts that hold the potential for its manipulation. On the contrary it is according to Norman that experience also plays a part in the formation of affordances [17, 18]. Thus, affordances are placed between actor and acted, in the relationship that holds between the object and the individual that is acting upon it.

Following this latter idea of affordances, we aim towards the extraction of the morphological affordances of the robot in the form of behaviours. These refer to the afforded behaviours of the morphology in the environment formed by the internal interactions of the robot's sensory and motor apparatus. As affordances are also present in the user's perception of the acting robot, our approach aims at an enacted approach through which the robot is able to demonstrate its afforded behaviours in its environment, while the user gets habituated in the 'new' creature and its behaviours and also builds up an experience of it acting. In that sense, the user learns afforded behaviours of the robot, while explores new possible ones by acting upon it (through our proposed system).

1.1.3 Intuition

Intuition is defined as the ability to understand something instinctively, without the need for conscious reasoning[1].

[1] Definition of 'intuition', Oxford Dictionary

This idea of, unique possibilities arising from the same structures, is applicable on the way we perceive control devices. Different people have the possibility of acting in a different way upon them. For such a process to be triggered, the human (operator) must have the possibility of freely manipulating the control mechanism. The interaction paradigm and the interfacing techniques should be able o support such activity. The interface must have the ability to adapt to the user. This idea, carries one of the most important aspects of the interfacing framework described here and results to the need of intuition by the user and intelligence from the interface.

Combining intuition with affordances as recognised characteristics of the interfacing mechanism, enables for an interface tailored to the user. Being able to capture ones instinctive perception on the affordances of the presented control device, has the potential for a thought free interaction. This, as the system is adapted and learns from the user's usage of the input device to recognise the natural, intuitive behaviours of the user as such emerge. By definition such a paradigm can ease the cognitive load for the manipulation of any input device.

At the same time, enabling the user to freely express the way of communicating their intentions for control, provides us with a new way of dealing with ergonomics. This as it is possible to adhere the personal preferences of the user that would otherwise be suppressed by the imposed manipulation of the device as inspired by its creator. In contrast to the norm of training the user to the device and the interaction supported, the device can be freely adapted and the interface trained for the usage instructed by the user.

1.2 Robot Behaviour Dynamics

In this section methods for the creation and formation of dynamic robotic behaviours are described. From the literature two ways of organising robot behaviour are found and two paradigms arise. Research following a top-down organisation of robotic behaviours deals with the production of pre-designed behaviours in a well structured manner. On the other hand research following a bottom-up organisation of robotic behaviours, describes how robotic behaviours can be synthesised based in low level interactions of the robot with its environment. In doing so, it exploits patterns and correlations in the sensorimotor interactions of the robot. Both ways can result in ‘intelligent’ systems, although in the first case most of the ‘intelligence’ lies with the designer of the system, while on the other ‘intelligence’ can be seen to emerge from seemingly trivial -or even naive- low level instructions. In the first case the potential of the robot is already known, its behaviours being well instructed and the reasoning behind them possible to be traced back. In the latter, it is the combination of the smaller component behaviours that produces the complexity of the robotic behaviours, in most cases able to only be qualitatively described.

Here, and in contrast with most approaches in human in the loop systems, we first pursue the creation of an autonomous robot loop. Segmenting this loop, we search for smaller consistent behaviours of the autonomous system to guide it. Thus, in this section we explore the possibilities for the creation of an autonomous loop as rich as possible, that can provide a deep understanding of the robot and its body in the context of its environment. The assumption is that having a rich autonomy, assumes rich behaviours being used from the robot. Even more, being able to tie such behaviours to the robot and its environment rather a pre-set goal, would mean that the enacted aspects of the embodiment are adequately explored and that there lies a knowledge of their co-dependence within the robot. A dependence that we can exploit, by extracting and reusing it, for the needs of the human operator rather than the needs of the autonomous system.

The cybernetic scope of the approach allows for non-optimality -in terms of control theory- of the explored behaviours, and endorses the idea of behaviours that are useful for the robot. It is in the field of cybernetics that robot-centred approaches first appeared, cases where minimal human intervention and self-organisation scaffolded emergent ‘intelligent’ and purposeful behaviours from seemingly simple structures.

Behaviour A well suited definition and explanation of behaviour is given by Nolfi [19]. There, behaviour is defined as a dynamic process based on the interactions of the robot's control system with its body and later with the environment. Behaviours are seen as dynamical processes that result from shorter time sequences of interactions. Based on the interactions of the three elements mentioned above - control system, body and environment - behaviours and behavioural properties emerge. These behaviours or their properties cannot be traced back to any of the elements taken in isolation. Thus, behaviours are innate to the agent, body and environment three-fold and can only be exhibited and studied as part of it.

Body Working on the same definition of behaviour as above, the body of a robot refers to the link between the control processes and the environment, while at the same time it frames the interactions with the environment. With the embodiment of a robot both the morphology and its material properties are addressed. Through the morphology the shape of the robot is addressed, its limbs and where they are attached, the available sensors and their location. With material properties, the reaction of the robots body parts to forces is addressed, as for example the deformation in the case of contact with environmental elements.

Control System Coming to the control system, it is here that information from the sensory apparatus of the robot is combined and with or without higher levels of reasoning the decision is taken as to how and when the motor apparatus should act. The control of a robotic system is a task of varying difficulty based of the morphology at hand. The degrees of freedom of the morphology as well as the particularities of the environment, allow for multiple solutions. The level of expected autonomy of the robot, ordered from high level commands (i.e. proceed to the next room) to low level commands (i.e. arrange a specific joint to certain degrees) allows and restrains for different applications. While, in most cases the level of expected autonomy of the robot is dictated by the performed task and goals. Based on the information flow that on the control system two main paradigms are shaped, these of top-down and bottom-up organisation.

1.2.1 Top - Down organisation

Traditional Artificial Intelligence (AI) research follows a top-down approach, usually involving a complicated, centralized controller that makes decisions based on access to all aspects of the global state, also referred to as top-down approach. Using such an approach, the capabilities of the robot are chosen beforehand and instructed to the robot. This procedure is usually described using symbolic formalisms described through mathematical logic, usually referred to as the symbolic AI. Such a procedure ‘builds in’ all of the cognitive abilities of the robot from the outset. The result is typically a ‘logical reasoning’ engine that imposes control over the entire robot and includes higher level cognitive functioning like knowledge representation, planning and decision making.

Under this paradigm, the information from the sensors of the robot is used to create a world model, a -usually- complex representation of the environment. This usually communicates with a reasoning - planning - module, where the decision of the robot’s action is taken. Finally, the decided action is executed by the robot in its environment. The complicacy of such a system is handled by the designers, the people programming these modules. Because of that, classical AI systems do not cope well with the uncertainty of real world situations and have limited flexibility in their operation.

1.2.2 Bottom - Up Organisation

There are, though, systems build with a bottom-up approach, where localized, parallel, and distributed low-level mechanisms interact with each other, providing the system with adaptive and complex behaviours. Behaviour Based robotics (BBR), Nonlinear Dynamics and Self Organisation, and Artificial Life (ALife) are research fields developing systems that follow this bottom-up approach.

Under this paradigm the control system is made so that it has dynamic internal representations of its state, which in turn provides flexibility in the actions it can take. The integration of the sensory information does not rely on well defined rules of logic but rather in distributed representations. Although such a representation is more difficult for the designers to analyse and understand, it provides robots with a more ‘personal’ view of the environment. As an example, a proximity sensor’s reading is used without any ground truth information available, making the system more robust against noise as well as malfunctions. Rather than imposing a barrier between hot and cold, such a notion arises from the sensors history of samples and thus is relative to itself. This, as the system learns to operate through experience and in response to the incoming information.

Furthermore, bottom-up approaches come with built-in learning mechanisms that provide such architectures with adaptivity and flexibility. As their representations get more and more examples their responses become better and more complex. This provides an alternative to the top-down approach where the knowledge base of the system is code beforehand. Here, given enough examples architectures become better, in cases reaching better results than engineered systems. At the same time, given their distributed nature, they have a closer relation to the structure and functioning of the biological brain. Examples of the widespread usage of such architectures can be found in cases of mimicking low-level brain functions. Such include image recognition, robotics and motor control, computer vision as well as speech recognition. In extend to the above and given their reactive nature, they can easily cope with data of the time domain. Here many examples of excellent performance are found in music, dynamic gesture recognition, video recognition as well as text generation amongst other.

Behaviour-Based Robotics

Behaviour Based Robotics (BBR) bridges the fields of artificial intelligence, engineering, and cognitive science [20]. In the context of behaviour based robotics, behaviours are defined as responses to stimuli, the perceived effect (through the sensors) of the environment on the robot [21]. This way BBR can be viewed as bottom up construction of behaviours, based on the structure of the internal mechanisms, namely the controllers of the system. These behaviours can either be used to maintain or achieve goals.

The idea here is that the system is able to perform a -usually small- set of simple behaviours. These behaviours are usually pre-programmed and part of the system from the design process. This set of behaviour that the robot has at its disposal is referred to as the behavioural repertoire of the robot. What is important and differentiates BBR to other approaches is that these behaviours being simple enough can be executed in parallel and also combined. While each behaviour alone is not enough for the system to be useful (e.g. autonomous wondering of a vacuum cleaner), the combination of smaller behaviours can achieve this results. For example, while avoiding obstacles, or wall following are not on their own enough for a system to autonomously explore an area, combined they create a more ‘intelligent’ machine. This, as the machine is not only to perform the two, but also their combination which brings us to the emergent properties of such architectures.

Given a small repertoire and a well defined environment, a system like the above can be analysed enough so that its behaviour is considered known. The application domain of most such system though is unstructured dynamic environments. This imposes a constrain in the analysis of such systems as there cannot be a benchmark or method to characterise them with scrutiny and in a rigorous way. This, becomes even more so the case when consider adaptive and flexible systems that change according to the changes of their environment and not always perform the same actions in similar situations.

To analyse such systems one needs to allow them to experience their environment. Since their behaviours are reactive to the stimuli of their environment, they can only be triggered as so. The internal representations and states of the robot really as much to the consistencies of the sensors as much to their inconsistencies. This notion of situatedness holds one of the key philosophical aspects of BBR and a view point towards the approach to the creation of ‘intelligent’ machines.

Non-linear Dynamics

Viewing an embodied agent as a complex dynamical system enables the employment of concepts such as self-organization and emergence rather than hierarchical top-down control. Working with the dynamics of the body, its properties, and their relation to the environment, behaviours are rooted to the characteristics and particularities of the robot and its environment. It is here that dynamical systems theory comes in place, as it provides the methodological and mathematical framework for the coupling of the sensing and acting dynamics of the robot. This in turn allows for the exploitation of statistical regularities in dynamics, enhancing the informational processing of the agent.

One of the very first such approaches is found in the work of William Grey Walter, as well as W. Ross Ashby, two seminal figures of the field of cybernetics. It was in Walter's work, and through the construction of the tortoises, that the dynamic interplay of sensing and acting came in place and highlighted the importance of situatedness. The emergent properties of the setup and the complexity of the behaviours the system was able to exhibit was profound for the time. The analysis of the reactive nature of the system provided the field of cybernetics with the insight that the complexity of our world could be emerging from simple rules and structures.

In the field of non-linear dynamics, a representative example of a bottom approach of self organisation is the work by Ralf Der on the homeokinesis principle [6]. Examples of applications of Dynamical Systems approach to the self organisation of the sensorimotor loop of robotics morphologies can be found in the work of Martius et. al. [7] and [22], Der [23] and [24], and Hesse et. al [25]. In their approach they realize the Dynamical Systems with Neural Networks, showing how; from simple structures and seemingly simple non linear approximations, behaviours can be shaped in robotic morphologies. The idea of goal oriented behaviours is not stated in their research, but has been pursued by others using Reinforcement Learning techniques to guide the exploration.

Artificial Life

In the field of ALife the applications of robot control are also notable [26]. Most of the systems here, combine Evolutionary Computations (EC) and Neural Networks (NN), to form the controllers of the robotic morphologies. Since EC methods are used, the behaviours evolved by the robot are goal oriented. Although the design of robot controllers is a major application of this field of research, the techniques are also used for map building, planning and human robot interaction [27], [28]. Notable is also the work of Stefano Nolfi in the field of Evolutionary Robotics. Evolutionary robotics is the attempt to develop robots through a self-organized process based on artificial evolution [29], and [30]. The main advantage of relying on self-organization is that the designer does not need to divide the desired behavior into simple basic behaviors to be implemented in separate layers of the robot control system [31].

1.3 Human Factors

1.3.1 Human Machine Interaction

Human Machine Interaction (HMI) describes how we as humans interact with dynamical mechanical ssytems (machines) through a human machine interface. A machine is defined as ‘any mechanical or electrical device that transmits or modifies energy to perform or assist in the performance of human tasks’[2] [32]. As an interface is described the shared border between two systems. Here, these are the machine and the input device. Since the manipulation of the input device is performed by the human and input devices are passive, in the sense that are fully guided by the human, the shared border of the two systems can be placed between the human and the machine.

The field of HMI covers the interaction of human with machines ranging from everyday kitchen appliances to computers and robots. Special attention will be given to Human-Robot Interaction later in this chapter. Through the field of HMI an overview of the objectives, approaches and the main problems are going to be introduced. Some of

[2] wordnet.princeton.edu, Definition of Machine.

the concepts apply in our line of research, while other apply to our greater idea and the demand for it's realisation.

Input Devices in HMI

Technologies and particular implementations may change, but the technologies available thus far can be separated in five main categories,

a) *Optic*;

b) *Acoustic*;

c) *Bionic*;

d) *Tactile*, and

e) *Motion*;

, technologies [32]. At the same time, input devices can be characterised by their degrees of freedom, both in linear and angular components. Also, according to their input specifications they can be divided to continuous and non-continuous, distinguishing this way the input from a computer mouse and a computer keyboard. There have been a number of studies and taxonomies attempting to organize this range of possibilities [33], [34].

The input device has a fundamental role in Human Machine Interaction, it dictates the possible input sequences of the user and so the way in which the user can communicate their intentions for control.

1. In the category of *Optic* input devices, the primary hardware used is usually a camera. The user doesn't have to physically manipulate the device, rather their hand motions and gestures are recorded and recognised. This makes such device easy to use, based on the embodiment of the control. Example usages of such HMI methods include self parking car [35], Head gesture controlled wheel

chair [36], eye blink control interface [37], eye tracking used for computer input control [38], Multi-touch interfaces [39].

2. *Acoustic* technologies mainly focus on speech recognition systems, providing a verbal way of communicating with machines. Typical applications include voice controlled wheel chairs [40], voice controlled home appliances [41], or even voice enabled in-car entertainment systems [42].

3. In the case of *Tactile* technologies, the user is required to have physical contact with the input device, pressing for example a button. Examples in this category range from computer keyboards to motion capturing gloves, although one would expect motion gloves to be part of the motion devices.

4. *Motion* technologies include devices capable of capturing motion, usually with the use of gyroscopes or accelerometers. Applications can be found in motion sensitive mouse interfaces [43] or teleoperation of humanoid robots [44].

Most studies on interfacing mechanisms for remote control of robotic morphologies are conducted using a fixed input device. Ellis et. al. have developed a haptic interface for robot teleoperation [45]. Chao Hu et.al. in [46] present a visual recognition method for mobile robot teleoperation using a camera for identifying human hand postures. Marin et. al. in [47] implement an interface using virtual reality techniques. They implement a multi-level architecture, where different interaction channels are available for the user to communicate their intentions for control. The channels vary from voice commands (top level) to remote programming (bottom level).

1.3.2 User Interfaces

In the context of human-machine interaction the notion of user interfacing arises. A user interface can be defined as the pathway of communication between two systems, namely the human operator, or user, and the machine, or robotic morphology. A user interface is found to potentially serve three roles; assist the correct and effective use of the system's capabilities, be proactive in the users problem solving and provide training [48]. The input data that the interface receives from the user are dictated by the input device used. Thus, having elaborated on the possible input devices, we now explore how we can support the communication with the user. The rules dictating the informational flow, the design and the architectures available.

The design of interfaces used and the interaction enabled by them are mostly judged on ergonomics [49]. In terms of HMI ergonomics relates to how the user will interact with a machine and how easy that interaction will be. The main goal of ergonomics can be stated as the design of equipment which is,

a) *Easy to remember*;

b) *Easy to learn*;

c) *Efficient to use*;

d) *Effective to use*;

e) *Enjoyable to use*; and

f) *Safe to use*

for the user.

Possible architectures governing such a communication are described in User Interface Management Systems (UIMS). UIMS are high-level interactive software applications, used to assist the computer-based system to communicate with the user, but also provide a separation of functionality between the application and the user interface components of the system [50]. The logical components of a UIMS are described by Green [51], as

a) the *presentation* of the interface to the user;

b) the mechanism for *dialog control*, this being the component bringing together interface and underlying system; and

c) the *application interface model* providing the semantics of the functionality of the underlying system;

, for the user.

Based on the work of Rasmussen [52] the levels of human-computer, a subclass of human-machine, interaction are abstracted in five levels from a physical to purpose basis. In their work Rasmussen and Vicente coined the term Ecological Interface Design (EID) [53]. EID focuses in the work domain of the interface (i.e. ecology) as they say, it facilitates and bounds purposeful patterns of activity, irrespective of human or automated agents performing the activities. EID is mostly concerned with the graphical aspects of interfaces, the way that information is show to the user to assist the interaction. Applications of EID can be found in process control [54], health care [55], command and control [56], and aviation [57]. Whereas more classical approaches to systems and interface design seek to optimize behaviour and look for particular ways of performing a task, EID tends to focus on 'control structure' instead of 'control behaviour' . This implies that operators are free to choose any strategy they prefer, as long as it does not violate work domain constraints. Something that aligns with the idea of controlling robotic morphologies, since it enables the user to bypass the optimal way of control if needed to.

The principle reflecting EID's approach to ecology is Ashby's Law of Requisite Variety [58], in the sense that in order for a representation to be effective for communication and control, it must be as complex as the problem solved. Although that seems similar to the SSSI (singe sensor single indicator) philosophy, this can be seen as making the problem more complicated than it needs to be [59].

Intelligent User Interfaces

The merge of artificial intelligence and human-computer interaction brings forward the idea of Intelligent Interfaces [60]. In their studies on intelligent user interfaces, Hefley et. al [61], provide a working definition, as systems *that build on facts and heuris-*

tic knowledge of an expert, together with techniques for reasoning about unstructured situations. In their research they use user interface management systems (UIMS) concepts as a basis for their research on intelligent interfaces. They distinguish between adaptive and flexible intelligent interfaces, with the first having the added capability to learn over time from experience to accommodate the user and their interaction. On the other hand, flexible interfaces deal with cases where the user can tailor the interface or the interface supports several styles of interaction.

1.3.3 Intuition in Human-Machine Interfaces

In this section we describe the main ideas guiding the interaction of the operator with the interface. An exemplar approach is described in Niwa et al. [2]. In their approach they define the interaction between the user and the interface as an 'intention translation' mechanism, by which user intentions are translated to instructions or commands that the interface can understand, so that the user can interact with it. In most interfaces users have to familiarise themselves with the interface in order to interact with it, read the user manual and understand the predefined mechanisms of interaction [62]. In a more complex interaction paradigm, where the actions to be performed are formed using simpler actions as building blocks, the user has to learn sequences of controls in order to 'communicate' their intentions to the interface. In such case, as the number of sequences, and so, the building blocks increase, the more laborious it becomes for the user to remember and execute them.

Providing a mapping between user intentions and robot behaviours can lead to an intuitive interface. The operator's intentions are taken into account through manipulations of the input device. In this reversed paradigm, users do not have to familiarise themselves with the interface (as is the usual case), but rather the interface can learn from the interaction with the user. Based on the reactions of the user to the exhibited behaviours of the robot, the interface is able to correlate the two, forming a control pattern. For that to happen, a level of consistency is expected from the users in the behaviours they exhibit. Same or similar input signals should be expected to yield the same robot behaviour as a response. Studies carried out, on a similar approach show up to 80% percent mapping accuracy in the interaction with a 17 degree of freedom robot, using an input device consisting of two joysticks [2].

1.3.4 Human Robot Interaction

Closely related to the field of Human-Machine Interaction, with the difference of the machine being more active in its environment is the field of Human-Robot Interaction [63]. Here, a split can be made between robots that have a level of autonomy and robots that don't. In the latter case it is easier to see the connection with the ideas of Human-Machine Interaction, as the robot is rather perceived as a mechanical structure that needs be operated. In the first case, the robot is a dynamic system that is able to perform in its environment and act in it in a purposeful manner. As such, the interaction paradigm shifts from the operational view of the machine to a more collaborative one where the robot's attention and desires are taken into account together with those of the human.

Autonomy In the case of robots with no level of autonomy usually the control takes place from the direct manipulation of the degrees of freedom (DOF) of the morphology. In the case of remote control, the input device needs to have at least the same amount of DOF so that the operator can achieve full functionality of the robotic morphology. Examples of direct manipulation are applied in humanoid robots, exploiting the resemblance of the operators and robots morphologies. This way the operators whole body [64], or parts of the it [65], are captured and mapped to the robotic ones, enabling a direct manipulation of the morphology. In most cases Kinect sensors are used, being easier to deploy and providing robust motion recognition out of the box. There are also cases where direct remote control is deployed using exoskeleton structures to capture the dynamics of the operators movements, having again humanoid robots as target.

In the case of robots with a level of autonomy, even if this only entails an active perception of the environment, the interaction is allowed a higher more human like level. Here, people can discuss with the robot or draw its attention to different elements of the environment [66]. Having a higher level of autonomy the robot can follow high level instructions which it can execute on its own demand. In such cases the robot can be refferd to as intelligent, but remains distant to a Turin-like intelligence, and many definitions are given in the literature [67, 68, 69]. Being able to follow directions, even high level ones (i.e. voice recognition, gesture recognition, attention), is distant from a machine that can form its own intentions and goals. It because of that lack of universality in its 'inteligence' that most research is residing in using predefined preprogrammed sets of behaviours that the robot 'knows' how to execute at the humans command. An important aspect of the robot's autonomy in HRI with mobile robots is neglect tolerance, which refers to the amount of time the robot can operate without human intervention [70].

Information Exchange The amount of information required to be exchanged between the human and the robot is yet another shaping characteristic of HRI. There exist measures for the interaction time required to perform a task [70], the cognitive load of the human -the amount of information one has to process for the interaction- [71], the situation awareness of the robot [72], as well as the shared understanding between human and robot [73, 74]. These measures come in accordance and work together with aforementioned ones, as for example high neglect tolerance allows for a shorter interaction time, potentially with a lower cognitive strain on the human.

Adaptability Finally one of the seminal aspects of HRI rises from the fact that for two complex systems to interact one or both should be adapted towards the other. Here three ways have been explored so far, having the human trained on the interaction's paradigm, the robot, or both.

Training of the operator is usually done through training manuals, instructions from a researcher, or istructions from the robot [75, 76, 77]. Such methods are usual in cases where the robot is designed to deal with a wide variety of human operators. The most straight forward way in tackling the differences of all operators is to train them in the particulars of the robot and interaction in hand. This includes the support of humans in the house, robot museum guides, and the general field of entertainment and educational robots.

The alternative is to train the robot for the interaction's paradigm and particulars. This can be done off-line as part of the design process [78, 79] or on-line through the interactions with the human [80, 81]. Through such training the robots perceptual capabilities can be improved [82, 80, 83], its autonomous capabilities [84], as well as its reasoning and planning capabilities [85]. Of great interest is the research enabling the robot to be trained for the interaction procedure with few training examples, making it applicable in real life scenarios and not highly time consuming [86].

1.4 Synopsis

Through the theoretical perspectives, existing works in the field of robot behaviour generation as well as human machine and robot interaction, key concepts of the literature have been introduced. Our goal being the entanglement of robot and human behaviours for the guidance of the robot, brings forward the idea of control. That is, the control of the robot according to the human's needs and desires. Even more so, the intuitive control of the robot, in that the human is enabled a training free interaction paradigm, rather than restricted to a pre-designed one. The seminal question answered in this thesis can be written as,

'How can a human intuitively regulate the state of a robot in its environment?'

Accepting the time dependency of behaviours in our problem, our *state* represents a depiction of our 'everything' within a time unit. And as we are interested in the state of the robot these observed variables shrink to the ones describing the robot. Here rises the need for a dynamical systems approach to robot behaviours, as they can deal with time, as they allow for a continuous approximation of the modelled system to be made.

As an embodied agent in the world the *robot*, through its sensing apparatus can make observations of the environment, of what surrounds it. In view of the previous paragraph we see that our state now shrinks to our robots sensory recordings at a given time unit. Here serves the ideas of enacted cognition and that of sensorimotor contingencies, as they place the robot in a seminal -central- position, enabling it with an understanding of its own body as well as its environment. And since the robot can act in its environment, its actions should also be part of its *state* as to be able to relate to them, their cause, and consequences. This following the idea of Access Consciousness and Sensorimotor Theory of Regan. Following these ideas it becomes clear the seminal role of the dynamics of the sensorimotor interactions between an embodiment and its environment. It is only through this 'loops' of continuous interaction that one becomes aware of the body, environment, and acting possibilities. In the understanding of the body we draw upon the ideas of Homeokinesis and the work of Ralph Der, while for the environment on the ideas of SMC and the work of Regan, finally for the possibilities -latent in the environment- on the ideas of Affordances and the works of Gibson and Norman.

The external source of commands to the robot, is the *human* operator. To control the robot one must be aware of the specifications of the robot and of its capabilities. And that in turn necessitates that the action possibilities of the robot, in its respective environment, need be exposed to the human in control. If not, an understanding of the robot's abilities within the environment cannot be formed by the human and so no control desires- thoughts- not patterns will be formed. While, to capture these control desires a sensory apparatus needs be found, strange or not to the robot. An external input device or a sensor -part of the robot- that can provide it with a notion of the human's actions. For this interaction to become intuitive, there are two possibilities. One possibility is for a mechanism and physical device to be found that are self descriptive and trigger consistent responses for their use to all humans. The other would be to exploit any given affordances of the device and 'learn' the human's intuitive control of it. Using any device would mean to enable the human operator to explore its affordances and learn how this device can relate to the robot's behaviours for each human operator. Establishing an adaptive and flexible interface between the input device and the robot that learns from the manipulations of the human in co-occurrence with behaviours from the robot. We follow the latter approach.

Ultimately, we pursue a solution to the problem by coupling the dynamics of human and robot. Sensorimotor dynamics that are formed through the actions of the embodiment in the environment. We pursue their regulation by means of external input, guided by the human mediated through a sensory system -input device. This, under the regime of adaptive systems that can be moulded to the particulars of the human and the robot.

Chapter 2

Methodological Background

Artificial Neural Architectures - from Autonomy to Guidance to Coupling

2.1 Introduction

Based on matters discussed in the introduction, the particulars of the ideas, principles and implementations are put forward. In this section, we elaborate on the methodologies used in our experiments and provide their mathematical background.

Our goal is to implement a system -framework- capable of supporting the intuitive control of robotic morphologies. Being able to support both different types of input devices and robotic morphologies, we need recognise high level aspects that guide and regulate their operation, through which we can establish a meaningful communication that enables the remote control of the robot, while satisfying the criteria set in the respective fields of robot control, human machine interaction and user interfaces.

Two are the main concerns of the framework, and two the main modules that need to cooperate. On one hand, the module that captures and communicates the human control behaviours and on the other, the one that explores, extracts and stores the robotic behaviours. Since our intention is for a system that can work independently of the robot and the input device present, the methods need to remain agnostic to the particulars of the mechanics. In doing so, providing the proposed methods should provide the system with both adaptability and flexibility.

The intuition acquisition mechanism suggested in the literature 1.3.3, provides us with a well defined starting point and supporting evidence for a way of capturing human intentions for control, that is human behaviours. Based on the observation of the previous chapter we recognise that behaviours are dynamic processes and as such time is essential for their characterisation. Thus, human behaviours can be captured as time varying configurations of the input device. That is, the configuration of degrees of freedom of the input device, according to time.

Taking into account the input devices presented in section 1.3.1, we observe that regardless of their particular characteristics, it is possible to acquire the captured input as time sequences. With this observation in mind we go forward and formulate the problem of acquisition of intentions from the operator as a problem of time sequence recognition. In doing so we can provide a user centred and adaptive method for the remote control, but the interaction would still need to be fragmented. Indeed, with this observation in mind, we put forward the idea of mapping rather that recognition (classification) and consequently work to provide a dynamic coupling of the human and robot behaviours. A case in which operator and operated co-exist and co-adapt to achieve on-line real time communication for control, a coupling.

User intentions for control are realised as time depended manipulations of the input device. Interfacing segmented manipulations of the input device, doesn't differ much from the idea of having a keyboard, with the keys replaced by time sequences. Being able to provide a way for real time recognition of the operators behaviours on the input device is a challenging problem, still not solved in the field of human machine interaction. In our approach not only we try for real time recognition, but we move a step further, trying to find a way of recognising combinations of behaviours exhibited by the user. Based on the idea of building blocks of behaviours we try for a system that can recognise entangled behaviours. Doing so necessitates that behavioural building blocks are also present on the robot and that they can sustain combinations retaining their behavioural 'meaning' -exhibit stable and smooth transitions, and meaningful combinations.

On the other hand of the interface towards the robot, our vision is for a system without a single target morphology. In doing so, we can only assume the acting and sensing apparatus of the morphology given, and thus work only with proprioceptive information. Under this restriction and in light of the theoretical foundations given in previous section we work in the direction of self-organisation of robotic behaviours. Developing a method able to explore the situated and enacted possibilities of a morphology is challenging, even more so to segment such behaviours in blocks that are reusable and also combinable.

Exploring possible robotic behaviours follows the idea of enacted cognition and allows for meaningful interactions of the robot within its environment to be discovered. In doing so, the robot is able to form behaviours depended on its morphology, its dynamics, and seminally its surrounding environment.

Ultimately on a second level, once the organisation of the sensorimotor loop is realised, a way is needed for the extraction of individual behaviours. First, the behaviours should be solely based in the self-organisation procedure underneath. Second, the segmentation to blocks should be done in an autonomous -automated- way, based on internal properties of the behaviours. Finally, such behaviours should not be overlapping as to eliminate redundancy and also to make their combinations useful. This in the sense that having the two very similar combined will not result to a new behaviour rather to a third one same as the other.

In what follows the methods are described and formalised. Starting from simple ones, we move on to the ones satisfying all the aforementioned criteria. This way we build towards the complexity of the system an element at a time. First, methods for the human side of the system are provided and, second, methods for the robot side.

2.2 Interfacing the Human Operator

2.2.1 DTW, Dynamic Time Warping

The Dynamic Time Warping (DTW) is a distance measure used mainly in speech recognition community. It allows a non-linear mapping of one signal on another by minimizing the distance of the two. The DTW algorithm calculates the distance between each possible pair of points out of two signals in terms of their associated feature values. It uses these distances to calculate a cumulative distance matrix and finds the least expensive path through this matrix. This path represents the ideal warp - the synchronisation of the two signals which causes the feature distance between their synchronised points to be minimised [87].

Calculating DTW Assuming two time series, Q of length n and S on length m ,

$$Q = q_1, q_2, ..., q_n \tag{2.1}$$

$$S = s_1, s_2, ..., s_m \tag{2.2}$$

To align the two series we construct an n-by-m matrix where the element in place (i, j) corresponds to the squared distance $d(q_i, s_j) = (q_i - s_j)^2$. To find the the best match between the two sequences, we find the path through the matrix that minimizes the total cumulative distance. Starting from the bottom left corner of the matrix and finishing at bottom right, the optimal path is,

$$DTW(Q,S) = min\{\sqrt{\sum_{k=1}^{K} w_k}\} \tag{2.3}$$

where w_k is the matrix element $(i, j)_k$ that also belongs to the k^{th} element of the wrapping path W. The wrapping path can also be found using dynamic programming with the following recurrence.

$$\gamma(i,j) = d(q_i,s_j) + min\{\gamma(i-1,j), \gamma(i,j-1), \gamma(i-1,j-1)\} \tag{2.4}$$

Multi-Dimensional DTW In the case of multi-dimensional time signals, a proposed approach is given by the MD-DTW, the multi-dimensional dynamic time warping [88]. The algorithm utilizes all the dimension to find the best synchronisation between the time series. Assuming that we have the same time series Q and S as above, but this time they are of dimension L, the MD-DTW follows the exact procedure as the single dimension DTW, but the distance matrix D is filled according to:

$$D(i,j) = \sum_{l=1}^{L} |Q(i,l) - S(j,l)| \tag{2.5}$$

A possible improvement for both DTW and MD-DTW is to calculate the measure on the first order derivatives of the features, synchronising this way the shapes (peaks and slopes) of the two signals. The algorithm for the calculations of the DDTW (Derivative Dynamic Time Warping) is the same as the one of DTW, with an extra step before the matrix calculations, to calculate the derivatives of the time sequence. An approximation of the derivative can be calculated, for a time sequence a at time t, as follows:

$$der(a(t)) = (a(t+1) - a(t-1))/2 \tag{2.6}$$

This method was applied for two-dimensional time sequences recognition and classification from a touchscreen device. The task was to control the behaviours of an E-puck robot, as part of the British Science Week (BSW) exhibition in Plymouth University.

2.2.2 Neural Networks

Although DTW has been applied and extended, and even though there are techniques of overcoming the necessary fragmentation of the input sequence, in this section we describe the potential Neural Networks (NN) have in the recognition of time sequences.

Introduction to Neural Networks

Neural networks (NN) can be seen as general function approximators. Depending on the layers they implement, they can be separated in linear and non-linear ones. At the same time, a further distinction can be made by the type of connections in the network. Having signals propagating only to next layer, defines a feed-forward network, whereas having signals traversing the network in many directions, defines a recurrent neural network.

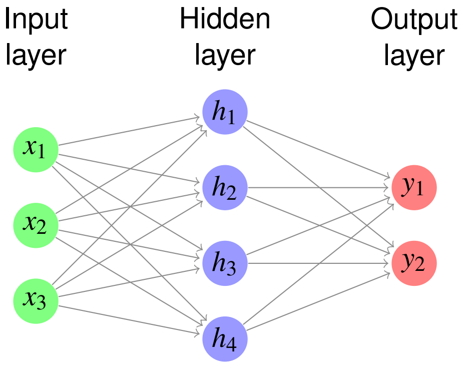


Figure 2.1: A 2 layer feed-forward Artificial Neural Network.

Another distinguishing factor is the learning process followed for updating the weights in the NN. In the case of supervised learning, one of most commonly found techniques is gradient descent. Supervised learning suggests that during the training both the input and the desired output are available. Giving the input to the network, propagating the signal and comparing the network output with the desired one, we get the error. Based on the error signal, the error gradient is calculated and the weights between the neurons are updated trying to minimize the error. The error is propagated to the neurons of each layer, assigning a 'blame' to each one for the error on the output. In 2 layer networks, delta rule is used for the adaptation of the weights.In the case of 3 or more layer networks the delta rule combined with the chain rule, gave rise to back-propagation of error algorithm, as defined first by Rumelhart, Hinton and Williams [89].

Finally, important to mention is the activation function of the neurons of the NN, which converts a neurons weighted input to its output activation. The selection of the activation function depends on the task performed by the network. It is here that the non-linearity of the network is formed and based on the activation function's range the potential approximation capabilities of the network.

A commonly used activation function is the sigmoid, calculated as

$$S(t) = \frac{1}{1+e^{-t}} \tag{2.7}$$

The use of the sigmoid function is based on the easily calculated first order derivative, given by

$$\frac{d}{dt}S(t) = S(t)(1-S(t)) \tag{2.8}$$

Another commonly used activation function is the hyperbolic tangent $tanh(x)$,

$$tanh(z) = \frac{sinh(z)}{cosh(z)} = \frac{\exp(2z)-1}{\exp(2z)-1} \tag{2.9}$$

with its derivative being,

$$\frac{d}{dz}tanh(z) = 1 - tanh^2(z) \tag{2.10}$$

Training of Neural Networks

Back Propagation of Errors Assuming the network in figure 2.2, and given an input vector x, a target output t and the output of the network y, the squared error function is

$$E = \frac{1}{2}(t-y)^2 \tag{2.11}$$

for each neuron n output of the neuron o_n is

$$o_n = f(in_n) = f(\sum_{k=1}^{m} w_{kj}x_k) \tag{2.12}$$

where the input in_n is the weighted sum of the neurons connected to it from the previous layer, m being the number of neurons connected to the neuron n, and w_{kj} is the weights of the connections. In the case where the neuron o is part of the 1^{st} layer the input in_n comes the input signal x_n. The function f is the activation function of the neuron and usually is non-linear and differentiable.

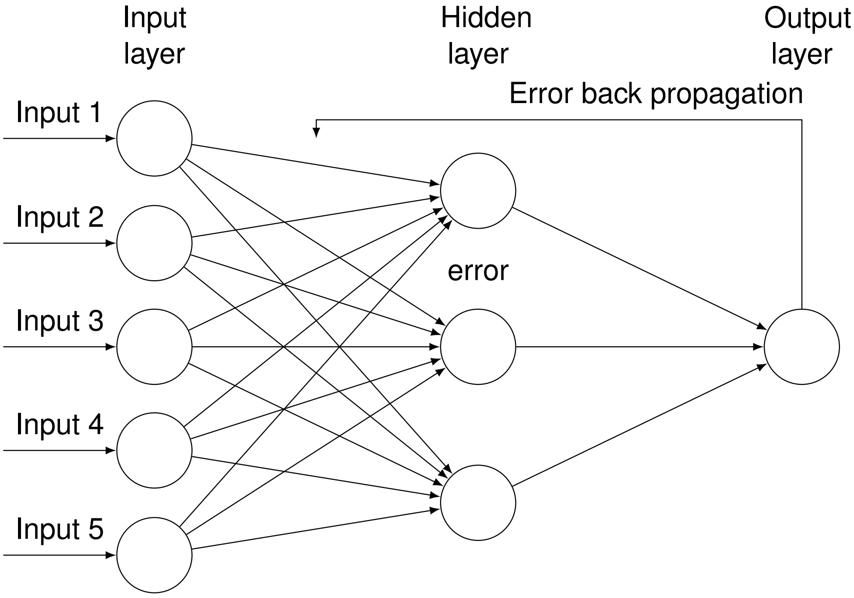


Figure 2.2: Schematic representation of the Backpropagation of the error in a 3 layer feed-forward Artificial Neural Network.

To adjust the weights one must create the error gradient. That is done by calculating the partial derivatives of the error with respect to a weight w_{ij}. In the case of multilayer networks and the backpropagation algorithm the chain rules is used

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial in_j}\frac{\partial in_j}{\partial w_{ij}} \tag{2.13}$$

in the general case the partial derivative of the error of neuron is calculated according to,

$$\frac{\partial E}{\partial w_{ij}} = \delta_j x_i \tag{2.14}$$

with

$$\delta_j = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial in_j} = \begin{cases} (o_j - t_j)f(in_j)(1 - f(in_j)) & \text{if } j \text{ is in the output layer} \\ (\sum_{l \in L} \delta_l w_{jl})f(in_j)(1 - f(in_j)) & \text{if } j \text{ is in any other layer} \end{cases} \tag{2.15}$$

with o being the output of the neuron, in being the input, L being the set of neurons receiving input from the neuron j. This way the update of the weights is done by gradient descent, using a learning rate ε according to the rule,

$$\Delta w_{ij} = -\varepsilon \frac{\partial E}{\partial w_{ij}} \tag{2.16}$$

Back Propagation Through Time BPTT is an extension of the normal BP algorithm described above, for the computation of the exact error gradient in recurrent neural architectures [90]. The idea behind this method is that the network errors and gradients are calculated in the span of time that the desired sequential behaviour is exhibited. Assuming behaviours take place from time t_0 to time t, we can unroll the recurrent network, as many times as our time steps, resulting into a Feed Forward network. Thus, the central idea behind BPTT is that to compute the $\partial J^{total}(t',t)/\partial w_{ij}$, the overall gradient of the network, one can simply calculate the partial derivatives with respect to each time step's weights in the Feed Forward network corresponding to w_{ij} and add them up.

The idea here is that the overall 'blame' or error of the model forms as a result of the smaller errors taking place in each time step of the produced signal. This way, the error can be better attributed once the recurrent network is transformed to a feed-forward equivalent, or 'unrolled ' in time. This way changes in weights are performed based on the whole of the outgoing signal and not only in a single time instance of it.

Given that $y_t = tanh(W_{in}x_t + Wh_{t-1})$ for the output $\hat{y_t}$ at time t the error E propagation for a signal with length T is calculated as,

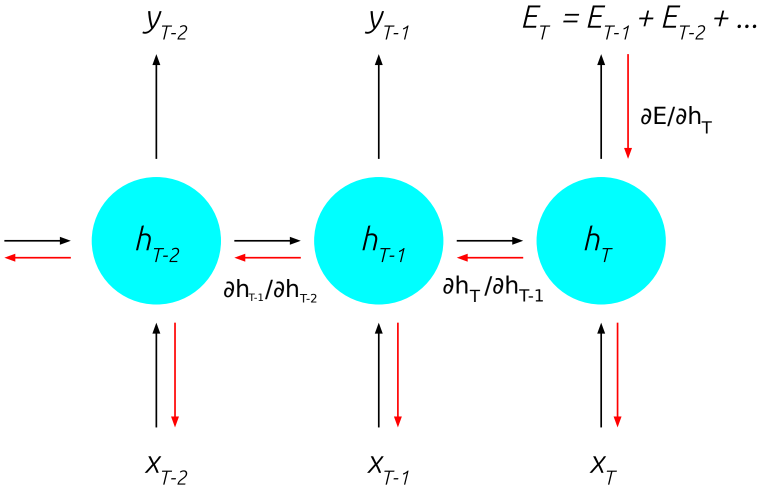


Figure 2.3: Schematic representation of Back Propagation Through Time.

$$\frac{\partial E}{\partial W} = \sum_{k=0}^{T} \frac{\partial E}{\partial \hat{y_T}} \frac{\partial \hat{y_T}}{\partial h_T} \frac{\partial h_T}{\partial h_k} \frac{\partial h_k}{\partial W} \tag{2.17}$$

with W being the weight matrix of the network, and h is the activation of the hidden layer at the indicated time step t.

From the above chain rule it is easy to see that the longer the time steps T for the backpropagation and the smaller the values in the layers, the gradient if forced to shrink exponentially fast [91]. Indeed, this problem of the BPTT algorithm makes it difficult for long range dependencies to be trained.

Feed Forward Neural Architectures

Convolutional Neural Networks Convolutional Neural Networks (CNN), are Feed-Forward Neural Networks comprised of one or more convolutional layers, usually followed by a fully connected network [92]. CNNs have local connections and shared weights, between neurons, in each layer making them more efficient to train than fully connected networks.

CNNs are assembled by a series of convolution layers, intercepted by sub-sampling

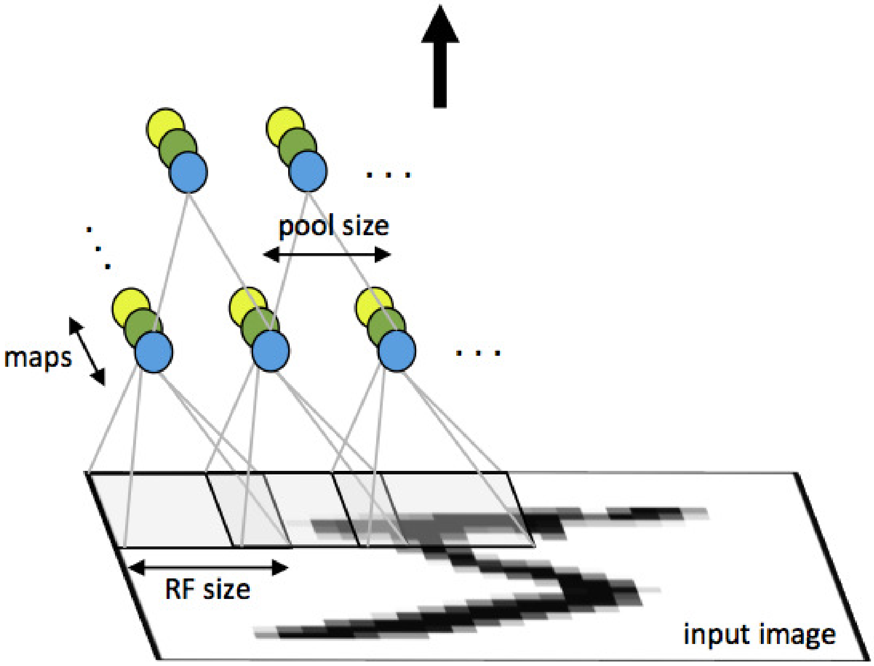


Figure 2.4: A schematic representation of a Convolution Layer

ones (e.g max-pooling) with a final linear classifier at the end (i.e. a fully connected single layer neural network) . Working with 2D convolutions (i.e. spatial convolutions), a typical setup entails the following features. Each convolution layer takes as input an image of $m \times n \times r$, where m in the height, n the width and r the number of channels. Convolution layers have k filters of size $n \times n \times q$, with n being usually smaller than m, $n << m$. The number of channels, q, may vary for each kernel. Each filter is then convolved with the image producing k feature maps of size $m - n + 1$. This operation gives rise to the locally connected structure of the layer. Each map is then sub-sampled with either mean or max pooling. This operation gives rise to the spatial invariance of the features recognised by CNNs. This, as features are aggregated from ever larger areas as they proceed to higher layer of network. The final layer usually consists of a fully connected linear layer that combines the highest level activations of the feature maps. An illustration of the architecture described can be seen in figure 2.4.

For the training of CNNs the backpropagation algorithm is used, to train both the top, fully connected network, and to propagate the errors to the convolution layers bellow.

This way the filters of each convolution layer can be adapted to minimise the set cost function. Thus, in a single architecture it is possible to train both the filters used over the image and their combinations for the classification. As noted in BPTT algorithm, 2.2.2, propagating an error through many layers makes the gradient shrink exponentially fast. This is why simpler activation functions have been used in Deep Convolution Neural Networks. At the same time usage of max pooling layers allows for the error gradient to be directed to single neurons rather groups of them, helping the gradient to vanish slower.

Time Aware Neural Architectures

Recurrent Neural Networks Dynamic temporal patterns can be acquired with the use of Recurrent Neural Networks. With the term dynamic temporal patterns we refer to signals that vary through time. There are many implementation paradigms, but our inspiration comes from the work of Jun Tani, in both time sequence recognitions [93] and multiple time scales dynamics acquisition [94]. In his work the recurrent network has a Jordan type structure [95] and it is trained with Backpropagation Through Time (BPTT) algorithm. In our implementation we are based on Elman [96] type structure, keeping recurrent the hidden layer.

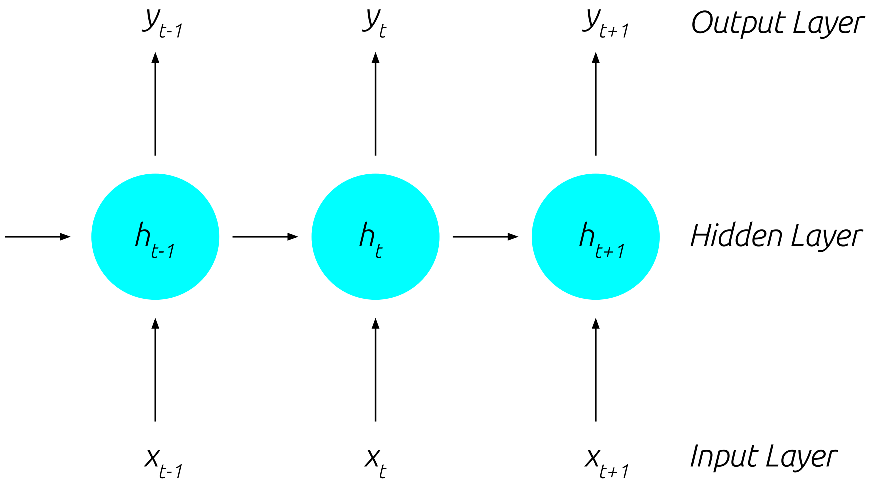


Figure 2.5: Schematic representation of an Elman type Recurrent Neural Network. Here the recurrency takes place in the hidden layer.

One can distinguish between different types of recurrent neural networks, by the way the neurons are connected. Fully recurrent neural networks structures have every neuron in each layer is connected to all the neurons in that layer and the neurons in the next layer. In the case of Jordan type architectures, the output of the network at time t is passed to the network together with input at time $t+1$. In the case of Elman type architectures the networks hidden layer at time t in passed to the network together with the input at time $t+1$.

Another important recurrent network architecture is found in Echo State Networks (ESN). Echo state networks (ESN) provide an architecture and supervised learning principle for recurrent neural networks (RNNs). The idea is to drive the large, randomly configured recurrent network with an input signal and acquire the output based on linear combinations of the reservoirs units. The structured randomness of the reservoir serves for both the

Recurrent Neural Network with Parametric Bias The idea of Parametric Biases (PB), as implemented by Jun Tani, provides a way for both generation and recognition of dynamic temporal patterns. PBs are units in the input layer of the network, adjusting themselves, according to networks dynamics. During the training phase and after the error has been propagated to the weights of the networks, the values of the PB units are adjusted, trying to further minimise the difference between the target and actual output. The update equations for the ith PB unit at time t are,

$$\delta\rho_t^i = k_{bp} \sum_{step=t-l/2}^{t+l/2} \delta_{step}^{bp^i} + k_{nb}(\rho_{t+1}^i - 2\rho_t^i + \rho_{t-1}^i) \tag{2.18}$$

$$\Delta\rho_t^i = \varepsilon\delta\rho_t^i + \eta\Delta\rho_{t-1} \tag{2.19}$$

$$\rho_t^i = f(\rho_t) \tag{2.20}$$

the term $\delta\rho_t$, the delta component of the internal value of the PB unit is calculated by the summation of two terms. The fist one represents the summation of the delta error propagated from the output units for l time steps, while the second term works as a low pass filter, inhibiting rapid changes in the PB unit, with k_{nb} being the coefficient for the filter. The update $\Delta\rho_t^i$ utilised a momentum term, the second term of the summation, for faster convergence. Finally, the activation of the PB unit is given through function f, a sigmoid function.

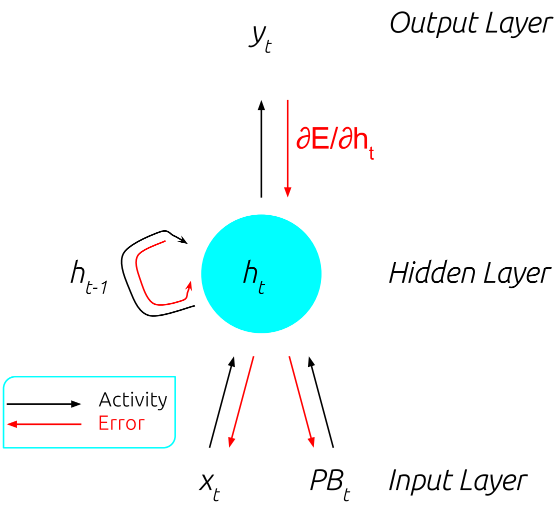


Figure 2.6: Schematic representation of a Recurrent Neural Network with Parametric Bias (RNNPB).

This updating step of the PB unit values is then performed, in real-time, once the network is trained, resulting to the recognition of the sequence fed on the network. Assuming the network N, that means that the temporal sequence $S(t)$ is generated as $S(t+1) = N(S(t))$. Once the network is trained, the input received at time $t+1$ represents the desired output of the network having as input the one of time t. Once the networks is trained the PB units are initialised with value zero, and a sequence is fed element-by-element in the network.The difference of the network output o and the desired output $S(t+1)$ in time $t+1$ can be calculated, given the known input from time t. Since the network is trained, the only possible source of error is identified in the PB units. So if an error appears, that error value is propagated to the PB units, updating them according to the same set of equations as in the training phase. That means that the network readjust the PB units values, to match the target output as best as possible. The resulting PB values, give us the ability to identify whether an unknown sequence is a sequence the network was trained on, since the PB values of the training phase are known.

This idea together with the formulation of multiple time scales RNN with PB units, as described in [94], allows us to train the network in time sequences captured by the input device of the interface. Having multiple time scales suits the idea of entangled behaviours captured by the input device, being decomposed according to their different temporal profiles. Initial experiments with this set up have shown good results in the recognition of the time sequence captured by a touch-screen interface. That is, the network, as show in the next chapter, can recognise, predict and classify the time series it is trained on, and generalise for the unknown ones. An extra dataset, captured from the manipulations of a Leap Motion device is also used for the testing of this architecture. Similar results are obtained from this dataset as well, with the results being explained in the next chapter.

Reservoir Based Architectures Echo State Networks (ESN) provide an architecture for efficient training of RNN in a supervised manner. One can distinguish two main components in an ESN. Firstly, he Dynamic Reservoir (DR), a large, random, recurrent neural network with fixed weights. The DR gets activated by the input and provides a non linear response for this input. And the output signal, which is trained as a linear combination of the activations of the DR. This way the computational resources and complexity required for the training RNNs is reduced to the adaptation of the output connections of the ESN.

Assume we have a ESN consisting of N reservoir units, K inputs and L outputs. First, we need to find the state, x, of the reservoir and based on the state and the input u, we can compute the output signal y. The state extended by the input, on which we base the computation of the output, will be referred to as the extended system state on the network, z. The extended system state, depending on the particulars of the implementation can also include the output of the reservoir, i.e. the output connections of the

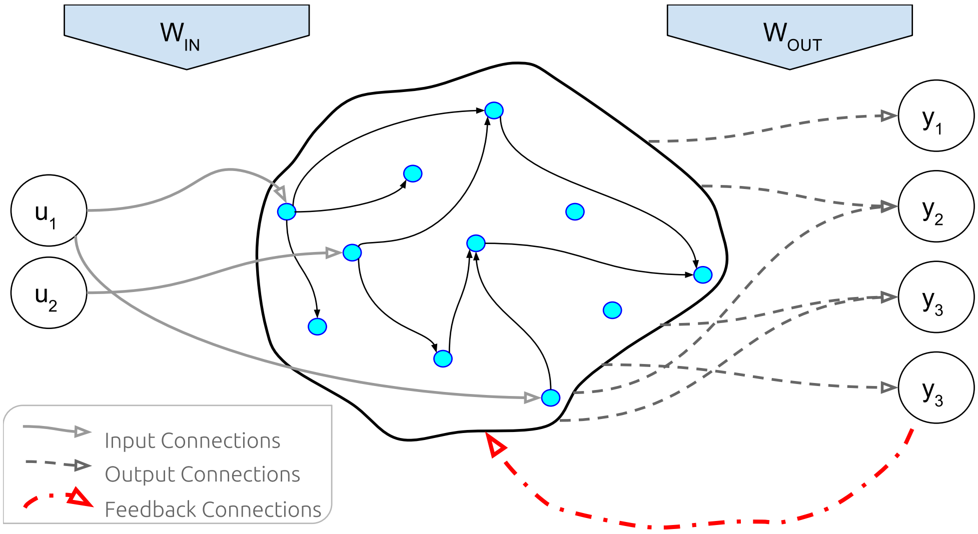


Figure 2.7: Schematic representation of an Echo State Network (ESN).

reservoir are recurrent.

So, the state update equation, for an ESN -without any recurrent output neurons- is,

$$\mathbf{x}(n+1) = f(\mathbf{W}\mathbf{x}(n) + \mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}^{fb}\mathbf{y}(n)) \; , \qquad (2.21)$$

where $\mathbf{x}(n)$ is the N-dimensional reservoir state, f is a sigmoid function (usually the logistic sigmoid or the tanh function), $\mathbf{W}$ is the $N \times N$ reservoir weight matrix, $\mathbf{W}^{in}$ is the $N \times K$ input weight matrix, $\mathbf{u}(n)$ is the Kdimensional input signal, $\mathbf{W}^{fb}$ is the $N \times L$ output feedback matrix, and $\mathbf{y}(n)$ is the L-dimensional output signal. In tasks where no output feedback is required, $\mathbf{W}^{fb}$ is nulled.

The extended system state $\mathbf{z}(n) = [\mathbf{x}(n); \mathbf{u}(n)]$ at time n is the concatenation of the reservoir and input states - and output in the case of output recurrency-.

The output is obtained from the extended system state by

$$\mathbf{y}(n) = g(\mathbf{W}^{out}\mathbf{z}(n)) \ , \tag{2.22}$$

where g is an output activation function (typically the identity or a sigmoid) and $\mathbf{W}^{out}$ is a $L \times (K+N)$-dimensional matrix of output weights.

For an ESN to function properly, the *echo state property* (ESP) is essential. ESP states that the dynamics of the DR will asymptotically washout, any information added by the input or feedback, from the initial conditions. It has been observed, that this can be achieved by scaling the *spectral radious* of the DR weights W to be less than unity. The ESP is then found to hold for the DR [97].

For the training of ESNs, let us assume a driving signal $u(1),\ldots,u(n_{max})$ and the extended states it generates -once passed to the network- $z(1),\ldots,z(n_{max})$. We collect the states in matrix S of size $n_{max} \times (N+K)$ and the desired outputs $d(n)$ in a matrix D of size $n_{max} \times L$. Usually, before each collection, based on the properties of the network, we apply a washout period, allowing the network to settle to the input provided.
Now, the desired output weights W^{out} can be calculated as follows. First, the correlation matrix of the extended system states are calculated, $R = S'S$. Then the cross-correlation matrix of the extended states against the desired outputs d, $P = S'D$. Finally, for the calculation one can either calculate the Wiener- Hopf solution

$$W^{out} = (R^{-1}P)' \tag{2.23}$$

or by calculating the pseudoinverse of S, $S^{\dagger}$ and then updating the weights

$$W^{out} = (S^{\dagger}D)' \tag{2.24}$$

2.3 Interfacing the Robotic Morphology

The autonomous discovery of available behaviours of a given morphology is the other fundamental element for designing an interfacing system that aims to reduce design constraints and maximise usability. To this end, we implemented a system consisting of two modules. One used for the exploration and self-organisation of the sensorimotor loop of the robot to be controlled and one for the extraction, storage and reuse of the acquired robot's behaviours.

The module for the self-organisation of the sensorimotor loop follows a dynamical system approach [98]. The realization of the dynamics of the robot and of the world is done using a Controller (K) and World Model (W) cooperating for the effective exploration of the robots dynamics and an accurate prediction of world states, respectively, as discussed in [7]. A very important feature of the approach is that there is no need for an extrinsic motivation or imposing of a target behaviour. This, as the method explores the robot's potential based on the intrinsic motivation to minimize the Time Loop Error (TLE) of the system. Indeed, given that the TLE is based on errors of the sensorimotor loop, the dynamical system formed is self-referential and thus self regulated.

For the extraction of behaviours from the robots an assembly of neural networks is used, working as individual controllers for each robot behaviour. First, this second level of abstracting the behaviours allows for a modular design, in that any other procedure for creating robot behaviours can be used. At the same time, extra controllers can be added with pre-defined behaviours resulting from explicit training of the robot to a behaviour. Reusing such behaviours is possible by placing the neural controller in control of the motor values of the robot. At the same time, given their formation, linear combinations of individual controllers is possible. This way not only the trained behaviours, but additional ones can be exhibited by the robot resulting from the combinations of the initial repertoire.

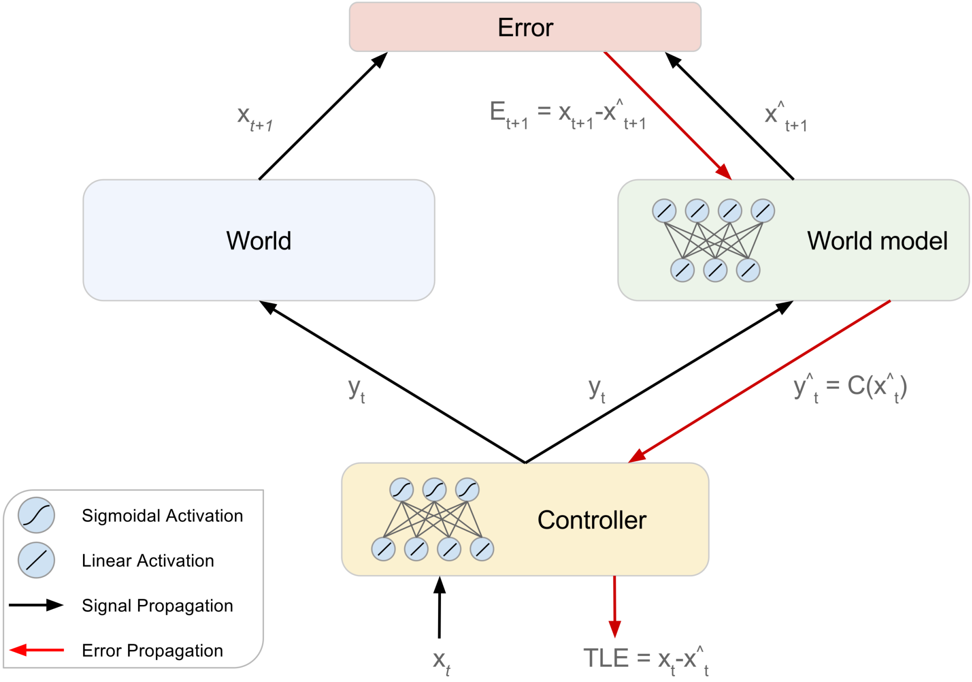


Figure 2.8: Schematic representation of the architecture for the self organisation of the sensorimotor loop of the robot.

2.3.1 Exploring the Robotic Morphology - Homeokinesis

The exploration module, in general, is described, according to time t, as:

$$\tilde{\mathbf{x}}_{t+1} = W(K(\mathbf{x}_t, C), A) \tag{2.25}$$

The controller K generates motor outputs $\mathbf{y}_t = K(\mathbf{x}_t, C)$ as a function of sensory inputs $\mathbf{x} = x_1, x_2, \dots, x_n$, in dependence on a set of parameters defined by the matrix $C_{n,n+1}$ and is defined by the equation:

$$K = g(\sum_{i=1}^{n} C_i x_i + C_{n+1}), \tag{2.26}$$

where g is a sigmoid function.

The world model $\tilde{\mathbf{x}}_{t+1} = W(\mathbf{y}_t, A)$ estimates future sensory inputs $\tilde{\mathbf{x}}_{t+1}$ from motor outputs $\mathbf{y}_t = y_1, y_2, \dots, y_n$ in dependence on a set of parameters defined by the matrix $A_{n,n+1}$. The parameter matrix of the world model, A, is adapted according to the Widrow - Hoff Learning Rule [99], delta rule, $\Delta w = +\eta E_W \mathbf{x}$ with the error, E_W, described by the function:

$$E_W = ||\mathbf{x}_{t+1} - \tilde{\mathbf{x}}_{t+1}||^2 \tag{2.27}$$

with learning rate η = 0.1.

The controller updates its parameter matrix by gradient descent with respect to the error function,

$$TLE = E_K = ||\mathbf{x}_t - \tilde{\mathbf{x}}_t||^2 \tag{2.28}$$

To calculate the above error (Time Loop Error - TLE), we find the $\tilde{\mathbf{x}}_t$ by calculating the motor input $\hat{\mathbf{y}}_t$ the world model should have in order to make a perfect prediction and then the sensory input the controller K should have to predict the motor output $\tilde{\mathbf{y}}_t$. The update on the controller parameter follows the rule $C_{t+1} = C_t - \varepsilon\frac{\partial E_K}{C}$, with a learning rate $\varepsilon = 0.01$.

2.3.2 Extracting Robotic Behaviours - Antagonistic Controllers

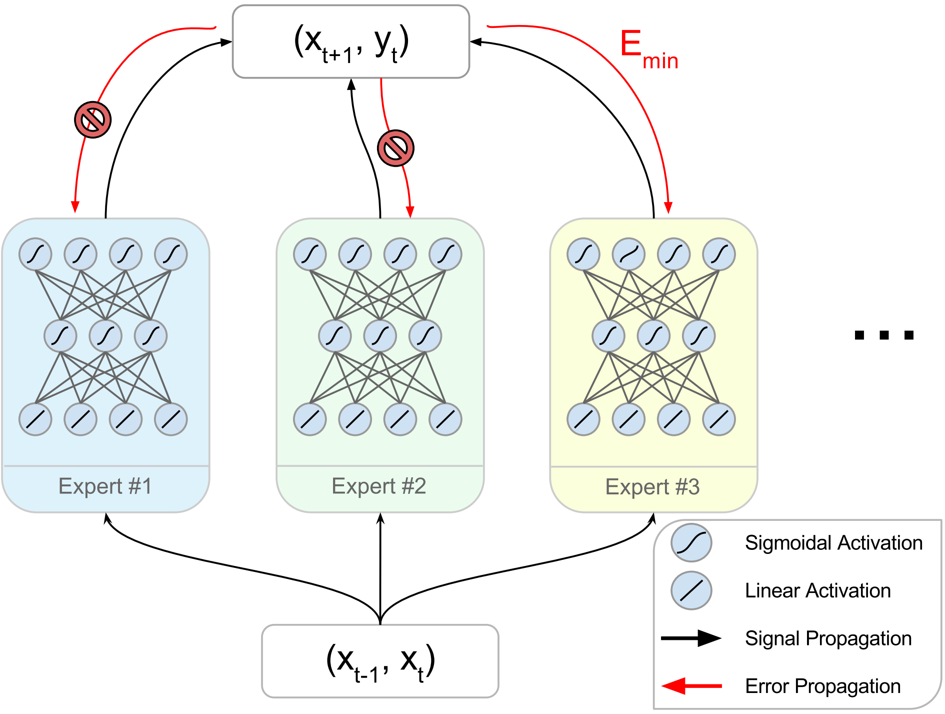


Figure 2.9: Schematic representation of the Neural Network architecture for behaviour extraction - Experts.

For the identification, storage and reuse of the different behaviours exhibited by the robot, we use a series of m neural networks. Each network is defined according to the equation,

$$(\mathbf{x}_{t+1}, \mathbf{y}_t) = N_i(\mathbf{x}_t, \mathbf{x}_{t-1}), \quad i = 1, \ldots, m \tag{2.29}$$

The neural networks, working in parallel, compete for the prediction of the motor command $\mathbf{y}_t$ of time t and the sensory input $\mathbf{x}_{t+1}$ of the next time step. It is a winner takes it all method, with only the winning network being allowed to train on the current data $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$. Because of that, each network specializes to a region of the sensorimotor space of the robot as discovered by the homeokinetic module.

The networks consist of 3 layers, input, output and a hidden layer. The hidden layer consists of sigmoid units whereas the input and output layers from linear units. No bias units are introduced in the networks.

The algorithm for the training of the networks is backpropagation, usually with learning rate $\eta = 0.01$. In each time step all the networks are activated with the same input and the one with the best approximation of the next sensor values and motor commands is selected as the winning network. The sample won is then added to the training dataset of the winning network and it is trained for another epoch. For the selection of network, a smoothed error is used, taking into account the past errors of the network.

Another possibility is using an online version for training. Updating the weights of the winner network in each step propagating only the current error E_t at time t. An advantage of this method is that the matrix multiplications for the weight updates are are simpler, in that the matrices are smaller. Thus this method becomes preferable in cases were the sampling rate for the sensor values of the robot is high and the computer system used offer limited parallelism on code execution (e.g. embedded systems) or limited memory.

In terms of efficiency, the distributed representations of the neural controllers offer a way of storing behaviours in the weights of the network, with very small memory requirements. At the same time, the distributed representations of neural networks are well suited for parallelisation. Multiple libraries and frameworks exist for efficient parallel algebraic calculations (i.e. Theano, TensorFlow, Keras, Cafe, DIGITS).

Triggering a robot behaviour -using a neural controller-, consists only of a forward pass of the sensory values of the robot through the network and applying the resulting motor commands.

Chapter 3

The Human Centric Approach to Robotic Control

Intuitive Control using Recurrent Neural Networks with Parametric Bias

3.1 Introduction

In this chapter we present a novel idea for the creation of an intelligent interface that allows the remote control of arbitrarily complex robotics morphologies by translating intuitive human behaviours into purposeful robotic actions. By taking inspiration from human robot interaction, ergonomic principles, and autonomous robotics this paper proposes a human-centric framework for robot control inspired by the current advancements in recurrent neural networks and self-organisation. In particular, we present an integrated approach based on neural networks for input acquisition from human operator and self-organisation for the acquisition of robot behaviours. We realise the interface as a kind of intelligent agent connecting the two end points of the system: Human and robot, providing an adaptive and intelligent interface for robot control. The present preliminary study shows the on-going results of the proposed methodology for both self-exploration of robotic morphologies and acquisition of human behaviours.

Human robot interaction and remote control have long been surging fields for both research communities and commercial markets. Whether robotic morphologies are built for entertainment purposes or for more “serious" applications, the remote control of such robots is one of the main forms of interaction between humans and robots.

Different types of robot morphologies have found applications in industry, rescue missions and military operations. In this context, we observe that the specific task often defines the constraints of the robot morphology and of the control mechanisms, and dictates the interfacing approach. Most systems, being specifically constructed for a given task, are designed with the restrictions already embedded into their operational mechanisms. This approach makes the robot usable and controllable by a human operator, but also drastically constrains its usage to a very specific context. To this end, a wide range of tele-robotic interfaces have been explored so far, some are very rigid devices that require a great deal of cognitive and manual effort, while other, more intuitive systems based on one-to-one body mapping, are in contrast very complex and expensive devices, often specifically tailored to a single robotic platform [1].

Tele-operation of complicated mechanical devices requires a great deal of knowledge about the interfacing mechanisms and the robotic morphology at hand, both from the operator and from the designer of the controlling interface, in order to make the interface tailored to the given robot and, often, to the operator (e.g. see the unique controllers designed to accommodate different types of motor disabilities). To obtain such knowledge the operator has to undergo a suitable training on the use of the interface. For the communication of a continuous control sequence, for example, in most of the approaches available so far, the operator has to pass a sequence of commands through a controlling device. This can be difficult to remember and prone to mistakes.

Starting from these observations, we aim to design an integrated methodology focused on the human and capable of seamlessly translating any type of human motion into meaningful robotics actions and behaviours, something that we can call, as the title suggest, a human-centric approach in designing tele-operation of robotic morphologies. The main concept is based on the principle that the interfacing controller should be capable to adaptively ‘understand’ and translate human motion into controlling commands for the robot, rather than having the human, the operator, learning the use of the interface. The proposed approach, therefore, relies more on the intelligence and learning capabilities of the controlling interface rather than on those of humans. We expect this should ease the cognitive demands in both designing and using the controlling device.

At the same time, we aim to extend the above idea to suit the control of arbitrary robotic morphologies. By exploiting techniques of self-organisation of robot behaviours, we are able to extract a behavioural repertoire of the robot to be controlled, without knowing in advance the kinematics and dynamics that characterise the morphology.

Ultimately, the interface can be regarded as a kind of cognitive agent that seats in between the robots and the human, and that try to minimise the control errors, adapting towards the robot and the operator at the same time. An agent that serves the human operator, while understanding the controlled robot.

3.2 Background

Two fundamental elements for constructing this kind of interface are based on understanding and constructing methods for autonomous exploration of robot behaviours on one hand, and finding a suitable methodology for human-machine interaction on the other.

3.2.1 Control of Robotic Systems

Controlling a robotic system can be a very difficulty task, depending on the morphology of the robot. Robots with 1 or 2 Degrees Of Freedom (DoF) can be easy to control, such as simple two-wheeled robots. Indeed, the control can have a comparable complexity of that of a remote controlled toy car. On the other hand, complex arrangements such as 4 or 6 legged robots, or humanoids, can be very difficult to control, especially for non-standard operational tasks (i.e., not simply going forward-backward and turning). In this cases, the designer of the controlling device has to decide the level of expected autonomy of the robot by implementing a series of controlling patterns of various complexity and abstraction, such as high level commands (i.e. proceed to the next room) or low level commands (i.e. arrange a specific joint to certain degrees). In most cases the level of expected autonomy of the robot is driven by the task and the goal.

In the case of robots with no level of autonomy the control is based upon the direct manipulation of the robot's DoF. In the case of remote control, the input device needs to have at least the same amount of DoF so that the operator can achieve full functionality of the robotic morphology [58]. Examples of such control techniques can be found in [64] using a full body mapping or part of it as in [65].

On the other hand, traditional Artificial Intelligence research follows a top-down approach in designing robot controllers, usually involving a complicated, centralised controller that makes decisions based on access to all aspects of the global state. There are though systems build from a bottom-up approach, where localized, parallel, and distributed low-level controller provide the robot with adaptive and complex behaviours. Behaviour Based robotics [21], Nonlinear Dynamics and Self Organisation [6], and Evolutionary Robotics [30] are research fields developing systems that follow this bottom-up approach.

For our purposes, particularly interesting is the non-linear dynamics approach put forward by Ralf Der and the homeokinesis principle [6], which is a representative example of a the bottom-up approach in robot control and exploits self- organisation. Other examples based on the same principle that exploit self-organisation of the sensorimotor loop in robotics morphologies can be found in the work of Martius et. al. [7] and Hesse et. al [25]. In their approach they use Neural Networks to show how from simple structures and non-linear approximations, behaviours can be discovered in robots with varying morphologies. The idea of goal oriented behaviours is not stated in their research, but has been pursued by others using Reinforcement Learning techniques to guide the exploration [100].

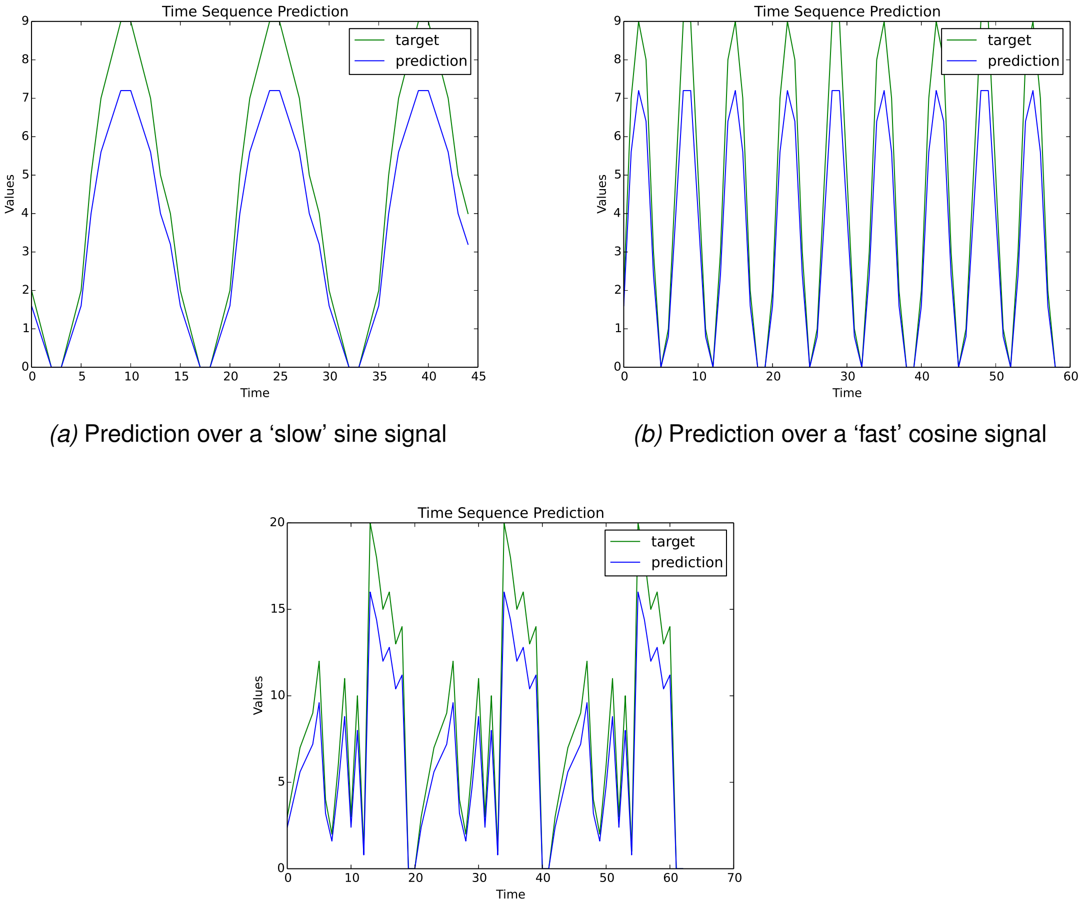
3.2.2 Human Machine Interaction

Thus far, Human Machine Interaction (HMI) systems are tightly designed around the applications and the machines to be operated. The design of interfaces to be used and the possible interactions between the human and the machine are typically based on ergonomic principles [49]. In terms of HMI, ergonomics relates to how the user will interact with a machine and how easy that interaction will be. The main goal of ergonomics can be stated as, the design of equipment which is, *a*) *Easy to remember*; *b*) *Easy to learn*; *c*) *Efficient to use*; *d*) *Effective to use*; *e*) *Enjoyable to use*; and *f*) *Safe to use*, for the user.

The concept of *affordances* was first introduced by J.J. Gibson [12]. It described the potential actions enabled by a given object, especially ones that is easily discoverable. The concept of affordance is applicable on the way we perceive control devices, as different people have the possibility of acting in a different way upon them. In this way the interface has the possibility to adapt to the user. This idea carries one of the most important aspects of the interfacing framework described here and allows the user to interact with the device in an intuitive way. We define here *Intuition* as the ability to understand something instinctively, without the need for conscious reasoning. Combining intuition with affordances permits to design an interface tailored for the user. Enabling the user to freely express the way of communicating their intentions for control through the interface provides us with a new way of dealing with ergonomics.

3.2.3 Intelligent User Interfaces

The merge of artificial intelligence and human-computer interaction brings forward the idea of Intelligent Interfaces [60]. In their studies on intelligent user interfaces, Hefley et. al [61], they describe intelligent interfaces as systems *that build on facts and heuristic knowledge of an expert, together with techniques for reasoning about unstructured situations*. In their research they use user interface management systems (UIMS) concepts as a basis for their research on intelligent interfaces. They distinguish between adaptive and flexible intelligent interfaces, with the first having the added capability to learn over time from experience to accommodate the user and their interaction, while flexible interfaces deal with cases in which the user can tailor the interface or when the interface can support several styles of interaction.



(a) Prediction over a 'slow' sine signal

(b) Prediction over a 'fast' cosine signal

(c) Prediction over a random signal

Figure 3.1: Prediction capabilities of the network, once trained over three time sequences

3.3 Extraction and Acquisition of Human and Robot Behaviours

In this sections the methodologies and the main principles at the basis of the implementation of a framework capable of supporting human intuitive control of robotic morphologies are described.

From an HMI point of view, following the work on humanoid robot control suggested by [2], operator's intention for control is captured as time varying configurations of an input device. In this paper a single methodology for the intuitive control of a humanoid robots is discussed, while in our approach we try to address the more general problem of acquisition of motion behaviour from the operator as a problem of time sequence recognition. In particular, we aim to capture the human intention for controlling the robot as time depended manipulations of the input device. Indeed, interfacing well defined segmented manipulations of the input device for remote control purposes is not different from the classical idea of having a set of discrete switches, such as a keyboard, with the switches replaced by time sequences. On the other hand, being able to provide a way for real time recognition of the operators behaviours on the input device is a challenging problem, still not conclusively solved in the field of human machine interaction. In our approach, the operator is free to perform any movement, as long as it is captured by the input device. In that sense, all hand movements are acceptable using a Leap Motion device, all body movements for a Kinect sensor and all buttons pressed in any order for any button based input device.

For the point of view of self-organisation and exploration of robot's behaviours, we present a model based on dynamical system which is capable to perform an exploration and extraction of behaviours of random robotic morphologies. With the term random, we suggest the idea of not having a single target morphology for the application of the interface. By coupling this system with a series of neural networks we are able to capture and reuse these behaviours as show in section 3.3.2. Behaviours include movements that are in general guided by the DoF of the robot, both translational and rotational. Also, based on the dynamics of the robot, more abnormal behaviours may appear, i.e. the octacrawl robot balancing on its tail. The results of the method are shown in section 3.3.2.

3.3.1 Interfacing the Human Operator

As stated before, the main challenge from an HMI perspective is that of properly sequencing and recognising the manipulation of an input device. To this end, among the many available solutions we concentrated mostly on Dynamic Time Warping and Recurrent Neural Networks.

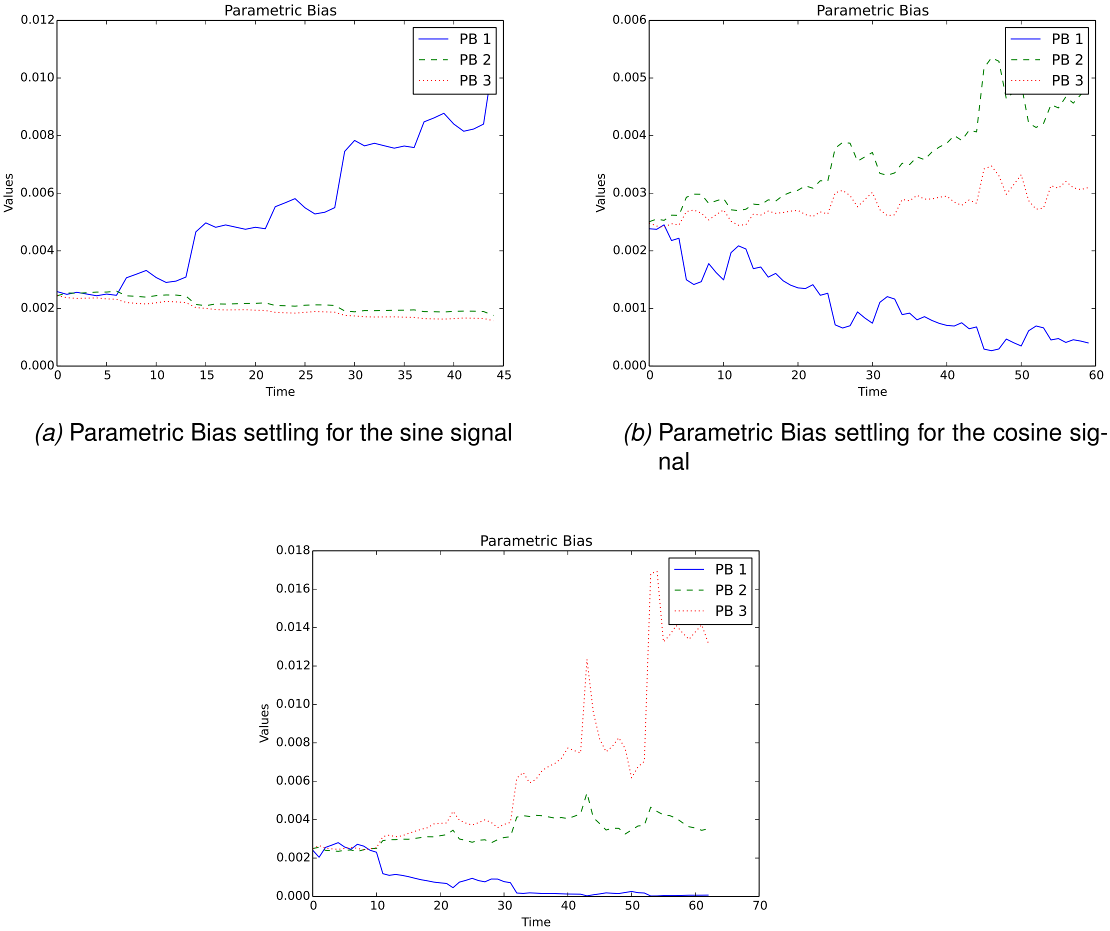
Dynamic Time Warping (DTW) is a distance measure used mainly in speech recognition community. It allows a non-linear mapping of one signal on another by minimizing the distance of the two. The DTW algorithm calculates the distance between each possible pair of points out of two signals in terms of their associated feature values. It uses these distances to calculate a cumulative distance matrix and finds the least expensive path through this matrix. This path represents the ideal warp - the synchronisation of the two signals which causes the feature distance between their synchronised points to be minimised [87]. Although DTW can provide a good measure of resemblance in time sequences it can only work once the control sequence is completed by the operator and their behaviour is captured. In the case of partial data, the sequence cannot provide enough information, even if stretched or squeezed in time, mainly because the method does not have the ability of completing a sequence by predicting the expected time steps.

Therefore, for flexibility reasons we focused our attention on Recurrent Neural Networks (RNN). Although this method produces a delay in the setup of the interface, due to the training of the RNN, the computational complexity of a trained RNN is very small and the representation of the trained sequences is very compact (the synaptic weights). In addition to DTW, RNN have also the ability to predict the next time steps according to the dynamic of the input, making the recognition faster and often without the need of presenting the full sequence.

Recurrent Neural Networks

There are many implementation paradigms for creating RNNs. Our approach is based on Jun Tani's works both in time sequence recognitions [93] and multiple time scales dynamics acquisition [94]. The main difference being that in his work the RNN has a *Jordan* type structure (recurrency on the output layer) and it is trained with Back-propagation Through Time (BPTT) algorithm [90]. In our implementation we implemented an *Elman* type structure with recurrency on the hidden layer, trained with standard Back-propagation [101].

The idea of Parametric Biases (PB) provides a way for both generation and recognition of dynamic temporal patterns. PBs are units in the input layer of the network capable of



(a) Parametric Bias settling for the sine signal

(b) Parametric Bias settling for the cosine signal

(c) Parametric Bias settling for the random signal

Figure 3.2: Recognition of the sequence by the self adaptation of the Parametric Bias units

adjusting themselves according to networks dynamics. During the training phase and after the error has been propagated to the weights of the networks, the values of the PB units are adjusted, trying to further minimise the difference between the target and actual output. The update equations for the ith PB unit at time t are,

$$\delta\rho_t^i = k_{bp} \sum_{step=t-l/2}^{t+l/2} \delta_{step}^{bp^i} + k_{nb}(\rho_{t+1}^i - 2\rho_t^i + \rho_{t-1}^i) \tag{3.1}$$

$$\Delta\rho_t^i = \varepsilon\delta\rho_t^i + \eta\Delta\rho_{t-1} \tag{3.2}$$

$$\rho_t^i = f(\rho_t) \tag{3.3}$$

the term $\delta\rho_t$, the delta component of the internal value of the PB unit, is calculated by the summation of two terms. The fist one represents the summation of the delta error propagated from the output units for l time steps, while the second term works as a low pass filter, inhibiting rapid changes in the PB unit, with k_{nb} being the coefficient for the filter. The update $\Delta\rho_t^i$ utilised a momentum term, the second term of the summation, for faster convergence. Finally, the activation of the PB unit is given through function f, a sigmoid function.

This updating step of the PB unit values is then performed, in real-time, once the network is trained. Given the network N, the temporal sequence $S(t)$ is generated as $S(t+1) = N(S(t))$. The input received at time $t+1$ represents the desired output of the network having as input the one of time t. Once the networks is trained the PB units are initialised with value zero, and a sequence is fed element-by-element to the network.The difference of the network output o and the desired output $S(t+1)$ in time $t+1$ can be calculated, given the known input from time t. Since the network is trained, the only possible source of error can be assigned to the PB units. If an error appears, that error value is propagated to the PB units, updating them according to the same set of equations as in the training phase. In this way the network readjusts the PB units values, to match the target output as best as possible. The resulting PB values give us the ability to identify whether an unknown sequence is a sequence the network was trained on, since the PB values of the training phase are known.

In the graphs we can observe both the ability of the RNN to predict the time sequences fed to it as in figure 3.1 and the recognition of the sequence as in figure 3.2. An important aspect of the setup, is the time step it takes for the network to recognise a given sequence. The figures 3.2a, 3.2b, 3.2c represent the sequences in 3.1a, 3.1b, 3.1c with lengths of 15, 20 and 20 respectively. We can see that the PB values settle very early, and they provide a correct recognition from steps 6, 5 and 12 respectively.

3.3.2 Interfacing the Robotic Morphology

Building the Behaviour Exploration Mechanism for the Robots

The autonomous discovery of available behaviours of a given morphology is the other fundamental element for designing an interfacing system that aims to reduce design constraints and maximise usability. To this end, we implemented a system consisting of two modules. One used for the exploration and self-organisation of the sensorimotor loop of the robot to be controlled and one for the extraction, storage and reuse of the acquired robot's behaviours. The robotic morphologies used for the experiments described in this paper are simulated with Open Dynamics Engine, ODE. The module for the self-organisation of the sensorimotor loop follows a dynamical system approach. The realization of the dynamics of the robot and of the world is done using a Controller (K) and World Model (W) cooperating for the effective exploration of the robots dynamics and an accurate prediction of world states, respectively, as discussed in [7]. Both are described by the equations described below.

The exploration module is described, according to time t, as:

$$\tilde{\mathbf{x}}_{t+1} = W(K(\mathbf{x}_t, C), A) \tag{3.4}$$

The controller K generates motor outputs $\mathbf{y}_t = K(\mathbf{x}_t, C)$ as a function of the sensory input $\mathbf{x} = x_1, x_2, \ldots, x_n$, depending on a set of parameters defined by the matrix $C_{n,n+1}$ and it is defined by the equation:

$$K = g(\sum_{i=1}^{n} C_i x_i + C_{n+1}), \tag{3.5}$$

where g is a sigmoid function.

The world model $\tilde{\mathbf{x}}_{t+1} = W(\mathbf{y}_t, A)$ estimates future sensory input $\tilde{\mathbf{x}}_{t+1}$ from the motor output $\mathbf{y}_t = y_1, y_2, \ldots, y_n$ depending on a set of parameters defined by the matrix $A_{n,n+1}$.

The parameter matrix of the world model, A, is adapted according to the delta rule [99],

$\Delta w = +\eta E_W \mathbf{x}$ with the error, E_W, described by the function:

$$E_W = ||\mathbf{x}_{t+1} - \tilde{\mathbf{x}}_{t+1}||^2 \tag{3.6}$$

with learning rate η = 0.1.

The controller updates its parameter matrix by gradient descent with respect to the error function,

$$E_K = ||\mathbf{x}_t - \tilde{\mathbf{x}}_t||^2 \tag{3.7}$$

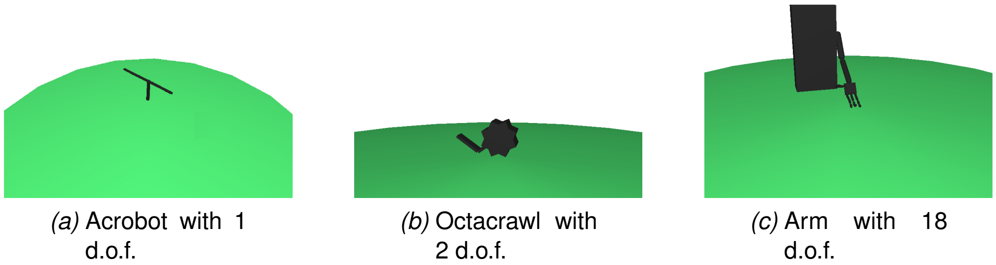
To calculate the above error, we find the $\tilde{\mathbf{x}}_t$ by calculating the motor input $\hat{\mathbf{y}}_t$ the world model should have in order to make a perfect prediction and then, the sensory input the controller K should have to predict the motor output $\tilde{\mathbf{y}}_t$. For updating the controller parameters the following rule is applied $C_{t+1} = C_t - \varepsilon \frac{\partial E_K}{C}$, with a learning rate $\varepsilon = 0.01$.

For the identification, storage and reuse of the different behaviours exhibited by the robot, we use a series of m neural networks (NNs), called *experts*. Each NN is defined according to the equation,

$$(\mathbf{x}_{t+1}, \mathbf{y}_t) = N_i(\mathbf{x}_t, \mathbf{x}_{t-1}), \quad i = 1, \ldots, m \tag{3.8}$$

The NNs, working in parallel, compete for the prediction of the motor command $\mathbf{y}_t$ of time t and the sensory input $\mathbf{x}_{t+1}$ of the next time step in a winner-takes-all method, with only the winning network being allowed to train on the current data $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$. Thanks to this process, each NN specialises to represent a region of the entire sensorimotor space of the robot.

The NNs consist of 3 layers, feedforward units where the hidden layer consists of sigmoid units, whereas the input and output layers of linear units. Online back-propagation is used to training the NN with learning rate $\eta = 0.01$. In each time step all NNs are activated with the same input and the one with the best approximation of the next sensor values and motor commands is selected as the winner. The sample is then added to



(a) Acrobot with 1 d.o.f.

(b) Octacrawl with 2 d.o.f.

(c) Arm with 18 d.o.f.

Figure 3.3: The different robotic morphologies used during the experiments

the training dataset of the winning NN and it is trained for another epoch.

Robot Behaviour Exploration, Extraction and Reuse Results

For testing purposes we applied the method described above to three different robotic morphologies, as seen in figure 3.3, with varying degrees of freedom and numbers of joints, respectively 1, figure 3.3a, 2, figure 3.3b and 18, figure 3.3c.

In figure 3.4 we can see how the experts are trained to identify different sensorimotor loops in the robot with 2 joints: The output of the network, describing each behaviour, as captured by the sensor values, stabilise and approximate the real ones more accurately as time and training size increase. In the example of figure 3.4, behaviour 1 stabilises faster than behaviour 2 as we can see from the convergence to a finite set of sensor values for each behaviour. This is caused by the difference in the size of the datasets for each behaviour. Some behaviours are more frequent than others making the dataset of the network describing them bigger and more accurate during the training phase. We can also observe a periodicity in the sensory values recorded, a direct result of the dynamical system approach used in the exploration mechanism. A consistent and stable over time behaviour is usually found when the system enters a basin of attraction, and progressively approaches the attraction point.

The behaviours observed vary between the morphologies explored. In the octacrawl morphology the method discovers among others, a way of moving forward, a way of jumping and a movement of the tail without changing position. Similarly, in the acrobot, behaviours include standing still upside, variable rotation speeds and a pendulum like behaviour. Finally, in the hand were in comparison less behaviours are extracted, we have up down movement of the whole arm, bending at the elbow and wrist.

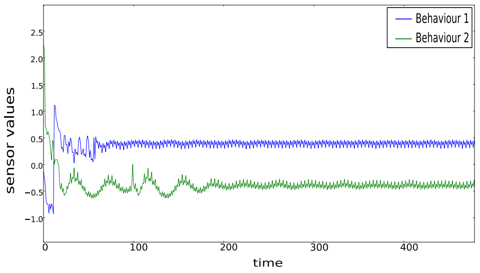
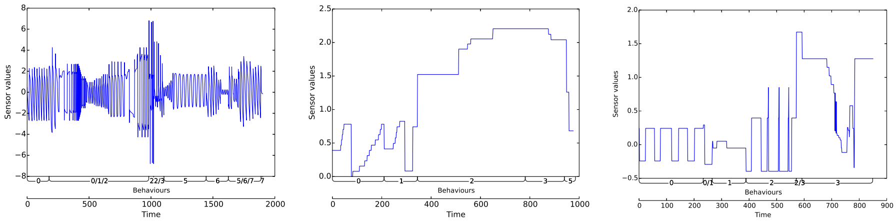


Figure 3.4: Plot of the sensor values for two different behaviours extracted from the octacrawl morphology, as they change through time during the learning phase

More interesting features of the system can be observed in the switching between behaviours. In figure 3.5a the behavioural changes of the robot with 1 joint are being displayed against time. The different behaviours become salient by the different sensor readings they produce. In figure 3.5c and 3.5b the behaviours of the 2-joints robot and the arm morphology are being displayed against time, respectively. Our interest in these graphs lies in the point of change between behaviours. In this context one behaviour is produced by activating the corresponding NN. The ID of the active network is noted in the horizontal axis, above time.

In all cases the exploration mechanism was able to identify and extract different behaviours. During testing these behaviours where triggered through the interface in random order and the sensor values of each morphology were recorded and correctly predicted by the network in control. In all graphs of the figure 3.5 we observe smooth changes in the sensory readings, regardless of the changes in behaviours. It appears that the system is able to produce and follow a trajectory from the old to the new attractor, and a consequently smooth transition in behaviours. In the first time steps following a behavioural change, we can observe the readjustment of the morphology, as recorded through the sensor values, smoothly moving towards the exhibition of the desired behaviour.

In the graphs of figure 3.5 it is also possible to observe the behaviour of the sys-



(a) Switching between behaviours using the acrobot morphology

(b) Switching between behaviours using the arm morphology

(c) Switching between behaviours using the octacrawl morphology

Figure 3.5: Switching between behaviours in the different morphologies used. In the horizontal axis the time and the ID of the expert(s) that control the robot. The figure show the expert ID and when two or more experts are in control at the same time the IDs are separated with '/'. The vertical axis shows the sensor values of the robot at a given time step.

tem in the case of simultaneous activations. When several behaviours are activated at the same time, results in the graphs show the ability of the system to mix the behaviours acquired seamlessly with no abnormal sensory readings, which indicates that no abnormal behaviours are exhibited by the morphology. In figures 3.5a, 3.5c we can observe the change in sensor values through time for the the robots with 1 or 2 joints respectively.

3.4 Conclusion

In this chapter we have described the principles, the background and the methodology for implementing an interfacing mechanism that allows the control of any type of robotic morphologies in an intuitive way through the manipulation of an arbitrary input device.

In particular, this chapter aims to show the methodological assumptions and technological building blocks of the proposed framework and the feasibility of the proposed system, based on the testing results of the technologies.

The proposed exploration mechanism for robot behaviours was successfully implemented. The robustness of the system is shown, both by the stability of the mechanism when switching between the self-organised behaviours, and by the ability of combining such behaviours. At the same time, we propose a mechanism able to support continuous interaction with the operator. The results of the method implemented show good recognition and prediction capabilities, providing a viable solution to the problem.

Given a suitable representation of the robot morphology and controller, and an intelligent interface, we have the potential of reducing the complexity that the user has to face in the interaction with a robotic system. The complexity of the controlled robot can be reduced by self-organising behaviours and capture the complexity of human behaviour as it could be exhibited through an arbitrary input device.

Chapter 4

Two-way Adaptive Interface for Intuitive Robot Control

Potentials of Echo State Networks in Human Behaviour Mapping

4.1 Introduction

In this chapter we present a continuous, on-line, real-time methodology for the remote control of mobile robots. Instead of the user adapting to the interface and control paradigm, the system proposed allows the user to shape the control motifs in their way of preference, moving away from the case where the user has to read and understand an operation manual. Starting from a tabula rasa basis, the system is able to identify control patterns (behaviours) for the given robotic morphology and successfully merge them with control signals from the user, regardless of the input device used. The structural components of the interface are presented and assessed both individually and as a whole.

Remote control of robots is usually seen as a classification problem, with the user acting on an input device, the system identifying the user's behaviour, and triggering the appropriate response on the robot. Under such a paradigm two variables need be fixed beforehand, that is the input device and the robotic morphology to be controlled.

Indeed, most research is performed targeting a specific robotic morphology [102, 103]. This allows for tailored solutions on the behaviour generation for the robot, solving inverse kinematic models or having hard-coded routines of interaction. Our approach working on a model free basis creates and adapts the robot's controller under a self-organising paradigm. Being agnostic towards the controlled robot highlights the necessity for a mechanism to explore the robots capabilities, with respect to its environment. It is through the interaction with the environment that the embodiment's properties can be revealed.

At the same time, the plethora of studies on remote control are formed tightly around the input device to be used [104, 105] and are commonly constructed by field experts, to be used by field experts. Complicated input devices and non-intuitive control patterns are created which the user has to learn in order to use the system. Our research aims at an intuitive control paradigm, where the user's intentions for control are formed and used for the interaction. Adaptive methodologies have only started appearing, most of them working under a classification paradigm [106, 107]. Although classification can provide a robust tool for input recognition, our approach provides a robust way of mapping inputs to a lower or higher dimensional space, allowing for the geometric properties of the input to be explored i.e. opposing behaviours having opposing mappings and the ability to mix behaviours.

Most robotic remote control systems rely on the intelligence and cognitive capabilities of the operator to understand the control paradigm and the robot's capabilities. The operator once familiar with the input device, its functioning and its potentials, has to understand the control paradigm and how the control flows are defined to be used. Indeed, in most cases the operator needs to acquire the knowledge required for control through a training procedure (i.e. reading the manual for operation, practice on the controls)[62].

The problem we try to solve, is a dual problem; mapping the human operator's control signals to the actions of the robotic morphologies. In doing so, we try to merge the two apparent dynamical systems involved. The one formed by the operator's input signals and the other by the robot's behaviours.

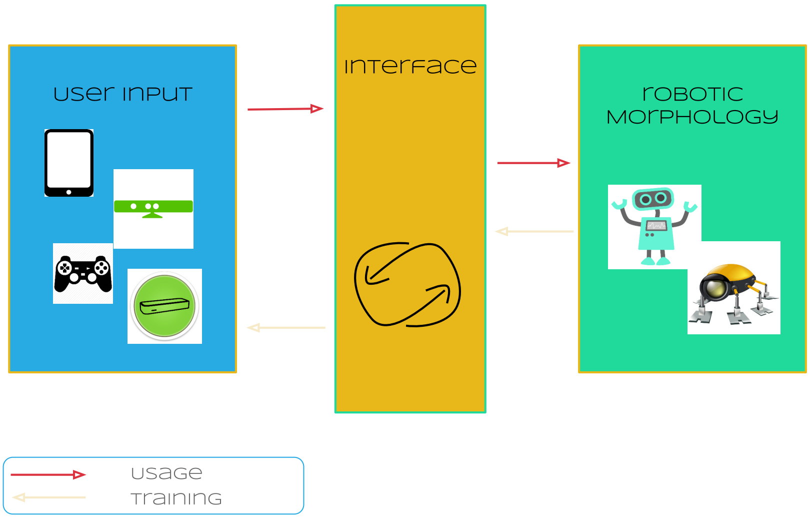


Figure 4.1: Schematic representation of KURE

From the operator's perspective; being able to observe the robot's embodiment interacting with the environment, allows for a better understanding of it. In this process, intentions for control can be formed. Being able to capture those intentions and associate them with the robot's behaviour, can result in a control flow - an intuitive control paradigm - tailored to each user and robot. This, with the operator being free to act upon the system - that acts as a mediating agent between user and robot - namely the interface.

4.2 Methods

The methods section is divided in two parts: covering the formation of the controllers for the robotic morphology, and the input acquisition from the user. Regardless of the robot to be controlled our intention is to extract primitive behaviours, capable of being combined, providing a rich enough scaffolding for the control of the robot. We avoid human intervention in the formation of the robotic behaviours in order to have an autonomous system. Indeed, under this scope the behavioural primitives should be formed through the interactions of the robot with the environment, allowing for a learning mechanism grounded to the robot.

There are numerous ways of forming controllers capable of achieving pre-specified behaviours, supervised learning, homoeostatic regulation, central pattern generators (CPG), evolutionary computation methods (EC), to name a few. All the above, although intrinsically different share the idea of external guidance. A teacher behaviour needs be formed for the robot to imitate in the case of supervised methods, external perturbations for a homoeostatic adaptation, the tuning for CPG, and a scalar measure for a target function in EC.

On the other hand, regardless of the input device used, our goal is to capture the intentions for control from the user as expressed through the input device. In this end we treat the input as a time sequence of manipulations of the input device. Our intention is to allow the user to freely interact with the input device forming their own personal control patterns. Segmenting the input sequence or using sliding windows techniques to imitate continuity on the input sequence is not a solution. Indeed, since our aim is to allow the user to form their interaction patterns, we cannot make any assumptions of the length of the time sequences (and thus on the size of the sliding window). At the same time, working under a mapping and not a classification paradigm we cannot use statistical methods. Finally, the system must not take long to initialise and adapt to the user preferences, as that would degrade the user experience. In our method the training time required for the input recognition subsystem is less that 10 seconds.

One solution could be the usage of Dynamic Time Warping (DTW), but in such a case the input should be segmented, violating our need for continuity. Hidden Markov Models could be used, but this would fall under a classification paradigm. Recurrent Neural Networks, pose a promising solution for our desired mapping, but the training techniques used (Back-Propagation through time) require a lot of time to train.

The system should work as a mediator between the robot and the operator; an interface connecting the two systems as seen in figure 6.2. In doing so, it should be capable of exploring the robots potential, the users behaviours and connect them seamlessly, placing the human in the loop.

4.2.1 Self-organisation of Robotic Behaviours

For the formation of the control sub-system for the robot, a self-referential dynamical system is derived and a principle for self-organisation of robotic behaviours [6, 108]. The idea here is to try and maintain a smooth control behaviour keeping the robot at a constant kinetic state. This property of the system (i.e. self excitation) was first formulated by Ralph Der and referred to as homeokinesis [109].

The learning in this procedure occurs based on the time-loop error; the error between the real behaviour and the model's prediction. Based on the homeokinetic principle the self-organisation of the sensory motor loop of the robotic morphology is possible without an external driving force (i.e. teacher signal or external perturbation). From this, a repertoire of behaviours emerges, which we are able to capture in the form of behavioural experts. These experts can later be reused and combined to control the robot. The behaviours, as demonstrated in the following section 6.4.1, vary in complexity, time, and are entirely based on the robot and its interactions within the environment.

For the exploration of the robot's capabilities we work as seen in [7]. We want to be able to produce motor outputs from sensory readings and from them predict the next sensory state of the robot. Creating a sensory-motor, and a motor-sensory mapping, allows us to derive an error signal for the update of the systems parameters.

For the two systems described above, the Controller K and the World Model W, their functions for operation follow the ones described in 2.3.

In order to use these behaviours, the sensor values from the robot are passed in the selected NN and the respective motor commands (output of the NN) are applied to the robot.

4.2.2 User Input as a Continuous Signal

Following the constrains mentioned in the beginning of the section, the system makes use of Echo State Networks. Combining the effect of multiple time scales and the possibility of mapping the time sequence dynamics to a fixed dimensional space, [110] formulated the echo state approach on training Recurrent Neural Networks, namely Echo State Networks (ESN). One of the most appealing features for our application, is the fact that the network is trained using linear regression on its output layer only, reducing the complexity of training with BPTT. The input signal propagated to the Dynamic Reservoir, expands in dimensions allowing for easier manipulation of the signal. In this setup the only trainable weights are output layer's, reducing the complexity of training to a matrix multiplication.

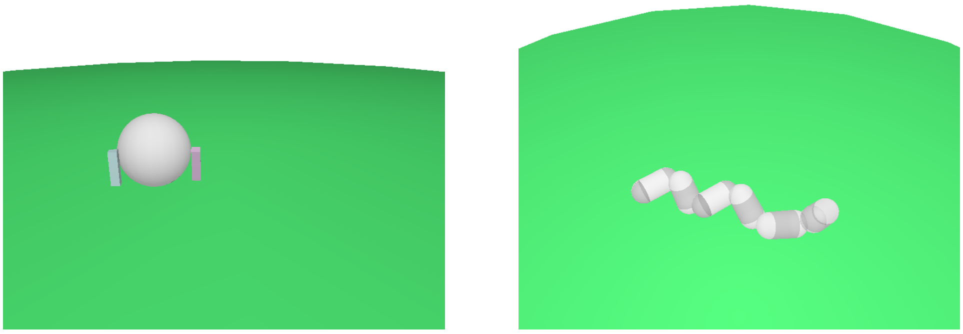
Echo State Networks (ESN) provide an architecture for efficient training of RNN in a supervised manner. One can distinguish two main components in an ESN. Firstly, the Dynamic Reservoir (DR), a large, randomly wired, recurrent neural network with fixed weights. The DR gets activated by the input and provides a non linear response for this input. And the output signal, which is trained as a linear combination of the activations of the DR. This way the computational resources and complexity required for the training RNNs is reduced to the adaptation of the output connections of the ESN.

The mathematical details for the creation of the Echo State Network follow the ones described in 2.2.2.

4.3 Experimental Results

Having elaborated on the methods to be used for the behaviour extraction from the robotic morphology and for the sequence recognition from the user, we now proceed with the description of our experimental setup with the results obtained by each individual component and the system as a whole.

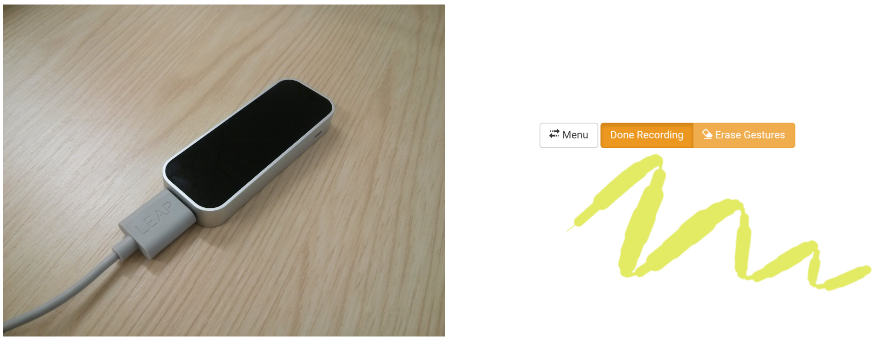
For our test scenarios we used two different - simulated- robotic morphologies (Fig.



(a) The spherewalker morphology.

(b) The snake morphology.

Figure 4.2: The two morphologies used in the experiments in the simulated environment.



(a) The Leap motion device

(b) The graphical interface for the touch-screen device. The yellow line shows the gesture being recorded. Disappears when a finger is not touching the screen.

Figure 4.3: The two interfaces used in the experiments.

4.2b, 4.2a) and two different input devices Fig. 4.3. The robotic morphologies were simulated using Open Dynamics Engine (ODE), through Python. In doing so, we were able to simulate the physical properties of the environment and so obtain a good representation of the dynamics of the morphology.

The 'spherewalker' morphology has two motors each with 1 Degree of Freedom (DoF), and two sensors, measuring the joint positions. The 'snake' morphology has five motors each with 1 Degree of Freedom (DoF), and five sensors, measuring the position of each joint. The values recorded from the sensors are normalised to fit $[-1, 1]$ in both morphologies.

(a) Smart phone device used as a touch screen input for KURE.

(b) Tablet device used as a touch screen input for KURE.

Figure 4.4: Two types of input devices used with KURE, for haptic 2-dimensional input signals. The difference is the screen size; 4.7 inches for the phone and 9.7 for the tablet.

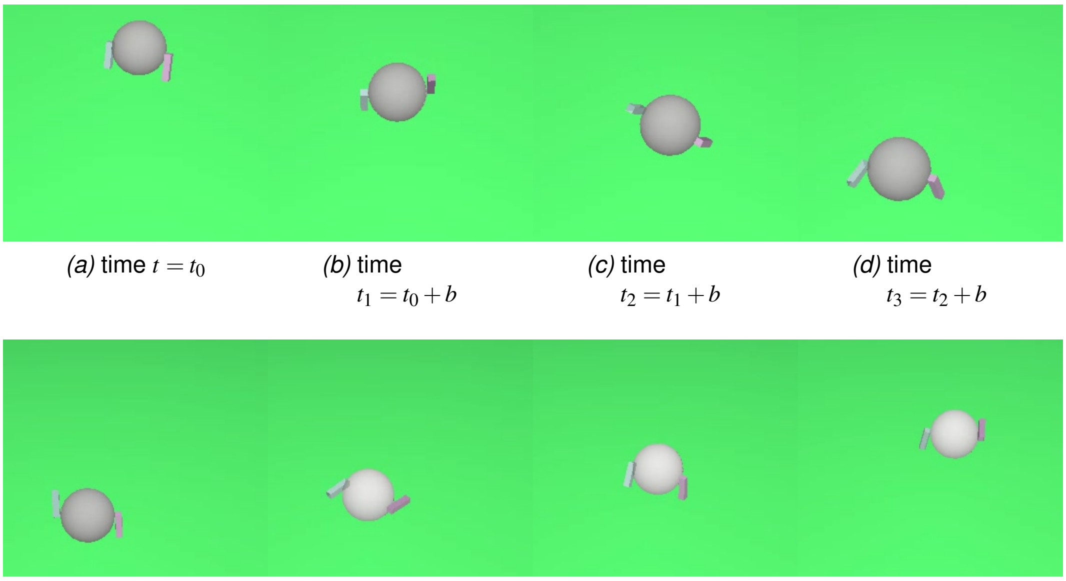
For the input devices, we capture a two dimensional signal from the touch-screen device, and a six dimensional signal from the Leap Motion device. In the case of the touch-screen device the input signal is 2-dimensional, using the horizontal and the vertical offset of the touch point at every time step. The signal values are normalised in $[0,1]$ for each dimension and captured at the frame rate allowed by the software used $(>30fps)$.

The Leap Motion device is a sensory device allowing for hand and finger positions in space, as input. Using the $JavaScript$ library provided by the manufacturer, and the same web-server setup with the touch-screen device we record six, 6, values to describe the hand posture at each frame. The values recorded represent the 3 rotational and 3 translational DoF of the palm of the hand.

4.3.1 Stage 1 - Training Towards the Robot

As illustrated in figure (6.2), the interface is working in the shared boundary between the two systems present: the robot and the input device. On the robot side, the interface captures the behaviours of the robot as sensory motor sequences. In our experimental setup, as sensory inputs we understand the joint positions of the robot. Thus, we work with proprioceptive sensory input to create the kinematic model of the robot. The motors of the robot can be controlled both via a PID controller and Torque from the same architecture with the same parameters, as through homeokinesis the controller network K adapts to perturbate the sensorymotor loop of the robot at hand.

Every time step (t) of the simulation, the module following a homekinetic adaptation produces motor commands, and a prediction of the resulting sensory state of the robot. In the next time step $(t+1)$ of the simulation the actual sensors are recorded and the time loop error of the homeokinetic control is calculated training the Controller K and the World Model W (see section 4.2.1). In parallel to this, in every time step (t) the 'expert' neural networks perform a forward pass, predicting the motor commands of time t and the sensory predictions of time $t+1$, of the homeokinetic module. Working in a winner takes all scheme, the network-expert with the best prediction adds its input and output to its dataset and a single step $(1$ epoch of training) of training is applied. This way each network specialises in a single behaviour, thus becoming an 'expert' of the behaviour.



(a) time $t = t_0$

(b) time $t_1 = t_0 + b$

(c) time $t_2 = t_1 + b$

(d) time $t_3 = t_2 + b$

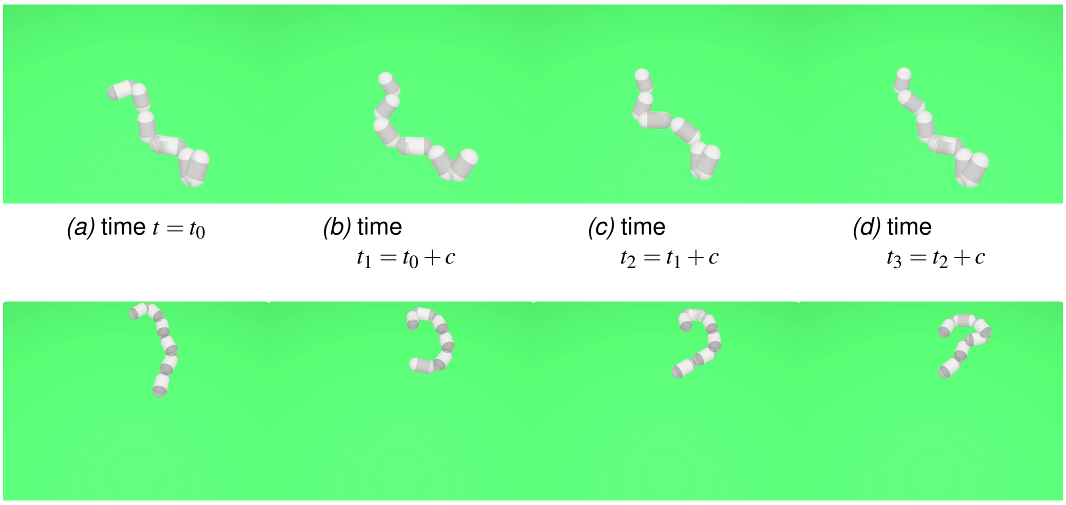
(e) time $t = t_0$

(f) time $t_1 = t_0 + b$

(g) time $t_2 = t_1 + b$

(h) time $t_3 = t_2 + b$

Figure 4.5: Behaviours of the sphere-walker morphology. The time constant b is set to $0.2sec$.



(a) time $t = t_0$

(b) time $t_1 = t_0 + c$

(c) time $t_2 = t_1 + c$

(d) time $t_3 = t_2 + c$

(e) time $t = t_0$

(f) time $t_1 = t_0 + c$

(g) time $t_2 = t_1 + c$

(h) time $t_3 = t_2 + c$

Figure 4.6: Behaviours of the snake morphology. The constant c is set to be $0.5sec$.

In figures (4.6) and (4.5), behaviours of the snake and the sphere-walker morphologies can be observed. We can see how a moving downwards and a moving upwards behaviours have been found for the the sphere-walker morphology. The graph displays snapshots of the simulated environment while the robot is being controlled by a behavioural expert. The same is seen with the snake morphology in figure (4.6). Two of the found experts are shown, as an example, controlling the robot and producing their respective behaviour.

As we described, and shown in [108], these behaviours can be intersected and also combined. Indeed, in our studies we have shown that the transition between them is smooth and so is the robot's resulting behaviour. In addition, we have shown how these behaviours can be linearly combined to produce new, stable, behaviours.

4.3.2 Stage 2 - Training Towards the Input Device

On the other side, the interface, after having trained on the robotic morphology, has to train on the user input. The system at this stage is able to stimulate different robotic behaviours. To capture the operators intentions for control, we reverse the information flow. In this stage the robot exhibits the behaviours extracted through the homeokinetic controller with the user responding to them. In order to form intentions for control the user observes the robot acting and responds with manipulations of the input device. At this stage the interaction between the human and the robot is recorded and the training set stored.

This set will be used to form the mapping from the K-dimensional space of the input device to the N-dimensional space of the experts. Since we want to perform a mapping, the output of the network is chosen to be coordinates in the N-dimensional expert-space. Working in the N-dimensional cube, each expert is found in each vertex of the N-dimensional unity hypercube. So, that expert-1, lies in $< 1_0, 0_1, 0_2, ..., 0_N >$, e.t.c.

The network is trained, performing linear regression on the output weights of the network for the whole dataset. The complexity of the calculations required is small enough to allow for the training of the network within $5s$. This makes it possible for the network to be trained for each user, as the system is about to be used.

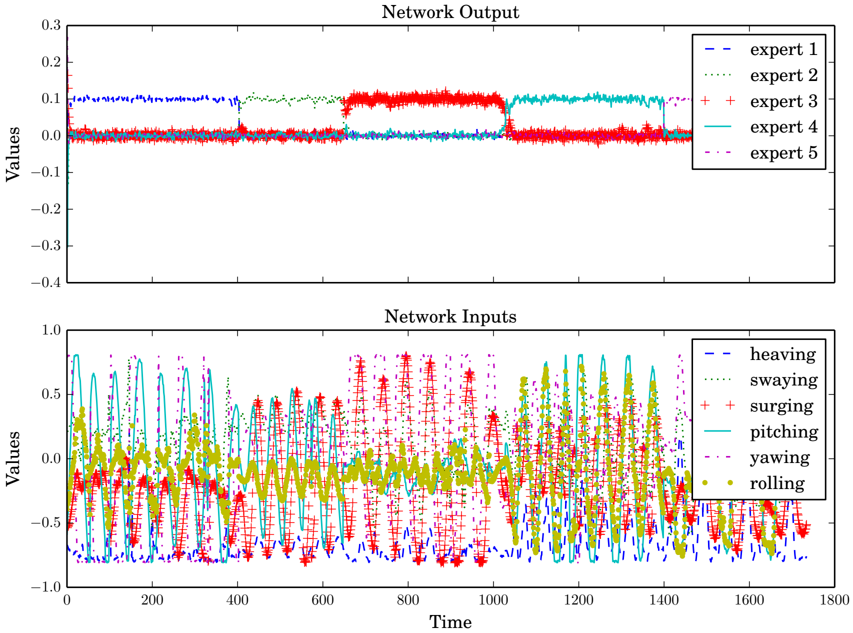


Figure 4.7: Validation of a trained ESN. In the top sub-graph is the output of the network, with each colour and line style representing a behaviour recognised in the input. On the bottom sub-graph is the input to the network, again each input node is depicted with a different colour and line style. All values are plotted against time.

In figure (4.7) we can see the validation of the training of the ESN. The x-axis represents time and y-axis represents input and output values, in the bottom and top graph respectively. In the top graph we can see the output of the ESN. Each colour describes a different dimension of the expert space. The dimension with the higher value is the representing dimension of the expert recognised in the inputs dynamics. In the bottom graph, the input of the network is depicted, showing the manipulations of the input device as they happen in time. Time is aligned through all three graphs.

In the case of figure (4.7), the input comes from a Leap Motion device. Input is acquired at the frame rate of the device (>30fps), from within the boundaries of $[-1, 1]$ for each DoF resulting in a cube where the interaction is recorded. The network is trained on the data provided and generates a perfect result for the training set (validation).

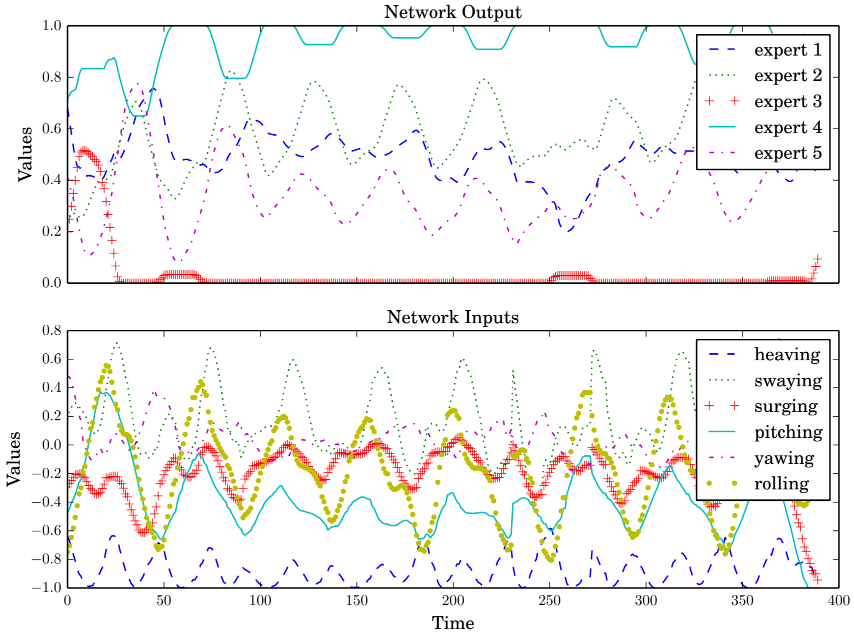


Figure 4.8: Usage of a trained ESN. In the top sub-graph the output of the network is depicted, with each colour and line style representing a pattern recognised in the input. In the bottom sub-graph the input as recorded from the input device is plotted. All values are plotted against time and are aligned as recorded.

For the trained network shown in (4.7) in order to test the capabilities of the network we have the user perform the gestures again in random order. This way we are able to test how well the network can cope the noise of real time usage of the device. Again, the Leap Motion is fed inputs and sampled at the frame rate of the device, with the input being recorded only within the boundaries of the cube mentioned above.

As long as the user is manipulating the input device, the ESN is activated with the recorded input. The ESN running real time, receiving input values at the frame rate of the input device, maps the input to the expert space. The network is able to recognise the input patterns of the user correctly.

In figure (4.8), we can see one of the behaviours; a cyclic motion of the hand in vertical space above the device. The x-axis represents time, while the y-axis the output, and input for the top and bottom graphs. Again each colour in the output represents a dimension of the expert-space.

4.3.3 Stage 3 - Usage of the Interface

Having trained both sides of the interface, the system is now ready to be used. The operator, manipulating the input device, provides the input to the ESN. The DoF of the input device are recorded continuously over time, producing the input sequence to the ESN. Each time step recorded is fed to the ESN, exciting the internal dynamics of the network and producing an output.

In figure (5.3), we can observe a close up of the recognised patterns from the ESN. CW notes a clockwise circle pattern on movement by the user, ACW an anticlockwise, and $Up-Down$, up down movement pattern of the hand. If we observe the first segment of the graph, as separated by the first vertical line, we see that the network correctly recognises a CW motion as input. What is more important is that the ACW motion is having a negative value, as the input pattern observed is "opposite" to it. In the next section we see the transition of the output to the $Up-Down$ motion. In this we observe that the network can mix the two in the output, while the ACW still remains negative, as the input is still opposing that pattern. Moving to the forth segment, the user is now performing an ACW input pattern and the ESN correctly recognises it. At the same time we observe that the CW pattern is negative as it is opposite to the one observe.

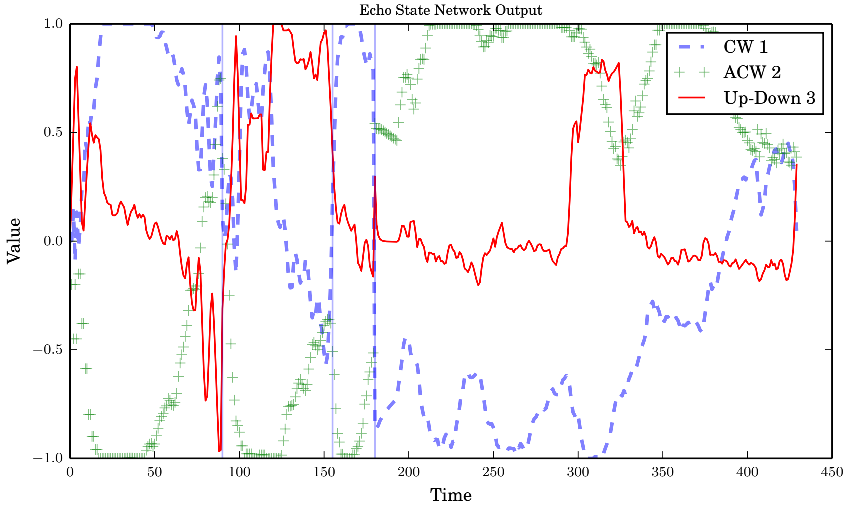


Figure 4.9: Usage of a trained ESN. In the graph the output of the network is depicted, with each colour and line style representing a pattern recognised in the input. All values are plotted against time

The $Up-Down$ recognition settles at 0 level again, until the user starts mixing the input patterns again, as observed at the mid-point of the segment.

Taking into account that the network has been trained by the responses of the user to robotic behaviours, we deem this an important feature of the system. As an example, lets assume the CW motion is mapped to the robot moving forward and the ACW backwards. Having opposite behaviours mapped being understood as opposite, provides the network with an "insight"; the user cannot be performing two opposing behaviours at the same time, but they can be performing the $Up-Down$ in combination with any of the above. At the same time, going back the expert networks we see that the combination of their outputs is done in a linear fashion. The contribution for each expert, in the final motor values of the robot, is calculated from the output of the ESN. This way, we observe that the robotic behaviour responding to the opposite motion, from the one observed in the input, not only is it suppressed, but also reversed, contributing to the correct behaviour being exhibited by the robot faster.

4.4 Conclusions

The interface is able to place the human operator in the loop of the robotic behaviours. In doing so, we establish a human centric control paradigm of robot control. Instead of having a learning procedure to train the operator on the usage of the interface, we adapt the system. User preferences, either as manipulations of the input device, or the input device itself (here Leap Motion and Touchscreen), fully shape the control experience.

We are able to provide a mapping between the two different time scales present; the manipulations of the input device happening according to the user preferences of the input device, and the robot behaviours guided by homeokinesis. Each point of the resulting expert-space represents two time sequences, able to unfold differently in time. Towards the robot, and through the expert-networks, each point is mapped to a robotic behaviour. Towards the human, each point is mapped to a time depended manipulation of the input device. This way both systems are mapped in a shared space, providing a robust and consistent way for control.

Chapter 5

An Adaptive Architecture for Human-Robot Behaviour Mapping

Functional Analysis of Reservoir Architectures for Robot Control

5.1 Introduction

In the field of pattern recognition, Recurrent Neural Networks provide a prominent solution. Reducing the cost of adaptation while having comparable performance Reservoir Computing has lately found many applications. In this paper, an architecture is described which performs pattern recognition in continuous time signals. The presented work is mostly centred on human-robot interaction and the need for an adaptive method to map control signals to robots behaviours. A supervised method is utilised for the training of the network and an unsupervised method for the adaptation of the reservoir. The proposed method is tested and analysed using a set of dynamic gestures. At the same time the feasibility and applicability of the proposed mechanism is tested under a scenario of robot navigation. Key properties of the setup are examined and tested.

Coupling the dynamics of humans' movements and the dynamics of a machine in order to control and direct the machine dynamics is a complex task. Mapping signals from one to the other in a continuous manner, in such a way that human users find intuitive and capable of expressing their own wishes and intentions, implies both detection and recognition of the input signals, as well as the full exploitation of their temporal aspects.

Moreover, such detection and classification of sequences should be performed on the fly, in order to make the user in full control of the machine, thereofore, a computational system that performs both tasks in real-time is of crucial importance in the field of human machine interaction. Whether the machine is a computer, a robot, or an integrated system in which both human's and machine's autonomy are involved (i.e. a self-driving car), being able to provide a direct and natural way of interaction between the human to the machines which are in control can ease the usage of such systems, and also bring them 'closer' to the operator. 'Closer' in the sense that users do not perceive the machine as an external entity, but a continuation and expansion of their own body. At the same time, the emergence of adaptive computational techniques allows for systems that seemlessly adapt to user preferences. Indeed, being able to connect humans and machines in such a way that the machine adapts to the user willing and intentions, rather than forcing the user to learn how to use and forcefully direct a given machine, has the potential to produce an easier, more comfortable and, above all, 'natural' usage of the system [62, 49]. Therefore, the approach towards a system that can adapt to the users, in order to detect and classify their actions, has an unquestioned importance in the advancement of action recognition systems, and ultimately in human-machine and human-robot interaction.

Adaptation towards the user is important, as it allows for personalised patterns of communication between the user and the machine. Indeed, besides improving the user experience, personalised controls can also enhance the usability of the system itself, making its usage easier and more intuitive. Adaptivity, in particular, can better accommodate user's needs, whether it is out of preference or necessary for the user's task to be accomplished (i.e., the machine to control has more degrees of freedom than the user). The challenge in this case, is to create a system capable of adapting to the user based on a very small set of training examples, in a short time and with a high degree of reliability. At the same time, to provide a natural way of communication between the user and the machine, the system should be able to recognise a specific sequence in a timely manner from a stream of data, effectively placing the human user in the interaction loop.

Imagine the action of driving a car: Not only a consistency and accuracy in recognising the driving commands is needed from the on-board mechanics and electronics of the car, but also the driver's ability to perform adjustments on the steering wheel - the input for the car control system - based on the car's behaviour is equally important. Likewise, when a human and a robot are coupled in their actions and behaviour, the extent at which the user performs an input command depends on how the robot implements the corresponding behaviour. Modulating the behaviour of the robot requires that the corresponding user behaviour is effectively recognised and propagated to the control system of the robot. In the particular case of continuous interaction, being able to inform the robot of the input magnitude or intensity is also fundamental. Thus, having user and robot behaviours coupled requires the following characteristics: (i) partial input observations to yield partial output results; (ii) the input signal's intensity to be propagated to the output; and (iii) smooth transitions in the recognition of different input signals.

Another important aspect of such interaction is time. Specifically, the time required for the computations of the recognition model and for handling the dynamics of the input signals. In this context, three are the main aspects that require attention: (i) the recognition model should be able to accommodate input patterns of different lengths; (ii) it should be trained and able to adapt to different users needs and preferences in a short time, so that the user does not disengage; (iii) the recognition should be implemented with a low complexity of computation. This is particularly important, as the recognition should take place fast enough for the system to have a timely response for the user.

Adaptive methodologies that present useful features like the ones above have only started to appear, most of them working under a classification paradigm [106, 107]. In this context, the challenges presented are mainly two: (a) detecting that a sequence is actually present in the data stream received from the input and (b) correctly classifying it. Most research features these two aspects with independent mechanisms [111, 112], however, having a unified mechanism for the two tasks, as discussed in the present paper, can provide beneficial aspects, as it might save computational resources overall and, at the same time, it might implement the recognition process faster.

Moreover, the task of dynamic sequence recognition becomes especially complicated when working with real and continuous streams of data and the complexity increases when the sequences have different lengths. Methods used for the classification span from distance measures (e.g. Dynamic Time Warping) [113, 114] and statistical models (e.g. Hidden Markov Models) [115, 116], to artificial neural architectures (e.g. Recurrent Neural Networks) [117, 118, 119, 120, 121] and hybrid solutions [122]. These methods vary in complexity and adaptability, with Recurrent Neural Networks being one of the most promising direction in the field [123]. Adaptation of RNNs though, is known to have high computational complexity. In addition, the training procedure is shown to have difficulties in finding good solutions, usually referred to as a gradient vanish problem [124].

Given the inner complexity of the recognition and classification task itself, working in real world environments is particularly difficult and demanding for adaptive models. Performance degrades rapidly when working directly with noisy user data taken from real input devices, making most methods not applicable in real world situations. Cleaning and preprocessing input data, as it is often required for model to work, is not a viable option when the fundamental demand is for a method that should be readily available to the user and work reliably in real-time. The task becomes even more difficult when the input is sampled in real time and is treated continuously. Not having the ability to segment the input data, i.e. not having a starting and stopping point, makes the usage of recurrent methods necessary, as they can integrate the signal continuously in time. On the other hand, training such models requires clean data to perform well, making them difficult to train with data obtained from real users. A potential solution in this case is a computational model that is able to capture the internal dynamics of a behaviour, such as an action performed by the user on a given input device, and thus provide a robust recognition.

A recurrent architecture that is shown to work well with noisy data under the restrictions mentioned above is the Echo State Network approach. ESNs seems to perform surprisingly well with noisy data directly taken from a user actions and can also adapt rapidly, making their usage for user oriented systems particularly appealing [121, 125, 126, 127, 128]. In the present paper, since we are interested in behaviour recognition, data comes directly from the user manipulations of an input device. Data can be noisy and the user repetition is not always perfect, resulting to training sets of data with high degrees of noise and variation between samples (e.g. gestures, behaviours). The ESN approach followed here provides a stable and robust mapping of the input commands for user behaviour recognition.

For the investigation and validation of the method, we have followed a methodology that encompasses three stages. Firstly, we establish the validity of the proposed setup and neural architecture by benchmarking and reporting its accuracy on the recognition of actions obtained from a publicly available dataset. This allows to compare the proposed neural architecture against alternative state-of-the-art methods and also against baseline methods. Secondly, we investigate the properties of the architecture on a dataset of sequences created in house with actions recorded by the experimenter with a Leap Motion device and made on purpose to better resemble the real ones that might be obtained by casual users. The intention is to get more detailed information about the property of the system on realistic sequences before exposing it to real users. Finally, we perform a user testing of the system on a small group of people, asking them to control a simulated robot. Characteristics of the neural architecture employed, as well as methodology and results of such investigations are described in the following sections.

5.2 Material and methods

5.2.1 Echo State Network

Echo State Networks (ESN) provide an architecture for efficient training of Recurrent Neural Networks (RNN) in a supervised manner [129, 130]. One can distinguish two main components in an ESN. Firstly, the Dynamic Reservoir (DR), a large, random, recurrent neural network with fixed weights. These weights get initialised once and are not adapted through the training procedure. The DR is activated by the input and the feedback from the output providing a non-linear response to the input signal. The neurons of the DR usually have sigmoidal activation functions, with hyperbolic tangents to be the prevailing choice. The second part of the ESN is the output, resulting from a linear combination of the reservoir's activations. Only these weights connecting the reservoir with the output are adapted through the training procedure.

For an ESN to function properly, the *echo state property* (ESP) is essential. ESP states that the dynamics of the DR will asymptotically washout, from the initial conditions. It has been observed, that this can be achieved by scaling the *spectral radius* of the DR weights W to be less than unity [97]. That is the largest eigenvalue of the weight matrix for the DR weights should be less that unity. This condition states that the dynamics of the ESN is uniquely controlled by the input, and the effect of the initial states vanishes.

The setting of spectral radius is also associated with the *memory* of the DR [97, 131]. That is the time steps it takes for the dynamics of the reservoir to washout and thus the past time steps for which information is incorporated to produce the output.

Echo State Network's Dynamics Formalisation Assuming an ESN consisting of N units in the DR, K input units and L output units. A matrix W_{in} of size $[K \times N]$ connecting the input to the DR, a matrix W of size $[N \times N]$ describing the connections amongst the DR units and a matrix W_{out} of size $[N \times L]$ connecting the DR to the output, and finally a matrix W_{outFb} of size $[L \times N]$ connecting the output to the DR establishing the feedback connections from the output to the DR.

Assuming time n, the input signal driving the reservoir is $u(n) = [u_1(n) \cdots u_K(n)]$, the state of the DR neurons is $x(n) = [x_1(n) \cdots x_N(n)]$ and the output signal is $y(n) = [y_1(n) \cdots y_L(n)]$.

The mathematical details for the creation of the Echo State Network follow the ones described in 2.2.2.

5.2.2 Training Procedure

During training the only weights adapted are the ones connecting the DR to the output, W_{out}. Let us assume a driving signal $u = [u(1), \ldots, u(n_{max})]$ and a desired output signal $d = [d(1), \ldots, d(n_{max})]$. The training procedure of the ESN involves two stages: (a) sampling and (b) weight computation.

Sampling In this stage the output is 'written' in output units, a procedure referred to as teacher forcing, and the input is provided through the input units. The network is initialised using a zero initial state x.

The network is driven by the input and output signals for n times $n = 0, \cdots n_{max}$, at each time step having as input $u(n)$ and teacher signal $d(n-1)$, this since there exists the feedback from the output. For the first time step where d does not exist, it is set to zero.

For each time step, after the washout period, the extended system states $z(n)$ and the teacher signal $d(n)$ are collected. The washout period includes those time steps just after the presentation of an input signal to the network where the systems extended states are discarded and not used in the training. This is to wait for the network to settle and the internal dynamics to stabilise and the network to settle to the input provided.

The extended states are collected in a matrix S of size $[n_{max} \times (N+K)]$ and the desired outputs $d(n)$ in a matrix D of size $[n_{max} \times L]$.

Now, the desired output weights W^{out} can be calculated as follows. First, the correlation matrix of the extended system states is calculated, $R = S'S$. Then, the cross-correlation matrix of the extended states against the desired outputs d, $P = S'D$. Finally, the calculation of the output weights of the network W_{out} is done by calculating the pseudoinverse of S, $S^{\dagger}$,

$$W^{out} = (S^{\dagger} D)' \tag{5.1}$$

5.2.3 Intrinsic Plasticity

Selecting the spectral radius of the reservoirs weight matrix is one of the most important parameters while using dynamic reservoirs. Intrinsic Plasticity (IP) provides an unsupervised method for the adaptation of the Dynamical Reservoir [133, 134]. The idea is that the activation functions of the neurons are adapted to fire under a certain, usually exponential, distribution. This results to sparse activations of the reservoir neurons, with each one capturing only important features of the input signal. The IP rule is local in space and time and aims at maximizing input to output information transmission for each neuron.

In our case, where the training data is noisy, IP is show to alleviate the overall performance helping in the decorrelation of the noisy input signals in the training procedure.

Using a hyperbolic tangent as an activation function for the reservoir's neurons, the intrinsic parametrisation can be derived by adding a gain *a* and a bias *b*, to the activation function $f'(x) = f(ax+b)$ and now working with f' as the activation function of the reservoir's neurons. Then, the online adaptation rule of IP according to [135] is derived to be,

$$\Delta b = -\zeta(-\mu\sigma^{-2} + y\sigma^{-2}(2\sigma^{2} + 1 - y^{2} + \mu y)) \tag{5.2}$$

$$\Delta a = \zeta a^{-1} + x\Delta b \tag{5.3}$$

where ζ is the learning rate for the IP, μ the mean of desired activation distribution and σ^2 it's variance. All signals, x, y and parameters a, b are of the same time step n.

5.2.4 Parametrisation of the system

The matrices W_{in} and W_{outFb} have 10% connectivity and are initialised in ranges $[-0.9, 0.9]$ and $[-10^{-4}, 10^{-4}]$ respectively. The DR matrix W has a 20% connectivity and is adapted through the IP rule, needing no explicit *spectral radius* setting. The parameter α in state calculation is set to 0.5 for both training and usage of the network. The size of the DR was chosen to be $N = 128$.

For the IP learning rule, the learning rate is $\zeta = 0.0001$, the mean $\mu = 0.0$ and the variance $\sigma^2 = 0.8$. The gain parameter a is initialised to unity, while the bias parameter b to zero.

For the adaptation of the ESN the training sequence is presented to the network and the IP rule is applied according to 5.2 and 5.3. Then the sequence is presented once more and the collection matrices S and D are created and the output weights W_{out} are calculated as described above in 5.1.

The optimal configuration was achieved by repeated experiments, although there exist methods for automatic or semi-automatic fixing of the parameters. The autonomous adaptation of the reservoir through the IP rule allows for a variability in the setting of the parameters, since the reservoir neurons are adjusted to have a maximal information transfer for the given input signal.

5.3 Experimental setup

5.3.1 Technical details

For the testing of the system a Leap Motion sensor was used, as seen in figure 4.3a. The system is initialised as described above for the input device. The ESN architecture described above was coded in Python using Theano [136]. A client-server model was implemented to provide the connection between the input device and the learning algorithm (i.e. ESN).

5.3.2 Input signal

The Leap Motion device is a sensory device providing tracking and skeleton data for hand and fingers positions in space. Using the *JavaScript* library provided by the manufacturer, and the client-server setup described above, we recorded six, 6, values to describe the hand position at each frame. The values recorded represent the 3 rotational and 3 translational DoF of the centre of the palm of the hand. The setup allows us to stream the input signal through the network at the sampling rate of the device i.e. $> 30fps$. A frame rate as high as 100fps was able to be produced, just for testing purposes of the setup.

There is no sub-sampling performed, nor for the training set acquisition nor for the testing phase. The device is sampled at each time step, and the sample is directed to the server side where it is fed to the ESN. The ESN provides an output for each time step, recorded and used for the analysis of the performance of the system in the results section.

5.3.3 Testing Cases

For the testing of the proposed system we have worked as follows:

- We tested the validity of the proposed method on a publicly available dataset, the Cornell Activity Dataset (CAD-120) [137]. This, in order to assess the quality of the work presented and to provide evidence of the generalisation capability and flexibility of the proposed method. Although CAD is rather distant to the application field of the proposed method, it allows for the comparison of our method against a baseline, while also shows the applicability of the setup regardless the type of input signal. The results of our method are reported and compared to alternative state-of-the-art computational methods.
- In order to investigate the system in more detail, an extra set of sequences was recorded by the experimenter using the Leap Motion as input device. This has produced a new dataset on which the proposed system has been further tested,

labelled in this work as (*Dataset Testing*). I this way, we are able to test the system with input sequences more applicable to our specific interest of robot control and also highlight general characteristics of the system.

- A small number of users were asked to control a simulated robot, visible on the screen of a computer, using the proposed system and the Leap Motion as input device. We refer to this test within this work as *User Testing*. In this phase of the testing, the users were asked to perform gestures using the Leap Motion device, in relation to behaviours of the simulated robot shown by the experimenter on the computer screen. It is important to mention here that the users were not instructed on the kind of gestures they should use in order to control the robot, allowing them to freely manipulate the input device at their own preferences. This resulted to different gestures being used by the users in relation to the same robot behaviour shown to them. This fact indicates the flexibility of the proposed system in personalising the control sequences and the associations between the user's gestures and the robot behaviours. Since the gestures performed by the users were different for each one of them, based on their preference, only the accuracy of the system is reported under this setup. Once the ESN was trained with the gestures performed by the users, they were asked to control the robot using their own provided gestures.

CAD-120 Testing

The CAD-120 is a publicly available dataset with the recording of skeleton data of 10 daily activities: *making cereal, taking medicine, stacking objects, unstacking objects, microwaving food, picking objects, cleaning objects, taking food, arranging objects, having a meal.* These activities are performed by 4 different people and each is repeated 3 or 4 times. For each person and repetition, a time series of the skeleton data is used as input for the network. Although the dataset offers a confidence value for the skeletal data at any point, all have been used regardless, since ESN are known to work well with noisy data. For the reporting of the accuracy of the method on the

CAD-120 dataset, a leave-one-person-out cross-validation scheme is used as found in the literature [137, 138]. Given the application domain - iterative robot control - from the dataset only the skeletal data were used as input (i.e. avoiding ground truth labels and objects in the scene), including erroneous entries. The accuracy is reported as the mean of the respective accuracies for each person. Following the accuracy reporting scheme found in literature, for each activity presented to the network the readouts were averaged for the whole length of the sequence.

Dataset Testing

In order to test the system in a setup more similar to its intended functionality the *Dataset Testing* was created in house by the experimenter. This dataset consists of 7 generic dynamic hand gestures performed using the Leap Motion device. In this case a different measure is used to calculate the accuracy, in order to highlight the mapping paradigm under which the method is used. We report the percentage of time the network output is indicating the correct input sequence presented, since we assume no segmentation of any sort of the input sequences, neither logical (e.g. by performing a moving average of the output), nor physical (e.g. by removing the hand from the Leap Motion's recording area before and after the performance of the gesture). The Leap Motion was selected for the testing as its larger input size is more demanding for the system. Each input gesture was repeated 3 times, the system was trained using two sequences out of the three and tested on the third, unseen, one. Thus, the accuracy is reported based on a 3-fold cross validation methodology. Each gesture of the training set includes the preparation, the nucleus, and the retraction of the gesture without tagging any of those moments [139, 140]. That is, each gesture includes the positioning of the hand within the device's receptive field and its removal. There was no care whether each execution of gestures was starting from the same point, nor that it had the same time span, nor that it was performed in the same manner, so to follow the exact same shape every time (e.g. performing a clockwise rotation of the same radius for the 3 times). This has been done in order to account for spatial and time variability between the input sequences.

The actions performed by the experimenter with the right hand within the range of the Leap Motion are the following:

Push The hand moves forward from the centre of the receptive field in the horizontal plane;

Pull The hand moves backwards from the centre of the receptive field in the horizontal plane;

Swipe-right Repeated swipe movements from the centre to the right of the receptive field in the horizontal plane;

Swipe-left Repeated swipe movements from the centre to the left of the receptive field in the horizontal plane;

Clockwise Circle the hand moves repeatedly clockwise in a circle within the receptive field in the vertical plane;

Anti-Clockwise Circle the hand moves repeatedly anticlockwise in a circle within the receptive field in the vertical plane;

Up-Down the hand moves up and down within the receptive field in the vertical plane.

It is important to note that sequences varies in length, as seen in table 5.1, which presents the 7 gestures recorded for the testing together with their length. Furthermore, it is also interesting to note that for the system to be able to discriminate between sequences 4, 5, and 6 it should be able to follow their ongoing dynamics. In fact, the higher and lower hand position in gesture 6 can also be found in gesture 4 and 5, as they are part of the circle described by the hand on the vertical plane. Similarly, gestures 2 and 3 share some of the hand positions with gestures 4 and 5, since the leftmost and rightmost points also belong to the circle described by the hand on the latter gestures.

Table 5.1: The 7 gestures recorded for testing. Their description is provided in the text. The sequence length is given in frames captured by the input device

Description	ID	Sequence length
Push	0	150
Pull	1	195
Swipe right	2	144
Swipe left	3	129
Clockwise Circle	4	225
Anti-Clockwise Circle	5	147
Up-Down	6	147

User Testing

In this final stage of the testing, eight participants were asked to perform gestures that they would deem appropriate in order to control 4 simple robotic behaviours shown to them with a simulated mobile robot. The robot selected for this test was a simple 2 D.o.F. differential drive mobile robot. That is, 2 drive wheels are mounted on a common axis and each wheel can independently be driven either forward or backward. A set of 4 behaviours were implemented on the robot: Forward, backward, clockwise rotation and anticlockwise rotation. The users were asked to perform their own set of input signals for these behaviours and then control the robot in a continuous fashion using their own generated signals. The system does not require that the users segment their input gestures, with transition between the input sequences being handled by the network's dynamics autonomously. The recording of the gestures was done in the same fashion as described in the previous section. The only relevant difference was that each gesture was performed only once for the training of the ESN. The participants were also asked, at the end of the testing, whether they realised any lags in the executions of the commands they sent to the robot.

In all cases where the system was used, each time point of a gesture performed is recorded, placed in a bucket and labelled with an index at the allowed frame rate of the Leap Motion. Once all gestures are performed, the network is trained, following the procedure described in section 5.2.2. The machine used for the training and testing of the system, in both test cases, was a mid-range laptop with an Intel Core i5-3340M

CPU @ 2.70GHz ÃŮ 4 (2 cores, 4 threads), with 3.7GB of RAM and without the use of any GPU acceleration methods. The training procedure took less than a second < 1 sec in all cases, even for the larger testing set.

5.4 Results

Results are split in three sections for the three test cases used. First the results from the *CAD-120 Testing* are reported, followed by the *Dataset Testing* and finally the results from the *User Testing*.

5.4.1 CAD-120 Testing

In table 5.2 the accuracy of the proposed method is reported. The skeletal recordings for each activity are used as input, while at the same time, all the available data regardless of their corresponding confidence value are used. The confidence value is available in the dataset and reports whether the given skeletal pose at a given frame is valid or not. For comparison the best results found in bibliography are reported for which the same input was used, that is, skeletal data without ground truth labels, or objects in the scene.

Table 5.2: Accuracy of the method on the CAD-120. * Koppula et.al reports on results with information about the objects in a scene.

Method	Accuracy($\%$)
[141]	70.2
[137] *	75.0
ESN	73.5

Although not many research reports classification results excluding objects in the scene, we can observe that the method presented here is able to achieve comparable performance.

At the same time, results show how the ESN architecture is able to handle the vastly different lengths of the recorded activities in the CAD-120, which varies from 150 to 900 frames.

In addition, the unsupervised adaptation of the reservoir through the IP rule, allows for a parametrisation of the network specific to the input sequences. Indeed, we observe that, because of the IP rule, a much smaller reservoir of only 128 neurons can be used, compared to the 300 units reservoir reported in [138]. We believe this is possible thanks to the IP rule, by which the activation function for each neuron is adjusted to maximise the information transfer. Furthermore, locality in time and space makes the adaptation computationally efficient [134].

5.4.2 Dataset Testing

After running the ESN according to the training procedure described, the system was always able in every case to converge and to find the right set of output weights for the task. Once the system is trained, sequences are then presented in random order to test the accuracy of the training. The network provides a response (output) for each time step an input is provided. Comparing the output with the gesture performed, we measured an accuracy of 87.8% for all the gestures performed. That is, 87.8% of the time steps an output was generated, it was indicating the correct gesture. It is worth to note here that this measure cannot reach 100% accuracy, since the ESN needs some time to stabilise its output for the input signal. For a more stable measure, the output of the network should have been segmented and observed only after the stabilisation. However, since we do not want to use any arbitrary set of parameters to judge the stabilisation point, we proceed with this holistic measure in the reporting of the results. Comparable performances, using less gestures, have been reported by Weber [142]. It is to be noted, however, that in Weber's work gestures have a starting and ending points, that we have not included in our work, to avoid any arbitrary interpretation of the gestures.

In table 5.3 a more detailed representation of individual results obtained for each sequence are presented. During testing, each gesture is recognised during the exhibition. What we present in table is the average of correct recognitions for all time steps each pattern is exhibited. Since the system is meant to provide a continuous output for every point of the sequence provided in input, we measure the percentage of correct

Table 5.3: Training and testing accuracy scores for the 7 gestures. Score is measured as the mean of recognised time steps for each gesture.

Description	Training	Testing
Push	.99	.99
Pull	.99	.99
Swipe right	.99	.98
Swipe left	.99	.72
Clockwise Circle	.99	.89
Anti-Clockwise Circle	.99	.70
Up-Down	.96	.88
Mean	.98	.87

recognition in time, as the input sequence is presented to the ESN. It can be seen as a measure of the correct mapping between input and output in time.

5.4.3 User Testing

As a final step of the testing phase, the system was finally exposed to users. That is, eight people were asked to use the system and control the simulated wheeled robot without a specific task. Their only goal was to control the robot in the way they wanted.

Results in this case are very similar to the ones observed with the *Dataset Testing* condition. The ESN was able to find the right set of output weights for all sequences provided by all height users every time. Although the input sequences recorded by the users where completely arbitrary and very different in terms of overall length and gesture patterns, the proposed architecture was able to cope with the incoming signal and mapping it to the output. Notably, the overall training performance was significantly increased with respect to the previous testing, since the network had only to distinguish between four input patterns, i.e. the four gestures associated to the four pre-coded movement of the robot.

Table 5.4 shows the lengths of the 4 recorded input patterns (gestures) from the participants. From the table we can observe the high variability of the gestures in term of length of the input and to also, therefore, highlight the capability of the setup to deal with different lengths. The input device was sampled at the maximum allowed frame rate (i.e. 100 fps) with the length of each sequence being the number of frames recorded

Table 5.4: The length of each of the 4 gestures used by each participant are presented. The table displays the variability of the length of the gestures. Fluctuations of the gesture lengths are observed between participants and between gestures. We can clearly observe that there is no particular tendency in terms of the lengths of the selected input sequences. The length is measured by the frames recorded for each sequence, with the average frame rate of the device being 100 frames per second.

Participant	G1	G2	G3	G4
P1	2613	841	975	1142
P2	210	180	192	121
P3	721	619	360	701
P4	205	409	384	602
P5	187	155	68	101
P6	207	128	203	266
P7	604	614	436	596
P8	241	521	522	492

from the user.

Furthermore, in order to provide a qualitative appreciation of the variability observed between user gestures, figure 5.1 shows the training set (i.e. the recorded sequences made by the Leap Motion device) of 3 users depicted within the 3 respective graphs. Each figure shows 6 lines representing the 6 D.o.F of the input device during the recording of the user. Those recorded values are then fed to the ESN as input. In each figure, the separation of the four sequences, representing the four gestures made by the user, is indicated above the graphs with the label G1, G2, G3, and G4 respectively, referring to the forward, backward, clockwise, and anticlockwise movements of the robot. By visually comparing the patterns for the three users it is possible to appreciate their differences, both *within* the same user and *between* users. As already seen in table 5.4, the differences in length of the sequences are noticeable. Moreover, it is also possible to appreciate the difference in which users have decided to associate their gestures to the four robot behaviours. Some users preferred periodic movements for all their input sequences (e.g. $P6$), while others chose more stable and non-periodic movements (e.g $P8$). At the same time, as shown in $P7$, the system was also able to handle cases were periodic and non-periodic input behaviours were mixed by the user.

After the test, each user was asked to respond to a questionnaire, in order to investigate

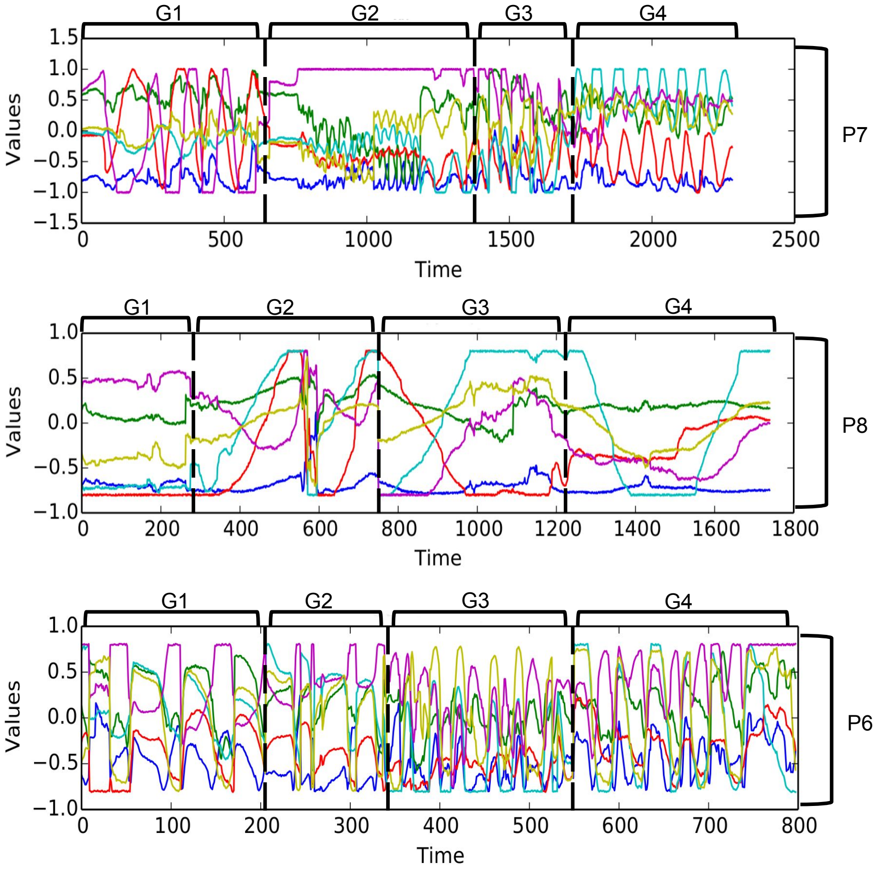


Figure 5.1: Three examples of four gestures, input behaviours, from participants P6, P7, and P8 used for the training of the ESN. The four different gestures are labelled with G1, G2, G3 and G4 at the top of each figure. Lines represent the 6 dimensions of the input signal of the Leap Motion device, plotted against time. It is possible to appreciate: (a) the visible differences in the quality of the input sequences and (b) the different lengths in time. That is, the different span along the x-axis.

the quality of the interaction and the feasibility of the methodology proposed. It is to be noted, in fact, that given the characteristic of the task presented the subjects, it is not possible to disentangle the input sequences performed by the users at run-time with the corresponding output and isolate the single gestures recorded during the training phase by the users itself, in order to make a comparison. This makes impossible to assess the accuracy of the network in the same way as it was done for the *Dataset Testing* condition. This is also the reason behind the creation of the *Dataset Test*, i.e., to have tangible and quantitative proof of the actual works of the ESN.

Besides the specific analysis of the responses, which is not central for this work, all of the users did not report any delay in the systems response. Seven subjects out of the eight reported that they felt in control of the robot by using the Leap Motion device. This indicates that the ESN was able to map their input signals in the corresponding robot behaviours, as they were expecting. Also, all users reported that the network's training time was short, most of them having not noticed it, and the training procedure short enough, having to perform only one repetition for each control signal.

5.5 Properties of the Echo-State Network and human-machine interface

During the running of the ESN and the tests that have been performed, a number of observations have led to a more detailed investigation of some aspects that represents particularly interesting features for the field of human-machine and human-robot interaction. Such interesting features that the proposed ESN shows regards the way in which it solves problems concerning the variability of the length of the patterns to be classified, the complexity posed by the real-time processing of the input streams and the huge amount of noise, which is typical of the raw data that we use as input for the system.

From the same perspective, the next sections present some of the properties that we have discovered by analysing the trained ESN. Those properties, together with the above features, we believe can have an interesting impact in the way in which a system like the one presented here can shed new lights on the construction of flexible interfaces between human and machines.

5.5.1 Variability in Pattern Length

As mentioned in section 5.4 the training patterns varied in length. This is a characteristic of all actions and behaviours performed by humans in real life and is also evident both in the CAD-120 dataset and in the dataset created by ourselves. Therefore, this is a fundamental problem that a human-machine interface has to face. Indeed, not confining the patterns to be of equal time scales, which will be artificial, allows for greater freedom for the user and enhances the robustness of a system trained on raw user data. The immediate benefit is that there is no need for explaining to the user the way in which the interface works. At the same time, the user will be free to behave in natural and intuitive way.

The high degree of recurrency within the DR allows for temporal dynamics of different time scales to be recorded and retained, without any explicit specification on the duration of the patterns. Both the number of neurons and the spectral radius of their connecting weights accounts for the memory size of the DR. In this work we have shown that it is possible to adapt those parameters in an unsupervised manner using the IP rule, allowing for the DR to adapt to the input behaviours. In this way we can obtain at the same time a general architecture capable of recognising sequences of different lengths and a specific adaptation towards a specific training set provided by the user.

5.5.2 Continuous Mapping from the Raw Data Input

Dynamic actions, such as hand gestures, generally contain three phases that overlaps in times: Preparation, nucleus, and retraction [139, 140], of which the nucleus is the most discriminative. The setup proposed is able to capture the discriminative part of a gesture without any explicit instructions about its location within the overall gesture performance. Thanks to this feature, transitions between gestures can be handled autonomously by the ESN. In turn, user input sequences do not need to be artificially and purposefully segmented, allowing for the continuity and natural flowing of the input to be preserved in the ESN output. It is this property of the setup that allows for the user to be placed in the loop of the controlled machine, or robot in the case of this work.

5.5.3 Geometrical Properties of the Input

Figure 5.3 shows a detail of three patterns from the condition *Dataset Testing* correctly recognised by the ESN : CW, which stands for a clockwise circle pattern performed by the user, ACW an anticlockwise circle pattern, and $Up-Down$, an up and down movement of the hand of the user, as recorded by the Leap Motion device. By observing the first segment of the graph and delimited by the first vertical line in the figure around $Time 100$, we can see that the network correctly recognises a CW gesture in input (the $Value$ of CW in the figure reaches 1.0). Interestingly, the ACW gesture at the same time shows a negative value. This observation can be explained by the fact that the ACW pattern is 'opposite' to the CW patter. Therefore, it suggests that geometrical properties of the input are retained and the spatial relationship between the two signals is captured and embedded from the network in its output signals.

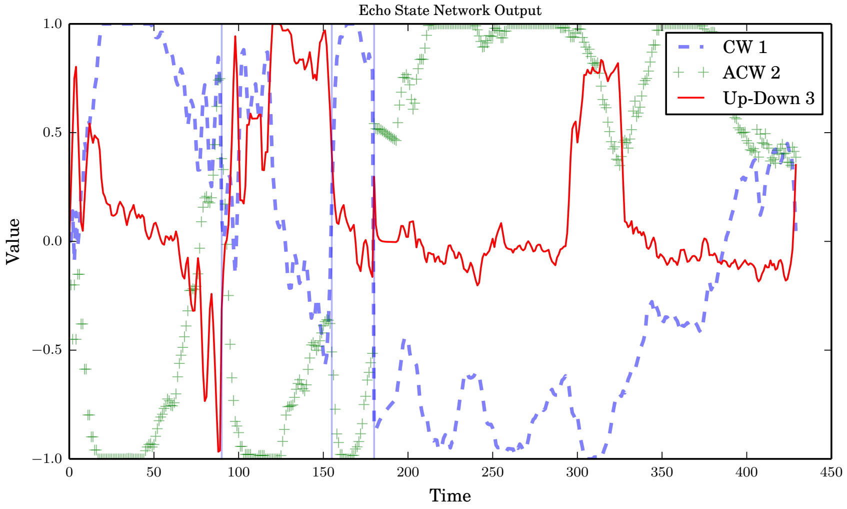


Figure 5.2: Usage of a trained ESN. The plot highlights how the geometrical properties of the input sequences are retained on the output of the network. The antagonistic behaviour between *Clockwise* (CW) and *Anticlockwise* (ACW) behaviours is shown, while the *Up-Down* motion recognition remains unaffected. In the graph the output of the network is depicted, with each colour and line style representing a pattern recognised in the input. All values are plotted against time

Similarly, in the next section the input behaviour changes from CW to a mixture of both CW and $Up-Down$, and ultimately to just $Up-Down$ around *Time* 130. This transition is also reflected to the output of the network, but, besides the two behaviours being mixed, ACW remains always negative and opposing the values of CW. This observation indicates that, although it is possible to mix behaviours, it remains impossible to do so with geometrically opposite ones. This is an interesting feature that, to our knowledge, cannot be found in other models. Similar dynamics can also be observed in the following section of the pattern, where the $Up-Down$ recognition settles around 0, the user is performing a ACW gesture and, as expected, the opposite CW pattern is negative.

Such observation indicates that geometrical properties of the input are propagated to the output. In our example, indeed, the clockwise and an anti-clockwise motion inhibit each other. By assuming that the network has been trained to control a moving robots, it is possible to grasp the importance of this feature. For example, lets assume the CW motion is mapped to the robot moving forward and the ACW backwards. Having opposite behaviours being interpreted as ‘opposite’ by the system, it provides the network with an ‘insight’: The user cannot perform two opposite behaviours at the same time, but it can perform the $Up-Down$ gesture in combination with any of the above. At the same time, the fact that the two behaviours are opposite is also maintained in the output. Assuming that the robot behaviours are combined in a linear fashion based on the network outputs, the recognition of ‘move forwards’ implicitly means for the system that ‘move backwards’ will hold opposite values (and negative in the specific implementation presented here).

5.5.4 Recognition Before the End of the Sequence

An important feature of the system presented in this work is that it provides the correct classification before the input sequence is completed. Given the feedback from the output is fed to the reservoir, the network is able to stabilise its dynamics and recognise a given pattern at an early stage of its presentation. This feature allows the system to have a fast response to the sequence in input, making it appealing for real time control cases.

Figure 5.3 shows an example of the recognition of the $pull$ (ID 1) sequence of the dataset. Similar behaviour is also shown by the network for the other sequences as well. That is, the network is able to classify the sequences before their completion. The

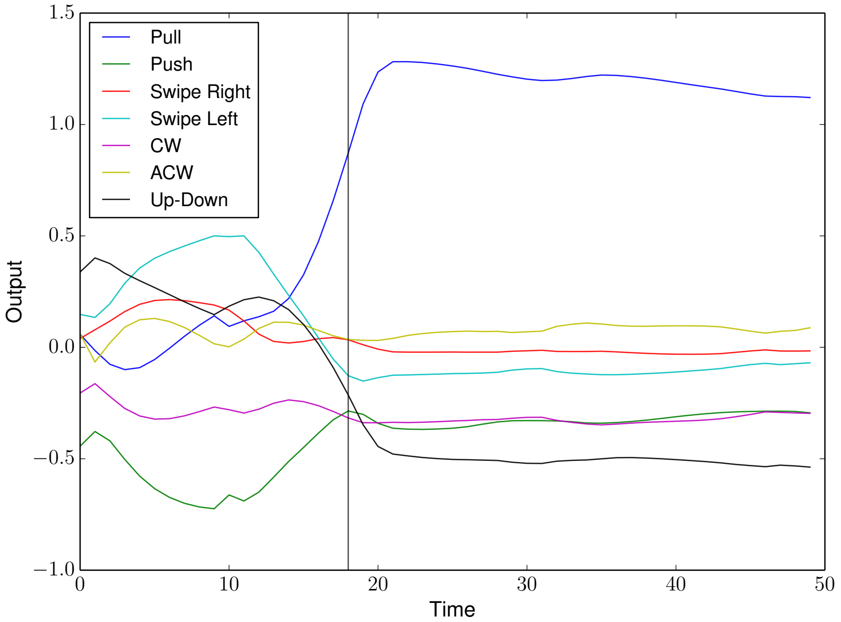


Figure 5.3: The figure shows the recognition of the *pull* (ID 1) input sequence from the initial presentation until the stabilisation of the network's output. In the graph the output of the network is depicted, with each colour and line style representing a pattern recognised in the input. All values are plotted against time

time steps required for the network to settle to a sequence can vary. This is expected as the sequences do not share the same length.

For a more comprehensive way of how the proposed architecture captures the dynamics of the input sequence, we tested the recognition with partial input sequences. Each input sequence was used to artificially create four new sequences, each one having 25%, 50%, 75% and 100% of the original sequence. Each sequence resulted from the initial one having the same starting point but a shorter time span, by omitting the remaining elements of the sequence. In this way, a set of 28 sequences were used for testing. The accuracy of the network is measured in the same fashion as before, reporting the percentage of time the network output indicates the correct input. Each sequence was presented to the network independently, resetting the network in between the sequences presented. At the same time, sequences were shuffled so as to eliminate any of their dynamics to be retained in the ESN's reservoir.

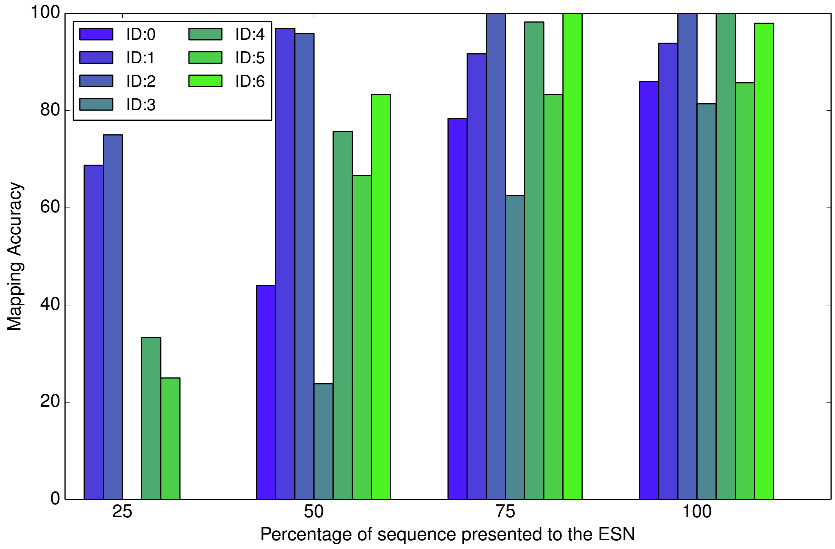


Figure 5.4: Accuracy of the ESN for partially observed inputs. Each colour represents an input sequence. The bars are grouped in four categories, each one representing the percentage of the signal presented to the network.

Figure 5.4 shows the results obtained by the test. From the bar chart it is possible to observe that the network produces the correct answer even from the initial 25% of some sequences (i.e. $IDs1$ and 2). At the same time, for most sequences it reaches a good performance with only half (50%) of the sequence being presented. When the 75% of the input sequence is presented the network is able to recognise all input patterns with a high level of accuracy, with the exception of pattern 3, which is the only one that reaches its maximum recognition rate only when the entire 100% of the pattern is presented.

Given the unified structure for the detection and recognition provided by the ESN, input patterns are detected before their completion. This allows for fast responses from the system, a feature necessary for real time control. It is shown that humans are very sensitive to the response time of user interfaces, with lags greater than 100ms perceived as annoying [143, 144]. Being able to provide feedback within the time span of a given input sequence, that is, during the execution of a gesture, is a challenge that ESN can achieve given the simplicity of the computations performed, which allows for very fast computation in comparison with other methods. Indeed, more complex classification systems perform even more costly computations with similar performances [117].

5.6 Conclusion

In this chapter an echo-state neural architecture for the recognition of continuous time signals is presented, together with a methodology for fast and efficient training. The proposed system is tested under two different paradigms. One to analyse its properties and one to test its real world applications. Through the testing useful properties are highlighted, analysed and their potentials are discussed. Under the scope of human-robot and human-machine interaction the system's applicability is discussed. At the same time the properties of the architecture are discussed independently, in order to allow and encourage usage of the method in other fields.

The findings of this chapter show that pattern recognition in continuous time signals is possible without the computational or algorithmic complexity of methods used so far in the field. The particular time signals considered here are coming from the manipulation of input devices within a human-machine interaction framework. The mapping that the proposed architecture provides was tested under a robot navigation task. In the field of robotics such an adaptive mechanism is shown to provides a just-in-time solution for a user centric system, capable of coupling the user's and robot dynamics in real time.

In the field of human machine interaction a surge in adaptive methods is being observed. Being able to adapt the communication mechanism that relates the machine to the user's preferences, can enhance the usability of the system, the performance of the communication, and decrease the training effort required by the operator in order to user the system [106, 107].

In the field of assistive robotics, ESN can provide a fast and reliable way of adapting the system to the users preferences. This may accommodate cases of increased of decreased mobility and the usage of unorthodox input devices. Being able to capture, train and recognise user behaviours from their preferred input method can be alleviating for use cases that cannot be taken into account in the design procedure.

Chapter 6

Dynamic Behaviour Coupling in Human-Robot Interaction

An experiment with Echo State Networks and the E-puck Robotic Platform

6.1 Introduction

In this chapter we present a novel approach to human-robot control. Taking inspiration from Behaviour Based robotics and self-organisation principles, we present an interfacing mechanism, with the ability to adapt both towards the user and the robotic morphology. The aim is for a transparent mechanism connecting user and robot, allowing for a seamless integration of control signals and robot behaviours. Instead of the user adapting to the interface and control paradigm, the proposed architecture allows the user to shape the control motifs in their way of preference, moving away from the case where the user has to read and understand an operation manual, or it has to learn to operate a specific device. Starting from a tabula rasa basis, the architecture is able to identify control patterns (behaviours) for the given robotic morphology and successfully merge them with control signals from the user, regardless of the input device used. The structural components of the interface are presented and assessed both individually and as a whole. Inherent properties of the architecture are presented and explained. At the same time, emergent properties are presented and investigated. As a whole, this paradigm of control is found to highlight the potential for a change in the paradigm of robotic control, and a new level in the taxonomy of human in the loop systems.

Our vision is for a system that is able to adapt both to the user and the robot, enabling a personalised communication path between the two. The user forms intentions for control and through the manipulations of the input device, communicates them to the robot. The procedure of command learning and recognition is implemented as a mechanism that interfaces the robot and the input device in such a way that, whether the input signals for activating the motor control are captured by an external hardware or acquired by the internal instruments of the robot (i.e. cameras), the system can actively recognise these input sequences and shape the robot's behaviour accordingly.

Most cases of remote robot control are tailored around specific robotic platforms or morphologies [145, 103]. In addition, most studies of remote control are tightly conceptualised around the input device to be used [104, 105]. Changing those assumptions requires a system that can handle multiple robotic morphologies as well as multiple input devices.

Here, a novel framework is presented for the autonomous dynamics behaviour integration between mobile robots and humans. Based on recurrent neural architectures the presented framework is able to generate, differentiate and extract dynamic behaviours from any mobile robot. At the same time, a novel paradigm of control is presented together with a novel adaptation technique for the user. Instead of the norm in control systems, the paradigm is shifted from classification to mapping, and thus robot and user dynamics are coupled to form the control patterns. Moreover, differently to most, if not all, available remote control systems, the robot is able to understand the user's intentions for control through the interaction of the two dynamics, thanks to the available sensors. In practice, the framework is able to (a) stimulate the user's intention for control by offering a set of pre-formed robot's behaviours, (b) capture this intention, and (c) store it in a efficient way, not only allowing reusability but also intuitive combinatorial between behaviours, as well as generative capabilities for adapting and creating new robot and user behaviours. To this extent, the novelty of the presented work sits within the context of the situated and embodied cognition paradigms, as well as within the behaviour based robotics approach to implement the robot control, indeed, both user's and robot's behaviours are strictly connected to the characteristics of the environment, of the robot morphology and of the input device, which are in turn entangled with the user's motor capabilities. Therefore, the novelty of the presented work lies in the unification of action, perception and intention in a rigorous analytical way, using time dependent methods, effectively providing a dynamical integration of all three.

In what follows we describe the method for autonomous acquisition of behaviours, interpreted in a modular fashion as in the case of behavioural-based robotics, formed through the dynamic interactions of the robot with its physical surroundings; and the method to perform the mapping of these behaviours to the relative input signals exhibited by the user. The former method is based on a dynamical system approach and a principle of self-excitation, namely homeokinesis. The latter method, based on Echo State Neural Networks, is capable of adapting to the dynamics of the input sequences and provides a robust mapping from the input space to the behavioural space of the robot. The methods used and the performance of the system are discussed and investigated in detail. The overall characteristics of the proposed framework are presented in detail in the 'Results' section.

6.2 Analysis of Existing Literature

6.2.1 Human Centric Systems

Our research is inspired by the fields of *human-machine* and *human-robot interaction*, as well as *self-organisation* concepts, with respect to embodied cognition. All those fields are brought into focus under the lights of cybernetic principles, where the system to be controlled, i.e. the robot, and the input system for the user, are both interpreted as complex systems dynamically interacting and coupling their behaviours. According to [146], control systems in general, fall into the category of 'Type_001 Cybernetics'. This type of cybernetics studies the cases were a self-governed system is governed from within by a single-self subject. In most systems of this type, there can be found two types of information flow. A cognitive flow, i.e. the quantitative information available by the system through its sensors, and a subjective flow, i.e. the experiential factors processed on the 'mind' of the system itself.

Human-Machine Interaction In Human Machine Interaction (HMI) interfacing mechanisms between the operator and the device to be controlled are tightly formed around the application field and the machine. To do so, the communication is mediated by an interface between the two systems. The design of the interfaces and the interaction enabled by them are mostly studied in the field of ergonomics [49]. In terms of HMI, ergonomics relates to how the user will interact with a machine and how easy that interaction will be.

Human-Robot Interaction Human-robot interaction is fundamentally different from typical human-computer interaction (HCI) in several dimensions. HRI differs from both HCI and HMI because it concerns systems showing complex, dynamic control systems, exhibiting a variable degrees of autonomy and cognition, and typically operating in changing, real-world environments. In addition, differences can be traced in the types of interactions (interaction roles); the physical nature of robots; the number of systems a user can simultaneously interact with; and the environment in which the interactions occur [147].

Most studies on interfacing mechanisms for remote control of robotic morphologies are conducted using a fixed input device. Ellis et. al. have developed a haptic interface for robot teleoperation [45]. Chao Hu et.al. in [46] present a visual recognition method for mobile robot teleoperation using a camera for identifying human hand postures. Marin et. al. in [47] implement an interface using virtual reality techniques. They implement a multi-level architecture, where different interaction channels are available for the user to communicate their intentions for control. The channels vary from voice commands (top level) to remote programming (bottom level).

Self-Organisation Autonomy in the exhibited behaviours of a robotic system has a key role, as it allows the robot to have an 'understanding' of its own kinematics and dynamics, its morphological constraints and the latent possibilities hidden in its environment. An autonomous robot is capable to anticipate the near future, the sequence of actions required for achieving a desired task and the transitions between them. Self-organisation at the level of robot's controller enables our approach to be agnostic towards the controlled robot, so that the robot itself can generate and 'discover' its own basic behaviours. This although, comes with the drawback that the user has only a limited, or null, role in the creation of such behaviours. A solution to this problem is discussed in the following section.

Affordances and Embodied cognition The concept of *affordances* was first introduced by J.J. Gibson [12]. It described the potential action enabled by an environment or a given object, especially one that is easily discoverable. These 'action possibilities' latent in the surroundings of an agent, need be discovered by the agent itself, providing it with a unique view.

This idea of unique possibilities arising from the same structures, is applicable on the way we perceive control devices. Indeed, different people may have the possibility of acting in different ways upon them. For such a process to be triggered, the human (operator) must have the possibility of freely manipulating the control mechanism. The interaction paradigm and the interfacing techniques should be able to support such activity. Human and robot should be interfaced in a transparent manner, such that supports the user's intuitive interactions. This interfacing mechanism should be able to adapt to the user, accommodating for their preferences, while informing the robot with the minimum possible delay. This idea carries one of the most important aspects of the work described here. As such, it has the potential of enabling intuitive interactions with the operator.

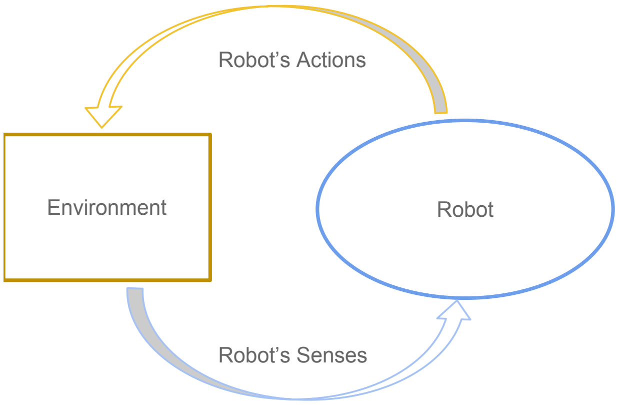


Figure 6.1: The sensorimotor loop of the robot. Behaviours are shaped through this cyclic interaction of the robot with its environment.

6.2.2 Emerging Robot Behaviours

Controlling a robotic system can be a very difficult task, depending on the morphology of the robot. Robots with 1 or 2 Degrees Of Freedom (D.o.F.) can be easy to control, such as simple two-wheeled robots. Indeed, the control can have a comparable complexity of that of a remote controlled toy car. On the other hand, complex arrangements such as 4 or 6 legged robots, or humanoids, can be very difficult to control, especially for non-standard operational tasks (e.g., not simply going forward-backward and turning). In this cases, the designer of the controlling device has to decide the level of expected autonomy of the robot by implementing a series of controlling patterns of various complexity and abstraction, such as high level commands (i.e. proceed to the next room) or low level commands (i.e. arrange a specific joint to certain degrees). In most cases the level of expected autonomy of the robot is driven by the task and the goal.

In the case of robots with no level of autonomy the control is based upon the direct manipulation of the robot's D.o.F.. In the case of remote control, the input device needs to have at least the same amount of D.o.F. so that the operator can achieve full functionality of the robotic morphology [58]. Examples of such control techniques can be found in [64] using a full body mapping or part of it as in [65].

In creating autonomous systems, two are the ways found in literature. First, that of traditional Artificial Intelligence research. Here, a top-down approach in designing

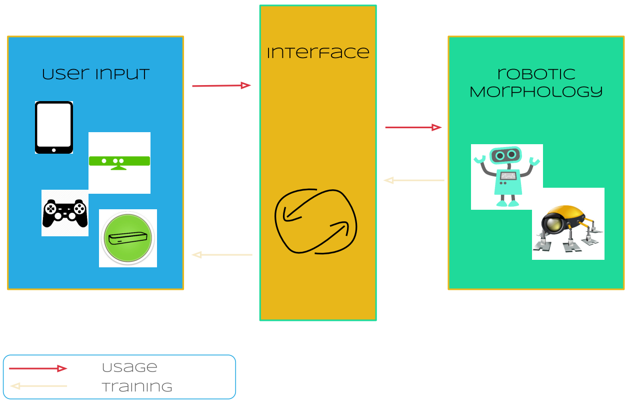


Figure 6.2: Schematic representation of the proposed interface

robot controllers is followed, usually involving a complicated, centralised controller that makes decisions based on access to all aspects of the global state, a view that dates back to 1970 [148]. Second, systems that rely of self-organisation, which could be referred to as 'action driven'. In such systems, build from a bottom-up approach, localized, parallel, and distributed low-level controllers provide the robot with adaptive and complex behaviours. This, based on the assumption that the complexity can be achieved base on the combinatorial effects of small simple behaviours [21].

The control of complex behaviours is said to be achieved through internal models [149]. The internal model is able to identify the expected outcome of an action and the sensory consequences of a motor command. The inverse model, on the other hand, is able to identify the motor command required for the desired sensory state to be achieved. To create such models, the idea of motor babbling [150] comes forward. Inspired by Piaget's suggestion on the stages of human motor development [151], it suggests babbling is the way for exploring the relations between motors and sensors. Despite the fact that the idea of Piaget of purposeless behaviours was later challenged by research showing purposeful exploration from the early stages of development [152], it remains a powerful paradigm in creating autonomous controller for complex robots.

Following this, in robotics similar methods have been proposed for the construction of internal models of behaviours. Under this paradigm, working in model-free case (i.e. not having a complete description of the robot's kinematics), robots are expected to form a model on a *tabula rasa* basis. Indeed, this is referred to as a 'cognitive capability', since this way an expectancy is formed with robot 'knowing' what move is to be performed and when, based on its own state and that of the environment. Paradigms of purposeless exploration have been suggested, through motor babbling in [153, 154]. Robots perform an exploration of their sensory-motor effects, establishing a model based on the expected sensory state produced by a given motor action. On the other hand, a purposeful way of exploring robotic morphologies has been put forward by homeokinesis [155] in rigid bodies, and with morphological computation [156, 157] in compliant bodies. The idea stems from the observation that behaviours can only be explored in a meaningful way if they are grounded on the robots body (sensors), motors, and environment (see figure 6.1).

6.2.3 User Behaviour Recognition

The proposed architecture, on the user side, should be able to understand the users intention for control, operate in real time, and be agnostic towards the input device and the morphology of the robots to be controlled. We understand the user intentions for control, as a series of manipulation sequences of the input device operated by the user over time. Although our problem could be seen as a time sequence classification, the need for real time control, and especially for time sequence combinations, does not allow for standard classification techniques to be used. What we want is a online and flexible mapping between the robot behaviours and the input signals, in the form of a temporal coupling between the two.

For time sequences recognition and combination [93, 94] proposed a Recurrent Neural Network (RNN) working with Parametric Biases (PB). This architecture allows for a mapping of the time sequence in the Parametric Bias (PB) space. The RNN is first trained in the time sequence using Back-Propagation Through Time (BPTT) [93], while the PB units are self organised depicting the differences in the sequence. In the operational mode, the PB are able to capture the present dynamics and move to values close to the trained ones, providing in this way a mapping of the overall RNN dynamics to an n dimensional space, n being the number of PB units. Another architecture has been proposed as an extension, capable of capturing multiple time scales of the time sequence presented to the network [158, 159]. This architecture also uses PB units, in the same manner as above, and it is shown to be able to extract features based on the different scales of sampling of the sequence. Both methodologies use BPTT to train the network. Although, methods for speeding up the training time are strongly required, given that the algorithm for BPTT has a complexity proportional to the length of the training set and the number of nodes of the RNN[160]. Experiments with this type of architectures can be found in [161], where the remote control of the robot behaviours is performed with the manipulation of the Parametric Bias units.

With the aim of combining the effect of multiple time scales and the possibility of mapping the time sequence dynamics to a fixed, and smaller, dimensional space than that of the system itself, [110] formulated the echo state approach on training Recurrent Neural Networks, namely Echo State Network (ESN). ESNs could be seen to work in the same logic as Support Vector Machines, projecting the sequence into a high dimensional space, where the problem becomes linearly separable. One of the most appealing features for our application, is the fact that the network is trained using linear regression on its last layer only, reducing the complexity of training with BPTT. The network is first presented with the input sequence and the values of the output units are replaced with the desired ones. The activation of the network based on the input is recorded and the output weights are computed through linear regression of the desired output on the network's state. Thanks to ESN properties, our proposed architecture is able to learn and adapt towards the time depended manipulations of the input device using an ESN approach.

The entire system therefore works in the following way: (i) Self-generated robot behaviours are exhibited to the user; (ii) the user is required to act on and manipulate the input device ‘as-if’ he was controlling the robot according to the behaviour exhibited; (iii) these input manipulation are captured and the system adapts (i.e. the ESN is trained) to map the time sequences of the user. Once the network is trained, the system can recognise the users commands and apply them to activate the robot accordingly.

6.3 Methods

In this section the methods selected are presented. First, we elaborate on the self-organisation of robotic behaviours and we continue with the input acquisition from the user.

Our vision is for a human-centric system, capable of understanding both the operator (human) and the operated (robot). As such, it can be seen as the cognitive architecture of the robot, capable of seamlessly integrating robots autonomous, self-generated, movements with the controlling intention injected in the system by the user through the input device. The system should be able to place the operator in-the-loop, regardless of the input device to be used and the robotic morphology at hand. For the human side, [162] mentions the importance of prediction in human robot interaction. We want to capture the real time manipulations of the input device by the operator, as they signal their intention for control. These behaviours (i.e. time depended manipulations), are then mapped to robotic behaviours, allowing for the operator to enter in the behavioural loop of the robot.

Under this paradigm, providing a rich and not restrictive repertoire of robot behaviours is essential. Self-organisation of the sensory-motor loop of the robot provides the needed variety and complexity of robotic behaviours [109].

The system should work as a mediator between the robot and the user: an interface connecting the two systems as seen in figure 6.2.

6.3.1 Self-organisation of Robotic Behaviours

In this section the methods for connecting the interface with the robotic morphology are discussed. Our goal is to explore the kinematics and dynamics of the robotic morphology as shaped through the interactions with the environment. This, together with a way of storing and reusing these behaviours found in the interaction of the robot with its environment. Indeed, in these sensory-motor contingencies of the complex system at hand, small independent controllers can be formed [21].

Methodologies for the autonomous exploration of the kinematics and dynamics can be found in the fields of artificial life and self-organisation [163, 164]. The methodology presented below is able to perform both; keeping in mind that the system should be capable of exploring the morphology as fast as possible and also using a modest amount of computational resources. Another constrain is on the variation of the robotic behaviours. Here, since we want the interface to be formed dynamically - based on the interaction with the user - the exploration cannot be driven by imposed goals. As imposed goals we refer to behaviours that emerge under a supervised training. Prototypical behaviours can be useful when the operational task is known in advance. Also, for prototypical behaviours to emerge an externally derived error signal must be used to train the controller. The construction of a teacher signal requires a simulated model of the robot or the operator to physically manipulate the robots in order to perform the action to be used as training set. In [165] the behaviours of the robot are shaped based on it's interaction with a specially constructed environment. In creating behaviours in a supervised manner the dynamics of embodiment are left unexplored. Indeed, the operators assumptions on the dynamics are imposed on the robot.

Conversely, in our proposed method, the interface should be able to capture the dynamic behaviours of the morphology as revealed under an unsupervised -self-organising-manner. In doing so, we remain agnostic towards the robot, where the control mechanism is based on the morphology and not on the designer's idea of the robot. The behaviours formed in this manner should solely rely on the dynamics and kinematics of the robot at hand, making the control 'natural' towards the morphology.

Homeostasis is described as the property of a system which tries to regulate internal values at a certain level. The regulation of the system arises from the negative feedback received by the system. The general idea here is that the system has sensors and actuators affecting it. The desired condition of the sensor is reached through activation of the actuator, based upon a negative feedback loop, i.e. an error. Provided external disturbances, the system can counteract them and maintain an equilibrium.

Based on the same idea, but formulating the equilibrium as part of the system, we can derive a self-referential dynamical system and a principle for self-organisation of robotic behaviours [6, 108]. The idea here is that we try to maintain a smooth control behaviour -instead of an internal variable - keeping the agent at a constant kinetic state. This property of the system -self excitation- gives rise to the name, homeokinesis.

Under the homeokinetic arrangement, learning occurs based on the error between the real behaviour -recorded by sensors- and the prediction of the robot's internal model. That is, the level at which the agent understands the robot's actions in the environment. Based on the homeokinetic principle the sensory motor loop of the controlled robotic morphology is self-organised. From the self-organisation, a repertoire of basic behaviours emerges [7], which we are able to capture in the form of behavioural 'experts'. These experts can be used later on by the operator and combined, in order to control the robot. The behaviours vary in complexity, time, and are entirely based on the interaction between the robot and the environment.

Homeokinetic Control

The self-organisation of the sensory-motor loop of the robot is realised as a dynamical system. For the exploration of the robot's capabilities we work as seen in [7, 22]. We want to be able to produce motor outputs from sensory readings and from them, predict the next sensory state of the robot. The creation of both a sensory-motor and a motor-sensory mapping allows us to derive an error signal for the update of the system parameters. The system is then able to create and adapt its motor-sensory mapping (referred to as the 'World Model'), in real-time, compensating for the misfit on the sensory values. The same error is also used to adapt the 'Controller', the module producing the sensory-motor mapping. This way, we perform an exploration of the kinematics and dynamics of the robots based on the robotic morphology itself.

Moreover, we are able to capture the dynamics exhibited by the robot as attractors formed in the behavioural space of the robot and reuse them. For this, we use a second module operating in parallel with the exploration module. This way, during the real time exploration of the robot's dynamics we are also able to have a series of controllers, in the form of basic behaviours, ready for the user to operate on. By activating each individual controller, the operator is able to manipulate the robot actions, driving the behaviour towards the basin of attraction described by the controller. We also show the ability to combine those basic behaviours, in order to exhibit combinations of behaviours.

The neural networks for the realisation of the above mentioned dynamical system are described below.

Both the Controller K and the World Model W are implemented as forward neural models with rate coding. The two networks working together describe the sensorimotor loop of the robot and are trained according to the homeokinetic principle. The exploration module is described, according to time t, as:

$$\tilde{\mathbf{x}}_{t+1} = W(K(\mathbf{x}_t, C), A) \tag{6.1}$$

The controller K generates motor outputs

$$\mathbf{y}_t = K(\mathbf{x}_t, C) \tag{6.2}$$

as a function of the sensory input $\mathbf{x} = x_1, x_2, \ldots, x_n$, depending on a set of parameters defined by the matrix $C\ [n,n+1]$ and it is defined by the equation:

$$K = g(\sum_{i=1}^{n} C_i x_i + C_{n+1}), \tag{6.3}$$

where g is a sigmoid function.

The world model $\tilde{\mathbf{x}}_{t+1} = W(\mathbf{y}_t, A)$ estimates future sensory input $\tilde{\mathbf{x}}_{t+1}$ from the motor output $\mathbf{y}_t = y_1, y_2, \ldots, y_n$ depending on a set of parameters defined by the matrix $A\ [n,n+1]$.

The parameter matrix of the world model, A, is adapted according to the delta rule [99], $\Delta w = +\eta E_W \mathbf{x}$ with the error, E_W, described by the function:

$$E_W = ||\mathbf{x}_{t+1} - \tilde{\mathbf{x}}_{t+1}||^2 \tag{6.4}$$

with learning rate η = 0.01.

The controller updates its parameter matrix by gradient descent with respect to the error function,

$$E_K = ||\mathbf{x}_t - \tilde{\mathbf{x}}_t||^2 \tag{6.5}$$

To calculate the above error, we find the $\tilde{\mathbf{x}}_t$ by calculating the motor input $\hat{\mathbf{y}}_t$ the world model should have in order to make a perfect prediction and then, the sensory input the controller K should have to predict the motor output $\tilde{\mathbf{y}}_t$. For updating the controller parameters we apply

$$C_{t+1} = C_t - \varepsilon \frac{\partial E_K}{C} \tag{6.6}$$

with a learning rate $\varepsilon = 0.1$. Matrix A is initialised from a uniform distribution in [0.5,1.5], while C in [1, 2].

For the identification, storage and reuse of the different behaviours exhibited by the robot, we use a series of m neural networks (NNs), called *experts*. Each NN is defined according to the equation,

$$(\mathbf{x}_{t+1}, \mathbf{y}_t) = N_i(\mathbf{x}_t, \mathbf{x}_{t-1}), \quad i = 1, \ldots, m \tag{6.7}$$

The NNs, working in parallel, compete for the prediction of the motor command $\mathbf{y}_t$ of time t and the sensory input $\mathbf{x}_{t+1}$ of the next time step in a winner-takes-all method, with only the winning network being allowed to train on the current data $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$. Thanks to this process, each NN specialises to represent a region of the entire sensorimotor space of the robot.

The NNs consist of 3 layers, feed-forward units where the hidden and output layers consist of sigmoid units, and the input layer of linear units. Online back-propagation is used to training the NN with learning rate $\eta = 0.1$. The size of the hidden layer is chosen to be 20. Assuming y to be the output vector of each neural network and x the input vector, we have

$$y = f(W_{hidden}h + b_h) \tag{6.8}$$

$$h = f(W_{input}x + b_x) \tag{6.9}$$

where f is the activation function, chosen to be a hyperbolic tangent. The matrices W_{input} and W_{hidden}, represent the weights from the input to the hidden and from the hidden to the output respectively. Finally, b_x and b_h represent the vectors for the bias units for the input and hidden layers respectively.

In each time step of the simulation the series of NNs are activated with the same input and the one with the best approximation of the next sensor values and motor commands is selected as the winner. The sample is then added to the training dataset of the winning NN which is then trained for one epoch. This way each network specialises in a single different behaviour of the robotic morphology.

The behaviour is efficiently stored in the distributed representation of the neural network being readily available for reuse by activating the neural network. When activating a NN we replace the Controller K and World Model W of the sensory-motor loop with the NN. Thus, the trained network is now producing the motor commands and the sensory predictions for the morphology.

6.3.2 User Behaviour Recognition

Adaptation towards the user is important, as it allows to exploit personalised patterns of communication between the user and the machine. Besides improving user experience, personalised control also enhances the usability of the system, making its usage easier and more intuitive. Adaptivity, in particular, can accommodate the user's needs, whether it is out of preference or necessary for the user itself (i.e. the machine to control has more degrees of freedom than the user, or the user can only benefit of a limited range of movements). The challenge in this case, is to create a system that is able to adapt to the user, based on a very small set of training examples, in a short time and be robust in the training.

At the same time, in order to provide a natural way of communication, the system should be able to recognise the sequence in a timely manner from a stream of data. Effectively, placing the human operator in the interaction loop.

Adaptive methodologies capable of showing the necessary behaviours have only started to appear, most of them working under a classification paradigm [106, 107]. The challenges presented here are two: (a) detecting that a sequence is actually present in the data stream received from the input and (b) correctly classifying it. Most research features these two aspects with independent mechanisms [111, 112]. Having a unified mechanism can save computational resources and produce faster recognitions.

Finally, another important aspect of the interaction is time. That is, the time required for the computations of the model to be performed and handling the dynamics of the input signals. Three are the main elements that require attention: (i) for the architecture, to accommodate for patterns of different lengths; (ii) to adapt in a short time, such that the user does not disengage; (iii) to perform the recognition with a low complexity of computation. This is important, as the recognition should take place fast enough for the system to have a timely response for the user.

The task of dynamic sequence recognition becomes especially complicated when working with a continuous streams of data. Breaking down the task, it can be seen to consist of two operations. One is the detection and the other the classification of the sequences. At the same time, the complexity increases when the sequences have different lengths (time spans). Methods used for the classification span from distance measures (e.g. Dynamic Time Warping) [113, 114] and statistical models (e.g. Hidden Markov Models) [115, 116], to artificial neural architectures (e.g. Recurrent Neural Networks)[166, 118, 119, 120, 121] and hybrid solutions [122]. These methods vary in complexity and adaptability, with Recurrent Neural Networks being one of the most prominent direction in the field [123]. Adaptation of RNNs though, is known to have high computational complexity. At the same time, the training procedure is show to have an impasse in finding good solutions, usually referred to as a gradient vanish problem [124].

Working in real world environments can be proven to be difficult and demanding for adaptive models. Performance degrades rapidly when working directly with user data, making most methods not applicable in real world situations. Cleaning data and preprocessing is not a viable option when the demand is for a method that should be readily available to the user. The task becomes even more difficult when the input is sampled in real time and is treated continuously. Not having the ability to segment the input data, thus not having a starting and stopping point, makes the usage of recurrent methods necessary as they can integrate the time signal continuously. On the other hand, training such models requires clean data to perform well, making them difficult to train with data obtained from real users. A potential solution in this case is a structure that is able to capture the internal dynamics of a behaviour (e.g. input sequence) and thus provide a robust recognition.

A recurrent architecture that is shown to work well with noisy data under the restrictions mentioned above is the Echo State Network approach. ESNs are seen to perform surprisingly well with noisy data directly taken from a user interaction and can also adapt rapidly, making their usage for user oriented systems appealing [121, 125, 126, 127, 128]. In our case of behaviour recognition, data comes directly from the user manipulations of an input device. Data can be noisy and the user repetition is not always perfect, resulting to training sets of data with a lot of noise and variation between samples (e.g. gestures, behaviours). The ESN approach followed here provides a stable and robust mapping of the input commands for user behaviour recognition.

Echo State Networks

The mathematical details for the creation of the Echo State Network follow the ones described in 2.2.2.

The network used for our setup has a reservoir of size 300, the spectral radius is set to $a = 0.995$. The feedback matrix is sampled from a uniform distribution in [-0.01, 0.01] and the input matrix in [-0.3, 0.3]. The sparsity of the reservoir, the input and feedback weights was set to 10%.

6.4 Experimental Setup

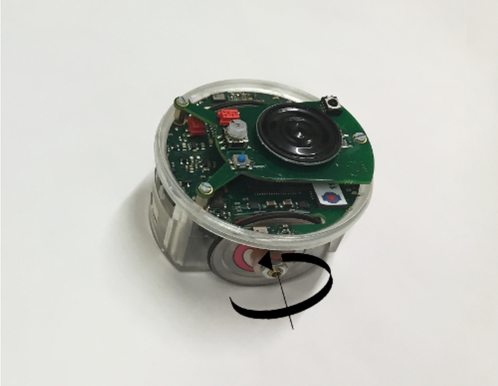


Figure 6.3: The E-puck robot used for the experiment.

The input device used for the experiment and test of the proposed system is the Leap Motion (see fig. 4.3a). It is equipped with two cameras. From these cameras the device creates a skeleton of the user's hand hovering above the device. In our case, the device is placed on a working surface facing upwards, and the user operates in the space above the device. The centre of the user's hand is recorded as input for our experiments. From the data provided from the device only 6 degrees of freedom (D.o.F.) are captured, representing the three rotational and three translational D.o.F of the centre of the hand. These are the 6 values that give the position and orientation of the hand in space, with the Leap Motion device as reference.

The robot to be controlled is the e-puck robot [167], which is a small two wheeled mobile robot. This choice of robot has been made based on its simplicity in order to ease the analysis. For the experiments, a simulated version of the robot is used. The control of the robot is done by adjusting the velocities of the wheels of the robot. Each wheel is controlled independently and can be set to positive and negative velocities, resulting in 2 controllable D.o.F for the robot. As sensory inputs, the positions of the robot's wheels are used. Thus, we work with proprioceptive sensory input to create the kinematic model and dynamic behaviours of the robot.

The proposed architecture works in two stages: (a) the robot self-discovers the behavioural possibilities it has; and (b) the user responds with commands for the robotic behaviours shown using the input device. From the interaction of the user with the robot, the behavioural associations between the two parties are formed. That is the dynamics of the robot's behaviours are coupled with the dynamics of user's actions on the input device. Using the input device, the user's intentions for control are expressed, with the robot changing its behaviour accordingly, following the dynamics in the users behaviour.

6.4.1 Stage 1 - Robotic Behaviour Exploration

As illustrated in figure (6.2), the architecture is placed between the two complex systems: The robot and the input device. On the robot side, the interface captures the behaviours of the robot at a sensory motor level, as a time sequences. On the input device's side, user behaviours are captured as timed sequences of the manipulations of the device by the user. In what follows the robot is the e-puck and the input device the Leap Motion Controller, as said.

The sensorimotor loop of the robot For every time step (t) the sensors of the robot are recorded with a frequency of 100Hz, the homeokinetic module of the architecture produces motor commands, and a prediction of the the resulting sensory state of the robot. In the next time step $(t+1)$ of the simulation the actual sensors are recorded and the time loop error of the homeokinetic control is calculated adjusting the behaviour of the robot. In parallel to this, in every time step (t) the 'expert' neural networks, the controllers, perform a forward pass, predicting the motor commands of time t and the sensory predictions of time $t+1$, of the homeokinetic module. Working in a winner takes all scheme, the network-expert with the best prediction adds the sensor input and motor command of that time step to its dataset, and a trains on its whole dataset once $(1$ epoch).

Through this procedure the robot explores and generates its own possibilities for movement in a structured and self-organised manner. In most research this procedure is addressed using motor babbling [168, 169]. Indeed, under this homoeostatic approach the robot learns to counteract external perturbations and through this interaction learns about its kinematics. However, under this approach the system cannot address the dynamics of the robotic morphology, while at the same time it is heavily dependent of the quality of the external perturbations. Instead we chose homeokinesis, in order to achieve a well structured exploration that is tied to both the robotic morphology and its environment. Through the homeokinetic rule the robot can start exploring its behavioural potentials based on internal perturbations.

The result of this procedure is a set of primitive, basic, behaviours that the robot can exhibit. Each behaviour is stored as a neural controller, becoming part of the robot's behavioural repertoire. As described and shown in [108], these behaviours can be intersected and also combined. Indeed, in their studies it is shown that transitions between them are smooth and so is the resulting robot's behaviour. Lastly, it is shown that these behaviours can be linearly combined to result new, stable, behaviours. Thus, at the end of this stage the robot is able to act in its environment, and also configuring the consequences of its actions to its sensors.

6.4.2 Stage 2 - Training Towards the User and the Input Device

Having adapted towards the robotic morphology, the architecture is now able to adapt towards the user. To stimulate the user, the previously explored robotic behaviour are exhibited by the robot in the simulated environment. The user, while observing these behaviours, responds by manipulating the input device in their way of preference. A schematic representation of the procedure can be seen in figure (6.4). The system does not impose any restriction on the users behaviour, as long as the behaviour is captured by the device. The only feedback given to the user at this stage is a notification that actions are recorded by the input device. Since the Leap Motion Controller does not require any physical contact, the user is informed when they exceed the devices recording radius. Indeed in this stage, the exploration goes towards the user, with them responding to the robot's actions. The architecture captures the user's responses as time sequences and maps them to the robotic behaviours, coupling the dynamics between the input device and the robot behaviour.

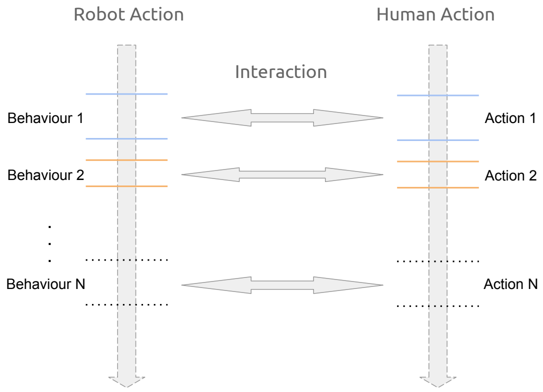


Figure 6.4: Schematic representation of how the user's behaviours are mapped to robotic ones. As the robot exhibits an action the user responds to it with a manipulation of the input device. At the time span of interaction the dynamics of the robotic behaviour are mapped to the dynamics of the user behaviour.

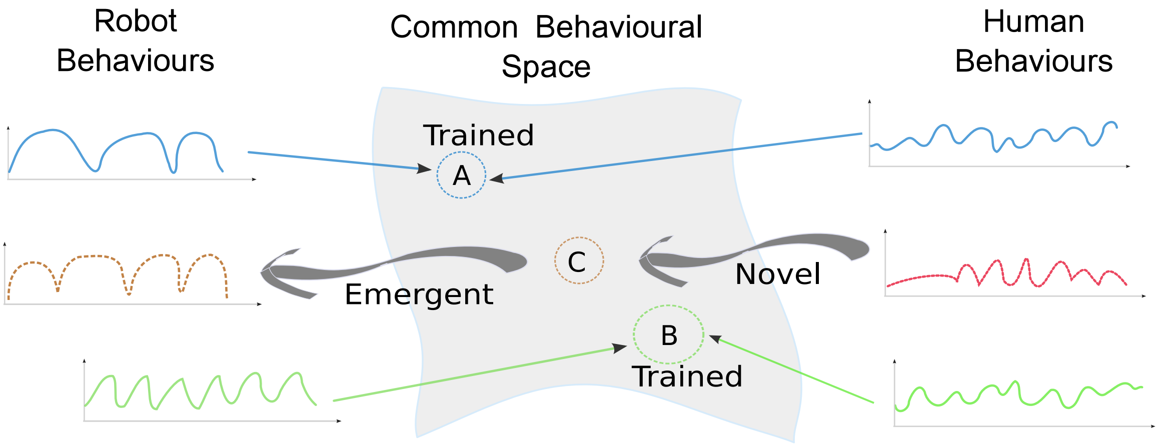


Figure 6.5: Operation of the trained system. The explored robot behaviours A and B (*Stage 1*) are coupled with the behaviours of the human (*Stage 2*). This creates the Common Behavioural space, which robot and human behaviours share. As a result robotic behaviours can be invoked based on the human input (*Stage 3*). At the same time, novel human behaviours can also be mapped to this space (as marked by C), generating emergent robot behaviours, based on combinations of A and B.

For the time span that a behaviour is exhibited by the robot, the input device is recorded and a dataset is created. In this stage we use an Echo State Network (ESN) to capture the dynamics of the input signal. The network is trained, performing linear regression on the output weights of the network for the whole dataset. The complexity of the calculations required is small enough to allow for the training of the network within $1s$. This makes it possible for the network to be trained for each user, as the system is about to be used.

At the end of this stage the architecture is adapted towards both robot and, ultimately, the user. Having the user responding to the robot's behaviours allows for the formation of intuitive control patterns. There is no need for learning from the user, since the architecture is being adapted to suit their control signals. At the same time, the proposed method is able to provide a continuous time mapping from the dynamics of the input device to the robotic behaviours. As soon and as long as the user acts upon the input device the signals are propagated through the ESN, activating the robotic controllers, resulting in a continuous robotic behaviour.

6.4.3 Stage 3 - Controlling the Robot

Having trained both sides of the interface, the system is now ready to be used. The user, manipulating the input device, provides the input to the ESN. The D.o.F of the input device are recorded continuously over time, producing the input sequence to the ESN. Each time step recorded is fed to the ESN, exciting the internal dynamics of the network.

The network output is then used to activate the related basic robotic behaviours. The combination of behaviours is realised as a linear combination of their outputs. Each of the expert-networks, gets as input the sensory state of the robot at time (t) and produces a motor command and a sensory prediction. The motor command passed to the robot is the combination of the motor commands as guided by the ESN's output. A schematic representation of the procedure can be seen in figure (6.5). Based on this arrangement the robot can smoothly switch between a controlled modality and an autonomous modality. Indeed, when there is no input present from the user, the controlling system gains full control of the robot and the robot is then able to self-sustain its sensori-motor loop.

In a more technical note it is important to note that the Echo State Network, the expert controllers (NNs) that generates the basic behaviours and the simulated environment run in parallel, for the above to be achieved. Despite the computational load, the interface is able to perform in frame rate of the input device, without requiring any down sampling. This, because the code has been optimised to work in parallel fashion. For the networks, we use Theano to perform faster, distributed computations, being able to port our code to GPU if needed. For our tests we were able to run the architecture in a machine using an Intel Core i5-3340M CPU @ 2.70GHz 4 (2 cores, 4 threads), with 3.7GB of RAM and without the use of GPU acceleration, in the frame rate of the Leap Motion Controller device ($> 100Hz$).

6.5 Results

The results obtained from the testing of the proposed architecture are now discussed and investigated in detail. The robotic behaviours, the user behaviour recognition, and the behaviour of the system are discussed and investigated closely.

6.5.1 Robot Behaviours

The 1st stage of the architecture's adaptation procedure results to the formation of the modular behaviours for the e-puck. The system works by generating commands in the form of wheel velocities, while using as sensory input only the wheel positions.

Through the homeokinetic adaptation the controllers formed for robot where only four, as expected, based on the low complexity of the controlled robot. We label the four behaviours based on the behaviour we observe on the robot, as seen on following table,

Behaviour	Left Wheel Velocity	Right Wheel Velocity
Forward	1.	1.
Backward	-1.	-1.
Left	-1.	1.
Right	1.	-1.

Table 6.1: The table displays the wheel velocities for the self-organised behaviours of the e-puck robot. The behaviours are the result of the architectures adaptation towards the robot (referred as 1st Stage in section 6.4.1).

6.5.2 User Behaviours

The 2nd stage of the adaptation of the architecture results to a mapping from the Leap Motion Controller to the e-puck behaviours. The user observing the robot responds with controls over the Leap Motion Controller. Based on these input signals the Echo State Network is trained.

In figure 6.6, the responses of the user to three of the four robot behaviours are plotted against time. The user inputs respond to forward, left and right movements of the robot, as seen from left to right. The recorded values from the input device are stored in a six-dimensional vector and for a whole input sequence in a matrix of size $[T \times 6]$, T being

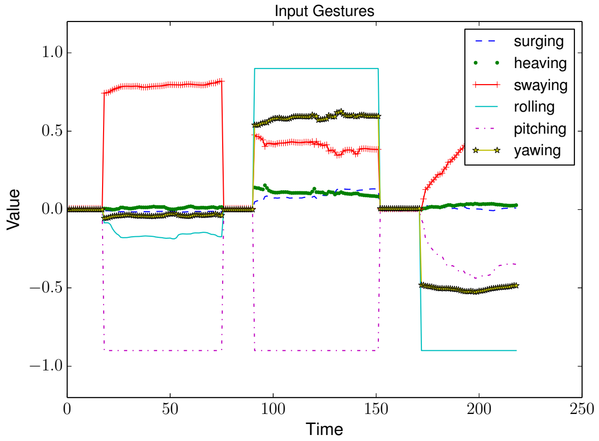


Figure 6.6: User input gestures for the system to train on. Each line style displays a D.o.F. captured by the input device. All values are plotted against time. The three behaviours seen expanding in time were acquired as responses to the forward left and right behaviours of the robot.

the length of the each sequence. There are only three of the four behaviours displayed as the backwards behaviour was not mapped to any input signal. This decision was taken to highlight some of the emergent properties of the architecture.

Pattern length variation A very useful property of the proposed architecture is that it does not impose any restrictions in the behaviour length of both user and robot. This since the sub-modules are designed to incorporate time in a non explicit way. The robot behaviours are stored in independent neural networks, each one having the possibility of storing a behaviour of different length to the others. This variability in the length of the robot's behaviour requires for the user's responses to follow the same variation. The Echo State Network used for the recognition of the user's input is able to handle variable lengths of input sequences and recognise them accordingly.

Simplicity in User Behaviour Capture Echo State Networks have a great capacity in handling noise. This allows for the architecture to capture and adapt to the user input without any preprocessing or special treatment of the input provided through the Leap Motion Controller. This feature of the architecture allows for the behaviours of the user to be captured without them being aware of the inner workings of the system. Rather, empowers them to behave in a natural and free way in the behaviours they exhibit and the input they provide.

6.5.3 Properties of the Architecture

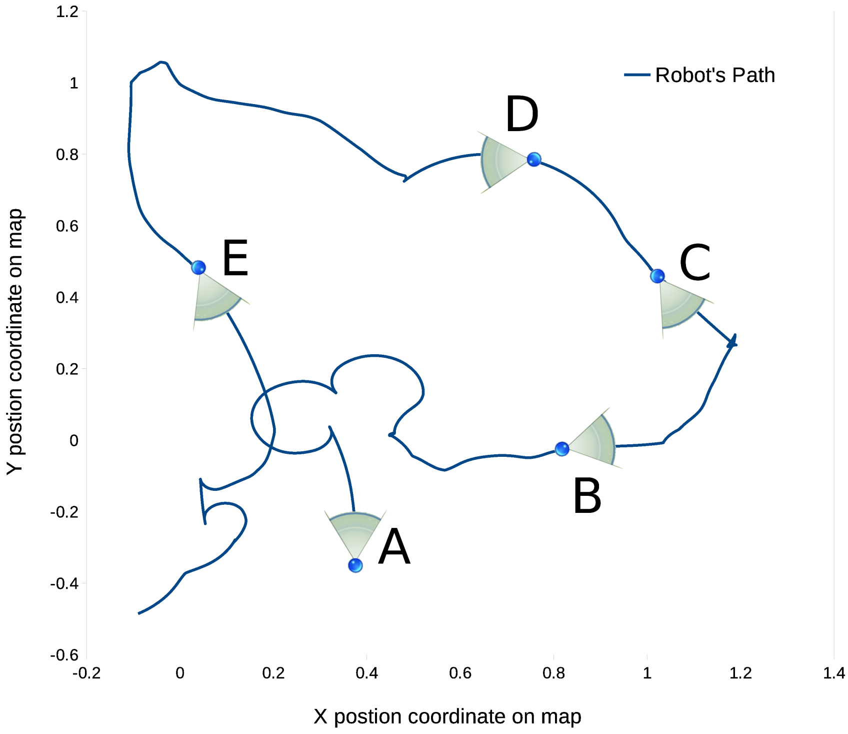


Figure 6.7: Plotting of the absolute position of the robot in the simulated environment during control. The robot is represented by the blue circle and the direction that the robot is facing is depicted by the gray triangle. The five different letters represent the location of the robot at different times. Based on the extracted robot behaviours and their modulation according to the user commands the robot is navigated in the simulated environment.

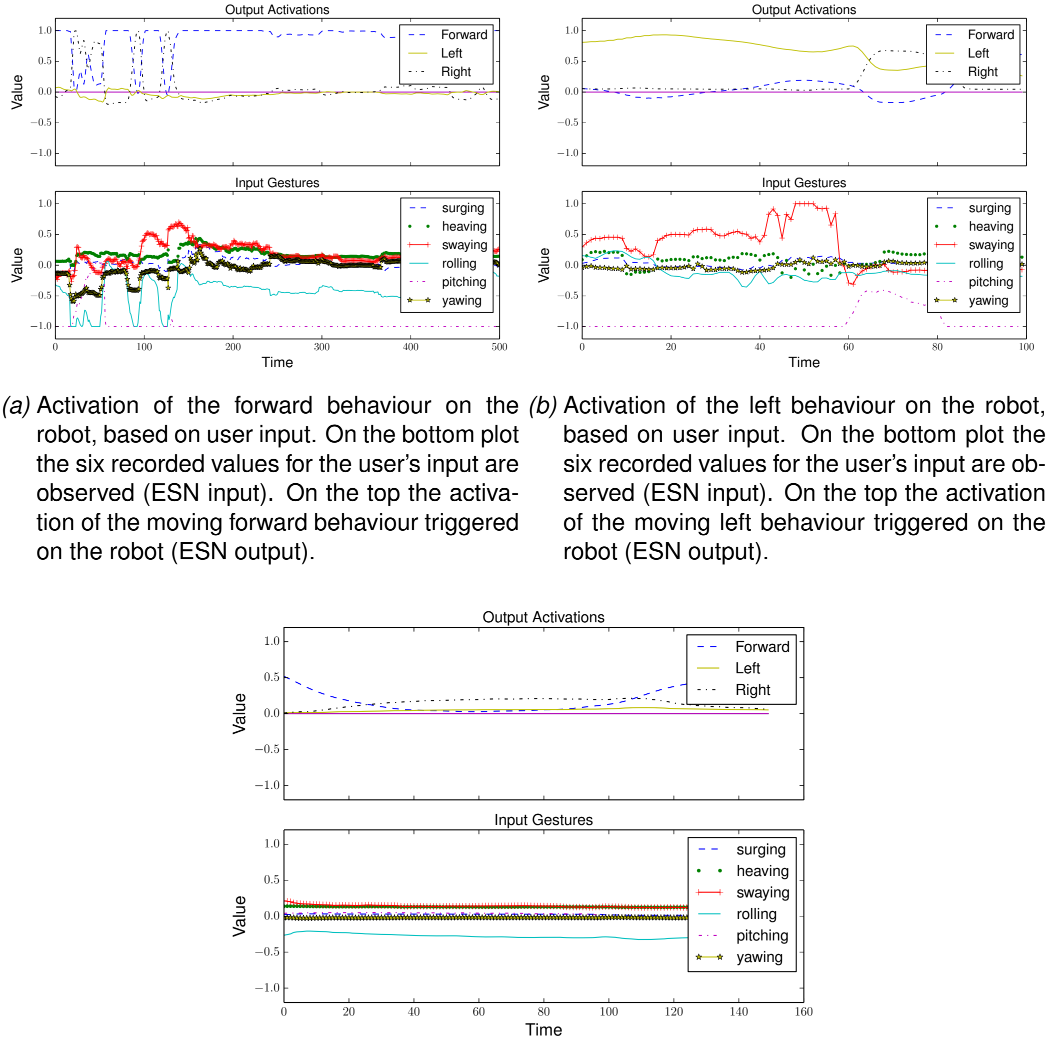
In figure 6.7 a visualisation of the absolute position of the robot in the world is provided, for the duration of the controlled period. The robot is initially placed at point A facing upwards as indicated in the graph. In location B small modulations of the robot's steering, produced by the user, are observed from the path. In location C the robot is moving backwards, exhibiting a behaviour for which the user has not indicated an input signal related to it. This and other emergent properties of the architecture are discussed later in section 6.5.3. Moving to location D, there is a slow left turn exhibiting the ability of the architecture not only to integrate but also modulate the robot's behaviours based on the modulation of the user's input. Finally, in location E a slow right turn is exhibited by the robot, again showing that this modulation holds for all robot and user behaviours and is a valid property of the architecture.

Using only three of the robot's explored behaviours - forwards, left, and right - the architecture is able to produce the missing one based on the inherent properties of the user behaviour recognition module, namely the ESN. The ESN can recognise and propagate the geometrical properties of the input to its output and thus to the robotic behaviours. Indeed, the robot is able to follow this path based on the system's capability for: (a) smooth transitions between robotic behaviours, (b) modulation of the robotic behaviours based on the modulation of the user input. The system is able to produce a smooth trajectory as well as grading wheel velocities based on the intensity of the input signal. Important to note here is the fact that both user and robotic behaviours are exhibited and coupled in real time.

Continuous Time Operation

The system couples user input and robot behaviour in continuous time. The input signals captured from the Leap Motion Controller at each time step are propagated to the ESN sub-module, which in turn, maps them to the robotic behaviours. Each robotic behaviour is realised by its own 'expert' neural controller. These expert are combined at each time step as dictated by the user behaviour recognition module, realised as an Echo State Network. In this section, we investigate the recognition capabilities of the ESN. Based on the user input the ESN should produce at each time step an output indicating the robot behaviours to be triggered.

In figure 6.8 examples are shown of the activations of the robot behaviours. Triggering of the forward (figure 6.8a), left (figure 6.8b), and right (figure 6.8c) behaviours are plotted. In each respective plot the continuous fashion of the input recognition can be seen. For each time step of input values from Leap Motion Controller (bottom plots) an output is generated for the activations of the behaviours on the robot.



(a) Activation of the forward behaviour on the robot, based on user input. On the bottom plot the six recorded values for the user's input are observed (ESN input). On the top the activation of the moving forward behaviour triggered on the robot (ESN output).

(b) Activation of the left behaviour on the robot, based on user input. On the bottom plot the six recorded values for the user's input are observed (ESN input). On the top the activation of the moving left behaviour triggered on the robot (ESN output).

(c) Activation of the right behaviour on the robot, based on user input. On the bottom plot the six recorded values for the user's input are observed (ESN input). On the top the activation of the moving right behaviour triggered on the robot (ESN output).

Figure 6.8: In the three plots the mapping between the user input and the three available robot behaviours is shown. In each figure the top plot represents the behaviour as triggered in the robot and the bottom the user input as recorded by the input device. All values are plotted against time. The time is synchronised between the top and the bottom plots of each figure, showing the real time coupling of user commands and robot behaviours.

Time span of behaviours It can be observed by time span of the behaviours in the plots of figure 6.8, that the network can recognise them even when they are stretched for more time steps that originally exhibited (in the second stage of the architecture's adaptation 6.4.2). This comes as an additional property of the system to the independent time span allowed for each input behaviour. The dynamics of the ESN can be stretched in time following the user's input behaviour and thus trigger the desired robot behaviour for longer.

As a validation of the user input recognition module of the architecture the distances between the behaviours recognised and the trained ones are calculated and shown in table 6.2. The distances are calculated using Dynamic Time Warping [87] as a distance measure, as it allows for the compared timed signals to have unequal lengths. From the table the accuracy of the method is shown as the input behaviour recognised is always the right one.

Train Test	Forward	Left	Right
Forward	**1.20**	2.28	1.87
Left	1.84	**1.11**	2.41
Right	1.45	2.38	**1.30**

Table 6.2: The table displays the distance between the users reference input behaviours (Train behaviours, provided at stage 2, section 6.4.2) and the behaviours recognised by the ESN as Forward, Left, and Right (Test behaviours, exhibited during operation). The lower the number, the lower the distance between the two. In **bold** the smallest value showing the closest behaviour to that of the user.

Transitions Between Behaviours

A very important aspect of the architecture is the transitions between robot behaviours under the command of the user. Having a continuous and smooth transfer from one behaviour to another necessitates the smooth integration of the user's input to the robot's behaviours. Moving a step closer, we also investigate how the transition between behaviours is performed in the motor level of the robot.

Transitions in Behavioural Level The transitions on a behavioural level can be observed from the plots of figure 6.8. Looking closely in figures 6.8b and 6.8c, it is possible to see on the top plots the smooth transitions between behaviours.

More specifically in figure 6.8b between time steps 60 and 80 a change in the input patterns from the user is observed (bottom plot). The 'swaying' measurement goes to zero while the 'pitch' of the hand motion increases. This in the input behaviour from the user is quickly propagated to the output of the ESN changing the behaviour mapping to the robot (top plot). The contribution of the 'Left' robot behaviour is lessened while that of the 'Right' behaviour is increased, becoming the main contributing behaviour (i.e. the one with the highest value).

In the same fashion a smooth transition between 'Forward' and 'Right' robot behaviours is observed in figure 6.8c. Between time steps 20 and 120 the 'Right' moving robot behaviour becomes the sole behaviour exhibited by the robot, having both 'Forward' and 'Left' mapped to near zero values (top plot).

Transitions in Motor Command Level The smooth transitions between robot behaviours can also be observed in the robot's motor values, as guided by the 'expert' controllers. The architecture is able to propagate the transitions observed in the behavioural level to the motor commands of the robot effectively.

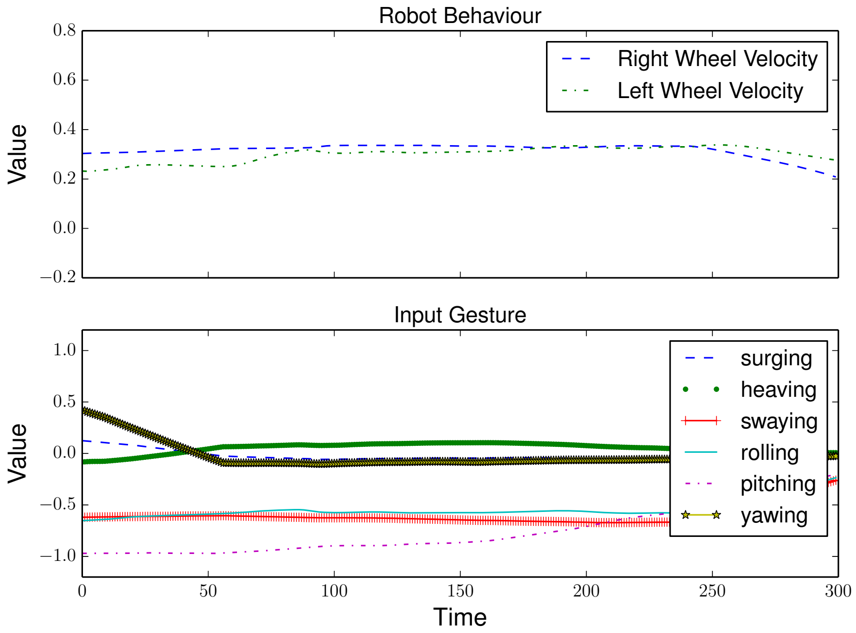


Figure 6.9: Transition from a left moving to a forward moving behaviour on the robot. Based on the change in the user's behaviour the change in the robot's motor commands is observed in the top plot. On the bottom plot the six recorded values for the user's input are observed (ESN input). On the top the motor commands of the robot are observed, measured in left in right wheel velocities ("expert's" control command).

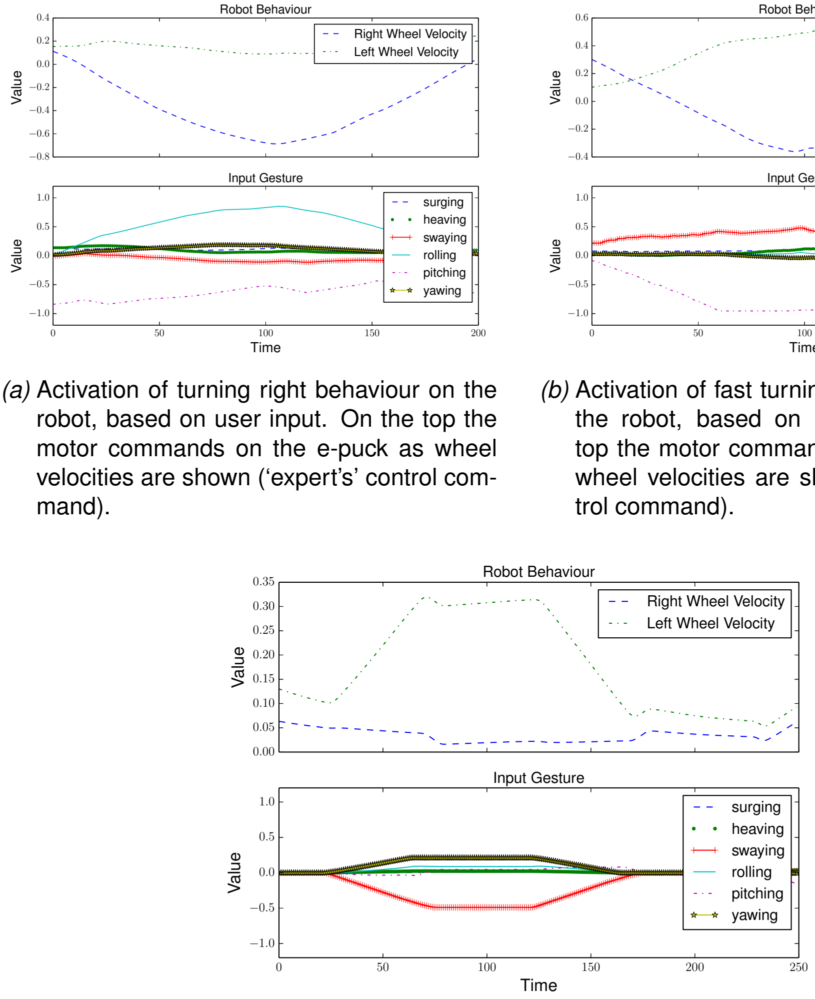
Transitions between user behaviours propagate to the ‘expert’ controllers of the robot resulting to stable and smooth transitions of motor commands for the roboot. From figure 6.9 we observe the change in the input signal just before time step 50 (bottom plot). The resulting change in the robot’s behaviour is seen on the top plot of the figure. The wheel velocities of the e-puck gradually change, with the increase of the left wheel velocity until the two wheel velocities are matched. This transition results to the robot changing its behaviour to forward moving (i.e. equal wheel velocities) from the initial turn left behaviour (i.e. greater velocity on the right wheel).

Modulation of Behaviours

Equally important with smooth transitions is the ability of the architecture to modulate the behaviours based on the modulation of the user input. This aspect also highlights the successful coupling of the input dynamics with those of the robotic behaviours. The intensity and the variation in the user’s input is propagated all the way to the motor commands of the robot, allowing the user to adjust the level at which robotic behaviours are exhibited. Since the e-puck is controlled through the velocities of the two wheels, we expect to see the robot being able to adjust the wheel speeds relative to the adjustments of the user’s input.

In figure 6.10 three examples are shown of the architecture’s ability to modulate the robot’s motor controls in accordance with the modulation to the user’s input. As seen in all of the three sub-figures these changes happen in the continuous, effectively embedding the user’s input signal into the e-puck’s behaviours.

All three sub-figures show a ‘turning right’ behaviour of the e-puck under the command of the user’s input. In figures 6.10a and 6.10b, a ‘fast’ turning of the e-puck is dictated by the user input while in figure 6.10c a slower more gradual turning. This can be observed in the difference between the wheel velocities commanded to the e-puck robot. While in 6.10a and 6.10b the difference approaches unity, in the case of 6.10c both speeds are closer, measuring approximately 0.3 of difference in velocity between the right and left wheel.



(a) Activation of turning right behaviour on the robot, based on user input. On the top the motor commands on the e-puck as wheel velocities are shown ('expert's' control command).

(b) Activation of fast turning right behaviour on the robot, based on user input. On the top the motor commands on the e-puck as wheel velocities are shown ('expert's' control command).

(c) Activation of slow turning right behaviour on the robot, based on user input. On the top the motor commands on the e-puck as wheel velocities are shown ('expert's' control command).

Figure 6.10: In the three plots the mapping between the user input and the robot's wheel velocities is shown. In each figure the top plot represents the behaviour of the robot as observed through the motor commands to the e-puck robot (i.e. wheel velocities). The bottom plots depict the user input as recorded by the input device. All values are plotted against time. The time is synchronised between the top and the bottom plots of each figure, showing the real time coupling of user commands and robot motors.

Another important observation is that the architecture is able to create all three possible combinations for the turning right behaviour. Looking at top plot of each respective figure, the commands to the e-puck wheel motors in velocities are depicted, from these we observe the following. In figure 6.10a the left wheel velocity is commanded to near zero values and the right wheel to negative values. In figure 6.10b the wheels have opposing velocities, with the right wheel having a negative velocity and the left wheel a positive one. Finally, in figure 6.10c the last possible combination of wheel velocities is observed, with the right wheel having near zero values and the left positive ones.

The explanation for the creation of these different motor modulations of the e-puck robot is found analysing the respective user's behaviour. Using Dynamic Time Warping and comparing the user's input behaviour to the ones they exhibited during the training procedure we obtain table 6.3.

Train / Test	Forward	Left	Right
Behaviour at 6.10a	1.68	**1.58**	2.43
Behaviour at 6.10b	1.93	2.88	**1.75**
Behaviour at 6.10c	**0.86**	1.59	1.79

Table 6.3: The table displays the distance measurements between the behaviours of figure 6.10 and the user's reference input behaviours (Train behaviours, provided at stage 2, section 6.4.2). The lower the number, the lower the distance between the two. In **bold** the smallest value showing the closest behaviour to that of the figure.

From the table it is observed that the behaviours are different as they result from the mix of the turning right input behaviour with other behaviours. Indeed mixing the turning right input with the moving forward results in the robot motors lowering the velocity of the right wheel to near zero values (3rd row of the table). While having a 'pure' turning right behaviour results to opposite wheel velocities. This since the pure turn right input should correctly activate a pure turn right behaviour of the robot, resulting to opposing wheel velocities. Finally, mixing the turn left with turn right behaviour the right wheel speed is commanded to negative values, with the left to near zero ones (1st row of the table).

Emergent Behaviours

Removing the backwards behaviour from the robot's behavioural repertoire highlights one of the emergent properties of the proposed architecture. Since the robot does not have the backwards behaviours there is also no user input associated with it. To this extent, both modules -the one for the robot behaviours and the one for the user behaviour recognition- are agnostic to the possibility of the robot moving in reverse.

The user's behaviour to trigger the forward behaviour on the robot can be described as 'a forward movement of the hand' above the Leap Motion Controller. The geometrically opposite behaviour could be said to be 'a backward movement of the hand' above the Leap Motion Controller along the axis it was initially moved forward. Since there is no 'backward' gesture in the training of the system, under any classification paradigm or otherwise recognition technique we would expect no behaviour to be triggered in the robot. To the contrary, in our case the formation of the coupling between the user's input behaviour dynamics and the robot's behaviour dynamics is such that the resulting robotic behaviour is moving backwards. This comes as an intuitive response from the system to the user movement, which also fulfills the expectation of the user. At the same time it follows through with the fundamental ideas of ergonomics. It increases controllability as it adds a new behaviour to the behavioural repertoire of the robot. Additionally, it makes the interpretation of the system easier by the user, enhancing the architecture's capability for interpreting the user's commands and intentions.

In figure 6.11 the e-puck's motor activations and the user's input triggering the backwards moving behaviour are displayed. In the bottom plot the user's input behaviour is observed. From the user's input behaviour we can observe that most values are similar with the case of forward moving behaviour, except from the 'pitching' and 'swaying' input's values that are reversed. When the 'reversed' input signal is fed to the ESN the output representing the forward behaviour becomes negative. This together with the linear combination of the 'experts' allow for the 'opposite from forward' behaviour to be exhibited by the e-puck.

Finally, a stopping behaviour emerged while using the system, as seen in figure 6.12. In the course of interaction, and with the user's behaviour being recorded with near zero values, the internal dynamics of the ESN start washing out. The 'memory' of the

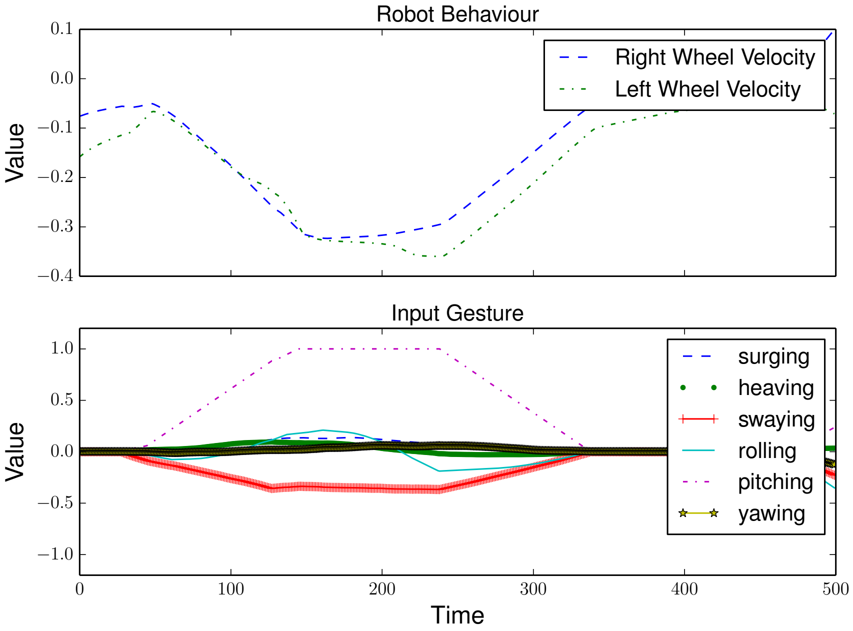


Figure 6.11: Activation of backwards moving behaviour on the robot, based on user input. On the top the motor commands on the e-puck as wheel velocities are shown ('expert's' control command). On the bottom plot the six recorded values for the user's input are observed (ESN input).

ESN (i.e. the dynamics of the recurrent connection's activations in the network) starts fading, the output levels of the network fade as well, reaching to near zero values. Since the user is still providing input, but such that the recorded values are zero, the network gradually lowers the activation of all behaviours, and this change is propagated to the robots motor commands. The velocity commands on the e-puck are decreased, reaching zero values, as seen in the top plot of the figure.

6.6 Conclusion

The architecture presented is capable of coupling user and robotic behaviours, enabling natural and intuitive control of the robot from the user. Indeed, a continuous control of the robot's behaviours is enabled based on the user's input signals. The methodology used and the procedure followed has no assumptions as of the robotic morphology nor for the input device. To this extend the architecture is agnostic to both, providing a general solution to the control of autonomous robots.

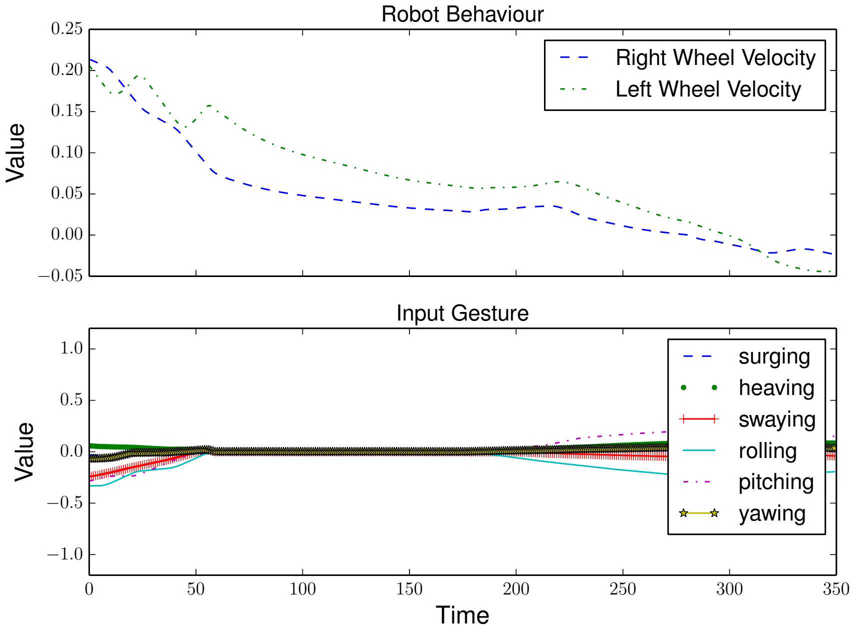


Figure 6.12: Stopping behaviour of the robot, based on user input. On the top the motor commands on the e-puck as wheel velocities are shown ('expert's' control command). On the bottom plot the six recorded values for the user's input are observed (ESN input).

Autonomous robotic behaviours have been explored, based on the principle of homeokinesis. These behaviours are grounded on the robot and its environment, and as such, allow for a meaningful representation of the robot's locomotive capabilities. Independent to the morphology, this exploration allows for the formation of a behavioural repertoire of the robot. The robot is then capable to display autonomy in the environment, being able to interact with it in a structured, predictable way. At the same time, the robotic behaviours provide the scaffolding for the display of more complex behaviours from the robot, through their combinations. This follows directly the idea of Behaviour Based Robotics, where complex behaviours can be formed from simple ones [21]. The exhibition of the behaviours, the transitions between them, and combinations of them are shown to be stable, robust, and replicable.

Furthermore, user behaviour is captured and mapped to the robotic one's independently of the input device used. Treating user input behaviours as time sequences of manipulations of the input device allows for pattern recognition methods to be used.

With the use of Recurrent Neural Network architectures, user input is coupled with robotic behaviours in a robust, and efficient way. At the same time, the methods used are of low computational complexity. This allows for the architecture to adapt to the user in a short amount of time, such that the system can be ready to use in less than a second. Overall, the architecture is able to adapt to the user and their control preferences, enabling an intuitive control paradigm. The user needs not to learn the system, rather the system learns the user. This is one of the highlights of the research presented in this paper, an architecture that can provide stable adaptation to the user, enhancing the usability of the system and its ergonomy.

From the establishment of the coupling between user and robot behaviours, a paradigm of continuous, real-time control emerges. From the separation of the robot and user modules the architecture is able to handle the different time scales present in both user and robot behaviours. Indeed, user behaviours of different lengths can be easily handled by the architecture, as the recurrent neural network is able to capture and recognise them in efficient manner. In the example of the Leap Motion used here, this enables the system to support both static and dynamic gestures. Adding to that, the dynamic gestures captured can be of different lengths (i.e. time spans) from each other. This follows the structure of the robot's behaviours, as through the modularity of the controllers the behaviours can be exhibited for multiple time lengths. Having both sub-modules varying in time, enables the system to couple the user's input behaviour to the underlying robotic behaviours, providing a real-time control architecture.

The architecture is able to handle the modulation of user input behaviours, being able to propagate them to the robotic ones. Working under a mapping paradigm, instead of a classification one, user behaviours can be recognised both when only a part is presented, or when mixed with each other. This feature is propagated to the robot behaviours, allowing for the partial activation, and the mixing of the self-generated primitive behaviours. As a direct result of this property, the architecture is able to handle transitions between behaviours as well.

An emergent property of the architecture is the ability to reverse behaviours, based on reversed input signals. Since the Leap Motion captures the location of the user's hand above the device, geometrical opposite input behaviours can trigger opposing robotic behaviours. Having not adapted on the reversed behaviour neither in the user side nor in the robot's side, the architecture is, nevertheless, able to handle a reversed input behaviour and also trigger the intuitive reversed robotic behaviour as a result. This feature of the setup highlights the robustness and the generalisation of the architecture while also providing support to the truthfulness of the approach towards human in the loop systems.

Ultimately, we can see the control method presented in this paper as an extension of the robot's sensory apparatus. The on-time connection provided by the architecture allows for the operator's experience of the environment to be mediated to the robot. Actions or reactions of the operator to their environmental stimuli are channelled to the robot through the interfacing of the architecture. Based on the ideas of situated and embodied cognition, we can investigate the way we communicate our movements to another morphology. The way that we understand and use our body. We can observe how the material agency of the input device affects and affords the user's control patterns. An investigation on how the mediated experience of another body -through the input device and interface- can result to a kinaesthetic experience, enhancing the way understand the morphology and its environment. As a parallel to Boden's 'conceptual spaces', this architecture aims to provide the constrains and allowances for the range of possible mappings between user and robotic morphology.

Chapter 7

Intuitive Control of the GummiArm Robot

Testing of the adaptive neural architecture to an 8 DOF robotic arm

7.1 Introduction

The work presented in this chapter, works in the paradigm of human-in-the-loop systems. We describe a novel framework able to autonomously form robotic behaviours and couple them with human ones. Being able to place the human in the loop seamlessly necessitates for methods beyond classification, capable of providing a continuous correspondence between the human and the robot. The proposed framework is described in detail together as are its component subsystems. The framework is tested with GummiArm, a 'soft' robotic arm that can almost entirely be printed on hobby-grade 3D printers. Our results are directed towards both the autonomous formation of robot behaviours and their effective coupling with human behaviours. Taking leverage of the 'soft' nature of the robot, we test our framework in the case of physical interaction and more precisely in a 'door opening' task. We find that the framework is able to both explore the robotic arm besides its complexity, and also couple robotic behaviours with human in an intuitive and natural way, being able to generalise to novel behaviours.

The research experiment presented here falls in line with the literature referenced in *Chapter 2* and the methodologies described in *Chapter 3*. To account for novelties and the extensions on the methodology we provide an overview in the *Methods, Section 8.2* of the chapter. For the background as well as the theoretical aspects of the work undertake the reader is refereed to *Chapter 2* of the thesis.

Control of complex robotic systems is a difficult task solved mostly with solutions tailored to specific robots. At the same, providing control -allowing a human to operate them- over such complex robotic systems is also dealt with in a case to case basis. The formation of an autonomous system that can perform both -controlling the robot while enabling its control from a human operator- is a challenging task in both levels.

The task of exploring the behavioural potentials of a robotic morphology, situated in the environment is a complex task. The theoretical and practical applications of self-organisation of robotic behaviours have shown good results, establishing a research field dealing with the autonomous formation of robot controllers with or without the need of external guidance.

The principals of Embodied and Enacted Cognition have brought forward the idea of systems that can act in response to their environmental stimuli and sensory information in a 'thoughtful' way. Making use of the complexity of the environment, adaptive learning mechanisms have been developed, mastering many locomotive and cognitive tasks. Acting within the environment through iterative procedures robot controllers can be formed, able to efficiently and effectively control robotic morphologies, tackling the problem of complexity, working with distributed representations of information. Artificial neural network architectures have been extensively used in the formation of such controllers given their outstanding performance and their computational efficiency (in part also because of advances in GPGPU computing methods and hardware).

Being able to explore a robotic morphology, requires that an ever adapting mechanism is put in place driving the robot's actions based on its perception. Having no external target, implies that the correction and adaptation procedure of the robot's controller can only work based on the error provided in sensory predictions. Incorporating the available information, while adapting on an on-line manner dynamical systems provide an ideal mathematical method for the formation and expression of such procedures.

Coming to the user part, in most cases it consists of providing a control mechanism and paradigm designed and tested by an engineer. This requires that the user has to study a manual and adapt themselves to the particulars of the mechanism and paradigm. Having a robust method that operates in a consistent manner is important. Equally important is that the mechanism is easy for the user to understand and efficient for them to control the robot. In most cases, having a pre-defined paradigm of control the idea of usability switches focus into stimulating (i.e. visualisations, graphics) the user in the right direction.

In what follows, we provide the methodological foundations and novelties for the proposed method, the experimental setup, the results and conclude with the an application outlook based on the results.

7.2 Methods

7.2.1 Exploring robotic behaviours - Homeokinetic Control

The self-organisation of the sensory-motor loop of the robot is realised as a dynamical system. For the exploration of the robot's capabilities we work as seen in [7, 22]. We want to be able to produce motor outputs from sensory readings and from them, predict the next sensory state of the robot. The creation of both a sensory-motor and a motor-sensory mapping allows us to derive an error signal for the update of the system parameters. The system is then able to create and adapt its motor-sensory mapping (referred to as the 'World Model'), in real-time, compensating for the misfit on the sensory values. The same error is also used to adapt the 'Controller', the module producing the sensory-motor mapping. This way, we perform an exploration of the kinematics and dynamics of the robots based on the robotic morphology itself.

Moreover, we are able to capture the dynamics exhibited by the robot as attractors formed in the behavioural space of the robot and reuse them. For this, we use a second module operating in parallel with the exploration module. This way, during the real time exploration of the robot's dynamics we are also able to have a series of controllers, in the form of basic behaviours, ready for the user to operate on. By activating each individual controller, the operator is able to manipulate the robot actions, driving the behaviour towards the basin of attraction described by the controller. We also show the ability to combine those basic behaviours, in order to exhibit combinations of behaviours.

The neural networks for the realisation of the above mentioned dynamical system are described below.

Both the Controller K and the World Model W are implemented as forward neural models with rate coding. The two networks working together describe the sensorimotor loop of the robot and are trained according to the homeokinetic principle. The exploration module is described, according to time t, as:

$$\tilde{\mathbf{x}}_{t+1} = A(y_t) + Sx_t + b + \xi_{t+1} \tag{7.1}$$

The controller generates motor outputs according to,

$$\mathbf{y}_t = g(Cx_t + h) \tag{7.2}$$

where C is the controllers matrix of size [*sensor_size, motor_size*], with its bias h, A the world models motor to sensor mapping of size [*motor_size, sensor_size*], S the sensor to sensor mapping of size [*sensor_size, sensor_size*], x_t the sensor values of time t, b the bias of the world model and g the activation function of the controller neurons, a *hyperbolic tangent* function, $tanh$. The error of the model at time $t+1$ is given by ξ_{t+1}.

For the calculation of the Jacobian matrix for the above sensorimotor model we work according to,

$$L = AG'C + S \tag{7.3}$$

with $G' = \delta g'(Cx+h)$, with g' being the derivative of the activation function of the controller's neurons.

Thus, the error - time loop error- for the adaptation of the model's parameters E is calculated as,

$$E = \xi^T \frac{1}{LL^T} \xi \tag{7.4}$$

the general parameter updates can be defined as

$$\Delta p = -\varepsilon_C \chi^T \frac{\partial L}{\partial p} \nu \tag{7.5}$$

with ε_C being the learning rule for the Controller parameters.

From the Time Loop Error, channel specific errors for the controller are calculated according to,

$$\varepsilon_i = 2\varepsilon_C \mu_i \zeta_i \tag{7.6}$$

with μ being,

$$\mu = G' A^T \chi \tag{7.7}$$

and ζ being,

$$\zeta = C\nu \tag{7.8}$$

with $\chi = \frac{1}{L^T}$ and $\nu = \frac{1}{L}\xi$ The adaptation of the World Model's and Controller's parameters follow,

$$\Delta A = \varepsilon_A \xi y \tag{7.9}$$

$$\Delta b = \varepsilon_A \xi \tag{7.10}$$

$$\Delta S = \varepsilon_A \xi x^T \tag{7.11}$$

for the World Model, with ε_A being the learning rate of the model. In our case set to $\varepsilon_A = 0.01$.

$$\Delta C = \varepsilon_C \mu v^T - qyx^T \tag{7.12}$$

$$\Delta h = -qy \tag{7.13}$$

with $q_{ij} = \delta_{ij}\varepsilon_i$ and δ being Kronecker delta. The channel specific error ε_i is given by 7.6. The overall learning rate for the controller ε_C was set to $\varepsilon_C = 0.1$.

For the initialisation of the matrices A, C, we apply a random oscillatory signal as the motor signal (i.e. bypassing the one generated by the model) and apply the equations 7.9, 7.10, 7.12, 7.13, 7.11 to adapt the initial weights of the models. The procedure stops once the Error produced by the World Model ξ stabilises. The adapted weights are then used for the initialisation of the homeokinetic control procedure.

The sensorimotor loop is from then on fully guided and regulated as described above.

7.2.2 Capturing robotic behaviours - Antagonistic Neural Networks

For the identification, storage and reuse of the different behaviours exhibited by the robot, we use a series of m neural networks (NNs), called *experts*. Each NN is defined according to the equation,

$$(\mathbf{x}_{t+1}, \mathbf{y}_t) = N_i(\mathbf{x}_t, \mathbf{x}_{t-1}), \quad i = 1, \dots, m \tag{7.14}$$

The NNs, working in parallel, compete for the prediction of the motor command $\mathbf{y}_t$ of time t and the sensory input $\mathbf{x}_{t+1}$ of the next time step in a winner-takes-all method, with only the winning network being allowed to train on the current data $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$. Thanks to this process, each NN specialises to represent a region of the entire sensorimotor space of the robot.

The NNs consist of 3 layers, feed-forward units where the hidden and output layers consist of sigmoid units, and the input layer of linear units. Online back-propagation is used to training the NN with learning rate $\eta = 0.1$. The size of the hidden layer is chosen to be 20. Assuming y to be the output vector of each neural network and x the input vector, we have

$$y = f(W_{hidden}h + b_h) \tag{7.15}$$

$$h = f(W_{input}x + b_x) \tag{7.16}$$

where f is the activation function, chosen to be a hyperbolic tangent. The matrices W_{input} and W_{hidden}, represent the weights from the input to the hidden and from the hidden to the output respectively. Finally, b_x and b_h represent the vectors for the bias units for the input and hidden layers respectively.

In each time step of the simulation the series of NNs are activated with the same input and the one with the best approximation of the next sensor values and motor commands is selected as the winner. The sample is then added to the training dataset of the winning NN which is then trained for one epoch. This way each network specialises in a single different behaviour of the robotic morphology.

The behaviour is efficiently stored in the distributed representation of the neural network being readily available for reuse by activating the neural network. When activating a NN we replace the Controller K and World Model W of the sensory-motor loop with the NN. Thus, the trained network is now producing the motor commands and the sensory predictions for the morphology.

7.2.3 Capturing User Behaviours - Echo State Networks

Echo State Networks (ESN) provide an architecture for efficient training of RNN in a supervised manner. One can distinguish two main components in an ESN. Firstly, he Dynamic Reservoir (DR), a large, random, recurrent neural network with fixed weights. The DR gets activated by the input and provides a non linear response for this input. And the output signal, which is trained as a linear combination of the activations of the DR. This way the computational resources and complexity required for the training RNNs is reduced to the adaptation of the output connections of the ESN.

Assume we have a ESN consisting of N reservoir units, K inputs and L outputs. First, we need to find the state, x, of the reservoir and based on the state and the input u, we can compute the output signal y. The state extended by the input, on which we base the computation of the output, will be referred to as the extended system state on the network, z. The extended system state, depending on the particulars of the implementation can also include the output of the reservoir, i.e. the output connections of the reservoir are recurrent.

So, the state update equation, for an ESN -without any recurrent output neurons- is,

$$\mathbf{x}(n+1) = f(\mathbf{W}\mathbf{x}(n) + \mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}^{fb}\mathbf{y}(n)) \tag{7.17}$$

where $\mathbf{x}(n)$ is the N-dimensional reservoir state, f is a sigmoid function (usually the logistic sigmoid or the tanh function), $\mathbf{W}$ is the $N \times N$ reservoir weight matrix, $\mathbf{W}^{in}$ is the $N \times K$ input weight matrix, $\mathbf{u}(n)$ is the Kdimensional input signal, $\mathbf{W}^{fb}$ is the $N \times L$ output feedback matrix, and $\mathbf{y}(n)$ is the L-dimensional output signal.

The extended system state $\mathbf{z}(n) = [\mathbf{x}(n); \mathbf{u}(n)]$ at time n is the concatenation of the reservoir and input states - and output in the case of output recurrency-.

The output is obtained from the extended system state by

$$\mathbf{y}(n) = g(\mathbf{W}^{out}\mathbf{z}(n)) \ , \tag{7.18}$$

where g is an output activation function (typically the identity or a sigmoid) and $\mathbf{W}^{out}$ is a $L \times (K+N)$-dimensional matrix of output weights.

For an ESN to function properly, the *echo state property* (ESP) is essential. ESP states that the dynamics of the DR will asymptotically washout, any information added by the input or feedback, from the initial conditions. It has been observed, that this can be achieved by scaling the *spectral radious* of the DR weights W to be less than unity. The ESP is then found to hold for the DR. In [131, 129] a more extensive discussion on the ESP and the dynamics of the network can be found.

For the training of ESNs, let us assume a driving signal $u(1), \ldots, u(n_{max})$ and the extended states it generates -once passed to the network- $z(1), \ldots, z(n_{max})$. We collect the states in matrix S of size $n_{max} \times (N+K)$ and the desired outputs $d(n)$ in a matrix D of size $n_{max} \times L$. Usually, before each collection, based on the properties of the network, we apply a washout period, allowing the network to settle to the input provided.

Now, the desired output weights W^{out} can be calculated as follows. First, the correlation matrix of the extended system states are calculated, $R = S'S$. Then the cross-correlation matrix of the extended states against the desired outputs d, $P = S'D$. Finally, for the output weight matrix is found by calculating the pseudoinverse of S, $S^{\dagger}$ and then updating the weights

$$W^{out} = (S^{\dagger}D)' \tag{7.19}$$

The network used for our setup has a reservoir of size 300, the spectral radius is set to $a = 0.995$. The feedback matrix is sampled from a uniform distribution in [-0.01, 0.01] and the input matrix in [-0.3, 0.3]. The sparsity of the reservoir, the input and feedback weights was set to 10%.

7.3 Experimental Setup

In this section the experimental setup is described. First we describe the robot morphology used GummiArm, then the input device used, and finally the particulars of the setup.

For the experimental setup we worked in multiple layers, testing our proposed framework in a real world scenario, validating the usability of the method, ensuring that the all the modules are able to cope with the complexity of the real world and a complex morphology, and finally to ensure that indeed a user can control these behaviours in a generative and intuitive manner.

First, the homeokinetic principles were put to test, to see weather it would be feasible for the homeokinetic method to explore the robot adequately and make use of the affordances of the environment. To address the former, we performed the behaviour exploration on the robot in a 'free' manner where the robot was not in contact with any objects in the environment. To address the affordance exploration we crafted the 'door handle' scenario. The robot was placed in front of a door and the idea here would be that the robot, once 'finding' the door would be able to explore its physical properties and making contact with the only movable part of the door (i.e. the handle) it would explore its affordances (i.e. twisting it) resulting to the opening of the door. The difficulty of the task was understood early on and the openness of the task was constrained by changing the initial position of the robot, placing its hand on the door handle as seen in picture 7.1. As seen in the picture the door handle was extended in length as to allow for a bigger force to be applied to the latch mechanism of the door and also to have a bigger contact point of the palm of the robot with the handle. The latter allowed for a better recording of the palm sensor of the robot as the area of contact would always be within the pressure sensor's receptive field. Other than that, the hand was free to detach from the door, if moved upwards or if the finger grip (hand contraction) would be reduced by the control algorithm (i.e. homeokinetic module).

Secondly, based on the robot's exploration of its environment, we needed to test whether

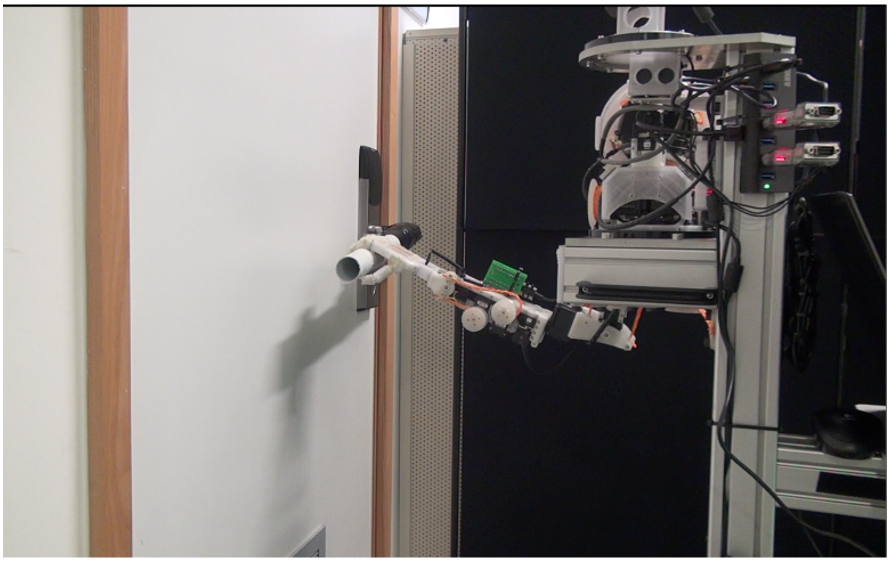


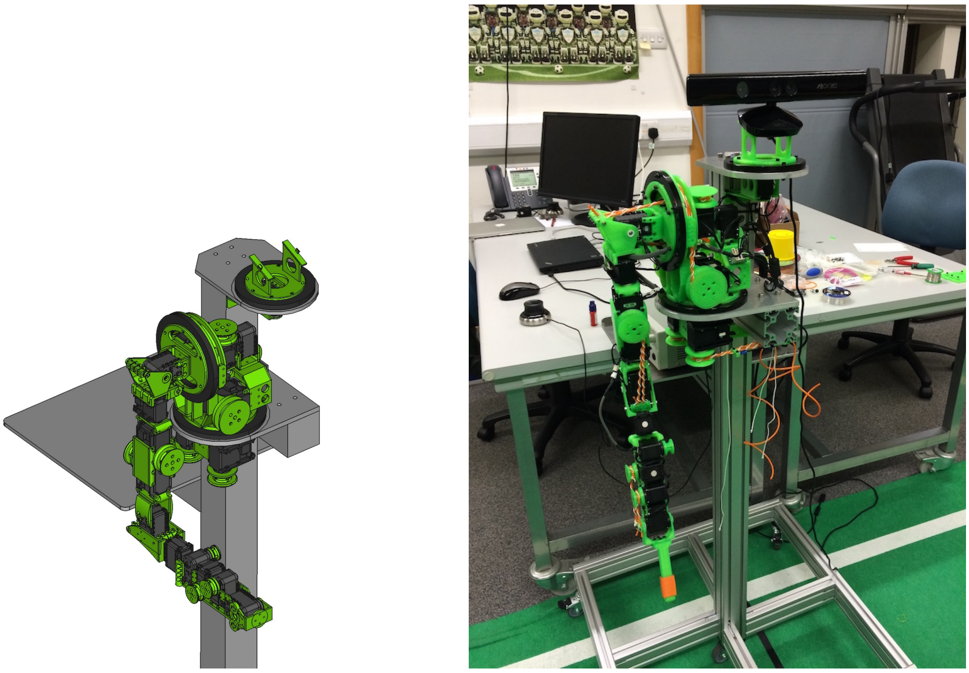
Figure 7.1: Initial position of the GummiArm for the experimental setup.

the extracted behaviours would be useful ones and able to be recreated by the 'experts' module. Once the exploration of the robot was complete testing whether the behaviours are useful and can reproduce a part of the exploration is essential. As the robot explores the door handle it is important to capture the 'door opening' behaviour of the robot and be able to trigger it on demand. This as the behaviour will be latter coupled with a behaviour from the user in order to establish the control paradigm.

Finally, capturing the user's behaviours through the input device was tested. Being able to recognise but also generalise the user's behaviours is essential as this enables the intuitive control paradigm as well as the accuracy of the control. Being able to generalise the control commands allows for the user to perform new, novel, control commands that the control module was never trained on. Having both, ensures that the spatio-temporal properties of the user behaviours are propagated to the robotic behaviours, enabling the emergence of new novel behaviours in that end.

7.3.1 GummiArm Robotic Arm

The GummiArm is a 'soft' robot arm that can almost entirely be printed on hobby - grade 3D printers. This enables rapid and iterative co - exploration of the arm mechanical structure and neural system, and provides a great platform for developing adaptive and bio - inspired behaviours. Viscoelastic actuator - tendon systems in an agonist - antagonist setup provide the arm with inherent damping, and stiffness that can be varied in real - time through co - contraction. Like the human arm it can therefore be 'soft' to absorb impacts, and to perform under uncertainty, but also stiffen up to be accurate. The idea behind its creation is to enable a sensorimotor learning that can exploit these properties, by taking inspiration from human motor control. The current architecture includes simple inverse models that enable a fast ballistic phase of movement, and predictive forward models for collision detection [170].



(a) Model of the robot.

(b) Photograph of the robot used in the experiments

Figure 7.2: GummiArm, a 'soft' robotic arm.

The robot consists of a 7 Degree of Freedom (DOF) arm, with 5 agonistic - antagonistic joints and a 3 DOF hand with touch sensing. In figure 7.3 the mechanical parts of the robot are sketched from a front and a side cut. The robot's hand is not depicted in the sketch. The *agonistic-antagonistic* joints allow for a variable stiffness of the joint, while the tendons elasticity provides the arm with passive compliance. The *viscoelastic* actuator-tendon system provides damping, given co-contraction of the tendons both during movements and after movement completion for accuracy.

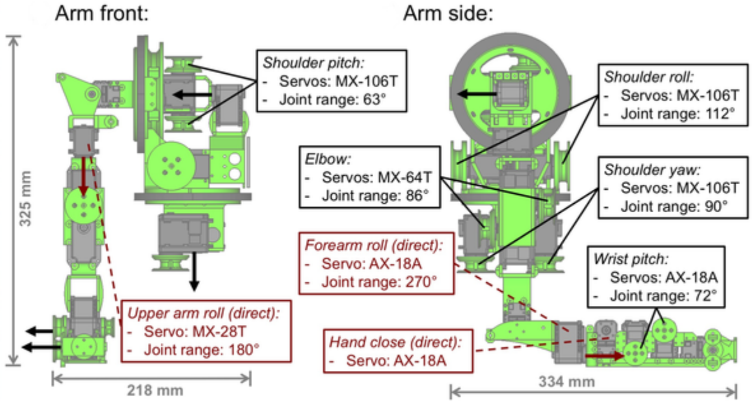


Figure 7.3: The mechanics of the GummiArm v2.1.0. Image taken from [170]

For the control of the arm we work with step-changes of equilibrium points in joint-space. The equilibrium points *p* range from -1 to 1 and are assumed to influence half the actuator range γ. At the same time, the co-contraction level of the joint is controlled were possible, allowing for variable stiffness to be applied by our controller.

7.3.2 Leap Motion Device

The input device used for the experiment and test of the proposed systems is the Leap Motion 4.3a. It is equipped with two cameras. From these cameras the device creates a skeleton of the user's hand hovering above the device. In our case, the device is placed on a working surface facing upwards, and the user operates in the space above the device. The position of the centre of the user's palm if recorded as input for out experiments. From the skeleton data provided from the device only 6 degrees of freedom are captured, representing the three rotational and three translational DOF of the palm.

7.4 Results

In this section the results of the (a) general behaviour exploration of the arm, (b) door handle scenario behaviour exploration from the arm, and (c) control of the behaviours extracted from the GummiArm.

7.4.1 General Behaviour Exploration with GummiArm

First the general possibilities of the arm and the homeokinetic module are explored. Each time step the input to the homeokinetic module is the sensory state s_t of the robot with the output y_t produced being directly passed for the control of the robot.

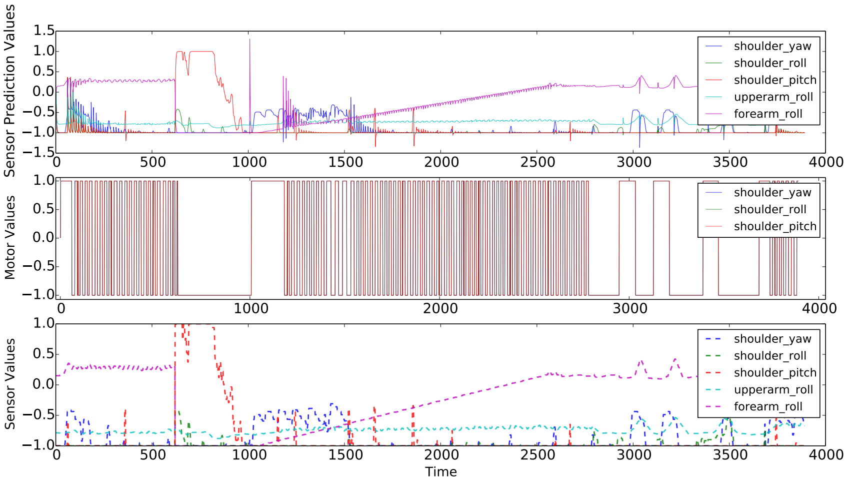


Figure 7.4: Mid air swinging behaviour of the GummiArm. On the bottom graph the sensory input of the robot is plotted according to time. On the middle, the motor commands passed to the robot from the homeokinetic network. On the top, the sensory predictions of the world model W of the homeokinetic module are shown. The time steps plotted are all the sampled frames as to increase the time resolution of the plot.

In figure 7.4 we observe a mid air swinging behaviour of the arm, guided by the shoulder joints. Working in synchrony the shoulder joints are able to achieve this oscillatory behaviour of the arm. Observing the changes in the frequency of the positive and negative values of the motors, we see that the behaviour is not a static one and perturbations caused by the movements in other joints cause the frequency to vary.

Being able to generate and maintain behaviours is important as the Antagonist Neural Networks module can acquire more samples from each behaviour being repeated in time. These changes allow for different networks to acquire sensorimotor data initially by having better predictions in particular areas of the sensorimotor space, and through time gain expertise of that area.

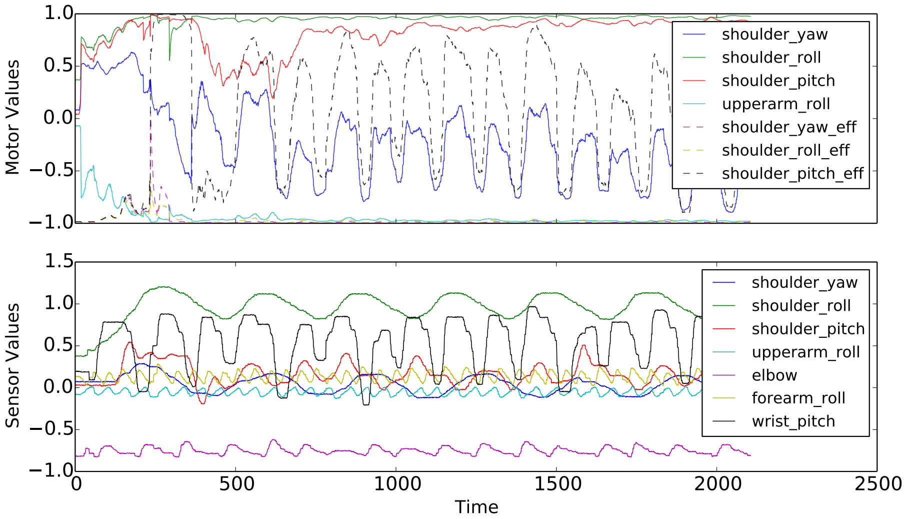


Figure 7.5: Shoulder play of the GummiArm. On the bottom graph the sensory input of the robot is plotted according to time. On the top, the motor commands passed to the robot from the homeokinetic network. The time steps plotted are all the sampled frames as to increase the time resolution of the plot.

In figure 7.5 an additional example of behaviour maintenance is seen. The homeokinetic module is able to maintain a fluid pattern of oscillations, once again allowing for the better training of the Antagonist Neural Networks module. In this figure we are able to observe the reaction of the whole arms sensory apparatus. The touch sensor is omitted as the hand is freely allowed to move in space and so there is no possible feedback from the palm. From the motor commands we observe the co-contraction of the shoulder yaw (*shoulder_yaw_eff*) and pitch (*shoulder_pitch_eff*) are quickly synchronised and that of the shoulder roll (*shoulder_roll_eff*) is minimised allowing for a free and fluid movement of the arm. The motor output driving the movement of the arm is seen to be the shoulder yaw motor (*shoulder_yaw*) ‘pacing’ the oscillatory behaviour.

In figure 7.6 we observe the coordination of the palm touch sensor (at the bottom plot) and the palm contraction (at the top plot). Through the adaptation of the sensorimotor loop of the robot the homeokinetic paradigm is able to locate the relation between hand contraction (*hand_contraction*) and the palm sensor values (*palm_touch*). At the same time the effect of the relation is propagated to the wrist of the arm, as seen on the changes in the values of the wrist’s pitch motor values (*wrist_pitch*) and co-contraction

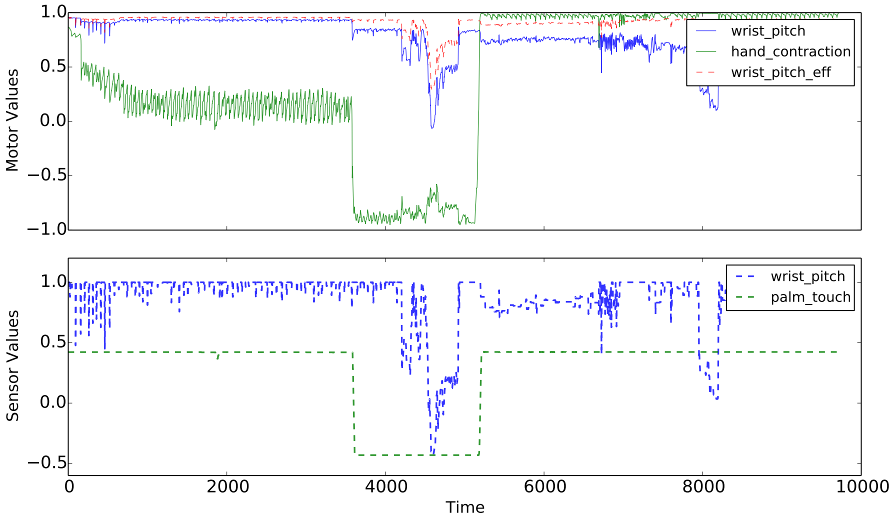


Figure 7.6: Changing modes of behaviour of the GummiArm. A closer look on the coordination of the palm touch sensor at the bottom and the palm effort at the top graph. Values are plotted against time. The time steps plotted are all the sampled frames as to increase the time resolution of the plot. The bottom graph shows the sensory input of the homeokinetic module while the top the motoric output.

levels of the antagonistic tendons of the wrist (*wrist_pitch_eff*). On a more behavioural level the robotic arm is showing a reaction of grasp pointing its palm towards the sensory input by means of twisting the wrist. This, in synchrony with a contraction of the hand, as to once again increase the sensory input and make the 'grasp' movement complete.

7.4.2 Door Handle Exploration with GummiArm

Here the potentials of the homeokinetic module in exploring the affordnces of the environment are explored. Each time step the input to the homeokinetic module is the sensory state s_t of the robot with the output y_t produced being directly passed for the control of the robot.

Being able to explore the environment sufficiently recognising and exploiting the affordances found in it is essential. This, as the proposed mechanism is able to capture important aspects of its environment, satisfying the situated and enacted paradigms which is meant to serve. In doing so, our proposed mechanism is able to extract and explore sensorimotor patterns that are inherent to the environment and the robotic morphology. The patterns of control result from the adaptation of the internal neural model of the sensorimotor loop of the robot, relying only on proprioceptive input. Doing so, we allow the robot to autonomously work on the formation of its behaviours without external human intervention. At the same time the modular fashion of the proposed system allows for extra behaviours to be used, behaviours not extracted from the self-organisation of the sensorimotor loop, by means of providing an artificial neural network describing the behaviour in response to the sensory input from the robot.

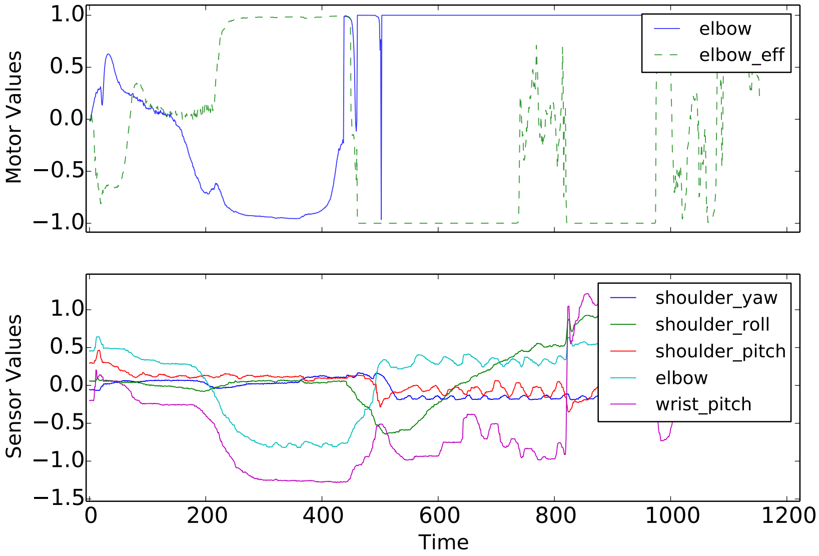


Figure 7.7: Change of the effort level of the elbow during the exploration of the door handle. On the bottom, the sensory input of the behaviour exploration module is plotted against time. On the middle, the motor output of the module and on the top the sensory predictions. The time steps plotted are all the sampled frames as to increase the time resolution of the plot.

In figure 7.7, we observe the adaptation of the elbow joint motor of GummiArm while the robot explores the 'door handle' environment 7.1. In the top plot the elbow motor command passed to the robot is plotted (*elbow*) and the co-contraction level as adjusted by the homeokinetic controller (*elbow_eff*). The sensor values recorded on the robot's shoulder joints (*shoulder_yaw, shoulder_roll, shoulder_pitch*) as well as on the elbow and wrist joints (*elbow, wrist_pitch*). We observe how the co-contraction of the elbow is increased as the elbow is pushing downwards the handle. Just after 400th time step the elbow values go up again, showing the elbow going to the opposite direction, where there is no resistance from the handle any more and the co-contraction levels are lowered to the minimum. This shows that the homeokinetic adaptation is able to exploit the correlations and the effects of the co-contraction level to the changes on the joint angles. As co-contraction is lowered the joint position set is less accurate allowing for a greater and more fluid exploration of the sensorimotor dynamics. At the same time when the robot finds feedback in the environment (i.e. door handle) the co-contraction level is increased, stabilising the effects the motor commands have to the joint positions. Stiffening up the tendons means that the robot can push the handle with more force, with the motor values forwarded to the robot having the desired and expected effected in the sensory state of the robot.

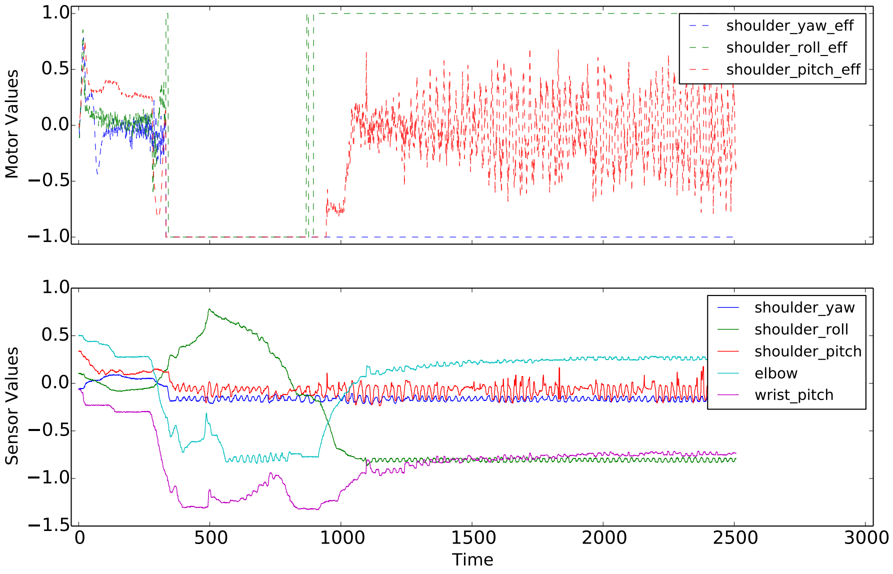


Figure 7.8: Change of the effort level of the shoulder during the exploration and opening of the door handle. On the bottom, the sensory input of the behaviour exploration module is plotted against time. On the middle, the motor output of the module and on the top the sensory predictions. The time steps plotted are all the sampled frames as to increase the time resolution of the plot.

In figure 7.8 we look closer to the motor value adaptation during exploration of the ‘door handle’ environment on shoulder motors of the robot 7.1. In the plot the co-contraction levels for the shoulder DOFs controlled by the homeokinetic module are plotted - shoulder yaw, roll and pitch (*shoulder_yaw_eff, shoulder_roll_eff, shoulder_pitch_eff*). Initially the co-contraction levels are close to zero, and then adjusted down to their minimum, allowing for the robot to maximise the 'looseness of the shoulder' increasing the effect the motor commands have on the sensory state of the robot. This is seen on the bottom plot where the sensor values recorded on the robot's shoulder joints (*shoulder_yaw, shoulder_roll, shoulder_pitch*) as well as on the elbow and wrist joints (*elbow, wrist_pitch*) are shown. At the point of pushing the handle downwards (i.e. the affordance is 'found') a new area of the sensorimotor space is 'accessed' by the robot, where the ever adapting homeokinetic module is able to explore sensorimotor patterns not available before. In doing so, we observe the co-contraction level of the shoulder yaw (*shoulder_yaw_eff*) if increased to maximum, stiffening the tendons, while the shoulder roll (*shoulder_roll_eff*) is lowered to the minimum loosening the tendons to the maximum level possible. In doing so the shoulder pitch (*shoulder_pitch_eff*) co-contraction levels can be explored and their effects in the sensorimotor patterns of the robot. All the above manipulations resulting from the autonomous adaptation of the sensorimotor model based on the homeokinetic learning rule. The shoulder roll is seen to follow an oscillatory behaviour which, as said before, allows for the sensorimotor patterns to be better learned by the Antagonistic Neural Networks module, thus producing expert networks for particular behaviours (i.e. areas in the sensorimotor space) of the robot.

In figure 7.9 we observe the changes on the GummiArm's hand contraction level (*hand_contraction*) for the same sensorimotor interaction as in figure 7.8. On the top plot of the figure, we observe that the co-contraction level initially and for a very short time goes up towards 0.5 and then to -0.5 from where it returns to a near zero 0.0 level. On the following time steps, we observe that the co-contraction of the palm is going to the lowest value, which for the hand means 'loosening' the grip. Although that seems counter-intuitive, after the 1000th time step we observe that once the door handle is pushed down the grip in again 'tightened' by means of higher co-contraction. The subsequent oscillatory behaviour of the co-contraction shows the exploration of the affordances of the door's

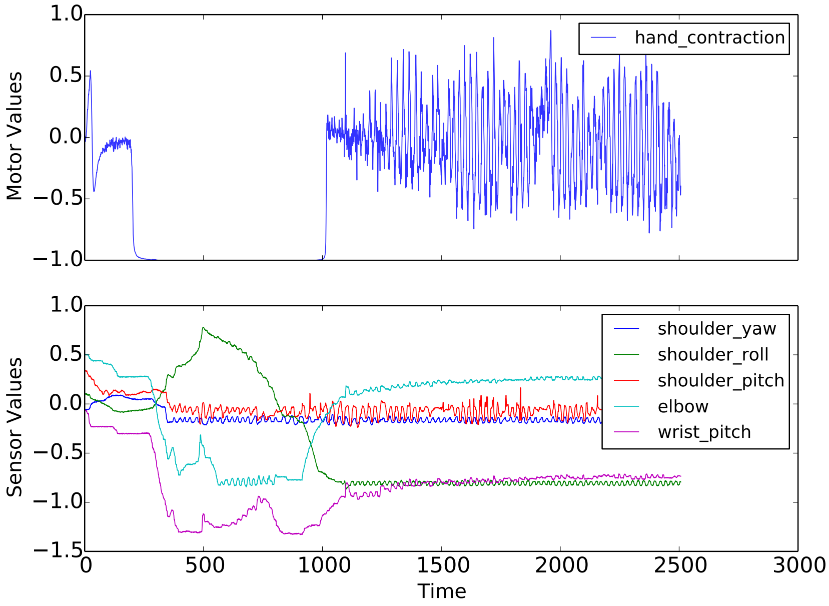


Figure 7.9: Change of the contraction level of the hand during the exploration and opening of the door handle. On the bottom, the sensory input of the behaviour exploration module is plotted against time. On the middle, the motor output of the module and on the top the sensory predictions. The time steps plotted are all the sampled frames as to increase the time resolution of the plot.

handle, where the hand ‘tightens’ and ‘loosens’ its grip, as to allow for an exploration of the emerging sensorimotor patterns.

7.4.3 Behavioural Control with GummiArm

In this section, the results and the coupling of the behaviours captured by the Antagonistic Neural Networks module with those from the user are elaborated. These ‘expert’ networks are placed in control of GummiArm, with the arm exhibiting the captured behaviours. The user observing these behaviours reacts with manipulations of the input device. In this reversed paradigm for the formation of control signals it the human that responds to the robot behaviours rather than the opposite, seen in most research. The idea here is that observing the arm performing the ‘learned’ behaviours the human operator forms intentions for control signals and communicates them through the input device. Doing so enables the user to form their own and personalised control paradigm and commands over the robot. Important to note here is that the user is able to perform any response to the robot's behaviour, thus enabled to take a 'first' person perspective or a 'third' person perspective over the robot's behaviours. In our method there is no dictation over the possibilities that the user has over the for the control of the robot. This is enabled by the adaptive mechanism dealing with the user input, the ESN module. Through the user's responses the control patterns are formed and a dataset is created for the adaptation of the ESN in supervised manner. Here another important aspect is the speed of the training, in which the ESN architecture has a linear complexity $O(N)$ over the length of the dataset sequences. This allows for an almost instant adaptation of the network in less than 1 second. This makes the paradigm applicable in real case scenarios and the user experience of the training smooth and without latencies.

To test the control possibilities of the robot behaviours and also display some of the fundamental properties of the architecture as a whole, only the 'open door' behaviour of the robot is controlled. The behaviour is exhibited by the robot as captured by the 'expert' neural network with the user responding with a manipulation of the input device to be associated with. Working with only one behaviour we can easily show how the proposed method is able to generalise to unknown input signals, how it deals with distortions of the input signal, having a closed system of only one behaviour. This allows us to be sure that only the effects of the single behaviour are generalised without having to take into account interference from other behaviours. Given the complexity of the robot it is important to keep the demonstration simple and to the point, as a more complicated example and testing scenario would more difficult to elaborate and display.

In figure 7.10 the input of the user is shown, for their response to the 'open door' behaviour of the robot. The input collected is short and also quite noisy as seen from the plot. The plot shows the 6 DOF captured by the input device in time. Essential part of the method is for the dataset to be created on spot. This means that the user input is recorded as the user's hand enters and leave's the devices recording field. No preprocessing of segmentation of the input is performed and the complexity of capturing the 'core behaviour' of the user is left to the ESN. Indeed, as it will be shown

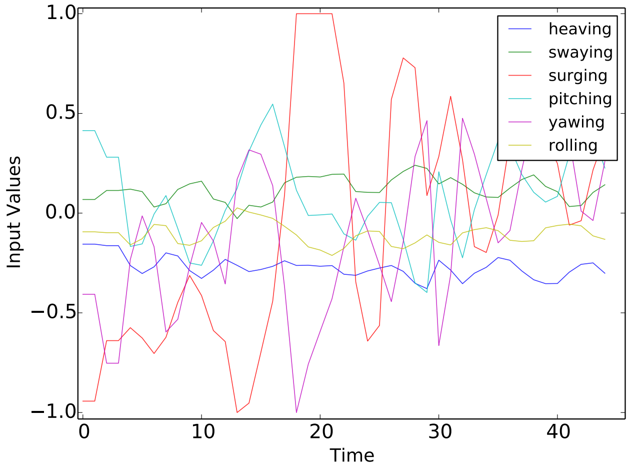


Figure 7.10: The user input recorded by the Leap Motion plotted against time.

later the detection as well as the recognition are performed by ESN.

Having captured the user's input behaviour, the ESN is trained and its output is then used as the weight for the activation of the behaviour. The output of the ESN for each user behaviour recognised is multiplied with the output of the corresponding 'expert' resulting to the level at which the corresponding motor output of the 'expert' is used to define the overall output for the robot's behaviour. The outputs of the experts are linearly combined with the output of the ESN used as a weighting for each output. In this case, having only one behaviour, we are able to investigate the responses of the ESN closely and observe how it deals with the incoming input signals in a continuous manner.

Once the ESN is trained, the user can then manipulate the device in a continuous manner producing a continuous output, thus resulting to a continuous control of the arm. If the user stops providing input the output of the ESN fades and the control of the robot terminates. Manipulating the input device again, the user can re-enter the robot's loop and control the robot. Based on the spatio-temporal variations of the input the ESN's output varies accordingly and as is shown in the rest of this section, generalises the user input commands, in cases triggering novel behaviours on the robot; behaviours that are spatially constructed based on the input present. As is described later and in the case of ‘door opening’ behaviour, the proposed system is able to generalise, capture (from the input) and apply (from the ‘experts’) a ‘closing the door’ behaviour.

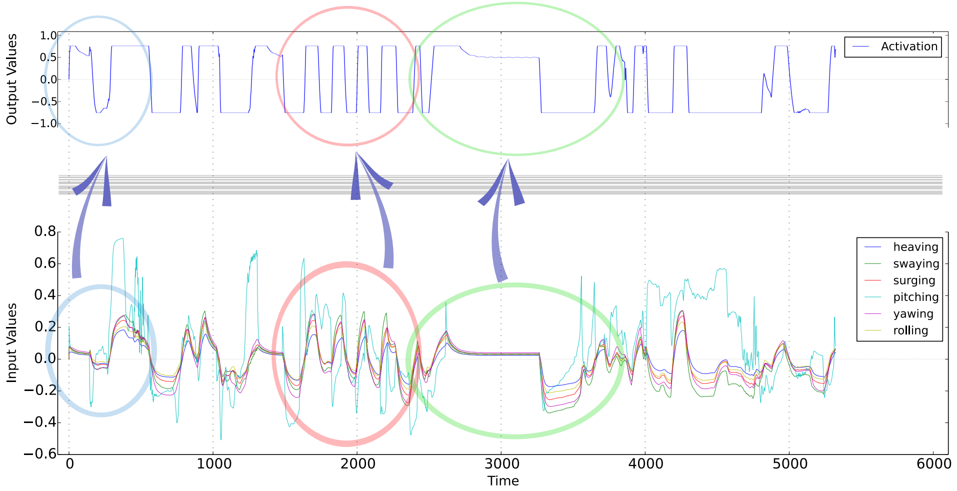


Figure 7.11: Coupling of human behaviour and robot movement. On the bottom the user input recorded by the Leap Motion is plotted against time. On the top the output of the ESN and activation level of the open door behaviour of the robot. The time steps plotted are all the sampled frames as to increase the time resolution of the plot.

In figure 7.11 the real time coupling of human and robot behaviour is shown. the user initially performs their recorded movement, but expanded in time (*blue circle* in the figure). The ESN is correctly recognising the user behaviour and produces a positive output, triggering the ‘expert’ accordingly. This results to the robot behaviour of the robot to be forwarded to the robot multiplied with a positive coefficient. In the rest of the *blue circle* of the figure we observe (on the bottom plot -user input-) that the user provides a geometrically opposite input, and the ESN is able to reverse its output and generate the ‘closing door behaviour’. The user’s behaviour could be seen as a ‘press’ behaviour above the LeapMotion device, with their hand moving vertically closer to the device. The user generating the ‘opposite’ behaviour of ‘pulling’ moving their hand vertically away from the device, produce an signal that has a geometrical relation ship with the input the ESN was trained on. This geometric relation is forwarded to the output of the ESN and subsequently to the robot, making it ‘close’ the door.

Later in time, the *red circle* shows how the temporal dynamics of the user input (bottom plot) are reproduced in the output (top plot, following the arrow) with the output of the ESN following the rhythm of the input. At the same time we observe that the same spatial effect explained before are also present being able to propagate the reverse robot behaviour. Overall, while the *blue circle* shows an opening and closing of the door, here we see a faster opening and closing behaviour triggered on the robot.

The *green circle* again shows how the temporal dynamics of the user input (bottom plot) are reproduced in the output (top plot, following the arrow) with the output of the ESN following this time the slow exhibition of the input. The ESN activation drops following the user’s input drop, while on last segment it is observed that the same slowing also hold in combination with the spatial properties of the signal, having the output reversed while prolonged in time.

Finally we observe around time 5000, that the ESN’s output is near zero although there is input present. This highlights the detection ability of the ESN module. Since the user input is not comparable to the spatio-temporal properties of the trained one the output goes to zero, showing a no-activation of the ‘open door’ ‘expert’.

7.5 Conclusion

The architecture presented is capable of coupling user and robotic behaviours, enabling natural and intuitive control of the robot from the user. A continuous control of the robot’s behaviours is enabled based on the user’s input signals. The methodology used and the procedure followed has no assumptions as of the robotic morphology nor for the input device. To this extend the architecture is agnostic to both, providing a general solution to the control of autonomous robots.

First, we show that the introduced complexity of a physical robot can be tackled by the exploration module following the homeokinetic rule. Introducing the extra motor controls for the co-contraction of the tendons, we have shown how the sensorimotor patterns are explored and the motor and co-contraction levels of the joints are treated differently in exploring the behavioural capacity of the robot.

Second, we show that the Antagonistic Neural Model, is able to capture the generated behaviours of the robot. Besides its simplicity, taking leverage of the re-occurrence of the robot's behavioural patterns it is able to properly disassemble the sensorimotor interactions of the robot into behaviours. At the same time we are able to show that the particular architecture, because of its distributed nature is able to tackle the complexity of on-line learning and do so with modest requirements in hardware, being run in a standard office laptop with 4Gb of RAM and a Intel i5 processor.

Third, we show that the ESN architecture provides a robust and capable user-behaviour recognition module. We have found that the spatio-temporal properties of the signal can be retained and that the proposed system can correctly propagate them to the controlled robot, besides its complexity.

Ultimately, with the formation of the 'door handle' scenario, we show that the proposed system can effectively explore the environment's affordances. At the same time we show that the sensorimotor patterns emerging from the afforded manipulations of the environment can be autonomously analysed to behaviours and that theses behaviours can be user by a user. To this extend we are also able to automatically formulate the control paradigm with the user without them having to know anything about the system or engage in any training. We show that the proposed method can work in real-life scenarios to establish autonomy and a behavioural repertoire from a given robot, as well couple these behaviours with user ones also retaining their spatio-temporal properties.

The idea put forward from this work, aims on the establishment of a new paradigm and niche in human robot control. This as it treats the robots as situated enacted agents, enabling them an understanding of their environment, while producing a simplification of it to behavioural segments. Removing thus a great burden from the designers, not only on the robot but also on the human side. This, as it allows for the automatic formation of a control paradigm, tailored to the user, their preferences and capabilities. This way a new approach of intelligent human-robot control emerges, where robot and human are treated equally in expectations; as much is the user expected to understand the robot and its behaviours, its the same the robot is expected to understand the human. Expectations from the robot are formed by observing it behave and expectations from the human are formed as they provide their feedback to the behaviours. As much is one expected to be consistent so is the other.

Conclusion

This thesis put forward a novel way of control for robotic morphologies, capable of adapting both to the user and the robot, while enabling a paradigm of intuitive control for the user. First, the theoretical background was given (*Chapter 1*) in order to frame the research methods followed, together with state-of-the-art method for both human behaviour recognition and robot behaviour generation. In *Section 1.1* the theoretical perspectives of Embodied Cognition are elaborated, introducing the main concepts guiding the experimental work undertaken. An overview of *Embodied Cognitive Science* (*Sub-Section 1.1.1*) was followed by an elaboration on the notion of *Affordances* and that of *Ecological Perception* shedding light to the potential and the importance of the interaction between the body and the environment. This section was concluded with a take on *Intuition* (*Sub-Section 1.1.3*) to build up for a theoretical framework allowing a new- fresh take on human robot control. The ideas put forward here, form the research lines and main theoretical assumptions followed in all the experiments conducted and also indicate the theoretical goal of this research. In the following section, an overview of the methods found in literature, for the construction of robotic behaviours, is given (*Section 1.2*). Both *top-down* and *bottom-up* approaches are expanded and state-of-the-art methods are considered in what is - a literature overview on *Robot Behaviour Dynamics*. In *Section 1.3* the focus changed for the theoretical and methodological approaches in interfacing humans and machines. The fields of *Human-Machine Iteraction*, *User Interfaces*, as well as *Human Robot Interaction* are analysed highlighting the potential, as well as, the necessity for our research.

Chapter 2 provided all the methodological background, mathematical foundations, and implementations necessary for the application and understanding of the theoretical work mentioned in Chapter 1. Methods for interfacing the human operator were formulated as well as methods for interfacing the robotic morphologies.

Chapter 3 presented an *intelligent interface enabling the remote of arbitrary complex robotic morphologies by translating intuitive human behaviours into purposeful robotic actions*. The usability and performance of Recurrent Neural Network with Parametric Bias (RNNPB) was shown in human behaviour recognition. Treating user's input as a timed sequence of manipulations of the input device, the usage of RNNPB was shown to be a valid method for the online adaptation of the proposed system to the user input. At the same time, the continuous accumulation of activations in the hidden layers of the RNNPB was shown to allow for a continuous interaction paradigm in comparison to the segmented one allowed by Dynamic Time Warping. On the robot's interface mechanism, the efficacy of the *homeokinetic* paradigm for the self organisation of the sensorimotor loop was shown and the proposed method of behaviour segmentation was tested in all three: behaviour extraction, generation and mixing.

In *Chapter 4*, a *two-way adaptive interface for intuitive robot control* was implemented and tested. The system incorporated an Echo State Network for the user's behaviour recognition and the *homeokinetic* exploration coupled with the *'expert' neural networks* for the control of the robotic morphology. The proposed interfacing system was tested with yet different robotic morphologies, establishing its robustness and universality, while for the user input two input devices were used highlighting the universality of the Echo State approach as well. Through the analysis of the proposed interface we were able to show that the system could effectively place the human in the loop of the robotic behaviours, while adapting to the multidimensional input from either input device. The continuity of the resulting interface mechanism was very surprising in that the interface was able to work in different time scales and manage the human and robotic behaviours concurrently.

Based on the effectiveness of the Echo State approach and the interface mechanism of *Chapter 4* in the next *Chapter 5* an extensive functional analysis of the Echo State architecture was conducted. To assess the applicability and potentials of Echo State networks, and Reservoir Neural Architectures in general, we followed a methodology encompassing three stages. First, a benchmark was performed as to establish the validity of the approach and place it amongst the state of the art. Second, functional properties were tested on dataset crafted by us. Finally, a user testing was performed to control the adaptation quality of the system. The variability between the user's input behaviours was analysed and shown not to have an impact over the systems performance. The system was shown to provide a continuous mapping from raw data inputs, in that no preprocessing was needed on the user's input sequences with the Echo State Network being able to handle its complexity as well as its inconsistency. The system was also shown to be able to propagate the geometrical properties of the input, recognising geometrically 'opposite' behaviours as recorded by the input device. Finally, since the system is aimed to work on a continuous manner the percentage of input needed to trigger the right recognition was analytically calculated and measured. The potential of the proposed system was also discussed under the general field of human-machine interaction and the field of assistive robotics.

In *Chapter 6* the proposed system was extensively tested in both ends- human and robot, in an experimental setup using an established robotic morphology, the 'e-puck' robot. In testing with a 'simple' robotic morphology, having only two motors, we were able to investigate the behaviour of the proposed system in detail, displaying the effects of the input commands to the motors of the robot. This entailed showing the properties of continuous-time operation enabled, the handling of transitions between user and robot behaviours and the emergent properties of the system. In an extensive experimental setup we were able to show that the proposed system is not only able to couple user and robot behaviours but also generalise to novel user inputs in an intuitive way. In particular we were able to generate a 'backwards' behaviour on the robot based on a novel input provided by the user. Indeed, the user providing the geometrically 'opposite' behaviour of forward the system was able to trigger the backwards behaviour on the robot. Through this experiment we were able to provide evidence for the *Dynamic Behaviour Coupling in Human-Robot Interaction*. This, since the system was able to create a 'common mapping' between the user and robot behaviours.

Finally, with the last experiment we were able to transition from simulated environments to a real world scenario, while also increase the complexity of the robotic morphology using an 8 DOF robotic arm, the GummiArm. In this experiment, we were able to show that the proposed system, as originally formulated in *Chapter 4*, is able to scale and handle both the complexity of the real world and highly complicated robots, while retaining its properties and providing an intuitive way of control. Constructing a 'door opening' task, we were able to show that not only the system is able to effectively explore the robot and capture behaviours, but that the enacted and embodied paradigms are more that well expressed by the system. The robot was able to explore the affordances of the door handle and open the door. The behaviour of 'door opening' was successfully captured and coupled with the user's input. Ultimately the user was not only able to trigger the 'open door' behaviour, but based on the generative nature of the proposed system also trigger a 'close door' behaviour on the robot providing the 'opposite' input behaviour.

Future Research Directions

The work presented in this thesis, provides a novel way of communication between human and robot. Working with adaptive and adaptable methodologies it has been possible to provide a new way forward in connect human and machine. In this thesis the design of a framework for the just-in-time creation and real-time usage of a communication pathway (i.e. interface) between the human and robot has been described and put to test. Having done so, unique possibilities arise for further research in two main directions, (a) self-organisation of robotic behaviours and (b) the adaptive behaviour-based communication of humans and machines. At the same time the technological readiness of the methods used as well as their computational needs, show a way forward in commercialising the framework as an Assistive Technology.

Embracing the complexity and diversity of each individual user and robot respectively, the framework has been designed not just as an adaptable one, but rather as reshape-able one. Indeed, adaptable would imply that the communication paradigm can be adjusted to the user. In the case of this work, the individual user and robot are able to both: (a) create their own behaviours and (b) build a way (interface) to communicate them to one another. While the ideas of User-Centered Design (UCD) exist in our work, the adaptation processes happen real-time through the interactions of user and machine, with both the user and the machine shaping their mutual communication (i.e. the Common Behavioural Space of chapter 6). An interesting research direction here, would be the mathematical formalisation and standardisation of a behaviour such that a communication protocol is established. Having a standard way of exchanging behavioural information would allow for an easier comparison of such adaptive methods, in the benefit of the research field.

A novel paradigm of tackling the communication of humans and machines Being highly personalised, created for and used by the individual user, the framework removes the design process from the lab and places it to the convenience of the user. In doing so, the work is readily applicable in the fields of assistive robotics and assistive technologies in general. This, as the system is capable of adapting to the user commands without any ground notion of 'right' or 'wrong'. The decisions for the communication of their intention are solely based on the individual without any explicit or implicit assumptions on their realisation. A good example of this is shown in chapter 5 where the human participants using the framework selected control strategies in driving a robot around its environment based on their own preference and idea of control resulting to examples of first or third person perspectives. At the same time, not all have had the same reactions and control patterns, showing that the afforded manipulations of the robot and the selected controller were differently perceived by users. Indeed, although 'mean' control behaviours could be calculated from an aggregation of their respective behaviours, the particularities of each user would not be captured following such a method. In this, we see a more general turn of Machine Learning and Artificially Intelligent applications towards the communication with the user. Such allows for a change in the relationship between users and technology, enabling new applications, while making it broadly inclusive. This as robots as well as virtual agents become ever more present in our everyday life.

Directed self-organisation of robotic behaviours the framework described here is able to self-organise robotic behaviours, in that the internal dynamics of the morphology account for the generated behaviours. An interesting point of research would be augmenting such behaviours with goal-directed ones, in practice allowing the user to shape the fundamental robotic behaviours. A second order of adaptation would allow user and robot not only to communicate to each other but also to shape each other. Designing through interacting becomes important as the complicacy of the available robots increases.

Common Behavioural Space the creation of a shared behavioural space between the robot and the user brings forward the idea of a calculus. Defining a calculus for such space would enable structured (by means of prediction) operation in such space. Although a step has been made in this thesis combining spatial and temporal properties of the behaviours, strictly defining such space and its properties provides an interesting research direction. In a metaphor, one can see the calculus as the grammatical rules given that the common behavioural space is the shared vocabulary.

Augmentative and Alternative Communication

Augmentative and Alternative Communication AAC defines an area of clinical practice that aims to help people with communication impairments. In this, AAC is seen as an intervention technique having the goal to enable communication pathways for cases of medical and developmental disorders. AAC mainly focuses on speech, as speech is widely seen as a primary method for communication, but also ranges from gestures to sign language and facial expression recognition.

AAC systems can be divided in two types: unaided and aided. Unaided AAC comprises of cases where there is no need for any external tool and include facial expressions, vocalisations, gestures, and sign languages. On the other hand, aided includes the usage of either electric and non-electric devices for the that are used to transmit or receive messages. Non-electrical aids are referred to as low-tech, while electrical ones are mentioned as high-tech in the literature.

It is here in the case of high-tech aids that the methods presented in this thesis fall in. Creating a system that can be trained to the user, without any prior knowledge of special abilities or preferences by them; enabling a continuous, robust and real-time communication that is computationally inexpensive.

The system described in this thesis, is able to communicate real-time, continuous, interactions to a set of robot behaviours. Being able to capture and make use of the spatio-temporal properties of the incoming signals makes a possible extension to voice quite straightforward. The spatio-temporal properties of voice signals can be easily handled by the methods presented, resulting to a continuous, real-time, system that can enable the communication between human and machine, or human to human. The 'Common Behavioural Space' put forward in chapter 6 (seen in figure 6.5) could be created between human-human behaviours (not human-robot), thus providing an adaptive and intuitive scaffolding for human to human communication. Our notion of behaviours is relatively loose allowing for most spatio-temporal signals to be treated as such. In creating this space we are able to create something analogous to a language, grounded to any of the signals the patient is able to broadcast.

There exists research pointing towards this direction, such as Patel and Roy [171] where they investigate teachable interfaces that can adapt to the preferences and abilities of the individual user. A more recent research line follows the Brain Computer Interfaces, as current technological advancements allow for noninvasive recording of brain waves. Such systems are used to people with severe speech and physical impairments (SSPI), although there still don't exist systems applicable to real world cases [172]. In other studies, Moore et al. 2016 make use of modern smartphones enabling individuals with severe dysarthria to build their own system of communication through speech or gesture input [173]. Such systems use much simpler methods to achieve their results, specifically Dynamic Time Warping, which shortcomings compared to our method have been shown in chapter 5.

In total, although the field of AAC was not targeted in this thesis, our methodological outcomes and final system can benefit applications in this research field. Given the plethora of users groups in AAC, adaptive technologies provide a way forward in tackling the complexity and diversity of medical cases. Instead of working with a ‘mean’ description of the each case the work in this thesis gives us the unique ability to embrace the complexity of the problem and provide communication pathways created by, and tailored to, the individual. At the same time, having no assumptions on the devices to be used by the proposed system, we can take a second step forward providing a unique interface for each user, rather that simply giving the possibility to adapt a given one. In this, the user can bring in their own ‘high-tech aiding’ equipment. Doing so, has the potential to ease the adaptation of the user, working with familiar devices, having already explored and acquainted oneself with their affordances.

List of references

Berto Gonzalez, Erin Carroll, and Celine Latulipe. Dance-inspired technology, technology-inspired dance. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, pages 398–407. ACM, 2012.

Masataka Niwa, Shinya Okada, Shota Sakaguchi, Kentaro Azuma, Hiroyuki Iizuka, Hideyuki Ando, and Taro Maeda. Detection and transmission of 'tsumori': an archetype of behavioral intention in controlling a humanoid robot. In *Proceeding of 20th International Conference on Artificial Reality and Telexistance (ICAT2010)*, pages 197–201, 2010.

Kimitaka Watanabe and Yosuke Yoneda. The world's smallest biped humanoid robot 'i-sobot'. In *Advanced Robotics and its Social Impacts (ARSO), 2009 IEEE Workshop on*, pages 51–53. IEEE, 2009.

Kouichi Watanabe, Ichiro Kawabuchi, Naoki Kawakami, Taro Maeda, and Susumu Tachi. Torso: Development of a telexistence visual system using a 6-dof robot head. *Advanced Robotics*, 22(10):1053–1073, 2008.

Terrence Fong and Charles Thorpe. Vehicle teleoperation interfaces. *Autonomous robots*, 11(1):9–18, 2001.

Ralf Der. Selforganized robot behavior from the principle of homeokinesis. *Proc. workhop soave*, pages 39–46, 2000.

Georg Martius, Katja Fiedler, and J Michael Herrmann. Structure from behavior in autonomous agents. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 858–862. IEEE, 2008.

Robert A. Wilson and Lucia Foglia. Embodied cognition. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2017 edition, 2017.

Thompson E Varela, F and E Rosch. *The embodied mind: Cognitive science and human experience*. Cambridge, MA: MIT Press, 1991.

J Kevin O'Regan and Alva Noë. A sensorimotor account of vision and visual consciousness. *Behavioral and brain sciences*, 24(05):939–973, 2001.

David Philipona, J Kevin O'Regan, and J-P Nadal. Is there something out there? inferring space from sensorimotor dependencies. *Neural computation*, 15(9):2029–2049, 2003.

James J Gibson. The theory of affordances. *Hilldale, USA*, 1977.

Frédérique de Vignemont. Acting for bodily awareness, 2014.

Yann Coello and Martin H Fischer. *Perceptual and emotional embodiment: Foundations of embodied cognition*, volume 1. Routledge, 2015.

Brian O'shaughnessy. *The will: a dual aspect theory*, volume 2. JSTOR, 1980.

Adrian John Tetteh Smith. Acting on (bodily) experience. *Psyche*, 15(1):82–99, 2009.

Donald A Norman. *The psychology of everyday things*. Basic books, 1988.

Donald A Norman. Affordance, conventions, and design. *interactions*, 6(3):38–43, 1999.

Stefano Nolfi. Behaviour as a complex adaptive system: On the role of self-organization in the development of individual and collective behaviour. *ComplexUs*, 2(3-4):195–203, 2004.

Judea Pearl. Behavior-based robotics. *The MIT Encyclopedia of the Cognitive Sciences*, page 74, 2001.

Ronald C Arkin. *Behavior-based robotics*. MIT press, 1998.

Georg Martius. Taming the Beast: Guided Self-Organization of Behavior in Autonomus Rboots. 2010.

R. Der, F. Hesse, and G. Martius. Learning to feel the physics of a body. *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, 2:252–257, 2005.

Ralf Der and Georg Martius. From motor babbling to purposive actions: Emerging self-exploration in a dynamical systems approach to early robot development. *From Animals to Animats 9*, pages 406–421, 2006.

Frank Hesse, Georg Martius, Ralf Der, and J. Michael Herrmann. A Sensor-Based Learning Algorithm for the Self-Organization of Robot Behavior. *Algorithms*, 2(1):398–409, March 2009.

Kyung-Joong Kim and Sung-Bae Cho. A comprehensive overview of the applications of artificial life. *Artificial life*, 12(1):153–82, January 2006.

I Harvey, P Husbands, D Cliff, A Thompson, and N Jakobi. Evolutionary robotics at sussex. In *Proc. Intl. Symposium on Robotics and Manufacturing*, 1996.

Inman Harvey. Artificial evolution and real robots. *Artificial Life and Robotics*, 1(1):35–38, 1997.

Stefano Nolfi and Dario Floreano. Synthesis of autonomous robots through evolution. *Trends in cognitive sciences*, 6(1):31–37, 2002.

Stefano Nolfi and Dario Floreano. *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. MIT press, 2000.

Stefano Nolfi. Evolutionary robotics: Exploiting the full power of self-organization. *Connection Science*, 10(3-4):167–184, 1998.

James Cannan and Huosheng Hu. Human-Machine Interaction (HMI): A Survey. *University of Essex*, 2011.

Stuart K. Card, Jock D. Mackinlay, and George G. Robertson. The design space of input devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 117–124, New York, NY, USA, 1990. ACM.

Jock Mackinlay, Stuart K. Card, and George G. Robertson. A semantic analysis of the design space of input devices. *Hum.-Comput. Interact.*, 5(2):145–190, June 1990.

Jae Kyu Suhr, Ho Gi Jung, Kwanghyuk Bae, and Jaihie Kim. Automatic free parking space detection by using motion stereo-based 3d reconstruction. *Machine Vision and Applications*, 21(2):163–176, 2010.

P Jia and H Hu. Active shape model-based user identification for an intelligent wheelchair. *International Journal of Advanced Mechatronic Systems*, 1(4):299–307, 2009.

MN Mamatha and S Ramachandran. Automatic eyewinks interpretation system using face orientation recognition for human-machine interface. *IJCSNS International Journal of Computer Science and Network Security*, 9(5):155–163, 2009.

I Scott MacKenzie. An eye on input: research challenges in using the eye for computer input control. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, pages 11–12. ACM, 2010.

Damien Michel, Antonis A Argyros, Dimitris Grammenos, Xenophon Zabulis, and Thomas Sarmis. Building a multi-touch display based on computer vision techniques. In *MVA*, pages 74–77, 2009.

JA Clark and RB Roemer. Voice controlled wheelchair. *Archives of physical medicine and rehabilitation*, 58(4):169–175, 1977.

Alfonso Gárate, Nati Herrasti, and Antonio López. Genio: an ambient intelligence application in home automation and entertainment environment. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, pages 241–245. ACM, 2005.

Ivan Tashev, Michael Seltzer, Yun-Cheng Ju, Ye-Yi Wang, and Alex Acero. Commute ux: Voice enabled in-car infotainment system. In *Mobile HCI*, volume 9, 2009.

Jun-Hyung Park, Hyun Suk Lee, Ju Seok Jeong, and Tae-Jeong Jang. Realization of a vibro-tactile glove type mouse. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, pages 255–256. ACM, 2009.

Nathan Miller, Odest Chadwicke Jenkins, Marcelo Kallmann, and Maja J Mataric. Motion capture from inertial sensing for untethered humanoid teleoperation. In *Humanoid Robots, 2004 4th IEEE/RAS International Conference on*, volume 2, pages 547–565. IEEE, 2004.

R.E. Ellis, O.M. Ismaeil, and M.G. Lipsett. Design and evaluation of a high-performance haptic interface. *Robotica*, 14:321–327, 5 1996.

Chao Hu, M.Q. Meng, P.X. Liu, and Xiang Wang. Visual gesture recognition for human-machine interface of robot teleoperation. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1560–1565 vol.2, Oct 2003.

R. Marin, P.J. Sanz, P. Nebot, and R. Wirz. A multimodal interface to control a robot arm via the web: a case study on remote programming. *Industrial Electronics, IEEE Transactions on*, 52(6):1506–1520, Dec 2005.

JL Alty and G Guida. The use of rule-based system technology for the design of man-machine systems. In *2nd IFAC Conference on Analysis and Evaluation of ManâĂŤ-Machine Systems, Varese, Italy*, 1985.

Ben Shneiderman. *Designing the user interface-strategies for effective human-computer interaction*. Pearson Education India, 1986.

H Rex Hartson and Deborah Hix. Human-computer interface development: concepts and systems for its management. *ACM Computing Surveys (CSUR)*, 21(1):5–92, 1989.

Mark Green. Report on dialogue specification tools. In *Computer Graphics Forum*, volume 3, pages 305–313. Wiley Online Library, 1984.

J Rasmussen. Mental models and their implications for design. In *Austrian Computer Society 6th Workshop on Informatics and Psychology*, 1987.

Kim J Vicente and Jens Rasmussen. Ecological interface design: Theoretical foundations. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(4):589–606, 1992.

C Reising Dal Vernon and Penelope M Sanderson. Work domain analysis and sensors i: principles and simple example. *International Journal of Human-Computer Studies*, 56(6):569–596, 2002.

Timothy R McEwen, John M Flach, and Nancy C Elder. Interfaces to medical information systems: Supporting evidenced based practice. In *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, pages 335–340. IEEE, 2014.

Daniel S Hall, Lawrence G Shattuck, and Kevin B Bennett. Evaluation of an ecological interface design for military command and control. *Journal of Cognitive Engineering and Decision Making*, 6(2):165–193, 2012.

Clark Borst, HCH Suijkerbuijk, Max Mulder, and MM Van Paassen. Ecological interface design for terrain awareness. *The International Journal of Aviation Psychology*, 16(4):375–400, 2006.

William Ross Ashby et al. An introduction to cybernetics. *An introduction to cybernetics.*, 1956.

Clark Borst, John M. Flach, and Joost Ellerbroek. Beyond Ecological Interface Design: Lessons From Concerns and Misconceptions. *IEEE Transactions on Human-Machine Systems*, 45(2):164–175, April 2015.

Joseph Sullivan, Joseph W Sullivan, Sherman W Tyler, et al. Intelligent user interfaces. 1994.

William E Hefley and Dianne Murray. Intelligent user interfaces. In *Proceedings of the 1st international conference on Intelligent user interfaces*, pages 3–10. ACM, 1993.

M Waleed Kadous, Raymond Ka-Man Sheh, and Claude Sammut. Effective user interface design for rescue robotics. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 250–257. ACM, 2006.

Michael a. Goodrich and Alan C. Schultz. Human-Robot Interaction: A Survey. *Foundations and TrendsÂő in Human-Computer Interaction*, 1(3):203–275, 2007.

Liying Cheng, Qi Sun, Han Su, Yang Cong, and Shuying Zhao. Design and implementation of human-robot interactive demonstration system based on kinect. In *Control and Decision Conference (CCDC), 2012 24th Chinese*, pages 971–975, May 2012.

Guanglong Du, Ping Zhang, Jianhua Mai, and Zeling Li. Markerless kinect-based hand tracking for robot teleoperation. *International Journal of Advanced Robotic Systems*, 9(10), 2012.

Alessandra Sciutti, Lars Schillingmann, Oskar Palinko, Yukie Nagai, and Giulio Sandini. A gaze-contingent dictating robot to study turn-taking. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*, pages 137–138. ACM, 2015.

James S Albus. Outline for a theory of intelligence. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):473–509, 1991.

Gordon Beavers and Henry Hexmoor. Types and limits of agent autonomy. In *International Workshop on Computational Autonomy*, pages 95–102. Springer, 2003.

Alex Meystel. Multiresolutional autonomy. In *Intelligent Control (ISIC), 1998. Held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS), Proceedings*, pages 516–519. IEEE, 1998.

Jacob W Crandall, Michael A Goodrich, Dan R Olsen, and Curtis W Nielsen. Validating human-robot interaction schemes in multitasking environments. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 35(4):438–449, 2005.

Thomas B Sheridan. *Humans and automation: System design and research issues*. John Wiley & Sons, Inc., 2002.

Mica R Endsley. *Designing for situation awareness: An approach to user-centered design*. CRC press, 2016.

Joan H Johnston, Stephen M Fiore, Carol Paris, and CA Smith. Application of cognitive load theory to developing a measure of team decision efficiency. Technical report, DTIC Document, 2002.

Gary Klein, Paul J Feltovich, Jeffrey M Bradshaw, and David D Woods. Common ground and coordination in joint activity. *Organizational simulation*, 53:139–184, 2005.

Jodi Forlizzi and Carl DiSalvo. Service robots in the domestic environment: a study of the roomba vacuum in the home. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 258–265. ACM, 2006.

Illah R Nourbakhsh, Clayton Kunz, and Thomas Willeke. The mobot museum robot installations: A five year experiment. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 4, pages 3636–3641. IEEE, 2003.

Candace L Sidner, Cory D Kidd, Christopher Lee, and Neal Lesh. Where to look: a study of human-robot engagement. In *Proceedings of the 9th international conference on Intelligent user interfaces*, pages 78–84. ACM, 2004.

Oliver Brock, Andrew Fagg, Roderic Grupen, Robert Platt, Michael Rosenstein, and

John Sweeney. A framework for learning and control in intelligent humanoid robots. *International Journal of Humanoid Robotics*, 2(03):301–336, 2005.

Monica N Nicolescu and Maja J Mataric. Learning and interacting in human-robot domains. *IEEE Transactions on Systems, man, and Cybernetics-part A: Systems and Humans*, 31(5):419–430, 2001.

Jerry Alan Fails and Dan R Olsen Jr. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 39–45. ACM, 2003.

Joe Saunders, Chrystopher L Nehaniv, and Kerstin Dautenhahn. Teaching robots by moulding behavior and scaffolding the environment. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 118–125. ACM, 2006.

Cynthia Breazeal, Daphna Buchsbaum, Jesse Gray, David Gatenby, and Bruce Blumberg. Learning from and about others: Towards using imitation to bootstrap the social understanding of others by robots. *Artificial life*, 11(1-2):31–62, 2005.

Deb K Roy. Learning visually grounded words and syntax for a scene description task. *Computer speech & language*, 16(3):353–385, 2002.

Joe Saunders, Chrystopher L Nehaniv, and Kerstin Dautenhahn. Using self-imitation to direct learning. In *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, pages 244–250. IEEE, 2006.

J Gregory Trafton, Alan C Schultz, Dennis Perznowski, Magdalena D Bugajska, William Adams, Nicholas L Cassimatis, and Derek P Brock. Children and robots learning to play hide and seek. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 242–249. ACM, 2006.

David W Franklin, Theodore E Milner, and Mitsuo Kawato. Single trial learning of external dynamics: What can the brain teach us about learning mechanisms? In *International Congress Series*, volume 1301, pages 67–70. Elsevier, 2007.

CA Ratanamahatana and E Keogh. Everything you know about dynamic time warping is wrong. *Third*, 2004.

GA Ten Holt, MJT Reinders, and EA Hendriks. Multi-dimensional dynamic time warping for gesture recognition. 2007.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. *ICML (3)*, 28:1310–1318, 2013.

Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

M Ito and Jun Tani. On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system. *Adaptive Behavior*, 12:93–115, 2004.

Jun Tani. Learning to generate articulated behavior through the bottom-up and the top-down interaction processes. *Neural Networks*, 16(1):11 – 23, 2003.

Michael I Jordan. Attractor dynamics and parallellism in a connectionist sequential machine. 1986.

Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

Herbert Jaeger. A tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the ‘echo state network’ approach. 2002. *Frauenhofer Institue for Autonomous Intelligent*, 2002:1–46, 2005.

Ralf Der and Georg Martius. *The playful machine: theoretical foundation and practical realization of self-organizing robots*, volume 15. Springer Science & Business Media, 2012.

Bernard Widrow, Marcian E Hoff, et al. Adaptive switching circuits. 1960.

SimÃşnC. Smith and J.Michael Herrmann. Homeokinetic reinforcement learning. In *Partially Supervised Learning*, volume 7081. Springer Berlin Heidelberg, 2012.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5, 1988.

Juergen Leitner, M Luciw, A Förster, and J Schmidhuber. Teleoperation of a 7 dof humanoid robot arm using human arm accelerations and emg signals. 2014.

Danelle Shah, Joseph Schneider, and Mark Campbell. A Sketch Interface for Robust and Natural Robot Control. *Proceedings of the IEEE*, 100(3):604–622, mar 2012.

Guillaume Doisy, Joachim Meyer, and Yael Edan. A Gesture Learning Interface for Simulated Robot Path ShapingWith a Human Teacher. *IEEE Transactions on Human-Machine Systems*, 44(6):788–795, dec 2014.

Neal Y. Lii, Zhaopeng Chen, Maximo a. Roa, Annika Maier, Benedikt Pleintinger, and Christoph Borst. Toward a task space framework for gesture commanded telemanipulation. *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 925–932, sep 2012.

Ying Yin. Real-time Continuous Gesture Recognition for Natural Multimodal Interaction. *Bulletin of Center for Educational Research and Development*, (21):1–8, 2014.

Saša Bodiroža, Helman I. Stern, and Yael Edan. Dynamic gesture vocabulary design for intuitive human-robot dialog. *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction - HRI '12*, page 111, 2012.

Christos Melidis and Davide Marocco. An exploration on intuitive interfaces for robot control based on self organisation. In Christopher J. Headleand, William J. Teahan, and Llyr Ap Cenydd, editors, *Artificial Life and Intelligent Agents*, volume 519 of *Communications in Computer and Information Science*, pages 73–79. Springer International Publishing, 2015.

Ralf Der, Georg Martius, and Frank Hesse. Let it roll - emerging sensorimotor coordination in a spherical robot. *ARTIFICIAL LIFE X: PROC. X INT. CONF. ON*, 2006.

Herbert Jaeger. *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the 'echo state network' approach*. GMD-Forschungszentrum Informationstechnik, 2002.

Sushmita Mitra and Tinku Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(3):311–324, 2007.

Natalia Neverova, Christian Wolf, Graham W Taylor, and Florian Nebout. Multi-scale deep learning for gesture detection and localization. In *Workshop at the European Conference on Computer Vision*, pages 474–490. Springer, 2014.

Saša Bodiroža, Guillaume Doisy, and Verena Vanessa Hafner. Position-invariant, real-time gesture recognition based on dynamic time warping. *HRI - ACM/IEEE International Conference on Human-Robot Interaction*, pages 87–88, 2013.

Z. Ren, J. Yuan, J. Meng, and Z. Zhang. Robust part-based hand gesture recognition using kinect sensor. *IEEE Transactions on Multimedia*, 15(5):1110–1120, Aug 2013.

Dan Xu, Xinyu Wu, Yen-Lun Chen, and Yangsheng Xu. Online dynamic gesture recognition for human robot interaction. *Journal of Intelligent & Robotic Systems*, 77(3-4):583–596, 2015.

D. Xu, Y. L. Chen, C. Lin, X. Kong, and X. Wu. Real-time dynamic gesture recognition system based on depth perception for robot navigation. In *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, pages 689–694, Dec 2012.

Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4207–4215, 2016.

Gonzalo Bailador, Daniel Roggen, Gerhard Tröster, and Gracián Triviño. Real time gesture recognition using continuous time recurrent neural networks. In *Proceedings of the ICST 2nd international conference on Body area networks*, page 15, 2007.

Tin Hninn Hninn Maung et al. Real-time hand tracking and gesture recognition system using neural networks. *World Academy of Science, Engineering and Technology*, 50:466–470, 2009.

Eleni Tsironi, Pablo Barros, and Stefan Wermter. Gesture Recognition with a Convolutional Long Short-Term Memory Recurrent Neural Network In : Proceedings of the European Symposium on Artificial Neural Networks ,. *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 213–218, 2016.

Doreen Jirak, Pablo Barros, and Stefan Wermter. Dynamic gesture recognition using echo state networks. In *Proceedings*, page 475. Presses universitaires de Louvain, 2015.

D. Wu, L. Pigou, P. J. Kindermans, N. D. H. Le, L. Shao, J. Dambre, and J. M. Odobez. Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1583–1597, Aug 2016.

Michal ČerÅĹanský and Peter TiÅĹo. Comparison of echo state networks with simple recurrent networks and variable-length Markov models on symbolic sequences. *Artificial Neural Networks - ICANN 2007*, pages 618–627, 2007.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Chunyang Sheng, Jun Zhao, Ying Liu, and Wei Wang. Prediction for noisy nonlinear time series by echo state network based on dual estimation. *Neurocomputing*, 82:186–195, 2012.

Rodrigo Laje and Dean V Buonomano. Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nature neuroscience*, 16(7):925–933, 2013.

David Sussillo and Larry F Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, 2009.

Lars Büsing, Benjamin Schrauwen, and Robert Legenstein. Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. *Neural computation*, 22(5):1272–1311, 2010.

Herbert Jaeger. The 'echo state' approach to analysing and training recurrent neural networks-with an erratum note.

Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *science*, 304(5667):78–80, 2004.

G Manjunath and Herbert Jaeger. Echo state property linked to an input: Exploring a fundamental characteristic of recurrent neural networks. *Neural computation*, 25(3):671–696, 2013.

R. Felix Reinhart and Jochen J. Steil. Reaching movement generation with a recurrent neural network based on learning inverse kinematics for the humanoid robot iCub. *9th IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS09*, pages 323–330, 2009.

Jochen Triesch. A gradient rule for the plasticity of a neuronâĂŹs intrinsic excitability. In *International Conference on Artificial Neural Networks*, pages 65–70. Springer, 2005.

Jochen J Steil. Online reservoir adaptation by intrinsic plasticity for backpropagation–decorrelation and echo state learning. *Neural Networks*, 20(3):353–364, 2007.

Benjamin Schrauwen, Marion Wardermann, David Verstraeten, Jochen J. Steil, and Dirk Stroobandt. Improving reservoirs using intrinsic plasticity. *Neurocomputing*, 71(7 - 9):1159 – 1171, 2008.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013.

Luiza Mici, Xavier Hinaut, and Stefan Wermter. Activity recognition with echo state networks using 3D body joints and objects category. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 465 – 470, Bruges, Belgium, April 2016.

Dariu M Gavrila. The visual analysis of human movement: A survey. *Computer vision and image understanding*, 73(1):82–98, 1999.

Adam Kendon. Current issues in the study of gesture. *The biological foundations of gestures: Motor and semiotic aspects*, 1:23–47, 1986.

Lukas Rybok, Boris Schauerte, Ziad Al-Halah, and Rainer Stiefelhagen. âĂIJimportant stuff, everywhere!âĂİ activity recognition with salient proto-objects as context. In *IEEE Winter Conference on Applications of Computer Vision*, pages 646–651. IEEE, 2014.

Cornelius Weber, Kazuhiro Masui, Norbert Michael Mayer, Jochen Triesch, and Minoru Asada. Reservoir computing for sensory prediction and classification in adaptive agents. *Machine Learning Research Progress*, 2008.

Stuart K Card, George G Robertson, and Jock D Mackinlay. The information visualizer, an information workspace. In *Proceedings of the SIGCHI Conference on Human factors in computing systems*, pages 181–186. ACM, 1991.

Robert B Miller. Response time in man-computer conversational transactions. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 267–277. ACM, 1968.

J Leitner, M Luciw, and A Förster. Teleoperation of a 7 DOF Humanoid Robot Arm Using Human Arm Accelerations and EMG Signals. *juxi.net*, 2014.

Qiangfu Zhao, John Brine, and Dimitar P. Filev. Defining Cybernetics. *IEEE Systems, Man, & Cybernetics Magazine*, 1(2):18–26, 2015.

Jean Scholtz. Theory and evaluation of human robot interactions. *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, 3:10 pp., 2003.

Herbert A Simon and Allen Newell. Human problem solving: The state of the theory in 1970. *American Psychologist*, 26(2):145, 1971.

D M Wolpert, R C Miall, and M Kawato. Internal models in the cerebellum. *Trends in Cognitive Sciences*, 2(9):338–347, 1998.

Andrew N Meltzoff and M Keith Moore. Explaining Facial Imitation: A Theoretical Model. *Early development & parenting*, 6(June):179–192, 1997.

Jean Piaget. *The origin of intelligence in the child*. Routledge & Paul, 1953.

Claes Von Hofsten. An action perspective on motor development. *Trends in cognitive sciences*, 8(6):266–272, 2004.

Yiannis Demiris and Anthony Dearden. From motor babbling to hierarchical learning by imitation: a robot developmental pathway. pages 31–37, 2005.

Pietro Morasso and Vittorio Sanguineti. Self-organizing body schema for motor planning. *Journal of Motor Behavior*, 27(1):52–66, 1995.

Ralf Der, Ulrich Steinmetz, Frank Pasemann, et al. Homeokinesis-a new principle to back up evolution with learning. 1999.

Helmut Hauser, Auke J Ijspeert, Rudolf M Füchslin, Rolf Pfeifer, and Wolfgang Maass. Towards a theoretical foundation for morphological computation with compliant bodies. *Biological cybernetics*, 105(5-6):355–370, 2011.

Rolf Pfeifer. *Morphological Computation: Connecting Brain, Body, and Environment*, pages 2–3. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

Yuichi Yamashita and Jun Tani. Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment. *PLoS Comput Biol*, 4(11):e1000220, 11 2008.

Jun Namikawa, Ryunosuke Nishimoto, and Jun Tani. A neurodynamic account of spontaneous behaviour. *PLoS Comput Biol*, 7(10):e1002221, 10 2011.

John F Kolen and Stefan C Kremer. *A field guide to dynamical recurrent networks*. John Wiley & Sons, 2001.

Christos Melidis and Davide Marocco. A human centric approach to robotic control. In *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, pages 2441–2446, Oct 2015.

Tetsuya Ogata and Shigeki Sugano. Imitation based Human-Robot Interaction -Roles of Joint Attention and Motion Prediction-. pages 283–288, 2004.

Ezequiel A Di Paolo. Organismically-inspired robotics: homeostatic adaptation and teleology beyond the closed sensorimotor loop. *Dynamical systems approach to embodiment and sociality*, pages 19–42, 2003.

Stefano Nolfi. Evolutionary Robotics: Exploiting the Full Power of Self-organization. *Connection Science*, 10(3-4):167–184, 1998.

Joe Saunders, Chrystopher L. Nehaniv, and Kerstin Dautenhahn. Teaching robots by moulding behavior and scaffolding the environment. In *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction*, HRI '06, pages 118–125, New York, NY, USA, 2006. ACM.

Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Francesco Mondada, Michael Bonani, Xavier Raemy, James Pugh, Christopher Cianci, Adam Klaptocz, Stephane Magnenat, Jean-Christophe Zufferey, Dario Floreano, and Alcherio Martinoli. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions*, volume 1, pages 59–65. IPCB: Instituto Politécnico de Castelo Branco, 2009.

M Zambelli and Y Demiris. Online multimodal ensemble learning using self-learned sensorimotor representations. *IEEE Transactions on Cognitive and Developmental Systems*, 2016.

Ivana Kajic, Guido Schillaci, S Bodiroza, and Verena V Hafner. A biologically inspired model for coding sensorimotor experience leading to the development of pointing behaviour in a humanoid robot. In *Proceedings of the Workshop" HRI: a bridge between Robotics and Neuroscience". 9th ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI 2014)*, 2014.

Martin F Stoelen, Fabio Bonsignorio, and Angelo Cangelosi. Co-exploring actuator antagonism and bio-inspired control in a printable robot arm. In *International Conference on Simulation of Adaptive Behavior*, pages 244–255. Springer, 2016.

Rupal Patel and Deb Roy. Teachable interfaces for individuals with dysarthric speech and severe physical disabilities. In *Proceedings of the AAAI Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 40–47, 1998.

M. Akcakaya, B. Peters, M. Moghadamfalahi, A. R. Mooney, U. Orhan, B. Oken, D. Erdogmus, and M. Fried-Oken. Noninvasive brain x2013;computer interfaces for augmentative and alternative communication. *IEEE Reviews in Biomedical Engineering*, 7:31–49, 2014.

Meredith Moore and Sethuraman Panchanathan. Translatable: Giving individuals with complex communication needs a voice through speech and gesture recognition. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 321–322. ACM, 2016.

Bound copies of published papers

An Exploration on Intuitive Interfaces for Robot Control Based on Self Organisation

Christos Melidis and Davide Marocco

School of Computing and Mathematics, Plymouth University, Plymouth, UK
{christos.melidis, davide.marocco}@plymouth.ac.uk

Abstract. In this paper we present the results of a preliminary study on behaviour extraction from arbitrary robotic morphologies. Our goal is to build a universal interface targeting all possible robotic morphologies. For the exploration of the capabilities of different morphologies, we focus on the self organisation of the sensorimotor loop for discovering behavioural capabilities. In this paper we briefly explain the core idea for such an interface and present preliminary results of our method together with future remarks.

Keywords: robotics, self organisation, operator behaviour acquisition

1 Introduction

The remote control of mechanical devices equipped with a large number of actuators, such as humanoid robots, is a challenging task. When dealing with the resulting large number of degrees-of-freedom, the nature of the interface provided to the human operator plays a fundamental role in the success of tele-robotic performance. A wide range of tele-robotic interfaces have been explored so far; some are very rigid devices that require a great deal of cognitive and manual effort, while other more intuitive systems, based on one-to-one body mapping, are in contrast very complex and expensive devices, often specifically tailored to a single robotic platform [3].

Our goal is the implementation of an agile interface able to control every possible robotic morphology, a universal interface. To do so we need an automated mechanism that can examine and explore the robotic morphology connected to the interface and extract interesting features, with respect to the desired control pattern. Our interest in this preliminary study is movement control. We identify interesting features as behaviours that can be produced by the robot and are meaningful to the user, according to the task in hand. The purpose of the interface is to map the behaviours of the operator to those produced by the robots, resulting in the association between the robots and operator behaviours. In order to achieve this, we reverse the informational flow of the interface, as suggested in [6]. The robot acts first and the operator responds to the exhibited behaviour with his own, through the input device. The input device thus, plays a critical part on the behaviours the operator can have. Multidimensional input devices, i.e. Kinect sensor, could enable a whole body mapping, whereas simpler ones, i.e. on-off switches or joysticks, are more restrictive [8], [9] and [2].

The interface is able to explore the capabilities of the robotic morphology based on the homeokinesis principle [1]. As described by Martius and colleagues in [5], self organisation of the sensorimotor loop can explore the behavioural repertoire of a robot. Based on this research we formulate the principles for the interaction between the interface and the robot. For the interaction between the interface and the human operator we propose a framework for a behaviour based interaction, though currently only capable of exhibiting a simple example of such interaction. In this study, we explore the applicability of the proposed method for behavioural exploration of the robotic morphology.

1.1 Operator Behaviour Acquisition

In this section we describe the main ideas guiding the interaction of the operator with the interface. As previously stated, the overall goal is to build a novel interface that connects intuitive human behaviours to robotics ones. Our approach follows the research described in [6]. In their approach they define the interaction between the user and the interface as an "intention translation" mechanism, by which user intentions are translated to instructions or commands that the interface can understand, so that the user can interact with it. In most interfaces users have to familiarise themselves with the interface in order to interact with it, read the user manual and understand the predefined mechanisms of interaction [4]. In a more complex interaction paradigm, where the actions to be performed are formed using simpler actions as building blocks, the user has to learn sequences of controls in order to communicate their intentions to the interface. In such case, as the number of sequences, and so, the building blocks increase, the more laborious it becomes for the user to remember and execute them.

Providing a mapping between user intentions and robot behaviours can lead to an intuitive interface. The operator's intentions are taken into account -through the manipulation of the input device- making the interfacing process easier and more personalised. In this reversed paradigm, users do not have to familiarise themselves with the interface, but the interface can learn from the interaction with the user. Based on the reactions of the user to the exhibited behaviours of the robot, the interface is able to correlate the two, forming a control pattern. For that to happen, a level of consistency is expected from the users in the behaviours they exhibit. Same or similar input signals should be expected to yield the same robot behaviour as a response. Studies carried out, on a similar approach show up to 80% percent mapping accuracy in the interaction with a 17 degree of freedom robot, using an input device consisting of two joysticks [6].

2 Materials and Methods

Based on the principles explained in the introduction, we implemented a system consisting of two modules. One used for the exploration and self organisation of

the sensorimotor loop of the robot and one for the extraction, storage and reuse of the acquired behaviours. The robotic morphologies used for the experiments described in this paper where simulated by the Open Dynamics Engine, ODE [7]. The module for the self organisation of the sensorimotor loop was implemented according to the system described in [5] and follows a dynamical system approach. The realization of the dynamics of the robot and the world is done using a Controller (K) and World Model (W) cooperating for the effective exploration of the robots dynamics and an accurate prediction of world states, respectively. Both are described by the equations bescribed below.
The exploration module, in general, is described, according to time t, as:

$$\tilde{\mathbf{x}}_{t+1} = W(K(\mathbf{x}_t, C), A) \tag{1}$$

The controller K generates motor outputs $\mathbf{y}_t = K(\mathbf{x}_t, C)$ as a function of sensory inputs $\mathbf{x} = x_1, x_2, \ldots, x_n$, in dependence on a set of parameters defined by the matrix $C_{n,n+1}$ and is defined by the equation:

$$K = g(\sum_{i=1}^{n} C_i x_i + C_{n+1}), \tag{2}$$

where g is a sigmoid function.
The world model $\tilde{\mathbf{x}}_{t+1} = W(\mathbf{y}_t, A)$ estimates future sensory inputs $\tilde{\mathbf{x}}_{t+1}$ from motor outputs $\mathbf{y}_t = y_1, y_2, \ldots, y_n$ in dependence on a set of parameters defined by the matrix $A_{n,n+1}$.
The parameter matrix of the world model, A, is adapted according to the Widrow - Hoff Learning Rule [10], delta rule, $\Delta w = +\eta E_W \mathbf{x}$ with the error, E_W, described by the function:

$$E_W = ||\mathbf{x}_{t+1} - \tilde{\mathbf{x}}_{t+1}||^2 \tag{3}$$

with learning rate $\eta = 0.1$.
The controller updates its parameter matrix by gradient descent with respect to the error function,

$$E_K = ||\mathbf{x}_t - \tilde{\mathbf{x}}_t||^2 \tag{4}$$

To calculate the above error, we find the $\tilde{\mathbf{x}}_t$ by calculating the motor input $\hat{\mathbf{y}}_t$ the world model should have in order to make a perfect prediction and then the sensory input the controller K should have to predict the motor output $\tilde{\mathbf{y}}_t$. The update on the controller parameter follows the rule $C_{t+1} = C_t - \epsilon \frac{\partial E_K}{C}$, with a learning rate $\epsilon = 0.01$.
For the identification, storage and reuse of the different behaviours exhibited by the robot, we use a series of m neural networks. Each network is defined according to the equation,

$$(\mathbf{x}_{t+1}, \mathbf{y}_t) = N_i(\mathbf{x}_t, \mathbf{x}_{t-1}), \quad i = 1, \ldots, m \tag{5}$$

The neural networks, working in parallel, compete for the prediction of the motor command $\mathbf{y}_t$ of time t and the sensory input $\mathbf{x}_{t+1}$ of the next time step. It is a winner takes it all method, with only the winning network being allowed to

train on the current data $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$. Because of that, each network specializes to a region of the possible motoric and sensory space.
The networks consist of 3 layers, input, output and a hidden layer. The hidden layer consists of sigmoid units whereas the input and output layers from linear units. No bias units are introduced in the networks.
The algorithm for the training of the networks is backpropagation, with learning rate $\eta = 0.01$. In each time step all the networks are activated with the same input and the one with the best approximation of the next sensor values and motor commands is selected as the winning network. The sample won is then added to the training dataset of the winning network and it is trained for another epoch. For the selection of network, a smoothed error is used, taking into account the past errors of the network.

3 Results

In this section we present the experimental results of the exploration method and the way by which the interface controls the different behaviours extracted. For testing purposes we applied the method to three different robotic morphologies as seen in figure 1, with varying degrees of freedom and numbers of joints. The acrobot has 1; figure 1(a), the octacrawl has 2; figure 1(b) and the arm has 18; shoulder, elbow and wrist pitch together with finger pitch for three joints in every finger, figure 1(c). In figure 2 we can see how the experts are trained

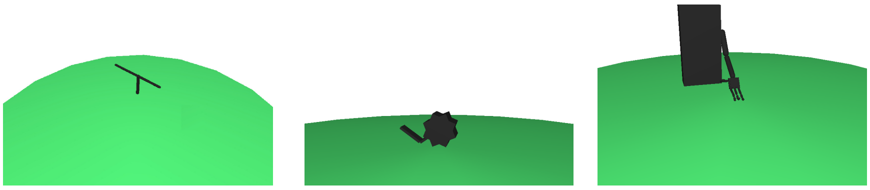

(a) Acrobot with 1 d.o.f. (b) Octacrawl with 2 d.o.f. (c) Arm with 18 d.o.f.

Fig. 1: The different robotic morphologies used during the experiments

to identify different sensor states. Here, only a couple of behaviours extracted from the octacrawl morphology are displayed. As we can see from the graph, the outputs of the network, describing each behaviour -as captured by the sensor values- stabilize and approximate the real ones more accurately as time and training size increment. In the example of figure 2, behaviour 1 stabilizes faster that behaviour 2 as we can see from the convergence to a finite set of sensor values for each behaviour. This is caused by the difference in the size of the datasets for each behaviour. Some behaviours are more frequent than others making the dataset of the network describing them to increase in size faster than others. We can also observe a periodicity in the sensory values recorded, a direct result of the dynamical system approach used in the exploration mechanism. A behaviour is usually found when the system enters a basin of attraction, and a long-time behaviour is exhibited by the system as it approaches the attraction point.

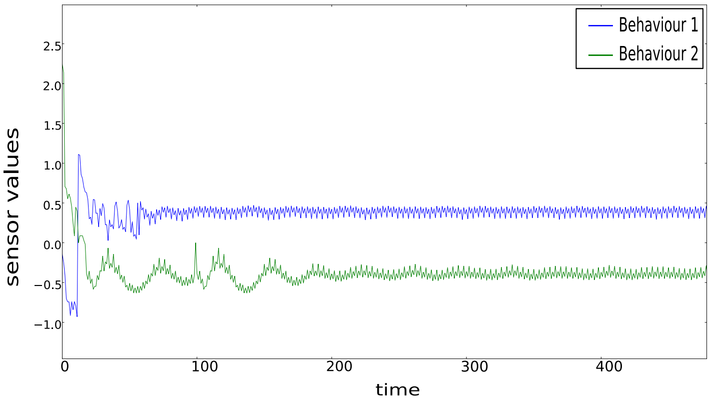
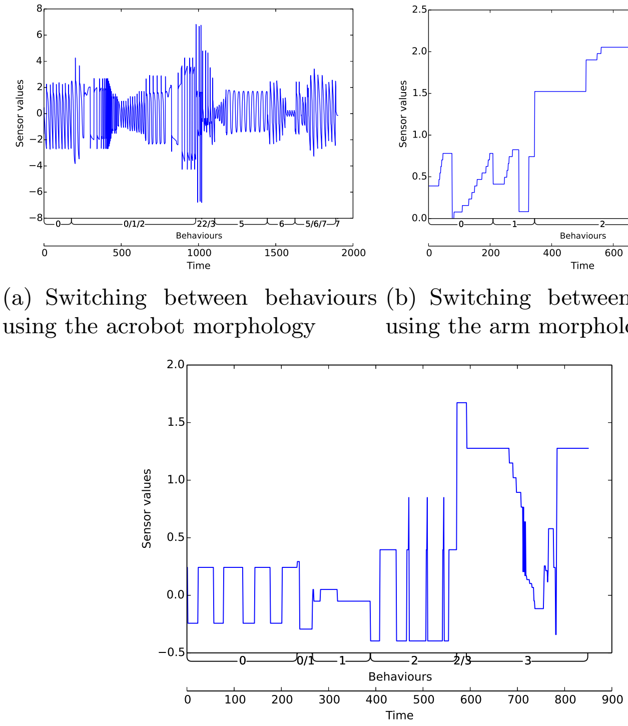


Fig. 2: Plot of the sensor values for two different behaviours extracted from the octacrawl morphology, as they change through time during the learning phase

Even more interesting features of the system can be observed in the switching between behaviours. In figure 3(a) the behavioural changes of the acrobot morphology are being displayed against time. The different behaviours become salient by the different sensor readings they produce. In figure 3(c) and 3(b) the behaviours of the octacrawl morphology and the arm morphology are being displayed against time, respectively. Our interest in these graphs lies in the point of change between behaviours. We exhibit a behaviour by activating the corresponding network. The id of the active network is noted in the horizontal axis, above time. For the rest of this section, behavioural change results from the change of network in charge. So, whenever a behavioural change is stated, the reader should keep in mind that the network in charge has changed in order to support the different dynamics dictated by the behaviour.
In all cases the exploration mechanism was able to identify and extract different behaviours. Theses behaviours where triggered through the interface in random order and the sensor values of each morphology were recorded and predicted by the network in control. In all graphs of the figure 3 we observe smooth changes in the sensory recordings, regardless of the changes in behaviours. The system, readjusts itself, following a trajectory to the new attractor, described by the network in control each instant. In the first time steps following a behavioural change, we can observe the readjustment of the morphology, as recorded through

the sensor values, so as to exhibit the desired behaviour.



(a) Switching between behaviours using the acrobot morphology (b) Switching between behaviours using the arm morphology

(c) Switching between behaviours using the octacrawl morphology

Fig. 3: Switching between behaviours in the different morphologies used. In the horizontal axis we have time, and the id of the expert(s) at control of the system. The expert id is displayed and when two or more experts are in control at the same time, their ids are separated with '/'. In the vertical axis the sensor values of the each robot are being displayed.

At the same graphs of figure 3 we can also observe the behaviour of the system in the case of simultaneous activations. In the horizontal axis we can see the behaviours exhibited by the morphology, separated with '/' when more than one behaviours are triggered. In the co-activation of behaviours we have the ability through the interface to adjust the level in which each behaviour contributes to the resulting one. The behaviours displayed in the graphs have been equally contributing to the behaviour exhibited, but experiments with different levels of contribution yield similar results. From the graphs we can see the ability of the system to mix the behaviours acquired seamlessly with no abnormal sensory readings or resulting behaviours being exhibited by the morphology. In figures 3(a), 3(c) we observe the change in sensor values through time for the acrobot and octacrawl morphologies respectively.

4 Conclusions

In this preliminary study of the proposed interfacing mechanism we were able to show that the proposed exploration mechanism for robot behaviours was successfully implemented. The robustness of the proposed mechanism was shown, both by the stability of the mechanism when switching between the explored behaviours, and by the ability of the explored behaviours to be combined together, potentially exhibiting more complex behaviours. The next step, will be the implementation of an interface based on the proposed interfacing principles, able to support continuous interaction with the user. Once the user is able to provide continuous feedback based on the robots behaviour, we could use that to guide the exploration of the behaviours towards desired ones, depending on the task. On the exploratory mechanism, a proposed extension would be the reuse of the extracted behaviours inside the self organising mechanism so as to guide the exploration towards more complex and fine grained behaviours. In this case the user could be the one deciding which behaviours should be extended and which not, tailoring the interface system according to their needs.

References

1. Der, R.: Selforganized robot behavior from the principle of homeokinesis. Proc. workhop soave pp. 39–46 (2000)
2. Fong, T., Thorpe, C.: Vehicle teleoperation interfaces. Autonomous robots 11(1), 9–18 (2001)
3. Gonzalez, B., Carroll, E., Latulipe, C.: Dance-inspired technology, technology-inspired dance. In: Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design. pp. 398–407. ACM (2012)
4. Kadous, M.W., Sheh, R.K.M., Sammut, C.: Effective user interface design for rescue robotics. In: Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction. pp. 250–257. ACM (2006)
5. Martius, G., Fiedler, K., Herrmann, J.M.: Structure from behavior in autonomous agents. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on. pp. 858–862. IEEE (2008)
6. Niwa, M., Okada, S., Sakaguchi, S., Azuma, K., Iizuka, H., Ando, H., Maeda, T.: Detection and transmission of" tsumori": an archetype of behavioral intention in controlling a humanoid robot. In: Proceeding of 20th International Conference on Artificial Reality and Telexistance (ICAT2010). pp. 197–201 (2010)
7. Smith, R., et al.: Open dynamics engine (2005)
8. Watanabe, K., Yoneda, Y.: The world's smallest biped humanoid robot i-sobot. In: Advanced Robotics and its Social Impacts (ARSO), 2009 IEEE Workshop on. pp. 51–53. IEEE (2009)

9. Watanabe, K., Kawabuchi, I., Kawakami, N., Maeda, T., Tachi, S.: Torso: Development of a telexistence visual system using a 6-dof robot head. Advanced Robotics 22(10), 1053–1073 (2008)
10. Widrow, B., Hoff, M.E., et al.: Adaptive switching circuits. (1960)

A Human Centric Approach to Robotic Control

Christos Melidis
School of Computing, Electronics and Mathematics
Plymouth University
Plymouth, United Kingdom
christos.melidis@plymouth.ac.uk

Davide Marocco
School of Computing, Electronics and Mathematics
Plymouth University
Plymouth, United Kingdom
davide.marocco@plymouth.ac.uk

***Abstract*—In this paper we present a novel idea for the creation of an intelligent interface that allows the remote control of arbitrarily complex robotics morphologies by translating intuitive human behaviours into purposeful robotic actions. By taking inspiration from human robot interaction, ergonomic principles, and autonomous robotics this paper proposes a human-centric framework for robot control inspired by the current advancements in recurrent neural networks and self-organisation. In particular, we present an integrated approach based on neural networks for input acquisition from human operator and self-organisation for the acquisition of robot behaviours. We realise the interface as a kind of intelligent agent connecting the two end points of the system: Human and robot, providing an adaptive and intelligent interface for robot control. The present preliminary study shows the on-going results of the proposed methodology for both self-exploration of robotic morphologies and acquisition of human behaviours.**

***Index Terms*—robotics, self-organisation, human behaviour acquisition, human-centric system, remote control, neural networks**

I. Introduction

Human robot interaction and remote control have long been surging fields for both research communities and commercial markets. Whether robotic morphologies are built for entertainment purposes or for more “serious” applications, the remote control of such robots is one of the main forms of interaction between humans and robots.

Different types of robot morphologies have found applications in industry, rescue missions and military operations. In this context, we observe that the specific task often defines the constraints of the robot morphology and of the control mechanisms, and dictates the interfacing approach. Most systems, being specifically constructed for a given task, are designed with the restrictions already embedded into their operational mechanisms. This approach makes the robot usable and controllable by a human operator, but also drastically constrains its usage to a very specific context. To this end, a wide range of tele-robotic interfaces have been explored so far, some are very rigid devices that require a great deal of cognitive and manual effort, while other, more intuitive systems based on one-to-one body mapping, are in contrast very complex and expensive devices, often specifically tailored to a single robotic platform [1].

Tele-operation of complicated mechanical devices requires a great deal of knowledge about the interfacing mechanisms and the robotic morphology at hand, both from the operator and from the designer of the controlling interface, in order to make the interface tailored to the given robot and, often, to the operator (e.g. see the unique controllers designed to accommodate different types of motor disabilities). To obtain such knowledge the operator has to undergo a suitable training on the use of the interface. For the communication of a continuous control sequence, for example, in most of the approaches available so far, the operator has to pass a sequence of commands through a controlling device. This can be difficult to remember and prone to mistakes.

Starting from these observations, we aim to design an integrated methodology focused on the human and capable of seamlessly translating any type of human motion into meaningful robotics actions and behaviours, something that we can call, as the title suggest, a human-centric approach in designing tele-operation of robotic morphologies. The main concept is based on the principle that the interfacing controller should be capable to adaptively ”understand” and translate human motion into controlling commands for the robot, rather than having the human, the operator, learning the use of the interface. The proposed approach, therefore, relies more on the intelligence and learning capabilities of the controlling interface rather than on those of humans. We expect this should ease the cognitive demands in both designing and using the controlling device.

At the same time, we aim to extend the above idea to suit the control of arbitrary robotic morphologies. By exploiting techniques of self-organisation of robot behaviours, we are able to extract a behavioural repertoire of the robot to be controlled, without knowing in advance the kinematics and dynamics that characterise the morphology.

Ultimately, the interface can be regarded as a kind of cognitive agent that seats in between the robots and the human, and that try to minimise the control errors, adapting towards the robot and the operator at the same time. An agent that serves the human operator, while understanding the controlled robot.

II. Background

Two fundamental elements for constructing this kind of interface are based on understanding and constructing methods for autonomous exploration of robot behaviours on one hand, and finding a suitable methodology for human-machine interaction on the other.

A. Control of Robotic Systems

Controlling a robotic system can be a very difficulty task, depending on the morphology of the robot. Robots with 1 or 2 Degrees Of Freedom (DoF) can be easy to control, such as simple two-wheeled robots. Indeed, the control can have a comparable complexity of that of a remote controlled toy car. On the other hand, complex arrangements such as 4 or 6 legged robots, or humanoids, can be very difficult to control, especially for non-standard operational tasks (i.e., not simply going forward-backward and turning). In this cases, the designer of the controlling device has to decide the level of expected autonomy of the robot by implementing a series of controlling patterns of various complexity and abstraction, such as high level commands (i.e. proceed to the next room) or low level commands (i.e. arrange a specific joint to certain degrees). In most cases the level of expected autonomy of the robot is driven by the task and the goal.

In the case of robots with no level of autonomy the control is based upon the direct manipulation of the robot's DoF. In the case of remote control, the input device needs to have at least the same amount of DoF so that the operator can achieve full functionality of the robotic morphology [2]. Examples of such control techniques can be found in [3] using a full body mapping or part of it as in [4].

On the other hand, traditional Artificial Intelligence research follows a top-down approach in designing robot controllers, usually involving a complicated, centralised controller that makes decisions based on access to all aspects of the global state. There are though systems build from a bottom-up approach, where localized, parallel, and distributed low-level controller provide the robot with adaptive and complex behaviours. Behaviour Based robotics [5], Nonlinear Dynamics and Self Organisation [6], and Evolutionary Robotics [7] are research fields developing systems that follow this bottom-up approach.

For our purposes, particularly interesting is the non-linear dynamics approach put forward by Ralf Der and the homeokinesis principle [6], which is a representative example of a the bottom-up approach in robot control and exploits self-organisation. Other examples based on the same principle that exploit self-organisation of the sensorimotor loop in robotics morphologies can be found in the work of Martius et. al. [8] and Hesse et. al [9]. In their approach they use Neural Networks to show how from simple structures and non-linear approximations, behaviours can be discovered in robots with varying morphologies. The idea of goal oriented behaviours is not stated in their research, but has been pursued by others using Reinforcement Learning techniques to guide the exploration [10].

B. Human Machine Interaction

Thus far, Human Machine Interaction (HMI) systems are tightly designed around the applications and the machines to be operated. The design of interfaces to be used and the possible interactions between the human and the machine are typically based on ergonomic principles [11]. In terms of HMI, ergonomics relates to how the user will interact with a machine and how easy that interaction will be. The main goal of ergonomics can be stated as, the design of equipment which is, *a*) *Easy to remember*; *b*) *Easy to learn*; *c*) *Efficient to use*; *d*) *Effective to use*; *e*) *Enjoyable to use*; and *f*) *Safe to use*, for the user.

The concept of *affordances* was first introduced by J.J. Gibson [12]. It described the potential actions enabled by a given object, especially ones that is easily discoverable. The concept of affordance is applicable on the way we perceive control devices, as different people have the possibility of acting in a different way upon them. In this way the interface has the possibility to adapt to the user. This idea carries one of the most important aspects of the interfacing framework described here and allows the user to interact with the device in an intuitive way. We define here *Intuition* as the ability to understand something instinctively, without the need for conscious reasoning. Combining intuition with affordances permits to design an interface tailored for the user. Enabling the user to freely express the way of communicating their intentions for control through the interface provides us with a new way of dealing with ergonomics.

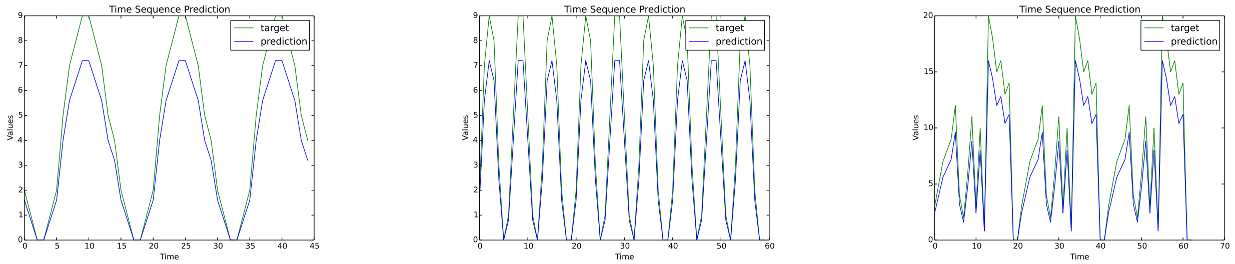
C. Intelligent User Interfaces

The merge of artificial intelligence and human-computer interaction brings forward the idea of Intelligent Interfaces [13]. In their studies on intelligent user interfaces, Hefley et. al [14], they describe intelligent interfaces as systems *that build on facts and heuristic knowledge of an expert, together with techniques for reasoning about unstructured situations.* In their research they use user interface management systems (UIMS) concepts as a basis for their research on intelligent interfaces. They distinguish between adaptive and flexible intelligent interfaces, with the first having the added capability to learn over time from experience to accommodate the user and their interaction, while flexible interfaces deal with cases in which the user can tailor the interface or when the interface can support several styles of interaction.

III. Extraction and Acquisition of Human and Robot Behaviours

In this sections the methodologies and the main principles at the basis of the implementation of a framework capable of supporting human intuitive control of robotic morphologies are described.

From an HMI point of view, following the work on humanoid robot control suggested by [15], operator's intention for control is captured as time varying configurations of an input device. In this paper a single methodology for the intuitive control of a humanoid robots is discussed, while in our approach we try to address the more general problem of acquisition of motion behaviour from the operator as a problem of time sequence recognition. In particular, we aim to capture the human intention for controlling the robot as time depended manipulations of the input device. Indeed, interfacing well defined segmented manipulations of the input



(a) Prediction over a ”slow” sine signal

(b) Prediction over a ”fast” cosine signal

(c) Prediction over a random signal

Fig. 1. Prediction capabilities of the network, once trained over three time sequences

device for remote control purposes is not different from the classical idea of having a set of discrete switches, such as a keyboard, with the switches replaced by time sequences. On the other hand, being able to provide a way for real time recognition of the operators behaviours on the input device is a challenging problem, still not conclusively solved in the field of human machine interaction. In our approach, the operator is free to perform any movement, as long as it is captured by the input device. In that sense, all hand movements are acceptable using a Leap Motion device, all body movements for a Kinect sensor and all buttons pressed in any order for any button based input device.

For the point of view of self-organisation and exploration of robot's behaviours, we present a model based on dynamical system which is capable to perform an exploration and extraction of behaviours of random robotic morphologies. With the term random, we suggest the idea of not having a single target morphology for the application of the interface. By coupling this system with a series of neural networks we are able to capture and reuse these behaviours as show in section III-B1. Behaviours include movements that are in general guided by the DoF of the robot, both translational and rotational. Also, based on the dynamics of the robot, more abnormal behaviours may appear, i.e. the octacrawl robot balancing on its tail. The results of the method are shown in section III-B2.

A. Interfacing the Human Operator

As stated before, the main challenge from an HMI perspective is that of properly sequencing and recognising the manipulation of an input device. To this end, among the many available solutions we concentrated mostly on Dynamic Time Warping and Recurrent Neural Networks.

Dynamic Time Warping (DTW) is a distance measure used mainly in speech recognition community. It allows a non-linear mapping of one signal on another by minimizing the distance of the two. The DTW algorithm calculates the distance between each possible pair of points out of two signals in terms of their associated feature values. It uses these distances to calculate a cumulative distance matrix and finds the least expensive path through this matrix. This path represents the ideal warp - the synchronisation of the two signals which causes the feature distance between their synchronised points to be minimised [16]. Although DTW can provide a good measure of resemblance in time sequences it can only work once the control sequence is completed by the operator and their behaviour is captured. In the case of partial data, the sequence cannot provide enough information, even if stretched or squeezed in time, mainly because the method does not have the ability of completing a sequence by predicting the expected time steps.

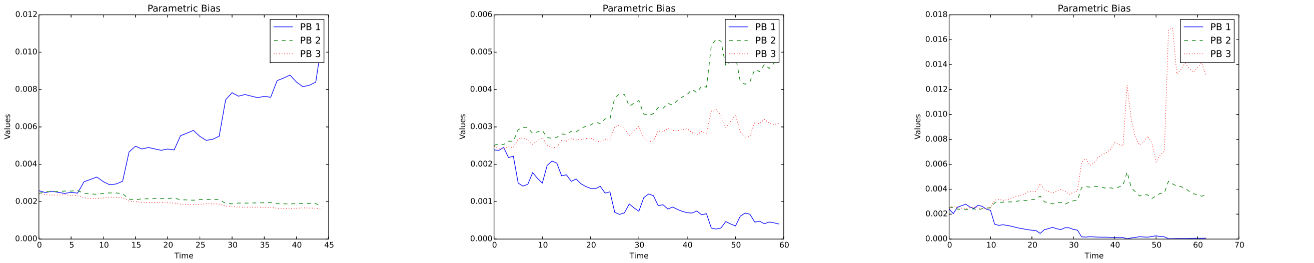
Therefore, for flexibility reasons we focused our attention on Recurrent Neural Networks (RNN). Although this method produces a delay in the setup of the interface, due to the training of the RNN, the computational complexity of a trained RNN is very small and the representation of the trained sequences is very compact (the synaptic weights). In addition to DTW, RNN have also the ability to predict the next time steps according to the dynamic of the input, making the recognition faster and often without the need of presenting the full sequence.

1) Recurrent Neural Networks: There are many implementation paradigms for creating RNNs. Our approach is based on Jun Tani's works both in time sequence recognitions [17] and multiple time scales dynamics acquisition [18]. The main difference being that in his work the RNN has a *Jordan* type structure (recurrency on the output layer) and it is trained with Back-propagation Through Time (BPTT) algorithm [19]. In our implementation we implemented an *Elman* type structure with recurrency on the hidden layer, trained with standard Back-propagation [20].

The idea of Parametric Biases (PB) provides a way for both generation and recognition of dynamic temporal patterns. PBs are units in the input layer of the network capable of adjusting themselves according to networks dynamics. During the training phase and after the error has been propagated to the weights of the networks, the values of the PB units are adjusted, trying to further minimise the difference between the target and actual output. The update equations for the ith PB unit at time t are,

$$\delta\rho_t^i = k_{bp} \sum_{step=t-l/2}^{t+l/2} \delta_{step}^{bp^i} + k_{nb}(\rho_{t+1}^i - 2\rho_t^i + \rho_{t-1}^i) \qquad (1)$$

$$\Delta\rho_t^i = \epsilon\delta\rho_t^i + \eta\Delta\rho_{t-1} \qquad (2)$$



(a) Parametric Bias settling for the sine signal

(b) Parametric Bias settling for the cosine signal

(c) Parametric Bias settling for the random signal

Fig. 2. Recognition of the sequence by the self adaptation of the Parametric Bias units

$$\rho_t^i = f(\rho_t) \tag{3}$$

the term $\delta\rho_t$, the delta component of the internal value of the PB unit, is calculated by the summation of two terms. The fist one represents the summation of the delta error propagated from the output units for l time steps, while the second term works as a low pass filter, inhibiting rapid changes in the PB unit, with k_{nb} being the coefficient for the filter. The update $\Delta\rho_t^i$ utilised a momentum term, the second term of the summation, for faster convergence. Finally, the activation of the PB unit is given through function f, a sigmoid function.

This updating step of the PB unit values is then performed, in real-time, once the network is trained. Given the network N, the temporal sequence $S(t)$ is generated as $S(t+1) = N(S(t))$. The input received at time $t+1$ represents the desired output of the network having as input the one of time t. Once the networks is trained the PB units are initialised with value zero, and a sequence is fed element-by-element to the network.The difference of the network output o and the desired output $S(t+1)$ in time $t+1$ can be calculated, given the known input from time t. Since the network is trained, the only possible source of error can be assigned to the PB units. If an error appears, that error value is propagated to the PB units, updating them according to the same set of equations as in the training phase. In this way the network readjusts the PB units values, to match the target output as best as possible. The resulting PB values give us the ability to identify whether an unknown sequence is a sequence the network was trained on, since the PB values of the training phase are known.

In the graphs we can observe both the ability of the RNN to predict the time sequences fed to it as in figure 1 and the recognition of the sequence as in figure 2. An important aspect of the setup, is the time step it takes for the network to recognise a given sequence. The figures 2(a), 2(b), 2(c) represent the sequences in 1(a), 1(b), 1(c) with lengths of 15, 20 and 20 respectively. We can see that the PB values settle very early, and they provide a correct recognition from steps 6, 5 and 12 respectively.

B. Interfacing the Robotic Morphology

1) Building the Behaviour Exploration Mechanism for the Robots: The autonomous discovery of available behaviours of a given morphology is the other fundamental element for designing an interfacing system that aims to reduce design constraints and maximise usability. To this end, we implemented a system consisting of two modules. One used for the exploration and self-organisation of the sensorimotor loop of the robot to be controlled and one for the extraction, storage and reuse of the acquired robot's behaviours. The robotic morphologies used for the experiments described in this paper are simulated with Open Dynamics Engine, ODE. The module for the self-organisation of the sensorimotor loop follows a dynamical system approach. The realization of the dynamics of the robot and of the world is done using a Controller (K) and World Model (W) cooperating for the effective exploration of the robots dynamics and an accurate prediction of world states, respectively, as discussed in [8]. Both are described by the equations described below.

The exploration module is described, according to time t, as:

$$\tilde{\mathbf{x}}_{t+1} = W(K(\mathbf{x}_t, C), A) \tag{4}$$

The controller K generates motor outputs $\mathbf{y}_t = K(\mathbf{x}_t, C)$ as a function of the sensory input $\mathbf{x} = x_1, x_2, \ldots, x_n$, depending on a set of parameters defined by the matrix $C_{n,n+1}$ and it is defined by the equation:

$$K = g(\sum_{i=1}^{n} C_i x_i + C_{n+1}), \tag{5}$$

where g is a sigmoid function.

The world model $\tilde{\mathbf{x}}_{t+1} = W(\mathbf{y}_t, A)$ estimates future sensory input $\tilde{\mathbf{x}}_{t+1}$ from the motor output $\mathbf{y}_t = y_1, y_2, \ldots, y_n$ depending on a set of parameters defined by the matrix $A_{n,n+1}$.

The parameter matrix of the world model, A, is adapted according to the delta rule [21], $\Delta w = +\eta E_W \mathbf{x}$ with the error, E_W, described by the function:

$$E_W = ||\mathbf{x}_{t+1} - \tilde{\mathbf{x}}_{t+1}||^2 \tag{6}$$

with learning rate η = 0.1.

The controller updates its parameter matrix by gradient descent with respect to the error function,

$$E_K = ||\mathbf{x}_t - \tilde{\mathbf{x}}_t||^2 \tag{7}$$

To calculate the above error, we find the $\tilde{\mathbf{x}}_t$ by calculating the motor input $\hat{\mathbf{y}}_t$ the world model should have in order to make a perfect prediction and then, the sensory input the controller K should have to predict the motor output $\tilde{\mathbf{y}}_t$. For updating the controller parameters the following rule is applied $C_{t+1} = C_t - \epsilon \frac{\partial E_K}{C}$, with a learning rate $\epsilon = 0.01$.

For the identification, storage and reuse of the different behaviours exhibited by the robot, we use a series of m neural networks (NNs), called *experts*. Each NN is defined according to the equation,

$$(\mathbf{x}_{t+1}, \mathbf{y}_t) = N_i(\mathbf{x}_t, \mathbf{x}_{t-1}), \quad i = 1, \dots, m \tag{8}$$

The NNs, working in parallel, compete for the prediction of the motor command $\mathbf{y}_t$ of time t and the sensory input $\mathbf{x}_{t+1}$ of the next time step in a winner-takes-all method, with only the winning network being allowed to train on the current data $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$. Thanks to this process, each NN specialises to represent a region of the entire sensorimotor space of the robot.

The NNs consist of 3 layers, feedforward units where the hidden layer consists of sigmoid units, whereas the input and output layers of linear units. Online back-propagation is used to training the NN with learning rate $\eta = 0.01$. In each time step all NNs are activated with the same input and the one with the best approximation of the next sensor values and motor commands is selected as the winner. The sample is then added to the training dataset of the winning NN and it is trained for another epoch.

2) Robot Behaviour Exploration, Extraction and Reuse Results: For testing purposes we applied the method described above to three different robotic morphologies, as seen in figure 3, with varying degrees of freedom and numbers of joints, respectively 1, figure 3(a), 2, figure 3(b) and 18, figure 3(c).

In figure 4 we can see how the experts are trained to identify different sensorimotor loops in the robot with 2 joints: The output of the network, describing each behaviour, as captured by the sensor values, stabilise and approximate the real ones more accurately as time and training size increase. In the example of figure 4, behaviour 1 stabilises faster than behaviour 2 as we can see from the convergence to a finite set of sensor values for each behaviour. This is caused by the difference in the size of the datasets for each behaviour. Some behaviours are more frequent than others making the dataset of the network describing them bigger and more accurate during the training phase. We can also observe a periodicity in the sensory values recorded, a direct result of the dynamical system approach used in the exploration mechanism. A consistent and stable over time behaviour is usually found when the system enters a basin of attraction, and progressively approaches the attraction point.

The behaviours observed vary between the morphologies explored. In the octacrawl morphology the method discovers among others, a way of moving forward, a way of jumping and a movement of the tail without changing position. Similarly, in the acrobot, behaviours include standing still upside, variable rotation speeds and a pendulum like behaviour. Finally, in the hand were in comparison less behaviours are extracted, we have up down movement of the whole arm, bending at the elbow and wrist.

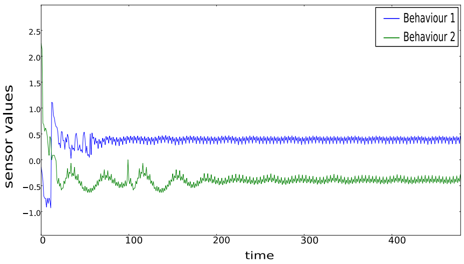


Fig. 4. Plot of the sensor values for two different behaviours extracted from the octacrawl morphology, as they change through time during the learning phase

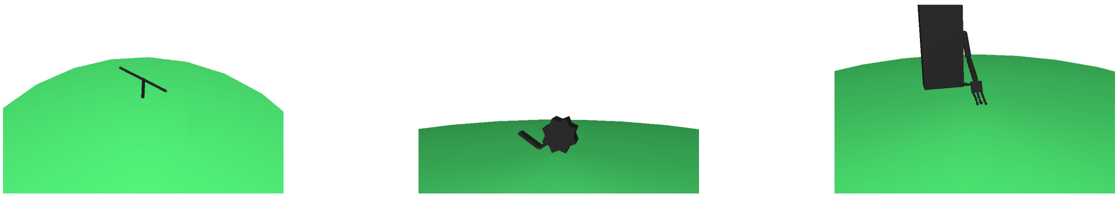
More interesting features of the system can be observed in the switching between behaviours. In figure 5(a) the behavioural changes of the robot with 1 joint are being displayed against time. The different behaviours become salient by the different sensor readings they produce. In figure 5(c) and 5(b) the behaviours of the 2-joints robot and the arm morphology are being displayed against time, respectively. Our interest in these graphs lies in the point of change between behaviours. In this context one behaviour is produced by activating the corresponding NN. The ID of the active network is noted in the horizontal axis, above time.

In all cases the exploration mechanism was able to identify and extract different behaviours. During testing these behaviours where triggered through the interface in random order and the sensor values of each morphology were recorded and correctly predicted by the network in control. In all graphs of the figure 5 we observe smooth changes in the sensory readings, regardless of the changes in behaviours. It appears that the system is able to produce and follow a trajectory from the old to the new attractor, and a consequently smooth transition in behaviours. In the first time steps following a behavioural change, we can observe the readjustment of the morphology, as recorded through the sensor values, smoothly moving towards the exhibition of the desired behaviour.

In the graphs of figure 5 it is also possible to observe the behaviour of the system in the case of simultaneous activations. When several behaviours are activated at the same time, results in the graphs show the ability of the system to mix the behaviours acquired seamlessly with no abnormal sensory readings, which indicates that no abnormal behaviours are exhibited by the morphology. In figures 5(a), 5(c) we can observe the change in sensor values through time for the the robots with 1 or 2 joints respectively.

IV. Conclusion

In this paper we have described the principles, the background and the methodology for implementing an interfacing mechanism that, when completed, will allow to control any type of robotic morphologies in an intuitive way through the manipulation of an arbitrary input device.

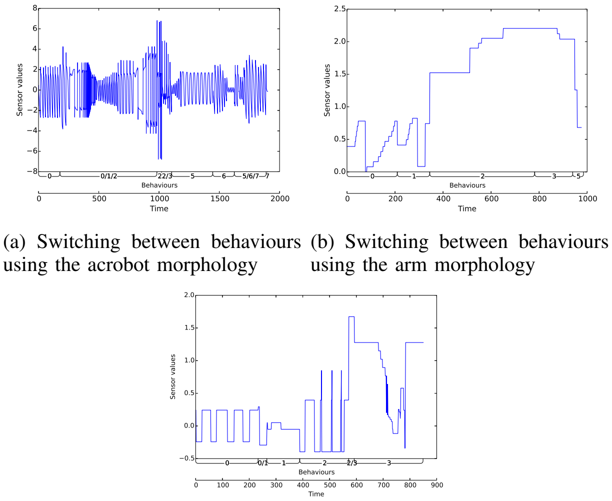


(a) Acrobot with 1 d.o.f.

(b) Octacrawl with 2 d.o.f.

(c) Arm with 18 d.o.f.

Fig. 3. The different robotic morphologies used during the experiments



(a) Switching between behaviours using the acrobot morphology

(b) Switching between behaviours using the arm morphology

(c) Switching between behaviours using the octacrawl morphology

Fig. 5. Switching between behaviours in the different morphologies used. In the horizontal axis the time and the ID of the expert(s) that control the robot. The figure show the expert ID and when two or more experts are in control at the same time the IDs are separated with '/'. The vertical axis shows the sensor values of the robot at a given time step.

In particular, despite the preliminary state of this research, the paper aims to show the methodological assumptions and technological building blocks of the proposed framework and the feasibility of the project, based on the testing results of the technologies.

The proposed exploration mechanism for robot behaviours was successfully implemented. The robustness of the system is shown, both by the stability of the mechanism when switching between the self-organised behaviours, and by the ability of combining such behaviours. At the same time, we propose a mechanism able to support continuous interaction with the operator. The preliminary results of the method implemented show good recognition and prediction capabilities, providing a viable solution to the problem.

The overall goal of this project is to build a novel interface that connects intuitive human behaviours to robots. Given a suitable representation of the robot morphology and controller, and an intelligent interface, we have the potential of reducing the complexity that the user has to face in the interaction with a robotic system. The complexity of the controlled robot can be reduced by self-organising behaviours and capture the complexity of human behaviour as it could be exhibited through an arbitrary input device.

REFERENCES

[1] B. Gonzalez, E. Carroll, and C. Latulipe, "Dance-inspired technology, technology-inspired dance," in *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*. ACM, 2012, pp. 398–407.

[2] W. R. Ashby *et al.*, "An introduction to cybernetics." *An introduction to cybernetics.*, 1956.

[3] L. Cheng, Q. Sun, H. Su, Y. Cong, and S. Zhao, "Design and implementation of human-robot interactive demonstration system based on kinect," in *Control and Decision Conference (CCDC), 2012 24th Chinese*, May 2012, pp. 971–975.

[4] G. Du, P. Zhang, J. Mai, and Z. Li, "Markerless kinect-based hand tracking for robot teleoperation," *International Journal of Advanced Robotic Systems*, vol. 9, no. 10, 2012.

[5] R. C. Arkin, *Behavior-based robotics*. MIT press, 1998.

[6] R. Der, "Selforganized robot behavior from the principle of homeokinesis," *Proc. workhop soave*, pp. 39–46, 2000.

[7] S. Nolfi and D. Floreano, *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. MIT press, 2000.

[8] G. Martius, K. Fiedler, and J. M. Herrmann, "Structure from behavior in autonomous agents," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 858–862.

[9] F. Hesse, G. Martius, R. Der, and J. M. Herrmann, "A Sensor-Based Learning Algorithm for the Self-Organization of Robot Behavior," *Algorithms*, vol. 2, no. 1, pp. 398–409, Mar. 2009.

[10] S. Smith and J. Herrmann, "Homeokinetic reinforcement learning," in *Partially Supervised Learning*. Springer Berlin Heidelberg, 2012, vol. 7081.

[11] B. Shneiderman, *Designing the user interface-strategies for effective human-computer interaction*. Pearson Education India, 1986.

[12] J. J. Gibson, "The theory of affordances," *Hilldale, USA*, 1977.

[13] J. Sullivan, J. W. Sullivan, S. W. Tyler *et al.*, "Intelligent user interfaces," 1994.

[14] W. E. Hefley and D. Murray, "Intelligent user interfaces," in *Proceedings of the 1st international conference on Intelligent user interfaces*. ACM, 1993, pp. 3–10.

[15] M. Niwa, S. Okada, S. Sakaguchi, K. Azuma, H. Iizuka, H. Ando, and T. Maeda, "Detection and transmission of" tsumori": an archetype of behavioral intention in controlling a humanoid robot," in *Proceeding of 20th International Conference on Artificial Reality and Telexistance (ICAT2010)*, 2010, pp. 197–201.

[16] C. Ratanamahatana and E. Keogh, "Everything you know about dynamic time warping is wrong," *Third*, 2004.

[17] M. Ito and J. Tani, "On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system," *Adaptive Behavior*, vol. 12, pp. 93–115, 2004.

[18] J. Tani, "Learning to generate articulated behavior through the bottom-up and the top-down interaction processes," *Neural Networks*, vol. 16, no. 1, pp. 11 – 23, 2003.

[19] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, 1988.

[21] B. Widrow, M. E. Hoff *et al.*, "Adaptive switching circuits." 1960.

KURE: Kinematic Universal Remote intErface A Human Centred Remote Robot Control Paradigm

Christos Melidis
School of Computing, Electronics and Mathematics
Plymouth University
Plymouth, United Kingdom
christos.melidis@plymouth.ac.uk

Davide Marocco
School of Computing, Electronics and Mathematics
Plymouth University
Plymouth, United Kingdom
davide.marocco@plymouth.ac.uk

***Abstract*—In this paper we present a novel approach to human-robot control. Taking inspiration from Behaviour Based robotics and self-organisation principles, we present an interfacing mechanism, named KURE in this paper, with the ability to adapt both towards the user and the robotic morphology. The aim is for a transparent mechanism connecting user and robot, allowing for a seamless integration of control signals and robot behaviours. Starting from a tabula rasa basis, KURE is able to identify control patterns (behaviours) for the given robotic morphology and successfully merge them with control signals from the user, regardless of the input device used. The structural components of the interface are presented and assessed both individually and as a whole.**

***Index Terms*—robotics, self-organisation, human behaviour acquisition, human-centric system, remote control, neural networks**

I. Introduction

This paper presents a new approach for the remote control of robotic morphologies. An approach that brings the operator closer to the controlled robot, enabling a more natural way of interaction. Our goal in this paper is to provide a methodology, together with examples to support it, that enables the remote control of arbitrary robotic morphologies, through arbitrary input devices. Given a robotic morphology we explore how it can be connected to, and controlled by an input device under the commands of a user.

Remote control, as it has been explored so far, is investigated using a specific robotic morphology [1], [2]. Targeting a specific robotic morphology, allows for robot specific solutions in control. Working with unknown robots, forces the methods followed here to be model-free, having no explicit - geometric - description of the robots kinematics. In addition, most studies of remote control are formed tightly around the input device to be used [3], [4] and are commonly constructed by field experts, to be used by field experts. To this end, our research follows current research in human-robot interaction and on flexible interfaces, were the interface is able to adapt to the specificities and preferences of each user. Our method aims towards a control paradigm suitable for both non-expert as well as expert users, as it does not require any knowledge about the robot, the control flow, or the input device. Most robotic remote control systems rely on the intelligence and cognitive capabilities of the operator to understand the control paradigm and the robot's capabilities. The operator once familiar with the input device, its functioning and its potentials, has to understand the control paradigm and how the control flows are defined to be used. Indeed, in most cases the operator needs to acquire the knowledge required for control through a training procedure (i.e. reading the manual for operation, practice on the controls) [5].

Furthermore, only the designers knowledge about the morphology is used. The morphology (i.e. the kinematics and dynamics) of the robot is not taken into account, resulting in inherent properties of the robot to remain unexplored. This can be argued to be not necessary or irrelevant in the control of robots with specific tasks. In our case, being agnostic towards the controlled robot highlights the necessity for a mechanism to explore the robots capabilities, with respect to its environment. It is through the interaction with the environment that the embodiment's properties can be revealed.

The problem we try to solve, is indeed a dual problem; merging the human operators control signals with the actions of the robotic morphologies. In doing so, we try to merge the two apparent dynamical systems involved. The one formed by the operators input signal and the other by the robots behaviour.

From the operators perspective being able to observe the robots embodiment interacting with the environment, allows for a better understanding of it. In this process, intentions for control can be formed. Being able to capture those intentions and associate them with the robots behaviour can result in a control flow - an intuitive control paradigm - tailored to each user and robot. This, with the operator being free to act upon the system - that acts as a mediating agent between user and robot - namely the interface.

II. Method

We see KURE as a human-centric system, capable of understanding both the operator (human) and the operated system (robot). Such interface should be able to place the operator in-the-loop seamlessly, regardless of the input device to be used and the robotic morphology at hand. For the human side, [6] mentions the importance of prediction in human robot interaction. We capture these real time manipulations of the input device by the operator, as they signal their intention for control. These behaviours (i.e. time depended manipulations), are then mapped to robotic behaviours, allowing for the operator to enter the behavioural loop of the robot. Operating at the joint level, the robot explores possibilities for movement. Self-organisation of the sensory-motor loop of the robotic morphology at hand provides the needed variety and complexity of behaviours without an explicit-external goal [7].

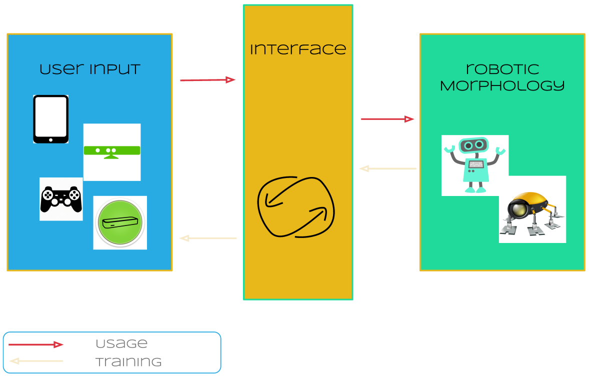


Fig. 1. Schematic representation of KURE

The system should work as a mediator between the robot and the operator; an interface connecting the two systems as seen in figure 1.

A. Self-organisation of Robotic Behaviours

In this section the methods for connecting the interface with the robotic morphology are discussed. Our goal is to explore the kinematics and dynamics of the robotic morphology as shaped through its interactions with the environment. This, together with a way of storing and reusing these behaviours. Indeed, in these sensory-motor contingencies of the robotic morphology at hand, small independent controllers can be formed [8], each one describing a behaviour of the robot.

To achieve this, a self-referential dynamical system is derived together with a principle for self-organisation of robotic behaviours [9], [10]. The idea here is to try and maintain a smooth control behaviour keeping the robot at a constant kinetic state. This property of the system (i.e. self-excitation) was first formulated by Ralph Der and referred to as homekinesis.

The learning in this procedure occurs based on the error between the real behaviour and the models prediction. Based on the homeokinetic principle the self-organisation of the sensory motor loop of the robotic morphology is possible without an external driving force (i.e. error or teacher signal). From this, a repertoire of behaviours emerges, which we are able to capture in the form of behavioural experts. These experts can later be reused and combined to control the robot. The behaviours, as demonstrated in the following section III-A, vary in complexity, time, and are entirely based on the robot, and the environment.

For the exploration of the robot's capabilities we work as seen in [11], [12]. We want to be able to produce motor outputs from sensory readings and from them predict the next sensory state of the robot. Creating a sensory-motor, and a motor-sensory mapping, allows us to derive an error signal for the update of the systems parameters.

For the two systems described above, the Controller K and the World Model W, their functions for operation are stated bellow. The exploration module is described, according to time t, as:

$$\tilde{\mathbf{x}}_{t+1} = W(K(\mathbf{x}_t, C), A) \tag{1}$$

The controller K generates motor outputs $\mathbf{y}_t = K(\mathbf{x}_t, C)$ as a function of the sensory input $\mathbf{x} = x_1, x_2, \ldots, x_n$, depending on a set of parameters defined by the matrix $C_{n,n+1}$ and it is defined by the equation, $K = g(\sum_{i=1}^{n} C_i x_i + C_{n+1})$, where g is a sigmoid function.

The parameter matrix of the world model, A, is adapted according to the delta rule [13], $\Delta w = +\eta E_W \mathbf{x}$ with the error, E_W, described by the function: $E_W = ||\mathbf{x}_{t+1} - \tilde{\mathbf{x}}_{t+1}||^2$ with learning rate η = 0.1.

The controller updates its parameter matrix by gradient descent with respect to the error function, $E_K = ||\mathbf{x}_t - \tilde{\mathbf{x}}_t||^2$ with $\tilde{\mathbf{x}}_t$ being the inverse solution of the Controller K for the motor commands of time t and the observed sensory state at $t + 1$.

For the identification, storage and reuse of the different behaviours exhibited by the robot, we use a series of m neural networks (NNs), called *experts*. Each NN is defined according to the equation,

$$(\mathbf{x}_{t+1}, \mathbf{y}_t) = N_i(\mathbf{x}_t, \mathbf{x}_{t-1}), \quad i = 1, \ldots, m \tag{2}$$

The NNs consist of 3 layers, feed-forward units where the hidden layer consists of sigmoid units, whereas the input and output layers of linear units. Back-propagation is used to training the NN with learning rate $\eta = 0.01$. The NNs, working in parallel, compete for the prediction of the motor command $\mathbf{y}_t$ of time t and the sensory input $\mathbf{x}_{t+1}$ of the next time step in a winner-takes-all method, with only the winning network being allowed to train on the current data $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$. Because of this competition, each of the expert NN specialises to represent a different behaviour of the robot.

In order to use these behaviours, the sensor values from the robot are passed in the selected NN and the respective motor commands (output of the NN) are applied to the robot.

B. User Input as a Continuous Signal

Combining the effect of multiple time scales and the possibility of mapping the time sequence dynamics to a fixed dimensional space, [14] formulated the echo state approach on training Recurrent Neural Networks, namely Echo State Networks (ESN). One of the most appealing features for our application, is the fact that the network is trained using linear regression on its output layer only, reducing the complexity of training with BPTT. The network is first presented with the input sequence and the values of the output units are replaced with the desired ones. The activation of the network based on the input is recorded and the output weights are computed through linear regression of the desired output on the network's state.

Echo State Networks (ESN) provide an architecture for efficient training of RNN in a supervised manner. One can distinguish two main components in an ESN. Firstly, the Dynamic Reservoir (DR), a large, randomly wired, recurrent neural network with fixed weights. The DR gets activated by the input and provides a non linear response for this input. And the output signal, which is trained as a linear combination of the activations of the DR. This way the computational resources and complexity required for training the RNN is reduced to the adaptation of the output connections of the ESN.

The state update equation, for an ESN consisting of N reservoir units, K inputs and L outputs - without any recurrent output neurons - is,

$$\mathbf{x}(n+1) = f(\mathbf{W}\mathbf{x}(n) + \mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}^{fb}\mathbf{y}(n)) \ , \quad (3)$$

where $\mathbf{x}(n)$ is the N-dimensional reservoir state, f is a sigmoid function (usually the logistic sigmoid or tanh function), $\mathbf{W}$ is the $N \times N$ reservoir weight matrix, $\mathbf{W}^{in}$ is the $N \times K$ input weight matrix, $\mathbf{u}(n)$ is the Kdimensional input signal, $\mathbf{W}^{fb}$ is the $N \times L$ output feedback matrix, and $\mathbf{y}(n)$ is the L-dimensional output signal. The extended system state $\mathbf{z}(n) = [\mathbf{x}(n); \mathbf{u}(n)]$ at time n is the concatenation of the reservoir and input states (and output in the case of output recurrency).

The output can be obtained from the extended system state by, $\mathbf{y}(n) = \mathbf{W}^{out}\mathbf{z}(n)$, where $\mathbf{W}^{out}$ is a $L \times (K+N)$-dimensional matrix of output weights.

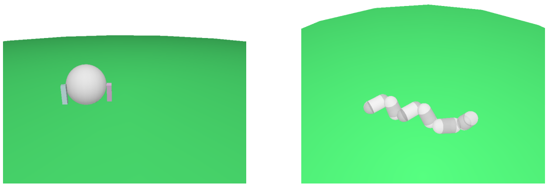
For the training of ESNs, assuming a driving signal $u(1), \ldots, u(n_{max})$, the extended states it generates -once passed to the network- are $z(1), \ldots, z(n_{max})$. We collect the states in matrix S of size $n_{max} \times (N+K)$ and the desired outputs $d(n)$ in a matrix D of size $n_{max} \times L$. Usually, before each collection, based on the properties of the network, we apply a washout period, allowing the network to settle to the input provided.

Now, the desired output weights W^{out} can be calculated as follows. First, the correlation matrix of the extended system states are calculated, $R = S'S$. Then the cross-correlation matrix of the extended states against the desired outputs d, $P = S'D$. Finally, for the calculation one can either calculate the Wiener- Hopf solution $W^{out} = (R^{-1}P)'$.

III. Experimental Results

In this section we describe the experimental setup and the results obtained by each individual component and the system as a whole.

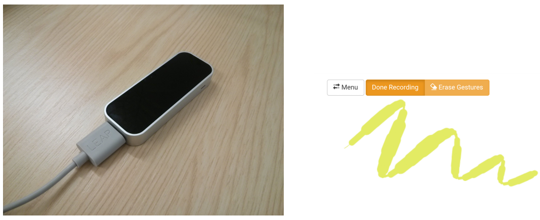
For our test scenarios we used two different - simulated- robotic morphologies (Fig. 2b, 2a) and two different input devices Fig. 3. The robotic morphologies were simulated using Open Dynamics Engine (ODE), through Python. This way, although simulated, the non-linearities and physical properties of the bodies are taken into account. Furthermore, latencies in the communication channels are dealt with twice. Firstly, by using buffered communication channels working under a tunnelling paradigm, where a channel is established only once. Secondly, during the adaptation procedure, where the robot and the input device are coupled through the users intentions for control using the channel with their potential latencies.



(a) The spherewalker morphology.

(b) The snake morphology.

Fig. 2. The two morphologies used in the experiments in the simulated environment.



(a) The Leap motion device

(b) The graphical interface for the touch-screen device.

Fig. 3. The two interfaces used in the experiments. In the graphical interface, the yellow line shows the gesture being recorded and disappears when the finger is not touching the screen.

The ”spherewalker” morphology has two motors each with 1 Degree of Freedom (DoF), and two sensors, measuring the joint positions. The ”snake” morphology has five motors each with 1 DoF, and five sensors, measuring the position of each joint. The values recorded from the sensors are normalised to fit $[-1, 1]$ in both morphologies.

Although robot kinematics are solved in most robot controllers, we work in a behavioural level. It is easy to see how a direct mapping of the robot's DOF can easily result in a bottleneck for the selected input device. On the other hand, even a two DOF input device can provide enough expressiveness for the control of a complex robots behaviours (e.g. touchscreen).

From the input devices, we capture a two dimensional signal from the touch-screen device, and a six dimensional signal from the Leap Motion device. In the case of the touch-screen device the input signal is 2-dimensional, using the horizontal and the vertical offset of the touch point at every time step. The signal values are normalised in $[0, 1]$ for each dimension and captured at the frame rate allowed by the software used $(> 30fps)$.

The Leap Motion device is a sensory device, allowing for hand and finger positions in space as input. Using the $JavaScript$ library provided by the manufacturer, and the same web-server setup with the touch-screen device we record six, 6, values to describe the hand posture at each frame. The

(a) Smart phone device used as a touch screen input for KURE.

(b) Tablet device used as a touch screen input for KURE.

Fig. 4. Two types of input devices used with KURE, for haptic 2-dimensional input signals. The difference is the screen size; 4.7 inches for the phone and 9.7 for the tablet.

values recorded represent the 3 rotational and 3 translational DoF of the palm of the hand.

Reversing the training paradigm, we employ machine learning techniques for the training of the interface to the user preferences. Supporting any given input device allows for this extra preference of the user to be taken into account. Indeed the idea can be highlighted even more with the use of a passive input device, like the Leap Motion, as it does not constrain the user in their manipulation patterns. The user does not have to interact with a physical material, rather their hand motions are being recorded, allowing for highly personalised patterns to be expressed. The continuity of the recording, on the other hand provides a challenging test bench for our system.

A. Stage 1 - Training Towards the Robot

As illustrated in figure (1), the interface is working in the shared boundary between the two systems present: the robot and the input device. On the robot side, the interface captures the behaviours of the robot as sensory motor sequences. In our experimental setup, as sensory inputs we understand the joint positions of the robot. Thus, we work with proprioceptive sensory input to create the kinematic model of the robot. The motor commands are passed as position commands on the joint motors of the robot, through a PID controller. Both the homeokinetic self-organisation and neural networks of the experts work and adapt real time on user demand.

Every time step (t) of the simulation, the homekinetic module produces motor commands, and a prediction of the the resulting sensory state of the robot. In the next time step $(t+1)$ of the simulation the actual sensors are recorded and the time loop error of the homeokinetic control is calculated training the Controller K and the World Model W (see section II-A). In parallel to this, in every time step (t) the 'expert' neural networks perform a forward pass, predicting the motor commands of time t and the sensory predictions of time $t+1$, of the homeokinetic module. Working in a winner takes all scheme, the network-expert with the best prediction adds its input and output to its dataset and a single step (1 epoch of training) of training is applied. This way each network specialises in a single behaviour, thus becomes an 'expert' of the behaviour.

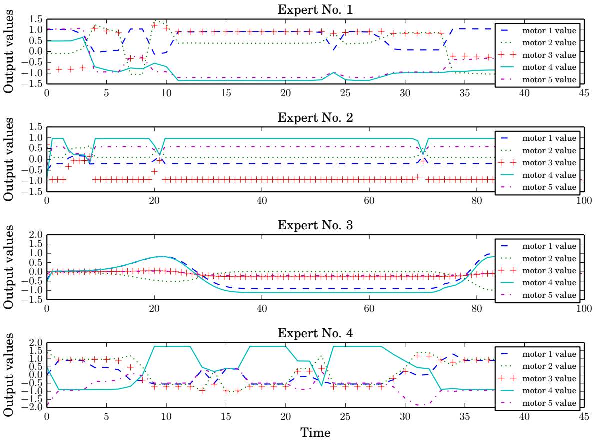


Fig. 5. Behaviours explored in the snake robot. In each sub-figure we can see the motor output of each expert respectively, for sensory input as described in section II-A.

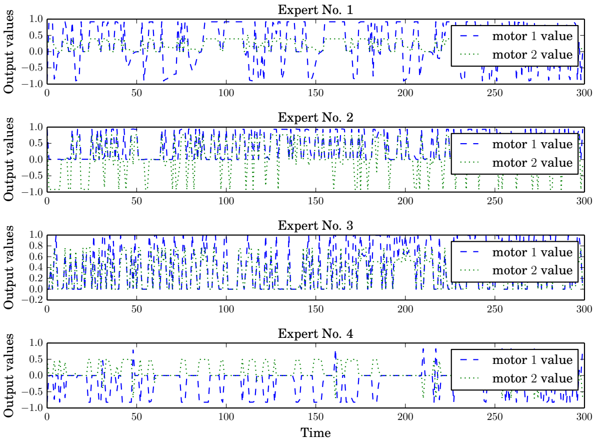
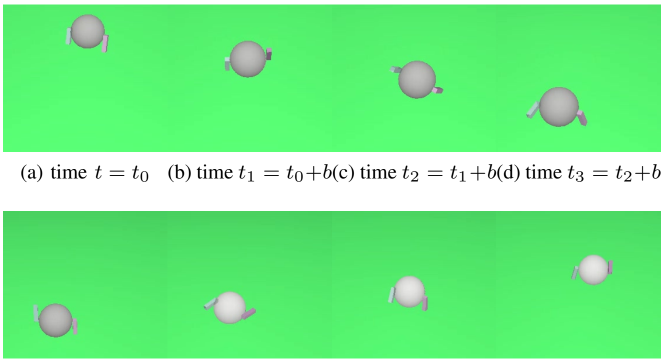


Fig. 6. Behaviours explored in the spherewalker robot. In each sub-figure we can see the motor output of each expert respectively, for sensory input as described in section II-A.

Indeed, each Neural Network ends up describing a different behaviour, being able to produce the corresponding motor commands based on the sensory input from the robot. In figures (5) and (6) we can see the behaviours described by the experts for the snake-like morphology and for the spherewalker morphology, respectively.

To generate both figures (5) and (6), we activated the experts (NNs) having as input their respective dataset, thus performing a validation of the behaviour they represent. In doing so we see the behaviours captured by each Neural Network and their differences. On each sub-figure we observe the behaviours as they expand in time. In more detail, the x-axis represents time, while the y-axis the networks outputs, i.e. the motor commands. For each behaviour we can see the different pattern followed in the motor domain. The robot for figure (5) has 5 motors and so the output of the network, while the robot of figure (6) has 2 motors. Each colour, and each line style,

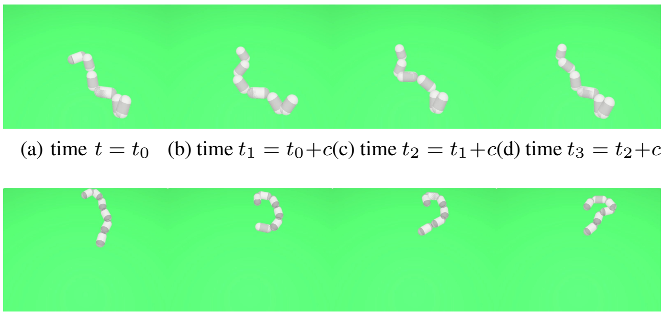


(a) time $t = t_0$ (b) time $t_1 = t_0 + b$(c) time $t_2 = t_1 + b$(d) time $t_3 = t_2 + b$

(e) time $t = t_0$ (f) time $t_1 = t_0 + b$(g) time $t_2 = t_1 + b$(h) time $t_3 = t_2 + b$

Fig. 7. Behaviours of the sphere-walker morphology. The time constant b is set to $0.2sec$.

represent a motor of the respective robotic morphology and show the values of the particular motor in time.



(a) time $t = t_0$ (b) time $t_1 = t_0 + c$(c) time $t_2 = t_1 + c$(d) time $t_3 = t_2 + c$

(e) time $t = t_0$ (f) time $t_1 = t_0 + c$(g) time $t_2 = t_1 + c$(h) time $t_3 = t_2 + c$

Fig. 8. Behaviours of the snake morphology. The constant c is set to be $0.5sec$.

In figures 8 and 7, behaviours of the snake and the sphere-walker morphologies can be observed. We can see how a moving downwards and a moving upwards behaviours have been found for the the sphere-walker morphology. The graph displays snapshots of the simulated environment while the robot is being controlled by a behavioural expert. The same is seen with the snake morphology in figure 8. Two of the found experts are shown, as an example, controlling the robot and producing their respective behaviour.

As we described and shown in [10], these behaviours can be intersected and also combined. Indeed, in our studies we have shown that the transition between them is smooth and so is the robot's resulting behaviour. Also, we have shown how these behaviours can be linearly combined to produce new, stable, behaviours.

B. Stage 2 - Training Towards the Input Device

On the other side, the interface, after having trained on the robotic morphology, has to train on the user input. To stimulate the user, each robotic behaviour is exhibited by the robot in the simulated environment. The user, while observing the robotic morphology, acts upon the input device in their way of preference. The system does not impose any restriction on the users behaviour, as long as the behaviour is captured by the device. Indeed in this stage, the exploration goes towards the user with them responding to the robot's actions. The system captures these time sequences and creates a dataset, having as an output the resulting 'expert'.

Since we want to perform a mapping, the output of the network is chosen to be coordinates in the N-dimensional expert-space. Working in the N-dimensional cube, each expert is found in each vertex of the N-dimensional unity hypercube. So, that expert-1, lies in $< 1_0, 0_1, 0_2, ..., 0_N >$.

For the time span that a behaviour is exhibited by the robot, the input device is recorded and a dataset is created. In this stage we use the Echo State Network (ESN) to capture the dynamics of the input signal. The network is trained, performing linear regression on the output weights of the network for the whole dataset. The complexity of the calculations required is small enough to allow for the training of the network within $5s$. This makes it possible for the network to be trained for each user, as the system is about to be used.

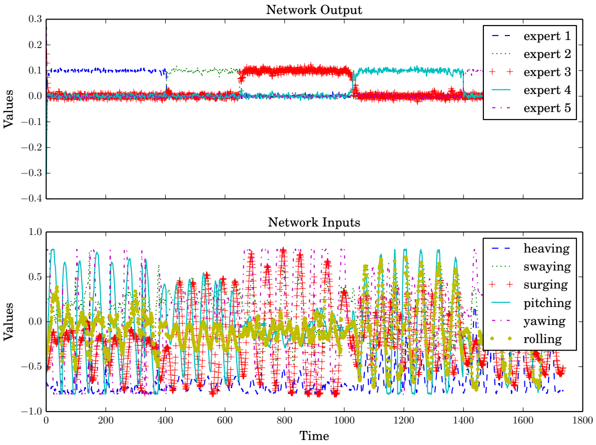


Fig. 9. Validation of a trained ESN. In the top sub-graph is the output of the network, with each colour and line style representing a behaviour recognised in the input. On the bottom sub-graph is the input to the network, again each input node is depicted with a different colour and line style. All values are plotted against time.

In figure (9) we can see the validation of the training of the ESN. The x-axis represents time and y-axis represents input and output values, in the bottom and top graph respectively. In the top graph we can see the output of the ESN. Each colour describes a different dimension of the expert space. The dimension with the higher value is the representing dimension of the expert recognised in the inputs dynamics. In the bottom graph, the input of the network is depicted, showing the manipulations of the input device as they happen in time. Time is aligned through all three graphs.

In the case of figure (9), the input comes from a Leap Motion device. Input is acquired at the frame rate of the device (> 30fps), from within the boundaries of $[-1, 1]$ for each DoF, resulting in a cube where the interaction is recorded.

The network is trained on the data provided and generates a perfect result for the training set (validation).

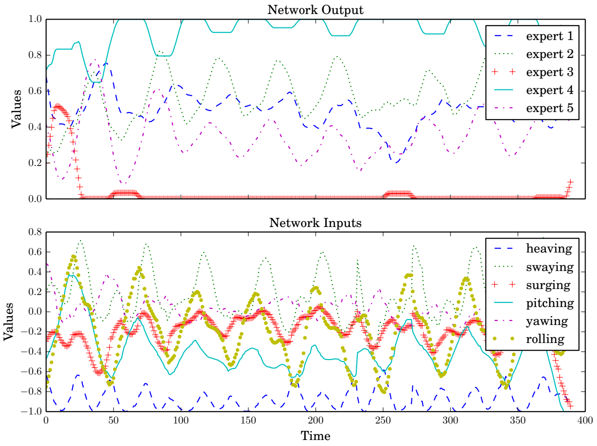


Fig. 10. Usage of a trained ESN. In the top sub-graph is the output of the network, with each colour and line style representing a behaviour recognised in the input. On the bottom sub-graph the input to the network is plotted, again each input node is depicted with a different colour and line style. All values are plotted against time.

For the trained network shown in (9), we now test the generalisation capabilities, having the user performing the gestures again in random order. This way we are able to test how well the network can cope the noise of real time usage of the device. Again, the Leap Motion is sampled at the frame rate of the device, with the input being recorded only within the boundaries of the cube mentioned above. In figure (10), we can see one of the behaviours: a cyclic motion of the hand in vertical space above the device. The x-axis represents time for all three graphs, while the y-axis the output, state, and input for the top, middle and bottom graphs. Again each colour in the output represents a dimension of the expert-space.

C. *Stage 3 - Usage of the Interface*

Having trained both sides of the interface, the system is now ready to be used. The operator, manipulating the input device, provides the input to the ESN. The DoF of the input device are recorded continuously over time, producing the input sequence to the ESN. Each time step recorded is fed to the ESN, exciting the internal dynamics of the network and producing an output, as seen in the graphs of the state of the ESN in figure (10). The recurrent and sparse connections in the Dynamic Reservoir of the network provide a rich high-dimensional representation of the input signal. This high-dimensional representation is then linearly combined to produce the output of the network.

The network output is then used to activate the experts accordingly. The output is represented as a vector in the expert-space. This vector, is then used to activate each of the experts, creating their linear combination. The combination of the experts is realised as a combination of their outputs, according to the linearity above. Each of the expert-networks, gets as input the sensory state of the robot at time (t) and produces a motor command and a sensory prediction. The motor command passed to the robot is the combination of the motor commands as guided by the ESN's output.

IV. Conclusions

We have shown how we are able to provide a mapping between the two different time scales present. The manipulations of the input device, happening according to the user preferences of the input device, and the robot behaviours, guided by homeokinesis. Each point of the resulting expert-space, represents two time sequences, able to unfold differently in time. Towards the robot, and through the expert-network, the point is mapped to a robotic behaviour. Towards the human, each point is mapped to a time depended manipulation of the input device. This way both systems are mapped in a shared space, providing a robust and consistent way for control.

Regarding future direction of the work presented, there is still a need for the unified mathematical formulation of the systems operation, now that the systems operation is confirmed experimentally. Also, the limitations are to the explored with the use of ever more distant (measured in DOF) input devices and robots.

References

[1] J. Leitner, M. Luciw, and A. Förster, "Teleoperation of a 7 DOF Humanoid Robot Arm Using Human Arm Accelerations and EMG Signals," *juxi.net*.

[2] D. Shah, J. Schneider, and M. Campbell, "A Sketch Interface for Robust and Natural Robot Control," *Proceedings of the IEEE*, vol. 100, no. 3, pp. 604–622, mar 2012.

[3] G. Doisy, J. Meyer, and Y. Edan, "A Gesture Learning Interface for Simulated Robot Path ShapingWith a Human Teacher," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 6, pp. 788–795, dec 2014.

[4] N. Y. Lii, Z. Chen, M. a. Roa, A. Maier, B. Pleintinger, and C. Borst, "Toward a task space framework for gesture commanded telemanipulation," *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pp. 925–932, sep 2012.

[5] M. W. Kadous, R. K.-M. Sheh, and C. Sammut, "Effective user interface design for rescue robotics," in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*. ACM, 2006, pp. 250–257.

[6] T. Ogata and S. Sugano, "Imitation based Human-Robot Interaction - Roles of Joint Attention and Motion Prediction-," pp. 283–288, 2004.

[7] R. Der, G. Martius, and F. Hesse, "Let it rollemerging sensorimotor coordination in a spherical robot," *ARTIFICIAL LIFE X: PROC. X INT. CONF. ON . . .*, 2006.

[8] R. C. Arkin, *Behavior-based robotics*. MIT press, 1998.

[9] R. Der, "Selforganized robot behavior from the principle of homeokinesis," *Proc. workhop soave*, pp. 39–46, 2000.

[10] C. Melidis and D. Marocco, "An exploration on intuitive interfaces for robot control based on self organisation," in *Artificial Life and Intelligent Agents*, ser. Communications in Computer and Information Science, C. J. Headleand, W. J. Teahan, and L. Ap Cenydd, Eds. Springer International Publishing, 2015, vol. 519, pp. 73–79.

[11] G. Martius, K. Fiedler, and J. M. Herrmann, "Structure from behavior in autonomous agents," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 858–862.

[12] G. Martius, "Taming the Beast: Guided Self-Organization of Behavior in Autonomous Rboots."

[13] B. Widrow, M. E. Hoff *et al.*, "Adaptive switching circuits." 1960.

[14] H. Jaeger, *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach*. GMD-Forschungszentrum Informationstechnik, 2002.

KURE: a Two-Way Adaptive System for Intuitive Robot Control

Christos Melidis[1] and Davide Marocco[2]

Abstract— **In this paper we present a novel approach to human-robot control. Instead of the user adapting to the interface and control paradigm, the system proposed allows the user to shape the control motifs in their way of preference, moving away from the case where the user has to read and understand an operation manual. Starting from a tabula rasa basis, the system is able to identify control patterns (behaviours) for the given robotic morphology and successfully merge them with control signals from the user, regardless of the input device used. The structural components of the interface are presented and assessed both individually and as a whole.**

I. INTRODUCTION

In this paper we present a continuous, on-line, real-time methodology for the remote control of mobile robots. The methodology presented, being agnostic to the specific systems at hand, targets any robot and any input device. Starting from a tabula rasa basis an open ended interface between the two systems is initialised and autonomously adapted to fit the robot and the input device at hand. The system is adapted through the interaction of the user with the input device, and the robot with its environment.

Remote control of robots is usually seen as a classification problem, with the user acting on an input device, the system identifying the user's behaviour, and triggering the appropriate response on the robot. Under such a paradigm two variables need be fixed beforehand, that is the input device and the robotic morphology to be controlled.

Indeed, most research is performed targeting a specific robotic morphology [1], [2]. This allows for tailored solutions on the behaviour generation for the robot, solving inverse kinematic models or having hard-coded routines of interaction. Our approach working on a model free basis creates and adapts the robot's controller under a self-organising paradigm. Being agnostic towards the controlled robot highlights the necessity for a mechanism to explore the robots capabilities, with respect to its environment. It is through the interaction with the environment that the embodiment's properties can be revealed.

At the same time, the plethora of studies on remote control are formed tightly around the input device to be used [3], [4] and are commonly constructed by field experts, to be used by field experts. Complicated input devices and non-intuitive control patterns are created which the user has to learn in order to use the system. Our research aims at an intuitive control paradigm, where the user's intentions for control are formed and used for the interaction. Adaptive methodologies have only started appearing, most of them working under a classification paradigm [5], [6]. Although classification can provide a robust tool for input recognition, our approach provides a robust way of mapping inputs to a lower or higher dimensional space, allowing for the geometric properties of the input to be explored i.e. opposing behaviours having opposing mappings and the ability to mix behaviours.

Most robotic remote control systems rely on the intelligence and cognitive capabilities of the operator to understand the control paradigm and the robot's capabilities. The operator once familiar with the input device, its functioning and its potentials, has to understand the control paradigm and how the control flows are defined to be used. Indeed, in most cases the operator needs to acquire the knowledge required for control through a training procedure (i.e. reading the manual for operation, practice on the controls)[7].

The problem we try to solve, is a dual problem; mapping the human operators control signals to the actions of the robotic morphologies. In doing so, we try to merge the two apparent dynamical systems involved. The one formed by the operators input signals and the other by the robots behaviours.

From the operators perspective; being able to observe the robots embodiment interacting with the environment, allows for a better understanding of it. In this process, intentions for control can be formed. Being able to capture those intentions and associate them with the robots behaviour, can result in a control flow - an intuitive control paradigm - tailored to each user and robot. This, with the operator being free to act upon the system - that acts as a mediating agent between user and robot - namely the interface.

II. METHOD

The methods section is divided in two parts: covering the formation of the controllers for the robotic morphology, and the input acquisition from the user. Regardless of the robot to be controlled our intention is to extract primitive behaviours, capable of being combined, providing a rich enough scaffolding for the control of the robot. We avoid human intervention in the formation of the robotic behaviours in order to have an autonomous system. Indeed, under this scope the behavioural primitives should be formed through the interactions of the robot with the environment, allowing for a learning mechanism grounded to the robot.

There are numerous ways of forming controllers capable of achieving pre-specified behaviours, supervised learning,

[1] C. Melidis is with the School of Computing, Electronics and Mathematics, member of the Cognition Institue, Plymouth University, Plymouth, Devon, UK, christos.melidis@plymouth.ac.uk

[2] D. Marocco is with the Dipartimento di Studi Umanistici, University of Naples "Federico II", via Porta di Massa 1, Naples, Italy and with the School of Computing, Electronics and Mathematics, Plymouth University, Plymouth, Devon, UK, davide.marocco@plymouth.ac.uk

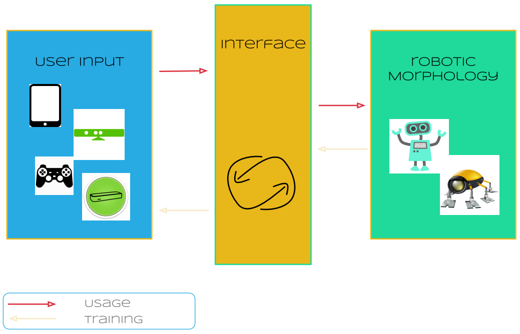


Fig. 1. Schematic representation of KURE

homoeostatic regulation, central pattern generators (CPG), evolutionary computation methods (EC), to name a few. All the above, although intrinsically different share the idea of external guidance. A teacher behaviour needs be formed for the robot to imitate in the case of supervised methods, external perturbations for a homoeostatic adaptation, the tuning for CPG, and a scalar measure for a target function in EC.

On the other hand, regardless of the input device used, our goal is to capture the intentions for control from the user as expressed through the input device. In this end we treat the input as a time sequence of manipulations of the input device. Our intention is to allow the user to freely interact with the input device forming their own personal control patterns. Segmenting the input sequence or using sliding windows techniques to imitate continuity on the input sequence is not a solution. Indeed, since our aim is to allow the user to form their interaction patterns, we cannot make any assumptions of the length of the time sequences (and thus on the size of the sliding window). At the same time, working under a mapping and not a classification paradigm we cannot use statistical methods. Finally, the system must not take long to initialise and adapt to the user preferences, as that would degrade the user experience. In our method the training time required for the input recognition subsystem is less that 10 seconds.

One solution could be the usage of Dynamic Time Warping (DTW), but in such a case the input should be segmented, violating our need for continuity. Hidden Markov Models could be used, but this would fall under a classification paradigm. Recurrent Neural Networks, pose a promising solution for our desired mapping, but the training techniques used (Back-Propagation through time) require a lot of time to train.

The system should work as a mediator between the robot and the operator; an interface connecting the two systems as seen in figure 1. In doing so, it should be capable of exploring the robots potential, the users behaviours and connect them seamlessly, placing the human in the loop.

A. Self-organisation of Robotic Behaviours

For the formation of the control sub-system for the robot, a self-referential dynamical system is derived and a principle for self-organisation of robotic behaviours [8], [9]. The idea here is to try and maintain a smooth control behaviour keeping the robot at a constant kinetic state. This property of the system (i.e. self excitation) was first formulated by Ralph Der and referred to as homeokinesis [10].

The learning in this procedure occurs based on the time-loop error; the error between the real behaviour and the model's prediction. Based on the homeokinetic principle the self-organisation of the sensory motor loop of the robotic morphology is possible without an external driving force (i.e. teacher signal or external perturbation). From this, a repertoire of behaviours emerges, which we are able to capture in the form of behavioural experts. These experts can later be reused and combined to control the robot. The behaviours, as demonstrated in the following section III-A, vary in complexity, time, and are entirely based on the robot and its interactions within the environment.

For the exploration of the robot's capabilities we work as seen in [11]. We want to be able to produce motor outputs from sensory readings and from them predict the next sensory state of the robot. Creating a sensory-motor, and a motor-sensory mapping, allows us to derive an error signal for the update of the systems parameters.

For the two systems described above, the Controller K and the World Model W, their functions for operation are stated bellow. The exploration module is described, according to time t and depending on a set of parameters defined by the matrix A, as:

$$\tilde{\mathbf{x}}_{t+1} = W(K(\mathbf{x}_t, C), A) \tag{1}$$

The controller K generates motor outputs $\mathbf{y}_t = K(\mathbf{x}_t, C)$ as a function of the sensory input $\mathbf{x} = x_1, x_2, \ldots, x_n$, depending on a set of parameters defined by the matrix $C_{n,n+1}$ and it is defined by the equation:

$$K = g(\sum_{i=1}^{n} C_i x_i + C_{n+1}), \tag{2}$$

where g is a sigmoid function.

The parameter matrix of the world model, A, is adapted according to the delta rule, $\Delta w = +\eta E_W \mathbf{x}$ with the error, E_W, described by the function: $E_W = ||\mathbf{x}_{t+1} - \tilde{\mathbf{x}}_{t+1}||^2$ with learning rate η = 0.1.

The controller updates its parameter matrix by gradient descent with respect to the error function, $E_K = ||\mathbf{x}_t - \tilde{\mathbf{x}}_t||^2$ with $\tilde{\mathbf{x}}_t$ being the inverse solution of the Controller K for the motor commands of time t and the observed sensory state at $t + 1$.

For the identification, storage and reuse of the different behaviours exhibited by the robot, we use a series of m neural networks (NNs), called *experts*. Each NN is defined according to the equation,

$$(\mathbf{x}_{t+1}, \mathbf{y}_t) = N_i(\mathbf{x}_t, \mathbf{x}_{t-1}), \quad i = 1, \ldots, m \tag{3}$$

The NNs consist of 3 layers, feed-forward units where the hidden layer consists of sigmoid units, whereas the input and output layers of linear units. Back-propagation is used to training the NN with learning rate $\eta = 0.01$. The NNs, working in parallel, compete for the prediction of the motor command $\mathbf{y}_t$ of time t and the sensory input $\mathbf{x}_{t+1}$ of the next time step in a winner-takes-all method, with only the winning network being allowed to train on the current data $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$. Because of this competition, each of the expert NN specialises to represent a different behaviour of the robot.

In order to use these behaviours, the sensor values from the robot are passed in the selected NN and the respective motor commands (output of the NN) are applied to the robot.

B. *User Input as a Continuous Signal*

Following the constrains mentioned in the beginning of the section, the system makes use of Echo State Networks. Combining the effect of multiple time scales and the possibility of mapping the time sequence dynamics to a fixed dimensional space, [12] formulated the echo state approach on training Recurrent Neural Networks, namely Echo State Networks (ESN). One of the most appealing features for our application, is the fact that the network is trained using linear regression on its output layer only, reducing the complexity of training with BPTT. The input signal propagated to the Dynamic Reservoir, expands in dimensions allowing for easier manipulation of the signal. In this setup the only trainable weights are output layer's, reducing the complexity of training to a matrix multiplication.

Echo State Networks (ESN) provide an architecture for efficient training of RNN in a supervised manner. One can distinguish two main components in an ESN. Firstly, the Dynamic Reservoir (DR), a large, randomly wired, recurrent neural network with fixed weights. The DR gets activated by the input and provides a non linear response for this input. And the output signal, which is trained as a linear combination of the activations of the DR. This way the computational resources and complexity required for the training RNNs is reduced to the adaptation of the output connections of the ESN.

The state update equation, for an ESN consisting of N reservoir units, K inputs and L outputs - without any recurrent output neurons - is,

$$\mathbf{x}(n+1) = f(\mathbf{W}\mathbf{x}(n) + \mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}^{fb}\mathbf{y}(n)) \ , \quad (4)$$

where $\mathbf{x}(n)$ is the N-dimensional reservoir state, f is a sigmoid function (usually the logistic sigmoid or tanh function), $\mathbf{W}$ is the $N \times N$ reservoir weight matrix, $\mathbf{W}^{in}$ is the $N \times K$ input weight matrix, $\mathbf{u}(n)$ is the Kdimensional input signal, $\mathbf{W}^{fb}$ is the $N \times L$ output feedback matrix, and $\mathbf{y}(n)$ is the L-dimensional output signal. In tasks where no output feedback is required, $\mathbf{W}^{fb}$ is set to zero.

The extended system state $\mathbf{z}(n) = [\mathbf{x}(n); \mathbf{u}(n)]$ at time n is the concatenation of the reservoir and input states (and output in the case of output recurrency).

The output can be obtained from the extended system state by,

$$\mathbf{y}(n) = g(\mathbf{W}^{out}\mathbf{z}(n)) \ , \quad (5)$$

where g is an output activation function (typically the identity or a sigmoid) and $\mathbf{W}^{out}$ is a $L \times (K + N)$-dimensional matrix of output weights.

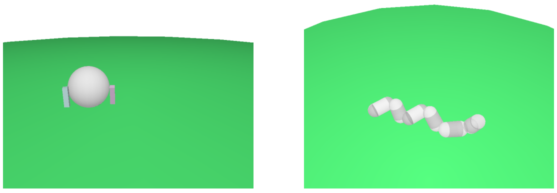
For the training of ESNs, let us assume a driving signal $u(1), \dots, u(n_{max})$ and the extended states it generates -once passed to the network- $z(1), \dots, z(n_{max})$. We collect the states in matrix S of size $n_{max} \times (N + K)$ and the desired outputs $d(n)$ in a matrix D of size $n_{max} \times L$. Usually, before each collection, based on the properties of the network, we apply a washout period, allowing the network to settle to the input provided.

Now, the desired output weights W^{out} can be calculated as follows. First, the correlation matrix of the extended system states are calculated, $R = S'S$. Then the cross-correlation matrix of the extended states against the desired outputs d, $P = S'D$. Finally, for the calculation we use the Moore - Penrose pseudoinverse of S, $S^{\dagger}$ and then updating the weights

$$W^{out} = (S^{\dagger}D)' \quad (6)$$

III. Experimental Results

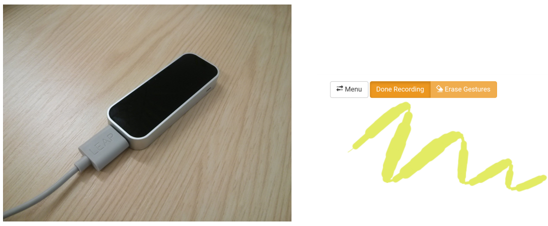
Having elaborated on the methods to be used for the behaviour extraction from the robotic morphology and for the sequence recognition from the user, we now proceed with the description of our experimental setup with the results obtained by each individual component and the system as a whole.



(a) The spherewalker morphology.

(b) The snake morphology.

Fig. 2. The two morphologies used in the experiments in the simulated environment.



(a) The Leap motion device

(b) The graphical interface for the touch-screen device. The yellow line shows the gesture being recorded. Disappears when a finger is not touching the screen.

Fig. 3. The two interfaces used in the experiments.

For our test scenarios we used two different - simulated- robotic morphologies (Fig. 2b, 2a) and two different input devices Fig. 3. The robotic morphologies were simulated using Open Dynamics Engine (ODE), through Python. In doing so, we were able to simulate the physical properties of the environment and so obtain a good representation of the dynamics of the morphology.

The "spherewalker" morphology has two motors each with 1 Degree of Freedom (DoF), and two sensors, measuring the joint positions. The "snake" morphology has five motors each with 1 Degree of Freedom (DoF), and five sensors, measuring the position of each joint. The values recorded from the sensors are normalised to fit $[-1, 1]$ in both morphologies.

(a) Smart phone device used as a touch screen input for KURE.

(b) Tablet device used as a touch screen input for KURE.

Fig. 4. Two types of input devices used with KURE, for haptic 2-dimensional input signals. The difference is the screen size; 4.7 inches for the phone and 9.7 for the tablet.

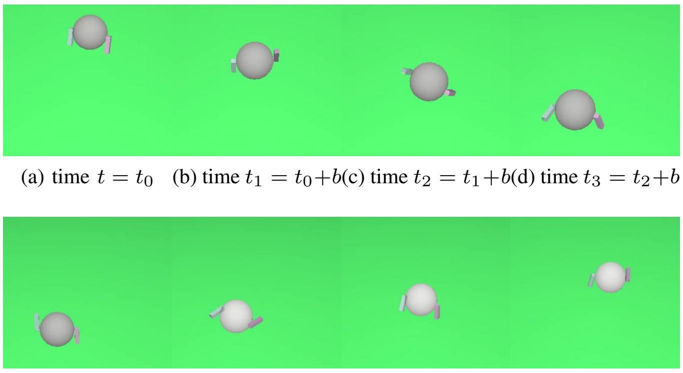
For the input devices, we capture a two dimensional signal from the touch-screen device, and a six dimensional signal from the Leap Motion device. In the case of the touch-screen device the input signal is 2-dimensional, using the horizontal and the vertical offset of the touch point at every time step. The signal values are normalised in $[0, 1]$ for each dimension and captured at the frame rate allowed by the software used $(> 30fps)$.

The Leap Motion device is a sensory device allowing for hand and finger positions in space, as input. Using the $JavaScript$ library provided by the manufacturer, and the same web-server setup with the touch-screen device we record six, 6, values to describe the hand posture at each frame. The values recorded represent the 3 rotational and 3 translational DoF of the palm of the hand.

A. Stage 1 - Training Towards the Robot

As illustrated in figure (1), the interface is working in the shared boundary between the two systems present: the robot and the input device. On the robot side, the interface captures the behaviours of the robot as sensory motor sequences. In our experimental setup, as sensory inputs we understand the joint positions of the robot. Thus, we work with proprioceptive sensory input to create the kinematic model of the robot. The motors of the robot can be controlled both via a PID controller and Torque from the same architecture with the same parameters, as through homeokinesis the controller network K adapts to perturbate the sensorymotor loop of the robot at hand.

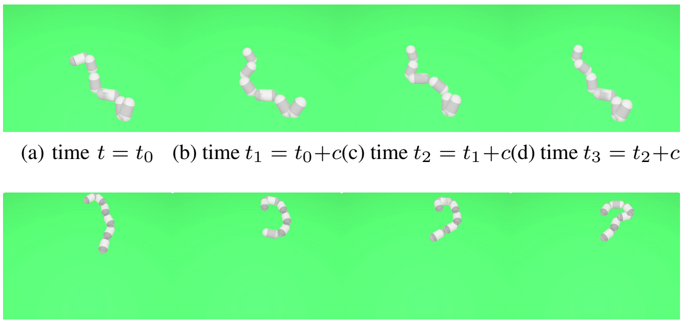
Every time step (t) of the simulation, the module following a homekinetic adaptation produces motor commands, and a prediction of the resulting sensory state of the robot. In the next time step $(t + 1)$ of the simulation the actual sensors are recorded and the time loop error of the homeokinetic control is calculated training the Controller K and the World Model W (see section II-A). In parallel to this, in every time step (t) the 'expert' neural networks perform a forward pass, predicting the motor commands of time t and the sensory predictions of time $t + 1$, of the homeokinetic module. Working in a winner takes all scheme, the network-expert with the best prediction adds its input and output to its dataset and a single step (1 epoch of training) of training is applied. This way each network specialises in a single behaviour, thus becoming an 'expert' of the behaviour.



(a) time $t = t_0$ (b) time $t_1 = t_0+b$(c) time $t_2 = t_1+b$(d) time $t_3 = t_2+b$

(e) time $t = t_0$ (f) time $t_1 = t_0+b$(g) time $t_2 = t_1+b$(h) time $t_3 = t_2+b$

Fig. 5. Behaviours of the sphere-walker morphology. The time constant b is set to $0.2sec$.



(a) time $t = t_0$ (b) time $t_1 = t_0+c$(c) time $t_2 = t_1+c$(d) time $t_3 = t_2+c$

(e) time $t = t_0$ (f) time $t_1 = t_0+c$(g) time $t_2 = t_1+c$(h) time $t_3 = t_2+c$

Fig. 6. Behaviours of the snake morphology. The constant c is set to be $0.5sec$.

In figures (6) and (5), behaviours of the snake and the sphere-walker morphologies can be observed. We can see how a moving downwards and a moving upwards behaviours have been found for the the sphere-walker morphology. The graph displays snapshots of the simulated environment while the robot is being controlled by a behavioural expert. The same is seen with the snake morphology in figure (6). Two of the found experts are shown, as an example, controlling the robot and producing their respective behaviour.

As we described, and shown in [9], these behaviours can be intersected and also combined. Indeed, in our studies we have shown that the transition between them is smooth and so is the robot's resulting behaviour. In addition, we have shown how these behaviours can be linearly combined to produce new, stable, behaviours.

B. Stage 2 - Training Towards the Input Device

On the other side, the interface, after having trained on the robotic morphology, has to train on the user input. The system at this stage is able to stimulate different robotic behaviours. To capture the operators intentions for control, we reverse the information flow. In this stage the robot exhibits the behaviours extracted through the homeokinetic controller with the user responding to them. In order to form intentions for control the user observes the robot acting and responds with manipulations of the input device. At this stage the interaction between the human and the robot is recorded and the training set stored.

This set will be used to form the mapping from the K-dimensional space of the input device to the N-dimensional space of the experts. Since we want to perform a mapping, the output of the network is chosen to be coordinates in the N-dimensional expert-space. Working in the N-dimensional cube, each expert is found in each vertex of the N-dimensional unity hypercube. So, that expert-1, lies in $< 1_0, 0_1, 0_2, ..., 0_N >$, e.t.c.

The network is trained, performing linear regression on the output weights of the network for the whole dataset. The complexity of the calculations required is small enough to allow for the training of the network within $5s$. This makes it possible for the network to be trained for each user, as the system is about to be used.

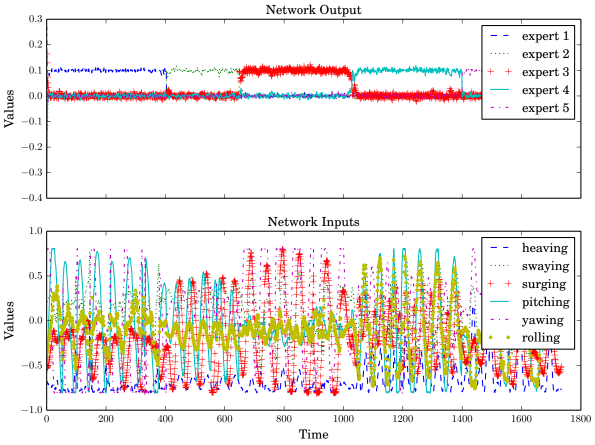


Fig. 7. Validation of a trained ESN. In the top sub-graph is the output of the network, with each colour and line style representing a behaviour recognised in the input. On the bottom sub-graph is the input to the network, again each input node is depicted with a different colour and line style. All values are plotted against time.

In figure (7) we can see the validation of the training of the ESN. The x-axis represents time and y-axis represents input and output values, in the bottom and top graph respectively. In the top graph we can see the output of the ESN. Each colour describes a different dimension of the expert space. The dimension with the higher value is the representing dimension of the expert recognised in the inputs dynamics. In the bottom graph, the input of the network is depicted, showing the manipulations of the input device as they happen in time. Time is aligned through all three graphs.

In the case of figure (7), the input comes from a Leap Motion device. Input is acquired at the frame rate of the device ($>$ 30fps), from within the boundaries of $[-1, 1]$ for each DoF resulting in a cube where the interaction is recorded. The network is trained on the data provided and generates a perfect result for the training set (validation).

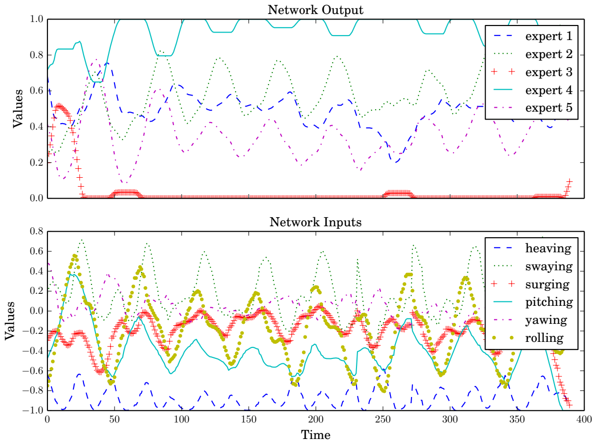


Fig. 8. Usage of a trained ESN. In the top sub-graph the output of the network is depicted, with each colour and line style representing a pattern recognised in the input. In the bottom sub-graph the input as recorded from the input device is plotted. All values are plotted against time and are aligned as recorded.

For the trained network shown in (7) in order to test the capabilities of the network we have the user perform the gestures again in random order. This way we are able to test how well the network can cope the noise of real time usage of the device. Again, the Leap Motion is fed inputs and sampled at the frame rate of the device, with the input being recorded only within the boundaries of the cube mentioned above.

As long as the user is manipulating the input device, the ESN is activated with the recorded input. The ESN running real time, receiving input values at the frame rate of the input device, maps the input to the expert space. The network is able to recognise the input patterns of the user correctly.

In figure (8), we can see one of the behaviours; a cyclic motion of the hand in vertical space above the device. The x-axis represents time, while the y-axis the output, and input for the top and bottom graphs. Again each colour in the output represents a dimension of the expert-space.

C. Stage 3 - Usage of the Interface

Having trained both sides of the interface, the system is now ready to be used. The operator, manipulating the input device, provides the input to the ESN. The DoF of the input device are recorded continuously over time, producing the input sequence to the ESN. Each time step recorded is fed to the ESN, exciting the internal dynamics of the network and producing an output.

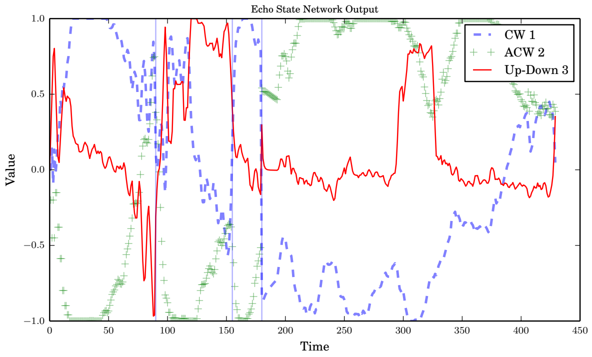


Fig. 9. Usage of a trained ESN. In the graph the output of the network is depicted, with each colour and line style representing a pattern recognised in the input. All values are plotted against time

In figure (9), we can observe a close up of the recognised patterns from the ESN. CW notes a clockwise circle pattern on movement by the user, ACW an anticlockwise, and $Up - Down$, up down movement pattern of the hand. If we observe the first segment of the graph, as separated by the first vertical line, we see that the network correctly recognises a CW motion as input. What is more important is that the ACW motion is having a negative value, as the input pattern observed is "opposite" to it. In the next section we see the transition of the output to the $Up-Down$ motion. In this we observe that the network can mix the two in the output, while the ACW still remains negative, as the input is still opposing that pattern. Moving to the forth segment, the user is now performing an ACW input pattern and the ESN correctly recognises it. At the same time we observe that the CW pattern is negative as it is opposite to the one observe. The $Up - Down$ recognition settles at 0 level again, until the user starts mixing the input patterns again, as observed at the mid-point of the segment.

Taking into account that the network has been trained by the responses of the user to robotic behaviours, we deem this an important feature of the system. As an example, lets assume the CW motion is mapped to the robot moving forward and the ACW backwards. Having opposite behaviours mapped being understood as opposite, provides the network with an "insight"; the user cannot be performing two opposing behaviours at the same time, but they can be performing the $Up - Down$ in combination with any of the above. At the same time, going back the expert networks we see that the combination of their outputs is done in a linear fashion. The contribution for each expert, in the final motor values of the robot, is calculated from the output of the ESN. This way, we observe that the robotic behaviour responding to the opposite motion, from the one observed in the input, not only is it suppressed, but also reversed, contributing to the correct behaviour being exhibited by the robot faster.

IV. CONCLUSIONS

The interface is able to place the human operator in the loop of the robotic behaviours. In doing so, we establish a human centric control paradigm of robot control. Instead of having a learning procedure to train the operator on the usage of the interface, we adapt the system. User preferences, either as manipulations of the input device, or the input device itself (here Leap Motion and Touchscreen), fully shape the control experience.

We are able to provide a mapping between the two different time scales present; the manipulations of the input device happening according to the user preferences of the input device, and the robot behaviours guided by homeokinesis. Each point of the resulting expert-space represents two time sequences, able to unfold differently in time. Towards the robot, and through the expert-networks, each point is mapped to a robotic behaviour. Towards the human, each point is mapped to a time depended manipulation of the input device. This way both systems are mapped in a shared space, providing a robust and consistent way for control.

REFERENCES

[1] J. Leitner, M. Luciw, A. Förster, and J. Schmidhuber, "Teleoperation of a 7 dof humanoid robot arm using human arm accelerations and emg signals," 2014.

[2] D. Shah, J. Schneider, and M. Campbell, "A Sketch Interface for Robust and Natural Robot Control," *Proceedings of the IEEE*, vol. 100, no. 3, pp. 604–622, mar 2012.

[3] G. Doisy, J. Meyer, and Y. Edan, "A Gesture Learning Interface for Simulated Robot Path ShapingWith a Human Teacher," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 6, pp. 788–795, dec 2014.

[4] N. Y. Lii, Z. Chen, M. a. Roa, A. Maier, B. Pleintinger, and C. Borst, "Toward a task space framework for gesture commanded telemanipulation," *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pp. 925–932, sep 2012.

[5] Y. Yin, "Real-time Continuous Gesture Recognition for Natural Multimodal Interaction," *= Bulletin of Center for Educational Research and Development*, no. 21, pp. 1–8, 2014.

[6] S. Bodiroža, H. I. Stern, and Y. Edan, "Dynamic gesture vocabulary design for intuitive human-robot dialog," *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction - HRI '12*, p. 111, 2012.

[7] M. W. Kadous, R. K.-M. Sheh, and C. Sammut, "Effective user interface design for rescue robotics," in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*. ACM, 2006, pp. 250–257.

[8] R. Der, "Selforganized robot behavior from the principle of homeokinesis," *Proc. workhop soave*, pp. 39–46, 2000.

[9] C. Melidis and D. Marocco, "An exploration on intuitive interfaces for robot control based on self organisation," in *Artificial Life and Intelligent Agents*, ser. Communications in Computer and Information Science, C. J. Headleand, W. J. Teahan, and L. Ap Cenydd, Eds. Springer International Publishing, 2015, vol. 519, pp. 73–79.

[10] R. Der, G. Martius, and F. Hesse, "Let it rollemerging sensorimotor coordination in a spherical robot," *ARTIFICIAL LIFE X: PROC. X INT. CONF. ON . . .*, 2006.

[11] G. Martius, K. Fiedler, and J. M. Herrmann, "Structure from behavior in autonomous agents," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 858–862.

[12] H. Jaeger, *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach*. GMD-Forschungszentrum Informationstechnik, 2002.

Noname manuscript No.
(will be inserted by the editor)

Intuitive control of mobile robots: An architecture for autonomous adaptive dynamic behaviour integration.

Christos Melidis · Hiroyuki Iizuka · Davide Marocco

Received: date / Accepted: date

Abstract In this paper we present a novel approach to human-robot control. Taking inspiration from Behaviour Based robotics and self-organisation principles, we present an interfacing mechanism, with the ability to adapt both towards the user and the robotic morphology. The aim is for a transparent mechanism connecting user and robot, allowing for a seamless integration of control signals and robot behaviours. Instead of the user adapting to the interface and control paradigm, the proposed architecture allows the user to shape the control motifs in their way of preference, moving away from the case where the user has to read and understand an operation manual, or it has to learn to operate a specific device. Starting from a tabula rasa basis, the architecture is able to identify control patterns (behaviours) for the given robotic morphology and successfully merge them with control signals from the user, regardless of the input device used. The structural components of the interface are presented and assessed both individually and as a whole. Inherent properties of the architecture are presented and explained. At the same time, emergent properties are presented and investigated. As a whole, this paradigm of control is found to highlight the potential for a change in the paradigm of robotic control, and a new level in the taxonomy of human in the loop systems.

Keywords Human-Robot Interaction · Recurrent Neural Networks · Remote Control · Distributed representations

1 Introduction

In this paper a human-centred, adaptive system for the remote control of autonomous robots is presented. Our vision is for a system that is able to adapt both to the user and the robot, enabling a personalised communication path between the two. The user forms intentions for control and through the manipulations of the input device, communicates them to the robot. The procedure of command learning and recognition is implemented as a mechanism that interfaces the robot and the input device in such a way that, whether the input signals for activating the motor control are captured by an external hardware or acquired by the internal instruments of the robot (i.e. cameras), the system can actively recognise these input sequences and shape the robot's behaviour accordingly.

Most cases of remote robot control are tailored around specific robotic platforms or morphologies [32,56]. In addition, most studies of remote control are tightly conceptualised around the input device to be used [17,33]. Changing those assumptions requires a system that can handle multiple robotic morphologies as well as multiple input devices.

Here, a novel framework is presented for the autonomous dynamics behaviour integration between mobile robots and humans. Based on recurrent neural architectures the presented framework is able to generate,

C. Melidis
School of Computing, Electronics and Mathematics, member of the Cognition Institue, Plymouth University, Plymouth, Devon, UK, PL48AA, E-mail: christos.melidis@plymouth.ac.uk

H. Iizuka,
School of Information Science and Technology, Hokkaido University, 5 Chome Kita 8 Jonishi, Kita Ward, Sapporo, Hokkaido Prefecture 060-0808, Japan

D. Marocco
Dipartimento di Studi Umanistici, University of Naples 'Federico II', via Porta di Massa 1, Naples, Italy, 80138
E-mail: davide.marocco@unina.it

differentiate and extract dynamic behaviours from any mobile robot. At the same time, a novel paradigm of control is presented together with a novel adaptation technique for the user. Instead of the norm in control systems, the paradigm is shifted from classification to mapping, and thus robot and user dynamics are coupled to form the control patterns. Moreover, differently to most, if not all, available remote control systems, the robot is able to understand the users intentions for control through the interaction of the two dynamics, thanks to the available sensors. In practice, the framework is able to (a) stimulate the users intention for control by offering a set of pre-formed robot's behaviours, (b) capture this intention, and (c) store it in a efficient way, not only allowing reusability but also intuitive combinatorial between behaviours, as well as generative capabilities for adapting and creating new robot and user behaviours. To this extent, the novelty of the presented work sits within the context of the situated and embodied cognition paradigms, as well as within the behaviour based robotics approach to implement the robot control, indeed, both users and robots behaviours are strictly connected to the characteristics of the environment, of the robot morphology and of the input device, which are in turn entangled with the user's motor capabilities. Therefore, the novelty of the presented work lies in the unification of action, perception and intention in a rigorous analytical way, using time dependent methods, effectively providing a dynamical integration of all three.

In what follows we describe the method for autonomous acquisition of behaviours, interpreted in a modular fashion as in the case of behavioural-based robotics, formed through the dynamic interactions of the robot with its physical surroundings; and the method to perform the mapping of these behaviours to the relative input signals exhibited by the user. The former method is based on a dynamical system approach and a principle of self-excitation, namely homeokinesis. The latter method, based on Echo State Neural Networks, is capable of adapting to the dynamics of the input sequences and provides a robust mapping from the input space to the behavioural space of the robot. The methods used and the performance of the system are discussed and investigated in detail. The overall characteristics of the proposed framework are presented in detail in the 'Results' section.

2 Analysis of Existing Literature

2.1 Human Centric Systems

Our research is inspired by the fields of *human-machine* and *human-robot interaction*, as well as *self-organisation* concepts, with respect to embodied cognition. All those fields are brought into focus under the lights of cybernetic principles, where the system to be controlled, i.e. the robot, and the input system for the user, are both interpreted as complex systems dynamically interacting and coupling their behaviours. According to [73], control systems in general, fall into the category of 'Type_001 Cybernetics'. This type of cybernetics studies the cases were a self-governed system is governed from within by a single-self subject. In most systems of this type, there can be found two types of information flow. A cognitive flow, i.e. the quantitative information available by the system through its sensors, and a subjective flow, i.e. the experiential factors processed on the 'mind' of the system itself.

Human-Machine Interaction In Human Machine Interaction (HMI) interfacing mechanisms between the operator and the device to be controlled are tightly formed around the application field and the machine. To do so, the communication is mediated by an interface between the two systems. The design of the interfaces and the interaction enabled by them are mostly studied in the field of ergonomics [58]. In terms of HMI, ergonomics relates to how the user will interact with a machine and how easy that interaction will be.

Human-Robot Interaction Human-robot interaction is fundamentally different from typical human-computer interaction (HCI) in several dimensions. HRI differs from both HCI and HMI because it concerns systems showing complex, dynamic control systems, exhibiting a variable degrees of autonomy and cognition, and typically operating in changing, real-world environments. In addition, differences can be traced in the types of interactions (interaction roles); the physical nature of robots; the number of systems a user can simultaneously interact with; and the environment in which the interactions occur [55].

Most studies on interfacing mechanisms for remote control of robotic morphologies are conducted using a fixed input device. Ellis et. al. have developed a haptic interface for robot teleoperation [20]. Chao Hu et.al. in [24] present a visual recognition method for mobile robot teleoperation using a camera for identifying human hand postures. Marin et. al. in [35] implement an interface using virtual reality techniques. They implement a multi-level architecture, where different interaction channels are available for the user to communicate their intentions for control. The channels vary from voice commands (top level) to remote programming (bottom level).

Self-Organisation Autonomy in the exhibited behaviours of a robotic system has a key role, as it allows the robot to have an 'understanding' of its own kinematics and dynamics, its morphological constraints and the latent possibilities hidden in its environment. An autonomous robot is capable to anticipate the near future, the sequence of actions required for achieving a desired task and the transitions between them. Self-organisation at the level of robot's controller enables our approach to be agnostic towards the controlled robot, so that the robot itself can generate and 'discover' its own basic behaviours. This although, comes with the drawback that the user has only a limited, or null, role in the creation of such behaviours. A solution to this problem is discussed in the following section.

Affordances and Embodied cognition The concept of *affordances* was first introduced by J.J. Gibson [21]. It described the potential action enabled by an environment or a given object, especially one that is easily discoverable. These 'action possibilities' latent in the surroundings of an agent, need be discovered by the agent itself, providing it with a unique view.

This idea of unique possibilities arising from the same structures, is applicable on the way we perceive control devices. Indeed, different people may have the possibility of acting in different ways upon them. For such a process to be triggered, the human (operator) must have the possibility of freely manipulating the control mechanism. The interaction paradigm and the interfacing techniques should be able to support such activity. Human and robot should be interfaced in a transparent manner, such that supports the user's intuitive interactions. This interfacing mechanism should be able to adapt to the user, accommodating for their preferences, while informing the robot with the minimum possible delay. This idea carries one of the most important aspects of the work described here. As such, it has the potential of enabling intuitive interactions with the operator.

2.2 Emerging Robot Behaviours

Controlling a robotic system can be a very difficult task, depending on the morphology of the robot. Robots with 1 or 2 Degrees Of Freedom (D.o.F.) can be easy to control, such as simple two-wheeled robots. Indeed, the control can have a comparable complexity of that of a remote controlled toy car. On the other hand, complex arrangements such as 4 or 6 legged robots, or humanoids, can be very difficult to control, especially for non-standard operational tasks (e.g., not simply going forward-backward and turning). In this cases, the designer of the controlling device has to decide the level of expected autonomy of the robot by implementing a series of controlling patterns of various complexity and abstraction, such as high level commands (i.e. proceed to the next room) or low level commands (i.e. arrange a specific joint to certain degrees). In most cases the level of expected autonomy of the robot is driven by the task and the goal.

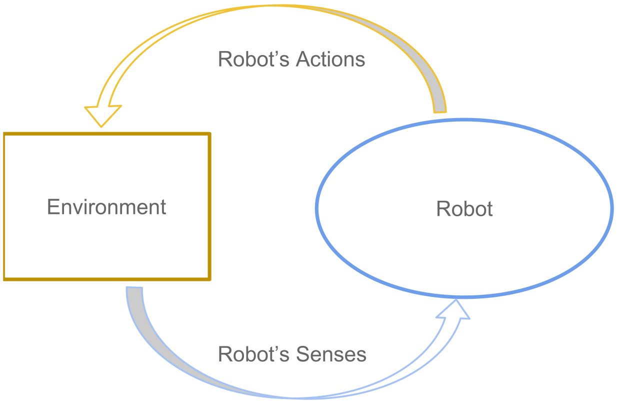


Fig. 1: The sensory motor loop of the robot. Behaviours are shaped through this cyclic interaction of the robot with its environment.

In the case of robots with no level of autonomy the control is based upon the direct manipulation of the robot's D.o.F.. In the case of remote control, the input device needs to have at least the same amount of D.o.F. so that the operator can achieve full functionality of the robotic morphology [2]. Examples of such control techniques can be found in [9] using a full body mapping or part of it as in [19].

In creating autonomous systems, two are the ways found in literature. First, that of traditional Artificial Intelligence research. Here, a top-down approach in designing robot controllers is followed, usually involving a complicated, centralised controller that makes decisions based on access to all aspects of the global state, a view that dates back to 1970 [59]. Second, systems that rely of self-organisation, which could be referred to as 'action driven'. In such systems, build from a bottom-up approach, localized, parallel, and distributed low-level controllers provide the robot with adaptive and complex behaviours. This, based on the assumption that the complexity can be achieved base on the combinatorial effects of small simple behaviours [1].

The control of complex behaviours is said to be achieved through internal models [66]. The internal model is able to identify the expected outcome of an action and the sensory consequences of a motor command. The inverse model, on the other hand, is able to identify the motor command required for the desired sensory state to be achieved. To create such models, the idea of motor babbling [41] comes forward. Inspired by Piaget's suggestion on the stages of human motor development [51], it suggests babbling is the way for exploring the relations between motors and sensors. Despite the fact that the idea of Piaget of purposeless behaviours was later challenged by research showing purposeful exploration from the early stages of development [64], it remains a powerful paradigm in creating autonomous controller for complex robots.

Following this, in robotics similar methods have been proposed for the construction of internal models of behaviours. Under this paradigm, working in model-free case (i.e. not having a complete description of the robots kinematics), robots are expected to form a model on a *tabula rasa* basis. Indeed, this is referred to as a 'cognitive capability', since this way an expectancy is formed with robot 'knowing' what move is to be performed and when, based on its own state and that of the environment. Paradigms of purposeless exploration have been suggested, through motor babbling in [11,45]. Robots perform an exploration of their sensory-motor effects, establishing a model based on the expected sensory state produced by a given motor action. On the other hand, a purposeful way of exploring robotic morphologies has been put forward by homeokinesis [14] in rigid bodies, and with morphological computation [22,50] in compliant bodies. The idea stems from the observation that behaviours can only be explored in a meaningful way if they are grounded on the robots body (sensors), motors, and environment (see figure 1).

2.3 User Behaviour Recognition

The proposed architecture, on the user side, should be able to understand the users intention for control, operate in real time, and be agnostic towards the input device and the morphology of the robots to be controlled. We understand the user intentions for control, as a series of manipulation sequences of the input device operated by the user over time. Although our problem could be seen as a time sequence classification, the need for real time control, and especially for time sequence combinations, does not allow for standard classification techniques to be used. What we want is a online and flexible mapping between the robot behaviours and the input signals, in the form of a temporal coupling between the two.

For time sequences recognition and combination [25, 62] proposed a Recurrent Neural Network (RNN) working with Parametric Biases (PB). This architecture allows for a mapping of the time sequence in the Parametric Bias (PB) space. The RNN is first trained in the time sequence using Back-Propagation Through Time (BPTT) [25], while the PB units are self organised depicting the differences in the sequence. In the operational mode, the PB are able to capture the present dynamics and move to values close to the trained ones, providing in this way a mapping of the overall RNN dynamics to an n dimensional space, n being the number of PB units. Another architecture has been proposed as an extension, capable of capturing multiple time scales of the time sequence presented to the network [70,46]. This architecture also uses PB units, in the same manner as above, and it is shown to be able to extract features based on the different scales of sampling of the sequence. Both methodologies use BPTT to train the network. Although, methods for speeding up the training time are strongly required, given that the algorithm for BPTT has a complexity proportional to the length of the training set and the number of nodes of the RNN[30]. Experiments with this type of architectures can be found in [40], where the remote control of the robot behaviours is performed with the manipulation of the Parametric Bias units.

With the aim of combining the effect of multiple time scales and the possibility of mapping the time sequence dynamics to a fixed, and smaller, dimensional space than that of the system itself, [27] formulated the echo state approach on training Recurrent Neural Networks, namely Echo State Network (ESN). ESNs could be seen to work in the same logic as Support Vector Machines, projecting the sequence into a high dimensional space, where the problem becomes linearly separable. One of the most appealing features for our application, is the fact that the network is trained using linear regression on its last layer only, reducing the complexity of training with BPTT. The network is first presented with the input sequence and the values of the output units are replaced with the desired ones. The activation of the network based on the input is recorded and the output weights are computed through linear regression of the desired output on the network's state. Thanks to ESN properties, our proposed architecture is able to learn and adapt towards the time depended manipulations of the input device using an ESN approach.

The entire system therefore works in the following way: (i) Self-generated robot behaviours are exhibited

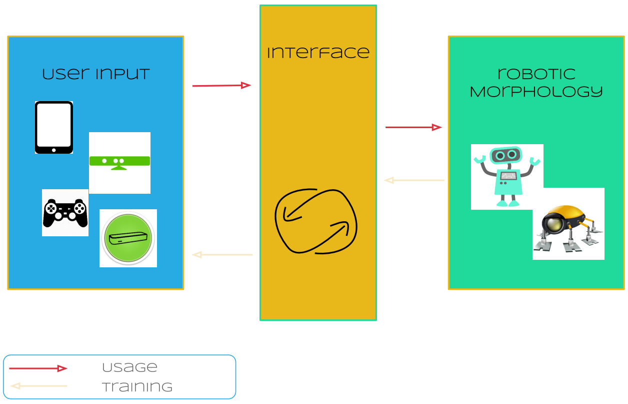


Fig. 2: Schematic representation of the proposed interface

to the user; (ii)the user is required to act on and manipulate the input device 'as-if' he was controlling the robot according to the behaviour exhibited; (iii) these input manipulation are captured and the system adapts (i.e. the ESN is trained) to map the time sequences of the user. Once the network is trained, the system can recognise the users commands and apply them to activate the robot accordingly.

3 Methods

In this section the methods selected are presented. First, we elaborate on the self-organisation of robotic behaviours and we continue with the input acquisition from the user.

Our vision is for a human-centric system, capable of understanding both the operator (human) and the operated (robot). As such, it can be seen as the cognitive architecture of the robot, capable of seamlessly integrating robots autonomous, self-generated, movements with the controlling intention injected in the system by the user through the input device. The system should be able to place the operator in-the-loop, regardless of the input device to be used and the robotic morphology at hand. For the human side, [49] mentions the importance of prediction in human robot interaction. We want to capture the real time manipulations of the input device by the operator, as they signal their intention for control. These behaviours (i.e. time depended manipulations), are then mapped to robotic behaviours, allowing for the operator to enter in the behavioural loop of the robot.

Under this paradigm, providing a rich and not restrictive repertoire of robot behaviours is essential. Self-organisation of the sensory-motor loop of the robot provides the needed variety and complexity of robotic behaviours [13].

The system should work as a mediator between the robot and the user: an interface connecting the two systems as seen in figure 2.

3.1 Self-organisation of Robotic Behaviours

In this section the methods for connecting the interface with the robotic morphology are discussed. Our goal is to explore the kinematics and dynamics of the robotic morphology as shaped through the interactions with the environment. This, together with a way of storing and reusing these behaviours found in the interaction of the robot with its environment. Indeed, in these sensory-motor contingencies of the complex system at hand, small independent controllers can be formed [1].

Methodologies for the autonomous exploration of the kinematics and dynamics can be found in the fields of artificial life and self-organisation [15,48]. The methodology presented below is able to perform both; keeping in mind that the system should be capable of exploring the morphology as fast as possible and also using a modest amount of computational resources. Another constrain is on the variation of the robotic behaviours. Here, since we want the interface to be formed dynamically - based on the interaction with the user - the exploration cannot be driven by imposed goals. As imposed goals we refer to behaviours that emerge under a supervised training. Prototypical behaviours can be useful when the operational task is known in advance. Also, for prototypical behaviours to emerge an externally derived error signal must be used to train the controller. The construction of a teacher signal requires a simulated model of the robot or the operator to physically manipulate the robots in order to perform the action to be used as training set. In [54] the behaviours of the robot are shaped based on it's interaction with a specially constructed environment. In creating behaviours in a supervised manner the dynamics of embodiment are left unexplored. Indeed, the operators assumptions on the dynamics are imposed on the robot.

Conversely, in our proposed method, the interface should be able to capture the dynamic behaviours of the morphology as revealed under an unsupervised -self-organising- manner. In doing so, we remain agnostic towards the robot, where the control mechanism is based on the morphology and not on the designer's idea of the robot. The behaviours formed in this manner should solely rely on the dynamics and kinematics of the robot at hand, making the control 'natural' towards the morphology.

Homeostasis is described as the property of a system which tries to regulate internal values at a certain level. The regulation of the system arises from the negative feedback received by the system. The general idea here is that the system has sensors and actuators affecting it. The desired condition of the sensor is reached through activation of the actuator, based upon a negative feedback loop, i.e. an error. Provided external disturbances, the system can counteract them and maintain an equilibrium.

Based on the same idea, but formulating the equilibrium as part of the system, we can derive a self-referential dynamical system and a principle for self-organisation of robotic behaviours [12,39]. The idea here is that we try to maintain a smooth control behaviour -instead of an internal variable - keeping the agent at a constant kinetic state. This property of the system -self excitation- gives rise to the name, homeokinesis.

Under the homeokinetic arrangement, learning occurs based on the error between the real behaviour -recorded by sensors- and the prediction of the robot's internal model. That is, the level at which the agent understands the robot's actions in the environment. Based on the homeokinetic principle the sensory motor loop of the controlled robotic morphology is self-organised. From the self-organisation, a repertoire of basic behaviours emerges [37], which we are able to capture in the form of behavioural 'experts'. These experts can be used later on by the operator and combined, in order to control the robot. The behaviours vary in complexity, time, and are entirely based on the interaction between the robot and the environment.

3.1.1 Homeokinetic Control

The self-organisation of the sensory-motor loop of the robot is realised as a dynamical system. For the exploration of the robot's capabilities we work as seen in [37, 36]. We want to be able to produce motor outputs from sensory readings and from them, predict the next sensory state of the robot. The creation of both a sensory-motor and a motor-sensory mapping allows us to derive an error signal for the update of the system parameters. The system is then able to create and adapt its motor-sensory mapping (referred to as the 'World Model'), in real-time, compensating for the misfit on the sensory values. The same error is also used to adapt the 'Controller', the module producing the sensory-motor mapping. This way, we perform an exploration of the kinematics and dynamics of the robots based on the robotic morphology itself.

Moreover, we are able to capture the dynamics exhibited by the robot as attractors formed in the behavioural space of the robot and reuse them. For this, we use a second module operating in parallel with the exploration module. This way, during the real time exploration of the robot's dynamics we are also able to have a series of controllers, in the form of basic behaviours, ready for the user to operate on. By activating each individual controller, the operator is able to manipulate the robot actions, driving the behaviour towards the basin of attraction described by the controller. We also show the ability to combine those basic behaviours, in order to exhibit combinations of behaviours.

The neural networks for the realisation of the above mentioned dynamical system are described below.

Both the Controller K and the World Model W are implemented as forward neural models with rate coding. The two networks working together describe the sensorimotor loop of the robot and are trained according to the homeokinetic principle. The exploration module is described, according to time t, as:

$$\tilde{\mathbf{x}}_{t+1} = W(K(\mathbf{x}_t, C), A) \tag{1}$$

The controller K generates motor outputs

$$\mathbf{y}_t = K(\mathbf{x}_t, C) \tag{2}$$

as a function of the sensory input $\mathbf{x} = x_1, x_2, \ldots, x_n$, depending on a set of parameters defined by the matrix C $[n, n+1]$ and it is defined by the equation:

$$K = g(\sum_{i=1}^{n} C_i x_i + C_{n+1}), \tag{3}$$

where g is a sigmoid function.

The world model $\tilde{\mathbf{x}}_{t+1} = W(\mathbf{y}_t, A)$ estimates future sensory input $\tilde{\mathbf{x}}_{t+1}$ from the motor output $\mathbf{y}_t = y_1, y_2, \ldots, y_n$ depending on a set of parameters defined by the matrix A $[n, n+1]$.

The parameter matrix of the world model, A, is adapted according to the delta rule [65], $\Delta w = +\eta E_W \mathbf{x}$ with the error, E_W, described by the function:

$$E_W = ||\mathbf{x}_{t+1} - \tilde{\mathbf{x}}_{t+1}||^2 \tag{4}$$

with learning rate $\eta = 0.01$.

The controller updates its parameter matrix by gradient descent with respect to the error function,

$$E_K = ||\mathbf{x}_t - \tilde{\mathbf{x}}_t||^2 \tag{5}$$

To calculate the above error, we find the $\tilde{\mathbf{x}}_t$ by calculating the motor input $\hat{\mathbf{y}}_t$ the world model should have in order to make a perfect prediction and then, the sensory input the controller K should have to predict the motor output $\tilde{\mathbf{y}}_t$. For updating the controller parameters we apply

$$C_{t+1} = C_t - \epsilon \frac{\partial E_K}{C} \tag{6}$$

with a learning rate $\epsilon = 0.1$. Matrix A is initialised from a uniform distribution in [0.5,1.5], while C in [1, 2].

For the identification, storage and reuse of the different behaviours exhibited by the robot, we use a series of m neural networks (NNs), called *experts*. Each NN is defined according to the equation,

$$(\mathbf{x}_{t+1}, \mathbf{y}_t) = N_i(\mathbf{x}_t, \mathbf{x}_{t-1}), \quad i = 1, \ldots, m \tag{7}$$

The NNs, working in parallel, compete for the prediction of the motor command $\mathbf{y}_t$ of time t and the sensory input $\mathbf{x}_{t+1}$ of the next time step in a winner-takes-all method, with only the winning network being allowed to train on the current data $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$. Thanks to this process, each NN specialises to represent a region of the entire sensorimotor space of the robot.

The NNs consist of 3 layers, feed-forward units where the hidden and output layers consist of sigmoid units, and the input layer of linear units. Online back-propagation is used to training the NN with learning rate $\eta = 0.1$. The size of the hidden layer is chosen to be 20. Assuming y to be the output vector of each neural network and x the input vector, we have

$$y = f(W_{hidden} h + b_h) \tag{8}$$

$$h = f(W_{input} x + b_x) \tag{9}$$

where f is the activation function, chosen to be a hyperbolic tangent. The matrices W_{input} and W_{hidden}, represent the weights from the input to the hidden and from the hidden to the output respectively. Finally, b_x and b_h represent the vectors for the bias units for the input and hidden layers respectively.

In each time step of the simulation the series of NNs are activated with the same input and the one with the best approximation of the next sensor values and motor commands is selected as the winner. The sample is then added to the training dataset of the winning NN which is then trained for one epoch. This way each network specialises in a single different behaviour of the robotic morphology.

The behaviour is efficiently stored in the distributed representation of the neural network being readily available for reuse by activating the neural network. When activating a NN we replace the Controller K and World Model W of the sensory-motor loop with the NN. Thus, the trained network is now producing the motor commands and the sensory predictions for the morphology.

3.2 User Behaviour Recognition

Adaptation towards the user is important, as it allows to exploit personalised patterns of communication between the user and the machine. Besides improving user experience, personalised control also enhances the usability of the system, making its usage easier and more intuitive. Adaptivity, in particular, can accommodate the user's needs, whether it is out of preference or necessary for the user itself (i.e. the machine to control has more degrees of freedom than the user, or the user can only benefit of a limited range of movements). The challenge in this case, is to create a system that is able to adapt to the user, based on a very small set of training examples, in a short time and be robust in the training.

At the same time, in order to provide a natural way of communication, the system should be able to recognise the sequence in a timely manner from a stream of data. Effectively, placing the human operator in the interaction loop.

Adaptive methodologies capable of showing the necessary behaviours have only started to appear, most of them working under a classification paradigm [71,5]. The challenges presented here are two: (a) detecting that a sequence is actually present in the data stream received from the input and (b) correctly classifying it. Most research features these two aspects with independent mechanisms [42,47]. Having a unified mechanism can save computational resources and produce faster recognitions.

Finally, another important aspect of the interaction is time. That is, the time required for the computations of the model to be performed and handling the dynamics of the input signals. Three are the main elements that require attention: (i) for the architecture, to accommodate for patterns of different lengths; (ii) to adapt in a short time, such that the user does not disengage; (iii) to perform the recognition with a low complexity of computation. This is important, as the recognition should take place fast enough for the system to have a timely response for the user.

The task of dynamic sequence recognition becomes especially complicated when working with a continuous streams of data. Breaking down the task, it can be seen to consist of two operations. One is the detection and the other the classification of the sequences. At the same time, the complexity increases when the sequences have different lengths (time spans). Methods used for the classification span from distance measures (e.g. Dynamic Time Warping) [4,53] and statistical models (e.g. Hidden Markov Models) [69,68], to artificial neural architectures (e.g. Recurrent Neural Networks)[43,3,38, 63,28] and hybrid solutions [67]. These methods vary in complexity and adaptability, with Recurrent Neural Networks being one of the most prominent direction in the field [8]. Adaptation of RNNs though, is known to have high computational complexity. At the same time, the training procedure is show to have an impasse in finding good solutions, usually referred to as a gradient vanish problem [23].

Working in real world environments can be proven to be difficult and demanding for adaptive models. Performance degrades rapidly when working directly with user data, making most methods not applicable in real world situations. Cleaning data and preprocessing is not a viable option when the demand is for a method that should be readily available to the user. The task becomes even more difficult when the input is sampled in real time and is treated continuously. Not having the ability to segment the input data, thus not having a starting and stopping point, makes the usage of recurrent methods necessary as they can integrate the time signal continuously. On the other hand, training such models requires clean data to perform well, making them difficult to train with data obtained from real users. A potential solution in this case is a structure that is able to capture the internal dynamics of a behaviour (e.g. input sequence) and thus provide a robust recognition.

A recurrent architecture that is shown to work well with noisy data under the restrictions mentioned above is the Echo State Network approach. ESNs are seen to perform surprisingly well with noisy data directly taken from a user interaction and can also adapt rapidly, making their usage for user oriented systems appealing [28, 57,31,61,6]. In our case of behaviour recognition, data comes directly from the user manipulations of an input device. Data can be noisy and the user repetition is not always perfect, resulting to training sets of data with a lot of noise and variation between samples (e.g. gestures, behaviours). The ESN approach followed here provides a stable and robust mapping of the input commands for user behaviour recognition.

3.2.1 Echo State Networks

Echo State Networks (ESN) provide an architecture for efficient training of RNN in a supervised manner. One can distinguish two main components in an ESN. Firstly, he Dynamic Reservoir (DR), a large, random, recurrent neural network with fixed weights. The DR gets activated by the input and provides a non linear response for this input. And the output signal, which is trained as a linear combination of the activations of the DR. This way the computational resources and complexity required for the training RNNs is reduced to the adaptation of the output connections of the ESN.

Assume we have a ESN consisting of N reservoir units, K inputs and L outputs. First, we need to find the state, x, of the reservoir and based on the state and the input u, we can compute the output signal y. The state extended by the input, on which we base the computation of the output, will be referred to as the extended system state on the network, z. The extended system state, depending on the particulars of the implementation can also include the output of the reservoir, i.e. the output connections of the reservoir are recurrent.

So, the state update equation, for an ESN -without any recurrent output neurons- is,

$$\mathbf{x}(n+1) = f(\mathbf{W}\mathbf{x}(n) + \mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}^{fb}\mathbf{y}(n)) \quad (10)$$

where $\mathbf{x}(n)$ is the N-dimensional reservoir state, f is a sigmoid function (usually the logistic sigmoid or the tanh function), $\mathbf{W}$ is the $N \times N$ reservoir weight matrix, $\mathbf{W}^{in}$ is the $N \times K$ input weight matrix, $\mathbf{u}(n)$ is the Kdimensional input signal, $\mathbf{W}^{fb}$ is the $N \times L$ output feedback matrix, and $\mathbf{y}(n)$ is the L-dimensional output signal.

The extended system state $\mathbf{z}(n) = [\mathbf{x}(n); \mathbf{u}(n)]$ at time n is the concatenation of the reservoir and input states - and output in the case of output recurrency-.

The output is obtained from the extended system state by

$$\mathbf{y}(n) = g(\mathbf{W}^{out}\mathbf{z}(n)) \ , \quad (11)$$

where g is an output activation function (typically the identity or a sigmoid) and $\mathbf{W}^{out}$ is a $L \times (K + N)$-dimensional matrix of output weights.

For an ESN to function properly, the *echo state property* (ESP) is essential. ESP states that the dynamics of the DR will asymptotically washout, any information added by the input or feedback, from the initial conditions. It has been observed, that this can be achieved by scaling the *spectral radious* of the DR weights W to be less than unity. The ESP is then found to hold for the DR. In [34,26] a more extensive discussion on the ESP and the dynamics of the network can be found.

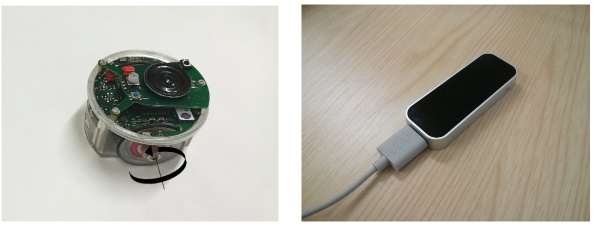
For the training of ESNs, let us assume a driving signal $u(1), \ldots, u(n_{max})$ and the extended states it generates -once passed to the network- $z(1), \ldots, z(n_{max})$. We collect the states in matrix S of size $n_{max} \times (N+K)$ and the desired outputs $d(n)$ in a matrix D of size $n_{max} \times L$. Usually, before each collection, based on the properties of the network, we apply a washout period, allowing the network to settle to the input provided.

Now, the desired output weights W^{out} can be calculated as follows. First, the correlation matrix of the extended system states are calculated, $R = S'S$. Then the cross-correlation matrix of the extended states against the desired outputs d, $P = S'D$. Finally, for the output weight matrix is found by calculating the pseudoinverse of S, $S^{\dagger}$ and then updating the weights

$$W^{out} = (S^{\dagger}D)' \tag{12}$$

The network used for our setup has a reservoir of size 300, the spectral radius is set to $a = 0.995$. The feedback matrix is sampled from a uniform distribution in [-0.01, 0.01] and the input matrix in [-0.3, 0.3]. The sparsity of the reservoir, the input and feedback weights was set to 10%.

4 Experimental Setup



(a) The E-puck robot used for the experiment.

(b) The Leap Motion Controller used as an input device.

Fig. 3: The robotic morphology and the input device used for the experiments.

The input device used for the experiment and test of the proposed system is the Leap Motion. It is equipped with two cameras. From these cameras the device creates a skeleton of the user's hand hovering above the device. In our case, the device is placed on a working surface facing upwards, and the user operates in the space above the device. The centre of the user's hand is recorded as input for our experiments. From the data provided from the device only 6 degrees of freedom (D.o.F.) are captured, representing the three rotational and three translational D.o.F of the centre of the hand. These are the 6 values that give the position and orientation of the hand in space, with the Leap Motion device as reference.

The robot to be controlled is the e-puck robot [44], which is a small two wheeled mobile robot. This choice of robot has been made based on its simplicity in order to ease the analysis. For the experiments, a simulated version of the robot is used. The control of the robot is done by adjusting the velocities of the wheels of the robot. Each wheel is controlled independently and can be set to positive and negative velocities, resulting in 2 controllable D.o.F for the robot. As sensory inputs, the positions of the robot's wheels are used. Thus, we work with proprioceptive sensory input to create the kinematic model and dynamic behaviours of the robot.

The proposed architecture works in two stages: (a) the robot self-discovers the behavioural possibilities it has; and (b) the user responds with commands for the robotic behaviours shown using the input device. From the interaction of the user with the robot, the behavioural associations between the two parties are formed. That is the dynamics of the robot's behaviours are coupled with the dynamics of user's actions on the input device. Using the input device, the user's intentions for control are expressed, with the robot changing its behaviour accordingly, following the dynamics in the users behaviour.

4.1 Stage 1 - Robotic Behaviour Exploration

As illustrated in figure (2), the architecture is placed between the two complex systems: The robot and the input device. On the robot side, the interface captures the behaviours of the robot at a sensory motor level, as a time sequences. On the input device's side, user behaviours are captured as timed sequences of the manipulations of the device by the user. In what follows the robot is the e-puck and the input device the Leap Motion Controller, as said.

The sensorimotor loop of the robot For every time step (t) the sensors of the robot are recorded with a frequency of 100Hz, the homeokinetic module of the architecture produces motor commands, and a prediction of the the resulting sensory state of the robot. In the next time step $(t+1)$ of the simulation the actual sensors are recorded and the time loop error of the homeokinetic control is calculated adjusting the behaviour of the robot. In parallel to this, in every time step (t) the 'expert' neural networks, the controllers, perform a forward pass, predicting the motor commands of time t and the sensory predictions of time $t+1$, of the homeokinetic module. Working in a winner takes all scheme, the network-expert with the best prediction adds the sensor input and motor command of that time step to its dataset, and a trains on its whole dataset once (1 epoch).

Through this procedure the robot explores and generates its own possibilities for movement in a structured and self-organised manner. In most research this procedure is addressed using motor babbling [72, 29]. Indeed, under this homoeostatic approach the robot learns to counteract external perturbations and through this interaction learns about its kinematics. However, under this approach the system cannot address the dynamics of the robotic morphology, while at the same time it is heavily dependent of the quality of the external perturbations. Instead we chose homeokinesis, in order to achieve a well structured exploration that is tied to both the robotic morphology and its environment. Through the homeokinetic rule the robot can start exploring its behavioural potentials based on internal perturbations.

The result of this procedure is a set of primitive, basic, behaviours that the robot can exhibit. Each behaviour is stored as a neural controller, becoming part of the robot's behavioural repertoire. As described and shown in [39], these behaviours can be intersected and also combined. Indeed, in their studies it is shown that transitions between them are smooth and so is the resulting robot's behaviour. Lastly, it is shown that these behaviours can be linearly combined to result new, stable, behaviours. Thus, at the end of this stage the robot is able to act in its environment, and also configuring the consequences of its actions to its sensors.

4.2 Stage 2 - Training Towards the User and the Input Device

Having adapted towards the robotic morphology, the architecture is now able to adapt towards the user. To stimulate the user, the previously explored robotic behaviour are exhibited by the robot in the simulated environment. The user, while observing these behaviours, responds by manipulating the input device in their way of preference. A schematic representation of the procedure can be seen in figure (4). The system does not impose any restriction on the users behaviour, as long as the behaviour is captured by the device. The only feedback given to the user at this stage is a notification that actions are recorded by the input device. Since the Leap Motion Controller does not require any physical contact, the user is informed when they exceed the devices recording radius. Indeed in this stage, the exploration goes towards the user, with them responding to the robot's actions. The architecture captures the user's responses as time sequences and maps them to the robotic behaviours, coupling the dynamics between the input device and the robot behaviour.

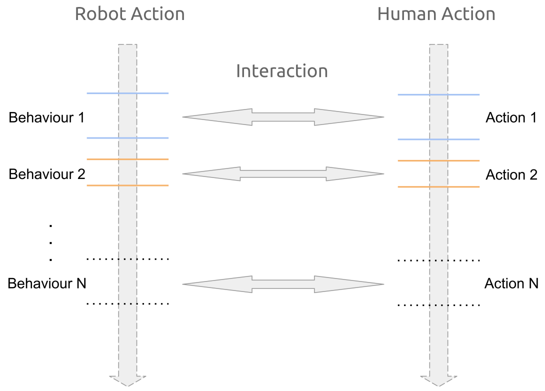


Fig. 4: Schematic representation of how the user's behaviours are mapped to robotic ones. As the robot exhibits an action the user responds to it with a manipulation of the input device. At the time span of interaction the dynamics of the robotic behaviour are mapped to the dynamics of the user behaviour.

For the time span that a behaviour is exhibited by the robot, the input device is recorded and a dataset is created. In this stage we use an Echo State Network (ESN) to capture the dynamics of the input signal. The network is trained, performing linear regression on the output weights of the network for the whole dataset. The complexity of the calculations required is small enough to allow for the training of the network within $1s$. This makes it possible for the network to be trained for each user, as the system is about to be used.

At the end of this stage the architecture is adapted towards both robot and, ultimately, the user. Having the user responding to the robot's behaviours allows for the formation of intuitive control patterns. There is no need for learning from the user, since the architecture is being adapted to suit their control signals. At the same time, the proposed method is able to provide a continuous time mapping from the dynamics of the

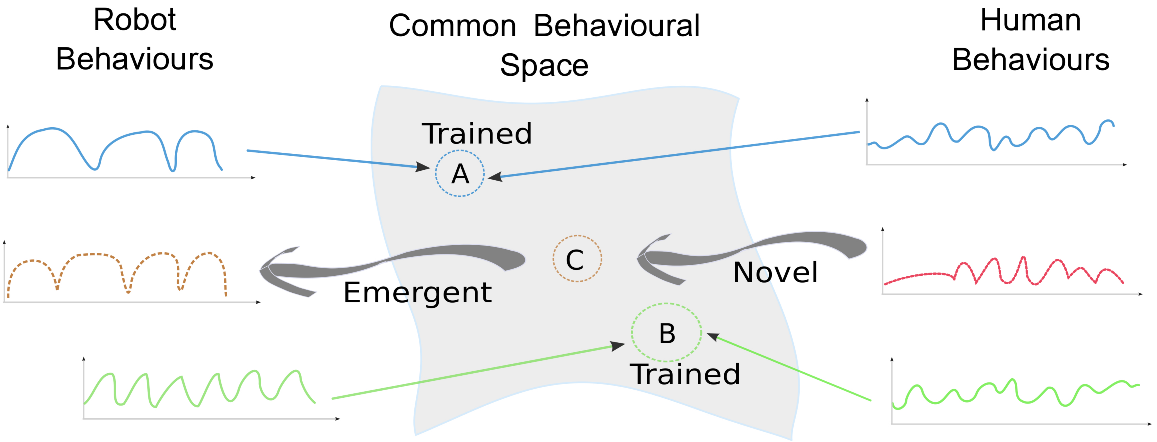


Fig. 5: Operation of the trained system. The explored robot behaviours A and B (*Stage 1*) are coupled with the behaviours of the human (*Stage 2*). This creates the Common Behavioural space, which robot and human behaviours share. As a result robotic behaviours can be invoked based on the human input (*Stage 3*). At the same time, novel human behaviours can also be mapped to this space (as marked by C), generating emergent robot behaviours, based on combinations of A and B.

input device to the robotic behaviours. As soon and as long as the user acts upon the input device the signals are propagated through the ESN, activating the robotic controllers, resulting in a continuous robotic behaviour.

4.3 Stage 3 - Controlling the Robot

Having trained both sides of the interface, the system is now ready to be used. The user, manipulating the input device, provides the input to the ESN. The D.o.F of the input device are recorded continuously over time, producing the input sequence to the ESN. Each time step recorded is fed to the ESN, exciting the internal dynamics of the network.

The network output is then used to activate the related basic robotic behaviours. The combination of behaviours is realised as a linear combination of their outputs. Each of the expert-networks, gets as input the sensory state of the robot at time (t) and produces a motor command and a sensory prediction. The motor command passed to the robot is the combination of the motor commands as guided by the ESN's output. A schematic representation of the procedure can be seen in figure (5). Based on this arrangement the robot can smoothly switch between a controlled modality and an autonomous modality. Indeed, when there is no input present from the user, the controlling system gains full control of the robot and the robot is then able to self-sustain its sensori-motor loop.

In a more technical note it is important to note that the Echo State Network, the expert controllers (NNs) that generates the basic behaviours and the simulated environment run in parallel, for the above to be achieved. Despite the computational load, the interface is able to perform in frame rate of the input device, without requiring any down sampling. This, because the code has been optimised to work in parallel fashion. For the networks, we use Theano to perform faster, distributed computations, being able to port our code to GPU if needed. For our tests we were able to run the architecture in a machine using an Intel Core i5-3340M CPU @ 2.70GHz 4 (2 cores, 4 threads), with 3.7GB of RAM and without the use of GPU acceleration, in the frame rate of the Leap Motion Controller device ($> 100Hz$).

5 Results

The results obtained from the testing of the proposed architecture are now discussed and investigated in detail. The robotic behaviours, the user behaviour recognition, and the behaviour of the system are discussed and investigated closely.

5.1 Robot Behaviours

The 1st stage of the architecture's adaptation procedure results to the formation of the modular behaviours for the e-puck. The system works by generating commands in the form of wheel velocities, while using as sensory input only the wheel positions.

Through the homeokinetic adaptation the controllers formed for robot where only four, as expected, based on the low complexity of the controlled robot. We label the four behaviours based on the behaviour we observe on the robot, as seen on following table,

Behaviour	Left Wheel Velocity	Right Wheel Velocity
Forward	1.	1.
Backward	-1.	-1.
Left	-1.	1.
Right	1.	-1.

Table 1: The table displays the wheel velocities for the self-organised behaviours of the e-puck robot. The behaviours are the result of the architectures adaptation towards the robot (referred as 1st Stage in section 4.1).

5.2 User Behaviours

The 2nd stage of the adaptation of the architecture results to a mapping from the Leap Motion Controller to the e-puck behaviours. The user observing the robot responds with controls over the Leap Motion Controller. Based on these input signals the Echo State Network is trained.

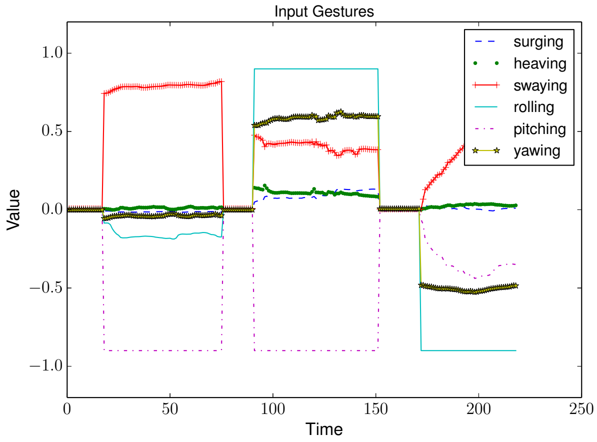


Fig. 6: User input gestures for the system to train on. Each line style displays a D.o.F. captured by the input device. All values are plotted against time. The three behaviours seen expanding in time were acquired as responses to the forward left and right behaviours of the robot.

In figure 6, the responses of the user to three of the four robot behaviours are plotted against time. The user inputs respond to forward, left and right movements of the robot, as seen from left to right. The recorded values from the input device are stored in a six-dimensional vector and for a whole input sequence in a matrix of size $[T \times 6]$, T being the length of the each sequence. There are only three of the four behaviours displayed as the backwards behaviour was not mapped to any input signal. This decision was taken to highlight some of the emergent properties of the architecture.

Pattern length variation A very useful property of the proposed architecture is that it does not impose any restrictions in the behaviour length of both user and robot. This since the sub-modules are designed to incorporate time in a non explicit way. The robot behaviours are stored in independent neural networks, each one having the possibility of storing a behaviour of different length to the others. This variability in the length of the robot's behaviour requires for the user's responses to follow the same variation. The Echo State Network used for the recognition of the user's input is able to handle variable lengths of input sequences and recognise them accordingly.

Simplicity in User Behaviour Capture Echo State Networks have a great capacity in handling noise. This allows for the architecture to capture and adapt to the user input without any preprocessing or special treatment of the input provided through the Leap Motion Controller. This feature of the architecture allows for the behaviours of the user to be captured without them being aware of the inner workings of the system. Rather, empowers them to behave in a natural and free way in the behaviours they exhibit and the input they provide.

5.3 Properties of the Architecture

In figure 7 a visualisation of the absolute position of the robot in the world is provided, for the duration of the controlled period. The robot is initially placed at point A facing upwards as indicated in the graph. In location B small modulations of the robot's steering, produced by the user, are observed from the path. In location C the robot is moving backwards, exhibiting a behaviour for which the user has not indicated an input signal related to it. This and other emergent properties of the architecture are discussed later in section 5.3.4. Moving to location D, there is a slow left turn exhibiting the ability of the architecture not only to integrate but also modulate the robot's behaviours based on the modulation of the user's input. Finally, in location E a slow right turn is exhibited by the robot, again showing that this modulation holds for all robot and user behaviours and is a valid property of the architecture.

Using only three of the robot's explored behaviours - forwards, left, and right - the architecture is able to

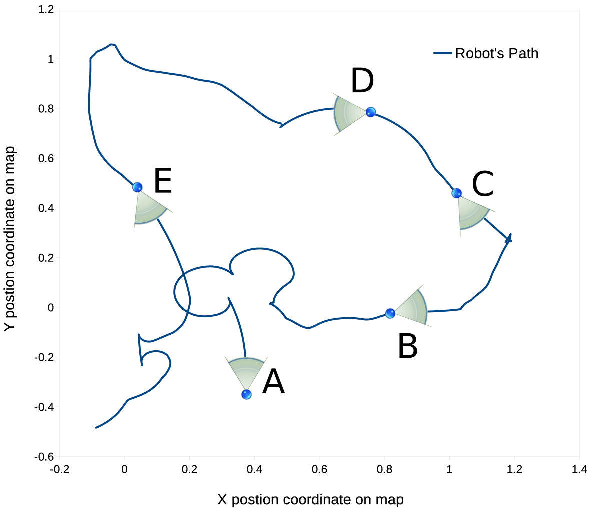


Fig. 7: Plotting of the absolute position of the robot in the simulated environment during control. The robot is represented by the blue circle and the direction that the robot is facing is depicted by the gray triangle. The five different letters represent the location of the robot at different times. Based on the extracted robot behaviours and their modulation according to the user commands the robot is navigated in the simulated environment.

produce the missing one based on the inherent properties of the user behaviour recognition module, namely the ESN. The ESN can recognise and propagate the geometrical properties of the input to its output and thus to the robotic behaviours. Indeed, the robot is able to follow this path based on the system's capability for: (a) smooth transitions between robotic behaviours, (b) modulation of the robotic behaviours based on the modulation of the user input. The system is able to produce a smooth trajectory as well as grading wheel velocities based on the intensity of the input signal. Important to note here is the fact that both user and robotic behaviours are exhibited and coupled in real time.

5.3.1 Continuous Time Operation

The system couples user input and robot behaviour in continuous time. The input signals captured from the Leap Motion Controller at each time step are propagated to the ESN sub-module, which in turn, maps them to the robotic behaviours. Each robotic behaviour is realised by its own 'expert' neural controller. These expert are combined at each time step as dictated by the user behaviour recognition module, realised as an Echo State Network. In this section, we investigate the recognition capabilities of the ESN. Based on the user input the ESN should produce at each time step an output indicating the robot behaviours to be triggered.

In figure 8 examples are shown of the activations of the robot behaviours. Triggering of the forward (figure 8a), left (figure 8b), and right (figure 8c) behaviours are plotted. In each respective plot the continuous fashion of the input recognition can be seen. For each time step of input values from Leap Motion Controller (bottom plots) an output is generated for the activations of the behaviours on the robot.

Time span of behaviours It can be observed by time span of the behaviours in the plots of figure 8, that the network can recognise them even when they are stretched for more time steps that originally exhibited (in the second stage of the architecture's adaptation 4.2). This comes as an additional property of the system to the independent time span allowed for each input behaviour. The dynamics of the ESN can be stretched in time following the user's input behaviour and thus trigger the desired robot behaviour for longer.

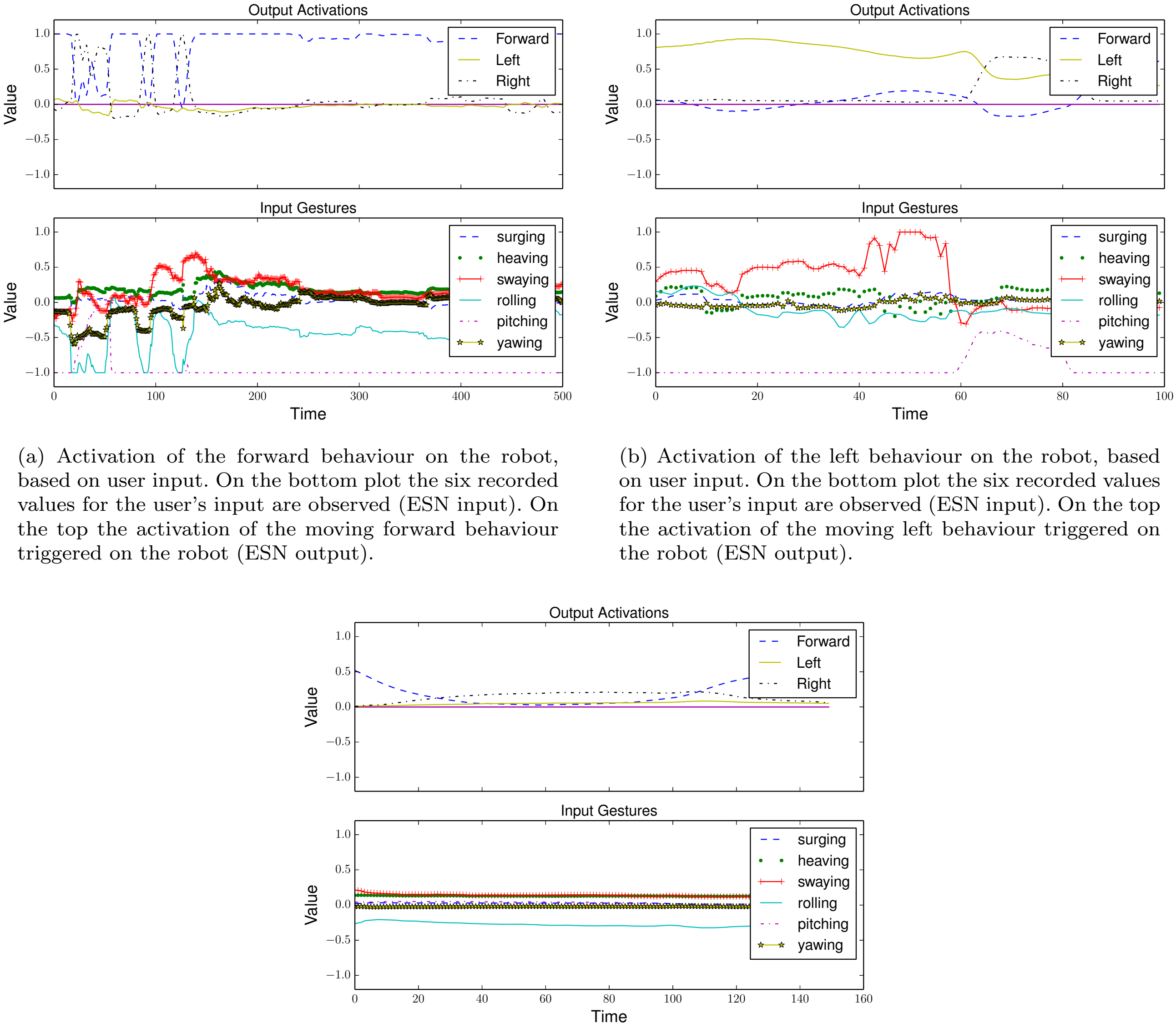
As a validation of the user input recognition module of the architecture the distances between the behaviours recognised and the trained ones are calculated and shown in table 2. The distances are calculated using Dynamic Time Warping [52] as a distance measure, as it allows for the compared timed signals to have unequal lengths. From the table the accuracy of the method is shown as the input behaviour recognised is always the right one.

Train / Test	Forward	Left	Right
Forward	**1.20**	2.28	1.87
Left	1.84	**1.11**	2.41
Right	1.45	2.38	**1.30**

Table 2: The table displays the distance between the users reference input behaviours (Train behaviours, provided at stage 2, section 4.2) and the behaviours recognised by the ESN as Forward, Left, and Right (Test behaviours, exhibited during operation). The lower the number, the lower the distance between the two. In **bold** the smallest value showing the closest behaviour to that of the user.

5.3.2 Transitions Between Behaviours

A very important aspect of the architecture is the transitions between robot behaviours under the command of the user. Having a continuous and smooth transfer from one behaviour to another necessitates the smooth



(a) Activation of the forward behaviour on the robot, based on user input. On the bottom plot the six recorded values for the user's input are observed (ESN input). On the top the activation of the moving forward behaviour triggered on the robot (ESN output).

(b) Activation of the left behaviour on the robot, based on user input. On the bottom plot the six recorded values for the user's input are observed (ESN input). On the top the activation of the moving left behaviour triggered on the robot (ESN output).

(c) Activation of the right behaviour on the robot, based on user input. On the bottom plot the six recorded values for the user's input are observed (ESN input). On the top the activation of the moving right behaviour triggered on the robot (ESN output).

Fig. 8: In the three plots the mapping between the user input and the three available robot behaviours is shown. In each figure the top plot represents the behaviour as triggered in the robot and the bottom the user input as recorded by the input device. All values are plotted against time. The time is synchronised between the top and the bottom plots of each figure, showing the real time coupling of user commands and robot behaviours.

integration of the user's input to the robot's behaviours. Moving a step closer, we also investigate how the transition between behaviours is performed in the motor level of the robot.

Transitions in Behavioural Level The transitions on a behavioural level can be observed from the plots of figure 8. Looking closely in figures 8b and 8c, it is possible to see on the top plots the smooth transitions between behaviours.

More specifically in figure 8b between time steps 60 and 80 a change in the input patterns from the user is observed (bottom plot). The 'swaying' measurement goes to zero while the 'pitch' of the hand motion increases. This in the input behaviour from the user is quickly propagated to the output of the ESN changing the behaviour mapping to the robot (top plot). The contribution of the 'Left' robot behaviour is lessened while that of the 'Right' behaviour is increased, becoming the main contributing behaviour (i.e. the one with the highest value).

In the same fashion a smooth transition between 'Forward' and 'Right' robot behaviours is observed in figure 8c. Between time steps 20 and 120 the 'Right' moving robot behaviour becomes the sole behaviour exhibited by the robot, having both 'Forward' and 'Left' mapped to near zero values (top plot).

Transitions in Motor Command Level The smooth transitions between robot behaviours can also be observed in the robot's motor values, as guided by the 'expert' controllers. The architecture is able to propagate the transitions observed in the behavioural level to the motor commands of the robot effectively.

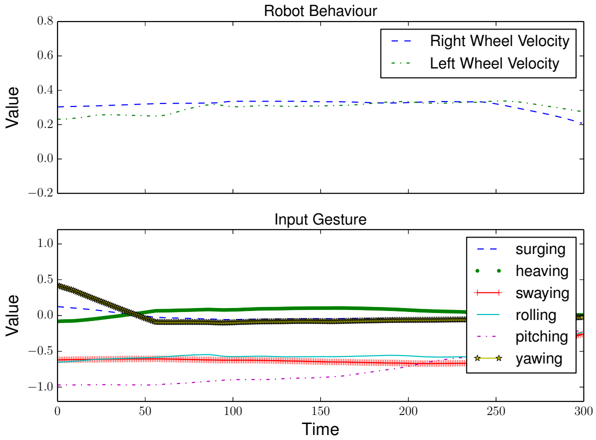


Fig. 9: Transition from a left moving to a forward moving behaviour on the robot. Based on the change in the user's behaviour the change in the robot's motor commands is observed in the top plot. On the bottom plot the six recorded values for the user's input are observed (ESN input). On the top the motor commands of the robot are observed, measured in left in right wheel velocities ("expert's" control command).

Transitions between user behaviours propagate to the 'expert' controllers of the robot resulting to stable and smooth transitions of motor commands for the roboot. From figure 9 we observe the change in the input signal just before time step 50 (bottom plot). The resulting change in the robot's behaviour is seen on the top plot of the figure. The wheel velocities of the e-puck gradually change, with the increase of the left wheel velocity until the two wheel velocities are matched. This transition results to the robot changing its behaviour to forward moving (i.e. equal wheel velocities) from the initial turn left behaviour (i.e. greater velocity on the right wheel).

5.3.3 Modulation of Behaviours

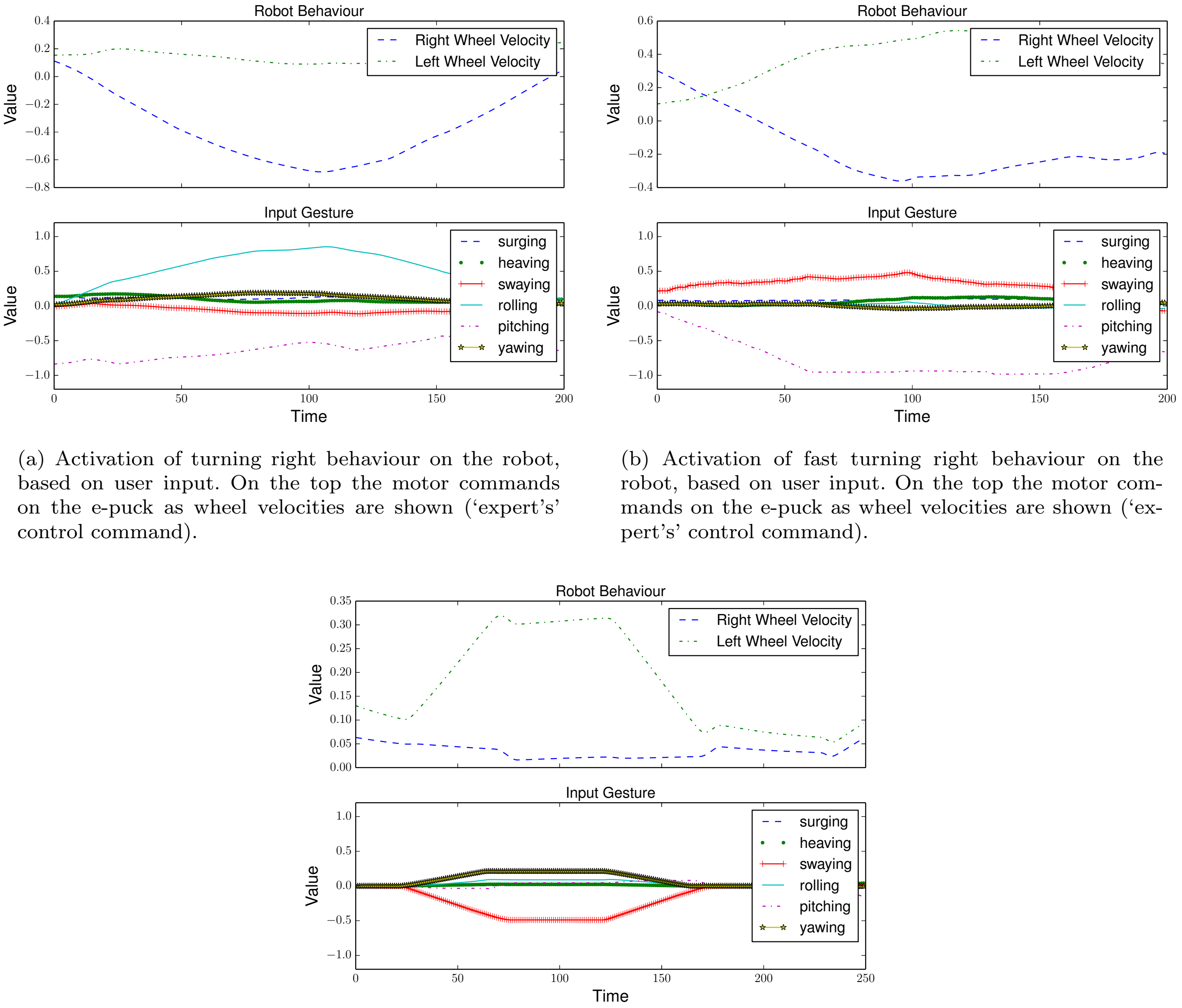
Equally important with smooth transitions is the ability of the architecture to modulate the behaviours based on the modulation of the user input. This aspect also highlights the successful coupling of the input dynamics with those of the robotic behaviours. The intensity and the variation in the user's input is propagated all the way to the motor commands of the robot, allowing the user to adjust the level at which robotic behaviours are exhibited. Since the e-puck is controlled through the velocities of the two wheels, we expect to see the robot being able to adjust the wheel speeds relative to the adjustments of the user's input.

In figure 10 three examples are shown of the architecture's ability to modulate the robot's motor controls in accordance with the modulation to the user's input. As seen in all of the three sub-figures these changes happen in the continuous, effectively embedding the user's input signal into the e-puck's behaviours.

All three sub-figures show a 'turning right' behaviour of the e-puck under the command of the user's input. In figures 10a and 10b, a 'fast' turning of the e-puck is dictated by the user input while in figure 10c a slower more gradual turning. This can be observed in the difference between the wheel velocities commanded to the e-puck robot. While in 10a and 10b the difference approaches unity, in the case of 10c both speeds are closer, measuring approximately 0.3 of difference in velocity between the right and left wheel.

Another important observation is that the architecture is able to create all three possible combinations for the turning right behaviour. Looking at top plot of each respective figure, the commands to the e-puck wheel motors in velocities are depicted, from these we observe the following. In figure 10a the left wheel velocity is commanded to near zero values and the right wheel to negative values. In figure 10b the wheels have opposing velocities, with the right wheel having a negative velocity and the left wheel a positive one. Finally, in figure 10c the last possible combination of wheel velocities is observed, with the right wheel having near zero values and the left positive ones.

The explanation for the creation of these different motor modulations of the e-puck robot is found analysing the respective user's behaviour. Using Dynamic Time



(a) Activation of turning right behaviour on the robot, based on user input. On the top the motor commands on the e-puck as wheel velocities are shown ('expert's' control command).

(b) Activation of fast turning right behaviour on the robot, based on user input. On the top the motor commands on the e-puck as wheel velocities are shown ('expert's' control command).

(c) Activation of slow turning right behaviour on the robot, based on user input. On the top the motor commands on the e-puck as wheel velocities are shown ('expert's' control command).

Fig. 10: In the three plots the mapping between the user input and the robot's wheel velocities is shown. In each figure the top plot represents the behaviour of the robot as observed through the motor commands to the e-puck robot (i.e. wheel velocities). The bottom plots depict the user input as recorded by the input device. All values are plotted against time. The time is synchronised between the top and the bottom plots of each figure, showing the real time coupling of user commands and robot motors.

Warping and comparing the user's input behaviour to the ones they exhibited during the training procedure we obtain table 3.

From the table it is observed that the behaviours are different as they result from the mix of the turning right input behaviour with other behaviours. Indeed mixing the turning right input with the moving forward results in the robot motors lowering the velocity of the right wheel to near zero values (3rd row of the table). While having a 'pure' turning right behaviour results to opposite wheel velocities. This since the pure turn right input should correctly activate a pure turn right behaviour of the robot, resulting to opposing wheel velocities. Finally, mixing the turn left with turn right

Train Test	Forward	Left	Right
Bhvr at 10a	1.68	**1.58**	2.43
Bhvr at 10b	1.93	2.88	**1.75**
Bhvr at 10c	**0.86**	1.59	1.79

Table 3: The table displays the distance measurements between the behaviours of figure 10 and the user's reference input behaviours (Train behaviours, provided at stage 2, section 4.2). The lower the number, the lower the distance between the two. In **bold** the smallest value showing the closest behaviour to that of the figure.

behaviour the right wheel speed is commanded to negative values, with the left to near zero ones (1st row of the table).

5.3.4 Emergent Behaviours

Removing the backwards behaviour from the robot's behavioural repertoire highlights one of the emergent properties of the proposed architecture. Since the robot does not have the backwards behaviours there is also no user input associated with it. To this extent, both modules -the one for the robot behaviours and the one for the user behaviour recognition- are agnostic to the possibility of the robot moving in reverse.

The user's behaviour to trigger the forward behaviour on the robot can be described as 'a forward movement of the hand' above the Leap Motion Controller. The geometrically opposite behaviour could be said to be 'a backward movement of the hand' above the Leap Motion Controller along the axis it was initially moved forward. Since there is no 'backward' gesture in the training of the system, under any classification paradigm or otherwise recognition technique we would expect no behaviour to be triggered in the robot. To the contrary, in our case the formation of the coupling between the user's input behaviour dynamics and the robot's behaviour dynamics is such that the resulting robotic behaviour is moving backwards. This comes as an intuitive response from the system to the user movement, which also fulfills the expectation of the user. At the same time it follows through with the fundamental ideas of ergonomics. It increases controllability as it adds a new behaviour to the behavioural repertoire of the robot. Additionally, it makes the interpretation of the system easier by the user, enhancing the architecture's capability for interpreting the user's commands and intentions.

In figure 11 the e-puck's motor activations and the user's input triggering the backwards moving behaviour are displayed. In the bottom plot the user's input behaviour is observed. From the user's input behaviour we can observe that most values are similar with the case of forward moving behaviour, except from the 'pitching' and 'swaying' input's values that are reversed. When the 'reversed' input signal is fed to the ESN the output representing the forward behaviour becomes negative. This together with the linear combination of the 'experts' allow for the 'opposite from forward' behaviour to be exhibited by the e-puck.

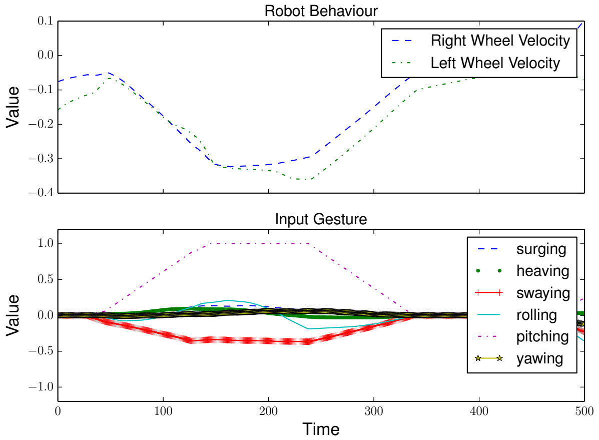


Fig. 11: Activation of backwards moving behaviour on the robot, based on user input. On the top the motor commands on the e-puck as wheel velocities are shown ('expert's' control command). On the bottom plot the six recorded values for the user's input are observed (ESN input).

Finally, a stopping behaviour emerged while using the system, as seen in figure 12. In the course of interaction, and with the user's behaviour being recorded with near zero values, the internal dynamics of the ESN start washing out. The 'memory' of the ESN (i.e. the dynamics of the recurrent connection's activations in the network) starts fading, the output levels of the network fade as well, reaching to near zero values. Since the user is still providing input, but such that the recorded values are zero, the network gradually lowers the activation of all behaviours, and this change is propagated to the robots motor commands. The velocity commands on the e-puck are decreased, reaching zero values, as seen in the top plot of the figure.

6 Conclusion

The architecture presented is capable of coupling user and robotic behaviours, enabling natural and intuitive control of the robot from the user. Indeed, a continuous control of the robot's behaviours is enabled based on the user's input signals. The methodology used and the

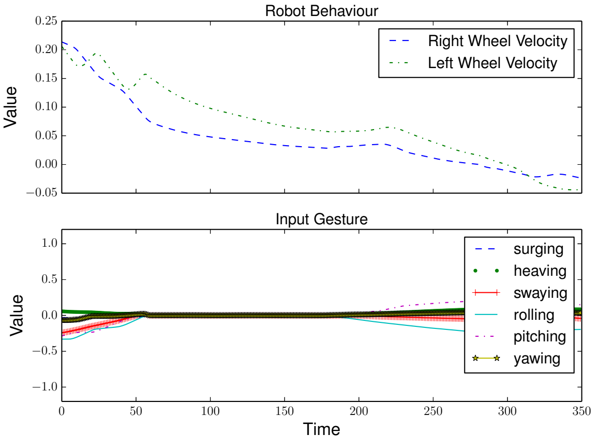


Fig. 12: Stopping behaviour of the robot, based on user input. On the top the motor commands on the e-puck as wheel velocities are shown ('expert's' control command). On the bottom plot the six recorded values for the user's input are observed (ESN input).

procedure followed has no assumptions as of the robotic morphology nor for the input device. To this extend the architecture is agnostic to both, providing a general solution to the control of autonomous robots.

Autonomous robotic behaviours have been explored, based on the principle of homeokinesis. These behaviours are grounded on the robot and its environment, and as such, allow for a meaningful representation of the robot's locomotive capabilities. Independent to the morphology, this exploration allows for the formation of a behavioural repertoire of the robot. The robot is then capable to display autonomy in the environment, being able to interact with it in a structured, predictable way. At the same time, the robotic behaviours provide the scaffolding for the display of more complex behaviours from the robot, through their combinations. This follows directly the idea of Behaviour Based Robotics, where complex behaviours can be formed from simple ones [1]. The exhibition of the behaviours, the transitions between them, and combinations of them are shown to be stable, robust, and replicable.

Furthermore, user behaviour is captured and mapped to the robotic one's independently of the input device used. Treating user input behaviours as time sequences of manipulations of the input device allows for pattern recognition methods to be used. With the use of Recurrent Neural Network architectures, user input is coupled with robotic behaviours in a robust, and efficient way. At the same time, the methods used are of low computational complexity. This allows for the architecture to adapt to the user in a short amount of time, such that the system can be ready to use in less than a second. Overall, the architecture is able to adapt to the user and their control preferences, enabling an intuitive control paradigm. The user needs not to learn the system, rather the system learns the user. This is one of the highlights of the research presented in this paper, an architecture that can provide stable adaptation to the user, enhancing the usability of the system and its ergonomy.

From the establishment of the coupling between user and robot behaviours, a paradigm of continuous, real-time control emerges. From the separation of the robot and user modules the architecture is able to handle the different time scales present in both user and robot behaviours. Indeed, user behaviours of different lengths can be easily handled by the architecture, as the recurrent neural network is able to capture and recognise them in efficient manner. In the example of the Leap Motion used here, this enables the system to support both static and dynamic gestures. Adding to that, the dynamic gestures captured can be of different lengths (i.e. time spans) from each other. This follows the structure of the robot's behaviours, as through the modularity of the controllers the behaviours can be exhibited for multiple time lengths. Having both sub-modules varying in time, enables the system to couple the user's input behaviour to the underlying robotic behaviours, providing a real-time control architecture.

The architecture is able to handle the modulation of user input behaviours, being able to propagate them to the robotic ones. Working under a mapping paradigm, instead of a classification one, user behaviours can be recognised both when only a part is presented, or when mixed with each other. This feature is propagated to the robot behaviours, allowing for the partial activation, and the mixing of the self-generated primitive behaviours. As a direct result of this property, the architecture is able to handle transitions between behaviours as well.

An emergent property of the architecture is the ability to reverse behaviours, based on reversed input signals. Since the Leap Motion captures the location of the user's hand above the device, geometrical opposite input behaviours can trigger opposing robotic behaviours. Having not adapted on the reversed behaviour neither in the user side nor in the robot's side, the architecture is, nevertheless, able to handle a reversed input behaviour and also trigger the intuitive reversed robotic behaviour as a result. This feature of the setup highlights the robustness and the generalisation of the architecture while also providing support to the truthfulness of the approach towards human in the loop systems.

Ultimately, we can see the control method presented in this paper as an extension of the robot's sensory apparatus. The on-time connection provided by the architecture allows for the operator's experience of the environment to be mediated to the robot. Actions or reactions of the operator to their environmental stimuli are channelled to the robot through the interfacing of the architecture. Based on the ideas of situated and embodied cognition, we can investigate the way we communicate our movements to another morphology. The way that we understand and use our body. We can observe how the material agency of the input device affects and affords the user's control patterns. An investigation on how the mediated experience of another body -through the input device and interface- can result to a kinaesthetic experience, enhancing the way understand the morphology and its environment. As a parallel to Boden's 'conceptual spaces', this architecture aims to provide the constrains and allowances for the range of possible mappings between user and robotic morphology.

From a more applicative perspective, in the field of human machine interaction a surge in adaptive methods is being observed. Being able to adapt the communication mechanism that relates the machine to the user's preferences can enhance the usability of such systems, the performance of the communication, and decrease the training effort required by the operator in order to user them [71, 5].

In particular, in the field of assistive robotics, our approach can provide a fast and reliable way of adapting the system to the users preferences. This may accommodate cases of increased of decreased mobility and the usage of unorthodox input devices. Being able to capture, train and recognise user behaviours from their preferred input method can be alleviating for use cases that cannot be taken into account in standard design procedures.

In the field of rehabilitation robotics, being able to train to each individual patient, enables their potential (no matter how limited) to be used to its maximum. While better usage of their body enables and works towards a better and faster recovery, working with an adaptive system, enables for the system to adapt to the user and thus allow for the maximum of their possibilities to be exploited. This parallel increment in the abilities of both system and patients has the potential to boost the rehabilitation effects [18]. Given the nature of the framework we are proposing, as the patient is allowed to be more expressive in their environment, the more possibilities for communication they will discover, enabling in turn explorative behaviours, that would enhance the self-driven motivation for improvement, which is already known to be beneficial in rehabilitation [10, 7, 60]. In cases where rehabilitation is not possible (e.g. Alzheimer or Parkinson disease) such interaction is found to slow that overall degradation of the patient's condition, by the same property of 'exercising' the available patient functionality [16].

In conclusion, besides the interesting technological challenge behind this work, and the proposed shift of paradigm from the adaptation of the user to the machine, to the reciprocal adaptation of the machine to the user, enabling then the machine of the necessary level of 'intelligence', we believe the general paradigm we propose may have in the future, beneficial effects on enduring and relevant societal problems, such as those related to motor and cognitive rehabilitation in general, and to the general problem of empowering disable people with more, diverse and adaptive control means toward the external world.

References

1. Arkin, R.C.: Behavior-based robotics. MIT press (1998)
2. Ashby, W.R., et al.: An introduction to cybernetics. An introduction to cybernetics. (1956)
3. Bailador, G., Roggen, D., Tröster, G., Triviño, G.: Real time gesture recognition using continuous time recurrent neural networks. In: Proceedings of the ICST 2nd international conference on Body area networks, p. 15. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (2007)
4. Bodiroža, S., Doisy, G., Hafner, V.V.: Position-invariant, real-time gesture recognition based on dynamic time warping. HRI - ACM/IEEE International Conference on Human-Robot Interaction pp. 87–88 (2013)
5. Bodiroža, S., Stern, H.I., Edan, Y.: Dynamic gesture vocabulary design for intuitive human-robot dialog. Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction - HRI '12 p. 111 (2012). DOI 10.1145/2157689.2157710
6. Büsing, L., Schrauwen, B., Legenstein, R.: Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. Neural computation **22**(5), 1272–1311 (2010). DOI 10.1162/neco.2009.01-09-947
7. Byl, N.N., Pitsch, E.A., Abrams, G.M.: Functional outcomes can vary by dose: learning-based sensorimotor training for patients stable poststroke. Neurorehabilitation and neural repair **22**(5), 494–504 (2008)
8. Čeranský, M., Tio, P.: Comparison of echo state networks with simple recurrent networks and variable-length Markov models on symbolic sequences. Artificial Neural NetworksICANN 2007 pp. 618–627 (2007)
9. Cheng, L., Sun, Q., Su, H., Cong, Y., Zhao, S.: Design and implementation of human-robot interactive demonstration system based on kinect. In: Control and Decision Conference (CCDC), 2012 24th Chinese, pp. 971–975 (2012). DOI 10.1109/CCDC.2012.6242992
10. Dayan, E., Cohen, L.G.: Neuroplasticity subserving motor skill learning. Neuron **72**(3), 443–454 (2011)
11. Demiris, Y., Dearden, A.: From motor babbling to hierarchical learning by imitation: a robot developmental pathway pp. 31–37 (2005)
12. Der, R.: Selforganized robot behavior from the principle of homeokinesis. Proc. workhop soave pp. 39–46 (2000)

13. Der, R., Martius, G., Hesse, F.: Let it rollemerging sensorimotor coordination in a spherical robot. ARTIFICIAL LIFE X: PROC. X INT. CONF. ON … (2006)
14. Der, R., Steinmetz, U., Pasemann, F., et al.: Homeokinesis-a new principle to back up evolution with learning (1999)
15. Di Paolo, E.A.: Organismically-inspired robotics: homeostatic adaptation and teleology beyond the closed sensorimotor loop. Dynamical systems approach to embodiment and sociality pp. 19–42 (2003)
16. Disbrow, E., Russo, K., Higginson, C., Yund, E., Ventura, M., Zhang, L., Malhado-Chang, N., Woods, D.L., Sigvardt, K.: Efficacy of tailored computer-based neurorehabilitation for improvement of movement initiation in parkinson's disease. Brain research **1452**, 151–164 (2012)
17. Doisy, G., Meyer, J., Edan, Y.: A Gesture Learning Interface for Simulated Robot Path ShapingWith a Human Teacher. IEEE Transactions on Human-Machine Systems **44**(6), 788–795 (2014). DOI 10.1109/THMS.2014.2331618
18. Draganski, B., Gaser, C., Busch, V., Schuierer, G., Bogdahn, U., May, A.: Neuroplasticity: changes in grey matter induced by training. Nature **427**(6972), 311–312 (2004)
19. Du, G., Zhang, P., Mai, J., Li, Z.: Markerless kinect-based hand tracking for robot teleoperation. International Journal of Advanced Robotic Systems **9**(10) (2012)
20. Ellis, R., Ismaeil, O., Lipsett, M.: Design and evaluation of a high-performance haptic interface. Robotica **14**, 321–327 (1996). DOI 10.1017/S0263574700019639
21. Gibson, J.J.: The theory of affordances. Hilldale, USA (1977)
22. Hauser, H., Ijspeert, A.J., Füchslin, R.M., Pfeifer, R., Maass, W.: Towards a theoretical foundation for morphological computation with compliant bodies. Biological cybernetics **105**(5-6), 355–370 (2011)
23. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
24. Hu, C., Meng, M., Liu, P., Wang, X.: Visual gesture recognition for human-machine interface of robot teleoperation. In: Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on, vol. 2, pp. 1560–1565 vol.2 (2003). DOI 10.1109/IROS.2003.1248866
25. Ito, M., Tani, J.: On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system. Adaptive Behavior **12**, 93–115 (2004)
26. Jaeger, H.: The echo state approach to analysing and training recurrent neural networks-with an erratum note
27. Jaeger, H.: Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach. GMD-Forschungszentrum Informationstechnik (2002)
28. Jirak, D., Barros, P., Wermter, S.: Dynamic gesture recognition using echo state networks. In: Proceedings, p. 475. Presses universitaires de Louvain (2015)
29. Kajic, I., Schillaci, G., Bodiroza, S., Hafner, V.V.: A biologically inspired model for coding sensorimotor experience leading to the development of pointing behaviour in a humanoid robot. In: Proceedings of the Workshop" HRI: a bridge between Robotics and Neuroscience". 9th ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI 2014) (2014)
30. Kolen, J.F., Kremer, S.C.: A field guide to dynamical recurrent networks. John Wiley & Sons (2001)
31. Laje, R., Buonomano, D.V.: Robust timing and motor patterns by taming chaos in recurrent neural networks. Nature neuroscience **16**(7), 925–933 (2013)
32. Leitner, J., Luciw, M., Förster, A.: Teleoperation of a 7 DOF Humanoid Robot Arm Using Human Arm Accelerations and EMG Signals. juxi.net
33. Lii, N.Y., Chen, Z., Roa, M.a., Maier, A., Pleintinger, B., Borst, C.: Toward a task space framework for gesture commanded telemanipulation. 2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication pp. 925–932 (2012). DOI 10.1109/ROMAN.2012.6343869
34. Manjunath, G., Jaeger, H.: Echo state property linked to an input: Exploring a fundamental characteristic of recurrent neural networks. Neural computation **25**(3), 671–696 (2013)
35. Marin, R., Sanz, P., Nebot, P., Wirz, R.: A multimodal interface to control a robot arm via the web: a case study on remote programming. Industrial Electronics, IEEE Transactions on **52**(6), 1506–1520 (2005). DOI 10.1109/TIE.2005.858733
36. Martius, G.: Taming the Beast: Guided Self-Organization of Behavior in Autonomus Rboots
37. Martius, G., Fiedler, K., Herrmann, J.M.: Structure from behavior in autonomous agents. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, pp. 858–862. IEEE (2008)
38. Maung, T.H.H., et al.: Real-time hand tracking and gesture recognition system using neural networks. World Academy of Science, Engineering and Technology **50**, 466–470 (2009)
39. Melidis, C., Marocco, D.: An exploration on intuitive interfaces for robot control based on self organisation. In: C.J. Headleand, W.J. Teahan, L. Ap Cenydd (eds.) Artificial Life and Intelligent Agents, *Communications in Computer and Information Science*, vol. 519, pp. 73–79. Springer International Publishing (2015)
40. Melidis, C., Marocco, D.: A human centric approach to robotic control. In: Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on, pp. 2441–2446 (2015)
41. Meltzoff, A.N., Moore, M.K.: Explaining Facial Imitation: A Theoretical Model. Early development & parenting **6**(June), 179–192 (1997). DOI 10.1002/(SICI)1099-0917(199709/12)6:3/4¡179::AID-EDP157¿3.0.CO;2-R
42. Mitra, S., Acharya, T.: Gesture recognition: A survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **37**(3), 311–324 (2007)
43. Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., Kautz, J.: Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
44. Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.C., Floreano, D., Martinoli, A.: The e-puck, a robot designed for education in engineering. In: Proceedings of the 9th conference on autonomous robot systems and competitions, vol. 1, pp. 59–65. IPCB: Instituto Politécnico de Castelo Branco (2009)
45. Morasso, P., Sanguineti, V.: Self-organizing body schema for motor planning. Journal of Motor Behavior **27**(1), 52–66 (1995)
46. Namikawa, J., Nishimoto, R., Tani, J.: A neurodynamic account of spontaneous behaviour. PLoS Comput Biol **7**(10), e1002,221 (2011). DOI 10.1371/journal.pcbi.1002221
47. Neverova, N., Wolf, C., Taylor, G.W., Nebout, F.: Multiscale deep learning for gesture detection and localization.

In: Workshop at the European Conference on Computer Vision, pp. 474–490. Springer (2014)

48. Nolfi, S.: Evolutionary Robotics: Exploiting the Full Power of Self-organization. Connection Science **10**(3-4), 167–184 (1998). DOI 10.1080/095400998116396
49. Ogata, T., Sugano, S.: Imitation based Human-Robot Interaction -Roles of Joint Attention and Motion Prediction- pp. 283–288 (2004)
50. Pfeifer, R.: Morphological Computation: Connecting Brain, Body, and Environment, pp. 2–3. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
51. Piaget, J.: The origin of intelligence in the child. Routledge & Paul (1953)
52. Ratanamahatana, C., Keogh, E.: Everything you know about dynamic time warping is wrong. Third (2004)
53. Ren, Z., Yuan, J., Meng, J., Zhang, Z.: Robust part-based hand gesture recognition using kinect sensor. IEEE Transactions on Multimedia **15**(5), 1110–1120 (2013). DOI 10.1109/TMM.2013.2246148
54. Saunders, J., Nehaniv, C.L., Dautenhahn, K.: Teaching robots by moulding behavior and scaffolding the environment. In: Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction, HRI '06, pp. 118–125. ACM, New York, NY, USA (2006). DOI 10.1145/1121241.1121263
55. Scholtz, J.: Theory and evaluation of human robot interactions. 36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the **3**, 10 pp. (2003). DOI 10.1109/HICSS.2003.1174284
56. Shah, D., Schneider, J., Campbell, M.: A Sketch Interface for Robust and Natural Robot Control. Proceedings of the IEEE **100**(3), 604–622 (2012). DOI 10.1109/JPROC.2011.2179772
57. Sheng, C., Zhao, J., Liu, Y., Wang, W.: Prediction for noisy nonlinear time series by echo state network based on dual estimation. Neurocomputing **82**, 186–195 (2012)
58. Shneiderman, B.: Designing the user interface-strategies for effective human-computer interaction. Pearson Education India (1986)
59. Simon, H.A., Newell, A.: Human problem solving: The state of the theory in 1970. American Psychologist **26**(2), 145 (1971)
60. Sterr, A., Freivogel, S.: Motor-improvement following intensive training in low-functioning chronic hemiparesis. Neurology **61**(6), 842–844 (2003)
61. Sussillo, D., Abbott, L.F.: Generating coherent patterns of activity from chaotic neural networks. Neuron **63**(4), 544–557 (2009)
62. Tani, J.: Learning to generate articulated behavior through the bottom-up and the top-down interaction processes. Neural Networks **16**(1), 11 – 23 (2003)
63. Tsironi, E., Barros, P., Wermter, S.: Gesture Recognition with a Convolutional Long Short-Term Memory Recurrent Neural Network In : Proceedings of the European Symposium on Artificial Neural Networks ,. Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN) pp. 213–218 (2016)
64. Von Hofsten, C.: An action perspective on motor development. Trends in cognitive sciences **8**(6), 266–272 (2004)
65. Widrow, B., Hoff, M.E., et al.: Adaptive switching circuits. (1960)
66. Wolpert, D.M., Miall, R.C., Kawato, M.: Internal models in the cerebellum. Trends in Cognitive Sciences **2**(9), 338–347 (1998). DOI 10.1016/S1364-6613(98)01221-2
67. Wu, D., Pigou, L., Kindermans, P.J., Le, N.D.H., Shao, L., Dambre, J., Odobez, J.M.: Deep dynamic neural networks for multimodal gesture segmentation and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(8), 1583–1597 (2016). DOI 10.1109/TPAMI.2016.2537340
68. Xu, D., Chen, Y.L., Lin, C., Kong, X., Wu, X.: Real-time dynamic gesture recognition system based on depth perception for robot navigation. In: Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on, pp. 689–694 (2012). DOI 10.1109/ROBIO.2012.6491047
69. Xu, D., Wu, X., Chen, Y.L., Xu, Y.: Online dynamic gesture recognition for human robot interaction. Journal of Intelligent & Robotic Systems **77**(3-4), 583–596 (2015)
70. Yamashita, Y., Tani, J.: Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment. PLoS Comput Biol **4**(11), e1000,220 (2008). DOI 10.1371/journal.pcbi.1000220
71. Yin, Y.: Real-time Continuous Gesture Recognition for Natural Multimodal Interaction. = Bulletin of Center for Educational Research and Development (21), 1–8 (2014)
72. Zambelli, M., Demiris, Y.: Online multimodal ensemble learning using self-learned sensorimotor representations. IEEE Transactions on Cognitive and Developmental Systems
73. Zhao, Q., Brine, J., Filev, D.P.: Defining Cybernetics. IEEE Systems, Man, & Cybernetics Magazine **1**(2), 18–26 (2015). DOI 10.1109/MSMC.2015.2421325