

2021-09-24

# Development of a Low-cost Data Glove using Flex Sensors for the Robot Hand Teleoperation

Zhu, S

<http://hdl.handle.net/10026.1/18374>

---

10.1109/isrimt53730.2021.9596972

2021 3rd International Symposium on Robotics & Intelligent Manufacturing Technology  
(ISRIMT)

IEEE

---

*All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.*

# Development of a Low-cost Data Glove using Flex Sensors for the Robot Hand Teleoperation

Shuo Zhu

Center for Robotics and Neural Systems, Plymouth University,  
UK, PL4 8AA

Ashraf Fahmy

ASTUTE 2020 in Future Manufacturing Research Institute,  
College of Engineering, Swansea University, Swansea, United  
Kingdom

Johann Sienz

ASTUTE 2020 in Future Manufacturing Research Institute,  
College of Engineering, Swansea University, Swansea, United  
Kingdom

Angus Stuttaford-Fowler

Center for Robotics and Neural Systems, Plymouth University,  
UK, PL4 8AA

Chunxu Li

Center for Robotics and Neural Systems, Plymouth University,  
UK, PL4 8AA

\* Corresponding author: Chunxu.li@plymouth.ac.uk

**Abstract** — This paper focuses on the control of a simulated robot hand using a data glove, which gathers information from resistors that measure the flexion of the user's fingers. The data glove used was a commercially available controller, using 5 flex sensors on each finger to collect data. Arduino Uno R3 microcontroller board was used to power the data glove which is based on the ATmega328P microcontroller. CoppeliaSim was used as a platform for simulation, with a compatible model for the hand found within the software's community. C++ is used within both the CoppeliaSim and Arduino environments, with a micro-USB connecting the two. This paper allows for real time control with the proposed data glove, which transmit the data in the form of a string of 5 different values. In the future, many adjustments and modifications could be made to allow for more precise control and eventually more degrees of freedom. A video demo for the proposed design can be seen from: <https://youtu.be/w-GqzigfkXY>.

**Keywords**-data glove; flex sensors; Arduino; CoppeliaSim; flexion

## I. INTRODUCTION

Converting sensor input to program output is the basis of many fields of robotics, from computer vision to virtual and augmented reality, both commercially and in research. This project was originally planned to use a real anthropomorphic robotic hand. However, current circumstances have led to the project being simulation-based [1]. A data glove is a piece of wearable technology that enables human-computer interaction, converting real-world movement of the hand into usable sensory data for software. Many approaches to capturing physical data have been developed over the past 50 years, starting even before the conception of the Sayre glove in 1977, transitioning from optical and magnetic-based motion capture to a more hands-on wearable form [2]. However, the data captured is often still used in conjunction with gyroscopic and inertial sensors for tracking the hand in 3D space.

Its applications can vary from biomedical research, the teleoperation of a system, to virtual and augmented reality, with different, and focus on precise finger control to more gesture-based means allowing for different techniques to be applicable in different scenarios [3]. This gives more creativity and

engineering freedom over the way in which the glove is designed to suit its functionality. The core principle of operation between gloves does remain very similar, in that flex sensors are still often placed on the fingers to measure finger flexion, whilst positional sensors track the whole hand. This paper will explore a variety of different sensors before choosing the most viable for the task of simple teleoperation. A simulated anthropomorphic hand will be controlled, with one degree of freedom per finger and each finger containing 3 coupled joints. The goal is to allow for operation of the aforementioned hand with a personally designed glove. By achieving this, the data glove will prove themselves as an effective tool for real-world teleoperation, as well as any other suitable task. The hope is to build on this in the future, so that positional and haptic feedback could be implemented within the glove, along with more degrees of freedom to allow for increased tracking precision.

Most data gloves available on the market can cost thousands of pounds, effectively limiting their accessibility to much of their potential target audience. Despite this, completing this research work will demonstrate that the key elements of these systems can be mirrored and even improved upon from our previous work [4]. This paper was based around both a robotic hand and a data glove, with more focus on designing and building a real-world hand. The initial requirements were to create both glove and hand with one degree of freedom per finger, and then expanding on that to give more degrees of freedom and control for both. These requirements form the basis of the aims of the project, with each aim representing a milestone towards assembling and testing the data gloves:

1. An anthropomorphic robotic hand will be modelled in CAD software, then exported and assembled as a robot within simulation software. Mapping the joint angles and distances would allow for forward and inverse kinematics to be used to calculate end effector location from known joint angles, or vice versa respectively.

2. The joint angle data collected will be used to control the simulated robotic hand. The data can be transferred over serial or writing to an array file to allow the simulation to read the joint angles and move to the target position.

3. A data glove will be made to be used in conjunction with the simulation, having already completed the software design for the operation of the hand.

## II. PROPOSED METHODOLOGIES

### A. Motion Tracking

The initial approach to capturing physical data from the hand was motion tracking in an enclosed environment. There were predominantly 4 methods employed to track the hands' position, which later developed into what is used today. The position and orientation data can be relayed to a receiver from gyroscopes and accelerometers. This method uses units that contain multiple tracking components but is susceptible to inaccuracies over time and gyroscopic drift, but this is correctable with filtering techniques such as the Kalman filter which is widely used today for all positional tracking techniques [5]. The most relevant to the project would be the aforementioned first methods. The first being represented by the Leap Motion Controller, an optical hand tracking module commonly used for joint angle and positional data acquisition [6]. Features such as fingertips are extracted from the camera's shot, and vectors are drawn in to represent position and direction. Angles can then be calculated using the intersection between vectors [7].

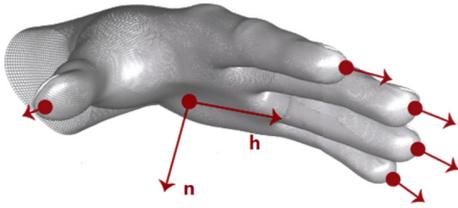


Figure 1: Features are extracted to draw vectors on each point, which can be used in finding the joint angles and relative position of the rest of the hand. Adapted from reference 5.

### B. Potentiometers and String Potentiometers

A potentiometer is a three-terminal adjustable voltage divider that comes in many forms, widely used for a plethora of different applications [8]. They typically consist of a sliding contact that moves along a resistive element, so that the position of the slider dictates the value of output resistance. Used in conjunction with an attached mechanism, such as a lever or joystick, they become a very versatile tool in measuring position in a system [9]. The voltage across the component can be found using the simple voltage divider rule [8]:

$$V_L = \frac{R_2 R_L}{R_1 R_L + R_2 R_L + R_1 R_2} \cdot V_S \quad (1)$$

Dividing through by  $R_L$  and cancelling gives the simplified and well-known form of the equation, as the load is large in comparison to other resistances:

$$V_L = \frac{R_2}{R_1 + R_2} \cdot V_S \quad (2)$$

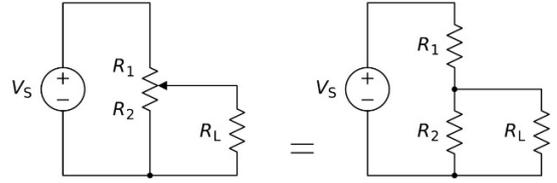


Figure 2: A further simplified version of the voltage divider circuit within the potentiometer, adapted from reference 7.

An extension of the potentiometer, the string potentiometer attaches a pullable string mechanism to the adjustable handle of the potentiometer, accompanied by a gearing and spring mechanism for multiple turns and returning the string potentiometer to a neutral position. The rotational sensor can be in the form of a potentiometer or a rotary encoder [10]. An electrical signal proportional to the cable's linear extension is produced, which can be differentiated as a function of position to give velocity.

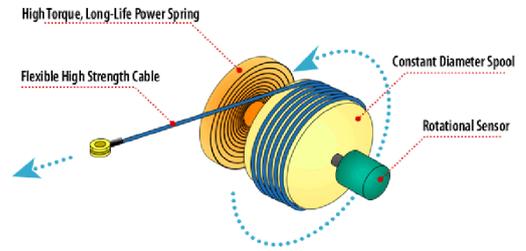


Figure 3: The inner workings of a string pot. The cable is pulled and extends from around the spool, which is measured by the rotational sensor. The spring returns the string pot back to a resting position. Adapted from reference 8.

### C. Flex Sensors

Flex sensors come in different varieties and have been used since the first data glove [2]. The Sayre glove used optical flex sensors consisting of a flexible tube housed with a light source at one end and a photovoltaic cell at the other. Bending the tube would evenly decrease the amount of light reaching the photocell [11]. This principle has been taken forward and adapted over time to make different kinds of flex sensors. The fibre optic flex sensor is similar to the optical flex sensor, but instead uses optical fibres.

Conductive ink flex sensors work in much the same way as the capacitive flex sensor. Set up in a similar fashion to the capacitive flex sensor, a strip of flexible resin has a conductive ink deposited thereon with a segmented conductor on top, forming a flexible potentiometer [12]. Bending the sensor decreases the cross-sectional area and an increase in length, resulting in a predictable change in resistance [13].

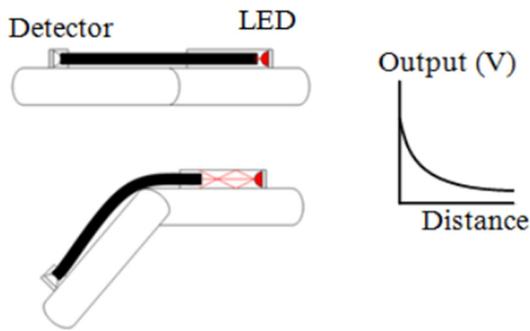


Figure 4: Principle of operation for a photocell flex sensor. Flexion results in a decrease in light intensity, as shown by the graph. Adapted from reference 9.

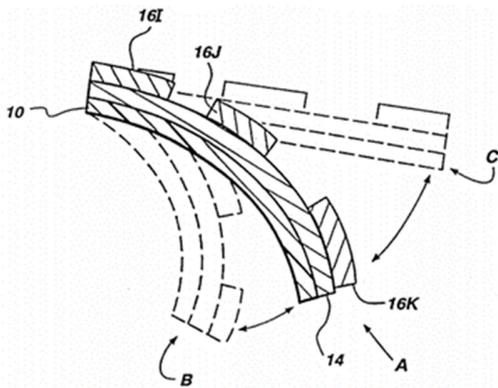


Figure 5: Operation of a conductive ink flex sensor. Different levels of bend result in different resistive properties of the interaction between the ink and conductive pads, which can be predictably measured. Adapted from reference 15.

These have recently been adapted to use a new conductive carbon material, commercially known as Velostat or Linqstat. It is used as a packaging material made of a polymeric foil impregnated with carbon black to make it electrically conductive, making it useful for protection against electrostatic shock [14]. It can be used in place of conductive ink, and is commonly used for most hobby flex sensors today.

#### D. Strain Gauges

A more unorthodox method would be to use strain gauges, or load cells for each finger to measure flexion. A load cell is a transducer that converts an applied force into a desired and measurable output. This can be in the form of detecting different properties, such as bending, compression, changes in the magnetic field as input, and electrical, hydraulic or pneumatic for output [15]. The most popular and relevant type of which is the strain gauge. These could be used at the knuckle of a data glove, so that flexion of the fingers causes an increase in strain proportional to the angle of bending. This would allow for not only joint angle calculation, but for precise control of how much force is applied across the robotic hand. It would also interact well with haptic feedback and play more into the illusion of a robotic hand being an extension of operator's own body.

### III. DEVELOPMENT OF THE CONTROL INTERFACE

#### A. Hardware Design

Traditional data gloves use lever-actuated potentiometers and do not involve the glove's or operator's own nervous system. They normally have a lever coupled to a potentiometer for each finger and the thumb, which attach via a red adjustable ring. The resistor values and range are normally unknown to a traditional data glove with lever-actuated potentiometers owing to its mechanical structure. Therefore, the resistances at rest and max flexion of the whole finger can not be calculated. This does not solve the resistance of just the potentiometer, but the theoretical resistance of the finger, as there is probably a potential divider circuit for each finger.

$$ADCOutput = \frac{V_{in}}{V_{ref}} \cdot ADCmaxcount \quad (3)$$

To overcome the above-mentioned issues, flex sensors were chosen as the optimal sensor for each finger, as they were easily available and relatively inexpensive compared to other options, given the lack of facilities and resources to 3D print or create a more advanced solution. Each flex sensor in its own potential divider circuit with a 10k resistor. Each flex sensor has a resting resistance of 30k and a fully bent resistance of 90k. Using these values, the voltage for each state and range of voltages can be calculated. In order for each sensor to have the full range of motion, they have to be able to bend and relax whilst staying seated on top of the finger. In this paper, small holes were cut into the sides of each finger so that zip ties could be used to create a track for the flex sensors to slide down. Each one is fixed at the top of the finger so that it will always bend from the top, allowing the rest of the sensor to slide through the zip ties up to the clinch connector. The wires coming from the sensors are fed through a small loop to avoid tangling, and each pair of wires leads to its respective voltage divider.

#### B. Software Design

The Arduino was used as the microprocessor of the data glove. The Arduino Analogue and Serial classes were used to transmit data from the glove to the simulation in the form of float values representing resistance. The simulation receives this data and converts it into a joint angle in radians, which each joint in the finger will move to. The loop runs at the clock speed of the Arduino, which is 16MHz. The simulation runs at a slower rate than the Arduino's clock speed, meaning it will never have to wait for new values to come up. As the values are not stored in an array or external file, a buffer is not needed to ensure proper data transfer. Storing these values would be helpful for replaying the movements of the hand, which could be used to teach an AI how to carry out a task, inspired from our previous work [16].

A model of the hand is established, and the measurement specification of the model allows the calculation of a homogeneous transformation matrix [17]. This would have allowed for a kinematic model to be produced that could use forward and inverse kinematics to calculate end effector locations and joint angles given values of one or the other. The hand was imported into Fusion360 as a single body mesh. Each link and separate body in the hand had to be manually separated to build the linked robot. To place the joints, a cylinder had to

be extracted from the housing of each joint. During the meshing and decimation process, the model's joints lost their cylindrical properties and became warped, which placed each joint off centre and made the collisions between each body a lot slower to calculate within the physics engine. In order to find the homogeneous transformation matrices, the Denavit-Hartenberg parameters need to be established.

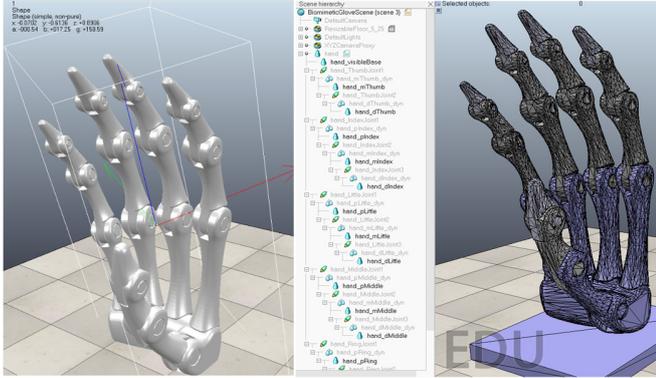


Figure 6: The information text in the top left shows the distance between two joints, as well as the difference in orientation and position. This can be used to fill out an approximate Denavit-Hartenberg table.

Each finger will be approximated to follow the Denavit-Hartenberg parameters [18]:

- The  $z$ -axis must be the axis of rotation or direction of motion.

- The  $x_n$ -axis must be perpendicular to both the  $z_n$  and  $z_{n-1}$  axes.

- The  $x_n$ -axis must intersect both the  $z_n$  and  $z_{n-1}$  axes.

- The  $y_n$  axis' coordinate frame must follow the right-hand rule.

Each joint in their respective fingers was aligned with each other using CoppeliaSim, the  $z$ -axis of each coordinate frame is already aligned and therefore alpha and  $z$ -axis displacement 'd' will always be zero. Theta denotes the angle of rotation through the  $z$  axis, and displacement 'a' represents distance between links. The thumb is the only negative value for theta as the thumb needs to rotate in the opposite direction (clockwise) to the other fingers. Each coordinate frame is linked to the next by a rotation through  $z$  between 0-63 degrees. A rotation in the  $z$  axis is represented by the matrix:

$$R_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ where } \theta = (-)63^\circ \quad (4)$$

To approximate further, each translation occurs along the  $y$ -axis, when in the simulation it is actually a combination of displacements in the  $x$  and  $y$  axes. The transformation matrix is given by:

$$T = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \text{ where } y = \text{displacement} \quad (5)$$

Therefore, the homogeneous transformation matrix between one coordinate frame and the next would be represented by:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (6)$$

$$H_n^{n-1} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & x \\ \sin\theta & \cos\theta & 0 & y \\ 0 & 0 & 0 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Filling the values in from the table for each finger would give the total homogeneous transformation matrix, used in forward and inverse kinematics.

#### IV. EXPERIMENTAL STUDIES

It was found in the test that once one finger was functioning correctly, the same code was used to read the joint angles of each finger, so that the fingers can operate together. The main problem was the speed at which the simulation was occurring, coupled with a constantly vibrating ring finger. To speed up the simulation, a load must be taken off the physics engine by decreasing the number of calculations it has to make. To do this, alternate links in the robot can be set to different mask levels within the simulation, so that they do not collide. Also, an invisible and simplified version of the model was created and attached to the visible model, so that the simplified version could run faster whilst the original can visually copy it.

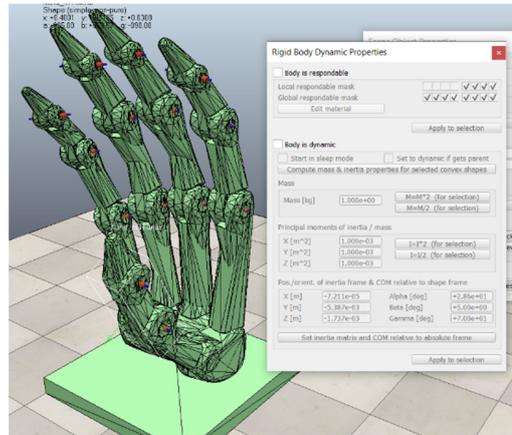


Figure 7: Simplified version of model to speed up the physics engine. The mask layers indicate which objects the individual body will collide with.

The green hand shown in Figure 7 is a simplified mesh version of the original hand. The local responsible layers show that the base of the thumb does not interact with anything on the first 4 mask layer boxes, i.e., the tip of the thumb and the base. The segments of the fingers alternate in masks so that adjacent parts do not collide but anything past that will, which is an effective way of both speeding up the simulation and fixing the issue of parts colliding with themselves.

The prototype hand could be now constructed. The index finger was completed first as a test for how fixing the sensors in place would work. It was found that fixing them at the tip and

letting the back slide freely was the optimal solution for bending each sensor fully whilst not obstructing the clinch connector.

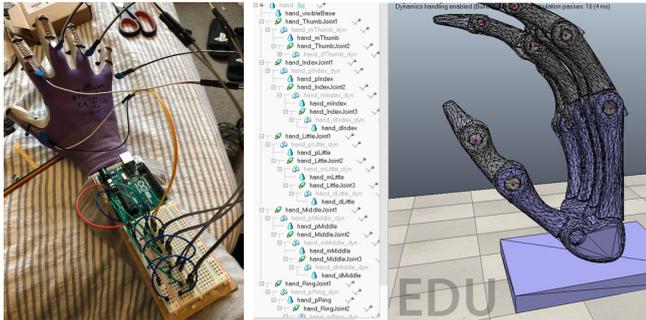


Figure 8: Glove building montage and the joint angles being printed to the CoppeliaSim output. They have been converted to radians by the simulation.

Putting a voltmeter in parallel with the output showed the voltage fluctuating around 1.2V when relaxed, slightly under the number predicted earlier. This dropped to around 400mV when flexed. The glove can now be plugged into the simulation. The values used within the code that convert and calibrate the float values into joint angles must be changed as well. A flat value of the minimum reading was subtracted from the resistance read from a finger, to give a number in degrees between 0 and approximately 70. This was then converted to radians to set the joint angle targets for each motor in the joints.

### V. ANALYSIS AND DISCUSSION

The experimental studies have proven successful in controlling a simulated anthropomorphic hand using our designed data glove. The model used for the hand was very complex with a large mesh, which caused the simulation to run slower, and made the importation process more difficult. Investing more time into making a new and simpler model in CAD would have decreased the number of simulation-based issues later on. This also would have fixed the misalignment of the joints and parts of the hand interacting with each other in unexpected ways. The jitter experienced in the hand is also partially due to the rate of change of joint angles. Currently, the joint angles are sent at a rate of 16Mhz and are subject to slight changes, which causes the hand to vibrate slightly at times. Taking a running average of the joint angles coming in would have decreased their variation and made operation smoother. The data glove and simulation are used together to find the joint angle data and convert it to a useable form. This would also allow the gloves to be more compatible with other systems without much modification. Moreover, storing the joint angles in an array allows for a recreation of the simulation, which would be helpful for teaching an AI to copy a task or replaying the users' actions.

### VI. CONCLUSION

In this paper, one low-cost data glove was designed to control a simulated robot hand in real time. The goal was met and demonstrated the key features of what forms a data glove. Different components and methods of data acquisition were explored before choosing suitable ones for a prototype glove that was made. The model was used for simulation of an anthropomorphic robotic hand and set up for simulated

teleoperation. Experimental studies validated the proposed control scheme and the success of the data glove design. Additionally, this paper was an opportunity to learn more about current industry applications and conventions. It is estimated that the addition of features such as position and orientation data, battery life and wireless connectivity, and haptic feedback could be implemented for a low cost, allowing the glove to compete on the market.

### REFERENCE

- [1] C. Li, A. Fahmy, S. Li, and J. Sienz, "An enhanced robot massage system in smart homes using force sensing and a dynamic movement primitive," *Frontiers in Neurobotics*, vol. 14, p. 30, 2020.
- [2] D. Sturman and D. Zeltzer, "A survey of glove-based input," *IEEE Computer Graphics and Applications*, vol. 14, no. 1, pp. 30–39, 1994.
- [3] C. Li, C. Yang, J. Wan, A. Annamalai, and A. Cangelosi, "Neural learning and kalman filtering enhanced teaching by demonstration for a baxter robot," in *2017 23rd International Conference on Automation and Computing (ICAC)*, pp. 1–6, IEEE, 2017.
- [4] C. Li, A. Fahmy, and J. Sienz, "Development of a neural networkbased control system for the dlr-hit ii robot hand using leap motion," *IEEE Access*, vol. 7, pp. 136914–136923, 2019.
- [5] J. LaViola, "A survey of hand posture and gesture recognition techniques and technology," *Brown university, providence, ri*, vol. 29, 1999.
- [6] "Ultraleap, World-leading Hand Tracking: Small. Fast. Accurate.," 2021.
- [7] Y. Du, S. Liu, L. Feng, M. Chen, and J. Wu, "Hand gesture recognition with leap motion," 2017.
- [8] HolyDumphy, "Potentiometer Adjustable Voltage Divider Introduction," 2020.
- [9] "Potentiometer, Preset Potentiometers and Rheostats," 2021.
- [10] "What is a String Pot: String Potentiometers," 2020.
- [11] L. Majeau, J. Borduas, S. Loranger, Y. El-Iraki, J. Lavoie, D. Banville, V. Latendresse, V. Beland, J. Daniel-Rivest, A. Thiaw, H.-D. Bambara, T. P. Beausoleil, W. Trottier-Lapointe, and J. Lapointe, "Dataglove for consumer applications," in *2011 7th International Workshop on Fibre and Optical Passive Components*, pp. 1–4, 2011.
- [12] G. B. Langford, *Flexible potentiometer in a horn control system*. United States of America Patent, may 1994.
- [13] "Flex Sensor Data Sheet," tech. rep., Spectra Symbol, 2001.
- [14] N. Drake, *Polymeric Materials for Electrostatic Applications: A Report from Rapra's Industry Analysis and Publishing Group*. 1996.
- [15] "What is a Load Cell and how does it work?."
- [16] C. Li, C. Yang, and C. Giannetti, "Segmentation and generalisation for writing skills transfer from humans to robots," *Cognitive Computation and Systems*, vol. 1, no. 1, pp. 20–25, 2019.
- [17] 90srobot, "Biomimetic Hand Endoskeleton."
- [18] M. W. Spong, S. Hutchinson, M. Vidyasagar, et al., *Robot modeling and control*, vol. 3. wiley New York, 2006.