2011

Application of Intermediate Multi-Agent Systems to Integrated Algorithmic Composition and Expressive Performance of Music

Kirke, Alexis

http://hdl.handle.net/10026.1/895

http://dx.doi.org/10.24382/4900
University of Plymouth

Copyright Statement

Application of Intermediate Multi-Agent Systems to Integrated Algorithmic Composition and Expressive Performance of Music

By

Alexis John Kirke

A thesis submitted to the University of Plymouth

In partial fulfilment for the degree of

Doctor of Philosophy

Faculty of Arts

September 2011

Application of Intermediate Multi-Agent Systems to Integrated Computer-aided Composition and Expressive Performance of Music

Abstract

We investigate the properties of a new Multi-Agent Systems (MAS) for computer-aided composition called IPCS (pronounced "ipp-siss") the Intermediate Performance Composition System which generates expressive performance as part of its compositional process, and produces emergent melodic structures by a novel multi-agent process. IPCS consists of a small-medium size (2 to 16) collection of agents in which each agent can perform monophonic tunes and learn monophonic tunes from other agents. Each agent has an affective state (an "artificial emotional state") which affects how it performs the music to other agents; e.g. a "happy" agent will perform "happier" music. The agent performance not only involves compositional changes to the music, but also adds smaller changes based on expressive music performance algorithms for humanization. Every agent is initialized with a tune containing the same single note, and over the interaction period longer tunes are built through agent interaction. Agents will only learn tunes performed to them by other agents if the affective content of the tune is similar to their current affective state; learned tunes are concatenated to the end of their current tune. Each agent in the society learns its own growing tune during the interaction process. Agents develop "opinions" of other agents that perform to them, depending on how much the performing agent can help their tunes grow. These opinions affect who they interact with in the future. IPCS is not a mapping from multi-agent interaction onto musical features, but actually utilizes music for the agents to communicate emotions. In spite of the lack of explicit melodic intelligence in IPCS, the system is shown to generate non-trivial melody pitch sequences as a result of emotional communication between agents. The melodies also have a hierarchical structure based on the emergent social structure of the multi-agent system and the hierarchical structure is a result of the emerging agent social interaction structure. The interactive humanizations produce micro-timing and loudness deviations in the melody which are shown to express its hierarchical generative structure without the need for structural analysis software frequently used in computer music humanization.

Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank Professor Eduardo R. Miranda for his support, direction and experience, as well as Simon Ible, John Matthias and other members of the Interdisciplinary Centre for Computer Music Research. I would also like to thank my family and friends for their support and encouragement. This work has been funded by the UK EPSRC project “Learning the Structure of Music” (EPD063612-1).

AUTHOR'S DECLARATION

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

This study was financed with the aid of a studentship from the Engineering and Physical Sciences Research Council.

Signed…….

Date…….

Chapter 1 – Introduction

This chapter introduces the motivation, background and original contributions presented in this thesis. The contributions are based around a new application of Intermediate Multi-agent Systems: applying them to combined computer-aided composition and expressive performance of music. The original contributions are a result of defining and investigating a multi-agent system called "IPCS" – the Intermediate Performance Composition System.

1.0 Motivation

Computer-aided composition (CAC) of classical music had been around since 1957 when The Illiac Suite for String Quartet - the first published composition by a computer - was published by Hiller and Isaacson (1959). Since then there has been a large body of such music and research produced, with many successful systems created for automated and semi-automated computer composition (Buxton 1977; Roads 1996; Miranda 2001a).

The generation of novelty is at the heart of many CAC systems. Without some way of generating new material, a CAC will churn out the same material time after time. To avoid this, many systems utilize some form of random number generation (Roads 1996). A more recent alternative is the use of complexity which is ordered but unpredictable. A popular type of system that generates such complexity is found in the field of Artificial Life or A-Life (Langton 1986). A-Life investigates systems related to life, their processes, and evolution; it does this most often through computer simulations and models. An example Artificial Life system is Cellular Automata (CA) – groups of artificial stationary growing and dying cells, which produce complex and eye-catching patterns, based only on very simple rules of interaction. Cellular Automata have two elements in common with much A-Life work which has made them attractive to

composers for use in CAC: they generate complexity with order and structure (Bedau 2003), and they inspire composers by their visual variety (Brown 2002). So although Cellular Automata and some other A-Life systems can generate unexpected complexity, there is an inherent order – they are not solely random. This is often called "Emergent" behaviour (Chalmers 2002).

One field which has a large intersection with Artificial Life is Multi-agent Systems, which (along with computer-aided composition and computer expressive performance) is one of the 3 key areas addressed by this thesis. These systems can be viewed as a step up in complexity from Cellular automata. Each agent in an MAS is a digital entity which can interact with other agents to solve problems as a group, though not necessarily in an explicitly co-ordinated way. Many tasks fall naturally into such a distributed realm – e.g. multiple web searchBots, server load balancing, or multiple automated enemies in a computer game; and there are a number of agent-oriented programming languages available (Georgeff 2009). What often separates agent-based approaches from normal object-oriented or modular systems is their emergent behaviour (Panzarasa and Jennings 2006). The solution of the problem tackled by the agents is often generated in an unexpected way due to their complex interactional dynamics, though individual agents may not be that complex. As with the application of Cellular Automata in CAC, these social dynamics can be both artistically functional – for example each agent in an ensemble can contribute a motif or play an artificial instrument in a piece of music; or artistically motivational, inspiring an algorithmic composer to produce the "music of artificial societies".

This thesis was initially motivated by an area of multi-agent systems which has had no implemented applications to the best of our knowledge in computer-aided composition: Intermediate MAS. The "Intermediate" label is explained as follows. There are multi-

agent systems which have no significant individual processing in an agent, no memory and simple reactive interaction rules; but from which significant group computational power can emerge. These are sometimes known as swarm intelligence systems (Bonabeau et al. 1999). These are almost as simple as Cellular Automata and have been investigated significantly in computer-aided composition. There are also MAS where each agent has a significant amount of AI processing ability, memory and/or interaction protocols. These are described as “heavy duty” agents by Parunak et al. (2006). This thesis will look at MAS existing between these two levels of complexity, that Parunak et al. (2006) describe as “Intermediate Agents”. They are more complex than simple swarm systems but no agent in an Intermediate MAS has heavy duty AI. Most MAS for computer-aided composition have either been of the low processing/swarm type or the heavy-duty type. There have been some musical Intermediate MAS – for example (Dahlstedt and McBurney 2006) and (Gong et al. 2005) - but none of the intermediate musical systems has been implemented and tested as a computer-aided composition system.

Another motivation for this thesis can be related to the idea of “Music Acts” (Murray-Rust and Smaill 2005), which examines language that not only provides information, but causes a change in the environment - i.e. “acts” on the environment. This can be compared to improvisers, whose music not only creates the overall improvisation, but also provides communicative cues to each participant in real time about what each musicians intentions are. Miranda (2003) presents a multi-agent system to investigate the evolution of musical culture. Agents develop their social bonds and musical repertoire using the same language – that of music. They do not have a separate protocol for bonding and for developing their musical repertoire. This thesis examines a system where the data and the protocol are the same – i.e. both are purely musical.

The question then arises in such a system – what is being communicated? This thesis' system is based around affective (emotional) communication. Whalley (2009) discusses the conversational mode of composing, and argues that a natural core in this mode is the use of affective information. It is also informative to observe that music is often called the language of emotions (Trainor and Schmidt 2003; Peretz et al. 1998). Furthermore designing music by emotional specification is a growing field in computer-aided composition, for example in soundtrack composition (Behravan 2007), or for computer game music (Livingstone 2008). So one natural approach is to consider that a society of agents may be using music to communicate emotions, and looking into how that could be used to specify compositions. A corollary to this design decision is that some method is needed by the agents to estimate the affective content of music that is passed on to them by another agent. So part of this thesis will be introducing a simple linear model for performing such an analysis.

Another element of computer-aided composition that will be addressed through multi-agent techniques is musical structure – Western music often has a hierarchical structure (Schoenburg 1970). For example - notes make up motifs, motifs make up phrases, phrases make up themes, themes make up sections, sections make up movements and movements make up a piece. This thesis investigates the application of a frequent method used in MAS – agent interaction success measures – to the issue of creating musical structure. A number of intermediate MAS have agents which develop opinion/interaction-history measures of each other, and these measures affect their future interaction. These opinion measure methods will be applied here to investigate mapping the resulting agent "social structure" to composing a musical hierarchical structure.

Having discussed computer-aided composition, intermediate multi-agent systems, and emotional communication, the final element motivating this thesis is expressive performance of music. In the early 1980s the seeds of a problem were sown as a result of synthesizers being developed and sold with built-in sequencers. The introduction of MIDI led to an explosion in the use of sequencers and computers, thanks to the new potential for connection and synchronisation. These computers and sequencers performed their stored tunes in perfect metronomic time, a performance which sounded "mechanical". They sounded mechanical because human performers normally perform expressively – for example speeding up and slowing down while playing, and changing how loudly they play. The performer's changes in tempo and dynamics, and other subtle musical features, allow them to express a fixed score – hence the term expressive performance (Widmer and Goebl 2004). Publications on computer expressive performance of music have lagged behind computer-aided composition by almost quarter of a century. But from the end of the 1980s onwards there was an increasing interest in automated and semi-automated Computer Systems for Expressive Music Performance (CSEMP). A CSEMP is a computer system which – given a score in some form – is able to generate expressive performances of music. For example software for music typesetting will often be used to write a piece of music, but some packages play back the music in a relatively mechanical way – the addition of a CSEMP enables a more "human sounding" playback, giving a better idea of how the final performance may sound.

Studies have found that a significant factor affecting a performer's deviation from the score is the musical structure of that score – i.e. the start and end of phrases and themes, etc. A significant amount of CSEMP effort is in either manually analysing the musical structure of the score/audio, or researching systems to do this (semi-)automatically. The automated approaches are not particularly accurate (Holland 2000; Cambouropoulos

2001; Hamanaka, et al 2004). However, many computer-aided composition systems generate a piece based on some local and/or global structure or musical form which can often be made explicitly available. So in computer music systems it could be more computationally efficient and more amenable to musicological accuracy to design a protocol allowing the computer composition system to communicate structure information directly to the CSEMP, or simply combine the systems into an integrated CAC and CSEMP system.

For example one could imagine a Finale-type (MakeMusic 2010) scoring tool which had some computer-aided composition algorithms, and linking the humanized playback facility of the scoring tool to the musical structures create by the composition algorithms. In this case it would be less computationally efficient and less able in its ability to find the actual structure, if the approach utilized involved separate composition and expressive performance systems – i.e. where a score is generated and the CSEMP sees the score as a black box and performs a complex structure analysis. The system presented in this thesis is designed to generate an expressive performance of its music as it composes it.

1.1 Thesis Aim

The aims of this work is as follows:

To investigate/explore the dynamics of a new system that extends the field of Computer-aided Composition and Computer Expressive Performance through the application of intermediate MAS to integrated algorithmic composition and performance.

1.2 Contributions

In the course of this research, the following contributions will be made:

- The demonstration of the applicability of intermediate multi-agent systems to algorithmic composition.
- A proof of concept that a multi-agent system can develop non-trivial melody pitch structures through affective interaction of agents without explicit melodic knowledge.
- A demonstration that multi-agent social structures can generate musical structure on thematic and sectional levels as well as on a note or phrase level.
- A proof of concept that combined algorithmic composition and expressive performance system applies expressive performance rules in a way that expresses the musical structure and without having to do an explicit musical analysis of the composition.
- A music-emotion analyzing model which takes as input a monophonic MIDI file and estimates its affective content.

1.3 Thesis Structure

The remainder of this thesis is structured as follows:

Chapter 2 – Literature Review. This chapter surveys the literature in the fields relating to the aims and contributions of this thesis. There is an overview of multi-agent systems, going on to divide them into low processing, intermediate, and higher processing types. Computer-aided Composition (CAC) in general is briefly reviewed, and then in more detail when it comes to the application of low processing, intermediate and heavy duty MAS for CAC. CAC systems which allow users to specify elements of the composition/performance using affective labels are also surveyed. This is followed by a review of the newer and less well-known field of computer systems for expressive performance of music. The chapter then surveys multi-agent systems for expressive

performance, of which there is only one; and concludes with a discussion of the areas lacking in the systems reviewed which will be addressed by the thesis system IPCS.

Chapter 3 – Method. Introduces IPCS – the Intermediate Performance Composition System, which is an intermediate multi-agent system designed to allow a user to generate music with integrated expressive performance. The system will be described in detail; including its affective, musical, and social networking aspects. The reasoning behind the design, as well as the way in which these details address what is lacking in previous systems reviewed, will be discussed as details are introduced, as well as the reasoning behind the design.

Chapters 4 – Experiments. The chapter lays out a series of experiments to investigate/explore the dynamics of IPCS. Most experiments focus on the adjusting various parameters of IPCS to see the effect on the resulting music and properties in the system. They will also be used to show how the performance elements express the structure of the music. An initial listening experiment will also be done. Results will be reported in this chapter for all experiments.

Chapters 5 – Discussions. The results of the experiments in the previous chapter are analysed and contextualised.

Chapters 6 – Conclusions and Future Work. The dynamics of IPCS which have been explicated and highlighted by the results and discussions will be summarized to describe more general properties of the system, together with concluding comments and potential further work suggested by this thesis.

Chapter 2 - Literature Review

This chapter presents a representative survey of past literature and research in the key fields relevant to this thesis:

- Multi-agent Systems in General (i.e. not specifically music-related)
- Multi-agent Systems for Computer-aided Composition
- Affective-based Systems for Computer-aided Composition
- Computer Expressive Performance
- Combined Computer Expressive Performance and Computer-aided Composition

2.0 Introduction

The purpose of this review is to contextualise the work of the thesis and to support the original contributions (highlighting any elements lacking in previous work in relation to the aims of this thesis). An overview of the review is shown in Figure 1.

2.1 Multi-agent Systems Overview

Multi-agent systems (MAS) are an extremely large field of research, dating back to the founding of Distributed AI in the late 1970s (Weiss 1999). The precise definition of multi-agent systems has been approached in different ways. For example Wooldridge (2004) focuses on agents that are situated in an environment, with each agent autonomous, and with explicit goal-directed behaviour – not purely reactive.

1. Multi-agent Systems (MAS)
    a. Low Processing MAS
    b. Heavy Duty MAS
    c. Intermediate MAS
2. Computer Aided Composition (CAC)
    a. Low Processing MAS for CAC
    b. Heavy Duty MAS for CAC
    c. Intermediate MAS for CAC
    d. Emotion-based CAC systems
3. Computer Systems for Expressive Performance
    a. Emotion-based CSEMPs
    b. Hierarchical Structure based-Systems
    c. MAS for Expressive Performance
4. Combined Performance and Composition Systems

Figure 1: Overview of Literature Review Chapter

However there are a number of multi-agent system where the agents have a central controller (Murray-Rust and Smaill 2005; Baxter et al. 2007), and there are a number of researchers who include reactive-based systems as multi-agent. This thesis will take this second, more wide-ranging approach. Such an MAS definition has been described as defining a complexity continuum going from agents with no or very low processing, up to agents with significant AI processing algorithms such as Expert Systems or Belief/Desire/Intention (Rao and Georgeff 1991); which Parunak et al. (2006) describe as Heavy Duty agents. They use the label Intermediate Agents for those which lie

between these levels. It is this Intermediate type which is the focus of this thesis. The review of the MAS field will examine some examples from each of these three complexity-levels of MAS moving upwards in individual agent complexity, starting with low processing MAS, followed by "Heavy Duty" MAS, and then Intermediate MAS.

Before doing this, Cellular Automata (Langton 1986) will be briefly examined. Although CA are usually considered a separate field to MAS, low-processing MAS have been described as a generalization of CA (Beni 2005; Spicher et al. 2009) – i.e. CA are one step below low processing MAS in complexity. So to complete the picture of the complexity continuum, we will begin with CA, especially since they have been used in Computer-aided Composition. Cellular Automata are most simply exemplified as a fixed grid in which cells can be turned black or white depending on which neighbouring cells are black and white. Cells could be thought of as being the location of immobile agents which generally have finite and potentially very short life spans. The most common CA is the Game of Life (Gardener 1970) with 3 rules: a cell dies if too many or too few cells are alive around it (simulating loneliness and overcrowding respectively); and a cell comes to life if just the right number are alive around it. Based on these simple rules and the cells which are painted black at the start of the iterations, great complexity can emerge – for example the moving "Glider Gun" in Figure 2.

CA have been applied to image processing (Rosin 2006), architectural design (Herr and Kvan 2007), generative art (Boden and Edmonds 2009) and algorithmic composition (Miranda 1994; Beyls 1989; Kirke and Miranda 2007).

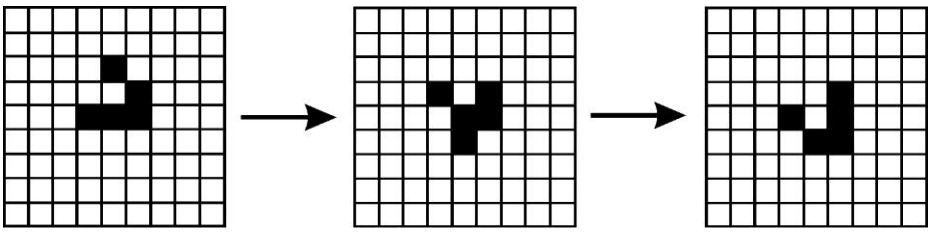

Figure 2: Game of Life is a two-dimensional cellular automaton where each cell can be in one of two possible states: alive or dead.

2.1.1 Low Processing Multi-Agent Systems

Having examined Cellular Automata, an overview of low processing MAS will now be given. (Note that in the next few sections, no musical low processing MAS will be reviewed, this will be saved for separate music-based sections later in the chapter.) Agents in Low Processing MAS are usually very simple, having no memory, no individual intelligence and no model of other agents (Parunak et al. 2007). They are sometimes called "swarm" systems, and one of the major inspirations for low processing MAS comes from the societies in nature such as ant, termite and some bee and wasp societies (Parker 1994). When many insects interact – even in this simple way – structure and "intelligent" behaviour can emerge. Examples are termite nest building, ant foraging and body collection (Beckers et al. 1995), and bees searching out a good source (Kelly 1995). It should be noted that although these *inspire* many low processing MAS systems, the actual "agent" in *real* insect swarms and colonies can be relatively advanced biological organisms – not the simple reactive ones found in artificial swarm intelligence systems. In fact, even the agents in low processing MAS could be seen as having some processing, however this processing could often be just as simply modelled and perceived as being equations of motion leading the agents to behave like particles in each others' force fields. As such, swarm MAS often have no explicit

communication system, as can be seen in Tables 2 and 3 at the end of the chapter. Furthermore, such swarm MAS interactions are usually only between nearby neighbours. We will now examine a selection of the main low processing multi-agent systems.

Boids. One of the first low processing MAS was Boids (Reynolds 1987), where a sequence of simple agents with basic interaction rules simulated the flocking of birds. The basic interaction rules were threefold: avoid getting too close to other agents ("separation"), move in the average direction of all neighbouring agents' motion ("alignment"), move towards the average position of all neighbouring agents ("cohesion"). As a result the Boids seemed to flock coherently, and navigate around obstacles – exhibiting swarm intelligence.

Particle Swarm Optimization. Swarm MAS were also applied to optimization theory through a Metaheuristic Optimization algorithm (Blum and Rolli 2003) called Particle Swarm Optimization (PSO) (Kennedy and Eberhart 1995). PSO was developed in the 90s inspired by similar elements as those which inspired Boids, but this time focusing on group searching behaviour rather than group travelling behaviour. Agents travel through the objective function space – like animals flocking in search of something (in this case the global optimum of the objective function). Each agent will tend to move in a direction towards a function of the positions of: (i) its neighbour who is in the most optimal position in the objective function space compared to other neighbours; (ii) the agent who is globally at the most optimal position found so far (i.e. not necessarily a neighbour, but any member of the population / flock); and (iii) the best objective function position the agent/particle has passed through itself in the past. Applications have included job scheduling on computer grids and data mining (Abraham et al. 2006).

BEES and Ants. The optimization search application has been extended using other swarm intelligence based metaheuristic algorithms such as the BEEs Algorithm (Pham et al. 2006) and Ant Colony Optimization (Dorigo and Glover 1999). BEES is based on the method that bees use to locate food sources from their hive. When a bee returns to the hive from a food source it performs a special "dance" to other bees, indicating the location of the food source and as a result of multiple bees doing multiple dances, a search process emerges. The BEES algorithm utilizes a simplified artificial version of the process for optimization searches. The Ant Colony system is a more widely studied approach based on how ants locate food. When ants move around looking for food they leave a pheromone trail which other ants can detect and will tend to follow. The more ants that find a particular food source, the more pheromone is laid to that source and the more ants will tend to follow the pheromone trail. Thus successful routes to food tend to be reinforced. The Ant Colony Optimization approach simulates this process with artificial pheromones and artificial ants moving around the objective function space. Ant Colony Optimization has found uses in areas included computer network load balancing (Sim and Sun 2003).

Other applications of Low Processing MAS have included cancer diagnosis (Xu et al. 2007) and to video watermarking (Lin and Liao 2008). They have also been applied to areas of the arts apart from music – for example SwarmArt (Jacob et al. 2007), a way of creating interactive visual art installations; and as a methodology for generating swarm imagery to be included in traditionally constructed arts (von Mammen et al. 2008).

To conclude this section on Low Processing MAS, a common property will be highlighted – low processing MAS are often homogeneous MAS; i.e. MAS systems in which all agents are initially indistinguishable by feature or ability. MAS where agents are *not* the same initially are called Heterogeneous MAS. This property is of interest as

there is an approximate correlation between level of MAS heterogeneity and processing/complexity in the literature. It can be seen in Tables 2 and 3 at the end of this chapter that the vast majority of described Low Processing MAS (5 out of 6) are homogeneous, a slightly lower proportion (9 of 11) of the described Intermediate MAS are homogeneous and none of the 12 Heavy Duty MAS are homogeneous. One reason for this is that the more complex agents can be, the more they can work within complex groups (i.e. have a higher social intelligence for dealing with heterogeneity).

2.1.2 "Heavy-Duty" Multi-Agent Systems

Having examined low processing MAS, some examples of heavy duty MAS which utilize more traditional AI techniques, or require heavier processing, will now be examined. It has already been noted that these systems tend to be heterogeneous. Another common thread running through much heavy duty MAS research is that there is some kind of control and planning hierarchy in the system – with the agents further up the hierarchy having a wider view than, and more control over, the agents lower in the hierarchy. For heavy duty MAS, 9 of the 12 rows described across Tables 2 and 3 have some form of initial control/planning hierarchy (whereas the proportions for Intermediate are 1 out of 11, and for lower processing none of the described systems have an explicit hierarchy). Once again, the increasing complexity of individual agents allows for more complex control structures.

Energy Management. James et al. (2006) introduce a multi-agent system for large-scale energy load management developed on the popular multi-agent platform JADE (Bellifemine et al. 1999). Agents try to optimize in real-time how much energy is generated in an electricity supply grid, and where to generate it, based on demand and on energy prices. The system is being experimented with on a physical test bed consisting of two "energy receivers": office-type artificial rooms, an office building

with a heating, ventilation and air-conditioning system; and a set of energy "producers": a 30kW turbine, a wind-turbine, and solar energy array. The system is heterogeneous with some heavy duty agents; agents requiring more processing power are run on PCs, whereas the energy receivers are controlled by agents running on PDAs. The more intensive agent types include: an agent dedicated to forecasting and gathering energy prices, an agent which manages the operation of multiple energy receivers and producers, and a monitoring agent which takes snapshots of various parameters across the whole agent system.

Emergency Response. Schoenharl et al. (2006) present a multi-agent system WIPER for detecting possible emergencies over a mobile-phone network, and also coming up with ways to deal with the emergency. Real-time data is collected from mobile phone network providers. Mobile phone data can be used to estimate location, movement and call activity, it is currently used to monitor traffic activity. There are mobile agents that can be migrated to the mobile phone provider (reducing network load), including mobile agents to pre-process and anonymise data from mobile phones, and data mining and anomaly detection agents. A simulation and prediction system based on agent design allows new algorithms to be swapped in and out – including heavy duty agents. And the general processing elements are agent-based to allow them to be distributed to optimize processing load. Users interact with the system via a web-based interface to stay updated in real-time about emergency situations.

Uninhabited Air Vehicles. (Baxter et al. 2007) UAVs have become better known in the last few years as a result of their use in recent armed conflicts. Normally a single UAV needs multiple people to control it. This multi-agent system is designed to allow a single user to control multiple UAVs. The heterogeneous system consists of multiple agent types which are hierarchically in control in approximately the following order: User,

Group, Search Specialist Planning, Attack Specialist Planning and a UAV agent for each UAV. The user agent is given an overview of the required task and controls group agents which are designed to execute and plan team tasks; though they may utilize specialist planning agents. There are five group-based behaviours including search for a target, attack a target and fly a route in formation. UAV agents are controlled in teams by the group agents and have five behaviours such as releasing a weapon, or taking a picture. The agents communicate using XML (Bray et al. 2000) and utilize the Belief/Desire/Intention method (Rao 1995) found in many heavy-duty multi-agent systems.

2.1.3 Intermediate Multi-Agent Systems

When working within an MAS paradigm not inspired by the collective behaviour of insects/flocks, researchers will tend to consider a wider-range of possible symbolic manipulation and storage, changing the potential areas and methodologies of application. In higher processing/Heavy Duty MAS, individual agents will have significant artificial intelligence. Multi-agent systems which are the agents do not individually have significant AI processing, but where agents are not acting like particles, insects or purely reactive entities, are labelled here as Intermediate MAS (Parunak et al. 2006). Criteria for Intermediate MAS can only be provided in a semi-fuzzy way, but are as follows, based on the literature reviewed in this thesis:

- No machine learning or AI in individual agents; i.e. no expert systems, neural networks, genetic algorithms, inductive logic, clustering, bayesian networks, reinforcement learning, symbolic deduction/reasoning/ problem solving, or planning.

- They have simple rules which can be adaptive.
- They do not utilize a fixed input output link between agents, e.g. their communication content is not fixed, and the agent they communicate with is not fixed.
- They change behaviour based on experience.
- Their memory changes and often expands through experience.

Common characteristics also found in Intermediate MAS include:

- Homogeneous (7 out of 9 in Tables 2 and 3) – agents in Intermediate systems are often the same (unlike Heterogeneous systems where agent's initial functionality can differ).
- Flat initial hierarchy (9 out of 9 in Tables 2 and 3) – agents in Intermediate systems commonly do not start with any one or group of agents having control over others.

It is helpful to draw a border somewhere – even if a semi-fuzzy one – due to the fact that researching with a swarm-based motivation will move work in a different direction to that found in intermediate MAS. Four examples of intermediate non-musical MAS will now be examined.

Portfolio Selection. Parkes and Huberman (2001) introduce an MAS approach to selecting stock portfolios. The usual approach is based on investors choosing a risk level they're comfortable with, and then using optimizations to find a set of stocks (a portfolio) with the maximum predicted profit for that level of risk. In the MAS approach

the agents each try out different portfolios in an artificial stock market and share with each other some information about how well their portfolios do; then an agent updates its portfolio based on the information it gets from other agents about their performances. The agents use a common method from MAS, called a "blackboard" to communicate – a data area that all agents can see and post to. When agents have finished interacting, a final portfolio is constructed based on the portfolios of the agents and the returns of the various stocks. The system was tested against FTSE 100 and Nikkei 225 (Valdez 2006) (which represent stock markets in London and Tokyo), and the results for the multi-agent approach outperformed against the market and portfolios constructed using the standard risk/return optimization systems. (The practical success of such a system in investing however will also depend on the accurate estimation of risks and returns of stocks, and the implementation of the trading decision.)

Data Mining. Di Fatta and Fortino (2007) introduces a multi-agent system for distributed data mining. The system specifically avoids using a managing agent – which is seen as a bottle-neck - and attempts to do the job in a purely peer-to-peer distributed approach. During the search process search agents work in parallel and communicate asynchronously, dynamically balancing the task load across the team. The system is tested for searching through the results of automated drug screening – with hundreds of thousands of potential molecules and their combinations. The resulting system shows that peer-to-peer approach works successfully compared to the approach using a centrally managing agent.

Collaborative Interface Agents. Lashkari et al. (1994) introduce the concept of Collaborative Interface Agents for email clients. While the user is manually using the email client, the agent attached to the client tries to analyse what actions a user takes with different types of received emails. For example do they delete them, store them in

a certain folder or forward them? So when an agent's client receives an email for its user, it checks to see if it can make a good prediction of an action to perform based on the nature of the email (e.g. the subject line), and suggest the action to the user to take; for example – move to Junk folder or Flag for Action. If an email agent does not have enough past experience to make a confident prediction then it can ask other more experienced email agents (on other users' email clients) for advice via the internet. These other agents can tell the first agent what action they would take (i.e. what their user would probably do). If the first user agrees then the local agent makes a note of which other agent it received the advice from and ups it view of the agent's usefulness. So as time goes on all agents will develop a view of which other agents are most useful to them (i.e. which other agent's users have similar requirements to their users); thus enabling them to pick which advice to take, which agents to trust, and to allow automation to develop. The issue of trust and agent popularity is significantly utilized in MAS – for example many MAS trust systems are reviewed by Ramchurn et al. (2004). In such systems agents keep some kind of records of interactions with other agents, and use it to decide on future interactions. These records create a form of social network (Sabater and Sierra 2002). It will be seen that the thesis system IPCS utilizes a form of past interaction measure between agents to encourage an emergent hierarchy to develop, whereby agents take on different roles in the composition and performance process.

Colour Grounding. There is an intimate relationship between multi-agent systems and multi-robot systems – in fact multi-robot systems could be described as "embodied" multi-agent systems. Steels and Belpaeme (2005) utilize intermediate multi-agent systems in an attempt to design robots which when placed in a new and open-ended environment (they give the example of a distant planet) are able to develop new categories which they can use for sensor input, processing, recognition and communication. So these would need to be shared categories amongst the agents and

which are not explicitly programmed. The example they choose to develop is for colour. The agents' artificial environment which they move in contains many different colours (mixes of Red, Green and Blue at different levels). When an agent experiences a colour it starts to self-organise a category for that colour, using a winner-take-all Radial Basis Function network (Bugmann 1998). It also attempts to assign a linguistic label to the category using a similar winner-take-all approach. As well as perceiving and learning colours, the agents play a "guessing game" with each other – an agent communicates one of its learned language symbols to a second agent. The second agent attempts to "guess" the colour the symbol refers to and communicates that colour back to the agent by "pointing to it" in the artificial environment. Based on the success of iterations of this guessing game, agents update their internal colour category and language models. Experiments are done with various learning mechanisms in this process, and the core result it is found that agents develop emergent common colour categories, despite a lack of "telepathy" or pre-design.

2.2 Computer-aided Composition

Having examined multi-agent systems in general, we will now look at applying these to the field of computer-aided composition. Computer-aided composition (CAC) has not been as clearly delineated as MAS. However there have been a number of reviews of such systems: by chronology (Hiller 1981; Burns 1994), by methodology, or by dimensions. Examples of the surveys that use the second two approaches are shown in Table 1.

It can be seen from Table 1 that a variety of methodologies have been applied to computer-aided composition. It can also be seen that some surveys have focused on methodologies, and some on functional dimensions. For our survey of CAC we will

focus partially on a functional perspective but mostly on methodologies, using four dimensions most appropriate to this thesis:

1. Swarm MAS for CAC
2. Heavy Duty MAS for CAC
3. Intermediate MAS for CAC
4. Affective-based CAC systems

The reason for the addition of the fourth item is that the core multi-agent mechanism utilized in IPCS – the thesis system – is based on affective (emotional) communication.

2.2.1 Low Processing MAS for CAC

We will now look at a significant proportion of the Low Processing MAS for CAC. Such systems utilize the emergent self-organisation of swarm intelligence to generate music.

Swarm Music. Blackwell and Bentley (2002) and Blackwell (2007) discuss the use of swarms in music. The focus is not so much on pre-organised music, where musical events are already largely defined in time, loudness and pitch. They instead discusses improvisation from experience and from analysis – arguing that musicians respond to each other in improvisation using local-based rules – just as insect swarms do. And thus self-organisation would be expected – as found in swarms – and hence improvisations have a structure despite the lack of conductor or followed score. They go further and argues that the rules are directly analogous to the types of rules used in Low Processing

MAS. As a result Blackwell/Bentley's basic swarm music approach is based on the 3 elements of the Low Processing BOIDS-type MAS discussed earlier: cohesion, separation and alignment.

Reference	**Dimensions/Methodologies**
(Papadopoulos and Wiggins 1999)	Mathematical Models, Knowledge-based Systems, Grammars, Evolutionary Methods, Systems which Learn, Hybrid Systems
(Jarvelainen 2000)	Stochastic Music; Cellular Automata; Flow control and grammars, etc; Fractals; Genetic Algorithms
(Miranda 2002)	Probabilities, Grammar and Automata; Iterative Algorithms: chaos and fractals; Neural Computation and Music; Evolutionary Music
(Ariza 2005)	Scale, Process model, Idiom-Affinity, Extensibility, Event Production, Sound Source, User Environment
(Wooler et al. 2005)	Linguistic/Structural, Interactive/Behavioural, Creative/Procedural, Biological/Emergent

Table 1: Survey of Surveys

The system is turned into a live algorithm which allows it to improvise interactively with human improvisers. The behaviour of the human improviser will place "attractors" in the space, which the swarms will tend to congregate towards. Swarms can actually move in 7 dimensions where the dimensions represent: note loudness, time between notes, pitch, time duration of note, number of simultaneous notes in phrase, number of ascending or descending pitches in a phrase, and similarity between successive phrases.

A number of additional mechanisms in the swarms have been introduced, allowing for more complex behaviours and music. In relation to this it should be noted that the Swarm Music System does not address the issue of larger scale musical structure; in fact Blackwell discusses how the adding of some deliberative mechanisms to the low processing MAS (i.e. making them intermediate MAS) may enable a more structured self-organised music to emerge.

Ant Colony System. As has already been mentioned, Ant Colony Search is an algorithm which has been developed using low processing multi-agent systems. Clair et al. (2008) utilize an artificial ant colony to generate music. Ants wander through graphs in multiple dimensions, where vertices trigger pitches and durations. As the ants wander the graphs, they leave an artificial pheromone trail, thus increasing the probability of other ants following that path. This leads to rhythmic and pitch cycles which create repeating sequences that can be musical patterns. Multiple ants moving can create harmonies. There are also "silent" ants, whose only function is to enforce structure through pheromone path generation. The main motivation for the system is actually its use as a live musical instrument. The user can play the ants as an instrument by either influencing adding notes they play to the graphs, or by "becoming an ant" themselves and creating pheromone paths around the graphs. Unlike the Swarm Music system above – which is compared to the process of musical improvisation, the Ant Colony System has no clear relationship to musical processes.

Swarm Orchestra. Bisig and Neukom (2008) use a swarm simulation library for BOIDS called ISO Flock and for sound called ISO Synth. ISO Synth is also able to deal with sound spacialisation and hence the swarms in their orchestra are played on a 20 speaker system with three dimensional sound projection, where the sound positioning can be

decided by swarming. They investigate a number of mappings from swarms to sound, including:

1. Additive Synthesis – mapping agent co-ordinates to a sine wave frequency and oscillator amplitude

2. Granular Synthesis (Miranda 1995) – agent velocity amplitude maps to grain playback rate, agent amplitude direction maps to where the grain is added in the final sound, and number of agent neighbours maps to amplitude of the grain.

3. Patch Construction – agent movements trigger synthesizer patch construction rules

4. Physical Representation – the agents are constrained by artificial physics, for example attached to spring networks, and the springs are also excited by white noise. The springs then generate sine waves which create a spectrum for a sound source.

The authors state that the main purpose of their work is to highlight how much untapped potential there still is in swarm music. However the more complex the mappings are made, the less strong the suggestive links between swarm music systems and human improvisation (the original motivation) become.

The Society of Musical Agents. Beyls' (2007) system is designed to create polyphonic music in real-time based on clusters which emerge in the agents' movement space. Agents have "personalities" defined by their repertoires of intervals, durations and velocities that they construct music from. Agents also have properties which define the distances at which they can communicate and cluster, and a flag which influences whether they move away from other agents or attempt to communicate with them. They also an activation level that effects the musical influence of the agents in clusters, and

an energy level which defines how they move. To generate music, the agents are separated into neighbourhood clusters depending on their locations and feature values. The highest energy cluster is selected and music is generated from the pattern using various algorithms. The paper focuses more on the complex but coherent patterns which emerge as a result agent clustering. These patterns – no matter how complex - do not lead to the paper addressing the issue of larger scale musical structure. (Also they continue to move swarm music away from its original inspiration of human improviser groups.)

2.2.2 Heavy Duty MAS for CAC

As the complexity of individual agents increases there is a tendency – in musical MAS systems – to lead to agents being design to specialize in different parts of the musical function; for example, conductors and controller agents, harmony and melody agents, etc. This tendency will be clear as a significant proportion of heavy duty MAS systems for CAC are now reviewed.

MMAS (Musical MAS). MMAS (Wulfhost et al. 2003a; Wulfhost et al. 2003b) is designed to have agents play along with each other in real time producing compatible harmonies and rhythms. One or more of the agents can optionally be a live human performer (via MIDI control). Agents have a beat detection sensor and beat inference system to help them synchronize their playing. They also attempt to detect the current harmonic context to help them select their next note(s), based on a static transition table constructed from 50 popular songs. The agent's musical knowledge is encoded with fuzzy logic rules such as "if accelerando is molto then dynamic is forte". They use a blackboard system to communicate. Tests were run in real-time with a multi-agent system including, and not including a human. The system was found to be able to

synchronise well with the agents and the human player. There are plans to implement affective modes and playing for the agents but the results of these implementations have not been reported. The system's focus on Fuzzy Logic rules means that interaction elements are not the core part of the creative process; they are used more to ensure compatibility between multiple sound generators.

Musical Agents. Each agent in Fonseka's (2000) system can run a script system flexible enough to perform many algorithmic composition functions. These scripts also define how the agents interact with each other. The agents are designed to be distributed over a computer network and to take into account network latency times in performing in real-time with each other. The performance format is MIDI. The agents utilize a central communication node. An example is given where a script is written for agents on multiple PCs to try to simulate the composition Paragraph 7 by *Cardew* (1967). In this all agents choose random notes to perform for a fixed period, then at the end of that period must pick a note their "neighbour" is currently performing for them to perform in the next period. This was found by Cardew to have complex emergent properties, and a similar behaviour was reported during agent testing. Although able to simulate an emergent process, this Musical Agents system is essentially series of algorithmic composition script intepreters which can synchronise over a network - the issue of multi-agent interaction for creativity is not addressed in depth.

Andante. Ueda and Kon (2003) is a multi-agent framework distributed across multiple computers – like Fonseka's Musical Agents, however it is not real time and agents in Andante can also migrate across the network. It is described as a framework for a mobile multi-agent musical system. A computer in the network can also contain multiple instances of an Andante agent. All agents utilize a common metronome. An example MAS is given made up of what are called "NoiseAgents". These agents

generate pitches whose features can be defined by types of noise; for example an agent's pitches could be generated by white noise, its durations by pink noise and its loudness by Brownian noise. Later work (Ueda and Kon 2004) introduces a scripting system which allows users to define agent parameters in time – e.g. for NoiseAgents what different types of noise are used, and what pitch ranges, etc. There do not seem to have been any investigations in how the agents could interact; probably because the paper is solely a framework introduction. Future work is planned to examine inserting real musicians into the process. The lack of reported implementation scenarios for the system is its main weakness.

VirtuaLatin. The VirtuaLatin system (Murray-Rust et al 2005) was not fully implemented as a multi-agent system. A single agent was produced for real-time rhythm accompaniment. The agent listens to the music it is to accompany, and develops an in-depth structural representation of music so it can play along. The planned MAS includes harmonic, rhythmic and pattern analysis, as well as a central conductor. An experiment was reported on the single agent system playing virtual timbalero, and listeners could not tell the difference between a human recording and the agent generated rhythms. However because of the lack of reported multi-agent implementations it is hard to evaluate this system as a musical MAS.

MAMA (Musical Acts – Music Agents). The VirtuaLatin team were later involved in this project (Murray-Rust and Smaill 2005). MAMA is a real-time multi-agent system based on a theory called of Musical Acts. This is inspired by Speech Acts Theory in language studies, which examines language that not only provides information, but causes a change in the environment - i.e. "acts" on the environment. This can be compared to musical improvisers, whose music not only creates the overall improvisation, but also provides communicative cues to each participant in real time

about what each musicians intentions are. Many MAS which act on shared objects have a separate protocol for acting on the objects to communicating with each other – e.g. an MAS which builds visual data will usually have a separate communication protocol to the data it is working on. This has relevance to the thesis system IPCS, which does not have any communication protocol separate to the object it acts on. In IPCS the data and the protocol are the same – i.e. both are the music being composed. However IPCS differs to MAMA in two key ways: MAMA is real-time, and uses higher levels of processing to allow an agent to infer information about others agents and their music, and to allow agents to play together musically. Furthermore, in MAMA, the user provides agents with a high level structure for the piece, a set of musical fragments with rules for how they can be used, and style constraints. Based on this MAMA can generate a performance. One example given is the Terry Riley piece In C, a piece in which every performance is actually a form of "re-composition" (Nyman 1974). MAMA also allows humans to be agents within the system. Although MAMA can produce music with a higher level structure, this is provided by the user, not generated my MAMA itself.

Kinetic Engine. Eigenfeldt (2009) describes a real-time performance multi-agent system for generating rhythms. The user sets parameters for the type of rhythms they want – how much syncopation and density of rhythm – and a central selector agent chooses material from the generator agents whose live rhythms best suit the parameters (which can be adjusted in real time during the performance). During the performance, the agents are continually generating new populations of rhythms using the genetic algorithms. One of the key problems with genetic algorithm generated music has been the "fitness bottleneck" – the evaluation of music by members of the population. This is addressed here by the user providing initial rhythmic source material in MIDI format – this material is analysed for musical features. And these features become the desirable

features within the genetic population algorithm. Eigenfeldt argues that the genetic crossover operator in genetic algorithms does not represent a useful musical transformation in this context. Therefore members of the GA population reproduce "asexually". A particular GA population member generates a first order Markov chain (Roads 1996) to represent its rhythmic transitions, and children are generated from this markov model. For clarity it should be noted, the agents themselves do not reproduce – each agent holds within it a GA population of tunes which is reproducing. It is reported that at times, no agents can supply the selector agent with rhythms that fulfil the user's real-time requirements for level of syncopation and density of rhythm; in these cases the agents' parameters will be shifted to attempt to match the GA output more closely with the user's requirements. From a composition point of view, Kinetic Engine is only generating rhythms not pitches. Furthermore its use of MAS methods is largely for parallelism of musical lines, its musical intelligence comes from Genetic Algorithms and Markov Chains.

CinBalada. Sampaio et al. (2008) introduce a system for polyphonic rhythm composition. Each agent is given a rhythmic role of type "base", "complementary base", "solo" or "fill". Before each bar is composed, agents with the same rhythmic role negotiate together about which will fulfil the role in the next bar – agents submit proposed patterns to each other and they are evaluated in competition. The evaluation includes how well the pattern for an agent evolves from the previous bar, and how well the pattern fits in with other patterns which have already been selected. A series of rhythmic compatibility measures was developed over a three year project in collaboration with musical experts. Agents base their composed patterns on a stored database of Brazilian percussive performance. Tests were done with listeners comparing CinBalada generated patterns with patterns generated randomly or by simpler criteria, and listeners showed some preference for CinBalada patterns. However, because of the

bar-to-bar compatibility approach, CinBalada does not generate any higher level musical structure.

AALIVE. Spicer et al. (2003) present a multi-agent system in which each agent has two internal oscillators – one for pitch and one for duration – synchronized to the tempo of the music. A human gives higher level musical goals, and the system attempts to implement them. Whenever an agent plays a note, the current state of the duration oscillator defines the note length, and the pitch oscillator the pitch. There are different classes of agents, for example: bass player, kick and snare drum player and arpeggio player. Some agents are designed to work in sub-groups where there is a lead agent and followers; the followers may have certain rules such as "follow the lead pitch an interval below", thus creating harmonies; or to re-align their rhythm pattern to creating interlocking rhythms. The human controller defines the starting duration and pitches for agents, and defines targets for average pitch and average duration of the MAS.

NetNeg. This is a system for composing First Species Counterpoint (Goldman et al. 1999). It makes use of an artificial neural network (ANN) and two agents. The ANN generates melodies for the other two agents, and these two agents then negotiate to try to produce notes which obey a selection of standard counterpoint rules. The ANN is trained to evaluate single line melodies based on lines from 16th century counterpoint. Suppose one of the two agents sends a certain note N to the ANN agent, then the ANN agent will return what it considers the best next note for the melody. It actually returns a vector of 13 numbers where each vector element represents a pitch, and the highest magnitude vector element will be at the most melodically "aesthetically pleasing" following note. The next most "pleasing" note will have the second highest magnitude for its vector element, and so forth. At each time step the two agents send a note each to the ANN, and the ANN sends a vector to each of the two agents. These two agents then

negotiate, using a counterpoint expert system (Waterman 1986) as their basis, to try to come up with a next pair of notes which are an optimal balance of counterpoint rules and melodic movement. NetNeg's biggest limitation for original composition is that it focuses on one type of music only – first species counterpoint.

Inmamusys. Delgado et al. (2008) present a multi-agent system that has three types of agents: manager, composing, and voice. The user specifies their compositional requirements to a manager agent using labels such as "worry", "happiness", and "chaos". A number of composing agents then compete to fulfil these requirements – with labels liked "muzak", "dark", "scales", and "random". Once the manager has selected a composing agent, that composing agent will execute a number of voice agents, of which there are four types: melody, accompaniment, drums and harmony. These agents then combine their outputs to make the final composition. The agents are based on an expert system approach to musical generation. In listening tests, four compositions were generated, one for "Happiness" and "Chaos", and two for "Worry". The tests gave a strong indication that there was a reasonable mapping between the input labels and the perception of the music by the listeners. IPCS provides a similar method for specifying compositions and for listening tests. It generates music with different estimated emotional content; and in the results chapter tests will be reported on how some listeners perceive the music. IPCS also attempts to utilize a hierarchical approach, but has - unlike Inmamusys - no initial hierarchy. The hierarchy is looser and emergent rather than fixed. Another difference is that Inanmusys voice agents work in parallel on the same tune, and also have more significant AI processing. It is hard to see what advantage Inmamusys gains from being a multi-agent rather than a modular expert system – hence its testing of the use of MAS interaction for composition is limited.

2.2.3 Intermediate MAS for CAC

Having reviewed low processing and heavy duty MAS for music, most of the Intermediate MAS for Computer-aided Composition will now be reviewed. Most of them have either not been implemented, have been designed for studying musical culture (rather than for composing), or solely generate rhythm rather than musical lines. A significant proportion of this section will involve comparing the systems to the thesis system IPCS, because the research reviewed here is the most closely related to IPCS.

Computer-Aided Multi-Agent Music Composition. The Dahltstedt and McBurney (2006) system seems to be more a proposal that has never been implemented. Their aim is to using MAS to better understand music and the processes of music composition, as well as to utilize them to compose with and to use in installations. The approach is to use agents which have different explicit goals that represent different parts of the process of music composition. An example is given of an agent whose goal is to reduce sound object density if the population of the system's "sound landscape" becomes too "cluttered"; another is given of an agent who does the opposite. Both agents would take into account the musical context while doing this. The researchers explicitly intend to utilise emergence (Bentley 2005) to generate interesting music. This is a similarity with IPCS, though key differences are: that agents here act on a common musical landscape together, whereas IPCS agents each have their own repertoires which can develop in parallel; and the agents in IPCS do not have explicit and distinct goals. Obviously a key limitation of the Computer-Aided Multi-Agent Music Composition is its lack of implementation - and hence lack of contribution to investigating MAS interaction in composition.

Cultural Evolution. Miranda's (2003) system generates musical motifs in a way designed to study the evolution of culture. In this case the agents use a two-way imitation procedure to bond socially. Agents can store a repertoire of tunes and have a

basic biological models of an adaptive voice box and auditory system. Agents pick other agents to interact with randomly. When two agents A and B interact the following process occurs: if agent A has tunes in its repertoire it picks one randomly and sings it, if not then it sings a random tune. These tunes are three notes long and do not grow in length. Agent B compares the tune from A to its own repertoire and if it finds one similar enough, plays it back to agent B as an attempted imitation. Then agent B makes a judgement about how good the imitation is. If it is satisfied with the imitation it makes a "re-assuring" noise back to agent A, otherwise it does not. Based on the success of the imitation Agents A and B update their repertoires and their voice box settings to try and improve their chances of socially bonding in later interactions – e.g. by deleting or re-enforcing tunes, or making random deviations to their voice box parameters. The aim of the system is to see how the repertoire is generated and affected under such social pressures. As a result of the social bonding interactions a community repertoire begins to emerge. An example run is reported of 5 agents and 5000 interactions; and it was found that the agents evolved a shared repertoire of tunes as an emergent property of the social pressure. From an algorithmic composition point of view, the system is clearly limited in the use of only three note tunes, and its use of randomness.

Sequence Evolution. Gong et al. (2005) produced a simple music composing system with a similar purpose to Miranda's (2003) - investigating the emergence of musical culture. The agents start with a set of random motifs, together with different agents being equipped with distinct but very simple aesthetic evaluation functions (for rhythm, pitch, etc.). An agent plays its tune to another agent and if the second agent finds the tune unpleasant, it modifies it (based on its musical evaluation), and plays it back to the first agent. If the first agent thinks the modified tune is better than its original, it deletes its original and stores the modified version. As agents interact this leads to "more pleasant" motifs emerging. Also, using an interaction-history measure, the social link

between first and second agent is strengthened so that they are more likely to interact in the future. However if the first agent does not prefer the modified tune to its own version, it discards it and the link between the two agents is not strengthened. It was found that in the emergent social network the agents tended to cluster according to their aesthetic preference function. This system has a couple of similarities to IPCS: it utilizes MAS social network/trust techniques to decide who interacts with whom, and in each interaction agents vary their repertoire based on their opinion of the other agent's repertoire. The key differences between this system and IPCS is that IPCS agents have no explicit evaluative musical intelligence, and the social network in IPCS is used to generate music structure within an agent's repertoire not to experiment with the clustering of agents according to their repertoires. Also the number of notes in Gong et al's system is very limited and tunes cannot grow in size – neither this system nor Miranda's above are practical composition systems at this stage.

A-Rhythm. Martins and Miranda's (2007) system sets out to examine the application of multi-agent systems to algorithmic composition, but has not yet fulfilled that goal. Current papers focus on, like Miranda's (2003) and Gong et al.'s (2005) systems, investigating the emergence of social clusters, and are solely based on rhythmic repertoire. A-Rhythm has some similarities to IPCS – the agents communicate one at a time serially and their musical content grows longer. However A-Rhythm lacks the ability to compose pitches, and focuses on rhythms. Also the similarity measures are more directly based on the music, rather than affective content of the music. Finally A-Rhythm uses measures for the "popularity" of rhythms in an agent's repertoire, but not for the "popularity" of agents. Agents in this system can transform their repertoires based on interaction – using certain rhythmic transformation rules (rather than ICPS's affective-based transformations). In the paper a number of experiments are done based on different interaction approaches, and the resulting population and repertoire

dynamics are examined, showing the potential for the emergence of structured rhythmic repertoires.

2.2.4 MAS and Musical Structure

It is worth highlighting at this point that none of the MAS CAC systems surveyed above were designed to deal with higher level musical structure – e.g. how to order themes and sections of a piece of music. They focus on the note to note, or phrase to phrase level. There are in the wider CAC field systems which address these higher level issues of musical structure. However Collins (2009) observes that algorithmic generation of musical form is one of the least advanced elements of computer-aided composition systems. In the next chapter it will be shown that an Intermediate MAS can be designed so as to incorporate a proposal for a emergent structural composition system arising from the social behaviour of the agents in the system.

2.2.5 MAS Summary

Table 2 gives an overview of the general (non-musical) Multi-Agent Systems (MAS) reviewed in this chapter, together with key descriptive properties. The whole field was not detailed but these 11 non-musical systems gave an overview of MAS in general, their common features, and introduced some of the key elements relevant to this thesis - e.g. intermediate vs. low processing MAS, agent interaction networks, etc.

Table 3 gives an overview of generative musical systems reviewed– both MAS-based and non-MAS-based. In terms of computer-aided composition, the review focus was on MAS-based computer-aided composition, though affective-based computer-aided composition will also be surveyed next because of its relevance to the thesis. The 18 musical MAS reviewed are, as far as is known, the majority of multi-agent composition and expressive performance systems published. The 3 non-MAS musical systems to be

reviewed in detail are affective-based composition and cover a significant proportion of the published systems. (Affective-based expressive performance is also discussed in the expressive performance section of the review.)

The Fields in table 2 are:

- "Complexity" – is the MAS low processing, intermediate or heavy duty?
- "Homog / Het" – is the MAS homogeneous or heterogeneous (i.e. do agents all start out the same, or are some different)?
- "Comm" – do the agents communicate, and if so do they do it synchronously or asynchronously (i.e. do they take it in turns to communicate and process, or do they do it concurrently)?
- "Initial Hierarchy" – is there a hierarchy of planning/control; are some agents dependent on others? Can some agents control others?

The choice of properties is both because they clarify the context for the thesis system IPCS, and are also some of the key defining features of any MAS. Some core elements of IPCS are that it is Intermediate complexity, homogeneous, that it composes music using direct communication, and that there is no central controlling agent or hierarchy. Hence the fields chosen for Table 2 for non-musical MAS. Table 3 goes into more detail about some of the secondary properties of IPCS and properties which are more relevant in the music field.

The additional fields in Table 3 (Musical MAS) are:

- “Num Tunes” – Does the system generate multiple tunes during its processing, or a single tune? (Multiple tunes does not mean polyphony but unconnected multiple tunes.)
- “Real-time” – is the music generated in real-time?
- “Size” – what is the order of the number of agents in the system?
- “Model / Func” – is the system designed to model some element of music, or as a computer-aided composition system?

IPCS is a non-real-time system which works with a small to medium number of agents (i.e. not hundreds of agents), it generates multiple tunes in parallel, and it is focused on computer-aided composition not on modelling the composition process or musical culture.

2.2.6 Affective-based CAC systems

One area of computer-aided composition which has received more attention recently is affectively-based computer-aided composition, for example the Inmamusys system described earlier (Delgado et al. 2008). There have been a number of questionnaire studies done which support the argument that music communicates emotions (Minassian et al. 2003; Lindstrom et al. 2003; Juslin and Laukka 2004). However a key point that needs to be clarified is the difference between induced emotion and perceived emotion (Juslin 2005). For example a listener may enjoy a piece of music like the famous Adagio for Strings by *Barber* (1938) which most people would describe as a “sad” piece of music. However, because they enjoy it, the induced emotion must be pleasurable (their valence is increased by listening), but the perceived emotion is sadness. Although there are some differences between perceived and felt emotions

(Evans and Schubert 2006), perceived and felt emotions are highly correlated (Bigand et al. 2005). IPCS, and the systems reviewed below, focus on the generation of music which causes certain emotional perceptions in the listener, rather than directly inducing an emotional state.

System	**Complexity**	**Homog / Het**	**Comm**	**Initial Hierarchy**
Boids (Reynolds 1987)	Low	Homog	No (see p29)	Flat
PSO (Kennedy and Eberhart 1995)	Low	Homog	No	Flat
BEES (Pham et al. 2006)	Low	Homog	Synch	Flat
Ant Colony (Dorigo and Glover 1999)	Low	Homog	No	Flat
Energy Management (James et al. 2006)	Heavy	Heterog	No	Hierarchy
Emergency Response (Schoenharl et al. 2006)	Heavy	Heterog	Asynch	Hierarchy
UAV (Baxter et al. 2007)	Heavy	Heterog	No	Hierarchy
Portfolio Selection (Parkes and Huberman 2001)	Int	Homog	Sync	Flat
Data Mining (Di Fatta and Fortino 2007)	Int	Homog	ASynch	Flat
Collab Interf. (Lashkari et al. 1994)	Int	Homog	Sync	Flat
Colour Grounding (Steels and Belpaeme 2005)	Int	Homog	Synch	Flat

Table 2: Overview of non-musical MAS examples reviewed

System	Complexity	Homog / Het	Comm	NumTunes	Initial Hierarchy	Real time	Size	Model / Func
Swarm Music (Blackwell and Bentley 2002)	Low	Het	No	1	Flat	Y	21	F
Ant Colony Music (Clair et al. 2008)	Low	Homog	No	1	Flat	Y		F
Swarm Orchestra (Bisig and Neukom 2008)	Low	Homog	No	1	Flat	Y		F
Society of Music Agents (Beyls 2007)	Low	Homog	Sync	1	Flat	N		F
MMAS (Wulfhost et al. 2003a)	Heavy	Het	ASync	1	Flat	Y	8	F
Musical Agents (Fonseka 2000)	Heavy	Het	Async	1	Flat	Y		F
Andante (Ueda and Kon 2003)	Heavy	Het	Async	1	Flat	Y		F
VirtuaLatin (Murray-Rust et al. 2005)	Heavy	Het	Sync	1	Hierarchy	N	1	F
MAMA (Murray-Rust and Smaill 2005)	Heavy	Het	ASync	1	Hierarchy	Y		F
Kinetic Engine (Eigenfeldt 2009)	Heavy	Het	ASync	1	Hierarchy	Y		F
CinBalada (Sampaio et al. 2008)	Heavy	Het	ASync	1	Flat	N		F
AALIVE (Spicer et al. 2003)	Heavy	Het	ASync	1	Hierarchy	Y		F
NetNeg (Goldman et al. 1999)	Heavy	Het	ASync	1	Hierarchy	N	3	F
Inmamusys (Delgado et al. 2008)	Heavy	Het	Sync	1	Hierarchy	N	9	F
CAC Multi-Agent (Dahltstedt and McBurney 2006)	Int	Het	ASync	1	Flat	N	6	F
Critic Culture (Gong et al. 2005)	Int	Hom	Sync	Multi	Flat	N		M
Miranda Culture (Miranda 2003)	Int	Hom	Sync	Multi	Flat	N	5	M
Artificial Musical Society (Martins and Miranda 2007)	Int	Het	Sync	1	Flat	Y		F
IPCS (thesis system)	Int	Hom	ASync	Multi	Flat	N	10	F

Table 3: Overview of Musical MAS reviewed

Another common theme running through some of the affective-based systems below is their method of affective specification – the Valence / Arousal approach. This is not the only possible approach to affective specification. It is just one of a variety of approaches which can be broadly divided in the Dimensional type and the Category type (Zentner et al. 2008). Category approaches range from basic emotion definitions – which assumes that some emotions are more fundamental than others and attempts to list these; to the more everyday emotion label systems – which do not attempt to categorize based on an emotion hierarchy. A version of the second type of category approach is used in Director Musices expressive performance approach (Sundberg et al. 1983; Friberg et al. 2006) – where six emotion labels are utilized – such as "Happy", "Angry", etc. Juslin's (2005) analysis of musical emotions also uses everyday categories. A more recent category-based approach for emotion is the Geneva Emotion Music Scales (GEMS) approach (Zentner et al. 2008) which attempts to provide categories optimal for musical emotion. This is done by first investigating through psychological tests which sorts of emotion are most commonly expressed to people by music. Although GEMS does get users to score the category with an integer from 1 to 5, the fact it has up to 45 categories puts it more in the realm of categorical than the dimensional systems now discussed.

The Dimensional approach to specifying emotion utilizes an n-dimensional space made up of emotion "factors". Any emotion can be plotted as some combination of these factors. For example, in IPCS and in CMERS, two dimensions are used: Valence and Arousal (Lang 1995). In this model, emotions are plotted on a graph with the first dimension being how positive or negative the emotion is (Valence), and the second dimension being how intense the physical arousal of the emotion is (Arousal). This is shown in Figure 3. Just as category approaches would not claim to list all possible emotions, so dimensional approaches do not claim to be complete. It is not known if emotions can be pinpointed based on unique independent dimensions. Other

dimensional approaches include the three dimensional valence/arousal/dominance system (Oehme at al. 2007). In this case Valence and Arousal have the same meaning as in the 2D version. However in the 2D approach Fear and Anger are both low valence, high arousal. In the 3D version, Dominance differentiates emotions such as anger (high dominance) and fear (low dominance); anger can be seen as more of an active emotion, fear as more of a re-active one. In (Canazza et al. 2004) a task-specific mood-space is constructed for expressive performance using experiments and principle component analysis. In this case the dimensions are not explicit.

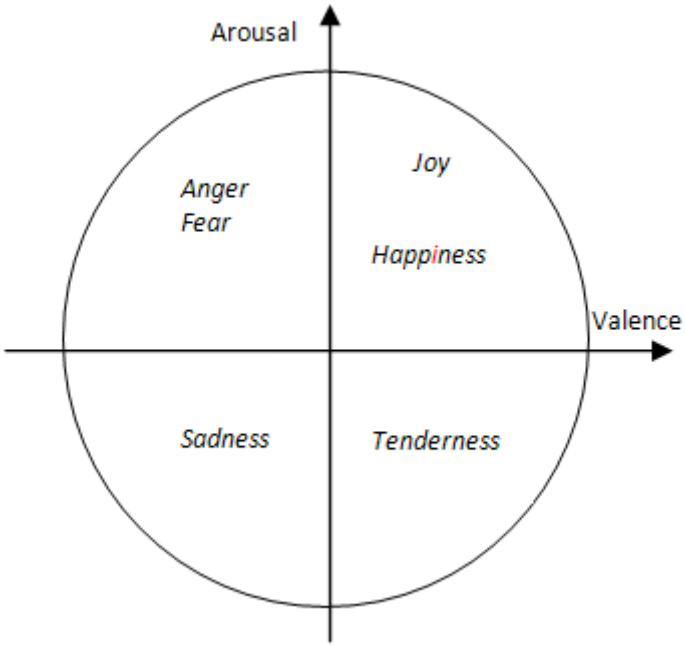

Figure 3: The Valence/Arousal Model of Emotion

One key difference between dimensional and category approaches is that dimensional approaches are much simpler to implement in continuous change systems. For example in a 6 category system like that used in Director Musices, there is no obvious way of moving continuously around the emotions – one can only switch between emotional types. In IPCS the concept of small changes to affective state is implemented. Hence the

choice of a dimensional approach seems natural. Furthermore because of the successful implementation of the 2D valence/arousal approach in other musical applications, it was this dimensional approach which was selected. So in IPCS affective states are defined by a pair of real numbers (*valence*, *arousal*). At the heart of affective-dimensional CAC systems tends to be a mapping from the desired valence / arousal of a section of music, to the resulting musical features such as key, tempo and pitch. This mapping may be a set of rules, or a linear or non-linear regression mapping. Three of these affective CAC will now be examined.

CMERS. Many of the affective-based systems are actually based around re-composition rather than composition; i.e. they focus on how to transform an already composed piece of music to give a different emotional effect – e.g. make it sadder, happier, etc. This is the case with the best known and most thoroughly tested system - the Computational Music Emotion Rule System (CMERS) (Livingstone et al. 2010; Livingstone et al. 2007). The rules for expressing emotions map valence and arousal onto such elements as modes and pitch class. These rules were developed based on the combining a large number of studies by psychologists into music and emotion. However it was found these needed to be supplemented by rules for expressive performance of the transformed music to express the emotion successfully. Hence CMERS is actually (like IPCS) an integrated composition and expressive performance. For this reason a more detailed review will be given at the end of this chapter, which focuses on combined composition / expressive performance systems. CMERS' key limitation as a composition system is that it is designed for re-composition, not for generating new material.

Knowledge-Base Approach. Oliveira and Cardoso (2009) also perform affective transformations on MIDI music, and utilizes the valence-arousal approach to affective specification. These are to be mapped on to musical features: tempo, pitch register,

musical scales, and instrumentation. A knowledge-base of musical features and emotion was developed based on musical segments with a known affective content. This knowledge-base was then used to train a generalized mapping of affective state to required music and a model was then generated based on Support Vector Machine (Shawe-Taylor and Cristianini 2000) regression. The model was tested for transforming the emotion of classical music – the current results are not as good as CMERS. One reason for this may be that this Knowledge-Base approach has the limitation that it is unable to generate expressive performances.

Affective GA Composition. Although Legaspi et al. (2007) utilise pre-composed music as its heart, it is more focused on composing new music. An affective model is learned based on score fragments manually labelled with their appropriate affective perception – this maps a desired affective state on to a set of musical features. The model is learned based on the machine learning approaches Inductive Logic Programming and Diverse Density Weighting Metric (Lloyd 1984; Maron and Lozano-Perez 1998). This is then used as a fitness function for a Genetic Algorithm – however the GA is also constrained by some basic music theory. The GA is then used to generate the basic harmonic structure, and a set of heuristics are used to generate melodies based on the harmonic structure. The system was trained with emotion label dimensions "favourable-unfavourable", "bright-dark", "happy-sad", and "heartrending-not heartrending". Listening tests were done on a series of eight bar tunes and the results obtained were considered promising, but indicated more development was needed. Once again, the system is lacking the ability to generate expressive performances.

2.2.7 MAS and CAC Summary

In the last few sections, multi-agent systems in general have been reviewed, including a focus on MAS for computer-aided composition. This was done because IPCS, the thesis system, is an MAS for computer-aided composition. There has also been an overview of affective-based computer-aided composition systems, once again an area closely related to IPCS. As well as being an MAS and affectively based, IPCS is also a system in which computer-aided composition and expressive performance are combined. The field of computer expressive performance is newer and less familiar than the field of computer-aided composition. Hence, before reviewing the handful of systems for integrated computer-aided composition and expressive performance, we will present a review and analysis of the field of computer systems for expressive performance of music.

Before moving on to this there is one final point to note. One implicit limitation of much of the CAC systems discussed relates to the subjective nature of musical performance (Ramirez et al. 2008). Ramirez et al. generate multiple performances in parallel so as to allow the user to select their subjective favourite from them. Aside from those systems designed for modelling purposes, the CAC systems reviewed all produce a single performance. (This issue will be re-visited in the Chapter 3.)

2.3 Computer Systems for Expressive Performance

This section surveys computer expressive performance, and will start by examining what expressive performance actually is.

2.3.1 Human Expressive Performance

How do humans make their performances sound so different to the so-called "perfect" performance a computer would give? In this thesis the strategies and changes which are not marked in a score but which performers apply to the music will be referred to as expressive *Performance Actions*. Two of the most common performance actions are changing the *Tempo* and the *Loudness* of the piece as it is played. These should not be confused with the tempo or loudness changes marked in the score, like accelerando or mezzo-forte, but to additional tempo and loudness changes not marked in the score. For example, a common expressive performance strategy is for the performer to slow down as they approach the end of the piece (Friberg and Sundberg 1999). Another performance action is the use of expressive *articulation* – when a performer chooses to play notes in a more staccato (short and pronounced) or legato (smooth) way. Those playing instruments with continuous tuning, for example string players, may also use expressive *intonation*, making notes slightly sharper or flatter; and such instruments also allow for expressive *vibrato*. Many instruments provide the ability to expressively change *timbre* as well.

Why do humans add these expressive performance actions when playing music? The context will be set for answering this question using a historical perspective. Pianist and musicologist Ian Pace offers up the following as a familiar historical model for the development of notation (though suggests that overall it constitutes an over-simplification) (Pace 2007):

> In the Middle Ages and to a lesser extent to the Renaissance, musical scores provided only a bare outline of the music, with much to be filled in by the performer or performers, freely improvising within conventions which were essentially communicated verbally within a region or locality. By the Baroque Era, composers began to be more specific in terms of requirements for pitch, rhythm and articulation, though it was still common for performers to apply embellishments and diminutions to the notated scores, and during the Classical Period a greater range of specificity was introduced for dynamics and accentuation. All of this reflected a gradual increase in the internationalism of music, with composers and performers travelling more widely and thus rendering the necessity for greater notational clarity as knowledge of local performance conventions could no longer be taken for granted. From Beethoven onwards, the composer took on a new role, less a servant composing to occasion at the behest of his or her feudal masters, more a freelance entrepreneur who followed his own desires, wishes and convictions, and wrote for posterity, hence bequeathing the notion of the master-work which had a more palpable autonomous existence over and above its various manifestations in performance. This required an even greater degree of notational exactitude; for example in the realms of tempo, where generic Italianate conventions were both rendered in the composer's native language and finely nuanced by qualifying clauses and adjectives. Through the course of the nineteenth century, tempo modifications were also entered more frequently into scores, and with the advent of a greater emphasis on timbre, scores gradually became more specific in terms of the indication of instrumentation. Performers phased out the processes of embellishment and ornamentation as the score came to attain more of the status of a sacred object. In the twentieth century, this process was extended much further, with the finest nuances of inflection, rubato, rhythmic modification coming to be indicated in the score. By the time of the music of Brian Ferneyhough, to take the most extreme example, all minutest details of every parameter are etched into the score, and the performer's task is simply to try and execute these as precisely as he or she can.

So in pre-20th century classical music there has been a tradition of performers making additions to a performance which were not marked in the score (though the reason Pace calls this history an oversimplification is that modern music *does* have the capacity for expressive performance, as we will discuss later).

A number of studies have been done into this pre-20th Century (specifically Baroque, Classical and Romantic) music performance. The earliest studies began with Seashore (1938), and good overviews include Palmer's (1997) and Gabrielsson's (2003). One element of these studies has been to discover what aspects of a piece of music – what *Musical Features* - are related to a performer's use of expressive performance actions.

One of these musical features expressed is the performer's *structural* interpretation of the piece (Palmer 1997). As has already been noted, a piece of music has a number of levels of meaning – a hierarchy. Notes make up motifs, motifs make up phrases, phrases make up sections, sections make up a piece (in more continuous instruments there are intranote elements as well). Each element - note, motif, etc. - plays a role in other higher elements. Human performers have been shown to express this hierarchical structure in their performances. Performers have a tendency to slow down at boundaries in the hierarchy – with the amount of slowing being correlated to the importance of the boundary (Clarke 1998). Thus a performer would tend to slow more at a boundary between sections than between phrases. There are also regularities relating to other musical features in performers' expressive strategies. For example in some cases the musical feature of higher pitched notes causes a performance action of the notes being played more loudly; also notes which introduce melodic tension relative to the key may be played more loudly. However for every rule there will always be exceptions.

Another factor influencing expressive performance actions is *Performance Context*. Performers may wish to express a certain mood or emotion (e.g. sadness, happiness) through a piece of music. Performers have been shown to change the tempo and dynamics of a piece when asked to express an emotion as they play it (Gabrielsson and Juslin 1996). For a discussion of other factors involved in human expressive performance, the reader is referred to Juslin (2003).

2.3.2 Computer Expressive Performance

Having examined human expressive performance, the question now becomes why is it desirable for *computers* to perform music expressively? There are at least four answers to this question:

1. *Investigating human expressive performance by developing computational models* – Expressive performance is a fertile area for investigating musicology and human psychology (Seashore 1938; Palmer 1997; Gabrielsson 2003). As an alternative to experimentation with human performers, models can be built which attempt to simulate elements of human expressive performance. As in all mathematical and computational modelling, the model itself can give the researcher greater insight into the mechanisms inherent in that which is being modelled.
2. *Playback on a music typesetting or composing tool which gives better insight into the final performance* – There are many computer tools available now for music typesetting and for composing. If these tools play back the compositions with expression on the computer, the composer will have a better idea of what their final piece will sound like. For example, Sibelius, Notion and Finale have some ability for expressive playback.
3. *Playing computer-generated music expressively* – There are a number of algorithmic computer-aided composition systems that output music without expressive performance but which audiences would normally expect to hear played expressively. These compositions in their raw form will play on a computer in a mechanical way. A CSEMP would allow the output of an algorithmic composition system to be played directly on the computer which composed it (for example in a computer game which generates mood music based on what is happening in the game).
4. *Playing data files* - a large number of non-expressive data files in formats like MIDI and MusicXML (Good 2001) are available on the internet, and they are used by many musicians as a standard communication tool for ideas and pieces. Without CSEMPs most of these files will playback on a computer in an unattractive way, whereas the use of a CSEMP would make such files much more useful.

2.3.3 A Generic Framework for Previous Research in Computer Expressive Performance

Figure 4 shows a generic model for the framework that most (but not all) previous research into CSEMPs tends to have followed (Kirke and Miranda 2010). The modules of this diagram and the connection system are described beneath Figure 4. The connecting lines between the modules/boxes indicate the direction of flow of data.

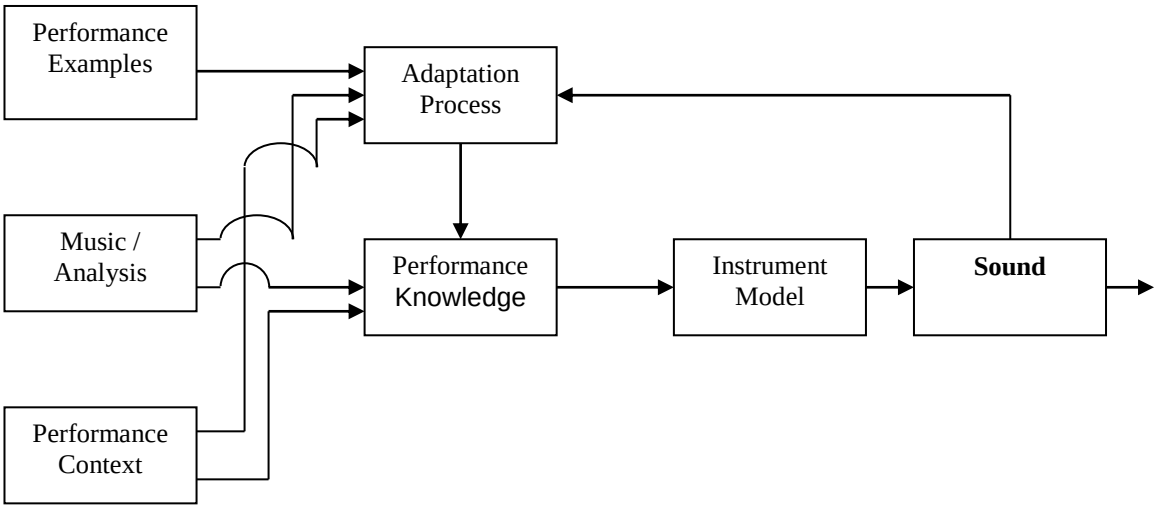

Figure 4: Generic model for most current CSEMPs

Performance Knowledge - This is the core of any performance system. It is the set of rules or associations that controls the performance action. It is the "expertise" of the system which contains the ability, implicit or explicit, to generate an expressive performance. This may be in the form of an Artificial Neural Network, a set of cases in a Case-based Reasoning system, or a set of linear equation with coefficients. To produce performance actions, this module uses its programmed knowledge together with any inputs concerning the particular performance. Its main input is the Music /

Analysis module. Its output is a representation of the performance of the musical input, including expressive performance actions.

Music / Analysis - The Music / Analysis module has two functions. First of all, in all systems, it has the function of inputting the music to be played expressively (whether in paper score, MIDI, MusicXML, audio or other form) into the system. The input process can be quite complex, for example direct paper score or audio input will require some form of analytical recognition of musical events. This module is the only input to the Performance Knowledge module that defines the particular piece of music to be played. In some systems, it also has a second function – to provide an analysis of the musical structure. This analysis provides information about the Music Features of the music – for example metrical, melodic or harmonic structure. (It was mentioned earlier how it has been shown that such structures have a large influence on expressive performance in humans.) This analysis can then be used by the Performance Knowledge system to decide how the piece should be performed. Analysis methods used in some of the systems include Lerdahl and Jackendoff's Generative Theory of Tonal Music (Lerdahl and Jackendoff 1983), Narmour's Implication Realisation (Narmour 1990), and various bespoke musical measurements. The analysis may be automated, manual or a combination of the two.

Performance Context - Another element which will affect how a piece of music is played is the performance context. This includes such things as how the performer decides to play a piece for example happy, perky, sad, or lovelorn. It can also include whether the piece is played in a particular style, e.g. baroque or romantic.

Adaptation Process - The adaptation process is the method used to develop the Performance Knowledge. Like the Analysis module this can be automated, manual or a combination of the two. In some systems, a human expert listens to actual musical

output of the performance system and decides if it is appropriate. If not then the Performance Knowledge can be adjusted to try to improve the musical output performance. This is the reason that in Figure 4 there is a line going from the Sound module back to the Adaptation Procedure module. The Adaptation Procedure also has inputs from Performance Context, Music / Analysis, Instrument Model, and Performance Examples. All 4 of these elements can influence the way that a human performs a piece of music, though the most commonly used is Music / Analysis and Performance Examples.

Performance Examples - One important element that can be incorporated in the Performance Knowledge building is the experience of past human performances. These examples can be used by the Adaptation procedure to analyse when and how performance actions are added to a piece of music by human performers. The examples may be a database of marked-up audio recordings, MIDI files together with their source scores, or (in the manual case) a person's experience of music performance.

Instrument Model - By far the most common instrument used in computer generated performance research is the piano. This is because it allows experiments with many aspects of expression, but requires only a very simple instrument model. In fact the instrument model used for piano is often just the MIDI/media player and soundcard in a PC. IPCS has been developed and evaluated using the Propellerhead Reason MIDI sequencer and piano sampler. In other systems something more complex can be used but still not part of the simulation system, for example a Yamaha Disklavier. However a few simulation systems use non-keyboard instruments, for example Saxophone, and Trumpet. In these cases the issue of a performance is more than just expressiveness. Just simulating a human-like performance, even if it is non-expressive, on these instruments is non-trivial. So systems simulating expressive performance on such instruments may

require a relatively complex instrument model in addition to expressive performance elements.

One element to notice in the above framework is there is nothing incorporated for composition. This is because the CSEMPs which we are aware of focus on performances of already-composed pieces of music.

The review presented here is based on three key areas relevant to this thesis:

1. Emotion-based Systems
2. Hierarchical Structure Systems
3. Multi-agent systems

Emotion-based systems are those systems which focus on allowing the user to specify performances based on emotion or mood, and IPCS utilizes such systems. Hierarchical structure systems are those which analyse the hierarchical structure of the music to generate an expressive performance; IPCS will be seen to expressively perform the hierarchical structure of its music. And the final category – although practically empty - is included because IPCS is a multi-agent system. The review covers around half of the CSEMPS published, however they contain many of the key elements of the broader CSEMP field, and they certainly demonstrate the areas of CSEMP research which are relevant to this thesis. The earlier overview of common CSEMP modules provides sufficient information about the field as a whole.

2.3.5 Emotion-based CSEMPs

Director Musices. Director Musices (DM) (Sundberg et al. 1983; Friberg et al. 2006) has been an ongoing project since 1982. Researchers including violinist Lars Fryden developed and tested performance rules using an analysis-by-synthesis method (later using analysis-by-measurement and studying actual performances). Currently there are around 30 rules which are written as relatively simple equations that take as input Music Features such as height of the current note pitch, the pitch of the current note relative to the key of the piece, or whether the current note is the first or last note of the phrase. The output of the equations defines the Performance Actions. For example the higher the pitch the louder the note is played, or during an upward run of notes, play the piece faster. Another DM rule is the Phrase Arch which defines a "rainbow" shape of tempo and dynamics over a phrase .The performance speeds up and gets louder towards the centre of a phrase and then tails off again in tempo and dynamics towards the end of the phrase. Some manual score analysis is required – for example harmonic analysis and marking up of phrase start and ends.

Each rule has a numeric "k-value" - the higher the k-value the more effect the rule will have and a k-value of 0 switches the rule off. For example the high-loud rule and some other loudness rules have their loudness increase results multiplied by k before adding them to the note loudness; and similarly rules which change duration will have the duration change multiplied by k before being used to change the note duration. The results of the multiple transformation rules are all added together linearly to get the final performance. DM's ability to be a semi-automated system comes from the fact it has a "default" set of k-values, allowing the same rule settings to be applied automatically to different pieces of music (though not necessarily with the same success). Rules are also included for dealing with non-monophonic music (Friberg et al. 2006). The "Melodic-sync" rule generates a new voice consisting of all timings in all other voices (if two

voices have simultaneous notes, then the note with the greatest melodic tension is selected.) Then all rules are applied to this synchronisation voice, and resulting durations are mapped back onto the original voices. The “Bar-sync” rule can also be applied to make all voices re-synchronise at each bar end.

DM has also been developed to enable emotional expression (Bresin and Friberg 2000), drawing on work by Gabrielsson and Juslin (1996). Listening experiments were used to define the k-value settings on the DM rules for expressing emotions. The music used was a Swedish nursery rhyme and a computer-generated piece in a minor mode written using Cope’s (2005) algorithmic composition system in the musical style of Chopin. Six rules were used from DM to generate multiple performances of each piece. Subjects were asked to identify a performance emotion from the list: fear, anger, happiness, sadness, solemnity, tenderness or no-expression. As a result parameters were found for each of the 6 rules which mould the emotional expression of a piece. For example for “tenderness”: inter-onset interval is lengthened by 30%, sound level reduced by 6dB, and two other rules are used: the Final Ritardando rule (slowing down at the end of a piece) and the Duration Contrast rule (if two adjacent notes have contrasting durations, increase this contrast).

Director Musices has been evaluated in a number of experiments. In (Friberg 1995) k-values were adjusted by a search algorithm, based on 28 human performances of 9 bars of Schumann’s Träumerei. A good correlation was found between the human performances and the resulting DM performance. Another experiment involved manually fitting to one human performance the first 20 bars of the Adagio in Mozarts sonata K.332 (Sundberg et al. 2003). The correlations were found to be low, unless the k-values were allowed to change dynamically when the piece was performed. An attempt was made to fit k-values using to a larger corpus of piano music using Genetic algorithms in (Kroiss 2000), and the results were found to give a low correlation as

well. In an attempt to overcome this (Zanon and De Poli 2003) allowed k-values to vary in a controlled way over a piece of music. This was tested on Beethoven's Sonatine in G Major and Mozart's K.332 piano sonata (the slow movement) – but the results were found to be poor for the Beethoven. The RenCon competition is a regularly organized event for comparing by jury the results of various CSEMPs (Suzuki 2003). In the first RenCon in 2002, the second prize went to a DM-generated performance, however the first placed system (a manually generated performance) was voted for by 80% of the jury. In RenCon 2005, a Director Musices default-settings (i.e. automated) performance of Mozart's Minuette KV 1(1e) came a very close 2nd in the competition. However 3 of the other 4 systems competing were versions of the DM-system.

The DM model has been influential, and as will be seen in the systems described later, DM-type rules appear repeatedly. Expressive performance in IPCS is based on DM. DM was chosen because it is a well-developed and tested system whose knowledge is explicit (i.e. not implicit in some kind of regression matrix), and can therefore be integrated as part of a composition process. One of DM's most significant weaknesses is that it rules do not incorporate the relationship between musical hierarchical structure and expressive performance. This relationship must be discerned by the user and the rules applied appropriately. As will be seen below, many later CSEMPs do incorporate this relationship as part of their core functionality. However this is not an issue when it comes to utilizing DM in IPCS, because IPCS itself generates its own musical structure, and applies DM in parallel, thus significantly mitigating DM's lack of structural awareness.

CaRo. CaRo (Canazza et al. 2000; Canazza et al. 2004; de Poli 2004) is a monophonic CSEMP designed to generate audio files which – like CMERS - express certain moods/emotions. It does not require a score to work from, but works on audio files which are mood-neutral. The files are however assumed to include the performer's

expression of the music's hierarchical structure. CaRo's performance actions at the note and intra-note level include changes to inter-onset interval, brightness, and loudness-envelope centroid. A linear model is used to learn actions - every action has an equation characterised by parameters called Shift and Range Expansion. Each piece of music in a particular mood has its own set of Shift and Range Expansion values.

CaRo also learns "how musical performances are organised in the listener's mind" in terms of moods: hard, heavy, dark, bright, light and soft. To do this, a set of listening experiments analysed by Principal Component Analysis (PCA) generate a two dimensional space that captures 75% of the variability present in the listening results; this space is used to represent listeners' experience of the moods. A further linear model is learned for each piece of music which maps the mood space onto Shift and Range Expansion values. The user can select any point in the mood space, and CaRo generates an expressive version of the piece. A line can be drawn through mood space, and following that line in time CaRo can generate a performance morphing through different moods. CaRo's has the ability to have a line drawn through the mood space. Users can draw trajectories through this space which create entirely novel performances; and this can be done in real time. Listening tests with MIDI piano gave results showing that the system gave a good modelling of expressive mood performances.

Emotional Flute. Camurri et al.'s (2000) Emotional Flute system uses explicit Music Features and Artificial Neural Networks (ANN), thus allowing greater generalisation than the related CaRo system. The music features are similar to those used in Director Musices. Expressive actions include inter-onset interval, loudness, and vibrato. Pieces need to be segmented into phrases before being input - this segmentation is performed automatically by another ANN. There are separate nets for timing and for loudness. There is also a third net for the duration of crescendo and decrescendo at the single note

level. However the nets could not be successfully trained on vibrato, so a pair of rules were generated to handle it. A flautist performed the first part of Telemann's Fantasia no.2 in nine different moods: cold, natural, gentle, bright, witty, serious, restless, passionate and dark. Like CaRo a 2-D mood space was generated and mapped on to the performances by the ANNs.

To generate new performances the network drives a physical model of a flute. Listening tests gave an accuracy of approximately 77% when subjects attempted to assign emotions to synthetic performances. To put this in perspective, even when listening to the original human performances, human recognition levels were not always higher than 77%; the description of emotional moods in music is a fairly subjective process.

SaxEx. The SaxEx system (Arcos et al. 1997; Lopez de Mantaras and Arcos 2002) was one of the first systems to learn performances based on the performance context of mood. SaxEx includes algorithms for extracting notes from audio files, and generating expressive audio files from note data. SaxEx also looks at intranote features like vibrato and attack, and needs a non-expressive audio file to perform transformations upon. Analysis methods used include the Generative Theory of Tonal Music (GTTM) (Lerdahl and Jackendoff 1983), and Implication Realisation (IR) (Narmour 1990). Other elements used to analyse the music are ideas from Jazz theory.

SaxEx was trained on cases from monophonic recordings of a tenor sax playing 4 Jazz standards with different moods (as well as a non-expressive performance). The moods were designed around three dimensions: tender-aggressive, sad-joyful, and calm-restless. The mood and local IR, GTTM and Jazz structures around a note are linked to the expressive deviations in the performance of that note. These links are stored as performance cases. SaxEx can then be given a non-expressive audio file and told to play it with a certain mood. A further AI method is used then to combine cases: Fuzzy Logic. For example - if two cases are returned for a particular note in the score and one

says play with low vibrato, and the other says play with medium vibrato, then fuzzy logic combines them into a low-medium vibrato. The learning of new CBR solutions can be done automatically or manually through a GUI.

2.3.6 Hierarchical Structure-based Systems

Hierarchical Parabola Model. One of the first CSEMPs with a hierarchical expressive representation was the Hierarchical Parabola Model (Todd 1985; Todd 1989; Todd 1992; Todd 1995). Todd argues it was consistent with a kinematic model of expressive performance, where tempo changes are viewed as being due to accelerations and decelerations in some internal process in the human mind/body, for example the auditory system. For tempo the hierarchical parabola model uses a rainbow shape like DM's phrase arch, which is consistent with Newtonian kinematics. For loudness the model uses a "the faster the louder" rule, creating a dynamics rainbow as well.

The key difference between DM and this hierarchical model is that the hierarchical model has greater expressive representation and wider performance action. Multiple levels of the hierachy are analysed using Lerdahl and Jackendoff's Generative Theory of Tonal Music (GTTM). GTTM Time Span Reduction (TSR) examines each note's musicological place in all hierarchical levels. The parabolas are generated at each level, from the note-group level upwards (Figure 5) and added to get the performance. This generation is done by a parameterized parabolic equation which takes as input the result of the GTTM TSR analysis.

The performance was shown to correlate well by eye with a short human performance but no correlation figures were reported. Clarke and Windsor (2000) tested the first four bars of Mozart's K.331; comparing two human performers with two performances by the Hierarchical Parabola model. Human listeners found the Parabola version

unsatisfactory compared to the human ones. In the same experiment however, the Parabola model was found to work well on another short melody. The testing also showed that the idea of "the louder the faster" did not always hold.

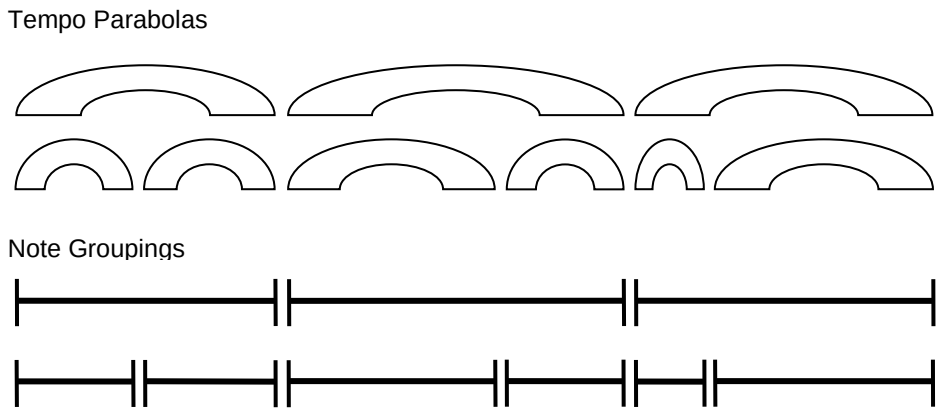

Figure 5: Todd Parabola Model

Composer Pulse and Predictive Amplitude Shaping. Manfred Clynes' Composer Pulse (Clynes 1986) also acts on multiple levels of the hierarchy. Clynes hypothesises each composer has a unique pattern of amplitude and tempo variations running through performances – a pulse. This is captured as a set of numbers multiplying tempo and dynamics values in the score. It is hierarchical with separate values for within the beat, the phrase and at multiple bar level. Table 4 shows the values of pulses for phrase level for some composers. The pulses were measured using a sentograph to generate pressure curves from musicians tapping their finger whilst thinking of or listening to a specific composer. Figure 6 shows the structure of a pulse set in three-time (each composer has a three-time and a four-time pulse set defined). This pulse set is repeatedly applied to a score end on end. So if the pulse is 12 beats long and the score is 528 beats, the pulse will repeat 528/12 = 44 times end on end.

Another key element of Clyne's approach is Predictive Amplitude Shaping. This adjusts a note's dynamic based on the next note simulating "a musician's unconscious ability to sculpt notes in this way" that "makes his performance flow beautifully through time, and gives it meaningful coherence even as the shape and duration of each individual note is unique." A fixed envelope shape model is used (some constants are manually defined by the user), the main inputs being distance to the next note and duration of the current note. So the Pulse/Amplitude system has only note level expressive representation.

Level 2 Composers' Pulses - 4 Pulse					
Beethoven	Duration Amplitude	106 1.00	89 0.39	96 0.83	111 0.81
Mozart	Duration Amplitude	105 1.00	95 0.21	105 0.53	95 0.23
Schubert	Duration Amplitude	97 1.00	114 0.65	98 0.40	90 0.75
Haydn	Duration Amplitude	108 1.00	94 0.42	97 0.68	102 1.02
Schumann	Duration Amplitude	96 0.60	116 0.95	86 0.50	102 1.17
Mendelssohn	Duration Amplitude	118 1.00	81 0.63	95 0.79	104 1.12

Table 4: Level 2 Composers' Pulses

Clynes' test of his own model (Clynes 1995) showed that a number of expert and non-expert listeners preferred music with a composers pulse than with a different pulse.

However not all tests on Clynes' approach have supported a universal pulse for each composer (Thompson 1989; Repp 1990), with suggestions instead that the pulse may be effective for a subset of a composer's work.

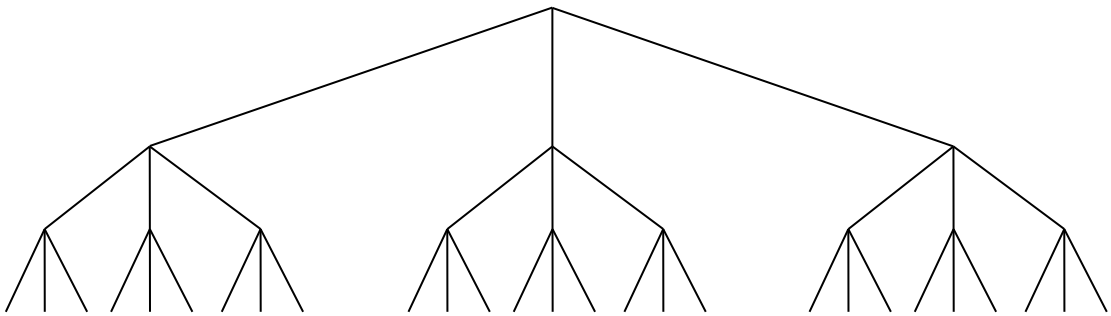

Figure 6: Structure of a pulse set in three-time

Kagurame. Kagurame (Suzuki et al. 1999; Suzuki 2003) is a case-based reasoning system which is designed to incorporate a wider degree of performance conditions than solely mood, for example playing in a Baroque or Romantic style. Rather than GTTM and IR, Kagurame uses its own custom hierarchical note structure analysis to develop and retrieve cases for expressive performance. Score analysis automatically divides the score into segments recursively with the restriction that the divided segment must be shorter than one measure. Hence manual input is required for boundary information for segments longer than one measure. The score patterns are derived automatically after this, as is the learning of expressive actions associated with each pattern. Kagurame acts on timing, articulation, and loudness. There is also a polyphony action called Chord Time Lag – notes in the same chord can be played at slightly different times. Results are reported for monophonic Classical and Romantic styles. It is reported that a high percentage of listeners guessed correctly whether the computer piece was Romantic or Classical style.

Combined Phrase-decomposition/PLCG. The combined Phrase-decomposition/PLCG system (Widmer and Tobudic 2003) also utilizes the hierarchical structure of the music. When learning the performance model, this CSEMP takes as input scores that have had their hierarchical phrase structure manually defined to three levels by a musicologist (who also provides some harmonic analysis), together with an expressive MIDI performance by a professional pianist. Tempo and Loudness curves are calculated from the MIDI performance, and then the system does a multi-level decomposition of these expression curves. This is done by fitting quadratic polynomials to the tempo and loudness curves (similar to the curves found in Todd's Parabola Model).

Once the curve-fitting has been done, there is still a "residual" expression in the MIDI performance. This is hypothesised as being due to note-level expression, and a previous note level expression algorithm – PLCG (Partition Learn Cluster Generalize) - developed by Widmer (2003) is run on the residuals to learn the note-level rules which generate this residual expression. PLCG learns multiple rule-sets in parallel, and clusters rules by similarity, and optimizes them into a final single note-level rule set. The learning of the higher structure level tempo and loudness deviations is done using a case-based learning type method - by a mapping from multiple-level features to the parabola/quadratic curves. An extensive set of music features are used including: length of the note group, melodic intervals between start and end notes, where the pitch apex of the note group is, whether the note group ends with a cadence, and the progression of harmony between start, apex and end.

To generate an expressive performance of a new score, the system moves through the score and in each part runs through all its stored musical features vectors learned from the training; it finds the closest one using a simple distance measure. It then applies the curve stored in this case to the current section of the score. Data for curves at different

levels, and results of the note level expression, are added together to give the expression performance actions.

DISTALL System. Widmer and Tobudic did further work to improve the results of the Combined Phrase-decomposition/PLCG, developing the DISTALL system (Tobudic and Widmer 2003a; Tobudic and Widmer 2003b; Tobudic and Widmer 2003c) for simulation. The learned performance cases in the DISTALL system are hierarchically linked, in the same way as the note groupings they represent. So when the system is learning sets of expressive cases, it links together the feature sets for a level 3 grouping with all the level 2 and level 1 note groupings it contains. When a new piece is presented for performance, and the system is looking at a particular level 3 grouping of the new piece, say X – and X contains a number of level 2 and level 1 subgroupings - then not only are the score features of X compared to all level 3 cases in the memory, but the subgroupings of X are compared to the subgroupings of the compared level 3 cases as well. There have been measures available which can do such a comparison in case-based learning before DISTALL (e.g. RIBL (Emde and Wettschereck 1996)). However DISTALL does it in a way more appropriate to expressive performance – giving a more equal weighting to subgroupings within a grouping. In tests for tempo, 11 of the 16 generated performances were better than a mechanical/neutral performance. Correlations varied from 0.89 for loudness in Mozart K283 to 0.23 for tempo in Mozart K332. The mean correlation for loudness was 0.7 and for tempo was 0.52.

2.3.7 Multi-agent Systems

Multi-Agent System with Imitation. The Multi-Agent System with Imitation (Zhang and Miranda 2007) is the only MAS for expressive performance we are aware of. It is influenced by Miranda's (2003) evolution of musical culture MAS study, and inspired by the hypothesis that expressive music performance strategies emerge through

interaction and evolution in the performers' society. The agents have fitness functions to evaluate theirs and other agents' performances. Rather than comparing the generated performances to actual performances, the fitness function here expresses constraints inspired by the generative performance work of Eric Clarke (1998). When a score is presented to an agent for performance, the system constructs a theoretical timing and loudness curve for the melody (one advantage of this CSEMP is that this music analysis is automatic). However this curve is not used directly to generate the actual performance, but to influence the evolution. The timing curve comes from an algorithm based on Cambouropoulos' (2001) Local Boundary Detection Model (LBDM) – the inputs to this model are score note timing, pitch and harmonic intervals. The resulting curve is higher for greater boundary strengths. The approximate loudness curve is calculated from a number of components – the harmonic distance between two notes (based on a measure by Krumhansl (1991)), the metrical strength of a note (based on the Melisma software (Temperley and Sleator 1999)), and the pitch height. These values are multiplied for each note to generate a loudness curve.

A fitness function is constructed referring to the score representation curves. It has 3 main elements – fitness is awarded if: (1) the pulse set loudness and timing deviations follow the same direction as the generated loudness and timing curves; (2) timing deviations are increased at boundaries; (3) timing deviations are not too extreme. Point (1) does not mean that the pulse sets are the same as the loudness and timing curves, but – all else being equal - that if the dynamic curve moves up between two notes, and the pulse set moves up between those two notes, then that pulse set will get a higher fitness than one that moves down there.

In this model each agent listens to other agents' monophonic performances, evaluates them, and learns from those whose performances are better than their own. Every agent's evaluation equation is their fitness function, and performance deviations are

modelled as a hierarchical pulse set (similar to Manfred Clynes pulse sets described earlier). When an agent hears a performance it evaluates as being better than its own, it attempts to imitate it but ends up giving a slightly mutated performance. This imitating agent then generates a pulse set internally that matches its mutated performance. So pulse sets are not exchanged, performances are (just as humans imitate behaviour, not neural and underlying biological processes). The performances of the system have not been evaluated in testing.

CSEMP	**Performance Knowledge**	**Analysis**	**Performance Context**	**Instrument Model (or Application)**	**Performance Actions**
Director Musices (Sundberg et al. 1983)	Rules	Custom	Mood space	All (Piano)	T/D/A/P
Hierarchical Parabola (Todd 1985)	Parabola equation	GTTM TSR	-	Piano	T/D
Composer Pulse (Clynes 1986)	Multiplier set	-	-	All	T/D
CaRo (Canazza et al. 2000)	Linear model	Custom	Mood space	All	T/K/D/A
Emotional Flute (Camurri et al. 2000)	ANN and rules	Custom	Mood space	Flute	T/D/V
SaxEx (Arcos et al. 1997)	Fuzzy Rules	Narmour / IR / GTTM, Custom	Mood space	Saxophone	T/D/V/K
Kagurame (Suzuki et al)	Rules	Custom	Performance conditions	Piano	T/D/A
PLCG /Phrase-decomposition (Widmer and Tobudic 2003)	Learned rules	Custom, Harmonic by musicologist	-	Piano	T/D/A
DISTALL (Tobudic and Widmer 2003a)	CBR	Custom, Harmonic by musicologist	-	Piano	T/D/A
MAS with Imitation (Zhang and Miranda 2007)	Pulse set	LBDM, Kruhmans, Melisma	-	Piano	T/D

Table 5: Expressive Performance Systems reviewed.

2.3.8 CSEMP Summary

Table 5 lists the systems reviewed together with information about their key modules for this thesis (as described for Figure 4 earlier). Note that the column for Instrument Model is also used for CSEMPs without an explicit instrument model, so as to show their applicability to various instruments. A number of abbreviations are used in Table 5 and throughout the thesis are explained in the List of Abbreviations in Appendix 5.

2.4 Combined Performance and Composition Systems

IPCS combines algorithmic composition and expressive performance. There has been some work into computer-aided composition systems which also generate expressive performances of music as part of their process. However the two systems we are aware of, aside from IPCS, were not actually designed for this. The Ossia system was originally designed for computer-aided composition, and it was found that the compositional algorithms also generated expressive performance-type elements. The CMERs system was designed to algorithmic transform previous composition affectively, but it was found that expressive performance algorithms were also needed to make the transformation effective in communicating emotion. These two systems will now be examined.

Ossia. Dahlstedt's (2007) Ossia is a CAC which incorporates both compositional and performance aspects. Ossia is able to generate entirely new and expressively performed compositions. Like IPCS, Ossia is generating novel compositions and performances, rather than learning how to perform like a human. Ossia generates music through a novel representational structure that encompasses both composition and performance –

Recursive Trees (generated by GAs). These are "upside down trees" containing both performance and composition information. The bottom leaves of the tree going from left to right represent actual notes (each with their own pitch, duration and loudness value) in the order they are played. The branches above the notes represent transformations on those notes. To generate music the tree is flattened – the "leaves" higher up act upon the leaves lower down when being flattened to produce a performance/composition. So going from left to right in the tree represents music in time. The trees are generated recursively – this means that the lower branches of the tree are transformed copies of higher parts of the tree.

Motifs are transformed into new motifs, and themes are transformed into new expositions. Ossia uses a novel transformation-based music representation. In Ossia, transformations of note, loudness and duration are possible – the inclusion of note transformations here emphasising the composition aspect of the Ossia. The embedding of these transformations into recursive trees leads to the generation of gradual crescendos, decrescendos and duration curves – which sound like expressive performance patterns to a listener. The trees also create a structure of themes and expositions. Ossia uses a GA to generate a population of trees, and judges for fitness using such rules as number of notes per second, repetivity, amount of silence, pitch variation, and level of recursion. These fitness rules were developed heuristically by Dahlstedt through analysis-by-synthesis methods. Dahlstedt observes "The general concept of recapitulation is not possible, as in the common ABA form. In terms of testing – the system has not been formally evaluated; though it was exhibited as an installation at Gaudeamus Music Week in Amsterdam.

Despite Ossia's power, it lacks documentation of in-depth examination of its processes and abilities. It also depends significantly on Dahlstedt's heuristic rules which limit the system's flexibility for other composers.

CMERS. The Computational Music Emotion Rule System (CMERS) has a rule-set of 19 rules and requires manual mark-up of phrases in the original score which is to be transformed. These rules are designed not only to inject expressive performance patterns into the score to generate human-like performances, but also to use expressive performance patterns and compositional transformations to express emotions to the listener. To this end CMERS is able to change the score itself, recomposing it. CMERS rules were constructed from a review of 20 studies of music and emotion. The rules for expressing emotions include: moving between major and minor modes, changing note pitch classes, and Director Musices-type rules for small changes in loudness and tempo. It was found that the addition of the microfeature humanisation rules improved the accuracy of the emotional expression (as opposed to solely using macrofeature "recomposition" rules). The rules for humanising the performance include some rules which are similar to Director Musices, such as Phrase Arch and emphasising metrically important beats.

A significant number of formal listening tests have been done by Livingstone, showing that CMERS can change the emotion of music with an average accuracy rate of 78%. However this is limited to four emotions – i.e. happy, sad, angry, or tender. Also it does not take into account the issue of musicality (Zahler 1991) – i.e. are the resulting re-compositions of lesser or greater musicality of the original? Also CMERS is more successful at influencing perceived emotions in musically trained than untrained subjects – and given that the majority of musical listeners are untrained it is possible that this factor could skew the results to appearing better than they are (half of their test subjects were musically trained). CMERS does not take into account structural and cumulative effects on emotional communication. For example the difference in the "sad"-communication in these two scenarios: (a) a happy section of music followed by a sad section of music; (b) an angry section followed by a sad section of music. Another

example would be – how does emotional communication get changed by the various different musical structure types, e.g. ABA, ABAB, AABB, etc?

It was found that CMERS is more successful than DM at expressing emotions, presumably because it adjusts both performative and compositional aspects. CMERS is one of the better tested systems in this review – one reason being that its aim is more measurable than that of a normal computer-aided composition system – i.e. affective transformation of a given score. In its current design it has limited use as a computer-aided composition system, and tests of its usage as such have not been documented. It will be seen later that CMERS is a key contributor to the composition algorithms used in IPCS. The fact that CMERS does not take into account structural effects is less important for IPCS, as in IPCS tune structure is built in parallel with the application of the performance algorithms. Also the limitation of CMERS testing to Happy, Sad, Angry and Tender is less a limitation and more a convenience for IPCS – given that it will be used in IPCS in conjunction with Director Musices (which incorporates similar divisions into Happy, Sad, Angry and Tender).

2.5 Summary

In this chapter we have made a representative survey of the areas involved in the technology behind this thesis – Multi-agent Systems, Computer-aided Composition and Computer Systems for Expressive Performance.

The purpose of this review was to contextualize the research reported in this thesis, and to help to demonstrate why the work contains original contributions. IPCS attempts to address the following elements which are missing from previous work. The systems reviewed…

- …do not address the issue of the generation of a larger scale musical structure.
- …(except ones for modeling) produce only one output tune for users to choose from.
- …that are MAS CAC systems, do not use an affective–based interaction model (a natural type of model to consider due to affective nature of music). They utilize other models, e.g. swarms, negotiation, synchronization.
- …which are interaction-based MAS CAC systems, either have short fixed tune length or do not generate pitches.
- …that are affective-based composition systems do not attempt to utilize the result of interaction emergence in their use of affectivity, or have only reported it as a potential non-implemented feature.
- …for computer-aided composition do not incorporate expressive performance algorithms.

These gaps have focused research in this thesis as follows. Intermediate multi-agent systems have been utilized - such systems maintain the focus on emergence, while incorporating Blackwell's (2007) ideas of increasing structured behavior, including "social" structures. The interaction dynamic chosen in this thesis is imitation, a multi-agent method which has been shown in the review to allow emergent musical behavior and repertoires (e.g. (Miranda 2003), (Gong et al. 2005)). To allow tunes to grow in length, imitated tunes were concatenated. The imitation dynamics are largely governed by an affective processing system, since music is often called a language of emotion, and the model of affectively communicating agents is as relevant to performing musician groups as swarming is to performing improvisers. Finally affective expressive performance was integrated into the imitation process based on one of the simplest CSEMPs (DM (Friberg et al. 2006)). In the next chapter will now describe the IPCS system in detail.

Chapter 3 – Method

An Intermediate Multi-agent Systems for Computer-aided Composition and Expressive Performance of Music will now be introduced – called the "Intermediate Performance Composition System" or IPCS (pronounced "ip-siss"). IPCS is not a Low Processing MAS because its agents have a growing memory of data about themselves and each other, and a music perception and deliberation function. IPCS is not heavy duty because its processing functions are a handful of static linear equation-based music transformations, and a static linear model of musical-affect estimation. Hence IPCS is an intermediate MAS and – as has been observed – has a number of elements in common with previous intermediate musical MAS. It is the only implemented Intermediate MAS designed for computer-aided melody composition of which we are aware, as well as the first integrated computer-aided composition and expressive performance approach that utilizes MAS.

There are two linguistic elements that need to be clarified for this chapter. The use of affective labels such as "Happy" and "Sad" are used to assist clarity in introducing the reader to the concepts of IPCS; they are not meant to be taken literally. For example "Happy" refers to a region of high valence and arousal values, and "Sad" refers to a region of low valence and arousal values. The same goes for any words which may seem to imply that agents have any kind of personification, or deeper intentional or biological model. Such language is merely a shorthand for clarifying functionality. A second issue is the use of the word "performance". We distinguish between the word "performance" and the phrase "expressive performance". An agent has a tune – when it performs that tune it both "re-composes" elements of the tune, and then expressively performs that tune using methods like those found in CSEMPs in the previous chapter. So a "performance" by an agent to another agent includes both compositional

transforms and expressive performance transforms. Thus the use of the word "performance" does not necessarily solely refer to CSEMP-type expressive performance. In fact *whenever* CSEMP-type expressive performance is being referring to the phrase "expressive performance" will be used explicitly.

3.1 IPCS Overview

Agents are initialized with a tune contain a single note, and over the interaction period longer tunes are built through interaction. Figure 7 shows a static representation of an IPCS collection of agents. Below this an informal six point overview of IPCS is now given to allow the reader to have an overall sense of the system before examining the detail. Below each point is an explanation of how the point addresses an issue arising from elements lacking in previous systems.

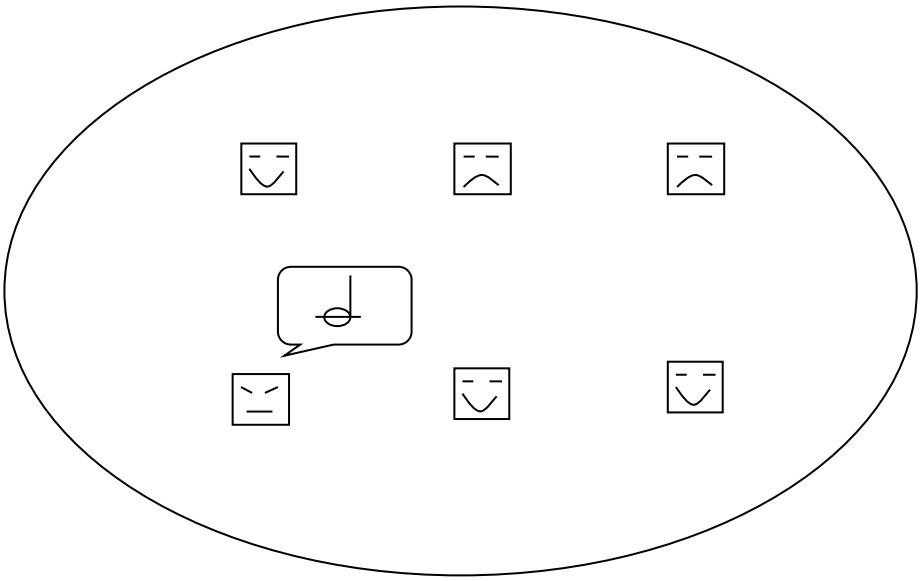

Figure 7: Six IPCS agents in a variety of affective states with on agent performing.

1. Size - IPCS consists of a small-medium size (2 to 16) collection of agents.

   Music composition and performance are in the end subjectively judged. Miranda et al (2010) address this by each agent generating its own variation of the final result. Each IPCS agent will produce its own tune. A small-to-medium number of agents allows all tunes to be auditioned and selected from by the user.

2. Music - Each agent can expressively perform monophonic tunes and learn monophonic tunes from other agents.

   Unlike previous computer-aided composition systems, expressive performance algorithms are incorporated by design.

3. Affective Performance - An agent has an affective state (an "artificial emotional state") which affects how it performs the music to other agents; e.g. a "happy" agent will perform "happier" music.

   There already exist systems based on simple social and group scenarios: e.g. (Blackwell and Bentley 2002) uses a simple swarming model, (Goldman et al. 1999) uses a simple negotiation model, and (Wulfhost et al. 2003b) uses synchronisation between players. But there is not an MAS that uses an affective–based interaction model (a natural type of model to consider due to affective nature of music) to generate melodies.

4. Affective Influence - An agent's affective state is in turn affected by the affective content of the music performed *to* it; e.g. if "sad" music is performed to a "happy" agent, the agent will become a little more "sad".

   The MAS and non-MAS affective-based composition systems reviewed do not attempt to utilize the result of interaction emergence in their use of affectivity, or have only reported it as a potential non-implemented feature.

5. Tune Learning - Agents will only learn tunes performed to them if the affective content of the tune is similar to their current affective state; learned tunes are concatenated to the end of their current tune.

   The use of affective similarity harks back to point (3) above, regarding interaction. Unlike (Miranda 2003) and (Gong et al 2005), tune lengths can increase. A simple tune length increase is desirable (concatenation) so as to focus musical structure generation on interaction-driven emergence. Concatenating at the end of tunes is one element designed to incorporate larger-scale musical structure into the composition, through tune development.

6. Agent Interaction Coefficient - Agents develop "opinions" of other agents that perform to them, depending on how much the agents can help their tunes grow. These opinions affect who they interact with in the future.

   It will be seen that the Interaction Coefficient facilitates the generation of a larger scale musical structure, an element lacking from previous MAS and affective CAC systems.

The use of affective performances for composition is also motivated by the observation that they provide a uniform approach for the generation of melodies and of themes and other structures. For example when agents are learning single notes from other agents, each of the new notes they learn will have been transformed – often to a new pitch. Thus the learning agent will be receiving and concatenating a series of notes which are transformed in pitch, thus creating a melody. Once agents have learned melodies and are then performing these to each other, transformed versions of *melodies* will be being learned, rather than single notes. This second aspect of transforming melodies to create new ones is a common model in algorithmic composition and expressive performance:

the development of a number of motifs which are transformed and combined into a structural hierarchy. A key difference in IPCS is that the same method that is used to create melodies (transformation) is also used to combine the melodies into larger forms. Also the compositions are being created here by communication between individual agents, whereas algorithmic composition systems usually are based on a single agent selecting, transforming and constructing.

3.2 IPCS Agents

An IPCS agent will be defined from a Data / Output / Input / Processing point of view.

3.2.1 Agent Data

An IPCS Agent contains the following data:

a. *Agent Tune* - a monophonic tune in MIDI format
b. *Agent Affective State* –a number pair <valence, arousal> representing the artificial affective state of the agent based on the valence/arousal model of affectivity discussed earlier.
c. *Interaction Coefficient List* – a list of interaction coefficients of all the other agents in the collection – these are non-negative floating point numbers.

The concept of Interaction Coefficient is used in IPCS to attempt to create emergent compositional hierarchies (as will be explained later). However another way of thinking of Interaction Coefficient at this point is to consider an imagined “motivation” for an IPCS agent. The aim of IPCS is – starting with each agent having a single note - to build actual tunes. So an agent should “want” notes. An agent A’s Interaction Coefficient

measure of another Agent B' is based on the note count and number of performances it has added from B to its own tune.

The reason Interaction Coefficient method was chosen for musical hierarchy generation is that – as seen in the review - such methods are common in intermediate multi-agent systems, though not previously for musical reasons. So it was desired to examine if such a standard MAS approach could provide larger scale musical structure. The valence / arousal model of emotion was chosen because of its successful use in a number of musical systems reviewed and because of its continuous nature (allowing continuous adjustment of agent state and musical features).

3.2.2 Agent Output

An IPCS agent can only output one thing – a performance of its current tune. This performance is affected by its internal processing based on its internal data. This allows the agent to perform its tune with affective re-composition and affective expressive performance.

3.2.3 Agent Input

An IPCS agent can receive two types of input:

a. *Other Agent Performance* – when another agent performs a tune to it, an agent can precisely store and can process that performance.
b. *Other Agent Identification* – the agent is able to identify unambiguously any agents during interaction cycles.

These allow the agent to imitate performances and grow their tunes, as well as to utilize the Interaction Coefficient functionality in choosing which other agents to perform to (thus incorporating the larger scale music structure functionality).

3.2.4 Agent Internal Processing

An IPCS agent has six basic internal processing functions. In brackets after each function, the reason for the function is given:

1. *Performance Output Choice*– Choose which agent to perform to, based on its Interaction Coefficient list of agents [large scale music structure].
2. *Performance Output Transform* - Play its current tune as a performance to another agent, with performance features based on its own current affective state [affective-based interaction for composition and expressive performance].
3. *Performance Input Estimate* – Estimate the affective content of a tune performed to it by another agent, and adjust its own internal affective state based on the affective content [to aid the below decision].
4. *Performance Input Choice* – Decide whether to concatenate a performance from another agent to its own current tune based on: (a) the affective content of that performance, and (b) how long the listening agent's current tune is - agents have a finite tune length memory [interaction control].
5. *Performance Input Interaction Coefficient* – Update its Interaction Coefficient measure of another agent based on that agent's performance [large scale music structure].
6. *Performance Input Add* – Add a performance by concatenating it to the end of its current tune [variable tune length].

The actual details of the processing are given later in this chapter. However it is worth noting now an element which will be discussed in more detail later: that the above behaviours, including the concatenation process, are designed to generate compositional structure even though the agents have zero compositional knowledge of melodic structure.

3.2.5 Agent Summary

Figure 8 summarizes an IPCS individual agent structure in a graphical form, showing inputs / outputs / data / processing. The in-depth details of the input / output / processing / data will be discussed in a later section. Before this the interaction protocol and cycle between agents in IPCS will be introduced.

3.3 IPCS Agent Interaction Cycle

The agent interaction cycle will now be defined in terms of: initial condition, repeated cycle, and ending condition.

3.3.1 Initial Condition

Initially all agents in IPCS are given the same single note. By default this is MIDI value 60, or middle C. The note will also have a default MIDI duration of 1 second. Both of these initial values can be adjusted by the user, but the same across all agents. In most musical Cellular Automata and low processing MAS, the music is designed to be an expression of the interaction patterns of the cells / agents. By only providing a single note to an agent, and giving all agents' MIDI notes the same values, it follows that any melodies are constructed only as a result of the agent interactions in this Intermediate MAS.

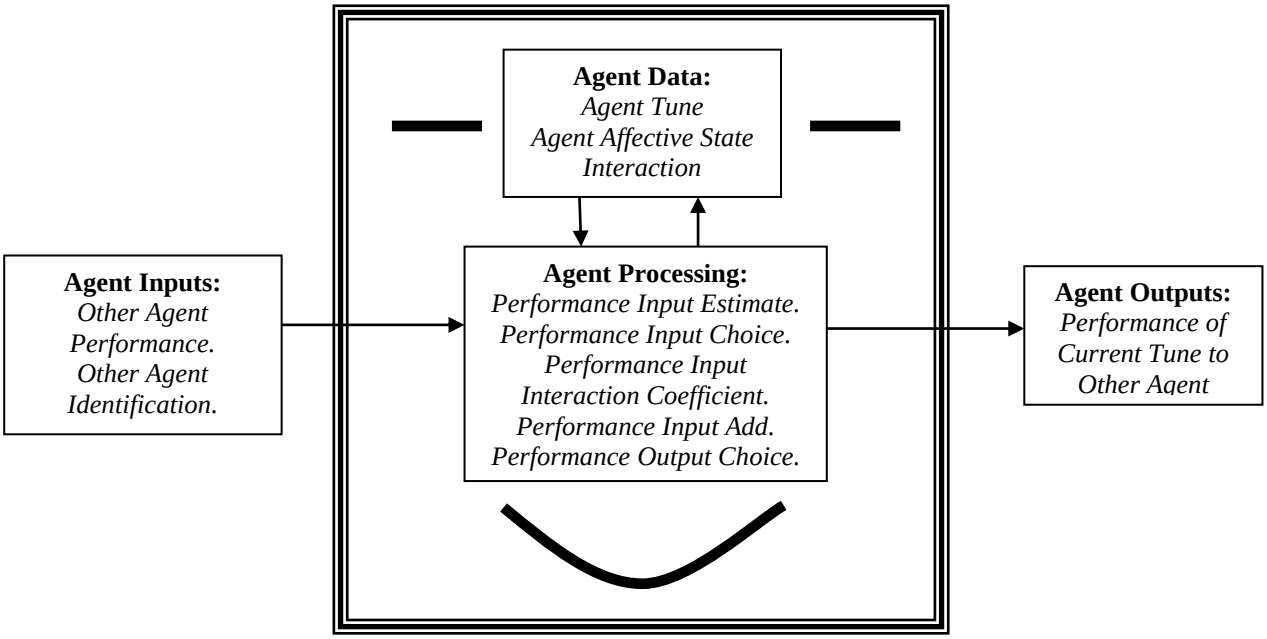

Figure 8: Summary of Agent Structure

The first agent to perform is always Agent number 1, the first candidate for being performed to is always Agent number 2, and then Agent number 3, etc. (Note: hereafter for readability whenever a description of any system focuses predominantly on two players the performer agent will always be labelled as Agent A and the candidate listener as Agent B. But for examples outside of this context, numeric indices will usually be used – i.e. Agent 1, 2, etc.)

3.3.2 Repeated Cycle

The IPCS cycle is as follows:

1. If Agent A's Interaction Coefficient measure for Agent B is below Agent A's average Interaction Coefficient for other agents, then ignore Agent B and select the next listener agent, repeating this test.
2. Agent A performs its tune T_A, adjusting the tune based on its own current affective state to give performance P_A.

3. Agent B estimates the affective content of Agent A's performance P_A.
4. If B's estimated affective content of P_A is close to its own current affective state, Agent B concatenates P_A to the end of its own tune T_B. Or to put it another way: $T_B = T_B + P_A$.
5. Agent B adjusts its own affective state towards its estimate of the affective content of performance P_A.
6. Agent B updates its Interaction Coefficient measure of Agent A proportional to the number of notes provided by Agent A in performance P_A.
7. Agent A turns its attention iteratively to the next agent, and returns to Step 1. Note: once all agents have been considered as candidates for performance by Agent A, a new performer agent is iteratively selected to perform , say agent B, and to listen, say agent C.

We will give two examples to help clarify the cycle process – one in which an agent fails in communicating its performance and one in which it succeeds. Note – in these examples we assume agents' estimation of affective content of tunes is perfect (as will be seen later the estimation algorithms are not perfect).

3.3.2.1 Cycle Example 1

In this example we examine three agents: (a) Agent 1 is the performer and starts by considering performing to Agent 2; (b) Agent 1's measure of Agent 2's Interaction Coefficient is very low in this example; (c) Agent 1's measure of Agent 3's Interaction Coefficient is very high; (d) Agent 1's affective state is high valence and high arousal – i.e. "happy"; Agent 3's affective state is low valence and low arousal – i.e. "sad".

i. Because Agent 1's Interaction Coefficient of Agent 2 is very low, Agent 1 does not even perform to Agent 2. It selects the next Agent iteratively – Agent 3.

ii. Agent 1's view of Agent 3's Interaction Coefficient is very high – so Agent 1 performs its tune T_1, adjusting it to make it "Happier" because of its high valence and arousal state, giving a performance P_1.

iii. Agent 3 estimates the affective content of Agent 1's performance P_1 and gets a result of high valence and arousal – i.e. it estimates it is a "happy" performance.

iv. Because Agent 3's affective estimate of Agent 1's tune is high valence and arousal but Agent 3's state is low valence and arousal – i.e. very different to "happy" - Agent 3 discards Agent 1's tune.

v. However Agent 3 still adjusts it owns affective state towards its estimate of the affective content of performance P_1 i.e. it becomes a little more "happy".

vi. Neither Agent makes any adjustment to their Interaction Coefficient measures since no performances were stored.

vii. Agent 1 remains the performer, and the next agent is iteratively chosen to listen – i.e. Agent 4.

3.3.2.2 Cycle Example 2

In this example: (a) Agent 1 is the performer and starts by considering performing to Agent 2; (b) Agent 1's measure of Agent 2's Interaction Coefficient is very high in this example; (c) Agent 1's affective state is low valence and medium arousal – i.e. "quite sad"; Agent 2's affective state is low valence and low arousal – i.e. "sad".

i. Agent 1's view of Agent 2's Interaction Coefficient is very high – so Agent 1 performs its tune T_1, adjusting it to make it "quite sad" because of its low valence and medium arousal state, giving a performance P_1.

ii. Agent 2 estimates the affective content of Agent 1's performance P_1 and gets a result of low valence and medium arousal – i.e. it estimates it is a "quite sad" performance.

iii. Because Agent 2's affective estimate of Agent 1's tune is low valence and medium arousal and Agent 2's state is low valence and low arousal – i.e. because "sad" and "quite sad" are close – Agent 2 concatenates Agent 1's performance P_1 to the end of its Tune: $T_2 = T_2 + P_1$

iv. Agent 2 adjusts it owns affective state towards its estimate of the affective content of performance P_1 i.e. it raises its arousal from "sad" to "quite sad".

v. Agent 2 has received a tune from Agent 1 so it increases its Interaction Coefficient estimate of Agent 1, proportional to the number of notes in the performance P_1.

vi. Agent 1 remains the performer, and the next agent is iteratively chosen to listen – i.e. Agent 3.

3.3.2.4 Cycle Diagram

Cycles happen in a particular iterative order as follows. Suppose there are three agents: 1, 2 and 3. In the first cycle 1 can perform to 2 and then 3; in the second 2 can perform to 1 and 3; and in the third 3 can perform to 1 and 2. In Figure 9 a summary of the repeated cycle is shown graphically.

3.3.3 Ending Condition

The agents continue in the above cycle until they encounter a composer-defined stopping condition. Example stopping conditions include (based on a composer-given value N) stopping when:

1. The number of cycles reaches N
2. The average agent tune length is greater than N
3. One agent reaches a tune length of greater than N
4. All agents' tunes stop growing in length

In experiments in this thesis, condition 1 alone is used. Though the others could be easily activated and utilized separately in logical combinations.

Note that IPCS has a maximum number of notes per agent parameter, separate to the above. The purpose of this is to prevent exponential tune length growth. For example imagine two agents swapping tunes repeatedly – since each swapped tune was built from previous swapped tunes there will be an exponential length growth. Hence the maximum agent note count.

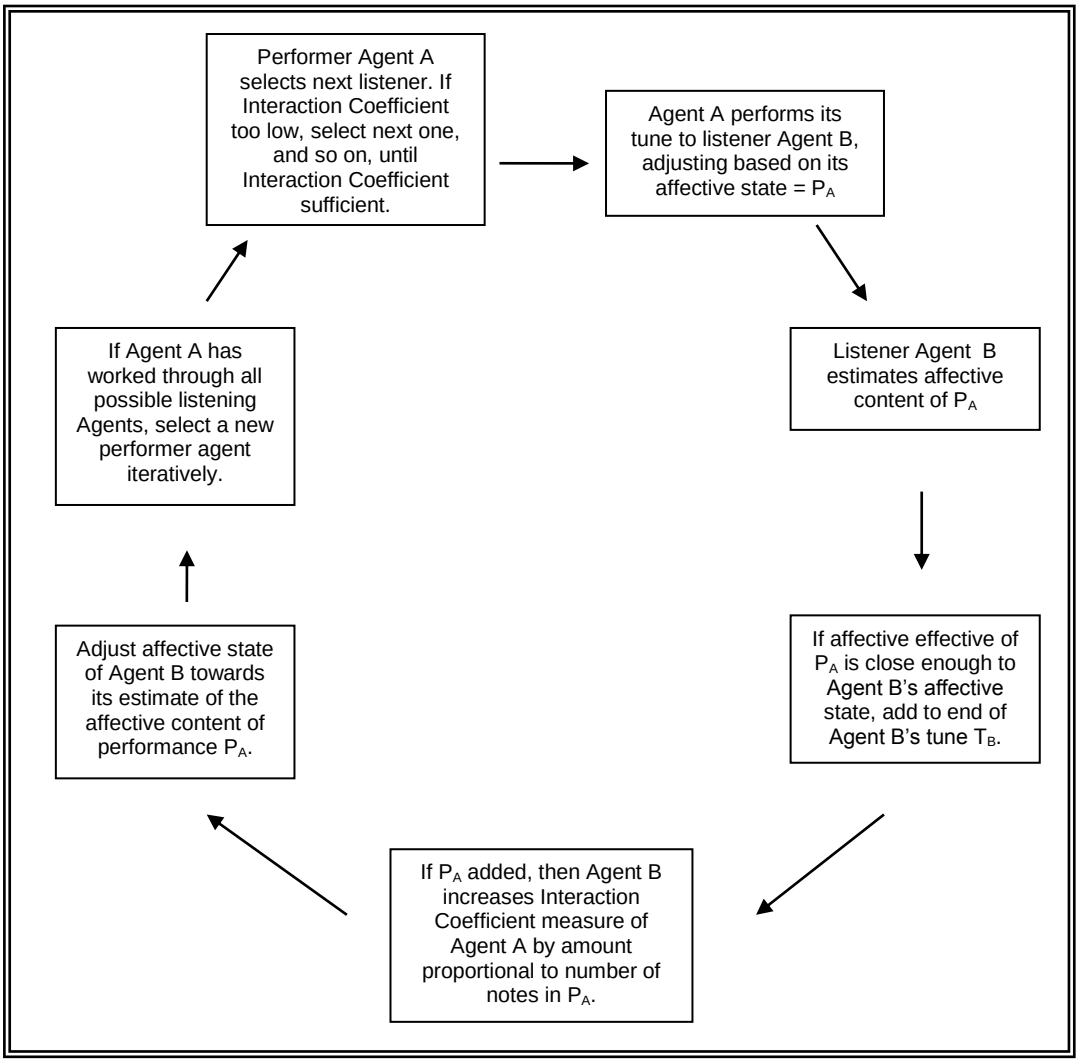

Figure 9: Summary of Agent Cycle

3.3.4 Affective Cycle

Having given an overview of the interaction and cycling protocol, and of the Data / Output / Input / Processing for individual IPCS agent, a more in-depth description of IPCS will now be presented in the next four sections. This will be based on what will be called the "Affective Cycle" which is:

1. Compositional Affective Transform
2. Expressive Performance Affective Transform
3. Performance Affective Content Estimation
4. Affective State Update

These four functions occur in that order in IPCS Agent cycling. The next sections will detail these four parts of the Affective Cycle.

3.4 Affective Cycle – Composition Affective Transform

Before performing its tune to another agent, an agent will transform its tune in a "compositional" way, and then transform it using expressive performance transformations. In this section the focus is on the "compositional" transformations. These are defined as transformations on the agent's MIDI tune that are not of the size and type found in Computer Systems for Expressive Performance. It was seen in the literature review in Chapter 2 that CSEMPs create patterns in music involve micro-changes in timing, loudness and pitch. In systems such as the Computational Music Emotion Rule System (Livingstone et al. 2007) which perform both compositional and expressive performance transformations, the compositional transformations involve larger changes in pitch (i.e. a semitone or more) and timing. The compositional

transformations used in IPCS are based on two sources: CMERS and Juslin's (2001) paper on musical features and emotional expression.

Two types of compositional transformations are applied - Linear Feature Transforms and a Key Mode Transform. The linear transformations – based on Livingstone at al, (2007) and Juslin (2001) are: (a) inter-onset interval and (b) duration are inversely proportional to arousal, (c) loudness is proportional to valence and arousal, and (d) pitch is proportional to valence and arousal (but with valence having twice as much influence). These are implemented using linear equations described in equations (1) to (4). It can be seen that equations (1) and (2) are basically the same, and lead to duration and onset being changed in lock-step. In other words the duration is only adjusted so as to keep it in line with the inter-onset interval; if inter-onset interval is changed by X%, then duration is changed by X%, and duration is never changed by the compositional rules unless inter-onset interval is changed by the rules.

$$IOI_i(A)' = IOI_i(A)(1 - \theta_{IOI} arousal_A) \tag{1}$$

$$dur_i(A)' = dur_i(A)(1 - \theta_{IOI} arousal_A) \tag{2}$$

$$loud_i(A)' = loud_i(A)\left(1 + \frac{\theta_{loud}}{2}(valence_A + arousal_A)\right) \tag{3}$$

$$pitch_i(A)' = pitch_i(A)\left(1 + \frac{\theta_{pitch}}{3}(2valence_A + arousal_A)\right) \tag{4}$$

When an agent A is about to perform and has a particular level of valence ($valence_A$) and arousal ($arousal_A$), it will first compositionally transform its stored tune based on the effects of equations (1) to (4). The primed values on the left hand side of the

equations are the defining features of the compositionally transformed music, and are used to unambiguously generate a transformed MIDI file. The pre-transformation values $IOI_i(A)$, $dur_i(A)$, $loud_i(A)$, and $pitch_i(A)$ are: the inter-onset interval between note i and the next note $i+1$ in seconds, the note duration in seconds, the MIDI loudness, and MIDI pitch of the i-th musical note of Agent A's stored tune. The post-transformation values are signified by a prime. The theta values, (θ_{onset}, θ_{loud}, and θ_{pitch}), are constant, fixed across all agents and define the affective sensitivity of the transformation – i.e. how much affect a change in Agent A's valence or arousal will have on the transformation. They are the maximum variation percentage bars around the current feature value. For example if theta is 0.25, then by Equation (1) the onset will vary from 25% below its current value to 25% above its current value when arousal varies from -1 to 1. If a transformation goes above the maximum MIDI value (127) then it is set to 127. Similarly if it goes below 1 it is set to 1. Note θ_{onset} is used both for onsets and duration so that as gaps between notes are increased or decreased, the duration of the same notes is increased and decreased by the same amount. (Note: for more information on the MIDI music representation parameters and their maximum and minimum values see Appendix 1.)

To give an example of the composition transformation: suppose agent A has a MIDI note with duration 0.5 seconds, MIDI pitch 60 (middle C), MIDI loudness 100, and inter-onset interval 1 second between its onset and the onset of the next note. And suppose the theta values for equations (1) to (4) are all 0.25. The examples for the results of the equations are given in Table 6 for difference values for valence and arousal, between -1 and 1.

The CMERS summary graph for its compositional affective rules is shown in Figure 10. It can be seen that the values in Table 6 are consistent with the properties of the Figure.

Valence	Arousal	dur	loud	pitch
0	0	0.5	100	60
-0.5	0	0.5	93.75	55
0.5	0	0.5	106	65
0	-0.5	0.5625	94	58
0	0.5	0.4375	106	63
1	1	0.365	125	75
-1	-1	0.375	125	75

Table 6: Example Modified Features with thetas of 0.25

Having looked at the linear transforms we will now look at the Mode transform. It is partly taken from Livingstone et al. (2007), and partly from Oliveira and Cardoso (2009). It is a simple transformation:

If valence > 0 transform performance to C major

If valence < 0 and arousal < 0 transform performance to C minor

If valence < 0 and arousal > 0 transform performance to "C semitone minor"

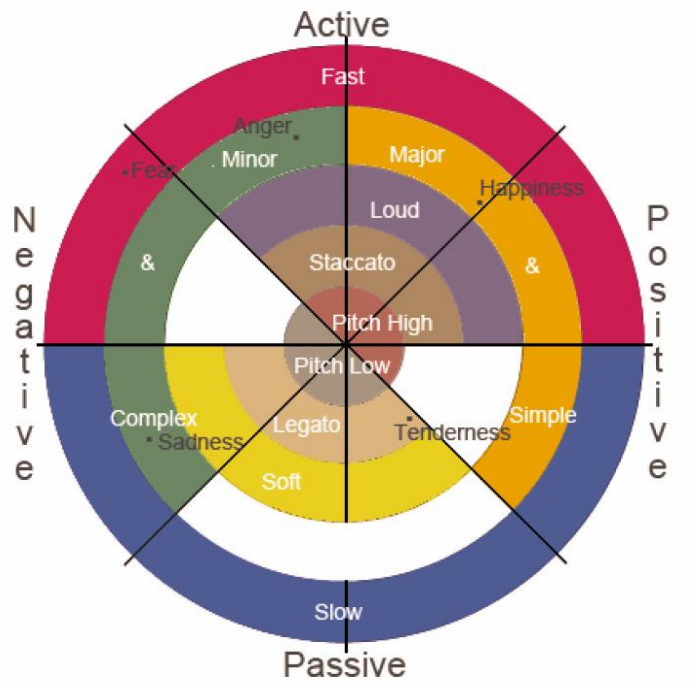

Figure 10: CMERS Music feature summary graph (Livingstone et al. 2007, p5)

The agent transforms its whole performance this way when it performs, i.e. the whole tune it has stored, made up of its initial tune and those it has concatenated during interactions. For positive emotion a major key i s utilized and for negative emotion with negative arousal (e.g. “sadness”) a minor key is utilized. For negative valence and positive arousal (e.g. “Anger” or “Fear”) each note in C major is moved alternately up or down a semitone; this is designed to inject an “atonal” element to the music. The transform is algorithmic and deterministic – it searches either side of the current notes for a note in the new mode which does not violate a MIDI boundary (i.e. not out of the MIDI 128 parameter range). The pseudo-code for the transformation is shown below for major and minor:

```
FOR each PITCH in MIDI file
    IF NOTE is NOT in required KEY THEN
        IF Note INDEX is ODD
            MOVE Note PITCH up a SEMITONE
        IF Note INDEX is EVEN
            MOVE Note PITCH down a SEMITONE
    IF NOTE PITCH now ABOVE MIDI Range (127)
        SET MIDI PITCH TO 125
    IF NOTE PITCH now BELOW MIDI Range (1)
        SET MIDI PITCH TO 3
```

For the “angry/fear” quadrant, the algorithm is the same, except it tries to move notes out of key to create an atonal effect:

```
FOR each PITCH in MIDI file
    IF NOTE is IN C MAJOR THEN
        IF Note INDEX is ODD
            MOVE Note PITCH up a SEMITONE
        IF Note INDEX is EVEN
            MOVE Note PITCH down a SEMITONE
    IF NOTE PITCH now ABOVE MIDI Range (127)
        SET MIDI PITCH TO 125
    IF NOTE PITCH now BELOW MIDI Range (1)
        SET MIDI PITCH TO 3
```

So suppose an agent A has stored a tune from a “happy” agent which is a major key. If agent A then performs its tune while “sad” it will convert all of its tune, including the

major part it received from another agent, into the minor mode. This version of IPCS has no ability for actual key composition functionality, hence the reason for using only C major and C minor.

The key change algorithms could have been implemented in alternative ways, perhaps with greater compositional intelligence. However the above algorithms were chosen to be as simple as possible, while being non-random. The alternating between even and odd pitch indices was used to avoid a drift for pitches being moved up or down, given that randomness was not being used to avoid a drift. This helps to ensure any music in IPCS is not generated by a random process.

3.5 Affective Cycle – Expressive Performance Affective Transform

Having applied affective compositional transforms to its tune, before performing an agent will next apply expressive performance transformations. The expressive performance adjustments made when the agent performs a tune are patterns made up of much smaller changes in note onset, amplitude and note duration. These adjustments are based on the Director Musices (DM) CSEMP already discussed in the previous chapter. (As has already been mentioned DM and its variants are widely accepted and tested systems, and are simple enough to be integrated as part of a composition process.) An IPCS agent performance is based on the following Director Musices rules:

- *High Loud* – Increase loudness as pitch increases.
- *Punctuation* – add a micro-pause and duration change at phrase boundaries to emphasise phrasing.

- *Duration Contrast* – relatively short notes are shortened and relatively long notes are lengthened.
- *Duration Contrast Articulation* – inserts a micro-pause between adjacent notes if the first note has an IOI between 30 and 600 milliseconds.
- *Phrase Arch* – adds an accelerando to the start of a tune and a decellerando to the end. Similarly adds a crescendo to the start and a decrescendo to the end.

These are all the rules identified by Bresin and Friberg (2000) as enabling emotional expressive performance through Director Musices, except for one. The one rule excluded from this list is Final Ritardando. This rule relates to the final slowing down that is often heard when a performer reaches the end of a piece of music. This rule is not utilised here because DM rules in our system are applied repeatedly as tunes are learned and communicated back between agents; whereas the Final Ritardando Rule should only be applied once and at the end of the piece. Applying it repeatedly every time an agent performs will lead to multiple Final Ritardandos appearing in the middle of the piece. All of the other rules (except Phrase Arch) are non-structural and can be viewed as emphasising their effects during their re-application as a tune iteratively expands in IPCS. The only other structural rule in the list – Phrase Arch – is actually applied repeatedly at multiple levels in Director Musices. So it is more reasonable to use such a structural rule in the repeated interactions of agents. In the case of all the Director Musices rules utilized, a natural question is whether such rules retain their purpose and meaning when repeatedly iterated as is done in IPCS. This will be addressed in the experiments in the next chapter.

3.5.1 Representative Affective States

Bresin and Friberg (2000) examined what selection of rules, and what parameter values for the rules, were appropriate to express certain emotions through performance. They only did experiments for distinct emotion labels, rather than continuous valence and arousal values – i.e. there is no simple mapping from continuous valence and arousal indices on to the parameters of the DM emotional expressive performance model. IPCS utilizes continuous values of valence and arousal. So for the emotional expressive performance system a similar methodology to that used by Livingstone et al. (2007) is utilised and four DM emotions are selected as representing the 4 quadrants of valence/arousal. Specifically happiness, sadness, anger, and tenderness are selected - as shown in Table 7. The origin for the 2-D representation used for this table is Figure 3 in Chapter 2. For example: if the agent valence is less than 0, and the agent arousal is greater than 0, then the chosen representative emotion would be "Anger". Table 8 shows the DM results from Bresin and Friberg (2000) for these four representative emotions. Note that the parameter "Turn" in the final row of Table 8 refers to the peak in the "rainbow" shape over the note grouping. For example, a Turn of 0.5 would mean the peak was in the middle, giving a symmetric shape; whereas a Tune of 0.25 would mean the peak was a quarter way in to the note grouping.

A key point is that Phrase Arch is defined in the DM Emotion System at two note grouping levels in the table: Levels 5 and 6. Examples of these grouping levels are shown in Figure 11. In the original Director Musices Emotion System, each grouping on Level 6 will have its own phrase arch generated for it, using the Level 6 parameters in Table 8; and each grouping on Level 5 will have its own phrase arch generated for it, using the Level 5 parameters in Table 8. However in IPCS, the application of Phrase Arch happens in a different way, as the rule is applied during and as part of composition. The aim of the IPCS approach is to generate a structurally meaningful set of accelerandi and deccelerandi. To understand this, there will now be a discussion of

compositional structure in IPCS and how it relates to expressive performance.

Valence	**Arousal**	**Representative Emotion**
Negative	Negative	Sadness
Positive	Positive	Happiness
Negative	Positive	Anger
Positive	Negative	Tenderness

Table 7: Representative Emotions for DM

	Happy	**Sad**	**Angry**	**Tender**
Tone IOI	Shorten 20%	Lengthen 30%	Shorten 15%	Lengthen 30%
Sound Level	Increase 3dB	Decrease 6dB	Increase 8dB	Decrease 6db
High Loud	k = 1.5	N/A	N/A	N/A
Punctuation	k = 2	N/A	k = 2	
Duration Contrast	k = 2	k = -2	k = 2	k = -4
Duration Contrast Articulation	k = 2.5	N/A	k = 1	N/A
Phrase Arch	N/A	Level 5: k = 1.5 Turn = 0.25 Level 6: k = 1.5 Turn = 2	Level 5: k = -0.75 Turn = 0.5 Level 6: k = -0.75 Turn = 0.25	N/A

Table 8: List of Director Musices Rule Parameters to Express Emotional States

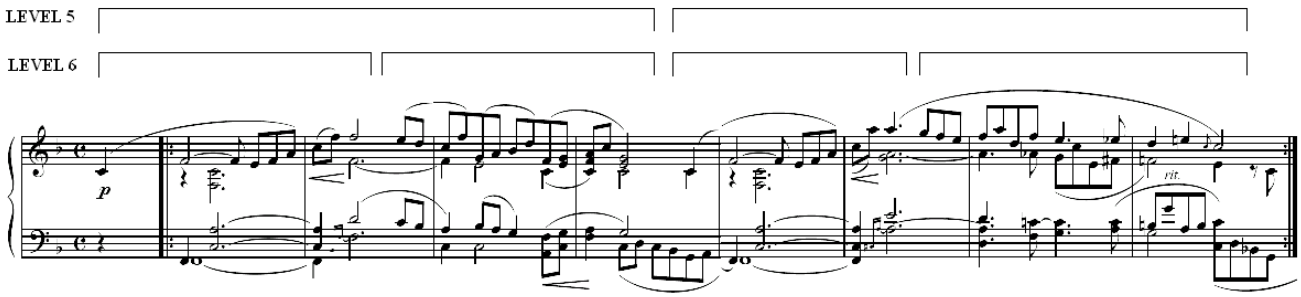

Figure 11: Examples of Phrase Arch Levels 5 and 6 in Schumman's Traumerei (Friberg 1995, p66)

3.5.2 Compositional Structure in IPCS

While the agents communicate and learn each other's tunes, their own tune builds in size. This creates a note grouping structure. Because of the nature of the interaction cycle, the generative structure is binary. In other words, at each iteration where agent A adds a performance to its tune, a new "XY" structure is produced. The X being agent A's original tune, and the Y being the new performance it is adding. Looking at this process building up, the hierarchical structure in Table 9 emerges. The notation in the table can give the impression that the right hand half of the binary growing structure is always simpler than left hand half. However bear in mind that the tunes added to the right are coming from other agents, and these agents are generating a hierarchical structure as well in their own repertoire. So the added "y"-tunes will all have their own internal grouping structure.

As has already been discussed, research into human performers has suggested that – although there are a number of factors involved - the main consistent contributing factor to the expressive deviations during a performance is the hierarchical grouping structure of the music. Human performers have been shown to express the hierarchical structure in their performances. As a result one of the most common observations in music performance studies (Clarke 1998; Clarke 1991; Shaffer and Todd 1987): performers slow down at boundaries in a musical piece, with the amount of slowing down being greater for the more significant the boundary.

Hence Chapter 2's review of CSEMPs showed two frequent commonalities:

1. Formal musical analysis - CSEMPS often base their performances on the hierarchical structure of the music requiring an analysis, sometimes by a musicologist.

2. Hierarchical combination of expressive performance deviations – Most hierarchical systems generate the final tempo and dynamics expressive deviations by combining separate multipliers calculated for each level of the analyzed or manually marked-up hierarchy.

Tune Added	**Structure**
Initial tune x_0	x_0
y_0	x_0y_0
y_1	$(x_0y_0; y_1)$
y_2	$\{(x_0y_0; y_1)\}\{(y_2)\}$
y_3	$(\{(x_0y_0; y_1)\}\{(y_2)\}; y_3)$
etc…	…

Table 9: Compositional Structure Growth in IPCS

As has been mentioned - a significant amount of CSEMP effort is often in analysing the musical structure of the score/audio. However, as has also been discussed, doing musical analysis reliably is nowhere near an automatic process. To clarify the desirability of this, we must re-visit the concept of musical hierarchical structure. A piece of music often has a number of levels of meaning – a hierarchy. For example: notes make up motifs, motifs make up phrases, phrases make up sections, sections make up movements, and movements make up a piece. Each element - note, motif, etc. - plays a role in higher elements. Figure 11 showed examples of two levels of the hierarchy.

The key CSEMP contribution in IPCS is that each time the structure of an agent's composition is extended by tune adding, it will be based on a performance from another

agent and this performance will have had a Phrase Arch applied by the other agent. By repeatedly applying Phrase Arch like this during multi-agent tune exchange it is aimed to simulate the building up of the parabolas, the first two levels of which are shown in Figure 5 in Chapter 2 in reference to Todd's Hierarchical Parabola System (the first CSEMP to implement this parabola hierarchy). However in order to utilize DM to do this in an MAS, some adjustments must be made.

3.5.3 DM Adjustments for MAS

Consider an IPCS system of two agents A and B. Suppose their tunes are both short motifs of 4 notes in length - these would be classed as Director Musices Level 7 motifs in Director Musices in the type of analysis shown in Figure 11 earlier. If A performs its tune, then B will add that to the end of its tune. B's tune will then be 8 notes long, more of a Director Musices Level 6 length tune from Figure 11 (though this level is not shown explicitly in Figure 11). Suppose when B performs its tune back to A, then A adds the tune to the end of its own, thus A's tune becomes 12 notes long, like a Level 5 tune from Figure 11. In addition to this length extension process, at each interaction a Phrase Arch is added by the performing agent over the tune it performs. However, the Director Musices Emotion System only defines its parameters for levels 5 and 6, and the composition method of the MAS will extend beyond such levels after only a few iterations. So in order to allow the application of the Phrase Arch rule to affective generation, a compromise was made whereby a single Phrase Arch rule was applied each time an agent performs; and the parameters of that rule are the average of the parameters for Levels 5 and 6 in Table 8. For "Sad" this gives k = 1.5 and Turn = 0.25; for "Angry" k = -0.75 and Turn = 0.375. Such an averaging and application will be further discussed in the results section.

So the process of agent communication and tune addition causes a hierarchical building up of tunes through different levels and a resultant hierarchical of performance Phrase Arches like Figure 5 in Chapter 2.

3.6 Affective Cycle – Performance Affective Content Estimation

Having introduced the first two parts of the Affective Cycle: Compositional and Expressive Performance Transforms, the part of the cycle involving an agent estimating a performance's affective content will be examined. A linear equation is used to model agent B's affective estimate of a performance by agent A – this is shown in equations (5) and (6).

$$valenceEst_B = x_p mean(pitch_A) + x_l mean(loud_A) + x_k mean(keyMode_A) + x_{IOI} mean(IOI_A) + x_0 \quad (5)$$

$$arousalEst_B = y_p mean(pitch_A) + y_l mean(loud_A) + y_{IOI} mean(IOI_A) + y_0 \quad (6)$$

In these equations $pitch_A$ and $loud_A$ refer to the average MIDI pitch and MIDI loudness of A's performance. $keyIndex_A$ is defined as having value 2 for a minor key, and 1 for a major key; and the key mode of A's tune is estimated using a key profile-based algorithm (Krumhansl and Kessler 1982). The x and y coefficients in the Equations are IPCS constants estimated by linear regression.

They are estimated in a one-off process as follows. A set of 1920 random MIDI files was generated, of random lengths between 1 and 128 notes. Each MIDI file was transformed for 10 known and equally spaced valence and arousal values between -1 and 1 using transformation equations (1) to (4), and key mode transformations. Then a

linear regression was run on the resulting transformed MIDI files against the known arousal and valence values – based on equations (5) and (6). The resulting coefficients are shown in Table 10. The average percentage errors – when tested on a separate 1920 transformed random files - were 10% for valence and 9% for arousal. These are considered to be sufficiently accurate. Actual human musical emotion recognition error rates can be as high as 23% (Camurri et al. 2000); and other far more complex artificial musical emotion detection systems have rates such as 81% (Legaspi et al. 2007). This linear estimation approach will be referred to as the IPCS Linear Estimator (ILE).

	x	*y*
Pitch	-0.00214	0.003025
Loudness	0.012954	0.052129
keyMode	1.1874	-1.4301
IOI	-0.6201	0.59736
Constant	0.61425	-4.5185

Table 10: Regression Results

It will be noted that only the compositional transformations are used in the regression above. This was because it was desired to keep the model flexible for use with and without expressive performance, since a composer may wish to compose a tune without expressive performance. It was found that using the ILE in IPCS with expressive performance transformations overlaid on agent B's performance did not cause excessive errors in the ILE affective estimate by agent A; this will be demonstrated in the results in the next chapter.

ILE is used in two aspects of IPCS – firstly for an agent to decide whether or not to add a performance to its own tune, and secondly for an agent to be influenced by the affective content of a performance it has heard. This second aspect, affective influence, will be detail in Section 3.7 on the last part of the affective cycle. But before that, the decision mechanism for when a performance is added to the end of a tune is examined.

3.6.1 Performance Adding Decision

Interaction decision methods used in previous Intermediate MAS for music creation divide into two types: comparing similarity of musical elements stored by agents (Martins and Miranda 2007; Miranda 2003), and critical evaluation of musical elements (Zhang and Miranda 2007; Gong et al. 2005). In IPCS it was desired to keep as much focus as possible on the patterns of interaction as a primary force generating the music structure. Hence the desire to avoid a function which critically evaluated musical elements, as found in the work of Zhang and Miranda (2007) and Gong et al. (2005). Looking next at the similarity approach in the work of Martins and Miranda (2007) and Miranda (2003), the observation was made that in IPCS tune lengths change. Comparing different size tunes in a meaningful way is not a trivial task. In the Miranda (2003) model all agent tunes remain the same length; in the Martins and Miranda (2007) a new rhythmic similarity measure was introduced to deal with rhythms of different lengths.

It is clear that using an affective comparison, rather than a direct tune comparison, not only bypasses any complexities of different tune length, but also provides potentially simpler ways of specifying compositions, since music is an “affective language” (Juslin 2005) and has even been used as a method of affective communication in robots (Bethel and Murphy 2006). (It was also realised that the use of an affective interaction approach

may provide interesting modelling tools for those wishing to investigate elements of musical culture; however this has not been explicitly developed or evaluated here).

So IPCS uses a meaningful meta-feature for comparison. However this comparison is a little different to that done in other music Intermediate MAS, in the sense that an IPCS agent does not compare the affective content of its stored tune and the performance it is hearing, but compares its own affective state with the affective content of the performance it is hearing. And its own affective state will be as a result of the performances it has heard in the past. The agent may not have stored all these performances. So there could be a substantial difference between an agent's affective state and the affective state of the tune it has stored. However there will not be such a difference between an agent's affective state and the affective state of its own *performance*. Because its own performance is transformed significantly by its own affective state. So agents are really comparing performances, not tunes. So agents will tend to add performances from other agents in a similar, but not necessarily the same affective state. Use of such an affective similarity test is expected to lower the feature differences between two tunes being added. This is because very different feature statistics will often imply a different affective state, and vice versa.

The precise method by which an agent updates its affective state will now be introduced.

3.7 Affective Cycle –Affective State Update

Equations (7) and (8) below are used to update the valence and arousal of agent B after a performance from agent A. The equations were designed to maintain a relatively simple agent model – they are independent and linear. In reality organisms' response to affective influence is almost certainly non-linear, and valence and arousal are not

independent dimensions. However the simplicity of (7) and (8) prevents stepping beyond the scope of the work here, and maintains the MAS as an intermediate system. The γ (Gamma) constant - between 0 and 1 - defines how sensitive an agent is to affective state change – i.e. the amount of change to valence (γ_v) and arousal (γ_a). If it is set to 1 then the new valence and arousal values will be totally changed to the estimated values of the performance the agent has just heard ($valenceEst_A$ and $arousalEst_A$ respectively – from equations (5) and (6)). A value of 0 will lead to no change. Values of γ_v and γ_a increasing from 0 and 1 will cause the estimate to have a proportionally greater effect.

$$valence'_B = (1 - \gamma_v)valence_B + \gamma_v valenceEst_A \tag{7}$$

$$arousal'_B = (1 - \gamma_a)arousal_B + \gamma_a arousalEst_A \tag{8}$$

Once the agent B has decided whether or not to append the performance from A (and if so, has done so), it will update its valence and arousal based on Equations (7) and (8) to from ($valence_B, arousal_B$) to ($valence'_B, arousal'_B$). In future, when it next performs a tune, it will transform it based on its new valence and arousal state. It is designed so that through this series of updating affective states and the agent tune communication and system, new musical structures will emerge.

3.8 Affective Cycle Summary

The affective cycle is summarized in a linear graphic in Figure 12.

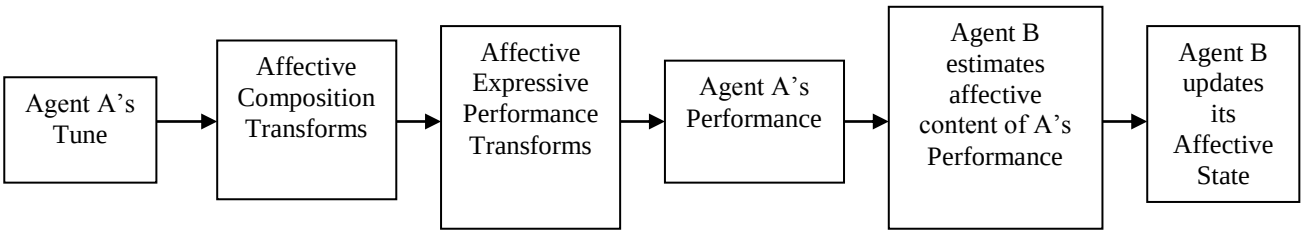

Figure 12: Affective Cycle

To further clarify the Affective Cycle, two examples of interaction cycles will be given in the next two subsections.

3.8.1 Affective Cycle Example 1 – Compositional Only

To focus the example, a two agent system is considered where it is assumed agents always have a high enough interaction coefficient to interact, and in which the affective similarity threshold is so high that agents always consider performances to be similar enough to their own affective state. There are two agents 1 and 2, each starts with a single note (middle C = MIDI value 60) of duration 1 second. Suppose Agent 1 is "happy" – i.e. valence/arousal is 0.5/0.5, and suppose Agent 2 is "sad" – i.e. valence/arousal is -0.5/-0.5. Let the agent valence and arousal update rates both be 0.1 (10%) and the affective pitch, onset and loudness influences be 0.1. The k-Value for expressive performance is set to 0. It will be assumed for simplicity in these examples that the affective estimation equation is 100% accurate. The first agent to perform will be agent 1 – it's default onset is 0 seconds, duration 1 second, pitch MIDI value 60, loudness MIDI value 60 – write this as <O0, D1, P60, L60>.

i. Agent 1's affective state of 0.5/0.5, when inserted into affective influence equations (1) – (4) leads to a new quadruple of <0O, 0.95D, 63P, 63V>.

ii. Next Agent 1 transposes the tune to C Major (since the agent's valence is positive). However, MIDI pitch 63 is already a tone of C Major, so there is no change. Thus it's "happy" state leads to a performance that is slightly shorter, higher pitch and slightly louder than its stored tune.

iii. Agent 2 estimates the affective content of Agent 1's performance – giving estimates of valence of 0.5 and arousal 0.5.

iv. Because the affective similarity threshold has been set so high, Agent 2 will add Agent 1's tune to the end of its own tune. Agent 2's tune will initially be <O0, D1, P60, L60>. So adding Agent 1's performance to the end gives Agent 2 a tune of [<O0, D1, P60, L60>, <O1, D0.95, P63, L63>].

v. Agent 2 updates its valence and arousal using equations (7) and (8) giving a new valence of -0.4 and a new arousal of -0.4. Thus its affective state becomes a little more "positive" and a little more "aroused".

vi. It is Agent 2's turn to perform. It's valence/arousal of -0.4/-0.4 causes it to turn its tune [<O0, D1, P60, L60>, <O1, D0.95, P63, L63>] using equations (1)-(4) into a performance of [<O0, D1.04, P58, L58>, <O1.04, D0.988, P60, L60>]. Note that the onset value for the second note is calculated by transforming the IOI using equation (1).

vii. Next Agent 1 transposes the tune to C Minor (since the agent's valence is negative). However, MIDI pitches 60 and 58 are already tones of C Minor, so there is no change. Thus its "sad" state leads to a performance that is slower, slightly lower pitch, and slightly quieter than its stored tune.

viii. Agent 1 estimates the affective content of Agent 2's performance – giving estimates of valence of -0.4 and arousal -0.4.

ix. Because the affective similarity threshold has been set so high, Agent 1 will add Agent 2's tune to the end of its own tune. Agent 1's tune will initially be <O0, D1, P60, L60>. So adding Agent 2's performance to the end gives Agent 1 a tune of [<O0, D1, P60, L60>,<O1, D1.04, P58, L58>, <O2.04, D0.988, P60, L60>].

x. Agent 1 updates its valence and arousal using equations (7) and (8) giving a new valence of 0.41 and a new arousal of 0.41. Thus its affective state becomes a little more "negative" and a little less "aroused".

xi. It is Agent 1's turn to perform…etc

The main element to note from this example is how melody pitches are generated. There is no explicit mapping from agent states to pitches (as in – for example - Cellular Automata music), nor is there explicit musical intelligence generating the pitches. The melodies are emergent – being produced as a result of the interactions and affective states of the agents.

3.8.2 Affective Cycle Example 2 – Expressive Performance Only

In this example, there are the global k-value is set to 0.2 rather than 0; and the pitch, onset and loudness affective influences are set to 0 – i.e. the compositional elements are removed. The third change is because expressive performance elements only apply when there are 4 or more notes in a tune. So Agent 1's tune will be a 4 note tune: middle C, D, E, F - all with duration 1 and onsets matching duration, and loudness 60. The first agent to perform will be agent 1 – it's tune is [<O0, D1, P60, L60>, <O1, D1,

P62, L60>, <O2, D1, P64, L60>, <O3, D1, P65, L60>]. Agent 2 will still be initialised with the tune <O0, D1, P60, L60>.

i. Agent 1's affective state of 0.5/0.5 – which gives a representative state of "Happy". When this is referenced with Table 8 - the director musices equations – it can be seen that "Happy" leads to high positive duration contrasts, a medium high-loud effect, no phrase arch, an increase in sound level of 3db, and reduction in IOI of 20%; those these will be moderated by the k-value of 0.2.

ii. This leads to the tune being transformed by the Director Musices equations into the following performance: [<O0.032, D0.96, P60, L69>, <O0.96, D0.96, P62, L71>, <O1.92, D0.96, P64, L73>, <O2.88, D0.976, P65, L74>]

iii. Agent 2 estimates the affective contact of Agent 1's tune – assume this is estimated correctly as valence/arousal 0.5/0.5.

iv. Because the affective similarity threshold has been set so high, Agent 2 will add Agent 1's tune to the end of its own tune. Agent 2's tune will initially be <O0, D1, P60, L60>. So adding Agent 1's performance to the end gives Agent 2 a tune of [<O0, D1, P60, L60>, <O1.032, D0.96, P60, L69>, <O1.96, D0.96, P62, L71>, <O2.92, D0.96, P64, L73>, <O3.88, D0.976, P65, L74>].

v. Agent 2 updates its valence and arousal using equations (7) and (8) giving a new valence of -0.4 and a new arousal of -0.4.

vi. It is Agent 2's turn to perform. Its valence/arousal of -0.4/-0.4 gives it a representative emotion of "Sad". So the Director Musices rules in Table 8 turn its tune [<O0, D1, P60, L60>, <O1.032, D0.96, P60, L69>, <O1.96,

D0.96, P62, L71>, <O2.92, D0.96, P64, L73>, <O3.88, D0.976, P65, L74>] into a performance of [<O0, D1.06, P60, L51>, <O1.094, D1.09, P60, L58>, <O2.078, D1.018, P62, L54>, <O3.095, D1.115, P64, L56>, <O4.112, D1.551, P65, L57>].

vii. Agent 1 estimates the affective content of agent 2's performance and adds it to the end of its own tune, and so forth…

One element to note here is how sensitive the features are to a global k-Value as low as 0.2. Hence the low k-values will be used in practice and in the experiments in Chapter 4.

3.9 Interaction Coefficient Update

Now the Affective Cycle has been defined, there is only one more core element of IPCS which needs to be detailed – Agent Interaction Coefficient. In line with the intermediate MAS approach, and the fact that some previous social "reputation" models have been quite simple (Sabater et al 2005), Agent Interaction Coefficient is implemented using a simple linear update equation (10), and a linear threshold decision function (9). Before an Agent A performs to an Agent B it compares its Interaction Coefficient measure of Agent B to the average of its Interaction Coefficient for other agents. The Interaction Coefficient that B has for A is written as *IC(B,A)*. So the comparison done is:

$$IC(B,A) > mean[IC(B, \text{all agents})] \qquad (9)$$

If it is not, then it does not perform to Agent B and moves on to the next agent.

These interaction coefficients are updated whenever an agent adds a performance to its tune. Suppose agent B adds a performance by agent A to its own tune, then it increases its Interaction Coefficient measure of agent A. Suppose Agent A's performed tune is of length N notes. Then the increase in B's Interaction Coefficient measure of Agent A is calculated using the Equation (9). The parameter *d* is a constant called the Interaction Coefficient Update Rate. *N* is the length of the tune added. The increase in Interaction Coefficient is proportional to the length of tune it has added. So the more notes in Agent A's performance, the greater its Interaction Coefficient will be viewed by Agent B.

$$IC(B,A) = IC(B,A) + d.N \qquad (10)$$

This can be visualised as an Agent's basic resources being tunes - so the more notes in an Agent's tune, the greater its potential Interaction Coefficient to other agents. However the actual reason for including Interaction Coefficient functionality, and making Interaction Coefficient proportional to the number of notes in a performing agent's tune is primarily to generate a "social" hierarchy amongst the agents which influences the hierarchy of the composed music. Bearing in mind that an agent will only perform to other agents with a high enough Interaction Coefficient, it can be seen that:

a. Agents which perform more than listen will tend to have lower interaction coefficients
b. Agents which mostly listen and store will have longer tunes and higher interaction coefficients

c. Agents with higher interaction coefficients will tend to be selected as listeners more often

So the system is designed to turn the agent population into a set of agents who tend to perform and have shorter tunes, and a set of agents who tend to listen and store. The aim is for lower Interaction Coefficient agents to be focused on providing lower elements of the musical hierarchy. So for example, it is hoped that agents with lower Interaction Coefficients would provide shorter note groupings ("motifs") to agents with medium Interaction Coefficients, and the "medium" agents would concatenate the motifs in longer note groupings ("phrases" or "themes"); and they would later perform to agents with higher interaction coefficients who would build the themes and phrases into larger note groupings ("sections"). The use of the words "motifs", "phrases", etc is not exact here, but is to help clarify the hierarchical structure.

3.11 Using IPCS

Based on the description in this chapter IPCS' total parameter set is described in Table 11 at the end of this chapter. The table also details the section in which each parameter is discussed in this chapter. This is a total of 18 parameters which is a large number – in particular given that in theory every agent could be given different versions of a significant portion of the parameters. This is not necessarily a problem as the system is still in the research stage and so has been investigated by an expert user. However for the system to be useable by someone who is less familiar with it, it would be necessary to provide a hierarchical GUI with some default values and heuristic controls. Examples that have already been implemented are entry boxes which allow a user to enter the number of "happy", "sad", "tender" and "angry" agents.

The core thing for a user to understand is what are the key defining parameters in Table 11 from a music feature point of view, and how they may be used. This will be investigated as part of the next two chapters. Once the necessary parameters have been set by the user, IPCS is activated and runs its interaction cycles until it reaches an appropriate stopping condition (detailed earlier). Once it completes, the user can audition or visually examine the tunes stored in the different agents in the population. From these the user can select the one they wish to utilize, or perhaps adjust the parameters and set the system running again.

3.12 Chapter Summary

In this chapter an Intermediate MAS for Combined Expressive Performance and Computer-aided Composition has been presented, together with some of the rationale and background for its features and functionality. The background for IPCS comes from a common model in algorithmic composition and expressive performance: the development of a number of motifs which are transformed and combined into a structural hierarchy; and the addition of micro-deviations to this composed piece to express the structure. A key motivator of IPCS development is to investigate the value of mapping this model onto common features found in intermediate multi-agent systems: memory, social networks, and communication.

Concerning this issue of “common features”, there was a change in approach needed when moving from low processing systems for music, to those of slightly higher individual complexity. The low processing and low complexity systems, e.g. swarms and Cellular Automata, are used more directly in music –the group dynamics of the system are more “directly” mapped onto music features such as pitch and timing. One of the reasons for this is that Cellular Automata and swarms are relatively highly

constrained systems with unique, structured and complex dynamics. Researchers implementing these systems for composition often state a desire to utilize this structured complexity and to create “CA music” or “Swarm music”. However once the individual units and interactions become more complex – e.g. in intermediate multi-agent systems – there are more degrees of freedom in designing the properties and interactions of the elements.

Successful music generated by the Game of Life or by a Boid swarm may be considered a representation and argument for the use of CA and swarms in music respectively. As has been seen in this chapter, elements often found in intermediate MAS –e.g. social hierarchies and memory (Sabater and Sierra 2002) – can be mapped on to musical features such as musical hierarchical structure and learning of tunes. So by implementing these features, it is hoped that IPCS will demonstrate the application of intermediate MAS beyond their functionality as test beds for musical culture simulation; specifically as actual musical composition tools.

In the next chapter a series of experiments are designed and run on IPCS, with the results being reported.

Parameter Name	Effect	Section
Number of Agents	Size of the agent population	3.1, p88
Number of Cycles	Number of potential interactions between agents	3.3.2.4, p98
Maximum Agent Notes	An agent which exceeds it max notes will stop adding to its tune	3.3.3, p98
Seed Note Duration	All agents get a single note with this duration to start	3.3.1, p94
Seed Note Pitch	All agents get a single note with this pitch to start	3.3.1, p94
Interaction Coefficient Initial	All agents start with this initial Interaction Coefficient measure of all other agents.	3.9, p122
Interaction Coefficient Update Rate	When agent A provides a tune to Agent B, this is the constant which multiplies the performance length of A to give Agent B's new value for Agent A's Interaction Coefficient	3.9, p122
Interaction Coefficient Threshold	If this value multiplied by the mean Interaction Coefficient of all agents to Agent A, is smaller than Agent B's Interaction Coefficient to A, then Agent A performs to Agent B.	3.9, p122
Similarity Threshold	If agent B estimates the affective content of Agent A's performance as being closer to its own affective state than this threshold measures, then the performance is added.	3.6.1, p115
Affective IOI Influence	How much Agent B's affective state affects the inter-onset interval of a tune it performs.	3.3, p94
Affective Loudness Influence	How much Agent B's affective state affects the loudness of a tune it performs.	3.3, p94
Affective Pitch Influence	How much Agent B's affective state affects the pitches of a tune it performs.	3.3, p94
Initial Valence	The initial valence an agent starts with	3.7, p116
Initial Arousal	The initial arousal an agent starts with	3.7, p116
Arousal Update Rate	How much an agent's arousal is affected by hearing a performance.	3.7, p116
Valence Update Rate	How much an agent's valence is affected by hearing a performance.	3.7, p103
k-Value	Intensity of effect of expressive performance equations.	3.5, p106

Table 11: IPCS Total Parameter Set

Chapter 4 - Experiments and Results

4.0 Parametric Experiments

The issue of how to evaluate an algorithmic composition and expressive performance system is by no means agreed in the research community. There are systems which attempt to simulate human approaches to composition, e.g. counterpoint (Goldman et al. 1999), or simulate human expressive performance (Widmer and Tobudic 2003). These can be more clearly evaluated in reference to their goals. However, as Holland (2000) observes, novel composition is a "wicked" problem in which research often has no clearly defined evaluatable goal. Pearce and Wiggins (2001) discuss the relative lack of evaluation in algorithmic composition research, and they have proposed a framework for evaluation. However this framework is still in the early stages of development.

Parametric/Example-based investigations are common in investigating algorithmic composition systems, e.g. (Beyls 2007; Fonkseka 2000; Anders 2007) and expressive performance (Lopez de Mantaras and Arcos 2002; Zhang and Miranda 2007). Parametric/example-based experiments were done analysing how IPCS output responded to various parameter changes, giving objective information for a potential user of IPSC's behaviour. Such experiments are important because they provide insight into the dynamics of the system. Because composition and performance systems "output" to human beings, it was decided that a listening experiment for IPCS would be conducted with 10 listeners. Most algorithmic composition systems do not include listening experiments in their evaluation procedures, hence is was felt a comprehensive listening experiment was not necessary for IPCS, but an initial test was felt to be helpful as an addendum to the main experiments. This initial experiment focuses on whether a user-defined strong affective bias in the agent initiation is expressed in the final music.

(On a more informal note there is an example of a complete computer-aided composition in Appendix 4, by the thesis author).

Parameter Name	**Default Value**	**Effect**
Seed Note Duration	1	Initial Seed note will be 1 second long
Seed Note Pitch	60	Initial seed pitch will be middle C
Interaction Coefficient Update Rate	0	Agents will never block interaction through Interaction Coefficient judgments
Interaction Coefficient Threshold	0	Agents will never block interaction through Interaction Coefficient judgments
Interaction Coefficient Initial	0.5	Agents will never block interaction through Interaction Coefficient judgments
Valence Similarity Influence	0.5	Valence and Arousal equally affect affective similarity measure
Arousal Similarity Influence	0.5	Valence and Arousal equally affect affective similarity measure
Similarity Threshold	$1x10^{10}$	Agents will never block interaction through lack of tune similarity
Affective Onset Influence	0.5	Equations (1) and (2): onset can vary by +/-50%
Affective Loudness Influence	0.5	Equation (3): loudness can vary by +/-50%
Affective Pitch Influence	0.5	Equation (4): pitch can vary by +/-50%
Arousal Update Rate	0	Agents affective state will stay constant
Valence Update Rate	0	Agents affective state will stay constant
k-Value	0	Agents will not apply expressive performance elements into their performance
Agent Maximum Note Count	$1x10^{10}$	Agent tune size is unlimited

Table 12: Minimum Complexity Parameter Set

In the experiments, different features of IPCS are added in a gradual increase in parametric complexity to give insight into the dynamics of the system. Initially IPCS is initialised with the Minimum Complexity Parameter Set – in this case the base

parameters are shown in Table 12. The aim of this minimum complexity set is to enable the investigation of the effects of number of cycles, number of agents, and to provide a base from which to develop further experiments into other parameter changes. The following experiment sets will be run:

1. Number of Agents, Cycles and Maximum Number of Notes on Tune Growth
2. Initial Affective State
3. Affective Similarity Threshold
4. Inter-agent Affective Update Rate
5. Interaction Coefficient
6. Expressive Performance
7. Listening Experiment

4.1 Experiment Set 1 – Number of Agents, Cycles, and Maximum Note Count

The purpose of this experiment was to examine the effects of number of agents and number of cycles on the growth of the agents' stored tunes. With the Minimum Parameter Complexity Set, agents will never fail to interact and build tunes. This experiment is designed to investigate what is a fundamental element of IPCS – potential exponential growth of the size of tunes. This element is due to the fact that agents learn each other's *complete* performances. This means that if Agent B learns Agent A's performance and performs it back to agent A, agent A will receive a version of its own performance back plus a version of Agent B's original tune. If this process is repeated it can be seen it is exponential. Similarity and Interaction Coefficient thresholds, and agent Maximum Note Count, will restrict this exponential growth. However to see the

patterns of the restriction it is first necessary to see the effect of the tune growth as the number of agents and the number of cycles changes.

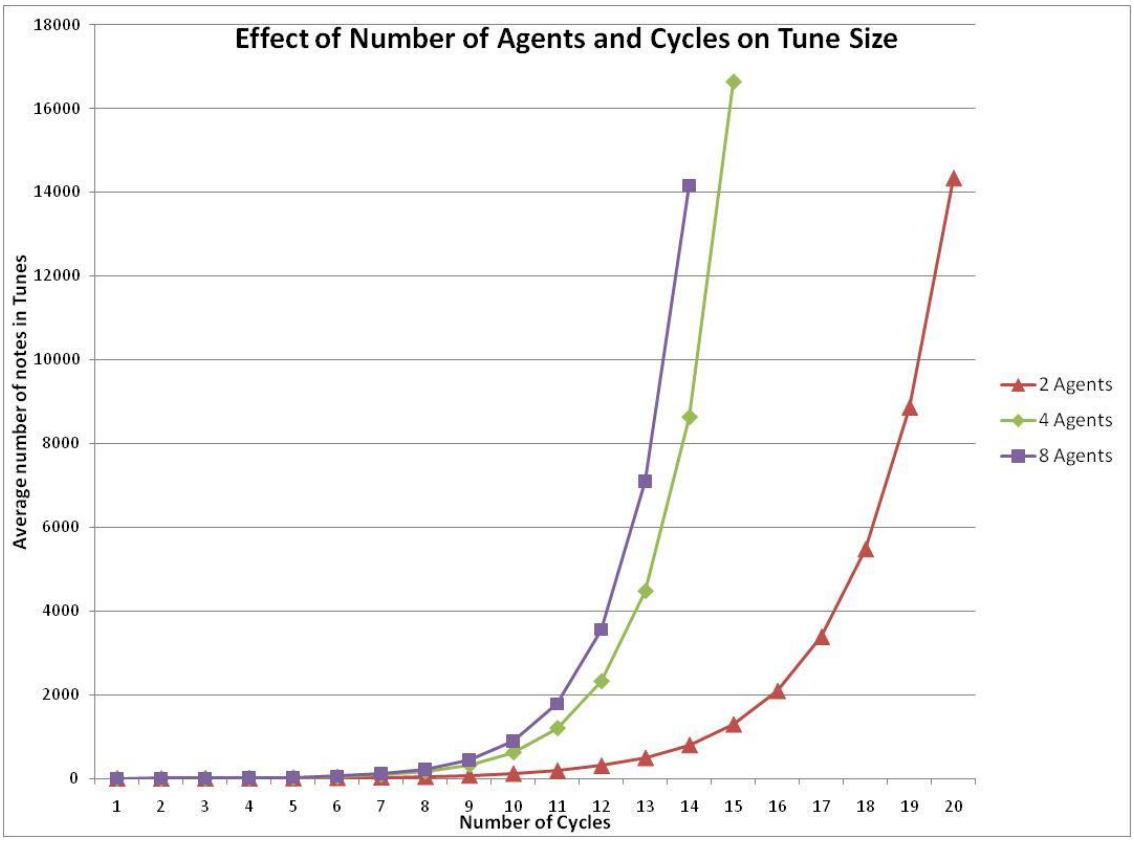

Figure 13: Effects of Number of Agents and Cycles on Tune Size

For this experiment, populations of 2, 4 and 8 agents were used. The results are shown in Figure 13. It was found that the tune growth meant that for a larger number of agents, only a smaller number of cycles were needed before the tune size became very large. This exponential growth is characterised by the fact that all 3 lines have an R^2 fit of greater than 0.999 to an exponential curve of the form ae^{bx}. For example for the 8 agent curve fits to Mean Tune Length = $0.94e^{0.98Cycles}$.

It can be seen that, as expected, the tunes grow rapidly, even in the two agent system. As well as creating tunes of an unlistenable length, this will also cause processing slow-

down on most PCs. This is the reason for including an agent Maximum Note Count. The value of 300 was used. The results of experiments for 2, 4 and 8 agents were repeated and are shown in Figure 14. It can be seen that having a maximum agent note count of 300 leads to the agent tunes being significantly shorter than 300. This is because agents are trying to add performances to tunes which are shorter than 300 notes, but the performances would then make the tune longer than 300. Hence the agents' tunes will stop growing significantly before they reach 300 notes in length. Furthermore, it can be seen that the 4 agent system's curve flattens out twice. The first plateau is due to some, but not all, of the agents being unable to increase their tune length.

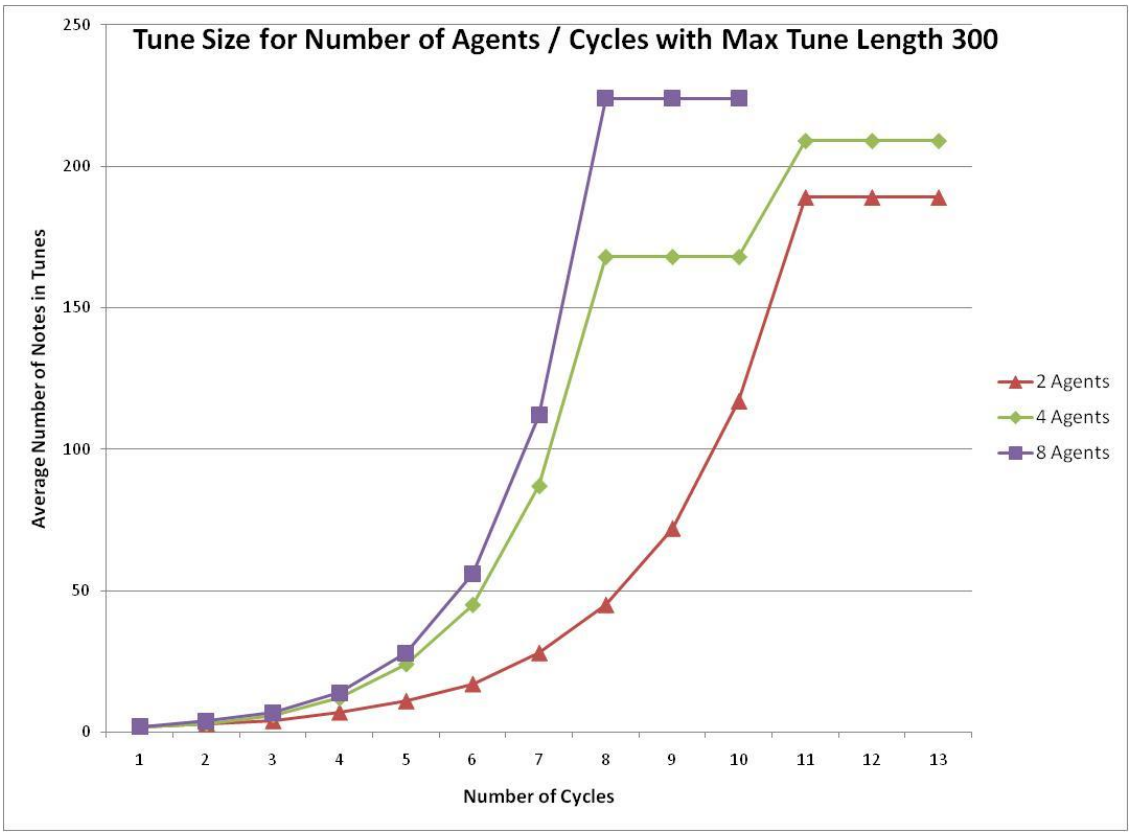

Figure 14: Effects of Max Tune Length of Tune Size

One note on agent population size – the average population size for reviewed musical MAS systems turned out to be 8, and this is the largest and most frequently used population size examined in experiments on IPCS. Apart from being helpfully comparative, due to it relating to average population size for musical MAS, it was also found to give meaningful results for analysis.

4.2 Experiment Set 2 – Initial Affective State

The purpose of this set of experiments is to show how agents' tunes are affected by their initial affective states. In the runs in Experiment 1, every tune was the same, except for their note count. All pitches were 1 second long, all inter onset intervals were 1 second long, and all notes were middle C. So by now adjusting the initial affective state of the agents, they will adjust their tune performances using equations (1) to (4), and thus the note features such as pitch, timing and loudness will be affected. The affective states were constructed based on medium levels of positive and negative arousal and valence, specifically Arousal = 0.5 or -0.5 and Valence = 0.5 or -0.5. For example [0.5, 0.5] would be positive valence and arousal (like "happiness"), whereas [-0.5, 0.5] would be a negative valence and positive arousal (like "anger"). Approximate representative labels for the other two states are "tenderness" [0.5, -0.5] and "sadness" [-0.5, -0.5]. These labels are used as helpful indicators giving a simpler intuitive marker for the experiments.

4.2.1 Two Agents

The affective update rates are all set to 0.5 for pitch, loudness and timing. 10 cycles were run. Figures 15 and 16 show how the initial affective states effect the resulting music features after the cycles are run. (Table 25A p255 in Appendix 3 shows the final mean loudness, and mean estimated valence and arousal of final tunes.) Letters are used

to represent initial affective states: “A” = Angry, “S” = Sad, “T” = Tender, “H” = Happy. Thus “HH” refers both agents initially having initial affective state Happy = [0.5, 0.5]; and “AS” refers to one agent being Angry = [-0.5, 0.5] and one being Sad = [-0.5, -0.5]. The figures also provide a series of dotted and dashed arrows which give insight into the changes in tune features as initial affective state combinations are changed; for example highlighting the progression in features going from AA to AT to TT in Figure 15. Note that in this diagram, and in future ones where the progression is shown as arrows, the selection of these trajectories is not exhaustive. (For example the arrows could have been shown going from TT to AT to AA). In this and other cases those trajectories which are most simply readable to be marked up are the ones which are chosen from the various which could have been marked up.

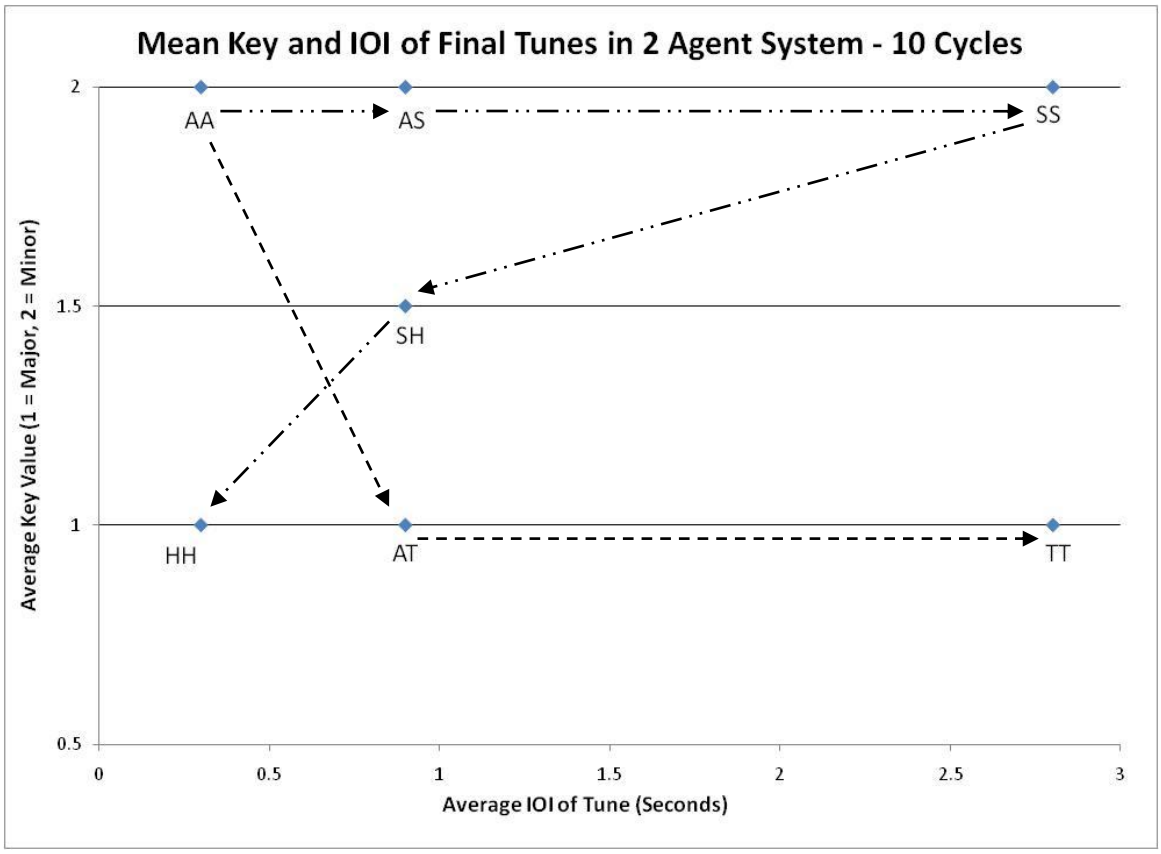

Figure 15: Effects of Initial Affective States in 2 Agent System and trajectories, for mean Key and IOI, 10 Cycles run

The plots are of music features are averaged across the 2 agent system. Note that Pitch values are MIDI values, IOI (inter-onset interval) is in seconds, and a keymode of 1 is major, whereas a keymode of 2 is minor. In Table 25A p255, Loudness is a MIDI value, the Columns Est Valence and Arousal refer to the IPCS linear estimation model being applied to all the agents tunes, and the resulting estimated valence and arousal being averaged across all agents.

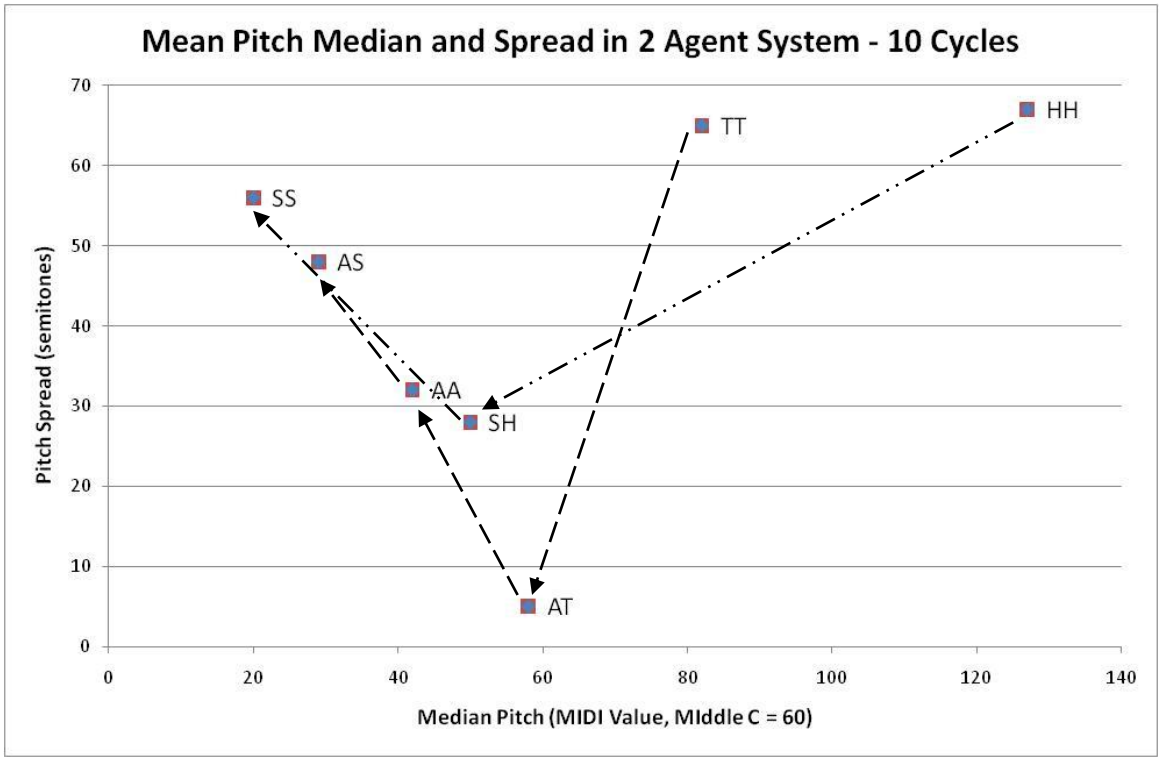

Figure 16: Effects of Initial Affective States in 2 Agent System and Trajectories for Pitch Median and Pitch Spread, 10 Cycles Run

Due to their size and numerousness, actual musical figures are placed in Appendix 2. Figures 54A, 55A and 56A (p228-30) in Appendix 2 show the actual MIDI note development of Agent 1's tunes for selected cycles for three of the runs in Figures 15 and 16: "Sad / Sad", "Sad / Angry" and "Sad / Happy". The content of Agent 1's tune is shown at the end of cycles 4, 6, 8 and 10 in each Figure. The "piano roll" notation is used to display pitches in this thesis. This notation is used as it is simple to interpret whether or not the reader is able to read standard music notation. On the y-axis of the piano roll notation two ticks are separated by a single semitone. The x-axis is in units of seconds. A dotted vertical line is drawn on the piano roll figures to show the location of Middle C. To represent a note of, for example, MIDI pitch 50 starting at time t_1, and finishing at time t_2, a solid line is drawn from time t_1 to t_2 on the graph at a height representing the MIDI value 50 on the diagram (which would the 10th tick below the dotted vertical line). So Figure 54A p228 for example ("Sad / Sad") shows three descending notes of increasing length, starting at Middle C. It can be seen in Figure 54A, the note groups are all being transformed pitch-wise by Equation 4 in the same direction because both agents have the same affective state, with low valence and low arousal. However the tune is not monotonic due to the fact that note groups are transformed and re-combined during the iterative process of agent communication. This process takes on a greater variety when the agents have different affective states as in Figure 55A p229. In 55A the second note in cycle four rises due to the fact that Agent 2 has an affective state with high valence and arousal.

Figures 57A, 58A, and 59A (p231-2) plot the loudness of the final tune for these runs. The Loudness graph is displayed by note index rather than time; the note index of the first note is 1 and the second note is 2, and so forth. So each point on the graph represents one note. The MIDI loudness value is goes from 0 = silent, up to 127 = maximum loudness. The single initial seed note of an IPCS agent always has MIDI

loudness 60. So in Figure 57A p231 the loudness stays at 60 for the first couple of notes, then falls to the mid-30s for 1 note, and rises to the mid-40s for the next note, and so on. The similarity between transformation equations (3) and (4) (p101) mean that the loudness in Figures 57A, 58A, and 59A p231-2 follows the pitch direction quite closely.

4.2.2 Eight Agents

A second set of experiments was done in this vein but increasing the population to 8 agents to see how a larger population interacts with the affective initiation functionality. Because adding 6 agents leads to a larger number of interactions per cycle, the number of cycles was lowered to 8, as this made the final number of notes of the same order as the 2 agent example above. A similar notation is used: "AAAAAAAA" means all 8 agents were initialised with "Angry" – i.e. valence -0.5, and arousal 0.5. "AAAASSSS" means that the first 4 agents were initialised "Angry" and the last 4 initialised "Sad". The results for various music features are shown in Figures 17 and 18. Figures 60A-62A (p233-5) show the development of Agent 1's tunes for even cycles for "SSSSSSSS", "SSSSHHHH", and "SSAAHHTT". Note that Agent 1's tune in Cycle 2 consists of one note only, hence the reason that the first piano roll in the figures is 1 second long and filled in black. Figures 63A-65A (p236-237) show the loudness for the runs.

Figure 17 and 18 actually compare the 2 and 8 agent systems music feature results on the same diagram. It can be seen that adding more agents does not in this case cause very large changes in statistics – apart from for "SSSSHHHH" vs. "SH". In fact looking at Figures 17 and 18 and the raw data tables in Appendix 3 (Table 25A and 26A p255) it can be seen that SSSSHHHH/SH is the only initial state which shows a large change in any final music features. Figure 19 shows the average of affective estimates on the tunes for different initial affective states. For each agent system (e.g. "HH") the affective state of the final tunes was calculated using the linear model and this average

is plotted in Figure 19. For readability Figure 19 is bounded at 0.5; this means the more extreme feature values in valence/arousal space are placed at the bounded extremes and labelled with their actual values.

Figures 17, 18 and 19 also have arrows to highlight the progression of features as initial affective states are changed. They furthermore have dashed ellipses to highlight how close together the resulting features of the 2 agent system are to their "equivalent" 8 agent system. Figure 19 has some additional sets of dashed lines. These are to highlight the relationship between the 2 agent HH-AA-TT-SS boundary in valence/arousal space, and the boundary HHHHHHHH-AAAAAAAA-TTTTTTTT-SSSSSSSS.

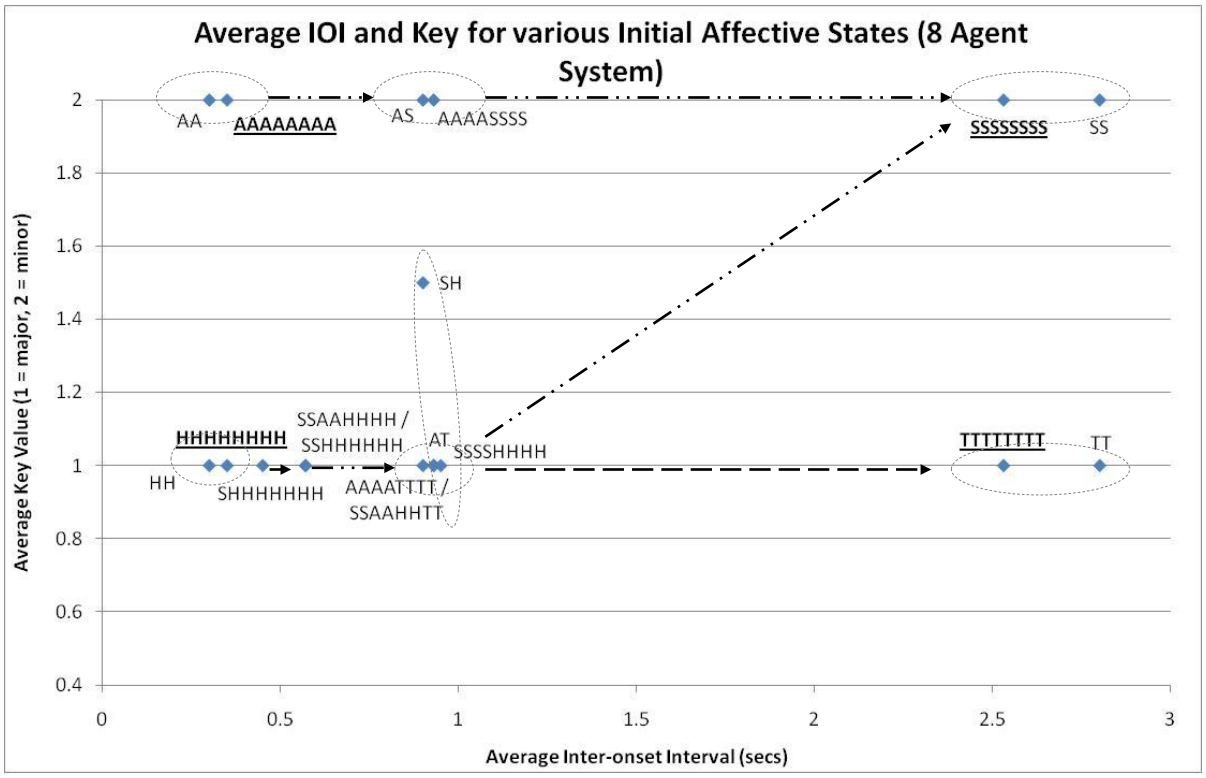

Figure 17: Effects of Initial Affective States in 8 Agent System on average IOI and Key, 8 Cycles Run

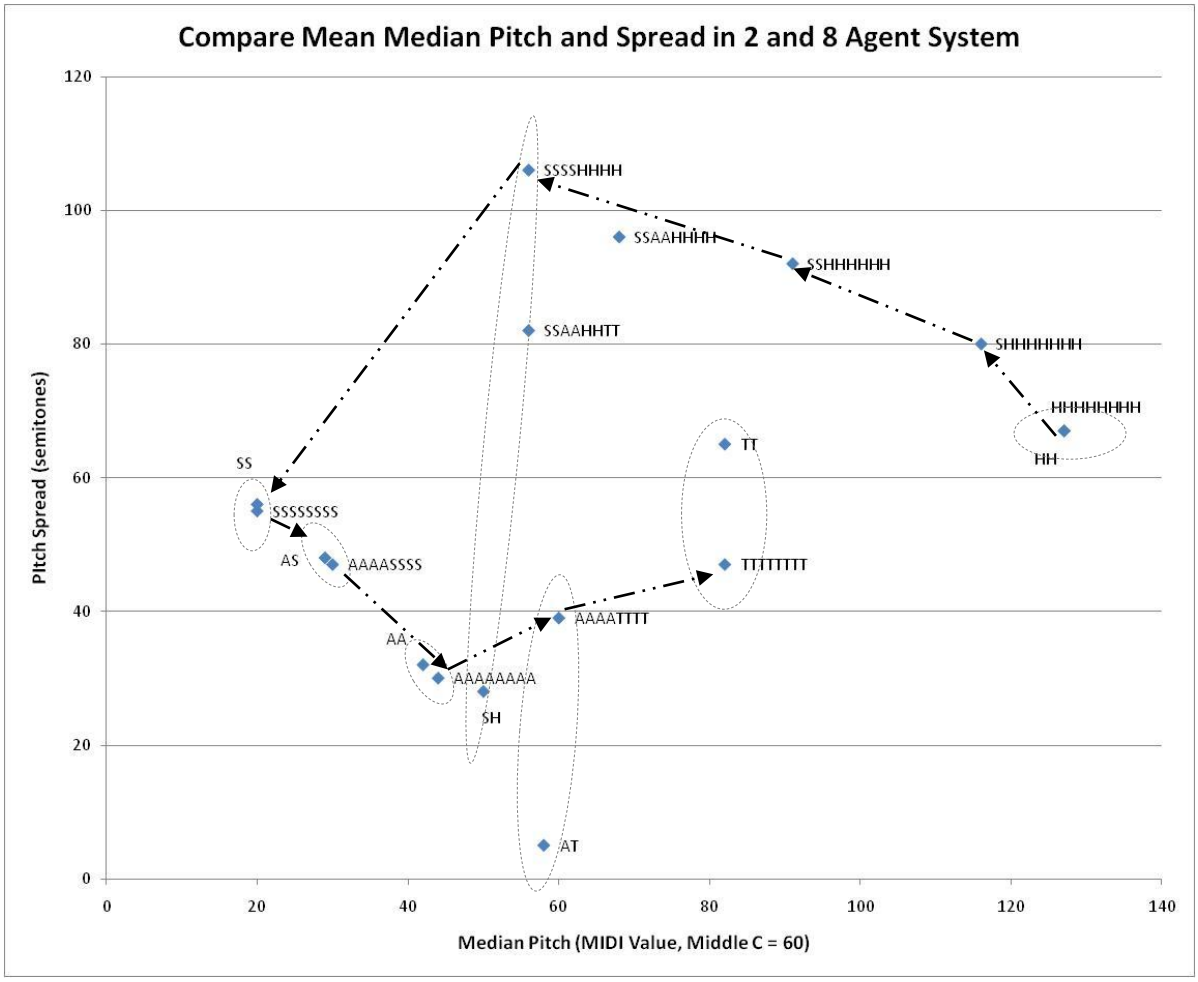

Figure 18: Effects of Initial Affective States in 8 Agent System on Pitch Median / Spread, 8 Cycles Run

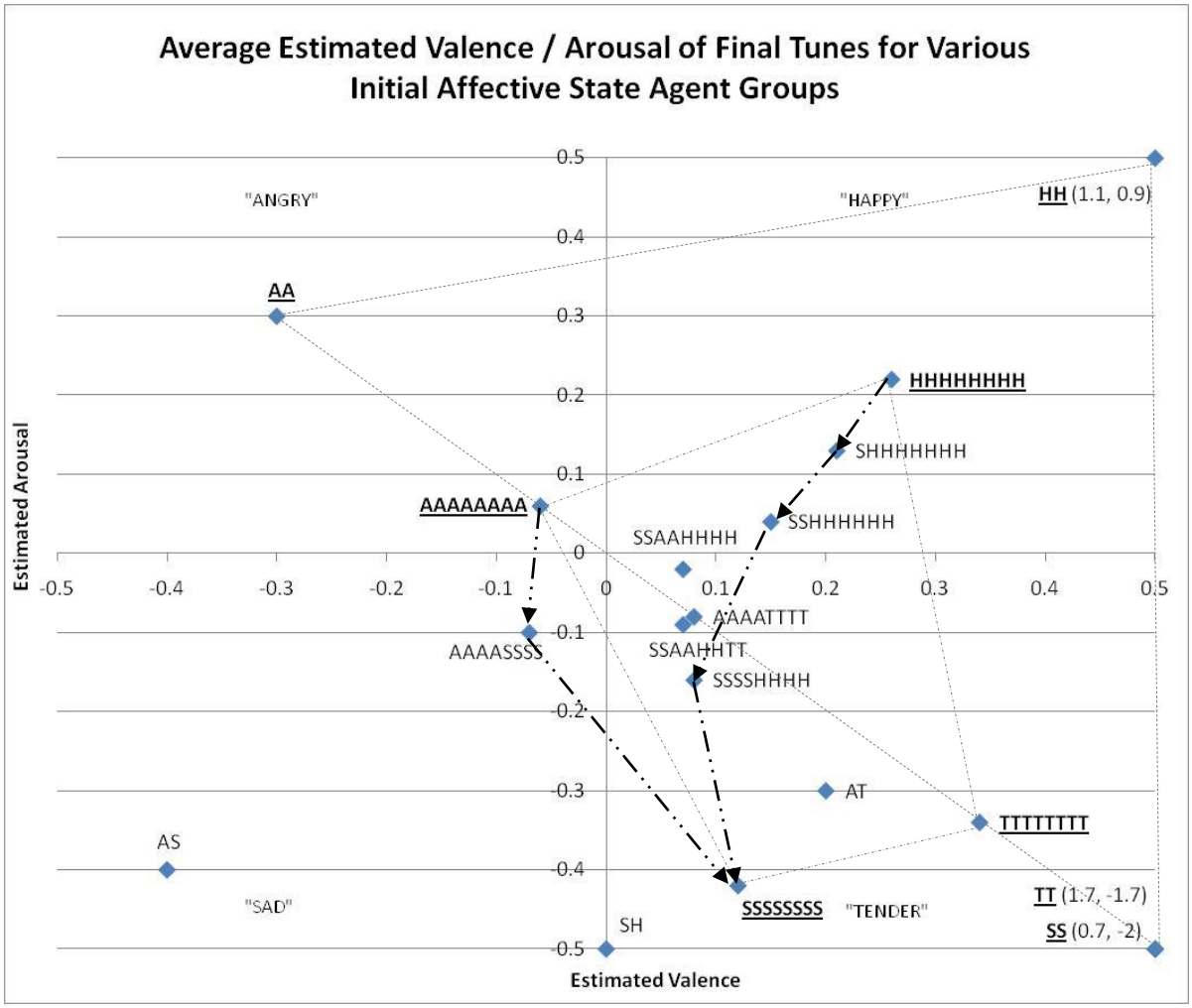

Figure 19: Average Estimated Valence / Arousal of Final Tunes for Various Initial Affective State Agent Groups

4.3 Experiment Set 3 –Affective Similarity Threshold

Having examined "uninhibited" tune learning in the previous experiments, the affective similarity threshold will now be included. The basic function of the similarity threshold is to examine the possibility of implementing some diversity between agents' tunes and an emergent affective tune structure (as well as to further limit tune growth, in a meaningful way). For an agent B to add a performance from agent A to its own tune, agent A's performance must have features which are sufficiently similar in affective implication to agent B's actual affective state. There are two types of similarity included

in each comparison – the valence and the arousal. By default they contribute 50% each to the total similarity. A similarity result of 0 means that the valence and the arousal for both are exactly the same; if there are differences the similarity result will be greater than 0.

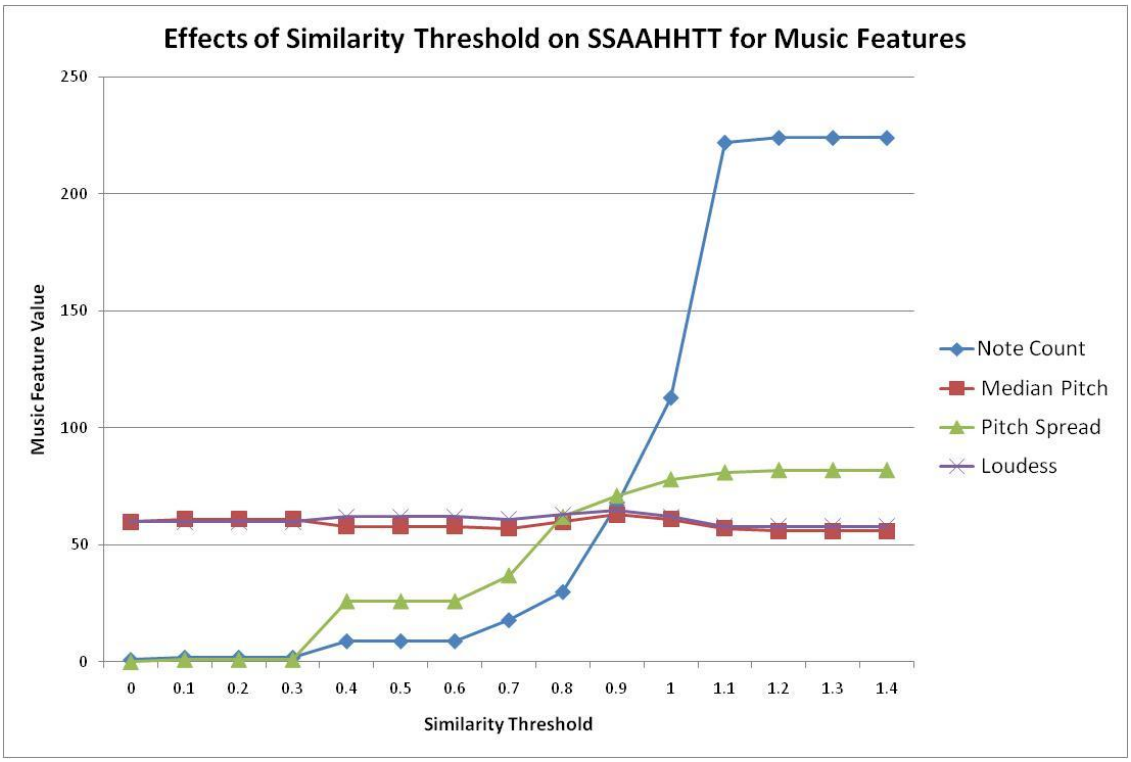

Figure 20: Effects of Similarity Threshold SSAAHHTT Agent System on Music Features

For this experiment affective similarity threshold was adjusted through a number of possible values. An 8 agent system was used with agents initialised to 2 happy, 2 sad, 2 angry, and 2 tender (hereafter referred to as an "Equal Spread" initialization), and it was run for 8 cycles. Figures 20 and 21 show the effects of similarity threshold on final tune lengths and musical features. Figure 22 shows the effects on the mean estimated valence and arousal of the final tunes.

Some immediate conclusions are that a similarity threshold of 0 causes no interaction (0 note count), and that a similarity threshold of 1.2 and above causes agents to freely interact (and therefore gives the same result without the similarity thresholding – SSAAHHTT in Figures 18/19).

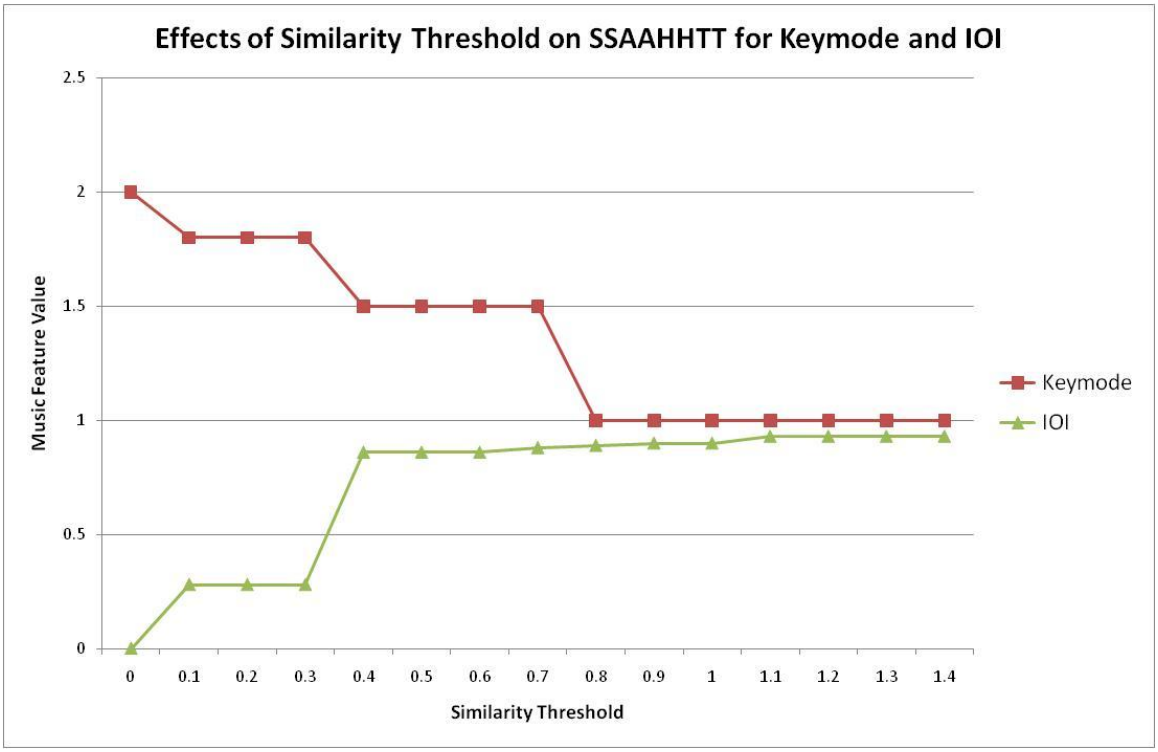

Figure 21: Effects of Similarity Threshold SSAAHHTT Agent System on Keymode and IOI

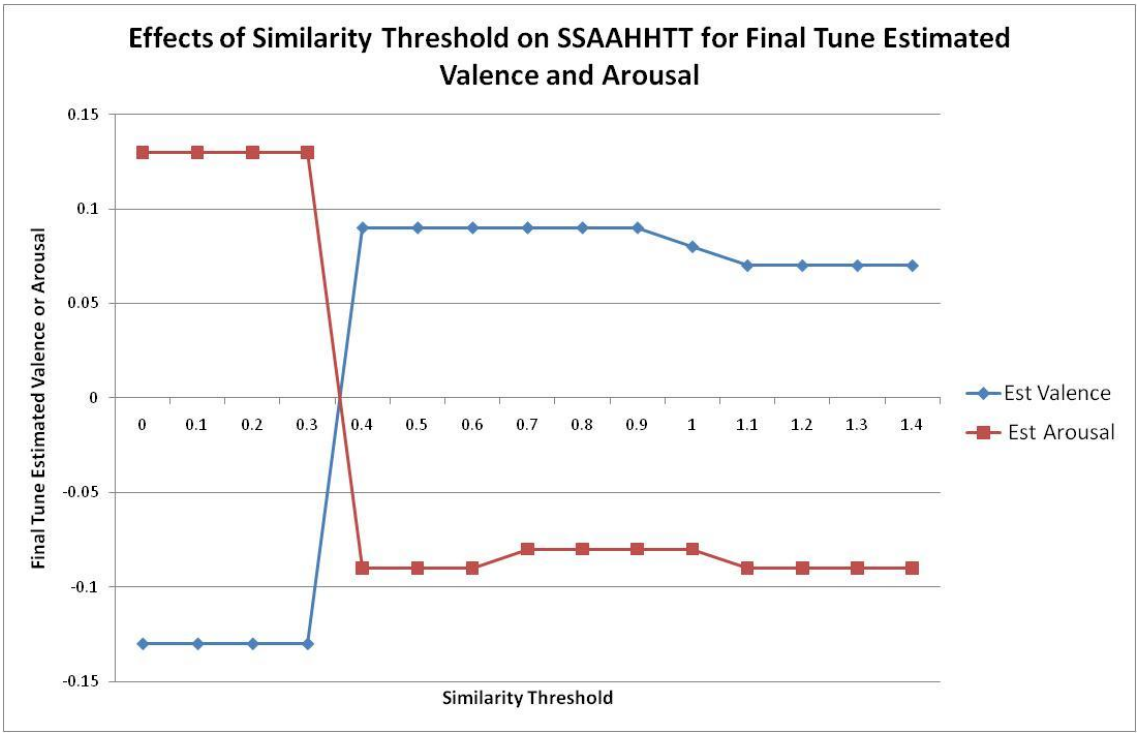

Figure 22: Effects of Similarity Threshold SSAAHHTT Agent System on mean Estimated Valence and Arousal of Final Tunes

4.4 Experiment Set 4 –Inter-Agent Affective Update Rate

In the previous experiment agents' affective states have remained constant. The agents' ability to be effected by affective content of performance will now be added. This element is parameterized by the Affective Update Rates in Equations (7) and (8) (p117). For example, an arousal update rate of 0.1 means that if an agent estimates a performance it hears to have an arousal *r*, then it will add 10% of *r* to 90% of its own arousal, thus moving its arousal toward *r* by 10%. Whereas an update rate of 0.3 will cause it to move its arousal toward *r* by 30%; and an update rate or 100% will cause it to replace its arousal value by *r*. Thus an increase in arousal and valence update rates will change the rate at which agents are influenced by each others' affective music. This

is with the aim of adding extra affective dynamics to the interaction process which it is hoped translate into musical dynamics over a longer time-scale.

Experiments were done first with the tune adding ability switched off. This will lead to agents continuously playing a one note performance to each other, with the note transformed based on their affective state. Figure 23 shows how agent valence and arousal states change for two different 2 Agent systems: one with an “angry” agent and a “neutral” agent; and one with an “angry” and a “sad” agent. “Neutral” means an agent with both initial valence and arousal equal to 0. To highlight the valence and arousal effects, when final mean valence is being plotted it is for runs with an arousal update of 0. And when final mean arousal is being plotted, it is for runs with a valence update of 0.

To examine how this translates to larger agent groups, an 8 agent system is considered with 2 of each type of affective state. Unlike the 10 cycles above, 8 will be used (since there are more agents). But once again there is no thresholding and no tune adding. Figure 24 shows the results of the 8 cycles.

It can be seen in Figure 23 that the average valence of the system is made more and more negative as the affective influence between agents is increased – and that the angry agent is dominating both mean valence and arousal of the 2 agent system. When neither agent is neutral the negative valence of the two agents re-enforce each other. In Figure 24 the negative valence and positive arousal dominates – making the group on average more “Angry”. This is because of the agent ordering, and because the linear model has a bias towards negative valence and positive arousal; for example in Tables 32A and 33A (p259) the average linear model estimates for the MAS’s constant tunes are valence = -0.13 and arousal = 0.13; when ideally the average tune valence and

arousal in that scenario should both be estimated as 0. Since the tune does not grow in length (as tune learning is switched off here) these biases have a significant effect.

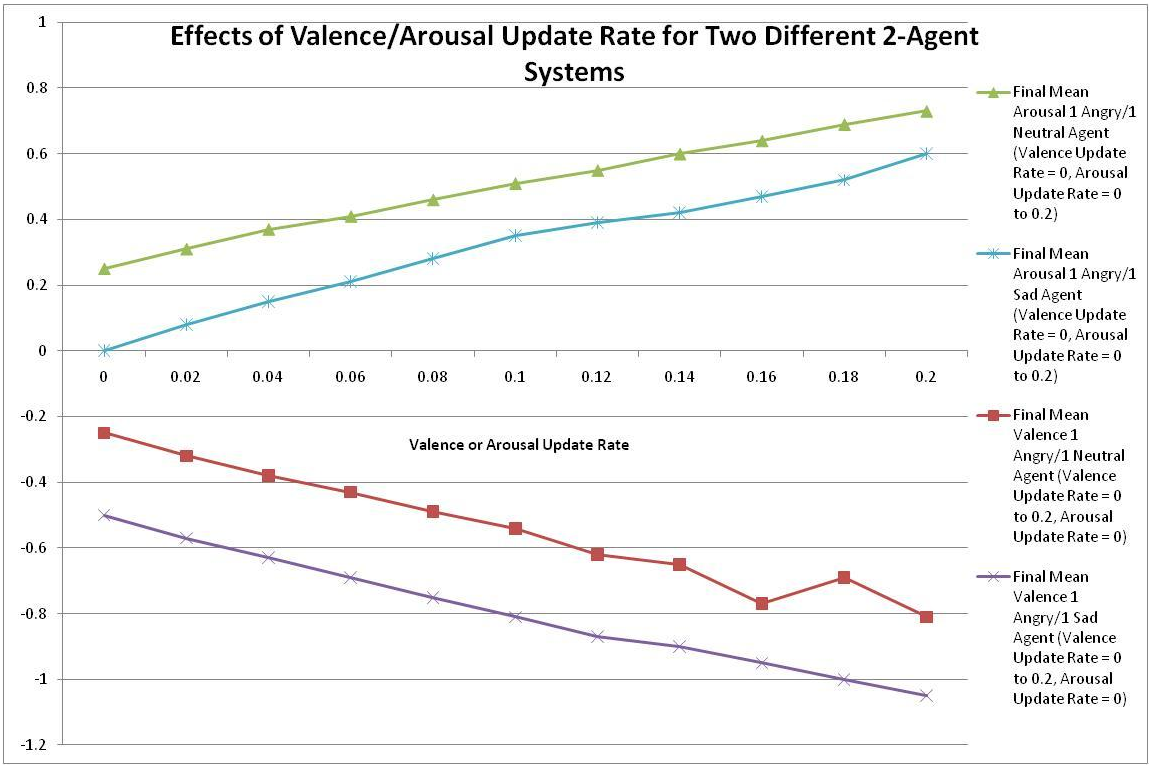

Figure 23: Effects of Arousal and Valence updates on 2-Agent Systems

Now that some insight into the dynamics of the update process in terms of agents has been developed, the tune learning system will be switched back on. To get some idea of the effect of affective update, Figures 25 and 26 show the same multi-agent systems with and without affective update rates contrasting their musical features. Figure 27 shows the effects of affective update on the mean estimated valence and arousal of the final tunes.

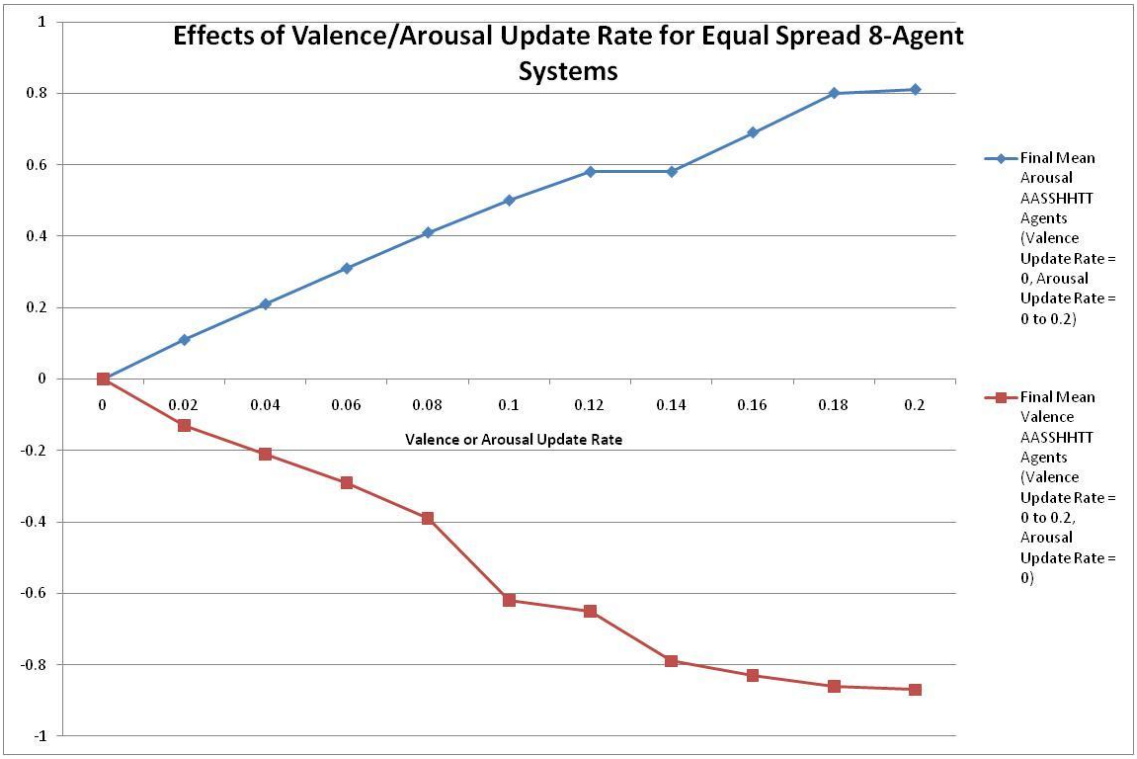

Figure 24: Effects of Arousal and Valence updates on an 8-Agent SSAAHHTT System

Piano roll figure 62A p235 ("SSAAHHTT" tune without affective update) in Appendix 2 is also rebuilt *with* affective update to give Figure 66A p238, showing the resultant tunes. Figure 28 shows the evolution of Agent 1's affective state, together with linear estimate of its tune's affective content. The dotted lines are the valence related values and the dash-dot lines are the arousal related values. It can be seen that Agent 1 starts with affective state of [-0.5, -0.5] ("Sad") and during interactions with Agent 2 (which is "Happy") the agent's valence moves higher (influenced by Agent 2) and its arousal moves lower.

The final element to be added in the affective cycle in this experiment is the Affective Threshold. Figures 20 and 21 in Experiment 3 showed the effect of similarity threshold for an 8 agent equally spread system. Figures 29 and 30 are these same calculations but

with Affective Update values of 0.1 included (note this is for 8 cycles again). For a Similarity Threshold of 1, Figure 31 shows the actual valence and arousal states two of the agents go through over time, together with the estimated affective content of their tune by the IPCS linear affective estimator.

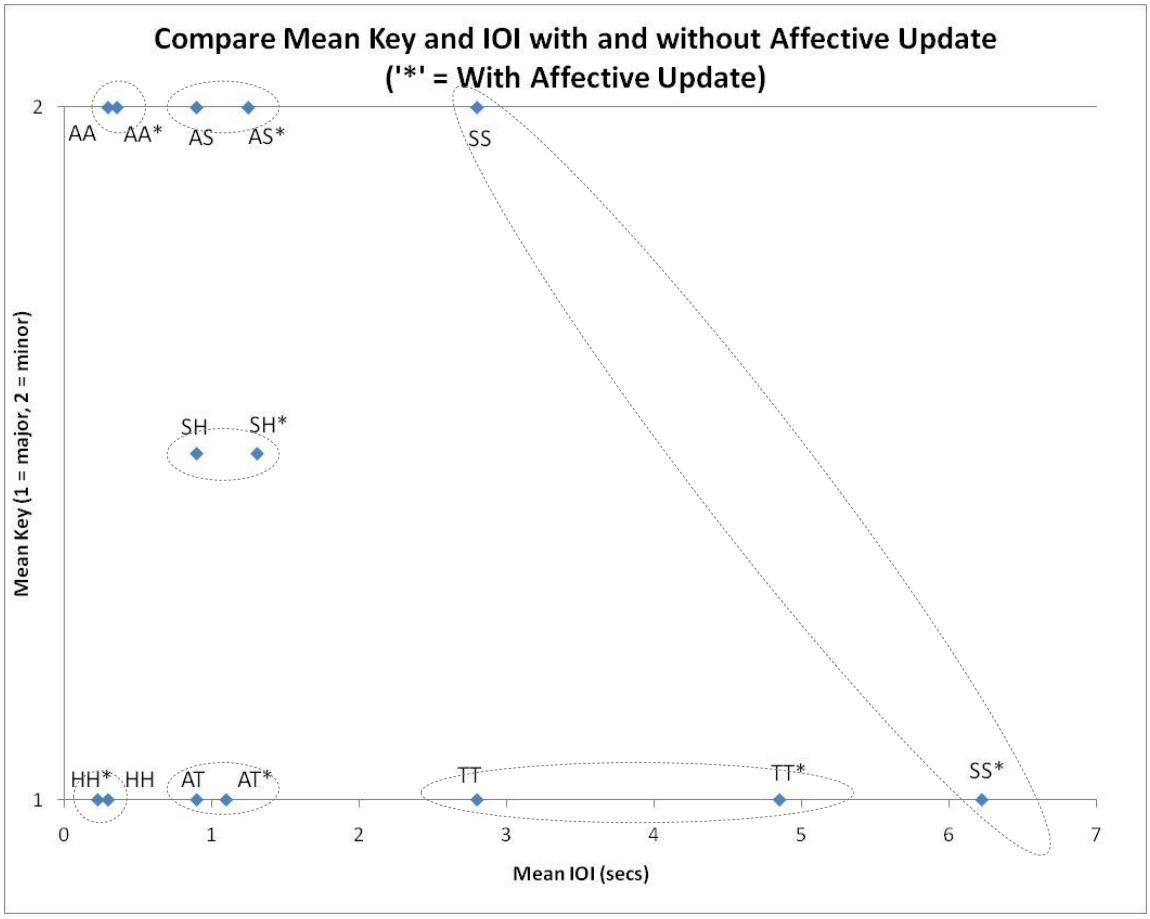

Figure 25: Compare 2-Agent System mean final Key and IOI with and without Arousal and Valence updates (‘*’ With Affective Updates = 0.1)

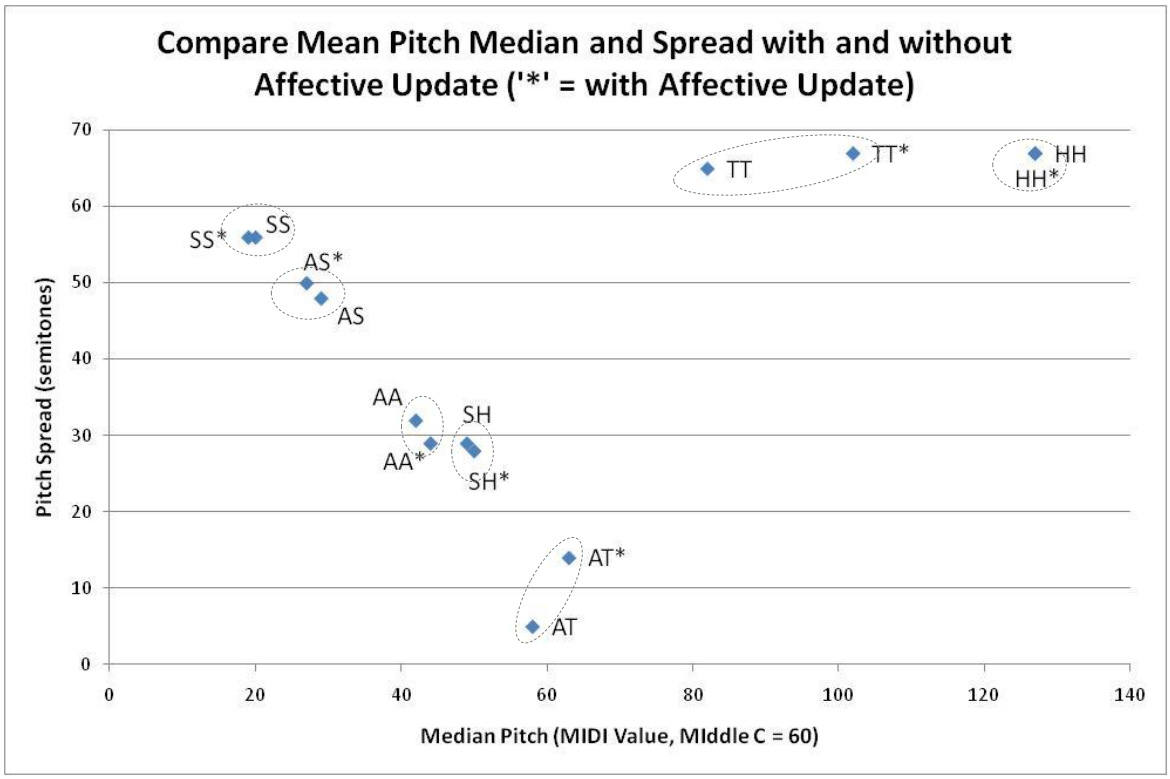

Figure 26: Compare 2-Agent System final mean Median Pitch and Spread with and without Arousal and Valence updates (‘*’ With Affective Updates = 0.1)

It can be seen from Figure 25 that in general the affective update tends to increase the mean IOI. In Table 34A p260 the tune lengths are increased as well – suggesting that the affective influences are bringing agent tunes closer together in features, by bringing agent states closer together. In Figure 31 it can be seen that the state and estimation tend to track each other once the tune lengths get larger. This is presumably because short tunes have too little information the Affective Linear Estimator to use in estimating the affective content of the tune. It can also be seen that the estimator over estimates the magnitude of the affective variables. Piano roll figure 70A p242 in Appendix 2 shows the tune evolution for agents 4 and 6.

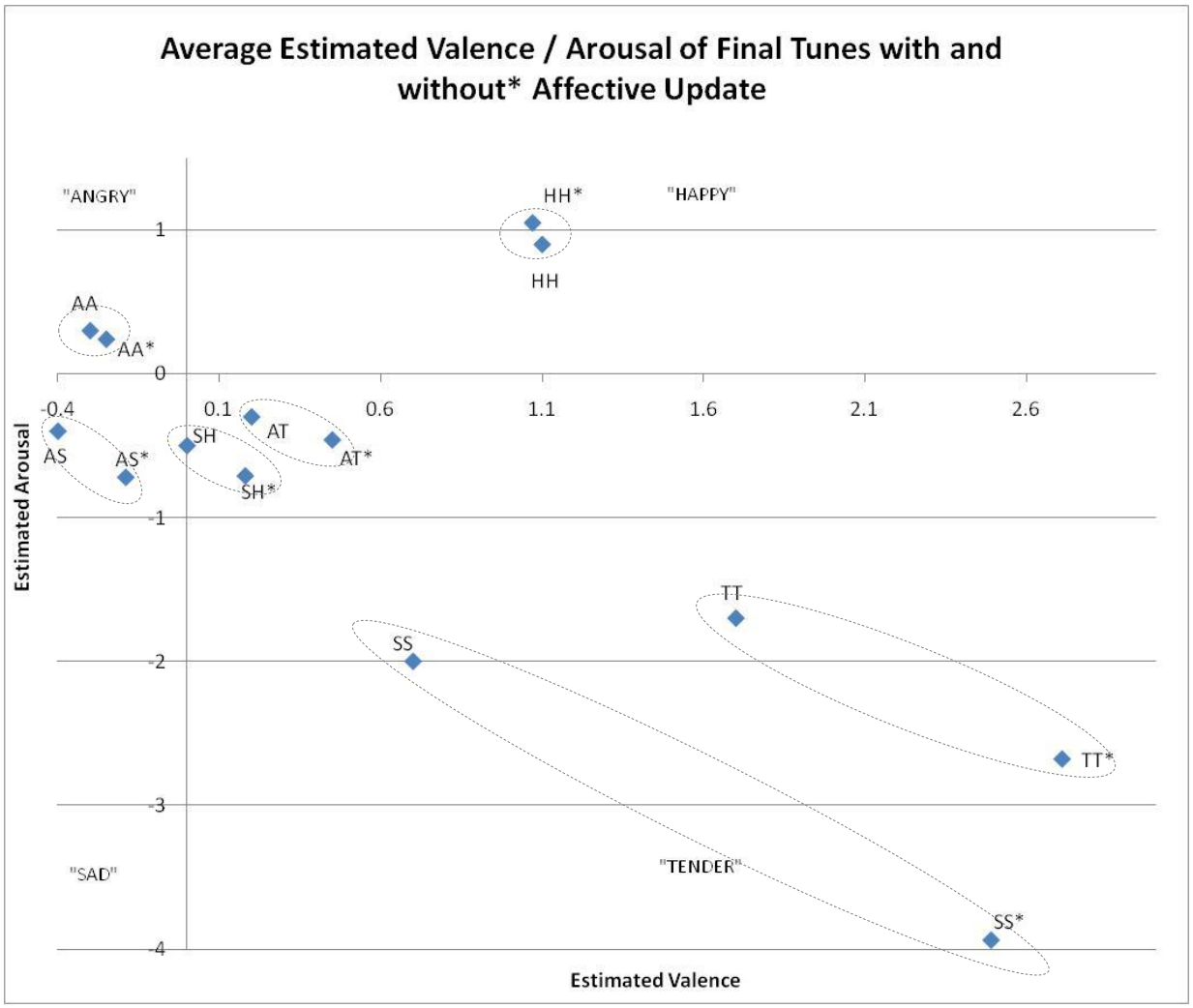

Figure 27: Average Estimated Valence / Arousal of Final Tunes with and without* Affective Update

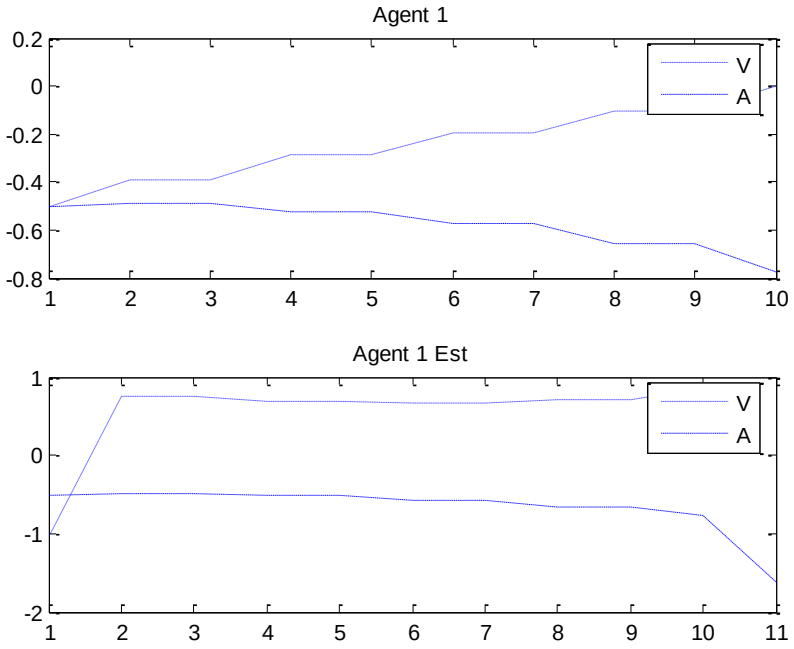

Figure 28: Affective State of Agent 1, and Estimated Affective State of Agent 1's tune

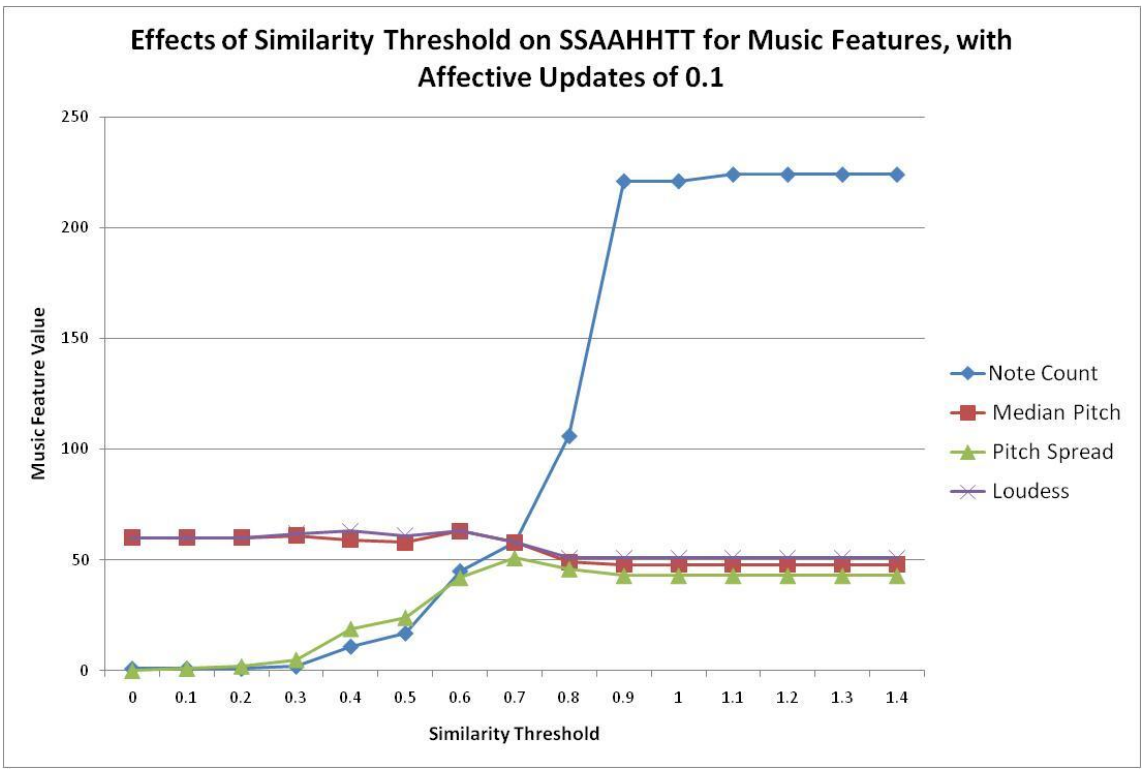

Figure 29: Effects of Similarity Threshold SSAAHHTT Agent System on Music Features with Affective Updates of 0.1

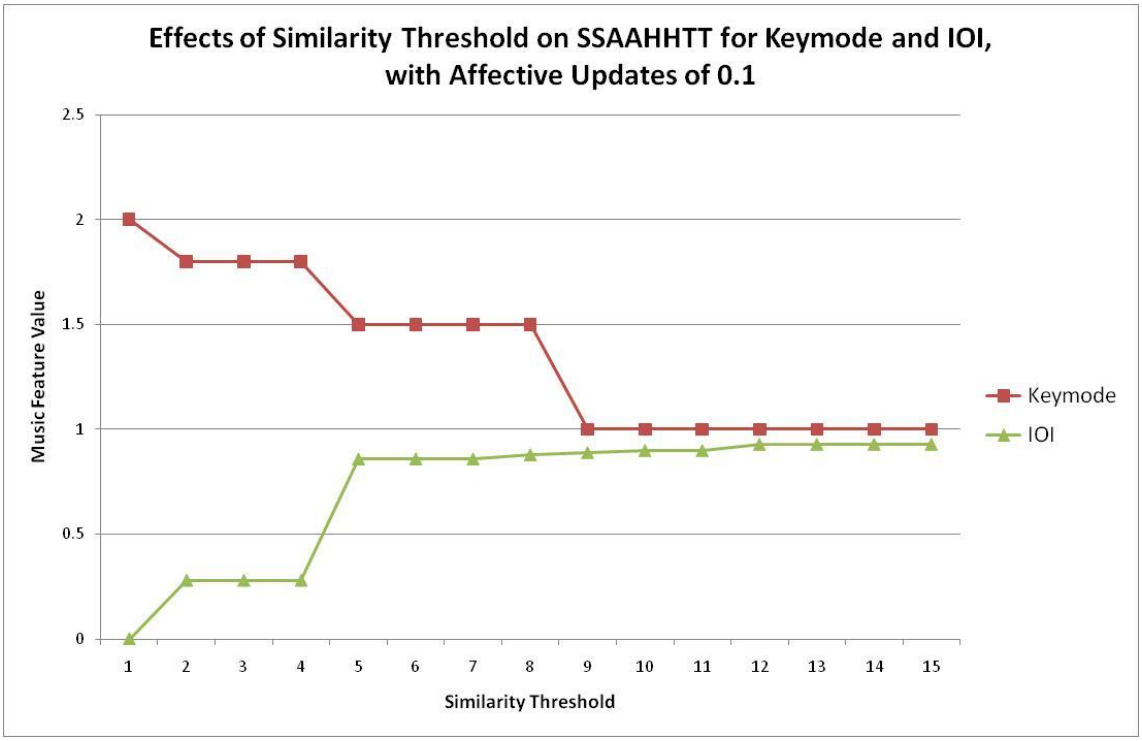

Figure 30: Effects of Similarity Threshold SSAAHHTT Agent System on Music Features with Affective Updates of 0.1

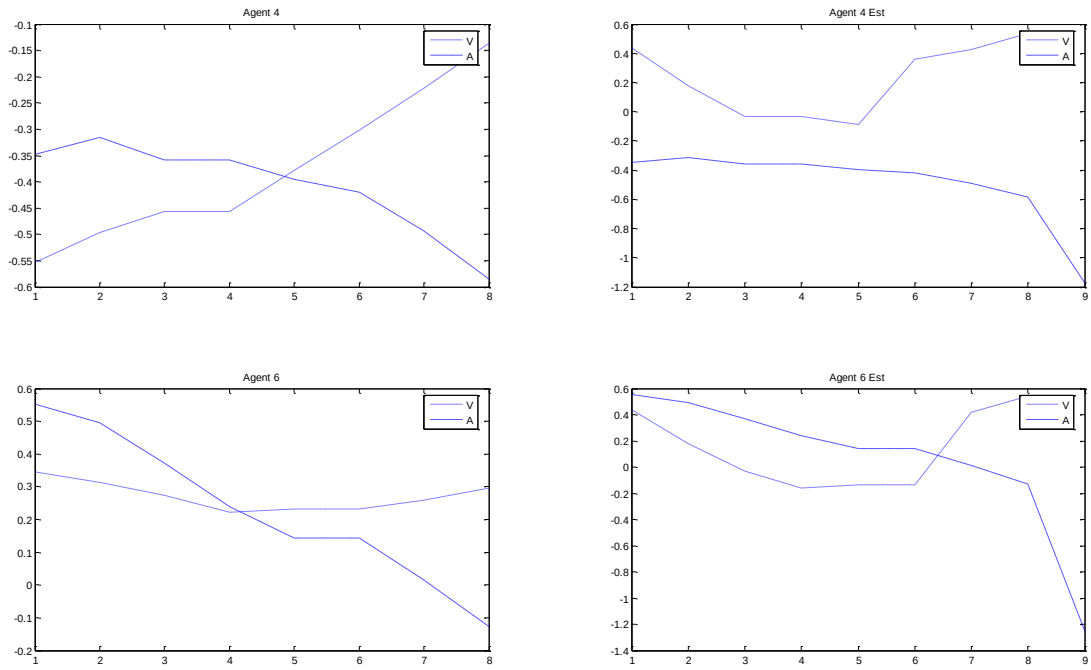

Figure 31: Valence / Arousal Profiles of Agents 4 ("Sad") and 6 ("Happy") Tune, Similarity Threshold 1

The tunes of Agent 4 and 6 in Figure 70A p242 are more similar than expected – possibly due to the higher similarity threshold and the affective influence the agents have on each other. The tunes are also repetitive. This raises the fact that there has been no attempt in the experiments so far to focus on "interesting" tunes – parameters have been acceptable if they have highlighted the dynamics of IPCS. A key example of this is that the affective influence on pitch was so large (0.5) that the pitch ranges are multiple octaves over just a few notes. As an example of a parameter set which was selected for "musical interest" consider Piano roll Figure 71A p243 in Appendix 2. In an attempt to introduce more diversity into the population, the similarity threshold was lowered from 1 to 0.5, and since this would decrease the number of expected interactions, the number of cycles was increased to 12. Furthermore the influence of affective state on pitch, timing and loudness influence was reduced from 0.5 to 0.1. This leads to a population with tunes such as those shown in Figure 71A p243. It can be seen there is greater diversity in the tune features. For example, Agent 1's tune is a significant transformation of Agent 3's tune, and Agent 5's tune is a combination of significant transformations of a similar tune to Agent 1 and 2. Agent 7's tune has been built into something much longer. Figure 72A p244 in Appendix 2 shows the loudness for Agent 5's final tune. (Note that there is also an example of a complete computer-aided composition using IPCS in Appendix 4).

4.5 Experiment Set 5 – Effects of Interaction Coefficient

Another factor will now be introduced: Interaction Coefficient. As has been mentioned, the Interaction Coefficient process is designed as an attempt to regulate tune growth in such a way that certain agents will become "tune provider" agents and some will become "tune receiver" agents. This is implemented by causing Agent B to increase its

Interaction Coefficient measure of Agent 1 if it learns a performance from agent 1; and that increase is proportional to the number of notes in Agent 1's performance. On top of this, in interactions, Agent 2 will only perform its tune to an Agent 3 if its Interaction Coefficient measure of Agent 3 is greater than some proportion of its mean Interaction Coefficient of all other agents (as in equations (9) and (10) in Chapter 3). So with Interaction Coefficient thresholding implemented, Agent 3 will generally only provide performances to agents which it values above average – i.e. which in the past have provided the performances with the most notes to Agent 3.

To display the effects of Interaction Coefficient thresholding, a series of runs was done to investigate the effect on musical features, and runs were done to demonstrate the segmenting of the agent population into "providers" and "receivers". Table 36A p261 in Appendix 3 gives a first indication of how agent tunes are affected by the Interaction Coefficient Threshold parameter. In this case, an 8 agent system was used with equally spread agent initial affective states, 300 memory size, 32 cycles, affective similarity threshold of 1, pitch update rate of 0.1, and IOI and loudness update rates of 0.5; valence and arousal update rates were set to 0.1. The Interaction Coefficient update rate was set to 0.2; this is the rate at which an agent's Interaction Coefficient Measure of other agents is updated.

Table 36A p261 in Appendix 3 shows the results. It can be seen that Pitch and Pitch Spread vary little, hence they are not plotted. Keymode and IOI are shown in Figure 32. A new measure is added to the results in Table 36A in Appendix 3 and Figure 34 - Equality. Equality is the coefficient of variation of the number of tunes received by agents, divided by the product of cycles and the number of agents (as a normalising denominator). Given the same average final number of notes, an experiment where the number of tunes received varies less between agents, will have a greater equality

measure than an experiment where the number of tunes received by agents varied a great deal. So if Interaction Coefficient divides agents into “givers” and “receivers”, then it would be expected to affect equality in some coherent way.

From Figure 34, it can be seen that with an Interaction Coefficient Threshold of 0 the maximum equality is calculated. As the Interaction Coefficient Threshold increases, this equality falls consistently. In Figure 33 it can also be see that tune average lengths generally drop (though in Table 36A p261 in Appendix 3 the spread of tune lengths on average tends to increase). To look into the effects of Interaction Coefficient in more detail one particular run is selected, Interaction Coefficient Threshold = 0.9; the run with the lowest equality. Piano roll Figure 73A p245 shows the final tunes for all the agents in this run. Figure 35 shows the mean Interaction Coefficient each agent has for every other agent, over all 32 cycles.

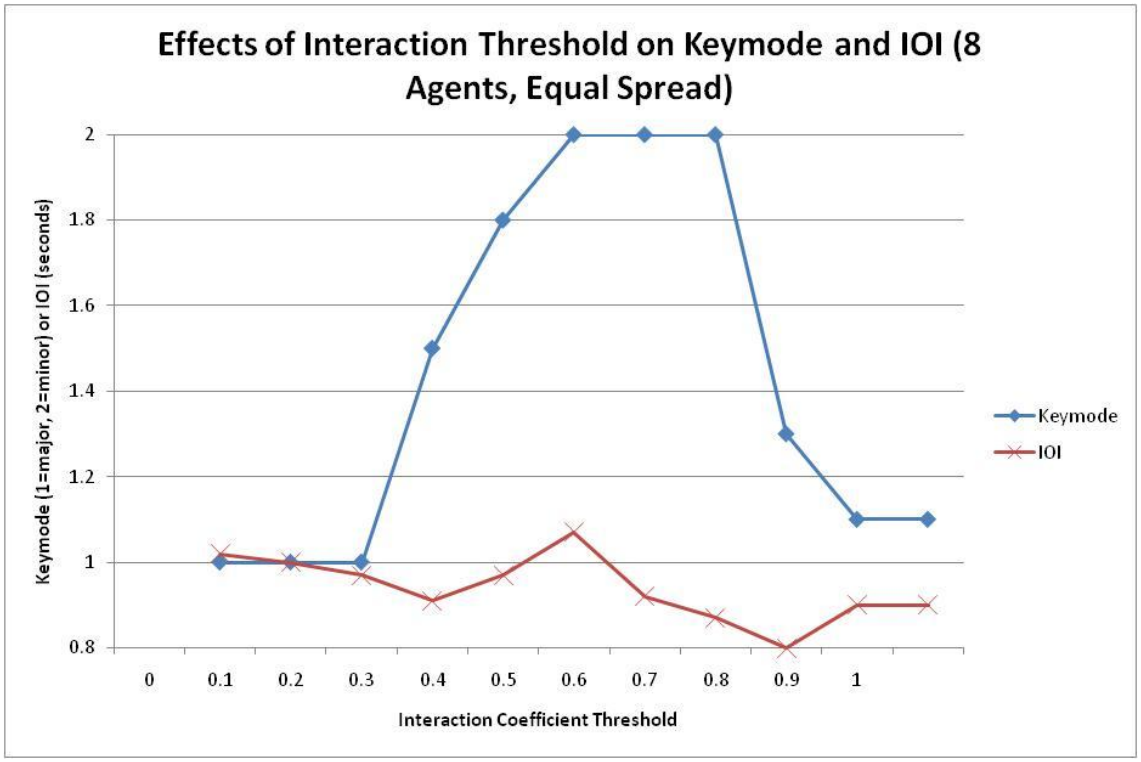

Figure 32: Effects of Interaction Threshold on Keymode and IOI (8 Agents, Equal Spread)

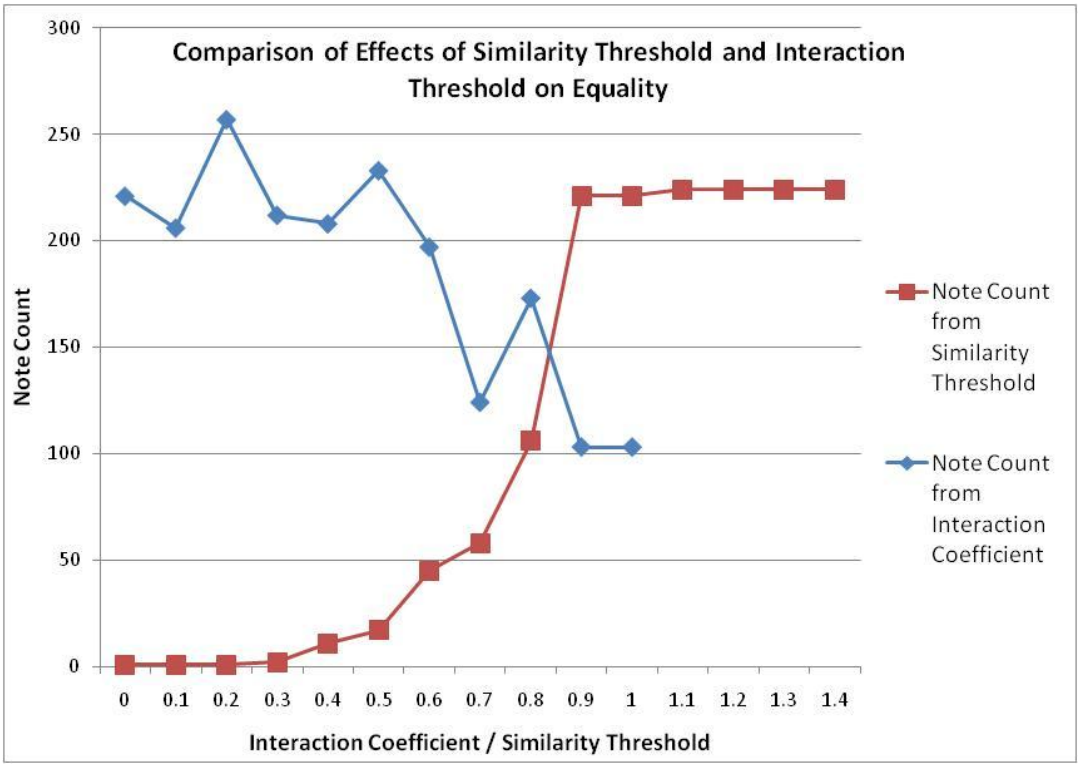

Figure 33: Comparison of Effect of Similarity Threshold and Interaction Threshold on Note Count

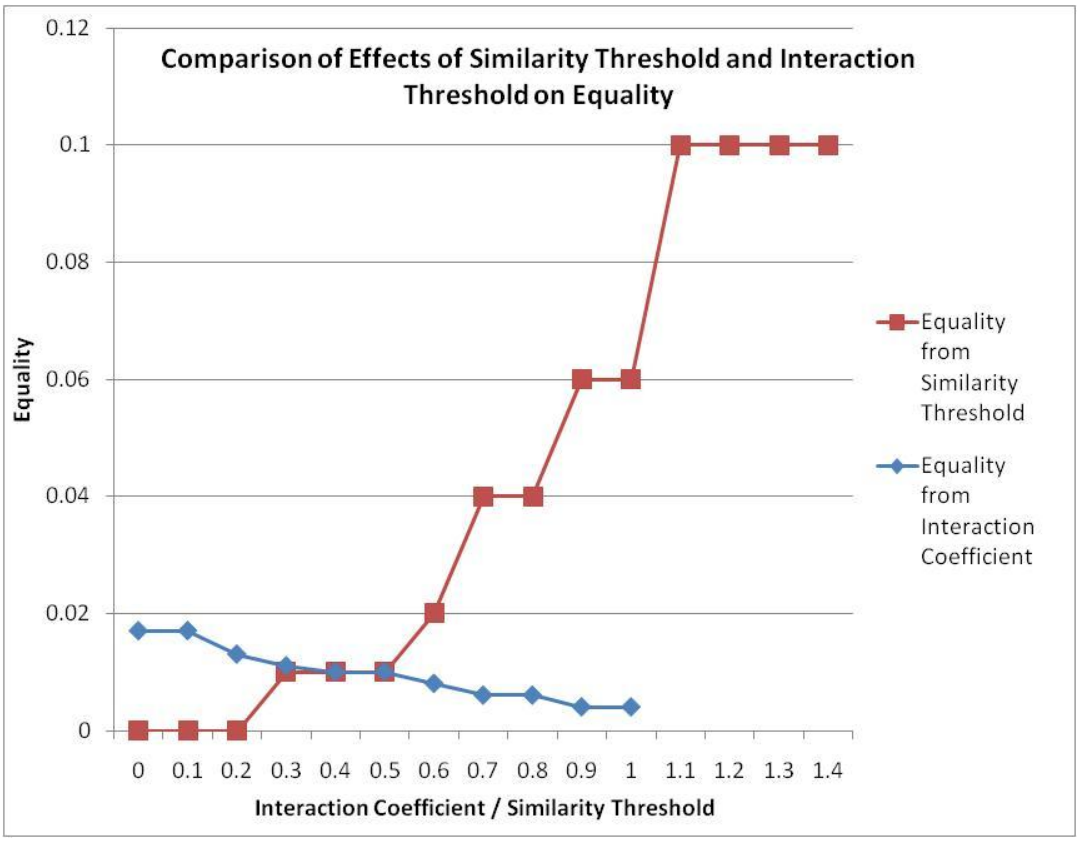

Figure 34: Comparison of Effects of Similarity Threshold and Interaction Threshold on Equality

Figure 34 also shows the effects of similarity threshold on equality. It may seem at first that Interaction Coefficient and Similarity Threshold are doing the same thing – but they are not. Looking at Figure 33 it can be seen that to reduce Equality to levels as low as those found for higher Interaction Coefficient thresholds, there is a drastic drop in mean tune length. This is because Similarity Threshold focuses on affective states, and Interaction Coefficient on tune length.

The number of notes the agents 1 to 8 have is respectively: 291, 102, 102, 102, 102, 102, 18, and 5. Looking at Figure 35, it can be seen that this relates to the Interaction Coefficient – the higher an agent's final Interaction Coefficient the higher its note count. Figure 36 shows a two-dimensional interaction diagram giving some idea of the evolution of which agents have the highest interaction coefficients for each other. The more intense and more red the colour, the higher the interaction coefficient, the more blue, the lower. So for example in cycle 2, agent's 2 to 6 have a good opinion of agent 1; and by cycle 20, the interaction coefficient patterns have pretty much settled down.

Table 13 shows the average number of tunes exchanged over cycles across the whole population. The pattern of interaction can be seen from another perspective in Table 14. Table 14 shows in each cycle which agents an agent receives tunes from. So for example in cycle 1, agents 2 to 6 receive tunes from agent 1. In cycle 2 agent 1 receives a tune from agent 2, but no other agents receive tunes. In this table it can be seen that the lower Interaction Coefficient agents tend to give out tunes, while the higher Interaction Coefficient agents tend to receive tunes. The lower numbered agents have higher Interaction Coefficient because of the ordering of agent interaction in each cycle. The lower numbered agents will be receivers first, and have a chance to build up the size of their tunes. Then when they become performers (givers) the lower agents will receive large tunes from them and their Interaction Coefficient will increase as a result.

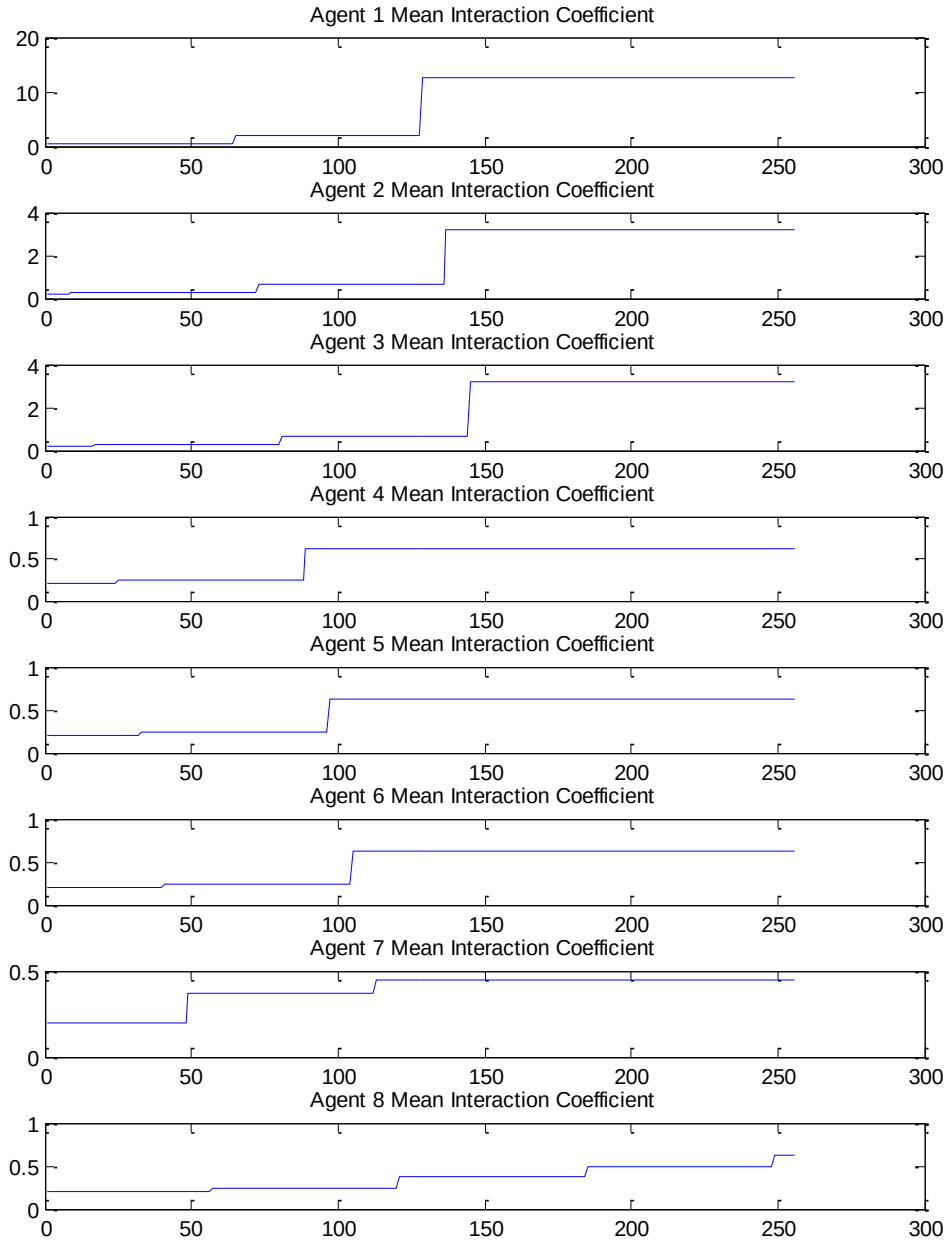

Figure 35: Change in mean Interaction Coefficient leading to tunes in Figure 73A

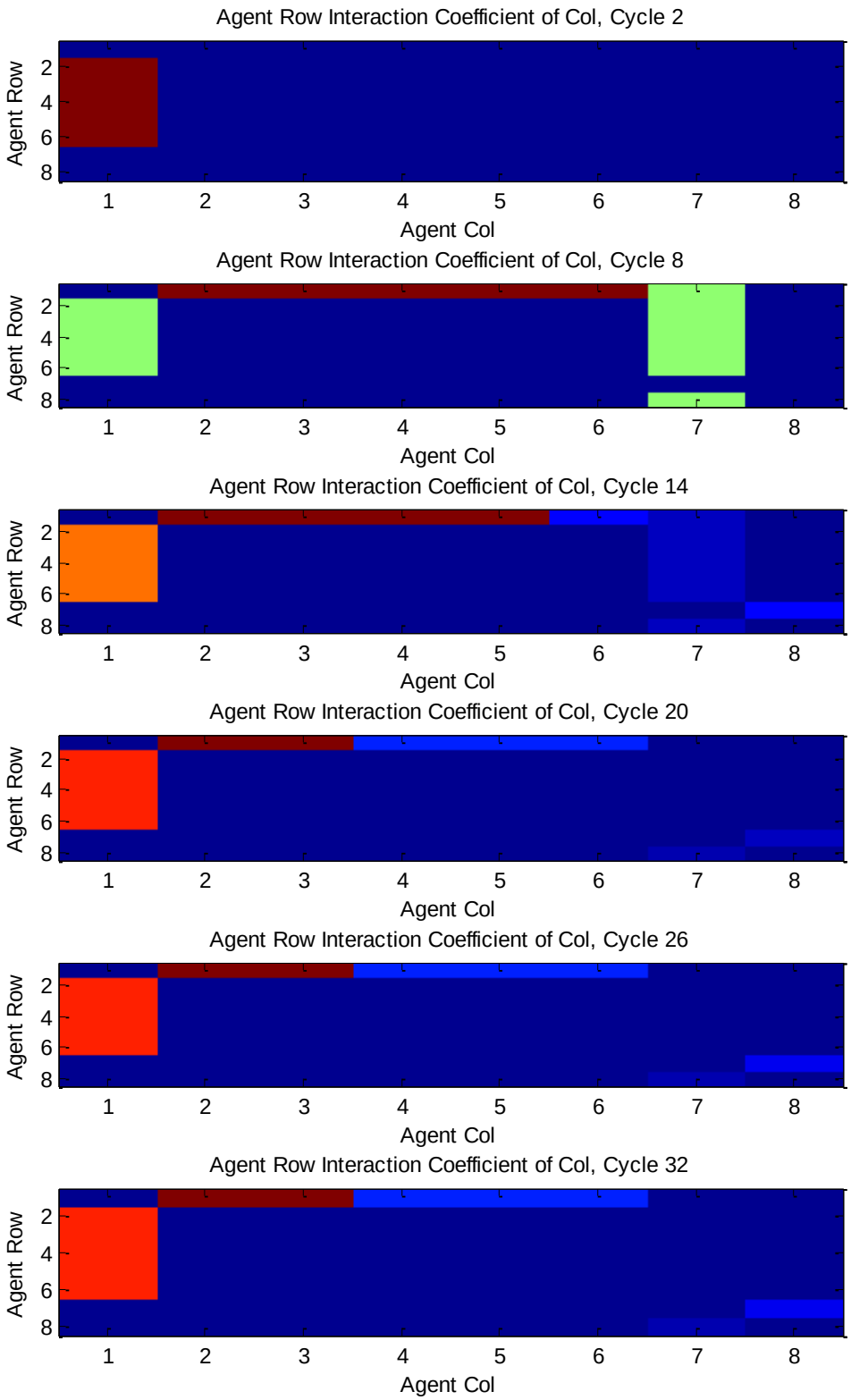

Figure 36: Interaction graph for 8 agent System

Agent		Total Count	Mean Interaction Coefficient
1	*in*	13	12.7
	out	15	
2	*in*	4	3.2
	out	3	
3	*in*	4	0.6
	out	3	
4	*in*	4	0.6
	out	2	
5	*in*	4	0.6
	out	2	
6	*in*	4	0.6
	out	2	
7	*in*	4	0.4
	out	8	
8	*in*	2	0.6
	out	4	

Table 13: Agent Tune Exchanges for 8 Agents, 32 cycles

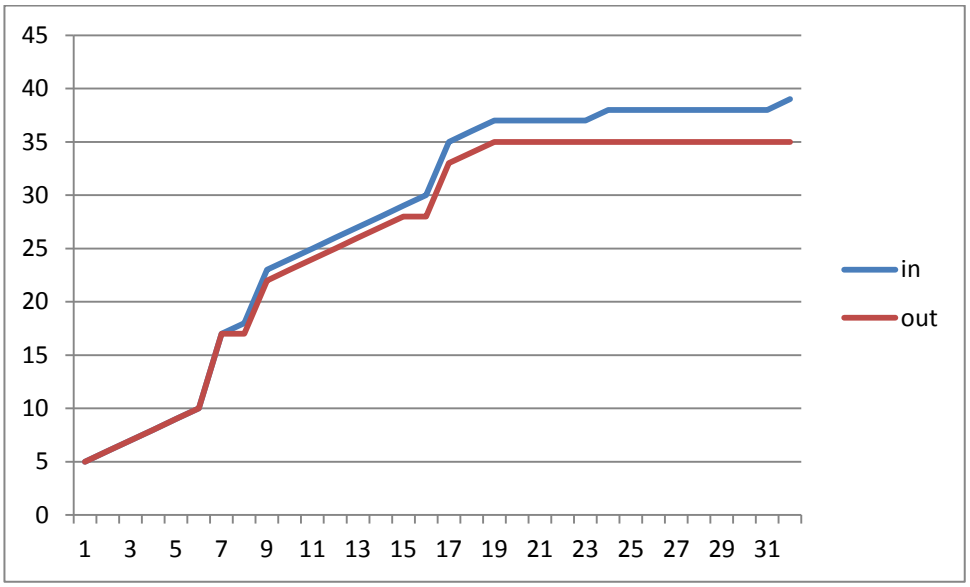

Figure 37: Cumulative count of tunes given out and received across all agents over 32 cycles

This can be further clarified by adding a new mode to IPCS operation: e-mode. In e-mode, rather than in each cycle agents performing sequentially, the even-numbered are performed to first and then the odd numbered. This produces the Interaction Coefficient pattern in Figure 38, and final tune lengths of 173, 267, 226, 164, 280, 226, 107, and 195. The final interaction coefficients are: 11.2, 3, 3.8, 5.6, 3.4, 8.5, 3.6, 3.3. Figure 38 contrasts strongly with that of the non-e-mode results in Figure 36.

The question arises – how does the "interaction coefficient structure" relate to the tune structure here? Intuitively it is fairly clear – but this can be clarified by considering two elements. Table 14 shows the progression of tunes received in the multi-agent system (the non-e-mode version earlier).

	Cycles																																
Agent	**1**				**5**					**10**					**15**					**20**					**25**					**30**		
1		2	3	4	5	6				2	3	4	5	6				2	3													
2	1						7	8									1															
3	1						7	8									1															
4	1						7	8									1															
5	1						7	8									1															
6	1						7	8									1															
7								8									8							8								8
8							7								7																	

Table 14: Agent tunes received by agent for 32 agents, non-e-mode

This shows the expected behaviour – agents with the shortest tunes (Agents 8 and 7) - the "motifs" - are tune providers. So agents 8 and 7 only receive tunes from each other, but provide tunes to agents 2 to 7. Agents 2 to 7 have tune lengths "in the middle" and provide tunes to the longest highest Interaction Coefficient agent – Agent 1.

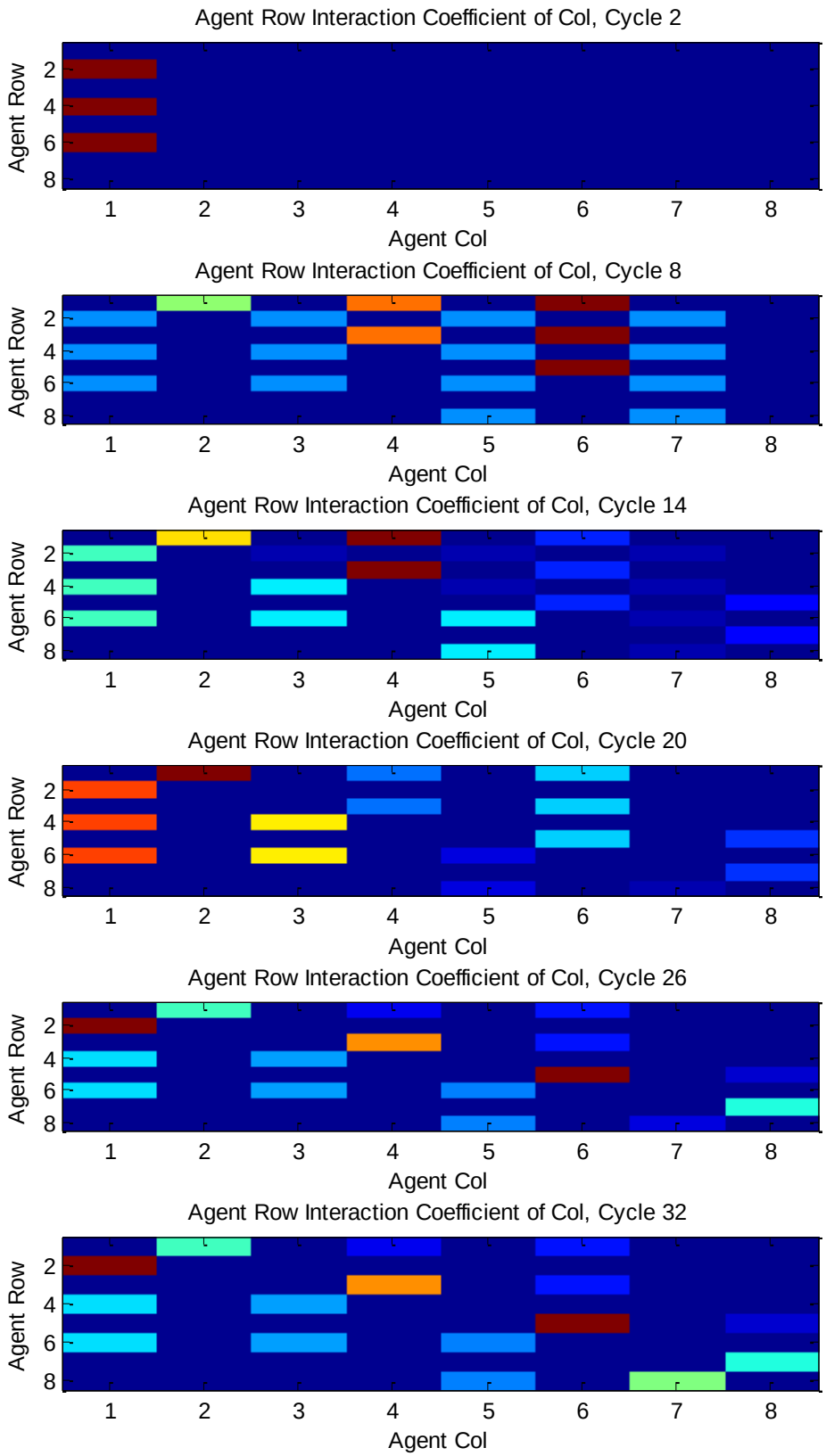

Figure 38: Interaction Coefficient Pattern, e-mode, for 8 agent systems

Thus the "interaction structure" has Agent 1 at the top, Agents 2 to 6 in the middle and Agents 7 and 8 are the bottom. It is important to note that although Agent 1 is most frequently receiving tunes it does give out tunes twice in the 32 cycles: very early on, and then halfway through. So the interaction structure is not one way, but is still heavily weighted toward a kind of hierarchy, which impacts the hierarchy of the music. To see this we first define a notation

X_i = Agent X's performance at the end of cycle i

Using this notation we can represent any agents tune. For example

Agent A's Stored tune = $B_jC_kD_l$

means that agent A's tune is made up of 3 sub-tunes: starting with a tune provided by agent B during cycle j, followed by tune provided by agent C during cycle k, and then followed by a tune provided by agent D in cycle l. Now consider that by Table 14 Agent 1's final tune could be written as:

$1_02_13_24_35_46_57_62_93_104_115_126_132_173_18$

Remember that an agent's tune can vary over different cycles – e.g. 3_{18} is not the same 3_2. Because the MAS is a closed deterministic system, all tunes in this structure are the result of a transformation on another agent's tune. So for example:

$2_1 = 2_01_0$'

$3_2 = 3_01_0$''

…

$7_6 = 7_01_0$'''

where the primes (‘) represent transformations on Agent 1’s tune due to Agent 1’s affective state at the time. In the next round of tunes being given to Agent 1 this gives:

$$2_9 = 2_1 7_7' 18_8' = (2_0 1_0') \, 7_7' 18_8'$$

…

This expansion can be continued until there is a full description of Agent 1’s tunes based on the way in which the tune grows. This description will show the building structure of the tune. (It will not necessarily show the *perceptual* structure of the tune – this is not claimed, but it will show how the tune was built from the motifs and phrases etc of other agents). This structure is clearly a function of the agent interaction hierarchy, and as has been seen this hierarchy is strongly influenced by the Interaction Coefficient functionality. Hence this supports the idea that the Interaction Coefficient provides a non-affective-based method for generating coherent hierarchical structure in the tunes, based on the emerging interaction structure in the multi-agent system.

4.6 Experiment Set 6 – Effects of Expressive Performance

In these experiments the effects of expressive performance on the system are investigated. Experiments were run with 8 agents with equal spread of initial affective states, and the standard initial parameter values already discussed. The result for different k-values (for 10 cycles, with the last section’s parameters, and seed duration 0.5) are shown in Figures 39 and 40 (detailed in Table 38A (p262 Appendix 3)). Note that the Estimated Tune Affective State columns were left out of Table 38A, as the k-Value had no effect. In an attempt to see how much the expressive performance elements can change agent interactions, experiments were done with a larger number of cycles in Table 39A p262 (for 24 cycles). It should be noted that the k-value is *per*

interaction – in other words expressive performance with that k-value is applied in every interaction.

And a different k-value will be applied a different number of times depending on how many times a sub-tune is exchanged. It can be seen that the more that the expressive performance deviations are applied (i.e. the larger the number of cycles), the greater their effect on the interaction of the agents and on their final note features. To show the effect on the actual tunes four tune and loudness plots are done: Figures 74A to 77A p246-7 in Appendix 2, of Agent 1's tune after 24 cycles with k = 0 and k = 0.2.

One of the expected more useful aspects of such an approach is the emergence of the expression of compositional structure. As mentioned in Chapter 3, a survey of previous CSEMPs shows two frequent commonalities:

1. Formal musical analysis - CSEMPS often base their performances on the hierarchical structure of the music thus requiring a pre-analysis, sometimes by a musicologist.

2. Hierarchical combination of microstructure deviations – Most hierarchical systems generate the final tempo and dynamics expressive deviations by combining separate multipliers calculated for each level of the hierarchy.

Hence a significant amount of CSEMP effort is often in analysing the musical structure of the score/audio. However, as a result of the way in which expressive performance is applied in IPCS the hierarchical structure is actually expressed by the performance elements.

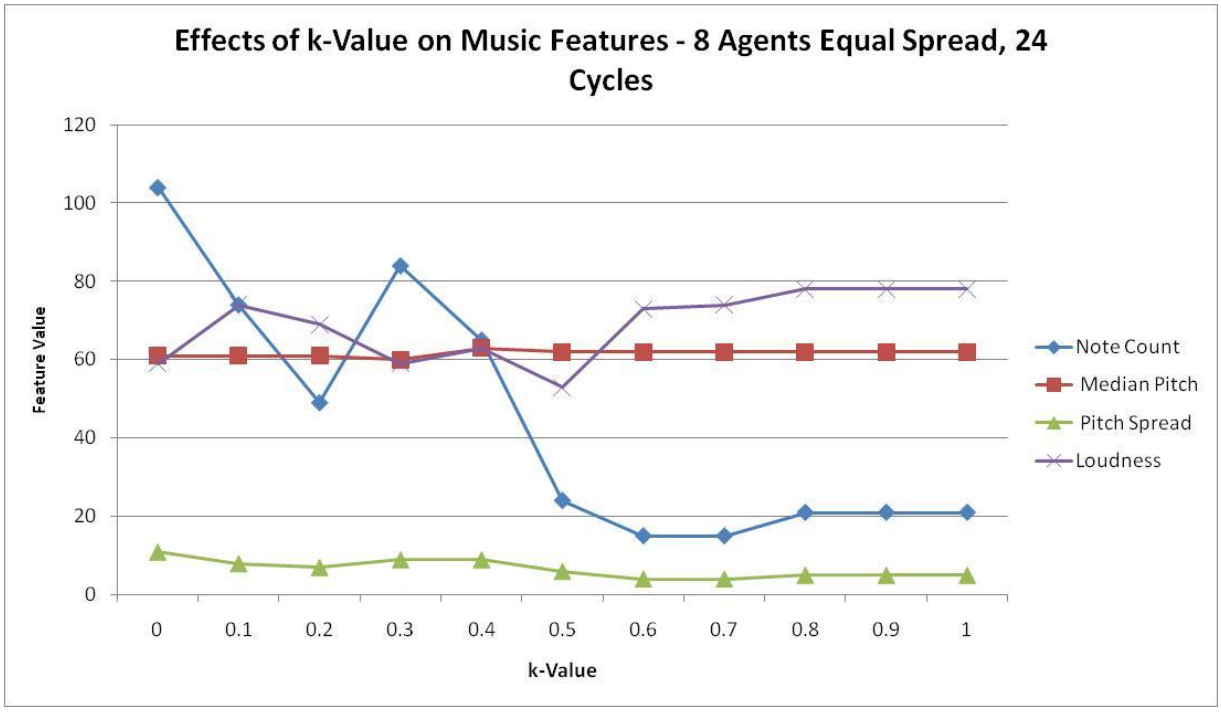

Figure 39: Effects of k-Value on various Music Features

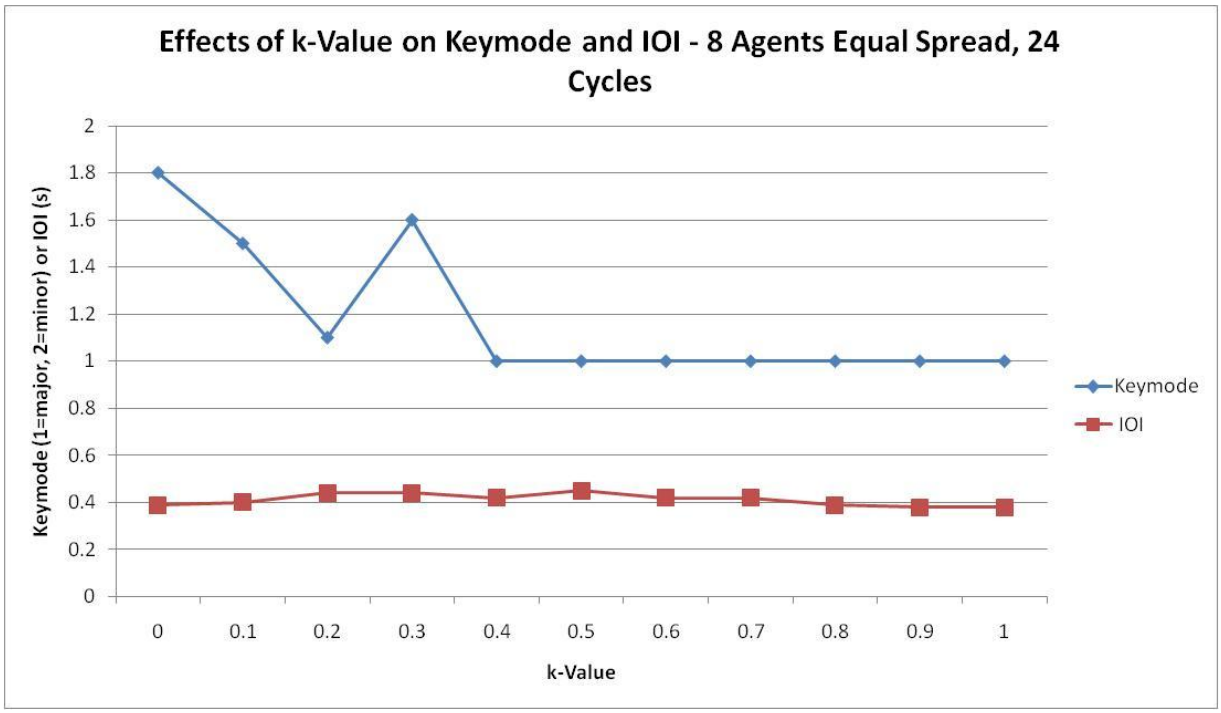

Figure 40: Effects of k-Value on IOI and Keymode

This can be understood by referring back to Experiment Set 4.5 where Agent 1's tune could be written as:

$1_0 2_1 3_2 4_3 5_4 6_5 7_6 2_9 3_{10} 4_{11} 5_{12} 6_{13} 2_{17} 3_{18}$

It was then described how the various subtunes were built up:

Agent 2's tune in cycle 1 = $2_0 1_0$'

Agent 3's tune in cycle 2 = $3_0 1_0$''

…

Agent 7's tune in cycle 6 = $7_0 1_0$'''

and:

Agent 2's tune in cycle 0 = $2_1 7_7' 18_8'$ = $(2_0 1_0')\ 7_7' 18_8'$

When k>0 each time a tune was performed by an agent, phrase punctuation is applied and for some affective states a Phrase Arch was placed over that performance. The longer the tune, the wider the phrase arch. Furthermore the starts and ends of a currently performed phrase arch by agent A may coincide with the start or end of a previous phrase arch imposed over part of A's tune learned earlier. As a result phrase punctuation and phrase arches combine hierarchically as agents interact, and more significant boundaries of tunes (which are maintained during interactions) will be repeatedly and coherently emphasised by the recursive imposition of performative phrasing. This leads to the phrasing expressing the grouping structure of the way the notes are built up. For example suppose we write the performance of tune x as *P(x)*, and the concatenation of tune x followed by tune y as *[x y]*; and let Agent A plays performs tune 2 to Agent B (who currently has a tune called 1), then Agent B will store tune:

[1 P(2)]

where *P(2)* is an expressively phrased version of tune 2. If Agent B performs this back to Agent A (imposing another phrase expression) then Agent B will have:

[2 P([1 P(2)])]

So here *P([1 P(2)])* is an expressively phrased version of the concatenation of tune *1* and tune *P(2)*. And then if Agent A performs back to Agent B, Agent B will have:

[[1 P(2)] P(2 P([1 P(2)]))]

So each time a tune is swapped back, a new phrase expression is added over all the "lower" tunes, and the borders of this phrase expression will emphasise certain tune and sub-tune boundaries. An actual example is shown now. To explicate this process, some adjustments had to be made to the system initialisation. If the normal 8 agents with multiple affective states was used, then it would not be clear how the combination of expressive performance and composition was working. This is because expressive phrasing works differently for different emotions – for example anger imposes a phrase arch with a negative k-value, and happiness and tenderness do not use phrase arch at all.

To simplify things, a 2 agent neutral system was run for 10 cycles with a k-value of 0.2 and maximum notes 150 (with Affective States Update, Similarity Thresholding and Interaction Coefficient not activated). The resulting tempos for Agent 1 for cycles 5,6,7,9, and 10 are shown in Figures 78A, 79A, and 80A p248-9 in Appendix 2. (No expressive performance deviations are added before cycle 5 due to the fact that agents need at least 2 notes in their tune to add expressive performance). Agent 1's final tune is shown in Figure 81A p249; and its loudness curve is given in Figure 82A p250. The growth of the expressive curves can be described as follows. Firstly Agent 1 adds Agent 2's expressive performance to the end of its own unexpressive tune giving Figure 78A

p248. This gets performed back to Agent 2 who then performs it back to Agent 1 resulting in Figure 79A p248.

A key thing to note here is the size of the deceleration between the tune added to Agent 1 in Figure 79A p248 is larger than the decelerations within the sub-tunes. These deceleration sizes actually capture the structure of the tune so far (from a generative point of view). If we call the two tunes in Figure 79A “phrases” and call the sub-tunes “motifs”, then the separation between “motifs” is expressed by decelerations, and the separation of phrases by the largest deceleration. This pattern continues into Figure 80A p249. Here a new performance is added to the end of Agent 1’s tune. If this added performance is labelled a “theme” and the tune which was in Agent 1’s store in Figure 79A p248 is called a “theme” as well; then the decelerations between “themes” are greater than the decelerations between “phrases”. Thus the expressive tempo does expresse the generative structure. The behaviour of the loudness is due to the fact that MIDI loudness is an integer parameter and so is rounded off.

This can be compared to one of the most common observations in music performance studies (Clarke 1998; Clarke 1991; Shaffer and Todd 1987): performers slow down at boundaries in a musical piece, with the amount of slowing down being greater for the more significant the boundary. It is clear from the process of IPCS tune growth that such an expression will always occur (though to make it noticeable with compositional elements added may require increasing the global k-value). This is the main demonstration of the advantage of such an approach. It should also be mentioned that this approach requires no musical analysis, and therefore could work with musical types for which analysis methods are more limited (Clarke 1991). On the other hand it should be mentioned that the expression is of the generative structure, not the perceptive one. In other words a human may hear different note groupings to those actually implicitly

assumed by IPCS. This expressive structure can be heard in the example computer-aided composition in Appendix 4.

4.7 Experiment Set 7 – Listening Experiment

The element now to be examined is the effect of specifying agent initial affective states on the subjective experience of affective communication by the music. In this experiment, the affective parameters were used to perform an initial examination of how starting settings in affective states effect the emotional perception in the listener. Performances were generated with agents with different initial affective states, the resulting perceptions were garnered from test subjects. The performances were played through the Grand Piano simulation in the sequencer software Reason (Carlson et al. 2003). Table 15 shows the initial affective states for the listening experiment. The lists of single letters in the second column refer to the initial affective states of agents in the group. For example "AAAAAAAA" (either A's) means that an 8 agent population was used, all of them with initial state of "Angry", i.e. valence of -0.5 and arousal 0.5. The order column is the order in which tunes were presented to test subjects.

Tunes	Initial Affective State	Order
1	AAAAAAAA	1
2	AAAAAASS	5
3	AAAAAAHH	9
4	AAAAAATT	13
5	SSSSSSSS	2
6	SSSSSSAA	6
7	SSSSSSHH	10
8	SSSSSSTT	14
9	HHHHHHHH	3
10	HHHHHHAA	7
11	HHHHHHSS	11
12	HHHHHHTT	15
13	TTTTTTTT	4
14	TTTTTTAA	8
15	TTTTTTSS	12
16	TTTTTTHH	16

Table 15: Initial Affective States for Listening Experiments

There is always a clear majority of agents (at least 6 out of 8) with the same initial affective state. So one example of a mixed affective set-up is “AAAAAASS” means 6 agents with valence/arousal -0.5/0.5 (“angry”) and 2 agents with valence/arousal -0.5/-0.5 (“Sad”). For each quadrant of the 2D affective space, there are 4 tunes with a clear agent majority in that quadrant. The mixed states provide two elements: (1) examining if the user’s perception is still skewed toward the agent majority; and (2) further exploring the parameter space of the agent’s affective behaviour to see what sort and shape of tunes emerge.

An 8 agent population was used, run for 10 cycles, with similarity threshold of 1, and a k-Value of 0. The valence and arousal update rates were set to 0.001, maximum tune size was 150, seed note duration 0.5. The Interaction Coefficient update rate and threshold was set to 0, as the desire was to focus on the perception effects of the affective cycle, rather than the structure elements. The tune presented to listeners was Agent 6’s tune at the end of all cycles. The reason for such a low update rate is that previous experiments showed that agents updating their affective states at a rate of 0.1 lead to quite large changes in affective state and thus to large changes in musical features. A more subtle method for regulating these large changes has not yet become clear, hence instead the update rate is set very low for the listening experiments. However the parameter was not to 0, and hence has some effect, particularly given the exponential interaction effects of the agents (for example in an exponential growth equation $(1+0.001)^{10} = 1.01$).

4.7.1 Generated Tunes

The observed features of the tunes generated for the listening experiment gives an insight into the function of IPCS. Bearing in mind that all other parameters were the same, and it was only the agent’s initial affective state that was changed, it was found

that a wide variety of features are generated purely through adjusting these two elements. A web page has been set up for this thesis and five of the tunes used can be found on that page (http://cmr.soc.plymouth.ac.uk/alexiskirke/ipcs.html). Figures 41 and 42, and Table 40A in Appendix 3 p263 show the features. The figures and table in themselves highlight some key aspects of IPCS. Because of the low value for affective update rate, the table can also be used to examine the effects of different initial affective states. The first thing to note is that whenever 6 of the 8 agents start in particular valence / arousal quadrant they remain there after 10 cycles, whether expressive performance is included or not (column "Agent Approx Label"). This is also true for the agents' average final tune effective features (column "Approx Tune Label").

4.7.2 Listening Experiment Set-up

The listening experiment involved 10 listeners each listening to the 16 tunes in Table 15. Three of the listeners were musicians or studying music, the others were non-musicians. Listeners were played the tunes and allowed to pause in between them manually. They were told to label each tune with one of four labels: "Happy", "Sad", "Tender" or "Angry". Listeners were also told to mark down their subjective scores for "Interesting" and "Enjoyment" for the tunes, interpreting those words as they saw fit – and scoring between 1 and 10 (10 for most "interesting", or most "enjoyable"). The table was in a form based on that shown in Table 16.

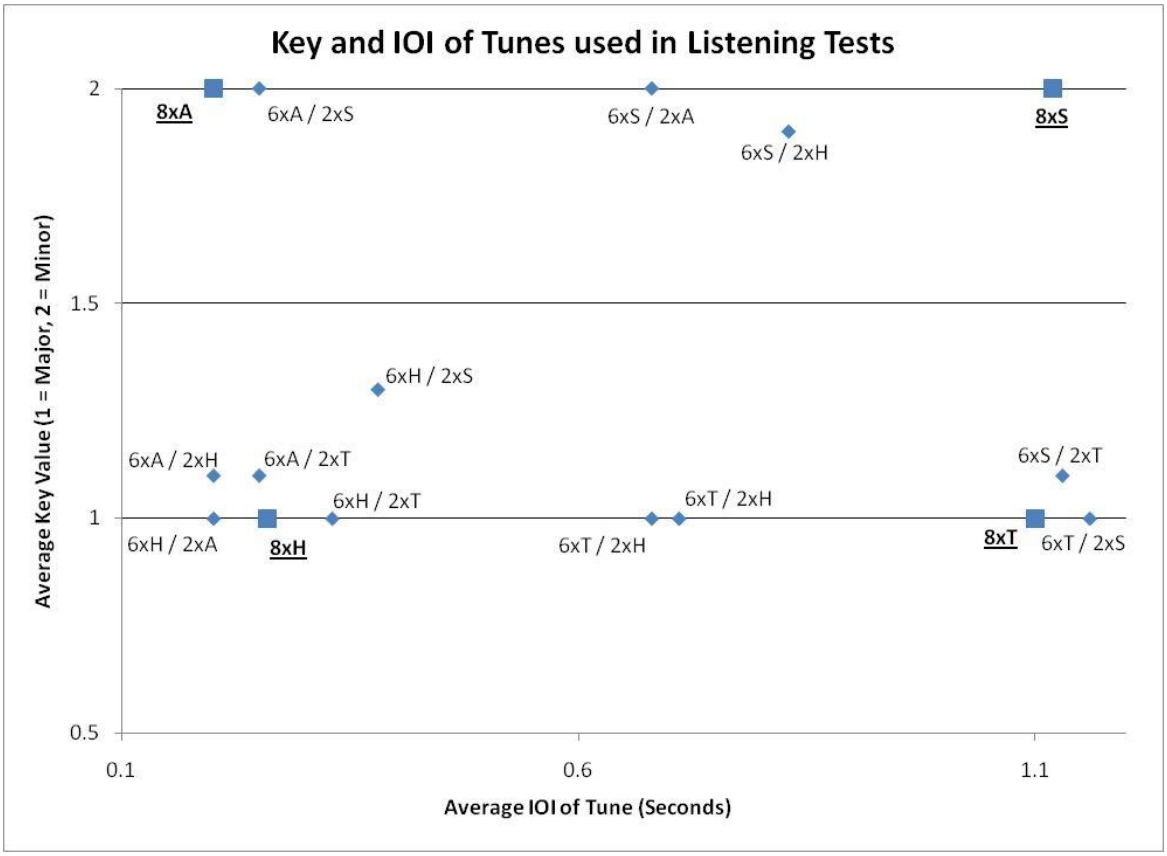

Figure 41: Average Keymode and Inter-onset Interval used in Listening Tests

Listeners completed their tests in one session, and sessions were spread over 2 weeks. Listeners sat at a desk with a laptop which contained 3 WAVs across which the test tunes were distributed in the order described earlier. The subjects would have the experiment explained and then put on headphones. The experimenter would be present for the first two tunes, and ensure the headphone volume was comfortable, before leaving them alone and explaining how they could trigger the two remaining tune files. In between each tune there was a short middle C beep to indicate tune start. There was a gap of approximately 8 seconds between each tune. The listener was given a sheet in which to mark up the results. They were also provided with an instruction sheet which they were given an opportunity to read and ask questions.

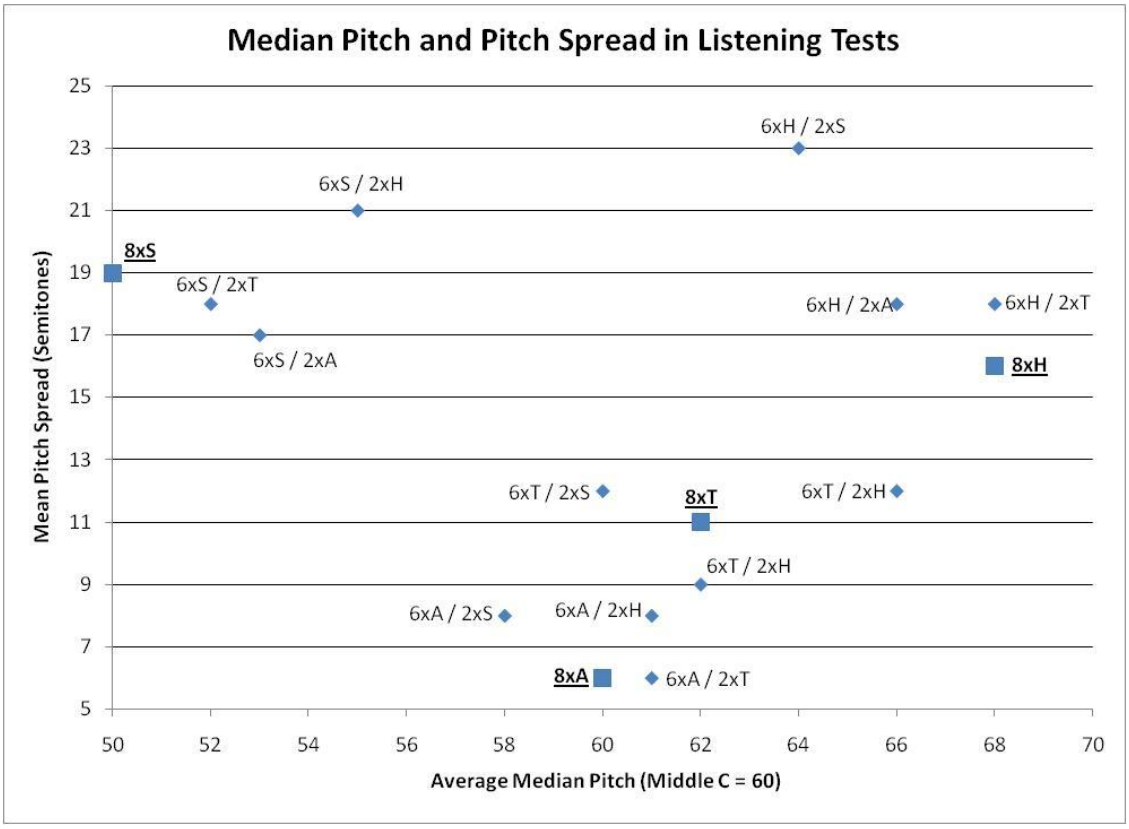

Figure 42: Average Median Pitch and Pitch Spread of Tunes used in Listening Tests

Tune	Happy	Sad	Angry	Tender	Interest? (1-10)	Enjoyment? (1-10)
1						
2 (repeat)						
3						
4						
…						

Table 16: Type of Format used for Listening Experiments

4.7.3 Listening Experiment Results

For calculating the results IPCS was defined as having a "Target" state. Each of the affective labels represents a valence arousal pair, e.g. "Happy" is valence/arousal = 0.5/0.5. The Target state is the initial affective state of the majority of agents in the

system. So for AAAAAAAA it is “Angry” or valence/arousal = -0.5/0.5; for HHHHHHSS it is “Happy” or valence/arousal = 0.5/0.5. The numeric difference in valence and arousal was calculated the valence/arousal implied by the user’s labelling of the tune (as “Happy”, “Sad”, “Tender” or “Angry”) and the Target valence/arousal based on majority initial agent state. So if the initial agent state was HHHHHHSS, the Target is valence/arousal = 0.5/0.5, and if the user selects “Angry” (which is valence/arousal = -0.5/0.5) as their affective estimate of the resultant tune, then the differences are valence error = 0.5-(-0.5) = 1, and arousal = 0.5-0.5 = 0.

Results are given for valence and arousal separately in Figure 43, for reasons that will become apparent. The numbers are shown in Table 41A in Appendix 3 p263. The figure shows how often the mean of how closely a listener’s experience of valence and arousal matched the initialized valence and arousal. The first thing to observe is that the differences are often twice as large for valence as for arousal. This reflects what was found in other research – that valence is much harder to quantify than arousal. The bracketed range given after each mean value is the 95% t-test range. The mean “accuracy” for valence is 71% and for Arousal 82%. The quadrant-based difference was also calculated. This is how often the user’s label matched precisely the majority label for the initial affective state. The quadrant-based matching is significantly lower, as seen in Figure 43. This gives a mean “accuracy” of 59% (which is significantly better than chance – 25%).

The data was also collated for “interest” and “enjoyment” by experimental subject (numbered 1 to 10) in Figures 44 and 45. The linear correlation between interest and enjoyment was 0.77 which is quite high, but shows they are not precisely the same. Another way of viewing the interest/enjoyment results is shown in Table 44A in Appendix 3 p265, showing initial affective state together with average Enjoyment and

Interest across listeners together with some key music features. The table is sorted by decreasing enjoyment.

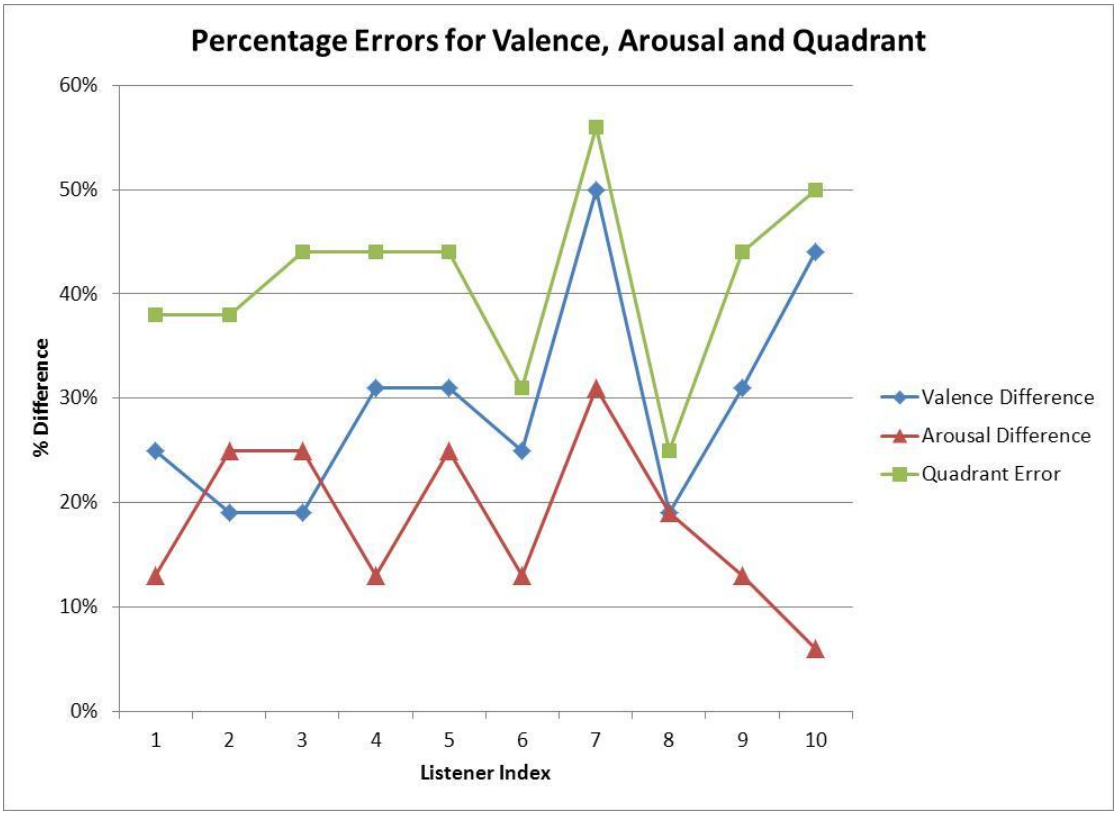

Figure 43: Percentage Errors for Valence only, Arousal only and Quadrant, for Listening Tests

It can be seen that the top 4 most enjoyable tunes have a majority of positive arousal and valence. However the 3 least enjoyable tunes have positive arousal, so assuming some independence of valence and arousal, it may suggest the top 4 tunes are the top 4 because of their positive valence. This is supported by the fact the next 4 most enjoyable tunes all have positive valence, but negative arousal. The order in which tunes were presented is also shown in column 4. There is a linear correlation of 0.44 between the order tunes were presented and the enjoyment of the tune; this may imply that in some sense listeners are learning to enjoy IPCS' music more as they hear more of it, but further experiments would need to be done. It can also be seen that the 3 least enjoyable

pieces had a very low pitch spread. Other features examined in the MAS which it was thought might contribute to enjoyment: mean IOI and mean number of notes, show no obvious correlation to enjoyment.

Table 45A in Appendix 3 p265 shows the same data but sorted by Interest. There are no obvious relationships apart from between presentation order and interest, where the correlation of 0.48 is not significantly higher than 0.44 for enjoyment. The "AAAAAAAA" tune is clearly far more interesting than it is enjoyable; the "TTTTTTAA" is much more enjoyable than it is interesting. Sad tunes are more interesting than tender majority tunes, but tender are more enjoyable than sad. But this is not a large difference, and neither is particularly interesting or enjoyable. Other data not displayed that was examined included tune length in seconds, but it was not significantly correlated with interest or enjoyment.

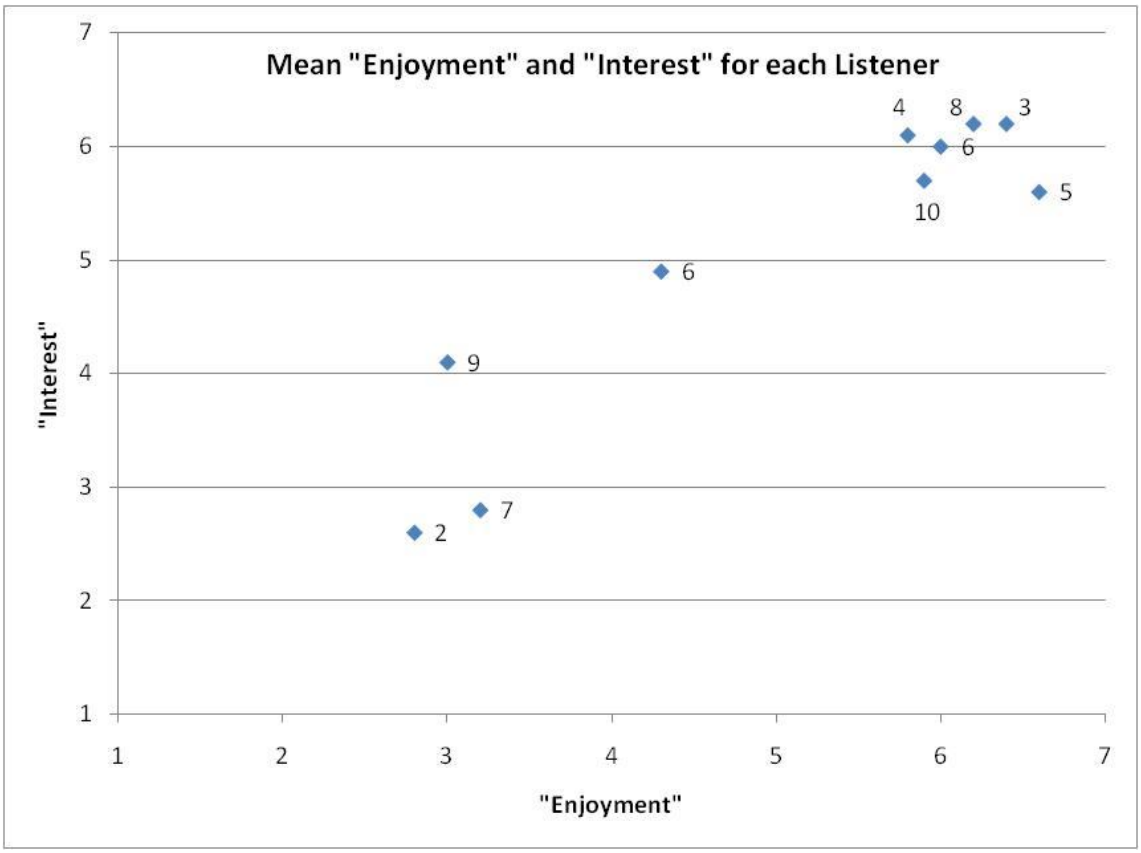

Figure 44: Average Enjoyment and Interest for Each Tune across Listeners

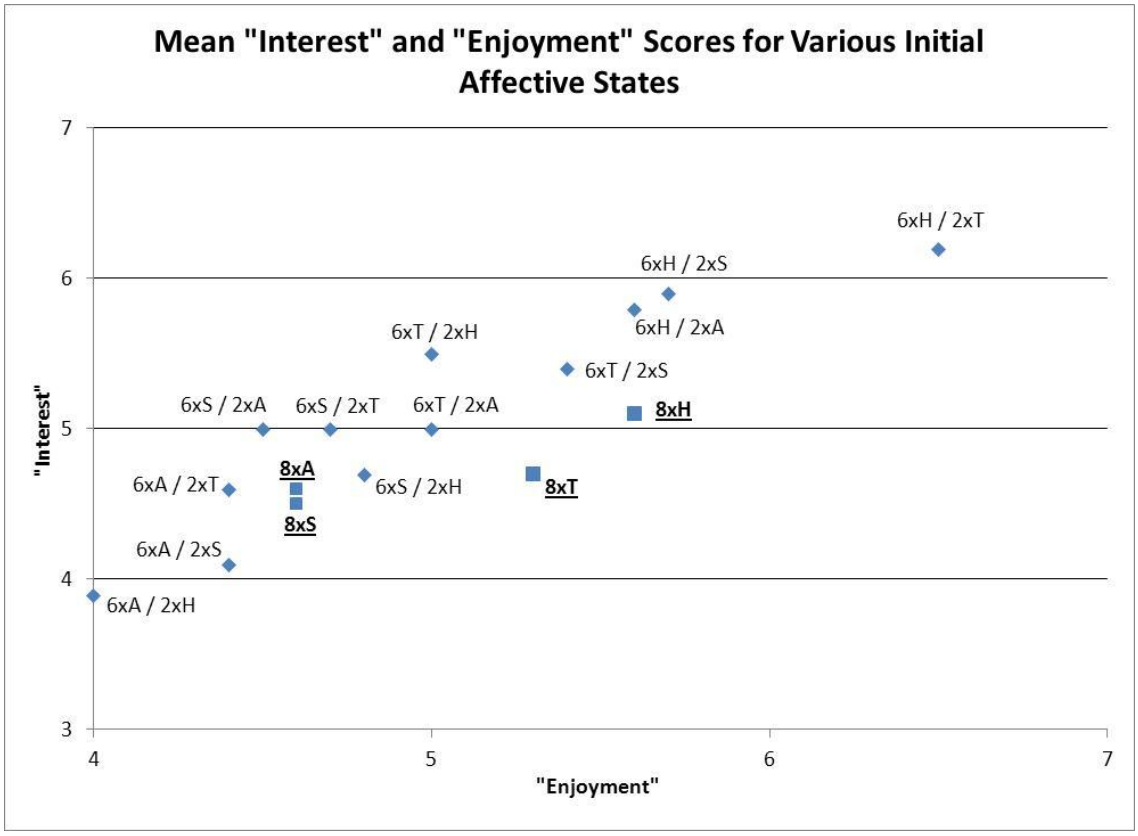

Figure 45: Average Interest and Enjoyment scores across Listeners, for the various Initial Affective States

4.7.3 Exploring IPCS Affective Space

It has already been mentioned that it was interesting to see the variety of melodies produced when preparing pieces for the listening tests. So these pieces will now be displayed to give a sense of the variety available in IPCS in Figures 46 to 53. These figures are displayed in a smaller format as it is mainly the variety of melodic patterns which is being demonstrated, rather than specific detail.

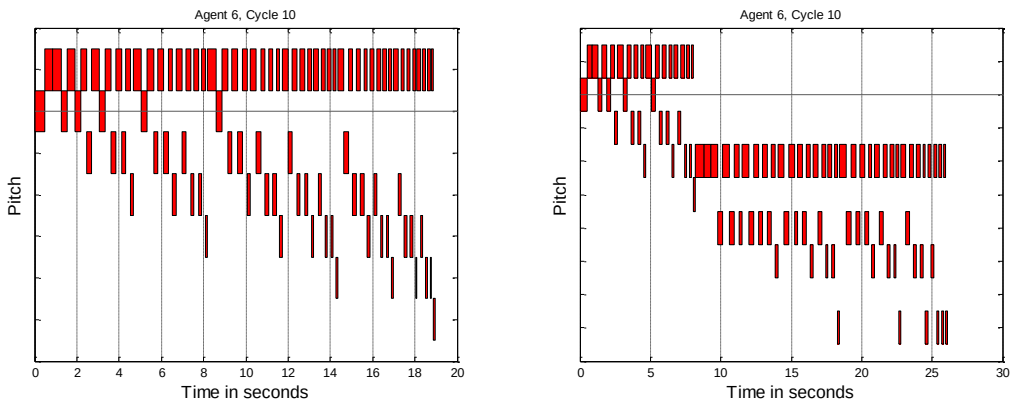

Figure 46: 8 Agents AAAAAAAA, 10 Cycles, and 8 Agents AAAAAASS, 10 Cycles

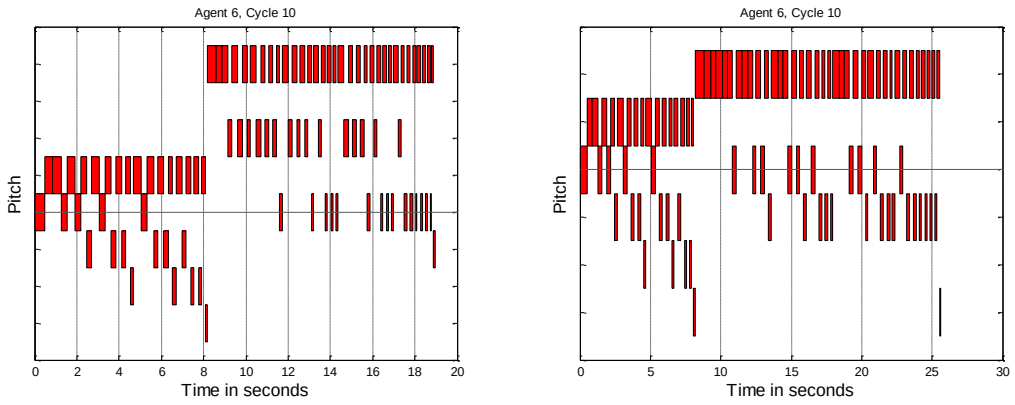

Figure 47: 8 Agents AAAAAAHH, 10 Cycles, and 8 Agents AAAAAATT, 10 Cycles

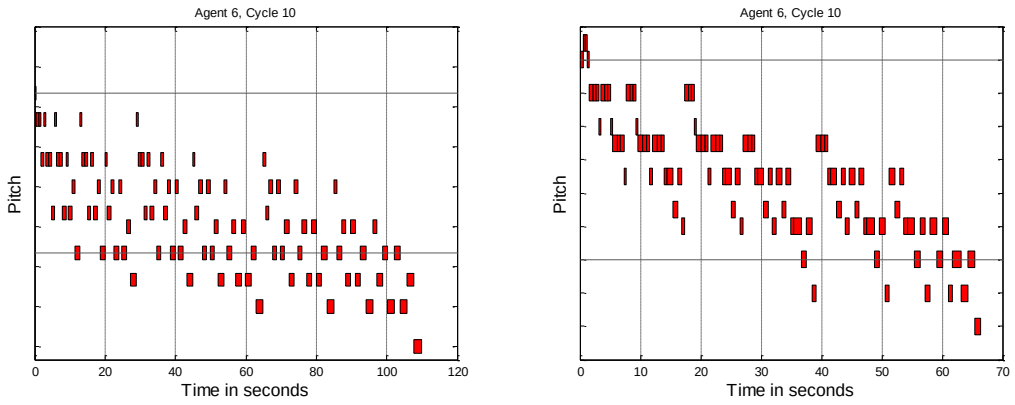

Figure 48: 8 Agents SSSSSSSS, 10 Cycles, and 8 Agents SSSSSSAA, 10 Cycles

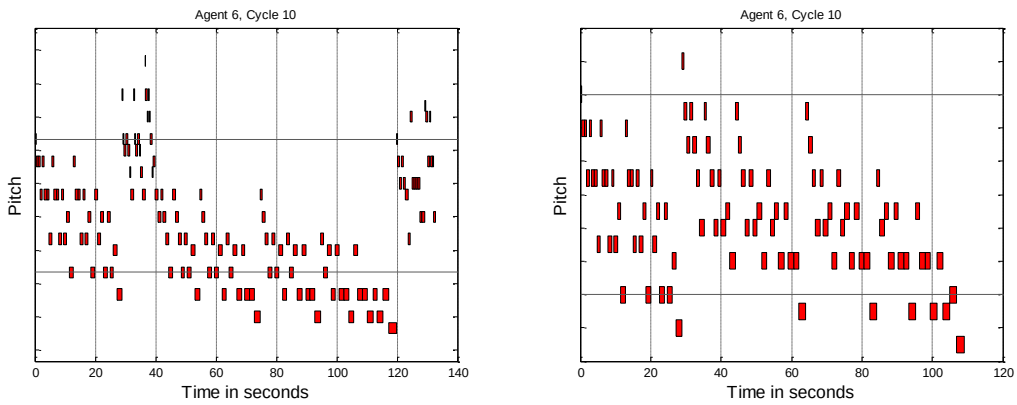

Figure 49: 8 Agents SSSSSSHH, 10 Cycles, and 8 Agents SSSSSSTT, 10 Cycles

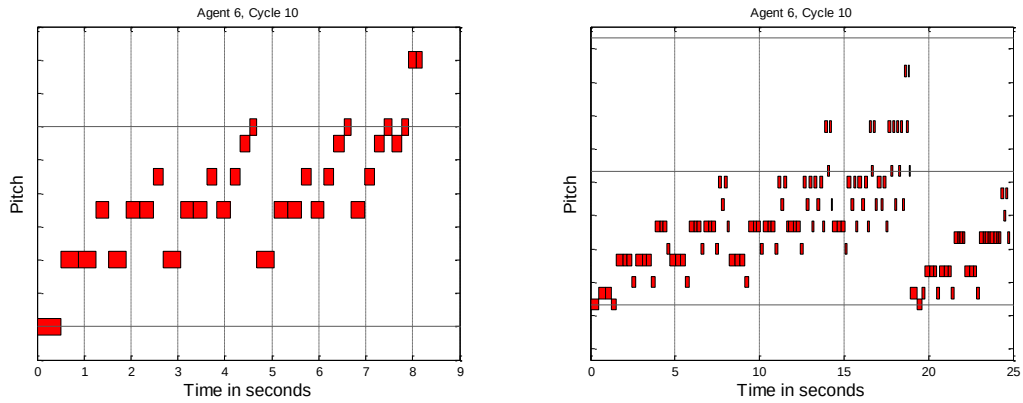

Figure 50: 8 Agents HHHHHHHH, 10 Cycles, and 8 Agents HHHHHHAA, 10 Cycles

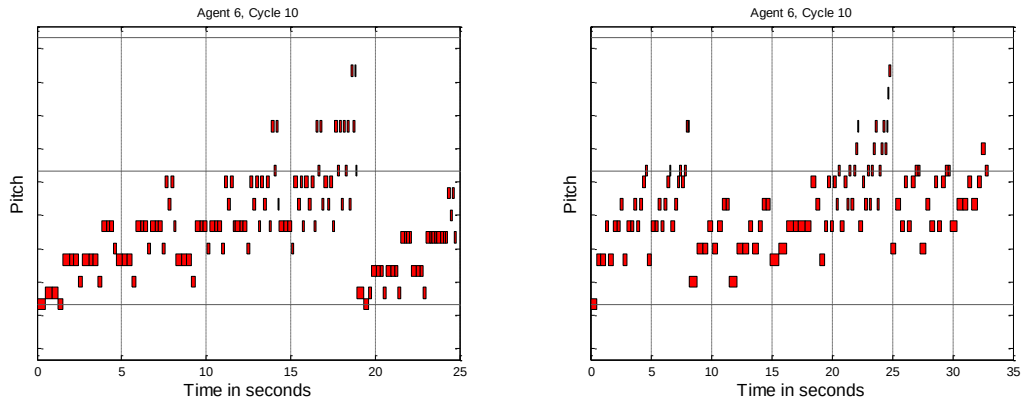

Figure 51: 8 Agents HHHHHHSS, 10 Cycles, and 8 Agents HHHHHHTT, 10 Cycles

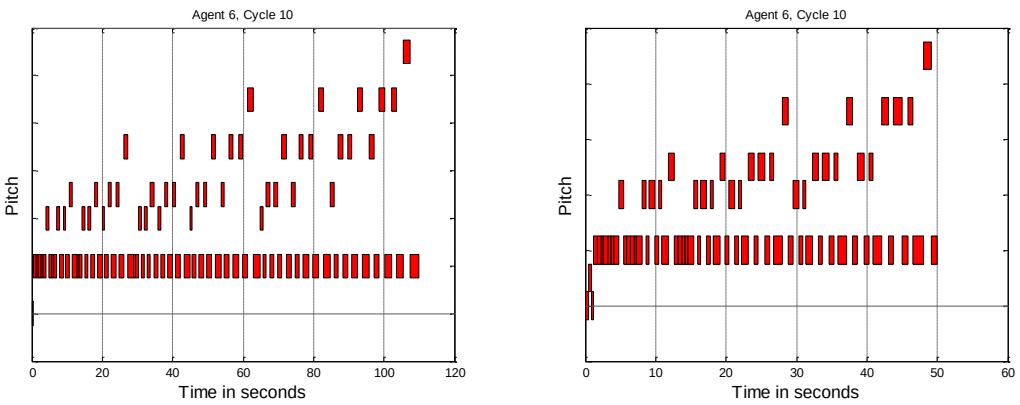

Figure 52: 8 Agents TTTTTTTT, 10 Cycles, and 8 Agents TTTTTTAA, 10 Cycles

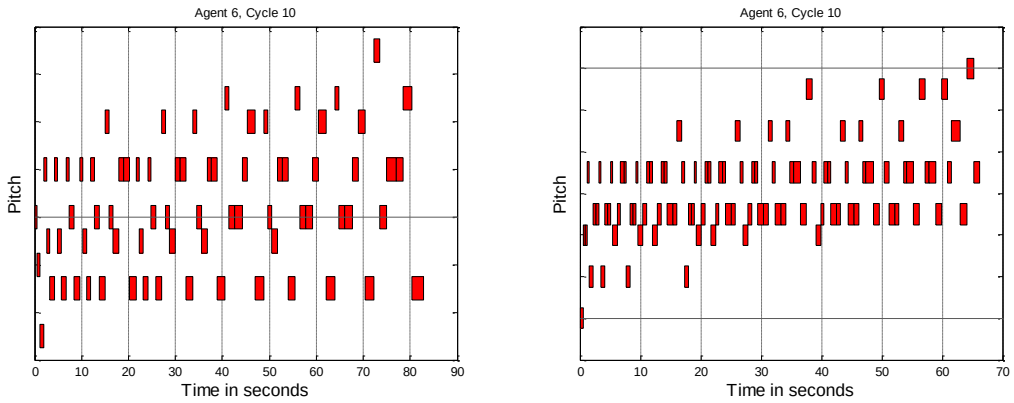

Figure 53: 8 Agents TTTTTTSS, 10 Cycles, and 8 Agents TTTTTTHH, 10 Cycles

4.9 Chapter Summary

In this chapter a number of experiments have been run - a total of 7 experiment sets.

These are all listed below:

1. Experiment Set 1 – Number of Agents, Cycles and Max Tune Length
2. Experiment Set 2 – Initial Affective State
3. Experiment Set 3 – Affective Similarity Threshold
4. Experiment Set 4 – Inter-Agent Affective State Influence
5. Experiment Set 5 – Interaction Coefficient

6. Experiment Set 6 – Emotional Expressive Performance
7. Experiment Set 7 – Listening Experiment

In the next chapter of the thesis, there will be a discussion of the results of these experiments in more depth.

Chapter 5 - Discussions

There will be now be further discussions of the results in the previous chapter. The discussions will be structured as follows:

1. Dynamics of Musical Features
    a. Pitch
    b. Timing
    c. Loudness
2. Software parameters
    a. Number of Cycles
    b. Number of Agents
    c. Affective State Musical Feature Influence
    d. Initial Affective States
    e. Interaction Coefficient
    f. Affective Similarity Threshold
    g. Affective Update Rates
3. Integration of Composition and Expressive Performance
4. Listening Experiment
5. Comparison with Previous Systems
    a. Computer-aided Composition
    b. Expressive Performance

5.1 Dynamics of the Musical Features

The experiments involved investigations of how IPCS responded musically to various parameter changes and various measures of IPCS's behaviour, or providing example outputs. Such experiments provided an insight into the mechanics and dynamics of the system. The behaviour of the 3 key musical features – Pitch, Loudness and Timing – will now be examined.

5.1.1 Pitch

A first observation that should be made is the success of IPCS in generating emergent melodies in an innovative way. Figure after Figure (e.g. 55A p228, 71A p243, 46-53 p178) show the building of non-trivial melodies. Often these melodies are very repetitive over time, due to the recursive and semi-continuous transformations of IPCS. However the fact that such melodies can be generated by local and non-globally controlled interactions based on artificial affective communication is a new approach to emergent melody. This in itself is a far more useful result when a significant variety of non-trivial tunes can be demonstrated from the output. This is the case here: while building the tunes for the listening experiments, a wide variety of different melody types were found (Figures 46 to 53 p178). These tunes are not simply linear runs of increasing pitch, random pitches, or simple repeated patterns, and are therefore significantly non-trivial (Section 5.1.4 below details melodic features regarding as non-trivial). In Computer-aided Composition melody is usually generated (a) randomly, (b) by mapping from some complex pattern generation system, or (c) by a system with trained/rule-designed melodic intelligence. For example the CAMUS (Miranda 1994) mapping from Cellular Automata to music, or the Blackwell and Bentley (2002) mapping from Swarm MAS to music. It could be argued that IPCS is following a similar approach to CAMUS and generating music mapped from patterns of complex multi-agent interaction. However a look at the underlying IPCS information process

shows agents exchanging a growing symbol set with a feedback mechanism based on the agents' artificial affective state. Firstly the music *is* the symbol set, not being mapped from it. Secondly - if one was going to choose a symbol set to use in a system which focuses on communicating emotion, music would be a logical choice. Thus it is argued that the results of IPCS melody generation are a new approach compared to the standard mapping methodology. IPCS does not use randomness to generate complexity, nor pre-defined or trained melodic artificial intelligence. The closest IPCS has to musical intelligence are its mappings from affective state to music features and vice-versa; these are the musical heart of IPCS in terms of functionality.

The question of whether such a melody generation approach has a meaningful tune output is another one. It certainly has *structure* which is one usual requirement for a tune. It also does not simply produce repeated notes, or monotonic runs of notes, or repetition of 2-3 note patterns. IPCS does sometimes produce monotonous repetitiveness (Figure 70A p242, Figure 60A p233). This repetitiveness can also be interesting – for example Figure 71A p243. It is interesting to note that the variety exists despite the fact that in all the experiments the seed pitch was set to 60. (Though it is clear that if the seed pitch is set higher then higher areas of the pitch space can be investigated.) Another important parameter for pitch was the Affective Pitch Influence. In a number of the first experiments it was set to 0.5 and this proved to create large pitch jumps (Figures 54A to 56A, p228). These sort of pitch skips are more appropriate to arpeggios or to occasional elements in melodies.

In terms of absolute pitch in Figure 42 p173 "Sad" gives the lowest average pitches, and "Happy" the highest ones. Median pitch can be adjusted significantly by adjusting in a consistent way initial affective states as shown by the trajectory in Figure 16 (p135): pitch goes down as HH transforms to SS through Tender and Angry, one agent at a

time. Median pitch is not so sensitive to increasing the number of agents (Figure 18, p139). Figure 41 (p173) shows that using initial affective state alone it is possible to get a range of pitch median pitches from MIDI pitch 50 to 70, with average pitch spreads from 5 to 25 semitones around this.

For pitch spread (i.e. the maximum minus the minimum pitch in a tune) - apart from AS in Figure 16 (p135)- the lowest pitch spreads are achieved by combining agents with different initial affective states. Pitch spreads are more sensitive than median pitch to increasing agent numbers (Figure 18, p139), and to similarity threshold changes (Figure 20 p141). Similarity threshold increases pitch spread as it is increased. In a number of cases, the pitch spread tends to change in relation to the length of the tune when changing a single parameter. For example Similarity Threshold in Figure 20 (p141) and k-Value in Figure 39 (p165). However this is not always the case – the Interaction Coefficient leads to significant changes in tune length while keeping the pitch spread approximately the same (Table 36A, p261). This suggests that a combination of Similarity Threshold and Interaction Coefficient could be used to control pitch spread if desired, though of course this would not be independent of other factors. The agent Initial Affective State in Table 40A (p263) leads to a more interesting relationship between pitch spread and tune length.

Pitch spread can also be adjusted using affective update rate (Figure 26, p148) though it can cause some runaway feedback for 2 agent systems involving “Tender”. For larger agent groups (8 agents) there is also a median pitch/pitch spread cycle spanning the space (Figure 18, p139) going from H->S->A->T and back to H. This cycle is bounded for pitch by 30 and 105 top and bottom, and for pitch spread at 20 and 125 at left and right, though this is for mean values.

Another element in the pitch structure is the key mode. Average key is mostly influenced by the majority initial affective state. It can be adjusted but tends to end up with mean agent keys close to being major or minor (Figures 17 and 41, p138 and 172). For the two agent case the exception is SH and for the 8 agent HHHHHHS. However 6S/2H, 6S/2T, 6A/2H, 6A/2T also show a little more variety in key behaviour. This may be because more agents allow more variety in key across the population in final tunes. Figure 41 p172 shows that agent groups with the majority of "Happy" and "Tender" in their initial state (e.g. HHHHHHAA) are less influenced by agents with different affective states. Those agent populations which start with an "angry" or "sad" majority are more influenced by agents with other affective states. For example an angry population with two tender (AAAAAATT) and an angry population with two happy (AAAAAAHH) give a final average key mode very close to major (1.1). Whereas a "Happy" majority is only influenced in average key mode when two "Sad" agents are added (HHHHHHSS); but still stays closer to a major than minor key mode. Of course this analysis is dependent on the key finding algorithm, which is in itself imperfect; but should be indicative. Interaction threshold impacts key – in particular between 0.3 and 0.9 (for SSAAHHTT in Figure 32 p154) the tunes are made more minor, and are major otherwise. This is perhaps because the lower agents are still tune providers in this range. At 0.9 the Angry and Happy agents are equally tune providers, but first two agents (Sad) become tune receivers and the last two (Tender) agents become tune providers. Similarity threshold moves key from major to minor as it increases (Figure 21, p142). This is partly because the key finding algorithm is biased to minor keys for shorter tunes. On average key will be more minor when using non-zero k-values (Figure 40, p165).

5.1.2 Timing

The type of timing being discussed here is the compositional, not expressive performance element. Looking at Figure 41 (p172) there is a wide variation of average IOI from 0.2 seconds (AAAAAAAA) up to 1.16 (TTTTTTSS). All “Happy” leads to a slightly lower tempo than all “Angry”, all “Tender” leads to a much lower tempo than all “Happy”, and all “Sad” is slightly lower tempo than all “Tender”. It is interesting to note that the fastest IOI is achieved using a single initial affective state type – “Angry” - whereas the slowest is achieved by mixing two types of initial affective state – “Tender” and “Sad”. There is no clear pattern in how adding minority affective states to a majority initial affective state changes timing, apart from the obvious one of influencing towards any added minority state.

It has been mentioned that median pitch showed a clear trajectory of reduction through H->T->A->S; however tempo reduction (IOI increase) moves from H&A->S or H&A->T (Figure 17, p138). All-H and All-A have similar tempos, as does all-T and all-S. Increasing similarity threshold reduces tempo in the SSAAHHTT case (Figure 21, p142), presumably because it increases interactions and thus the influence of the first agents (S and A). Affective update rate has significantly more effect on mean tempo than it does on median pitch (Figure 25/26, p147/148) – in particular for low arousal combinations.

Timing has proved to be a more problematic part of IPCS than pitch. Across a number of different parameter settings IPCS tunes often exhibit a global acceleration or deceleration of tempo (called by musicians an *accelerando* and *decellerando* respectively), with or without expressive performance included. For example Figures 62A (p235) and 61A (p234) show an accelerando, Figures 54A (p228) and 60A (p233) shows a decellerando. This is due to the following. Suppose there is a 2 Agent system, and Agent A is “Sad” and performs to Agent B who is “Neutral”. Agent A’s “Sad” state

will cause its performance to lower the tempo of the tune it performs. Suppose Agent B adds that tune. It will also lower its own affective state towards “Sad”. So when Agent B performs the tune back to Agent A, Agent B’s “sadder” state will cause it to decelerate the tune even more which in turn will make Agent A’s affective state more “Sad”. Agent A will now contain a version of its tune, followed by a transformation (with possible additions) of a slowed down version of this tune. As this process continues the agents end up with a decelerating tune being built. The same can happen for, say, happy agents with accelerando. It was hoped that during experiments parameter settings would become clear which could lead to a complex “balance” in this process of affective influence. Figure 66A (p238) shows the global deceleration becoming local. But outside of specific cases, no general balance has yet been found – though this does not mean it does not exist. Thus some using IPCS may choose to minimise the ability in IPCS of agents to affectively influence each other, as was done in the listening test.

On a note-to-note level, timing has the opposite issue – too little variance (whereas pitch variance is more interesting). One reason for this is that timing changes are perceptually fairly continuous, whereas pitch frequency is quantized into jumps and key modes. It may be that that a timing quantization method should utilized to create more interesting timing. This would create more of a timing “palette”, instead of a semi-continuous time transformation. Such an approach would actually be closer to the way that composers are inspired to work by common music notation – i.e. changes in tempo are marked by themes or sections, and are only occasionally continuous (e.g. with accelerando markings). Composers generally rely on the performer to insert continuous timing changes. So it may have made more sense to have compositional timing changes being quantized and allow the expressive performance algorithms to insert some continuous timing. This is something that will be explored in future work.

Although duration composition is influenced by affective state, it is only in such a way as to reduce or increase duration in proportion to the IOI.

5.1.3 Loudness

The type of loudness being discussed here is compositional, not expressive performance. It could perhaps be seen as unusual to include loudness as a compositional element. But one only needs to take a look at a number of scores to see a plethora of loudness indicators (*f*, *p*, crescendi, etc) put in as part of the composition process; these refer to instructions to play loudly (*forte)*, quietly (*pianissimo*), louder over time (*crescendo*), etc. However the loudness generated during IPCS composition has been shown to be different to this. For example Figure 57A (p231) shows significant changes in the loudness between individual notes, exceeding 200% change between notes. There are slower patterns – e.g. a gradual global decrescendo in the figure. But in some cases a listener will hear the note by note changes as well. Figure 59A (p232) – where 2 agents have opposite initial affective states – shows significant changes between almost every note. Larger numbers of agents with a wider affective initialization seem to mitigate these rapid changes: for example Figure 65A (p237) with 8 agents with equal affective spread shows more stability and smaller changes in the loudness on a note by note level. Similarly Figure 72A (p244) shows a more stable and much *smaller* change in loudness (down to under 5%). This illustrates an issue with utilizing an affective transformation system for compositional purposes. Experiments which have developed rules for affective transformations – e.g. CMERS and Director Musices - have focused on affective communication, *not on compositional interest and enjoyment*. This could be seen in IPCS as a weakness, because rules are being utilized which were not originally designed for use as compositional tools. It could also be seen as a strength – as non-standard approaches to composition may enable to the user to find tunes of greater interest. Whichever of these possibilities is dominant, the loudness composition also

raises the issue of notation – such frequent and non-continuous changes in loudness would be hard to notate for human performance.

5.1.4 Summary of Non-trivial Musical Features

The following are the key features of the experiments which support the non-trivial musical features of IPCS melodies (in spite of the lack of explicit melodic intelligence in the system):

1. **Tune length:** music has been generated up to around 50 seconds long. If the tunes were only 3 to 6 notes or a few seconds long, these would be trivial.
2. **Pitch and timing change direction:** the melodies are not just simple directional patterns, like single repeated notes, uniformly rising or falling patterns, or repeated "zig-zags".
3. **Pitch repetition not too high:** The melodies are not just groups of repeated notes, e.g. 5 notes at one pitch, then 4 notes at another, etc. The pitches vary much more than that.
4. **Pitch repetition not to low:** However they do not vary all the time – there are times when notes are repeated 2 or 3 times, as one would expect in melodies.
5. **Timing repetition not too high or too low:** a similar observation as with pitch in 3 or 4 above. We have generated melodies where the tempo does not stay constant, but also the tempo does not simply seem to vary randomly.
6. **Pitch jumps not too small or large:** it would seem odd if note only raised or lowered by one pitch in all tunes. In music there are sometimes larger jumps. However it would also seem odd if all the pitch changes were large. We have shown that tunes can be generated which avoid these two extremes.

7. **Motifs structure:** the melodies contain recognizable note groupings which are repeated and transformed to different pitches and timings. This is expected by western listeners who usually look to identify a motif structure.

In terms of flexibility, Figures 41 and 42 (p172/173) show some gaps in the feature space, but generally span much of the feature space. The major gaps are because the system seems to cluster towards on average having tunes in major or minor keys. Note that all these observations are made in the context of computer-aided composition. IPCS does not generate entire pieces of music. In our investigations we have only seen it generate tunes of less than 60 seconds.

5.2 Software Parameters

A key element with IPCS is finding the correct parameter settings to create non-trivial (or "interesting") tunes. This is easier to do for shorter tunes than longer ones. However – it is possible to write longer tunes for example - Figure 49 (p179) with 8 agents (SSSSSSHH). It could be argued that even with parameters which lead to repetitive larger scale structure, IPCS could be used for phrase generation. This is sometimes done with other A-life/Cellular Automata approaches (Miranda 2001b).

A few of the other key parameters will now be examined in IPCS.

5.2.1 Number of Cycles

A larger number of cycles increases the number of interactions agents can have. In each cycle a selected agent has the opportunity to perform to all other agents. Another option for defining a cycle would have been that all agents interact with all other agents. This approach was used at first but was found to lack control – because the effect of a single

cycle would grow exponentially with the number of agents. A potential effect of number of cycles is saturation of tune features – for example pitch and loudness – due to recursive interactions. It could also cause a runaway tempo growth – since tempo is not capped in MIDI like loudness and pitch. The effect of cycles is particularly important to take into account, since a 1 cycle increase (say from 12 to 13 cycles in Figure 13 p131) can lead almost doubling size, in terms of number of notes.

5.2.2 Number of Agents

One effect of number of agents is closely related to number of cycles. A larger number of agents will cause potentially more interactions for a single cycle. So increasing the number of agents increases the potential tune length – Figure 13 p131 shows this. A similar effect is found looking at Figure 14 p132.

Another effect of number of agents is it allows the mixing together of more groups of initial affective states. For example a two agent system does not allow combinations of more than 2 initial affective states, whereas eight agents can combine 4 different initial affective states easily. Figures 17 and 18 (p138/139) show that for most cases, increasing the number of agents does not create as large a change in features as changing the combinations of initial affective states. This is a useful property as it means that one's experience of composing with smaller numbers of agents gives some intuitive sense of a starting point for composing with larger numbers. The exception for this is Pitch spread which has some quite large changes when TT->TTTTTTTT, AT->AAAATTTT, and SH->SSSSHHHH. For AAAATTTT and SSSSHHHH the pitch spread for 2 agents is dramatically lower than for 8 agents – and the average keymode changes significantly as well. For SSSSSSSS, AAAASSSS, and AAAAAAAA the only real differences in results (apart from tune lengths) are shown in Figure 19 p140 in the final tune affective estimates.

The larger agent populations seem to reduce the average magnitude of the affective variables in Figure 19, while usually retaining the same quadrant. The Figures dotted boundary lines show that the valence/arousal of 8-agent systems is roughly bounded within the 2 agent systems' valence/arousal behaviour. So increasing the number of agents has the effect of reducing the changes in valence and arousal of agents. Thus it regulates the runaway tendencies of IPCS for positive feedback.

5.2.3 Affective State Influences

The effect of an agent's affective state on the tune features pitch, loudness and timing is key. The initial experiments used values of 0.5 for timing, loudness and pitch respectively. It was immediately seen (Figures 54A p228 and 55A p229) that this led to large pitch changes. Hence the pitch influence was reduced to 0.1 in later experiments. What was interesting, but not investigated further, was that there was no need to reduce the loudness influence explicitly, but that the addition of other IPCS features such as similarity threshold seemed to mitigate the effects. The timing influence value is problematic for two reasons: firstly it does not separate out onset and duration; and secondly because increasing it as a means of increasing timing diversity in a tune also leads to a potential increase in the self-perpetuating runaway of tempo in recursive reactions.

5.2.4 Initial Affective States

Initial affective state settings were found to be some of the most effective tools in IPCS for manipulating tune features (Figures 17/18 p138/139). It also allows for some balancing to lower undesirable recursive effects – e.g. in Figure 19 p140 there is a clear trajectory going from 8H to 8S and from 8A to 8S. It is interesting to note that the equal weighted group (SSAAHHTT) is very roughly in the middle of the bound area by the 8H, 8S, 8T, 8A. As in fact are all of the 8 agent combinations tested. Setting initial affective states was shown by the linear affective estimator in Figure 19 p140 and by the

listening experiment (Figure 43 p175) to have a significant influence on the communicated affectivity after 10 cycles in an 8 agent system. So to a degree the user can utilize the initial affective states to control the affective communication of the composition. However it is not clear whether after more cycles than 10 this effect will continue. Also the ILE predicts the 8S system ends up in the Tender rather than the Sad quadrant in Figure 19 p140.

Figure 19 p140 shows that AA, HH and TT all end up in the same quadrant as they started, according the ILE. But SS ends up in the Tender quadrant. TT is the most extreme in its resulting valence/arousal, followed by SS, then HH, and then AA. Only AS populates the Sad quadrant. So the Sad quadrant generally seems to be a repellor, whereas the Tender quadrant seems to be an attractive zone. “Tender” dominates 5 out of 8 rows in Table 26A p255 in terms of tune features. This happens even if the initial affective states are not in the “Tender” quadrant. A similar tendency of “Tender” to be an attractor is shown in Tables 36A and 37A (p261). Why this should be is not clear from the current experiments. Tender is high valence and low arousal. It suggests that the compositional transformation rules (Equations (1) to (4)), when all else is equal, may tend to be attracted towards this region in recursive application.

It is useful to note that because the listening experiments showed a communication accuracy of 71% for valence and 82% for arousal, and the IPCS Linear Affective Estimator (in Column 11 Table 40A p263) showed an accuracy of 100% for estimating the communicated valence and arousal quadrant, then the initial listening experiment provide initial further support for the validity of the IPCS Linear Affective Estimator as a contribution.

5.2.5 Interaction Coefficient

The interaction coefficient was shown by – for example – Figure 73A (p245) that it could be used to cause an emergent division of the population into motif/phrase

providers and motif/phrase receivers. This behaviour is a fairly complex feature of IPCS and there was no investigation into a variety of compositions utilising it. To fully understand its application would require a rather higher dimensional investigation possibly of the order of the length of the whole parametric results section as it already is. However it is considered that the initial claim has been sufficiently supported – that there can be a link between the multi-agent social network and the musical form. Though, as was stated for expressive performance, this musical form is the one defined by the IPCS generative process and could be at odds with a human's perception of the structure in some cases.

5.2.6 Affective Similarity Threshold

The functionality of the Affective Similarity Threshold was to control tune growth and to examine the possibility of it imposing an emergent affective tune structure. The first of these was found to be true in Figure 20 (p141), where lowering the threshold led to smaller and smaller tune lengths. It also shows a significant effect on pitch spread, which is to be expected as agents with different median pitches will often have different affective states, making them less like to interact. In terms of its effect on tune structure, it is not clear if it imposes an affective structure. Comparing, for example, Figure 66A (p238) which has no affective threshold to Figure 62A (p234) which has an affective threshold of 1, the main element apparent is that the motifs or phrases that the tune is broken down into are shorter. This is useful to note as it means that the affective threshold is not here reducing tune length by simply stopping interactions at some point where tunes diverge too far from each other, but that it reduces the tune length by being selective of material from early in the generative process. The final structure of the 2 tunes are quite similar by eye (as are their statistics in Row 8 Table 26A p255 and Table 27A p256), with the global upwards pitch movement, except for a lower pitch version of the theme around the middle of the tune. So the similarity threshold here is

maintaining the structure, but reducing the size of the elements (and slightly reducing the pitch spread).

Figure 20 p141 also shows that for thresholds below 0.4 there are no significant interactions between any agents. Even for agents with the same initial affective state, once tunes have been changed sufficiently, they diverge too far from each other and interactions cease. With affective threshold 1.2 and above, agents interact as if the threshold was infinite. Figure 22 p143 shows that between 0.4 and 1.2 increasing the threshold causes the resulting mean tune valence to reduce very slightly, and the result estimated arousal to change in a small n-shape. This n-shape may be related to the peak in median pitch and loudness which occurs at similarity threshold of 0.9. The reason for this peak is not clear, but is possibly something to do with higher valence or arousal agents being able to have more influence for this setting. Once the threshold goes above the value some lower valence or arousal agents may be allowed back into the tune interaction process, and below this value the higher valence and arousal agents may be blocked from taking part in the interaction.

5.2.7 Affective Update

The affective state influence between agents was aimed at adding extra affective dynamics to the interaction process which is was hoped would translate into musical dynamics over a longer time-scale. Looking at Figures 25 to 27 (p147-149) (and Table 34A (p260) with affective update active, and Table 25A (p255) without affective update active) there are a number of differences. A key thing to observe in Figure 27 p149 is that all but one approximate affective labels (quadrants) of the average final tune affective states are the same, if both agents have the same initial affective state. The exception is Sad / Happy which goes from Sad to Tender. The update rate has a very large effect on TT and SS extending them outward on both valence and arousal. Interestingly the same figure shows the update rate limits changes with negative

valence, and increases those with positive valence (except HH). Figures 25 and 26 show that IOI and median pitch are changed by a large amount for TT, and IOI and key are changed by a large amount for SS as well.

It has already been observed that IPCS dynamics do have tendencies to move in certain directions and not others – so this possibly explains the non-symmetric effects of the affective update rules.

5.3 Integration of Composition and Expressive Performance

The effects of integrating expressive performance and computer-aided composition in IPCS are now discussed. Figures 79A to 82A (p248) support that IPCS does utilize the properties of the integration. In IPCS the structural analysis is automated, and "perfect" in one sense. When a human composes a piece of western music they naturally think in terms of motifs, phrases and sections (if not consciously). The composer will perceive these as being units and build using them. A fellow human listener will *tend* to perceive the same elements as being units. However it is not clear how IPCS's building units are related to human perception; or if there is a way of encouraging this using IPCS's parameters.

There will be a tendency, if somebody hears a motif or theme repeated, to hear it as a motif or theme. Examples of IPCS tune with this property include Figure 60A (p233), Figure 66A (p238) and Figures 46 to 53 (p178). However it may be that the natural human perception may be to hear theme as starting in the middle of one of IPCS's themes, and then ending in the middle of the next IPCS theme. So if IPCS uses expressive performance on these two themes, the performance may actually *confuse* a listener's perception of the tune. This can be looked at in a more creative way. In many cases, non-trained listeners can find it hard to make out the structure of contemporary

classical music. Thus it is not a requirement for a piece of music's structure to be perceivable. From a creative point of view, and because IPCS is not a composition simulation system, it could be argued that this non-perceptual constraints in IPCS could lead to more interesting music, though this remains unexamined.

Another element of the combined system is that the music is being shaped by the expressive performance as well as the compositional elements. Looking at Figures 39 and 40 p165 (Table 39A p262) the effects of expressive performance can be seen on compositional elements. In the table and figure the most obvious element is that expressive performance causes greater changes in tunes, which leads to lower tune lengths – since these changes will take performances above the affective similarity threshold more often. Loudness trends upwards the increase in expressive performance, possibly linked to the increase in agent valence as k-value increases. Though it is interesting to note that when k reaches a certain critical value, the average agent affective state is suddenly dominated by the "Angry" quadrant. The average key mode starts quite major, then once k goes above 1 becomes quite minor, and for the highest three values of k sits somewhere between major and minor.

5.4 Listening Experiment

In a system like Livingstone's CMERS there was a primary purpose of communicating emotion – it was the system's contribution. For this reason CMERS evaluation involves twice as many listeners and extensive test design and statistical analysis of the tests as IPCS. However the primary contribution of IPCS is the investigation of the utilization of intermediate multi-agent systems to combined computer-aided composition and expressive performance. The use of affective states in the agents was primarily a method to develop a musically-relevant interaction of sufficient complexity to generate

non-random but non-trivial material. IPCS is more closely comparable in *aim* to Ossia (Dahlstadt 2007) or the Cellular Automata work – CAMUS - of Miranda (1994) than to CMERS, however IPCS's underlying methods bear comparison with CMERS.

Because of this change in focus, IPCS uses a similar approach to that found in such systems as Ossia and CAMUS – the investigation by parametric examples. Similarly the system is run in a number of configurations to investigate its ability to produce non-trivial musical material in a structured way. However, the decision to include a listening experiment was partially motivated by Papadopoulos and Wiggins (1999) comments on algorithmic composition not being viewed as a "creative and meaningful process". So one way of doing this was to run an initial experiment to examine the affective interpretation of the music in relation to initial parameter settings, as well as some "creative" evaluation using subjective labels of "interesting" and "enjoyable". However because parametric investigations are the standard methodology in algorithmic composition, only an initial listening experiment was done (see also Section 5.5.2).

It should also be noted that the listening experiment essentially looked at the results of recursive application of an initial affective bias to the final perception of the music. Basing the results on "accuracy" is not meant to imply that IPCS is designed to produce "Happy" music after eight happy agents have interacted ten times. It is meant to imply that where IPCS to show no relationship between the initial affective states, and the affective perception after a small number of cycles, this would be sufficiently counterintuitive to a composer so as to reduce IPCS utility. So a higher "accuracy" implies a more understandable relationship between IPCS' initial affective states, and the affective perception of a final tune.

Looking now at the results, it should be observed that a pigeonholing for test-subject affective response strategy was deliberately used because of the highly subjective nature

of defining emotions. Two listeners did comment on this. But if the listeners were allowed to write down their own emotions then it would be difficult to collate data. However this may have had the effect of increasing the apparent accuracy. The final accuracy for selected affective label (“Happy”, “Sad”, “Tender”, “Angry”) – Figure 43 p175 (Table 41A p263) - was 59% which was more than double the chance value of 25%. Because the aim of IPCS is not primarily emotional communication, but composition, this level is not unreasonable. It is significantly less than CMERS (Livingstone 2009), which uses the same emotion labels, and has average accuracy of 78%. It should be noted that unlike IPCS, CMERS uses well-known pre-composed tunes by Mendelssohn, Mozart and Beethoven, utilizing both compositional and expressive performance rules. The results in IPCS are based on compositional rules only, and are newly composed tunes. Furthermore, as can be seen from the results, tunes in IPCS with different initial agent affective states will be different tunes, whereas the same base tunes are presented to CMERS listeners but with different emotional elements changed. However it is interesting to note that the original version of CMERS, which only used compositional rules, achieved an accuracy of 63%, very similar to IPCS without expressive performance.

Although Director Musices-type rules were not tested in IPCS for their emotional effect, it is worth noting that Director Musices was tested by Bresin and Friberg (2000) to give an accuracy of 63% across 7 emotion labels (including the 4 used in IPCS testing). Livingstone (2010) retested and came up with an accuracy result for Director Musices of 49%. Once again, these compare with IPCS’s results.

For IPCS, when the results are separated into valence and arousal, the accuracy increases to 71% for Valence and 82% for Arousal. This supports the idea that IPCS is more successful at controlling arousal than valence, and also is compatible with the

previous research suggesting valence is harder to control than arousal. It should also be noted that Figures 46 to 53 (p178) show that changing two of the agent's initial affective states in a system of 8 agents can have a significant effect on the final tune. The results for IPCS are also based on the idea that a majority of agents being a certain affective state should, after 10 cycles, lead to the user perceiving that state. If results are limited to only those experiments where all 8 agents had the same affective state, the accuracy results for valence and arousal become 70% and 90%. Figure 43 (p175) shows clearly that valence errors dominate the quadrant errors and that the system communicated far better to some listeners than others.

It is worth noting that the division of affective state into the two dimensions of valence and arousal, although common in much emotional research, is not a perfectly accurate model. So the results of these tests are dependent on the accuracy of this dimensional description. It is used confidently here because of its standard utilization across a number of fields.

Another element of the listening experiments was the responses for "enjoyment" and "interest". It is not clear how the users utilized these scores – whether they were used in an absolute sense or a relative sense. If they were used in an absolute sense then the average score of 5 for enjoyment and interest is fairly encouraging for monophonic tunes which were automatically composed using a process with no explicit compositional intelligence. On the other hand – assuming absolute scoring - a highest average score of 6.6 for enjoyment and 6.1 for interest shows there is significant work to be done with IPCS, either improving the system, or only utilizing it for shorter phrases. If the scoring was relative then it should be noted that the mean scores spread from 6.6 down to 2.8 for enjoyment, and from 6.2 down to 2.6 for interest. If a rough

linear scale is assumed, then listeners like some tunes almost 3 times as much as others, or found some tunes almost 3 times as interesting as others.

Figure 44 (p176) supports the idea of a spread of listener experience, and they appear to group into cluster of "likers" around 6/6, and the cluster of dislikers around 3/3, plus one listener in the middle around 5/5. The correlation between interest and enjoyment is clear in this figure. Figure 45 (p177) shows that angry seems to be the least enjoyable and boring as an initial affective state. Happy was rated the most interesting and enjoyable on average. It is not clear that mixing up initial affective states made things more interesting or enjoyable. Happy is rated a little more enjoyable and interesting than Tender on average.

5.5 Comparison with Previous Systems

The issue of how these results highlight the relationship between IPCS and previous work will now be discussed.

5.5.1 Computer-Aided Composition

IPCS was not designed as a system for emulating composition traditions in detail – it differs radically in usage from such systems such as David Cope's EMI (Cope 2005) used in the emotional DM tests, which can simulate composers such as Mozart; or systems such as William Shottstaedt's (Shottstaedt 1989) program simulating Palestrina-type counterpoint rules.

IPCS applies intermediate multi-agent systems in an attempt to enable to a composer to generate emergent material which they may not have come upon themselves. This could be compared to some of the work of Iannis Xenakis (1963) who utilized computers to generate material which he then used as the basis for some compositions. The first algorithmic composition by Hiller and Isaacson (1959) attempted to produce the whole

piece of music using a computer. Based on the results so far IPCS may not be suitable for whole pieces in its current form due to such elements as the repetitiveness over longer time scales and the potential for global accelerandos and decellerandos.

It is worth dwelling on the comparison between IPCS and Miranda's CA approach in CAMUS. There were two main significant insights in Miranda's approach here: (1) that CA provided complexity without random non-structure; and (2) the discovery of a mapping of visual patterns to musical features which created compositionally interesting phrases and developments that could be moulded by a composer manually into a full piece. In IPCS the complexity without randomness comes from multiple agents with differing affective states, and the performances affecting the musical content of the tunes, and the multiple interactions. Whereas the CA method utilized by Miranda (Game of Life) was well established before he did his work, the multi-agent interaction structure and properties underlying IPCS were only *suggested* by previous work – in particular Miranda's Musical Culture work (2003) and his work with Martins (2007), and the Sequence Evolution paper of Gong, Zhang and Wu (2005).

Because of this it was not clear before the experiments that the needed complexity would be generated with no random behaviour (those systems which inspired IPCS all had random elements in some way embedded). However it is clear from – for example – Figure 71A (p243) and Figures 46 to 53 (p178), that sufficiently different tunes can be generated within the limits of IPCS's behaviour so far investigated. However it is also felt that the "edge of chaos" has not been firmly located in IPCS. This refers to the idea from Langton (1990) that when Cellular Automata rules become too stable the behaviour is uninteresting and monotonous, but when they are too unstable the behaviour is for chaotic and seemingly unstructured. But there is a point at which the rules are "just right", and unexpected ordered complexity emerges at the "edge of chaos". The Game of Life rules utilized by Miranda are an example of rules at this point

of complexity. Because IPCS has more parameters than a 1 or 2-dimensional Cellular Automata, the testing has not yet settled on finding a parameter space where a user can be ensured that IPCS will generate interesting behaviour.

However looking at the results like Figure 71A p243 it appears that some kind of balance between the similarity threshold and the number of cycles/agents may be fruitful. And similarly a balance between the spread of initial affective states, the affective influence and update rules, and the number of cycles/agents could be productive. Dahlstedt (2007) describes how a great deal of time was spent fine tuning the rules which would lead to the Ossia algorithm generating interesting music. He utilizes a significant number of heuristics which guide the music generation to stop it become uninteresting. However there is still no guarantee of all tunes being of interest. Ossia is designed as a fixed system, but IPCS encourages the user/composer to experiment with the balance of order and chaos in the system. So Ossia has the advantage of the composer knowing they are more likely to get an interesting output. But the disadvantage of Ossia is that it is almost a creative act in itself, it is *owned* by Dahlstedt because of the time he spent fine tuning the rules. Other composers would probably have less interest in using the system, as they may see it as a creative product of Ossia. Whereas the flexibility of IPCS (leading to its potential instability or monotony) is more suited to computer-aided composition.

Looking now at the other systems most closely related to IPCS – the multi-agent computer-aided composition systems – there are a number of similarities and differences highlighted by the results. The swarm music of Blackwell (2007) focuses on the mapping approach – but also focuses on being a live system. This allows live collaborators to interact with the musical swarm and push it out of less interesting areas and into more interesting areas. Because IPCS is not real-time, if it becomes locked into a self-perpetuating behaviour (e.g. rising tempo) the composer cannot break it out of

this. The Ant Colony system (Clair et al. 2008) is also a live system and has this live interaction possibility. Looking at the Heavy duty MAS systems now, the real-time MMAS (Wulfhost et al. 2003a; Wulfhost et al. 2003b) system highlights the limitations of the monophonic approach in IPCS. The MMAS agents have the ability to infer each other's tempo and thus synchronise, whereas IPCS agents build their music independently.

A similar synchronisation method like MMAS is used in the heavy-duty Andante system (Ueda and Kon 2003). The Andante and Musical Agent systems (Fonseka 2000) both require significant scripting from a composer, and are more a framework requiring algorithmic scripting when compared to IPCS. An IPCS user can begin generating non-trivial music with perhaps 10-15 minutes of having the system explained; e.g. being told the number of cycles and agents to use and then manipulating the initial affective state and similarity threshold. Only a few parameters need to be changed in IPCS to lead to significantly different music. Andante and Musical Agent have great flexibility but require coding from the ground up compared to IPCS. MAMA (Murray-Rust and Smaill 2005) has similar advantages and issues – it is real-time and synchronized with musical intelligence, but requires the user to provide musical fragments and rules for how they are used, as well as style constraints. It should also be noted that MAMA is a system designed so that agents communicate by music as well as make music. It is worth re-visiting that point since the results of the IPCS tests show that agents are able to send and receive emotional communication via music. For example in Figure 28 (p150) Agent 1's affective state updates when it hears Agent 2's tune. These communications then have an effect on the future music created and communicated.

CinBalada (Sampaio et al. 2008) and NetNeg (Goldman et al. 1999) are perhaps the closest of the heavy-duty systems to being a "simulation" of human musical composition. CinBalada's design motivation is largely to simulate certain types of

Brazilian Rhythms. This has the advantage of being based in a pre-defined and appreciated musical culture, but of course does not have as much flexibility of IPCS to be used as a computer-aided composition system. Similarly with NetNeg – which uses 16th Century counterpoint rules. Another element with NetNeg is its fixed population size.

One element with many of the above heavy-duty systems is their motivation to utilize multi-agent systems for producing multiple lines of a composition/performance in *parallel*. This has the effect that after a run of the system, one composition will be produced. An interesting element of IPCS is that after a run, there will be as many compositions as there are agents. A user can potentially audition multiple compositions to find the one they prefer. This can be compared to the work of Ramirez et al. (2008) which utilizes Genetic Algorithms and multi-agent systems to produce multiple results at the end of a run. They argue that because music performance is a subjective task, the user should be presented with multiple results so they can subjectively choose the one or ones they want. It can be seen in the IPCS results – for example Figures 71A (p243) and 73A (p245) – that although there are often a number of similar tunes in all the agents at the end of the run, they are not all the same. Thus there is some variety for a user to choose from.

A close relative of IPCS in the heavy-duty systems is Inmamusys which uses multi-agent systems to compose affectively labelled music (Delgado et al. 2008). Now that it has been seen that IPCS does provide the emergence of non-trivial music from simple rules, the status of Inmamusys as an MAS will be re-evaluated. In many ways Inmamusys is almost indistinguishable from an object oriented program. It is not clear how the multi-agent nature of Inmamusys contributes to the composition described in the paper. Whereas IPCS's music clearly emerges from simple interactions in a homogeneous multi-agent system. The testing of Inmamusys utilize 20 listeners, like

CMERS, but only 4 examples – with emotion labels “worry”, “happiness”, “chaos” and “worry”. The results are not given numerically but in pie charts and reading these leads to the system being 40% accurate for “worry”, 90% accurate for “happy”, and 60% accuracy for “chaos” – giving an average of 58% - around the same as IPCS. However, the Inmamusys testing included the option for the user to mark up results as “indifferent” or “fear”, as well as the 4 that were actually being expressed by the system. This would lead naturally to lower results because of less of a pigeonholing than IPCS. So it could be estimated that Inmamusys is *at least* as successful as IPCS in emotional communication. However Inmamusys is designed for emotional communication and is a heavy-duty MAS.

A number of the heavy-duty systems allow humans to participate in the process, or are planning to investigate that potential. IPCS does not allow this to the same degree because it is not real time. However, there is no reason why a user could not change the initial tune of a number of agents to be larger motifs, rather than middle C. Thus in a sense the human composer can become part of the process. This is really an element of future work, but is included here for direct comparison with real-time heavy-duty systems.

The intermediate MAS discussed earlier are either modelling systems, rhythm based, or not implemented. Looking at the non-implemented Computer-Aided Multi-Agent Music Composition system (Dahltstedt and McBurney 2006) the main thing to observe is that in spite of no explicit musical goals, IPCS has been shown to generate developing musical patterns. The A-Rhythm (Martins and Miranda 2007) system does highlight the timing weaknesses in IPCS in its method. If IPCS could generate rhythmic structure like A-Rhythm then it would be a significantly improved system. From another perspective, IPCS unlike A-Rhythm does generate *both* rhythm and pitch, though the second more successfully than the first. And it does it utilizing emergence in a homogeneous multi-

agent system, like A-Rhythm. The key extra element found in IPCS is the affective interaction protocol and agent state.

It is clear that in spite of the IPCS tendency to repetitiveness over larger time scales it is able to produce phrases and themes, and therefore is a step on from Miranda's (2003) Cultural Evolution approach, and the Gong et al. (2005) Sequence Evolution approach which have small fixed numbers of notes. However it has proved harder in IPCS than in the Sequence Evolution approach to understand how agent states will shape a tune. In the sequence Evolution system, agents' musical preferences can be adjusted in a more intuitive way. In IPCS a musical "meta-property" is used – affective state – and a very simple model of it. Hence it is not so transparent what will come out of the agent interactions, based on their affective parameters. It is interesting that in spite of the variation between agent tunes at the end of runs that has been mentioned - in Figures 71A (p243) and 73A (p245) - there is often a similarity in the tunes across the population (as can be seen in these two figures). This is reminiscent of a common repertoire of the community emerging in Miranda's (2003) system. Another related element of IPCS is the affective influence agents have on each other through their music, which can lead to agents affective states moving closer, as seen in Figure 28 (p150) and Figure 31 (p151) – reminiscent of the equilibrium found in many modelling MAS.

5.5.2 Expressive Performance

One immediate effect of expressive performance (i.e. k not being 0) is the requirement to increase the similarity threshold or the number of cycles – as seen in Figures 39 and 40 (p165) (Tables 38A and 39A, p262). This element is fairly unique to IPCS as the only other expressive performance multi-agent system (Zhang and Miranda 2007; Miranda et al. 2010) uses a different type of thresholding – agents only alter their

expressive performance if another agent's expressive performance is judged as being "better" than theirs. Like IPCS, their approach was evaluated not by listening tests but by examining in examples and parametric tests whether the resulting performance expressed the note structure of the system. However this evaluation was not done as explicitly as in IPCS because the note groupings in the music used (Fur Elise, Beethoven) were auto-analysed using LBDM (Cambouropoulos 2001). Furthermore even if LBDM was perfect, it does not follow that their system expresses the *hierarchical* note groupings.

This approach to evaluation, without listening tests, was also followed by SaxEx (Arcos et al. 1998), Hierarchical Parabola Model (Todd 1985; Todd 1995), and Ossia. (This should be qualified by saying that Ossia was not designed as an expressive performance system.) Of the seven CSEMPS in Chapter 2 which are evaluated by listening tests – CaRo (Canazza et al. 2000; de Poli 2004), Emotional Flute (Camurri et al. 2000), Composer Pulse (Clynes 1986), Kagurame (Suzuki et al. 1999; Suzuki 2003), CMERS and PLCG/Distall (Tobudic and Widmer 2003a; Tobudic and Widmer 2003b) – 3 of them are evaluated for their effect on emotional communication (as opposed on their success in expressive performance techniques *per se*). The two systems with the closest goal of IPCS in terms of expressive performance are the Hierarchical Parabola Model and PLCG/Distall. Both are focusing on the relationship between expressive performance and the hierarchical structure of the music. Neither combined composition and expressive performance systems, so both require an in-depth analysis of the music to be expressively performed. The Hierarchical Parabola Model needs a multi-level TSR analysis to be done (Lerdahl and Jackendoff 1983); and the PLCG/Distall requires a human musicological analysis.

PLCG/Distall does not give examples of how the hierarchical structure is expressed, but focuses on evaluating how successful the performance is. It can be seen from Figures

78A to 80A (p249) how the expressive performance leads to tempo in IPCS expressing the hierarchical grouping structure of the music. The IPCS advantage of combining expressive performance with composition also bears comparison with Director Musices itself – which normally requires manual mark-up before applying Phrase Arch. Though it is key to note that the analysis methods utilized to prepare for an expressive performance in all three of the systems discussed above focus on perceptual grouping, whereas IPCS focuses on the grouping *as built by the IPCS composition*.

Kagurame has the ability to provide expressive performance for homophonic music (i.e. melody with chords). A similar approach is used in DISTALL to allow expressive performance of homophonic music. Both the DISTALL and Kagurame systems have been successfully evaluated with homophony.

A final note must be mentioned relating to Ossia. The expressive performance in Ossia is more deeply integrated into the composition process, and could be described as being "more emergent" than IPCS. Also from a subjective point of view, informal listening to Ossia output and comparing with IPCS would lead to the conclusion that – in general – Ossia can give more interesting and pleasant performed output than IPCS. However, as has already been noted, Ossia was not *designed* as a combined composition and performance system, nor tested as one – e.g. does it express its compositional hierarchical grouping structure? Furthermore – as has already been noted – Ossia is not necessarily suitable for use in general by composers for reasons already discussed. These factors, combined with the results found for IPCS in the previous chapter, support the idea that IPCS is still a significant contribution, in the context of Ossia, to Composition and Expressive Performance systems.

5.5.3 L-Systems

Lindermeyer Systems (or L-Systems) (Rozenberg and Salomaa 1980) were originally developed to model the development of plant growth. This is done using an approach similar to that found in the re-writing rules of formal grammar systems (Chomsky 1956). They have been used in algorithmic composition (McCormack 1996) and have enough similarities to the IPCS tune development approach to warrant a comparison here. A simple L-System model can be described as a set of variables, an initialised state and rules which transform variables, as follows:

Variables – x y

Initialisation – x

Rules – [x→xy], [y→x]

A series of iterations are run starting with the initial string 'x' and applying all rules that can be applied at each iteration. Table 17 shows the first 6 iterations, and the rules applied at each step for the first 6 iterations.

Iteration	**Result**	**Rules applied**
0	x	n/a
1	xy	[x→xy]
2	xyx	[x→xy] **xy**y [y→x] xy**x**
3	xyxxy	[x→xy] **xy**yx [y→x] xy**x**x [x→xy] xyy**xy**
4	xyxxyxyx	[x→xy] **xy**yxxy [y→x] xy**x**xxy [x→xy] xyx**xy**xy [x→xy] xyxxy**xy**y [y→x] xyxxyxy**x**
5	xyxxyxyxxyxxy	…
6	xyxxyxyxxyxxyxyxxyxyxxyxxyxyxxyxxy	…

Table 17: Simple L-System growth

To see how this relates to IPCS consider the version of Table 9 (p111) labelled as Table 18. In this table the structure markings used in Table 9 have been removed and two columns have been added to the right. One is the way that the tune transformation could be represented as a replacement transformation like that used in L-Systems, and a fourth column showing a possible general transformation rule which has been applied. Based on this the tune development has some relationship to the following L-System:

Variables – x_0, y_0, y_1, …, y_M

Initialisation – x_0

Rules – [$x_0 \rightarrow x_0y_0$], [$y_i \rightarrow y_iy_{i+1}$]

Tune Added by Agent	**Agent new Tune**	**Replacement**	**"Rule"**
Initial tune x_0	x_0		
y_0	x_0y_0	$x_0 \rightarrow x_0\ y_0$	$x_0 \rightarrow x_0y_0$
y_1	$x_0y_0y_1$	$y_0 \rightarrow y_0y_1$	$y_i \rightarrow y_iy_{i+1}$
y_2	$x_0y_0\ y_1y_2$	$y_1 \rightarrow y_1y_2$	$y_i \rightarrow y_iy_{i+1}$
y_3	$x_0y_0y_1y_2y_3$	$y_2 \rightarrow y_2y_3$	$y_i \rightarrow y_iy_{i+1}$
etc…	…		

Table 18: Compositional Growth in IPCS

A key element of the above system which makes it unusual is that the variable set is limited but needs to be as extensive as the generative ability. This is not the case usually with an L-System which uses a relatively small simple set of variables to generate significant complexity. Also in an L-System all rules which can be applied are applied

at each iteration. Systems which do not have this universal application approach are more similar to formal grammars. However, even if IPCS is considered in the context of systems which do not require universal replacement rule application (e.g. grammar systems), IPCS's use of similarity thresholding and affective-recomposition/performance lead to a significant difference. At each line in replacement, there is a decision – is the similarity close enough? And before replacement is done, the replacing symbol is further transformed. So in the special case when these elements are disabled in IPCS, it can be comparable to generative grammar systems.

Both L-System and generative grammars have been applied to algorithmic composition (McCormack 1996)(Morgan 2007). This has often involved a mapping from the graphical interpretation of the L-System onto musical features. Consider the scenario where the mapping involves one note for each node of vertex of the graphical representation. The addition of similarity threshold in IPCS can be viewed as a method for reducing the exponential growth in the number of notes in the composition. The affective re-composition transformations could be viewed, in this context, as an additional layer of musical development process on top of the string replacement generating the self-similar note structure. Not all L-System musical approaches use such a simple mapping. But even in the more complex cases, a key differences remains between information-swapping multi-agent systems and L-Systems systems. An L-System has fixed rules. Although both have elements of self-similarity, the multi-agent system is in essence generating new rules to potential levels of complexity which could not have been envisaged by the initiator of an L-System.

Chapter 6 – Conclusions and Future Work

This, the final chapter of the thesis will draw conclusions based on the results and discussions, and discuss possibilities for future work.

6.1 Conclusions

At this point, the contributions of this work will be re-stated from the first chapter:

1. The demonstration of the applicability of intermediate multi-agent systems to algorithmic composition.
2. A proof of concept that a multi-agent system can develop non-trivial melody pitch structures through affective interaction of agents without explicit melodic knowledge.
3. A demonstration that multi-agent social structures can generate musical structure on thematic and sectional levels as well as on a note or phrase level.
4. A proof of concept that combined algorithmic composition and expressive performance system applies expressive performance rules in a way that expresses the musical structure and without having to do an explicit musical analysis of the composition.
5. A music-emotion analyzing model which takes as input a monophonic MIDI file and estimates its affective content.

Contribution 1 has been addressed by Chapters 3 and 4, IPCS is an intermediate system – it is not a swarm system nor does it utilize any significant AI processing (e.g. expert systems). The affective inference system - a single linear equation – and the affective expression system – three linear transforms based on a handful of parameters – could not be classified as significant AI processing. It use for algorithmic composition has

been supported within the scope of the time and space available for the work. In particular it was felt that the experiments showed how number of agents and cycles, initial affective state, affective state update rate and similarity threshold could be utilized by a user in shaping a variety of tunes. (It is also worth noting that IPCS has been used by the author to compose a piece for solo piano – see Appendix 4).

Contribution 2 has also been demonstrated because the agent has no explicit knowledge of how melodies should be built: which notes should follow which, how they should repeat, etc. However Chapter 5 Section 5.1.4 p190 gives a list of properties was given for non-trivial melodies and these were largely fulfilled by a number of outputs demonstrated from IPCS in the various piano roll figures. An agent's only compositional knowledge of music is its ability to extract affective data from the whole and impose affective features on the whole. Contribution 3 was also demonstrated in Chapter 4 Section 4.5 p152. There were two demonstrations: (1) it was demonstrated diagrammatically how the interaction structure would relate to the musical structure; (2) an example were given showing the musical structure building up and how it related to the agents social structure.

Contribution 4 was supported in Chapter 4 Section 4.6 in particular p166 where a demonstration was given showing how IPCS utilized integrated expressive performance to express the grouping structure of its tunes. And an example tune generation was given supporting this. The running of multiple examples to generate a statistical demonstration was made difficult by the fact that IPCS is an *integrated* system – in other words as soon as an musical features - apart from expressive performance elements – are added to the mix, the expressive performance elements become part of the combined whole and hard to separate out. However it was seen that in a pure "non-compositional" system, the expressive performance expressed the structure. And although in the compositional mode new timing and loudness information is added, it

does not remove the expressive performance elements; hence they will remain, continuing at some level (depending on the size of the k-value) to express the structure. Contribution 5 was supported in Chapters 4 and 5 in two ways: (1) offline tests in Chapter 3 Section 3.6 p114 gave a sufficient accuracy relative to human accuracy; (2) the relationship between the Linear Estimator's final affective estimate and the listening accuracy described in Section 5.2.4 p194.

6.2 Future Work

This final section of the thesis will examine ideas for future work on IPCS. It will divided into eight subsections:

a. Further parametric investigation
b. Composition issues
c. Initialisation
d. Agent interactions
e. Structure
f. Expressive performance
g. Affectivity
h. Listening experiments

6.2.1 Further parametric investigation

It became clear when discussing results that there is a significant element that could be added early on in the parametric testing of IPCS – and it is the effect of different size pitch, onset and loudness influences. These parameters were used in the testing of other elements of IPCS, but not investigated themselves – due to what seemed an obvious

intuitive sense of their effect. However this assumption now seems erroneous due to the complex nature of multiple IPCS interactions. So this would be a priority element of future work. Another element which was not investigated was the effect of changing Seed Note Pitch. Setting this significantly higher or lower will tend to cause tunes to be seen to have different affective states, by equations (5) and (6). Similarly the seed note's duration could be investigated – the only values utilized so far have been 0.5 and 1 seconds.

It has already been highlighted that that a better sense needs to be gained of the parametric areas of productive complexity in IPCS – i.e. it's "edge of chaos". One method not fully investigated was the usage of larger numbers of agents and cycles and maximum note counts but with stricter interaction controls – e.g. similarity threshold and interaction coefficient. As well as manually investigating, it may be possible to attempt this using Genetic Algorithms. Key parameters of IPCS could be encoded as genomes and the fitness function could either be a human scoring by eye, and/or various automated musical evaluation methods (Kirke and Miranda 2007).

The cycling approach to IPCS had a fairly profound effect on which agents developed which tunes. In many MAS there is a random or more complex process deciding which agents interact with which. In IPCS the linear movement through agents leads to more predictable interaction networks between agents. There was an example given where odd agents interacted with even numbered agents, and it was seen how this changed the interaction networks. It may be possible to experiment with different approaches to see which gives the most productive musical feature set.

The interaction coefficient could be more deeply investigated in terms of how it behaves. It would be useful to show what regions of the various interaction coefficients actually lead to the division of the agents in providers and receivers; and to show what

effect the Interaction Coefficient actually has on the music subjectively. It would also be interesting to investigate in more depth how this behaviour varies with different other parameter settings such as similarity threshold and affective update rates.

Something which was utilized once but not again was the use of neutral agents – i.e. agents initialised to 0 valence and arousal. These agents have the ability to build tunes with the minimum of affective influence on the tune. It would be interesting to see if there are parameter sets which could actually cause neutral agents to compose pitch, and see what sort of tunes develop. Furthermore adding them to the mix of an agent group may help to mitigate some of the effects of self-reinforcing music features such as tempo. When a non-neutral agent combines two sub-tunes and passes them on, it will always change the tunes in some way – not so a neutral agent.

6.2.2 Composition issues

The problem of repetitiveness in IPCS, caused by its recursive nature, could be further investigated. Like Ossia, IPCS is limited in its ability to generate larger scale ABC type structures. One possibility is to have a "large" transformation every so often. In this case some agent's tunes are so radically transformed as to turn them from an A to a B or a C. This could be achieved by utilizing some form of basic scripting language which defines how the system behaves as cycles continue, perhaps reinitializing some agents in some way, etc. The continuous and often linear nature of IPCS's simple transforms is a significant limiting element, so the issue could be addressed by looking for more complex non-linear transformations. The related issue of runaway growth in timing could perhaps be further addressed using the already discussed idea of timing palettes and quantization. This would force timing and loudness into certain pigeon-holes, just as pitch is forced into certain notes values. This approach would also make the output of

IPCS more notatable if it is to be played by a human. Loudness could be investigated in the opposite way – i.e. find methods to make it smoother, for example crescendos and so forth. At the moment loudness tends to change in larger steps and over phrase-size time scales.

The timing quantization may help to address another issue with IPCS – the fact that duration is not compositionally utilized. One possibility would be to have a different palette for duration than for onset. Another is to have a duration transformation equation which is independent of the onset equation. This could be done by making the duration equation focus on how legato or staccato a note is - both of which have affective implications. Another way of approaching the timing issues could be to have agents not always add performances to the beginning of their tunes. For example they could add them alternately to the beginning and end of their tunes. In this case the self-enforcing timing or loudness sweeps would no longer sound continuous but would be broken up in a way which may be more compositionally interesting. The whole issued of timing could actually be sidestepped in one piece of future work – remove all timing elements and see how IPCS functions as a system for generating semi-repetitive patterns – e.g. for minimalist composition.

It would be helpful to investigate the melodies generated from a more musicological point of view to see what the precise effects of – for example - the similarity threshold and affective state influence - have on the melodic properties. This could perhaps be done by an independent musically trained person asked to give more in-depth comments. It could also be attempted using automated analysis tools, e.g. the work of Hamanaka, et al (2004).

6.2.3 Initialisation

At the start of interactions all agents in IPCS were given the same single note value to demonstrate the emergence of music by multi-agent interactions. However there is no reason that this should be the case outside of this demonstration. Agents could be initialised with different notes. In fact agents could be initialised with multiple notes by a human user – giving them pre-composed phrases. These approaches could be investigated to see their musical results. As well as more complex note initialization, it is possible to imagine more complex affective state initializations. Agent's affective states could be initialized anywhere on the continuous two dimensional space of valence and arousal. Different initialisation patterns could be investigated – for example a sine wave of arousal and a cosine wave of valence across the agents.

Random initialisation methods could be used across various IPCS parameters. This could provide an extra element of exploratory approach for the user. Another way to look for new areas of interest in the space would be to initialise agents with non-global values in some of the parameters which are currently global. For example initial Interaction Coefficient parameters are currently global across all agents, as are agent affective state update rates and affective influence on pitch, onset and loudness. Some of these could be potentially initialised non-globally and/or randomly.

6.2.4 Agent Interactions

Agent interactions in IPCS are currently a closed system. Once the user sets them running they cannot be influenced. However it is possible to imagine opening up agent interactions to the user at certain cycles, allowing the user to interact with agents

themselves acting like an agent, or like a “god” of the system. Possible interaction protocols could be investigated to allow this. Another way of inserting an extra element into interactions is to allow certain agents to be more “affectively competent” than others – to insert an error into the affective feature estimation linear equation. This error could be different for different agents, or global. Alternatively, or additionally, entirely new ways of making tune adding decisions could be investigated. The similarity threshold may not be the only possible approach – perhaps the use of a non-meta-comparison would be more effective – i.e. compare actual tune features. Similarly there is scope for investigating if an Interaction Coefficient function based purely on tune length is the most efficient at generating a musically relevant interaction structure in the agent group.

An additional element to consider in agent interaction is an investigation into alternatives to the way in which agents update their tunes. Currently in IPCS agents add the tune to the end of their own tune. In other intermediate MAS modelling systems it is more common to use some form of “influence”. In other words an agent updates its tunes based on certain features of the tune it is hearing, as opposed to just adding the tune to the end of its own. The listening agent could still extend the length of its tune – for example if the tune it is listening to is longer than its own, the agent could extend its own tune by repeating elements of it. But the actual features of its tune may be adjusted in some way by the incoming tune to make the tunes more similar.

6.2.5 Affectivity

The model of emotion in IPCS – valence and arousal - is a simplification. A more accurate or higher dimensional model may have compositional advantages. Similarly emotional influence by one agent on another is a linear and unbounded process; and it

would be useful to see if making it more realistic or complex has musical benefits. The system currently allows agents affective state to go from minus infinity to infinity; some bounding system may be helpful in limiting the runaway self-perpetuating music feature transforms that can occur in IPCS. The linear affective music feature estimation system could definitely be improved upon –making it polynomial for example.

The affective Similarity Threshold and its effect on longer term structure has not been fully investigated. There was less detail than expected on its precise musical note by note effect or its effect on larger scale structure. Further work could be done looking in to this. There could also be further work on the inter-agent affective state influence, and how it changes melody structure. Also here was some evidence for the emergence of common affective states due to affective influence; it would be useful to investigate this further and see and what its precise effect on music is. The same goes for the affective dominances found in the experiments – for example the average agent affective states were often attracted to the "Tender" quadrant. It would be helpful to know why this was, how it could be better controlled, and what its effect on the music was. Finally, both of these elements point toward the possibility of IPCS being used as some form of modelling tool for music and affectivity in society – though this was not the intent, and it has not been evaluated this way. Future work could perhaps be done in collaboration with scientists more embedded in the relevant fields.

6.2.6 Expressive Performance

Currently in IPCS note groupings are expressed by the expressive performance based on how they are built, not how they are perceived. It would be helpful to find ways of testing how closely the current algorithm's note groupings are to human perceptions. To

generate perceptual note groupings may require some change in the interaction algorithm, for example the similarity and interaction coefficient system.

It was difficult to judge how subjectively pleasant the repeated recursive addition of Director Musices rules are in IPCS. It may be possible to do a recursive reverse engineering of the rules to find a set of more primitive rules from which DM-type behaviour will emerge. Some work could be done experimenting with recursive methods of phrase arch, i.e. what k-Values etc should be used. One possibly interesting side effect of the recursive application of k-Values – and the fact it is embedded in the composition process, is that it may be possible to actually overdrive the expressive performance algorithms to create unique compositional elements. This may also be a way of creating more interesting rhythms in IPCS.

Currently both IPCS and CMERS use an approach to emotional expressive performance which focuses on approximating some features by utilizing a certain quadrant-based approximations. For example all affective states with positive valence and positive arousal are considered to be "Happy" in terms of selecting parameters for expressive performance in IPCS. It would be useful to investigate if the expressive performance rules could be made more continuous – for example utilizing some form of linear interpolation within each quadrant. This would lead to linear mappings of the type found in the IPCS compositional elements. It was desired to focus on a deterministic algorithm in designing IPCS and to avoid random methods as a way of creating complexity; however another change which could improve the expressive performance is including some elements of randomness. There is a DM rule for doing this, designed to simulate small human random deviations in timing and loudness.

Other ways of extending the evaluation of the expressive performance would be to do a comparison with the other multi-agent system for expressive performance – the Zhang

and Miranda (2007) system. It may be possible to generate IPCS tunes which are the same in basic structure when k is zero and when it is non-zero. In that case the tunes could be put into the Zhang/Miranda system and the results compared to the output of IPCS with k > 0. Another way of evaluating IPCS expressive performance would be to get musicologists – and automated systems like LBDM - to divide up some IPCS tunes into their hierarchical structure, and see how accurately they can do it. If there are large inaccuracies this supports the idea of combined expressive performance and composition algorithms providing more accurate performative expressions of structure. However a method would need to be found for ensuring that the grouping in the IPCS tunes was also their perceptual groupings. Otherwise the human musicologist may differ from IPCS because IPCS is perceptually "wrong" in its grouping.

Ossia embeds humanized performance in its composition at the deepest level, as a result of its recursive algorithms. This came out as an accidental result of its design. It would be interesting to see if the expressive performance elements of IPCS could be deliberately redesigned to work at a deeper level in the composition. Though one disadvantage of the deep embedding in Ossia is that removing the humanization/expressive performance would cause the composition to unravel, whereas in IPCS the expressive performance can be turned on and off.

6.2.7 Listening Experiment

A key gap in the listening experiment was seeing if the IPCS expressive performance algorithms increase the affective perception as much as CMERS's use of expressive performance does for it; experiments could be run examining this. The listening experiment could be extended to get a clearer statistical view of IPCS properties – perhaps extending them to 20 listeners, like CMERS. A larger number of tune samples

could be used as well – keeping the same initial affective states but adjusting other IPCS parameters. Listening experiments could also be done with a wider range of mixed affective states – for example half and half (rather than 6 of one and 2 of the other). Also the assumption that any agent majority initialized in a state implies that state should be communicated could be modified. For example the initial state AAAAAATT could be calculated as having a particular valence and arousal average across the 8 agents, and the ability of the agents to communicate *that* average would become the measure tested in listening tests. It would furthermore be useful to collaborate with a music psychologist in designing the tests to maximize their usefulness and validity – for example different ways of users filling in perceived emotion, and different labels or scoring systems for "enjoyable" or "interesting".

A control group for the tests would help to clarify the implications of the results. For the control group, the IPCS compositional transformations could be replaced by a set of randomly generated transformations based which would generate feature transformations of the order of those found in the actual compositional transformations, but in an affectively unspecified direction. A similar sized group of listeners could be tested, with each having their own set of randomly generated transforms. The results would give a baseline with which to compare the results for the original non-random compositional transforms. It may help to clarify any affective or pigeonholing bias and expectations – for example would the users' accuracy for each of the four affective quadrants really be 25% when transformations where essentially random?

Other parameters of IPCS could be investigated for their effect on listeners' experience – for example interaction coefficient. The inter-agent Affective State Update was set very low for the listening experiment, it would be helpful to include it at a higher value for a future listening experiment. Listening experiments involving musicological

evaluation by experts would be enable greater insight into IPCS's strengths, weaknesses and flexibility.

Appendix 1 – MIDI

MIDI (Musical Instrument Digital Interface) is a standard music representation that is utilized in IPCS. MIDI data is digital and stored in a standard file format. The elements of MIDI used in IPCS relate only to MIDI's ability to represent tunes. MIDI files used in IPCS thus represent a list of notes. The notes can represent a polyphonic tune or a melody. A note can have a number of parameters. MIDI files are utilized in IPCS using a programming tool called the Matlab MIDI Toolbox (Eerola and Toiviainen 2004). The toolbox makes key parameters in a MIDI note file available to a programmer. For each note in IPCS, 4 parameters in the MIDI toolbox are used as described below.

Note Onset

The time in seconds when a note starts, e.g. 0, 0.1, 1.5, etc. The start of the MIDI file is time 0.

Note Duration

The length of the note in seconds, e.g. 0.33, 1, 2.

Note Pitch

The pitch of the note – integer value 0 to 127. Middle C is 60, Middle C# is 61, B below Middle C is 59.

Note Amplitude

The loudness of the note – integer value 0 to 127. 0 is silent, 127 is maximum loudness.

For IPCS's purposes a MIDI file consists of a list of parameters of this type, with eac of the four parameters given for every note represented in the MIDI file.

Appendix 2 – Piano roll and Loudness Graphs

This Appendix contains the large piano roll and loudness graphs generated in the experiments in Chapter 4.

Experiment Set 2 – Initial Affective State

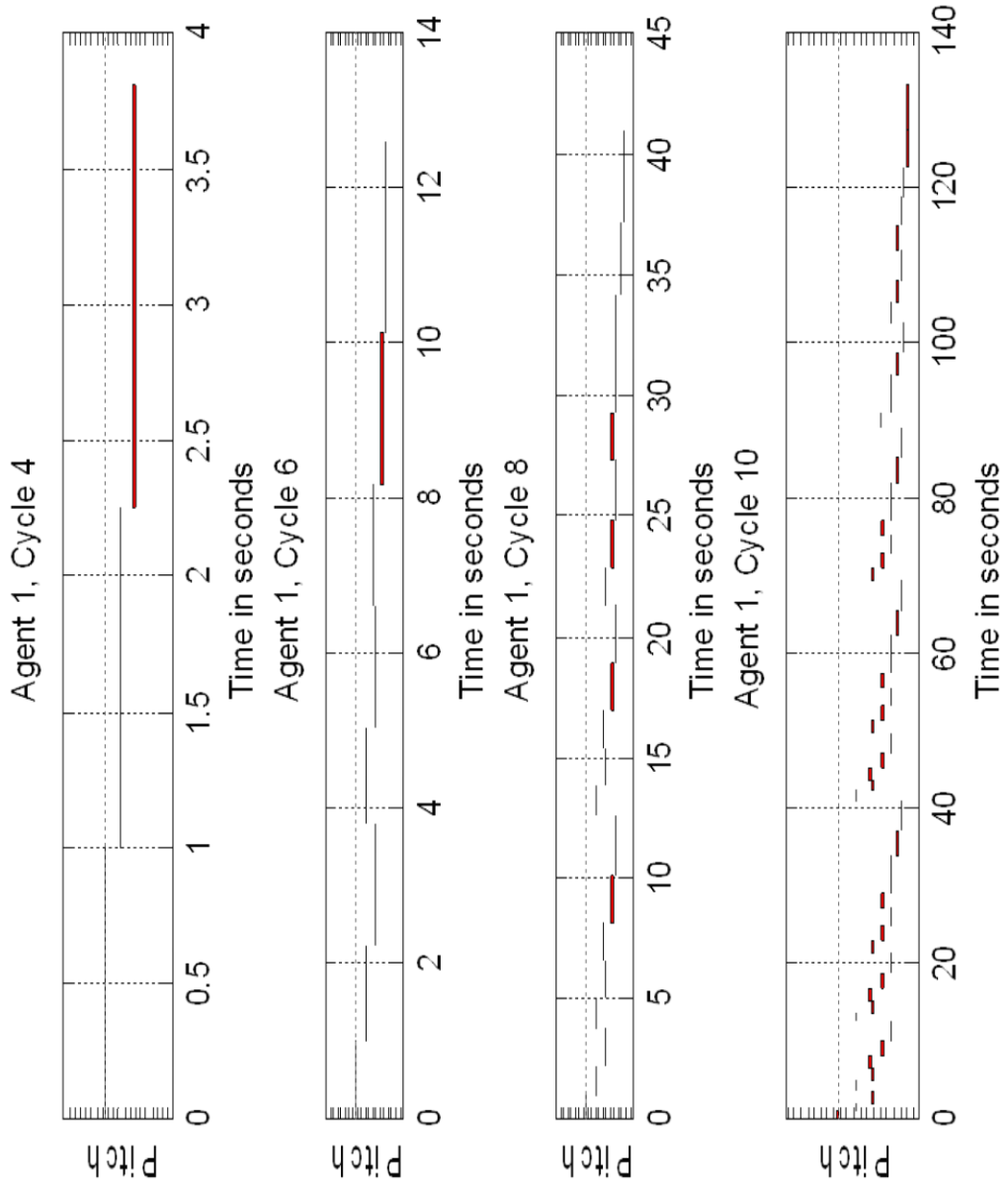

Figure 54A: Tunes for "Sad / Sad" – Agent 1

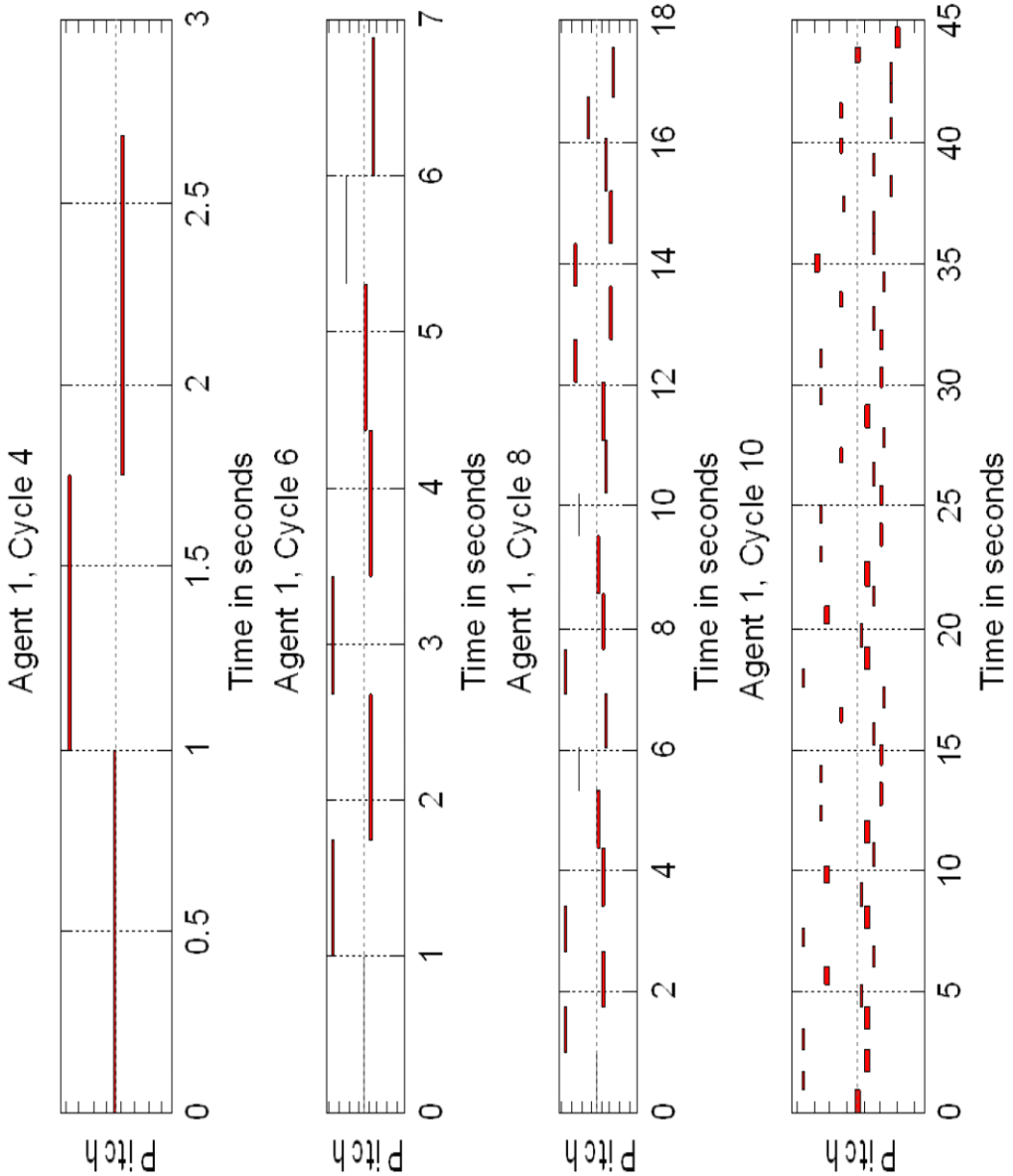

Figure 55A: Tunes for "Sad / Angry" – Agent 1 – Row 2 of Table 25A

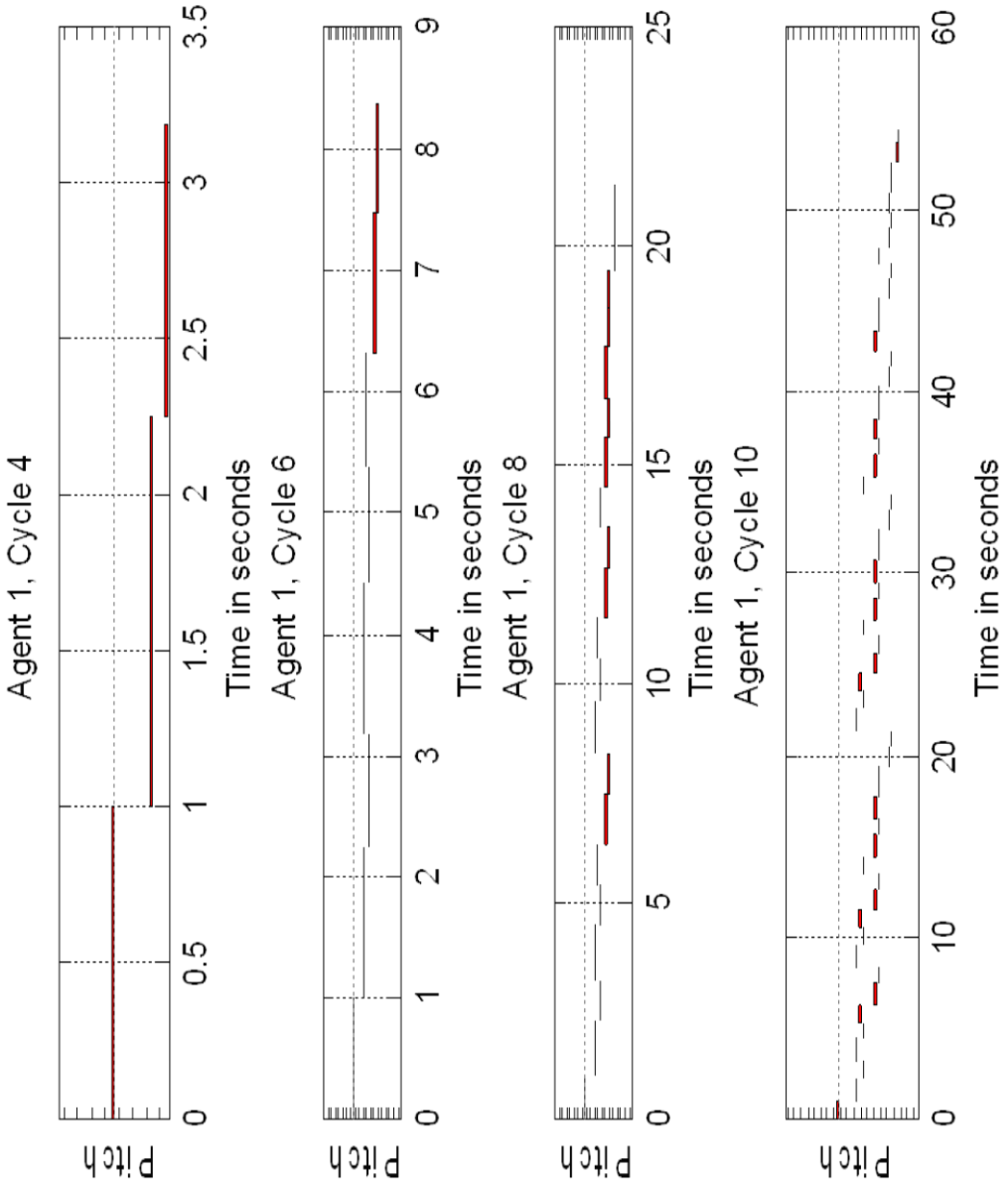

Figure 56A: Tunes for "Sad / Happy" – Agent 1 – Row 5 of Table 25A

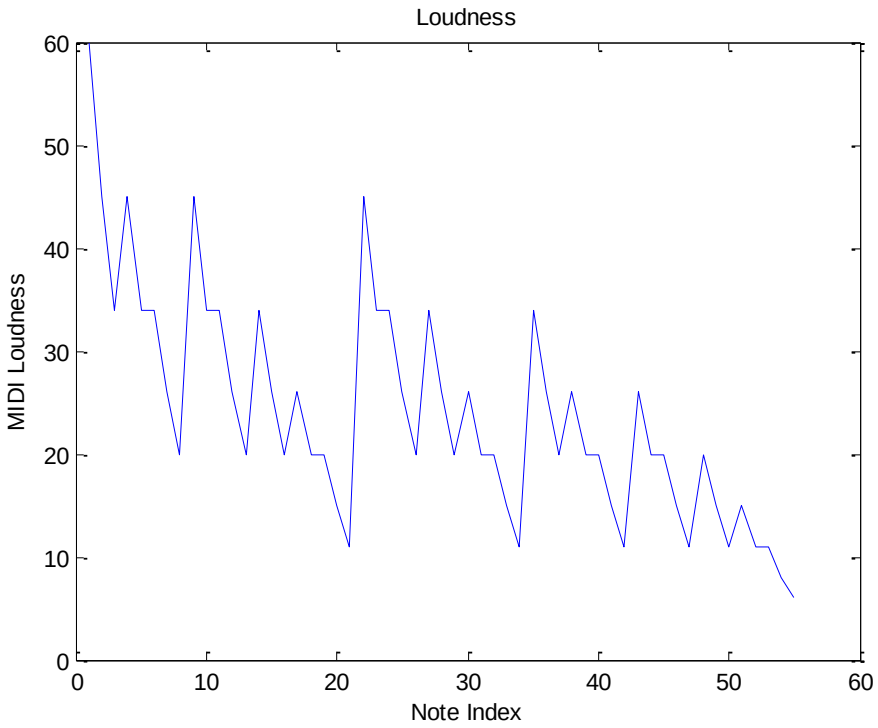

Figure 57A: Loudness for "Sad / Sad" – Agent 1 – Row 1 of Table 25A

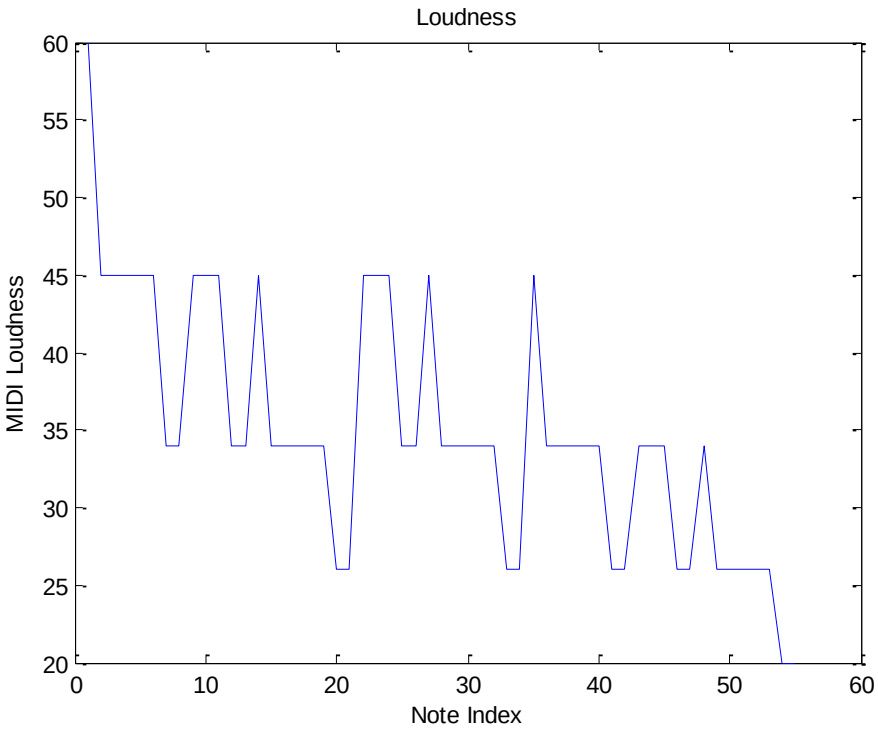

Figure 58A: Loudness for "Sad / Angry" – Agent 1 – Row 2 of Table 25A

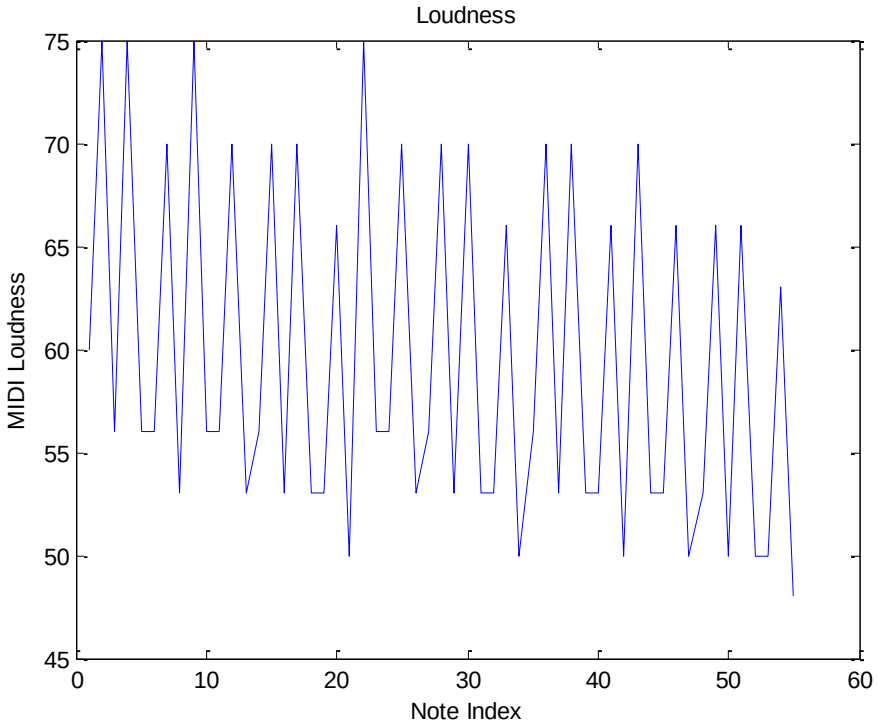

Figure 59A: Loudness for "Sad / Happy" – Agent 1 – Row 5 of Figure 25A

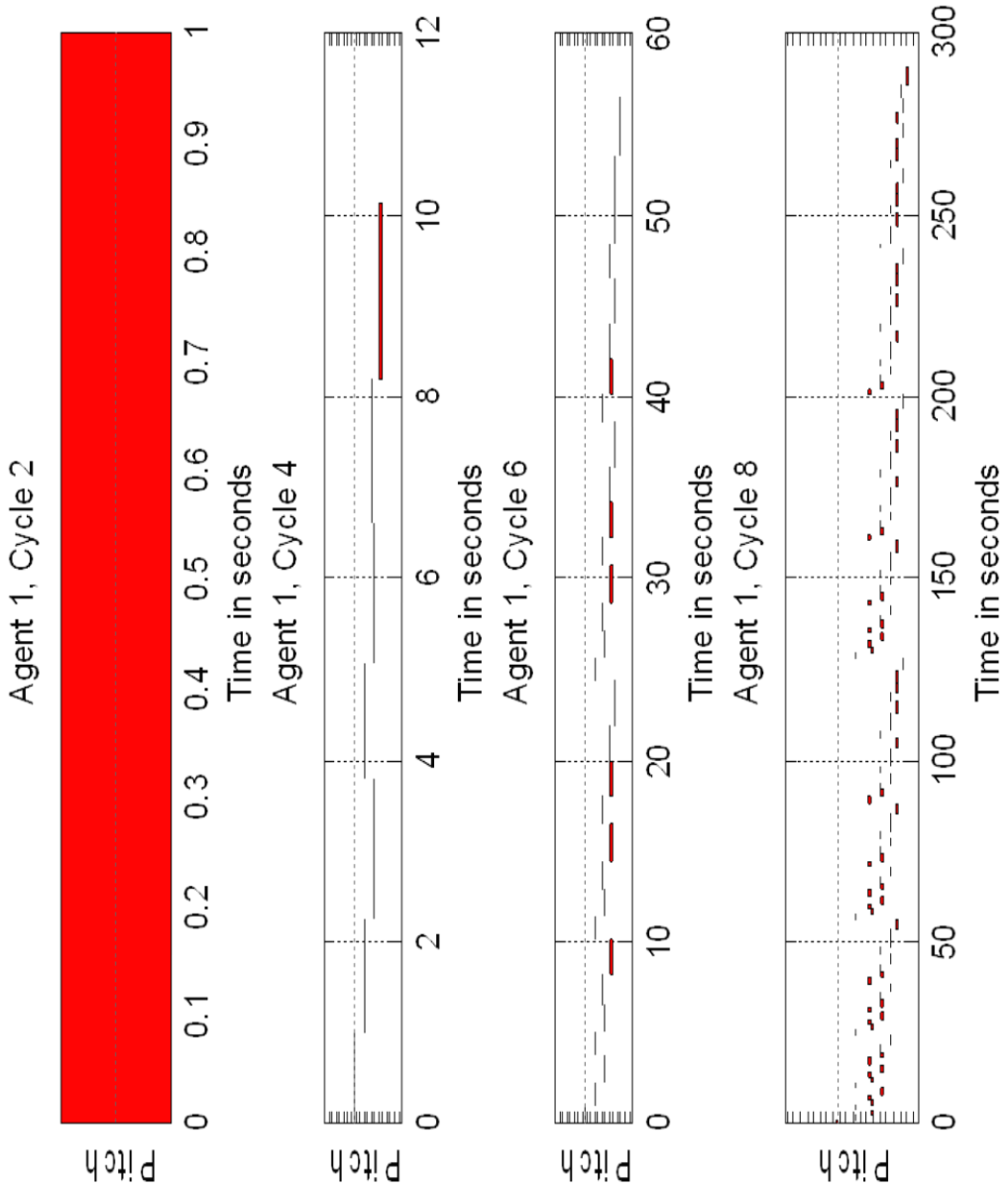

Figure 60A: Tunes for all "Sad" – Agent 1 (8 Agents)

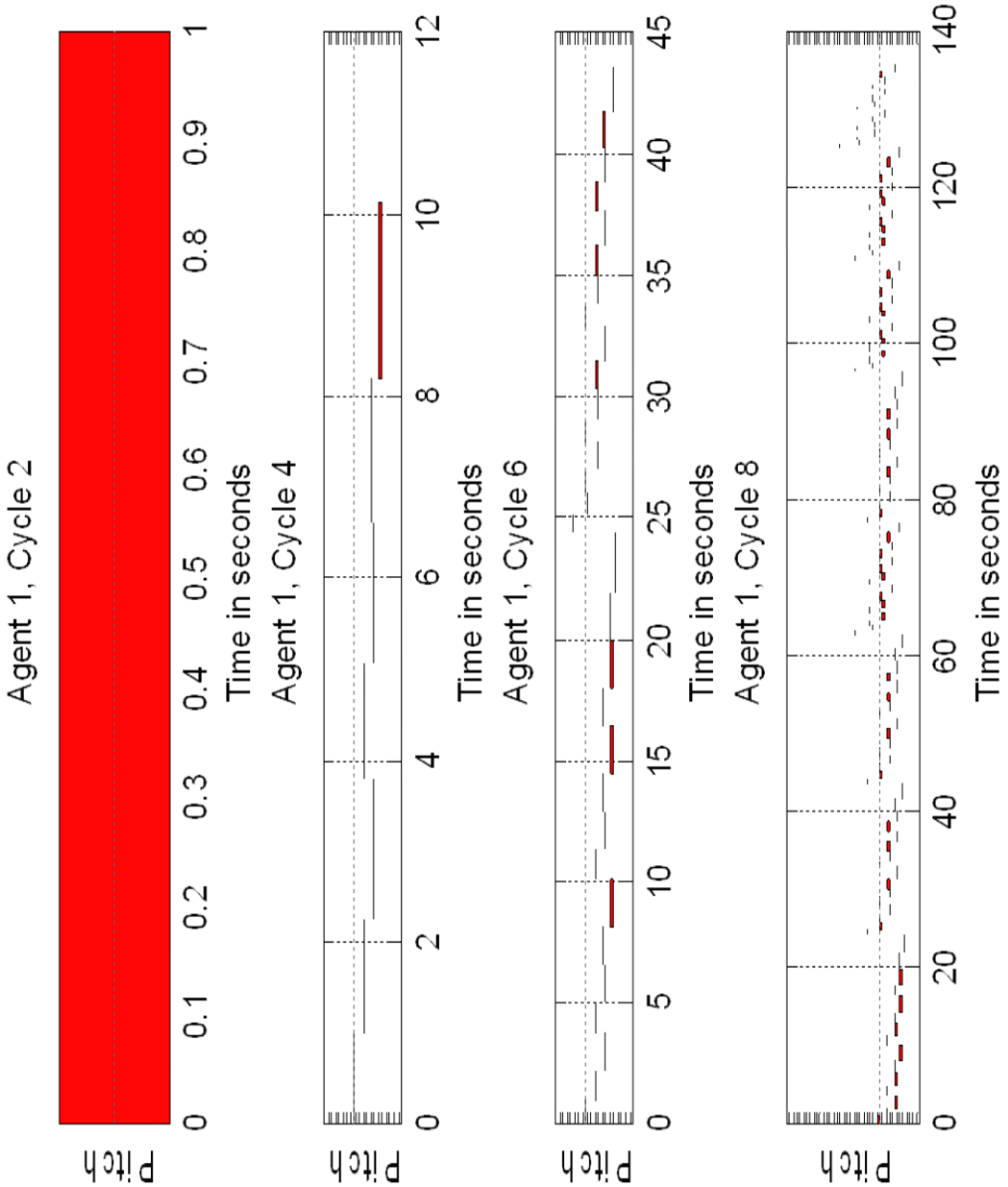

Figure 61A: Tunes for half "Sad", half "Happy" – Agent 1 (8 Agents)

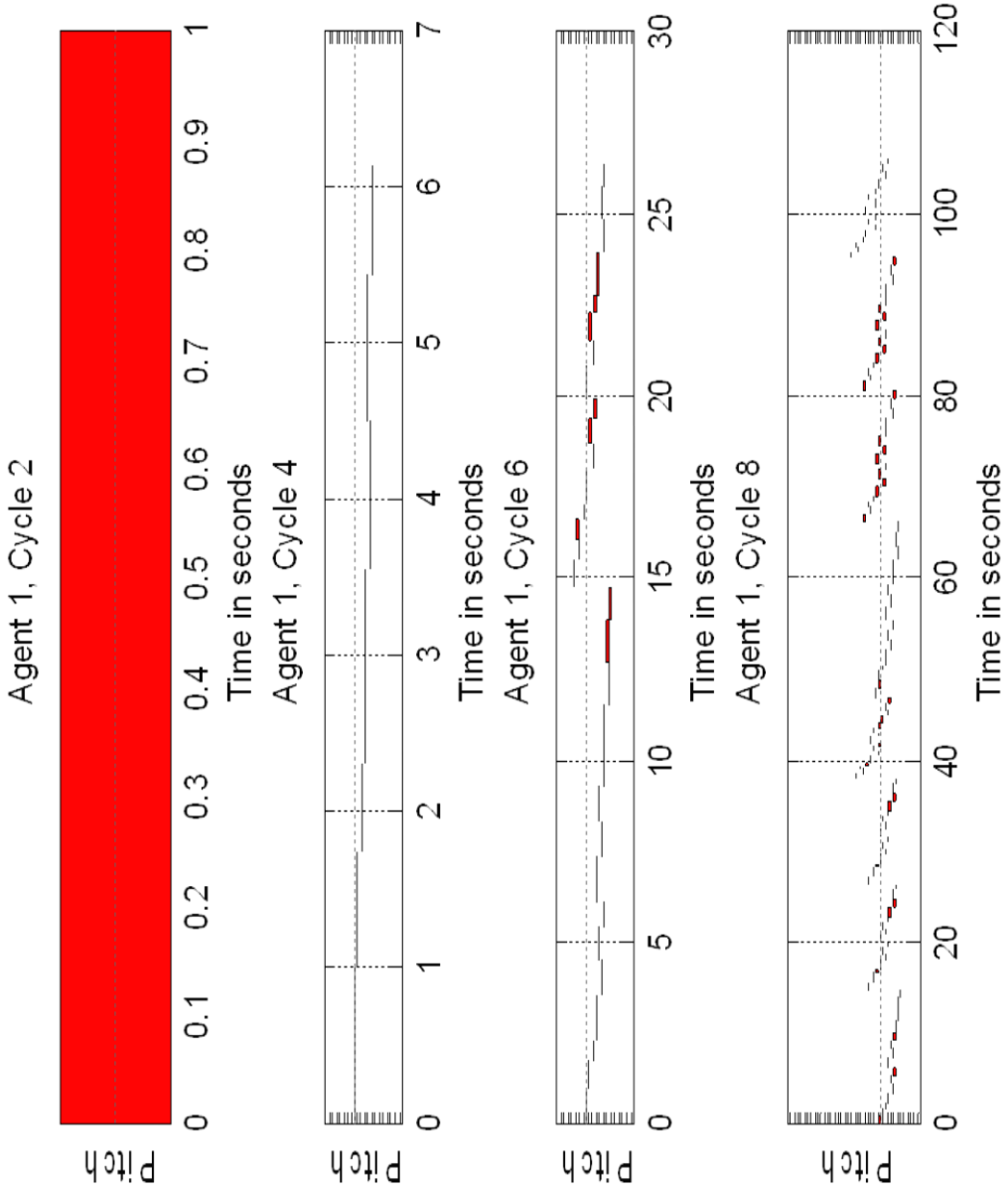

Figure 62A: Tunes for 2 happy, 2 sad, 2 angry, 2 tender – Agent 1 (8 Agents)

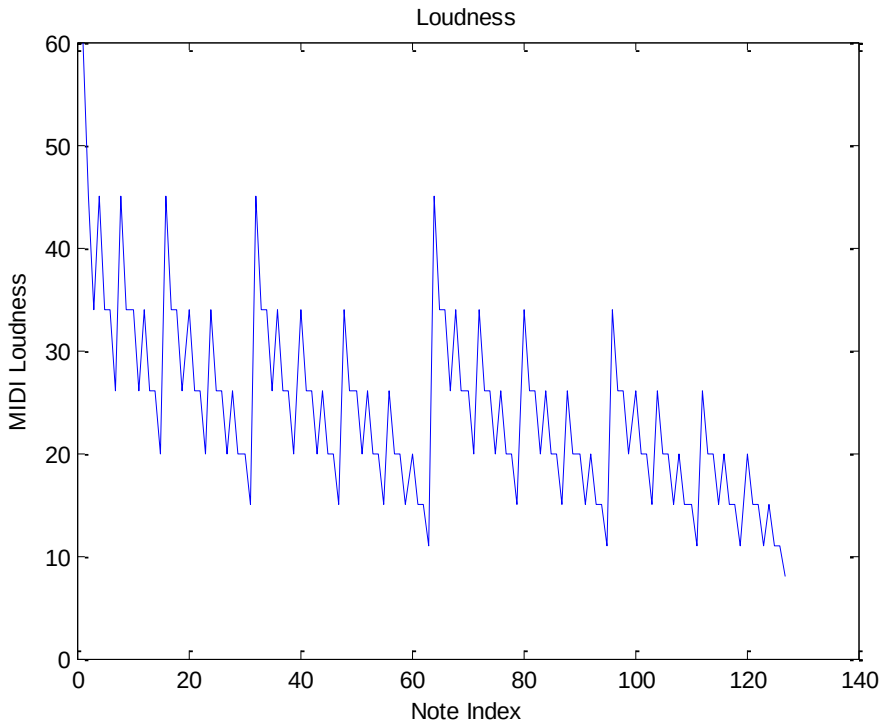

Figure 63A: Loudness for all "Sad" – Agent 1 (8 Agents)

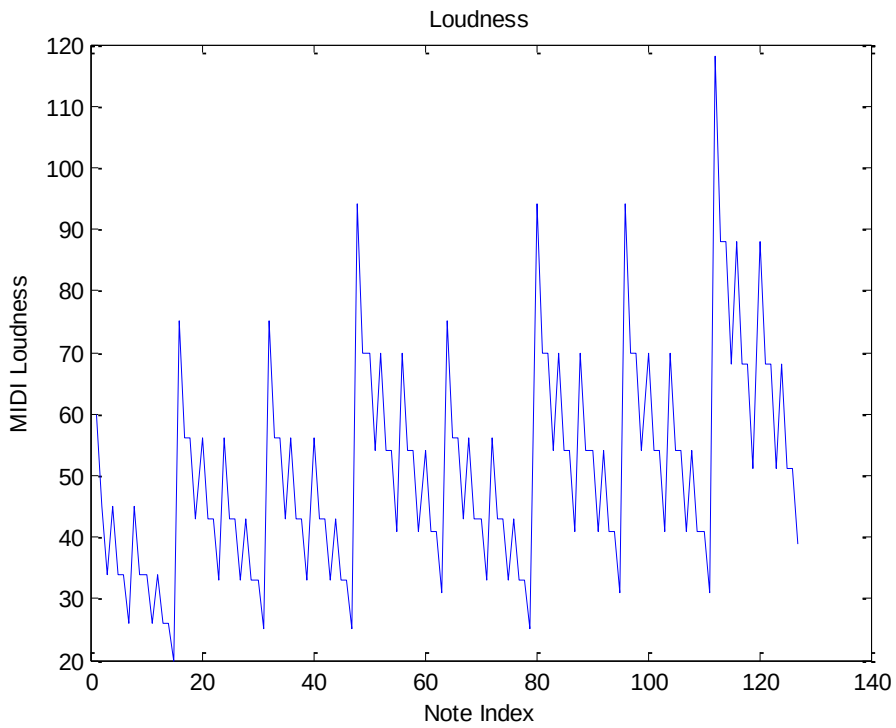

Figure 64A: Loudness for half "Sad", half "Happy" – Agent 1 (8 Agents)

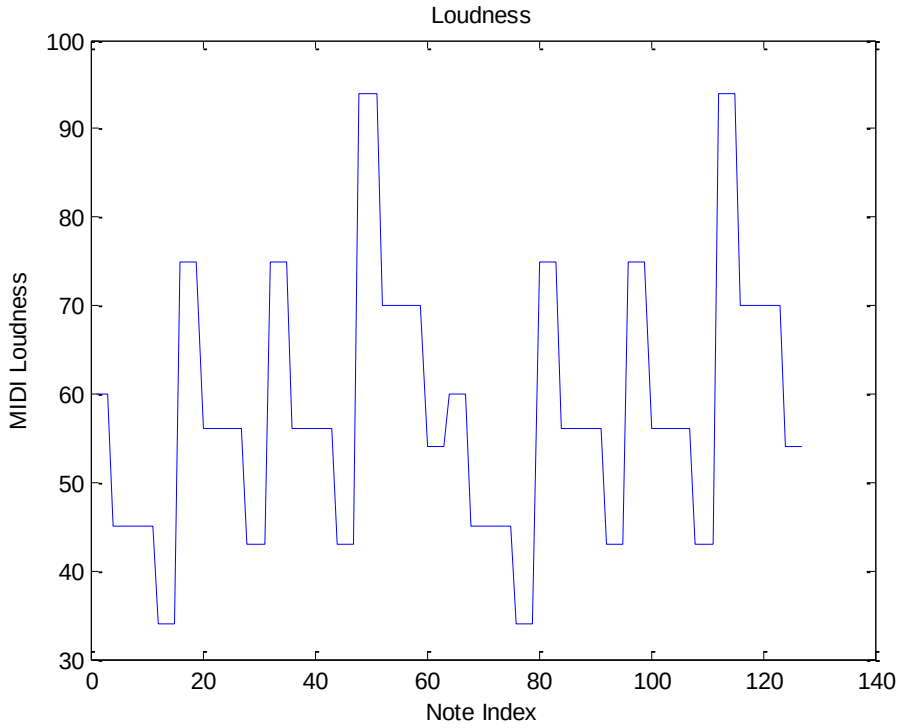

Figure 65A: Loudness for 2 happy, 2 sad, 2 angry, 2 tender – Agent 1 (8 Agents)

Experiment Set 3 –Affective Similarity Threshold

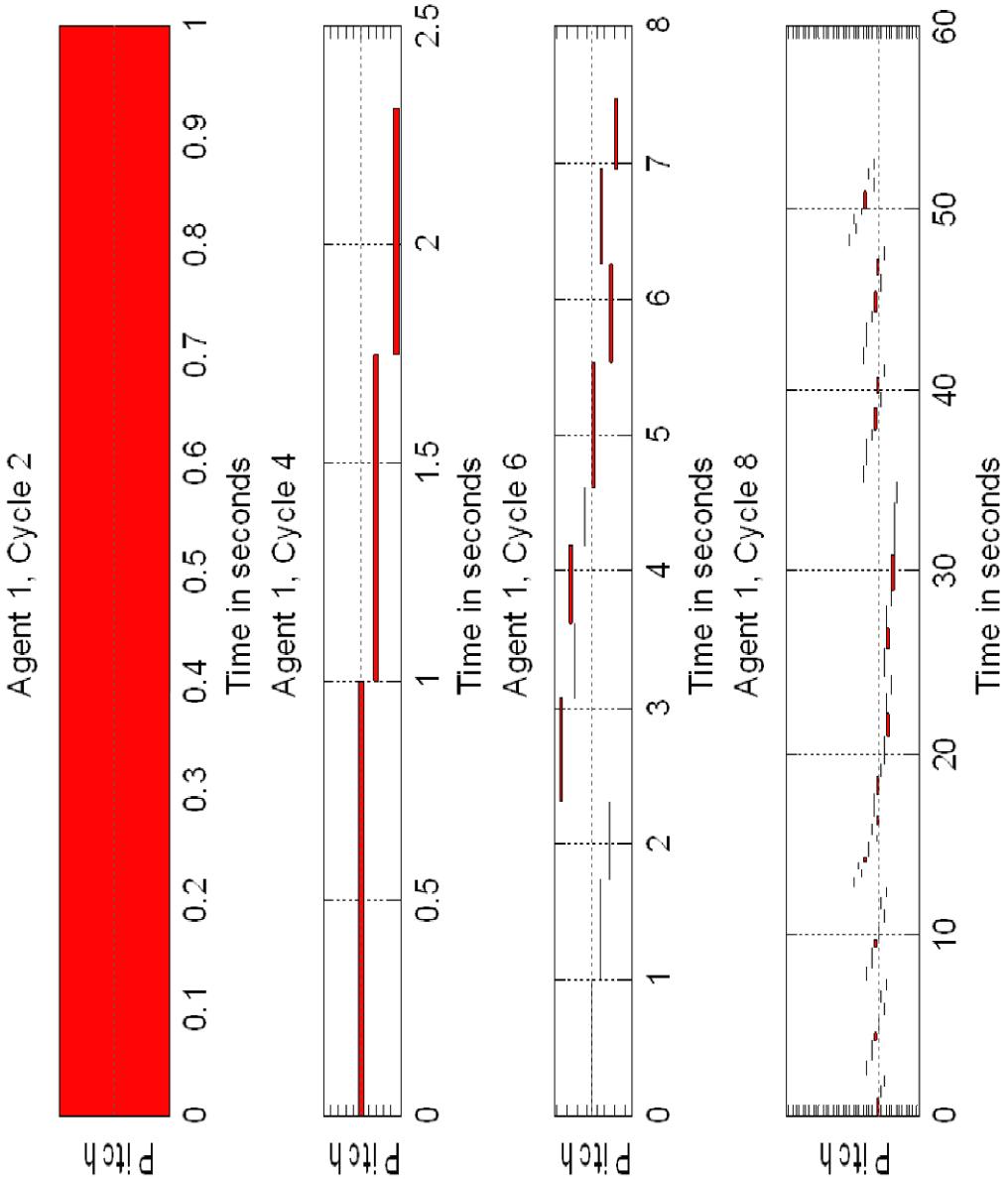

Figure 66A: Eight Agents, Equal Spread of Initial Affective States, Affective Similarity Threshold 1

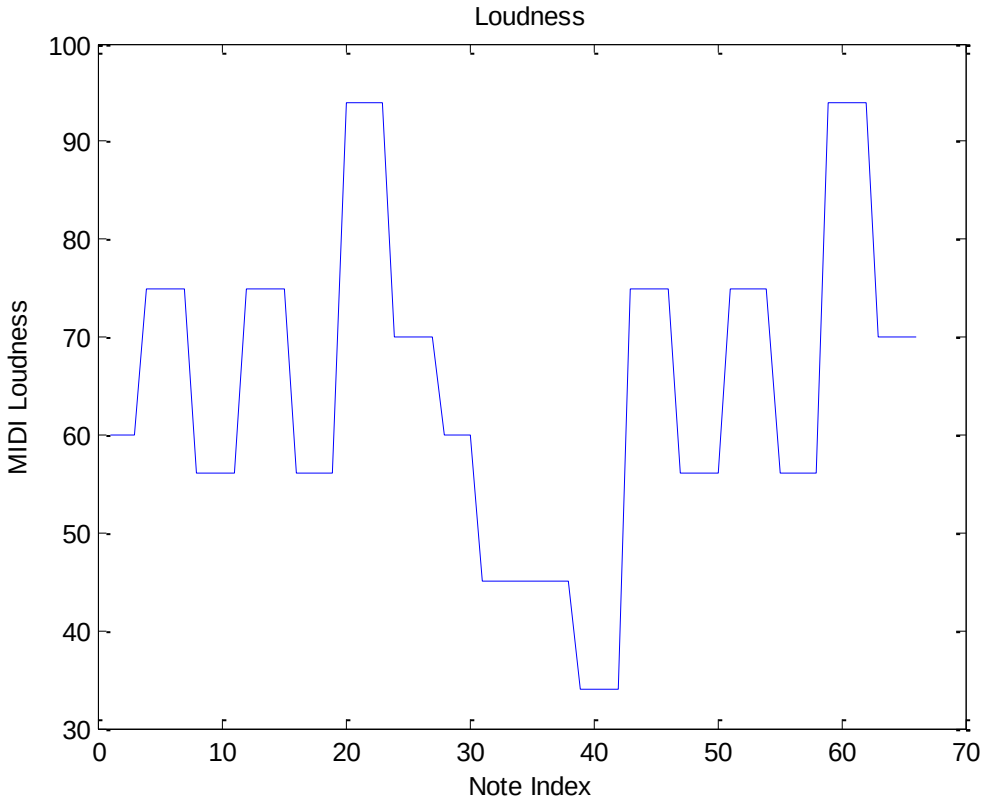

Figure 67A: Loudness for 8 Agents, Equal Spread of Initial Affective States, Affective Similarity Threshold 1

Experiment Set 4 –Inter-Agent Affective Update Rate

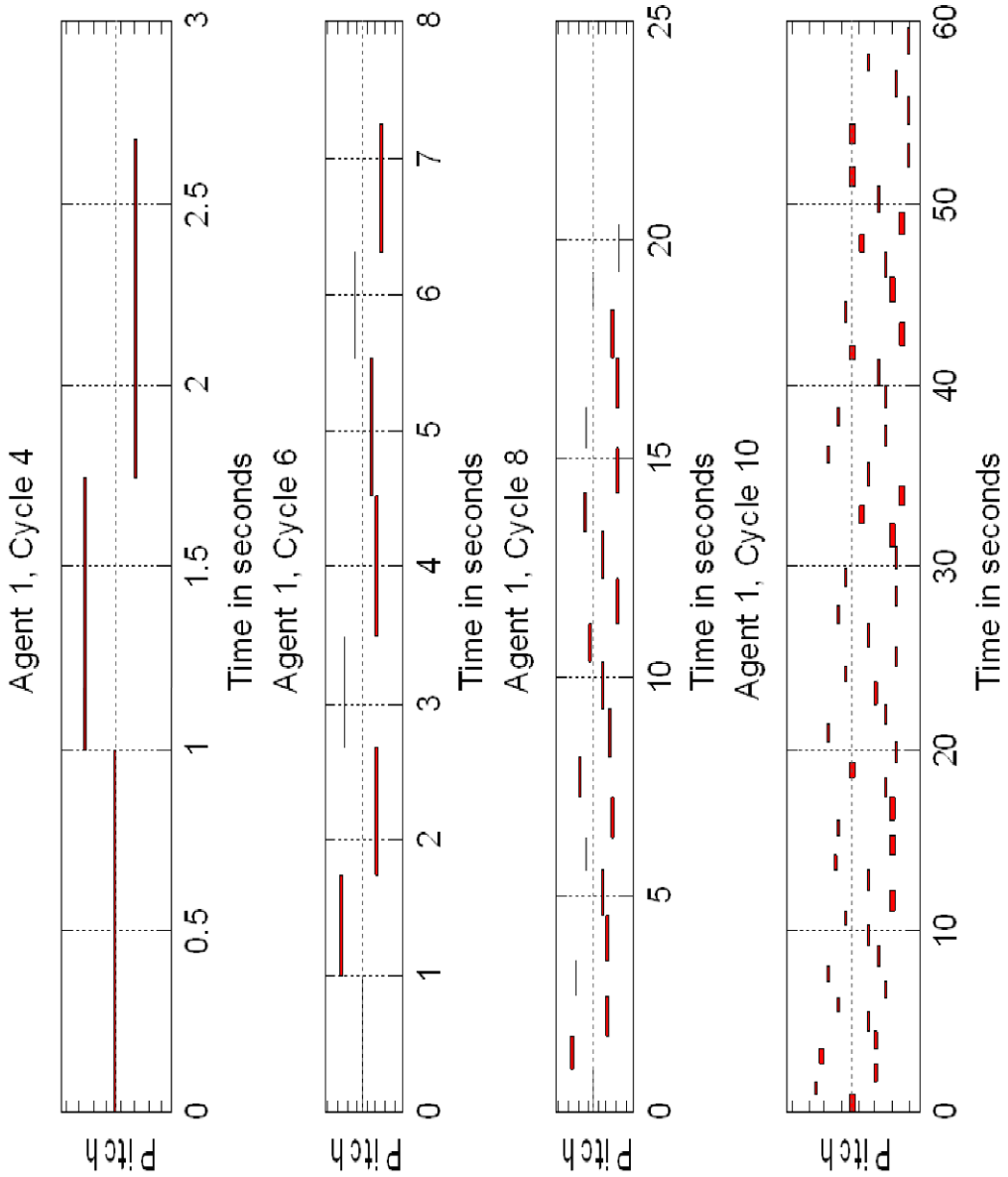

Figure 68A: Tunes for "Sad / Happy" – Agent 1 – Affective Updates 0.1

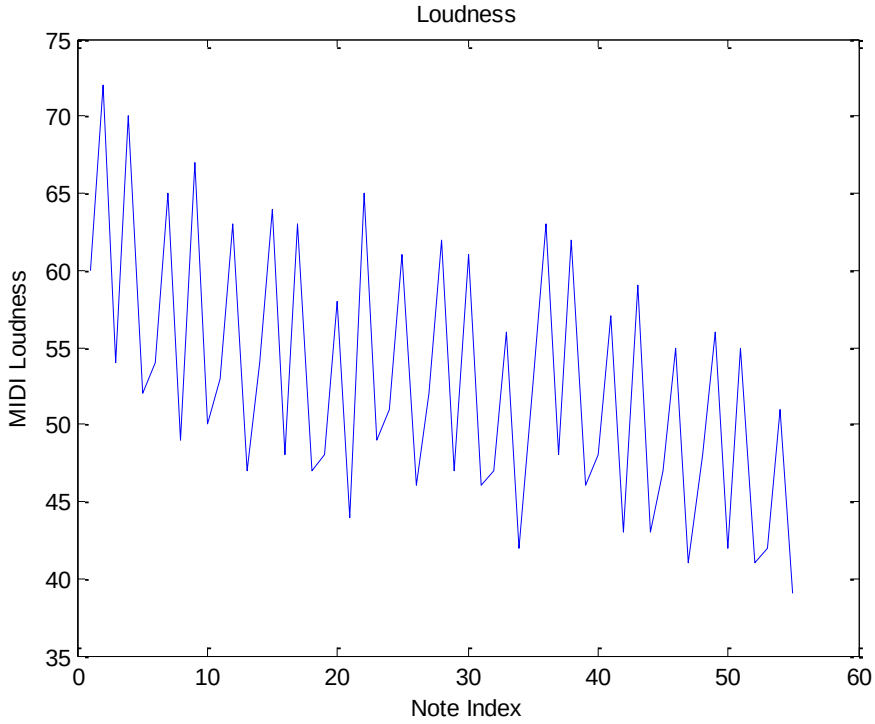

Figure 69A: Loudness for "Sad / Happy" – Agent 1 – Affective Updates 0.1

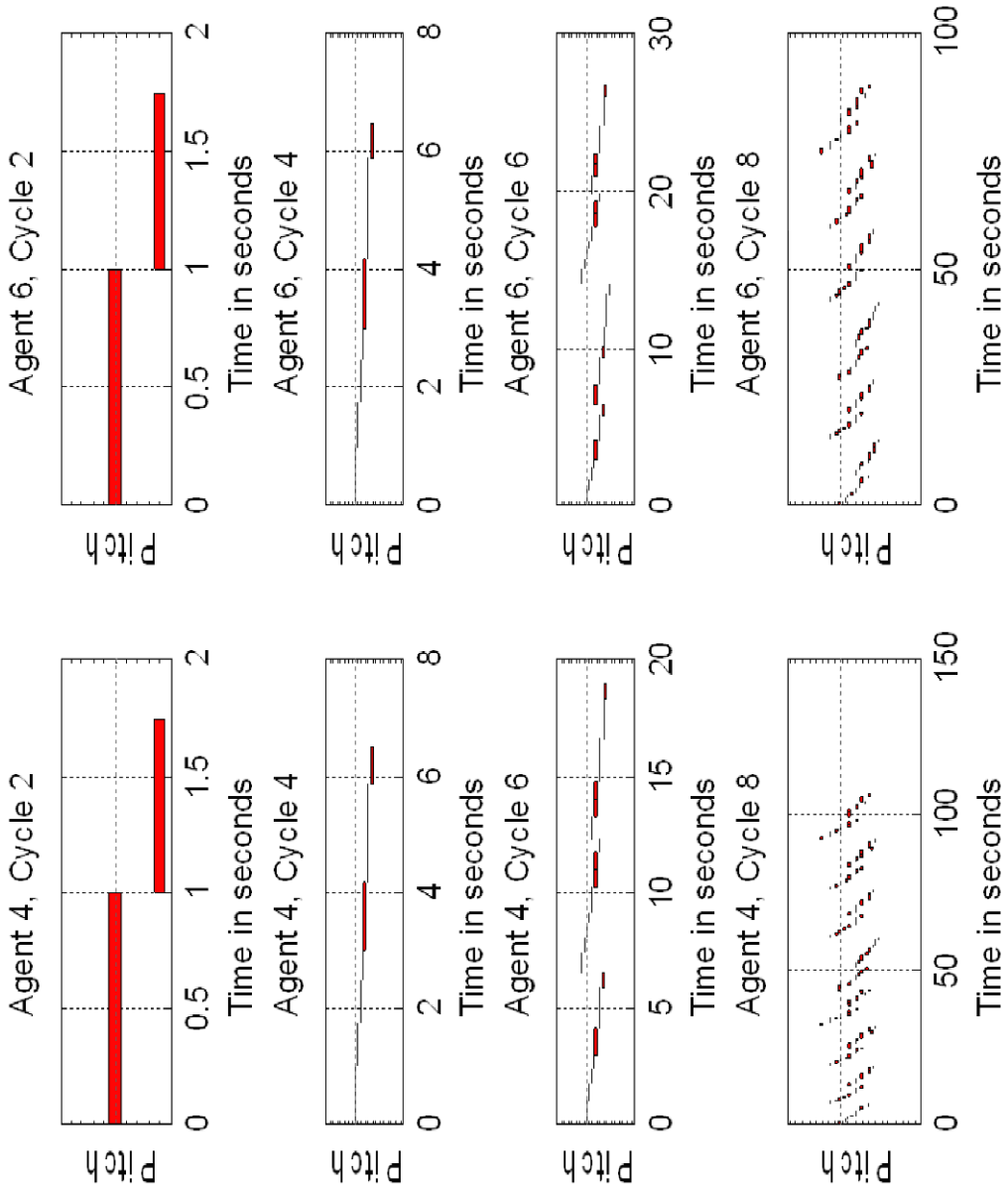

Figure 70A: Agents 4 (“Sad”) and 6 (“Happy”) Tunes

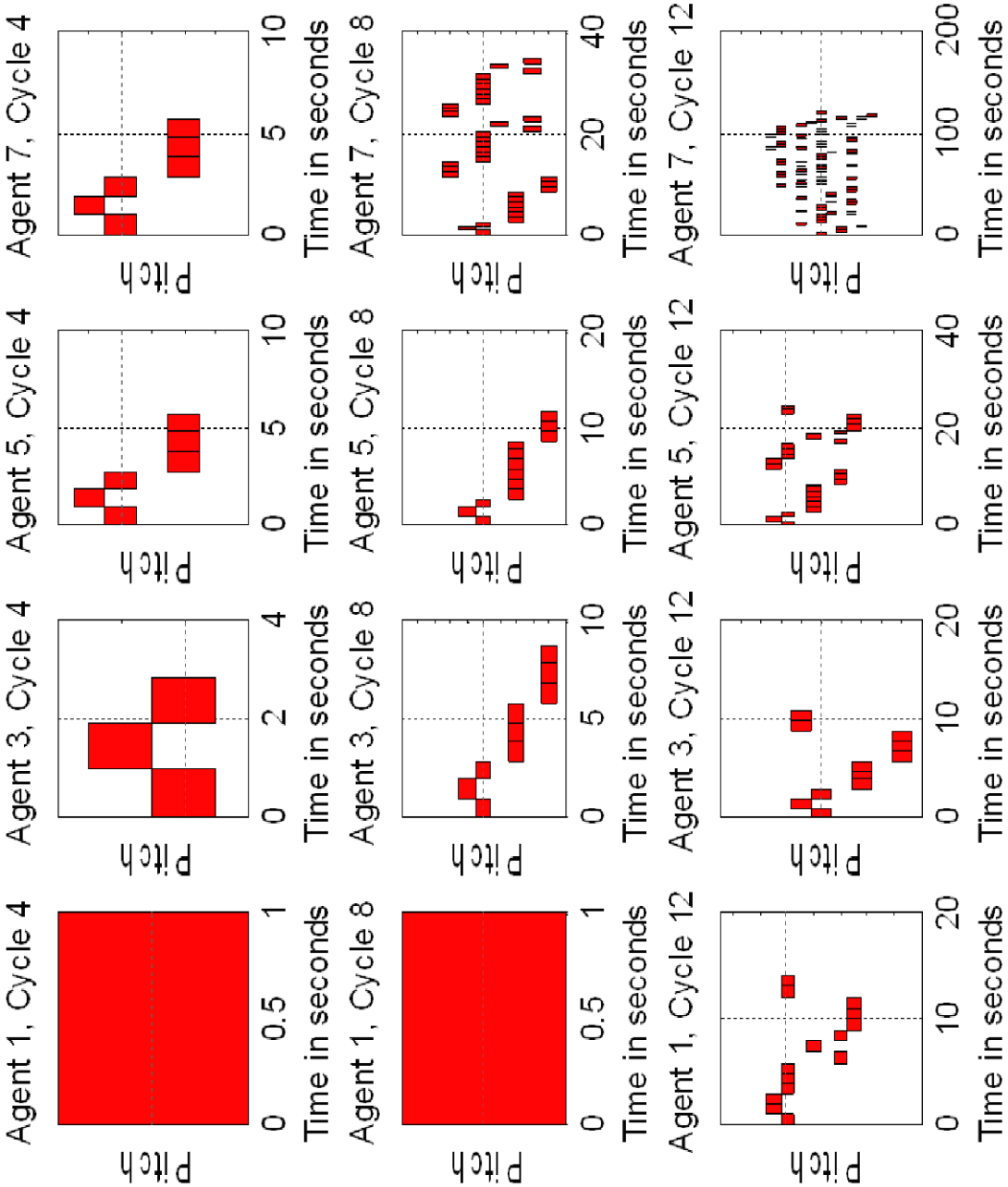

Figure 71A: Four Agents from an 8 Agent population using "musically" selected parameters, 12 cycles

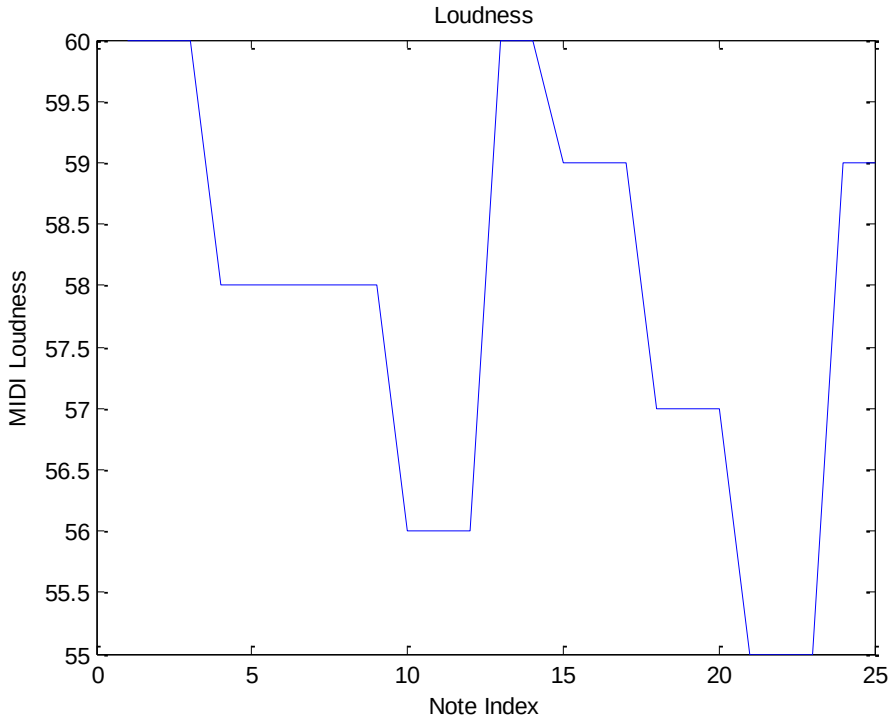

Figure 72A: Loudness for Agent 5 after 12 cycles

Experiment Set 5 – Effects of Interaction Coefficient

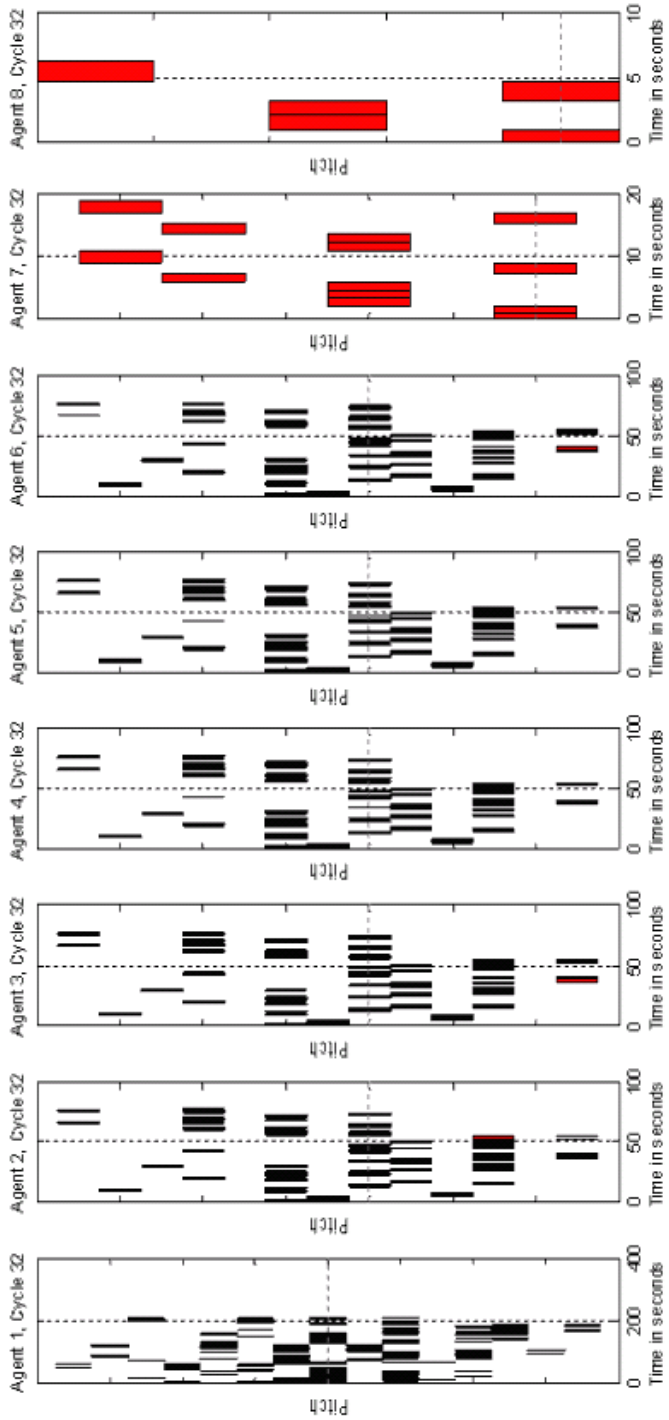

Figure 73A: Final tunes for row 10 of Figure 36A – i.e. Interaction Coefficient Threshold 0.9

Experiment Set 6 – Effects of Emotional Expressive Performance

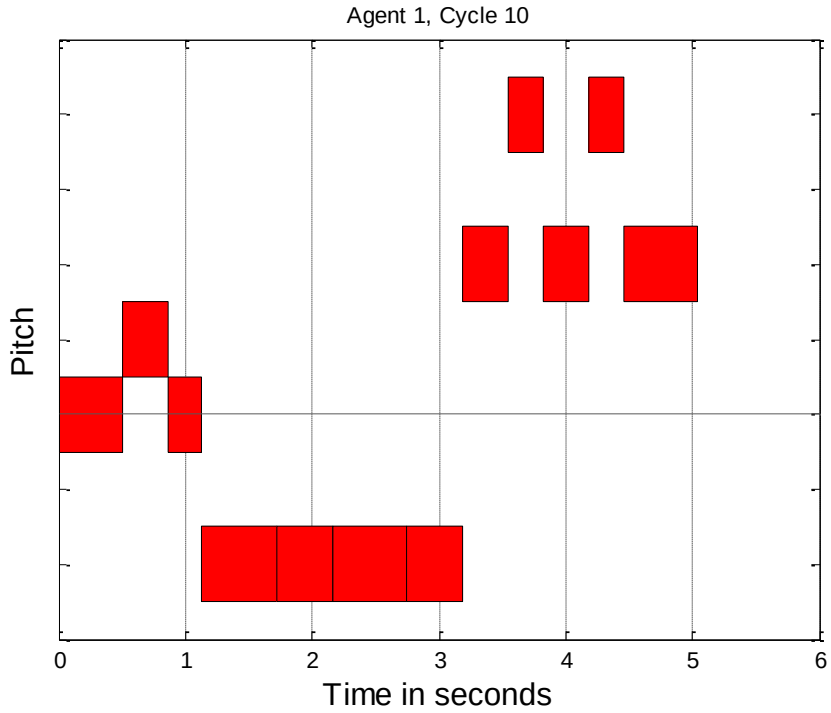

Figure 74A: Agent 1's tune with k = 0, 24 cycles

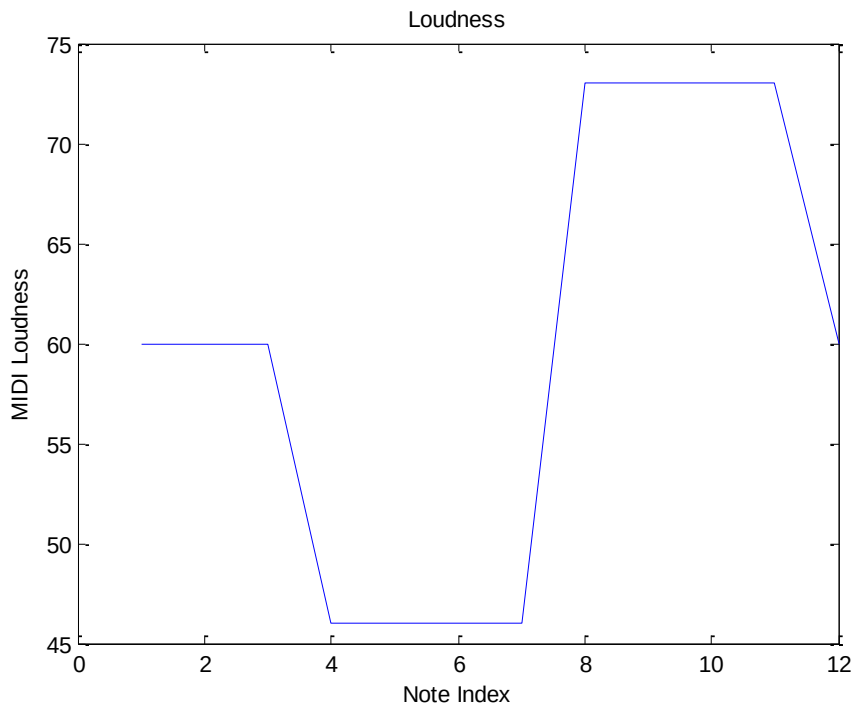

Figure 75A: Agent 1's loudness with k = 0, 24 cycles

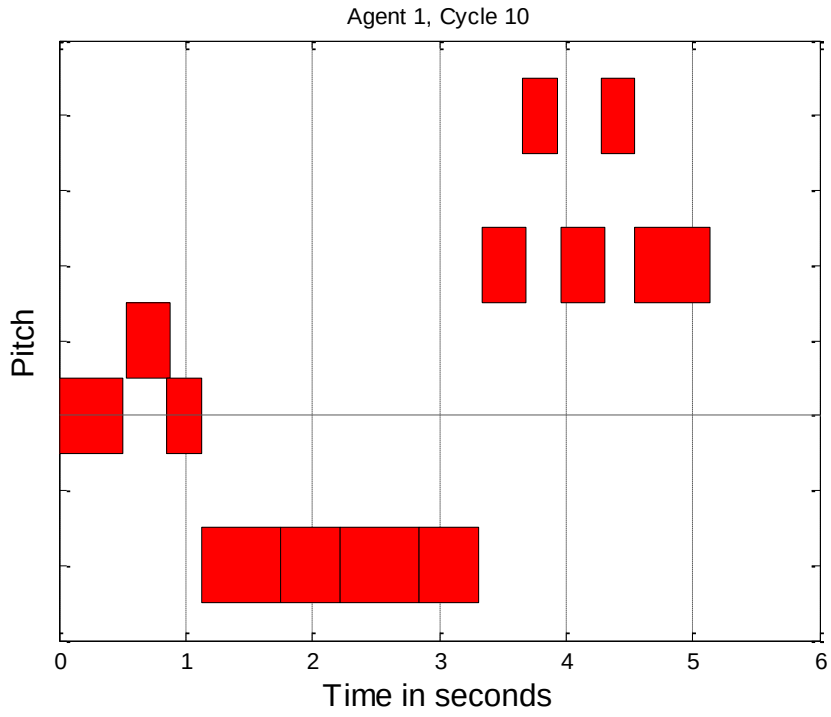

Figure 76A: Agent 1's tune with k = 0.2, 24 cycles

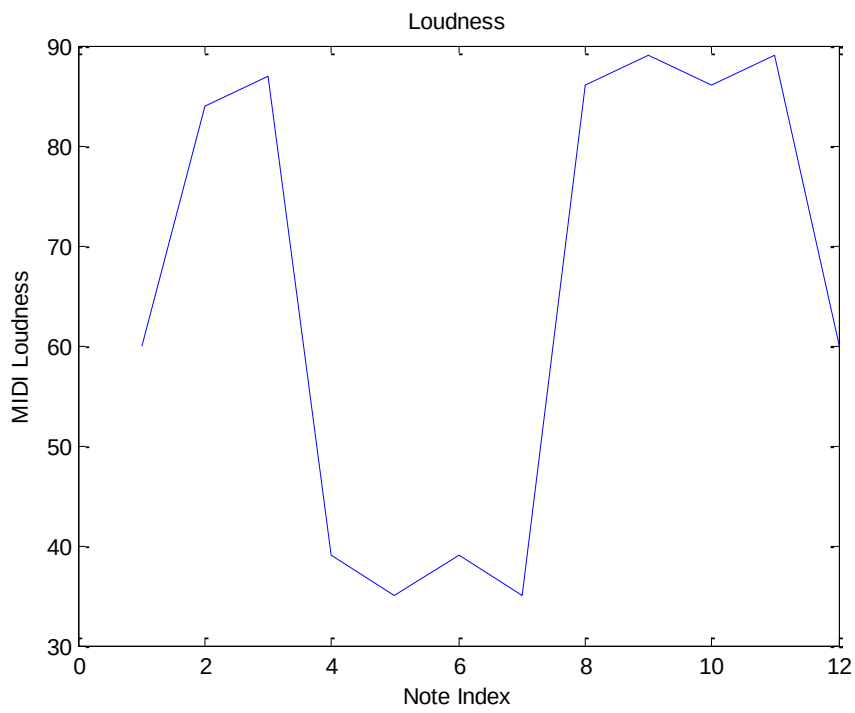

Figure 77A: Agent 1's loudness with k = 0.2, 24 cycles

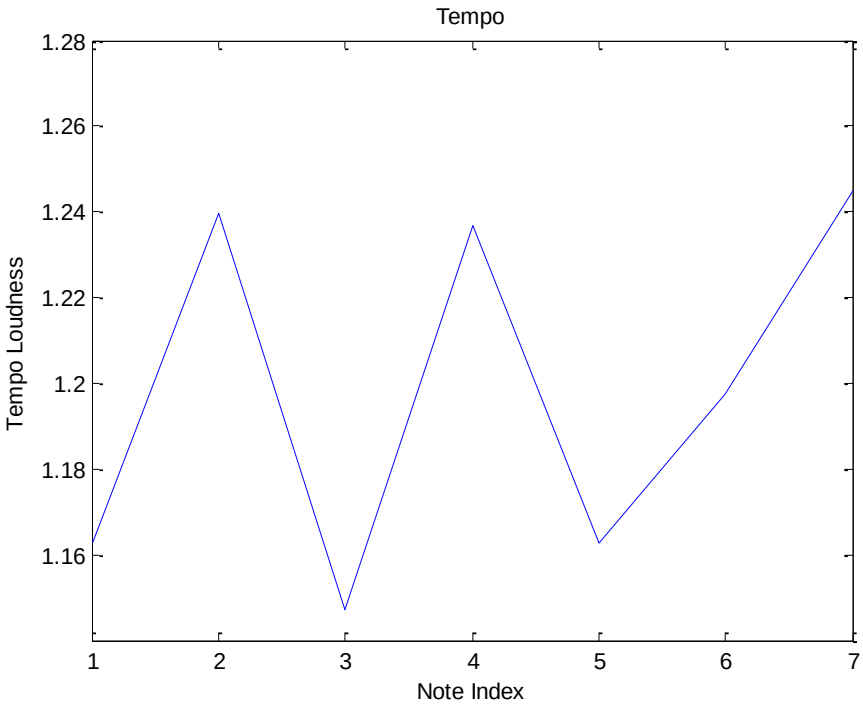

Figure 78A: Agent 1 - 2 Neutral Agents after 5 and 6 cycles, k = 0.2, Tempo

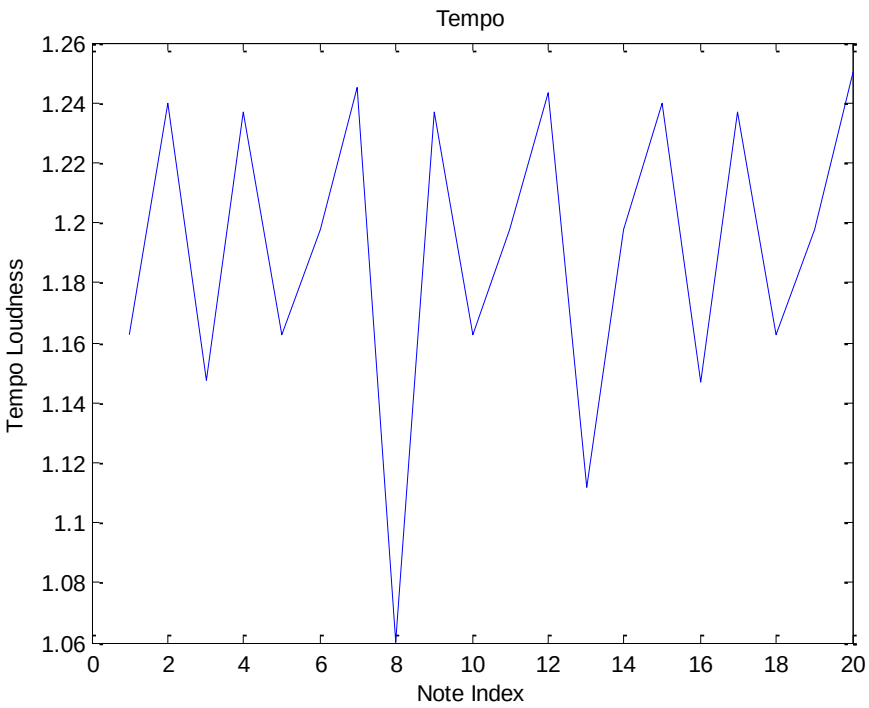

Figure 79A: Agent 1 - 2 Neutral Agents after 7 and 8 cycles, k = 0.2, Tempo

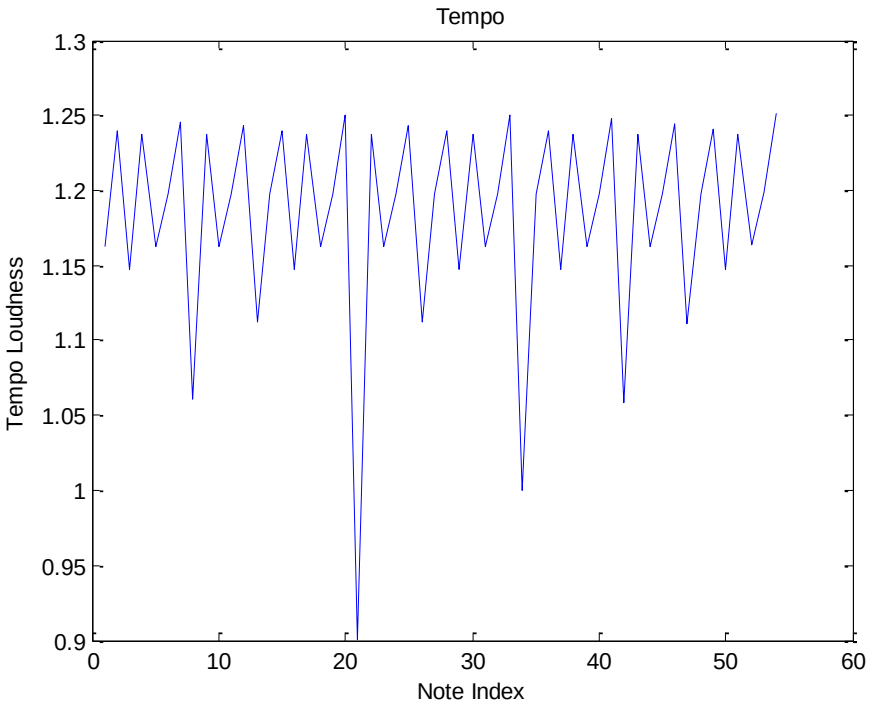

Figure 80A: Agent 1 - 2 Neutral Agents after 9 and 10 cycles, k = 0.2, Tempo

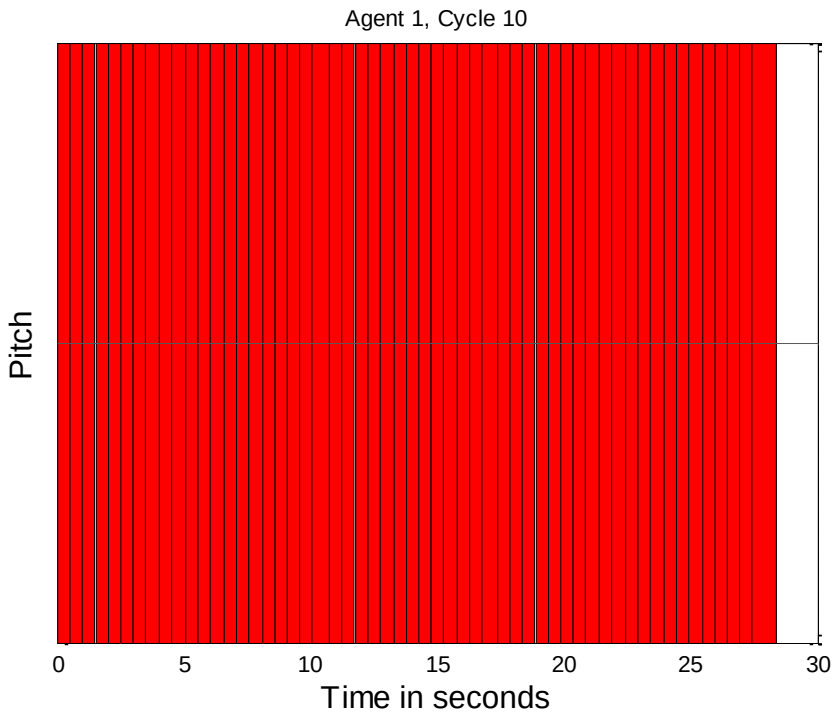

Figure 81A: Agent 1 - 2 Neutral Agents after 10 cycles, k = 0.2, Tune

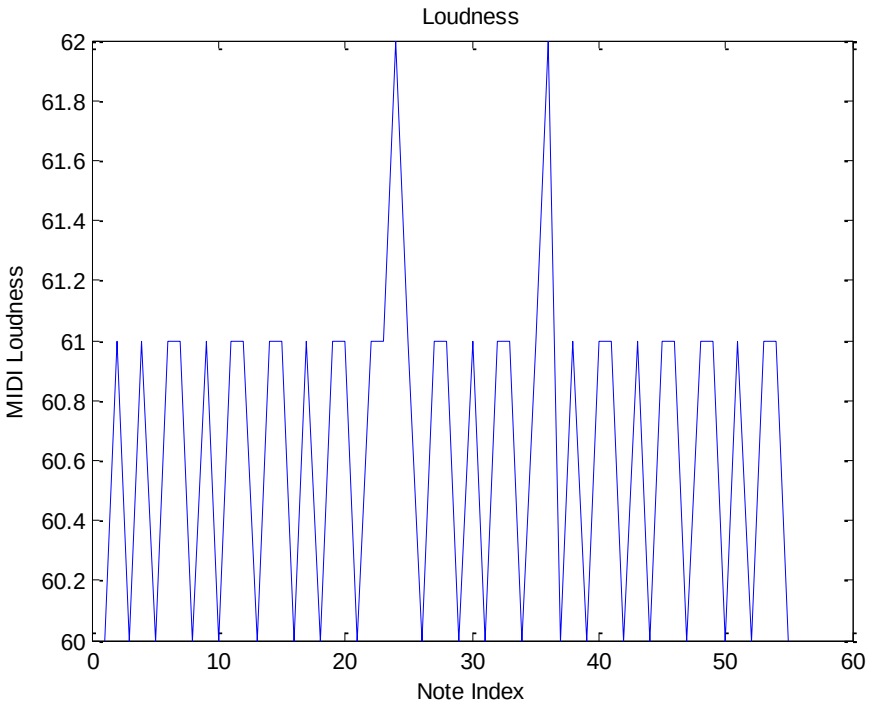

Figure 82A: Agent 1 - 2 Neutral Agents after 10 cycles, k = 0.2, Loudness

Appendix 3 – Results Tables

This Appendix contains the tables of data generated in the experiments in Chapter 4.

Experiment Set 1 – Number of Agents, Cycles, and Maximum Note Count

Number of Cycles	Mean Note Count	Note Count Standard Dev
1	2	1
2	3	1
3	4	1
4	7	2
5	11	4
6	17	6
7	28	9
8	45	15
9	72	24
10	117	39
11	189	63
12	305	102
13	494	165
14	799	267
15	1292	431
16	2091	698
17	3383	1129
18	5473	1827
19	8856	2956
20	14329	4784

Table 19A: 2 Agents, 20 Cycles

Number of Cycles	Mean Note Count	Note Count Standard Dev
1	2	1
2	3	1
3	6	2
4	12	3
5	24	6
6	45	12
7	87	23
8	168	44
9	324	85
10	625	164
11	1204	316
12	2322	610
13	4476	1175
14	8628	2265
15	16631	4367

Table 20A: 4 Agents, 15 Cycles

Number of Cycles	Mean Note Count	Note Count Standard Dev
1	2	0
2	4	1
3	7	1
4	14	3
5	28	6
6	56	11
7	112	22
8	224	44
9	447	89
10	893	177
11	1782	353
12	3556	704
13	7098	1405
14	14168	2805

Table 21A: 8 Agents, 14 Cycles

Number of Cycles	Mean Note Count	Note Cout Standard Dev
1	2	1
2	3	1
3	4	1
4	7	2
5	11	4
6	17	6
7	28	9
8	45	15
9	72	24
10	117	39
11	189	63
12	189	63
13	189	63

Table 22A: 2 Agents, 13 Cycles, Max Note Count 300

Number of Cycles	Mean Note Count	Note Count Standard Dev
1	2	1
2	3	1
3	6	2
4	12	3
5	24	6
6	45	12
7	87	23
8	168	44
9	168	44
10	168	44
11	209	46
12	209	46
13	209	46

Table 23A: 4 Agents, 14 Cycles, Max Note Count 300

Number of Cycles	Mean Note Count	Note Count Standard Dev
1	2	0
2	4	1
3	7	1
4	14	3
5	28	6
6	56	11
7	112	22
8	224	44
9	224	44
10	224	44

Table 24A: 8 Agents, 10 Cycles, Max Note Count 300

Experiment Set 2 – Initial Affective State

Affective State Agent 1	Affective State Agent 2	Median Pitch	Pitch Spread	Loud	Keymode	IOI	Est Valence	Est Arousal	Approx Label
[-0.5, -0.5] "Sad"	[-0.5, -0.5] "Sad"	20	56	20	2	2.8	0.7	-2.0	"Tender"
[-0.5, 0.5] "Anger"	[-0.5, -0.5] "Sad"	29	48	34	2	0.9	-0.4	-0.4	"Sad"
[-0.5, 0.5] "Anger"	[-0.5, 0.5] "Anger"	42	32	60	2	0.3	-0.3	0.3	"Angry"
[-0.5, 0.5] "Anger"	[0.5, -0.5] "Tender"	58	5	60	1	0.9	0.2	-0.3	"Tender"
[-0.5, -0.5] "Sadness"	[0.5, 0.5] "Happy"	50	28	52	1.5	0.9	0.0	-0.5	"Sad"
[0.5, -0.5] "Tenderness"	[0.5, -0.5] "Tenderness"	82	65	60	1	2.8	1.7	-1.7	"Tender"
[0.5, 0.5] "Happy"	[0.5, 0.5] "Happy"	127	67	119	1	0.3	1.1	0.9	"Happy"

Table 25A: Effects of Initial Affective State – 2 agent population, 10 cycles

Agents Profile	Median Pitch	Pitch Spread	Loud	Keymode	IOI	Est Valence	Est Arousal	Approx Label
SSSSSSSS	20	55	21	2	2.53	0.12	-0.42	Tender
AAAASSSS	30	47	36	2	0.93	-0.07	-0.10	Sad
AAAAAAAA	44	30	60	2	0.35	-0.06	0.06	Angry
AAAATTTT	60	39	60	1	0.93	0.08	-0.08	Tender
SSSSHHHH	56	106	57	1	0.95	0.08	-0.16	Tender
TTTTTTTT	82	47	60	1	2.53	0.34	-0.34	Tender
HHHHHHHH	127	67	119	1	0.35	0.26	0.22	Happy
SSAAHHTT	56	82	58	1	0.93	0.07	-0.09	Tender
SSAAHHHH	68	96	73	1	0.57	0.07	-0.02	Tender
SSHHHHHH	91	92	89	1	0.57	0.15	0.04	Happy
SHHHHHHH	116	80	106	1	0.45	0.21	0.13	Happy

Table 26A: Effects of Initial Affective State – 8 agent population

Experiment Set 3 –Affective Similarity Threshold

Similarity Threshold	Note Count	Note Count Std	Median Pitch	Pitch Spread	Loudess	Keymode	IOI	Est Valence	Est Arousal	Approx Label
0	1	0	60	0	60	2	0.00	-0.13	0.13	Angry
0.1	2	1	61	1	60	1.8	0.28	-0.13	0.13	Angry
0.2	2	1	61	1	60	1.8	0.28	-0.13	0.13	Angry
0.3	2	1	61	1	60	1.8	0.28	-0.13	0.13	Angry
0.4	9	9	58	26	62	1.5	0.86	0.09	-0.09	Tender
0.5	9	9	58	26	62	1.5	0.86	0.09	-0.09	Tender
0.6	9	9	58	26	62	1.5	0.86	0.09	-0.09	Tender
0.7	18	22	57	37	61	1.5	0.88	0.09	-0.08	Tender
0.8	30	19	60	62	63	1	0.89	0.09	-0.08	Tender
0.9	67	35	63	71	65	1	0.90	0.09	-0.08	Tender
1	113	37	61	78	62	1	0.90	0.08	-0.08	Tender
1.1	222	43	57	81	58	1	0.93	0.07	-0.09	Tender
1.2	224	44	56	82	58	1	0.93	0.07	-0.09	Tender
1.3	224	44	56	82	58	1	0.93	0.07	-0.09	Tender
1.4	224	44	56	82	58	1	0.93	0.07	-0.09	Tender

Table 27A: Effects of Similarity Threshold

Experiment Set 4 –Inter-Agent Affective Update Rate

Affective Update Valence	Note Count	Median Pitch	Loud	Mode	Est Tune Valence	Est Tune Arousal	Approx Tune Label	Agent Valence	Agent Arousal	Agent Approx Label
0	1	60	60	2	-0.52	0.51	Angry	-0.25	0.25	Angry
0.02	1	60	60	2	-0.52	0.51	Angry	-0.32	0.25	Angry
0.04	1	60	60	2	-0.52	0.51	Angry	-0.38	0.25	Angry
0.06	1	60	60	2	-0.52	0.51	Angry	-0.43	0.25	Angry
0.08	1	60	60	2	-0.52	0.51	Angry	-0.49	0.25	Angry
0.1	1	60	60	2	-0.52	0.51	Angry	-0.54	0.25	Angry
0.12	1	60	60	2	-0.52	0.51	Angry	-0.62	0.25	Angry
0.14	1	60	60	2	-0.52	0.51	Angry	-0.65	0.25	Angry
0.16	1	60	60	2	-0.52	0.51	Angry	-0.77	0.25	Angry
0.18	1	60	60	2	-0.52	0.51	Angry	-0.69	0.25	Angry
0.2	1	60	60	2	-0.52	0.51	Angry	-0.81	0.25	Angry

Table 28A: Two Agents, 1 Angry 1 Neutral, Affective Threshold $1x10^{10}$, Aff Update Arousal = 0

Affective Update Arousal	Note Count	Median Pitch	Loud	Mode	Est Tune Valence	Est Tune Arousal	Approx Tune Label	Agent Valence	Agent Arousal	Approx Agent Label
0	1	60	60	2	-0.52	0.51	Angry	-0.25	0.25	Angry
0.02	1	60	60	2	-0.52	0.51	Angry	-0.25	0.31	Angry
0.04	1	60	60	2	-0.52	0.51	Angry	-0.25	0.37	Angry
0.06	1	60	60	2	-0.52	0.51	Angry	-0.25	0.41	Angry
0.08	1	60	60	2	-0.52	0.51	Angry	-0.25	0.46	Angry
0.1	1	60	60	2	-0.52	0.51	Angry	-0.25	0.51	Angry
0.12	1	60	60	2	-0.52	0.51	Angry	-0.25	0.55	Angry
0.14	1	60	60	2	-0.52	0.51	Angry	-0.25	0.60	Angry
0.16	1	60	60	2	-0.52	0.51	Angry	-0.25	0.64	Angry
0.18	1	60	60	2	-0.52	0.51	Angry	-0.25	0.69	Angry
0.2	1	60	60	2	-0.52	0.51	Angry	-0.25	0.73	Angry

Table 29A: Two Agents, 1 Angry 1 Neutral, Affective Threshold $1x10^{10}$, Aff Update Valence = 0

Affective Update Valence	Note Count	Est Tune Valence	Est Tune Arousal	Approx Tune Label	Agent Valence	Agent Arousal	Agent Approx Label
0	1	-0.52	0.51	Angry	-0.50	0.00	Sad
0.02	1	-0.52	0.51	Angry	-0.57	0.00	Sad
0.04	1	-0.52	0.51	Angry	-0.63	0.00	Sad
0.06	1	-0.52	0.51	Angry	-0.69	0.00	Sad
0.08	1	-0.52	0.51	Angry	-0.75	0.00	Sad
0.1	1	-0.52	0.51	Angry	-0.81	0.00	Sad
0.12	1	-0.52	0.51	Angry	-0.87	0.00	Sad
0.14	1	-0.52	0.51	Angry	-0.90	0.00	Sad
0.16	1	-0.52	0.51	Angry	-0.95	0.00	Sad
0.18	1	-0.52	0.51	Angry	-1.00	0.00	Sad
0.2	1	-0.52	0.51	Angry	-1.05	0.00	Sad

Table 30A: Two Agents, 1 Angry 1 Sad, Affective Threshold $1x10^{10}$, Aff Update Arousal = 0

Affective Update Arousal	Est Tune Valence	Est Tune Arousal	Approx Tune Label	Agent Valence	Agent Arousal	Agent Approx Label
0	-0.52	0.51	Angry	-0.50	0.00	Sad
0.02	-0.52	0.51	Angry	-0.50	0.08	Angry
0.04	-0.52	0.51	Angry	-0.50	0.15	Angry
0.06	-0.52	0.51	Angry	-0.50	0.21	Angry
0.08	-0.52	0.51	Angry	-0.50	0.28	Angry
0.1	-0.52	0.51	Angry	-0.50	0.35	Angry
0.12	-0.52	0.51	Angry	-0.50	0.39	Angry
0.14	-0.52	0.51	Angry	-0.50	0.42	Angry
0.16	-0.52	0.51	Angry	-0.50	0.47	Angry
0.18	-0.52	0.51	Angry	-0.50	0.52	Angry
0.2	-0.52	0.51	Angry	-0.50	0.60	Angry

Table 31A: Two Agents, 1 Angry 1 Sad, Affective Threshold $1x10^{10}$, Aff Update Valence = 0

Affective Update Valence	Est Tune Valence	Est Tune Arousal	Approx Tune Label	Agent Valence	Agent Arousal	Agent Approx Label
0	-0.13	0.13	Angry	0.00	0.00	Neutral
0.02	-0.13	0.13	Angry	-0.13	0.00	Sad
0.04	-0.13	0.13	Angry	-0.21	0.00	Sad
0.06	-0.13	0.13	Angry	-0.29	0.00	Sad
0.08	-0.13	0.13	Angry	-0.39	0.00	Sad
0.1	-0.13	0.13	Angry	-0.62	0.00	Sad
0.12	-0.13	0.13	Angry	-0.65	0.00	Sad
0.14	-0.13	0.13	Angry	-0.79	0.00	Sad
0.16	-0.13	0.13	Angry	-0.83	0.00	Sad
0.18	-0.13	0.13	Angry	-0.86	0.00	Sad
0.2	-0.13	0.13	Angry	-0.87	0.00	Sad

Table 32A: Eight Agents, Equal Spread, Affective Threshold $1x10^{10}$, Aff Update Arousal = 0

Affective Update Arousal	Est Tune Valence	Est Tune Arousal	Approx Tune Label	Agent Valence	Agent Arousal	Agent Approx Label
0	-0.13	0.13	Angry	0.00	0.00	Neutral
0.02	-0.13	0.13	Angry	0.00	0.11	Angry
0.04	-0.13	0.13	Angry	0.00	0.21	Angry
0.06	-0.13	0.13	Angry	0.00	0.31	Angry
0.08	-0.13	0.13	Angry	0.00	0.41	Angry
0.1	-0.13	0.13	Angry	0.00	0.50	Angry
0.12	-0.13	0.13	Angry	0.00	0.58	Angry
0.14	-0.13	0.13	Angry	0.00	0.58	Angry
0.16	-0.13	0.13	Angry	0.00	0.69	Angry
0.18	-0.13	0.13	Angry	0.00	0.80	Angry
0.2	-0.13	0.13	Angry	0.00	0.81	Angry

Table 33A: Eight Agents, Equal Spread, Affective Threshold $1x10^{10}$, Aff Update Valence = 0

Affective State Agent 1	Affective State Agent 2	Median Pitch	Pitch Spread	Loud	Keymode	IOI	Est Valence	Est Arousal	Approx Label
[-0.5, -0.5] "Sad"	[-0.5, -0.5] "Sad"	19	56	16	1	6.22	2.49	-3.94	Tender
[-0.5, 0.5] "Anger"	[-0.5, -0.5] "Sad"	27	50	31	2	1.25	-0.19	-0.72	Sad
[-0.5, 0.5] "Anger"	[-0.5, 0.5] "Anger"	44	29	60	2	0.36	-0.25	0.24	Angry
[-0.5, 0.5] "Anger"	[0.5, -0.5] "Tender"	63	14	60	1	1.10	0.45	-0.46	Tender
[-0.5, -0.5] "Sadness"	[0.5, 0.5] "Happy"	49	29	47	1.5	1.31	0.18	-0.71	Tender
[0.5, -0.5] "Tenderness"	[0.5, -0.5] "Tenderness"	102	67	60	1	4.85	2.71	-2.68	Tender
[0.5, 0.5] "Happy"	[0.5, 0.5] "Happy"	127	67	122	1	0.23	1.07	1.05	Happy

Table 34A: Effects of Initial Affective State – 2 agent population, affective updates 0.1

Similarity Threshold	Note Count	Note Count Std	Median Pitch	Pitch Spread	Loud	Mode	IOI	Est Tune Valence	Est Tune Arousal	Approx Tune Label	Agent Valence	Agent Arousal	Agent Approx Label
0	1	0	60	0	60	2	0.00	-0.13	0.13	Angry	-0.48	0.43	Angry
0.1	1	0	60	1	60	1.9	0.13	-0.13	0.13	Angry	-0.48	0.43	Angry
0.2	1	0	60	2	60	1.8	0.25	-0.13	0.13	Angry	-0.42	0.40	Angry
0.3	2	1	61	5	62	1.8	0.45	-0.13	0.13	Angry	-0.42	0.40	Angry
0.4	11	13	59	19	63	1.6	0.85	0.10	-0.04	Tender	0.30	-0.22	Tender
0.5	17	19	58	24	61	1.5	0.88	0.09	-0.08	Tender	0.27	-0.28	Tender
0.6	45	23	63	42	63	1	0.91	0.09	-0.08	Tender	0.27	-0.28	Tender
0.7	58	23	58	51	58	1	0.96	0.07	-0.10	Tender	0.21	-0.29	Tender
0.8	106	37	49	46	51	1	1.04	0.06	-0.13	Tender	0.14	-0.43	Tender
0.9	221	44	48	43	51	1	1.01	0.05	-0.12	Tender	0.08	-0.36	Tender
1	221	44	48	43	51	1	1.01	0.05	-0.12	Tender	0.08	-0.36	Tender
1.1	224	44	48	43	51	1	1.01	0.05	-0.12	Tender	0.08	-0.36	Tender
1.2	224	44	48	43	51	1	1.01	0.05	-0.12	Tender	0.08	-0.36	Tender
1.3	224	44	48	43	51	1	1.01	0.05	-0.12	Tender	0.08	-0.36	Tender
1.4	224	44	48	43	51	1	1.01	0.05	-0.12	Tender	0.08	-0.36	Tender

Table 35A: Effects of Similarity Threshold on Affective Update

Experiment Set 5 – Effects of Interaction Coefficient

Interaction Coefficient Threshold	Note Count	Note Count Std	Median Pitch	Pitch Spread	Loud	Key mode	IOI	Est Tune Valence	Est Tune Arousal	Approx Tune Label	Equality
0	221	44	59	11	51	1	1.02	0.05	-0.12	Tender	0.017
0.1	206	53	59	11	51	1	1.00	0.05	-0.12	Tender	0.017
0.2	257	24	59	11	50	1	0.97	0.07	-0.17	Tender	0.013
0.3	212	38	58	12	49	1.5	0.91	-0.02	-0.10	Sad	0.011
0.4	208	64	57	11	46	1.8	0.97	-0.02	-0.14	Sad	0.010
0.5	233	42	56	11	43	2	1.07	-0.01	-0.14	Sad	0.010
0.6	197	42	58	13	48	2	0.92	-0.01	-0.08	Sad	0.008
0.7	124	90	58	13	51	2	0.87	-0.01	-0.06	Sad	0.006
0.8	173	64	62	15	60	1.3	0.80	0.01	-0.01	Tender	0.006
0.9	103	86	61	10	59	1.1	0.90	0.14	-0.14	Tender	0.004
1	103	86	61	10	59	1.1	0.90	0.14	-0.14	Tender	0.004

Table 36A: Effects of Interaction Coefficient Threshold: 8 agents, equal spread, 1 Similarity Threshold, 0.1 valence/arousal update; 0.2 Interaction Coefficient Update, 0.2 Interaction Coefficient, 32 Cycles

Similarity Threshold	Note Count	Note Count Std	Median Pitch	Pitch Spread	Loud	Mode	IOI	Est Tune Valence	Est Tune Arousal	Approx Tune Label	Equality
0	1	0	60	0	60	2	0.00	-0.13	0.13	Angry	0.00
0.1	1	0	60	1	60	1.9	0.13	-0.13	0.13	Angry	0.00
0.2	1	0	60	2	60	1.8	0.25	-0.13	0.13	Angry	0.00
0.3	2	1	61	5	62	1.8	0.45	-0.13	0.13	Angry	0.01
0.4	11	13	59	19	63	1.6	0.85	0.10	-0.04	Tender	0.01
0.5	17	19	58	24	61	1.5	0.88	0.09	-0.08	Tender	0.01
0.6	45	23	63	42	63	1	0.91	0.09	-0.08	Tender	0.02
0.7	58	23	58	51	58	1	0.96	0.07	-0.10	Tender	0.04
0.8	106	37	49	46	51	1	1.04	0.06	-0.13	Tender	0.04
0.9	221	44	48	43	51	1	1.01	0.05	-0.12	Tender	0.06
1	221	44	48	43	51	1	1.01	0.05	-0.12	Tender	0.06
1.1	224	44	48	43	51	1	1.01	0.05	-0.12	Tender	0.10
1.2	224	44	48	43	51	1	1.01	0.05	-0.12	Tender	0.10
1.3	224	44	48	43	51	1	1.01	0.05	-0.12	Tender	0.10
1.4	224	44	48	43	51	1	1.01	0.05	-0.12	Tender	0.10

Table 37A: Effects of Similarity Threshold for 8 agents, equal affective spread, 8 cycles

Experiment Set 6 – Effects of Emotional Expressive Performance

k Value	Note Count	Note Count Std	Median Pitch	Pitch Spread	Loudness	Keymode	IOI	Agent Valence	Agent Arousal	Agent Approx Label
0	13	8	61	5	60	1.8	0.40	-0.15	0.14	Angry
0.1	13	8	61	5	67	1.1	0.39	-0.08	0.15	Angry
0.2	12	6	61	5	73	1.1	0.40	-0.04	0.19	Angry
0.3	8	6	61	4	69	1.3	0.42	-0.03	0.22	Angry
0.4	5	4	61	3	60	1.3	0.44	0.00	0.25	Happy
0.5	5	4	61	3	60	1.3	0.44	0.02	0.26	Happy
0.6	5	4	61	3	61	1.3	0.44	0.03	0.27	Happy
0.7	5	4	61	3	61	1.3	0.44	0.03	0.27	Happy
0.8	5	4	61	3	61	1.3	0.44	0.03	0.27	Happy
0.9	5	4	61	3	61	1.3	0.44	0.03	0.27	Happy
1	5	4	61	3	61	1.3	0.44	0.03	0.27	Happy

Table 38A: Effects of k-Value Expressive Performance, 8 agents, Equal Spread, 10 cycles

k Value	Note Count	Note Count Std	Median Pitch	Pitch Spread	Loud	Key mode	IOI	Est Tune Valence	Est Tune Arousal	Approx Tune Label	Agent Valence	Agent Arousal	Agent Approx Label
0	104	85	61	11	59	1.8	0.39	0.05	-0.05	Tender	-0.16	0.16	Angry
0.1	74	73	61	8	74	1.5	0.40	0.02	-0.09	Tender	0.02	0.27	Happy
0.2	49	64	61	7	69	1.1	0.44	0.00	-0.12	Sad	0.03	0.23	Happy
0.3	84	53	60	9	59	1.6	0.44	-0.17	-0.10	Sad	-0.02	0.25	Angry
0.4	65	59	63	9	63	1.0	0.42	-0.07	-0.09	Sad	0.17	0.44	Happy
0.5	24	20	62	6	53	1.0	0.45	-0.09	-0.11	Sad	0.16	0.42	Happy
0.6	15	14	62	4	73	1.0	0.42	0.02	-0.02	Tender	0.43	0.71	Happy
0.7	15	14	62	4	74	1.0	0.42	0.02	-0.02	Tender	0.44	0.72	Happy
0.8	21	19	62	5	78	1.0	0.39	0.02	-0.02	Tender	0.48	0.75	Happy
0.9	21	19	62	5	78	1.0	0.38	0.02	-0.02	Tender	0.49	0.76	Happy
1	21	19	62	5	78	1.0	0.38	0.02	-0.02	Tender	0.52	0.75	Happy
1.1	21	19	62	5	78	1.0	0.38	0.02	-0.02	Tender	0.52	0.75	Happy
1.2	21	19	62	5	78	1.0	0.38	0.02	-0.02	Tender	0.52	0.76	Happy
1.3	21	19	62	5	78	1.0	0.38	0.02	-0.02	Tender	0.51	0.76	Happy
1.4	21	19	62	5	78	1.0	0.38	0.02	-0.02	Tender	0.51	0.76	Happy
1.5	21	19	62	5	78	1.0	0.38	0.02	-0.02	Tender	0.51	0.76	Happy
1.6	21	19	62	5	78	1.0	0.37	0.02	-0.02	Tender	0.51	0.77	Happy
1.7	21	19	62	5	78	1.0	0.37	0.02	-0.02	Tender	0.50	0.77	Happy
1.8	21	19	62	5	78	1.0	0.37	0.02	-0.02	Tender	0.50	0.77	Happy
1.9	21	19	62	5	78	1.0	0.37	0.02	-0.02	Tender	0.50	0.77	Happy
2	21	19	62	5	78	1.0	0.37	0.02	-0.02	Tender	0.49	0.78	Happy

Table 39A: Effects of k-Value Expressive Performance, 8 agents, Equal Spread, 24 cycles

Experiment Set 7 – Listening Experiments

Agents Profile	Note Count	Note Count Std	Median Pitch	Pitch Spread	Loud	Mode	IOI	Est Tune Valence	Est Tune Arousal	Approx Tune Label	Agent Valence	Agent Arousal	Agent Approx Label
AAAAAAAA	112	22	60	6	60	2	0.20	-0.09	0.09	Angry	-0.50	0.50	Angry
AAAAAASS	112	22	58	8	53	2	0.25	-0.12	0.05	Angry	-0.50	0.25	Angry
AAAAAAHH	112	22	61	8	67	1.1	0.20	-0.01	0.07	Angry	-0.25	0.50	Angry
AAAAAATT	111	22	61	6	60	1.1	0.25	-0.03	0.03	Angry	-0.25	0.25	Angry
SSSSSSSS	112	22	50	19	24	2	1.12	-0.09	-0.22	Sad	-0.50	-0.51	Sad
SSSSSSAA	112	22	53	17	31	2	0.68	-0.14	-0.11	Sad	-0.50	-0.25	Sad
SSSSSSHH	71	39	55	21	37	1.9	0.83	-0.09	-0.05	Sad	-0.25	-0.26	Sad
SSSSSSTT	111	22	52	18	27	1.1	1.13	-0.02	-0.27	Sad	-0.25	-0.51	Sad
HHHHHHHH	28	6	68	16	103	1	0.26	0.17	0.20	Happy	0.51	0.51	Happy
HHHHHHAA	111	27	66	18	97	1	0.20	0.16	0.21	Happy	0.25	0.51	Happy
HHHHHHSS	132	17	64	23	72	1.3	0.38	0.05	0.06	Happy	0.25	0.25	Happy
HHHHHHTT	78	40	68	18	98	1	0.33	0.11	0.11	Happy	0.51	0.26	Happy
TTTTTTTT	112	22	62	11	60	1	1.1	0.13	-0.13	Tender	0.50	-0.50	Tender
TTTTTTAA	96	21	62	9	60	1	0.71	0.05	-0.05	Tender	0.25	-0.25	Tender
TTTTTTSS	96	21	60	12	46	1	1.16	0.07	-0.19	Tender	0.25	-0.51	Tender
TTTTTTHH	112	22	66	12	76	1	0.68	0.12	0.01	Happy	0.50	-0.24	Tender

Table 40A: Features of Listening Experiment Tunes

Listening Experiment	Valence Difference	Arousal Difference
1*	25%	13%
2	19%	25%
3	19%	25%
4	31%	13%
5	31%	25%
6	25%	13%
7*	50%	31%
8	19%	19%
9	31%	13%
10*	44%	6%
Mean	29% *[23,36]*	18% *[13,23]*

Table 41A: Results of Listening Experiments

Listening Experiment	Combined Error
1*	38%
2	38%
3	44%
4	44%
5	44%
6	31%
7*	56%
8	25%
9	44%
10*	50%
Mean	41% *[36, 47]*

Table 42A: Results of Combined Matching Difference for Listening Experiments

Listening Experiment	Enjoyment	Interest
1*	6.0	6.0
2	2.8	2.6
3	6.4	6.2
4	5.8	6.1
5*	6.6	5.6
6	4.3	4.9
7*	3.2	2.8
8	6.2	6.2
9	3.0	4.1
10*	5.9	5.7
Mean	5.0 *[4.1, 5.9]*	5.0 *[4.1, 5.9]*

Table 43A: Results of Listening Experiments for Enjoyment and Interest

Initial Affective State	Enjoyment	Interest	Order	Mean Number of Notes	Mean IOI	Mean Pitch Spread
HHHHHHTT	6.5	6.2	12	78	0.33	18
HHHHHHSS	5.7	5.9	11	132	0.38	23
HHHHHHAA	5.6	5.8	10	111	0.2	18
HHHHHHHH	5.6	5.1	9	28	0.26	16
TTTTTTSS	5.4	5.4	15	96	1.16	12
TTTTTTTT	5.3	4.7	13	112	1.1	60
TTTTTTHH	5	5.5	16	112	0.68	76
TTTTTTAA	5	5	14	96	0.71	60
SSSSSSHH	4.8	4.7	7	71	0.83	37
SSSSSSTT	4.7	5	8	111	1.13	27
AAAAAAAA	4.6	4.6	1	112	0.2	6
SSSSSSSS	4.6	4.5	5	112	1.12	19
SSSSSSAA	4.5	5	6	112	0.68	17
AAAAAATT	4.4	4.6	4	111	0.68	6
AAAAAASS	4.4	4.1	2	112	0.25	8
AAAAAAHH	4	3.9	3	112	0.2	8

Table 44A: Results of Listening Experiments average across Listeners, sorted by Enjoyment

	Interest	Enjoyment	Order	Mean Number of Notes	Mean IOI	Pitch Spread
HHHHHHTT	6.2	6.5	12	78	0.33	18
HHHHHHSS	5.9	5.7	11	132	0.38	23
HHHHHHAA	5.8	5.6	10	111	0.2	18
AAAAAAAA	5.5	5	16	112	0.68	76
SSSSSSHH	5.4	5.4	15	96	1.16	12
HHHHHHHH	5.1	5.6	9	28	0.26	16
SSSSSSSS	5	5	14	96	0.71	60
SSSSSSAA	5	4.7	8	111	1.13	27
AAAAAASS	5	4.5	6	112	0.68	17
SSSSSSTT	4.7	5.3	13	112	1.1	60
TTTTTTSS	4.7	4.8	7	71	0.83	37
TTTTTTTT	4.6	4.6	1	112	0.2	6
TTTTTTHH	4.6	4.4	4	111	0.68	6
AAAAAATT	4.5	4.6	5	112	1.12	19
AAAAAAHH	4.1	4.4	2	112	0.25	8
TTTTTTAA	3.9	4	3	112	0.2	8

Table 45A: Results of Listening Experiments average across Listeners, sorted by Interest

Appendix 4 – "ASH", a Computer-aided Composition using IPCS

"ASH" is a 2 minute classical piece for solo piano composed by the author with the help of IPCS. ASH can be found online at the thesis web page:

http://cmr.soc.plymouth.ac.uk/alexiskirke/ipcs.html

All of the right-hand (treble clef part) of ASH was composed using a single IPCS run, except for 4 consecutive notes used to create some chord harmonies about a third of the way through the composition, and the last few notes of the composition. The parameters for the IPCS run were as follows:

Number of Agents / Cycles – 8 / 16

Initial Affective States - AASSSSHH (2 "Angry" 4 "Sad", 2 "Happy")

Seed Duration - 0.25

Seed Pitch - 60

maxAgentMemorySize – 250 notes

Similarity Threshold - 1

Affective Onset Influence - 0.1

Affective Loudness Influence - 0.1

Affective Pitch Influence - 0.1

Affective Update Arousal - 0.01

Affective Update Valence - 0.01

kValue - 0.2

These parameters were arrived at by the composer through manual experimentation. It did not take a great deal of time, due to the experience gained running the parametric experiments. It was known that 8 agents created more complexity than 2 so the composer used 8 agents. The arousal and valence affective updates were kept low to prevent any “runaway” accelerandos or deccelerandos. Similarly a low but non-zero k-Value was selected. The composer wanted to produce a performance, not just a composition. However too large a k-value could lead to excessive performative deviations due to recursive application. The composer knew he wanted some shorter notes and so set a low seed duration. Various values were tried, as well as various values of initial affective states.

Once the composer was happy with the tune that was produced by one of the agents, he placed it into a composing tool (Propellorhead Reason) and experimented with different global tempos (though keeping the relative durations and onsets of notes the same). Piano was selected as the instrument as it is the easiest one to simulate. The left-hand (bass clef part) of ASH was composed manually by ear. The composition of the left hand was heavily influenced by at least two elements: the strong structure created by IPCS, and the “humanized” nature of the right-hand part (due to the non-zero k-Value). The non-zero k-Value helped to encourage the composer to be far more free with their timing in a way they found attractive. The strong structure already apparent in the IPCS tune shaped how the left-hand developed and repeated, but also encouraged the more free timing (since the IPCS structure held the tune together strongly).

The composition highlighted the fact that IPCS does not really produce tune-endings. So a few notes had to be added manually at the end to create a satisfactory conclusion. The fact IPCS did not produce a natural ending, and the desire to not interfere with the interesting structures IPCS had produced, meant that the human composed ending

became slightly unusual in nature. But this was seen by the composer as a positive point.

It was satisfying to note that the integrated expressive performance could be clearly heard and appreciated in the final composition. However this also means it is not possible to produce a meaningful music notation score, because the timings would come over as far too complex in common notation.

Appendix 5 – List of Abbreviations

A	Articulation
ANN	Artificial Neural Network
BBN	Bayesian Belief Network
CBR	Case-based Reasoning
CSEMP	Computer System for Expressive Music Performance
CMERS	Computational Music Emotion Rule System by Livingstone et al
D	Dynamics
DM	Director Musices (KTH System)
EC	Evolutionary Computing
GA	Genetic Algorithm
GP	Genetic Programming
GPR	Gaussian Process Regression
GTTM	Lerdahl and Jackendoff's Generative Theory of Tonal Music
GUI	Graphical User Interface
HMM	Hidden Markov Model
IBL	Instance-based Learning
ILE	IPCS Linear Estimator
IOI	Inter-onset Interval
IR	Narmour's Implication/Realisation Theory of Melody
K	Attack
KCCA	Kernel Canonical Correlation Analysis
kNN	k-Nearest Neighbour
KRR	Kernel Ridge Regression
LBDM	Local Boundary Detection Model of Cambouropoulos
MAS	Multi-agent System
MIDI	Musical Instrument Digital Interface
MusicXML	Music Extensible Markup Language
MIS	Music Interpretation System by Katayose et al
N	Note addition/consolidation
P	Pitch

PCA	Principal Component Analysis
PSO	Particle Swarm Optimisation
T	Tempo
TSR	Time Span Reduction Technique (from GTTM)
V	Vibrato

References

ABRAHAM, A., GUO, H., AND LIU, H. 2006 Swarm intelligence: foundations, perspectives and applications. In *Swarm Intelligent Systems, Studies in Computational Intelligence*, N. NEDJAH and L. MOURELLE, Eds. Springer-Verlag, 3-25.

ANDERS, T. 2007. Composing Music by Composing Rules: Design and Usage of a Generic Music Constraint System. Thesis/Dissertation, Queens University, Belfast

ARCOS, J.L., DE MANTARAS, R.L., AND SERRA, X. 1997. SaxEx: A Case-based Reasoning System for Generating Expressive Musical Performances. In *Proceedings of 1997 International Computer Music Conference*, Thessalonikia, Greece.

ARCOS, J.L., LOPEZ DE MANTARAS, R., AND SERRA, X. 1998. Saxex: A Case-Based Reasoning System for Generating Expressive Musical Performance. *Journal of New Music Research 27*, 194-210.

ARIZA, C. 2005. Navigating the Landscape of Computer-Aided Algorithmic Composition Systems: A Definition, Seven Descriptors, and a Lexicon of Systems and Research. In *Proceedings of the 2005 International Computer Music Conference*, San Francisco, California.

BAXTER, J., HORN, G., AND LEIVERS, D. 2007. Fly-by-Agent: Controlling a Pool of UAVs via a Multi-Agent System. In *Proceedings of the Twenty-seventh SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (AI-2007)*, Cambridge, UK.

BECKERS, R., HOLLAND, O.E., AND DENEUBOURG, J.L. 1995. From local actions to global tasks: stigmergy and collective robotics. In *Proceedings of the Fourth International Conference on the Synthesis of Living Systems*, Cambridge, MA, USA.

BEDUA, M. 2003. Artificial life: organization, adaptation and complexity from the bottom up. *Trends in Cognitive Sciences 7*, 505-512.

BELLIFEMINE, F., POGGI, A., AND RIMASSI, G. 1999. JADE: A FIPA-Compliant agent framework. In *Proceedings of the 4th International Conference on Practical Applications of Intelligent Agents and Multi-Agents*, London, UK.

BENI, G. 2005 From Swarm Intelligence to Swarm Robotics. In *Swarm Robotics*, G. BENI, Eds. Springer Berlin / Heidelberg, 1-9.

BENTLEY, P.J. 2005. Everything Computes. In *Proceedings of the Third International Conference on Generative Art*, Melbourne, Australia.

BEYLS, P. 1989. The Musical Universe of Cellular Automata. In *Proceedings of the 1989 International Computer Music Conference*, San Francisco, California.

BEYLS, P. 2007. Interaction and Self-organisation in a Society of Musical Agents. In *Proceedings of ECAL 2007 Workshop on Music and Artificial Life (MusicAL 2007)*, Lisbon, Portugal.

BIGAND, E., VIEILLARD, S., MADURELL, F., MAROZEAU, J., AND DACQUET, A. 2005. Multidimensional scaling of emotional responses to music: The effect of musical expertise and of the duration of the excerpts. *Cognition and Emotion 19*, 1113-1139.

BISIG, D. AND NEUKOM, N. 2008. Interactive Swarm Orchestra, an Artificial Life Approach to Computer Music. In *Proceedings of the 11th Generative Art Conference*, Milan, Italy.

BLACKWELL, T. AND BENTLEY, P. 2002 Improvised Music with Swarms. In *Proceedings of the 2002 World Congress on Evolutionary Computation*, Honolulu, Hawaii.

BLACKWELL, T. 2007 Swarming and Music. In *Evolutionary Computer Music*, E. MIRANDA and J. BILES, Eds. Springer, 194-217.

BLUM, C. AND ROLI, A. 2003. Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Computing Surveys 35*, 268-308.

BODEN, M. AND EDMONDS, E. 2009. What is generative art? *Digital Creativity 20*, 21-46.

BONABEAU, B., THERAULAZ, G., AND DORIGO, M. 1999. *Swarm Intelligence: From Natural To Artificial Systems*. Oxford University Press.

BRESIN, R. AND FRIBERG, A. 2000. Emotional Coloring of Computer-Controlled Music Performances. *Computer Music Journal 24*, 44-63.

BROWN, A. 2002. Opportunities for Evolutionary Music Composition. In *Proceedings of 2002 Australasian Computer Music Conference*, Melbourne, Australia.

BUGMANN, G. 1998. Classification using Networks of Normalized Radial Basis Functions. In *Proceedings of the 1998 International Conference on Advances in Pattern Recognition (ICAPR'98)*, Plymouth, UK.

Burns, K. 1994. The History and Development of Algorithms in Music Composition, 1957-1993. Thesis/Dissertation, Ball State University

BUXTON, W.A.S. 1977. A Composers Introduction to Computer Music. *Interface 6*, 57-72.

CAMBOUROPOULOS, E. 2001. The Local Boundary Detection Model (LBDM) and its Application in the Study of Expressive Timing. In *Proceedings of the 2001 International Computer Music Conference*, Havana, Cuba.

CAMURRI, A., DILLON, R., AND SARON, A. 2000. An Experiment on Analysis and Synthesis of Musical Expressivity. In *Proceedings of 13th Colloquium on Musical Informatics*, L'Aquila, Italy.

CANAZZA, S., DRIOLI, C., DE POLI, G., RODÀ, A., AND VIDOLIN, A. 2000. Audio Morphing Different Expressive Intentions for Multimedia Systems. *IEEE Multimedia 7*, 79-83.

CANAZZA, S., DE POLI, G., DRIOLI, C., RODÀ, A., AND VIDOLIN, A. 2004. Modeling and control of expressiveness in music performance. *The Proceedings of the IEEE 92*, 686-701.

CARLSON, L., NORDMARK, A., AND WIKLANDER, R. 2003. *Reason Version 2.5 - Getting Started*. Propellorhead Software.

CHOMSKY, N. 1956. Three Models for the Description of Language. *IRE Transactions on Information Theory 2*, 113-123.

CLAIR, R., MONMARCHE, N., AND SLIMANE, M. 2008. Interactions between an artificial colony of musical ants and an impaired human composer: towards

accessible generative arts. In *Proceedings of the 11th Generative Art Conference*, Milan, Italy.

CLARKE, E. AND WINDSOR, W. 2000. Real and Simulated Expression: a Listening Study. *Music Perception 17*, 277-313.

CLARKE, E.F. 1991 Expression and communication in musical performance. In *Music, Language, Speech and Brain*, J. SUNDBERG, L. NORD, and R. CARLSON, Eds. Macmillan Press,

CLARKE, E.F. 1998 Generative Principles in Music Performance. In *Generative Processes in Music: The Psychology of Performance, Improvisation, and Composition*, J.A. SLOBODA, Eds. Clarendon Press, 1-26.

CLYNES, M. 1986. Generative principles of musical thought: Integration of microstructure with structure. *Communication and Cognition 3*, 185-223.

CLYNES, M. 1995. Microstructural musical lingusitics: Composer's pulses are liked best by the best musicians. *Cognition 55*, 269-310.

COLLINS, N. 2009. Form and Algorithmic Composition. *Contemporary Music Review 28*, 103-114.

COPE, D. 2005. *Computer Models of Musical Creativity*. MIT Press.

DAHLSTEDT, P. AND MCBURNEY, P. 2006. Musical agents: Toward Computer-Aided Music Composition Using Autonomous Software Agents. *Leonardo 39*, 469-470.

DAHLSTEDT, P. 2007. Autonomous Evolution of Complete Piano Pieces and Performances. In *Proceedings of ECAL 2007 Workshop on Music and Artificial Life (MusicAL 2007)*, Lisbon, Portugal.

DE POLI, G. 2004. Methodologies for expressiveness modeling of and for music performance. *Journal of New Music Research 33*, 189-202.

DELGADO, M., FAJARDO, W., AND MOLINA-SOLANA, M. 2008. Inmamusys: Intelligent multiagent music system. *Expert Systems with Applications 36*, 4574-4580.

DI FATTA, G. AND FORTINO, G. 2007. A customizable multi-agent system for distributed data mining. In *Proceedings of the 2007 ACM symposium on Applied computing*, Seoul, Korea.

DORIGO, M. AND GLOVER, F. 1999 The ant colony optimization meta-heuristic. In *New ideas in optimization*, D. CORNE, M. DORIGO, and F. GLOVER, Eds. McGraw-Hill, 11-32.

EEROLA, T. AND TOIVIANEN, P. 2004. MIR in Matlab: The MIDI Toolbox. In *Proc. 5th International Conference on Music Information Retrieval*, Barcelona, Spain.

EIGENFELDT, A. 2009 The Evolution of Evolutionary Software: Intelligent Rhythm Generation in Kinetic Engine. In *Applications of Evolutionary Computing*, Springer, 498-507.

EMDE, W. AND WETTSCHERECK, D. 1996. Relational Instance Based Learning. In *Proceedings of 13th International Conference on Machine Learning*, Bari, Italy.

EVANS, D., HEUVELINK, A., AND NETTLE, D. 2003. The evolution of optimism: A multi-agent based model of adaptive bias in human judgement. In *Proceedings of the AISB'03 Symposium on Scientific Methods for the Analysis of Agent-Environment Interaction* , Wales, UK.

EVANS, P. AND SCHUBERT, E. 2006. Quantification of Gabrielsson's relationships between felt and expressed emotions in music. In *Proceedings of the 9th International Conference on Music and Cognition*, Bologna, Italy.

FONSEKA, J. 2000. Musical Agents. Thesis/Dissertation, Monash University

FRIBERG, A. 1995. Matching the rule parameters of Phrase Arch to performances of "Träumerei:" A preliminary study. In *Proceedings of the KTH Symposium on Grammars for Music Performance*, Stockholm, Sweden.

FRIBERG, A. AND SUNDBERG, J. 1999. Does music performance allude to locomotion? A model of final ritardandi derived from measurements of stopping runners. *Journal of Acoustical Society of America 105*, 1469-1484.

FRIBERG, A., BRESIN, R., AND SUNDBERG, J. 2006. Overview of the KTH rule system for musical performance. *Advances in Cognitive Psychology 2*, 145-161.

GABRIELSSON, A. AND JUSLIN, P. 1996. Emotional expression in music performance: between the performer's intention and the listener's experience. *Psychology of Music 24*, 68-91.

GABRIELSSON, A. 2003. Music performance research at the millennium. *Psychology of Music 31*, 221-272.

GEORGEFF, M. 2009. The gap between software engineering and multi-agent systems: bridging the divide. *International Journal of Agent-Oriented Software Engineering 3*, 391-396.

GOLDMAN, C.V., GANG, D., ROSENSCHEIN, J.S., AND LEHMANN, D. 1999. NetNeg: A Connectionist-Agent Integrated System for Representing Musical Knowledge. *Annals of Mathematics and Artificial Intelligence 25*, 69-90.

GONG, T., ZHANG, Q., AND WU, H. 2005. Music evolution in a complex system of interacting agents. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, Edinburgh, UK.

GOOD, M. 2001. MusicXML for notation and analysis. *Computing in Musicology 12*, 113-124.

HAMANAKA, M., HIRATA, K., AND TOJO, S. 2004. An Automatic Music Analyzing System based on GTTM: Acquisition of Grouping Structures and Metrical Structures. *Joho Shori Gakkai Kenkyu Hokoku 2004*, 1-8.

HASHIDA, M., NAGATA, N., AND KATAYOSE, H. 2006. Pop-E: a performance rendering system for the ensemble music that considered group expression. In *Proceedings of 9th International Conference on Music Perception and Cognition*, Bologna, Spain.

HERR, C.M. AND KVAN, T. 2009. Adapting cellular automata to support the architectural design process. *Automation in Construction 16*, 61-69.

HILLER, L. AND ISAACSON, L. 1959. *Experimental Music. Composition with an Electronic Computer*. McGraw Hill.

HOLLAND, S. 2000 Artificial Intelligence in Music Education: a critical review. In *Readings in Music and Artificial Intelligence*, E. MIRANDA, Eds. Harwood Academic Publishers, 239-274.

JACOB, C., HUSHLAK, G., BOYD, J., NUYTTEN, P., SAYLES, M., AND PILAT, M. 2007. SwarmArt : Interactive Art from Swarm Intelligence. *Leonardo 40*, 248-254.

JAMES, G., COHEN, D., DODIER, R., PLATT, G., AND PALMER, D. 2006. A deployed multi-agent framework for distributed energy applications. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, Hakodate, Japan.

JUSLIN, P. AND LAUKKA, P. 2004. Expression, perception, and induction of musical emotion: a review and a questionnaire study of everyday listening. *Journal of New Music Research 33*, 216-237.

JUSLIN, P. 2005 From Mimesis to Catharsis: expression, perception and induction of emotion in music. In *Music Communication*, D. MIELL, R. MACDONALD, and D.J. HARGREAVES, Eds. Oxford University Press, 85-116.

JUSLIN, P. 2003. Five Facets of Musical Expression: A Psychologist's Perspective on Music Performance. *Psychology of Music 31*, 273-302.

JUSLIN, P. 2001. Communicating Emotion in Musical Performance: a Review and Theoretical Framework. In Juslin, P.N., Sloboda, J.A., Music and Emotion: Theory and Research (pp309-337), New York: Oxford University Press.

KELLY, K. 1995. *Out of Control: the new biology of machines*. Fourth Estate Paperbacks.

KENNEDY, J. AND EBERHART, R. 1995. Particle Swarm Optimization. In *Proceedings of the IEEE International Conf.on Neural Networks*, Perth, Australia.

KIRKE, A. AND MIRANDA, E. 2007. Capturing the Aesthetic: Radial Mappings for Cellular Automata Music. *Journal of the ITC Sangeet Research Academy 21*, 15-23.

KIRKE, A.J. AND MIRANDA, E.R. 2009. A Survey of Computer Systems for Expressive Performance of Music. *ACM Computing Surveys 42*, 3:1-3:41.

KRUMHANSL, C. AND KESSLER, E. 1982. Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological Review 89*, 334-368.

KRUMHANSL, C. 1991. *Cognitive Foundations of Musical Pitch*. Oxford University Press.

LANG, P. 1995. The emotion probe: Studies of motivation and attention. *The American Psychologist 50*, 372-385.

LANGTON, C. 1990. Computation at the edge of chaos. *Physica D: Nonlinear Phenomena 42*, 12-37.

LANGTON, C.G. 1986. Studying Artificial Life with Cellular Automata. *Physica D 22*, 120-149.

LASHKARI, Y., METRAL, M., AND MAES, P. 1994. Collaborative Interface Agents. In *Proceedings of the 1994 National Conference on Artificial Intelligence*.

LEGASPI, R., HASHIMOTO, Y., MORIYAMA, K., KURIHARA, S., AND NUMAO, M. 2007. Music Compositional Intelligence with an Affective Flavor. In *Proceedings of the 2007 International Conference on Intelligent User Interfaces*, Honolulu, Hawaii, USA.

LERDAHL, F. AND JACKENDOFF, R. 1983. *A Generative Theory of Tonal Music*. The MIT Press.

LIN, D.-T. AND LIAO, G.-J. 2008. Swarm Intelligence Based Fuzzy C-Means Clustering for Motion Vector Selection in Video Watermarking. *International Journal of Fuzzy Systems 10*, 185-194.

LINDSTROM, E., JUSLIN, P., BRESIN, R., AND WILLIAMON, A. 2003. Expressivity comes from within your soul. *Research Studies in Music Education 20*, 23-47.

LIVINGSTONE, S. 2008. Changing Musical Emotion through Score and Performance with a Computational Rule System. Thesis/Dissertation, The University of Queensland

LIVINGSTONE, S., MUHLBERGER, R., BROWN, A., AND THOMPSON, W.F. 2010. Changing Musical Emotion: A Computational Rule System for Modifying Score and Performance. *Computer Music Journal 34*, 41-64.

LIVINGSTONE, S.R., MUHLBERGER, R., BROWN, A.R., AND LOCH, A. 2007. Controlling Musical Emotionality: An Affective Computational Architecture for Influencing Musical Emotions. *Digital Creativity 18*, 43-53.

LLOYD, J. 1984. *Foundations of logic programming*. Springer-Verlag.

LOPEZ DE MANTARAS, R. AND ARCOS, J. 2002. AI and music: From composition to expressive performance. *AI Magazine 23*, 43-57.

MARON, O. AND LOZANO-PEREZ, T. 1998. A framework for multiple-instance learning. In *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems*, Denver, CO, USA.

MARTINS, J.M. AND MIRANDA, E.R. 2007. Emergent Rhythmic Phrases in an A-Life Environment. In *Proceedings of ECAL 2007 Workshop on Music and Artificial Life (MusicAL 2007)*, Lisbon, Portugal.

MCCORMACK, J. 1996. Grammar-based music composition. *Complex International 3*.

MINASSIAN, C., GAYFORD, C., AND SLOBODA, J.A. 2003. Optimal experience in musical performance: a survey of young musicians. In *Proceedings of the 2003 Meeting of the Society for Education, Music and Psychology Research*, London, UK.

MIRANDA, E. 1994. Music composition using cellular automata. *Languages of Design 2*, 105-117.

MIRANDA, E. 1995. Granular Synthesis of Sounds by means of a Cellular Automaton. *Leonardo 28*, 297-300.

MIRANDA, E., KIRKE, A., AND ZHANG, Q. 2010. Artificial Evolution of Expressive Performance of Music: An Imitative Multi-Agent Systems Approach. *Computer Music Journal 34*, 80-96.

MIRANDA, E. 2001. Evolving cellular automata music: From sound synthesis to composition. In *Proceedings of 2001 Workshop on Artificial Life Models for Musical Applications*, Prague, Czech Republic.

MIRANDA, E.R. 2001. *Composing Music With Computers*. Focal Press.

MIRANDA, E.R. 2002. Emergent Sound Repertoires in Virtual Societies. *Computer Music Journal 26*, 77-90.

MIRANDA, E.R. 2003. On the evolution of music in a society of self-taught digital creatures. *Digital Creativity 14*, 29-42.

MORGAN, N. 2007. Transformation and mapping of L-Systems data in the composition of a large-scale instrumental work. In *Proceedings of ECAL 2007 Workshop on Music and Artificial Life (MusicAL 2007)*, Lisbon, Portugal.

MURRAY-RUST, D.S. AND SMAILL, A. 2005. Musical Acts and Musical Agents. In *Proceedings of the 5th Open Workshop of MUSICNETWORK:* Vienna, Austria.

MURRAY-RUST, D.S., SMAILL, A., AND MAYA, M.C. 16-8-2005. VirtuaLatin - Towards a Musical Multi-Agent System. In *Proceedings of Sixth International Conference on Computational Intelligence and Multimedia Applications*, Las Vegas, NV, USA.

NARMOUR, E. 1990. *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*. The University of Chicago Press.

NYMAN, N. 1974. *Experimental Music: Cage and Beyond*. Cambridge University Press.

OEHME, A., HERBON, A., KUPSCHICK, S., ZENTSCH, E. 2007. Physiological correlates of emotions, in *Proceedings of the 2007 Conference on Artificial Intelligence and Simulation of Behaviour*. Newcastle, UK.

OLIVEIRA, A.P. AND CARDOSO, A. 2009. Automatic Manipulation of Music to Express Desired Emotions. In *Proceedings of the 6th Sound and Music Computing Conference*, Porto, Brazil.

PACE, I. 2007. Complexity as Imaginative Stimulant: Issues of Rubato, Barring, Grouping, Accentuation and Articulation in Contemporary Music, with Examples from Boulez, Carter, Feldman, Kagel, Sciarrino, Finnissy. In *Proceedings of the 5th International Orpheus Academy for Music & Theory*, Gent, Belgium.

PALMER, C. 1997. Music performance. *Annual Review of Psychology 48*, 115-138.

PANZARASA, P. AND JENNINGS, N. 2006 Collective cognition and emergence in multi-agent systems. In *Cognition and Multi-Agent Interaction: From cognitive modelling to social simulation*,Cambridge University Press, 401-408.

PAPADOPOULOS, G. AND WIGGINS, G. 1999. AI methods for algorithmic composition: A survey, a critical view and future prospects. In *Proceedings of the AISB'99 Symposium on Musical Creativity*, Brighton, UK.

PARKER, L. 1994. Heterogeneous Multi-Robot Cooperation. Thesis/Dissertation, MIT, Massachusetts, USA.

PARKES, D. AND HUBERMAN, B. 2001. Multiagent Cooperative Search for Portfolio Selection. *Games and Economic Behavior 35*, 124-165.

PARUNAK, H., NIELSEN, P., BRUECKNER, S., AND ALONSO, R. 2007. Hybrid Multi-Agent Systems: Integrating Swarming and BDI Agents. In *Proceedings of the 2006 Workshop on Engineering Self-Organizing Agents*, Hakodate, Japan.

PEARCE, M. AND WIGGINS, G. 2001. Towards a framework for the evaluation of machine compositions. In *Proceedings of the AISB2001 Symposium on AI and Creativity in the Arts and Sciences*, Brighton, UK.

PERETZ, I., GAGNON, L., AND BOUCHARD, B. 1998. Music and emotion: perceptual determinants, immediacy, and isolation after brain damage. *Cognition 68*, 111-141.

PHAM, D.T., GHANBARZADEH, A., KOC, E., OTRI, S., RAHIM, S., AND ZAIDI, M. 2006. The Bees Algorithm - A Novel Tool for Complex Optimisation Problems. In *Proceedings of IPROMS 2006 Conference*, Oxford, UK.

RAMCHURN, S., HUYUNH, D., AND JENNINGS, N. 2004. Trust in multi-agent systems. *The Knowledge Engineering Review 19*, 1-25.

RAMIREZ, R., HAZAN, A., MAESTRE, E., AND SERRA, X. 2008. A Genetic Rule-Based Model of Expressive Performance for Jazz Saxophone. *Computer Music Journal 32*, 38-50.

RAO, S. AND GEORGEFF, M. 1991. Modeling Rational Agents within a BDI-Architecture. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, Boston, MA, USA.

REPP, B.H. 1990. Composer's pulses: Science or art. *Music Perception 7*, 423-434.

REYNOLDS, C. 1987. Flocks, herds and schools: a distributed behavioral model. *Computer Graphics 21*, 25-34.

ROADS, C. 1996. *The Computer Music Tutorial*. MIT Press.

ROSENBERG, G. AND SALOMAA, A. 1980. *The Mathematical Theory of L Systems.* MIT Press.

ROSIN, P.L. 2006. Training cellular automata for image processing. *IEEE Transactions on Image Processing 15*, 2076-2087.

SABATER, J. AND SIERRA, C. 2002. Reputation and social network analysis in multi-agent systems. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, Bologna, Italy.

SABATER, J. AND SIERRA, C. 2005. Review on Computational Trust and Reputation Models, *Artificial Intelligence Review* 24, 33-60.

SAMPAIO, P.A., RAMALHO, G., AND TEDESCO, P. 2008. CinBalada: a multiagent rhythm factory. *Journal of the Brazilian Computer Society 14*, 31-49.

SCHOENBERG, A. 1970. *Fundamentals of Music Composition*. Mackays of Chatham.

SCHOENHARL, T., MADEY, G., SZABO, G., AND BARABASI, A. 2006. WIPER: A Multi-Agent System for Emergency Response. In *Proceedings of the 3rd International ISCRAM Conference*, Newark, NJ, USA.

SEASHORE, C.E. 1938. *Psychology of Music*. McGraw-Hill.

SHAFFER, L. AND TODD, N. 1987 The interpretive component in musical performance. In *Action and Perception in Rhythm and Music*,R.Swed. Acad. Music, 139-152.

SHAWE-TAYLOR, J. AND CRISTIANINI, N. 2000. *An Introduction to Support Vector Machines*. Cambridge University Press.

SHOTTSTAEDT, W. 1989 Automatic counterpoint. In *Current directions in computer music research*,MIT Press, 199-214.

SIM, K. AND SUN, W. 2003. Ant colony optimization for routing and load-balancing: survey and new directions. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems, Man and Cybernetics 33*, 560-572.

SPICER, M., TAN, B., AND TAN, C. 2003. A Learning Agent Based Interactive Performance System. In *Proceedings of the 2003 International Computer Music Conference*, San Francisco, USA.

SPICHER, A., FATES, N., AND SIMONIN, O. 2009. From reactive multi-agent models to cellular automata -- illustration on a diffusion-limited aggregation model. In *Proceedings of 1st International Conference on Agents and Artificial Intelligence*, Porto, Portugal.

STEELS, L. AND BELPAEME, T. 2005. Coordinating perceptually grounded categories through language: a case study for colour. *Behavioral and Brain Sciences 28*, 469-489.

SUNDBERG, J., ASKENFELT, A., AND FRYDÉN, L. 1983. Musical performance. A synthesis-by-rule approach. *Computer Music Journal 7*, 37-43.

SUZUKI, T., TOKUNAGA, T., AND TANAKA, H. 1999. A Case Based Approach to the Generation of Musical Expression. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, August 1999*.

SUZUKI, T. 2003. Kagurame Phase-II. In *Proceedings of 2003 International Joint Conference on Artificial Intelligence (Working Notes of RenCon Workshop)*, Acapulco, Mexico.

TEMPERLEY, D. AND SLEATOR, D. 1999. Modeling meter and harmony: a preference rule approach. *Computer Music Journal 23*, 10-27.

THOMPSON, W.F. 1989. Composer-specific aspects of musical performance: An evaluation of Clynes' theory of pulse for performances of Mozart and Beethoven. *Music Perception 7*, 15-42.

TOBUDIC, A. AND WIDMER, G. 2003. Playing Mozart phrase by phrase. In *Proceedings of the 5th International Conference on Case-based Reasoning*, Trondheim, Norway.

TOBUDIC, A. AND WIDMER, G. 2003. Relational ibl in music with a new structural similarity measure. In *Proceedings of the 13th International Conference on Inductive Logic Programming*, Szeged, Hungary.

TOBUDIC, A. AND WIDMER, G. 2003. Learning to Play Mozart: Recent Improvements. In *Proceedings of 2003 International Joint Conference on Artificial Intelligence (Working Notes of RenCon Workshop)*, Acapulco, Mexico.

TOBUDIC, A. AND WIDMER, G. 2003. Technical Notes for Musical Contest Category. In *Proceedings of 2003 International Joint Conference on Artificial Intelligence (Working Notes of RenCon Workshop)*, Acapulco, Mexico.

TODD, N.P. 1985. A model of expressive timing in tonal music. *Music Perception 3*, 33-58.

TODD, N.P. 1989. A computational model of Rubato. *Contemporary Music Review 3*, 69-88.

TODD, N.P. 1992. The Dynamics of Dynamics: A Model of Musical Expression. *Journal of the Acoustical Society of America 91*, 3540-3550.

TODD, N.P. 1995. The kinematics of musical expression. *Journal of Acoustical Society of America 97*, 1940-1949.

TRAINOR, L. AND SCHMIDT, L. 2003 Processing emotions induced by music. In *The cognitive neuroscience of music*, I. PERETZ and R. ZATORRE, Eds. Oxford University Press, 310-324.

UEDA, L.K. AND KON, F. 2003. Andante: A Mobile Musical Agents Infrastructure. In *Proceedings of the 9th Brazilian Symposium on Computer Music*, Campinas, Brazil.

UEDA, L.K. AND KON, F. 2004. Mobile musical agents: the andante project. In *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, Vancouver, Canada.

VALDEZ, S. 2006. *An Introduction to Global Financial Markets*. Palgrave Macmillan.

VON MAMMEN, S., JACOB, C., AND WONG, J. 2008 Virtual constructive swarms: Compositions, inspirations, and designs. In *Proceedings of EvoWorkshops 2008*,Springer-Verlag, 491-496.

WEISS, G. 1999 Prologue of Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. In *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, G. WEISS, Eds. MIT Press, 1-23.

WHALLEY, I. 2009. Software Agents in Music and Sound Art Research/Creative Work: current state and a possible direction. *Organised Sound 14*, 156-167.

WIDMER, G. 2003. Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries. *Artificial Intelligence 146*, 129-148.

WIDMER, G. AND TOBUDIC, A. 2003. Playing Mozart by analogy: Learning multi-level timing and dynamics strategies. *Journal of New Music Research 32*, 259-268.

WIDMER, G. AND GOEBL, W. 2004. Computational Models of Expressive Music Performance: The State of the Art. *Journal of New Music Research 33*, 203-216.

WOOLDRIDGE, M. 2004. *An Introduction to Multi-Agent Systems*. John Wiley and Sons.

WOOLER, R., BROWN, A., MIRANDA, E., BERRY, R., AND DIEDERICH, J. 2005. A framework for comparison of process in algorithmic music systems. In *Proceedings of Generative Arts Practice 2005*, Sydney, Australia.

WULFHOST, R.D., FLORES, L.V., NAKAYAMA, L., FLORES, C.D., ALVARES, O.C., AND VICARI, R.M. 2003. An Open Architecture for a Musical Multi-Agent System. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, Melbourne, Australia.

WULFHOST, R.D., NAKAYAMA, L., AND VICARI, R.M. 14-7-2003. A Multiagent Approach for Musical Interactive Systems. In *Proceedings of 2nd International Joint Conference on Autonomous Agents and Multi-Agent System (AAMAS 2003)*, Melbourne, Australia.

XENAKIS, I. 1963. *Formalized Music: Thought and Mathematics in Composition*. Pendragon Press.

XU, R., ANAGNOSTOPOULOS, G., AND WUNSCH, D. 2007 A hybrid of neural classifier and swarm intelligence in multi-class cancer diagnosis with gene

expression signatures. In *Computational Intelligence in Bioinformatics*, G. FOGEL, D. CORNE, and Y. PAN, Eds. IEEE Press / Wiley, 3-20.

ZAHLER, N. 1991, Questions about interactive computer performance as a resource for music composition and problems of musicality. In *Proceedings of the 1991 International Computer Music Conference (ICMC 1991)*, ICMA, Glasgow, UK.

ZENTNER, M., GRANDJEAN, D., SCHERER, K. 2008. Emotions Evoked by the Sound of Music: Characterization, Classification, and Measurement. *Emotion 8:4*, 494-521.

ZHANG, Q. AND MIRANDA, E.R. 2007. Evolving Expressive Music Performance through Interaction of Artificial Agent Performers. In *Proceedings of ECAL 2007 Workshop on Music and Artificial Life (MusicAL 2007)* , Lisbon, Portugal.