DEVELOPING AND COMPOSING FOR A ROBOTIC MUSICIAN USING DIFFERENT MODES OF INTERACTION

Mason Bretan, Marcelo Cicconet, Ryan Nikolaidis, Gil Weinberg

Georgia Institute of Technology
Center for Music Technology, Atlanta, GA
masonbretan,cicconet,ryannikolaidis@gmail.com, gilw@gatech.edu

ABSTRACT

This paper describes three compositions written for the improvisational robotic marimba player, Shimon. The three compositions demonstrate multiple modes of interaction between Shimon and a human performer. Some of these modes are created using techniques from pattern recognition, tempo detection, and computer vision. We illustrate the process of composing and developing for Shimon while attempting to maintain creativity and musicality considering the limitations and opportunities that emerge when working with a mechanical apparatus.

1. INTRODUCTION

Research in robotic musicianship opens the door for new and interesting methods of creating and performing computer music. The combination of software and mechanics provides unique opportunity and limitation for developers, composers, and performers. Robotic musicians greatly differ from the pure software based computer music applications which bestow an unlimited library of timbre, software, physical mechanism, and any additional vision methods when referring to Shimon we refer to the amalgamation of physical mechanisms, software, human-computer interaction, and computer vision methods, including methods such as pattern recognition, tempo detection, improvisation, and gesture recognition. It is important to note that when referring to Shimon we refer to the amalgamation of software, physical mechanism, and any additional vision system.

2. COMPOSING FOR SHIMON

There are a few mechanical idiosyncrasies inherent to Shimon that make composing and developing both interesting and challenging. These characteristics influence both the creative and scientific processes of working with Shimon.

2.1. Shimon’s Arms

Servo motors, solenoids, and rotors actuate Shimon’s performance. As shown in Figure 1 Shimon has four arms, each holding two mallets. We control Shimon’s arms by sending midi messages (consisting of midi note number and velocity) at the instance a note should sound. There is a fixed 500ms delay from the time the message is sent to the time the arm moves into position and the mallet strikes the corresponding key. However, because messages are sent to individual arms it is very possible that when a message is sent to a specific arm to move to a certain location there may be another arm occupying the space at that location. When this occurs there are four options: play the note in another available octave, do not play the note at all, move the arm that is currently occupying the space, or play the note using the arm currently in its location. The third and fourth options require the programmer to anticipate the location of each arm and move them accordingly so that the arms locations do not conflict and no notes are dropped. We have adopted the term “path planning” to describe this process.

For many of our applications we use a general purpose path planning algorithm which first selects the arm closest to the note to play and then if for any reason cannot play the note (most likely because it is too close to the vicinity of an adjacent arm) it attempts to play the note in an alternative octave. Finally if none of these octaves exist the note is dropped. This general purpose path planning algorithm is sufficient most of our applications. However, in some cases (when it is essential to the music for Shimon to play specific motifs exactly as written) a more sophisticated predetermined path planning is necessary.

2.2. Head Movement

In addition to path planning, programming the movement of Shimon’s head is another feature unique to robotic musicianship. Hoffman and Weinberg [4] demonstrated the advantage of visual cues such as arm movement and mallet strikes. Through empirical study, we found that well programmed head movements can provide even more valuable visual cues to an accompanying musician. These head related visual cues include gestures varying from Shimon looking towards a person versus its own arms, positioning its head high above the marimba versus very close, and nodding to the beat versus more generic breathing gestures. These cues provide an environment more similar to human-human interaction as opposed to human-software interaction. Shimon develops a defined personality and any interaction becomes social and naturally engaging. In our compositions a combination of previously developed higher-level head gestures (such as head banging to the beat) are used in coalescence with lower level commands (such as head and neck rotation).

3. BAFANA

Bafana was inspired by street performances of African marimba bands. The piece moves through several phases that use a number of repetitive interlocking rhythmic motifs, played interdependently on two marimbas by Shimon and a human performer. Shimon listens to the human improvisation, detects specific motifs, and responds in a manner that is times predictable and at times surprising. Each phase presents a different situation for the human player as Shimon is programmed to respond differently. The piece begins with a playful vision-based interaction between the human and the robot that introduces some of the musical motifs in a social non-rhythmic context. The human player then chooses any of nine possible motifs to start the piece, plays it in a loop, and awaits Shimon’s response. In the current implementation of the piece, Shimon’s personality develops from predictable consonant responses at the beginning, to more surprising chromatic reactions towards the end, challenging the human player to constantly listen and carefully adapt to the evolving nature of the music.

3.1. Part 1 - Vision Based Interaction

In the first section Shimon responds to human arm movements analyzed by the Microsoft Kinect. The Kinect is able to provide positional values describing the depth, height, and width of each arm in relation to the camera. Each arm triggers one of two motifs (a total of four motifs). The horizontal position determines the particular motif to be played. The depth and height values influence the tempo and velocity, respectively. Though Shimon may take a more deterministic role, the importance of detecting physical gestures to influence the music still resides. Humans use a plethora of physical movements to convey emotional or descriptive pieces of information. In performance humans often cue downbeats or transitions through exaggerated gestures. In Bafana, physical communication with Shimon parallels the performance of a conductor, in which the conductor makes movements to elicit musical responses from the musicians. Figure 3 shows a human performer in this state of interaction with Shimon.

3.2. Part 2 - Melodic Interaction

The vision-based interaction serves as a precursor to the more involved melodic interaction, which serves as the principal mode of interaction for the piece. As previously
DEVELOPING AND COMPOSING FOR A ROBOTIC MUSICIAN USING DIFFERENT MODES OF INTERACTION

Mason Bretan, Marcelo Cicconet, Ryan Nikolaidis, Gil Weinberg
Georgia Institute of Technology
Center for Music Technology, Atlanta, GA
masonbretan,cicconet,ryan Nikolaidis@gmail.com, gilw@gatech.edu

ABSTRACT

This paper describes three compositions written for the improvisational robotic marimba player, Shimon. The three compositions demonstrate multiple modes of interaction between Shimon and a human performer. Some of these modes are created using techniques from pattern recognition, tempo detection, and computer vision. We illustrate the process of composing and developing for Shimon while attempting to maintain creativity and musicality considering the limitations and opportunities that emerge when working with a mechanical apparatus.

1. INTRODUCTION

Research in robotic musicianship opens the door for new and interesting methods of creating and performing computer music. The combination of software and mechanics provides unique opportunity and limitation for developers, composers, and performers. Robotic musicians greatly differ from the pure software based computer music applications which bestow an unlimited library of timbres, floating point precision, and extremely low latency. Some of the most distinctive differences prevalent in robotic musicians are their ability to create natural acoustic sound using traditional instruments, their mechanical limitations (encompassing everything from the rate of movement to the mere fact that their extremities occupy physical space), and perhaps most importantly their ability to provide visual feedback through a much more natural and engaging experience for an interacting human musician.

Robotic musicianship research has been a major focus at the Georgia Tech Center for Music Technology (GTCMT), most notably with the development of Haile [8] (a robotic percussionist) and Shimon [9] (a robotic marimba player). In the last year GTCMT researched multiple approaches to musical human-robotic interaction. This work culminated in compositions each building from these different approaches.

This paper describes three interactive compositions written for Shimon and a human performer. With each piece we address underlying facets of robotic musicianship including compensating for mechanical latencies, machine listening and generative note algorithms, computer vision, robotic head movement, and visual cues. Each composition uniquely demonstrates different approaches to creating music in the context of human-robotic interaction. These approaches involve various aspects of machine listening and computer vision methods, including methods such as pattern recognition, tempo detection, improvisation, and gesture recognition. It is important to note that when referring to Shimon we refer to the amalgamation of software, physical mechanism, and any additional vision system.

2. COMPOSING FOR SHIMON

There are a few mechanical idiosyncrasies inherent to Shimon that make composing and developing both interesting and challenging. These characteristics influence both the creative and scientific processes of working with Shimon.

2.1. Shimon’s Arms

There is a fixed 500ms delay from the time the message is sent to the time the arm moves into position and the mallet strikes the corresponding key. However, because messages are sent to individual arms it is very possible that a message is sent to a specific arm to move to a certain location there may be another arm occupying the space at that location. When this occurs there are four options: play the note at another available octave, do not play the note at all, move the arm that is currently occupying the space, or play the note using the arm currently in its location. The third and fourth options require the programmer to anticipate the location of each arm and move them accordingly so that the arms locations do not conflict and no notes are dropped. We have adopted the term “path planning” to describe this process.

For many of our applications we use a general purpose path planning algorithm which first selects the arm closest to the note to play and then if for any reason cannot play the note (most likely because it is in too close to the vicinity of an adjacent arm) it attempts to play the note in an alternative octave. Finally if none of these octaves exist the note is dropped. This general purpose path planning algorithm is sufficient most of our applications. However, in some cases (when it is essential to the music for Shimon to play specific motifs exactly as written) a more sophisticated predetermined path planning is necessary.

2.2. Head Movement

In addition to path planning, programming the movement of Shimon’s head is another feature unique to robotic musicianship. Hoffman and Weinberg [4] demonstrated the advantage of visual cues such as arm movement and mallet strikes. Through empirical study, we found that well programmed head movements can provide even more valuable visual cues to an accompanying musician. These related visual cues include gestures varying from Shimon looking towards a person versus its own arms, positioning its head high above the marimba versus very close, and nodding to the beat versus more generic breathing gestures. These cues provide an environment more similar to human-human interaction as opposed to human-software interaction. Shimon develops a defined personality and any interaction becomes social and naturally engaging. In our compositions a combination previously developed higher-level head gestures (such as head banging to the beat) are used in coalescence with lower level commands (such as head and neck rotation).

3. BAFANA

Bafana was inspired by street performances of African marimba bands. The piece moves through several phases that use a number of repetitive interleaving rhythmic motifs, played interdependently on two marimbas by Shimon and a human performer. Shimon listens to the human improvisation, detects specific motifs, and responds in a manner that is times predictable and at times surprising. Each phase presents a different situation for the human player as Shimon is programmed to respond differently. The piece begins with a playful vision-based interaction between the human and the robot that introduces some of the musical motifs in a social non-rhythmic context. The human player then chooses any of nine possible motifs to start the piece, plays it in a loop, and awaits Shimon’s response. In the current implementation of the piece, Shimon’s personality develops from predictable consonant responses at the beginning, to more surprising chromatic reactions towards the end, challenging the human player to constantly listen and carefully adapt to the evolving nature of the music.

3.1. Part 1 - Vision Based Interaction

In the first section Shimon responds to human arm movements analyzed by the Microsoft Kinect. The Kinect is able to provide positional values describing the depth, height, and width of each arm in relation to the camera. Each arm triggers one of two motifs (a total of four motifs). The horizontal position determines the particular motif to be played. The vertical position influences the tempo and velocity, respectively. Though Shimon may take a more deterministic role, the importance of detecting physical gestures to influence the music still resides. Humans use a plethora of physical movements to convey emotional or descriptive pieces of information. In performance humans often cue downbeats or transitions through exaggerated gestures. In Bofuji, physical communication with Shimon parallels the performance of a conductor, in which the conductor makes movements to elicit musical responses from the musicians. Figure 3 shows a human performer in this state of interaction with Shimon.

3.2. Part 2 - Melodic Interaction

The vision-based interaction serves as a precursor to the more involved melodic interaction, which serves as the principal mode of interaction for the piece. As previously
mentioned Shimon and the human performer influence each other by choosing from a library of nine motifs.

The human performer initiates the transition from vision to melodic based interaction by playing any single motif. Because the motifs are known a priori, Shimon’s motif detection consists of a pattern matching algorithm which overlays the input midi values over the hardcoded values of each motif. A string edit distance threshold describing the number of differences between the actual motif and the input allows for errors. At the time of detection of this first motif the tempo is determined through a measurement of the inter-onset-intervals (IOI) and set for the whole piece. At this point Shimon begins nodding its head at a rate described by the tempo signaling to the human that it is listening and has matched tempo. During this portion of the piece the human lead while Shimon responds to each motif by imitating, repeating, and synchronizing with the human player. After a predefined number of motif changes has occurred Shimon no longer limits performance to playing in sync with the human. In fact, Shimon is programmed to not play in sync, but rather in canon. The repetitive nature of the beginning becomes less apparent as the patterns interlock, creating new and interesting music.

The music created in the first part of the piece is predominantly driven by the human’s decisions with Shimon’s melodic responses coming directly from the motifs chosen by the human. In the remainder of the piece the robot begins to make stochastic decisions and influence the progression of the piece. The algorithm continues to detect the specific motif of the human, but responds based on predetermined weights applied to each of the motifs. This weighting describes the probability of one motif being played with another. For each motif the human plays, Shimon answers with a degree of chance and surprise, possibly playing multiple motifs simultaneously. The progression of the piece is obvious as the interlocking of several motifs becomes more complex.

4. INTERPRETATIONS

Interpretations is a piece for Shimon and drumset. In this piece Shimon makes musical decisions based on a stylistic analysis of the drummer’s playing. Shimon runs a program which intelligently determines whether the drummer is playing in time or out of time. Because the motifs are known a priori, Shimon is programmed to not play in sync with the human. In the remainder of the piece the robot begins to make stochastic decisions and influence the progression of the piece. The algorithm continues to detect the specific motif of the human, but responds based on predetermined weights applied to each of the motifs. This weighting describes the probability of one motif being played with another. For each motif the human plays, Shimon answers with a degree of chance and surprise, possibly playing multiple motifs simultaneously. The progression of the piece is obvious as the interlocking of several motifs becomes more complex.

4.1. Feature Extraction and Style Estimation

An algorithm using rule-based artificial intelligence techniques was developed to estimate tempo and style on a

4.1.1. Tempo Detection

A piezo-electric sensor was put on each drum (bass drum, snare, small tom, and floor tom) with each signal being fed into MaxMSP. Using Puckette’s spectral difference onset detection object [6] an accurate symbolic representation of each onset (representing the specific drum) was established. Shimon, using this symbolic representation with the IOIs a tempo was able to be measured. In order to estimate a tempo the algorithm was trained with a model describing a simple, typical pattern of one measure played by the drummer. As shown in Figure 4 this model describes a general pattern in which the bass drum is most likely to hit on beats 1 and 3 and the snare drum on beats 2 and 4. If this exact pattern is played Shimon calculates the tempo by simply divided the temporal length of the pattern by four (based on a fixed four beats per measure). The model describes another piece of useful information, which is in general the bass and snare alternate a total of four times from one measure to the next. So another method to extract the tempo is to measure the length of time it takes for four alternations to occur. Further analysis is computed using the tempo calculated from the previous measure to create a sixteenth note valued time grid. Each onset must be within a 1.5% threshold of the total time to fit the grid. A weighted ratio of the onsets that fit the grid versus the onsets that did not describe the probability of the tempo. Additional weight is given to the bass drum and snare onsets if they fit the grid and the model. When the ratio falls below a threshold a new tempo is estimated by using the alternations between the bass and snare and a new time grid is created with the value.

Through this process of determining the tempo it is possible to concurrently utilize the degree of deviation from the original model to simultaneously decide whether or not the drummer is playing in time or out of time. Because Shimon always attempts to calculate a tempo this deviation can be considered a confidence value describing how likely the calculated tempo is to the actual tempo. When the confidence falls below a threshold a decision is made that the drummer is now freely improvising without a set tempo. The number of onsets, the percent of onsets that fit as a sixteenth of the calculated tempo, and the number of different drum hits are the features which determine this confidence value.

4.1.2. Pattern Recognition and Discovery

Unlike Bafana which uses pattern recognition to find specific precomposed patterns, a primary focus of this piece is to discover improvised patterns. In interactive free improvisation one musician may repeat a motif or short musical pattern which intelligently determines whether the drummer is playing in time or out of time. In order to estimate a tempo the algorithm was trained with a model describing a simple, typical pattern of one measure played by the drummer. As shown in Figure 4 this model describes a general pattern in which the bass drum is most likely to hit on beats 1 and 3 and the snare drum on beats 2 and 4. If this exact pattern is played Shimon calculates the tempo by simply divided the temporal length of the pattern by four (based on a fixed four beats per measure). The model describes another piece of useful information, which is in general the bass and snare alternate a total of four times from one measure to the next. So another method to extract the tempo is to measure the length of time it takes for four alternations to occur. Further analysis is computed using the tempo calculated from the previous measure to create a sixteenth note valued time grid. Each onset must be within a 1.5% threshold of the total time to fit the grid. A weighted ratio of the onsets that fit the grid versus the onsets that did not describe the probability of the tempo. Additional weight is given to the bass drum and snare onsets if they fit the grid and the model. When the ratio falls below a threshold a new tempo is estimated by using the alternations between the bass and snare and a new time grid is created with the value.

Through this process of determining the tempo it is possible to concurrently utilize the degree of deviation from the original model to simultaneously decide whether or not the drummer is playing in time or out of time. Because Shimon always attempts to calculate a tempo this deviation can be considered a confidence value describing how likely the calculated tempo is to the actual tempo. When the confidence falls below a threshold a decision is made that the drummer is now freely improvising without a set tempo. The number of onsets, the percent of onsets that fit as a sixteenth of the calculated tempo, and the number of different drum hits are the features which determine this confidence value.

### Figure 4
A model describing a typical drum pattern of one measure where F represents the bass drum and B represents the snare drum. This model is used as a foundation to estimate tempo, when and where a measure starts, and how freely a drummer is playing based on a measure of deviation from the model.

4.1.2.1. Pattern Recognition and Discovery

The primary features Shimon is programmed to respond to during the free improvisation is note density, recurring patterns, dynamic, and specific drums hit. As the note density increases Shimon begins to play more dissonant notes. This is done by adding new notes to the melody based on a transition matrix from a first order Markov chain of the melody. When the drummer completely moves away from the model but is still in tempo (meaning each onset fits a sixteenth of the tempo), Shimon generates a precomposed melody and generates notes based purely on the transition matrix. For each sixteenth note in a measure the likelihood that Shimon plays a note is directly correlated with the note density of the drummer.

### 4.2. Musical Response

Shimon’s response to the drummer’s play is dependent on the results of the style estimation.

#### 4.2.1. In Time

Shimon uses notes from a precomposed melody, but how Shimon chooses to use the notes varies depending on the drummer’s play. The closer the drummer’s pattern follows the model the more likely Shimon imitates the pattern. When the drummer is playing a pattern which is known a priori, Shimon imitates the precomposed melody. As the drummer begins to or- nament the pattern with additional onsets Shimon also or- naments the melody. This is done by adding notes to the melody based on a transition matrix from a first order Markov chain of the melody. When the drummer completely moves away from the model but is still in tempo (meaning each onset fits a sixteenth of the tempo), Shimon generates a precomposed melody and generates notes based purely on the transition matrix. For each sixteenth note in a measure the likelihood that Shimon plays a note is directly correlated with the note density of the drummer.

#### 4.2.2. Out of Time and Improvisation

The primary features Shimon is programmed to respond to during the free improvisation is note density, recurring patterns, dynamic, and specific drums hit. As the note density increases Shimon begins to play more dissonant notes. This is done by adding new notes to the melody based on a transition matrix from a first order Markov chain of the melody. When the drummer completely moves away from the model but is still in tempo (meaning each onset fits a sixteenth of the tempo), Shimon generates a precomposed melody and generates notes based purely on the transition matrix. For each sixteenth note in a measure the likelihood that Shimon plays a note is directly correlated with the note density of the drummer.
mentioned Shimon and the human performer influence each other by choosing from a library of nine motifs.

The human player initiates the transition from vision to melodic based interaction by playing any single motif. Because the motifs are known in advance, Shimon’s motif detection consists of a pattern matching algorithm which overlays the input midi values over the hardcoded values of each motif. A string edit distance threshold describes the number of differences between the actual motif and the input allows for errors. At the time of detection of this first motif the tempo is determined through a measurement of the inter-onset-intervals (IOI) and set for the whole piece. At this point Shimon begins nodding its head at a rate described by the tempo signaling to the human that it is listening and has matched tempo. During this portion of the piece the human performs while Shimon responds to each motif by imitating, repeating, and synchronizing with the human player. After a predefined number of motif changes has occurred Shimon no longer continues performance to playing in sync with the human. In fact, Shimon is programmed to not play in sync, but rather in canon. The repetitive nature of the beginning becomes less apparent as the patterns interlock, creating new and interesting interactions.

The music created in the first part of the piece is predominantly driven by the human’s decisions with Shimon’s melodic responses coming directly from the motifs chosen by the human. In the remainder of the piece the robot begins to make stochastic decisions and influence the progression of the piece. The algorithm continues to detect the specific motif of the human, but responds based on predetermined weights applied to each of the motifs. This weighting describes the probability of one motif being played with another. For each motif the human plays, Shimon answers with a degree of chance, possibly playing multiple motifs simultaneously. The progression of the piece is obvious as the interlocking of several motifs becomes more complex.

4. INTERPRETATIONS

Interpretations is a piece for Shimon and drumset. In this piece Shimon makes musical decisions based on a stylistic analysis of the drummer’s playing. Shimon runs a program which intelligently determines whether the drummer is playing in a defined tempo or is improvising more freely. Then based on this decision responds in different ways. The pattern matching and response theme of Bafana largely inspired the writing of this piece. Interpretations similarly explores pattern recognition and matching algorithms, but in the context of real-time improvised performance. Many of the algorithms in this piece are inspired by the writing of this piece.

4.1. Feature Extraction and Style Estimation

An algorithm using rule-based artificial intelligence techniques was developed to estimate tempo and style on a measure to measure basis. This method helped to calculate several parameters depicting the drummer’s play. The first consisted of a boolean decision determining whether the drummer’s play was to be “in time” or “out of time.”

4.1.1. Tempo Detection

A piezo-electric sensor was put on each drum (bass drum, snare, small tom, and floor tom) with each signal being fed into MaxMSP. Using Puckette’s spectral difference onset detection object [6] an accurate symbolic representation of each onset (representing the specific drum) was established. Then, using this symbolic representation with the IOIs a tempo was able to be measured.

In order to estimate a tempo the algorithm was trained with a model describing a simple, typical pattern of one measure played by the drummer. As shown in Figure 4 this model describes a general pattern in which the bass drum is most likely to hit on beats 1 and 3 and the snare drum on beats 2 and 4. If this exact pattern is played by Shimon calculates the tempo by simply dividing the temporal length of the pattern by four (based on a fixed four beats per measure). The model describes another piece of useful information, which is in general the bass and snare alternate a total of four times from one measure to the next. So another method to extract the tempo is to measure the length of time it takes for four alternations to occur.

Further analysis is computed using the tempo calculated from the previous measure to create a sixteenth note valued tatum grid. Each onset must be within a 1.5% threshold of the tatum to fit the grid. A weighted ratio of the onsets that fit the grid versus the onsets that did not describe the probability of the tempo. Additional weight is given to the bass drum and snare onsets if they fit both the grid and the model. When the ratio falls below a threshold a new tempo is estimated by using the alternations between the bass and snare and a new tatum grid is created with the value.

Through this process of determining the tempo it is possible to concurrently utilize the degree of deviation from the original model to simultaneously decide whether or not the drummer is playing in time or out of time. Because Shimon always attempts to calculate a tempo this deviation can be considered a confidence value describing how likely the calculated tempo is to the actual tempo. When the confidence falls below a threshold a decision is made that the drummer is now freely improvising without a set tempo. The number of onsets, the percent of onsets that fit as a sixteenth tatum of the calculated tempo, and the number of different drums hit are the features which determine this confidence value.

4.1.2. Pattern Recognition and Discovery

Unlike Bafana which uses pattern recognition to find specific, precomposed patterns, a primary focus of this piece is to discover improvised patterns. In interactive free improvisation one musician may repeat a motif or short musical pattern which biện a tempo pattern of one measure where F represents the bass drum and R represents the snare drum. This model is used as a foundation to estimate tempo, when and where a measure starts, and how freely a drummer is playing based on a measure of deviation from the model.

4.2. Musical Response

Shimon’s response to the drummer’s play is dependent on the results of the style estimation.

4.2.1. In Time

Shimon uses notes from a precomposed melody, but how Shimon chooses to use the notes varies depending on the drummer’s play. The closer the drummer’s pattern follows the more the robot will interpret Shimon’s hit. As the note density of the precomposed melody and generates notes based purely on the transition matrix. For each sixteenth note in a measure the likelihood that Shimon play a note is directly correlated with the note density of the drummer.

4.2.2. Out of Time and Improvisation

The primary features Shimon is programmed to respond to during the free improvisation is note density, recurring patterns, dynamic, and specific drums hit. As the note density increases Shimon begins to play more dissonant notes. This is done by transforming the note chosen by the Markov chain such as shifting the note a half step or augmented 4th. A ratio describing the percentages of each different drum hit is continuously measured for the last 40 onsets. This ratio describes the variety of drums being hit and as this variety increases Shimon will continue to make note transformations by transforming the octave the note is played so that at the drummer’s peak variety each note Shimon plays is at a distance of at least 12 half steps from the previous note. The drummer’s dynamic is mapped to the probability of Shimon notes in unison. Using the pattern discovery algorithm Shimon continuously looks for drum patterns of significance. When one pattern looks for continguously repeated patterns. To do this a simple string matching function is used. The function works by taking a string of length n, let’s call it StringA, and concatenating it with itself, creating StringB with length 2n. StringA + StringA = StringB

4.3. Interpretations

StringA = StringB, let’s call it StringA′

StringA′ = StringA + StringB

This StringA′ is compared to a concatenation of StringA and the next contiguous string of length n, which we can call StringB. If StringA′ is equal to StringB then a contiguously repeated pattern exists. Here, n must be greater than three and at least two different drums must be represented.
is found Shimon creates another competing pattern derived of notes calculated from the most current feature values. Similarly to the stochastic processes in Bafana Shimon may create and repeat a pattern of his own and the drummer can choose to respond to how he sees musically fit.

4.3. Head Movement

Shimon is programmed to visually communicate social processes through the use of head movements. These visual cues help the human performer to better understand how Shimon is interpreting the drum playing. Shimon looks toward the drummer when either a new tempo is detected or it has a low confidence for the current calculated tempos. This not only gives the perception of Shimon “listening,” but also informs the drummer of Shimon’s confidence level in the calculated tempo. During sections when Shimon is improvising the neck moves on its vertical axis so that Shimon looks closer towards the marimba to simulate concentration. Depending on the level of improvisation (creating patterns during the out of time section versus merely coloring the melody during the in time section) Shimon will adjust the neck level informing the drummer of the current state Shimon has detected.

5. GESTURE DETECTION

The percussionist trains the system before the performance with the three gestures he or she would like to use to send messages to Shimon. To train the system the musician simply creates a gesture (in this case moving a mallet in a particular way) and the Wii data is recorded and used as the training data for the system. Figure 7 illustrates some examples of gestures that were used for the piece. The data represents the 2-dimensional position of the n points defining the gesture.

The percussionist trains the system before the performance with the three gestures he or she would like to use to send messages to Shimon. To train the system the musician simply creates a gesture (in this case moving a mallet in a particular way) and the Wii data is recorded and used as the training data for the system. Figure 7 illustrates some examples of gestures that were used for the piece. The data represents the 2-dimensional position of the n points defining the gesture. After the beginning of the gesture is recognized, the final section of 8 beats is scheduled to start after the current session of 8 beats.

5.1. Training the System

The first gesture triggers Shimon to play a section. Before the beginning of the song, the motif is played as soon as the gesture is recognized. After the beginning of the song, once the gesture is detected, the motif is scheduled to play in the next section of 8 beats. A second gesture controls how Shimon improvises during the section. Shimon is using a simple improvisational method in which it only uses notes from the major or minor scale degrees of the current chord in the progression. The gesture will trigger either a low or high note density. The third gesture controls when to end the piece. Once this gesture is recognized, the final section of 8 beats is scheduled to start after the current session of 8 beats.

6. DISCUSSION AND CONCLUSION

Using Shimon we attempt to demonstrate the capability of a robotic system to adequately perform written scores, improvise, and respond to human performers all while enriching the music through added levels of sophistication (such as Shimon playing complex musical lines while concurrently analyzing a human’s play). Several other robotic systems such as the Logos Foundation robots, LEMUR bots, or the KarmetiK Machine Orchestra have already demonstrated this. Aside from the musical response and unlike the other robotic systems much of the interactive focus is spent programming Shimon to visually communicate musical or software processes in a manner much more rich than purely solenoids striking a drum or piano keys moving up and down. Through our own interaction with Shimon and working with other musicians who have performed our compositions we have learned that such visual cues and naturally engaging social interaction is essential to successful human-robotic performance and as shown by [5] is important to audience appreciation as well. Visual cues play a significant role in human-human interactive performance and through our compositions we attempt to demonstrate that some of these interactions can be successfully translated to human-robotic performance.

Each composition tackles a unique mode of interaction including aspects of auditory and visual communication. Though musicians enjoy the experience of working with Shimon in these specific interactive scenarios they often long for additional ways to communicate. For example, performers of Bafana often ask if Shimon will respond to tempo changes or a more improvisatory style of play. Though Bafana was written to be performed in a certain manner, requests of this nature indicate that a truly successful robotic musician should be able to simultaneously make use of all of these interactive modes such as tempo detection or gesture recognition much like a human does. As we continue to compose and develop for Shimon we will work towards a system that simultaneously embodies many of these interactive features.

REFERENCES


is found, Shimon creates another competing pattern derived from notes calculated from the most current feature values. Similarly to the stochastic processes in Bafana, Shimon may create and repeat a pattern of his own and the percussionist can choose to respond to how he sees musically fit.

4.3. Head Movement

Shimon is programmed to visually communicate software processes through the use of head movements. These visual cues help the human performer to better understand how Shimon is interpreting the drum playing. Shimon looks toward the drummer when either a new tempo is detected or it has a low confidence for the current calculated tempo. This not only gives the human a sense of Shimon “listening,” but also informs the drummer of Shimon’s confidence level in the calculated tempo. During sections when Shimon is improvising the neck moves on its vertical axis so that Shimon looks closer towards the marimba to simulate concentration. Depending on the level of improvisation (creating patterns during the out of time section versus merely coloring the melody during the in time section) Shimon will adjust the neck level informing the drummer of the current state Shimon has detected.

5. GESTURE DETECTION

The percussionist trains the system before the performance with the three gestures he or she would like to use to send messages to Shimon. To train the system the musician simply creates a gesture in this case moving a mallet in a particular way and the Wii data is recorded and used as the training data for the system. Figure 7 illustrates some examples of gestures that were used for the piece. The data represents the 2-dimensional position of the n points defining the gesture.

The first used the gesture follower technology developed at IRCAM [1, 2]. Even though the gesture follower is provided as a MaxMSP external, we decided to implement our own version of the code in order to better understand the algorithm. The implementation was made in Objective-C, using the GHMM (General Hidden Markov Model) library [7]. The implementation of curve recognition using an HMM works well, but we feel it is not robust enough against false positives. Instead we use a simpler method that produces better results. Our curve matching algorithm consists of measuring the Euclidean distance of the last n points detected by the Wii with the points of the trained gesture, where n is the length of the gesture. If the total distance is within a certain threshold then the last n points is considered as one of the three gestures and the message is sent to Shimon.

The percussionist creates and repeats a pattern of his own and the percussionist can choose to respond to how he sees musically fit.

5.1. Training the System

5.2. Detecting Gestures and Performing

The gestures are detected in real-time during the performance, so the percussionist is able to communicate with the robot while playing. The three gestures for this composition, the number of recognizable gestures supported by Shimon is virtually unlimited.

The first gesture triggers Shimon to play a section. Before the beginning of the song, the motif is played as soon as the gesture is recognized. After the beginning of the song, once the gesture is detected, the motif is scheduled to play in the next session of 8 beats. A second gesture controls how Shimon improvises during the section. Shimon is using a simple improvisational method in which it only uses notes from the major or minor scale degrees of the current chord in the progression. The gesture will trigger either a low or high note density. The third gesture controls when to end the piece. Once this gesture is recognized, the final section of 8 beats is scheduled to start after the current session of 8 beats.

6. DISCUSSION AND CONCLUSION

Using Shimon we attempt to demonstrate the capability of a robotic system to adequately perform written scores, improvise, and respond to human performers all while enriching the music through added levels of sophistication (such as Shimon playing complex musical lines while concurrently analyzing a human’s play). Several other robotic systems such as the Logos Foundation robots, LEMUR bots, and the KarmetiK Machine Orchestra have already demonstrated this. Aside from the musical response and unlike the other robotic systems much of the interactive focus is spent programming Shimon to visually communicate musical or software processes in a manner much more rich than purely solenoids striking a drum or piano keys moving up and down. Through our own interaction with Shimon and working with other musicians who have performed our compositions we have learned that such visual cues and naturally engaging social interaction is essential to successful human-robotic performance and as shown by [5] is important to audience appreciation as well. Visual cues play a significant role in human-human interactive performance and through our compositions we attempt to demonstrate that some of these interactions can be successfully translated to human-robotic performance.

Each composition tackles a unique mode of interaction including aspects of auditory and visual communication. Though musicians enjoy the experience of working with Shimon in these specific interactive scenarios they often long for additional ways to communicate. For example, performers of Bafana often improvise a more improvisatory style of play. Though Bafana was written to be performed in a certain manner, requests of these nature indicate that a truly successful robotic-musician should be able to simultaneously make use of all of these interactive modes such as tempo detection or gesture recognition much like a human does. As we continue to compose and develop for Shimon we will work towards a system that simultaneously embodies many of these interactive features.

REFERENCES


