# The Effects of Latency on Ensemble Performance

by
Nathan Schuett
May, 2002

Advisor:
Chris Chafe

Research sponsored by
National Science Foundation

CCRMA
DEPARTMENT OF MUSIC
Stanford University
Stanford, California 94305

THE EFFECTS OF LATENCY ON ENSEMBLE PERFORMANCE

Nathan Schuett
Stanford University, 2002

# Abstract

Methods for audio transport over networks have developed to a point where experiments with professional-quality musical collaboration are now possible. Low-latency, low-jitter, next-generation networks have been tested in a variety of musical scenarios including performers playing together across continental distances. The latency problem inherent in long-haul network paths is well known but is less well understood in terms of its effect on real-time musical collaboration. We have begun a series of experiments testing the effect of latency on ensemble performance. The performances will be evaluated based on their tempo direction, average beat duration, and standard deviation of their beat durations.

The goal is to define the Ensemble Performance Threshold (EPT), or the level of delay at which effective real-time musical collaboration shifts from possible to impossible. Our motivation is the need for a "latency design spec" (in msec) for engineering new systems that support truly natural feeling audio collaboration environments. This study served as a pilot study to investigate whether an EPT exists and to determine the general effects of latency on musical performance.

Conclusions were as follows: (1) The direction of the tempo was a very useful indicator of whether a performance was being hindered by the effects of latency. If the delay was greater than 30 msec, the tempo would begin to slow down. This gives a solid indication that EPT for impulsive, rhythmic music lies between 20-30 msec. (2) A coping strategy was discovered that allowed the performers to maintain a solid tempo up to 50-70 msec of delay. The strategy can be quickly summarized as a leader - follower relationship. Unfortunately, this strategy results in a severe decrease of synchrony on the leader's end. (3) It is most likely that EPT varies depending on the type of music (speed, style, attack times of instruments, etc). (4) When delay is between 10-20 msec each way, it may be providing a stabilizing effect on the tempo. 10-20 msec of delay may be better for ensemble performance than 0 msec of delay. (5) The EPT determined in the electronic delay tests was much lower than the EPT estimated in the outdoor delay tests. This is predicted to be due to the lack of auditory cues in the electronic tests such as reverb and variable amplitude which were present in the outdoor tests.

# Table of Contents

# Chapter I
# **Introduction**

## 1.1. Summary

High-bandwidth audio streaming in real-time has recently become feasible. This is a result of the emergence of low-latency, low-jitter, next-generation networks. Collaborative musical performance is now possible over the Internet with professional-quality sound. The latency problem inherent in long-haul network paths is well known, but is less well understood in terms of its effect on real-time musical collaboration.

This study served as a pilot study to investigate the effects of latency on collaborative performance.

## 1.2. Internet Performance

Latency between New York and California measured over the Abilene network (Internet2's next-generation test bed) is 33 msec (which is within a factor of 2 of the speed of light). Network jitter has been measured on the order of 4%. Sound travels at around 345m/s, so the equivalent distance acoustically is around 12m, well within the dimensions of a large concert stage. Since musicians can effectively perform together at this distance, it is hypothesized that they should also be able to perform together if their signals are delayed electronically by a similar amount. The assumption that electronic latency correlates with a physical distance between players was tested in a preliminary study at Stanford involving two drummers outside. This preliminary study will be described in more detail in Section 2.1.4.

## 1.3. Thesis Scope

1. Determine and document the effects of delay on two-way musical performance.

2. Attempt to isolate a critical delay "comfort level" for playing rhythmic music under delay constraints. This will be called the Ensemble Performance Threshold (EPT).

3. Identify any other differences that distinguish a "telejam" from an ensemble performance in the same acoustic space.

4. In order to answer the above three questions, a quantitative method must be developed for analyzing ensemble performance. The method must be accurate and repeatable.

# Chapter II
# **Historical Review of Relevant Research**

## 2.1. Latency

There is a curious lack of research that deals with delay and its effects on music. Perhaps this is because of the hard to quantify characteristic of music and the lack of technology capable of performing such analyses.

Dave Phillips, who maintains the *Linux Music & Sound Applications Website*, writes that, "studies have indicated that the ear is sensitive to timing differences at the millisecond level, perhaps even down to a single millisecond." He also claims that latencies under 7 msec are not typically perceptible and should be considered acceptable for desktop and semiprofessional audio applications [1].

### 2.1.1. Effects on Telephone Conversation

The general consensus of much study on voice transmission is that one-way delay of less than 100 milliseconds (msec) is imperceptible to most users. Delays in the range of 100 – 300 msec are considered to be noticeable, but tolerable. Latencies greater than 300 msec are not tolerable, as they result in a speak-and-wait conversation [2].

### 2.1.2. Effects on Ensemble Performance

Jeremy Cooperstock, from McGill University claims that there are two EPTs for ensemble performance based on size of the ensemble alone. He claims that based on research studies, large ensembles can tolerate up to 40 msec of latency while small ensembles can tolerate only up to 5 msec [3].

#### Large Ensemble (8 or more players)

Cooperstock points to two studies by Rasch which showed that a typical delay between the first and the last attack between performers who are playing a single note was approximately 40 msec for large ensembles [4]. For example, if a symphony orchestra were to play a single note simultaneously, the time between the earliest musician's and the latest musician's entries on that note will be approximately 40 msec. This is not surprising, since many stages have dimensions as large as 40 ft - the distance traveled by a sound wave in air in 40 msec. Cooperstock then concludes that a two-way latency of up to 40 msec is an acceptable maximum delay for large ensemble performance.

Small Ensemble

Cooperstock's estimate of a 5 msec EPT for small ensembles is based largely on practical experience. Looking at the evolved seating arrangements of string quartets and trios, it is easy to see that the players try to sit very close to one another. He claims that when musicians in such a group are separated by more than roughly 2 m, difficulties in the ensemble are incurred [3]. Thus, he sets the EPT at 5 msec.

### 2.1.3. Electronically Manipulated Delay Experiments

The first such experiment involved placing two trumpeters in separate rooms. The experiments were run at the Banff Center for the Arts. Microphones and headphones allowed the players to hear each other. A TCP-based, 1-channel, bi-directional application was tuned to provide delays of about 200 msec. The musicians were initially mystified by trying to perform in such a situation (especially with no visual cue for starting together). It only became possible to avoid recursive tempo slowing when one player agreed to play behind the other [6].

There were no tools available to analyze the above experiment. It was evaluated by the ears of trained musicians.

### 2.1.4. Spatially Manipulated Delay Experiments

Our experiments to investigate whether electronic latency correlates with physical distance began with outdoor recordings using two drummers separated by increasing distances. They played a set of examples of graduated rhythmic complexity, and delay was added between the performers by increasing the physical distance between them. The players were facing away from each other so as to avoid visual cues. One side effect in the spatially manipulated environment is the decrease in amplitude, on the order of 6 dB every time the distance from the source is doubled [7].

A critical latency threshold was found when the players were 100 ft apart. It takes sound approximately 100 msec to travel this distance. Surprisingly, breakdown of their ensemble playing was as easily revealed when keeping simple time as when playing a duo of highly syncopated music. When positioned closer than about 33 m. (ca. 100 msec. delay time), the players synchronized well. As they played farther apart their "rhythmic flywheels" appeared to have difficulty phase locking. Then, at a point far enough beyond the critical threshold, they locked in a mode one-half cycle out of phase. The unstable region between the locked regions contains "chasing" where they seemed to be hopping between different modes.

## 2.2. Tempo

It is predicted that the tempo of a performance will be the most affected by delay. Tempo is commonly referred to as the speed of a composition. Before the introduction of the metronome, tempos were suggested rather broadly by a collection of terms such as *Andante*, *Allegro*, and *Presto*. With the aid of a metronome though, composers could define a more specific tempo in beats per minute.

A beat is a term that defines a steady recurring pulse. A single beat consists of a point in time (an event) and a duration to the next point in time. A fast tempo has short beats, and thus several beats per minute, while a slow tempo has long beats.

When a tempo is measured in beats per minute, the actual desired length of each individual beat can be calculated. For example, a piece of 120 bpm would have an ideal single beat duration of 0.5 sec. If the tempo were perfectly rigid, there would be 0.5 sec between the onset of one event and the onset of the next event.

Tempo can change over time by speeding up or slowing down, or it may remain constant. A perfectly constant tempo is impossible without the use of a metronome or computer, however experienced musicians can maintain a tempo that is very strict.

### 2.2.1. Tempo Studies

A study by Terry Kuhn and Edith Gates entitled "*Discrimination of Modulated Beat Tempo by Professional Musicians*" showed professional musicians could identify tempo slow downs more accurately and sooner than tempo increases. Thus, tempo decreases are more easily perceivable than tempo increases [8].

For the purpose of ensemble performance, tempo slow downs may cause a greater problem for the performers, as the decrease in tempo will quickly be perceived as abnormal.

A follow-up study by Kuhn and Gates entitled "*Effect of Notational Values, Age, and Example Length on Tempo Performance Accuracy*" tested to see whether perception of tempo correlates with performance tempo. Kuhn theorized that tempo would tend to slide in the direction that was least perceptible. This turned out to be the case, as subjects evidenced a tendency to increase tempo during a clapped performance [9].

In an ensemble performance then, it is predicted that the tempo of the piece should be slightly increasing.

#### 2.2.2. Tempo During a Performance

No performer can maintain a perfectly strict tempo without the aid of a metronome. Tempos are bound to fluctuate a little over the course of a piece. In fact, players normally alter the tempo of a performance briefly for effect. They may accelerate or slow down to convey emotion. For the purpose of this study, players were instructed to play as rigidly as possible.

#### 2.2.3. Tempo Evaluation

The method used in this study was to analyze the length of each beat duration over the course of the piece. In a perfectly rigid tempo, each beat duration should be almost identical. With human performers this is impossible. Even if the players maintain a relatively constant tempo throughout the piece, each beat duration is going to fluctuate somewhat. How much fluctuation is too much, though?

Can a measure be developed that will determine if the individual beats are fluctuating too much, thus classifying a tempo as irregular?

### 2.3. Rhythmic Synchronization

Adrian Freed, a researcher at the University of California at Berkeley states that the ear will notice the misplacement of a rhythmic event in a sequence if the event is more than 10 msec out of place [10].

The Weber ratio for regularity discrimination has been shown to be ~2% of the beat. That is, subjects tend to hear rhythmic phrases in which consistent deviations are less than 2% of the beat period as regular. For example, with a beat duration of 250 msec, deviations of 5 msec and less for each beast would still be perceived as regular [11].

# Chapter III
# **A Tool for Tempo Analysis**

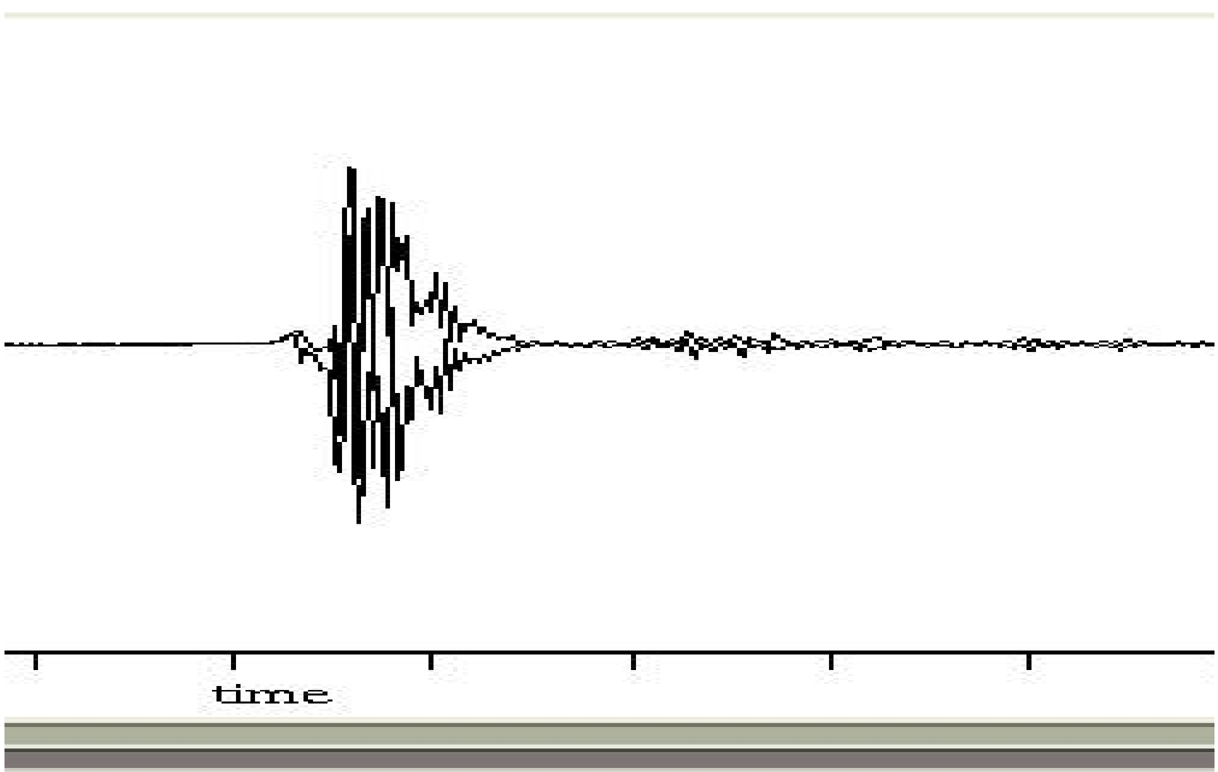
In this study, a system for quantitative analysis of tempo was designed that makes use of automatic event detection. It consists of two separate programs. The first program creates a manageable amplitude envelope file for the original signal. The second program analyzes that envelope, locating events and ultimately determining a tempo.

The complete code for the two programs can be found in Appendices A and B.

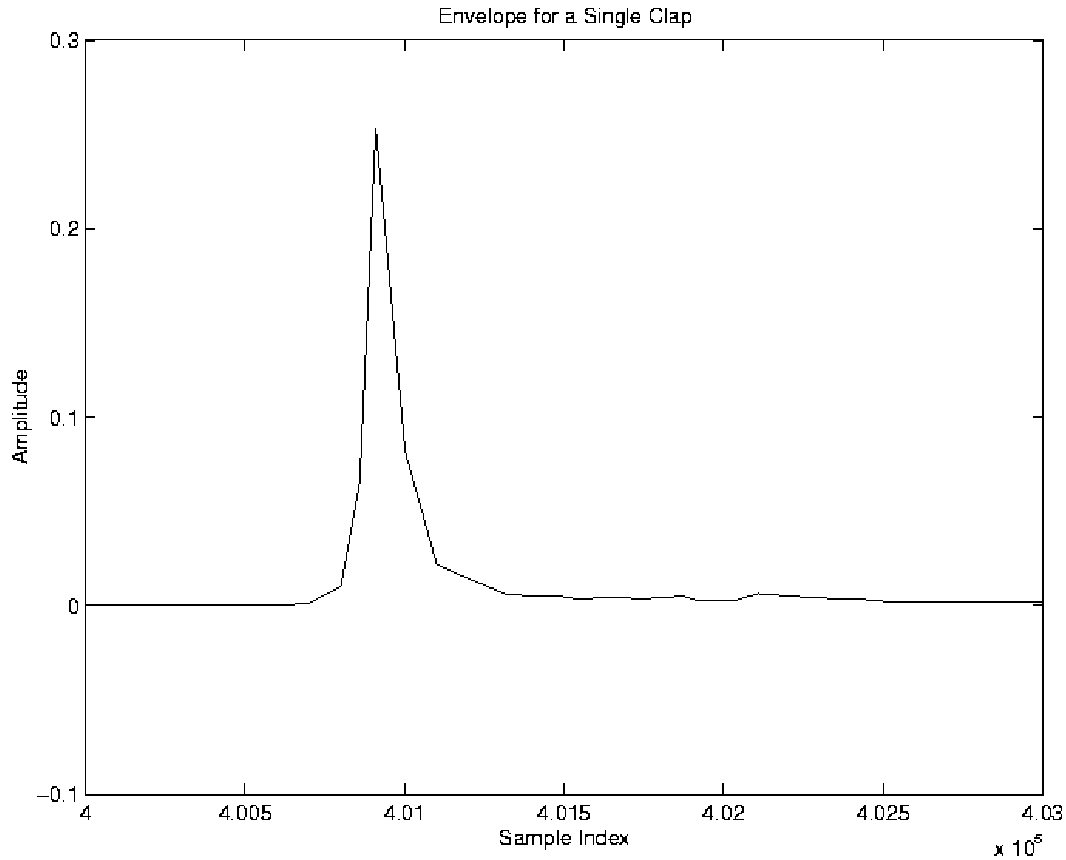
### **3.1. Phase 1: Local Maximums**

The first program was a modification of STK's playN function. It plays through a .wav file and writes the maximum amplitude for every 100 samples (2.27 msec) to a .m file for Matlab. This creates an envelope for the signal, helping to reduce the amount of data from the original signal significantly.

The Original Signal:



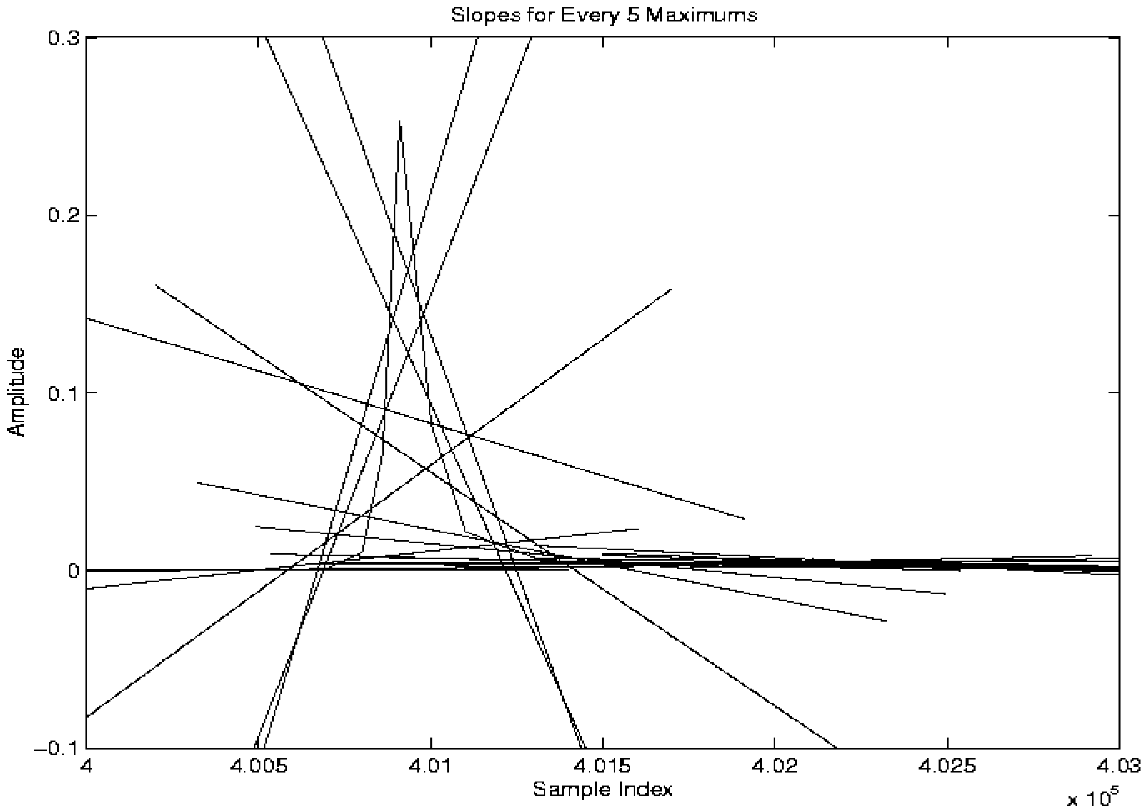
Envelope of the Signal:




### 3.2. Phase 2: The Surfboard Method of Event Detection

The second program then sorts through all of the maximum amplitudes to determine which maximums actually correspond with events from the performance. This program uses an algorithm known as the "surfboard" method from Andrew Schloss' *On the Automatic Transcription of Percussive Music – From Acoustic Signal to High-Level Analysis* [12].

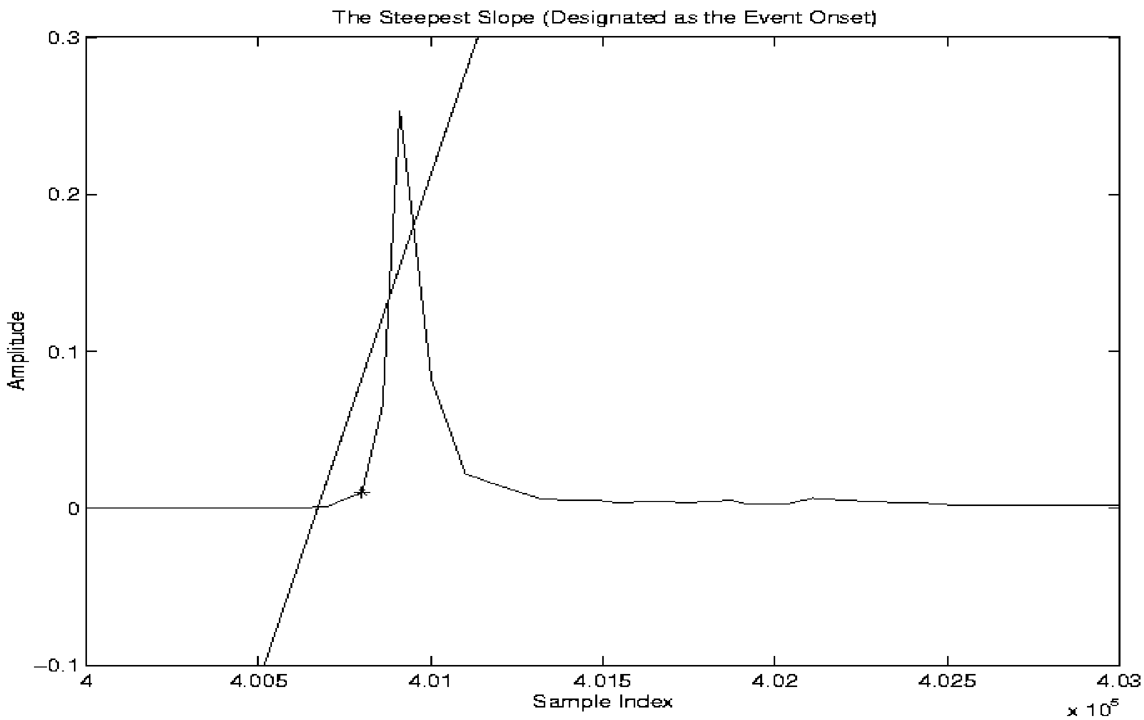
The surfboard method involves calculating several linear regression lines between the stored maximum amplitudes. The regression line moves one point at a time through the maximum amplitude file, while approximating n points at a time (n was 5 in this study). This creates several overlapping line segments that "float" over the data like a surfboard.



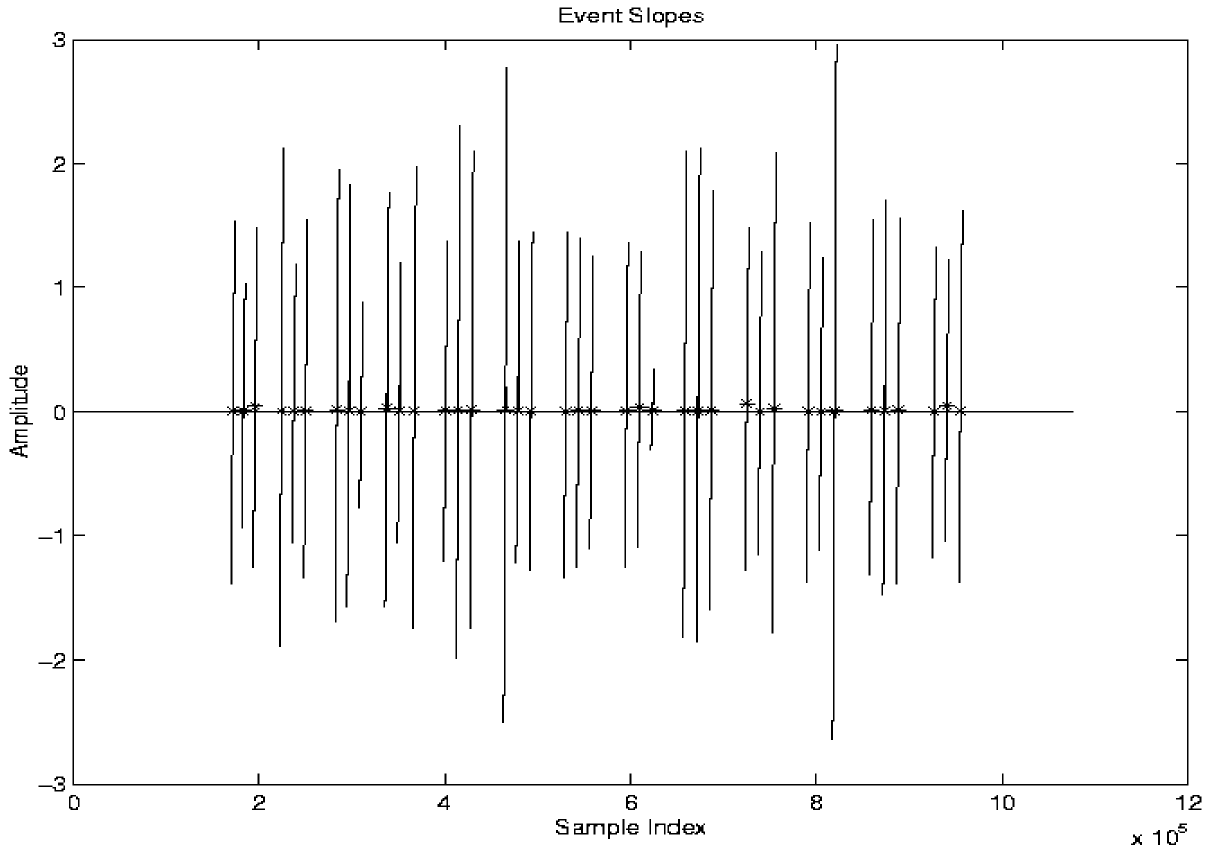

### 3.3. Phase 3: Determining Events

All of the slopes for the entire sound file are stored in an array and the maximum slope is determined. A function iterates through the slope array looking for slopes that are within a certain arbitrary threshold of the maximum slope. This can be adjusted based on the particular recording to get the best results. The event detector used in this study classified events as having a slope within 3% of the maximum slope.

It must be noted though, that not all slopes above the threshold were classified as events. That would result in double or triple counting events. This was avoided by examining the data closer once an above-threshold slope was found. The local area would be searched for any larger slopes, and the largest local slope would be classified as the event. The sample index of the point in the middle of the regression line is then recorded as the note onset.




The graph below shows the location of all the events in a particular sound file.




3.3.1. Error of the Event Detector

Maximum amplitudes are selected for every 100 samples. Theoretically, this window could shift location by a maximum of 100 samples each way from the maximum. This means the event could be pinpointed anywhere within those 200 samples. Converted to milliseconds, that puts the maximum error around 4.5 msec.

(44100 samples / sec) = (44.1 samples / msec)

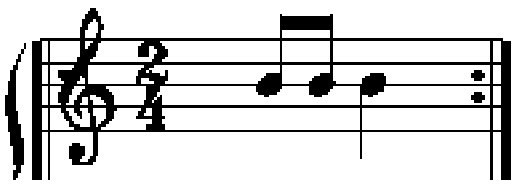
200 samples / 44.1 msec = 4.5 msec

**3.4. Phase 4: Tracking the Tempo**

Once all of the events have been classified, their sample indices are stored in an array. This array now contains all of the note onsets. Simple conversion can change sample indices into milliseconds. Then, the duration between note onsets, or the beat length, can be calculated.

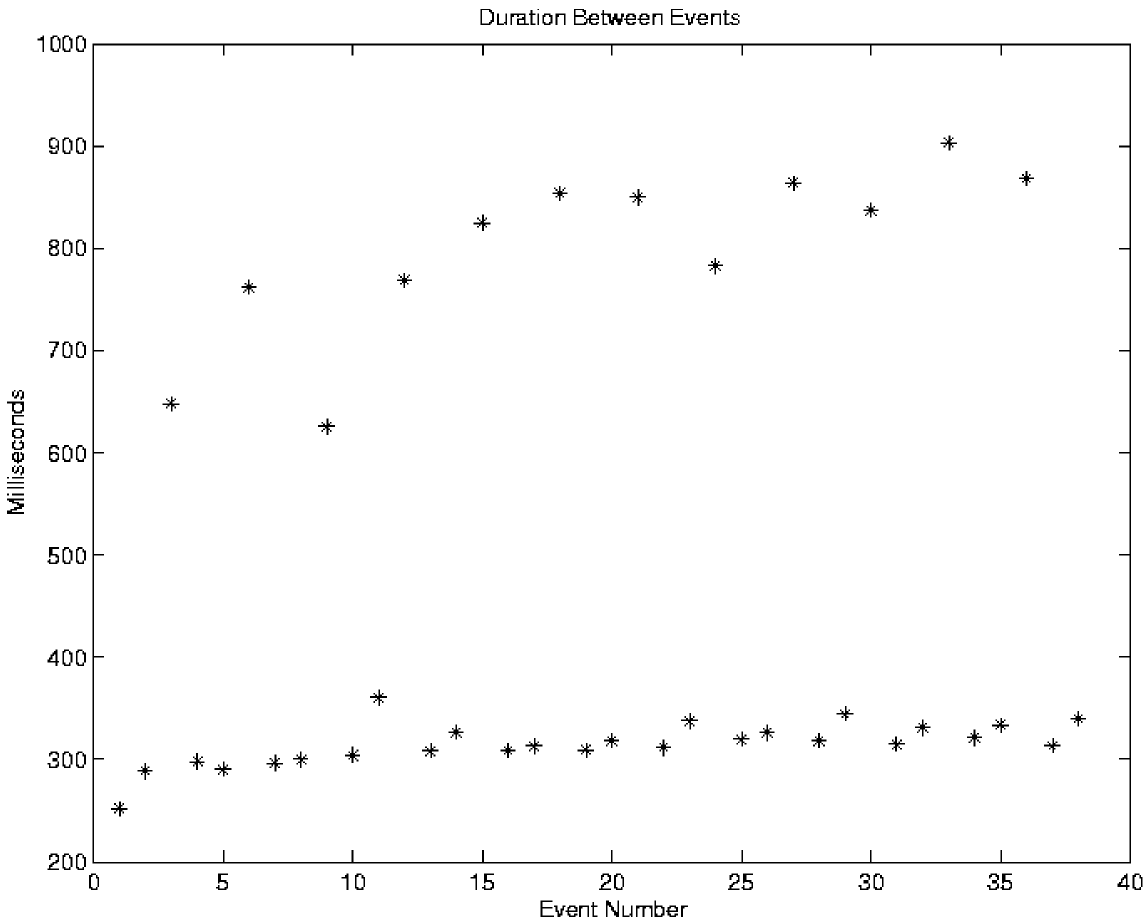
With the relatively simple rhythms of these experiments, it was adequate to make the smallest durations of the sound file the ‘beat’ of the piece in a beats per minute calculation.

For example, one of the rhythms used in the experiments was a repeating pattern of eighth note - eighth note - quarter note.



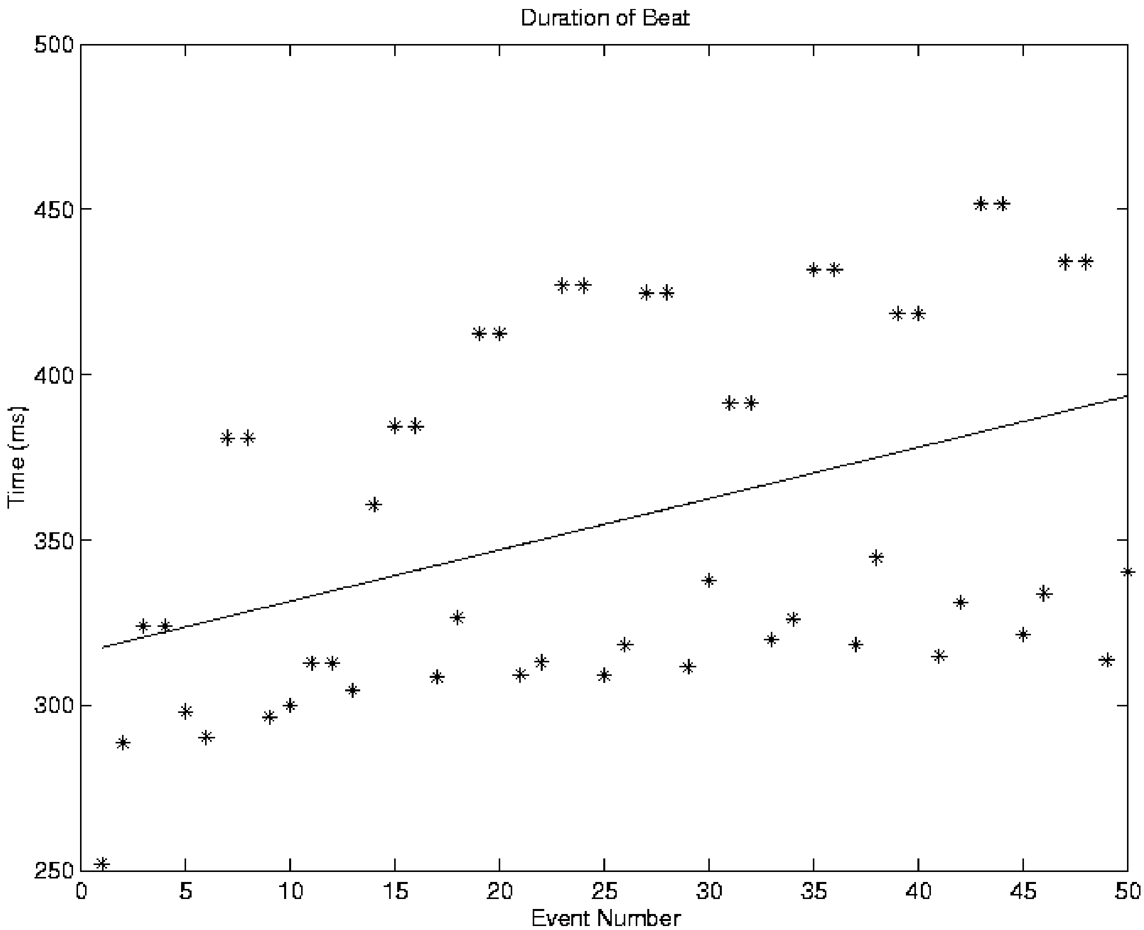
In this case, the eighth note would serve as the beat.  The quarter note, would then need to be divided by two to give an equivalent duration.

The graph below shows the beat durations for a recording of the above rhythm:




The durations above 600 msec are the durations for the quarter notes in the pattern. The durations around 300 msec are the durations for the eighth notes. Notice that there are two eighth note durations for every quarter note duration, following the rhythm that is displayed above.

In order to make the data manageable, the long beats must be represented on the same time scale as the short beats. The quarter note to eighth note ratio is 2:1, so the long beats must be divided by two. The figure below shows all the beat durations on the same scale. Notice that the long beats are represented by two stars. This actually indicates two beats, and the duration will be counted double in any calculations.




### 3.5. Phase 5: Analyzing the Data

After collecting all the beat durations, the tempo analyzer provides three measures for each sound file:

1. average beat duration (msec) of the performance
2. standard deviation of the beat durations (how rigid is the tempo)
3. slope of the beat durations (is the tempo slowing down or speeding up?)

Looking at the figure above, the regression line through the beat durations indicates the direction of the tempo. There is a positive slope to the regression line, indicating that the performance is slowing down.

The three outputs above will be used in combination to measure the effectiveness of the performances. It is hoped that the three measures supplied by the tempo analyzer will be sufficient in quantitatively evaluating the tempo of an ensemble performance.

# Chapter IV
# **Empirical Research**

Again, to frame the goals:

1. This is a pilot study to investigate the effects of latency on collaborative performance.
2. It will attempt to isolate and define an Ensemble Performance Threshold (EPT)
3. And finally, it will identify differences between normal ensemble performance (same room) and remote ensemble performance.

## 4.1. Method for Electronically Manipulated Delay Experiments

The experiment devised was intended to simulate performance over the net.

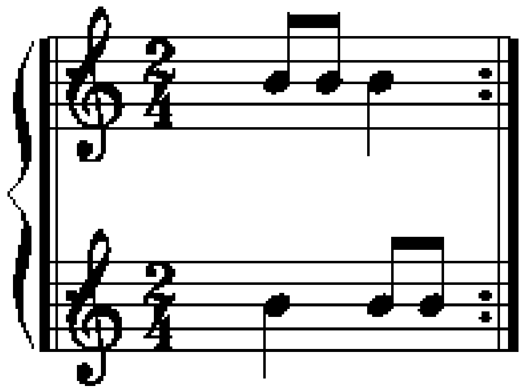
There were two players in separate rooms (isolated both visually and aurally) and latency was artificially added to each of their signals. Each performer could hear his own dry signal, but would hear his partner's delayed signal through headphones. The delay was modulated using a patch within the Mackie Digital Console.

There were 5 different testing scenarios:

| Scenario | Rhythm | Starting Tempo (msec) | Coping Strategy | Metronome for starting pulse | Delay Administered |
|---|---|---|---|---|---|
| 1 | Rhythm 1 | 250 | True Ensemble | Yes | Sequential Choice |
| 2 | Rhythm 1 | 250 | Leader / Follower | Yes | Sequential Choice |
| 3 | Rhythm 1 | 250 | Combination | No | Random, Blind Choice |
| 4 | Rhythm 1 | 400 | Combination | No | Random, Blind Choice |
| 5 | Rhythm 2 | 300 | Combination | No | Sequential Choice |

### 4.2. Scenario 1

Scenario 1 consists of a simple interlocking rhythmic pattern (Rhythm 1):



The above rhythm was clapped by two performers. Performer 1 claps the top rhythm while performer 2 claps the bottom rhythm.

Both voices articulate a single rhythmic motif, but the motif is offset by one quarter note. This insures that there are points of synchronization at each beat, and that there is a limited amount of independence built into each part.

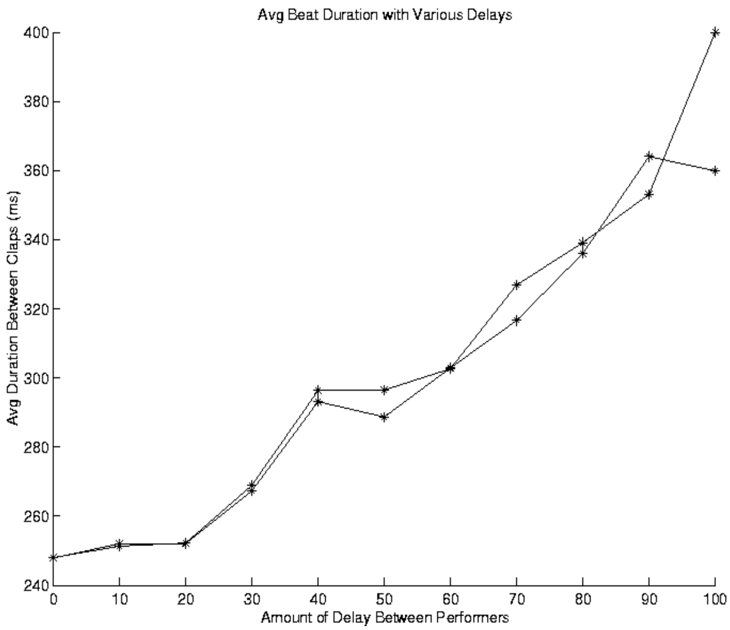
#### 4.2.1. Experiment

The performers practiced clapping together with 0 msec of delay until they felt confident they could perform without error. The first recording was made with 0 msec of delay. After that, the delay between signals rose by 10 msec for each take. The players were informed of the amount of delay as it rose until the experiment was stopped at 100 msec.

#### 4.2.2. Results

The delay in Scenario 1 ranges from 0 - 100 msec. The performers were instructed to play the rhythmic excerpt as accurately as possible.

As the latency increased though, this became difficult and the players could not maintain synchrony. The result was that each performer would try to line up their beat with the other's beat, but they were actually entering late due to the delay. This resulted in recursive tempo slowing, which is shown by the average tempo graph below. The data for this graph is organized based on the average tempo of a performance when subjected to a certain degree of latency. For each latency performance, all the beat durations of that recording were averaged. The number of beats in each performance varied depending on how long the performers played, which was usually around 30 beats.  Each line represents one of the performers.



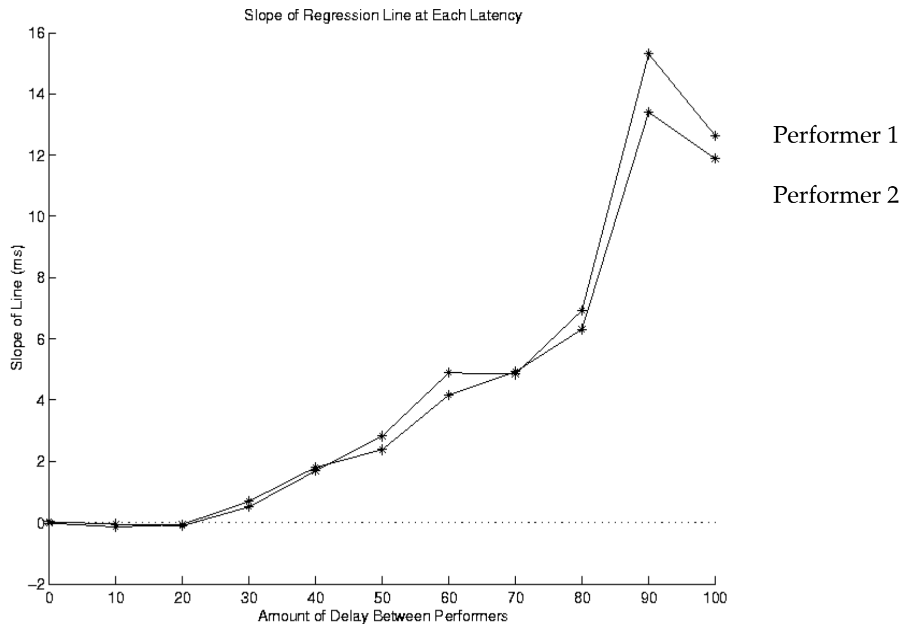

Performer 1

Performer 2

The graph above shows that the average beat duration remains fairly constant up to 20 ms of latency.  Once delay reaches 30 msec, though, both players begin to slow down the duration between beats.

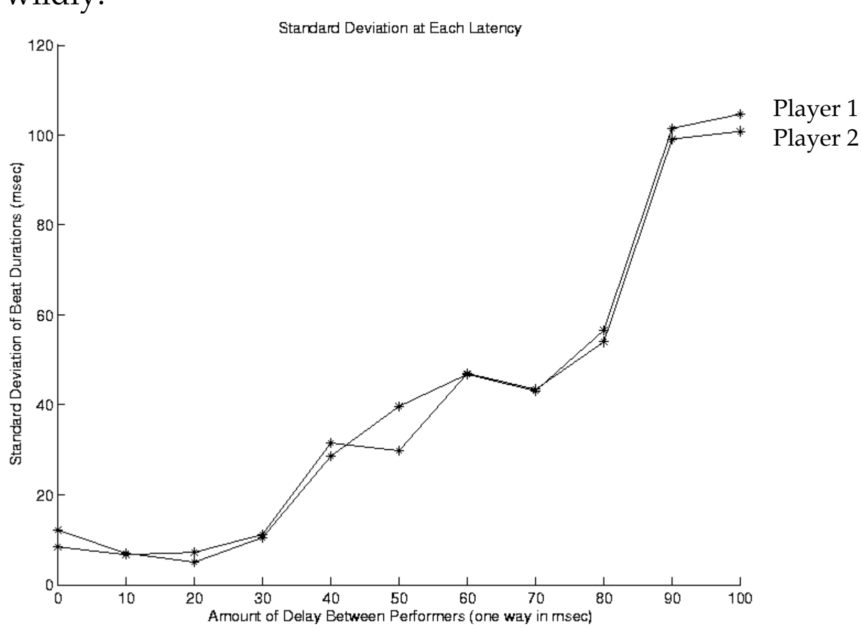
The sharp jump between 20-40 msec indicates the EPT, where performance is possible but is beginning to be affected quite strongly by the delay.  Beyond 40 msec, however, all ensemble characteristics are lost, as synchrony and tempo are compromised.

The recursive tempo slowing found throughout scenario 1 is also shown well by the slope graph below.  This graph displays the slope of each performance that was subjected to varying degrees of latency.




A positive slope indicates that the tempo is slowing down. Around 20-40 msec, the slope becomes positive. This is where the tempo begins to show the effects of delay by decreasing. As the delay increases to 80 and 90 msec, the tempo continues to slow down more and more.

The standard deviation graph below also helps in pinpointing the EPT. The standard deviation was calculated over the course of a performance for all of its beat durations. A high standard deviation means the tempo was fluctuating wildly.



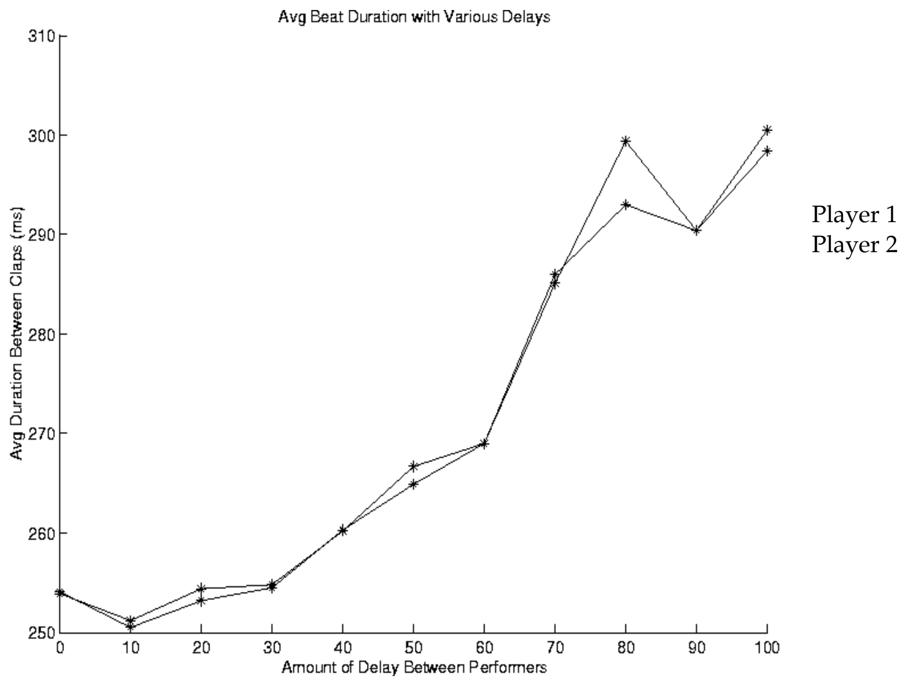
There is a noticeable leap in the standard deviation between 30-40 msec, which means that the tempo started to fluctuate when subjected to those delays. This adds merit to the suggested EPT of 20-40 msec.

**4.3. Scenario 2**

Scenario 2 uses the same experiment setup as scenario 1. The difference lies in how the performers reacted to the delay. Prior to the takes, the performers agreed on a strategy for coping with the latency. They decided to focus more on maintaining a strict tempo than synchronizing the beats.

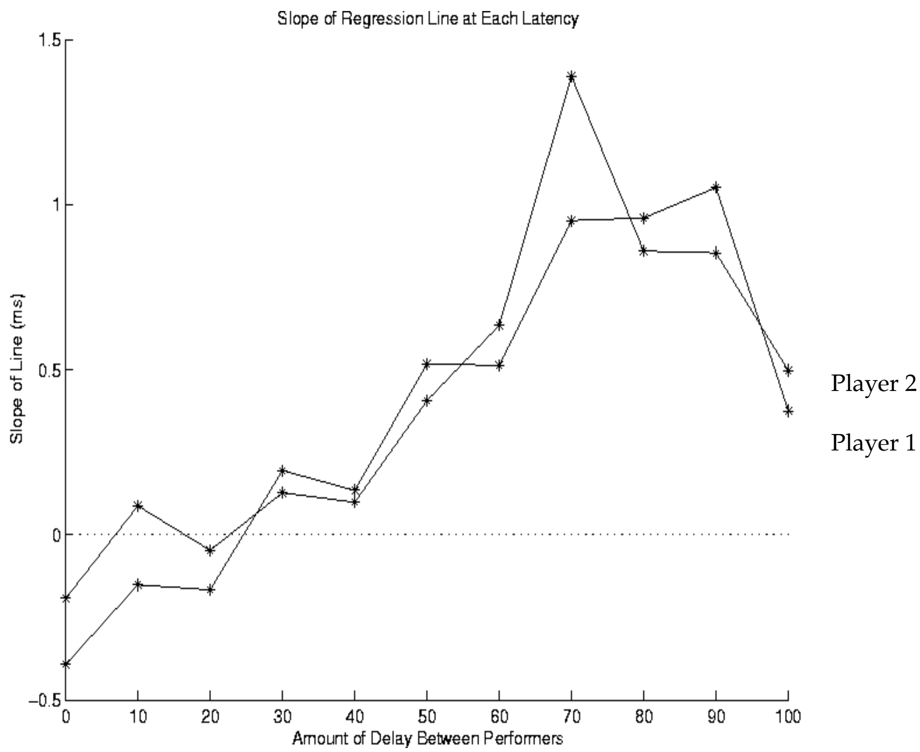
This selective listening resulted in a steadier overall tempo, even as the precision of the ensemble became increasingly problematic as the amount of delay increased. The interactivity between the parts was compromised by the selection of specific musical elements to which each performer was responding. It resulted in a leader / follower relationship, where the follower's signal was consistently late in arriving to the leader, but would be ignored by the leader who was focusing solely on his own tempo. The leader, therefore, was not really playing in an ensemble, but rather was just keeping time for the follower.

The graph below shows that the tempo slows down a little as delay is increased, with the average beat duration reaching a maximum value of 300 msec. However, the tempo does not slow down nearly as much as with scenario 1, where the average beat reached 400 msec.




Also of note is the shifting of the peak to a higher latency. The average tempo drops at the greatest rate between 60-80 msec. This indicates that the EPT has shifted to a higher latency with the new coping strategy.

The slope graph also indicates that tempo is not being affected as much by the delay. At 70 msec of delay, the tempo is slowing down by barely 1.5 msecs per beat, whereas with Test 1, the slope was decreasing by almost 6 msecs per beat at 70 msec of delay.

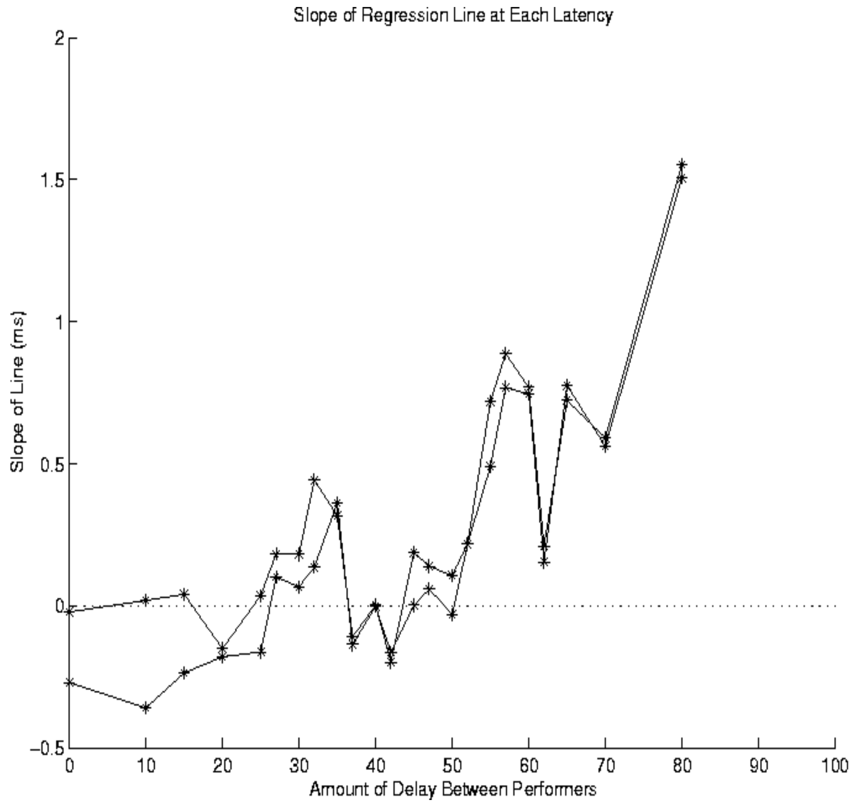



While the leader / follower coping strategy does not simulate true ensemble performance, it does allow reasonably solid performance up to a higher EPT.

### 4.4. Scenario 3

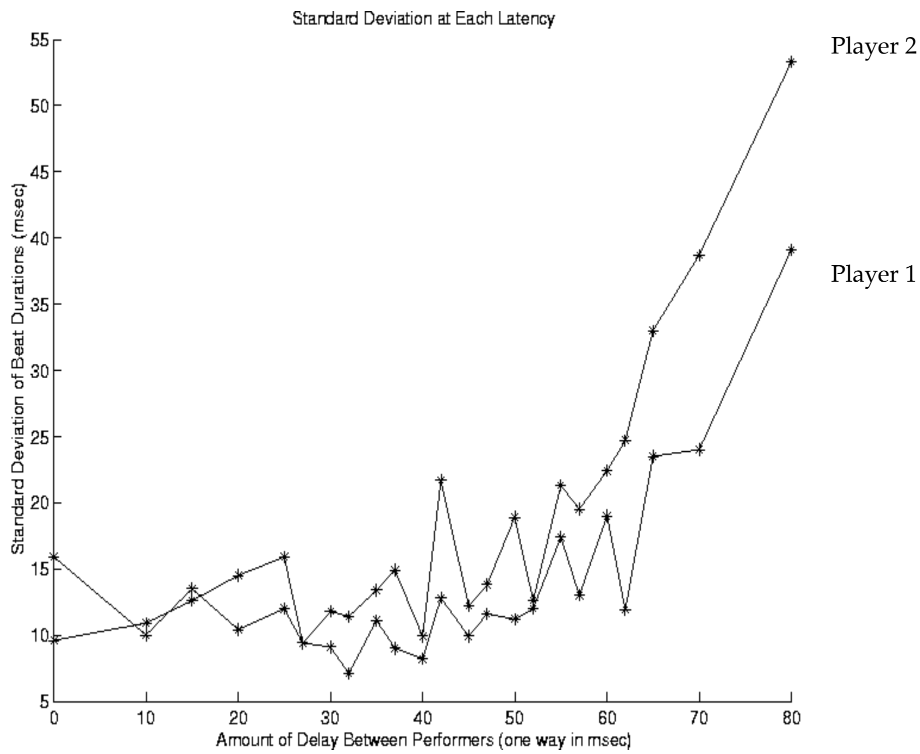
Scenario 3 was run to determine whether the results from the previous two tests were repeatable. It followed the same experiment setup as the previous tests. The key difference with scenario 3 was that the performers did not know the delay prior to each recording. They simply had to begin playing and adjust to whatever delay was present. Also, the delay was varied randomly from take to take, whereas in the previous scenarios it was sequentially increased after every take.

The results were similar to Scenario 2. The players were focusing on tempo rather than synchronization. They were most likely playing in another leader / follower relationship.




The most significant increase in slope happens between 50 – 80 msec. This is where the performance is slowing down at a rapid rate. The rapid slow down helps pinpoint the EPT in this range for scenario 3.

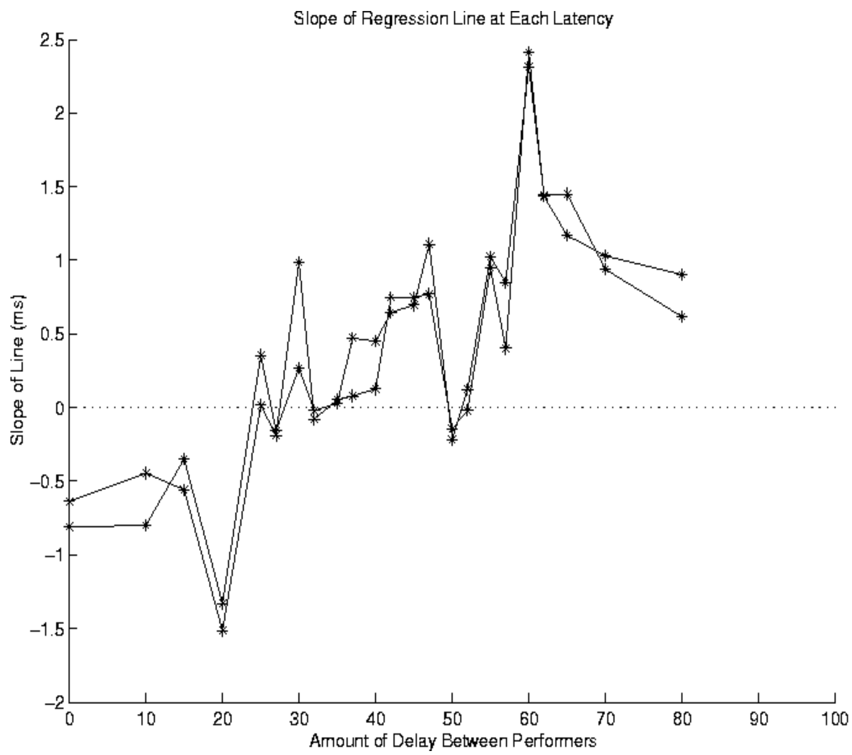
The standard deviation graph has a significant peak between 60-80 msec. This means that the tempo is fluctuating wildly when subjected to latencies in this range. This suggests an EPT of 60-80 msec for Scenario 3.

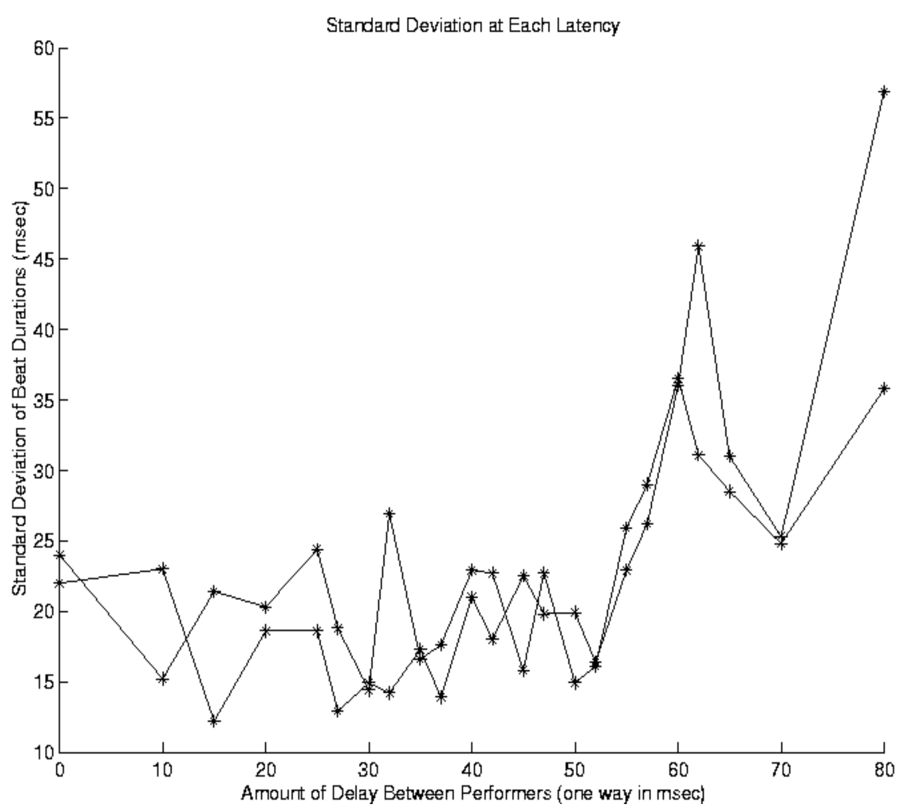



### 4.5. Scenario 4

Test 4 was also run to determine whether the results were repeatable. The key difference in this test was that the starting tempo was instructed to be much slower.

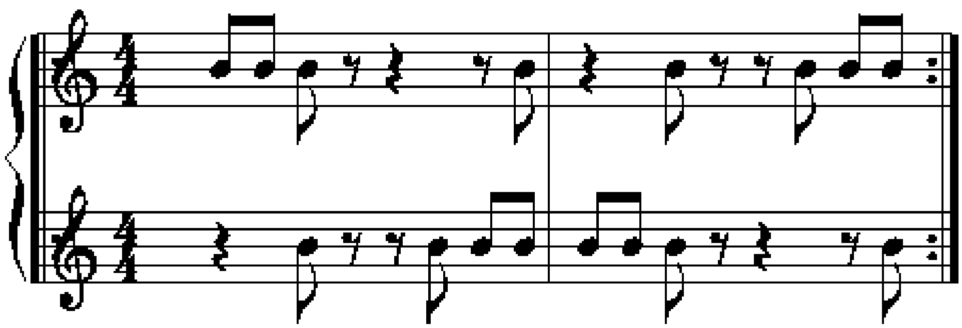
The speed of the tempo did not in fact affect the EPT of a leader / follower performance. The peaks of both the slope and the standard deviation graphs indicate an EPT of 50-70 msec.


Slope of Regression Line at Each Latency
Slope of Line (ms)
Amount of Delay Between Performers
2.5
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
10
20
30
40
50
60
70
80
90
100

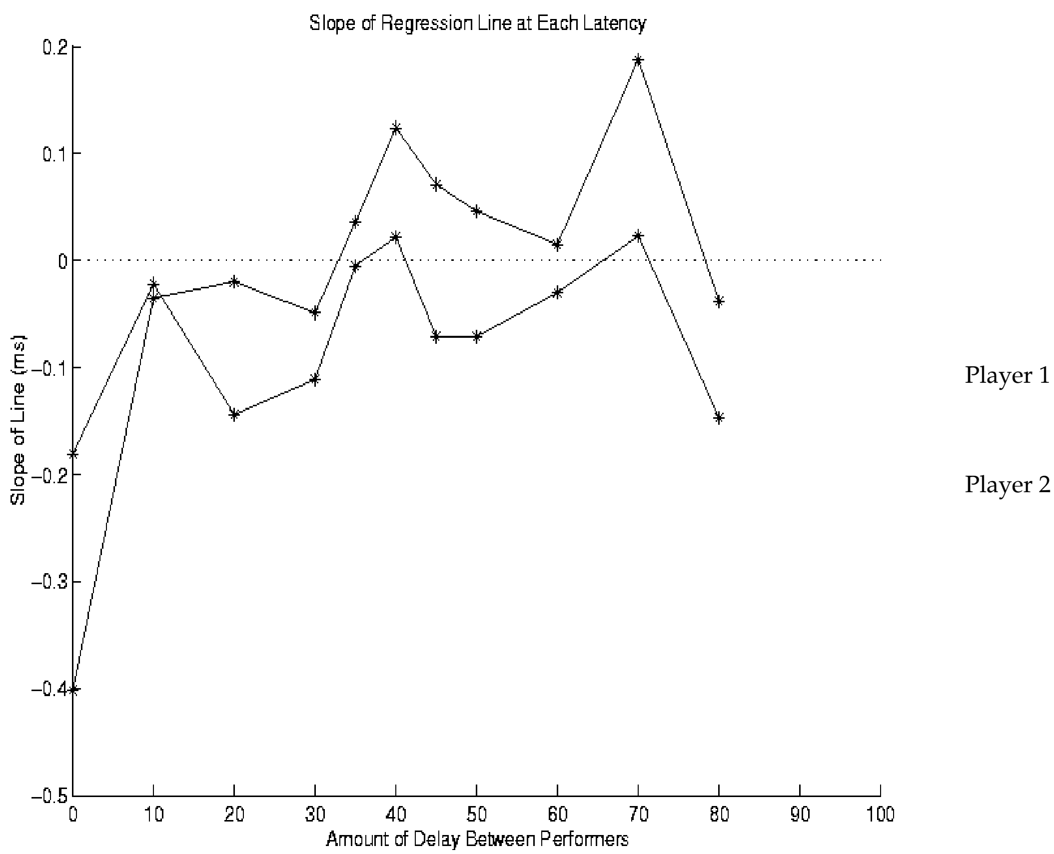

Standard Deviation at Each Latency
Standard Deviation of Beat Durations (msec)
Amount of Delay Between Performers (one way in msec)
60
55
50
45
40
35
30
25
20
15
10
0
10
20
30
40
50
60
70
80

### 4.6. Scenario 5

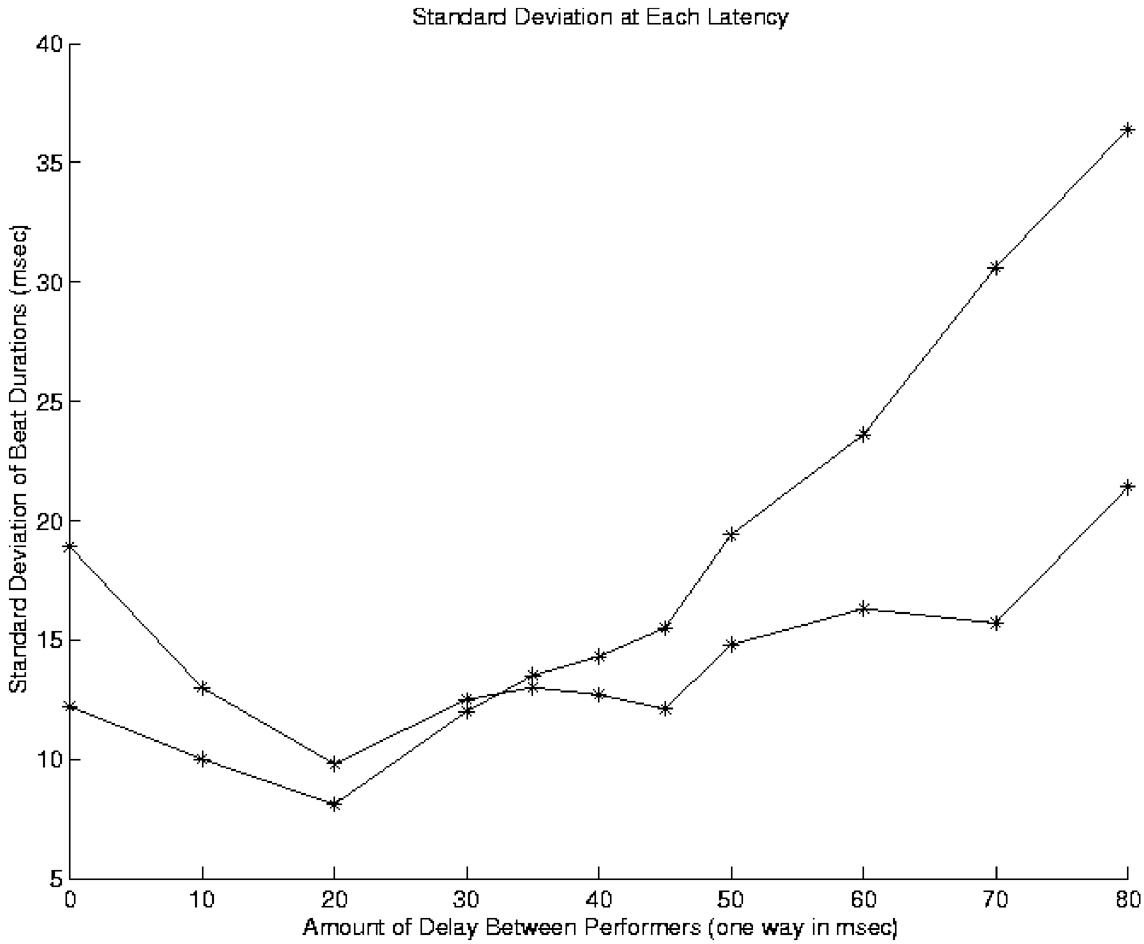
Test 5 followed the same setup as the previous experiments. The key difference was in the performance of a new rhythm (Rhythm 2):



In this scenario, the slope measure did not turn out to be a very good indicator of EPT. This is because the players were able to maintain a very strict tempo (close to 0 slope) for all latencies, as shown below.



Looking at the standard deviation graph below, it is easy to notice that the standard deviation increases the most between 50-80 msec. EPT would most likely lie in this range.




## 4.7. Summary of Suggested EPTs

| Scenario | Level of Interactivity | EPT |
|---|---|---|
| 1 | True Ensemble | 20-40 msec |
| 2 | Leader / Follower | 50-70 msec |
| 3 | Combination | 60-80 msec |
| 4 | Combination | 50-70 msec |
| 5 | Combination | unclear (50-70?) |

# Chapter V
# **Discussion / Conclusions**

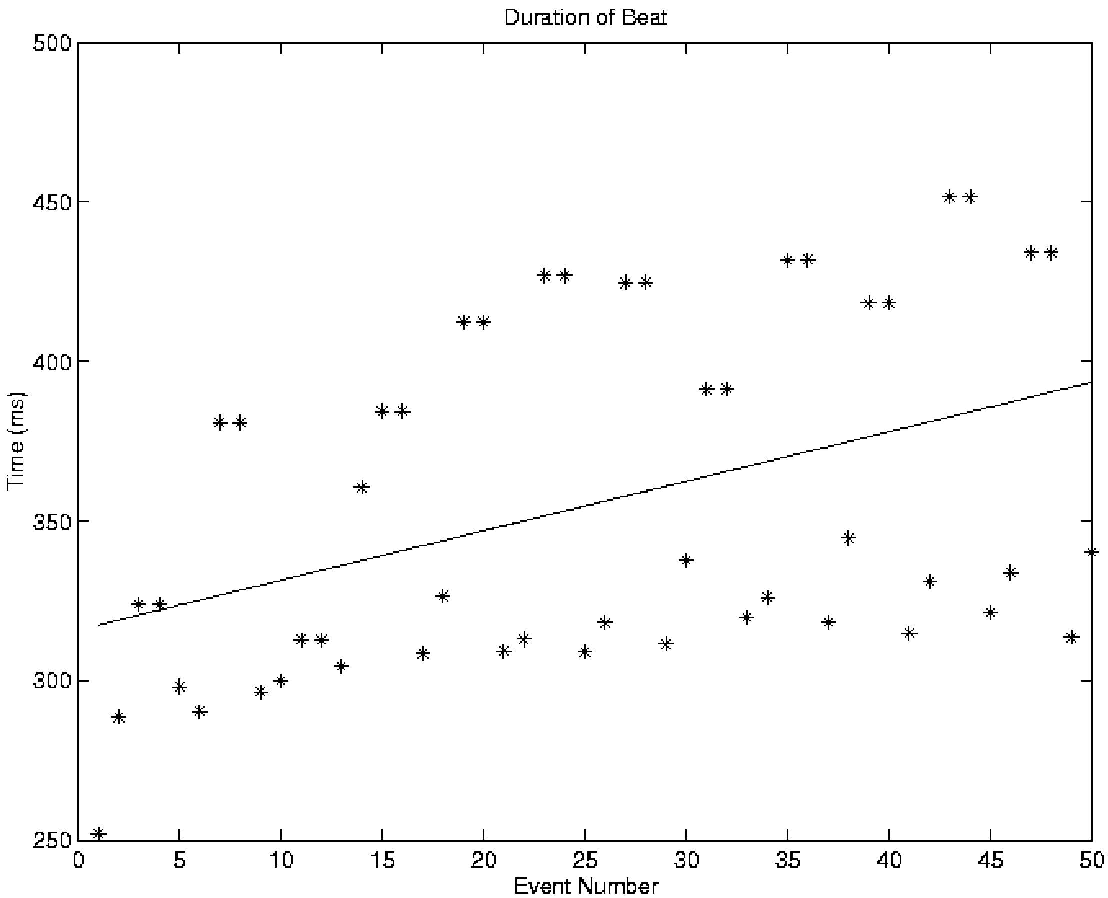
These tests were an initial simulation of an internet-based real-time interactive performance environment. The tests examined the breakdown of simultaneous musical performance resulting from varying amounts of delay between performers.

### 5.1. General Effects

The general effect of delay between performers was a tendency to slow down the tempo.

### 5.2. Swinging Beats

An interesting effect was witnessed when the delay started to become perceptible to the performers. They began to unwittingly swing the long beats. As the Event Detector shows below, the quarter notes were consistently given more time than their eighth note counterparts.




This trend was usually only seen after the EPT has been crossed, and was pretty consistent for all latencies after the EPT. Also, both performers would swing the long beats during the same performance, but the long beats do not line up with each other rhythmically. Thus, the discontinuity in the rhythm is coming from this swinging of the long beats.

### 5.3. Two Coping Strategies and their Respective EPTs

The performers found two strategies for coping with the delay. With the first strategy, each performer attempts to synchronize his own pattern with the sounding result of the other's pattern. The result is that each performer compensates for the delay by entering late and slowing down the overall tempo with each iteration of the pattern. In this test, interactivity is prioritized resulting in a true ensemble performance.

The second strategy results in a leader / follower relationship. This results in selective listening, with a steadier overall tempo but less synchronization as the amount of delay increases. The interactivity between the parts is compromised by the selection of specific musical elements to which each performer is responding. The follower thinks the two signals are synchronized perfectly, whereas the leader must consistently ignore the follower's late entries and hold a constant tempo. The leader is not performing in true ensemble fashion.

It is intuitive that each strategy for coping with the delay would have a separate Ensemble Performance Threshold. As far as the tempo tracker is concerned, the threshold for true ensemble performance lies around 20-40 msec. The threshold for the leader / follower scenario extends to 50–70 msec. Again, these figures are only based on the rigidity of the tempo, not on the synchronization of the beats.

These results appear to push the limits of 5 and 40 msec set by Jeremy Cooperstock up to a higher level of delay. This is encouraging for the future of remote ensemble performance.

### 5.4. Quantitative Tempo Analysis

Based on the results of the delay experiments, a basic measure has been developed for the analysis of a performance. A poor performance can be classified as having: a positive slope, a high standard deviation of beat durations, and a high level of note asynchrony.

It remains difficult to quantifiably determine in a general way whether a performance's tempo is solid or not. Difficulty will be introduced when performers exhibit expressive timing, rubato, and other tempo smearing tactics. Other difficulties stem from the fluid nature of music in general. However, if the players are instructed to maintain as rigid a tempo as possible, the three measures should be useful for future testing situations.

#### 5.4.1. Tempo Direction Measure

Whether the tempo is speeding up or slowing down actually turned out to be a good indicator of whether an ensemble performance is adequate or not. If the slope is negative (the performance is speeding up), it closely resembles normal performance, as Kuhn showed. However, if the beat durations have a positive slope (the performance is slowing down), the performers are feeling the effects of the latency.

#### 5.4.2. Standard Deviation Measure

The standard deviation measure was a good accompanying measure to the tempo direction measure. It was a good indicator of tempo fluctuation from beat to beat, which helps identify when delay's effects are taking hold of the performance.

### 5.4.3. Synchrony Measure

A measure of synchronicity is an essential piece of the analysis of an ensemble performance. There was no test of synchrony for these experiments, and it definitely would have helped in narrowing down the EPT. Future studies must incorporate a tool for analyzing synchrony between beats.

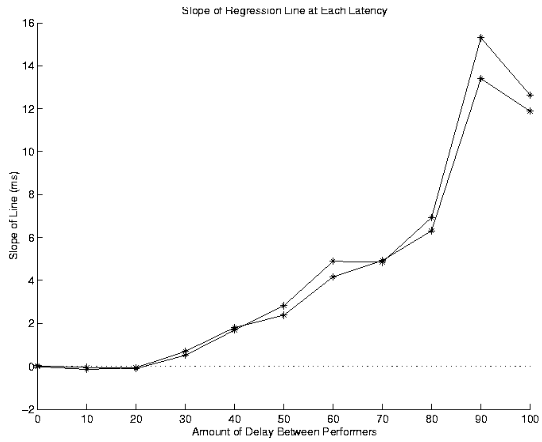
The question is, which signals do you analyze? You must analyze what each performer is hearing. First, you must analyze Performer 1's dry signal with Performer 2's delayed signal. Then, analyze Performer 2's dry signal combined with performer 1's delayed signal. If they are truly playing in an ensemble manner, both analyses should yield similar results - there should be equivalent amounts of difference between the synchrony of the notes regardless of which performer's perspective is being analyzed. If the players are participating in a leader / follower strategy, the follower's perspective should be synchronized well with the leader's beats, while the leader's perspective will be marred by the consistent late beats of the follower.

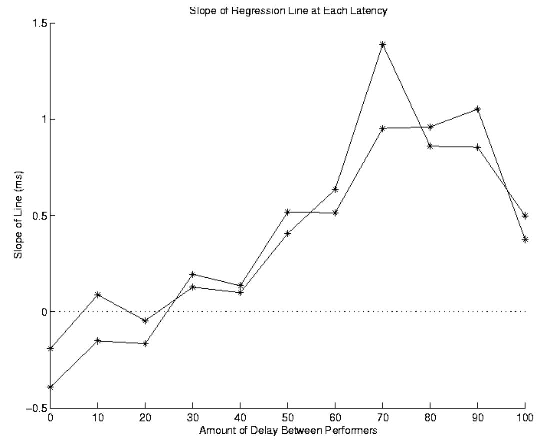
## 5.5. Adding Limited Delay may Actually Improve Performance

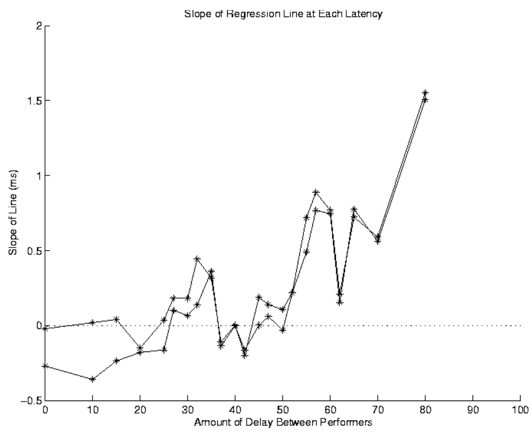
The tempo direction measure illuminated a very interesting phenomenon:

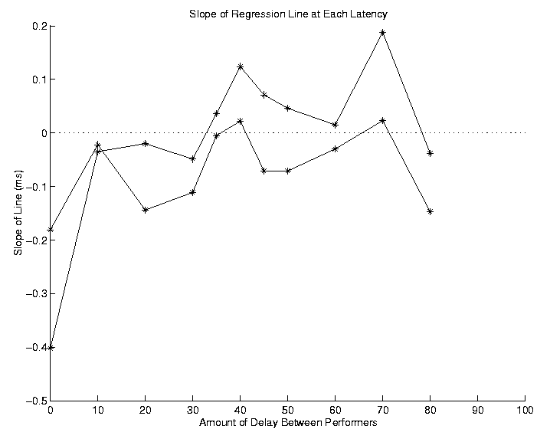
The performers noted that a particular performance with around 20 msec of delay was quite easy to play and the tempo was very stable. It is possible that a certain amount of delay could actually help synchronize an ensemble performance. After all, if the players were playing in the same room, there would be initial delay correlated with their distance apart and there would also be reverberant delayed signals.

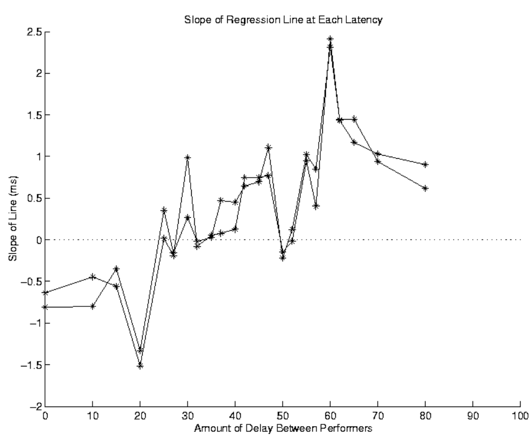
Looking at each graph of the slopes it is easy to see that between 20 and 30 msec, the slopes of the beat durations cross the 0 plane, indicating that the tempo is starting to slow down.



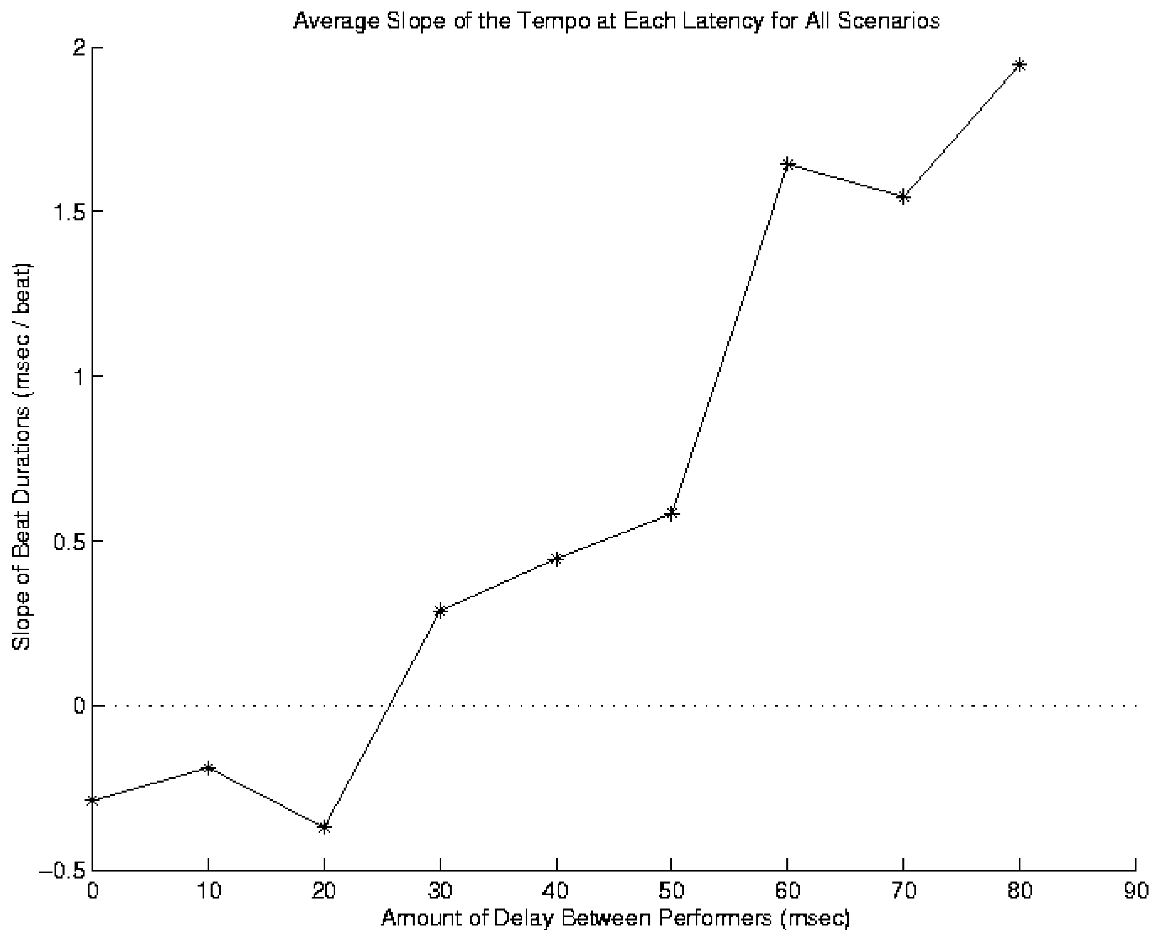



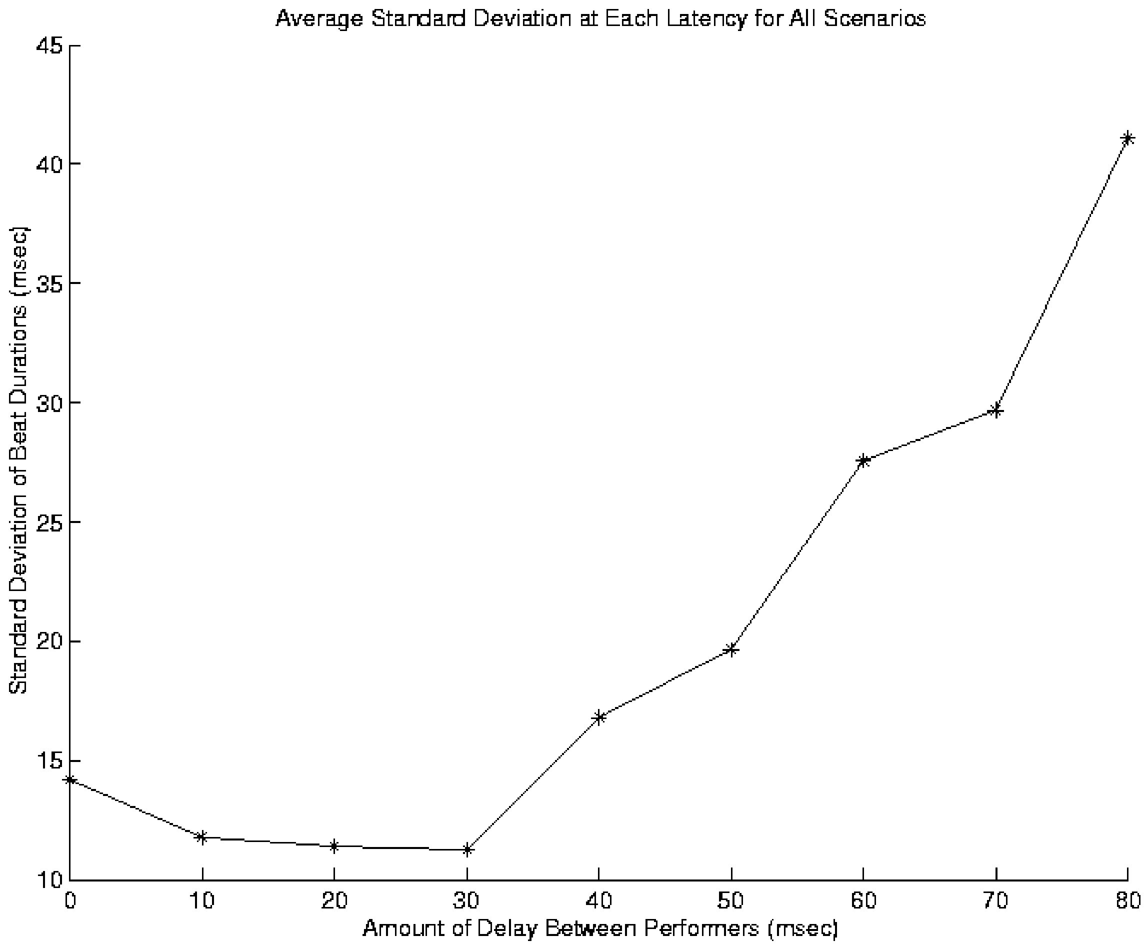




More importantly, though, between 10-20 msec, the slope is very close to 0 for all the tests. This indicates that the tempo is neither slowing down nor speeding up with this amount of delay. This could indicate that with a certain amount of latency, the tempo becomes more stable as the performers lock on together.




Further evidence of this phenomenon can be seen by looking at the standard deviation graph below, which shows that the average beat duration during the performance is relatively more stable at latencies of 10-30 msec than at latencies of 0 and 40 msec.




This hypothesis, that a certain amount of latency may actually help stabilize a performance, seems to fit well with the spatial set up of normal ensemble performances. Players are never separated by a distance equivalent to 0 msec. Rather, they are usually separated by somewhere between 4 - 20 msec (which converts to approx. 4 - 20 ft).

These results merit further study.

**5.6. Reverberation**

One of the puzzling results of this study is the notion that ensemble performance is more robust when the delay is spatially manipulated (outdoor drummers were able to stabilize their tempo at a distance of 100 ft, or 100 msec) versus when it is electronically manipulated (EPTs of 20-40 msec and 50-70 msec). This is thought to be a result of auditory cues such as reverb that are present when the two performers share the same acoustic space but are absent from the electronic tests.

5.6.1. A Preliminary Test

A preliminary test was conducted to determine whether adding reverb to the player's signals would help increase the threshold of delay in the electronic tests. The tests were run in the same manner as Scenarios 1-5, only reverb was artificially added to each signal through a digital plug-in. The hope was that reverb would provide a sort of "auditory cushion" that would mush the signal, making it easier for the players to synchronize with each other. If this were the case, effective performance in the electronic environment should have approached the 100 msec threshold that was demonstrated in the spatial delay experiments.

The tests did not indicate that synthesized reverb was providing any positive effect. The tempo still began to break down around 30-40 msec. These tests were not analyzed using the event detector, but were evaluated by the ear of trained musicians who agreed that the threshold was around 30-40 msec.

The results do not mean that reverb is completely ineffective and does not play a role in the synchronizing of tempos. Rather, our suggested interpretation hinges around the fact that the reverb applied to each signal was too artificial. It did not in any way make it seem like the two players were sharing the same auditory space.

5.6.2. A Proposed Solution

We found that reverb does not help put isolated players in the same room if it's the wrong reverb. The right reverb would be one that "encloses" both players such that all delays are relative to their respective positions. That means transporting some of the reflection paths over the net separate from the dry signals (if done waveguide-style) or using convolution to position each performer within the same room using two-location, stereo, impulse responses.

## 5.7. Future Study

5.7.1. Multiple Performers

It would be interesting to see how additional performers would affect the stability of the tempo and the synchronization of an ensemble performance.

My guess is that additional performers would provide a stabilizing influence on the tempo, although synchrony would still break down at a certain point. Jason Cooperstock also believes that additional performers help to raise the EPT.

#### 5.7.2. Real Music vs. Clap Tests

Hopefully, in further studies, automatic event detection can be applied to instrumental music as opposed to just percussive music. This could then help to shed light on the issue of whether the size of the ensemble affects EPT. My hypothesis is that the slow attack time makes synchronization difficult for a string quartet, not the size of the ensemble.

Real music would also show whether a more complex, non-repeating rhythm would lower the EPT. This is highly likely, as it was too easy for the performers in these tests to pay attention to only their individual rhythm.

#### 5.7.3. Different EPT's for Different Types of Music

More tests could be done to determine whether different types, styles and speeds of music have different EPTs.

#### 5.7.4. Improvisation

Improvisation would be an interesting test. After a certain amount of delay, certain types of musical improvisation would become impossible.

#### 5.7.5. Shifting Between Coping Strategies

As the delay approaches the threshold of playability, the players may actually be shifting between coping strategies. They could shift from true ensemble to leader / follower once the delay reaches the true ensemble threshold. This definitely confounds results, but would help with performances over distances just above the EPT.

### 5.8. Application

#### 5.8.1. Long-Distance Sessions over the Net

The theoretical roundtrip time across the US and back is approximately 40 msec. Our experiments with very good networks have achieved RTT as low as 75 msec. This means that effective musical collaboration is possible at distances as great as the continent.

In fact, Stanford's SoundWire group demonstrated that ensemble performance was possible by recording high-bandwidth, remote performances over Internet 2. For these performances, there were two players: a cellist and a pianist. The cellist played an electric cello from various locations on the East Coast, while the pianist was always stationed in the studio at CCRMA where the sessions were recorded to tape. The piece performed was Brahms' Sonata for Piano and Violoncello in E minor, Op. 38. It was chosen for its rhythmic complexity and variability, thus forcing an ensemble environment. Three East Coast NGI sites were chosen: McGill University, Internet 2 Headquarters, and Princeton University.

Firewalls, congestion, address problems, and dropouts were all experienced in these one-day sessions, but one of the sessions, from Internet2 headquarters at Armonk, NY, worked surprisingly well.

The distance between CCRMA and Internet2 headquarters in Armonk, New York is approximately 9000 km round trip. RTT of around 75 msec was achieved and sustained. There were very few dropouts, and the RTT was just on the "hairy edge" for an unencumbered performance (37.7 msec one-way). The performers maintained a relatively rigid tempo. It sounds decent to the untrained ear, but actually oscillates somewhat over the course of the performance. Microphones and headphones were connected and compared to a telephone connection also open between the same rooms. Network RTT was nearly as good as the telephone's, and conversation seemed comfortable. The audio quality was of course much better.

### 5.8.2. Design Specs

Our motivation is the need for a "design spec" in engineering new systems that support truly natural feeling audio collaboration environments. With solid results from the delay experiments, efforts can be made to design systems that will maintain latency below the ensemble performance thresholds. For instance if two players wanted to perform together over the internet, they could be discouraged from attempting a performance if the latency were above the threshold for their particular type of music.

It is very likely that audio will become one of the driving forces for Internet engineering, particularly with regard to evaluating QoS.

### 5.8.3. A Little Latency Could be Good Latency

Dave Phillips, who maintains the *Linux Music & Sound Applications Website*, writes, "Any real-time or interactive software hopes for zero perceptible delay..."

This has been the general consensus among software and hardware developers for many years. However, this study indicates that the addition of some latency (even perceptible latency) could be beneficial to interactive performance. The just-noticeable latency seems to help make each performer's tempo more rigid, and helps stem the constant tendency to speed up.

Recording studios and audio software companies may find this conclusion very useful. Adding around 5-20 msec of latency between the performers in interactive sessions could be very beneficial to the overall quality of the music. Any time two people are isolated acoustically, but are playing through headphones, adding 5-20 msec of latency could be beneficial.

After all, when people are performing together in the same room, their signals never arrive instantaneously. They are always naturally delayed by physical distance between the performers and the slurring provided by reverberation off the walls.

### 5.9. Conclusions

Conclusions were as follows:

(1) The direction of the tempo was a very useful indicator of whether a performance was being hindered by the effects of latency. If the delay was greater than 30 msec, the tempo would begin to slow down. This gives a solid indication that EPT for impulsive, rhythmic music lies between 20-30 msec.

(2) A coping strategy was discovered that allowed the performers to maintain a solid tempo up to 50-70 msec of delay. The strategy can be quickly summarized as a leader - follower relationship. Unfortunately, this strategy results in a severe decrease of synchrony on the leader's end of the performance.

(3) It is most likely that EPT varies depending on the type of music (speed, style, attack times of instruments, etc).

(4) When delay is between 10-20 msec each way, it may be providing a stabilizing effect on the tempo. 10-20 msec of delay may be better for ensemble performance than 0 msec of delay.

(5) The EPT determined in the electronic delay tests was much lower than that estimated in the outdoor delay tests. This is predicted to be due to the lack of auditory cues in the electronic tests such as reverb and variable amplitude which were present in the outdoor tests.

# Appendix A. Amplitude Envelope Code

```
/*****************************************/
/*
 Program that outputs an amplitude envelope to a file.
 Original playN Code by Gary Scavone
 Modified by Nathan Schuett, 2001

 This program is currently written to load and play
 a WAV file.  It determines the maximum and minimum
 amplitudes for every n samples.  The sample index and
 amplitude are both written to a Matlab file (ampenv.m).
*/
/*****************************************/
#include "iostream.h"
#include "RtWvOut.h"
#include "WavWvIn.h"
        FILE* textfile;

void usage(void) {
 /* Error function in case of incorrect command-line
   argument specifications
 */
 printf("\nuseage: playN N file fs \n");
 printf("    where N = number of channels,\n");
 printf("    file = the .wav file to play,\n");
 printf("    and fs = the sample rate.\n\n");
 exit(0);
}

int main(int argc, char *argv[])
{
 // minimal command-line checking
 if (argc != 4) usage();

 int chans = (int) atoi(argv[1]);

 // Define and load the SND soundfile
 WvIn *input;
 try {
  input = new WavWvIn((char *)argv[2], "oneshot");
 }
```

```
 catch (StkError& m) {
  m.printMessage();
  exit(0);
 }
 // Set playback rate here
 input->setRate(atof(argv[3])/SRATE);

 // Define and open the realtime output device
 WvOut *output;
 try {
    output = new RtWvOut(chans);
 }
 catch (StkError& m) {
  m.printMessage();
  exit(0);
 }


/*************************************/
 double sbmax, sbmin, sampleperiod;
 int i, maxtime, mintime, lowfreq, numsamps;

 sbmax=-1.0;
 sbmin=1.0;
 i = maxtime = mintime = 0;
 cout << "How many samples would you like in the period? ";
 cin >> numsamps;
 textfile = fopen("/user/n/nschuett/ampenv.m","w");
 fprintf(textfile, "B=[");

//****** Here's the runtime loop ******///

 while (!input->isFinished()) {
  i++;
  double tmp = input->tick();
  if (tmp >= sbmax) {sbmax=tmp;maxtime=i;}
  if (tmp <= sbmin) {sbmin=tmp;mintime=i;}
  if ((i % numsamps) == 0) {
       fprintf(textfile, "%d %1.12f \n", maxtime, sbmax);
       sbmax=-1.0;sbmin=1.0;maxtime=0;mintime=0;}
  output->tick(tmp);
 }
```

```
  //***** Clean up **********///
  delete input;
  delete output;
  fprintf(textfile, "]");
  fclose(textfile);
  printf("textfile closed\n");

}
```

# Appendix B. Event Detector / Tempo Analyzer Code

```
%%%%%%% An event detector built for Matlab %%%%%%%%%
%%%%%%% Written by Nathan Schuett        %%%%%%%%%
%%%%%%% July, 2001                       %%%%%%%%%


%%% Includes %%%

ampenv;
format long;
format compact;


%%% Initialize %%%

surflength = 5;
halflength = 2;      %%% (surflength/2) rounded down; %%%
slopethresh = .03;

[numrows, numcols] = size(B);
numslopes=numrows-halflength;


%%% Calc Slope for each sample %%%

for x = (halflength+1):numslopes
  sl = polyfit(B(x-halflength:x+halflength,1),B(x-halflength:x+halflength,2), 1);
  xpoint(x)=B(x,1);
  ypoint(x)=B(x,2);
  slope(x) = sl(:,1);
  yint(x) = sl(:,2);
%%%  fprintf('%12.10f is slope %12.10f is y-int. \n', slope(x), yint(x));
%%%  inp=input('Press Return');
end


%%% Graph all the slopes %%%

%   hold on;                %%% prevents graph rewriting each time %%%
%for i=halflength+1:numslopes
%   xtix=linspace(xpoint(i)-2000,xpoint(i)+2000,1000);
%   ypt = polyval([slope(i),yint(i)],xtix);
```

```
%   plot(xtix,ypt,'-r')
%end
%   plot(B(:,1),B(:,2), 'k')     %%% The original envelope curve %%%


%%% Search through slope array %%%
%%% Find maximum slope %%%

  n=halflength+1;
  maxslope = 0;
while n <= numslopes
  if slope(n) >= maxslope
    maxslope = slope(n);
  end
n=n+1;
end

fprintf('%12.10f is maximum slope. \n', maxslope);


%%% Wait for User %%%
   inp=input('Press Return');


%%% Set threshold as slopethresh * maximum slope %%%
%%% If a slope is > than slopethresh * max, examine data closer.
%%% Find the max slope in surrounding area.  That is the event.

  num = 1;
  k=halflength+1;
while k <= (numslopes-50)
  if slope(k) >= (maxslope * slopethresh)
    tempmaxslope = slope(k);
    tempx = xpoint(k);
    tempy = ypoint(k);
    tempyint = yint(k);
    for t = 1:15
      if slope(k+t) >= tempmaxslope
        tempmaxslope = slope(k+t);
        tempx = xpoint(k+t);
        tempy = ypoint(k+t);
        tempyint = yint(k+t);
      end
```

```
    end
    eventslope(num) = tempmaxslope;
    eventx(num) = tempx;
    eventy(num) = tempy;
    eventyint(num) = tempyint;
    num = num + 1;
    k = k + 50;
  else k=k+1;
  end
end


%%% print out sample index of events %%%

for q = 1:(num-1)
   eventms(q) = eventx(q)/44.1;
   fprintf('%12.10f is slope. %3.0f is sample index =  %2.6f ms. \n',
eventslope(q),eventx(q),eventms(q))
end


%%% pause until ready %%%

inp=input('Press Return');
hold off;


%%% Graph the event slopes and event points %%%

plot(B(:,1),B(:,2), 'k')     %%% The original envelope curve %%%
hold on;                     %%% prevents graph rewriting each time %%%

for i = 1:(num-1)
  xtix=linspace(eventx(i)-2000,eventx(i)+2000,1000);
  ypt = polyval([eventslope(i),eventyint(i)],xtix);
  title('Event Slopes')
    xlabel('Sample Index')
    ylabel('Amplitude')
  plot(xtix,ypt,'-r')
  plot(eventx(i),eventy(i),'b*')
end
```

```
%%% pause until ready %%%

inp=input('Press Return');
hold off;


%%% print out offset between events %%%
%%% offset is measured as difference between event q and q+1 %%%

for q = 1:(num-2)
   offset(q) = eventms(q+1)-eventms(q);
   fprintf('Offset between event %3.0f and event %3.0f = %2.6f ms. \n',
q+1,q,offset(q))
end



%%% Graph the offset %%%

for i = 1:(num-2)
  title('Duration Between Events ')
    xlabel('Event Number')
    ylabel('Milliseconds')
  plot(i,offset(i),'b*')
  hold on;             %%% prevents graph rewriting each time %%%
end


%%% Find the minimum offset time %%%

minoffset = offset(1);
for i = 2:(num-2)
    if offset(i) < minoffset
      minoffset = offset(i);
    end
end


%%% Find all the small offsets %%%
%%%          and
%%% Assign the number of large offsets that go with each small offset that
%%%follows %%%
```

```
bigpersmall(1) = 0;
bigpersmall(2) = 0;
smoffsetcounter = 1;
smoffset(smoffsetcounter) = minoffset;

for i = 1:(num-2)
    if offset(i) < (smoffset(smoffsetcounter) * 1.5)  %%% then it's small event %%%
      smoffsetcounter = smoffsetcounter + 1;
      smoffset(smoffsetcounter) = offset(i);
      bigpersmall(smoffsetcounter+1) = 0;
    else                                  %%% it's a missed beat or double beat %%%
      bigpersmall(smoffsetcounter) = bigpersmall(smoffsetcounter) + 1;
    end
end


%%% Print out the small offsets and the number of large offsets associated with
%%% each %%%

for i = 1:smoffsetcounter
  fprintf('Small offset %3.0f = %2.6f ms. \n', i,smoffset(i))
    fprintf('It has %3.0f large offsets with it. \n', bigpersmall(i))
end

%%% For every small offset, divide the large offsets associated with it
%%% by 2,3,4,or 5

    origoffsetcounter = 0;
    adjcounter = 0;
    if bigpersmall(1) >= 1
      for bps = 1:bigpersmall(1)
            if 3.0 <= (offset(origoffsetcounter+bps)/smoffset(1))
              if 3.5 <= (offset(origoffsetcounter+bps)/smoffset(1))
                if 4.5 <= (offset(origoffsetcounter+bps)/smoffset(1))
                  for t = 1:5
                    adjcounter = adjcounter + 1;
                    adjoffset(adjcounter) = (offset(origoffsetcounter+bps)/5);
                  end
                        origoffsetcounter = origoffsetcounter + 1;
                else
                  for t = 1:4
                         adjcounter = adjcounter + 1;
                    adjoffset(adjcounter) = (offset(origoffsetcounter+bps)/4);
```

```
            end
            origoffsetcounter = origoffsetcounter + 1;
          end
        else
          for t = 1:3
            adjcounter = adjcounter + 1;
            adjoffset(adjcounter) = (offset(origoffsetcounter+bps)/3);
          end
          origoffsetcounter = origoffsetcounter + 1;
        end
      else
          for t = 1:2
            adjcounter = adjcounter + 1;
            adjoffset(adjcounter) = (offset(origoffsetcounter+bps)/2);
          end
          origoffsetcounter = origoffsetcounter + 1;
      end
  end

 end      %%% bigpersmall(1) = 0.  So, do nothing because smoffset(1) %%%
       %%%    is always the minoffset (it's not a real event) %%%


for i = 2:smoffsetcounter
 adjcounter = adjcounter + 1;
 adjoffset(adjcounter) = smoffset(i);
 origoffsetcounter = origoffsetcounter + 1;

 if bigpersmall(i) >= 1
  for bps = 1:bigpersmall(i)
    if 3.0 <= (offset(origoffsetcounter+bps)/smoffset(i))
      if 4.3 <= (offset(origoffsetcounter+bps)/smoffset(i))
       if 4.5 <= (offset(origoffsetcounter+bps)/smoffset(i))
        for t = 1:5
         adjcounter = adjcounter + 1;
         adjoffset(adjcounter) = (offset(origoffsetcounter+bps)/5);
        end
       else
        for t = 1:4
                adjcounter = adjcounter + 1;
          adjoffset(adjcounter) = (offset(origoffsetcounter+bps)/4);
        end
       end
```

```
          else
            for t = 1:3
              adjcounter = adjcounter + 1;
              adjoffset(adjcounter) = (offset(origoffsetcounter+bps)/3);
            end
          end
        else
            for t = 1:2
              adjcounter = adjcounter + 1;
              adjoffset(adjcounter) = (offset(origoffsetcounter+bps)/2);
            end
        end
      end
    origoffsetcounter = origoffsetcounter + bigpersmall(i);
    end
end


%%% pause until ready %%%

inp=input('Press Return');
hold off;


%%% print out offset between events %%%
%%% offset is measured as difference between event q and q+1 %%%

for q = 1:adjcounter
   fprintf('Offset %3.0f = %2.6f ms. \n', q, adjoffset(q))
end


%%% Graph the offset %%%

for i = 1:(adjcounter)
    title('Duration of Beat')
    xlabel('Event Number')
    ylabel('Time (ms)')
  plot(i,adjoffset(i),'b*')
  hold on;          %%% prevents graph rewriting each time %%%
end
```

```
%%% Draw a best fit line through the offsets %%%
 slope1 = polyfit(1:adjcounter,adjoffset(1:adjcounter),1);
 itsslope = slope1(:,1);
 yint = slope1(:,2);
%%%

%%% Print out the slope and yint %%%
 fprintf('The slope of the beats = %4.6f \n', itsslope)
 fprintf('The yint = %4.6f \n', yint)
%%%

%%% Graph the slope %%%

  hold on;                  %%% prevents graph rewriting each time %%%
  xtix = linspace(1,adjcounter,300);
  ypt = polyval([itsslope,yint],xtix);
  plot(xtix,ypt,'-r')
%%%


%%% Calculate the Average of the adjusted offsets %%%

sumoffset = 0;
for i = 1:adjcounter
  sumoffset = sumoffset + adjoffset(i);
end
avgoffset = sumoffset/adjcounter;
fprintf('The Average offset = %4.6f ms \n', avgoffset)


%%% Calculate the Standard Deviation of the offsets %%%

sdsum = 0;
for i = 1:adjcounter
 sdsum = sdsum + (adjoffset(i) - avgoffset)*(adjoffset(i) - avgoffset);
end
 stdeviation = sqrt(sdsum/adjcounter);
 fprintf('Standard Deviation = %4.6f ms \n', stdeviation)
```

## REFERENCES


[1] http://www.oreillynet.com/lpt/a/467

[2] http://www.networking.ibm.com/voice/voice-data.html

[3] http://ww2.mcgill.ca/icc/canarie/learnWare/milestone1/App1.htm

[4] Rasch, R.A. "The perception of simultaneous notes as in polyphonic music." Acustica. Vol. 40 (1978) 21-33.

[5] Rasch, R.A. "Synchronization in performed ensemble music." Acustica. Vol. 43 (1979) 121-131.

[6] Chafe, C., et al. *A Simplified Approach to High Quality Music and Sound over IP*,Proc. Workshop on Digital Audio Effects (DAFx-00), Verona, Italy, 159-163, 2000.

[7] Rossing, Thomas D. The Science of Sound. 2nd ed. Massachusetts: Addison-Wesley, 1990.

[8] Kuhn, T. "Discrimination of Modulated Beat Tempo by Professional Musicians." Journal of Research in Music Education. Vol. 22 No. 4. (270-277).

[9] Kuhn, T. and Gates, E. "Effect of Notational Values, Age, and Example Length on Tempo Performance Accuracy." Journal of Research in Music Education. Vol. 23 No. 3. (203-210).

[10] http://cnmat.cnmat.berkeley.edu/~adrian/latency.html

[11] Gordon, J. (1984.) "Perception of Attack Transients in Musical Tones." CCRMA Report No. STAN-M-17.

[12] Schloss, Andrew W. (1985) "On the Automatic Transcription of Percussive Music – From Acoustic Signal to High-Level Analysis." CCRMA Report No. STAN-M-27.