Kalman Filtering for Computer Music Applications

by

Manjinder Singh Benning

B.Eng, University of Victoria, 2004

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

Kalman Filtering for Computer Music Applications

By

Manjinder Singh Benning
B.Eng, University of Victoria, 2004

Supervisory Committee

Dr. Peter Driessen, Supervisor
(Department of Electrical and Computer Engineering)

Dr. George Tzanetakis, Member
(Department of Electrical and Computer Engineering)

Dr. Andrew Schloss, Outside Member
(Department of Music)

Dr. Ali Shoja, External Examiner
(Department of Computer Science)

Supervisory Committee

Dr. Peter Driessen, Supervisor
(Department of Electrical and Computer Engineering)

Dr. George Tzanetakis, Member
(Department of Electrical and Computer Engineering)

Dr. Andrew Schloss, Outside Member
(Department of Music)

Dr. Ali Shoja, External Examiner
(Department of Computer Science)

Abstract

This thesis discusses the use of Kalman filtering for noise reduction in a 3-D gesture-based computer music controller known as the Radio Drum and for real-time tempo tracking of rhythmic and melodic musical performances. The Radio Drum noise reduction Kalman filter is designed based on previous research in the field of target tracking for radar applications and prior knowledge of a drummer's expected gestures throughout a performance. In this case we are seeking to improve the position estimates of a drum stick in order to enhance the expressivity and control of the instrument by the performer. Our approach to tempo tracking is novel in that a multi-modal approach combining gesture sensors and audio in a late fusion stage lead to higher accuracy in the tempo estimates.

Table of Contents

List of Tables

List of Figures

Acknowledgments

Love to my Mother for her selflessness when dealing with life's challenges, your work ethic, patience, and calm nature inspire me. May we continue to learn from each other through the destruction of our ego.

For all of the Benning family I wish for compassion, non-judgment, openness, and personal evolution. Thank you all for being. May we learn to exist together with smiles, non-violent communication and a healthy diet.

Eternal blessings to my extended soul island family. You create me, inspire me and teach me. I wish to appreciate you more with deeper compassion and love. You let the music flow.

I thank myself for the self-discipline to grow out of unhealthy patterns, push to maintain openness for new experiences and continuing to search for happiness and truth through selfless service for the ones I love. I will always be learning.

Thanks to Dr. Michael McGuire for his guidance through the challenging world of estimation.

Chapter 1

1 Introduction – Motivation

Since its conception in the 1960's, Kalman filtering has been extensively applied to a variety of engineering problems, ranging from object tracking to auto-pilot navigation as well as to remote sensing and geophysical exploration. In the relatively newer interdisciplinary field of computer music, the potential of the Kalman filter has only been realized in the past few years. Cheaper access to fast computing technology coupled with high-level intuitive programming interfaces has sped up the emergence of the "artistic scientist". Algorithms and methods typically reserved for the mathematically minded are now being explored for more right-brained endeavors leading to novel applications of science and engineering in the realm of the arts. This work primarily concerns itself with two artistic applications of the Kalman filter: Improved tracking of percussive gestures using a 3-D gesture-based musical controller, and machine perception of tempo during a musical performance using sensor/audio fusion. In both cases we are challenged with the problem of tracking the hidden state of a dynamic time varying system in real-time with noise corrupted observations.

The ability for an artist to reliably translate their emotional intentions into musical expression depends on the quality of their instrument. The motivation behind the first application mentioned is to reduce the noise in the Radio Drum 3-D gesture

sensing system. The Radio Drum uses capacitive sensing to locate the position of two radio frequency emitting drum sticks moving above its surface. This system is prone to random electromagnetic disturbances created both within the system itself and within the environment of a typical performance. A multi-model Kalman filtering system is used to reduce the effect of these disturbances and improve the tracking of the performers drum sticks leading to enhanced expressivity with the controller.

In a different application, tempo of musical performances of the North Indian Sitar string instrument, and the North Indian Tabla Drums is tracked using a Kalman filter based algorithm. We show how tempo estimates of musical performances may be improved by combining sensor and audio data. The tempo tracker takes noisy onsets from analyzed audio and sensor data as inputs, and outputs beat periods. Many instances of the algorithm are run in parallel each with different sensor inputs that are subsequently fused to obtain a more accurate tempo track than that of a single sensor. In the case of the Sitar, both the performer and the instrument were augmented with a variety of sensors including an inertial sensor and a thumb pressure sensor [1, 2]. In the case of the Tabla Drums, the performer wore a wrist mounted 3-D accelerometer unit known as the KiOm [3]. Various combinations of sensor and audio data were tested to improve accurate machine perception of tempo.

Chapter 2 discusses related work involving Kalman filtering, percussive gesture sensing and wearable sensors in the context of computer music. In Chapter 3 a variety of sensing systems used as experimental implements in this research are described. Chapter 4 and Chapter 5 discuss the design, implementation, and testing of a multi-model Kalman filter used for improved gesture tracking of the Radio Drum

system. Finally, Chapter 6 discusses the experiments and results of a Kalman filter based algorithm used for tempo tracking of a variety of sensor augmented musical performances.

Chapter 2

2 Related Work in Computer Music

In this chapter we will take a look at previously explored work relevant to this thesis: Kalman Filtering in the field of computer music, gesture based percussive oriented controllers, and wearable sensors used to acquire musically relevant gesture information.

2.1 Kalman Filtering in Computer Music

Compared to extensive use in the fields of control, aerospace, and economics, Kalman filtering has seen relatively minimal and only recent application in computer music. For an explanation of Kalman filter basics refer to section 4.2. With the earliest work appearing in 1993 [4], Kalman filtering has been used in applications for audio signal/speech restoration, auditory scene analysis and pitch tracking, beat tracking, gesture tracking and audio localization. All of these computer music problems share the commonality that noisy data or measurements coupled with a system model are used to develop a more accurate estimate of a dynamic, time-varying, hidden state.

2.1.1 Audio Restoration

Audio/speech signal restoration has application in telephony and more specifically UDP audio and speech transmission over the internet, where data packets may be dropped. The restoration of scratched or damaged audio stored on hard media such as compact disc, vinyl, and magnetic tape is also of interest.

The earliest work [4], concerns itself with the specific task of restoring a noise corrupted flute performance in real-time using a bank of linear Kalman filters running in parallel. This work models the sound of a flute as a sum of four sinusoids, the fundamental plus three main harmonics. A group of dynamic models track the most likely movement of these sinusoids against the true measured fundamental and three corresponding harmonics.

The work explained by Bari et al. [5] and originally conceived in [6], attempts to restore recordings of electronic music by modeling the audio as an autoregressive (AR) process and obtaining estimates of the slow varying AR parameters with an extended or non-linear Kalman filter. A second extended Kalman filter is used to detect and eliminate outliers such as pops and clicks. An AR/Kalman filter based approach is also used in [7] to improve quality of a transmitted speech signal. Work has been done by Cemgil [8] to restore missing data in an audio signal using a Kalman filter and a phase vocoder.

2.1.2 Tracking and Localization

Tracking refers to the problem of estimating the position of an object from noisy measurements. This is the problem addressed in this thesis with regards to the Radio

Drum. A related area, speaker localization, refers to finding the position of persons in a room based only on audio received through a microphone array.

Only one example was found where a Kalman filter was used for position tracking in a computer music application. A multi-user, polyphonic sensor stage environment that maps position and gestures of up to four performers to the pitch and articulation of distinct notes was developed by the MIT media lab. Kalman filtering was used to improve the position estimates of the performers, acquired with an ultrasonic tracking system [9].

Talker localization has application in video conferencing environments where a camera may be automatically steered towards the speaker. A Kalman filter based approach was attempted as early as 1997 where noisy position estimates are derived from a time-delay-based algorithm from a 16 microphone array [10]. The noisy estimates are processed through a Kalman filter and smoothed. This system is capable of tracking multiple speakers any of which may be moving. Two dynamic models, one for a static speaker and one for a speaker in motion, are needed to track the various motions possible in such an application. An Interacting Multiple Model is used to distinguish between the two models in order to provide the optimal filtering based on the speakers motion. Building on this, a more advanced approach was attempted in 2006 [11] which makes use of an extended Kalman filter and provides much more accurate results over the previous approach.

2.1.3 Auditory Scene Analysis and Pitch Tracking

As early as 1996, Kalman filter based methods have been used to track the frequency partials of audio signals; leading to the development of tools for auditory scene analysis, and more specifically, polyphonic pitch tracking.

Initial work in [12] attempted to track the most significant sound stream from a mixture. This work uses a non-linear Kalman filter to track the sounds fundamental pitch and associated harmonics. Similar work in [13] uses Kalman filtering to identify and transcribe multiple brass voices in a monophonic recording of a performance. A similar approach employing Kalman filtering is used in [14], however this works novelty lies in an improved partial peak detector. Recently, Cemgil [8] developed a graphical model approach to polyphonic pitch tracking. This work has the advantages of being computational efficient, able to track 'virtual' polyphonic pitch, where the fundamental and lower harmonics maybe missing, and extensible to broader auditory scene analysis problems.

2.2 Gesture Based Percussion/Conductor Controllers

In section we will describe related work in the area of drum-like or conductor-like gesture sensing. We group these two gesture types because of the overlap of data that we typically like to acquire from such interfaces: position, acceleration, periodicity or tempo, and strikes. The original work in this respect is the Radio Baton or Radio Drum which is a focus of this thesis. See Section 3.1 for a detailed overview of the Radio Drum.

In 1997 Teresa Marrin of the MIT media lab published work on the Digital Baton [15]. Used primarily by conductors, this interface incorporated 3-axis accelerometers, an external optical tracking sensor, and piezo-resistive strips to acquire 2-D position, 3-D orientation, and finger and palm pressure respectively. The system was incorporated into Paradiso's Brain Opera interactive installations [16]. The simpler, WorldBeat Baton [17] uses only infrared to track 2-D position and the Aobachi interface [18] augments traditional Japanese taiko drums sticks with 3-axis accelerometers and 2-axis gyroscopes for wireless tracking of drum strokes and other drum related gestures.

The 'air percussion' controller known as the Flock of Birds [19] uses electromagnetic sensing to track the position, tempo and virtual whacks of two drum sticks in a bounded region of sensing. Different from the Radio Drum system, the 'air percussion' controller does not provide a surface on which to strike the sticks, however, similar sensor noise plagues both systems. The Flock of Birds interface uses Linear Predictive Coding (LPC) to combine the predicted next sample of gesture with the next measured sample to arrive at a smoothed estimate [20]. Our work attempts to solve a similar problem using a model based statistical approach.

2.3 Wearable Sensors

Previous works on wearable or playable sensors in the academic community have mostly involved the use of accelerometers to obtain performer acceleration and tilt data. A survey of these designs can be found in the author's previous work [3]. Works

by Yeo [21] and Bowen [22], not mentioned in Kapur et. al [3] have also attempted accelerometer-based designs. Benbasat et al. [23], implemented the Sensor Stack, incorporating inertial, tactile, and sonar distance sensing into a small modular unit that was embedded in a shoe.

This year saw the advent of more complicated wearable wireless inertial sensor designs incorporating accelerometers, gyroscopes, and magnetometers. Aside from the WISP [2], used for the tempo tracking experiments discussed in Chapter 6 of this thesis, two new designs were unveiled at the New Interfaces for Musical Expression conference in New York. These are the Celeritas, an inertial sensor used for interactive solo or group dance performances [24] and Ircam's gesture follower interface [25].

Chapter 3

3 Experimental Implements

In this chapter we will outline the various data acquisition systems that were used for the two branches of research relevant to this thesis: Position Tracking and Tempo Tracking. First we will discuss the history and design of the Radio Drum system, used to acquire 3-D position data of a drum stick in the space over a capacitive sensing surface. We will then discuss the implements used for the tempo tracking experiments. These include the audio data collected from either a microphone or a piezo electric pickup, a wearable midi-enabled 3-D accelerometer named the KiOm, a wearable wireless inertial measurement unit named the Wireless Inertial Sensing Package (WISP), and a multi-modal sensor augmented hyper-instrument named the ESitar. The current Radio Drum system, the KiOm, the WISP and the ESitar were all designed and built at the University of Victoria.

3.1 The Radio Drum

The Radio Drum, originally known as the Radio Baton, is a 3 dimensional musical controller that tracks the x, y, and z position, z velocity and detects surface whacks of one or two drum sticks over its surface. Originally designed and built at Bell Laboratories in the 1980's to be used as a 3 dimensional mouse, the Radio Drum has now evolved to become a pioneering instrument in computer music performance.

Tracking of the sticks are performed through capacitive sensing. Drum stick tips are coiled with conducting wire through which a driving signal of 20-30 KHz is sent. Four antennas on the surface of the Radio Drum output varying signal strength depending on the position of the sticks.

3.1.1 History of the Radio Drum or Radio Baton

Max Mathews adopted the original Radio Baton from Bell Labs and adapted it for artistic use [26]. The sensing surface was comprised of a “backgammon board” antenna configuration. The signal strengths out of the four corners were used to compute a rough estimate of the sticks position. Figure 3.2 shows this original configuration [27].

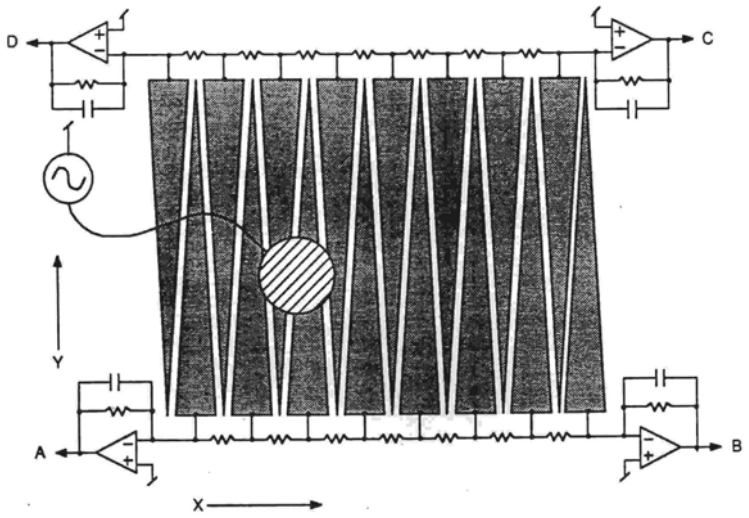

Figure 3.1. *Original "Backgammon" configuration of antenna*

Figure 3.2 shows another configuration of the stick and surface in which the signal strength of 5 antenna plates were used to acquire an estimate of the sticks position.

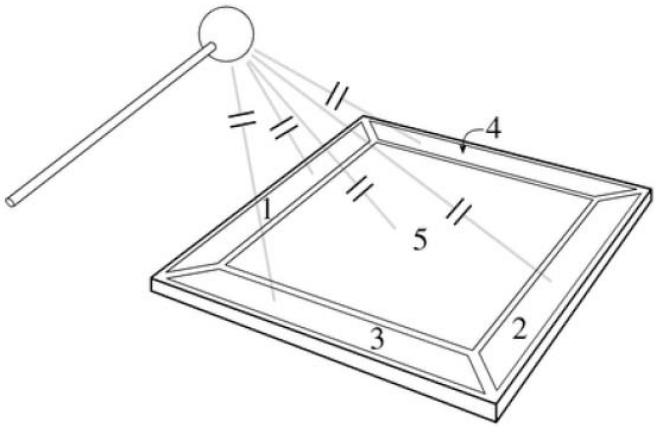

Figure 3.2. *Older Radio Baton stick and surface with 5 antennas*

Unacceptable non-linear behavior across the sensing surfaces and inaccurate position estimates led to the most recent antenna geometry design, shown in Figure 3.3.

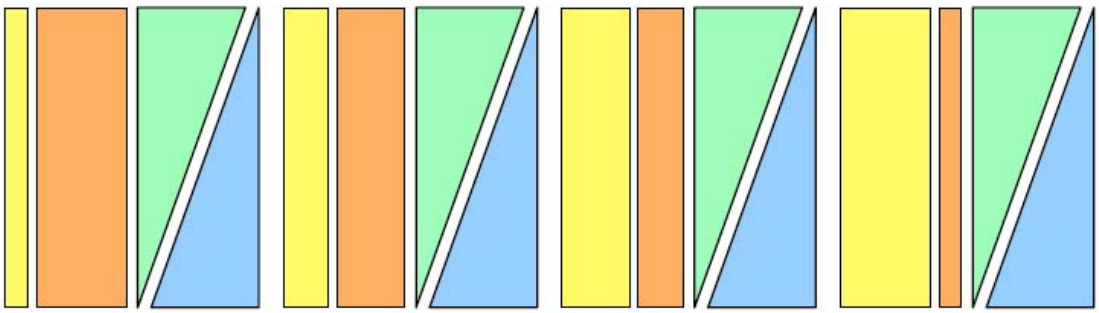

Figure 3.3. *Most recent "backgammon/rectangle" Radio Drum antenna geometry (Image Courtesy of Ben Neville)*

This “backgammon/rectangle” geometry decouples x and y position estimation, reducing non-linearity and antenna noise, thus improving the 3-D position estimate. The two signal outputs of the orange and yellow rectangular strips are combined in a geometric equation to obtain the x coordinate. Similarly, the y coordinate is obtained by geometrically combining the signals strengths of the two triangular strips. The z coordinate is obtained by summing the outputs of all four antennas.

Traditionally, a custom made hardware unit was designed to both transmit the driving signals for each stick and acquire the antenna output to calculate position and detect surface whacks. Position of the stick was output every 50ms and surface whacks were detected when a stick passed below a minimum z position threshold. Along with the detection of a whack, a velocity was also calculated as a function of the distance the stick traveled below the z position threshold. However, due to noise in the system and the nature of the position based whack algorithm, this estimate of whack velocity would often be inconsistent with perceived velocities performed by a musician [28]. Difficulties associated with troubleshooting and reprogramming of the hardware embedded algorithms stagnated any further evolution of the Radio Drum system. This led to the development of the current Radio Drum system known as the Audio Input Drum, a term coined by its creator Ben Neville.

3.1.3 The Audio Input Drum

The Audio Input Drum system alleviates the inconveniences of the original Radio Drum hardware by giving access to the 4 raw antenna signals in high-level software. This is achieved with a commercially available audio interface capable of sampling at 64 KHz or greater at 4 pre-amplified inputs connected to a modern desktop or laptop computer running either Windows or Mac OSX. Outputs of the audio card transmit a sinusoidal carrier signal to each stick, typically of 30 kHz and 26 kHz. The emitted sinusoidal carrier signals are amplitude modulated by a performer's gestures, received

by the 4 antennas, amplified at the audio interface inputs by +60dB and digitized. The audio interface of choice is the Fireface 800 by RME[1].

In real-time software, the digitized raw antenna signals are separated via 2 biquadratic bandpass filters centered at the two stick carrier frequencies and demodulated to recover the gesture manipulated antenna signal strengths. The demodulation is done by downsampling each raw antenna signal by 32. The phase of each carrier wave is adjusted to ensure that the peaks of the raw antenna signals are being picked, providing the greatest signal to noise ratio. All processing of the digitized raw antenna data is done in real-time by the commercially available Max/MSP/Jitter software[2], developed by cycling74 for computer music and video applications. Max/MSP/Jitter provides an intuitive graphically based language ideal for rapid prototyping of audio and video related ideas and solutions. The move from hardware based processing to software enabled a deeper analysis of the raw antenna signals leading to an improved velocity and acceleration based whack detection algorithm. For a more in depth explanation of the Audio Input Drum see Ben Neville's Masters Thesis [28]. Throughout the rest of this thesis we will still refer to the new and improved Audio Input Drum as the Radio Drum.

3.1.4 Non-Realtime Analysis and Design

For all the initial Kalman filtering work described in this thesis, Radio Drum antenna data was recorded at a sampling rate of 96 KHz with 24 bit resolution. All gesture

[1] http://www.rme-audio.com
[2] http://www.cycling74.com

analysis and Kalman filtering is performed in non real-time with the Matlab[3] computing software. The incoming digital data is recorded into 4 channel AU NeXT audio files through Max/MSP. In Matlab, the data from each antenna is filtered through a biquad bandpass filter with a Q=6 and centre frequency of 30 KHz. This is to band limit the signal and isolates the carrier wave. Demodulation is performed by downsampling the band limited signal by 32, being sure to start on a carrier peak. The band-limiting ensures that no aliasing will happen. The sample rate is reduced to 96/32=3 KHz. The x, y, and z relative position voltages are then calculated from the four demodulated antenna signals. See equation 4.10.

3.2 Audio

Audio data of the Tabla's and the Sitar was needed for experiments involving tempo tracking of musician performance. Audio of the Tabla performance was recorded with a Shure SM-57 microphone sampled at 44.1 KHz at the pre-amplified inputs of a Motu audio interface. The Sitar audio was recorded by placing a custom built peizo electric pickup on the bridge of the instrument. The Sitar audio was also sampled at 44.1 KHz by the Motu audio interface. The Root Mean Square (RMS) of every 512 audio samples was calculated and used for performance onset detection. The onsets were used as input into the Kalman filter based tempo tracking algorithm. RMS was calculated according to the equation below.

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} \qquad (3.1)$$

[3] http://www.mathworks.com

3.3 KiOm

The KiOm [3], shown in Figure 3.4, was designed by Ajay Kapur of the University of Victoria. The design of this wearable sensor is described in this section. The KiOm was placed on the right hand wrist of a Tabla player to collect 3 dimensional acceleration data of the performer's hand. The center piece of this controller is the Kionix KXM52-1050[4] three-axis accelerometer. The three streams of analog gesture data from the sensor is read by the internal ADC of a PIC Microchip 18F2320[5]. These streams are converted to MIDI messages for use with any musical hardware/software synthesizers and programs.

The dimensions of the KiOm are 3 inches by 3 inches by 2 inches. It weighs approximately 100 grams. A majority of the space and weight of the device is due to the 9-volt battery used to power the MIDI out port at 5-volts. The KiOm also has a power switch with LED to allow the user to know when the device is on/off. Two buttons are also built in for users to have control of different modes/controls/parameters of their compositions.

[4] http://www.kionix.com
[5] http://www.microchip.com

Figure 3.4. *KiOm Encasement and inside circuit board*

Onsets were derived from the acceleration data and used as input into the tempo tracking algorithm

3.4 WISP

The Wireless Inertial Sensor Package (WISP) [2] is a miniature Inertial Measurement Unit (IMU) designed by Bernie Till at the University of Victoria; Specifically designed for the task of capturing human body movements. It can equally well be used to measure the spatial orientation of any kind of object to which it may be attached. Thus the data from the WISP provides an intuitive way for a performer to control an audio and video synthesis engine. The performer is free to move within a radius of about 50m with no other restrictions imposed by the technology such as weight or wiring.

The WISP is a highly integrated IMU with on-board DSP and radio communication resources. It consists of a triaxial differential capacitance accelerometer, a triaxial magnetoresistive bridge magnetometer, a pair of biaxial vibrating mass coriolis-type rate gyros, and a NTC thermistor. This permits temperature-compensated measurements of linear acceleration, orientation, and

angular velocity. The first generation prototype of WISP, shown in Figure 3.5 next to a Canadian two-dollar coin, uses a 900 MHz transceiver with a 50Kb/s data rate. With a volume of less than 13cm^3 and a mass of less than 23g, including battery, the unit is about the size of a largish wrist watch. The WISP can operate for over 17 hours on a single 3.6V rechargeable Lithium cell, which accounts for over 50% of the volume and over 75% of the mass of the unit.

The fundamental difference between the WISP and comparable commercial products is that the WISP is completely untethered (the unit is wireless and rechargeable) in addition to being far less expensive. All comparable commercial products cost thousands of dollars per node, require an external power supply, and are wired. A wireless communication option is available in most cases, but as a separate box which the sensor nodes plug into. As can be seen in Figure 3.5, the small size and flat form-factor make it ideal for unobtrusive, live and on-stage, real-time motion capture.

Figure 3.5. *The WISP in contrast to a Canadian two dollar coin*

Figure 3.6 shows an end-to-end block diagram of the system. Although only one WISP is shown in the figure, the system uses time-division multiplexing to allow any number of WISPs to coexist on a single radio channel, subject only to aggregate data rate limitations. A channel can accommodate 4 WISPs, each sampling at a rate of 80Hz or 8 WISPs at 40Hz and so on.

The windows-based visual basic WISP application sends out the roll (rotation about x), pitch (rotation about y) and yaw (rotation about z) angles from the sensing unit over the Open Sound Control protocol [29]. These angles are commonly used in aerospace literature to describe, for example, the orientation of an aircraft [30].

The WISP was mounted on the upper wrist area of a sitar player to capture subtle orientation information of the performer's strumming hand. Onsets derived from the WISP data were used for tempo tracking of the sitar performance rather than the KiOm because the movements of a sitar player's upper wrist are more subtle than that of a tabla hand drummer's gestures.

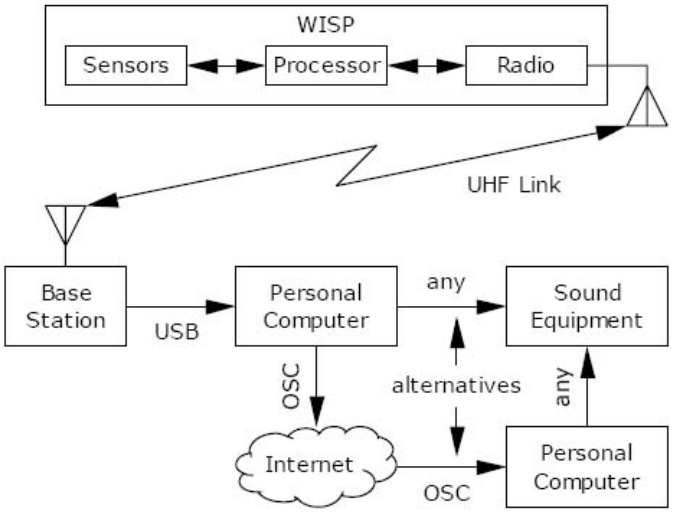

Figure 3.6. *WISP end to end block diagram*

3.5 Electronic Sitar (ESitar)

The sitar is the prevalent stringed instrument of North Indian classical music traditionally employed to perform ragas. It is distinguished by its gourd resonating

chamber, sympathetic strings, and curved frets that allow the incorporation of microtones into melodic phrasing.

The current version of the *ESitar* was designed and built by Ajay Kapur at the University of Victoria. Based on his older version [1], the newer *ESitar* uses methods and theory obtained from three years of experience of touring and performing. The first step was to find a sitar maker in India to custom design an instrument with modifications to help encase the electronics. One major change to the traditional sitar was the move to worm-gear tuning pegs for the six main strings. This allows the sitar to remain in tune through all the intense bending during performance, and makes the instrument more accessible to western music students. A second tumba (gourd) was also created to encase a speaker to allow for digital sound to resonate through the instrument as well as serve as a monitor for the performer. The bridge, traditionally made of ivory, and then deer bone was upgraded to black ebony wood from Africa, which generates an impressive clear sound and requires less maintenance. The frets themselves were pre-drilled to allow easy installation of the resistor network.

The newer *ESitar* made a platform change from the Atmel[1] to the PIC[2] microcontroller, based on the mentoring of Eric Singer the creator of the League of Electronic Music Urban Robots (LEMUR). A massive improvement was encasing the microchip, power regulation, sensor conditioning circuits, and midi out device in a box that fits behind the tuning pegs on the sitar itself. This reduces the number of wires, equipment, and complication needed for each performance. This box also has two potentiometers, six momentary buttons, and four push buttons for triggering and setting musical parameters.

The *ESitar* uses a resistor network for fret detection. Military grade resistors at 1% tolerance were used in this new version for more accurate results. Soldering the resistors to the pre-drilled wholes in the frets provided for a more reliable connection that does not have to be re-soldered at every sound check. A force sensing resistor used to obtain thumb pressure proves to be useful in obtaining rhythmic data and pluck direction (right image of Figure 3.7). There is a 3-axis accelerometer embedded in the controller box at the top of the neck (left image of Figure 3.7), to capture ancillary sitar movement, as well as serve as yet another means to control synthesis and audio effect parameters.

Figure 3.7. *ESitar controller head, neck and thumb sensor*

Onsets of the performance were obtained from the thumb sensor and fret data. These onsets were input into the tempo tracking algorithm along with the onsets generated from the sitar audio RMS data.

Chapter 4

4 Improved Gesture Tracking of the Radio Drum Part 1: Preliminary Design of the Kalman Filter

In this chapter we describe the preliminary design issues related to developing a multi-model Kalman filter used to improve position tracking of the Radio Drum 3 dimensional gesture-based musical controller. The goal of our work is to accurately track Radio Drum gestures through noisy measurement signals. We begin this chapter by explaining the motivation for improving the tracking of a Radio Drum stick's position. We then go on to introduce the Kalman filter algorithm and discuss the development of a measurement model and a dynamic model of the Radio Drum system, both of which are needed to fully specify a Kalman filter. We end this chapter with a brief conclusion.

A measurement model describes the relationship between the unknown quantities or parameters of a system with the systems known measurements [31]. An understanding of the noise, specifically the variances, covariances, and autocovariances of the Radio Drum antennas and calculated positions with respect to time and position is paramount to developing a measurement model. A thorough investigation of the time and position dependant noise characteristics of the Radio Drum system is performed. From this work, a method for obtaining a single covariance matrix, used in the measurement model, is described. A dynamic model describes the evolution of our state, in our case the x, y and z positions, over time.

Our model is based on a simple kinematics model of motion in space. With the specification of both the measurement and dynamic models the Kalman filter is able to distinguish between system noise and performer gesture to provide an improved track of the Radio Drum stick.

4.1 Motivation for Kalman Filtering

Figure 4.1 shows a 3 dimensional plot of the Radio Drum stick location while it is rapidly moving over the surface. Notice how the track is rather fuzzy or noisy. Kalman filering is the ideal algorithm to track such motion through noisy measurements. There are many reasons as to why one may want to have a smoother track of position over the Radio Drum surface. A popular way to use the Radio Drum is to map out the surface into several rectangular or square regions. Each region, when whacked or hovered over by the stick, may trigger a different sound sample or control an infinite number of other parameters in the virtual space. Position ambiguity due to noise in the signal may cause unintended regions to trigger causing the wrong sample to play or wrong parameter to change. The number of control regions one may define and intentionally trigger is proportional to the uncertainty in the position estimate.

More certainty in the z position of the stick also leads to a higher degree of expressivity when whacking the surface of the Radio Drum. The current whack detection algorithm developed by Ben Neville analyzes the velocity and acceleration of the z position to determine when a whack is triggered. A velocity threshold is set based on the variance of the stationary noise of the antennas. In order for the whack

detection algorithm to consider any candidate for a whack, the z velocity must go below this threshold. Kalman filtering of our stick position leads to a smaller whack threshold, enabling softer whacks to be detected; whacks that would otherwise be buried in noise. Section 5.3.3 will discuss the improvements on the whack detection in more detail.

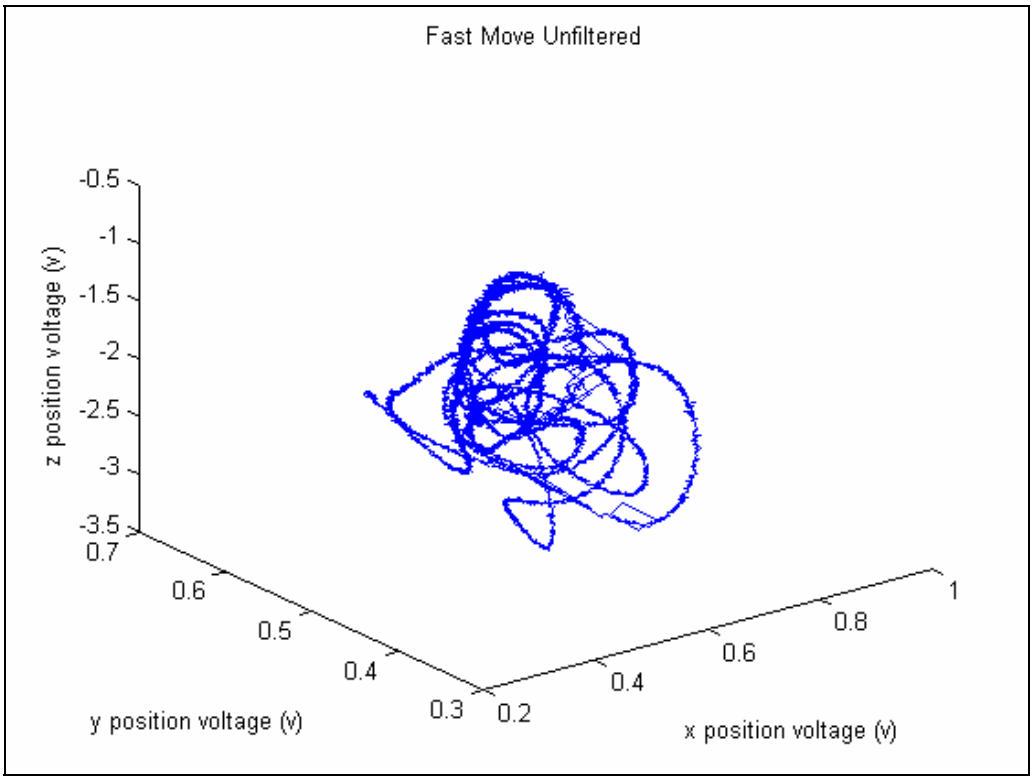

Figure 4.1. *3D track of unfiltered Radio Drum stick under fast motion*

4.2 Introduction to Kalman Filtering

The Kalman Filter is an optimal recursive linear estimator. With prior knowledge of the system and measurement devices, all available measurements are processed to estimate the desired unknown parameters. The Kalman Filter processes the measurements in a linear fashion minimizing the error between the estimated

parameters and the actual parameters [31]. The Kalman filter algorithm, named after its inventor Rudolph Kalman, was developed in the 1960's and used by NASA to estimate the trajectories for the Apollo space program [32]. Since then it has been used widely in many fields.

The various parts of the Kalman Filter are derived from a measurement model and dynamic model of our system. Both of these models have to be completely specified before a Kalman Filter can be designed. To relate the known measurements of the system to the unknown hidden parameters a linear model of this form is used.

$$Z(k) = H(k)X(k) + V(k) \qquad (4.1)$$

Where *Z(k)* denotes the measurement vector, *H(k)* denotes the observation matrix, *X(k)* denotes the unknown parameters; in our case voltages describing the relative position of the stick; and *V(k)* the measurement noise. Matrix *R(k)* is defined as the covariance of *V(k)*.

A dynamic model describes how the unknown parameters or state of the system changes from one time instant to the next.

$$X(k+1) = \Phi(k)X(k) + \Gamma U(k) + W(k) \quad (4.2)$$

Where *X(k+1)* and *X(k)* represent the state from consecutive time instants. The state transition matrix *Φ(k)* describes how the state of the system evolves over time. *U(k)* and Γ are the input, and input control matrices respectively. These allow for the model to account for external input. *W(k)* represents the dynamic noise matrix. The Kalman Filter assumes a zero mean, Gaussian white noise process. *W(k)* allows for random disturbances to the model. The covariance's of the disturbances are contained

in the matrix *Q(k)*. Table 4.1 describes the matrices used in the Kalman Filter that haven't already been mentioned.

Matrix	Description
X(k|k)	The current filtered state estimate, derived from X(k|k-1), K(k), and Z(k).
X(k|k-1)	The predicted state estimate, derived from X(k|k), Φ(k), Γ(k) and U(k).
P(k|k)	The estimated error covariance. This can be seen as the estimated accuracy of X(k|k).
P(k|k-1)	The predicted estimate error covariance, derived from Φ(k), Q(k), and P(k|k).
K(k)	The Kalman gain, derived from H(k), P(k|k-1), R(k)

Table 4.1. *Other matrix quantities of a Kalman filter*

4.2.1 Basics of Kalman Filter Function

The Kalman Filter works in two distinct phases: Prediction and Correction. Using the dynamic model, the state *X(k|k-1)* is predicted from the previous estimate of the state, *X(k-1|k-1)*. The prediction is done according to.

$$X(k \mid k-1) = \Phi(k)X(k-1 \mid k-1) + \Gamma(k)U(k) \qquad (4.3)$$

The correction phase then weighs the predicted state with the difference between the current measurement and the predicted state to come up with the next state estimate.

$$X(k \mid k) = X(k \mid k-1) + K(k)Z'(k) \qquad (4.4)$$

Z'(k) represents the difference between the current measurement vector, *Z(k)* and the predicted estimate vector *X(k|k-1)*. This is known as the innovation. The gain matrix

K(k) decides how much weighting to give to the innovation. *K(k)* will be small if the measurement noise, *R(k)*, of our system is large and *P(k|k-1)*, our accuracy of predicted estimate, is small. Intuitively, the gain matrix will adjust more weighting to the prediction if our measurements are noisy. On the other hand, if we are not so confident in our prediction, our measurements will be favoured. The Kalman gain is shown below. *S(k)* is the covariance of the innovations sequence shown below**.**

$$Z'(k) = Z(k) - H(k)X(k \mid k-1) \qquad (4.5)$$

$$S(k) = E\{Z(k)Z'(k)\} = H(k)P(k \mid k-1)H(k)'+R(k) \quad (4.6)$$

$$K(k) = P(k \mid k-1)H^T(k)S^{-1}(k) \qquad (4.7)$$

$$P(k \mid k-1) = \Phi(k)P(k \mid k)\Phi'(k) + Q(k) \qquad (4.8)$$

$$P(k \mid k) = (I - K(k)H(k))P(k \mid k-1) \qquad (4.9)$$

4.3 Measurement Model for the Radio Drum

In this section we will define the various quantities of the measurement model.
For a measurement model of the Radio Drum we must define *Z(k)*, the measurements, *X(k)* the state, *V(k)* and *R(k)*, the measurement noise and its covariance matrix, and H(k), the observation matrix.

Z(k) is a 3 by 1 column vector containing the raw measured *x*, *y* and *z* coordinates of the stick within the Radio Drum boundaries. The coordinates *x*, *y*, and *z* are expressed in voltages derived from the demodulated antenna signals. The

translations from antenna to *x*, y, and z positions are shown below. It is important to have independent measurements for *Z(k)* at each time step. If this is not so, once the measurement model is coupled with a time based dynamic model describing the movements of the Radio Drum performer, the Kalman filter system will confuse measurement noise with performer gesture. This problem is discussed later in the context of results.

$$x = \frac{a1}{a1 + a4}$$

$$y = \frac{a2}{a2 + a3}$$

$$z = a1 + a2 + a3 + a4 \qquad (4.10)$$

The state vector *X(k)* contains the x, y, and z position, velocity, and acceleration voltages fully describing the motion of a stick at any time instant.

$$X(k)=[x\ dx/dt\ d^2x/dt^2\ y\ dy/dt\ d^2y/dt^2\ z\ dz/dt\ d^2z/dt^2]^T \qquad (4.11)$$

H(k) is a 3 by 9 matrix that relates our hidden state *X(k)* to the measurements, *Z(k)*. When *H(k)* and *X(k)* are multiplied only the positions remain. Table 4.2 shows the entries of *H(k)*.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Table 4.2. *Observation matrix, H(k)*

The last term in Equation 4.1, *V(k)*, represents the noise that is added to the Radio Drum's true position to give the measured position. We can characterize this noise across the three position voltages through the covariance matrix of *V(k)*, *R(k)*. A covariance matrix will have the variances of the three position voltages x, y, and z along its diagonal elements: (1,1), (2,2), and (3,3) respectively and the covariances of x and y, x and z, and y and z at (1,2), (1,3), and (2,3) respectively. To understand the nature of the noise and obtain a covariance matrix it is important for each set of measurements, *Z(k)*, to be independent in time. Furthermore, our covariance matrix must not be position dependant. Meaning that it must be constant all over the Radio Drum surface, or at least with in a known bound. Unfortunately, both of these assumptions are not true of the Radio Drum system. Next we will show that our noise is a non-white process and then show that the covariance is in fact dependant on stick position.

4.3.1 Problems with Getting Independent Measurements

The three figures below show autocovariances of the four demodulated antenna signals with the stick held stationary at the centre of the x-y plane at heights of 0, 8, and 16 cm respectively for intervals of 30 seconds. The equation below defines the autocovariance of a sequence X_i with mean u, k is the lag index.

$$\gamma(k) = E[(X_i - u)(X_{i-k} - u)] \qquad (4.12)$$

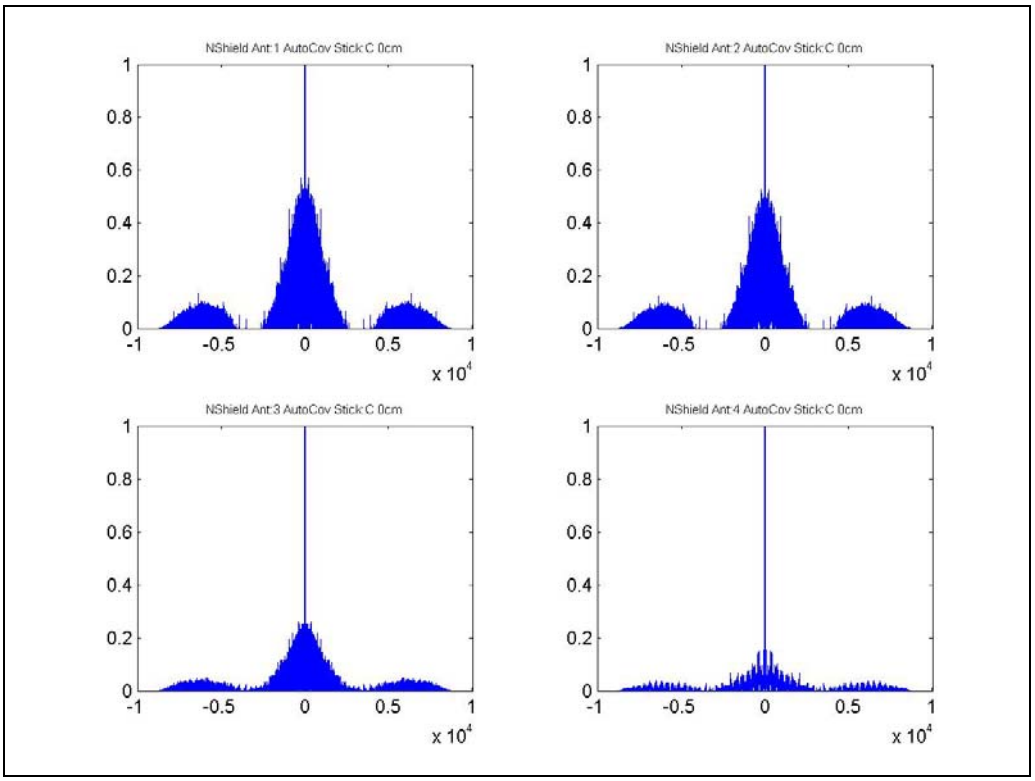

Figure 4.2. *Autocovariance of demodulated antenna signals, stick at centre on surface (z=0)*

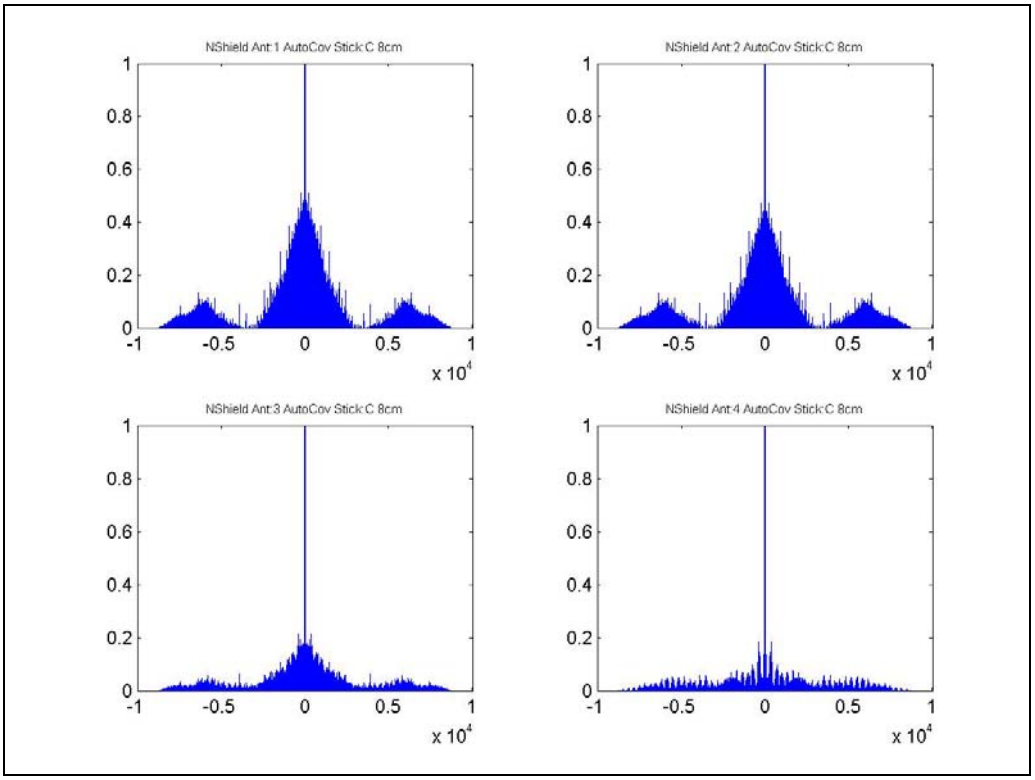

Figure 4.3. *Autocovariance of demodulated antenna signals stick at centre z=8cm*

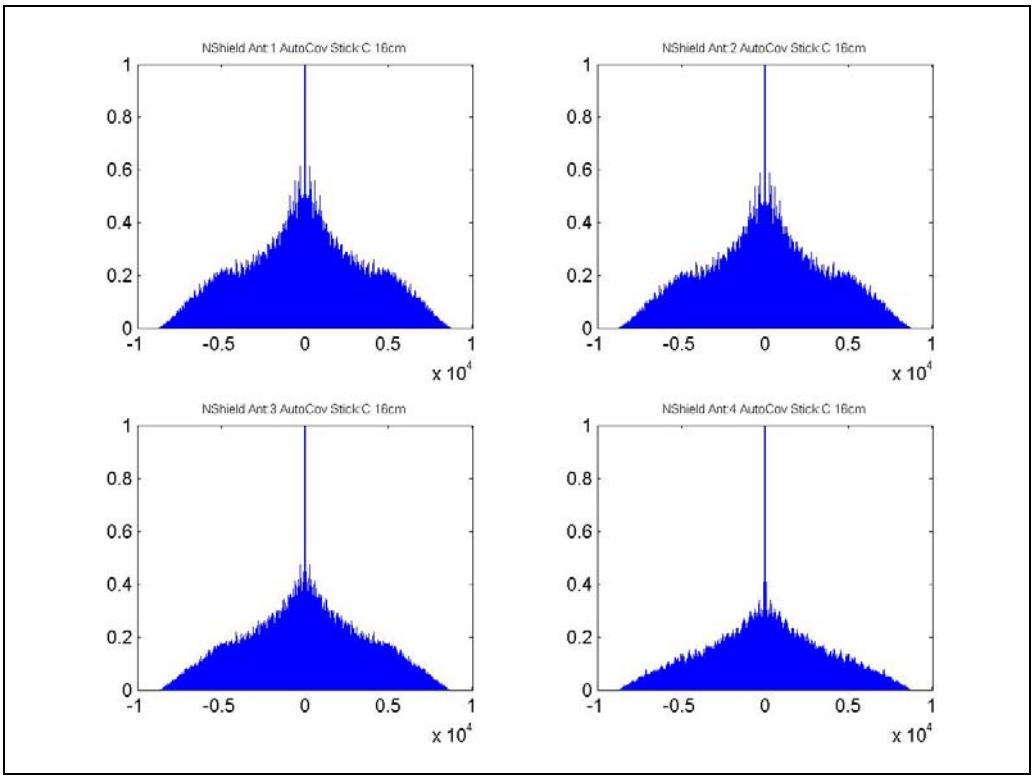

Figure 4.4. *Autocovariance of demodulated antenna signals stick at centre z=16cm.*

Since a periodicity in the autocovariance function proves the existence of periodicity in the time domain signal we can conclude that the demodulated antenna signals have some underlying low frequency periodicity [33]. This periodicity is clear for all antennas in the 0cm case and antenna 1 and 2 in the 8cm case. Although periodicity is not obvious in the 16cm case, there is a high level of correlation in the signal up to 5000 lags or 5000/3000=1.67 seconds. With the observed periodicity and/or large correlation of signal noise in the above figures it is apparent that we are not able to collect independent consecutive measurements for a measurement model of the Radio Drum. Likely candidates for the source of this periodicity in the noise are the Macbook Pro laptop computer and the Fireface 800 audio interface and other external interference. A variety of tests were performed to discover the cause of this problem.

4.3.1.1 Where are the low frequency oscillations coming from?

The anechoic electronically shielded chamber in the Engineering Lab Wing at the University of Victoria was used to perform a variety of experiments to single out the source of periodic interference. To single out the effect of the laptop computer, antenna data of a stationary stick at the centre of the antenna surface was acquired for 30 seconds with the antenna surface electronically shielded from the laptop. This was achieved by placing the stick and the antenna inside the anechoic chamber while the laptop and audio interface acquired the signals outside the chamber. Figure 4.5 shows that the antenna signals still exhibit periodicity despite the absence of a laptop and audio interface.

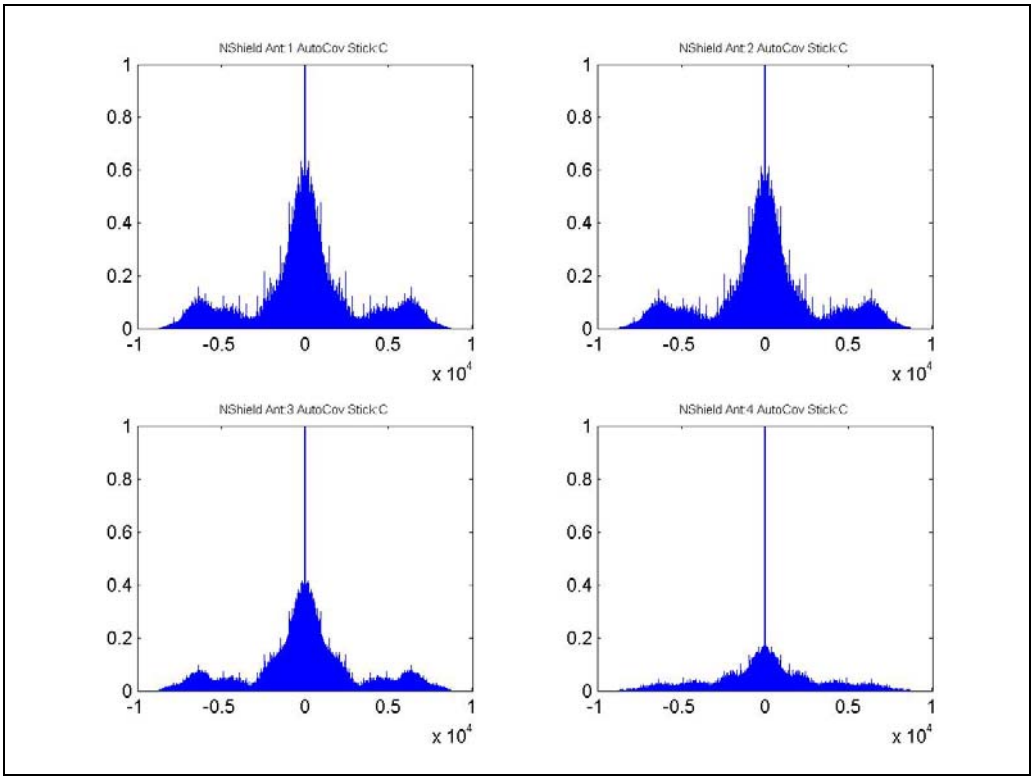

Figure 4.5. *Autocovariance of laptop seperated demodulated antenna signals, stick at centre on surface.*

To single out the effect of the external environment, the same experiment was performed with the entire Radio Drum system inside the shielded chamber. Figure 4.6 shows that despite the effect of external interference, the noise still exhibits periodicity.

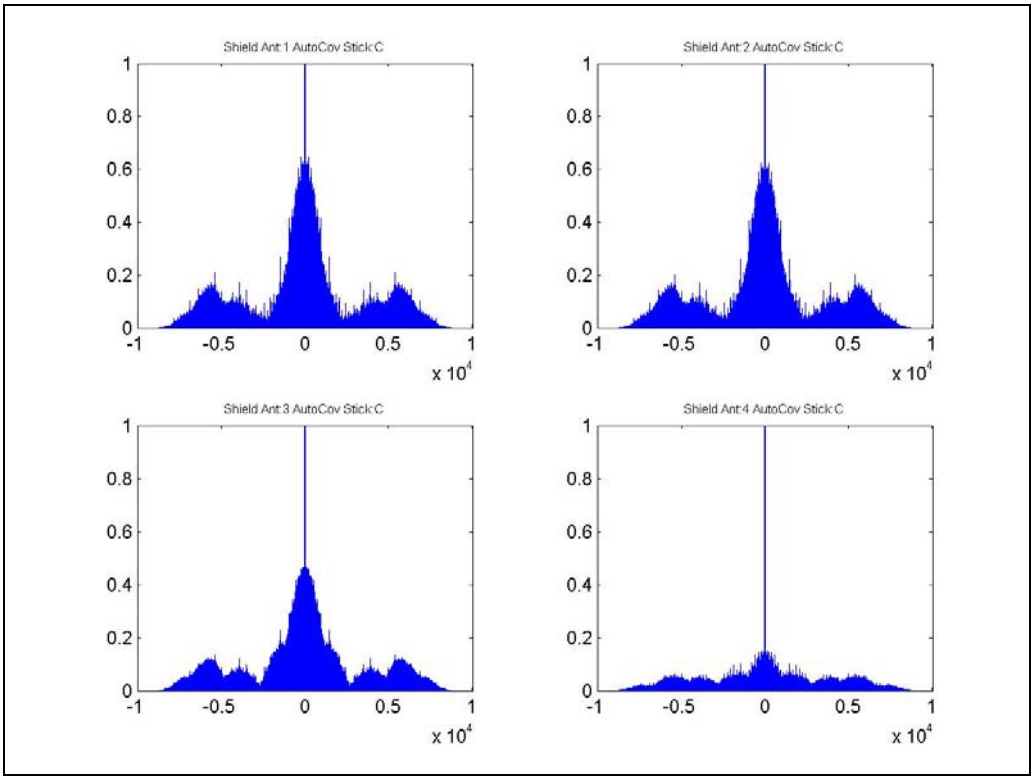

Figure 4.6. *Autocovariance of shielded demodulated antenna signals, stick at centre on surface.*

The Fireface audio interface, independent of the Radio Drum surface and stick, was also tested. To achieve this, a 30 KHz carrier wave from output 1 of the Fireface was attenuated by a simple voltage divider circuit and looped back into each of the

Fireface inputs: 7, 8, 9, and 10. The voltage divider was designed in such a way to attenuate the carrier by 82dB. This was chosen to mimic the attenuation of the carrier by the Radio Drum system with the stick on the surface at the centre position. At the surface of the drum at the centre position the attenuation of each antenna is -22dB. This includes the +60dB gain at the Fireface inputs. Figure 4.7 shows the autocovariance of three seconds of the looped attenuated demodulated carrier wave. Inputs 7, 8, and 9 shows the same correlation, at around 5000 lags, as the autocovariances of the Radio Drum antennas do. Input 7 shows the most obvious correlation at around 5000 lags while input 8 and 9 are subtle. Input 10 shows characteristics closer to white noise.

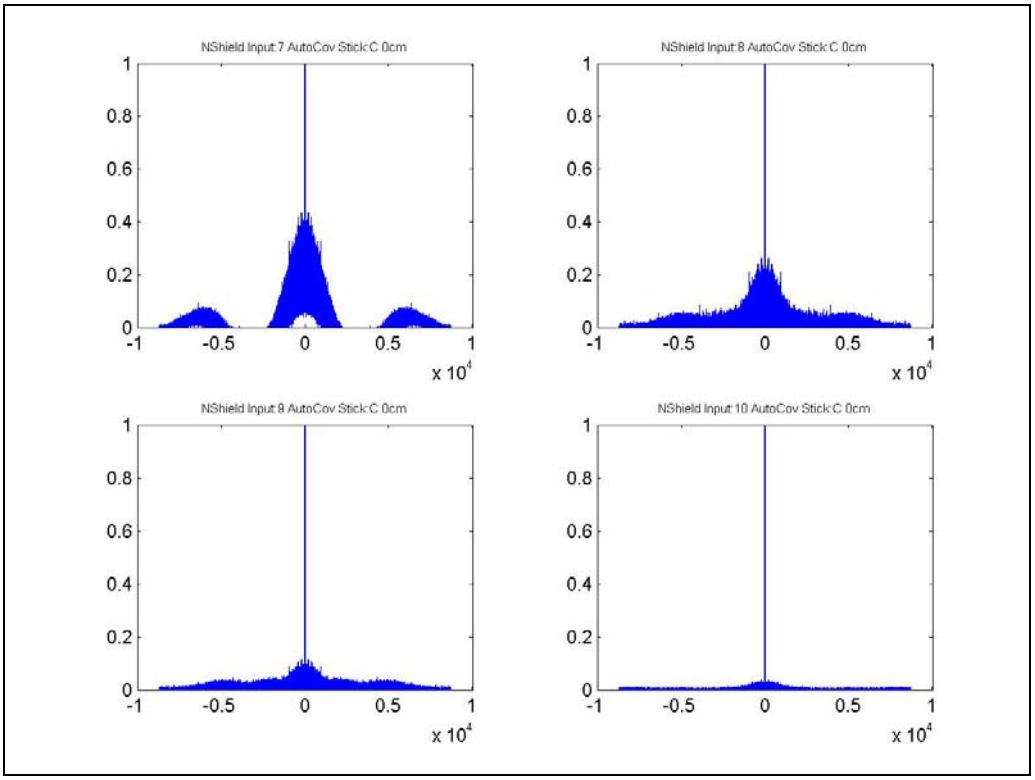

Figure 4.7. *Autocovariance of looped attenuated demodulated carrier signal*

This is an interesting result because it proves that the low frequency oscillations are happening independent of the Radio Drum surface and stick. It is also interesting to

note that inputs of the Fireface show decreasing autocovariances from input 7 through 10. And since the voltage to each of the inputs was uniform, this leads us to believe that the inputs of the Fireface are not. Figure 4.8 shows the power spectrum of one second of antenna data captured with the stick resting at the centre of the surface. The FFT was performed on one second of data with a window size of 96K samples giving us a resolution of 1 Hz. Typically, antennas 1, 2, 3, and 4 connect to the Fireface inputs 7, 8, 9, and 10 respectively. However for this experiment, the connections were reversed; antennas 4, 3, 2, 1 connected to inputs 7, 8, 9, and 10. Similar to the plots in Figure 4.7, inputs 9 and 10 of the Fireface show lower noise levels.

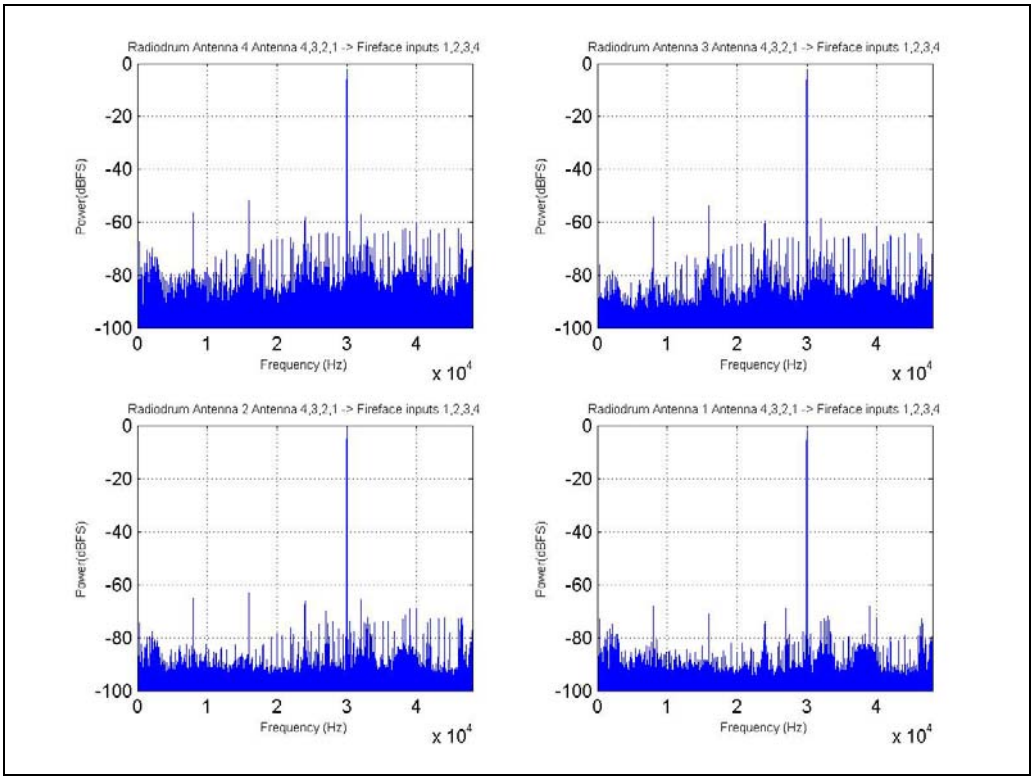

Figure 4.8. *Power spectrum of Radio Drum antennas reversed at the inputs to the Fireface audio interface. Shown clockwise: Input 7, 8 ,9, 10*

With these results we are led to believe that our measurement model is dependant on the connection of antenna outputs to audio interface inputs. To prove that the Firface audio card is in fact introducing a non-white noise process, another audio interface was tested.

The Tascam FW-1804 audio inteface was setup with the Radio Drum system to mimic the Radio Drum system with the Fireface. The same 30 KHz carrier wave was sent to the stick from the left output channel of the Tascam at full output gain. Antenna's 1, 2, 3, and 4 were connected to inputs 1, 2, 3, and 4 of the Tascam at full input gain at a sample rate of 96 KHz. The attenuation of the carrier wave with the stick at the centre of the Radio Drum resting on the surface was -22dB. This is the same attenuation seen with the Fireface.

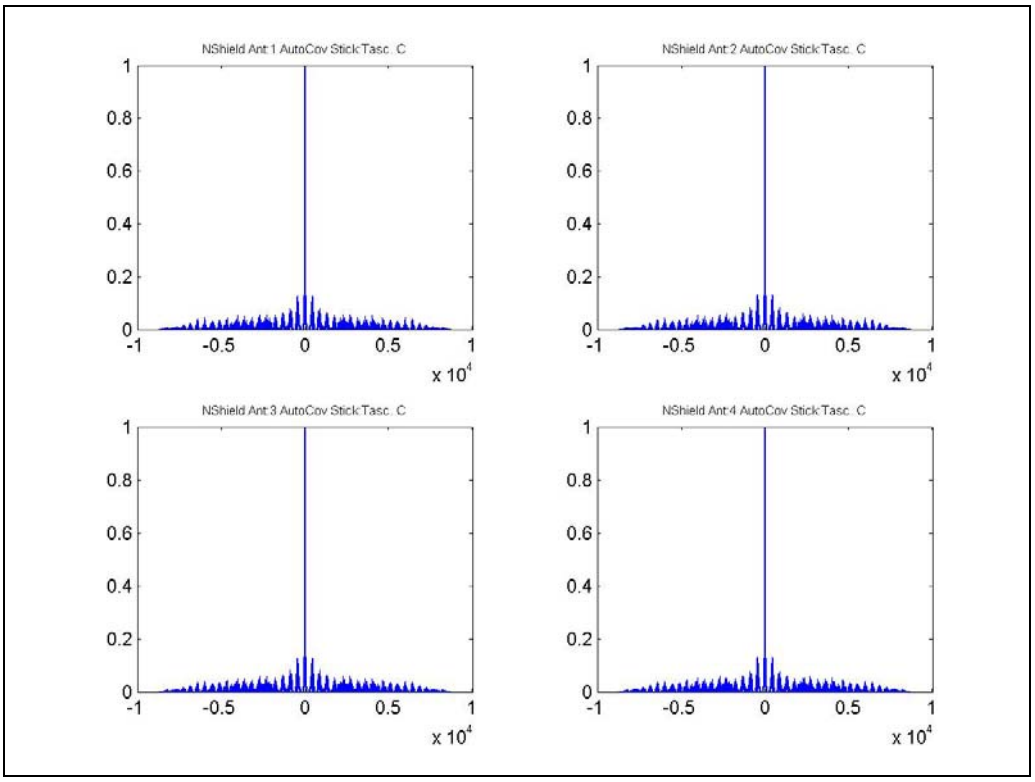

Figure 4.9. *Autocovariance of demodulated antenna signals, stick at centre on surface. Taken with Tascam FW-1804.*

Figure 4.9 shows that the Tascam audio interface does not exhibit the same side lobes at 5000 lags as is true with the Fireface interface. To complete our comparison to the Fireface, the looped, attenuated carrier test was repeated with the Tascam. The exact same attenuation circuit and length of data was used. Comparing to Figure 4.7, Figure 4.10 does not show the same demodulated antenna correlation at around 5000 lags. This once again reinforces that the low frequency oscillation in the demodulated antenna data is caused by the Fireface audio card.

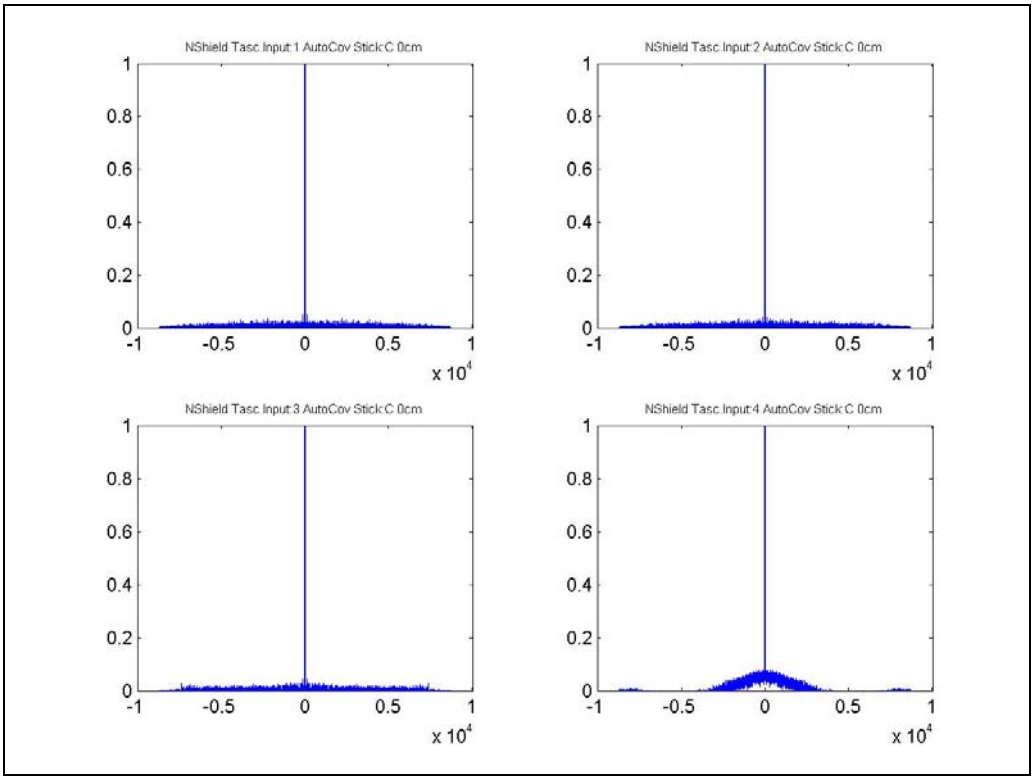

Figure 4.10. *Autocovariance of looped attenuated demodulated carrier signal. Taken with Tascam.*

For comparision sake power spectral plots of antennas 1, 2, 3, 4 connected to inputs 1, 2, 3, 4 of the Tascam audio interface are shown in Figure 4.11. The levels of noise

at each input are similar. However, the noise in the Tascam is greater than the Fireface. This is expected since the Tascam is less than half the cost of the Fireface. In chapter 5 we will show Kalman filter tracking results on gestures acquired using the Tascam interface.

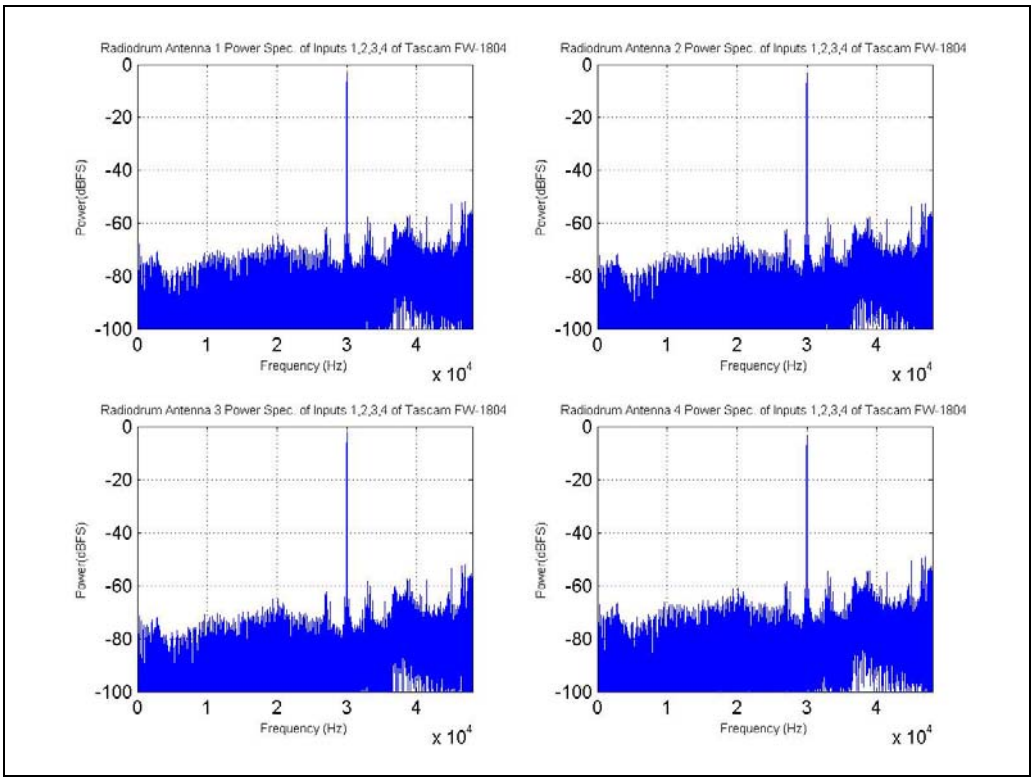

Figure 4.11. *Power spectrum of Radio Drum antenna's at the inputs to the Tascam audio interface. Shown clockwise: Inputs 1, 2 ,3, 4*

4.3.2 Determination of R(k), the System Noise Covariance Matrix

The measurement model requires a constant covariance matrix to describe the measurement noise of the Radio Drum system. This matrix defines the variance and covariance of the x, y, and z positions in units of voltages. To find this we need to determine a covariance matrix at various locations across the domain of the Radio

Drum and analyze if and how the covariance matrix varies as a function of stick position.

Radio Drum antenna data was collected using the Fireface audio interface with the stick at rest for 29 seconds at 9*6=54 different location. Figure 4.12 shows the spatial sample points on the surface of the Radio Drum. Sampling took place at each of these points at heights of 0, 8, 16, 24, 32, and 40 centimeters.

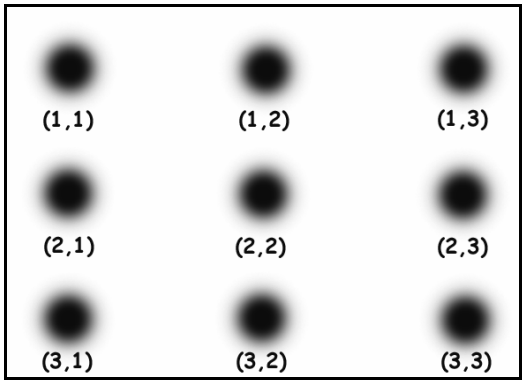

Figure 4.12. *Radio Drum stick locations*

These sampling locations were selected to get a fair representation of the antenna signal noise across the entire domain of the Radio Drum. The data was collected into 4 channel NeXt au files at a sample rate of 96 KHz at 24 bit. The 54, 4 channel data files were then demodulated and the x, y, and z voltages were calculated. The Matlab COV function was used to get 54 covariance matrices describing the noise across the entire domain of the Radio Drum surface. The variance of the position in centimeters of x, y and z and the covariances of x and y, x and z, and y and z were plotted as a function of x (left to right), y (top to bottom), and height.

It is important to note that since the z voltage is non-linear as a function of stick height we can not accurately translate the z position voltage into centimeters. In

the next section however, we continue to define z position variances in terms of centimeters to get a feel for the dependence of the noise on stick height.

4.3.2.1 Covariance as a function of position plots

Figure 4.13 and Figure 4.14 plots the variances and covariances of position as a function of x. The stick was moved along the x axis of the drum on the surface (0cm height). The sample points correspond to (2,1), (2,2), and (2,3) on Figure 4.12. The variance of x, y, and z position stays relatively low, below 0.2 cm^2, and constant and our positions stay relatively uncorrelated as the stick moves from left to right on the surface.

Figure 4.15 and Figure 4.16 and plot the variances and covariances of position as a function of y. The stick was moved along the y axis of the drum on the surface (0cm height). The sample points correspond to (1,2), (2,2), and (3,2) on Figure 4.12. We can conclude the same things for the stick moving along the y axis as we did for the x axis.

Figure 4.17 and Figure 4.18 plot the variances and covariances of position as a function of height. The stick was moved along the z axis of the drum at point (2,2) with z=0, 8, 16, 24, 32, and 40 cm. The x and y positions maintain a relatively small and constant variance up to a height of around 18cm, at which point the x and y noise become position dependant. The z position exhibits a much larger increase in noise as the stick reaches above 16cm. The z position variance climbs to roughly 20 cm^2 as the stick moves above 32cm. Clearly, the z position noise is height dependant. The

covariance of x, z, and y, z show that as z positions estimate increases, the x and y estimates also increase. This non-linearity is prevalent above a stick height of 32cm.

Figure 4.19 and Figure 4.20 once again plots the variances and covariances of position as a function of height. The stick was moved along the z axis of the drum at point (1,1) with z=0, 8, 16, 24, 32, and 40 cm. These plots show similar results to the plots where the stick was moved up the centre of the surface.

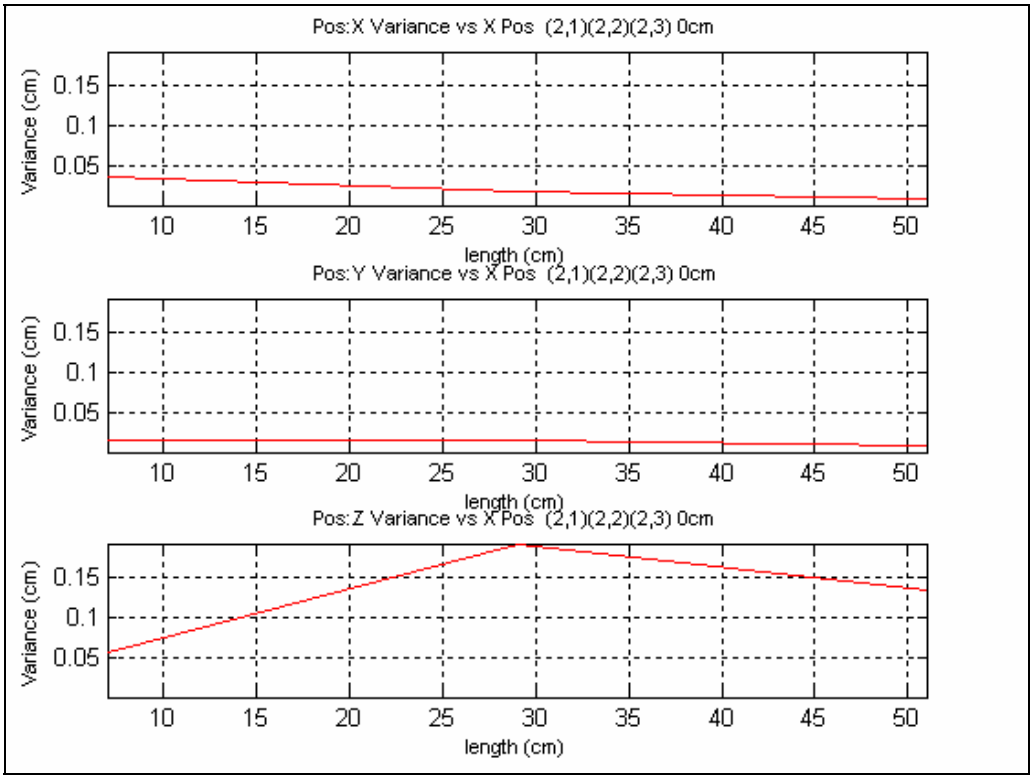

Figure 4.13. *x, y, and z variance as a function of x position (2,1)(2,2)(2,3)*

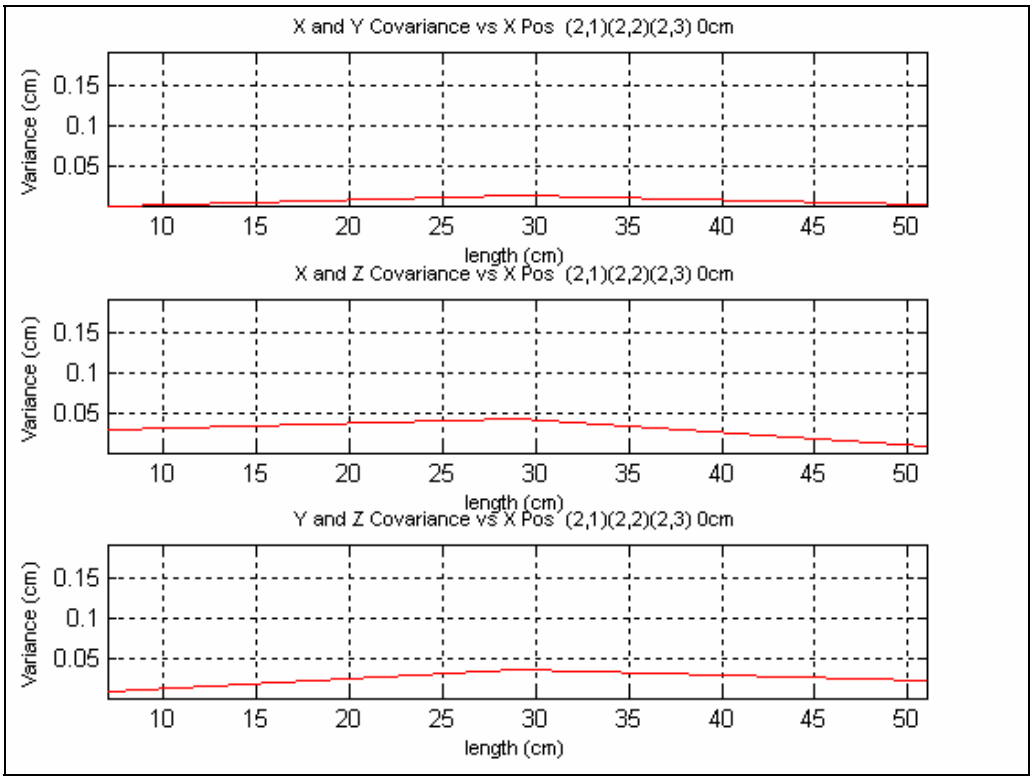

Figure 4.14. *x, y, z covariances as a function of x position (2,1)(2,2)(2,3)*

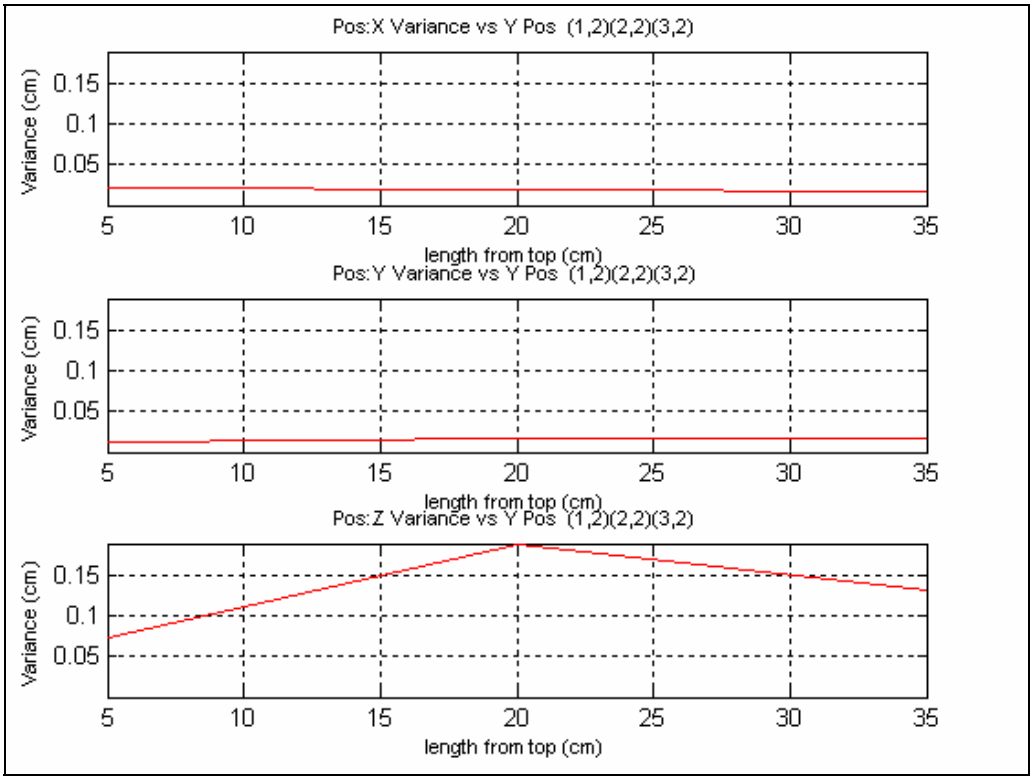

Figure 4.15. *x, y, z variances as a function of y position (1,2)(2,2)(3,2)*

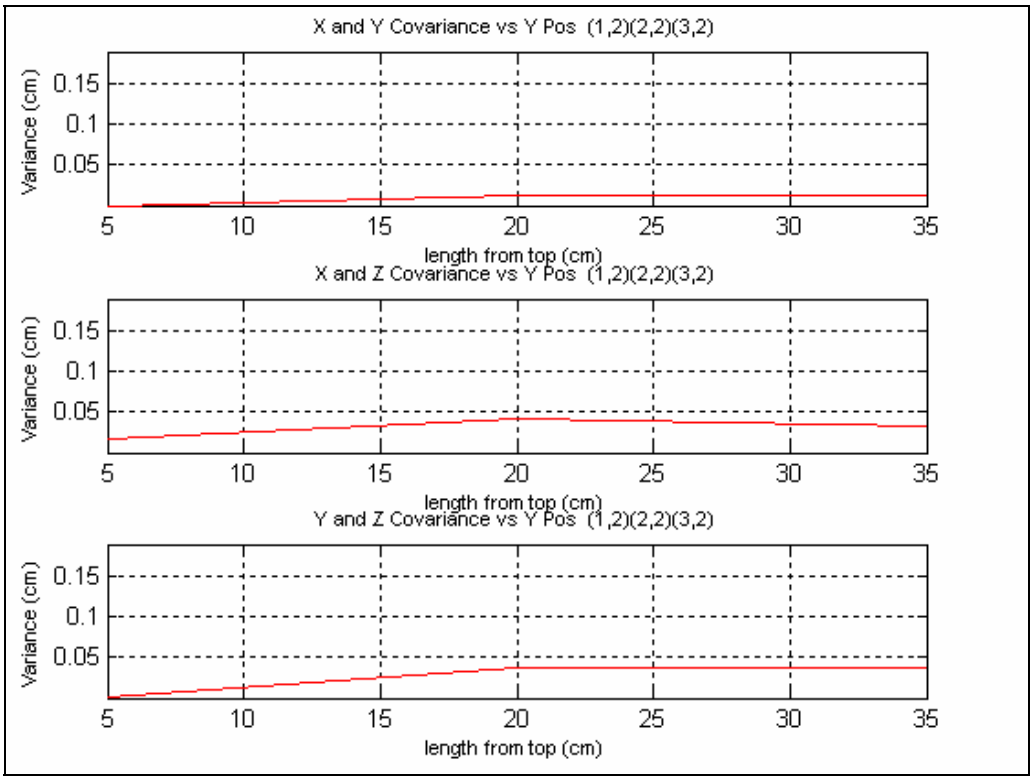

Figure 4.16. *x, y, z covariances as a function of y position (1,2)(2,2)(3,2)*

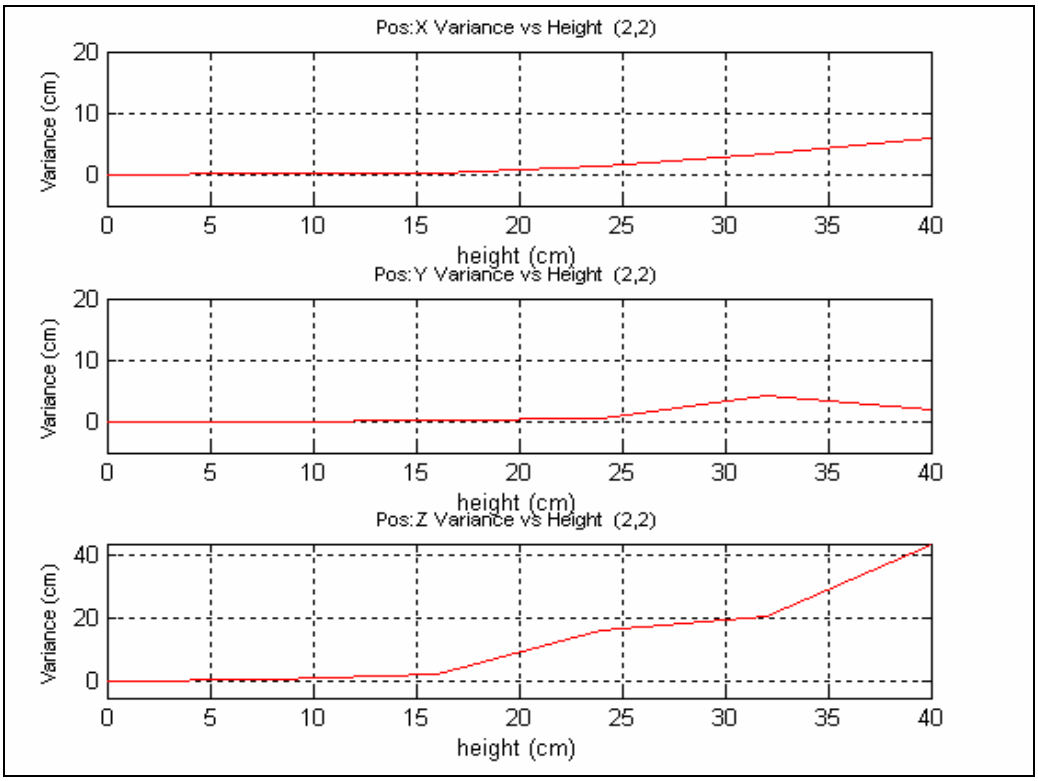

Figure 4.17. *x, y, z variances as a function of height (2,2)*

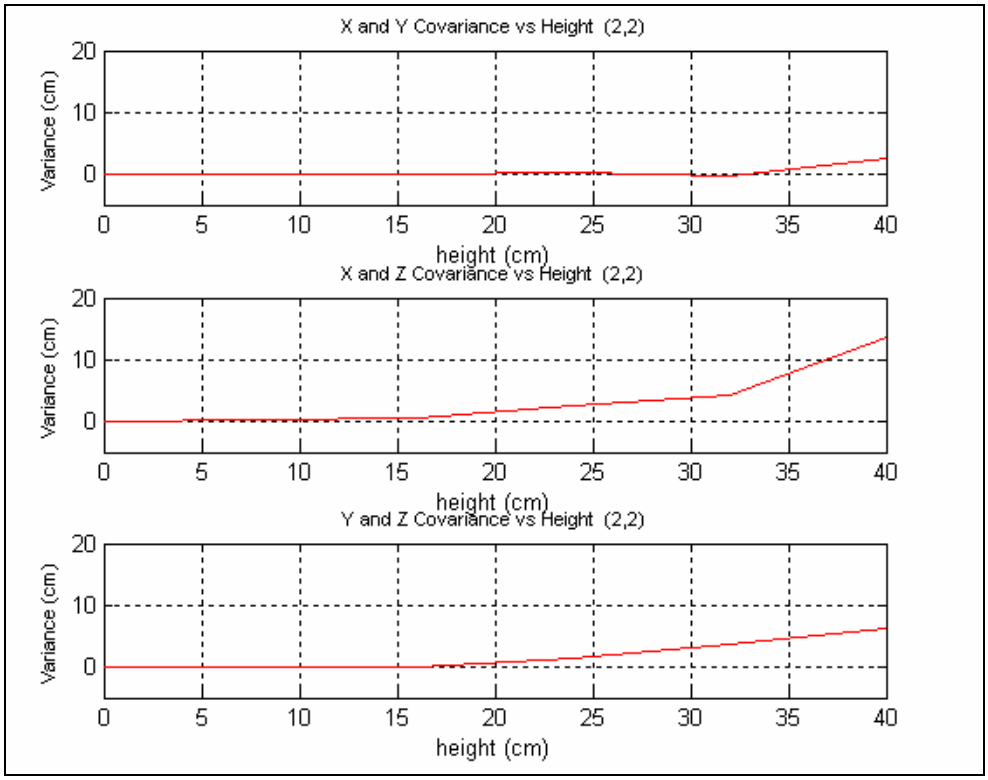

Figure 4.18. *x, y, z covariances as a function of height (2,2)*

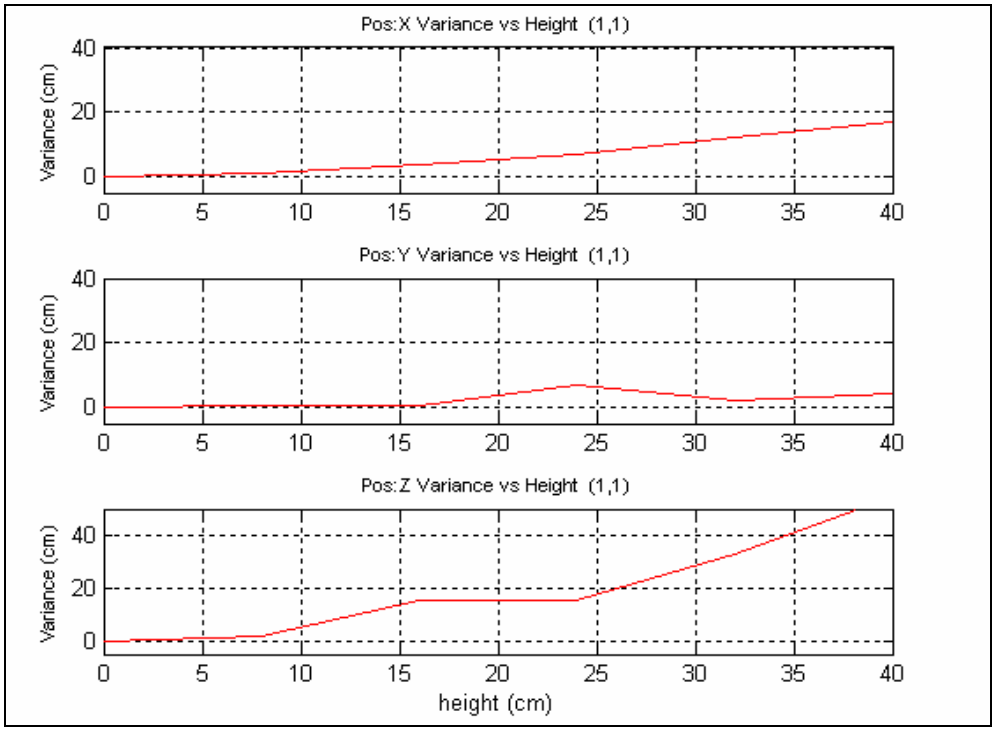

Figure 4.19. *x, y, z variances as a function of height (1,1)*

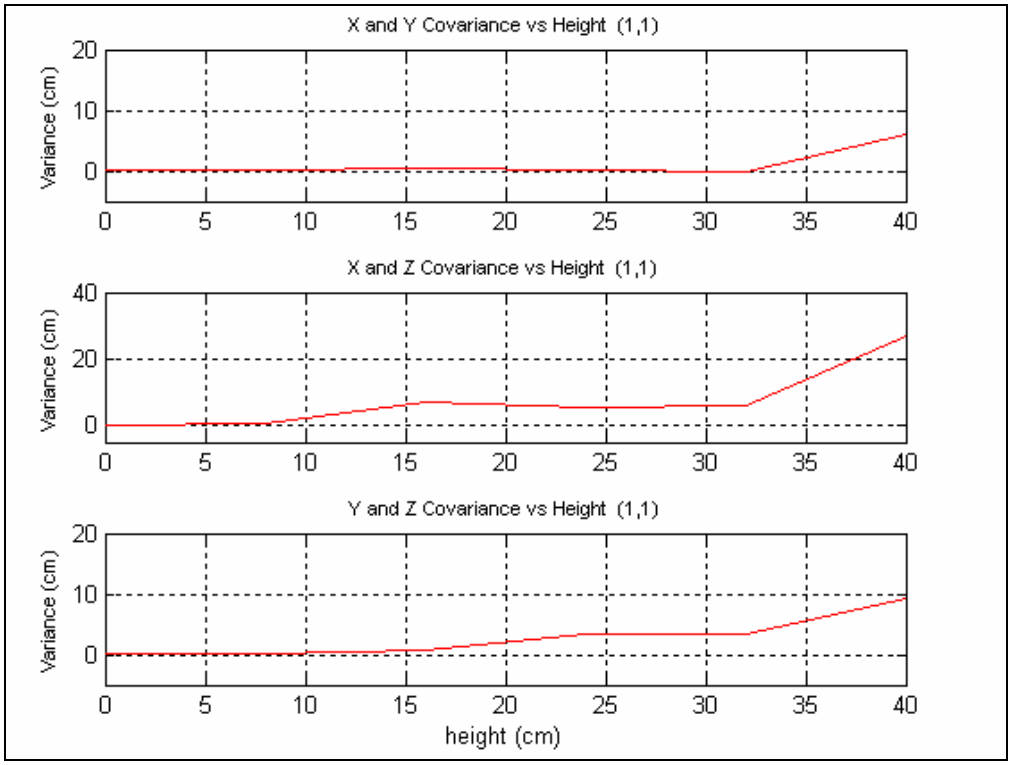

Figure 4.20. *x, y, z covariances as a function of height (1,1)*

Through the preceding analysis we can conclude that the noise does increase as the stick is lifted higher off the surface. We also see a non-linear behavior of the x and y position at stick heights greater than 32 cm. The z position exhibits the greatest amount of noise versus height. This is justified since the z position is the sum of all four noisy antenna signals. Figure 4.23 and Figure 4.24 plot the same variances and covariance of Figure 4.17 and Figure 4.18 however in the units of voltage. These plots confirm the magnitude of the z variance compared to that of the x and y coordinates.

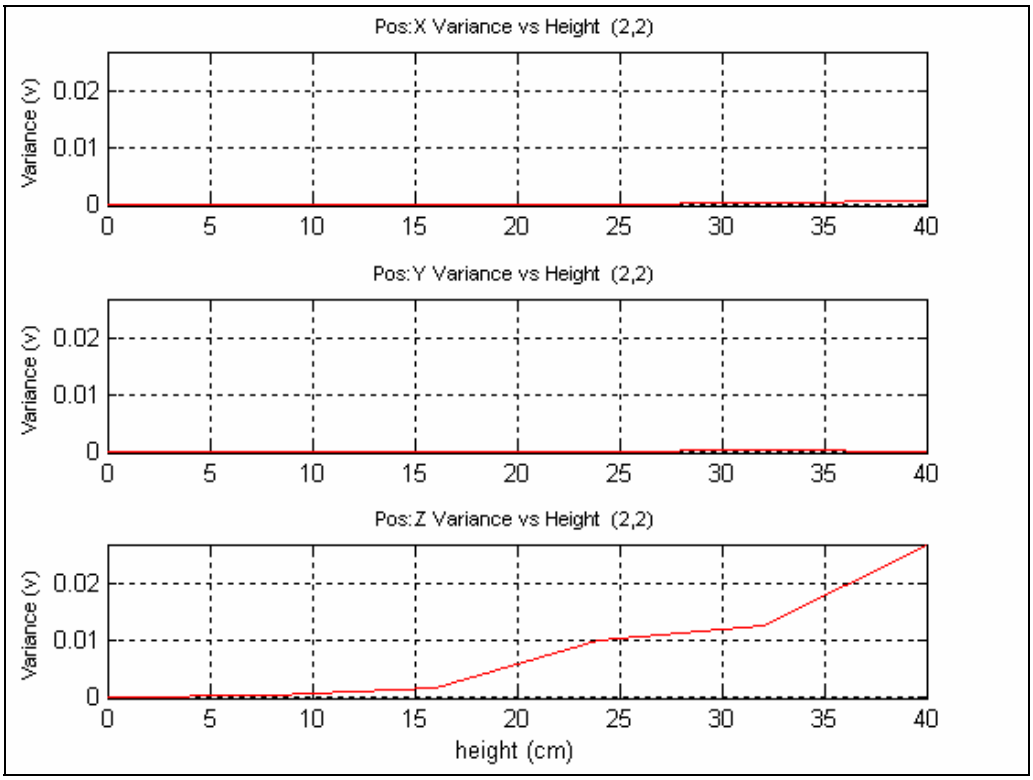

Figure 4.21. *x, y, z voltage variances as a function of height (2,2)*

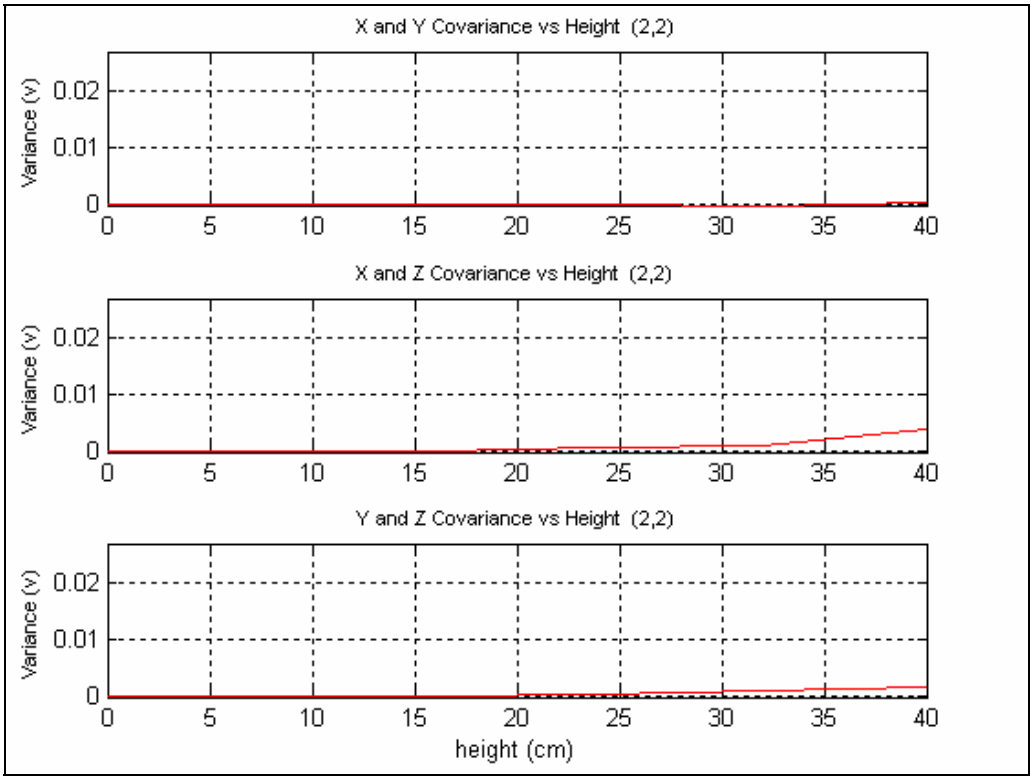

Figure 4.22. *x, y, z voltage covariances as a function of height (2,2)*

4.3.2.2 Bounding the Radio Drum Domain and Finding a Single Covariance Matrix

The following mesh plots show how the variance of the z position varies as the stick moves over the surface at a constant height. The points x position and y position correspond to the grid locations on the surface shown in Figure 4.12.

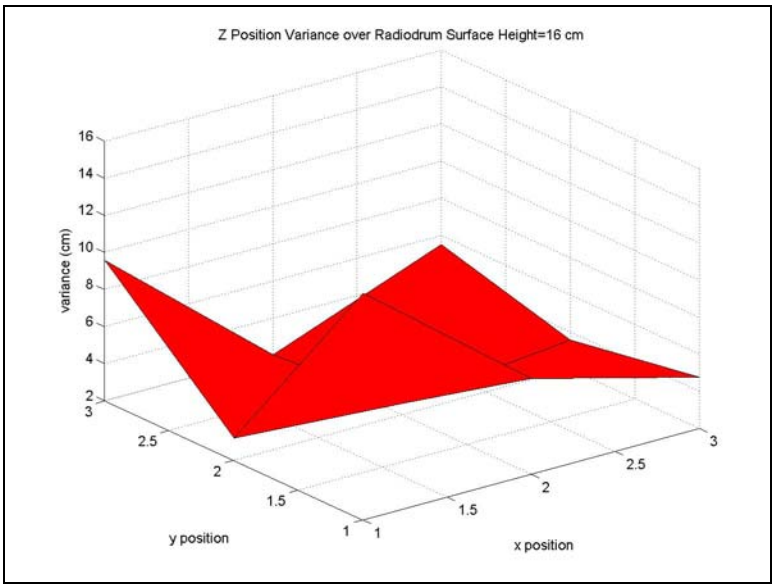

Figure 4.23. *Z variance over Radio Drum surface at height 16cm*

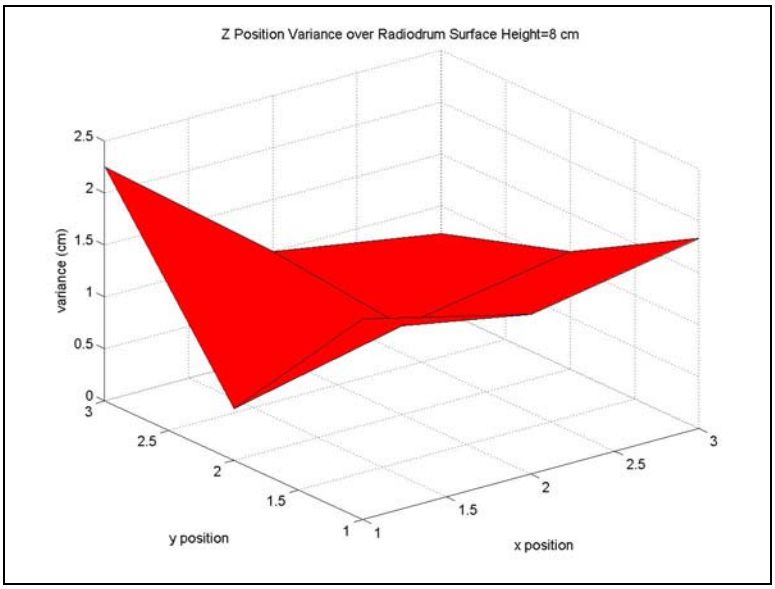

Figure 4.24. *Z variance over Radio Drum surface at height 8cm.*

As you can see even the variance of the z position noise is not constant at a constant height. The noise increases the edge of the surface. As expected, the z position variance at a height of 16cm is greater and has a greater range of fluctuation across the surface compared to the z position variance at 8cm. This analysis was performed at heights of 0, 24, 32, and 40 cm as well as the plotted results at 8 and 16 cm. Table 4.3 summarizes the maximums, minimums with corresponding grid points, and range of fluctuation of z position variance at different heights off the Radio Drum surface.

Height	**Max Z Variance (cm^2)** **Grid Position**	**Min Z Variance (cm^2)** **Grid Position**	**Range (cm^2)**
0cm	0.19 (2,2)	0.021 (2,1)	0.17
8cm	2.3 (3,1)	0.49 (2,1)	1.8
16cm	14.2 (1,1)	2.0(3,2)	12.2
24cm	20.3 (1,3)	6.9(2,1)	13.4
32cm	35.8 (1,2)	14.9 (3,3)	20.9
40cm	47.1 (1,1)	16.1 (3,2)	31.0

Table 4.3. *Z position variance max and mins with increase in height*

It is clear that as the stick moves higher the z variance increases and fluctuates more across the surface.

It is obvious that our system has position dependant noise. This means that the whole domain of the Radio Drum system noise cannot be described by a single covariance matrix. Since our measurement model requires a constant covariance matrix across the whole domain of the Radio Drum, we must limit our domain to that in which a single covariance matrix may be used. Since x and y position variance stay relatively constant over the surface we are mainly concerned with limiting the height for which our measurement model will be valid.

A single covariance matrix was chosen for the following ranges: x 54.3cm, y 38cm, and z 24cm. The x and y were chosen as the distances from the left edge to the right edge and the top edge to the bottom edge respectively. An edge corresponds to where the metal strips begin on the Radio Drum surface. Z was chosen based on Table 4.3. Within these limits we have confidence in our measurement model.
To obtain a single covariance matrix all covariance matrices over the bounding range were averaged. The table below shows the matrix entries for this final covariance matrix in demodulated antenna voltages.

$$R(k) = \begin{pmatrix} 1.951 & 0.7239 & 6.221 \\ 0.7239 & 1.110 & 3.310 \\ 6.221 & 3.310 & 46.30 \end{pmatrix} * 10^{-4}$$

Table 4.4. *Averaged Covariance Matrix for Radio Drum*

Where the diagonal elements (1,1), (2,2), and (3,3) are the x, y, and z variances respectively. (1,2) is the x, y covariance, (1,3) is the x, z covariance and (2,3) is the y, z covariance. Next we will discuss the development of the dynamic model for the Kalman filter algorithm.

4.4 Dynamic Model of a Radio Drum Stick

The dynamic model for the Radio Drum system was inspired by extensive work already published in the field of missile tracking for military applications. Tracking of a Radio Drum stick is analogous to tracking a missile through the earth's atmosphere using radar. The Singer Kalman Filter designed for tracking a maneuvering target was

used for our application [34, 35]. For convenience we once again show the linear dynamic model relationship.

$$X(k+1) = \Phi(k)X(k) + \Gamma U(k) + W(k)$$

Where *X(k)* and *X(k+1)* contain the position, velocity and acceleration estimates of the x, y, and z coordinates.

Singer takes into account a dynamic acceleration noise process that is not necessarily white. Physically, this means that a force may be applied to the stick, by a performer, over a certain window of time during a maneuver. The correlation time coefficient τ describes to what degree the acceleration of the stick is correlated or how long the maneuver takes and σ^2 is the variance of the acceleration over a gesture, a measure of how erratic the motion is. L equals the number of lags in samples. The autocorrelation of the acceleration is shown below.

$$E[a(t)a(t+L)] = \sigma^2 e^{(-\frac{|L|}{\tau})} \quad (4.13)$$

The *Φ(k)* matrix, shown in Table 4.5, describes how the Radio Drum stick's location is updated over consecutive time instants. T equals the sampling period, 1/3000 seconds.

$$\Phi(k) = \begin{bmatrix} 1 & T & \tau^2[-1 + \frac{T}{\tau} + e^{(-\frac{T}{\tau})}] \\ 0 & 1 & \tau[1 - e^{(-\frac{T}{\tau})}] \\ 0 & 0 & e^{(-\frac{T}{\tau})} \end{bmatrix}$$

Table 4.5. *Phi transition matrix*

Considering just the x location, the equations below show how the state parameters are updated based on the transition matrix.

$$x(k+1) = x(k) + Tv(k) + \Phi_{13}a(k)$$

$$v(k+1) = v(k) + \Phi_{23}a(k) \qquad (4.14)$$

$$a(k+1) = \Phi_{33}a(k)$$

In our case there are no inputs to the system therefore *U(k)=0* and *Γ(k)=0*.

Since the Kalman filter assumes a random white disturbance vector, the random acceleration component of *W(k)* is processed through a whitening filter giving rise to a *Q(k)* matrix of this form.

$$Q(k) = E[W(k)W^T(k)] = \frac{2\sigma^2}{\tau}\begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \qquad (4.15)$$

The q values are functions of τ, the correlation time and T, the sampling period. For the derivation of *Q(k)* see [34].

$$q11 = \frac{1}{(2*a^5)}*(1-\exp(-2*a*T)+2*a*T+2*a^3*\frac{T^3}{3}-2*a^2*T^2-4*a*T*\exp(-a*T))$$

$$q12 = \frac{1}{(2*a^4)}*(\exp(-2*a*T)+1-2*\exp(-a*T)+2*a*T*\exp(-a*T)-2*a*T+a^2*T^2)$$

$$q13 = \frac{1}{(2*a^3)}*(1-\exp(-2*a*T)-2*a*T*\exp(-a*T))$$

$$q22 = \frac{1}{(2*a^3)}*(4*\exp(-a*T)-3-\exp(-2*a*T)+2*a*T)$$

$$q23 = \frac{1}{(2*a^2)}*(\exp(-2*a*T)+1-2*\exp(-a*T))$$

$$q33 = \frac{1}{(2 * a\ )} * (1 - \exp(-2 * a * T))$$

$$a = \frac{1}{\tau}$$

Due to the large amount of noise inherent in our measurements, we cannot simply take the autocorrelation of the acceleration of a variety of Radio Drum gestures to obtain suitable values for τ and σ^2. We can only manually adjust the parameters around nominal values that we might expect and compare the results to our noisy gesture data.

4.5 Preliminary Design Conclusions

In this chapter we discussed the design of a measurement, and dynamic model for the Radio Drum system. An in depth analysis of the noise inherent in the Radio Drum system has led us to the conclusions that our measurement noise is correlated both over time and space. We proved the existence of low frequency periodicity in the demodulated antenna signals, the source of which is the Fireface 800 audio interface. We also proved that the variance of the noise is dependant on the stick's location above the surface. Variance of the noise increases and significant negative correlation is observed in the x and y positions as the stick moves higher. Even though we were unable to obtain independent measurements of the noise, a covariance matrix of the measurement noise was defined; the consequences of which are to be discussed later. In the next chapter we will fully develop our design of a multiple model approach to the Radio Drum Kalman filtering problem, including model calibration and tracking results.

Chapter 5

5 Improved Gesture Tracking of the Radio Drum Part 2: Multiple Model Design and Tracking Results

During a performance a Radio Drum stick can experience a wide range of accelerations. Slower moving subtle gestures often maintain a constant velocity while surface whacks experience extreme changes in velocity from when the downward moving stick hits the surface, stops, and then bounces back upward all in a few milliseconds. A single dynamic model is not able to accurately predict this diverse range of motion, therefore multiple Kalman filters, each with a specific dynamic model, must be run in parallel to capture the full range of stick motion.

This chapter will discuss the design of each one of these dynamic models and show tracking results on specific gestures. We will then go on to discuss an Interacting Multiple Model Estimator (IMM) used to combine the outputs of each model giving a single estimate of the sticks position based on a model likelihood weighting criterion. Tracking results on a variety of typical performance gestures, acquired with the Fireface 800 audio interface, will be shown and we will demonstrate how Kalman filtering can lead to a higher level of expressivity when whacking the surface. Furthermore, tracking results on Radio Drum gestures, acquired with the cheaper Tascam audio interface, will also be shown and discussed. We end this chapter with a look into the future of the Radio Drum system. Figure 5.1 is a block diagram of the Radio Drum system showing the flow of the Radio Drum

antenna signals from the instrument into the software and into the Interacting Multiple Model estimator.

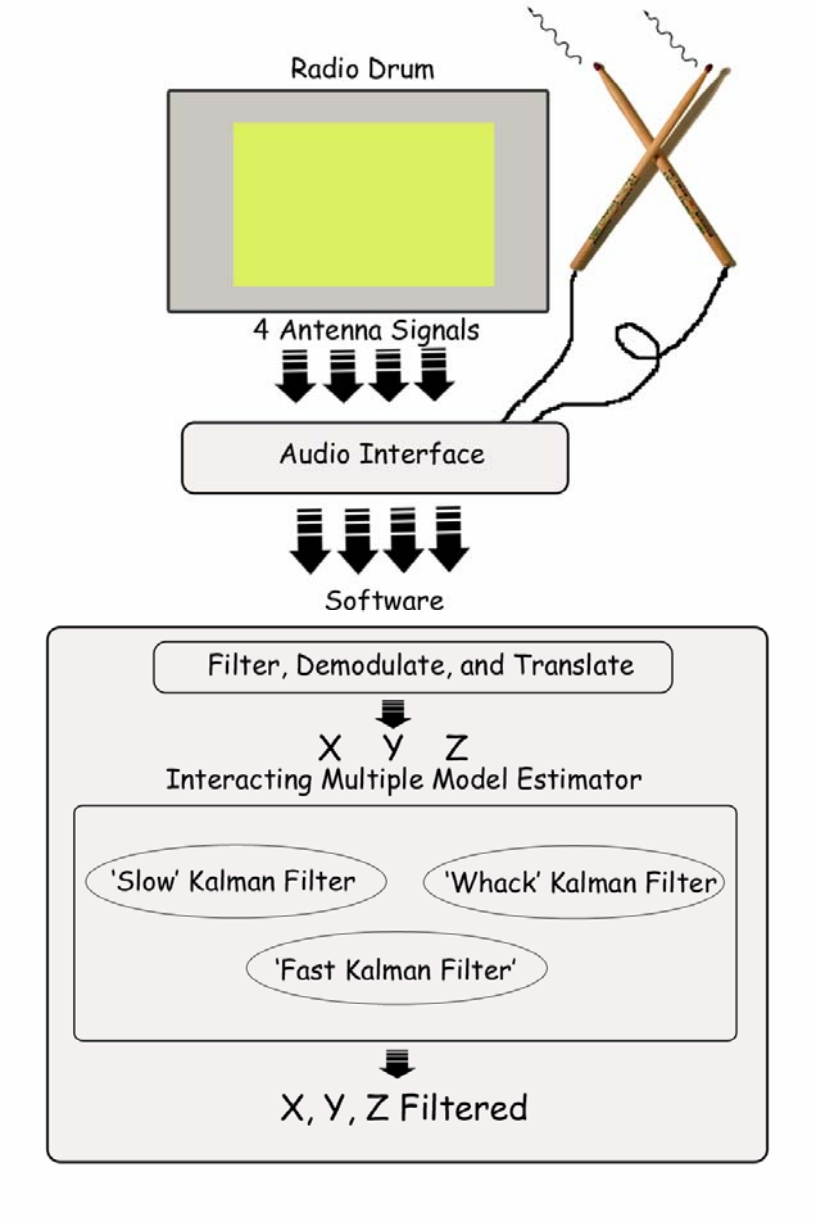

Figure 5.1. *Radio Drum Kalman Filtering Block Diagram*

5.1 Multiple Motion models for the Radio Drum Stick

As discussed in the previous chapter, the Singer dynamic model assumes a non-white acceleration noise process when modeling targets such as missiles, or in our case a drum stick. This means that as a Radio Drum stick moves through a maneuver, the performer's motion exerts a force on to it over a window of time. The acceleration of a maneuver can be described by τ, the correlation time of a maneuver, and σ^2, the variance of the maneuver's acceleration. To accommodate for the variety of stick accelerations expected during any Radio Drum performance, three discrete modes of performance were specified; to each of which a dynamic model, with specific tunings of τ and σ^2, was designed. We will call these modes 'slow move', 'fast move', and 'whacks'. 'slow move' and 'fast move' exhibit slow and fast movement all over the Radio Drum surface. 'whacks' is the gesture of a stick hitting the surface.

In the next subsections we tune our 3 models by comparing each Kalman filter's output position estimate to the raw position estimate of the stick. We consider the Kalman Filter as doing well when the filtered track follows the mean of the noisy raw track. Although sufficient, this criterion for tuning is flawed because we do not have any ground truth as to the exact track that the stick is taking. We will discuss this more in future work.

5.1.1 Tuning for the Slow Move Gesture

Figure 5.2, Figure 5.3, and Figure 5.4 below show the raw Radio Drum position voltages in green with the Kalman Filtered estimate of the track in black, x, y, and z are shown respectively. The gesture was recorded for 5 seconds. τ =T*100 sec and

σ^2=0.1m/s^2 were used to specify the acceleration. These values represent slow maneuvering with minimal acceleration.

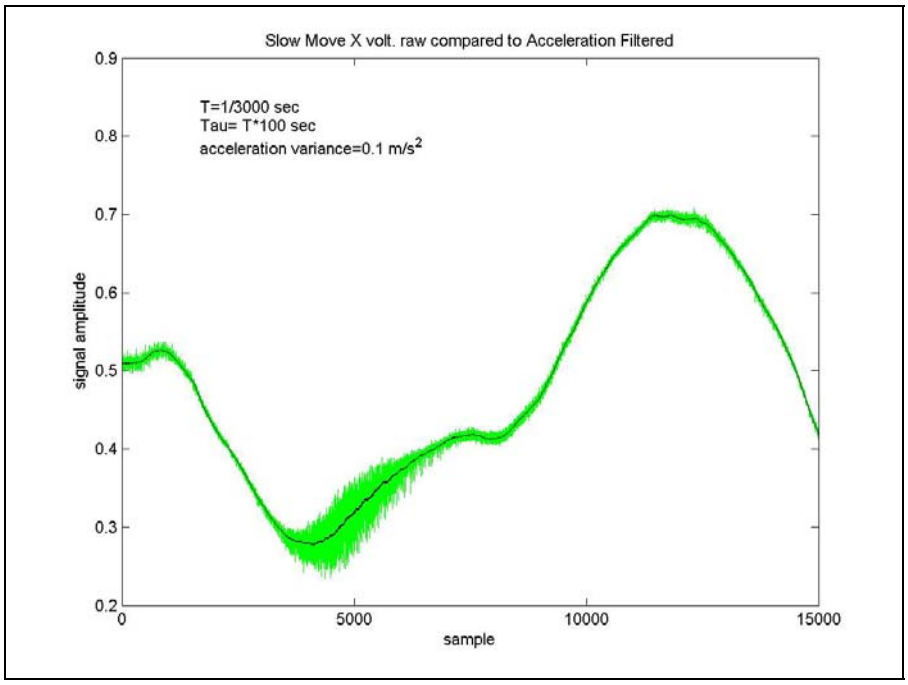

Figure 5.2. *x voltage of Slow Move gesture filtered*

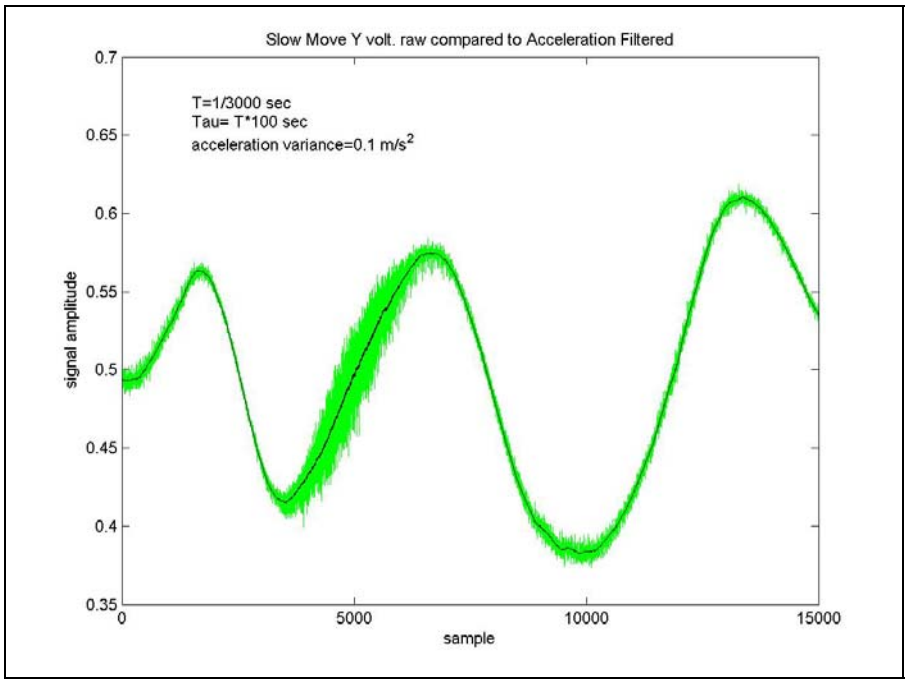

Figure 5.3. *y voltage of Slow Move gesture filtered*

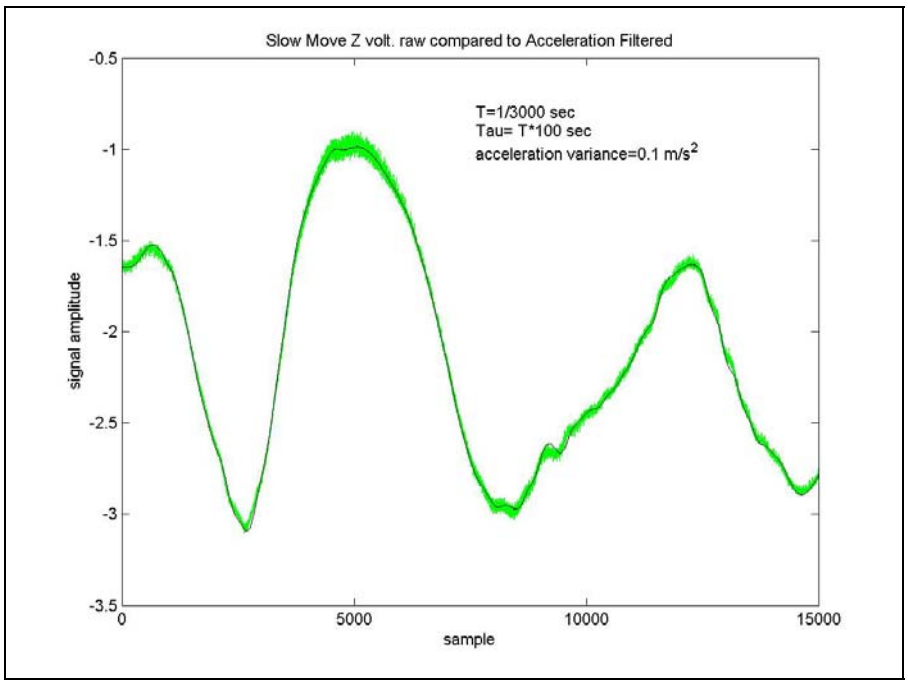

Figure 5.4. *z voltage of Slow Move gesture filtered*

The x and y positions seem to track well through noisy data and for the most part so does the z. However, the z track shows a little deviation at some points; particularly at 9000 samples. Figure 5.5 shows a close up of the z position voltage track. The Kalman track deviates at around 9200 samples. The Kalman Filter takes too long to catch up with the quick movement of the stick.

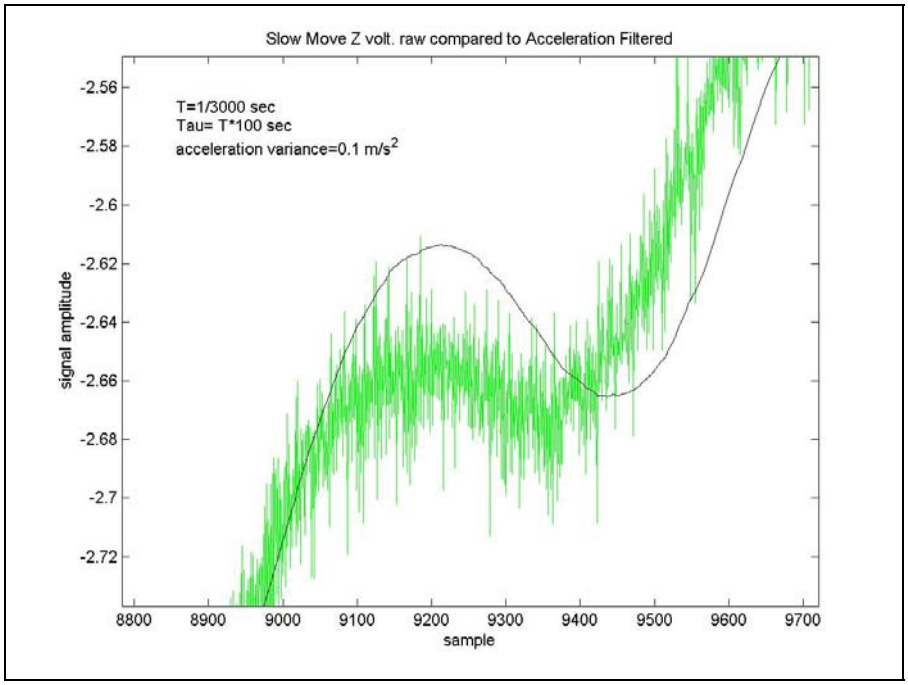

Figure 5.5. *z voltage of Slow Move gesture filtered close up*

With τ =T*100 sec and σ^2=100 m/s^2 the tighter track of Figure 5.6 can be achieved. Further testing was done on the Fast Move gesture.

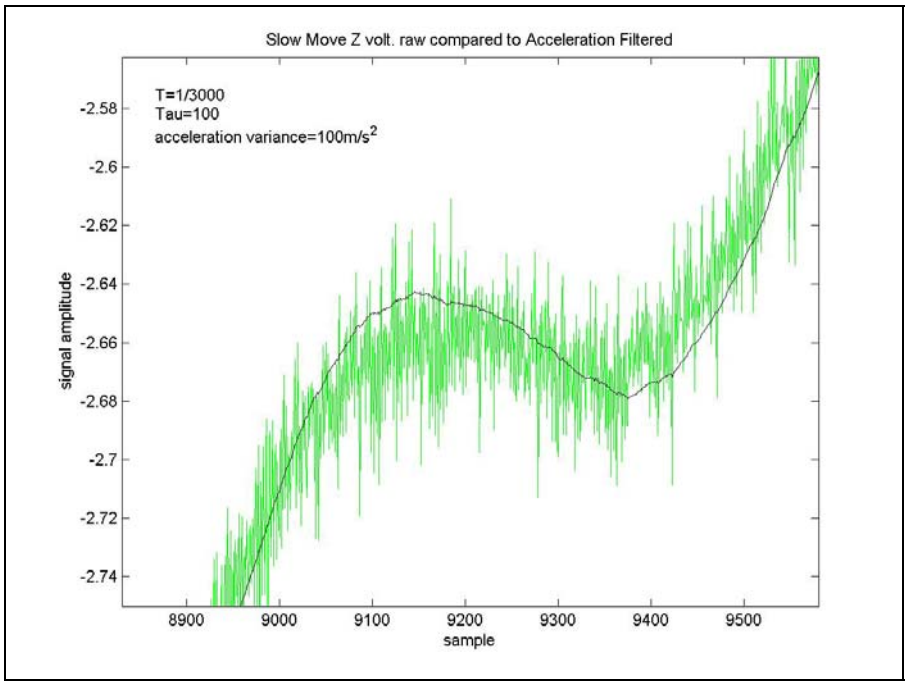

Figure 5.6. *z voltage of Slow Move gesture filtered close up*

Figure 5.7 and Figure 5.8 below show 3 dimensional plots of the raw and filtered stick track. The filter does a good job of tracing the correct path with minimal variance.

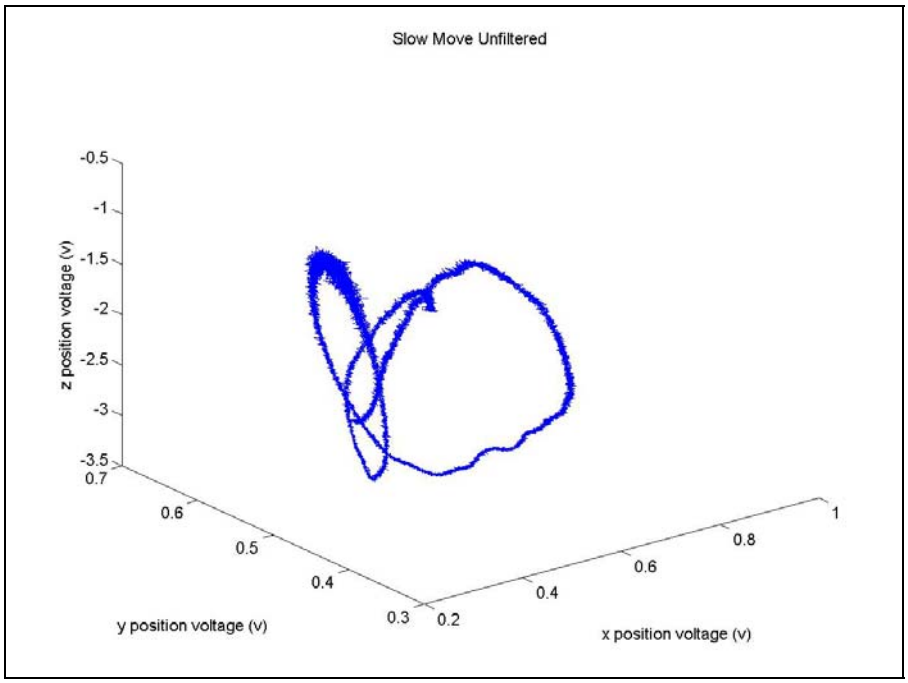

Figure 5.7. *Raw 3D track of Slow Move Gesture*

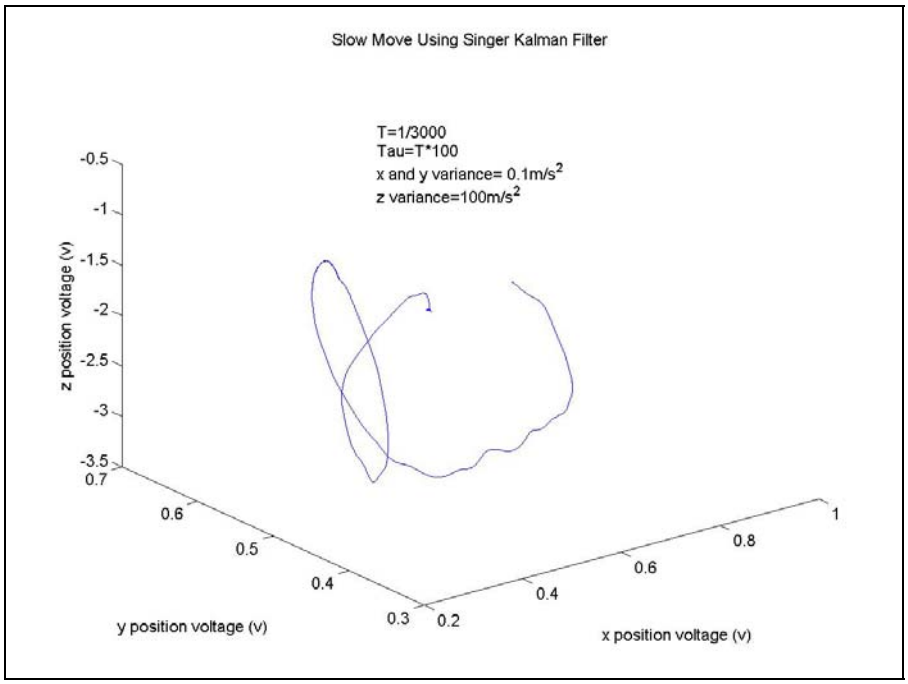

Figure 5.8. *Filtered 3D track of Slow Move Gesture*

5.1.2 Tuning for the 'Fast Move' Gesture

Figure 5.10 shows the raw and filtered z position track of 5 seconds of a Fast Move gesture. The same parameters that were used for the Slow Move gesture were used for this plot. It is obvious that such a long time constant and low acceleration variance cannot deal with the faster movements of the gesture. This becomes apparent at the extreme points of the motion where the stick experiences quick changes in velocity. Figure 5.11 shows a close up of some of the extreme points of the motion. The difference of 0.15 volts at sample 4600 from the raw data to the estimated track leads to an error of about 6 cm in space. This sort of inaccuracy is unacceptable. Similar results were seen for the x and y tracks.

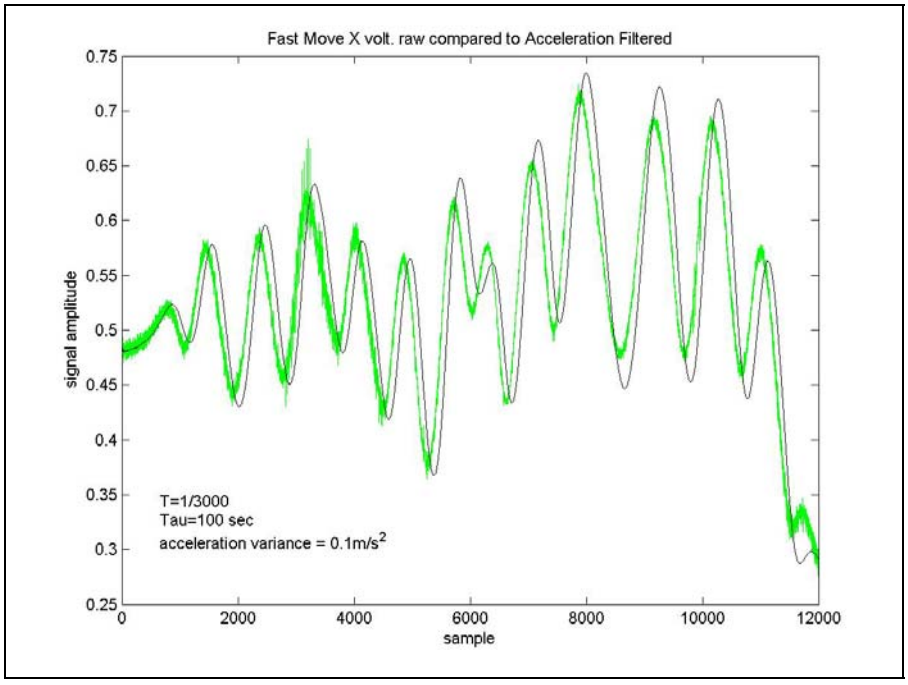

Figure 5.9. *x voltage of Fast Move gesture filtered*

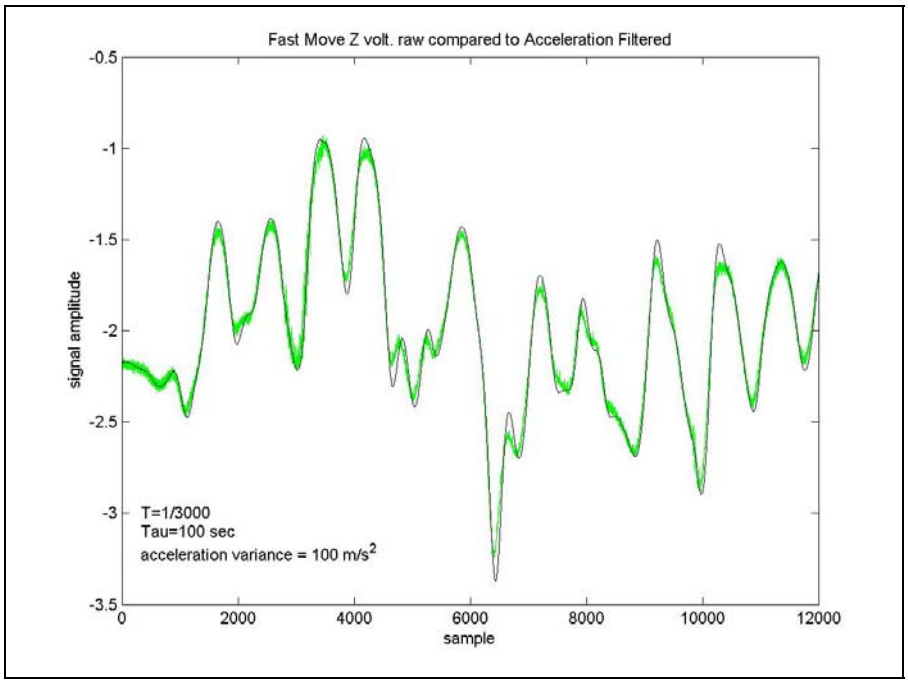

Figure 5.10. *z voltage of Fast Move gesture filtered*

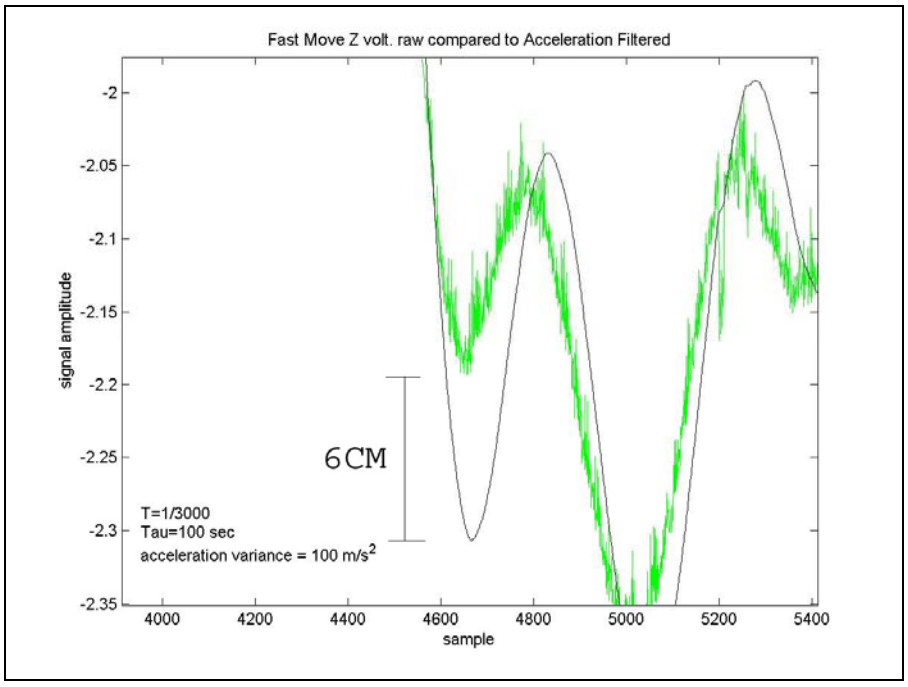

Figure 5.11. *z voltage of Fast Move gesture filtered close up*

An initial attempt at tuning the parameters to deal with such high accelerations was made. Figure 5.12 shows the same fast maneuver as in Figure 5.11. With $\tau=T*65$ sec and a $\sigma^2 = 80000\ m/s^2$ the filter seems to obtain a tighter track. However, the filtered position estimate exhibits much greater variance.

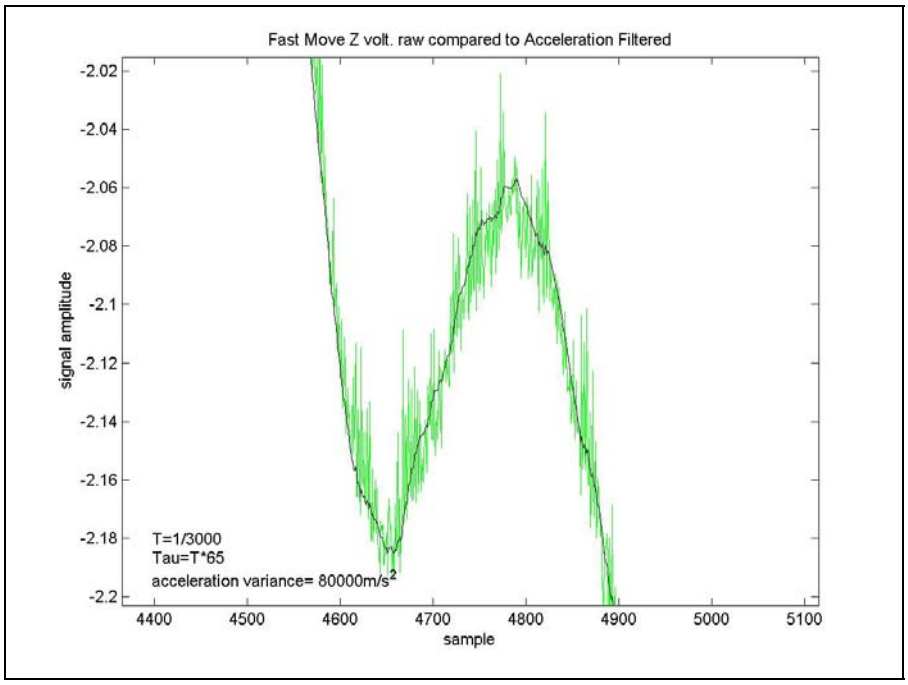

Figure 5.12. *z voltage of Fast Move gesture filtered close up using $\tau=T*65$, $\sigma^2=8000m/s^2$*

Figure 5.13, Figure 5.14, and Figure 5.15 show the estimated track of the Kalman Filter for the Fast Move gesture after the τ and σ^2 parameters have been tuned with $\tau=T*65$ sec and $\sigma^2=800\ m/s^2$. The Kalman Filter was able to track the fast maneuvering of the stick at the expense of higher variance in the estimates.

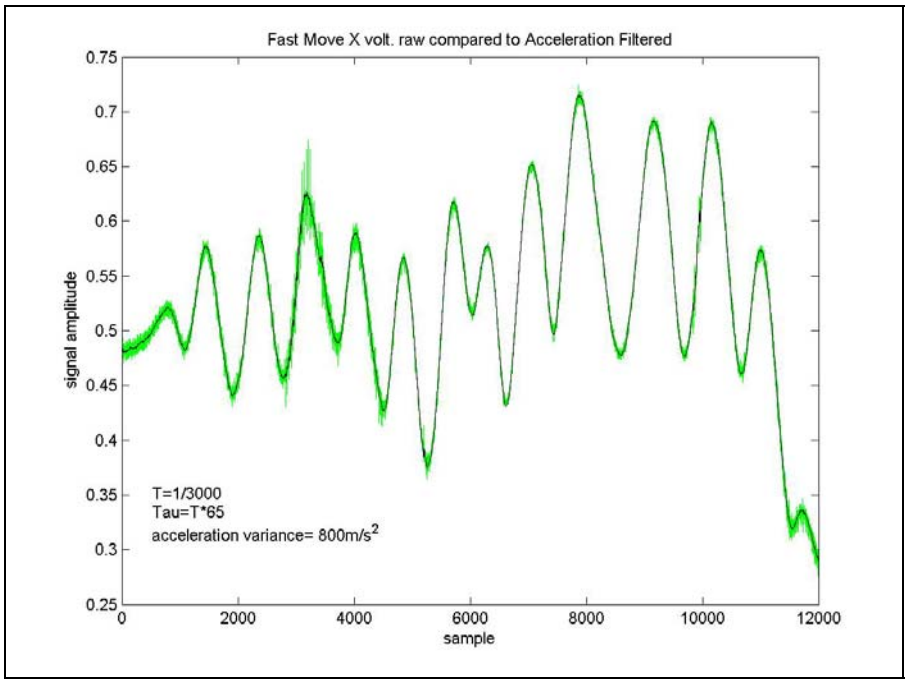

Figure 5.13. *x voltage of Fast Move gesture filtered close up using* $\tau = T*65$, $\sigma^2 = 800 m/s^2$

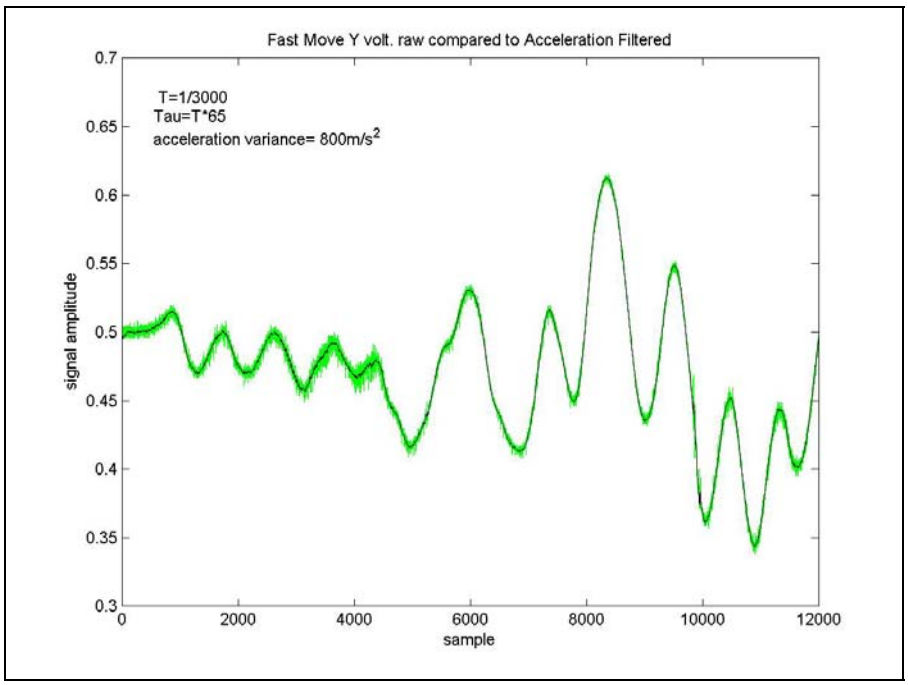

Figure 5.14: *y voltage of Fast Move gesture filtered close up using* $\tau = T*65$, $\sigma^2 = 800 m/s^2$

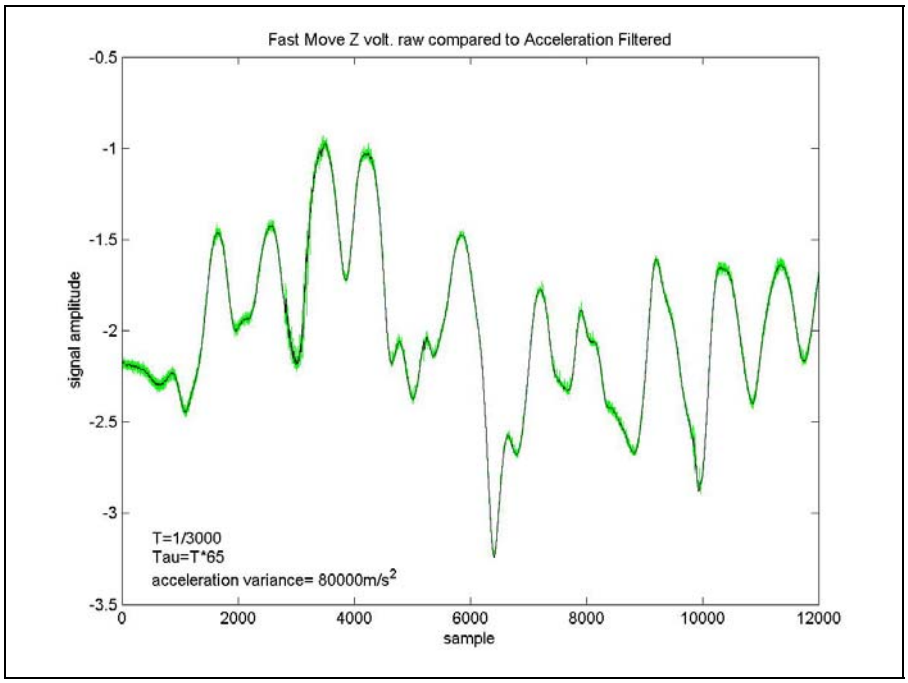

Figure 5.15. *z voltage of Fast Move gesture filtered close up using $\tau=T*65$, $\sigma^2=8000m/s^2$*

Figure 5.16 and Figure 5.17 below show 3 dimensional plots of the raw and filtered stick track for the Fast Move gesture. The filter does a good job of tracing the correct path, but with a higher variance in the estimates when compared to the track of the Slow Move gesture (Figure 5.8).

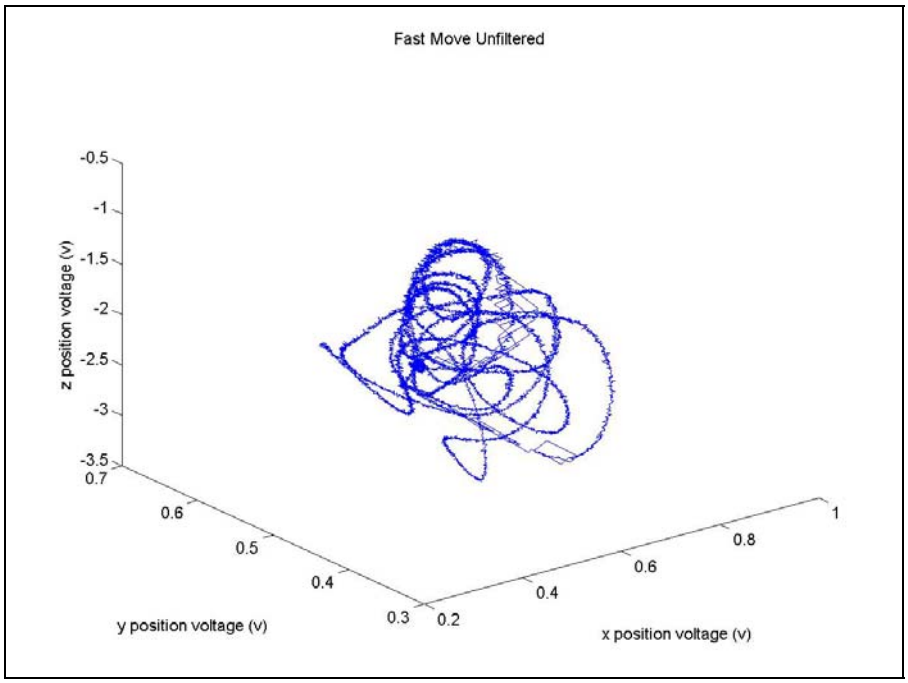

Figure 5.16. *Raw 3D track of Fast Move Gesture*

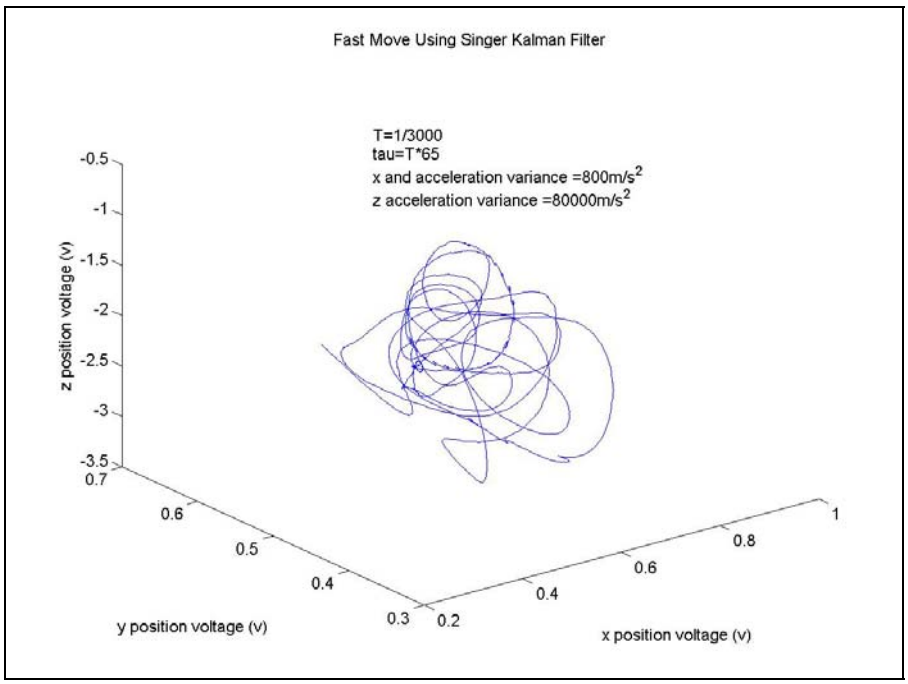

Figure 5.17. *Filtered 3D track of Fast Move Gesture*

5.1.3 Tuning for the 'Whack' Gesture

A whack gesture is the act of hitting the drum stick down onto the surface similar to the way one would hit a real drum with a drum stick. This is by far the most challenging case for the Kalman Filter since the motion of the z position experiences extreme bursts of acceleration when the stick the hits the surface. However, the x and y positions are simple to track because they stay relatively constant over the course of a whack event.

5.1.3.2 Tuning x and y for the Whack gesture

Figure 5.18 and Figure 5.19 show the raw and filtered tracks of the x and y voltages as the stick hits the surface repeatedly.

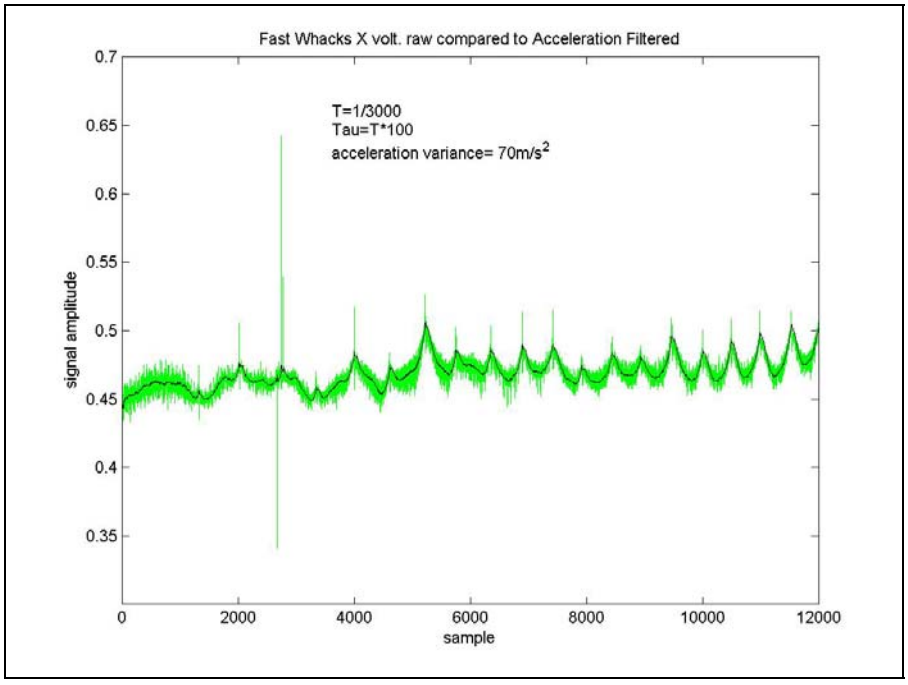

Figure 5.18. *x voltage of Whack gesture filtered*

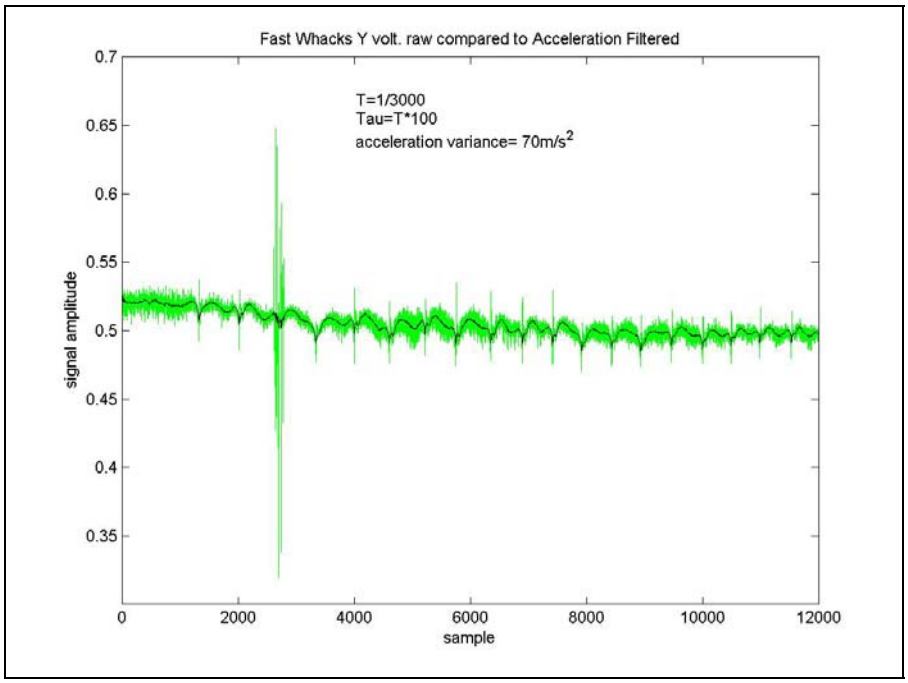

Figure 5.19. *y voltage of Whack gesture filtered*

Since the majority of the movement is in the z direction, filtering of the x and y signals can provide a smooth estimate. Values of $\tau=T*100$ sec and $\sigma^2=70\ m/s^2$ were used. Even though the x and y position doesn't change much, small impulses are experienced when the stick hits the surface. The filter does well ignoring the burst of noise at 2800 samples.

5.1.3.3 Tuning z for the Whack gesture

Figure 5.20 shows 19 whack events over the period of 4 seconds. The same tuning parameters τ and σ^2 used for the Fast Move are used here. It is hard to see how the filter is performing from this image. Figure 5.21 shows a single whack raw and filtered.

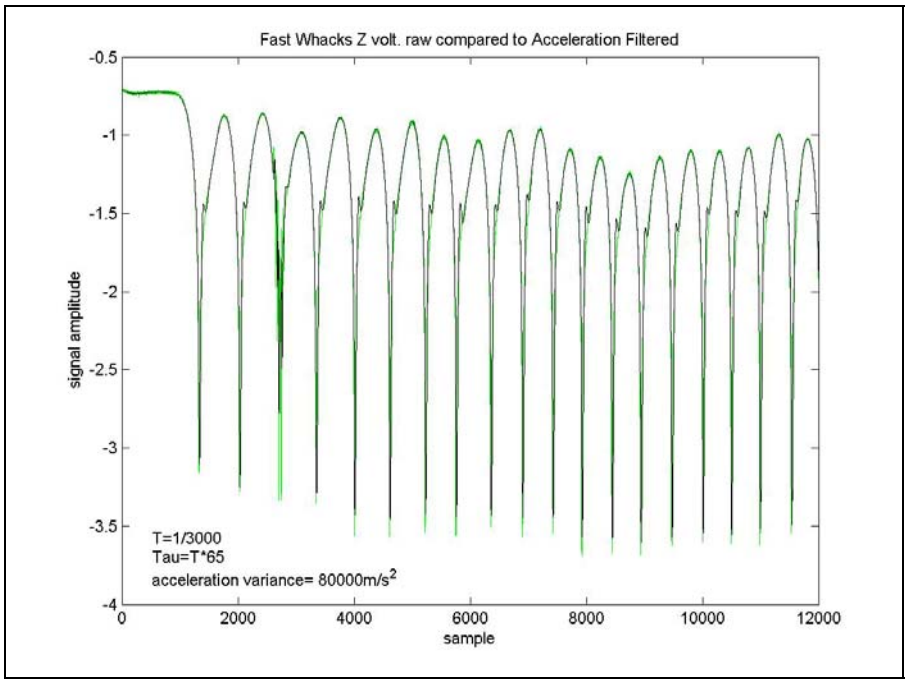

Figure 5.20. *z voltage of Whack gesture filtered*

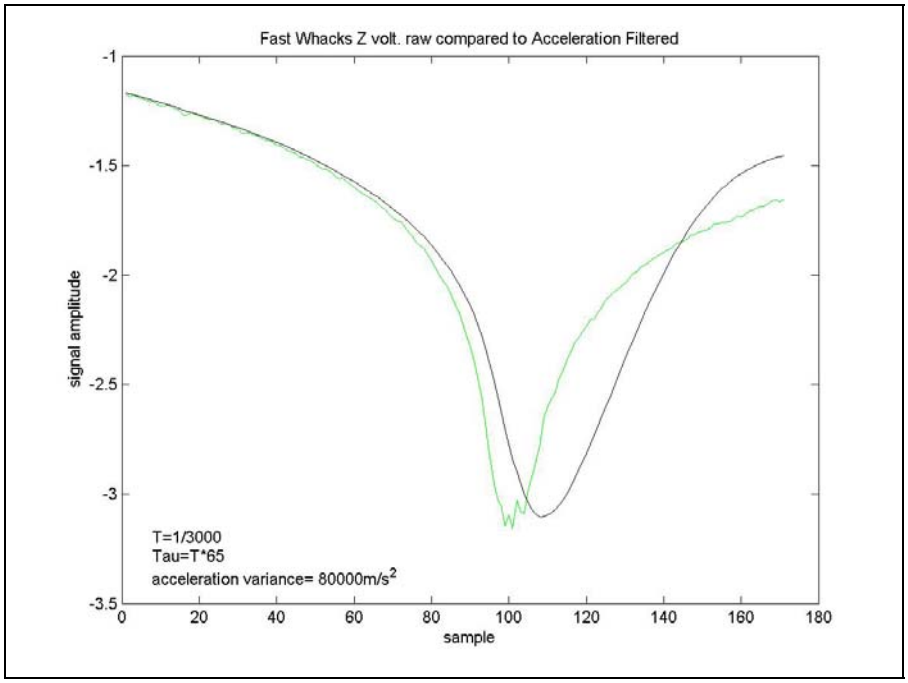

Figure 5.21. *z voltage of Whack gesture filtered close up*

The Kalman filter output is lagging 9 samples or 3 milliseconds. It is unacceptable to add this much delay onto the whack since latency is the worst enemy of a good whack detection algorithm. Furthermore, the shape of the whack is severely skewed by the filter. This is due to the filter not being able to adapt fast enough to the rapid changes of acceleration. By adjusting the Singer model parameters to extreme values it is possible to get a more precise track of a whack. Figure 5.22 plots the dashed filtered track almost exactly on top of the raw whack.

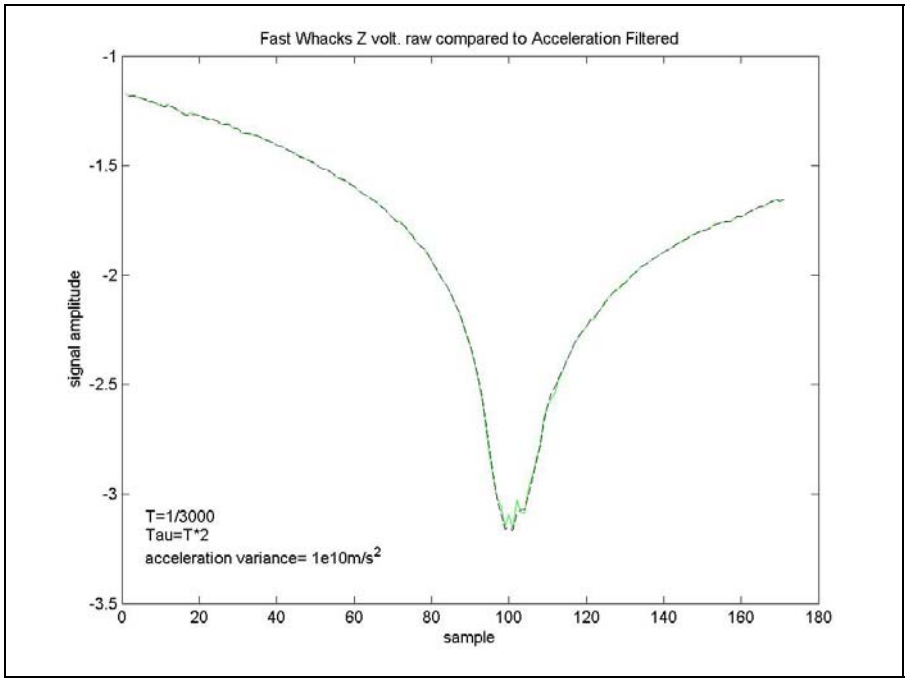

Figure 5.22. *z voltage Whack gesture filtered $\tau=2*T$ $\sigma^2=1e10 m/s^2$*

Values of τ=T*2 sec and a σ^2=1e10 m/s^2 were used. The value of τ makes sense since the acceleration is very impulsive, but the acceleration variance of 1e10 m/ s^2 seems a little extreme. Of course these parameters are not meant to give a smooth estimate of the track but rather stay precisely on the track at any cost.

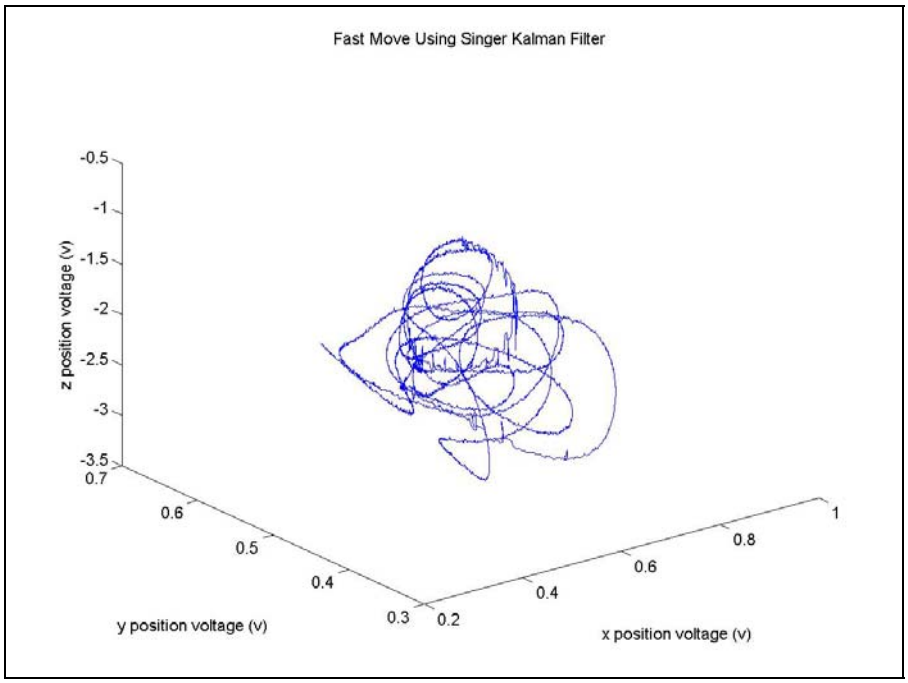

Figure 5.23. *Filtered 3D track of Fast Move Gesture using Whack parameters*

Figure 5.23 illustrates the pay off of accuracy versus variance. The same parameters used to track whacks were used to track the Fast Move gesture, see Figure 5.17. As you can see the variance of the estimates has increased, but the filtered track does not vary from the raw measurement. This is because we have specified large values for the covariance matrix of our dynamic noise process, which equates to a larger Kalman gain, forcing the filter to pay more attention to the measurements, see the Kalman gain in Equation 4.7. Even though the estimated track resembles the measurements more closely, there is some smoothing being achieved. It is clear that our three defined modes of gesture cannot simply be tracked by a single model. Next we discuss how our three models are combined to give a single improved position estimate regardless of gesture type.

5.2 The Interacting Multiple Model Estimator (IMM)

The Interacting Multiple Model Estimator (IMM) algorithm [36] provides a single output of the estimated state of a dynamic system from a combination of subfilter outputs, each of which, in our case, are tuned to a specific dynamic model. The weighting of each model's state estimate in the final combined state estimate is proportional to each model's likelihood function at the current iteration of the algorithm. The likelihood of each model is calculated from the normal distribution. Where *j* is the model index.

$$\Lambda_j = \mathrm{N}[z(k); \hat{z}_j(k), S_j] \quad (5.1)$$

Where *z(k)* are the current measurements, $\hat{z}_j(k)$ are the current model's predicted estimates, and S^j is the covariance of the model's innovations sequence, the difference between the predicted measurements and actual measurements.

Inputs to each model's filter at the start of an iteration, *k*, is a weighted combination of the outputs of each model's state estimate, $X_j(k-1)$, and covariance, $P_j(k-1)$ from the previous iteration, *k-1*. Weighting of each model *j*'s state estimate and covariance is governed by model *j*'s likelihood, Λ_j and model *j*'s switching probability to the current model. Since a first order Markovian process governs model transition, a model transition probability matrix, p_{ij} is defined; the transition probability from model *i* to model *j*.

A single iteration of the IMM algorithm involves five steps [36]:

1. Calculation of the mixing probabilities

The probability that mode M_i was in effect at k-1 and M_j is in effect at k:

$$u_{i|j}(k-1 \mid k-1) = \frac{1}{\bar{c}_j} p_{ij} u_i(k-1)$$

Where,

$$\bar{c}_j = \sum_{i=1}^{r} p_{ij} u_i(k-1)$$

r equals the number of models. In our case 3.

2. Filter initial conditions through mixing:

At each iteration of filtering a Kalman filter requires the previous filtered estimate, *x'(k-1|k-1)* and the previous filtered covariance, *P'(k-1|k-1)* as inputs. From these inputs the next prediction is made. In the IMM algorithm, the inputs to each subfilter are equal to a weighted combination of each subfilter's output at the previous iteration:

$$x'^{0j}(k-1 \mid k-1) = \sum_{i=1}^{r} x'^{i}(k-1 \mid k-1) u_{i|j}(k-1 \mid k-1) \quad \textit{for } j=1,..,r$$

3. Calculate likelihood of each model

The likelihoods are calculated for each model using *Z(k)*, the new measurements and each model's filtered output x^j*(k-1|k-1)* of each model.

4. Mode probability Update

Here we calculate the normalized probability for each model based on the likelihoods of the previous step:

$$u_j(k) = \frac{1}{c}\Lambda_j(k)\bar{c}_j \quad for\ j=1,\ldots,r$$

Where,

$$c = \sum_{j=1}^{r}\Lambda_j(k)\bar{c}_j$$

5. Subfilter combination

   The output estimate is a weighted combination of each individual subfilter's output:

$$x'(k \mid k) = \sum_{j=1}^{r} x'^{j}(k \mid k)u_j(k)$$

   The filtered state covariance is also updated in a similar way but is not shown here.

5.2.1 Gesture Mode Transitions

We choose to model the discrete switching of Radio Drum gesture modes as a first order Markov process. This means that the probability of being in the current mode, slow, fast, or whack, only depends on the previous mode or gesture performed. This is a fair assumption to make since our sampling rate, 3000 Hz, is high enough to ensure that a performer will not move through more than one mode in between sampling intervals. This allows us to define a mode transition probability matrix, (see

Table 5.1), used to govern the mixing probabilities for the first step of the IMM algorithm mentioned above.

Table 5.1 displays the mode transition matrix used in the IMM. We have labeled modes 1, 2, and 3 as ‘slow move’, ‘fast move’, and ‘whack’ respectively. The diagonal terms define the probability of staying in the current mode at the next sample. For example if the performer is in mode 1, or ‘slow move’, then they will stay in ‘slow move’ with a probability of 99%. The same for ‘fast move’, however, one will stay in ‘whack’ mode with only 40% probability. The off diagonal terms represent different mode transitions. Entry (1,2) is the probability of transitioning from mode 1 to 2 or ‘slow move’ to ‘fast move’. Entry (3,1) is probability of going from ‘whack’ to ‘slow move’ and so forth. There are a few important things to note here. The transitions from ‘slow move’ to ‘whack’ directly and vice versa are impossible. This simply means that the stick must continuously speed up or slow down before and after whacking the surface. The transitions (2,1) and (2,3) are the same because in a typical performance the performer is equally likely to whack or slow down if the stick is moving in a fast maneuver There is no ‘master’ set of transition probabilities that will work for all types of performance and it is difficult to intuitively set these values, they must be set through testing on different gesture types. We feel that the ones listed in the table worked well in our experiments.

The quantities in the transition matrix may be modified or customized for a specific performance. For instance if it was known before hand that a specific performance may involve slow gesture with no whacks then we may modify our transition table accordingly making the probability of transitioning into whack mode

zero. And vice-versa, if a performance is expecting only whacks then transitions to slow mode maybe set to zero.

$$\begin{bmatrix} 0.99 & 1-p11 & 0 \\ \frac{1-p22}{2} & 0.99 & \frac{1-p22}{2} \\ 0 & 1-p33 & 0.4 \end{bmatrix} = \begin{bmatrix} 0.99 & 0.01 & 0 \\ 0.005 & 0.99 & 0.005 \\ 0 & 0.6 & 0.4 \end{bmatrix}$$

Table 5.1. *Radio Drum performance mode transition probabilities*

5.3 IMM Tracking Results

In this section we will show the tracking results on a few different gestures using the interacting multiple model estimator.

To begin, an experiment was performed to see how the IMM filter would perform as the stick was moved from the surface of the Radio Drum to a height of 45cm over 1 second at constant velocity. The x and y positions were held constant as the stick traversed the z axis. Figure 5.24, Figure 5.25, Figure 5.26 show the z voltage, and x and y positions respectively. Contrary to the results in section 4.3.2.1 the noise in the z signal does not seem to increase much with height. This maybe due to phase cancellation by summing the antenna signals to get the z location. Further investigation is required to understand the z noise as a function of height.

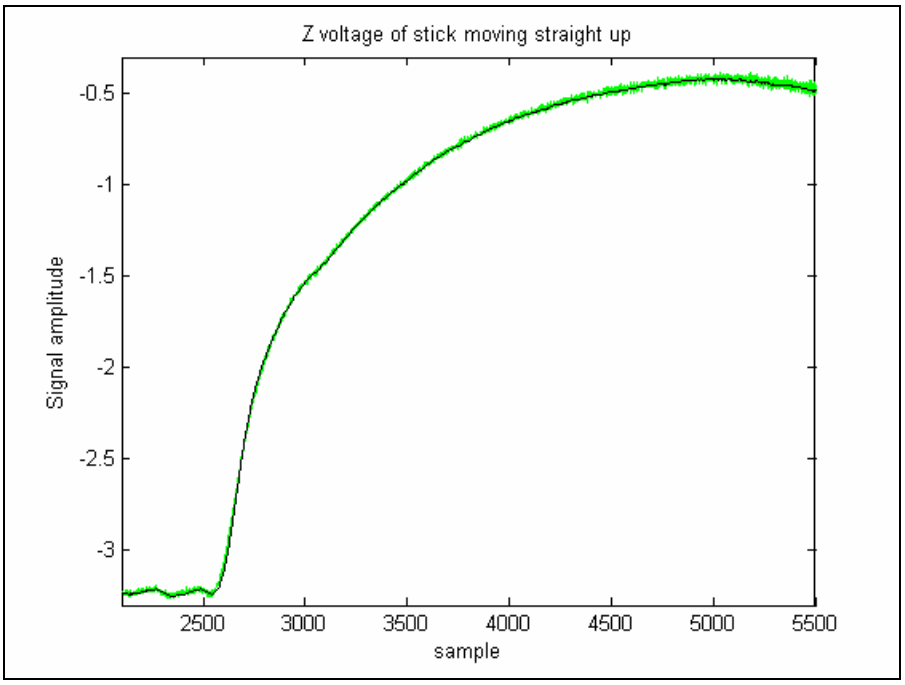

Figure 5.24. *Z stick location moving up to a height of 45cm*

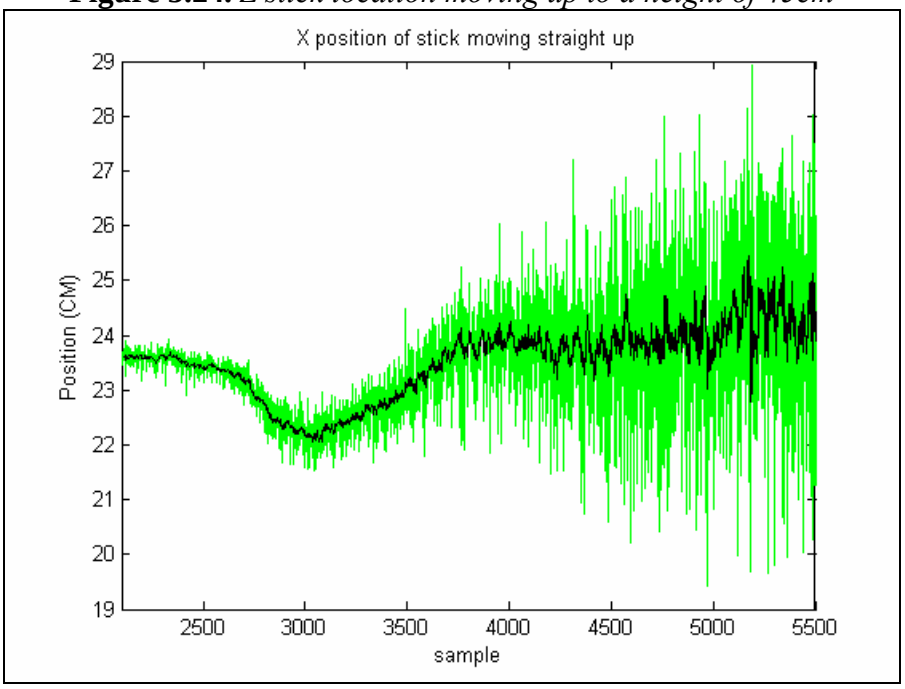

Figure 5.25. *The x location shows increasing noise and non-linearity as the stick moves up upwards*

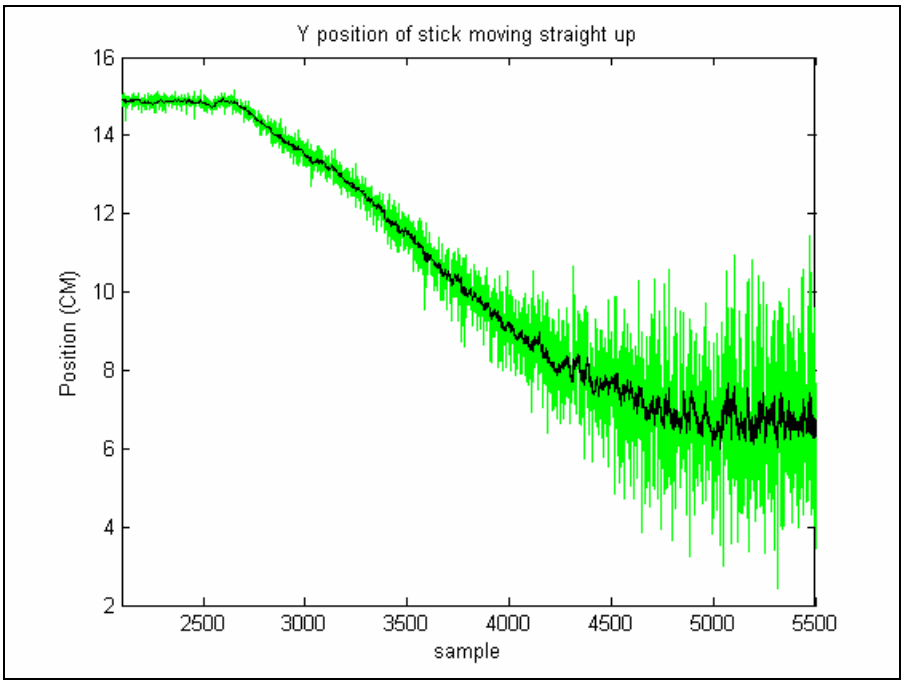

Figure 5.26. *The y location also shows increasing noise and severe non-linearity as the stick moves up upwards*

As expected the noise in the x and y location greatly increases with height. The x and y locations also begin to show non-linear movement not too high above the surface. The x location recovers as the stick reaches approximately 30cm but the y location continues to veer off to a deviation of 8cm as the stick reaches full height. The IMM filtered track, in black, manages to reduce the maximum uncertainty of the x position from 9cm to 3cm and from 7cm to 2cm for y as the stick reaches a height of 45cm. However, the IMM cannot compensate for the non-linearity in x and y. Further work must be done to understand the source of the non-linear correlation between z and x and y.

Figure 5.27 and Figure 5.28 show just over 2 seconds of a slow gesture with the stick moving to a maximum height of approximately 25cm around sample 2000. The noise on the x and y position estimates increases as the stick height increases.

The green line represents the raw unfiltered data obtained from the Radio Drum. The black line running through the green plots shows the filtered output of the 3 model IMM. The filtering has reduced the maximum uncertainty from 10 cm's to a few millimeters for the x position and from 7 cm's for the y position.

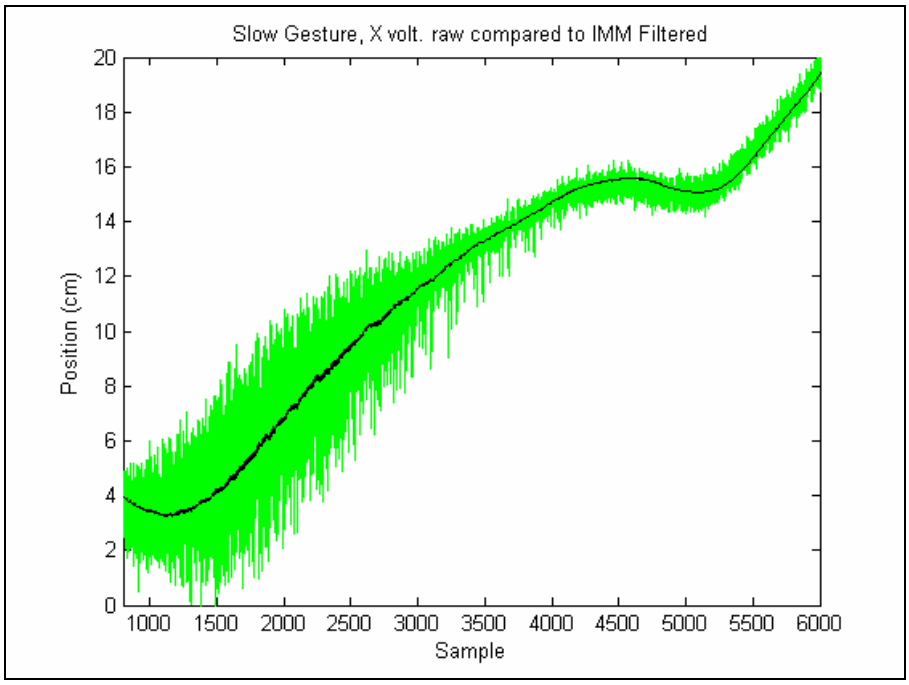

Figure 5.27. *IMM Filtered Radio Drum x position of slow gesture*

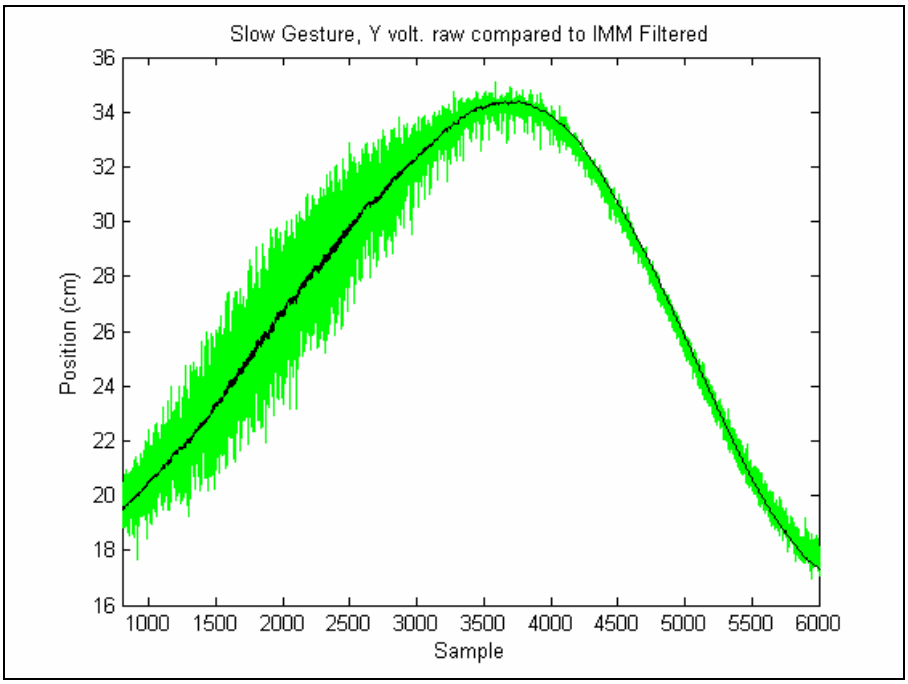

Figure 5.28. *IMM filtered Radio Drum y position of slow gesture*

We see here that the IMM successfully recognized the mode as 'slow move' and gave more weighting to the dynamic model tuned for smaller accelerations. However, the algorithm is a little slow to react to the y accelerations from sample 4000 to 4500; the filtered estimate tracks on the upper boundary of the measurements.

Figure 5.29 plots the raw and corresponding black IMM filtered z track for a slow to fast gesture transition. The raw data exhibits a correlated noise process. Although a reasonable position noise reduction from accuracy of 6cm to 1cm is observed, the IMM has trouble estimating the best track for the slow gesture up to sample 3400. The 'fast move' and 'whack' model's outputs are being favored over the 'slow move' model. An IMM favoring the 'slow move' model up to sample 3400 would give a position accuracy of a few millimeters. Since the 'fast move' and 'whack' models for the z axis have such large variances, the 'slow move' model's distribution gets 'swallowed' and never becomes more likely over the other models.

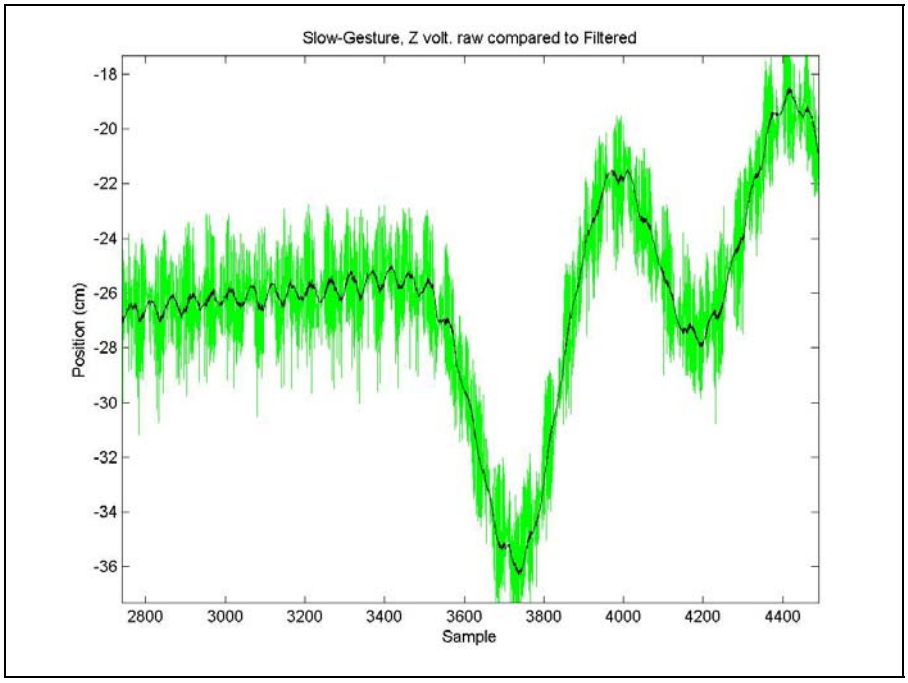

Figure 5.29. *IMM Filtered Radio Drum signal, slow to fast gesture*

In section 4.3.1 we discussed the correlated nature of the measurement noise in the Radio Drum system. Figure 5.29 illustrates how periodicity in the measurement noise can be mistaken as gesture by the Kalman filter. In this case the Kalman IMM output tracks the sinusoidal nature of the noise as if the performer is shaking the stick up and down at a rate of approximately 3000/100=30 Hz. Any tuning of the current Kalman filter IMM algorithm will not give a better track for the z position of this example.

Figure 5.30, Figure 5.31, and Figure 5.32 show the raw and IMM filtered x, y, and z coordinates as the stick moves through the three defined gesture modes. The gesture begins with fast movement that leads to a whack and finishes with slow

movement. Figure 5.33 shows a close up of the z coordinate at the moment of the whack. Figure 5.34 shows the normalized model probabilities.

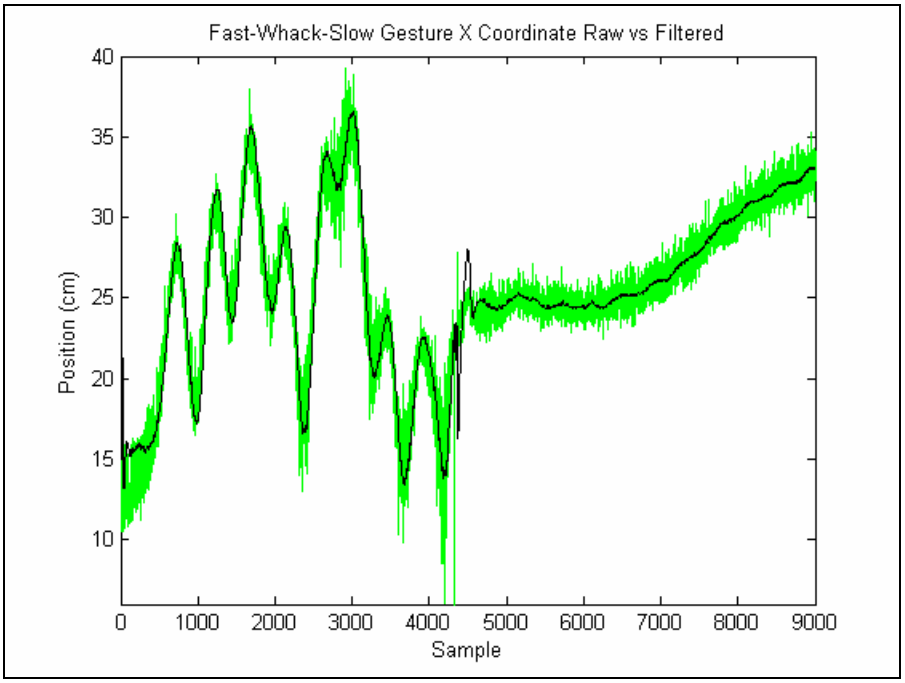

Figure 5.30. *X coordinate of a 'fast move' to 'whack' to 'slow move'gesture*

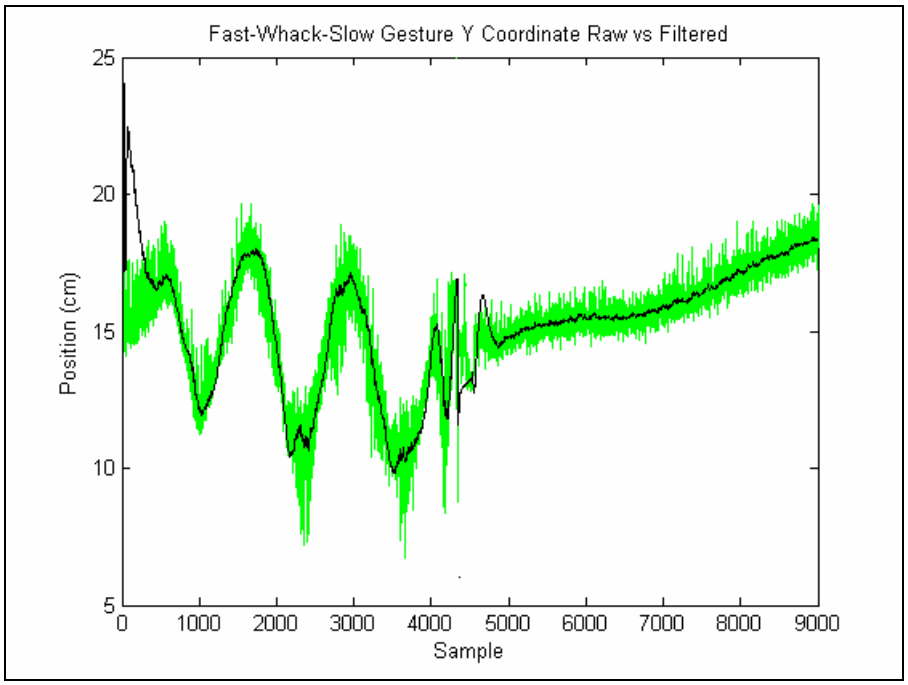

Figure 5.31. *Y coordinate of a 'fast move' to 'whack' to 'slow move'gesture*

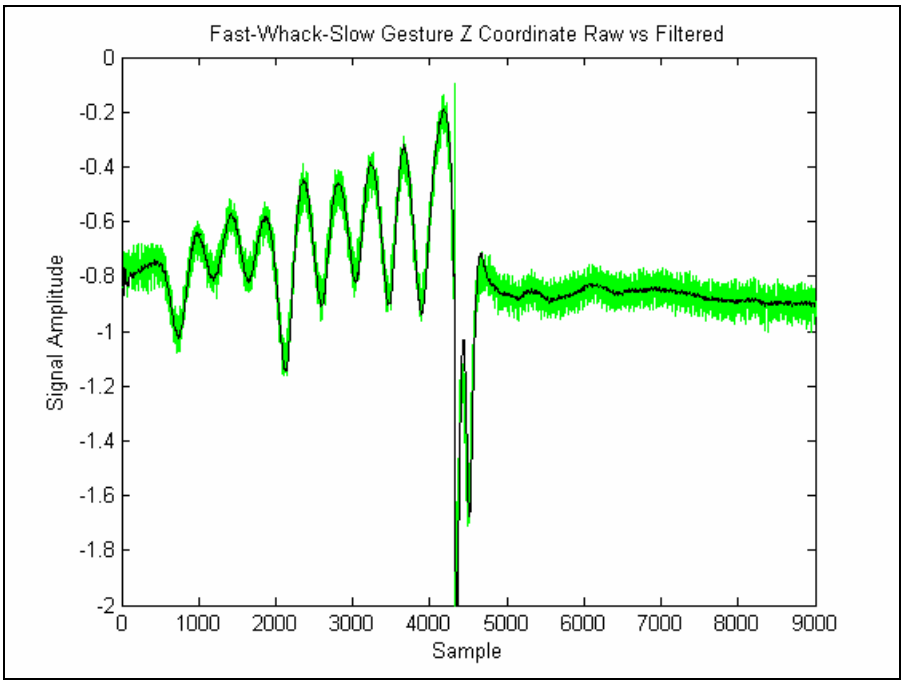

Figure 5.32. *Z coordinate of a 'fast move' to 'whack' to 'slow move'gesture*

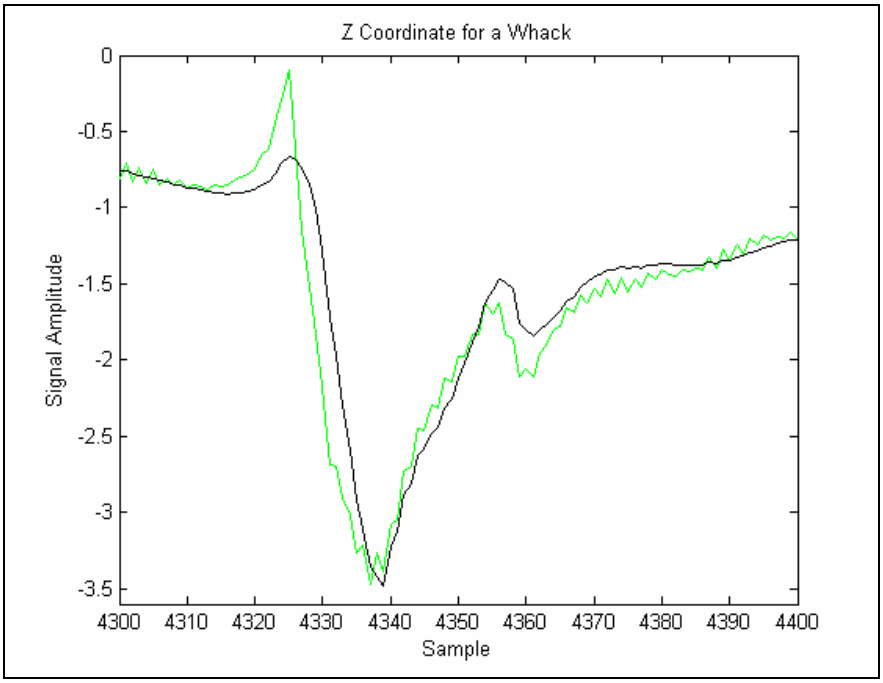

Figure 5.33. *Z coordinate of surface whack from a 'fast move' to 'whack' to 'slow move'gesture*

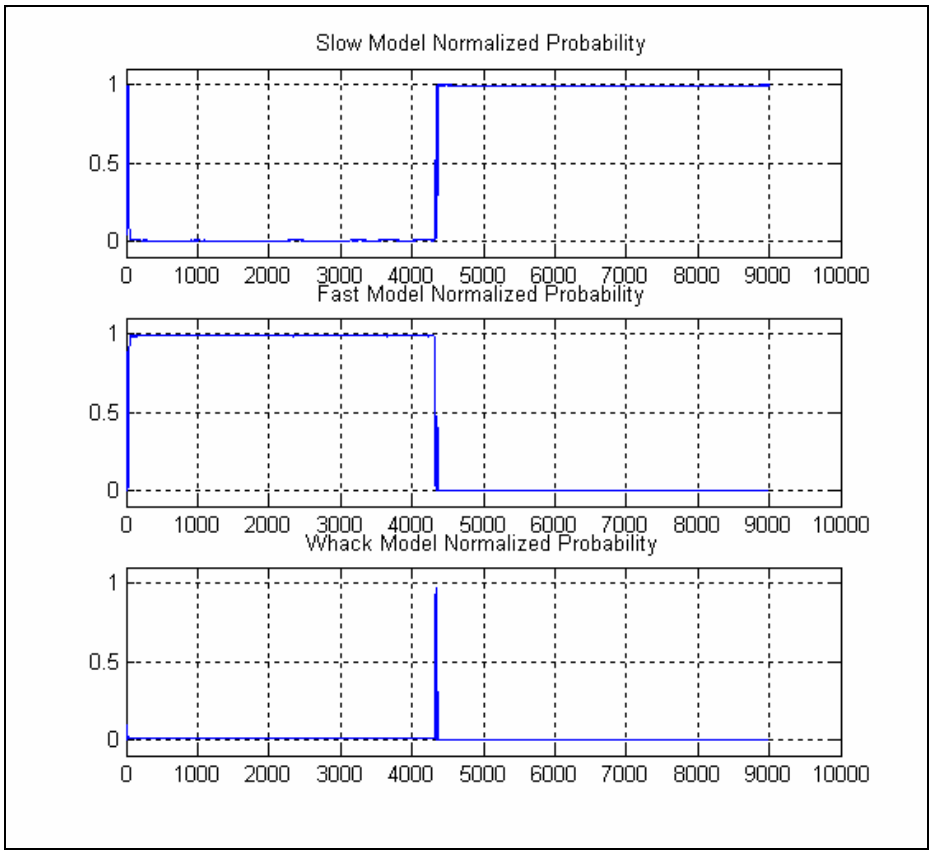

Figure 5.34. *IMM mode probabilities for Slow, Fast, and Whack models*

The IMM tracks the x, y, and z stick position relatively well, including the surface whack itself. When the whack is performed at around sample 4300 disturbance is created in the x and y position estimates causing slight momentary deviation from the intended track. However, from Figure 5.34 we can see that the IMM correctly tracks the slow to fast to whack gesture transitions. From the 0 to 4300 samples the IMM favours the 'fast move' model. At the moment of the whack the 'whack' model's probability jumps up. After the whack the IMM correctly favours the 'slow move' model.

As a comparison, Figure 5.35, is included to show how the fast-whack-slow gesture would be tracked if only a single Kalman filter, tuned for slow gestures, was used. We see that the single Kalman filter cannot follow the increased accelerations

of the fast and whack sections. However, it catches up to the slower section at the end. The z coordinate track of Figure 5.32 does a better job at following the intended gesture but with an increased variance in the position estimate.

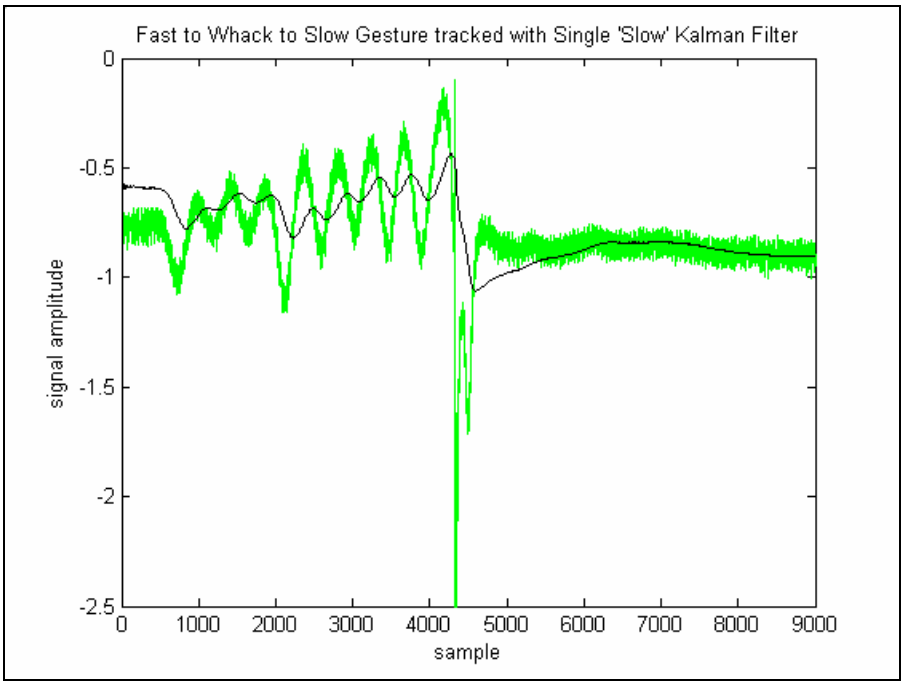

Figure 5.35. *Z coordinate of surface whack from a 'fast move' to 'whack' to 'slow move'gesture tracked with a single 'slow move' model Kalman filter*

5.3.3 Improvements to whack detection

IMM Filtering of our 3-D position signal leads to enhanced expressivity for the performer and a more accurate estimate of the x and y whack location.

5.3.3.1 Detection of Softer Whacks

The current whack detection algorithm looks for sign changes in the acceleration signal to determine when the stick has made contact with the Radio Drum surface. More precisely, a whack is detected if the velocity of the stick is at a minimum and the z position is below a certain threshold [28]. We look for an acceleration sign

transition from negative to positive to determine the minimum velocity. A statistical algorithm derived from sequential analysis theory [37] tracks sign changes of the noisy acceleration signal. A region of indifference is defined between zero and the acceptance point shown below. Where α and β are the miss and false alarm probabilities respectively, n is the expected number of samples needed before the acceptance point is reached and σ^2 is the variance of the acceleration signal [28].

$$\theta_0^2 = \frac{-2\sigma^2}{n}\left[\log\frac{1-\beta}{\alpha} + (1-\alpha)(\log\frac{\beta}{1-\alpha}) - \log\frac{1-\beta}{\alpha})\right]$$

In short, a smaller acceptance point, θ_0, leads to a more sensitive whack detector, and since θ_0 is directly proportional to σ^2, a smaller variance in the acceleration signal leads to the sensing of softer whacks, giving the performer a greater range of expression when striking the surface. The next three figures show the raw and IMM filtered z position, velocity, and acceleration of a single surface whack respectively.

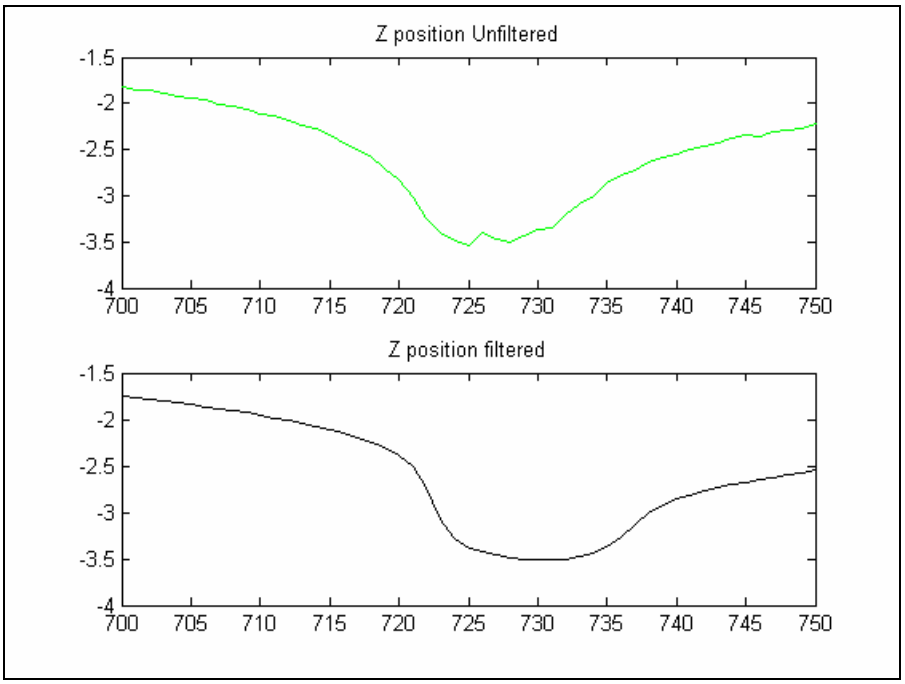

Figure 5.36. *Raw and IMM filtered Z position of a whack*

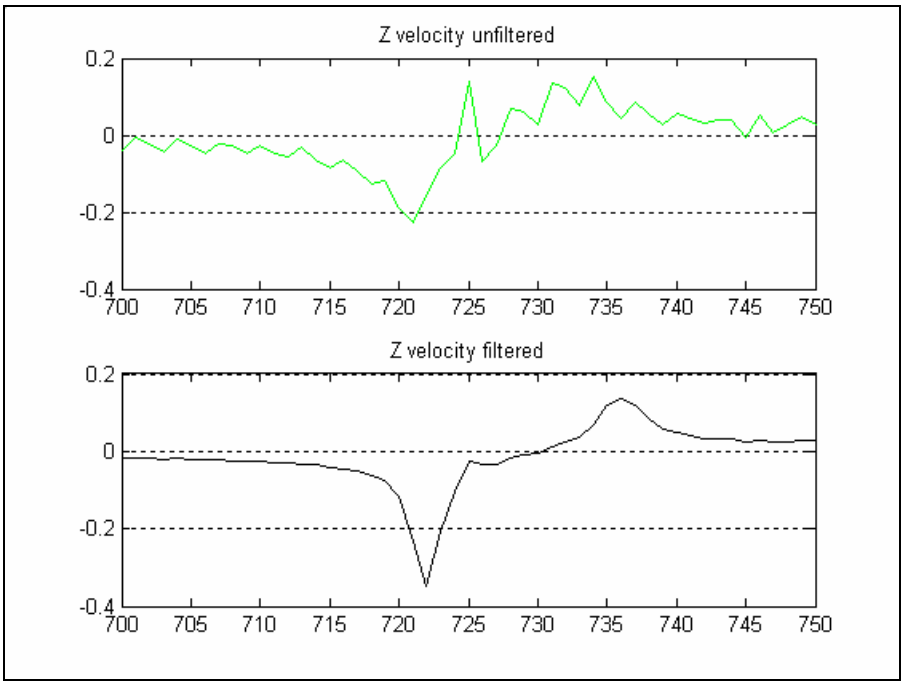

Figure 5.37. *Raw and IMM filtered Z velocity of a whack*

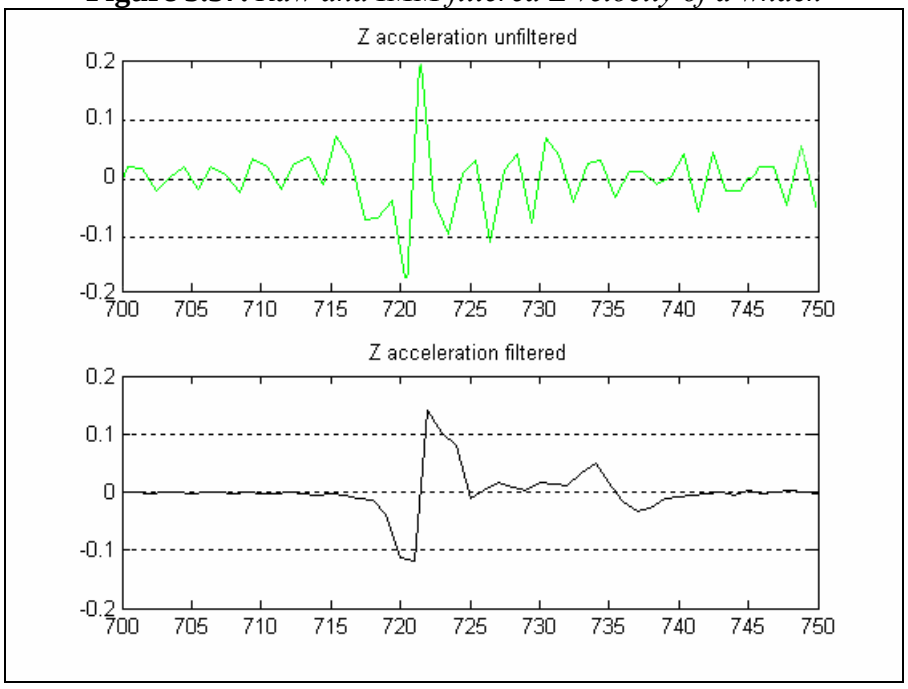

Figure 5.38. *Raw and IMM filtered Z acceleration of a whack*

Filtering of the position signal significantly reduces the variance in the velocity and acceleration estimate. A number of different filtered whacks were analyzed and in all

cases the acceleration signal exhibited smaller variance. Looking closely at the velocity and acceleration figures, we see that the IMM filtered whack trigger points (ie velocity minimum or acceleration zero point) lag behind the measurement by 1 sample. This introduces an additional latency of 32/44100=0.72 ms into our whack detection; a small price to pay for increased expressivity. Since our work is done in non real-time in the Matlab environment, we cannot truly test the improvements to whack gesture expression until a real-time implementation of the IMM filtering algorithm is written.

5.3.3.2 Improvements to x and y Whack Location

As the stick makes contact with the Radio Drum sponge surface sudden deviations in the x and y positions are observed, see Figure 5.39 and Figure 5.40.

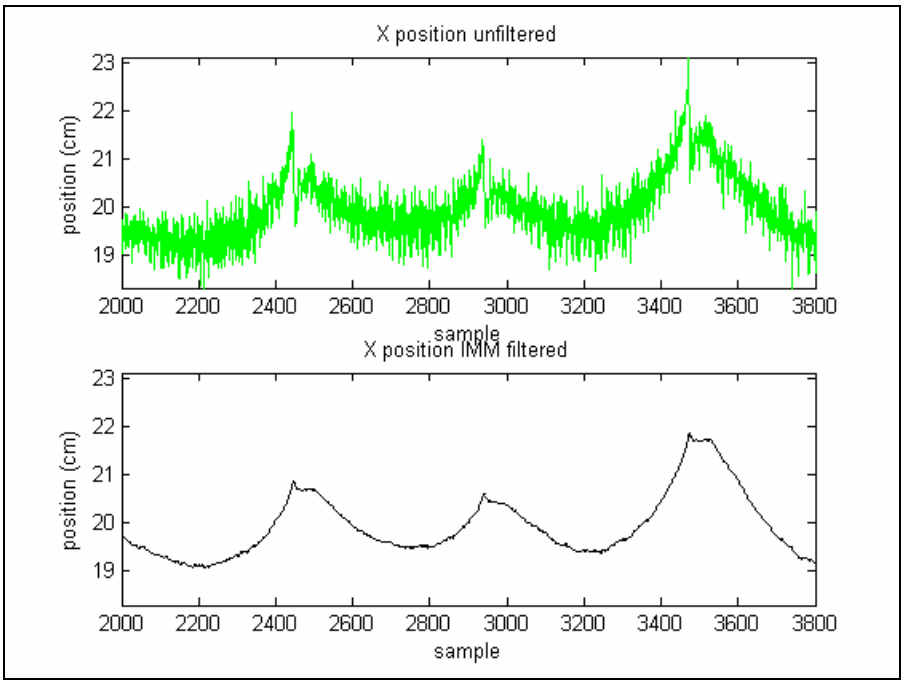

Figure 5.39. *X position during three surface whacks*

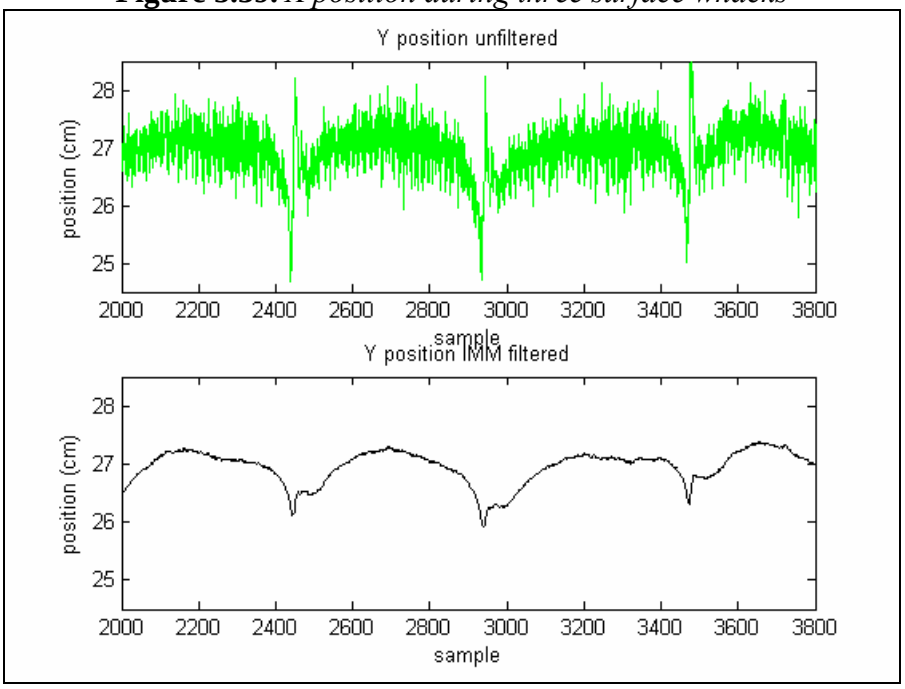

Figure 5.40. *Y position during three surface whacks*

Around the exact moment of contact the estimated raw x and y positions show discontinuous jumps as great as 4 cm when the actual x and y location of the stick is

not changing. Although the reason for these jumps has not been fully investigated, they are thought to originate from the release of built up electrostatic charge in the sponge pad on the Radio Drum's surface [28].

Shown on the second plot of the figures above, the filtering of the x and y position significantly reduces the effects of the jumps. However, the x and y positions still deviate in a range of 1-2.5 cm. Further investigations with alternate Radio Drum pad material should be performed to find the cause and solution to this problem.

5.3.4 Testing a Cheaper Audio Interface

Discussed in section 4.3.1.1**,** the Tascam FW-1804 audio interface can be purchased for as low as $400 US while the Fireface 800, currently used in the Radio Drum system, costs from $1500 - $2200 US depending on the retailer. Table 5.2 compares features of the two audio interfaces pertinent to the Radio Drum system.

Fireface 800	Tascam FW-1804
4 Microphone Inputs 60 dB gain	4 Microphone Inputs 50 dB gain
8 outputs, max 6.9 v rms	Stereo outputs, max 7.7 v rms
24 bit up to 192 KHz	24 bit up to 96 KHz

Table 5.2. *Fireface 800 versus the Tascam FW-1804*

Both interfaces have the features for a fully operational Radio Drum system. However, the Tascam is made of cheaper components, making it more susceptible to noise.

The Tascam audio interface was tested with a variety of Radio Drum gestures. The measurement model used for tracking the Fireface gestures had to be modified to work with the larger noise covariances of the Tascam. This was achieved by simply scaling up the previous measurement noise covariance matrix, used for the Fireface,

until satisfactory tracking of Tascam gestures was accomplished. A full noise assessment of the Tascam should be performed before any further experimentation is attempted. Below we show the x, y, and z raw and IMM filtered positions of a 'fast move' gesture.

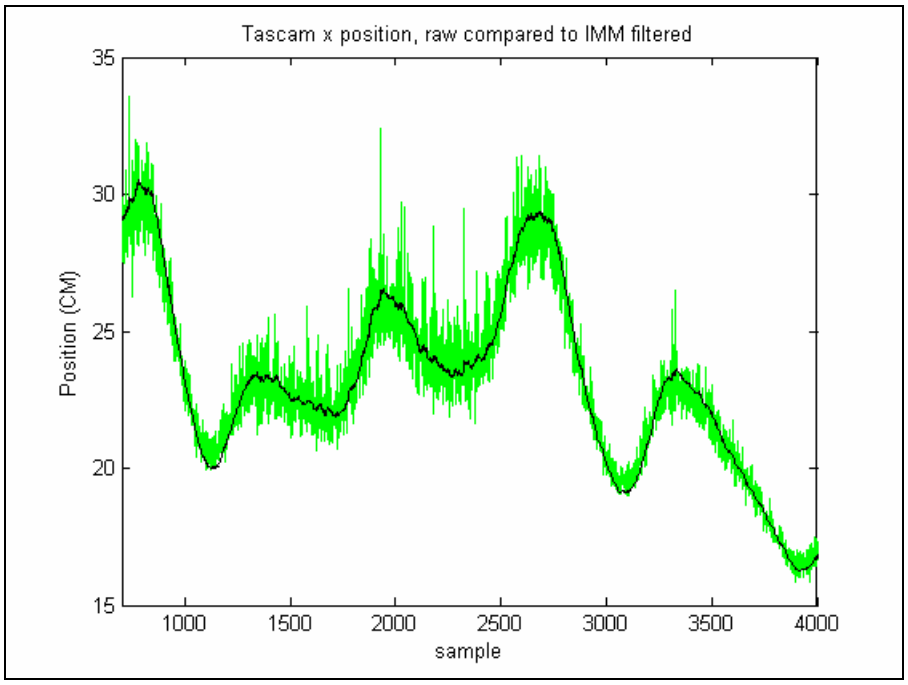

Figure 5.41. *x position of a 'fast move' gesture acquired with the Tascam*

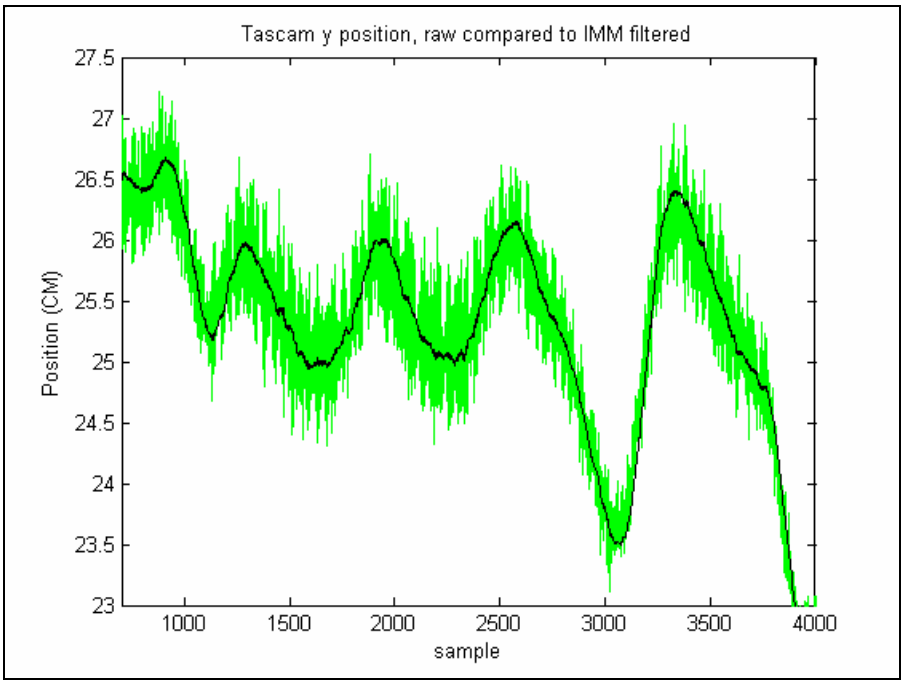

Figure 5.42. *y position of a 'fast move' gesture acquired with the Tascam*

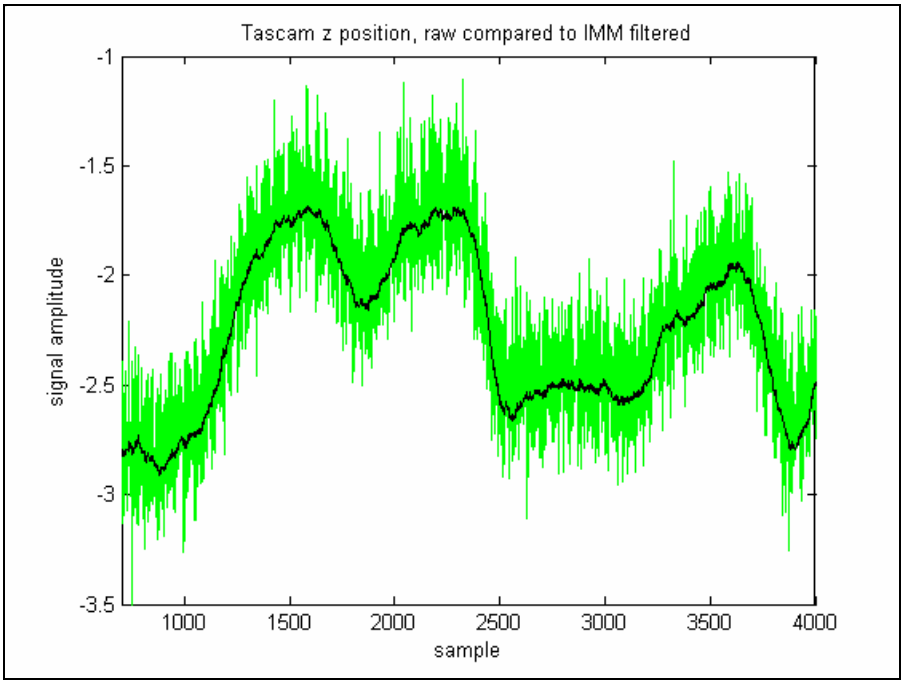

Figure 5.43. *z position of a 'fast move' gesture acquired with the Tascam*

At around sample 2000 in Figure 5.41 the raw x position data, acquired through the Tascam, shows a spike in the noise reaching from 24 cm to 33 cm. The IMM filter

successfully eliminates the outlier and reduces the average noise level of this ‘fast move’ gesture from 4 cm to less than 1 cm. Similar results can be seen in Figure 5.42 and Figure 5.43 where the y and z position is tracked through noisy measurements. Notice how the z coordinate vertical axis is labeled as signal amplitude and not as position. Due to the non-linearity of z coordinate we are not able to convert the z signal voltages into centimeters.

5.4 Discussion and Conclusion

In this chapter we discussed the development and testing of a multiple model tracking filter designed to adapt to the full range of gestures possible with the Radio Drum 3-D computer music controller.

Three linear dynamical models were tuned to track three discrete modes of gesture: ‘slow move’, ‘fast move’, and ‘whack’. Tracking performance of our Interacting Multiple Model estimator was evaluated on a variety of gestures spanning the three modes. Satisfactory results were observed for gestures acquired with both the higher quality Fireface 800 audio interface and the cheaper Tascam FW-1804.

We saw how IMM filtering of the Radio Drum signal can lead to enhanced expressivity and a more accurate x and y position estimate when whacking the surface. We also saw how correlated measurement noise can confuse the IMM filter and be wrongly perceived as gesture.

5.5 Future Radio Drum Work

There are many avenues for further research in dealing with improved gesture tracking of the Radio Drum. These are compiled in a list:

- Build an active filter to amplify signal and match impedance
- Different surface material to alleviate possible static charge build up
- Use estimation techniques to deal with correlated measurement noise
- Realtime implementation and testing in Max/MSP
- Tascam noise analysis if we want to use it in future
- Investigate non-linearity of Radio Drum and apply 3D function, possibly using Vicon Motion capture System
- Optimization of IMM Filtering and further analysis of parameters. Including mode transition matrix, dynamic model parameters, measurement noise.

Chapter 6

6 Tempo Tracking of North Indian Musical Performance Using a Kalman Filter

6.1 Introduction and Motivation

The "intelligence" of interactive multimedia systems of the future will rely on capturing data from humans using multimodal systems incorporating a variety of environmental sensors. Research on obtaining accurate perception about human action is crucial in building intelligent machine response. This chapter describes experiments for improving the accuracy of machine perception of tempo in the context of music performance using a Kalman filter based approach.

Conducting these types of experiments in the realm of music is obviously challenging, but fascinating at the same time. This is facilitated by the fact that music is a language with traditional rules, which must be obeyed to constrain a machine's response. Therefore the evaluation of successful algorithms by scientists and engineers is feasible. More importantly, it is possible to extend the number crunching into a cultural exhibition, building a system that contains a novel form of artistic expression, which can be used on stage.

More specifically, this chapter describes a multimodal sensor capturing system, capable of tempo tracking both melodic and rhythmic musical performance. Tempo tracking was attempted on a set of sitar and tabla performances separately.

Sensors for extracting performance information are placed on the performer and, in the case of the sitar, the instrument. A robotic drummer has been built to accompany the sitar player. In this research, the authors ask the question: Can tempo tracking of rhythmic and melodic performances be improved by combining audio and gesture data?

Analysis of accuracy of various methods of achieving this goal is presented. For each signal (sensors and audio) we extract onsets that are subsequently processed by Kalman filtering [31] for tempo tracking [38]. Late fusion of the tempo estimates is shown to be superior to using each signal individually. Kalman filtering lends itself well to the problem of tempo/beat tracking since, similar to the Radio Drum, noisy measurements (onsets) are observed from which a hidden state (tempo) is estimated.

The goal of this work is to improve tempo tracking in human-machine interaction. Tempo is one of the most important elements of music performance and there has been extensive work in automatic tempo/beat tracking on audio signals [39-44]. We extend this work by incorporating information from sensors in addition to the audio signal. Without effective real-time tempo tracking, human-machine performance has to rely on a fixed beat, making them sound dry and artificial. The area of machine musicianship is the computer music communities' term for machine perception. Robert Rowe (who also coined the term machine musicianship) describes a computer system which can analyze, perform and compose music based on traditional music theory [45]. Other systems which have influenced the community in this domain are Dannenberg's score following system [46], George Lewis's Voyager [47], and Pachet's Continuator [48]. The idea of extending traditional acoustic

instruments with sensors to capture performance information has been explored in [49].

Section 6.2 discusses our system as a whole: data acquisition, onset detection, the tempo tracker and late fusion techniques. Section 6.3 outlines the experimental procedure and shows the results of the experiments influencing design decisions for the real-time system. Section 6.4 contains concluding remarks.

6.2 Data Acquisition, Onset Detection, Tempo Tracking, and Late Fusion

For both melodic and rhythmic performances, onsets are detected from a variety of sensor signal streams and inputted into a parallel bank of Kalman filters. The outputs of the Kalman filters are fused to arrive at an estimate of tempo for each performance.

6.2.1 Data Acquisition

6.2.1.1 ESitar

Melody based Tempo tracking was done using the ESitar, discussed in 3.5. For our experiments we recorded data sets of a performer playing the ESitar with a WISP on the right hand to capture strumming wrist orientation. Audio data of the sitar was captured at a sampling rate of 44100 Hz. Thumb pressure and fret sensor data synchronized with audio analysis windows were recorded with Marsyas [15] at a sampling rate of 44100/512 Hz using Musical Instrument Digital Interface (MIDI) streams from the ESitar. Orientation data from the Open Sound Control (OSC) [16] streams of the WISP were also recorded synchronized to the audio window. We chose

to use a WISP to detect performer wrist movement due to the WISP's ability to capture subtle movements with small accelerations. Figure 6.1 shows a block diagram of our full system for tempo tracking of the ESitar.

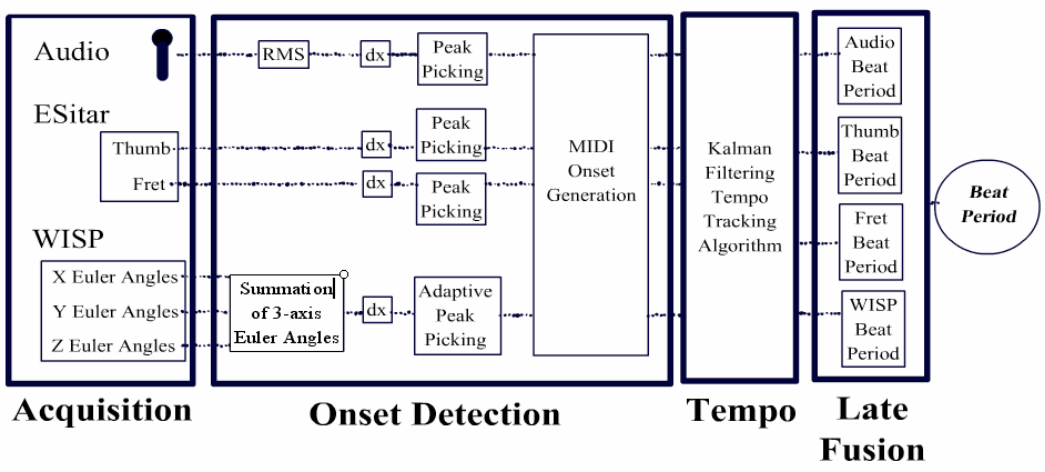

Figure 6.1. *Block diagram of ESitar tempo tracking*

6.2.1.2 Tabla Drums

Experiments on tempo tracking of rhythmic performance were conducted using tabla drums. The tabla is a traditional north Indian percussion instrument consisting of two separate drums, a dahina and a bayan. The dahina is a higher pitch drum requiring more erratic and typically quicker hand/wrist motions than that of the lower pitch bayan. 3 axes of acceleration data from a KiOm mounted on the performer's dahina wrist along with the audio signal were synchronized and captured at 44100/512 Hz and 44100 Hz respectively. We chose to use a KiOm, instead of a WISP, to capture the erratic wrist movements of a tabla drummer's dahina hand. The dahina hand of a tabla drummer experiences much higher accelerations than that of a sitar player's.

6.2.2 Onset Detection

Onset detection algorithms were applied to the audio and sensor signals to gather periodicity information of the music performances. A peak-picking algorithm is applied to the derivatives of each signal to find onset locations. Once a signal derivative exceeds a pre-defined threshold an onset is detected. A window of time, around 50-100 ms, must elapse before another onset can be detected. An adaptive peak-picking algorithm is applied on the WISP data to compensate for the large variability in wrist movement during performance.

6.2.2.1 Audio

The peaks of the first derivative of the Root Mean Square (RMS) of the audio signals were detected as audio onsets. Figure 6.2 and Figure 6.3 show the audio signal, computed RMS, and detected onsets for a few seconds of a tabla and sitar performance respectively. In both cases the detector does a good job at finding the onsets. However, some smaller amplitude onsets under the threshold are ignored.

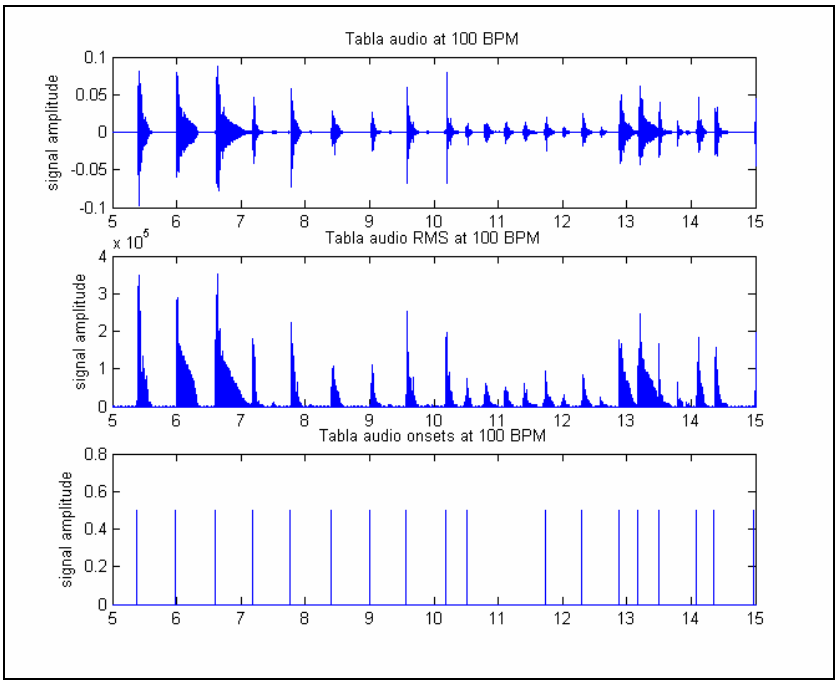

Figure 6.2. *Audio, RMS, and detected onsets of tabla performance*

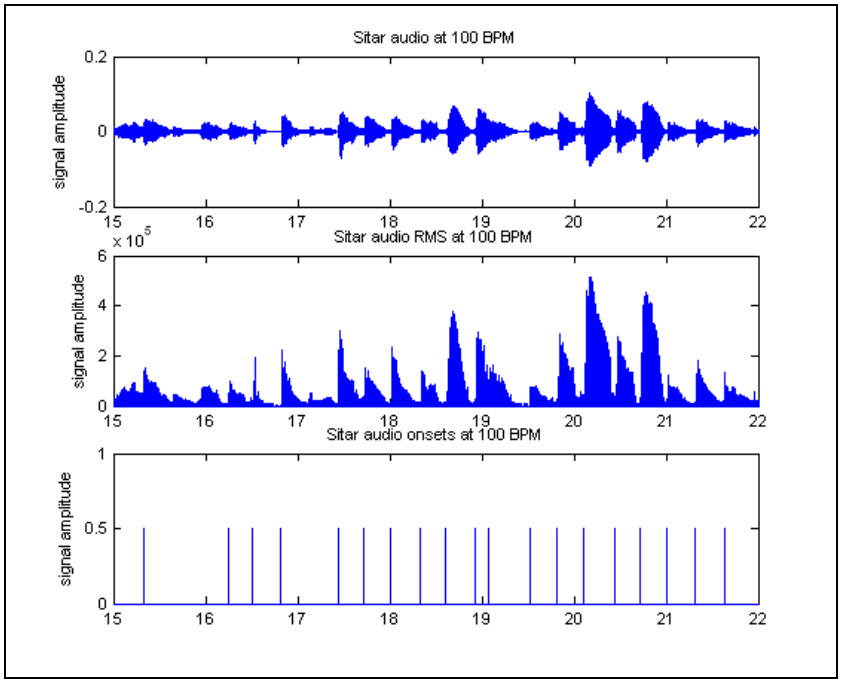

Figure 6.3. *Audio, RMS, and detected onsets of sitar performance*

6.2.2.2 Thumb Pressure

Figure 6.4 shows the same section of sitar audio plotted against the acquired thumb pressure data and detected onsets. Here we see that the detected onsets don't necessarily coincide with the audio onsets, however they do give us the onsets of an up or down strum. The up turned onsets represent down strokes, when the thumb sensor is being pressed, and the down turned onsets show the advent of up strokes.

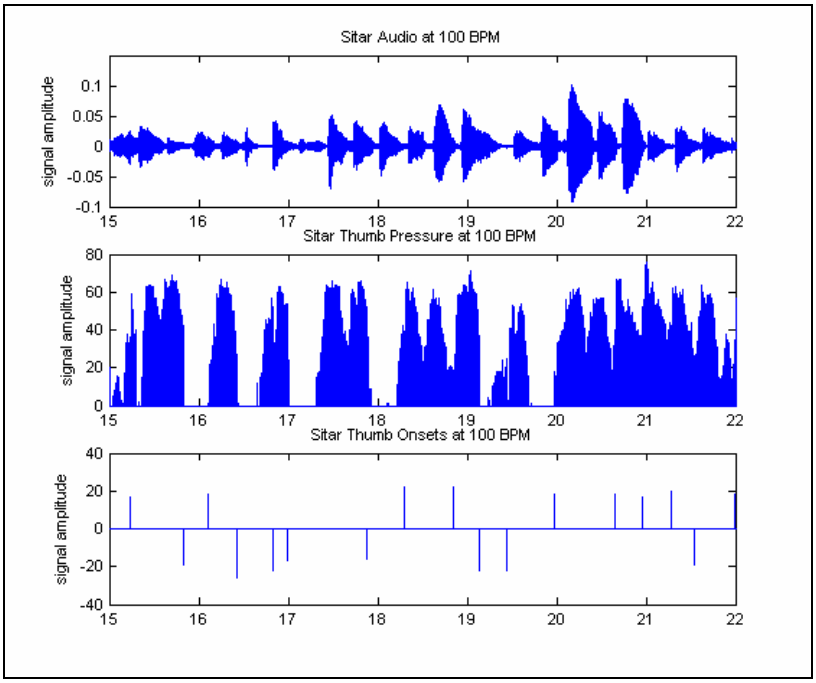

Figure 6.4. *Audio, thumb pressure, and detected thumb onsets of sitar performance*

Since we are interested in tracking the tempo of the performance and not the beat, the thumb data is useful to us, providing periodicity information about the player's rhythm.

6.2.2.3 Fret Data

Figure 6.5 show the audio from the same sitar performance along with ESitar fret data and the detected onsets from the fret data. Looking at the second plots we can see how the fret data maybe be modeled as discrete. Each fret on the ESitar is represented by its corresponding voltage. However there is noise in the signal, sometimes making it challenging to distinguish between adjacent frets.

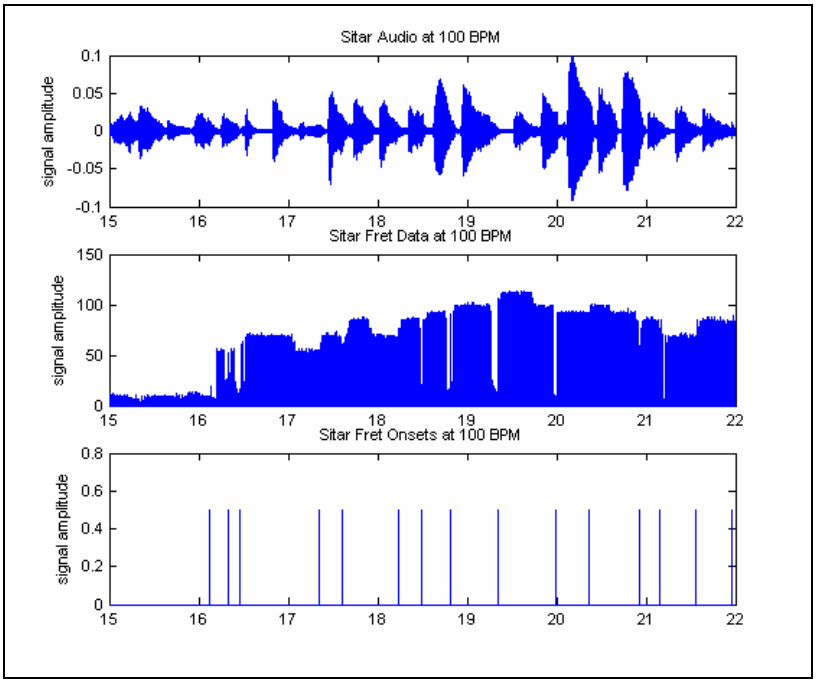

Figure 6.5. *Audio, fret data, and detected fret onsets of sitar performance*

Similar to the thumb pressure data the onsets from the fret data may not directly correspond to the audio RMS onsets. There maybe a delay between the instant when a sitar player touches the fret and when they strum the note. However, as the results will show, onsets from fret data can improve tempo tracking.

6.2.2.4 WISP

The first derivative of the magnitude of the three Euler orientation angles of the WISP was examined for strum gesture onsets. Figure 6.6 shows the same sitar audio snippet previously examined, the magnitude of the three Euler angles, defining the WISP's orientation, and the detected onsets.

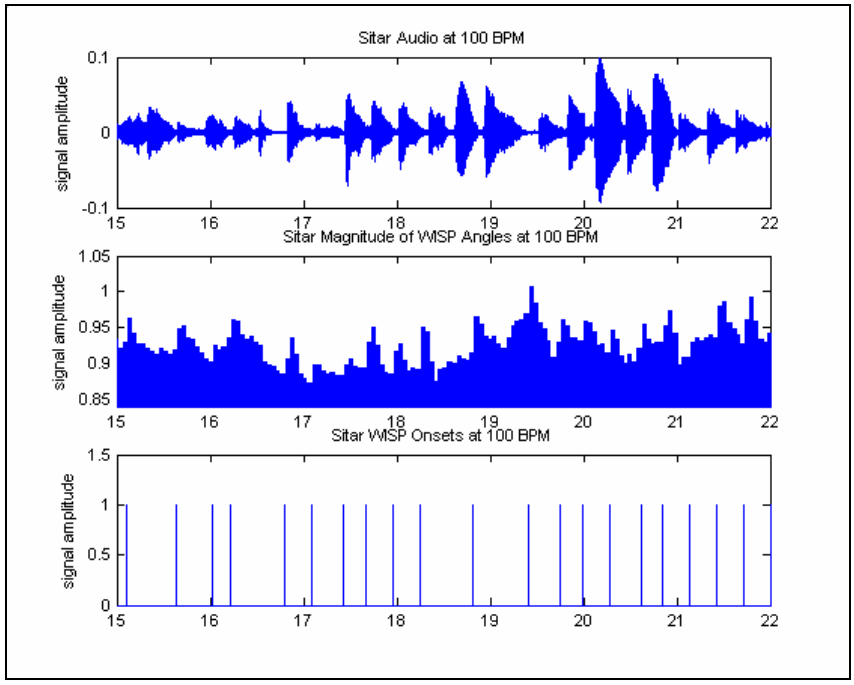

Figure 6.6. *Audio, WISP angle magnitude, and detected WISP onsets of sitar performance*

In the case of the WISP we used an adaptive threshold technique to track the large variations of the WISP magnitude signal. The equation below shows that our threshold, θ, at time n is a function of the previous onset and time.

$$\theta(n) = \alpha * mag[\tau(k-1)] * \beta^{(n-\tau(k-1))} \quad (6.1)$$

More specifically, the current threshold $\theta(n)$, is equal to a portion of the magnitude of the previous onset, $\tau(k-1)$, which slowly decays according to the β term. This term insures that an extremely high onset magnitude does not set our threshold too high thus preventing us from detecting anymore onsets. In our case $\alpha=0.6$ and $\beta=0.998$. These values were picked through trial and error.

6.2.2.5 KiOm

The first derivative of the magnitude of the three axis of acceleration of the KiOm was examined for drum gesture onsets.

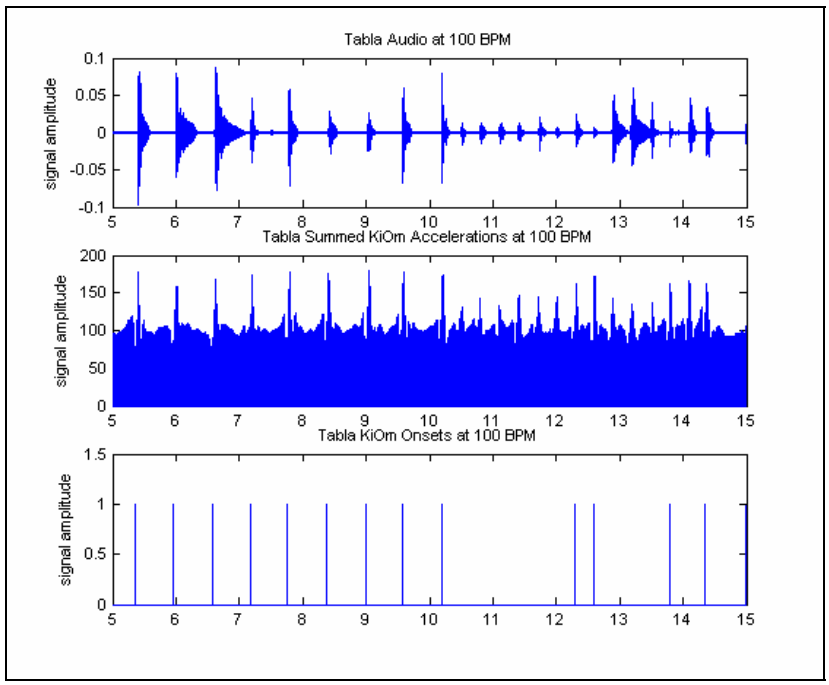

Figure 6.7. *Audio, KiOm acceleration magnitude, and detected KiOm onsets of tabla performance*

The detected onsets from the KiOm data are remarkably close to the actual audio onsets. This is due to the obvious drum hit gestures of a drummer, meaning that the

delay time between wrist motion and sound production on the drum membrane is minimal.

6.2.3 Kalman Filter Based Tempo Tracker

Real-time tempo tracking is performed using a bank of probabilistic Particle Filters. The algorithm tests various hypotheses of the output of each sensor stream's Kalman Filter against noisy onset measurements providing an optimal estimate of the beat period and beat [50]. Noisy onset measurements, extracted from the various sensor streams, are used as input to a real-time implementation of the tempo tracking algorithm [38]. In order to model the onset sequence we use a linear dynamical system as proposed in Cemgil [50]. The state vector x_k describing the system at a certain moment in time consists of the onset time τ_k and the beat period Δ_k defined as follows:

$$x_k = \begin{pmatrix} \tau_k \\ \Delta_k \end{pmatrix} = \begin{pmatrix} 1 & \gamma_k \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tau_{k-1} \\ \Delta_{k-1} \end{pmatrix} + w_k \qquad (6.2)$$

The current state vector depends on the previous state vector x_{k-1}, the beat period Δ_k and the score difference (γ_k). The score difference is defined as the rhythmic unit between two onsets in a musical performance. The bottom line of Figure 6.8 shows the value of γ_k as a score progresses.

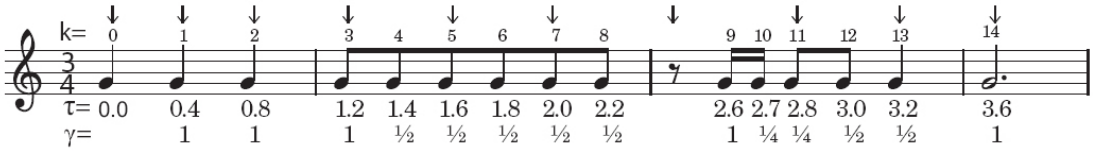

Figure 6.8. *Score with note index, score difference, and onset time (courtesy of Tim van Kasteren)*

During quarter note intervals the score difference is 1, during 8th notes ½, and during 16th notes ¼ and so on. We call γ_k a switching variable since its value in the dynamical system will change depending on when onsets occur. If we knew the score ahead of time we would be able to calculate the sequence of score differences and correctly predict the next onset in each case. However, in an improvised performance there is no predefined score, therefore some way is needed to decide on which value of γ_k to use at each time step. This is achieved through testing many hypotheses of γ_k and selecting the one with the highest probability. This process is known as particle filtering.

By setting the dynamic process noise, *w(k)*, interpreted as how well the performer stays on tempo, the system's sensitivity to tempo fluctuations can be controlled. In the case when the performer makes a momentary mistake and fumbles, we don't want our system to react right away but to observe the next few onsets and wait for the performer to recover from the mistake. If we expect significant tempo variations during the performance then we may increase the sensitivity to tempo change by increasing the dynamic noise. Recall from section 4.2.1 that the Kalman filter will place more weighting on the measurements as the covariance of the predicted estimate increases; the predicted estimate covariance is driven by the dynamic noise process.

For more details on the tempo tracker, including particle filtering, please refer to Appendix A. A fixed set of default filter parameter were used throughout all experiments. This was to ensure some kind of consistency to help determine the merit our multi-modal approach for onset generation.

6.2.4 Late Fusion

When looking at the estimated tempo tracks of each stream discontinuities are observed. In many of these cases the tracker is falsely tracking double, half or 3/2 errors caused by the wrong interpretation of the performance. 3/2 can be attributed to the tempo tracker falsely following triple meter onsets [51]. To counteract these issues we employ a simple late fusion of the various tempo tracker outputs to get a more accurate estimate of the tempo. For our experiments the trackers output a value for beat period every second. In the case of the ESitar we have four trackers running in parallel taking onset inputs from the RMS of the audio, the WISP, the thumb pressure sensor, and the fret data. In the case of the tabla we only have two: audio and KiOm.

Here is the late fusion Matlab code which includes all ESitar tempo tracking streams:

```
1   thresh=5; //BPM
2   for i=2:length(RMS)
3           if(abs(RMS(i)-temp)>thresh)                         //if there is an RMS discontinuity.
4
5                   if(abs(WISP(i)-temp)<thresh)                //if no discontinuity. in WISP tempo
6                   temp=WISP(i);                               //Use tempo from WISP stream
7                   out=WISP(i);
8                           elseif (abs(Thumb(i)-temp)<thresh)//if no discontinuity. in thumb
9   tempo
10                                  temp=Thumb(i);              //Use tempo from thumb stream
11                                  out=Thumb(i);
12                          else
13                                  temp=Fret(i);               //Use tempo from fret stream,
14  worst case
15                                  out=Fret(i);
16                  end
17           else
18                  temp=RMS(i)                                 //Use tempo from RMS stream
19                  out=RMS(i);
20          end
21                  Periods(i-1)=out;
22          end
23
24  end
```

Table 6.1. *Late Fusion Code*

This code looks for discontinuities in the estimated tempos from each stream in a hierarchical fashion. A discontinuity is defined by a tempo jump of thresh=5 bpm or more from one output to the next.

Line 3 decides if there is a discontinuity in the tempo track from the RMS, if so then we look to the WISP track. If the WISP tempo track has a discontinuity then we look to the thumb track. As a last resort we go to the tempo track from the fret data. If all streams have discontinuities then we keep the previous estimate of the tempo. Various permutations of these streams were explored before we were convinced of the best possible tempo estimates. This code example was used to give

an idea of how to deal with all the streams. Later we will see how ignoring some of the streams and combining others can give us the best possible tempo estimate.

Figure 6.9 shows the tempos derived from the four ESitar streams for a 40 second sitar performance at 120 BPM. The performer played along to a constant tempo click track. Here we see that the final combined estimate is better than any of the individual streams, alleviating the problem of large tempo discontinuities.

In the following section we explain our experimental procedure and discuss the accuracy of the various individual and fused estimated beat period streams for sitar and tabla performance are evaluated.

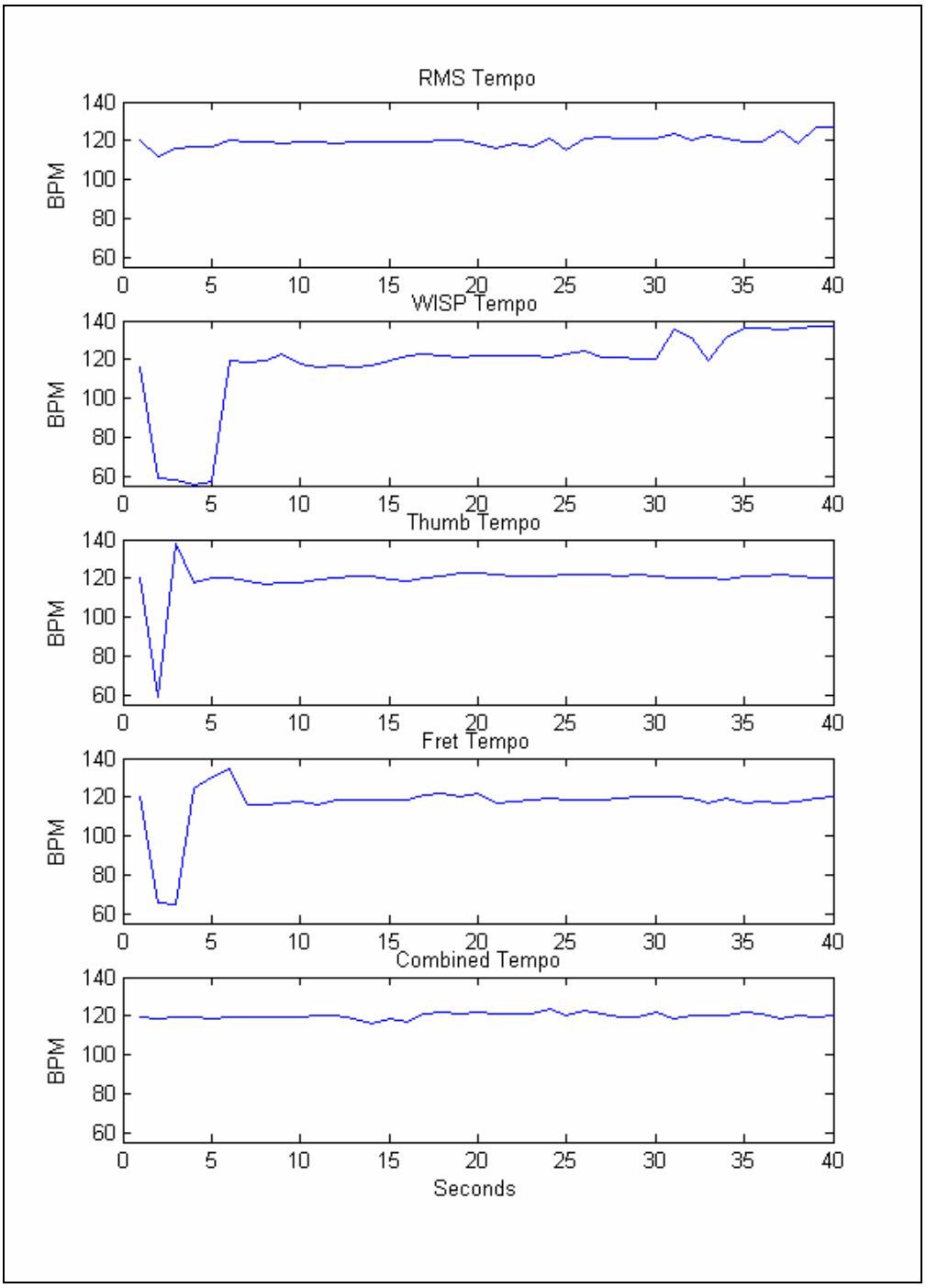

Figure 6.9. *The four streams of tempo (top 4 plots) combined to get the final estimate (bottom) for a 40 second 120 BPM performance of sitar*

6.3 Experiments and Results

6.3.1 Sitar

While playing the sitar, the performer listened to a constant tempo metronome through headphones. 104 trials were recorded, with each trial lasting 30 seconds. Trials were evenly split into 80, 100, 120, and 140 BPM, using the metronome connected to the headphones. The performer would begin each trial by playing a scale at a quarter note tempo, and then a second time at eight notes. The rest of the trial was an improvised session in tempo with the metronome.

Table 6.2 shows the percentages of frames for which the tempo was correctly estimated. We consider the tempo to be correct if it is with +/- 3 BPM of the intended tempo. We can conclude that when using a single acquisition method, the WISP obtained the best results at slower tempos, and the audio signal was best for faster tempos. Overall, the audio signal performed the best as a single input, whereas the fret data provided the least accurate information. Figure 6.10 and Figure 6.11 show summary histograms of the ratio between the estimated tempo and the correct tempo for the RMS and fret based tempo streams for the entire sitar data set. The histograms plot a total of 104 trials*30 secs = 3120 beat periods for the sitar data set. The ratios are plotted on a log2 scale where the zero point indicates correct tempo while -1, and +1 indicate half and double errors respectively. Errors of 3/2 noticed at 0.6 on the log2 scale can be attributed to the tempo tracker falsely following triple meter onset. The fret data exhibits large errors of 7/4. It is obvious here that the tempo track from the fret data is less accurate than that of the RMS.

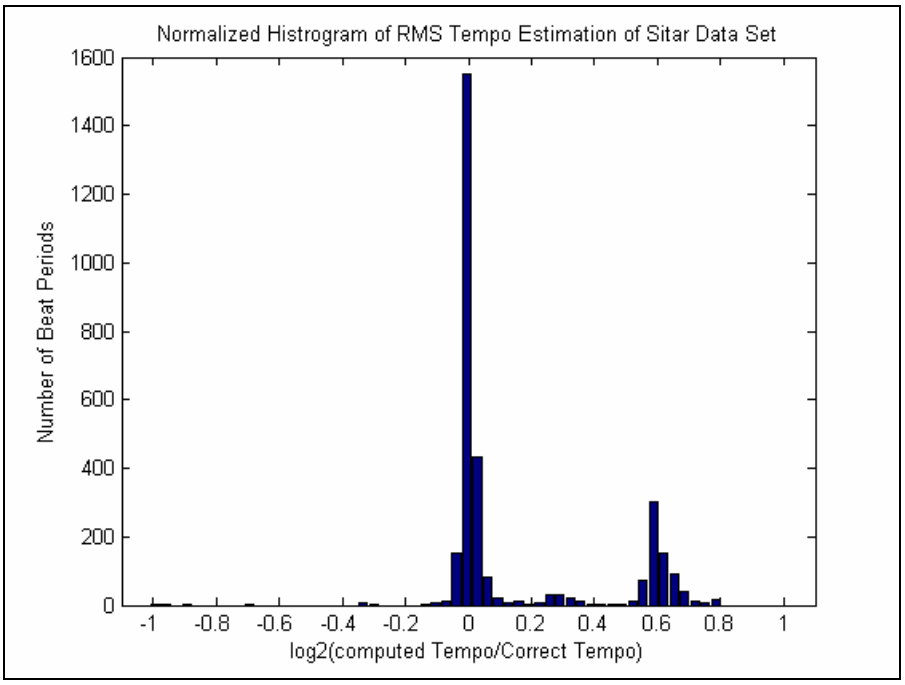

Figure 6.10. *Normalized summary log2 plot of RMS tempo tracking for the sitar data set*

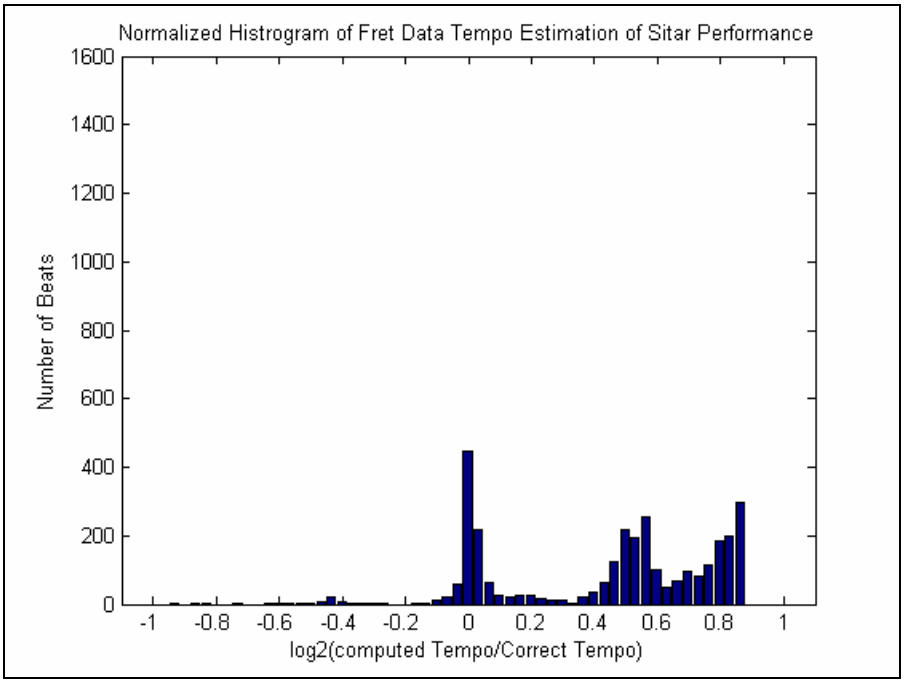

Figure 6.11. *Normalized summary log2 plot of fret data tempo tracking for the sitar data set*

When performing late fusion, the acquisition methods are considered in the order listed on bottom half of Table 1. For example the Audio/WISP/Thumb/Fret row shows the results of late fusion when the hierarchy of streams goes from Audio, being our best estimate to WISP to Thumb to Fret being our worst; similar to the late fusion algorithm of Table 6.1

Signal	Tempo (BPM)			
	80	*100*	*120*	*140*
Audio	46%	85%	86%	80%
Fret	27%	27%	57%	56%
Thumb	35%	62%	75%	65%
WISP	50%	91%	69%	53%
LATE FUSION:				
Audio/WISP/Thumb/Fret	45%	83%	89%	84%
Audio/WISP/Thumb	55%	88%	90%	82%
Audio/ WISP	58%	88%	89%	72%
Audio/Thumb	57%	88%	90%	80%
WISP/Thumb	47%	95%	78%	69%

Table 6.2. *Sitar tempo tracking results*

Overall, the audio stream fused with the thumb sensor gave the most accurate tempo estimate over all tempos, see Figure 6.12.

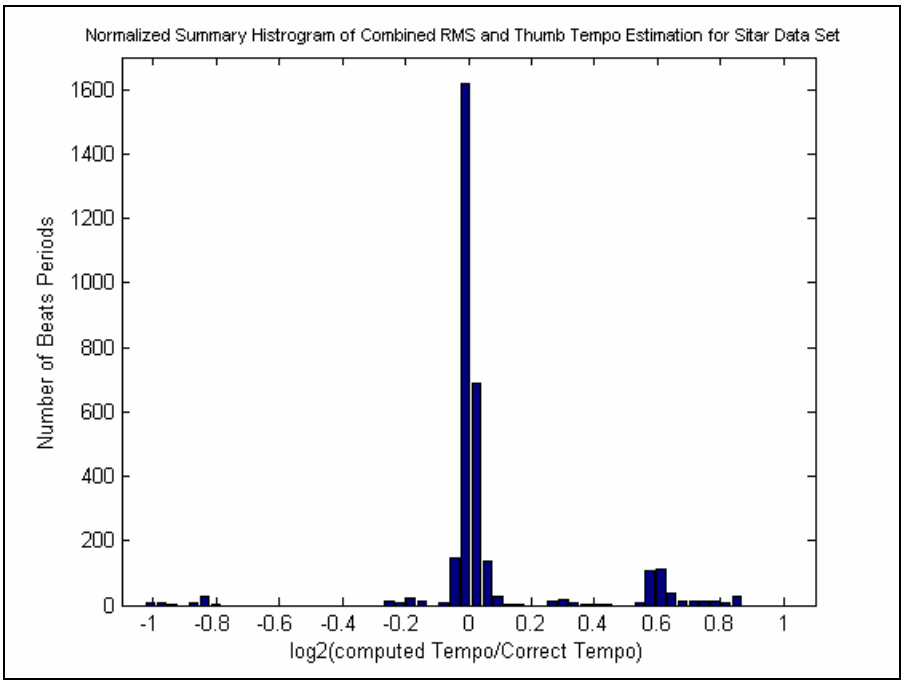

Figure 6.12. *Normalized summary log2 plot of fused RMS, and thumb data tempo tracking for the sitar data set*

We see that by fusing the thumb and RMS audio streams we get a more accurate estimate of tempo with a significant reduction in triple meter errors.

6.3.2 Tabla

While playing the tabla, the performer listened to a constant tempo metronome through headphones. 52 trials were recorded, with each trial lasting 30 seconds. Similar to the sitar data set, trials were evenly split into 80, 100, 120, and 140 BPM, using the metronome connected to the headphones. The performer would begin each trial by playing a simple fixed rhythmic pattern at a quarter note tempo, and then a second time at eight notes. The rest of the trial is improvised in tempo with the metronome.

Signal	Tempo (BPM)			
	80	*100*	*120*	*140*
Audio	67%	86%	79%	72%
KiOm	65%	81%	68%	60%
LATE FUSION:				
Audio/ KiOm	91%	93%	85%	80%

Table 6.3. *Tabla tempo tracking results*

Table 6.3 shows the tempo tracking results for 80, 100, 120, and 140 BPM of the tabla data set. We again see here that fusing the two streams provides us with a more accurate tempo estimate over all tempos. The greatest improvements are at 80 BPM. Figure 6.13, Figure 6.14, and Figure 6.15 show the summary histograms for the RMS, KiOm, and combined streams respectively. The fused results exhibit less scattering of outliers, however increasing the half tempo errors by a small amount.

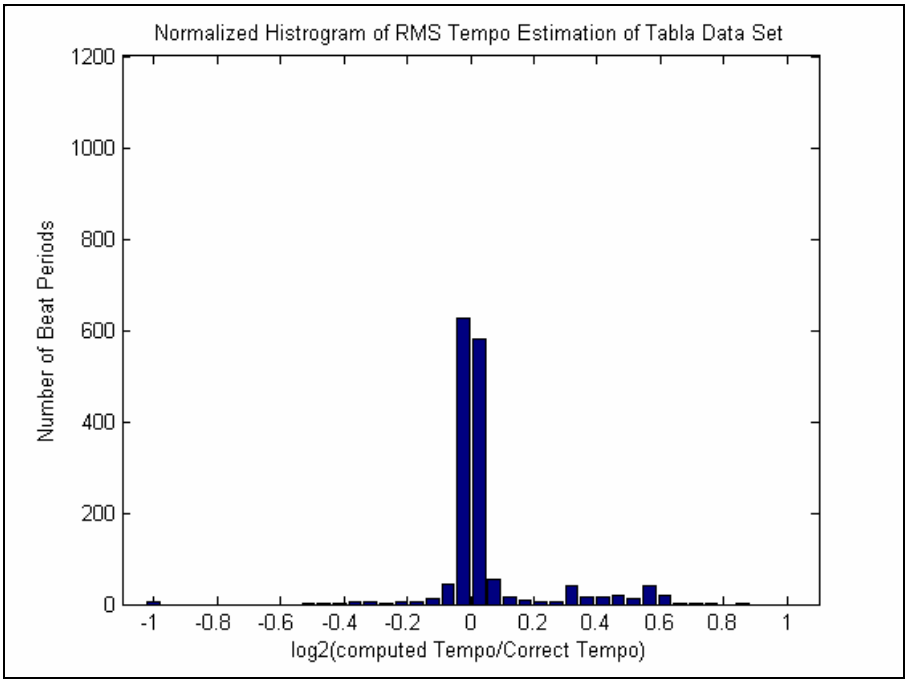

Figure 6.13. *Normalized summary log2 plot of RMS tempo tracking for the tabla data set*

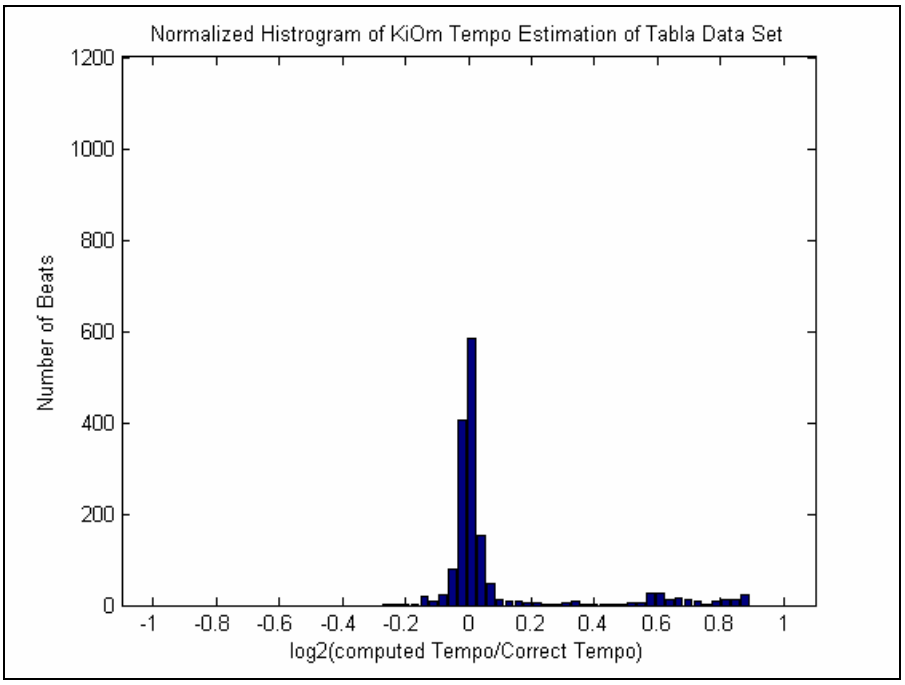

Figure 6.14. *Normalized summary log2 plot of KiOm tempo tracking for the tabla data set*

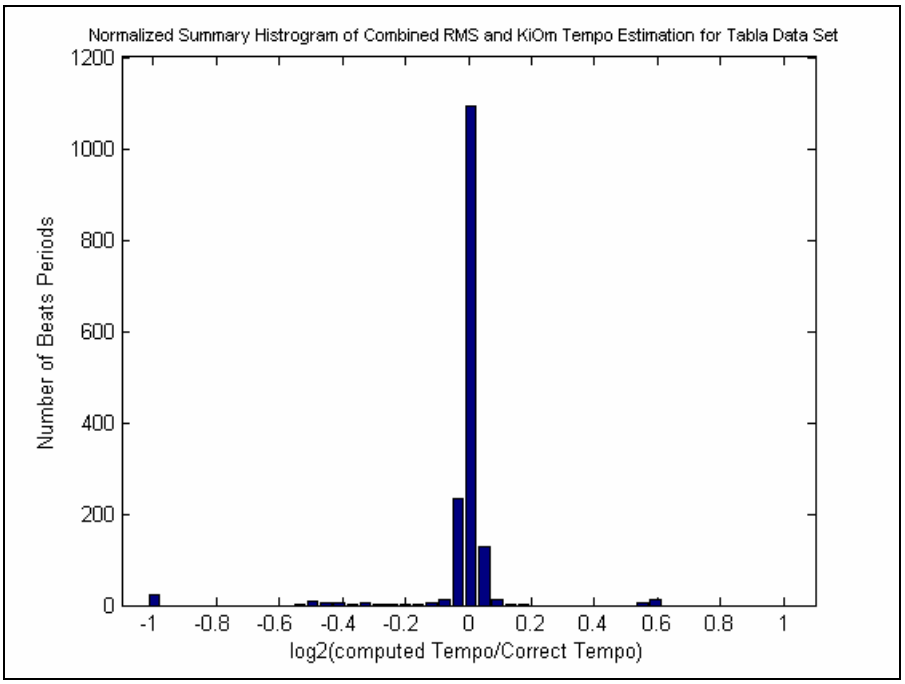

Figure 6.15. *Normalized summary log2 plot of fused RMS and KiOm tempo tracking for the tabla data set*

6.4 Concluding Remarks and Future Work

Tempo tracking of North Indian tabla and sitar performances was attempted. A variety of sensor streams, attached to both the performer's body and the instrument, were acquired to capture rhythmically relevant gesture information. Onsets were detected from the streams and input into parallel bank of Kalman filter based tempo trackers. The outputs of which were fused to arrive at a more accurate estimate of the performance tempo.

In the case of sitar, fusing of the thumb pressure and audio RMS tempo streams provided the best tempo estimate. In the case of the tabla experiments, fusing of the two streams greatly improved results. The worst tracking results and greatest improvements were seen at 80 BPM: an improvement of 12% for sitar and 24% for tabla. This makes sense since at slower tempos the inter-onset time increases thus leaving more room for human expression on top of the beat, thus more room for error.

For future work we would like to experiment with adjusting various parameter of the Kalman/Particle filtering algorithm to better adapt to various tempi and types of music. For instance increasing the dynamic noise, w(k), tells the model to expect greater performer deviation from the ideal onsets. This may improve tracking at slower tempos. The measurement noise, v(k), could be more accurately defined for each stream since the quality and precision of these onsets vary based on the sensor. Furthermore, a set of prior terms, as discusses in Appendix A, may favour a certain set of score differences or rhythmic units more for Indian music over western music. There is much research to be done in this area.

To further our research in multi-modal tempo tracking, we would like to design and execute experiments in which the performer plays to a slowly varying tempo. The figure below shows the tempo tracker successfully following a sequence of onsets that are speeding up.

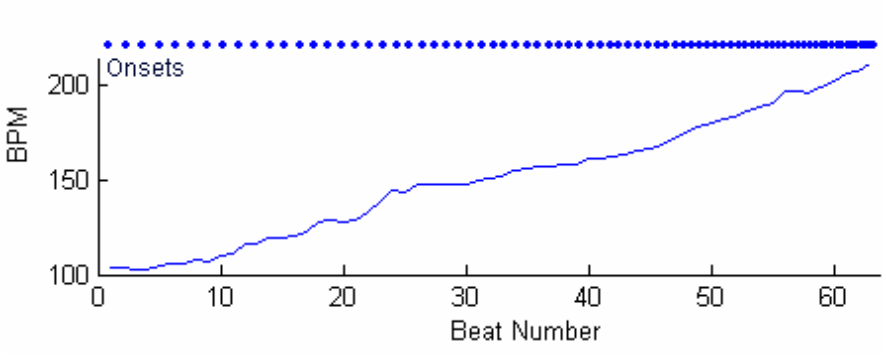

Figure 6.16. *Tempo track with increasing tempo*

Chapter 7

7 Conclusions and Contributions

In this thesis we concerned ourselves with two applications of the Kalman filter: Noise reduction of computer music controller and Real-time tempo tracking of rhythmic and melodic performances. Both of these engineering problems involve the estimation of the hidden state of a time-varying dynamical system achieved through the processing of observable noisy measurements with knowledge of the expected measurement and dynamic noise processes. In the case of the Radiodrum, the hidden state was comprised of the x, y, and z positions, velocities, and accelerations and the measurements were the x, y, and z positions derived from the noisy antenna signals. The measurement noise was defined by sampling stationary antenna signals at various points in the domain of the Radiodrum and the dynamic noise was defined for three gesture types using the Singer model for maneuvering targets. In the case of tempo tracking the hidden state consisted of the current onset time and the beat period. We consider the evolution of this state as dynamic since the beat period changes over time depending on the onsets. The measurements, in this case, are noisy onsets derived from analysis of audio and performer gesture data. Default parameters for measurement and dynamic noise were used and held constant over the course of all experiements.

7.1 Conclusions: Radiodrum Noise Reduction

Noise reduction on the Radiodrum stick's position estimate was achieved using a multiple model approach involving three parallel Kalman filters. Three linear dynamical models were designed to describe the motion of a Radiodrum stick moving with slow, fast, and extreme accelerations. Extreme accelerations correspond to the stick 'whacking' the surface of the drum. A single measurement model was defined based on the variance of stationary stick data at various locations in the Radiodrum domain. Three Kalman filters, with unique dynamical systems, were used to process the noisy x, y, and z measurement data and track the hidden x, y, and z positions of the dynamical system. An Interacting Multiple Model Estimator was designed to combine the three sub-filters outputs to provide a single estimate of the x, y, and z positions for all possible gestures of the Radiodrum.

Through the testing of our system on a variety of gestures we showed that IMM filtering of the Radiodrum measurement data leads to lower noise in the x, y, and z estimates and increased expressivity when whacking the surface. We also showed that the cost of our system could be greatly reduced through the use of a cheaper audio interface coupled with the IMM filtering.

7.2 Contributions: Radiodrum Noise Reduction

Here we list the novel aspects of the author's work discussed in this thesis pertaining to the Radiodrum system.

- Detailed noise analysis of the Radiodrum system, including the definition of an x, y, z noise covariance matrix used in the IMM
- Design of three dynamical systems to model the diverse motion of a Radiodrum stick. Acceleration variance and time correlation constants were experimentally defined for each gesture type
- Design and implementation of an Interacting Multiple Model Estimator. This involved the definition a mode transition probability matrix.
- Software implementation of IMM Kalman filter system in Matlab
- Testing of various gesture types using two different audio interfaces in preparation for a real-time version.

7.3 Conclusions: Real-Time Tempo Tracking

Real-time tempo tracking experiments were performed on North Indian melodic and rhythmic musical performances using a parallel bank of Kalman based particle filters. A complete real-time Kalman filter based tempo tracker, designed by Ali Cemgil [50] and implemented by Tim van Kasteren [38] was modified to track multiple streams of musical onsets acquired through analysis of audio and sensor data of the performer. The outputted tempo estimates from each stream was then combined in a late fusion stage to arrive at a more accurate final tempo estimate.

For the melodic experiments, 104 sitar performances at 80, 100, 120 , and 140 BPM's were tracked using the audio and sensor data from the ESitar[1], including thumb pressure and fret signals, and orientation data from the WISP[52]. For the rhythmic experiments, 52 tabla performances also at 80, 100, 120, and 140 BPM were tracked using the audio from the tabla and acceleration data from a wrist mounted KiOm[53].

In both cases the tempo estimates were improved when audio and sensor streams were fused. For the sitar, fusing of the audio with the thumb pressure sensor provided the overall best tracking performance. In the case of the tabla, fusing the audio with the KiOm stream also significantly improved the tempo track.

7.4 Contributions: Real-Time Tempo Tracking

Here we list the novel aspects of the author's work discussed in this thesis pertaining to the tempo tracking.

- Derived onsets from RMS feature from audio, ESitar thumb pressure, ESitar fret data, WISP orientation data, and KiOm acceleration data
- Developed an adaptive threshold algorithm for the WISP data
- Modified C++ tempo tracking code to work with parallel data streams and a variety of onsets
- Designed and implemented a late fusion scheme to combine sub-filter outputs
- Designed tempo tracking experiments and evaluation techniques

Bibliography

[1] A. Kapur, A. Lazier, P. Davidson, R. S. Wilson, and P. R. Cook, "The Electronic Sitar Controller," presented at New Interfaces for Musical Expression, Hamamatsu, Japan, 2004.

[2] B. Till, M. Benning, N. Livingston, "Wireless Inertial Sensing Package (WISP)," presented at New Interfaces for Musical Expression, New York, U.S.A, 2007.

[3] A. Kapur, A. Tindale, P. Driessen, "The KiOm: A Paradigm for Collaborative Controller Design," presented at International Computer Music Conference, New Orleans, U.S.A, 2006.

[4] G. S. M. K. Islam, "Detection and restoration of sound of flute embedded in noise using real-time Kalman filter," presented at IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY, U.S.A, 1993.

[5] A. Bari, S. Canazza, D. Giovanni, and M. A. Gian, "Towards a Methodolody for the Restoration of Electroacoustic Music," *Journal of New Music Research*, vol. 30, pp. 351-363, 2001.

[6] M. Niedzwiecki, "Identification of time-varying processes in the presence of measurement noise and outliers," presented at 11th IFAC Symposium on System Identification, Tokyo, Japan, 1997.

[7] M.Ning, R. A. Goubran, "A Perceptual Kalman Filtering-Based Approach fir Speech Enhancement," presented at IEEE 7th International Symposium on Signal Processing and its Applications, Paris, France, 2003.

[8] A. T. Cemgil, "Probabalistic Phase Vocoder and its Application to Interpolation of Missing Values in Audio Signals," presented at 13th European Signal Processing Conference, Antalya, Turkey, 2005.

[9] M. Reynolds, B. Schoner, J. Richards, K. Dobson, and N. Gershenfeld, "An Immersive, Multi-User, Musical Stage Environment," presented at Siggraph, Los Angeles, U.S.A, 2001.

[10] D. E. Sturin, M. S. Brandstein, and H. F. Silverman, "Tracking Multiple Talkers Using Microphone-Array Measurments," presented at International Conference on Acoustics, Speech, and Signal Processing Munich, Germany, 1997.

[11] U. Klee, T. Gehrig, and J. McDonough, "Kalman Filters for Time Delay of Arrival-Based Source Localization," *EURASIP Journal on Applied Signal Processing*, vol. 2006, 2006.

[12] K. Nishi, S. Ando, and S. Aida, "Optimum Harmonics Tracking Filter for Auditory Scene Analysis," presented at International Conference on Acoustics, Speech and Signal Processing, Atlanta, U.S.A, 1996.

[13] A. Sterian and G. H. Wakefield, "A Model-Based Approach to Partial Tracking for Musical Transcription," presented at SPIE Conference, San Diego, U.S.A, 1998.

[14] H. Satar-Boroujeni and B. Shafai, "Kalman Filtering Application in Automatic Music Transcription," presented at IEEE Conference on Control Applications, Toronto, Canada, 2005.

[15] T. Marrin, "Possibilities for the Digital Baton as a General-Purpose Gestural Interface," presented at Special Interest Group on Computer-Human Interaction, Atlanta, U.S.A, 1997.

[16] J. Paradiso, "The Brain Opera Technology: New Instruments and Gestural Sensors for Musical Interaction and Performance," *Journal of New Music Research*, vol. 28, pp. 130-149, 1999.

[17] J. O. Borchers, "WorldBeat: Designing a Baton-Based Interface for an Interactive Music Exhibit," presented at Special Interest Group on Computer Human Interaction, Atlanta, U.S.A, 1997.

[18] D. Young and I. Fujinaga, "Aobachi: A New Interface for Japanese Drumming," presented at New Interfaces for Musical Expression, Hamamatsu, Japan, 2004.

[19] C. Havel and H. Desainte-Catherine, "Modeling and Air Percussion for Composition and Performance," presented at New Interfaces for Musical Expression, Hamamatsu, Japan, 2004.

[20] V. Goudard, C. Havel, S. Marchand, and M. Desainte-Catherine, "Data Anticipation for Gesture Recognition in the Air Percussion," presented at International Computer Music Conference, Barcelona, Spain, 2005.

[21] W. S. Yeo, "The Bluetooth Radio Ball Interface: A Wireless Interface for Music/Sound Control and Motion Sonification," presented at International Computer Music Conference, New Orleans, U.S.A, 2006.

[22] A. Bowen, "SoundStone: A 3-D Wireless Music Controller," presented at New Interfaces for Musical Expression, Vancouver, Canada, 2005.

[23] A. Y. Benbasat, J.A. Paradiso, "A Wireless Modular Sensor Achitecture and its Application in On-Shoe Gait Analysis," presented at IEEE Sensors Conference, 2003.

[24] G. Torre, B. O'Flynn, P. Angove, "Celeritas: Wearable Wireless System," presented at New Interfaces for Musical Expression, New York, U.S.A, 2007.

[25] F. Bevilacqua, E. Flety, N. Leroy, "Wireless Sensor Interface and Gesture Follower for Music Pedagogy," presented at New Interfaces for Musical Expression, New York, U.S.A, 2007.

[26] R. Boie, M. Mathews, and W. A. Schloss, "The radio drum as a synthesizer controller," presented at International Computer Music Conference, Ohio, U.S.A, 1989.

[27] L. W. R. R. Boie, and E. R. Wagner, "Gesture sensing via capacitive moments," AT&T Bell Labs.

[28] B. Neville, "Gesture analysis through a computer's audio interface: The Audio-Input Drum," in *Interdisciplinary*, vol. M.A.Sc. Victoria: University of Victoria, 2007.

[29] M. Wright and A. Freed, "Open SoundControl: A New Protocol for Communicating with Sound Synthesizers," presented at International Computer Music Conference, Thessaloniki , Greece, 1997.

[30] R. B. McGhee, E. R. Bachmann, and M. J. Z. X. Yun, "Real-TimeTracking and Display of Human Limb Segment Motions Using Sourceless Sensors and a

Quaternion-Based Filtering Algorithm – Part I: Theory.," Naval Postgraduate School, Monterey, U.S.A, 2000.

[31] J. M. Mendel, *Lessons In Estimation Theory For Signal Processing, Communications, and Control*. New Jersey, U.S.A: Prentice Hall PTR, 1995.

[32] wikipedia.org, "Kalman Filter," 2005.

[33] P. Z. P. Jr, *Probability, Random Variables, and Random Signal Principles*. New York, U.S.A: McGraw-Hill Inc, 1993.

[34] E. Brookner, *Tracking and Kalman Filtering Made Easy*. New York, U.S.A: John Wiley and Sons Inc., 1998.

[35] R. A. Singer, "Estimating Optimal Tracking Filter Performance of Manned Maneuvering Targets," *IEEE Transactions On Aerospace and Electronic Systems*, vol. AES-6C4, pp. 473-484, 1970.

[36] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York, U.S.A: John Wiley and Sons Inc, 2001.

[37] A. Wald, *Sequential Analysis*. New York: John Wiley and Sons Inc, 1947.

[38] T. v. Kasteren, "Realtime Tempo Tracking Using a Kalman Filter," in *Computer Science*, vol. MASc. Amsterdam: University of Amsterdam, 2006.

[39] S. Dixon, "A Beat Tracking System for Audio Signals," presented at Diderot Forum on Mathematics and Music, Austrian Computer Society, Austria, 1999.

[40] M. Goto, "An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds," *Journal of New Music Research*, vol. 30, pp. 159-171, 2001.

[41] M. E. P. Davies and M. D. Plumbley, "Beat Tracking with a Two State Model," presented at IEEE International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, U.S.A, 2005.

[42] M. E. P. Davies and M. D. Plumbley, "Causal Tempo Tracking of Audio," presented at International Symposium on Musical Information Retrieval, Barcelona, Spain, 2004.

[43] M. M. S. Hainsworth, "Beat Tracking with Particle Filtering Algorithms," presented at IEEE Workshop of Signal Processing to Audio Acoustics, New Paltz, New York, 2003.

[44] W. A. Sethares, R. D. Morris, and J. C. Sethares, "Beat Tracking of Musical Performances Using Low-Level Audio Features," *IEEE Transactions on Speech and Audio Processing*, vol. 13, pp. 275-285, 2005.

[45] R. Rowe, *Machine Musicianship*. Cambridge, MA: MIT Press, 2004.

[46] R. B. Dannenberg, "An On-line Algorithm for Real-Time Accompaniment," presented at International Computer Music Conference (ICMC), Paris, France, 1984.

[47] G. Lewis, "Too Many Notes: Computers, Complexity and Culture in Voyager," *Leonardo Music Journal*, vol. 10, pp. 33-39, 2000.

[48] F. Pachet, "The Continuator: Musical Interaction with Style," presented at International Computer Music Conference (ICMC), Ohio, U.S.A, 2002.

[49] J. C. T. Machover, "Hyperinstruments: Musically Intelligent and Interactive Performance and Creativity Systems," presented at International Computer Music Conference (ICMC), Ohio, U.S.A, 1989.

[50] A. T. Cemgil, "Bayesian Music Transcription," vol. PhD. Nijmegen: Radboud University of Nijmegen, 2004.

[51] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An Experimental Comparison of Audio Tempo Induction Algorithms," in *IEEE Transactions on Speech and Audio Processing*, vol. 14, 2006.

[52] B. Till, M. Benning, and N. Livingston, "Wireless Inertial Sensing Package (WISP)," presented at New Interfaces for Musical Expression, New York, U.S.A, 2007.

[53] A. Kapur, M. Benning, A. Tindale, and P. Driessen, "The KiOm: A Paradigm for Collaborative Controller Design," presented at International Computer Music Conference, New Orleans, U.S.A, 2006.

Appendix A

Real-Time Tempo Tracking Using a Kalman Filter

In this appendix we will explain in more detail the operation of the real-time tempo tracking algorithm, used in chapter 6, to perform the sitar and tabla tempo tracking experiments. The tracker was designed by Ali Taylan Cemgil during his Phd at the Radboud University of Nijmegen in the Netherlands [50]. A fully functional real-time C++ implementation was written and tested by Tim Van Kasteren [38] during his masters work at the University of Amsterdam. Tim kindly contributed his source code as well as his support to our tempo tracking experiments. This appendix is a summary of a more detailed explanation of the Kalman filter based tempo tracker published in [38].

We begin by introducing a mathematic model for tempo tracking in non-real time on a known score of music. This leads to the development of the dynamic and measurement models used by the Kalman filter. We then extend our dynamic model to deal with any real-time musical performance by introducing a particle filter.

A.1 Mathematical Model

Figure A.1 shows a simple musical score. The number k, located on the top of the score represents the note or onset number and c, located below, is the score position. The score position can be thought of as the place the note takes in the score with integer values falling on quarter notes.

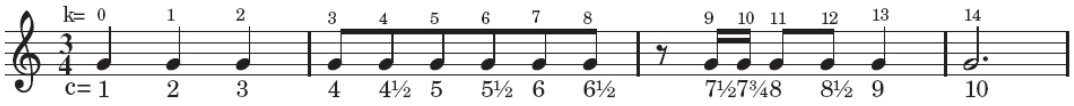

Figure A.1. *Score with note index and score positions (courtesy of Tim van Kasteren)*

Next we will introduce the score difference γ. This represents the difference of the score position between consecutive notes, $\gamma_k = c_k - c_{k-1}$. The bottom line of Figure A.2 shows values of γ for our simple score. Notice that the first note does not have a value for γ since there is no previous note that it can be compared to. Examining the score positions of Figure A.1 we can verify the correctness of the score differences of Figure A.2. The line second from the bottom of Figure A.2 shows the onset time, τ, of each note. This refers to the moment in time, in seconds, when the note is played.

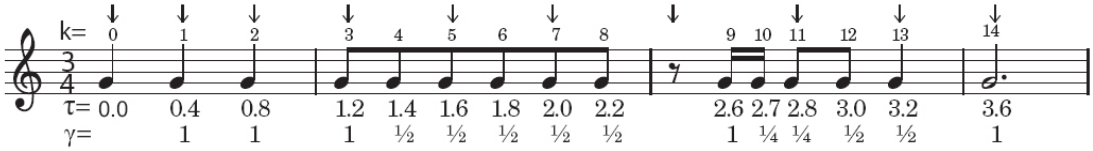

Figure A.2. *Score with note index, score difference, and onset time (courtesy of Tim van Kasteren)*

The onset time of each note depends on what tempo the score is performed at. Tempo refers to the number of beats per minute (bpm). The down pointing arrows above the score of Figure A.2 show where the beat lands. In this case, the arrows are at the locations of quarter notes or where you would typically snap your fingers or clap your hand along with the music. Another way to define tempo is by the amount of time elapsed between consecutive beats; this is called the beat period, Δ. The equation below shows the relationship between consecutive onsets on the score. This can be verified by examining Figure A.2 and assuming Δ=0.4 or 150 bpm.

$$\tau_k = \tau_{k-1} + \gamma_k \Delta_{k-1} \qquad \text{(A.1)}$$

We are now ready to define the dynamic model.

A.2 Dynamic and Measurement Models for the Tempo Tracker

Similar to the Radio Drum stick motion, the dynamic model can be defined by a linear equation of this form.

$$x_k = \Phi x_{k-1} + w_k \quad \text{(A.2)}$$

Where x is the state vector containing τ and Δ, Φ is the state transition matrix relating consecutive instances of x, and w is a dynamic Gaussian white noise vector. The equation below shows the complete dynamic model. This model enables us to generate onsets times produced by a known score at a known tempo. Where the score difference γ, in the dynamic model, changes based on the score.

$$\begin{pmatrix} \tau_k \\ \Delta_k \end{pmatrix} = \begin{pmatrix} 1 & \gamma_k \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tau_{k-1} \\ \Delta_{k-1} \end{pmatrix} + w_k \qquad \text{(A.3)}$$

W, the dynamic noise vector, represents onset and tempo variations away from the real score which is introduced by a human performer. The dynamic noise accounts for temporal expression of a musician; such as rhythmic effects caused by the delaying of notes. In the Kalman filter, w is defined through its covariance matrix Q.

The measurement model, shown below, relates the hidden state x to the measured onsets y_k. The measurement of the onsets may be done in a variety of ways,

the easiest being from a performance played on a midi keyboard. The onset time of each midi notes can be used as input to the tracker.

$$y_k = (1 \quad 0)\begin{pmatrix} \tau_k \\ \Delta_k \end{pmatrix} + v_k \qquad \text{(A.4)}$$

V, the measurement noise is described through its covariance matrix, R, and represents the confidence we have in our measurements or method of obtaining onsets.

A.3 Particle Filtering with a Switching Variable

Since the score difference, γ, changes depending on where we are in the score, we define it as a discrete switching variable. We say discrete because γ can only take up values defined on a musical grid for example 2, 1.75, 1.5, 0.5 0.25. A series of dynamic models differing only by their value of γ have to be defined to deal with all possible score differences in any given performance. And since in a real-time musical performance the score differences are not known in advance, a way is needed for the switching Kalman filter to decide which score difference to use at each time step.

From the large number of possible values of γ, a subset can be defined using the ideal score difference calculated below.

$$\begin{aligned} & e_k = y_k - Cx_{k|k-1} = 0 \\ & y_k - C\Phi x_{k|k-1} = 0 \\ & y_k - (1 \quad 0)\begin{pmatrix} 1 & \gamma_k \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \tau_{k-1} \\ \Delta_{k-1} \end{pmatrix} = 0 \qquad \text{(A.5)} \\ & y_k - \tau_{k-1} - \gamma_k \Delta_{k-1} = 0 \\ & \frac{y_k - \tau_{k-1}}{\Delta_{k-1}} = \gamma_k \end{aligned}$$

The innovation e_k is the difference between the next measured onset and the predicted onset based on the previous filtered onset time. Ideally, when our model and our measurements perfectly coincide, e_k equals zero. A subset of quantized score differences chosen in the range around the ideal score difference are tested as candidates for the next filtering step. Each value of γ in the subset is referred to as a particle. All possible outcomes of the state, x are calculated using each particle. The likelihood of each outcome is then calculated using e_k to sample from a normal distribution, $N(0,S(k))$ where $S(k)$ is the covariance of the innovations.

$$S(k) = HP(k \mid k-1)H' + R(k) \quad \text{(A.6)}$$

Often several particles may have the same likelihood, meaning that various combinations of period and score difference may lead to the same onset. For example a period of 1 sec or 60bpm with a score difference of 1 will lead to the onset as a period of 0.5 sec, or 120bpm with a score difference of 2. A prior term is multiplied by the likelihood to help dissolve such ambiguities. Priors are defined for each value of the score difference and can be modified depending on the genre of music being tracked. Certain traditions of music may for example favour quarter notes at lower tempos over eighth notes at higher tempos. Based on the probability calculated from the likelihood and prior the n best particles are chosen to reproduce children at the next iteration and the beat period generated from the particle with the highest probability is outputted.