MULTI-TAP EXTENDED KALMAN FILTER FOR A PERIODIC WAVEFORM WITH UNCERTAIN FREQUENCY AND WAVEFORM SHAPE, AND DATA DROPOUTS

by

Justin Saboury

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Engineering

at

The University of Wisconsin-Milwaukee

August 2019

ABSTRACT

MULTI-TAP EXTENDED KALMAN FILTER FOR A PERIODIC WAVEFORM
WITH UNCERTAIN FREQUENCY AND WAVEFORM SHAPE, AND DATA DROPOUTS

by

Justin Saboury

The University of Wisconsin-Milwaukee, 2019
Under the Supervision of Professor Brian Armstrong

Gait analysis presents the challenge of detecting a periodic waveform in the presence of time varying frequency, amplitude, DC offset, and waveform shape, with acquisition gaps from partial occlusions. The combination of all of these components presents a formidable challenge. The Extended Kalman Filter for this system model has six states, which makes it weakly identifiable within the standard Extended Kalman Filter network. In this work, a novel robust Extended Kalman Filter-based approach is presented and evaluated for clinical use in gait analysis. The novel aspect of the proposed method is that at each sample, the present and several past observations are used to update the system state, strengthening the state identification. These past observations are referred to as delay-line taps.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to thank my advisor, Professor Brian Armstrong, for his encouragement, patience and dedication to educating students. His passion for teaching is inspiring, and his interest in ensuring the material is understood has been invaluable to me. Thanks to my committee members, Professor Istvan Lauko and Professor Jun Zhang. I would also like to thank my parents for supporting me at every step in my education and in life. Thank you, Michelle, for encouraging me. And of course, thank you to all my friends and family for the support through the years.

1 Introduction

1.1 Motion tracking for gait analysis

In a variety of applications of image technology, such as medical image analysis and human motion modeling, tracking and analyzing the motion of nonrigid objects is of growing interest to researchers in a wide spectrum of disciplines. Motion of rigid parts occurs in situations where individual rigid parts of an object move independently of one another. In this case, the motion of each constituent part is rigid, but the motion of the whole object is non-rigid. Usually, this concept is categorized as "articulated motion." Kambhamettu, Goldgof, Terzopoulos, and Huang [1] proposed an extended classification scheme based on the degree of non-rigidity of the objects. Figure 1 shows the classification tree for objects in motion, according to Kambhamettu's definition. The definitions of the motion classes are briefly described as follows:

- Rigid motion preserves all distances and angles and has no associated non-rigidity.
- Articulated motion is piece-wise rigid motion. The rigid parts conform to the rigid motion constraints, but the overall motion is not rigid.

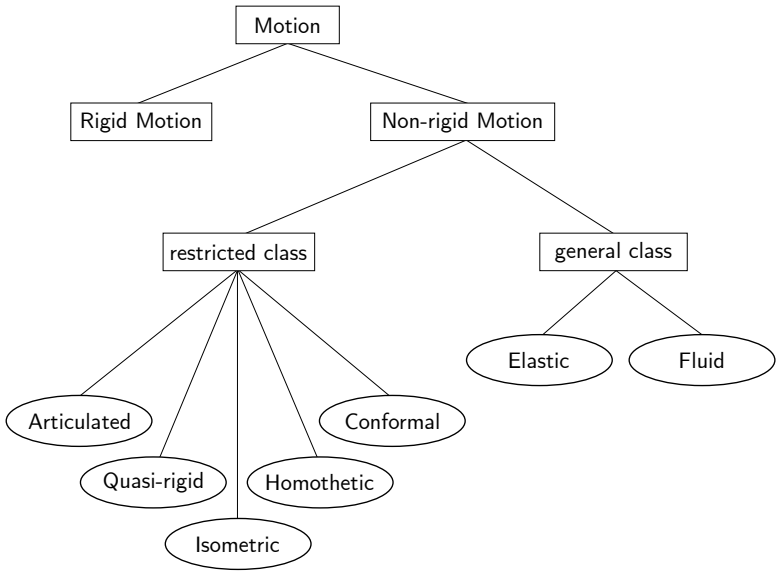

Figure 1: Classification tree for motion of objects defined by Kambhamettu et al.

In motion analysis, the motion vectors of different body parts reveal the underlying structure of the object. Thus, high-level processing, such as segmentation, gait recognition, or motion prediction can use the results of motion analysis. In Niyogi and Adelson's work [2], the model of the gait is the basis for the motion analysis and gait recognition process. With an accurate method of determining heel strike (HS) and toe off (TO) times, in addition to collecting motion tracking data for velocity and position, walking trials

may be broken into gait "cycles", resembling a periodic oscillator [3]. An example of this oscillation is shown in Figure 2.

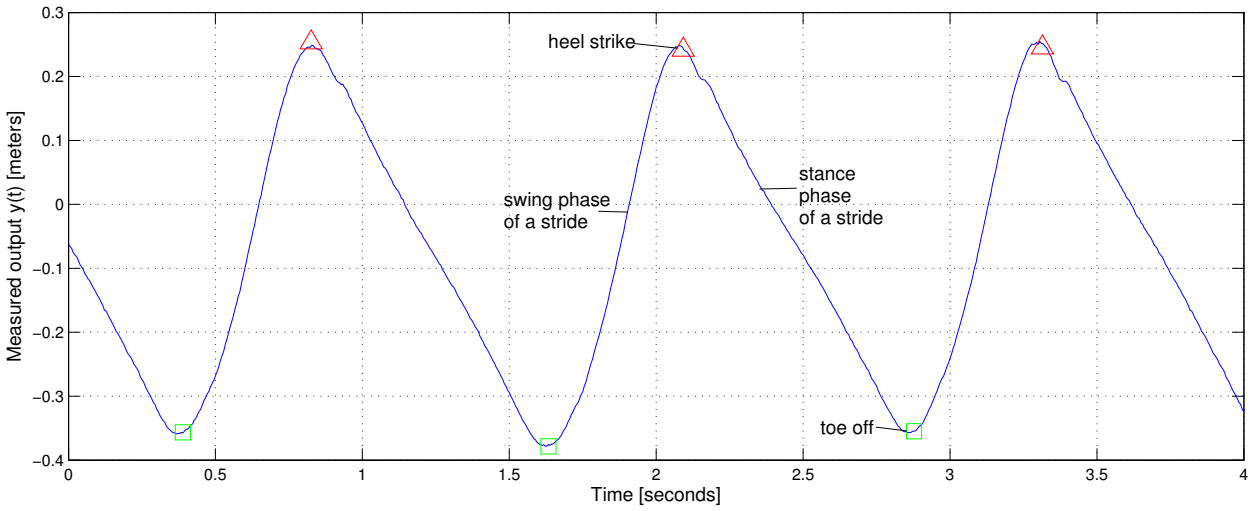

Figure 2: Experimental data for gait motion captured at 2 mph, showing typical measured ankle motion on left ankle for subject in normative data collection.

1.2 An introduction to Moiré Phase Tracking

Moiré Phase Tracking (MPT) is a 3D motion tracking technology developed by Brian Armstrong [4] that utilizes uses a single camera as a sensor and tracks specially designed targets (refer to Figure 3). Additional cameras may be added to the setup to expand field of view. The target relies on the detection of the starburst landmark in the center of the target, the location of the four circular landmarks, and the resolution of the periodic moiré patterns. The analysis of all of this information yields six degree of freedom (6-DOF) pose estimation of X, Y, Z and pitch, roll and yaw [5]. A complete system with camera is shown in Figure 4.

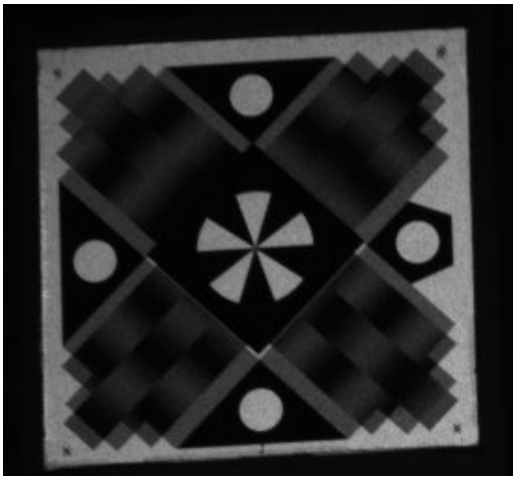

Figure 3: Moiré Phase Tracking Target

Figure 4: Single camera, single tracker system using Moiré Phase Tracking. The apparatus consists of a camera and reflective marker capable of rapidly and accurately collecting six degree of freedom data for each marker.

1.3 How MPT is used in gait analysis

In gait analysis, while ground reaction force data collection using force plates provides an accurate means of identifying HS and TO events, the size of the plates is not always suitable for individuals with both short and long stride lengths, and the plates require clean hits on the force platform [6]. Additionally, the plates can be expensive. In some applications, motion tracking data can be collected and combined with data from a force plate to create a simulation of the human anatomy extremities that can be performed and analyzed. From the analysis, HS and TO event times can be determined [7, 8, 9]. The need to determine HS and TO times and provide this feedback in real-time is growing in the field of physical therapy. This need drives the search for a low-cost method of determining gait HS and TO that accommodates variation in stride length and speed.

Since MPT can be used to track the motion of nonrigid objects with a high level of accuracy, for this study, MPT is used to track a human subject in motion without the use of force plates. Multiple targets are affixed to an individual, who will usually walk or run on a treadmill (frequently used in locomotion research because it offers a controlled and convenient environment for testing and facilitates the use of static cameras). An example of this type of setup is illustrated in Figure 5. Analysis of collected data can yield information about gait movement real time, and even send bio-feedback in real-time prediction to the subject using the treadmill.

Figure 5: Measurement apparatus for collecting gait movement data. The apparatus utilizes two cameras to capture a larger field of view and operates with eight markers, a single marker placed per body segment.

1.4 Goals and objectives

This thesis aims to demonstrate the performance of a filter that can reliably detect a periodic signal in the presence of missing observations and changing frequency, waveform shape, amplitude, and DC offset, with the purpose of detecting Heal Strike and Toe Off times. The filter needs to provide this feedback real-time, with a low computational cost and high degree of reliability and accuracy. Because of the real-time requirement and the presence of noise and gaps in the data, a model-based filtering approach is used. Several technical challenges associated with implementation of such a filter are outlined in the following sections.

1.5 Challenges

1.5.1 Phase and amplitude variation

During a single standard gait cycle, the phase (or position along the cycle) moves until the start of the next gait cycle. One challenge of predicting the next position or phase is accounting for changes in amplitude. Amplitude may vary depending on the subject's stride length and treadmill speed and the system model must accommodate abrupt changes.

1.5.2 Rate and DC offset variation

The system model must accommodate shifts in frequency, or stride rate, measured as the time it takes to complete one full stride. Since it is rare for a human subject to maintain constant gait velocity, even under constant treadmill speeds, the velocity of the gait cycle will shift during the cycle. Additionally, walking or running at higher speeds usually accompanies an increase in stride rate. Refer to Figure 6. Many subjects exhibit characteristics of an asymmetric waveform at lower speed, changing DC offset, and gaps in data collection at higher speeds as the waveform shifts to a symmetrical shape. Since movement speed is proportional to the length of the stride and gait cycle frequency, at higher movement speeds, when the stride length remains consistent, the stride frequency increases.

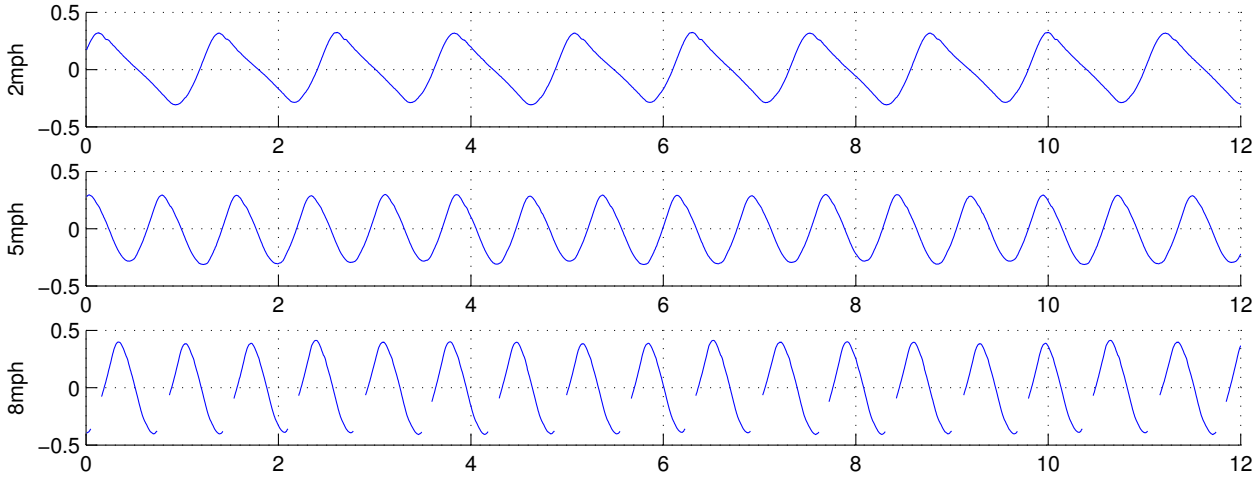

Figure 6: Ankle movement in a pelvis-center coordinate frame, motion shown in meters, taken from experimental data tracking the left leg. Data was collected using camera setup as shown in Figure 5. Note the asymmetry exhibited at lower speed and gaps in data collected at higher speed.

The DC offset of the oscillator system can vary also. Thinking about the cycle of the foot as centered under the pelvis, the mean position of the center of the foot becomes sustainably negative at higher speeds. Therefore, the offset must be taken into account in the filtering model.

Figure 7 illustrates variation over a broader range of samples with an example data set containing experimental data from 2,859 trials over varying treadmill speed.

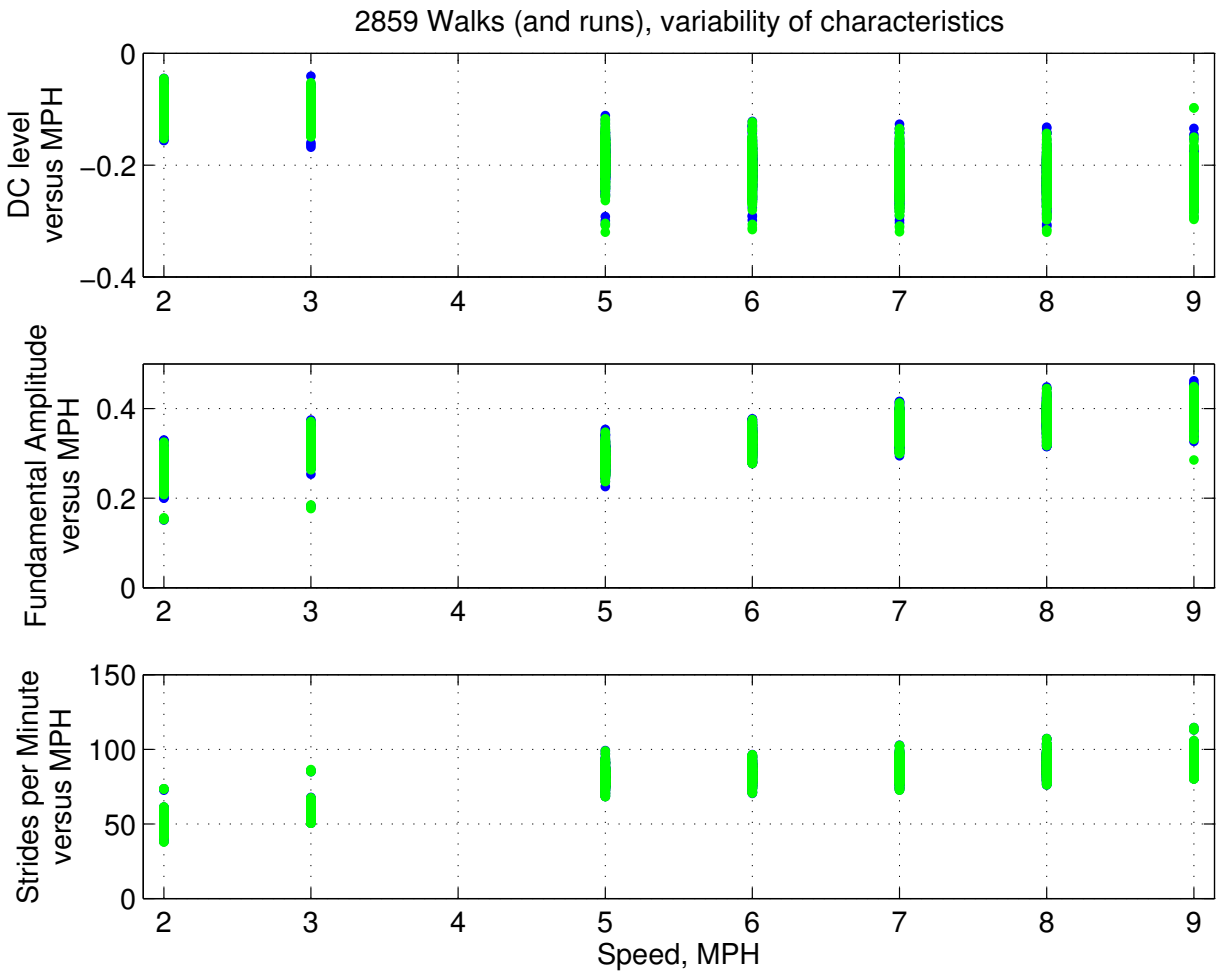

Figure 7: Variability of gait movement cycle characteristics. Left leg plotted blue behind right leg in green. Note the change in stride rate within a population set for a given treadmill speed, and the mean increase in stride rate at higher MPH. DC level variation trends negative with higher treadmill speed.

1.5.3 Waveform shape variation

Variation in the waveform shape of the system is mainly caused by shifts in the symmetry of the periodic waveform as stride rate changes, as illustrated by Figure 6. Figure 8 shows the distribution of the second harmonic magnitude and phase for all treadmill speeds.

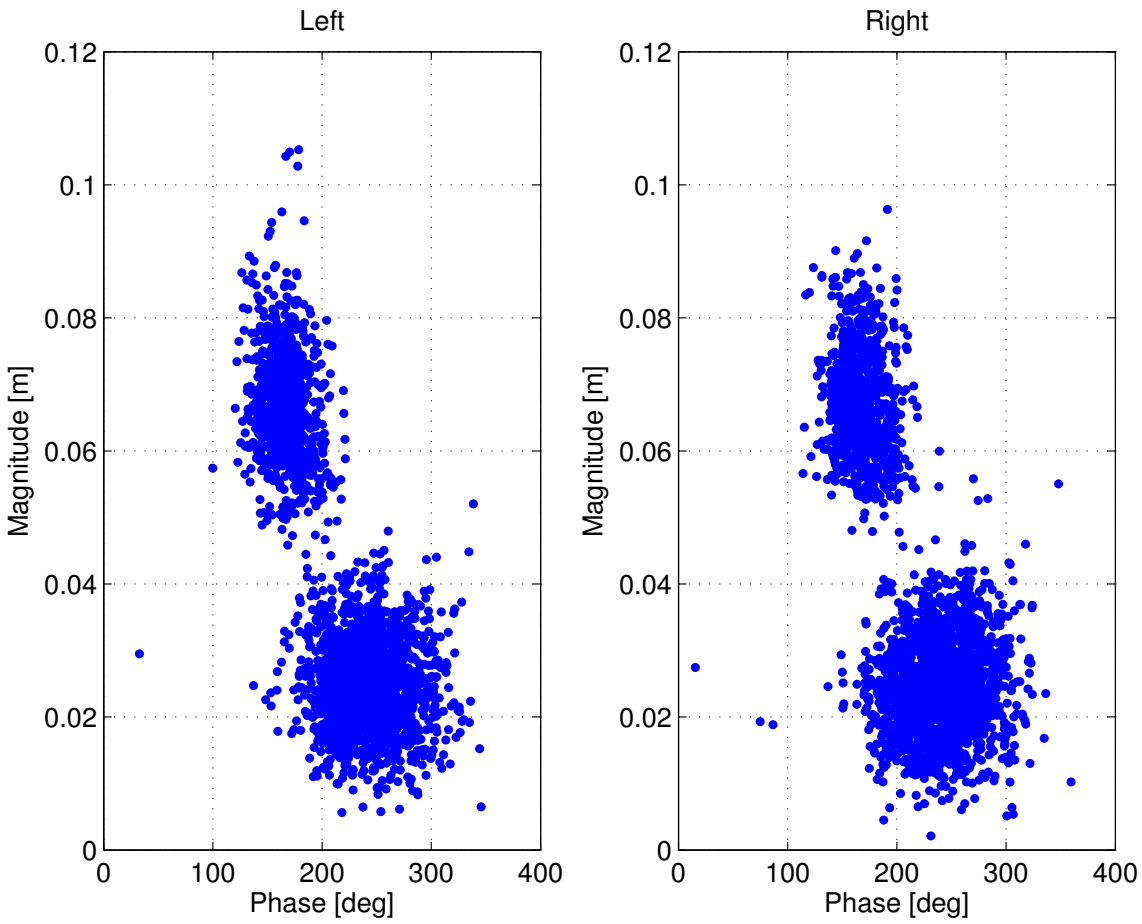

Figure 8: Variability of second harmonic characteristics. All trials from Figure 7 are plotted in the distribution plots, one plot for each leg. Each leg plot shows two distinct clusters, with the larger magnitude cluster of points attributed to walking trials and the lower magnitude cluster attributed to running trials. Note that each trial plotted varies by speed of the trial and the individual.

1.5.4 Intermittent data capture

One significant challenge in signal processing is accommodating an abrupt absence of sampling data due to an obstruction in data. Many methods deal with this problem by adding supplemental observers to the system or interpolation between lost data points. For the model-based filter, if prior state information can be stored, it can be used with the system model to provide accurate predictions in cases with insufficient data measurements.

2 Literature Review

Conventional Kalman filtering requires the definition of a dynamic model, a model that accounts for time-dependent changes in the system. Adaptive Kalman filtering algorithms, by definition, adapt the filter in real time to correspond to the temporal variation of the errors involved. The following subsections explore common Kalman filtering and similar processing techniques in the motion tracking application space.

2.1 The Kalman Filter modeling approach

2.1.1 Overview and setup of the linear, time variant Kalman Filter

The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. The filter is essentially a set of mathematical equations that falls into two groups: time update equations and measurement update equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the *a priori* estimates for the next time step. The measurement update equations are responsible for the feedback—i.e., for incorporating a new measurement into the *a priori* estimate to obtain an improved *a posteriori* estimate.

The time update equations can also be thought of as "predictor" equations, while the measurement update equations can be thought of as "corrector" equations. Therefore, the final estimation algorithm resembles that of a predictor-corrector algorithm for solving numerical problems optimally, in the sense that it minimizes the estimated error covariance when some presumed conditions are met [10, 11].

2.1.2 Prediction (*a priori)* equations

At each discrete time increment, a linear operator is applied to the state to generate a prediction of the new state, with some noise mixed in. The true state of the system at time k evolves from the state at (k -1) according to the following

$$x_k^* = F_k \, x_{k-1}^* + B \, u_{k-1} + w_{k-1} \tag{1}$$

where $x_k^* \in \mathbb{R}^n$ is the state vector of the model containing n state variables. The $n \times n$ matrix F_k in Equation (1) relates the state at the previous time step to the state at the current step. The $n \times 1$ matrix B relates the optional control input $u_k \in \mathbb{R}^1$ to the state x. Random variable $w_{k-1} \in \mathbb{R}^n$ represents the process noise.

The estimate state prediction equation is

$$\hat{x}_{k|k-1} = \hat{x}_k^- = F_k\,\hat{x}_{k-1} + B\,u_{k-1} \tag{2}$$

where the notation $\hat{x}_{n|m}$ generally represents the estimate of x at time n given observations up to time $m \leq n$. For simplicity, this paper will refer to estimates at step k where the known estimate is from one step behind, as $\hat{x}_k^-$. Additionally, the star modifier is used to denote a true value, hat modifier is used to denote an estimate, tilde modifier to denote residual error between estimate and true, and bar modifier denotes an observation. Note for this specific application discussed in this thesis, we do not have a control input, so the $B\,u_{k-1}$ term drops from the state equations.

The measurement of the output, $\bar{z}_k \in \mathbb{R}^{\mathrm{m}}$, where m is the number of observations in the system at each time step, is

$$\bar{z}_k = H_k\,x_k^* + v_k \tag{3}$$

Random variable $v_k \in \mathbb{R}^{\mathrm{m}}$ represents the measurement noise, and the measurement noise and process noise w_{k-1} are assumed to be independent of each other with normal probability distributions

$$p(w) \sim N(0, Q) \tag{4}$$

$$p(v) \sim N(0, R) \tag{5}$$

where Q is the process noise covariance and R the measurement noise covariance.

The $m \times n$ matrix H_k in measurement Equation (3) is referred to as the *observation model* and relates the true state x_k^* to the measurement $\bar{z}_k$ [10, 11].

Using Equation (2) for the predicted state estimate, we also find the predicted covariance

$$P_k^- = F_k\,P_{k-1}\,F_k^T + Q_k \tag{6}$$

where P_{k-1} is the *a posteriori* covariance from the previous step, or an initial seed value for the first step.

2.1.3 Correction (*a posteriori*) equations

The correction update begins by computing the Kalman gain, K_k, chosen as the gain factor that minimizes the *a posteriori* error covariance equation.

$$K_k = P_k^-\,H_k^T\,(H_k\,P_k^-\,H_k^T + R_k)^{-1} \tag{7}$$

The next step is to measure the process to obtain $\bar{z}_k$, and generate an *a posteriori* state estimate as a

linear combination of an *a priori* estimate $\hat{x}_{k-1}$ and a weighted difference between an actual measurement $\bar{z}_k$ and a measurement prediction $H\,\hat{x}_k$ as shown:

$$\hat{x}_k = \hat{x}_k^- + K_k\,(\bar{z}_k - H_k\,\hat{x}_k^-) \tag{8}$$

The final step is then to obtain the *a posteriori* error covariance estimate:

$$P_k = (I - K_k\,H_k)\,P_k^- \tag{9}$$

After each time and measurement update pair, the process is repeated with the previous *a posteriori* estimates used to predict the new *a priori* estimates [10, 11].

2.2 The Extended Kalman Filter

The Extended Kalman Filter (EKF) is similar to the Kalman Filter, but can also be used in non-linear systems, because it linearizes the transformations around the current mean and covariance via Taylor Expansions. In the EKF, the state transition and observation models need not be linear functions of the state but may instead be functions [12].

For the EKF, a *nonlinear* function f in the difference equation relates the state at the previous time step k -1 to the state at the time step k.

$$x_k^* = f(x_{k-1}^*, u_{k-1}, w_{k-1}) \tag{10}$$

Just like the state transition model, the *observation model* is now a *nonlinear* function h in the measurement equation that relates the state x_k^* to the measurement $\bar{z}_k$:

$$\bar{z}_k = h(x_k^*) + v_k \tag{11}$$

The steps of the EKF are as follows:

Algorithm 1 Extended Kalman Filter

Prediction Step:

1. Predict the state

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k) \tag{12}$$

2. Predict the error covariance matrix

$$P_k^- = F_k \, P_{k-1} \, F_k^T + Q_k \tag{13}$$

⇕

Correction Step:

1. Calculate the Kalman Gain

$$S_k = H_k \, P_k^- \, H_k^T + R_k \tag{14}$$

$$K_k = P_k^- \, H_k^T \, S_k^{-1} \tag{15}$$

2. Correct the *a priori* state estimate

$$\tilde{z}_k = \hat{z}_k - \bar{z}_k \tag{16}$$

$$\hat{x}_k = \hat{x}_k^- - K_k \tilde{z}_k \tag{17}$$

3. Correct the *a posteriori* error covariance matrix estimate

$$P_k = (I - K_k H_k) P_k^- \tag{18}$$

With:

$\hat{x}_k$ the updated *a posteriori* state estimate;

P_k the updated *a posteriori* error covariance matrix (a measure of the estimated accuracy of the state estimate);

F_k the state-transition model, applied to the previous state x_{k-1};

H_k the observation model;

$\hat{x}_k^-$ the predicted *a priori state* estimate;

P_k^- the predicted *a priori error* covariance;

Q_k, the covariance of the process noise;

R_k, the covariance of the observation noise;

$\tilde{z}_k$, the innovation, or measurement pre-fit residual (difference between the observed measurement and the predicted measurement);

$\bar{z}_k = h\left(x_k^*\right) + v_k$, the observation measurement;

$\hat{z}_k = h\left(\hat{x}_k^-\right)$, the predicted measurement;

S_k, the innovation covariance;

K_k, the Kalman gain

The EKF is a two step process: predict and correct. Refer to Algorithm (1).

1. **Prediction**
   (a) Predict the next state from the current state using the system model
   (b) Predict the error covariance of the next state prediction
2. **Correction** (using measurement data)
   (a) Compute a gain factor (Kalman Gain) that minimizes the error covariance.
   (b) Correct state estimation by adding the product of the Kalman gain and the prediction error to the prediction.
   (c) Correct the error covariance estimation using the Kalman gain.

After Correction, the filter iterates to the next observation, and the Prediction-Correction cycle continues [12].

2.3 Modified filtering methodologies

The nonlinear filtering problem has long been in the mainstream of research, with many results reported in this field [13, 14]. Estimation using measurements that are intermittent or degraded are well known to be one of the most frequently occurred phenomena in networked systems [15]. In these cases, perfect communication is not always available, and often the system measurement degrades in a probabilistic way. In cases where data dropout or intermittent observations affect filter performance, a modified version of the Kalman filter may be employed. In one example, "anchor" sensors may transmit measurement signal if they can detect the target, or transmit empty packets containing special flags. By defining the packet dropouts, the observation and prediction steps can be determined with a modifier that takes the dropouts into account [16].

2.3.1 Modeling intermittent observations as a Bernoulli process

Motivated by using loss of data in navigation and tracking applications, Sinopoli et al. [17] considered defining the problem of performing Kalman filtering with intermittent observations. The group chose to represent systems with intermittent observations as systems which contained an independent and identically distributed (i.i.d) Bernoulli process γ_t, with the arrival of the observation at each discrete time t defined as a binary random: either the filter received the observation, or the observation was dropped from transmission.

$$\gamma_t = \begin{cases} 1, & \text{the filter successfully got the observation} \\ 0, & \text{the observation is dropped during transmission} \end{cases} \tag{19}$$

Other groups have built on this analysis. Liu, Li, Li, Fernando and Iu [18] use the Bernoulli observation model to analyze the prediction error covariance matrix, P_k^-, and estimation error covariance matrix P_k in nonlinear systems. With this arrival information in each timestamped observation, one is able to determine the minimum "critical" probability λ that $\gamma_t = 1$ for nonlinear systems to guarantee P_k^- falls within a mean boundary. Additionally, the estimation error P_k was found to stabilize under certain conditions. By making this definition, a minimum valid observation rate is defined for the system, and the expected state error covariance can be upper and lower bounded. Building on using the Bernoulli process to describe the arrival of observations, Kar, Sinopoli and Moura [19] used a discrete version of the Riccati equation to describe the evolution of the state error covariance in a linear system. Hu, Wang, Liu and Gao followed this work [20, 21], investigating an array of discrete time-varying coupled stochastic complex networks with missing measurements. They developed a new state estimation algorithm with covariance constraints, and the estimator parameter was characterized by the solutions to two Riccati based difference equations.

2.3.2 Modeling intermittent observations as a Markovian model

Another method of quantifying the effect of packet loss adapts the use of the Markovian model where the packet process is described as a binary Markov chain [22], inspired by the so-called Gilbert-Ellitt (GE) channel. In the Markovian model, the packet loss rate is temporally correlated. In this example, the necessary and sufficient stability condition for the second-order systems and certain classes of higher-order systems are explicitly given in You et al. [23]. Rohr, Marelli, and Fu [24] derived necessary and sufficient stability conditions for a class of degenerate linear systems.

2.4 Multiple Model Adaptive Estimation (MMAE)

Some algorithms such as the Multiple Model Adaptive Estimation (MMAE), run several Kalman filters in parallel, sometimes with each filter using a different model [25, 26, 27]. The MMAE algorithm is often used to select either a single 'best' Kalman filter solution, or combine the output from all the Kalman filters in a single solution. Each filter is tuned to describe a particular fault status of the system. The output of each filter is then weighted by its corresponding probability, based on the measurement history. Refer to Figure (9).

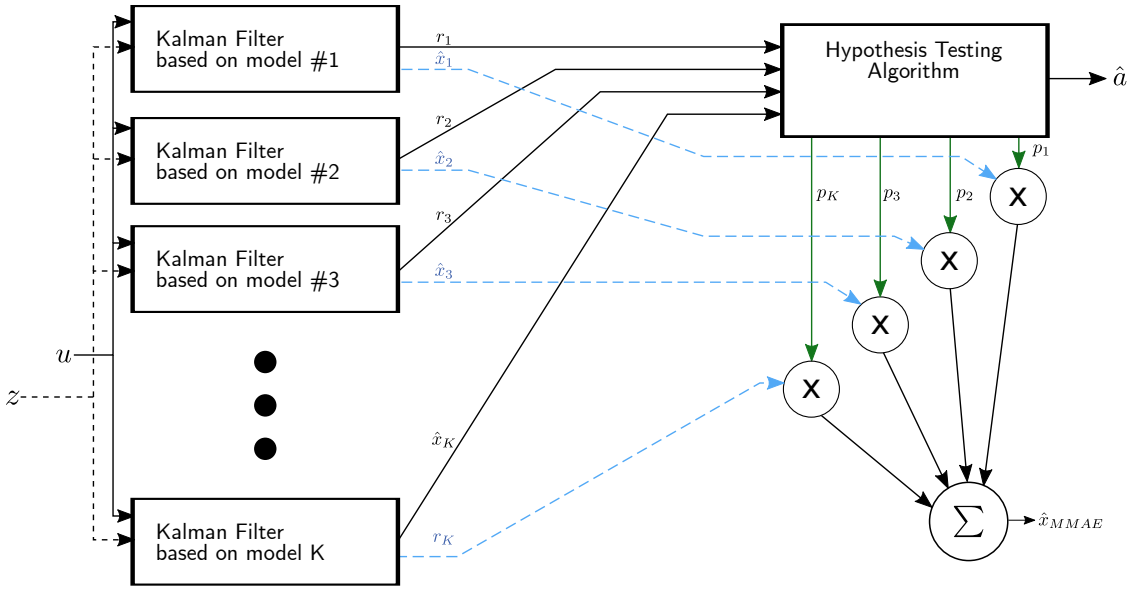

Figure 9: MMAE algorithm as defined by Hanlon and Maybeck [29]. The hypothesis testing algorithm uses the residuals r_k produced by the individual Kalman filters to compute conditional probabilities p_k, which help determine an estimate of the true parameter vector $\hat{a}_k$. The conditional probabilities also help inidicate the correctness of the individual filter models and can be used to form a probability weighted average state estimate $\hat{x}_{MMAE}$.

The MMAE method is a good choice for the detection of actuator as well as sensor faults, as long as the expected faults can be hypothesized by a reasonable number of Kalman filters. However, the number of addressable faults is rather restricted, and this method reaches its limits as soon as the actual occurring fault does not closely match the predefined fault hypothesis. In addition, there is a large computational burden imposed by running multiple Kalman filters [28]. MMAE methods have not yet proven effective with time varying frequency and waveform shape.

2.5 Adaptive Linear Neural Networks (ADALINE)

In recent years, many researchers have focused on the application of an adaptive linear neural network (ADALINE) for parameter estimation because of its low computational complexity, minimum tracking error and fast convergence rate [30, 31, 32, 33]. At a high level, the weight vector of the ADALINE is updated by the standard least mean square algorithm. Refer to Figure 10. $x(k)$ is the input on the network, $y(k)$ is the network output, and the error signal is $e(k) = \hat{y}(k) - y(k)$, where $\hat{y}(k)$ is the desired output. To summarize, the output vector $y(k)$ is compared with the desired output vector $\hat{y}(k)$, and on the basis of the corresponding error signal, the weights are adjusted until the measured output matches the desired output.

The main characteristic of the artificial neural network is the training algorithm, which is usually the processes of modifying the weight using the Windrow-Hoff delta rule:

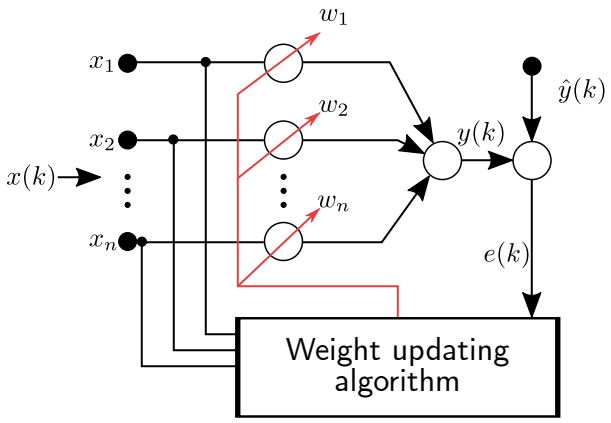

Figure 10: Flow chart detailing the ADALINE process. $x(k)$ is the input on the network, $y(k)$ is the network output, and the error signal is $e(k) = \hat{y}(k) - y(k)$, where $\hat{y}(k)$ is the desired output.

$$W(k+1) = W(k) + \frac{\eta\, e(k)\, x(k)}{\lambda + x^T\,(k)\, x(k)} \tag{20}$$

where λ is a small quantity to make $\lambda + x^T(k)\, x(k) \neq 0$. The error $e(k)$ will be brought to zero when perfect learning occurs and the weight vector will yield the Fourier coefficients of the signal. η is the learning rate and is chosen by a Genetic Algorithm, and is usually between $0 < \eta < 2$ to ensure stability and make the tracking error $e(k)$ converge to zero [34].

Convergence time and estimation accuracy show promise for harmonic systems with changes in frequency, varying offset level, amplitude and phase of the harmonics in the presence of random noise, with accurate frequency tracking demonstrated [30]. However, this algorithm does not show acceptable convergence behavior in many different situations like steady state with high noise level, intermittent or absent observation data, and real-time implementation.

2.6 Conclusions

While randomly distributed nonlinear systems have received particular research attention in the past decade, particularly in the fields of statistical study, the recursive filtering problem with randomly distributed nonlinearity and correlated noise has not yet received adequate attention, likely due to mathematical complexity and added computational cost. The recursive filter design problem continues to be investigated for nonlinear systems with multiple intermittent measurements and cases when random parameter matrices and correlated noise are both present in the model. While the challenge of unknown varying frequency, DC offset and waveform shape with data dropouts may not be uncommon, the methods discussed in this section have not been proven to adequately resolve the challenge.

3 The Multi-Tap EKF

The basic definition of a Kalman filter implies it is a recursive estimator. This means that only the estimated state from the previous time step and the current observation are needed to compute the estimate for the current state. What if this idea were extended, so that previous observations are used? This concept drives the concept and creation of the Multi-Tap Extended Kalman Filter. To summarize at a high level, the Multi-Tap EKF differs from the Standard EKF in that it incorporates past observations. Figure 11 illustrates the high-level difference between the Standard EKF and Multi-Tap EKF. Note that from an overall system perspective, the system contains no control input signal u_k.

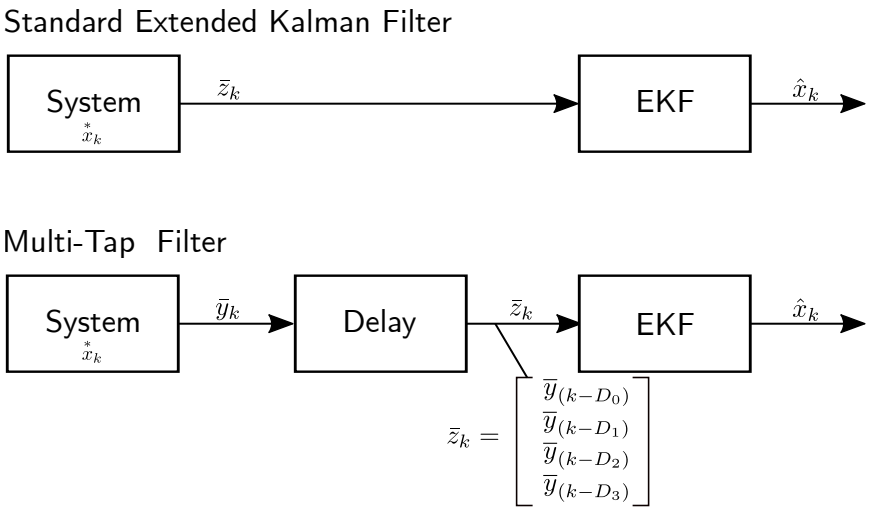

Figure 11: Block diagram highlighting the difference in system definition between a Standard and Multi-Tap EKF. Note that multiple observations enter into the EKF much like a tapped delay line.

3.1 Selecting the taps

If the idea of using multiple observations for the Extended Kalman Filter is used, there should be a strategy in selecting the previous observation points in time, or "taps". Sufficient number of taps should be selected to adequately capture at least one cycle of the harmonic oscillation; refer to Figure 12. Four taps were chosen in order to constrain the frequency and DC offset amplitude, even with poorly estimated initial values. The filter is still computationally efficient without requiring the use of more taps. For this application, it makes sense to set the most recent tap, D_0, to zero, so that $\bar{y}_{(k-D_0)} = \bar{y}_k$. Our initial observations consist of:

$$\bar{z}_k = \begin{bmatrix} \bar{y}_{(k-D_0)} \\ \bar{y}_{(k-D_1)} \\ \bar{y}_{(k-D_2)} \\ \bar{y}_{(k-D_3)} \end{bmatrix} \tag{21}$$

where $\bar{y}_{(k-D_t)}$ is an observation that occurred at sample $k - D_t$. For this application, tap locations were chosen non-uniformly at sample spacing 0, 10, 40, 80 in order to maximize the probability that the filter will track system performance and minimize the probability that it will fit to a harmonic of the data.

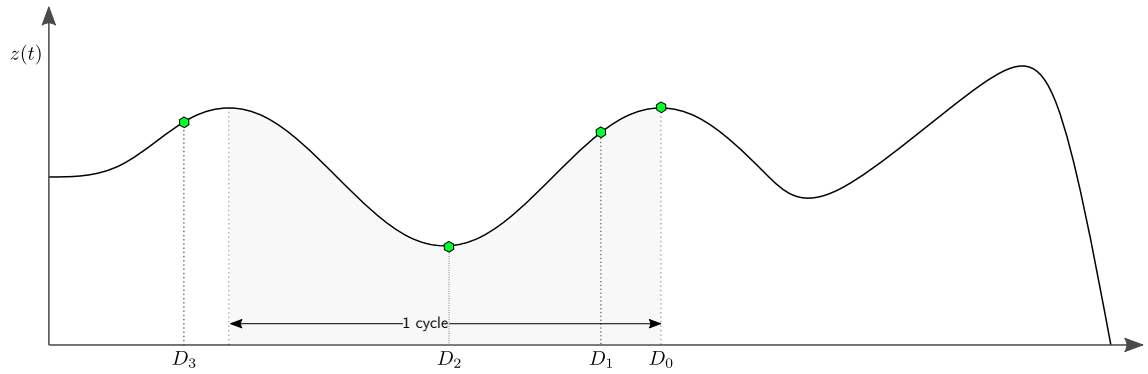

Figure 12: Multi-Tap distribution of the Extended Kalman Filter, shown as an illustration. Note that the set of points is distributed across 1 cycle of the waveform.

Figure 13 provides an overview of the Prediction and Correction steps for the Standard EKF and Multi-Tap EKF. Previous tapped observations are incorporated in the Correction cycle of the Multi-Tap EKF.

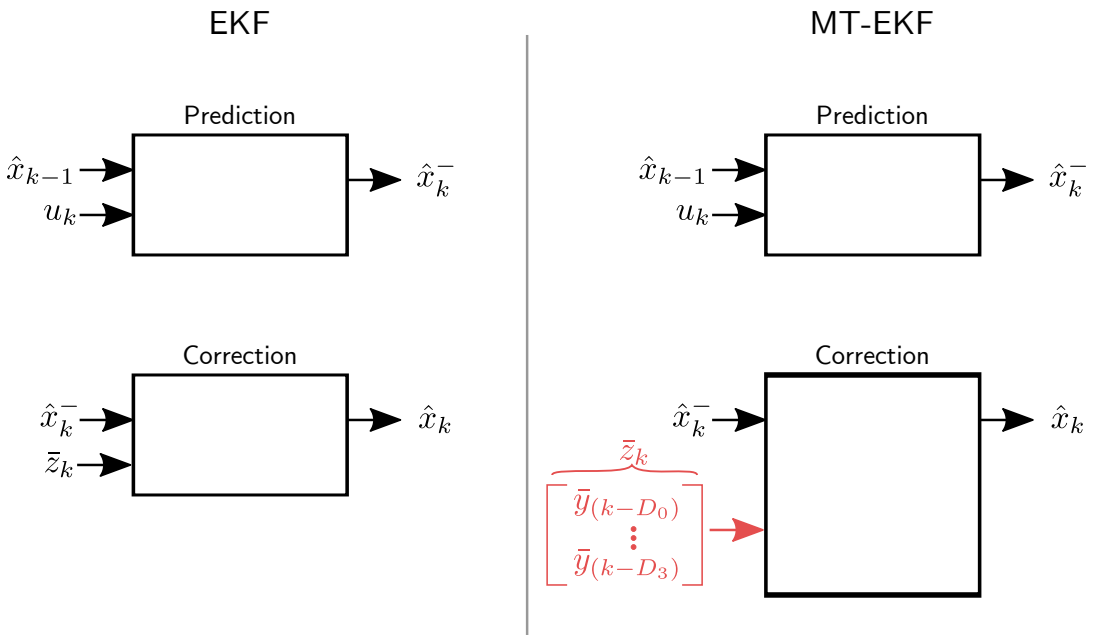

Figure 13: Traditional Kalman Filter Prediction and Correction cycles shown to the left. The Multi-Tap EKF (right) uses several observations as input to the Correction cycle.

3.2 Setting up the state space model

3.2.1 Initial model description and discussion

This section is included to understand the model and its components. When tracking gait movement, we can model our system as a harmonic oscillator with a DC offset. In polar coordinates, we can define our system output as a function of time t:

$$y(t) \in \mathbb{R}^1, \quad y(t) = A_0 + A_1 \cos(\omega t + \phi_1) + A_2 \cos(2\omega t + \phi_2) \tag{22}$$

where $A_0 \in \mathbb{R}^1$ represents the DC offset [V],

$A_1 \in \mathbb{R}^1$ is the magnitude of the fundamental [m],

$A_2 \in \mathbb{R}^1$ is the magnitude of the second harmonic [m],

$\omega \in \mathbb{R}^1$ is the rotation velocity [rad/s], assuming for the moment velocity is constant,

$\phi_1 \in \mathbb{R}^1$ is the rotation angle offset [rad], and

$\phi_2 \in \mathbb{R}^1$ is the second harmonic offset [rad].

3.2.2 Casting the model in rectangular coordinates with constant ω

It is advantageous to define the output function, Equation (22), in terms of rectangular coordinates:

$$
\begin{aligned}
x_1(t) &= A_1 \cos(\omega\, t \,+\, \phi_1) \\
y_1(t) &= A_1 \sin(\omega\, t \,+\, \phi_1) \\
x_2(t) &= A_2 \cos(2\,\omega\, t \,+\, \phi_2) \\
y_2(t) &= A_2 \sin(2\,\omega\, t \,+\, \phi_2)
\end{aligned} \tag{23}
$$

Then,

$$
y(t) = A_0(t) + x_1(t) + x_2(t) \tag{24}
$$

Having our state variables in terms of rectangular coordinates allows us to use simpler partial derivatives in constructing the state transition matrix F_k. The partial derivatives take the form:

$$
\begin{aligned}
\dot{x}_1(t) &= -\omega\, y_1(t) \\
\dot{y}_1(t) &= \omega\, x_1(t) \\
\dot{x}_2(t) &= -2\,\omega\, y_2(t) \\
\dot{y}_2(t) &= 2\,\omega\, x_2(t)
\end{aligned} \tag{25}
$$

3.2.3 Modeling in rectangular coordinates with time varying $\omega(t)$

In the previous subsections, we assumed that gait velocity ω does not vary with time. It is more realistic to represent ω as time-varying. Because we also want to represent the system in terms of discrete samples, we can define ω_k as the amount of rotation (refer to Figure 14) that occurs from k to $k+1$. Therefore, ω_k can be defined as the stride frequency, expressed in radians/sample.

$$
\omega_k \in \mathbb{R}^1, \quad \omega_k = \int_{\tau=t_k}^{t_{k+1}} \omega(\tau)\, d\tau \qquad \text{[radians/sample]} \tag{26}
$$

Note that in Equation (26), $\omega(\tau)$ is represented as continuous, while ω_k is discrete. Since we are representing our system in rectangular coordinates, we don't have to define ω_k as an integration through an angle, which helps avoid partial derivatives in the time domain.

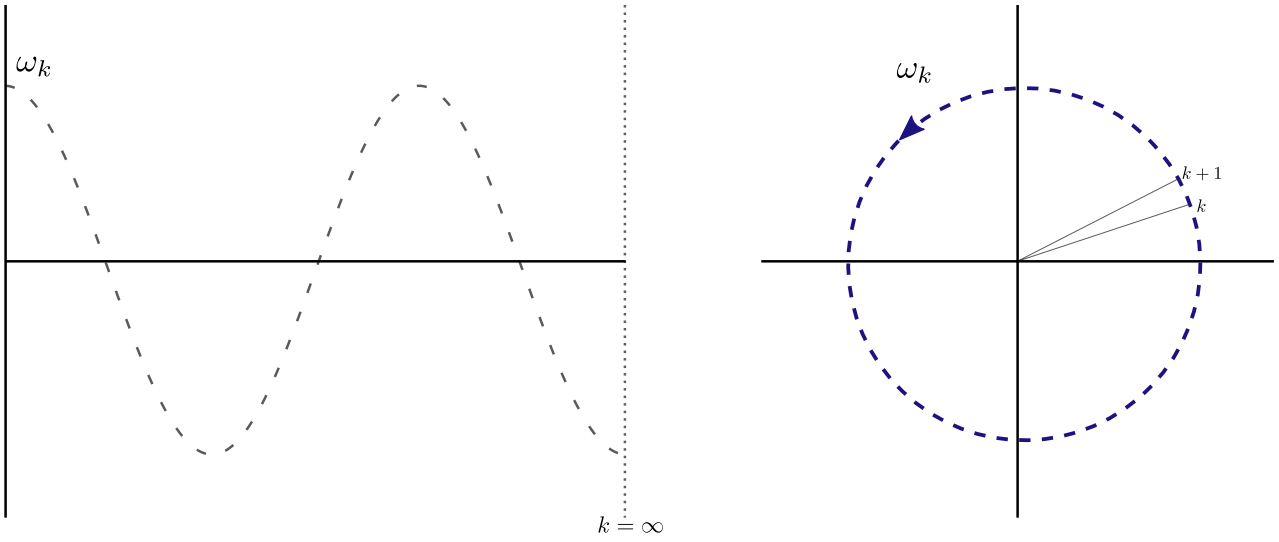

Figure 14: Unit circle representation of harmonic oscillator system. Dashes indicate samples in a discrete system.

Taking the modeling parameters thus far into account, the state vector for this system, using rectangular coordinates, can consist of six parameters. The rotation angle and its second harmonic comprise the first four variables of the state vector: x_1, y_1, x_2, and y_2. Here, x_1 and y_1 form the fundamental, with the magnitude (or amplitude) being the norm of $[x_1, y_1]$ and the phase equal to $atan2\,(y_1, x_1)$. Additionally, x_2 and y_2 form the second harmonic, with magnitude equal to the norm of $[x_2, y_2]$ and the phase equal to $atan2\,(y_2, x_2)$. Each of x_1, y_1, x_2, and y_2 evolves with sample step k.

In order to account for the DC offset (referred to as A_0 in Equation (22)), we add an additional state variable to the model, d_k. Finally, as discussed in the beginning of the section, the model must also be sensitive to changes in gait velocity, so a variable is added to the state vector representing gait rotation per sample, ω_k.

$$x_k^* = \begin{bmatrix} x_1^*(k) \\ y_1^*(k) \\ x_2^*(k) \\ y_2^*(k) \\ d_k^* \\ \omega_k^* \end{bmatrix} \tag{27}$$

$$\hat{x}_k = \begin{bmatrix} \hat{x}_1(k) \\ \hat{y}_1(k) \\ \hat{x}_2(k) \\ \hat{y}_2(k) \\ \widehat{d}_k \\ \hat{\omega}_k \end{bmatrix} \tag{28}$$

Recall that x_k^* and $\hat{x}_k$ represent the true state and estimate of the state respectively.

The Multi-Tap EKF operates by removing observations from the state update equation that contain data dropouts. By only allowing valid observed data into the observation matrix H_k, and by extension the innovation $\tilde{z}_k$, intermittent dropouts or gaps in data do not have an effect on the updated state estimate.

With Equation (27) as the state vector, our measurement vector is:

$$\bar{z}_k = \begin{bmatrix} d_k^* + x_1^*(k - D_0) + x_2^*(k - D_0) \\ d_k^* + x_1^*(k - D_1) + x_2^*(k - D_1) \\ d_k^* + x_1^*(k - D_2) + x_2^*(k - D_2) \\ d_k^* + \overset{*}{x_1}(k - D_3) + x_2^*(k - D_3) \end{bmatrix} + \begin{bmatrix} v(k - D_0) \\ v(k - D_1) \\ v(k - D_2) \\ v(k - D_3) \end{bmatrix} \tag{29}$$

Note the 2nd, 3rd, and 4th elements of the vector used "tapped" historic output samples. Recall D represents a sample offset, as illustrated in Figure 12.

The predicted state estimate for this model, using Equation (10), is

$$\hat{x}_k^- = f(\hat{x}_{k-1}) = \begin{bmatrix} \hat{x}_1(k-1)\,\cos(\hat{\omega}_{k-1}) + \hat{y}_1(k-1)\,\sin(\hat{\omega}_{k-1}) \\ \hat{y}_1(k-1)\,\cos(\hat{\omega}_{k-1}) - \hat{x}_1(k-1)\,\sin(\hat{\omega}_{k-1}) \\ \hat{x}_2(k-1)\,\cos(2\,\hat{\omega}_{k-1}) + \hat{y}_2(k-1)\,\sin(2\,\hat{\omega}_{k-1}) \\ \hat{y}_2(k-1)\,\cos(2\hat{\omega}_{k-1}) - \hat{x}_2(k-1)\,\sin(2\,\hat{\omega}_{k-1}) \\ \hat{d}_{k-1} \\ \hat{\omega}_{k-1} \end{bmatrix}, \tag{30}$$

Recall there is no u_k control input; any changes to gait correspond to noise w_k. The state transition matrix F_k then, is equal to

$$F_k = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_{k-1}^-} = \begin{bmatrix} \cos(\hat{\omega}_{k-1}) & \sin(\hat{\omega}_{k-1}) & 0 & 0 \\ -\sin(\hat{\omega}_{k-1}) & \cos(\hat{\omega}_{k-1}) & 0 & 0 \\ 0 & 0 & \cos(2\,\hat{\omega}_{k-1}) & \sin(2\,\hat{\omega}_{k-1}) \\ 0 & 0 & -\sin(2\,\hat{\omega}_{k-1}) & \cos(2\,\hat{\omega}_{k-1}) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & (-\hat{x}_1(k-1)\,\sin(\hat{\omega}_{k-1}) + \hat{y}_1(k-1)\,\cos(\hat{\omega}_{k-1})) \\ 0 & (-\hat{x}_1(k-1)\,\cos(\hat{\omega}_{k-1}) - \hat{y}_1(k-1)\,\sin(\hat{\omega}_{k-1})) \\ 0 & 2\,(-\hat{x}_2(k-1)\,\sin(2\,\hat{\omega}_{k-1}) + \hat{y}_2(k-1)\,\cos(2\,\hat{\omega}_{k-1})) \\ 0 & 2\,(-\hat{x}_2(k-1)\,\cos(2\,\hat{\omega}_{k-1}) - \hat{y}_2(k-1)\,\sin(2\,\hat{\omega}_{k-1})) \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{31}$$

As explained in Section 2.2, the observation model is now a nonlinear function. Because the Multi-Tap EKF depends on previous observations, we can derive the observation from Equation (11) and express in terms of the state vector $\hat{x}_k^-$. For this application, we will use (up to) four previous observations to construct the observation model.

$$\hat{z}_k = h(\hat{x}_k^-) = \begin{bmatrix} \widehat{d}_k + \hat{x}_1^-\,(k) + \hat{x}_2^-\,(k) \\ \widehat{d}_k + \cos(D_1\;\hat{\omega}_k^-)\;\hat{x}_1^-\,(k) - \sin(D_1\;\hat{\omega}_k^-)\;\hat{y}_2^-\,(k) + \cos(2\;D_1\;\hat{\omega}_k^-)\;\hat{x}_2^-\,(k) - \sin(2\;D_1\;\hat{\omega}_k^-)\;\hat{y}_2^-\,(k) \\ \widehat{d}_k + \cos(D_2\;\hat{\omega}_k^-)\;\hat{x}_1^-\,(k) - \sin(D_2\;\hat{\omega}_k^-)\;\hat{y}_2^-\,(k) + \cos(2\;D_2\;\hat{\omega}_k^-)\;\hat{x}_2^-\,(k) - \sin(2\;D_2\;\hat{\omega}_k^-)\;\hat{y}_2^-\,(k) \\ \widehat{d}_k + \cos(D_3\;\hat{\omega}_k^-)\;\hat{x}_1^-\,(k) - \sin(D_3\;\hat{\omega}_k^-)\;\hat{y}_2^-\,(k) + \cos(2\;D_3\;\hat{\omega}_k^-)\;\hat{x}_2^-\,(k) - \sin(2\;D_3\;\hat{\omega}_k^-)\;\hat{y}_2^-\,(k) \end{bmatrix} \tag{32}$$

The observation matrix is then determined similarly to the state transition matrix, by taking the Jacobian of h.

$$H_k = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_k^-} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ \cos(D_1\ \hat{\omega}_k^-) & -\sin(D_1\ \hat{\omega}_k^-) & \cos(2\ D_1\ \hat{\omega}_k^-) & -\sin(2\ D_1\ \hat{\omega}_k^-) & 1 \\ \cos(D_2\ \hat{\omega}_k^-) & -\sin(D_2\ \hat{\omega}_k^-) & \cos(2\ D_2\ \hat{\omega}_k^-) & -\sin(2\ D_2\ \hat{\omega}_k^-) & 1 \\ \cos(D_3\ \hat{\omega}_k^-) & -\sin(D_3\ \hat{\omega}_k^-) & \cos(2\ D_3\ \hat{\omega}_k^-) & -\sin(2\ D_3\ \hat{\omega}_k^-) & 1 \end{bmatrix} \tag{33}$$

$$\begin{bmatrix} 0 \\ \left(-\sin(D_1\ \hat{\omega}_k^-)\ D_1\ \hat{x}_1(k) - \cos(D_1\ \hat{\omega}_k^-)\ D_1\ \hat{y}_1(k) - 2\ \sin(2\ D_1\ \hat{\omega}_k^-)\ \hat{x}_2(k) - 2\ \cos(2\ D_1\ \hat{\omega}_k^-)\ \hat{y}_2(k)\right) \\ \left(-\sin(D_2\ \hat{\omega}_k^-\ D_2\ \hat{x}_1(k) - \cos(D_2\ \hat{\omega}_k^-)\ D_2\ \hat{y}_1(k) - 2\ \sin(2\ D_2\ \hat{\omega}_k^-)\ \hat{x}_2(k) - 2\ \cos(2\ D_2\ \hat{\omega}_k^-)\ \hat{y}_2(k)\right) \\ \left(-\sin(D_3\ \hat{\omega}_k^-)\ D_3\ \hat{x}_1(k) - \cos(D_3\ \hat{\omega}_k^-)\ D_3\ \hat{y}_1(k) - 2\ \sin(2\ D_3\ \hat{\omega}_k^-)\ \hat{x}_2(k) - 2\ \cos(2\ D_3\ \hat{\omega}_k^-)\ \hat{y}_2(k)\right) \end{bmatrix}$$

3.3 Steps of the Multi-Tap EKF

The first steps in Algorithm 1 involve finding the prediction of the state and covariance. Refer to Figure 15. Since the predicted state estimate is found with Equation (30), the next step is to find the predicted estimate covariance, as shown in Equation (13). This equation uses F_0, given by Equation (31), and user-defined initial covariance P_0 and process noise Q_0.

Next, Kalman gain K_k is found using the observation model H_k (refer to Equation (14) and (15)). The state estimate is also updated, *a priori*, using our observation estimate and measurement, per Equations (16) and (17).

The Multi-Tap Kalman Filter then makes several actions, depending on how many valid observed measurements there are at time k.

1. First, the number of valid measurements from k to $k - D_n$ is determined, where D_n is the tap the furthest sample in the past (for our application, it is D_3).

2. The observation H_k is then adapted to fit the number of valid measurements, as are the residual measurement vector $\tilde{z}_k$ and observation noise R_k.

3. Next, the Kalman gain is found using the refactored H_k and R_k.

4. Finally, the state estimate and error covariance are determined.

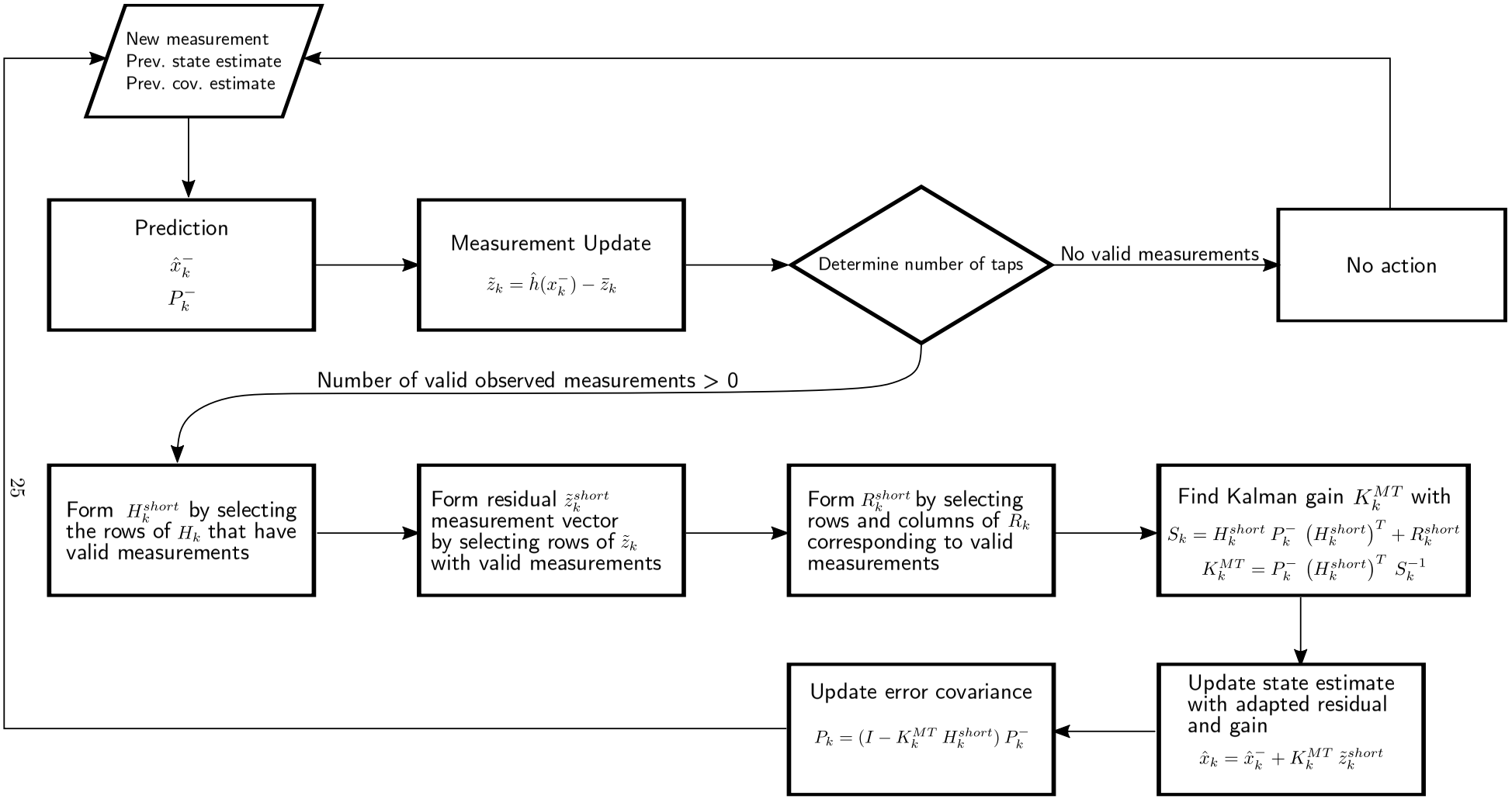

Figure 15: Functional block diagram of the Multi-Tap Extended Kalman Filter. H_k, $\tilde{z}_k$, and R_k are modified (*short*) to omit invalid measurements. The equations to find the Multi-Tap Kalman gain K_k^{MT} mirror Equations (14) and (15). The *posteriori* state estimate equation mirrors Equation (17). The error covariance equation using adapted Kalman gain and observation matrices mirrors Equation (18).

4 Analysis of experimental data

4.1 How to observe the state adjustment

Since the direction of the state space model rotates over time, it would be advantageous to construct a statistic that evaluates the filter's performance over a given period. Since we have unobservable modes with a single tap, we want to show that we have comparably fast convergence and lock in of our states with multiple taps. In other words, in order to analyze the performance of the Multi-Tap EKF, we need to observe how the difference between the estimate and true state evolves over each step and compare this observation with a traditional standard EKF approach. Therefore, it is beneficial to define the state adjustment at time step k in terms of the state adjustment at *k-1*, or $\tilde{x}_k = E_k\ \tilde{x}_{k-1}$, where E_k can be referred to as a "State Error Transition Matrix."

4.1.1 Basic expressions for the state estimate, residual, and observation

This subsection defines a basic set of equations for the state and its residual correction, as well as the observation, for use in the following sections.

Referring to Equation (10), the true state of our system model takes the following form:

$$x_k^* = f(x_{k-1}^*) + w_{k-1}. \tag{34}$$

Recall from Equation (12) that $\hat{x}_k^- = f(\hat{x}_{k-1})$ and assume no control signal u_k is present for this system model.

Assuming the difference between the estimate and true state is

$$\tilde{x}_k = \hat{x}_k - x_k^*, \tag{35}$$

we can rearrange to form

$$\hat{x}_k = \tilde{x}_k + x_k^*. \tag{36}$$

The true output of the system takes the form

$$\overset{*}{z}_k = h(x_k^*), \tag{37}$$

making the estimate of the output

$$\hat{z}_k = h(\hat{x}_k), \tag{38}$$

and the measurement of the output

$$\bar{z}_k = h(x_k^*) + v_k, \tag{39}$$

where v_k is measurement noise. Recalling the expression for the *a priori* state estimate, defined in Equations (16) and (17), the state estimate is expressed as

$$\hat{x}_k = \hat{x}_k^- - H_k\,\tilde{z}_k = \hat{x}_k^- - K_k(\hat{z}_k - \bar{z}_k). \tag{40}$$

Now substituting our output estimate and measurement using Equations (38) and (39), we arrive at

$$\hat{x}_k = \hat{x}_k^- - K_k\left(h\left(\hat{x}_{k-1}\right) - h\left(x_{k-1}^*\right) - v_{k-1}\right). \tag{41}$$

4.1.2 Expressing $\tilde{x}_k$ in terms of $\tilde{x}_{k-1}$

Recall the goal of this analysis is to express the evolution of the state adjustment in the form:

$$\tilde{x}_k = E_k\,\tilde{x}_{k-1}. \tag{42}$$

where E_k is the State Error Transition Matrix.

As outlined in Section 4.1.1,

$$x_k^* = f(x_{k-1}^*) + w_{k-1}\ ,\ \hat{x}_k^- = f(\hat{x}_{k-1}) \tag{43}$$

Recall that due to the non-linear nature of process, the covariance prediction and update equations cannot use f and h directly. Rather they use the Jacobian of f and h, which we assume are known. The Jacobians are defined as

$$F_k = \left.\frac{\partial f(\cdot)}{\partial x}\right|_{\hat{x}_{k-1}}\ ,\ H_k = \left.\frac{\partial h(\cdot)}{\partial x}\right|_{\hat{x}_k^-}. \tag{44}$$

When $\hat{x}_{k-1} \simeq x_{k-1}^*$ (or $\|\tilde{x}_{k-1}\|$ is small), we can estimate $f(\hat{x}_{k-1}) - f(x_{k-1}^*)$ as the first term of the Taylor Series Expansion

$$f(\hat{x}_{k-1}) - f(x_{k-1}^*) \simeq F\tilde{x}_{k-1} \tag{45}$$

Note that if $\|\tilde{x}_{k-1}\|$ were not small, the other terms present in the Taylor series become significant and don't drop out.

Using Equation (40), we know that our update takes the form $\hat{x}_k = \hat{x}_k^- - K_k(\tilde{z}_k) - v_k$, or

$$\hat{x}_k = f(\hat{x}_{k-1}) - K_k \left(h\left(\hat{x}_k^-\right) - h\left(x_k^*\right) - v_k \right) \tag{46}$$

From Taylor Series Expansion, and approximation using the first term,

$$H_k \tilde{x}_{k-1} \simeq h\left(\hat{x}_{k-1}\right) - h\left(x_{k-1}^*\right) \tag{47}$$

$$F_{k-1} \tilde{x}_{k-1} \simeq f\left(\hat{x}_{k-1}\right) - f\left(x_{k-1}^*\right) \tag{48}$$

If we refer to $\tilde{z}_k$ as

$$\tilde{z}_k = \hat{z}_k - \bar{z}_k \;=\; h\left(\hat{x}_k^-\right) - \left(h\left(x_k^*\right) - v_k\right), \tag{49}$$

and substitute $\hat{x}_k^-$ and x_k^* using Equation (43), we get

$$\tilde{z}_k = h\left(f(\hat{x}_{k-1})\right) - h\left(f(x_{k-1}^*) + w_k\right) - v_k. \tag{50}$$

By Taylor Series Expansion (Equations (47) and (48)) and assuming $\|w_k\|$ is small,

$$\tilde{z}_k \simeq h\left(f(\hat{x}_{k-1})\right) - h\left(f(x_{k-1}^*)\right) - H_k w_k - v_k. \tag{51}$$

By the chain rule of differentiation,

$$\frac{\partial h\left(f(x_k)\right)}{\partial x} = \left.\frac{\partial h}{\partial x}\right|_k \left.\frac{\partial f}{\partial x}\right|_k = H_k F_k. \tag{52}$$

Therefore, Equation 51 takes the form

$$\tilde{z}_k \simeq\; H_k\, F_k\, \left(\hat{x}_{k-1} - x_{k-1}^*\right) - H_k w_k - v_k. \tag{53}$$

Using Equation (35), we can also write $\tilde{z}_k \simeq H_k\, F_k\left(\tilde{x}_{k-1}\right) - H_k\, w_k - v_k$. Therefore, rewriting Equation (46) in terms of $\tilde{z}_k$,

$$\hat{x}_k \;\simeq\; f(\hat{x}_{k-1}) - K_k\left(H_k\, F_k\, \tilde{x}_{k-1} - H_k\, w_k - v_k\right). \tag{54}$$

Solving for $\tilde{x}_k$ using Equation (43),

$$\tilde{x}_k = \hat{x}_k - x_k^* = f(\hat{x}_{k-1}) - K_k\,(H_k\,F_k\,\tilde{x}_{k-1} - H_k w_k - v_k) - f(x_{k-1}^*) + w_{k-1}. \tag{55}$$

We now make our estimation substitution using Equation (48),

$$\tilde{x}_k = F_k\,\tilde{x}_{k-1} - K_k\,H_k\,F_k\tilde{x}_{k-1} + K_k\,H w_k + K_k\,v_k - w_{k-1}. \tag{56}$$

Refactoring terms on the right side in terms of $\tilde{x}_{k-1}$,

$$\tilde{x}_k = (F_k - K_k\,H_k\,F_k)\,\tilde{x}_{k-1} + g_k\,, \qquad g_k = K_k\,H\,w_k + K_k\,v_k - w_{k-1}, \tag{57}$$

we can now define E_k, the State Error Transition Matrix, as

$$E_k = F_k - K_k\,H_k\,F_k. \tag{58}$$

E_k determines the stability of the observed and convergence rate. Now, we may rewrite Equation (57) as

$$\tilde{x}_k = E_k\,\tilde{x}_{k-1} + g_k \tag{59}$$

4.2 Finding the State Estimate Error Transition Matrix for an entire stride

If we take a sample size from an experimental data set tracking the ankle, we can record a vector of State Error Transition matrices, E_k, which extends over the entire sample set, or a period of samples within that set.

Each state adjustment $\tilde{x}_k$ is a product of the previous state and the other state adjustment matrices before it, such that $\tilde{x}_2 = E_2\,\tilde{x}_1$, $\tilde{x}_3 = E_3\,\tilde{x}_2 = E_3\,E_2\,\tilde{x}_1$...

Alternatively, to get from step k_1 to k_2, where $k_2 - k_1$ is the number of samples it takes to complete one stride,

$$\tilde{x}_{k_2} = \left(\prod_{j=k_1}^{k_2} E_j\right)\tilde{x}_{k_1} \tag{60}$$

where $\tilde{x}_{k_1} \in \mathbb{R}^6$. We can also assign $\Phi_{k_1}^{k_2} = \prod_{j=k_1}^{k_2} E_j$, where $\Phi_{k_1}^{k_2} \in \mathbb{R}^{6\text{x}6}$ for our system model. Φ can then be referred to as the Product State Error Transition Matrix. Expressed in equation form,

$$\tilde{x}_{k_2} = \Phi_{k_1}^{k_2}\, \tilde{x}_{k_1}$$

4.3 Analyzing real data sets

For this data set analysis, 2,859 data records of individual experimental trials were examined. Trials spanned across various individual age, gender and size categories and covered treadmill paces of 2, 3, 5, 6, 7, 8, and 9 miles per hour. After heel-strike and toe-off positions were determined using post processing and local minima and maxima search techniques, a full stride was taken from each of the trials and $\Phi_{k_1}^{k_2}$ was determined for that stride, by taking the product of the State Transition Error Matrices through all of the samples, as explained in Equation (60).

For 5 mph data, shown in Figure 16, the maximum eigenvalues of $\Phi_{k_1}^{k_2}$, defined as $\bar{\lambda}_{\Phi}$, is quite large for the one-tap and two-tap conditions as compared with the three and four-tap implementation.

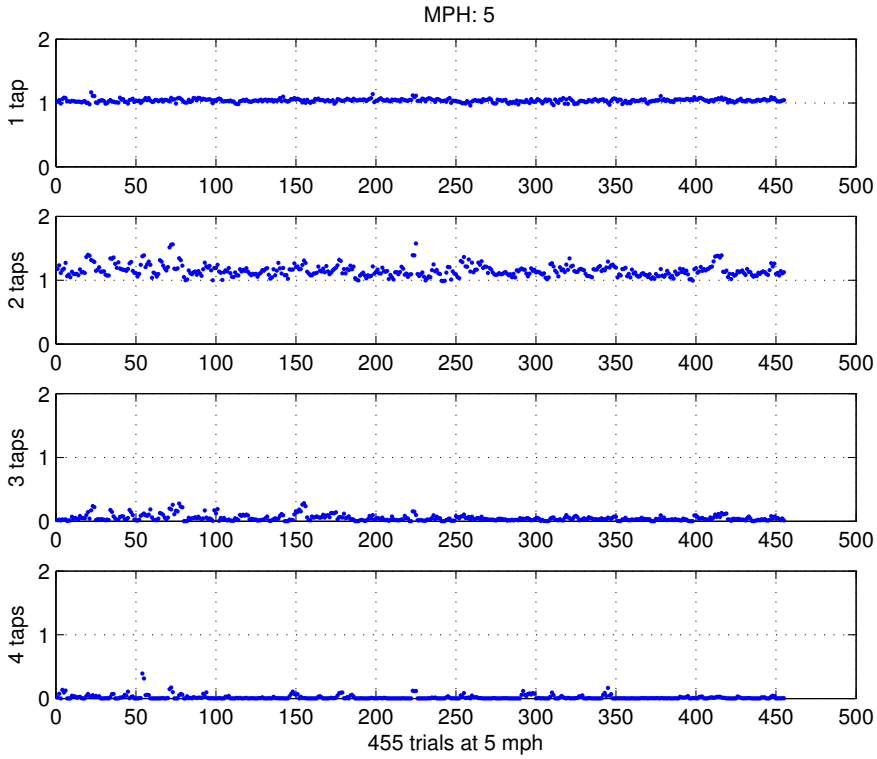

Figure 16: Maximum eigenvalues $\bar{\lambda}_{\Phi}$ taken for 5 mph data. For each tap number configuration (1-4 taps), each point represents the max eigenvalue of the Product State Error Transition Matrix over one complete stride.

The figure shows that some errors may not be reduced by 1 or 2-tap implementations, but strongly reduced by having 3 or 4 taps.

The data at 8 mph, as shown in Figure (17), shows more varying max eigenvalues $\bar{\lambda}_{\Phi}$ for the 3-tap implementation.

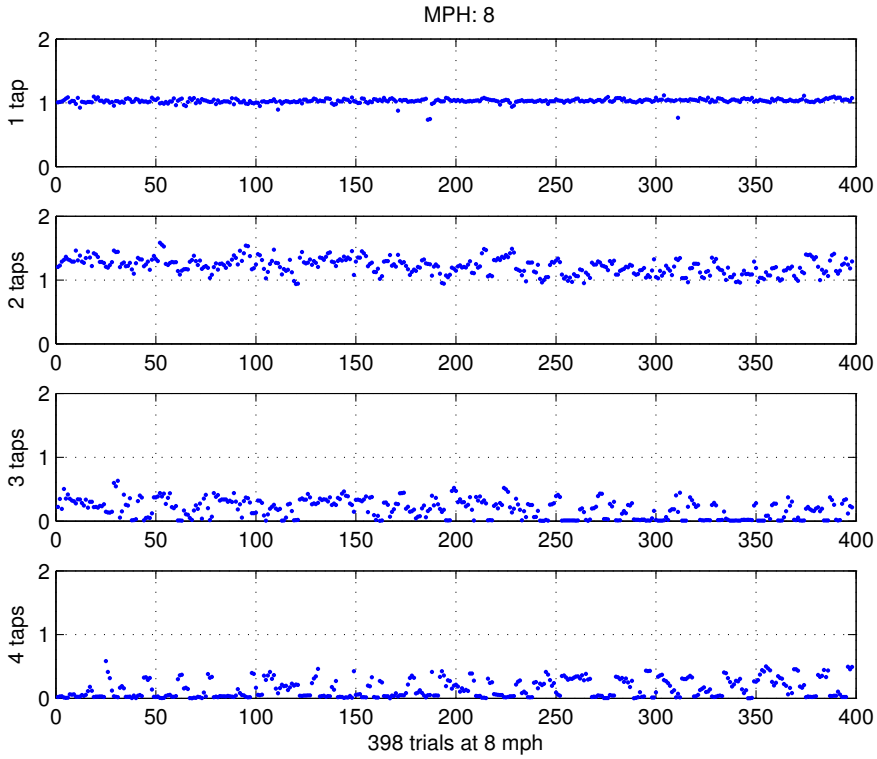

Figure 17: Maximum eigenvalues $\bar{\lambda}_{\Phi}$ taken for 8 mph data. For each tap number configuration (1-4 taps), each point represents the max eigenvalue of the Product State Error Transition Matrix over complete stride.

With the stride changing, the number of samples per stride changing, and dropout frequency changing, any combination of these varying characteristics can result in a change in the eigenvalues of $\Phi_{k_1}^{k_2}$.

The data at 2 mph, as shown in Figure (18), shows that in the 3 and 4-tap filter implementations, max eigenvalues $\bar{\lambda}_{\Phi}$ vary substantially as compared with the higher speeds.

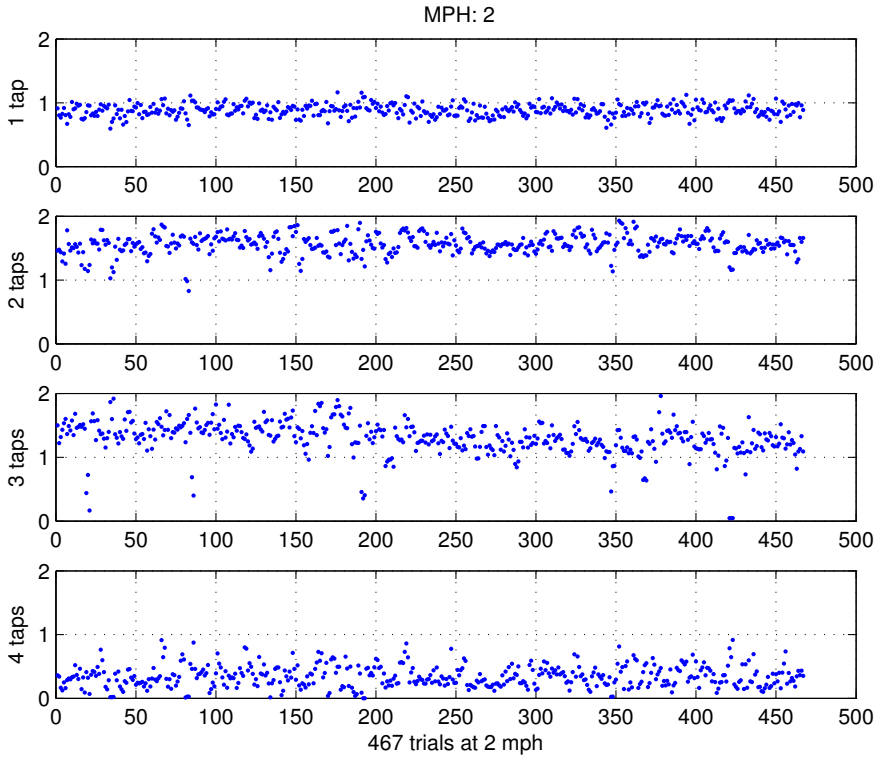

Figure 18: Maximum eigenvalues $\bar{\lambda}_{\Phi}$ taken for 2 mph data. For each tap number configuration (1-4 taps), each point represents the max eigenvalue of the Product State Error Transition Matrix over one complete stride.

The Product State Error Transition Matrix is shown to be unstable with a standard one-tap or two-tap implementation, but more stable with three or more taps. Note that having the second harmonic in the state model increases the uncertainty in the estimation. However, because the waveform changes shape (it is more sawtooth and asymmetric at low speed, while sinusoidal and symmetric at high speed), having the second harmonic permits asymmetry.

If the maximum maximum eigenvalues $\bar{\lambda}_{\Phi}$ for Figures 16, 17, and 18 are plotted, we see in Figure 19 that only at speeds at 5 mph and above are the 3 and 4-tap configurations stable for all trials. The 4-tap implementation is the only configuration which is stable for all trials at all speeds.

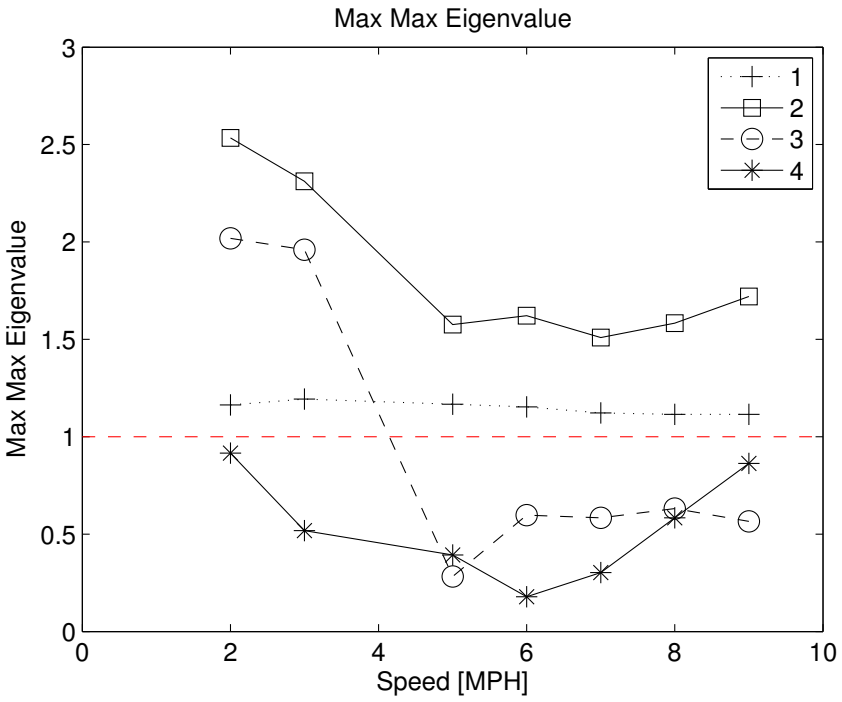

Figure 19: Maximum maximum eigenvalues $\bar{\lambda}_{\Phi}$ taken for all speeds, all trials.

5 Analysis using simulated data

This section will use simulated data to evaluate the performance of the Multi-Tap EKF. Simulated signals were generated for several trials with varying speed and dropout rate, with assumptions made on DC offset and frequency drift amplitude and sensor signal noise.

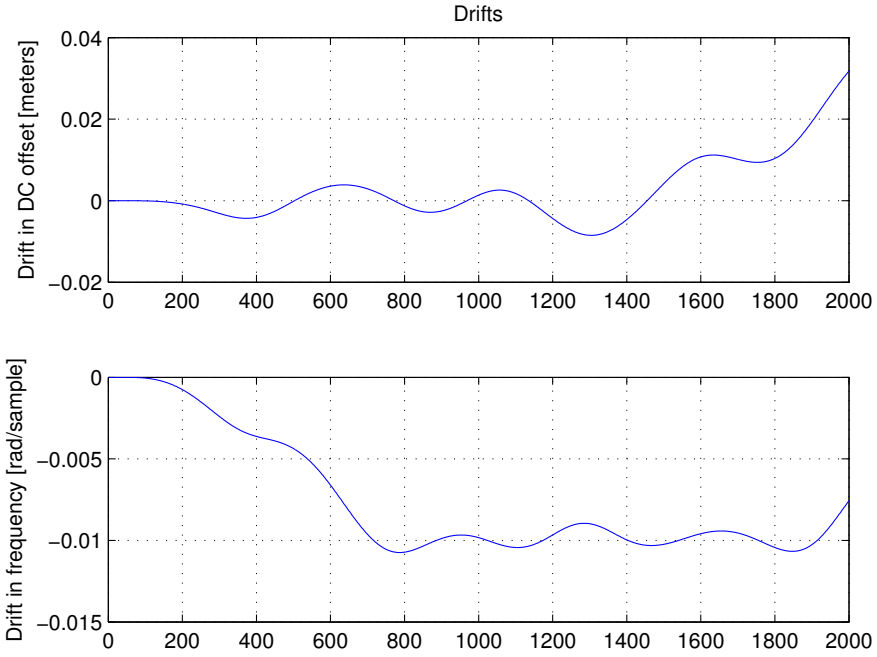

Figure 20: Example simulated DC offset and frequency drift

Over a sample size of 1000 trials looking at variations in dropout from 0-50%, stride rate varying from 20-120 strides per minute, with each variation running from 1-4 taps in the multi-tap filter, several trends emerged. Figure 21 shows one set of data for a 3-tap configuration on a single trial, where each state estimate and true value are plotted.

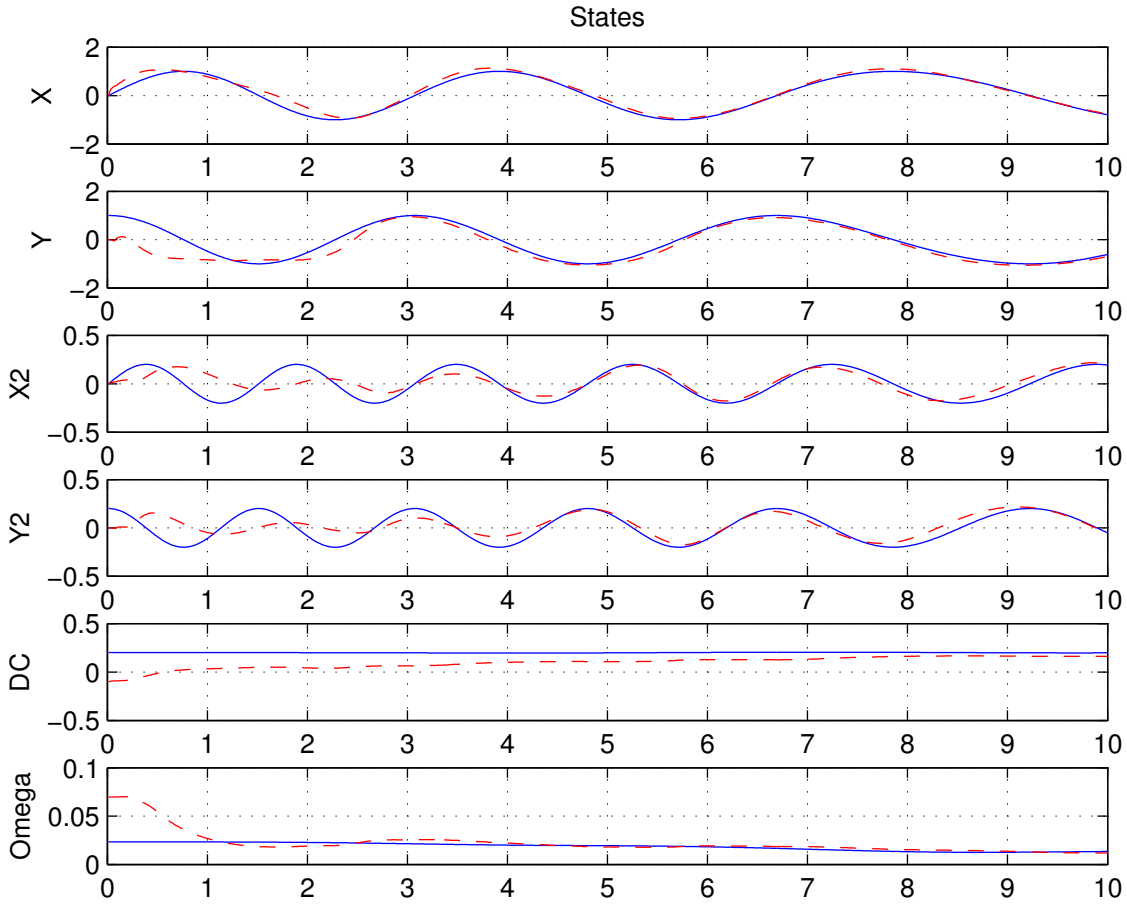

Figure 21: State estimates for 3-tap configuration across one trial of data containing more than 10 seconds of strides, with a stride rate of 20 strides per minute and dropout rate of 0%. True state shown in blue solid line, estimate shown in red dotted line. Performance of various tap configurations and correlation of tap number with stride rate and dropout percent were studied.

Studying the performance of the Multi-Tap configuration against the standard EKF, the DC residual correction is lower when utilizing three or more taps. Above 100 strides/min, the performance of the standard EKF breaks down as larger errors in the DC drift appear. Refer to Figure 22.

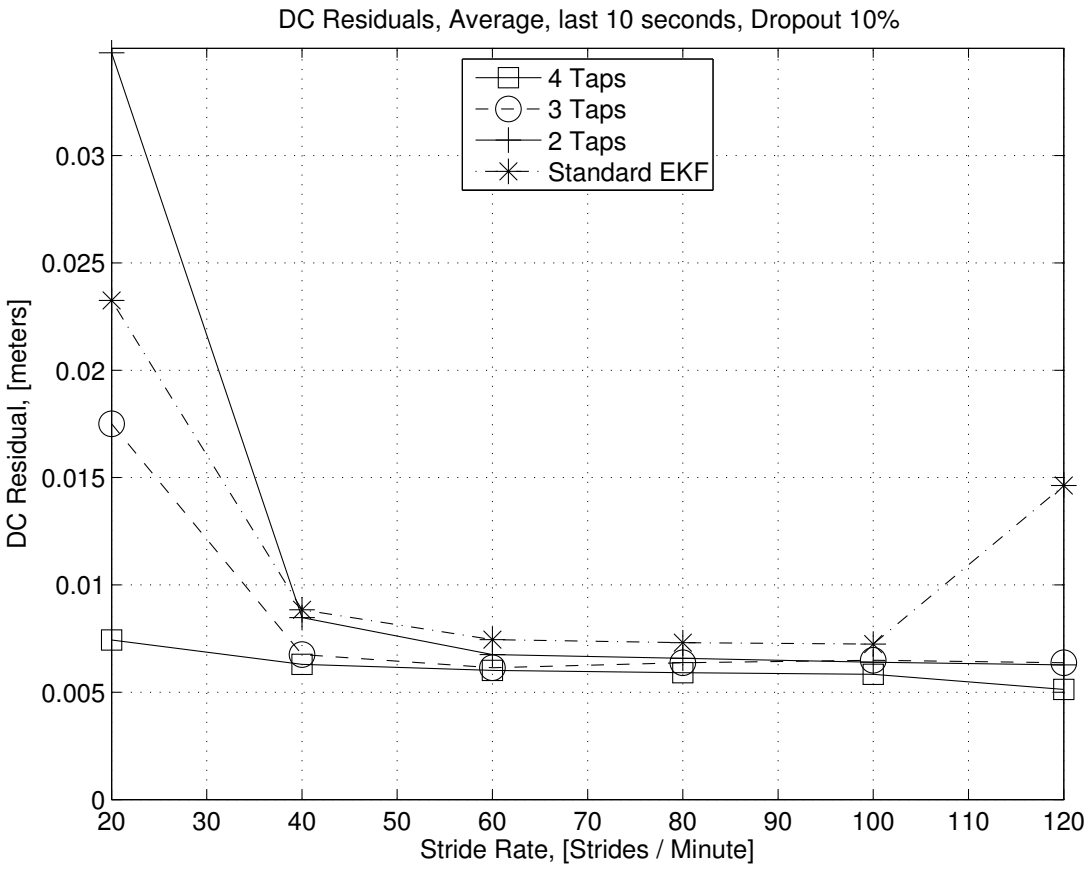

Figure 22: DC residual correction across various stride rates, data collected across 1000 trials with varying stride rate, assuming 10% dropout

Observing the DC residual correction over time for a single stride rate, as shown in Figure 23, all Multi-Tap configurations outperform the standard EKF with regard to the speed of convergence of the residual error to zero, a measure of how fast the filter locks in. With the ultimate goal of locking in and providing real-time feedback within 1-2 strides, the 4-tap configuration meets the objective.

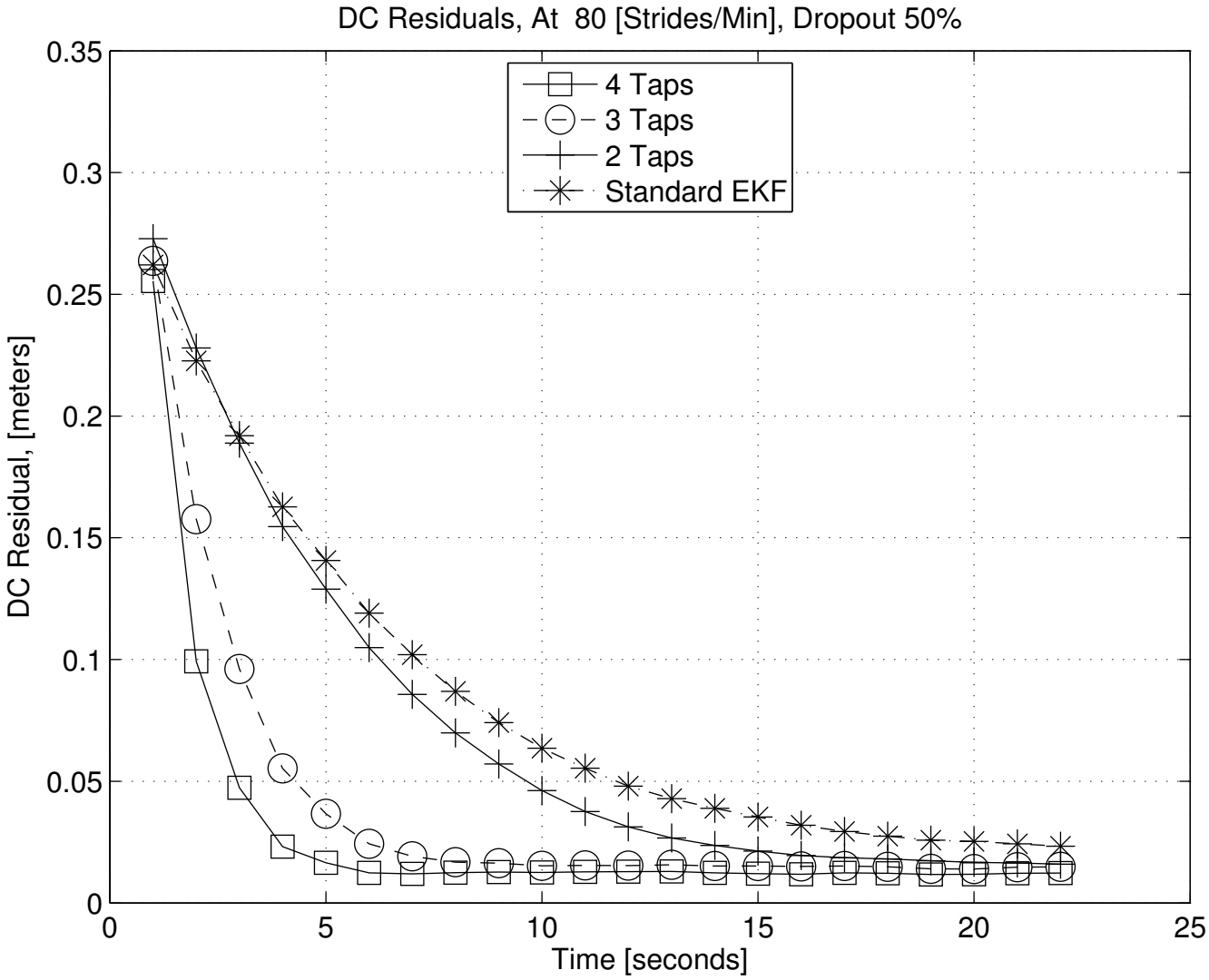

Figure 23: DC residual correction at 80 strides per minute. Data collected across 1000 trials with varying stride rate, assuming 50% dropout. Note the convergence time improvement of the 3 and 4 tap configurations.

A performance improvement can also be observed when viewing the average rate residual over the last 10 seconds (allowing sufficient time for each filter configuration to lock in). Figure 24 shows that at lower stride rates, three and four-tap configurations require the least amount of correction using the residual, with the standard EKF errors growing by approximately 5 times at stride rates above 100 strides/minute.

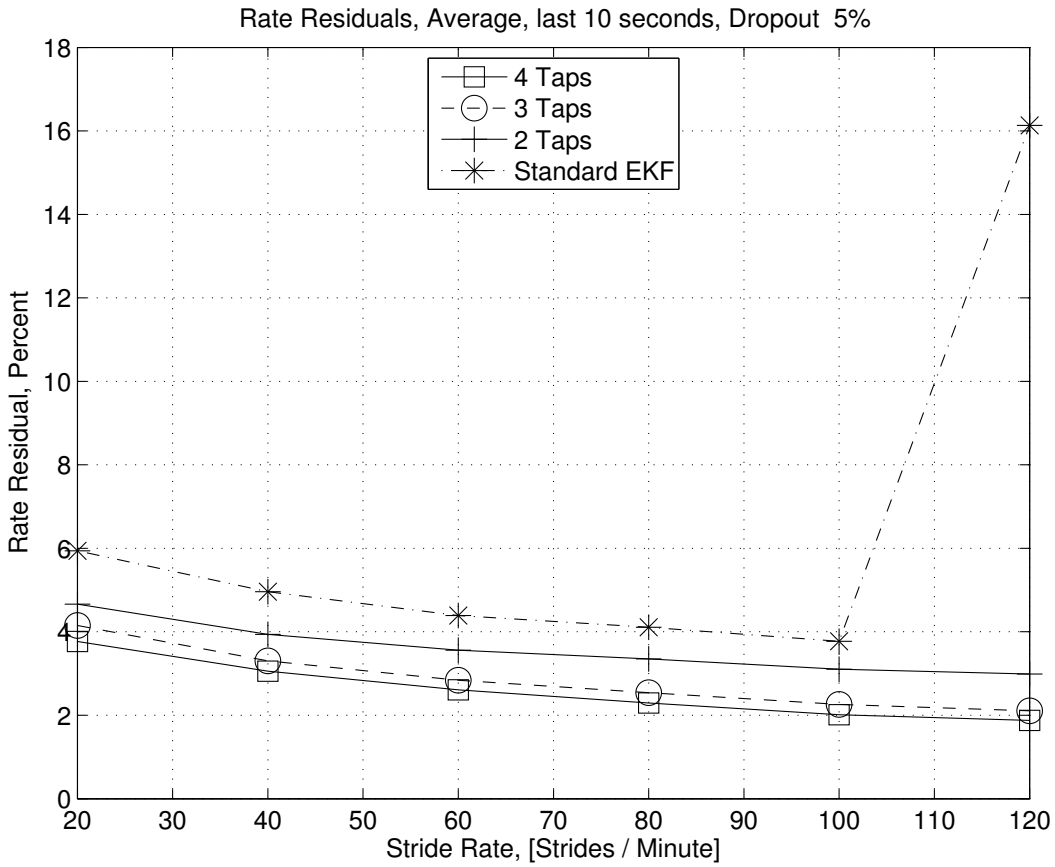

Figure 24: Rate residual correction across various stride rates, data collected across 1000 trials. Dropout at 5%.

Higher data dropout rates, such as at 50% shown in Figure 25, result in larger residuals, with some larger residuals being introduced at lower stride rates. In this condition, the Multi-Tap EKF manages to keep residual error under 15%. As before, at stride rates above 100 strides/min, standard EKF shows poor performance (as measured by size of the average residual), when the data dropout rate is 50%.

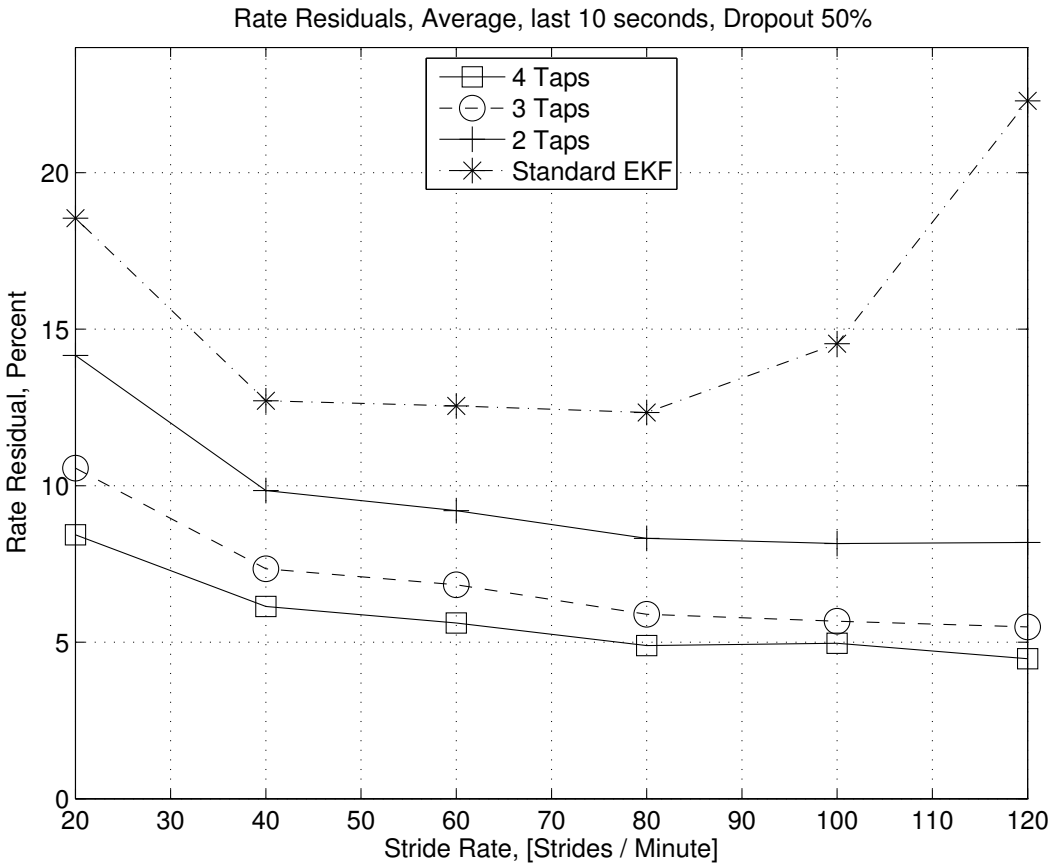

Figure 25: Rate residual correction across various stride rates, data collected across 1000 trials. Dropout at 50%.

Similar to Figure 23, when observing residual rate correction over time for a single stride rate, a significant improvement is observed in residual convergence to zero, even changing in configuration from single tap (standard EKF) to dual tap configurations. The three and four-tap configurations converge within a 5-second window, but even after allowing the standard EKF a full 20 seconds to converge, the three and four-tap configurations show an improvement almost by a factor of 3, once stabilized.

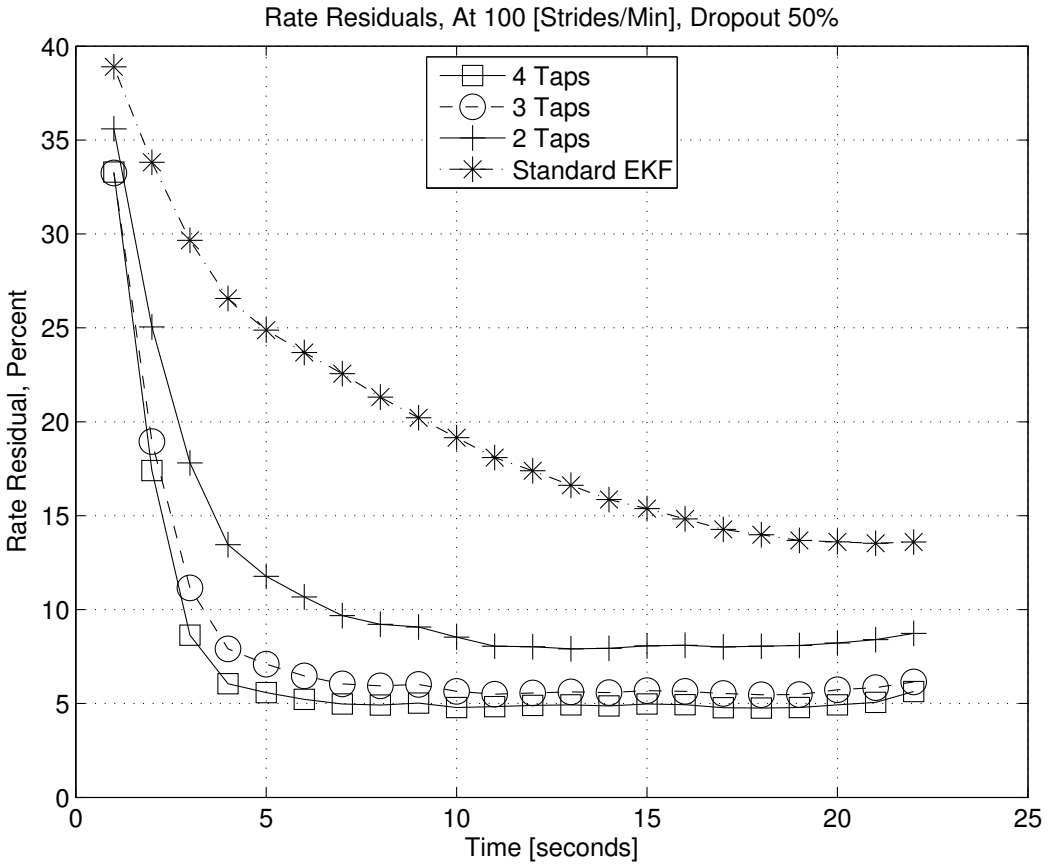

Figure 26: Rate residual correction for 100 strides per minute. Data collected across 1000 trials with varying stride rate, assuming 50% dropout.

6 Conclusion and Discussion

A novel structure for the Extended Kalman Filter is introduced, which makes repeated use of previous observations. By doing so, there is substantial improvement in facing many signal source challenges. We have shown that this novel construction is able to reliably lock into signals that exhibit varying frequency, DC offset, amplitude and waveform shape in the presence of dropouts in data. Having all of these challenges motivates us to use the Multi-Tap Extended Kalman Filter.

6.1 Impact of this Thesis

A method has been presented which can reliably lock onto and estimate the state of challenging gait data. This method can be utilized in similar real-time applications where it is advantageous to use a model-based filtering method to solve the following challenges: periodic signal with changing waveform shape, changing frequency and amplitude, changing DC offset, and data dropouts.

6.2 Future investigations

6.2.1 Preventing second harmonic lock on

It would be worth investigating methods of preventing the Multi-Tap EKF to lock onto the second harmonic in situations where it is not desired. For example, with our current filter, if the initial stride rate estimate is 60 strides/minute, and the actual stride rate is 120 strides/minute, the second harmonic of the model can lock on. An improvement to the Multi-Tap EKF defined thus far would be to measure the magnitude of the fundamental and the magnitude of the second harmonic. The fundamental magnitude should be much larger than the second harmonic magnitude; if this property is not obeyed, the filter can be reinitialized with adjusted initial conditions.

6.2.2 Finding the correct covariance estimate P_k

Within the Kalman Filter framework, P_k is determined based on an assumption of uncorrelated samples. However, by using a given sample several times in the calculation of the state adjustment, previous observations will have influence again and again on the state estimate. The hypothesis of uncorrelated $\tilde{x}_k$ and v_k therefore is violated and P_k is not computed correctly. It may be worth estimating the effect that a miscalculation of P_k has on the calculation of the Kalman gain K_k. For the scope of this paper, it has been assumed that this effect is insignificant since the Moiré Phase Tracking system utilized has high accuracy at sub-millimeter levels, while variations in the movement are repeatedly sampled on the order of centimeters.

In other words, the process noise, w_k, is likely more significant than the measurement noise, v_k. Future studies can quantify the effect the of the correlation.

6.2.3 Real time tap delay adjustment

Perhaps another optimization would involve being able to tune the sampling delay (values D_1, D_2, D_3 in Equation (21), illustrated in Figure 12). If the filter displays signs of poor lock-on time or tracking, the delay between taps or insertion of additional taps could be adjusted real-time.

References

[1] C. Kambhamettu, D. B. Goldgof, D. Terzopoulos, and T. S. Huang, "Nonrigid motion analysis," in *Handbook of pattern recognition and image processing (vol. 2)*, pp. 405–430, Academic Press, Inc., 1994.

[2] S. A. Niyogi, E. H. Adelson, *et al.*, "Analyzing and recognizing walking figures in xyt," in *CVPR*, vol. 94, pp. 469–474, 1994.

[3] J. Zeni Jr, J. Richards, and J. Higginson, "Two simple methods for determining gait events during treadmill and overground walking using kinematic data," *Gait & posture*, vol. 27, no. 4, pp. 710–714, 2008.

[4] B. Armstrong, T. Verron, L. Heppe, J. Reynolds, and K. Schmidt, "RGR-3D: Simple, cheap detection of 6-DOF pose for tele-operation, and robot programming and calibration," in *Proc. 2002 Int. Conf. on Robotics and Automation*, pp. 2938–2943, IEEE: Washington, 2002.

[5] K. M. O'Connor, B. S. R. Armstrong, J. Weinhandl, T. P. Kusik, and R. T. Barrows, "Validation of a single camera 3D motion tracking system," in *Proc. Annual Meeting of the American Society of Biomechanics*, p. 980, ABS: State College, Penn., Aug 2009.

[6] R. M. Kiss, "Comparison between kinematic and ground reaction force techniques for determining gait events during treadmill walking at different walking speeds," *Medical engineering & physics*, vol. 32, no. 6, pp. 662–667, 2010.

[7] J. K. De Witt, "Determination of toe-off event time during treadmill locomotion using kinematic data," *Journal of Biomechanics*, vol. 43, no. 15, pp. 3067–3069, 2010.

[8] F. Alvim, L. Cerqueira, A. D. Netto, G. Leite, and A. Muniz, "Comparison of five kinematic-based identification methods of foot contact events during treadmill walking and running at different speeds," *Journal of applied biomechanics*, vol. 31, no. 5, pp. 383–388, 2015.

[9] L. Smith, S. Preece, D. Mason, and C. Bramah, "A comparison of kinematic algorithms to estimate gait events during overground running," *Gait & posture*, vol. 41, no. 1, pp. 39–43, 2015.

[10] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[11] G. Welch, G. Bishop, *et al.*, "An introduction to the kalman filter," 1995.

[12] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[13] R. Caballero-Águila, A. Hermoso-Carazo, J. D. Jiménez-López, J. Linares-Pérez, and S. Nakamori, "Recursive estimation of discrete-time signals from nonlinear randomly delayed observations," *Computers & Mathematics with Applications*, vol. 58, no. 6, pp. 1160–1168, 2009.

[14] A. Hermoso-Carazo and J. Linares-Pérez, "Nonlinear estimation applying an unscented transformation in systems with correlated uncertain observations," *Applied Mathematics and Computation*, vol. 217, no. 20, pp. 7998–8009, 2011.

[15] S. Dey, A. S. Leong, and J. S. Evans, "Kalman filtering with faded measurements," *Automatica*, vol. 45, no. 10, pp. 2223–2233, 2009.

[16] P. Chen, H. Ma, S. Gao, and Y. Huang, "Modified extended kalman filtering for tracking with insufficient and intermittent observations," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[17] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[18] X. Liu, L. Li, Z. Li, T. Fernando, and H. H. Iu, "Stochastic stability condition for the extended kalman filter with intermittent observations," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 3, pp. 334–338, 2017.

[19] S. Kar, B. Sinopoli, and J. M. Moura, "Kalman filtering with intermittent observations: Weak convergence to a stationary distribution," *IEEE Transactions on Automatic Control*, vol. 57, no. 2, pp. 405–420, 2012.

[20] J. Hu, Z. Wang, B. Shen, and H. Gao, "Quantised recursive filtering for a class of nonlinear systems with multiplicative noises and missing measurements," *International Journal of Control*, vol. 86, no. 4, pp. 650–663, 2013.

[21] J. Hu, Z. Wang, S. Liu, and H. Gao, "A variance-constrained approach to recursive state estimation for time-varying complex networks with missing measurements," *Automatica*, vol. 64, pp. 155–162, 2016.

[22] M. Huang and S. Dey, "Stability of kalman filtering with markovian packet losses," *Automatica*, vol. 43, no. 4, pp. 598–607, 2007.

[23] K. You, M. Fu, and L. Xie, "Mean square stability for kalman filtering with markovian packet losses," *Automatica*, vol. 47, no. 12, pp. 2647–2657, 2011.

[24] E. R. Rohr, D. Marelli, and M. Fu, "Kalman filtering with intermittent observations: On the boundedness of the expected error covariance," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2724–2738, 2014.

[25] C.-B. Chang and M. Athans, "Hypothesis testing and state estimation for discrete systems with finite-valued switching parameters," tech. rep., Massachusetts Institute of Technology Electronic Systems Lab, 1977.

[26] C.-B. Chang and M. Athans, "State estimation for discrete systems with switching parameters," *IEEE Transactions on Aerospace and Electronic Systems*, no. 3, pp. 418–425, 1978.

[27] X.-R. Li and Y. Bar-Shalom, "Multiple-model estimation with variable structure," *IEEE Transactions on Automatic control*, vol. 41, no. 4, pp. 478–493, 1996.

[28] C. Hide, T. Moore, and M. Smith, "Multiple model kalman filtering for gps and low-cost ins integration," in *Proceedings of ION GNSS*, pp. 21–24, 2004.

[29] P. D. Hanlon and P. S. Maybeck, "Multiple-model adaptive estimation using a residual correlation kalman filter bank," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 2, pp. 393–406, 2000.

[30] S. Nanda, M. Biswal, and P. Dash, "Estimation of time varying signal parameters using an improved adaline learning algorithm," *AEU-International Journal of Electronics and Communications*, vol. 68, no. 2, pp. 115–129, 2014.

[31] M. Joorabian, S. Mortazavi, and A. Khayyami, "Harmonic estimation in a power system using a novel hybrid least squares-adaline algorithm," *Electric power systems research*, vol. 79, no. 1, pp. 107–116, 2009.

[32] G. W. Chang, C.-I. Chen, and Q.-W. Liang, "A two-stage adaline for harmonics and interharmonics measurement," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 2220–2228, 2009.

[33] A. Sarkar, S. R. Choudhury, and S. Sengupta, "A self-synchronized adaline network for on-line tracking of power system harmonics," *Measurement*, vol. 44, no. 4, pp. 784–790, 2011.

[34] P. Dash, D. Swain, A. Routray, and A. Liew, "Harmonic estimation in a power system using adaptive perceptrons," *IEEE Proceedings-Generation, Transmission and Distribution*, vol. 143, no. 6, pp. 565–574, 1996.