Spring 2022

ONLINE PILOT MODEL PARAMETER ESTIMATION FOR LOSS-OF-CONTROL PREVENTION IN AIRCRAFT SYSTEMS

Frederick Schill
schillf@my.erau.edu

ONLINE PILOT MODEL PARAMETER ESTIMATION FOR LOSS-OF-CONTROL PREVENTION IN AIRCRAFT SYSTEMS

By

Frederick Thomas Schill

A Thesis Submitted to the Faculty of Embry-Riddle Aeronautical University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Aerospace Engineering

April 2022

Embry-Riddle Aeronautical University

Daytona Beach, Florida

ONLINE PILOT MODEL PARAMETER ESTIMATION FOR LOSS-OF-CONTROL

PREVENTION IN AIRCRAFT SYSTEMS

By

Frederick Thomas Schill

This Thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Hever Moncayo, Department of Aerospace Engineering, and has been approved by the members of the Thesis Committee. It was submitted to the Office of the Senior Vice President for Academic Affairs and Provost, and was accepted in the partial fulfillment of the requirements for the Degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE

Hever Moncayo Digitally signed by Hever Moncayo
Date: 2022.04.21 10:34:17 -04'00'

Chair, Dr. Hever Moncayo

Richard J. Prazenica Digitally signed by Richard J. Prazenica
Date: 2022.04.22 09:03:07 -04'00'

Member, Dr. Richard Prazenica

Yan Tang Digitally signed by Yan Tang
DN: cn=Yan Tang, o=Embry-Riddle Aeronautical University, ou=Mechanical Engineering, email=yan.tang@erau.edu, c=US
Date: 2022.04.23 22:15:05 -04'00'

Member, Dr. Yan Tang

Daewon Kim Digitally signed by Daewon Kim
Date: 2022.04.26 09:45:36 -04'00'

Graduate Program Coordinator,
Dr. Daewon Kim

4/26/2022

Date

Dean of the College of Engineering,
Dr. James W. Gregory

Date

Associate Provost of Academic Support,
Dr. Christopher Grant

Date

ACKNOWLEDGMENTS

This section will be used to thank everyone who has supported me and my professional development over the many years I've been in school. Every person mentioned here has contributed to my education and supported me throughout years of difficult circumstances in more ways than I can mention.

First and foremost, I'd like to thank my mother and my father for their continued supportive talks and help with my finances whenever I found myself in troubled times. They have helped me focus more on my studies throughout my undergraduate career until I could finally cover my own way through graduate school through fellowships and university support.

On that note, I would like to thank my advisor, Dr. Moncayo, for his continued support in my research. His encouragement brought me into the Doctorate program here at Embry-Riddle and he has helped me gain a better understanding of concepts I could never learn in a classroom. His generosity in the past few years has made my time in graduate school considerably more welcoming and motivating.

Ken Shishino, Chris Leirer, and John Wardell were my closest friends during these past few years and helped me overcome many personal obstacles in achieving the completion of this thesis. I would like to thank each of you as well as every friend I have met in my time here at Embry-Riddle.

To each of my fellow lab-mates at the Advanced Dynamics and Control Lab, thank you for always being willing to answer questions and for your support throughout each of my years in graduate school.

Last but not least, I would like to thank my girlfriend, Minna Wamhoff, for believing in me for the many years we have been together. She has been lovingly supportive of me every step of the way and has always been a welcome balance to my school life.

ABSTRACT

A pilot is a highly nonlinear and incredibly complex controller whose responses are difficult to predict. Many accidents have occurred from pilot error before or after failures and almost always after entering areas of the flight envelope considered as Loss-of-Control regimes. If a pilot's inputs to the flight control system can be predicted, then the introduction of dangerous flight conditions can be readily avoided. Avoidance could take the form of a warning indicator or augmentation of the pilot's inputs. The primary difficulty lies in how to actually predict how the pilot will perform in the future.

Methods to solve this problem are focused around the McRuer pilot model which simplifies the pilot response to a four-parameter equation that has been the focus of most recent solutions. Many recent attempts at solving this problem have found promising results in Wavelets, Most Likelihood Estimation, Extended Kalman Filters, and Unscented Kalman Filters. This thesis applies two new methods to the estimation problem and suggests a modification to one.

The three methods investigated in this thesis are a modified form of the Unscented Kalman Filter, Fourier Transform Regression with Time Domain derivatives, and Adaptive Neural Networks. The Unscented Kalman Filter holds merit in many estimation problems for its ability to handle model nonlinearities and noise in the systems and sensors. In this respect, it held the best solution for this work given that it could correctly estimate the parameters. However, the filter had to be finely tuned to reach a solution. The Fourier Transform Regression method could only handle time-invariant pilot model parameters due to its usage of batches of data. Once the parameters began varying with time, the solutions began having singularities. The adaptive neural networks showed promise being that they are stochastic estimators, but the solutions held within show they need more development to become a viable solution to this problem. It is recommended that deep reinforcement learning or combinations of these algorithms be applied to this estimation problem in the future to determine a more robust solution that can estimate the pilot's intent online.

TABLE OF CONTENTS

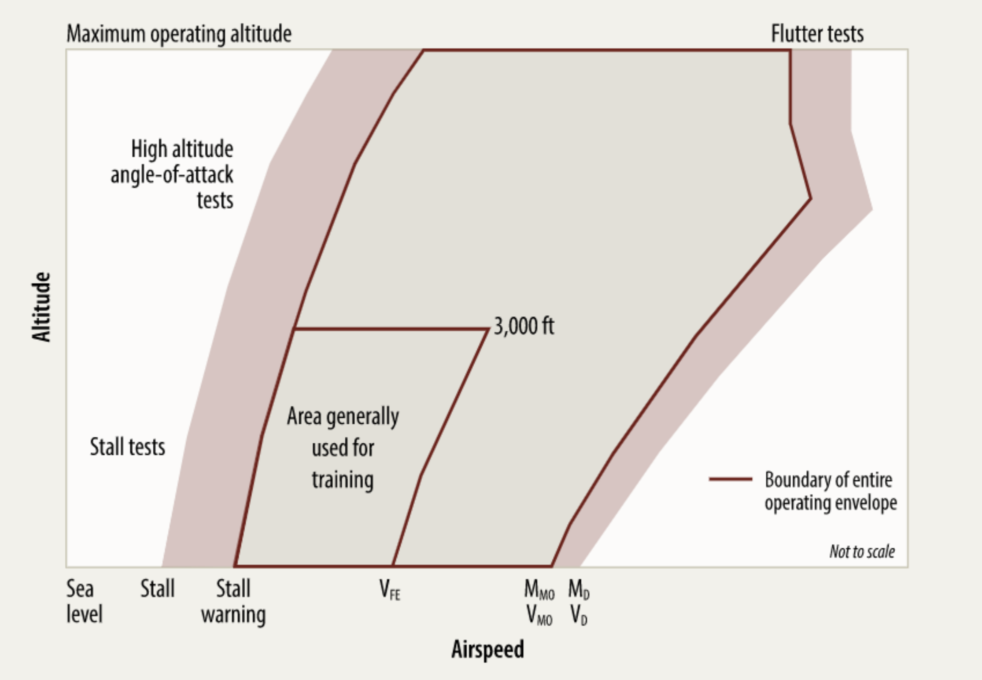
LIST OF FIGURES

LIST OF TABLES

Table **Page**

1 Introduction

One of the greatest causes of fatal aviation accidents remains the Loss-of-Control In-Flight (LOC-I). Many of these incidents tend to have root causes in errors performed by the pilot during the flight profile, though some can be a subsequent effect from other failures in the aircraft. The unexpected surprises in flight could include bird strikes, control problems, landing gear malfunctions, or engine failures. LOC-I is defined as an unintended departure from aircraft controlled flight, typically due to entering areas of the flight regime outside of the normal flight envelope [7]. This departure from safe flight can be readily seen in an Altitude-Velocity flight envelope, as shown in Figure 1.1. In almost every case, LOC-I will inevitably result in a stall or spin with an eventual loss of the aircraft. This will potentially result in the loss of the passengers and crew on board as well.



Maximum operating altitude
Flutter tests
High altitude angle-of-attack tests
Altitude
3,000 ft
Area generally used for training
Stall tests
Boundary of entire operating envelope
Not to scale
Sea level
Stall
Stall warning
V_{FE}
M_{MO} V_{MO}
M_D V_D
Airspeed

Figure 1.1 Altitude-Velocity Flight Envelope Showing Loss-of-Control Boundaries [1].

LOC-I is prevalent in all forms of aviation. However, General Aviation (GA) is much more accessible for the general public and therefore accounts for the higher portion of these incidents. In statistics from the National Transportation Safety Board (NTSB): between 2008 and 2014, 47 percent of fatal fixed-wing GA accidents in the United States involved pilots losing control of their aircraft in flight, resulting in 1,210 fatalities [8]. While these statistics have marginally improved in recent years, the more measures that can be taken to reduce this footprint on fatal aviation accidents, the safer the skies will be for society as a whole.

Recent studies have been performed which have classified regions of the flight envelope as controllable or not controllable. Others have implemented methods to inform the pilots when particular control actions will lead to a LOC-I regime. However, each of these solutions relies on a reactive methodology whereas the aim of this research is being proactive in an attempt to predict what the pilot will do based on their mental state and activity throughout the flight. If the pilot's intent can be correctly identified, then the formerly mentioned algorithms can inform the pilot such that a LOC-I accident can be readily avoided.

The pilot's mental state, behavior, experience, and exhaustion are all factors in determining how a pilot reacts to different scenarios. This adds an immense amount of complexity in a simulation environment when attempting to capture these changes. This is amplified by the lack of an exact mathematical model for a pilot under non-ideal conditions. This could mean a pilot who is hyper, exhausted, inexperienced, or under some other inhibiting condition could provide inputs to the flight control system that were unforeseen in the design of the simulation.

If a model of the pilot can be accurately and precisely estimated online throughout the flight profile, it is possible that the response of the pilot can be predicted for a few seconds into the future. This could allow predictive warning algorithms to notify the pilot and avoid potentially dangerous situations. This could effectively eliminate avoidable cases of LOC-I that have remained prevalent throughout the aviation industry.

1.1 Thesis Objectives

In this phase of research, the primary goal is to estimate a mathematical model of a human pilot providing elevator inputs to a pitch system. If a pilot's model can be readily identified, then future research can work towards predicting the intent the pilot may have for future control commands. This thesis will address four technical goals. The first of which is the identification of a pilot model which effectively captures the complex response of a human pilot. The second is the modification of that mathematical model to be amenable to estimation techniques in time or frequency domain. The third is to identify and tailor estimation methods around the parameters of the chosen pilot model. The last is to measure the accuracy of the estimation methods through statistical and frequency response analyses. Once all of these goals are achieved, the pilot's intent for the Pilot-Vehicle System can be used to determine future characteristics of the system.

This research aims to serve as a basis for future deep learning algorithms to better predict the onset of LOC-I. It will also apply to later research of Pilot Induced Oscillations (PIOs) and their prevention. Efforts will also be made to estimate mathematical models of pilots under different mental conditions such that a database of such parameters can be made available for the use of simulation technologies in the future.

1.2 Thesis Outline

This thesis first begins by introducing the background of each of the methods utilized in Chapter 2. This includes Kalman filtering, frequency domain parameter identification, and neural networks. Then, the McRuer pilot model is introduced along with the various forms used between all of the estimation techniques in Chapter 3. Afterwards, each method discussed prior is shown tailored to the estimation problem in Chapter 4. Lastly, the results of the Unscented Kalman Filter, Fourier Transform Regression, and Neural Network methods are shown in Chapter 5. Finally, conclusions about the data are drawn and recommendations for future work are discussed in Chapter 6.

2 Background

Online and offline estimation methods have been attempted in the past 10 years. Some have been successful, but each has particular difficulties due to the many factors that need to be taken into account. This Chapter will detail the the relevant tested methods and their associated backgrounds, but the estimation algorithms used in this research are shown in Section 4. Previous attempts to solve this problem will also be presented here to serve as a comparison for the results in Section 5.

Four different methods have been applied to this estimation problem in the past; the first of which was attempted in 2011 by Zaal and Sweet [2]. They referenced offline estimation techniques and attempted online estimation using Wavelets and Maximum Likelihood Estimation (MLE). The relevant results of their work are shown in Figure 2.1.

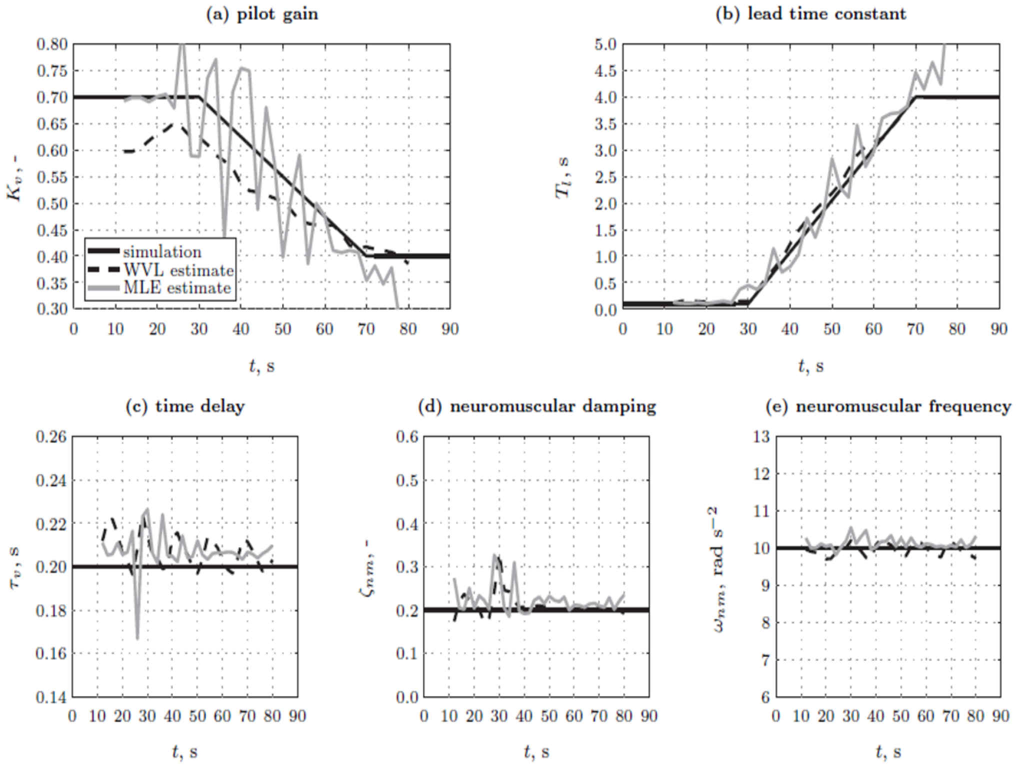


Figure 14: Estimated time-varying pilot model parameters from wavelet and MLE methods (P_n=0).

Figure 2.1 Results from Zaal and Sweet [2] showing Wavelets and Maximum Likelihood Estimation Techniques for Pilot Model Parameter Estimation.

Altogether, they found that wavelets were sensitive to measurement noise and could not converge to a correct solution while there was any amount of pilot remnant. This meant that the estimation could only work on a quasi-linear model without any remnant portion that accounts for the nonlinearity of the pilot's response. Wavelets are also computationally expensive and difficult to use in a real-time setting. This impedance is amplified by the fast-paced environment and possibility of quick maneuvers that aircraft tend to perform. MLE was found to give accurate results with less sensitivity to measurement noise and pilot remnant, but it could not detect rapid changes in the pilot's control behavior [2].

More recently, Mandal and a research group at West Virginia University performed similar testing using pilot data over the course of a few years between 2013 and 2016. Their work centered around using an Extended Kalman Filter and an Unscented Kalman Filter for the estimation. Due to the necessity of the Jacobian in an EKF, too many nonlinearities were uncaptured and the estimation suffered as a result [9]. Consequentially, they developed a UKF algorithm and showed that, under particular circumstances, the estimation tracks the true parameters well. This was under the condition that the measurement and process covariance matrices were finely tuned to the current problem. However, small changes in the matrices would change the results drastically. In this way, the UKF acted similar to the wavelets method in that there was high sensitivity to the noise in the system [10]. On that note, Mandal's effort inspired much of what is discussed in this thesis. The goal became developing a method which could capture changes in the pilot's mathematical model parameters without the sensitivity seen by the other methods mentioned.

2.1 Kalman Filters

Kalman Filtering techniques have historically been used to estimate parameters which cannot be measured otherwise. In aircraft systems, Kalman filters can be utilized for measurement noise reduction from the multitude of onboard sensors such as flight data probes and inertial measurement units. They can also be used for the estimation of Euler angles from angular rates, Euler angles from GPS Position, or various other combinations of aircraft states. These characteristics and uses allow Kalman Filters to act as a versatile solution for analytical problems in Pilot-Vehicle System (PVS) modeling.

Kalman filters rely on two primary steps in order to perform estimation. The first is a dynamics propagation step which creates an a-priori prediction of the current state and the second is a correction which adjusts the estimate according to the error in the measurement. There are three primary forms of Kalman filters that are typically used throughout aircraft control system design. The first and simplest is the Linear Kalman Filter (LKF), which relies on the simplicity of linear dynamics and output equations in order to perform estimation about an equilibrium point. The second and hybrid version is the Extended Kalman Filter (EKF), which uses the nonlinear model, but must linearize it at each time step. The last consists of Sigma-Point Kalman Filters (SPKF), which propagate a set of generated points through the nonlinear system dynamics. Specifically for this study, the SPKF known as an Unscented Kalman Filter (UKF) will be primarily used. UKF utilizes statistical guesses of variations in the state estimates through an Unscented Transformation (UT) to create a set of sigma points, each of which is propagated through the system dynamics and averaged before being corrected. Each method is described in detail in the following sections.

It is important to note that LKF, or Linear Quadratic Estimation (LQE), is the only method which finds an optimal solution. The EKF can potentially find an optimal solution, but that requires that the nonlinear system begins behaving mostly linearly. The UKF instead utilizes the nonlinear equations directly and typically does not converge to an optimal solution. However, it does more accurately converge to the system's true mean and covariance.

To show the capability of Kalman Filters with noise reduction and estimation, the simple mass-spring-damper problem with a noisy sensor can be considered. This system is shown in Figure 2.2 with an onboard accelerometer providing the noisy measurement for the filter ($\ddot{x}(t)$). The noise shown has variance of $0.005m/s^2$ and zero mean. The mass spring damper equation of motion and the respective state space are shown in Equations 2.1 and 2.2.

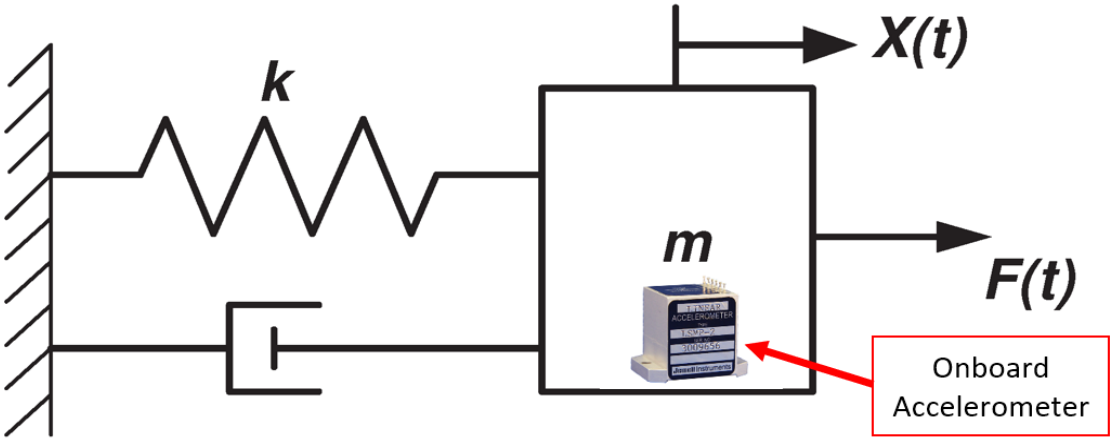


Figure 2.2 Mass-Spring-Damper used in Kalman Filter Estimation Example [3].

$$m\ddot{x} + c\dot{x} + kx = \tau \tag{2.1}$$

$$\dot{\vec{x}} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tau \tag{2.2}$$

The top graph in Figure 2.3 depicts the noise reduction capability of the LKF through the application to the linear accelerometer sensor data. Through this, the ability of the Kalman filter to correct data that has substantial noise can be seen. An extension of this is shown in the bottom graph of Figure 2.3 as the estimation of the position and velocity states using the sensor measurements. This depicts the capability of the filter to predict states that are unknown to the PVS from related sensor measurements while also canceling out a substantial amount of the noise. Altogether, the Kalman filters are an extremely useful tool that serve as a basis for the work performed in this study. The solution could have been further tuned by tweaking the Q-covariance matrix or better matching the R-covariance matrix to the sensor noise. However, this exemplifies the point that these filters must be manually tuned until they perform as intended. In some cases, the algorithm can be extremely sensitive to these adjustments.

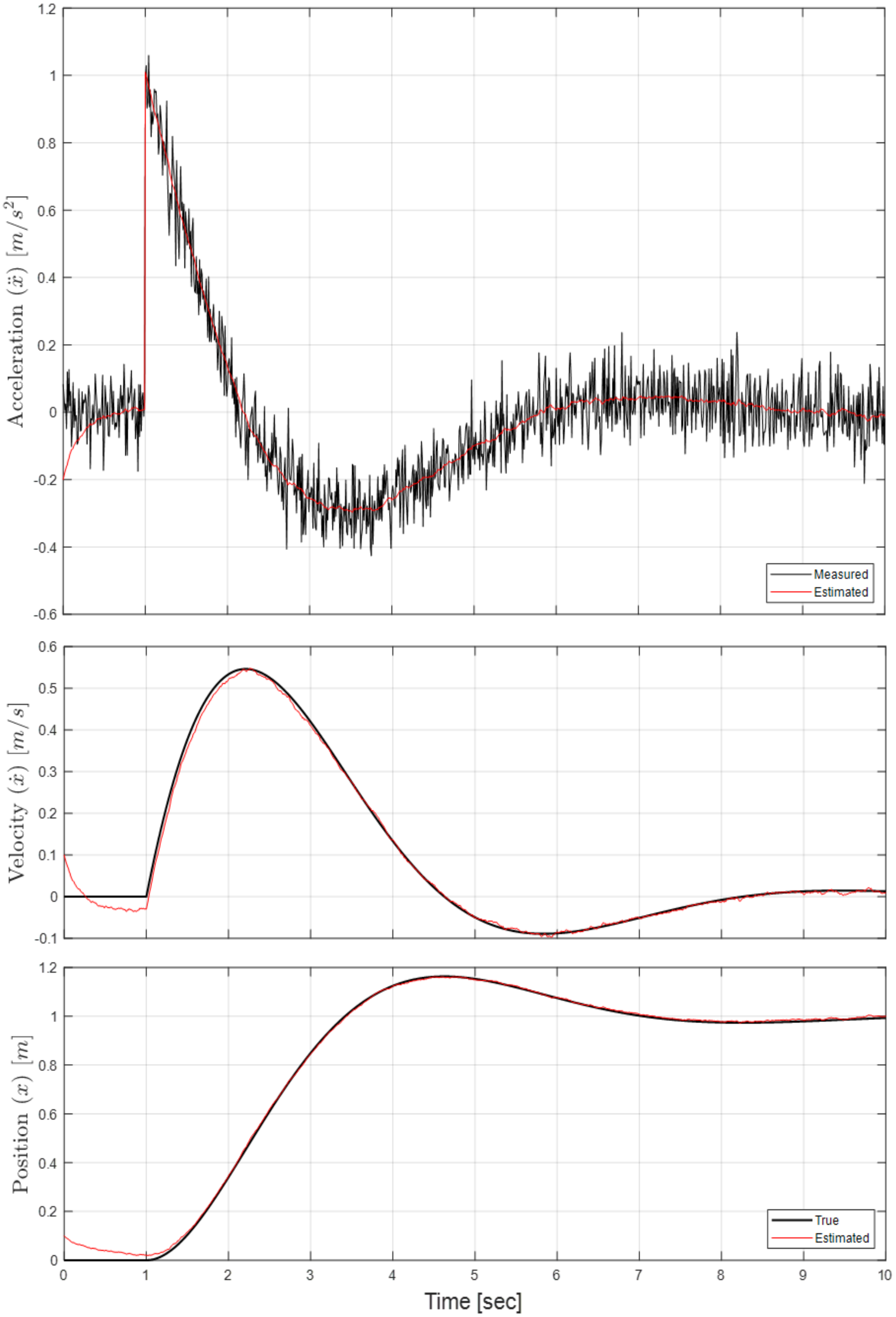


Figure 2.3 Kalman Filter Estimation example of a mass-spring-damper second order system. Results show noise reduction and estimation capabilities.

2.1.1 Linear and Extended Kalman Filters

As stated before, the LKF relies on the existence of a linear model for the dynamical system. The linear model is described by the dynamics matrix, A, and the measurement equation, C, shown below as Jacobians of the nonlinear equations evaluated at the equilibrium point.

$$A = \left.\frac{\partial \vec{f}}{\partial \vec{x}}\right|_{\vec{x}_e} \qquad C = \left.\frac{\partial \vec{h}}{\partial \vec{x}}\right|_{\vec{x}_e} \tag{2.3}$$

Noise assumptions are made which will serve as predictions and tuning parameters for the performance of the filter. The Q-matrix specifies the process noise. This could include vibration or wind, which could affect the states. The R-matrix signifies the measurement noise. This typically consists of the covariances of the sensor measurements and is not typically adjusted. Finally, the P_0-matrix is an initial covariance assumption which signifies the confidence in the initial condition of the filter. High covariance in the P matrix shows low confidence in the estimator initial conditions. These assumptions are shown in the equations below:

$$Q_k = E[w_k w_k^T] \qquad R_k = E[v_k v_k^T] \tag{2.4}$$

$$w_k \sim N(0, Q_k) \qquad v_k \sim N(0, R_k) \tag{2.5}$$

$$P_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] \tag{2.6}$$

where w and v are the process and measurement noise, respectively. E represents the expected value and N(α,β) represents a normal distribution with mean α and covariance β. Once all of these assumptions have been developed, the iterative portion of the algorithm begins. The first steps involve the calculation of the a-priori estimate and covariance. This is shown in Equations 2.7 and 2.8 as the prediction step.

$$\hat{\vec{x}}_{k|k+1} = A\hat{\vec{x}}_k + B\vec{u}_k \tag{2.7}$$

$$P_{k|k+1} = AP_kA^T + Q_k \tag{2.8}$$

Then, the Kalman gain, state estimation, and covariance matrix are all updated in the correction step as shown in Equations 2.9 through 2.11.

$$K_k = P_{k|k+1}C_k^T(C_kP_{k|k+1}C_k^T + R_k)^{-1} \tag{2.9}$$

$$\hat{\vec{x}}_k = \hat{\vec{x}}_{k|k+1} + K_k(\vec{y}_k - \hat{\vec{y}}_k) \tag{2.10}$$

$$P_k = [I - K_kC_k]P_{k|k+1} \tag{2.11}$$

These steps can then be iterated at each time step. Ideally, the covariance matrix P will reach a steady state value and the estimated states should begin to match the true values with little error and reduced noise. Should the solution not converge, then the equations may be too highly nonlinear for a linearized form to estimate well. An EKF can solve some of these problems.

The Extended Kalman Filter EKF has a similar form to the LKF, with a slight but significant change. The Jacobian is evaluated at each time step instead of at an equilibrium point. This causes Equation 2.4 to become 2.12.

$$A_k = \left.\frac{\partial \vec{f}}{\partial \vec{x}}\right|_{\vec{x}_k} \qquad C_k = \left.\frac{\partial \vec{h}}{\partial \vec{x}}\right|_{\vec{x}_k} \tag{2.12}$$

The prediction and correction steps have the same form as the LKF model. However, this Jacobian being calculated at each time step can vastly increase the computational power needed and can slow down the estimation technique considerably. This problem is only amplified when one considers the usage of EKF on higher order systems which could have an even greater number of states and an exponentially growing number of partial derivatives which need to be calculated. The EKF can handle small nonlinearities, but it cannot completely capture all nonlinear remnants.

2.1.2 Unscented Kalman Filter

The Unscented Kalman Filter takes a different approach to the Kalman estimation problem. The prediction step of a normal Kalman Filter is split into a state prediction step and an observation step. In the prediction, the estimation relies on the introduction of a series of sigma points which will each be propagated through the system dynamics and measurement function equations. This is handled by the Unscented Transformation (UT). The UT has three parameters which determine the selection of the sigma points for the statistically based estimation. α and κ determine the spread of the sigma points ($10^{-4} < \alpha < 1$, $\kappa \approx 1$). Then, β provides information about the statistical distribution (typically β=2 for Gaussian). These feed into the sigma point creation and weighting equations shown below.

$$\lambda = \alpha^2(L+\kappa) - L \tag{2.13}$$

$$\eta_0^n = \frac{\lambda}{L+\lambda} \tag{2.14}$$

$$\eta_0^c = \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta \tag{2.15}$$

$$\eta_i^n = \eta_i^c = \frac{1}{2(L+\lambda)}; i = 1, ..., 2L \tag{2.16}$$

where L is the length of the state vector, η is a vector of weights, and λ is a scaling factor. From these parameters and the covariance matrix, P_x, the sigma points are generated using Equation 2.17.

$$\chi_k = \begin{bmatrix} \bar{x}_k & \bar{x}_k + \sqrt{L+\lambda}\sqrt{P_{x_k}} & \bar{x}_k - \sqrt{L+\lambda}\sqrt{P_{x_k}} \end{bmatrix} \tag{2.17}$$

where $\sqrt{P_x}$ is the lower Cholesky decomposition of the covariance matrix such that $P_{x_k} = \sqrt{P_{x_k}}\sqrt{P_{x_k}}^T$. Each sigma point is then passed through the nonlinear system dynamics equations, weighted, and then summed. To complete the state prediction step, the state covariance matrix is then updated based on the new a-priori estimate.

$$\chi_{k|k+1}^i = f(\chi_k^i, u_k); i = 0, ..., 2L \tag{2.18}$$

$$\hat{x}_{k|k+1} = \sum_{i=0}^{2L} \eta_i^n \chi_{k|k+1}^i \tag{2.19}$$

$$P_{x_{k|k+1}} = Q_k + \sum_{i=0}^{2L} \eta_i^c (\chi_{k|k+1}^i - \hat{x}_{k|k+1})(\chi_{k|k+1}^i - \hat{x}_{k|k+1})^T \tag{2.20}$$

Once the estimate is updated, the sigma points must be propagated through the observation equations as well. Similar to the predictor equations, the output is calculated for each point, weighted, and then compared against the weighted average to update the covariance matrix. These steps are shown below.

$$\Psi_{k|k+1}^i = h(\chi_{k|k+1}^i, u_k); i = 0, ...2L \tag{2.21}$$

$$\hat{y}_{k|k+1} = \sum_{i=0}^{2L} \eta_i^n \Psi_{k|k+1}^i \tag{2.22}$$

$$P_k^{yy} = R_k + \sum_{i=0}^{2L} \eta_i^c (\Psi_{k|k+1}^i - \hat{y}_{k|k+1})(\Psi_{k|k+1}^i - \hat{y}_{k|k+1})^T \tag{2.23}$$

To finalize the observation step, the cross-covariance is calculated as follows:

$$P_k^{xy} = \sum_{i=0}^{2L} \eta_i^c (\chi_{k|k+1}^i - x_{k|k+1}^i)(\Psi_{k|k+1}^i - \hat{y}_{k|k+1})^T \tag{2.24}$$

Finally, the measurement correction can be applied using the Kalman gain similar to the LKF and EKF methods.

$$K_k = P_k^{xy}(P_k^{yy})^{-1} \tag{2.25}$$

$$\hat{x}_{k+1} = \hat{x}_{k|k+1} + K_k(y_k - \hat{y}_{k|k+1}) \tag{2.26}$$

$$P_{x_{k+1}} = P_{x_{k|k+1}} - K_k P_k^{yy} K_k^T \tag{2.27}$$

This equation is then used recursively throughout the data in order to estimate desired parameters. The version of this algorithm applied to the pilot model parameter estimation problem is presented in Section 5.

2.2 Parameter Identification in Frequency Domain

Parameter Identification (PID) serves as a linear method for estimating full state spaces (A, B, C, and D matrices) from the states and outputs of an unknown system. PID has seen uses in fault-tolerant control systems [11] in order to detect when a failure has occurred and to take proper action. Due to the linearization methods in Chapter 4, the method can be adapted to this estimation problem under certain circumstances. PID is beneficial as it is relatively simple to implement with no tuning parameters and noise is automatically filtered out; however it is limited to being used on a linear system, which means that nonlinear dynamics may not be fully captured.

The method of PID that this thesis focuses on is the Fourier Transform Regression with derivative in Time Domain (FTR-TD). This takes the basic form of the regression problem shown in Equation 2.28 at each sample batch of samples [12].

$$E\dot{z}(t) + Fz(t) = x^T(t)\hat{\Theta} \tag{2.28}$$

Applying a discrete Fourier transform to Equation 2.28 gives Equation 2.29. Written out for n number of points in a batch gives Equation 2.30. The solution to the regression equation then becomes Equation 2.31.

$$j\omega E\tilde{z}(\omega) + F\tilde{z}(\omega) = \tilde{x}^T(\omega)\hat{\Theta} \tag{2.29}$$

$$\begin{bmatrix} j\omega_1 E\tilde{z}(\omega_1) + F\tilde{z}(\omega_1) \\ \vdots \\ j\omega_n E\tilde{z}(\omega_n) + F\tilde{z}(\omega_n) \end{bmatrix} = \begin{bmatrix} \tilde{x}^T(\omega_1) \\ \vdots \\ \tilde{x}^T(\omega_n) \end{bmatrix} \hat{\Theta} \tag{2.30}$$

$$\hat{\Theta} = [Re(X^*X)]^{-1} Re(X^*Y) \tag{2.31}$$

X^* is the complex conjugate transpose. If the derivative terms, $E\tilde{z}(\omega_n)$, and non-derivative terms, $F\tilde{z}(\omega_n)$, are provided appropriately, the entire state space can be estimated. If $F\tilde{z}(\omega_n)$ is provided as the outputs concatenated to a set of zeros for each state, the C and D matrices can be estimated from this method as well.

2.3 Artificial Neural Networks

Artificial Neural Networks (ANNs) serve as open methods of estimating unknown parameters from known data. There does not need to be a dynamic equation linking the inputs to the outputs; rather the outputs' correlation to the inputs is identified and captured through the adjustment of weights between the hidden layer of the neural network and the input and output layers. The basic structure is shown in Figure 2.4.

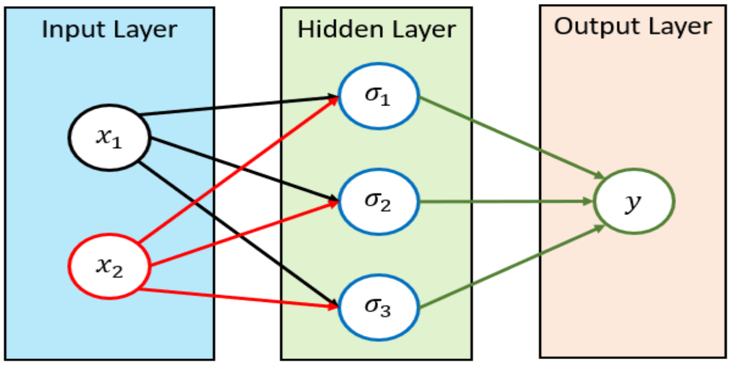


Figure 2.4 High-level Feedforward Neural Network architecture.

ANNs consist of three primary components: the input layer, output layer, and a variable number of hidden layers. The example in Figure 2.4 shows 2 inputs, $x_{1,2}$, 1 hidden layer, and 1 output, y. Typically, the neurons in the hidden layers contain activation functions such as Sigmoidal or Rectified Linear Unit (ReLU). These functions are shown in Figure 2.5.

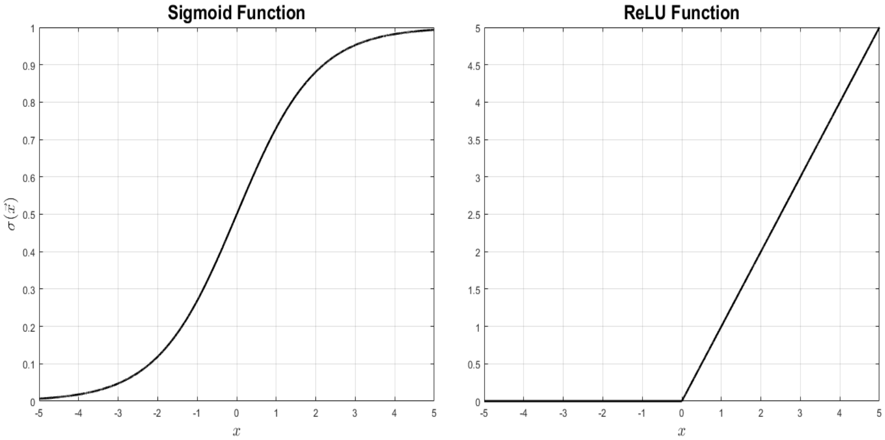


Figure 2.5 Graphical representation of typical activation functions used in hidden layer neurons of artificial neural networks.

As can be seen by Figure 2.5, the sigmoid function bounds the output between 0 and 1 no matter what range the input values have. This can be useful in probability or logistic predictions. The ReLU function is unbounded in the positive x direction, but provides zero for any negative input values. This could prove useful to effectively "turn off" certain neurons when the inputs are not necessary for its use, but that may affect the ability of the model to train properly. The rectification of negatives can be somewhat compensated for by using a "Leaky" ReLU function which has a positive slope in the negative input region. Some other options for the activation functions are hyperbolic tangent, lower-order polynomials, linear functions, or step functions. Each of these are shown in Figure 2.6. In any case, the activation function acts as a primary method of adjusting the performance of the NN to a multitude of problems. The summarizing equation of a NN is shown in Equation 2.32.

$$y(t) = V^T \vec{\sigma}(W^T \vec{x}) \tag{2.32}$$

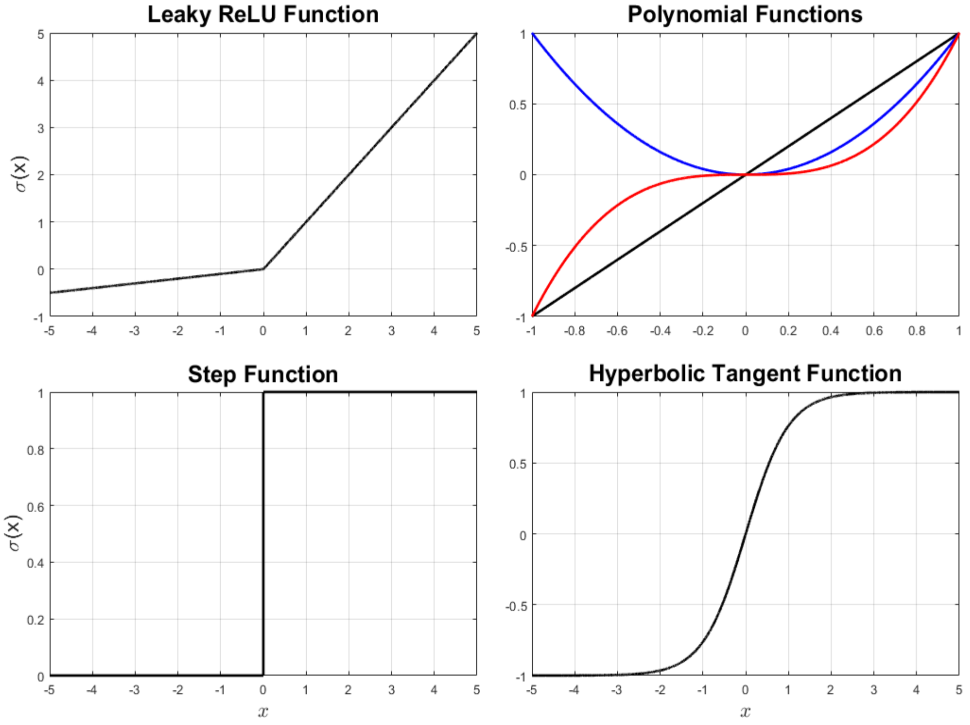


Figure 2.6 Graphical representation of other more uncommon activation functions used in hidden layer neurons of artificial neural networks.

Where W is the weighting matrix between the inputs and the hidden layer while V is the weighting matrix between the hidden layer and the outputs. Note that there could be multiple layers of neurons, but this research focuses on the implementation of a single layer. W and V are typically defined as shown in Equation 2.33 with m as the number of inputs, n as the hidden layer neuron count, and p as the number of outputs.

$$W = \left[\begin{array}{ccc} b_1 & \dots & b_n \\ \hdashline w_{1,1} & \dots & w_{1,n} \\ \vdots & \ddots & \vdots \\ w_{m,1} & \dots & w_{m,n} \end{array}\right] \epsilon \mathbb{R}^{(m+1)xn} \qquad V = \left[\begin{array}{ccc} d_1 & \dots & d_p \\ \hdashline v_{1,1} & \dots & v_{1,p} \\ \vdots & \ddots & \vdots \\ v_{n,1} & \dots & v_{n,p} \end{array}\right] \epsilon \mathbb{R}^{(n+1)xp} \tag{2.33}$$

The neurons and the output layer may have bias added in the form of b_i and d_i, respectively. These offsets can also be calculated and added into the matrix form of the neural network equation. In conventional uses of neural networks, the ANN is trained through a set of data with known output values. Training back-propagates changes to the weighting matrices W and V until a solution is reached that minimizes the given loss function. Gradient Descent (GD) is one of the most common training algorithms in feed-forward neural networks, but may require many iterations and will have slower convergence. Newton's method using the Hessian has also been utilized. A comparison of the convergence of the GD and Newton methods can be seen in Figures 2.7 and 2.8. Once the system is trained, the designer can test the outputs on a new known data set to gauge the performance of the ANN.

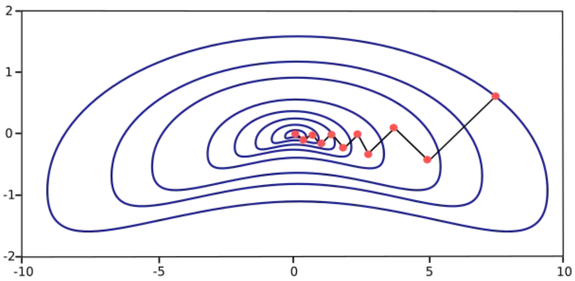


Figure 2.7 Gradient Descent Training method [4].

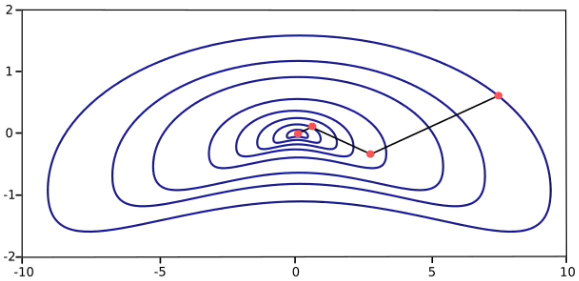


Figure 2.8 Newton's Hessian Training method [4].

3 Problem Formulation

This chapter details the format of the simulation problem along with detailed descriptions of the pilot models used. The simulation environment is based in Matlab/Simulink 2021a and utilizes a known pitch system with the McRuer pilot model acting as a controller. The estimation problem is in parallel to this PVS. The algorithms presented in this paper aim to estimate the parameters associated with this mathematical pilot model such that the intent of the pilot can be readily predicted.

3.1 Pilot Model(s)

McRuer and Krendel's paper provides the pilot model utilized in this study[13]. They provide two primary mathematical models which can be used for different levels of fidelity desired for the simulation environment. The first is the McRuer Precision Pilot Model which takes into account the neuromuscular properties of a pilot. The second is a simplified version which narrows the scope of the problem to four parameters. This study utilizes a simple 3rd-order pitch system to act as the plant while the pilot model acts as the controller. The error provided to the model is the difference in the desired to the achieved pitch angle. The pilot model is then attempting to minimize this error by providing an elevator deflection to the pitch system. One important note is that the actuator dynamics are not simulated. Therefore, the pilot model's output would normally need to be translated to a true elevator deflection [13]. The McRuer precision model is shown in Equation 3.1.

$$P(s) = K_p e^{-\tau s}\left(\frac{T_{Lead}s+1}{T_{Lag}s+1}\right)\left(\frac{T_K s+1}{T'_K s+1}\right)\left\{\frac{1}{(T_{N_1}s+1)[(\frac{s}{\omega_N})^2+(\frac{2\zeta_N}{\omega_N})s+1]}\right\}E(s) \quad (3.1)$$

Where P is the pilot output, E is the error, K_p is the gain, τ is the time delay, T_{Lead} and T_{Lag} are the mid-frequency phase lead and lag constants, T_K and T'_K are the low-frequency lead and lag constants, T_{N_1} is the neuromuscular system lag constant, ζ_N is the neuromuscular system damping, and ω_N is the neuromuscular system natural frequency. Commonly used simplifications of the precision model involve disregard for very-high and very-low frequency capturing terms and reduction to four core terms: K_p, T_{Lead}, T_{Lag}, and τ. These four terms can effectively capture all of the major mid-frequency action elements [13]. This simplified model, which is of core use to this study, is shown in frequency domain as Equation 3.2 and in time domain as Equations 3.3 and 3.4.

$$P(s) = K_p \frac{T_{Lead}s + 1}{T_{Lag}s + 1} e^{-\tau s} \tag{3.2}$$

$$\dot{x}(t) = -\frac{1}{T_{Lag}} x(t) + e(t - \tau) \tag{3.3}$$

$$p(t) = -\frac{K_p}{T_{Lag}^2}(T_{Lag} - T_{Lead})x(t) - \frac{K_p T_{Lead}}{T_{Lag}} e(t - \tau) \tag{3.4}$$

Each model can be used for the pilot model parameter estimation problem, but the simplified model serves the purposes of this study well. A linearized form of this, which is compatible with some of the estimation techniques presented, is shown in Chapter 4. This model may not be able to perfectly match the output of a human pilot. For instance, while a mathematical model may see an error of 0.01 degrees in pitch angle, a pilot will almost never be able to see that level of detail. Also, since it is a quasi-linear system with a nonlinear remnant, some non-linearity or higher order behavior will be lost. However, this mathematical model is still capable of giving basic information about the behavioral properties of the human controller through the four parameters.

3.2 Pilot/Vehicle System

The pilot serves as the controller which attempts to reduce the error between the desired and the achieved pitch angle. Therefore, some assumptions must be made in order to prove the effectiveness of the methods discussed. The first assumption is that the pitch system has known dynamics. As a characteristic of the aircraft dynamics, it is a reasonable assumption that the pitch system is known and that a model can be readily developed. The second is that only certain measurable parameters should be available for the estimation techniques. This adds to the applicability to a real-world environment. A simulation which utilizes every variable that is available to it may work in testing, but cannot feasibly be implemented. The third is that the stability of the estimation methods is important for convergence but it does not compromise or interfere with the stability of the core Pilot-Vehicle System. This allows stability proofing to be unnecessary.

The PVS is shown in Figure 3.1. Note that the parameter estimation is in parallel as mentioned before. The techniques compared within this study vary in the inputs they require. The most ideal case would be to only utilize the $p(t)$ and $\theta(t)$ since they are both measurable or otherwise capable of being estimated. The pitch system shown is used in reference to the work done by Mandal [10]. The pitch system is a simple third order dynamic model as shown in Equations 3.5 and 3.6.

$$\begin{bmatrix} \dddot{X}_\theta \\ \ddot{X}_\theta \\ \dot{X}_\theta \end{bmatrix} = \begin{bmatrix} -3.59 & -22.50 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \ddot{X}_\theta \\ \dot{X}_\theta \\ X_\theta \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} p(t) \tag{3.5}$$

$$\theta(t) = \begin{bmatrix} 0 & -15.44 & -59.93 \end{bmatrix} \vec{X}_\theta \tag{3.6}$$

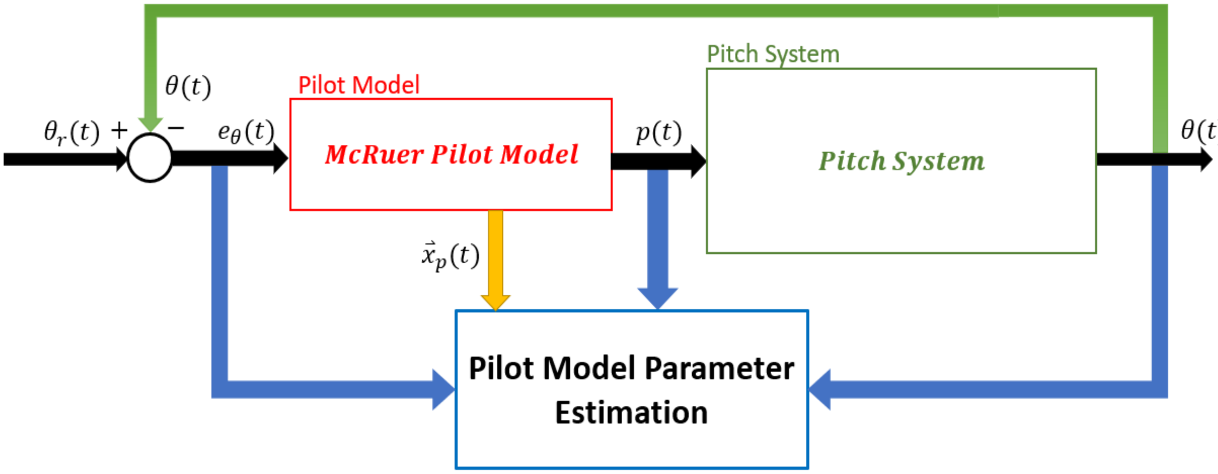


Figure 3.1 Simulation overview diagram. High level description of the major components of the simulation.

The PVS and results in Figure 3.2 show pitch tracking when given a desired pitch angle in radians. Note that the desired pitch angle was not scaled for a particular problem. The desired values are for testing and example. The corresponding pilot output is given in Figure 3.3. These examples show constant pilot model parameters; however, the simulation was made to be compatible with time-varying ones as well. This could be used to reproduce Pilot-Induced Oscillations or other non-optimal outputs from the pilot. An example is shown in Figure 3.4 which shows a chirp-signal resulting in PIOs after some time.

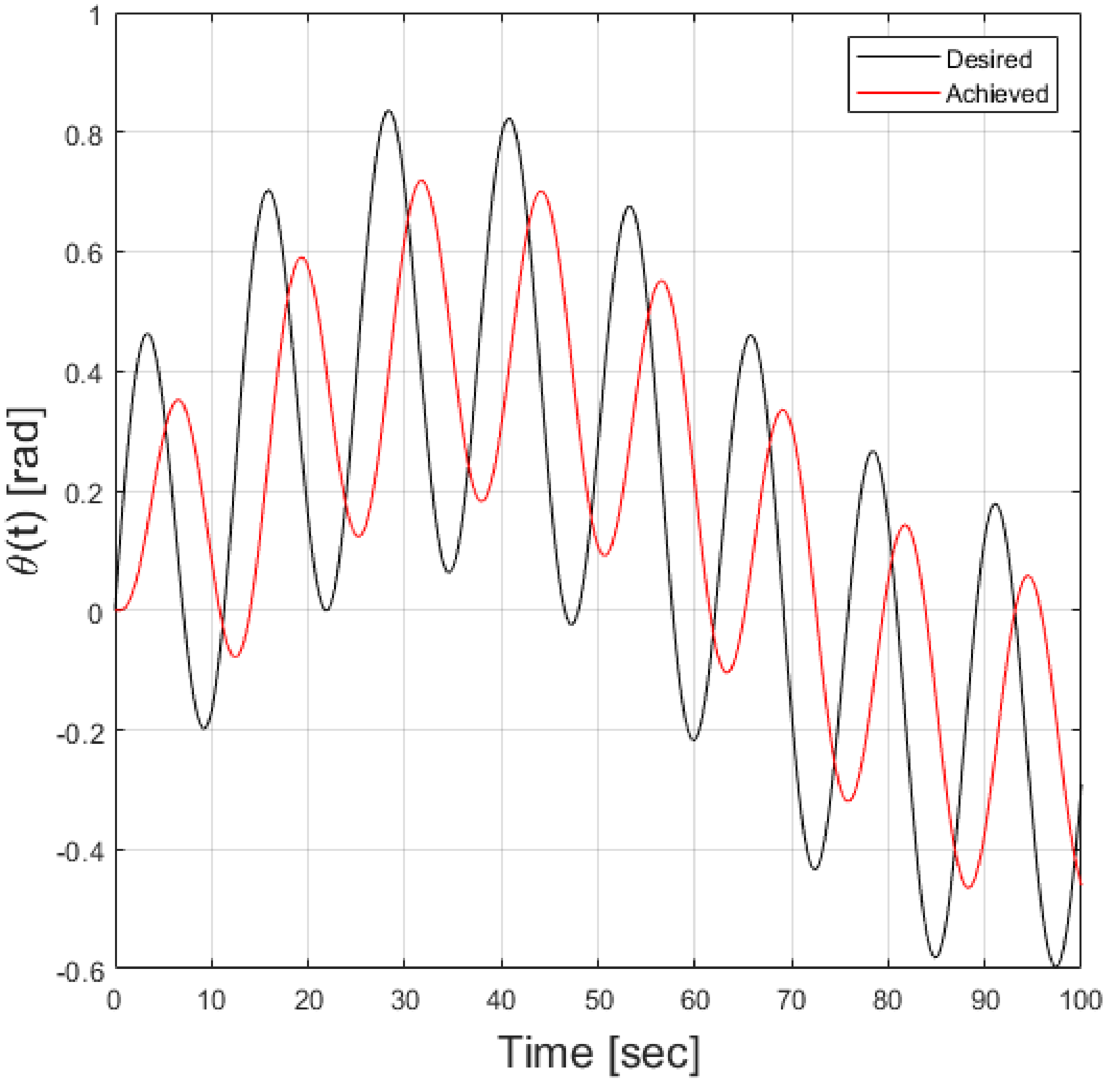


Figure 3.2 Pilot Vehicle System - Pitch System tracking from simulation environment.

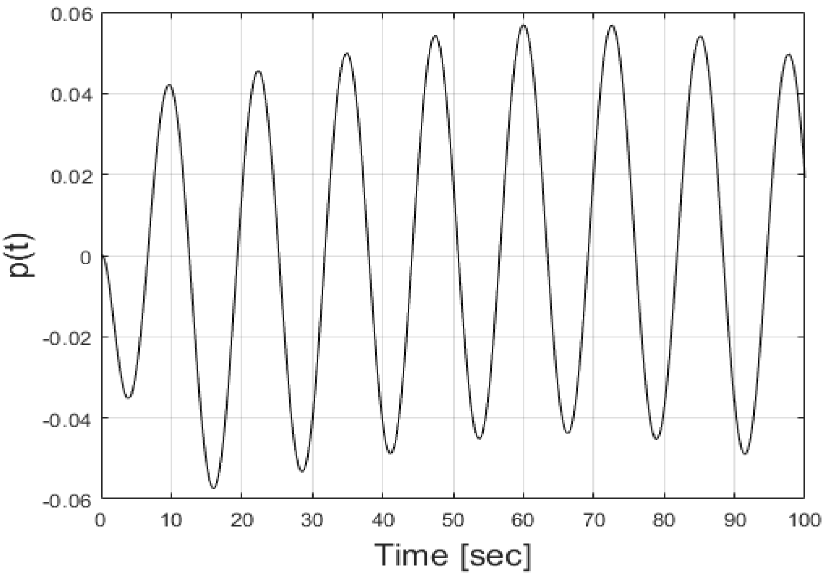


Figure 3.3 Pilot Vehicle System - Pilot controller output attempting to minimize pitch error in simulation environment.

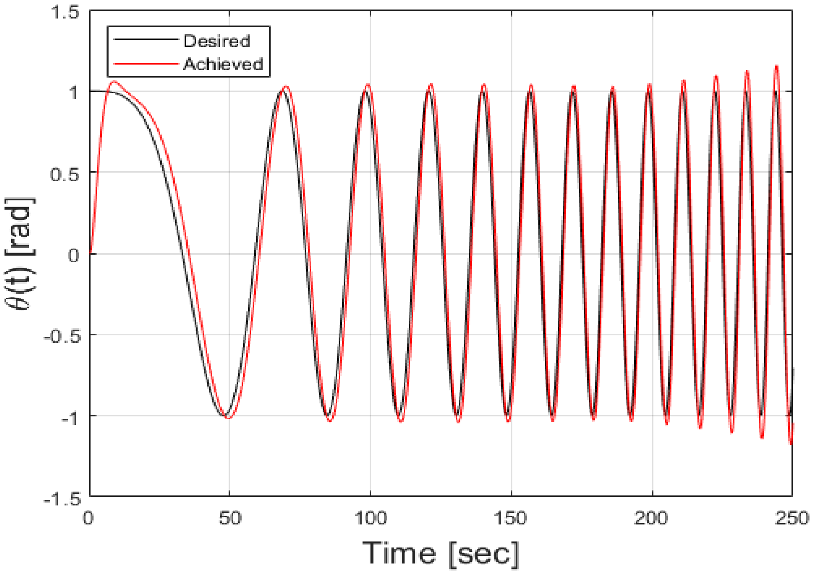


Figure 3.4 Pilot Vehicle System - slow onset of Pilot Induced Oscillations using the simulated pilot model.

Given that this mathematical model of the pilot can be tuned to match the pilot's response, the problem then becomes focused around estimating each of these parameters well enough such that the pilot's mental state and intent for future inputs can be predicted.

4 Pilot Model Estimation Methods

This chapter details the methodology and mathematical algorithms for the methods utilized in this research, all of which were identified as possible solutions for the parameter estimation of the McRuer human pilot model. Included in this section are the adaptation of Unscented Kalman Filters (UKF), frequency-domain Parameter Identification (PID), and Neural Networks (NN).

4.1 Linearization of the Pilot Model

To make the pilot model compatible with some forms of estimation presented here, a state space form was developed from a linearized version of the pilot model. One UKF method (UKFx10) and PID both rely on this form while the neural networks and the other UKF method (UKFx9) are able to attempt estimation of the parameters directly. The time delay in the pilot model equation is replaced using a first-order Pade approximation in order to facilitate reorganization into state space form. The Pade approximation and altered frequency domain pilot model are shown in Equations 4.1 and 4.2., respectively.

$$e^{-\tau s} = \frac{2-\tau s}{2+\tau s} \tag{4.1}$$

$$P(s) = \frac{-\tau T_{Lead} K_{pilot} s^2 + (2T_{Lead} - \tau) K_{pilot} s + 2K_{pilot}}{\tau T_{Lag} s^2 + (2T_{Lag} + \tau) s + 2} \tag{4.2}$$

From this version, a controllable canonical form can be developed as shown in Equations 4.3 and 4.4. Note that the C matrix has values due to the altered pilot model in Equation 4.2 having an equal number of poles and zeros.

$$\begin{bmatrix} \ddot{X}_p \\ \dot{X}_p \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix} \begin{bmatrix} \dot{X}_p \\ X_p \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{4.3}$$

$$p(t) = \begin{bmatrix} b_2 - b_0 a_2 & b_1 - b_0 a_1 \end{bmatrix} \begin{bmatrix} \dot{X}_p \\ X_p \end{bmatrix} + \begin{bmatrix} b_0 \end{bmatrix} e_\theta \tag{4.4}$$

The variables a_{1-2} and b_{0-2} serve as placeholders and are defined in Equations 4.5 and 4.6,respectively. These five parameters will form the estimated states for the UKF and PID methods discussed in the following sections.

$$a_1 = \frac{2T_{Lag} + \tau}{\tau T_{Lag}} \quad a_2 = \frac{2}{\tau T_{Lag}} \tag{4.5}$$

$$b_0 = \frac{-T_{Lead} K_{Pilot}}{T_{Lag}} \quad b_1 = \frac{(2T_{Lead} - \tau)K_{pilot}}{\tau T_{Lag}} \quad b_2 = \frac{2K_{pilot}}{\tau T_{Lag}} \tag{4.6}$$

Note that estimating these a's and b's will require solving a system of equations for the four model parameters which could add significant computation if performed at each time step. To account for this, the equations can be analytically solved to give the solution presented in Equations 4.7 and 4.8. These equations show that K_{pilot} can be solved first followed by T_{Lag}, then τ, and finally T_{Lead}.

$$K_{pilot} = \frac{b_2}{a_2} \qquad T_{Lag} = \frac{(a_1 + \frac{b_1}{K_{pilot}})}{a_2(1 - \frac{b_0}{K_{pilot}})} \tag{4.7}$$

$$\tau = \frac{2}{a_2 T_{Lag}} \qquad T_{Lead} = \frac{-2b_0}{a_2 \tau K_{pilot}} \tag{4.8}$$

4.2 Unscented Kalman Filter

The Unscented Kalman Filter presented in Section 2.1.2 can be adapted to fit the pilot model estimation problem. The first portion of this work is replicated from Mandal's dissertation [10]. To begin, the estimated variables are set up to include the 5 a's and b's found through the linearization of the pilot model into state space form. This altered state vector is shown in Equation 4.9.

$$\hat{\vec{X}} = \begin{bmatrix} a_1 & a_2 & b_0 & b_1 & b_2 & \hat{\vec{X}}_{pilot} & \hat{\vec{X}}_{pitch} \end{bmatrix}^T \tag{4.9}$$

The state matrix includes the pilot and pitch system states which are dependent on the 5 estimated a's and b's as shown in Equations 4.3 and 4.4. Given that the state vector contains a total of ten ($L = 10$) states, the parameters for the UKF can be set such that it uses a

Gaussian distribution ($\beta = 2$) with a "normal" amount of spread in the sigma points ($\alpha = 1$ and $\kappa = 0$). Recall the noise assumptions below in Equations 4.10 and 4.11.

$$Q_k = E[w_k w_k^T] \qquad R_k = E[v_k v_k^T] \tag{4.10}$$

$$P_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] \tag{4.11}$$

The Q and R matrices are set in Equation 4.12. The R-covariance matrix is set up such that the variance of the pilot output sensor and the pitch sensor are both equal. For both of these outputs, it is important to note that the sensors are theoretical and the results they provide may need to be estimated through other methods before being provided to the algorithms used here. The Q-covariance matrix is tuned in each case in order to improve the output results. This results in multiple solutions for Q to achieve acceptable results in the pilot model parameters.

$$Q = \begin{bmatrix} q_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & q_L^2 \end{bmatrix} \qquad R = \begin{bmatrix} 0.07^2 & 0 \\ 0 & 0.07^2 \end{bmatrix} \tag{4.12}$$

Then, the initial estimated state can be set using Equation 4.13. The sigma points are generated using Equation 2.17 and then propagated through the system dynamics using Equation 4.14. The pitch system is considered known and remains as a linear dynamic equation. The pilot model is also considered to take the state space form shown in Equations 4.3 and 4.4. The version shown here is in discrete form as is more intuitive to implement in the Matlab Simulink environment. $\vec{w}_k$ represents the matrix which generates the sigma points propagated through the prediction equation 4.14.

$$\hat{\vec{X}}_0 = \begin{bmatrix} \hat{a}_{1_0} & \hat{a}_{2_0} & \hat{b}_{0_0} & \hat{b}_{1_0} & \hat{b}_{2_0} & \hat{\vec{X}}_{pilot_0} & \hat{\vec{X}}_{pitch_0} \end{bmatrix}^T \tag{4.13}$$

$$\hat{\vec{X}}_{k+1} = \begin{bmatrix} \hat{a}_{1_{k+1}} \\ \hat{a}_{2_{k+1}} \\ \hat{b}_{0_{k+1}} \\ \hat{b}_{1_{k+1}} \\ \hat{b}_{2_{k+1}} \\ \hat{\vec{X}}_{pilot_{k+1}} \\ \hat{\vec{X}}_{pitch_{k+1}} \end{bmatrix} = \begin{bmatrix} \hat{a}_{1_k} \\ \hat{a}_{2_k} \\ \hat{b}_{0_k} \\ \hat{b}_{1_k} \\ \hat{b}_{2_k} \\ \hat{\vec{X}}_{pilot_k} + (\hat{A}_{pilot_k}\hat{\vec{X}}_{pilot_k} + B_{pilot}e_{\theta_k})dt \\ \hat{\vec{X}}_{pitch_k} + (A_{pitch}\hat{\vec{X}}_{pitch_k} + B_{pitch}\hat{p}_k)dt \end{bmatrix} + \vec{w}_k \tag{4.14}$$

Then, the measurement function can be set as Equation 4.15 assuming that the pilot output, $p(t)$, and pitch output, $\theta(t)$ are both being compared to the estimated measurements from the UKF. Since the A, C, and D pilot matrices in Equations 4.14 and 4.15 depend on the a and b parameters being estimated at each time step, they are shown as estimated matrices as well.

$$\hat{\vec{Y}} = \begin{bmatrix} \hat{p}_{k+1} \\ \hat{\theta}_{k+1} \end{bmatrix} = \begin{bmatrix} \hat{C}_{pilot_{k+1}}\hat{\vec{X}}_{pilot_{k+1}} + \hat{D}_{pilot_{k+1}}e_{\theta_{k+1}} \\ C_{pitch}\hat{\vec{X}}_{pitch_{k+1}} \end{bmatrix} \tag{4.15}$$

The correction step of the algorithm is the same as described in Chapter 2. The UKF algorithm will filter out noise in the output measurements, estimate all of the parameters given in the state vector, and attempt to minimize the error between the estimated and true output measurements. Should the noise assumptions in the Q-matrix be set too high, the outputs may be matched so well that the noise is not filtered out. Should they be too low, the solution may not converge at all. Due to this, the Q-matrix must be finely tuned to reach an acceptable solution in a particular case. Figure 4.1 shows the overview of the inputs and outputs of the UKF algorithm. The inputs are assumed to be known for the sake of the solution.

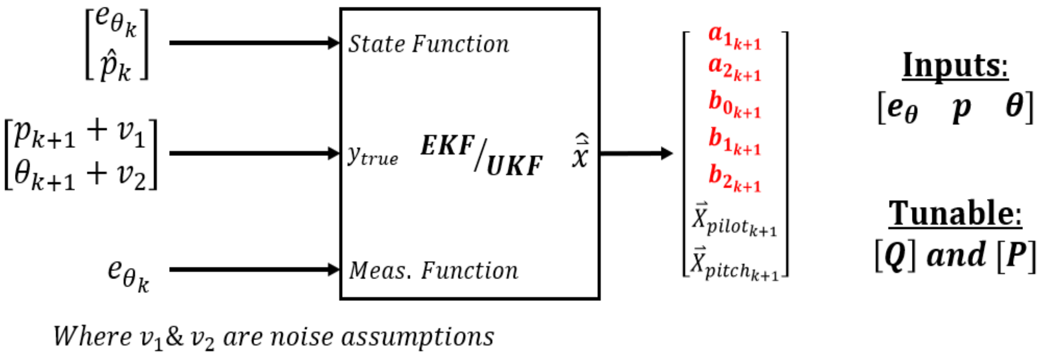


Figure 4.1 Unscented Kalman Filter diagram overview showing the inputs required and outputs given. This solution is tailored for the pilot model estimation problem.

As mentioned previously, the algorithm just described was recreated from the work of Mandal [10] and was found to have promising results even with the sensitivity of the process noise covariance matrix, Q. In the results section, this will be referred to as the primary UKF algorithm or UKFx10. In their work, the problem still persisted that the filter could not robustly predict the parameters for other cases beyond one Q was initially tuned for.

The estimated states can alternatively be modified to avoid the step of solving a system of equations by turning the state matrix defined in Equation 4.9 into Equation 4.16.

$$\hat{\vec{X}} = \begin{bmatrix} \hat{K}_p & \hat{T}_{Lead} & \hat{T}_{Lag} & \hat{\tau} & \hat{\vec{X}}_{pilot} & \hat{\vec{X}}_{pitch} \end{bmatrix}^T \tag{4.16}$$

This allows the nonlinear equations of the pilot model to be used directly and opens up the possibility for the precision model to be used in the future. This alteration changes Equations 4.14 and 4.15 into Equations 4.17 and 4.18. Performing direct estimation of the pilot model parameters cuts out the intermediate step of the system of equations and allows easier computation in flight control computers. This also eliminates the use of five equations to solve for four parameters and consequentially having multiple solutions for every time step. The new equations for the pilot states come from Equation 3.3 as well as the typical expression for discrete time state propagation. The measurement equation is taken from Equation 3.4.

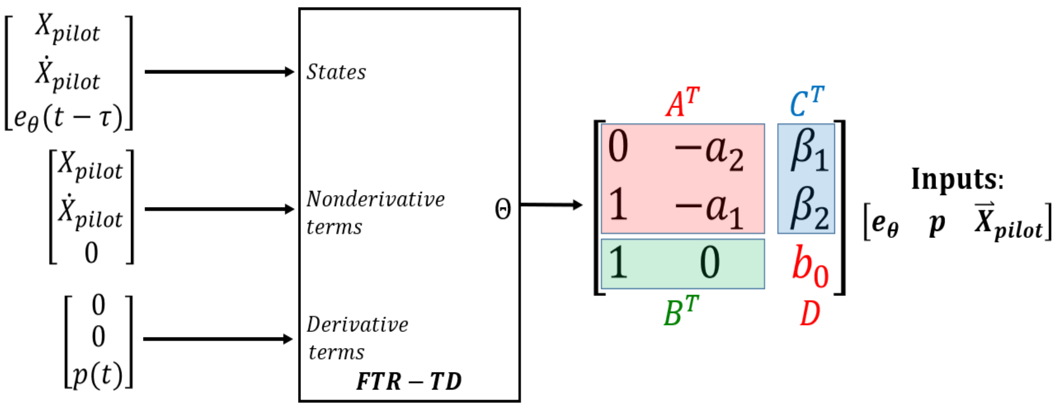
$$\hat{\vec{X}}_{k+1} = \begin{bmatrix} \hat{K}_{p_{k+1}} \\ \hat{T}_{Lead_{k+1}} \\ \hat{T}_{Lag_{k+1}} \\ \hat{\tau}_{k+1} \\ \hat{X}_{pilot_{k+1}} \\ \dot{\hat{X}}_{pilot_{k+1}} \\ \hat{\vec{X}}_{pitch_{k+1}} \end{bmatrix} = \begin{bmatrix} \hat{K}_{p_k} \\ \hat{T}_{Lead_k} \\ \hat{T}_{Lag_k} \\ \hat{\tau}_k \\ \hat{X}_{pilot_k} + (\dot{\hat{X}}_{pilot_k})dt \\ \frac{-\hat{X}_{pilot_k}}{T_{Lag_k}} + e_\theta(t - \tau_k) \\ \hat{\vec{X}}_{pitch_k} + (A_{pitch}\hat{\vec{X}}_{pitch_k} + B_{pitch}\hat{p}_k)dt \end{bmatrix} + \vec{w}_k \tag{4.17}$$

$$\hat{\vec{Y}} = \begin{bmatrix} \hat{p}_{k+1} \\ \hat{\theta}_{k+1} \end{bmatrix} = \begin{bmatrix} -\frac{K_{p_{k+1}}}{T^2_{Lag_{k+1}}}(T_{Lag_{k+1}} - T_{Lead_{k+1}})\hat{X}_{pilot_{k+1}} - \frac{K_{p_{k+1}}T_{Lead_{k+1}}}{T_{Lag_{k+1}}}e_\theta(t - \tau_{k+1}) \\ C_{pitch}\hat{\vec{X}}_{pitch_{k+1}} \end{bmatrix} \tag{4.18}$$

One drawback is the lack of estimation capability for the pure time delay, τ. Since the time delay is implicitly a part of the error input, the UKF does not converge to a solution since the delay does not internally affect the pilot and pitch outputs. However, the other three pilot model parameters can be readily estimated with similar limitations to the method presented by Mandal. This method is referred to as UKFx9 within the results section due to the nine estimated states.

4.3 Frequency Domain Parameter Identification

The parameter identification solution does not need to be modified as it is directly estimating the full state space model. The outputs take on a particular format given the order and number of inputs given to the algorithm. The PID solution takes on the general form given in Figure 4.2. This is the only method presented in this thesis which absolutely requires the system to be in state space form. As such, the outputs are given in such a form that each element can be related back to the a's and b's of the linearized pilot model in state space form from Equations 4.3 and 4.4. These can then be solved in a system of equations to give the four pilot model parameters like the primary UKF algorithm.



$\begin{bmatrix} X_{pilot} \\ \dot{X}_{pilot} \\ e_\theta(t-\tau) \end{bmatrix}$
$\begin{bmatrix} X_{pilot} \\ \dot{X}_{pilot} \\ 0 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 0 \\ p(t) \end{bmatrix}$
States
Nonderivative terms
Derivative terms
FTR – TD
Θ
A^T
C^T
$\begin{bmatrix} 0 & -a_2 & \beta_1 \\ 1 & -a_1 & \beta_2 \\ 1 & 0 & b_0 \end{bmatrix}$
B^T
D
Inputs:
$[e_\theta \quad p \quad \vec{X}_{pilot}]$

Figure 4.2 Parameter Identification overview of system schematic. Diagram shows the inputs and outputs required and the format of the output.

Yet again, as with the primary UKF algorithm, a_{1-2} and b_{0-2} can be solved for the pilot model parameters and runs into the same problem where multiple solutions can exist.

4.4 Pre-Trained Neural Networks

Due to the highly nonlinear nature of a human pilot's response, pre-trained stochastic estimation such as in the typical form of a neural network is unlikely to produce viable robust solutions. Preliminary results were obtained from a constant-weighting neural network to determine its capability to estimate the parameters. Figure 4.3 depicts the neural network results after being trained using a Levenberg-Marquardt technique with a Mean-Squared Error (MSE) performance index. Figure 4.4 depicts the same neural network run on a new data set with nonlinear pilot model parameter variation.

If the inverse case is considered where the neural network is trained on the more highly nonlinear pilot model parameters, then Figures 4.5 and 4.6 result. As shown in these four figures alone, the capability of the pre-trained neural networks in this estimation problem is limited. It does not matter whether the NN is trained on highly varying data which covers a range of possible inputs nor does it matter if the NN is trained on near constant data so it has better chance of convergence. The pilot as a controller is too complex of a system for static stochastic estimation. Therefore, adaptive solutions must be considered and become the focus of future interest for this research.

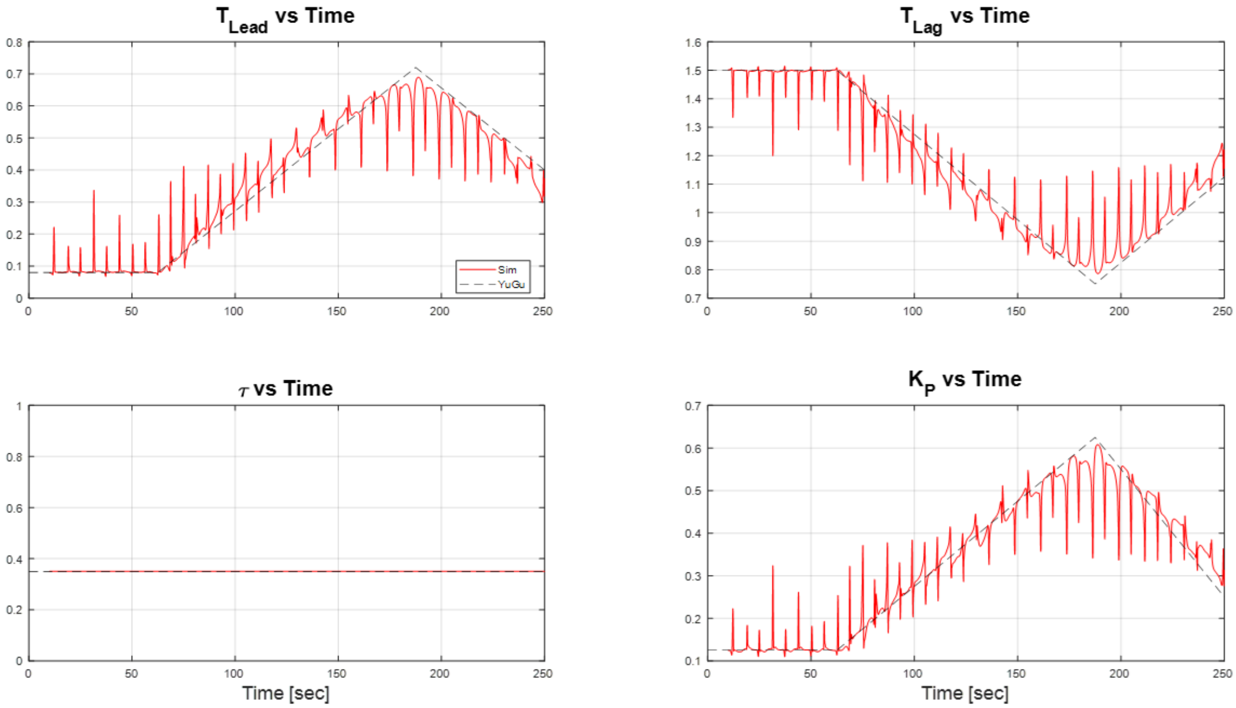


Figure 4.3 Neural Network Preliminary Results depicting the deployed neural network run on the near constant parameters it was trained on.

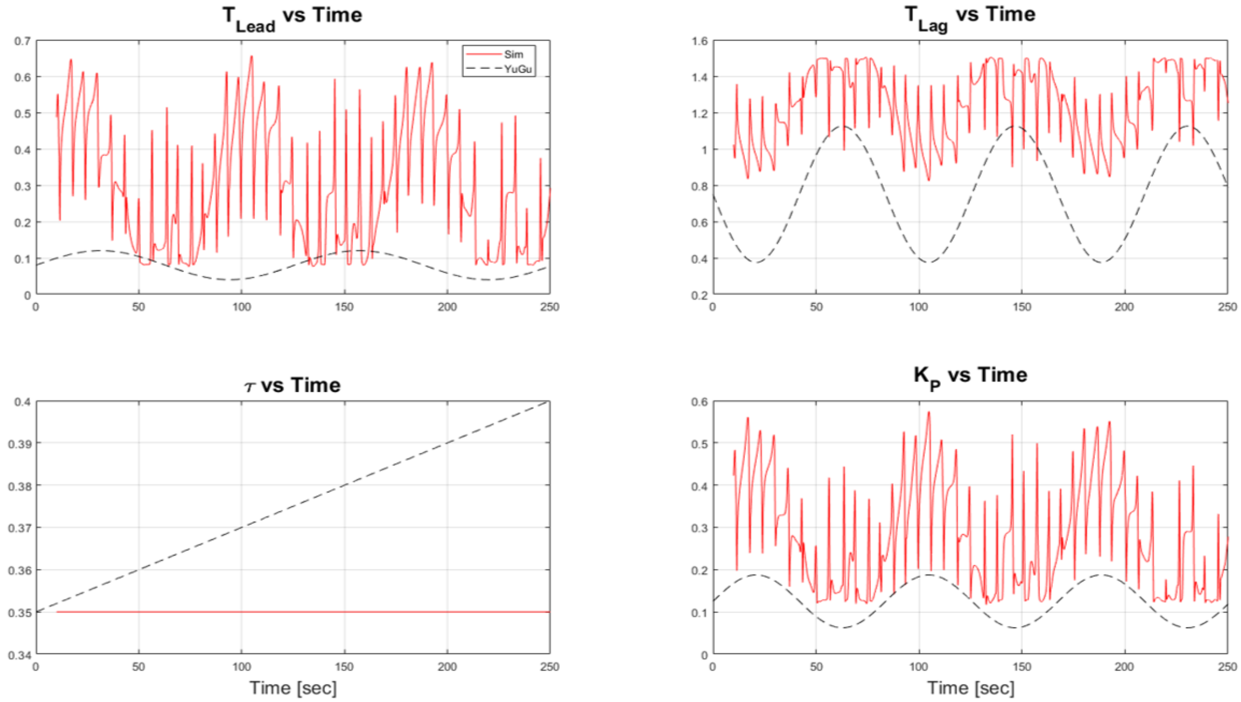


Figure 4.4 Neural Network Preliminary Results depicting the deployed neural network run over a more highly varying pilot model.

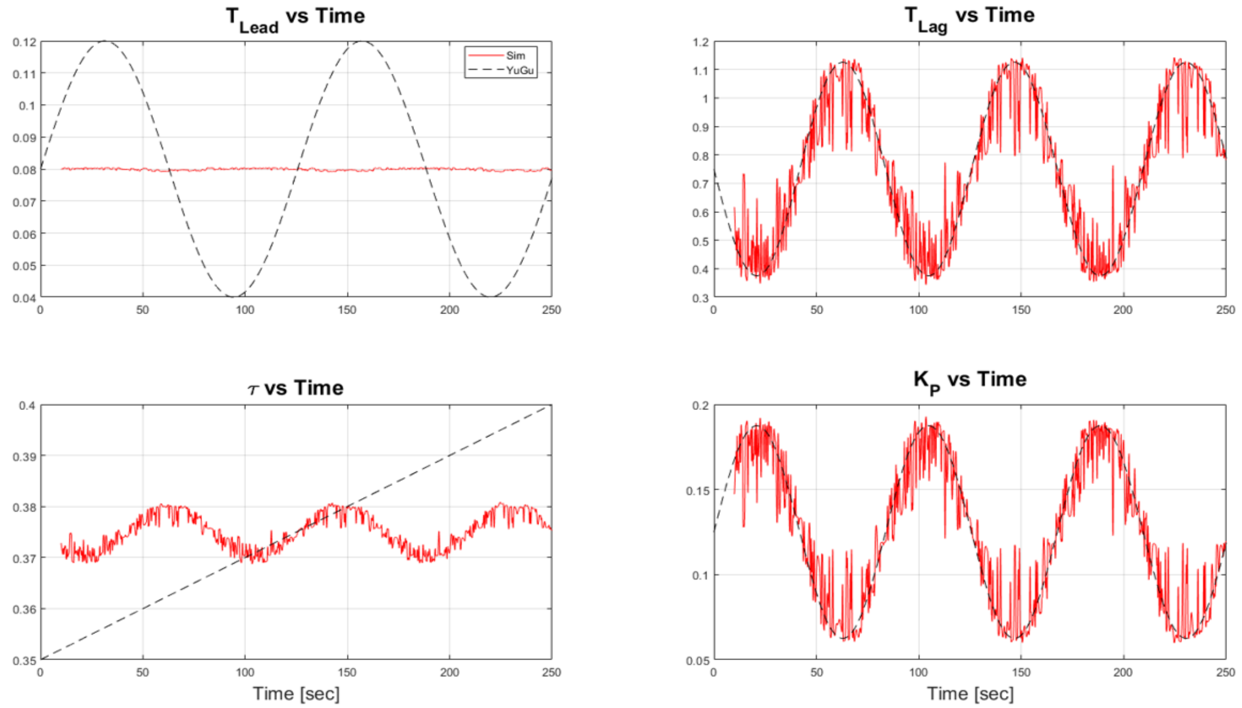


Figure 4.5 Neural Network Preliminary Results depicting the deployed neural network run on the highly varying parameters it was trained on.

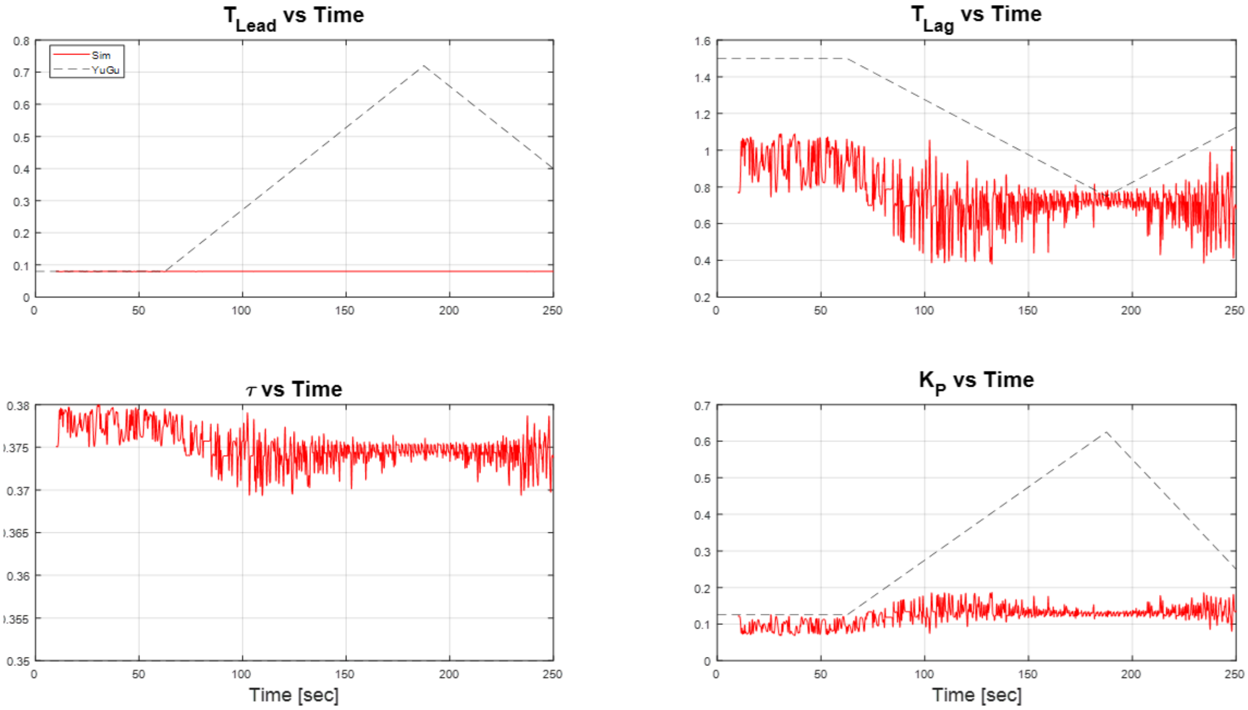


Figure 4.6 Neural Network Preliminary Results depicting the deployed neural network run over a linearly varying pilot model.

4.5 Adaptive Neural Networks

Given the limitations presented by the pre-trained neural networks, the next logical step would be to take advantage of their stochastic nature. By utilizing the training methods continuously throughout simulation, the weights can be persistently changed to accommodate new inputs or new pilot model parameter variations. For this reason, Adaptive Neural Networks (ANNs) have proven to be a useful resource for this estimation problem since they can stochastically predict seemingly unrelated parameters from known values. The primarily considered version, as is readily used in a number of different estimation problems is a Multilayer Perceptron ANN. The structure of the core neural network used in this application is given in Figure 4.7.

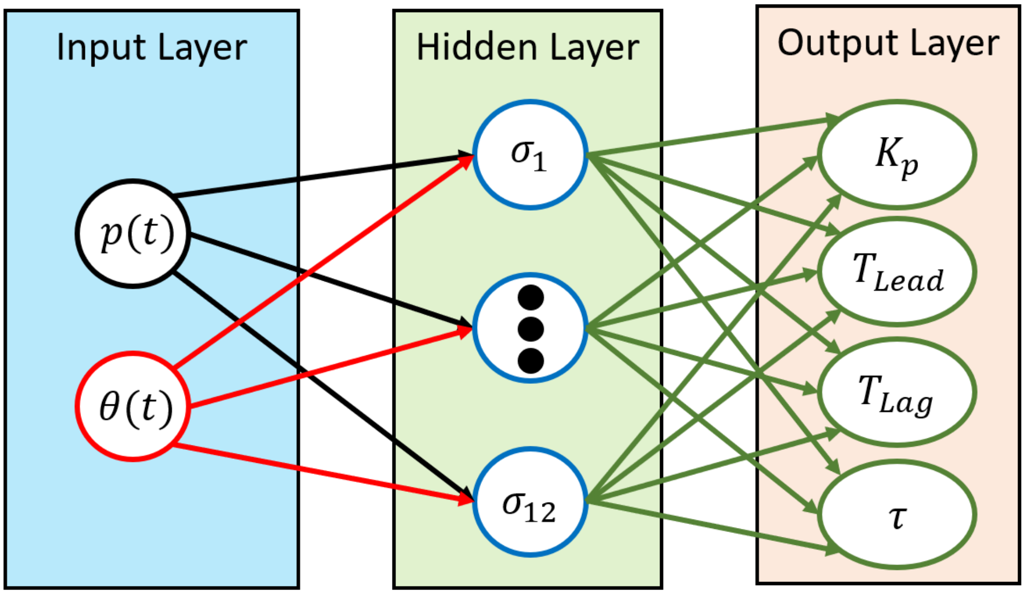


Figure 4.7 Neural Network format overview showing the inputs and output formats and general structure. This system has twelve neurons in a single hidden layer between two inputs and four outputs.

A neural network typically utilizes a set of training data to converge to input and output weights, W and V, which minimize a given loss function. The neural network is then deployed with these weights to determine the viability and robustness of its solution to a broad range of input data. The training methods used in this study are the same ones used in Giampiero Campa's Generalized Multilayer Perceptron (GMLP) ANN [14].

The Multilayer Perceptron (MLP) ANN utilizes back propagation for weight update laws. W and V are both dependent on one other such that each update as a system of linear first order ordinary differential equations as shown in equations 4.19 and 4.20.

$$\dot{\hat{W}} = -\Gamma_1[\hat{\vec{x}}\vec{e}^T\hat{V}^T\hat{\sigma}' + \lambda_1||\vec{e}||\hat{W}] \tag{4.19}$$

$$\dot{\hat{V}} = -\Gamma_2[(\hat{\sigma} - \hat{\sigma}'\hat{W}^T\hat{\vec{x}})\vec{e}^T + \lambda_2||\vec{e}||\hat{V}] \tag{4.20}$$

Where $\hat{W}$ and $\hat{V}$ are the input and output weighting matrices concatenated with the corresponding layer's biases, $\Gamma_{1,2}$ are the adaptive gains, $\vec{e}$ is the error in the NN outputs, and $\lambda_{1,2}$ are forgetting factors which counteract the uncontrolled parameter growth commonly found in adaptive control theory. Solving Equations 4.19 and 4.20 continuously, the input and output weighting matrices can be adaptively updated throughout the flight profile similar to the online form of the UKF presented before.

The neuron activation functions present within the GMLP network are Sigmoidal, though this could be modified for ReLU or hyperbolic tangent functions as mentioned in Chapter 2. This Sigmoidal activation function is especially useful as the pilot model parameters should only be positive values.

One primary setback of the provided tool is the restriction of having the same number of error inputs as there are outputs. The adaptive portion of the algorithm uses the errors of the parameters to update the weights, so the error in the pilot model parameters is not available. In this case, the system is set up as if it were a secondary or parallel dynamical system to the core PVS. This means that the output pilot model parameters are applied to a separate pilot model such that the pitch error taken from the PVS can be plugged in to get a resulting $p(t)$ output. The system is set up as shown in Figure 4.8. The error between the true pilot output and the neural network estimated pilot output is used as the loss function for updating all four output parameters. This is a major hindrance to the capability of the ANN as the resulting error trend is shown propagating in all four outputs and the parameters cannot update independently.

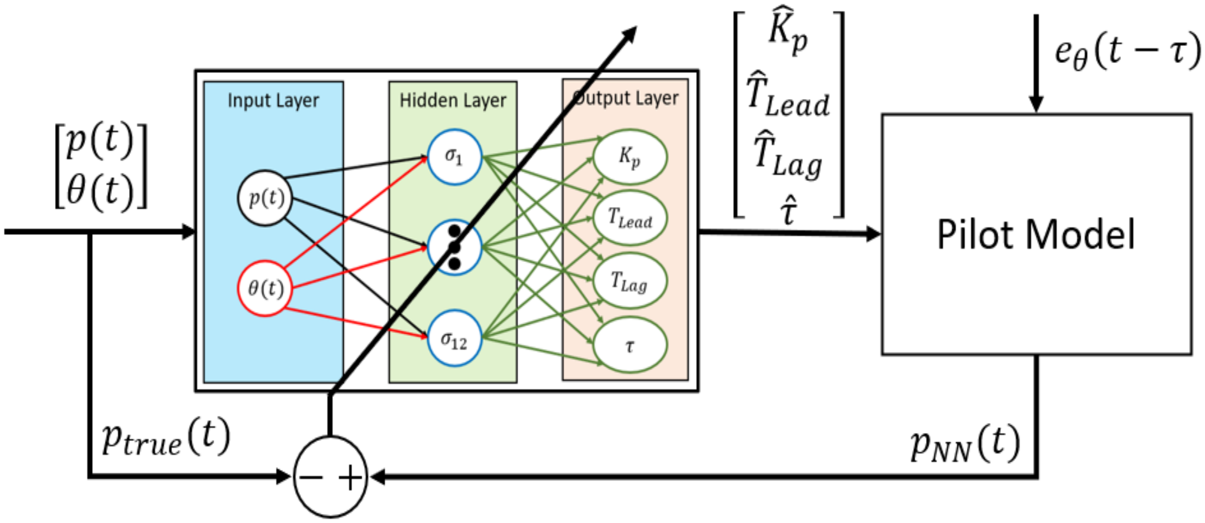


Figure 4.8 Adaptive Neural Network architecture with weights updated by the error in the pilot output. The estimated pilot model parameters are passed into a tertiary pilot model. The neural network attempts to minimize the error in the pilot output by adjusting the estimated pilot model parameters.

An adaptive neural network allows the possibility of the constantly changing dynamics of a highly nonlinear pilot to be estimated. A fixed neural network in this case has difficulty as there is not always a simple mathematical link between the parameters and the output that the model gives. This discontinuity is only amplified by the nature of a human pilot acting as the controller being estimated. Adaptive neural networks allow broader implementations such that a single network may be able to estimate a pilot's behavior throughout the flight profile.

5 Numerical Simulations

This section details the results of the algorithms shown in the previous chapters. The Unscented Kalman Filter, Parameter Identification, and Neural Networks were all implemented and the results are shown. The four input cases used to collect these results are presented first. The input variations consist of two different pitch commands and either time variant or time-invariant pilot model parameters. The results are organized such that each method is presented individually and a summary of the data is presented in the final section which compares the individual algorithm performances. The assumptions and tuning parameters modified for each case are also presented.

5.1 Input Cases

There are four input cases used in this study. They consist of combinations of two pitch command cases and two variations of pilot model parameters. The first pitch command is a high-frequency and high-amplitude sum of sines signal which aimed to meet the Persistence of Excitation requirement for the estimation to converge to the true values. The second pitch command is a pitch doublet which is much more normal for a pilot to perform within typical flight operation. The two pilot model cases are time-invariant and time-varying parameters.

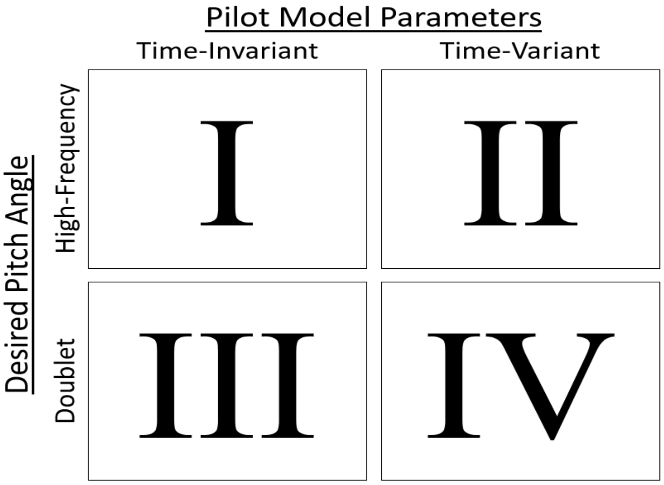


Figure 5.1 The format and description of the results cases. Both high frequency and doublet desired pitch signals are used to show the viability for convergence for the time-invariant and time-variant pilot model parameters

The four cases are chosen such that the robustness of each estimation technique can be displayed. Each demonstrates the potential extendability of their respective algorithms or lack thereof. The structure of the cases used for the results is shown in Figure 5.1. The results are presented in the results section under the labels for each case. The specific description of each variant of the parameters is described in Section 5.1.2 while the description of the desired pitch angles requested of the pilot model is shown in Section 5.1.1.

5.1.1 Pitch Commands

The first case utilizes a sum of sines input for the desired pitch angle and constant pilot model parameters. The desired pitch signal is shown in Figure 5.2. The signal is comprised of 3 sine wave signals with different frequencies and amplitudes. The information to rebuild the sine wave is shown in Table 5.1.

Table 5.1 Sine waves used in the creation of the high frequency pitch command for cases 1 and 2. The aim is high frequency and high amplitude to properly excite the system.

	Equation	A [rad]	ω [rad/s]
$\theta_{des,1} =$	$10\frac{\pi}{180}sin(0.25t)$	0.1745	0.25
$\theta_{des,2} =$	$5\frac{\pi}{180}sin(1t)$	0.0.0873	1.00
$\theta_{des,3} =$	$1\frac{\pi}{180}sin(0.125t)$	0.0.0175	0.125

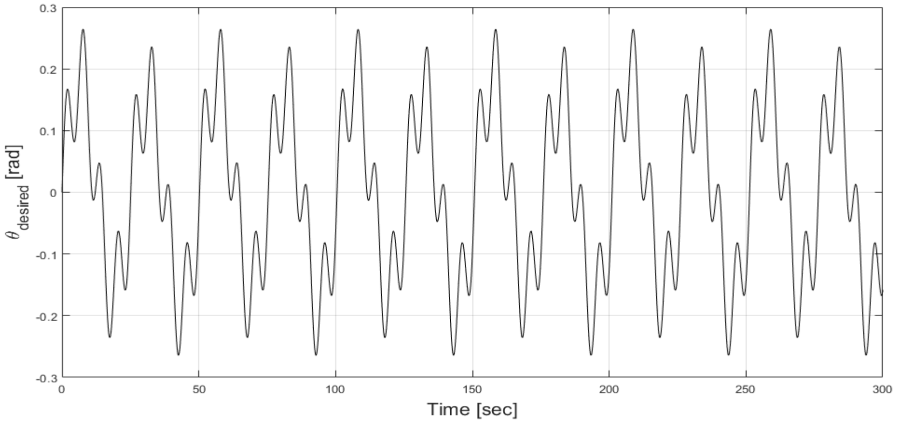


Figure 5.2 Depicts the high frequency pitch commanded in the Pilot Vehicle system. This case was used for proving the convergence of the algorithms presented and was primarily focused around achieving the Persistence of Excitation requirement for convergence.

The second pitch command is more reasonable for a pilot to perform in a typical flight profile. While it may not properly excite the system, the goal is to see how or if the estimation techniques converge to any solution or stray from their initial conditions given less extreme inputs. This second signal is a simple repeated doublet and is shown in Figure 5.3.

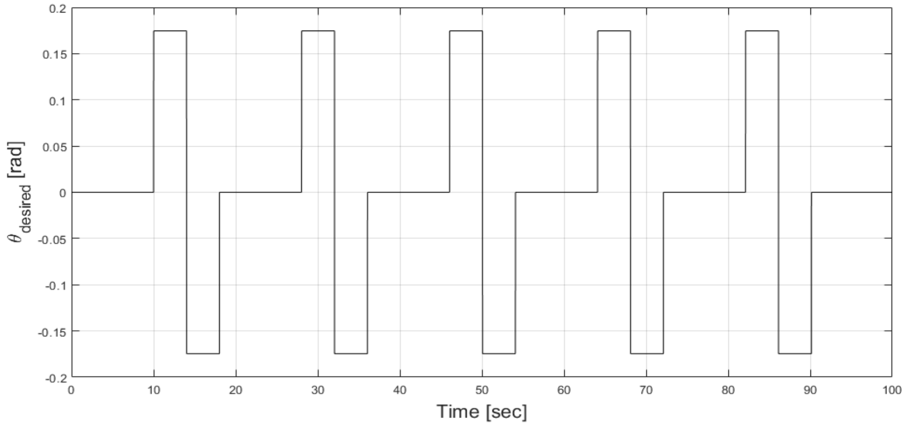


Figure 5.3 Lower frequency doublet commanded of the pilot model to simulate more realistic pilot commands that may not meet excitation requirements for convergence of the parameters. This input is used for cases 3 and 4.

5.1.2 Pilot Model Parameters

The two cases considered for the pilot model parameters are time-varying and time-invariant. Batch methods tend to have a more difficult time converging to a solution in time-varying cases as opposed to iterative methods. Therefore, the inclusion of both displays the strengths and weaknesses of all methods presented along with the two pitch input cases from Section 5.1.1. First, the time-invariant parameters are shown in Table 5.2. The pure time delay is the only dimensionalized value.

In the time-variant data, note that the parameters only change linearly whereas a true pilot would more likely have a much more gradual change in the values. As part of a typical Lead-Lag compensator, the Lag constant is kept higher than the Lead constant throughout the entire period of time for the sake of the input model. Figure 5.4 shows the time-varying pilot model parameters used for cases 2 and 4.

Table 5.2 Time-invariant pilot model parameters used to produce the results shown in cases 1 and 3. These values were derived from Monte Carlo simulations and retrieved from similar studies of the McRuer model [5] [6].

Parameter	Value
K_p	0.54
T_{Lead}	0.32
T_{Lag}	0.4
τ	0.25s

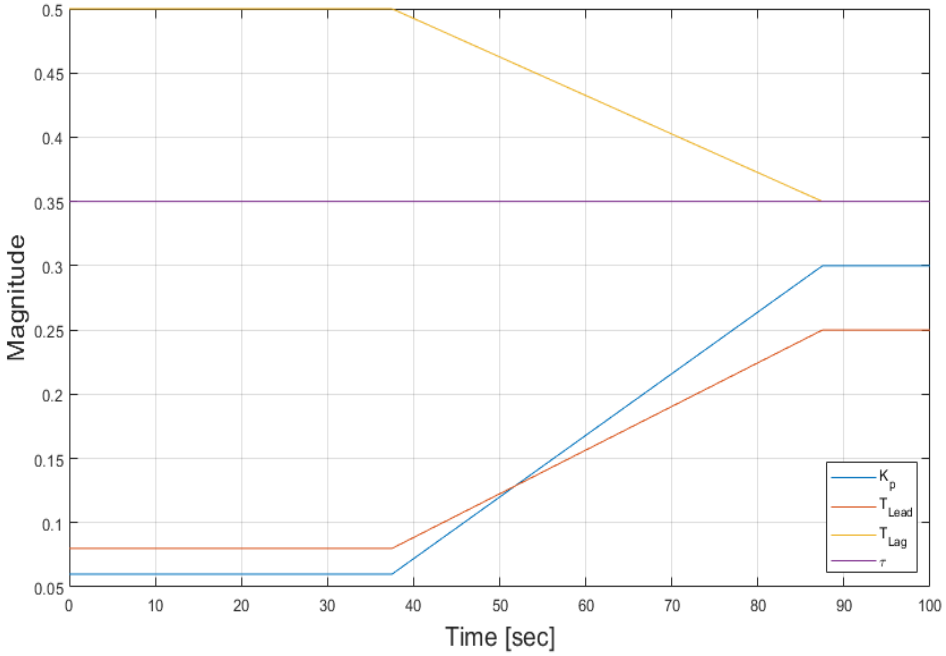


Figure 5.4 Time-Varying model parameters used in cases 2 and 4 of the results. The lead constant is always less than the lag constant as is typical of a lead-lag compensator. The results are linear but would more accurately depict a pilot if they changed more gradually.

5.2 Unscented Kalman Filter Results

Two Unscented Kalman Filter algorithms were used in this study. One UKF estimated the state space through a linearized form of the pilot model (UKFx10), while the other estimated the four pilot model parameters directly (UKFx9). Only the UKFx9 results are shown since UKFx10 can be found in Mandal's research[9][15][10] and because the method can result in singularities in some of the parameters. Refer to Section 4.2 for the details of the differences between these two methods.

5.2.1 Nine-State Unscented Kalman Filter

CASE 1: Case 1 uses a high frequency input signal with time-invariant pilot model parameters. The initial conditions and noise assumptions are shown in Equations 5.1-5.4. It is reasonable to assume that the pitch and pilot model states are zero initially as there is no desired pitch commanded before t=0. It would add fidelity to include aircraft trim conditions as the initial pitch angle. Note that, since the assumption is made that the initial conditions are not equal to the time invariant parameters, the P-covariance matrix is set as nonzero diagonal. This equates to relative confidence that the initial conditions are close to the true values but not exact.

$$X_0 = \begin{bmatrix} \underbrace{0.06}_{K_p} & \underbrace{0.08}_{T_{Lead}} & \underbrace{0.20}_{T_{Lag}} & \underbrace{0.35}_{\tau} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.1}$$

$$Q = \begin{bmatrix} 3x10^{-7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1x10^{-8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2x10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1x10^{-6} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1x10^{-8} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1x10^{-8} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1x10^{-10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1x10^{-10} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1x10^{-10} \end{bmatrix} \tag{5.2}$$

$$R = \begin{bmatrix} 0.0001 & 0 \\ 0 & 0.01 \end{bmatrix} \tag{5.3}$$

$$P_0 = 0.05\mathbb{I}_{9x9} \tag{5.4}$$

As can be seen by Equation 5.2, the Q-matrix has incredibly small covariances which are sensitive to small alterations. The resulting pilot model parameter tracking is shown in Figure 5.5.

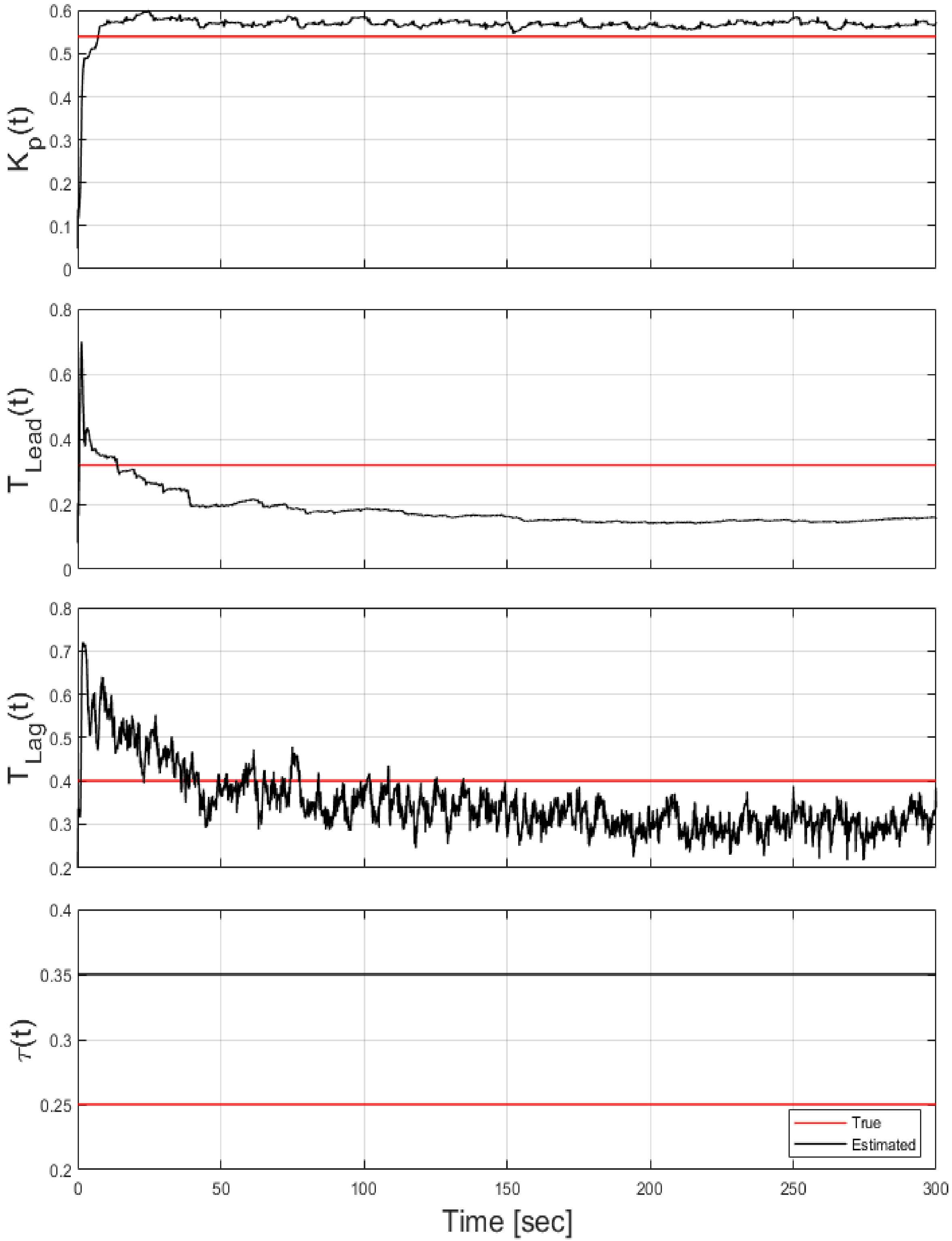


Figure 5.5 UKFx9 Case 1 Results - Pilot model parameter convergence.

An important note is that the time delay does not update and may be a shortcoming of this method. Since the delay is implicit in the input error signal, the UKF has difficulty updating it in this format. Due to the similarity in the pilot model parameters, it is safe to assume that the pilot output and, subsequently, the achieved pitch angle would match as well. These results are shown in Figures 5.6 and 5.7. The figures are zoomed in such that the estimation can be more easily seen.

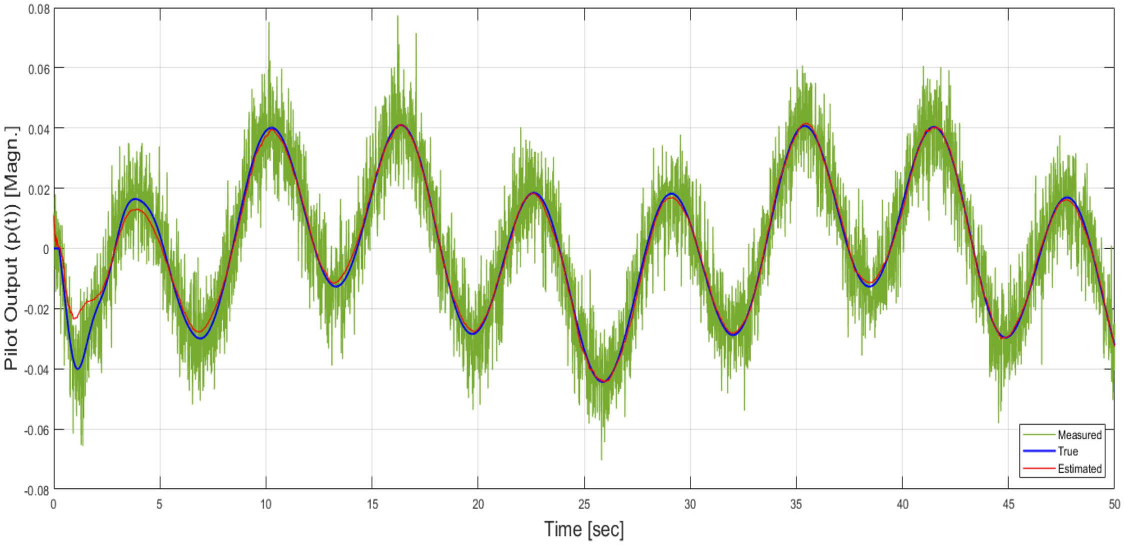


Figure 5.6 UKFx9 Case 1 Results - Pilot model output convergence.

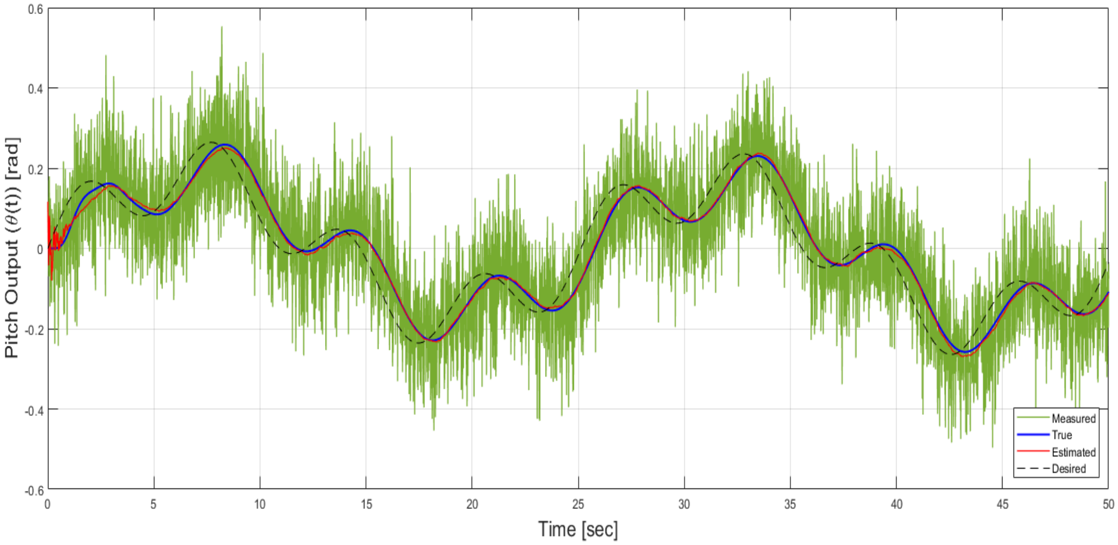


Figure 5.7 UKFx9 Case 1 Results - Pitch model output convergence.

The UKF also converged to the system covariance in the form of the P-matrix. The graph of these parameters is shown in Figure 5.8.

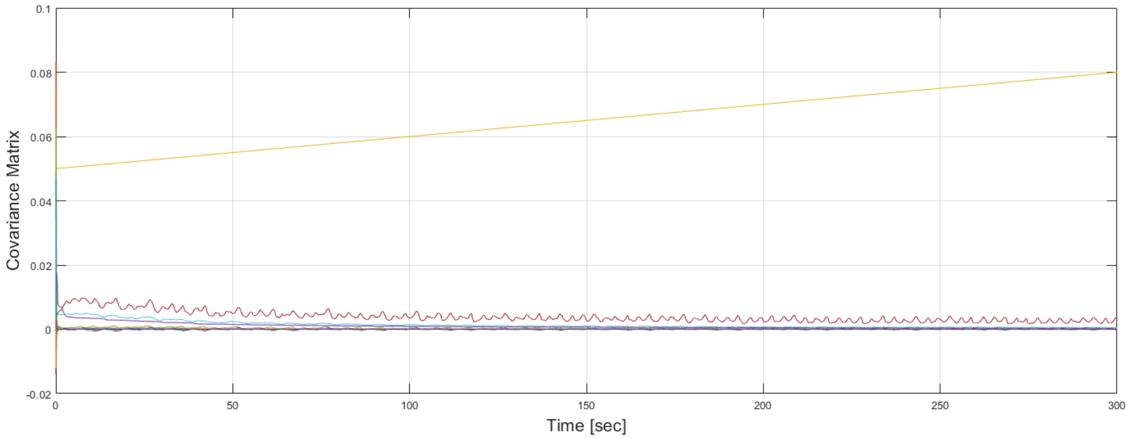


Figure 5.8 UKFx9 Case 1 Results - System covariance matrix convergence.

The resulting estimated pilot model's bode plots can be compared to see how well the frequency response matches the original system. In Figure 5.9, the magnitude and phase diagrams of the original system and the estimated system are compared.

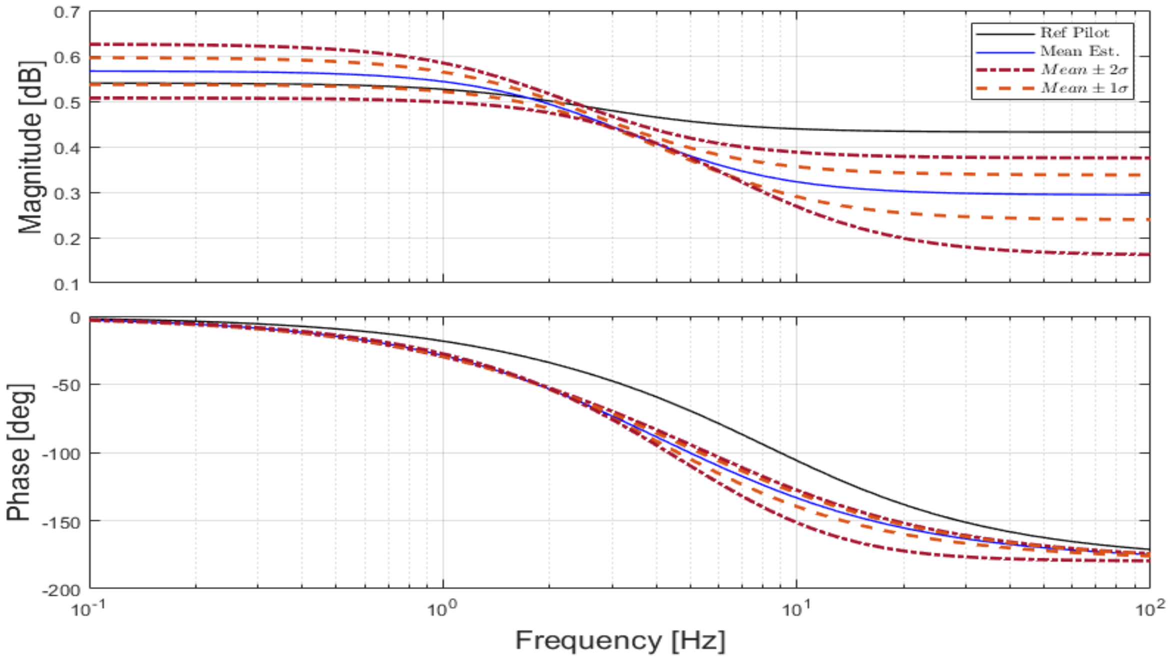


Figure 5.9 UKFx9 Case 1 Results - Bode plot comparison between real and estimated systems.

CASE 2: Case 2 uses a high frequency input signal with time-varying pilot model parameters. The ICs are shown in Equations 5.5-5.8. Using the ICs from case 1 resulted in a solutions which converged to other off-nominal values. Given those results, the system needed to be re-tuned for each input case.

$$X_0 = \begin{bmatrix} \underbrace{0.06}_{K_p} & \underbrace{0.08}_{T_{Lead}} & \underbrace{0.20}_{T_{Lag}} & \underbrace{0.35}_{\tau} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.5}$$

$$Q = \begin{bmatrix} 1x10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1x10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1x10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1x10^{-8} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1x10^{-8} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1x10^{-9} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1x10^{-9} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1x10^{-9} \end{bmatrix} \tag{5.6}$$

$$R = \begin{bmatrix} 0.0001 & 0 \\ 0 & 0.01 \end{bmatrix} \tag{5.7}$$

$$P_0 = 0.05\mathbb{I}_{9x9} \tag{5.8}$$

The resulting pilot model parameter tracking is shown in Figure 5.10. The output results are shown in Figures 5.11 and 5.12. The system covariance matrix (P-matrix) convergence is shown in Figure 5.13. Since this is a time-varying case for the pilot model parameters, a mean cannot be taken of the whole dataset. Instead, three bode plots are presented in Figures 5.14 through 5.16 which show the progress of convergence to the true pilot model's frequency response characteristics. They are labeled as "Beginning", "Middle", and "End" to depict a sample in the first 50 points, in the middle of the set, and in the last 50 points.

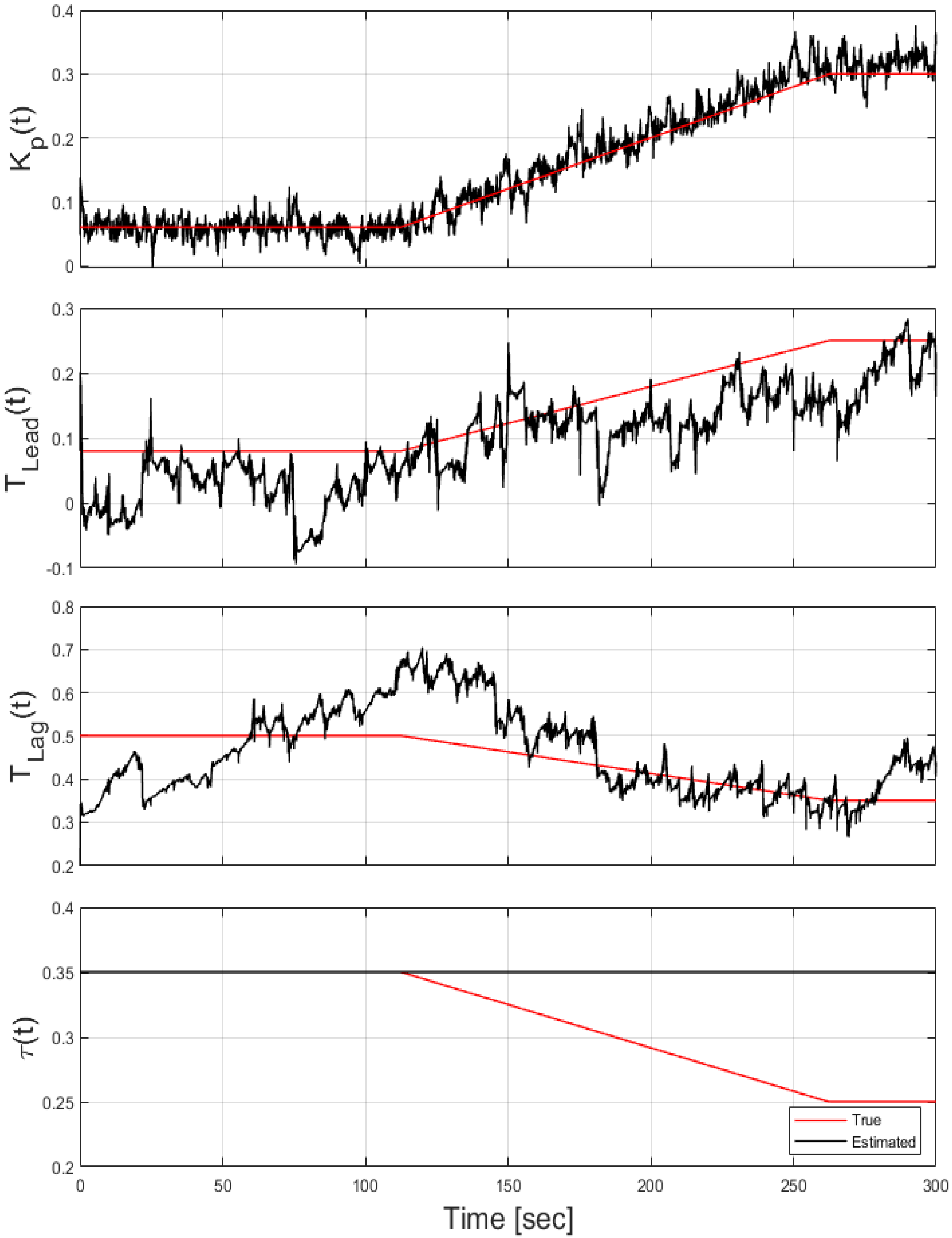


Figure 5.10 UKFx9 Case 2 Results - Pilot model parameter convergence.

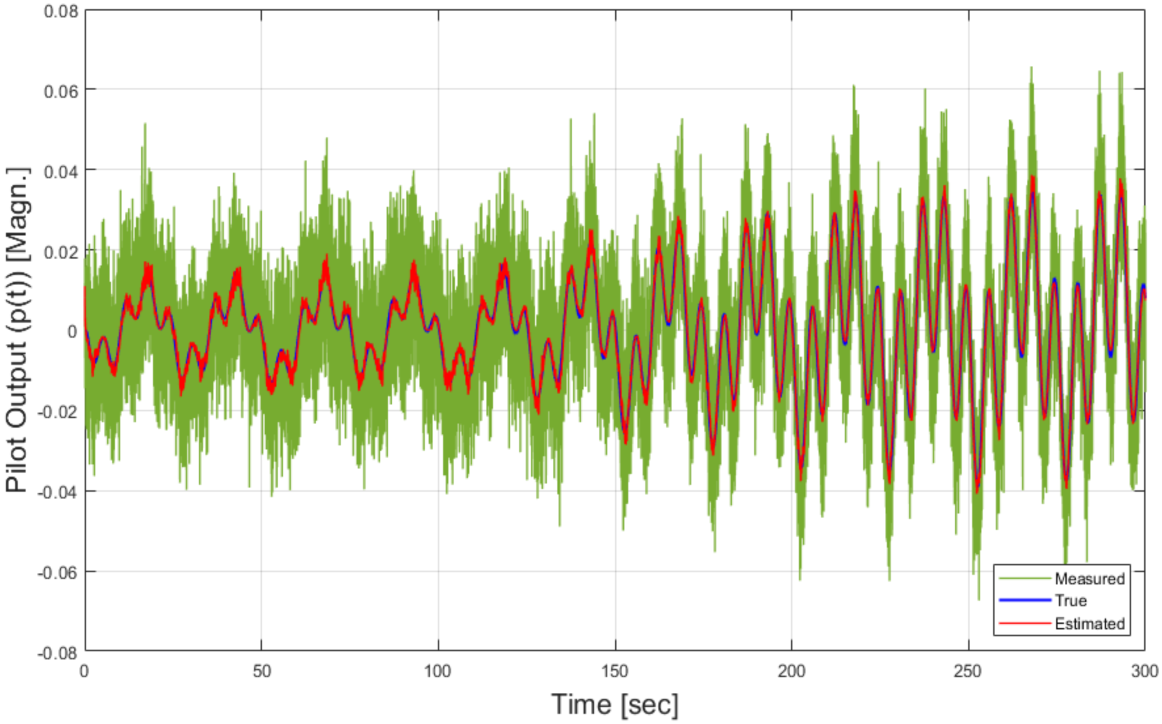


Figure 5.11 UKFx9 Case 2 Results - Pilot model output convergence.

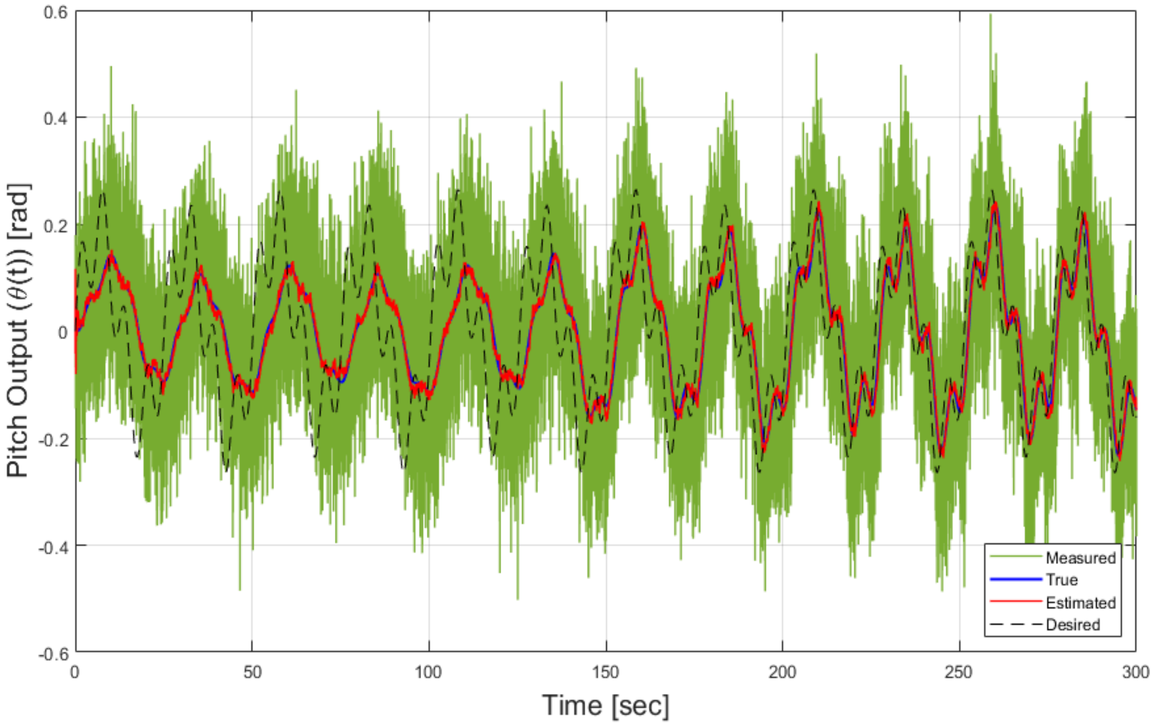


Figure 5.12 UKFx9 Case 2 Results - Pitch model output convergence.

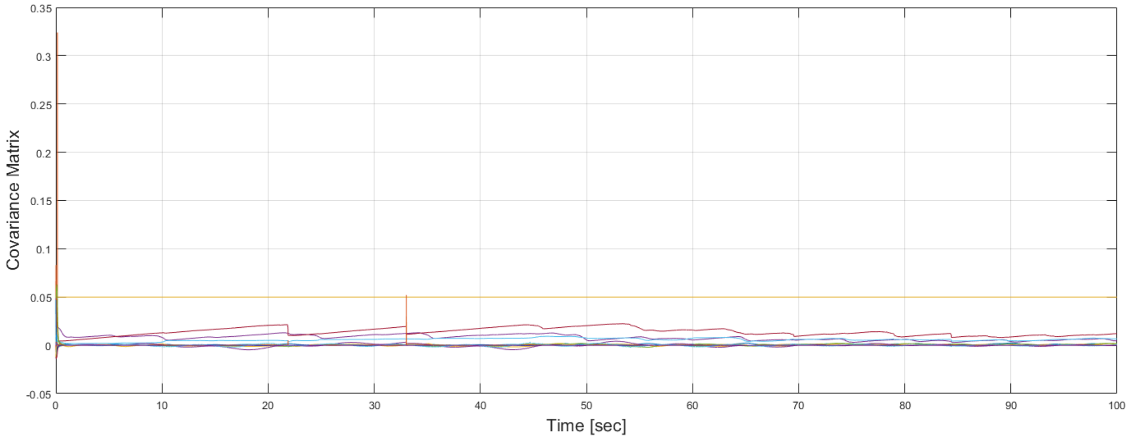


Figure 5.13 UKFx9 Case 2 Results - System covariance matrix convergence.

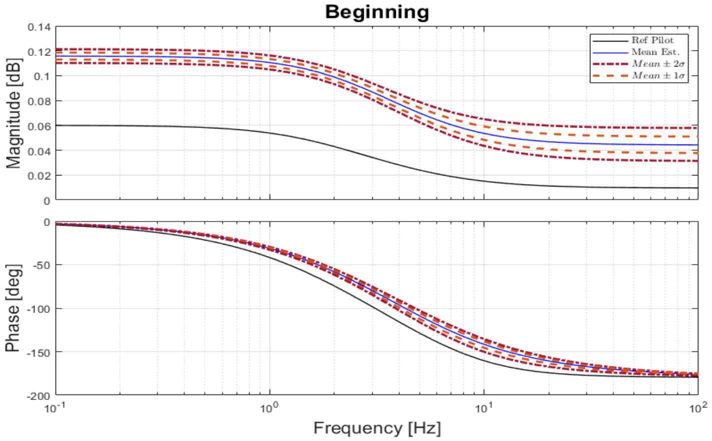


Figure 5.14 UKFx9 Case 2 Results - Beginning Bode plot comparison between real and estimated systems.

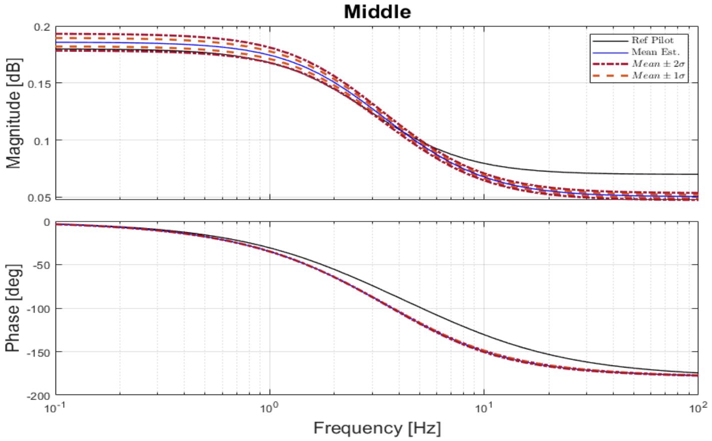


Figure 5.15 UKFx9 Case 2 Results - Middle Bode plot comparison between real and estimated systems.

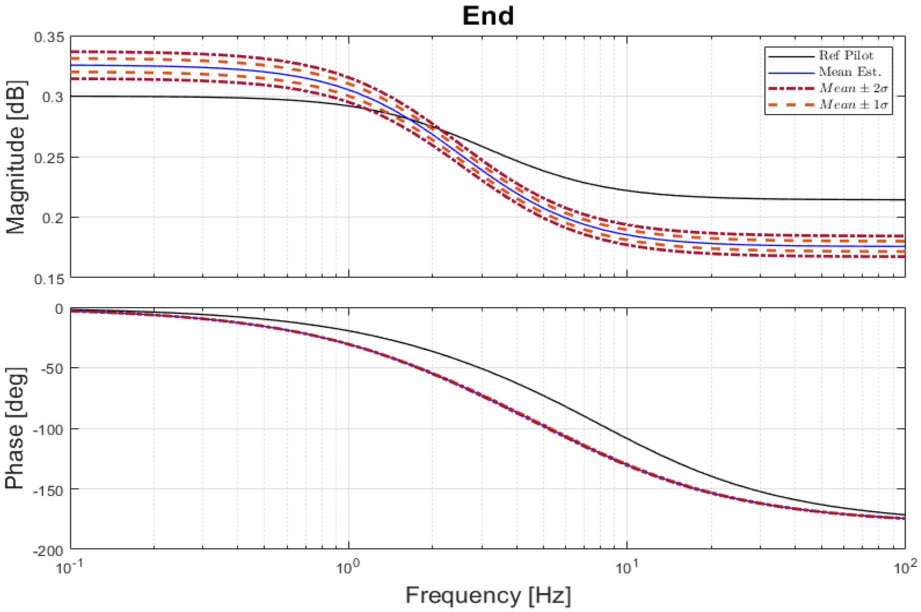


Figure 5.16 UKFx9 Case 2 Results - Ending Bode plot comparison between real and estimated systems.

CASE 3: Case 3 uses a repeated doublet input signal with time-invariant pilot model parameters. The only initial condition that is different than case 1 is the tuned Q-covariance matrix, which is shown in Equation 5.9.

$$Q = \begin{bmatrix} 2x10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1x10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1x10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2x10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2x10^{-6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1x10^{-9} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1x10^{-9} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1x10^{-9} \end{bmatrix} \tag{5.9}$$

The resulting pilot model parameter tracking is shown in Figure 5.17. The output results are shown in Figures 5.18 and 5.19. The system covariance matrix (P-matrix) convergence is shown in Figure 5.20. A bode plot showing the frequency response of the estimated vs the true system is shown in Figure 5.21.

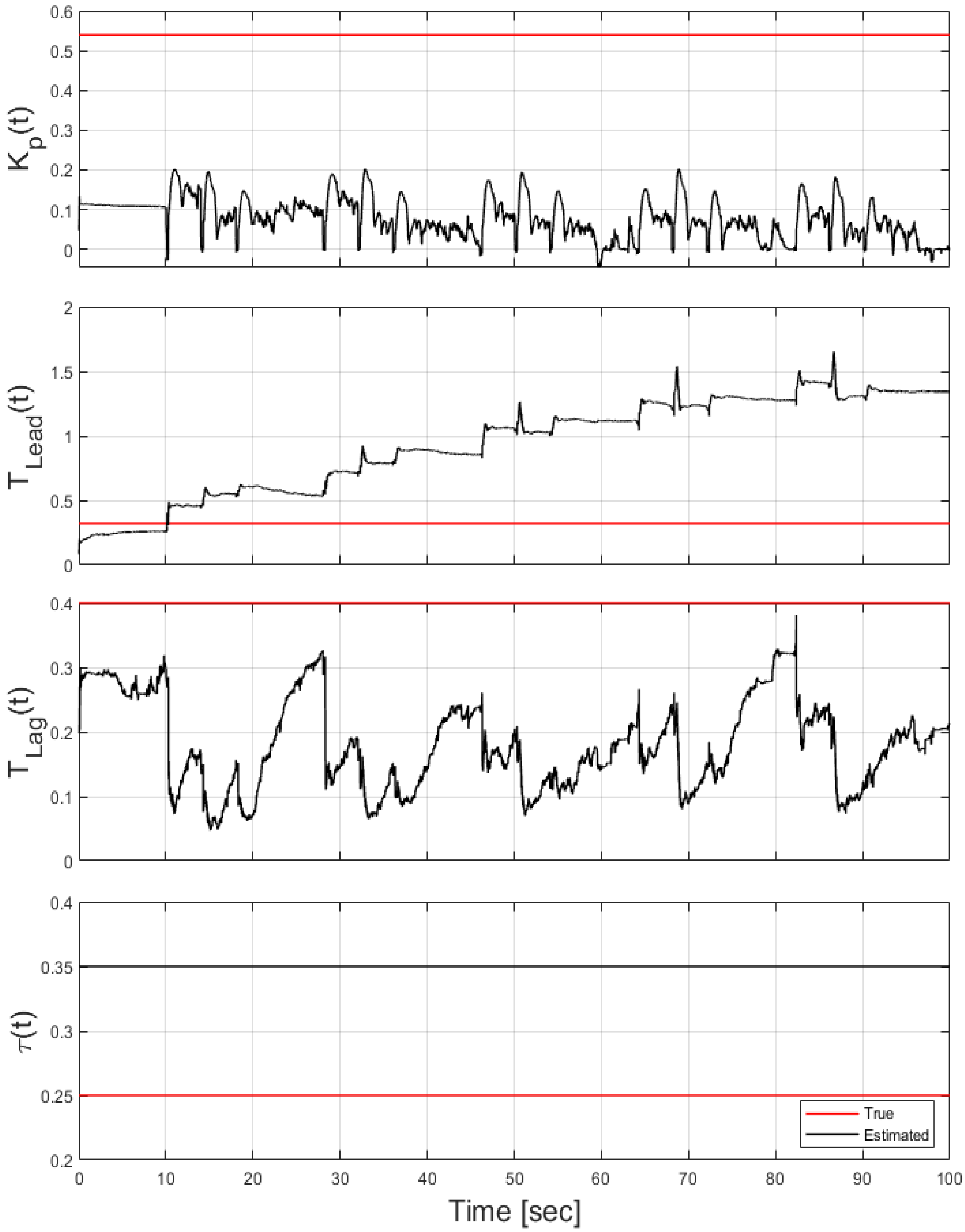


Figure 5.17 UKFx9 Case 3 Results - Pilot model parameter convergence. The pilot is acting as a proportional controller in this case.

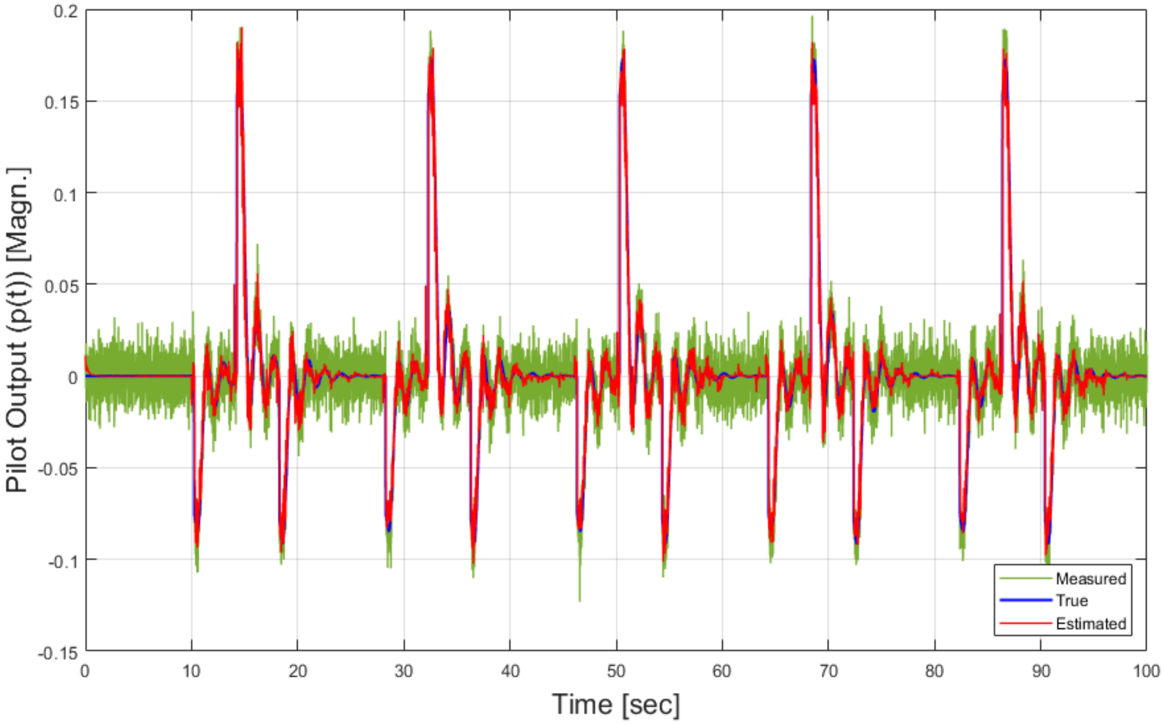


Figure 5.18 UKFx9 Case 3 Results - Pilot model output convergence.

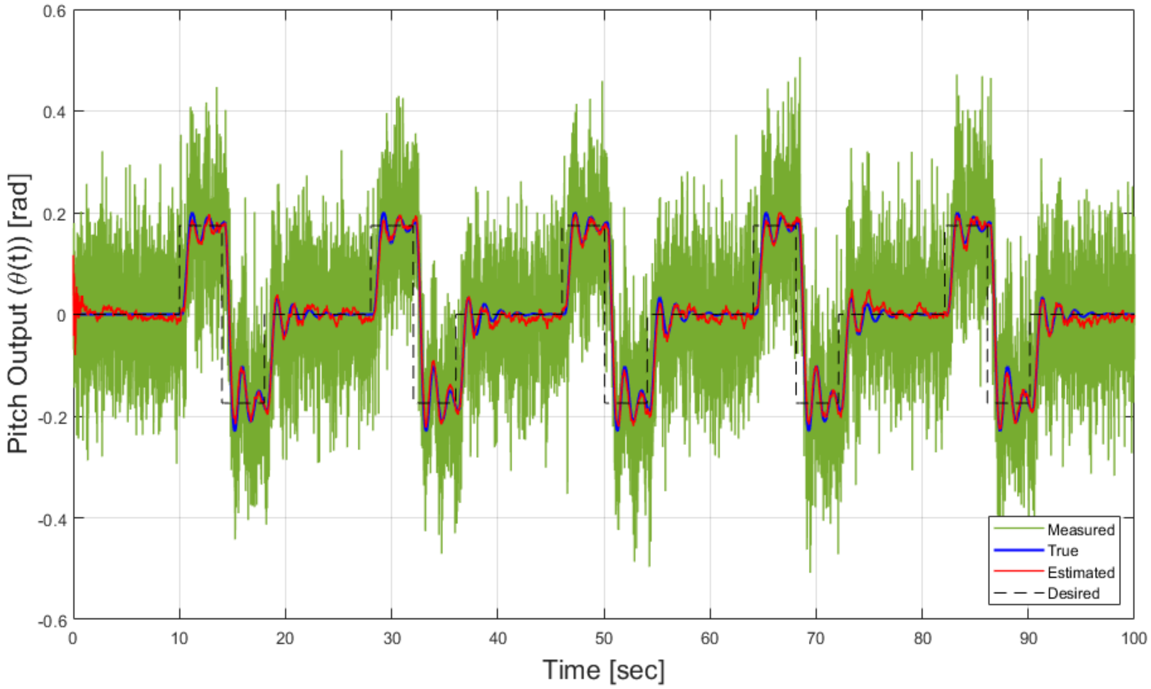


Figure 5.19 UKFx9 Case 3 Results - Pitch model output convergence.

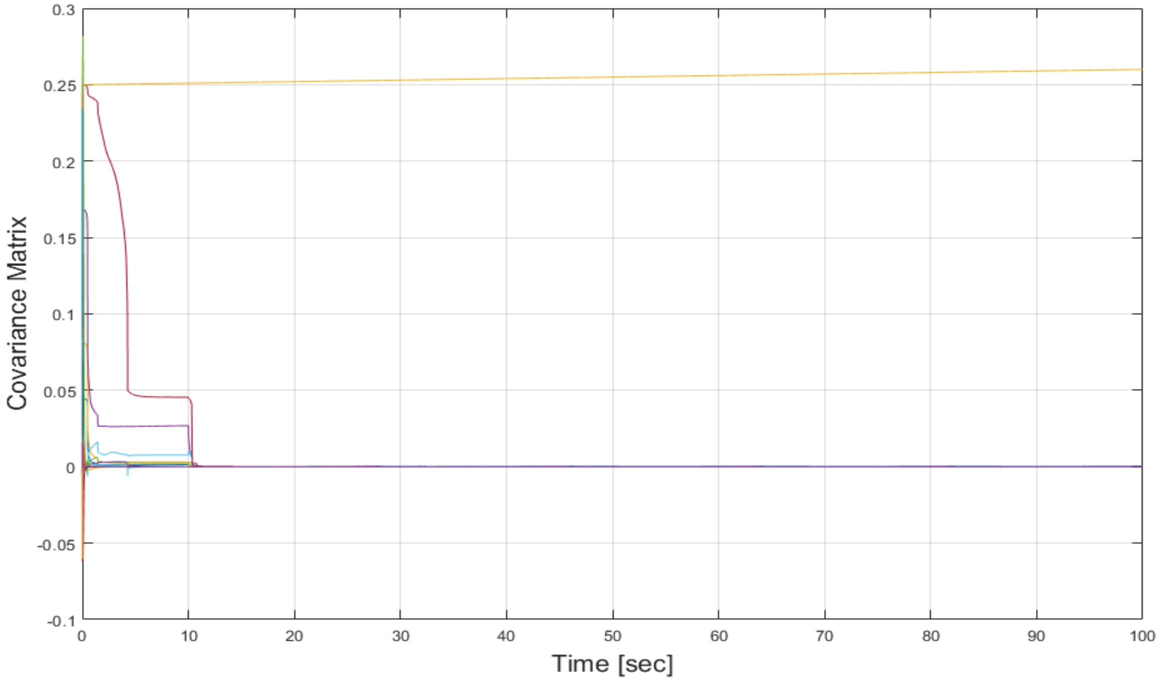


Figure 5.20 UKFx9 Case 3 Results - System covariance matrix convergence.

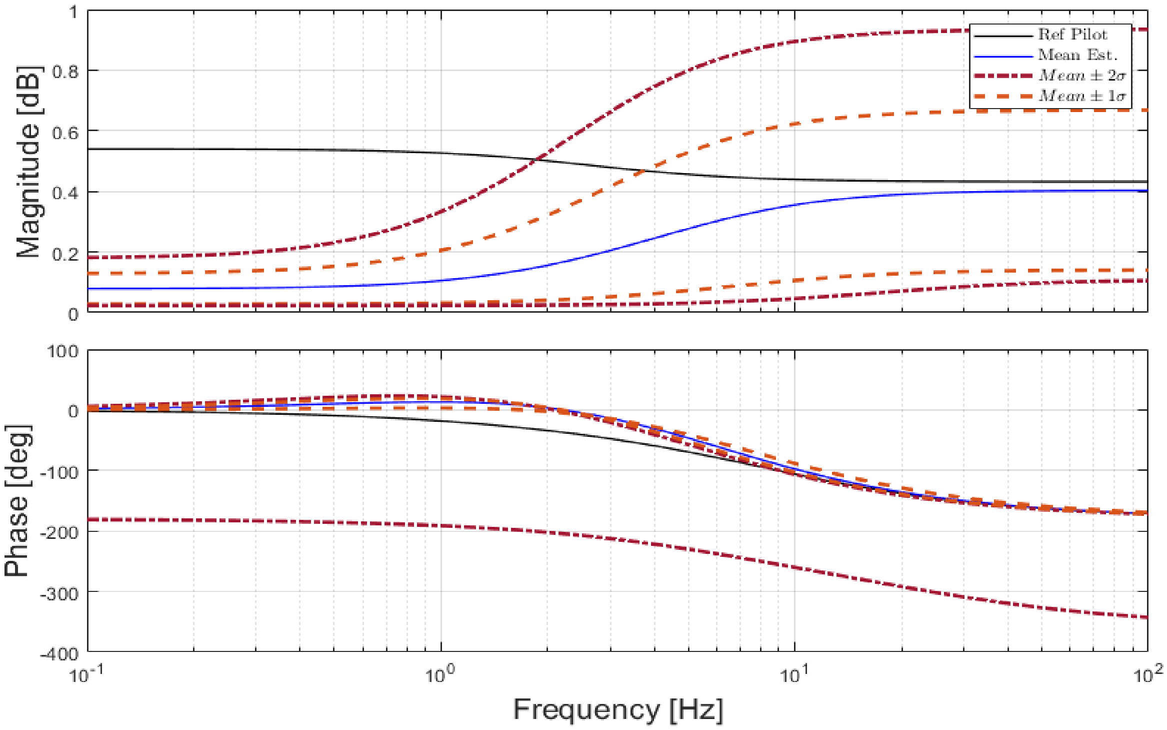


Figure 5.21 UKFx9 Case 3 Results - Bode plot comparison between real and estimated systems.

CASE 4: Case 4 consists of the repeated doublet input with time-varying parameters. This is a considerably more difficult case for which to achieve convergence. The only IC that was changed is the Q matrix, and it is shown in Equation 5.10.

$$Q = \begin{bmatrix} 1x10^{-6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1x10^{-6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1x10^{-6} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1x10^{-8} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1x10^{-9} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1x10^{-9} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1x10^{-9} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1x10^{-9} \end{bmatrix} \tag{5.10}$$

The resulting pilot model parameter tracking is shown in Figure 5.23. The output results are shown in Figures 5.22 and 5.24. The system covariance matrix (P-matrix) convergence is shown in Figure 5.25. Three bode plots are presented in Figures 5.26 through 5.28 which show the progress of convergence to the true pilot model's frequency response characteristics.

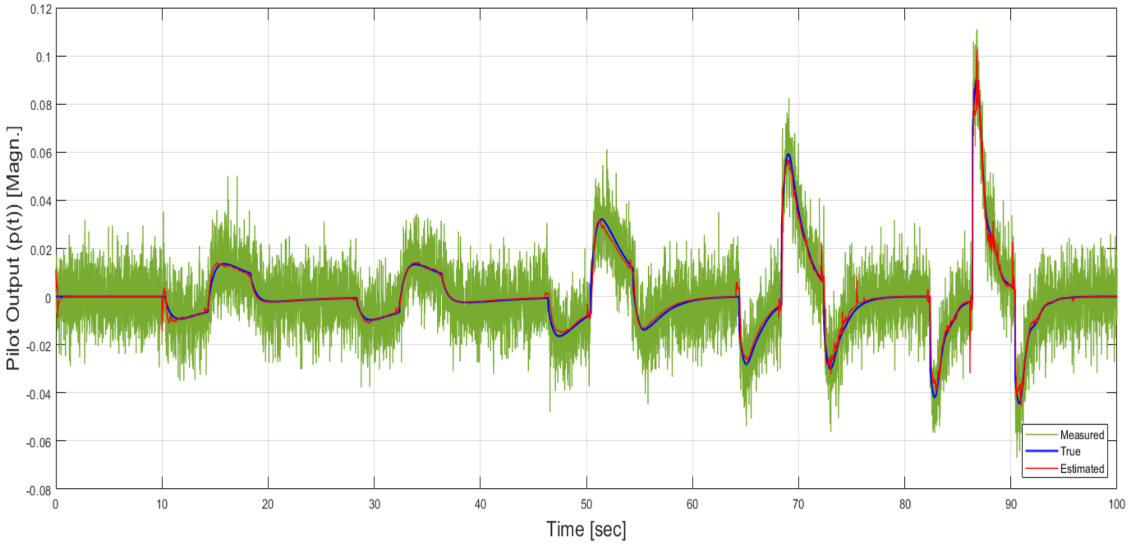


Figure 5.22 UKFx9 Case 4 Results - Pilot model output convergence.

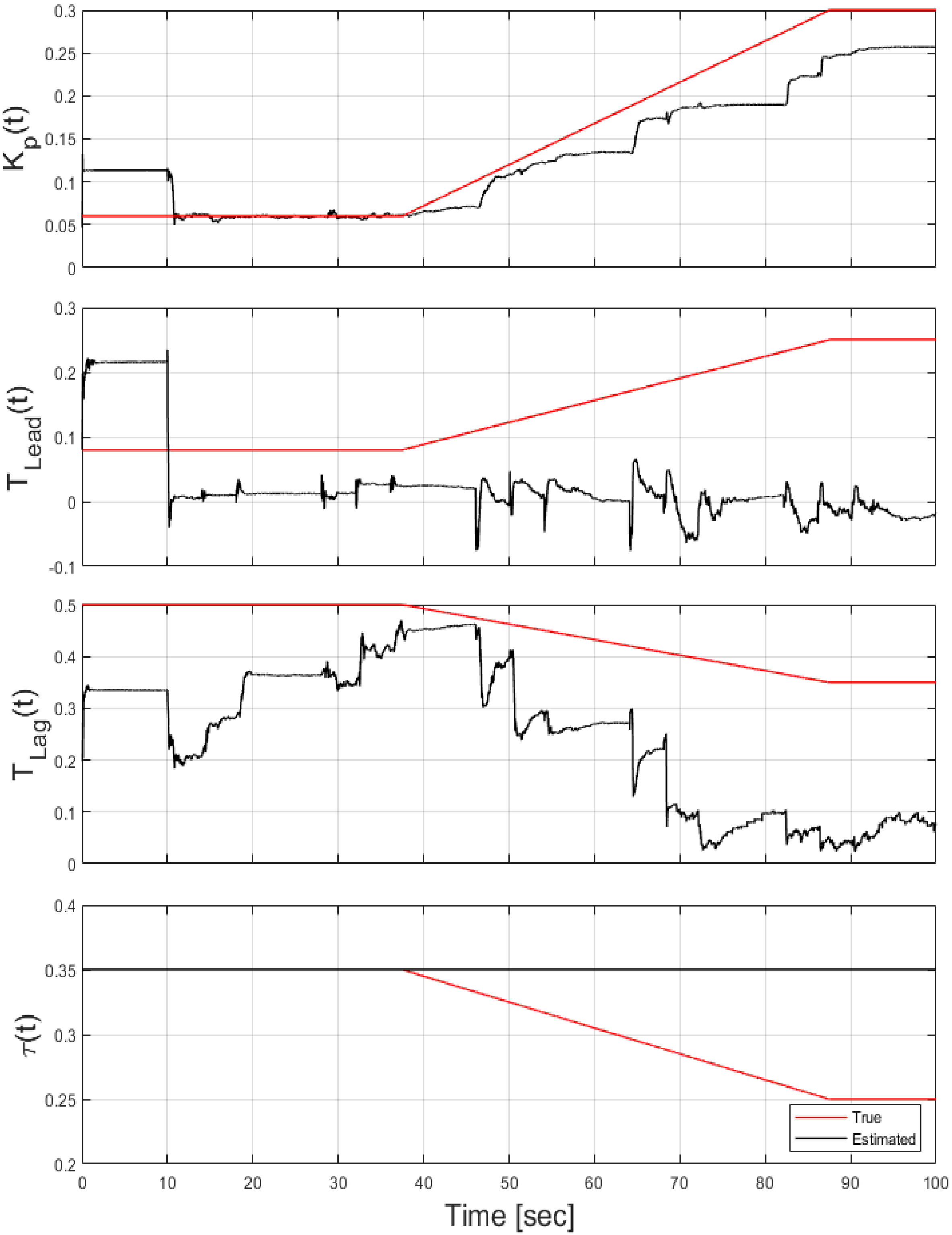


Figure 5.23 UKFx9 Case 4 Results - Pilot model parameter convergence.

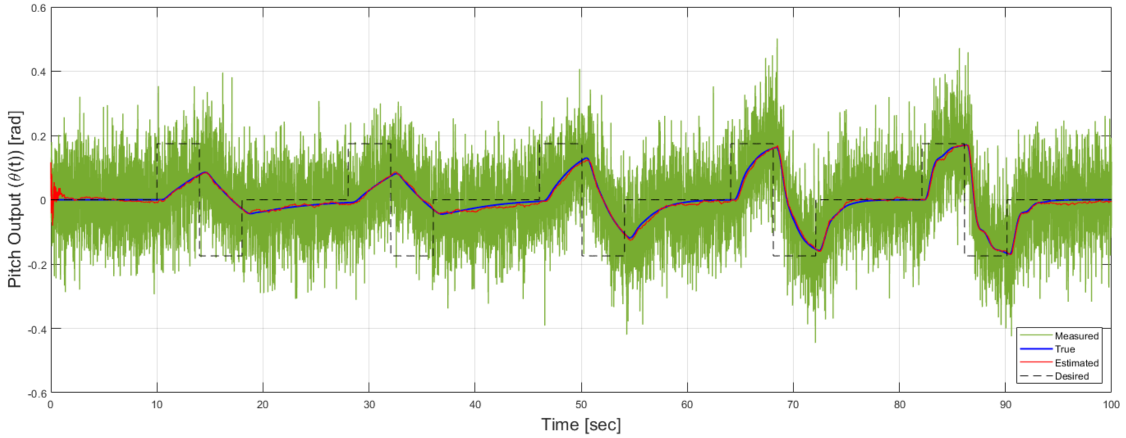


Figure 5.24 UKFx9 Case 4 Results - Pitch model output convergence.

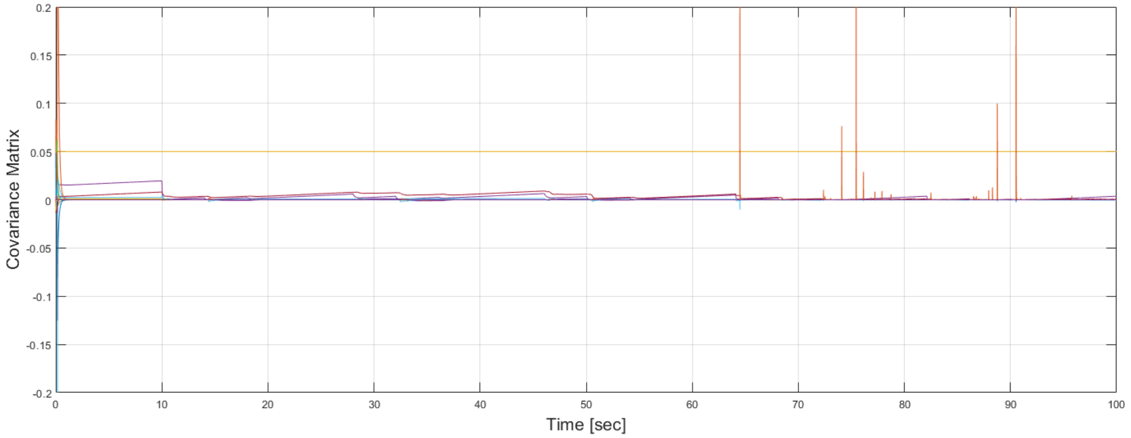


Figure 5.25 UKFx9 Case 4 Results - System covariance matrix convergence.

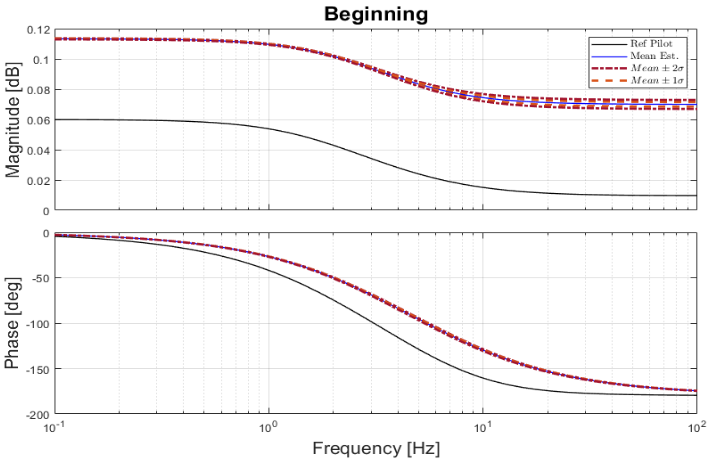


Figure 5.26 UKFx9 Case 4 Results - Beginning Bode plot comparison between real and estimated systems.

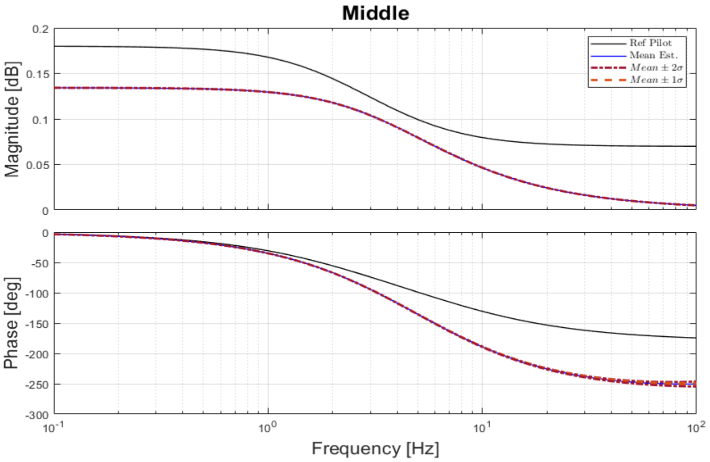


Figure 5.27 UKFx9 Case 4 Results - Middle Bode plot comparison between real and estimated systems.

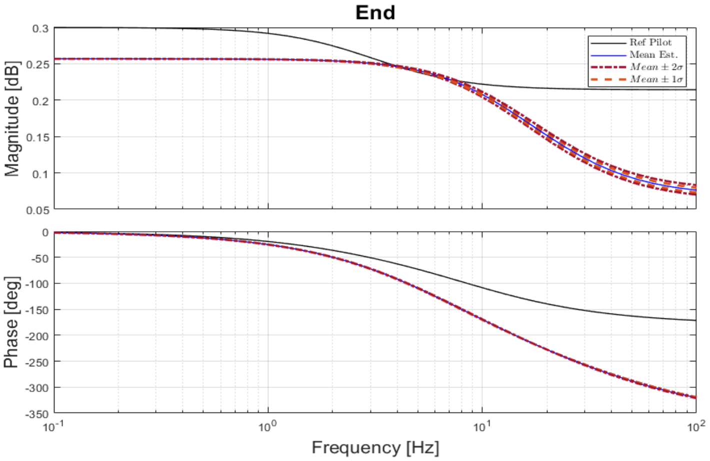


Figure 5.28 UKFx9 Case 4 Results - Ending Bode plot comparison between real and estimated systems.

5.2.2 UKFx9 Results Discussion

Between the four cases presented for the UKFx9 algorithm, it can readily be seen that the Q-matrix is only slightly tweaked on the order of magnitude of 10^{-6}. Among the pilot model parameter plots, it can be seen that the solutions converge to values which may or may not be the true solution. Even with this, the UKF algorithm is able to converge to an estimated model which nearly matches the pilot and pitch outputs.

Given that the desired pitch angle requested of the pilot properly excites the system for the algorithm to converge, the solution eventually reaches estimated values for the pilot model parameters which closely matches the plant's outputs and frequency domain response. One interesting set of plots are Figures 5.5 and 5.11 where the convergence of the parameters T_{Lead} and T_{Lag} seems to depend on the pilot model closely tracking the desired pitch angle. The solution does not seem to converge to the true values unless the pitch error is minimized by the pilot acting as a controller. In cases 3 and 4, due to the simplicity of the desired pitch angle (doublet), the pilot model converged to a simple proportional controller which only needed to control the response with K_p and did not need the Lead-Lag Compensator. Yet again, this shows that there are multiple mathematical models which could achieve the response. A summary of the mean squared error for all cases is shown in Table 5.3.

Table 5.3 UKFx9 Case Results: Table Summary of Mean Squared Errors

	Mean-Squared Error			
Parameter	Case 1	Case 2	Case 3	Case 4
K_p	0.016	0.0004	0.2160	0.0014
T_{Lead}	0.0226	0.0041	0.5173	0.0263
T_{Lag}	0.0081	0.0074	0.0530	0.0462
τ	0.0100	0.0029	0.0100	0.0029
$p(t)$	$2.01x10^{-6}$	$1.70x10^{-6}$	$4.95x10^{-6}$	$5.46x10^{-6}$
$\theta(t)$	$2.93x10^{-5}$	$8.56x10^{-5}$	$4.65x10^{-5}$	$3.95x10^{-5}$

5.3 PID - Fourier Transform Regression

The goal of PID in this solution is to converge to the state space form given by the linearization of the pilot model. However, this system cannot be tuned other than changing the cutoff frequency, ω, sample time, dt, and number of points per batch, n. The cutoff frequency is applied through a simple first order transfer function which approximates Equation 5.11. The initial conditions are set according to Equation 5.12.

$$Y(s) = \frac{a}{s+a} \tag{5.11}$$

$$\omega = 1Hz \quad dt = 0.05s \quad n = 128 \tag{5.12}$$

The desired values are arranged in the form depicted by 4.2 to get $\hat{\Theta}$,. In this case, the PID solution requires that the entire state space is estimated directly as shown in Section 5.1.2. Equations 4.5 and 4.6 must then be used to solve for the pilot model parameters.

CASE 1: High frequency desired pitch input with time-invariant parameters. Figure 5.29 shows the capability of FTR-TD to arrive at a solution.

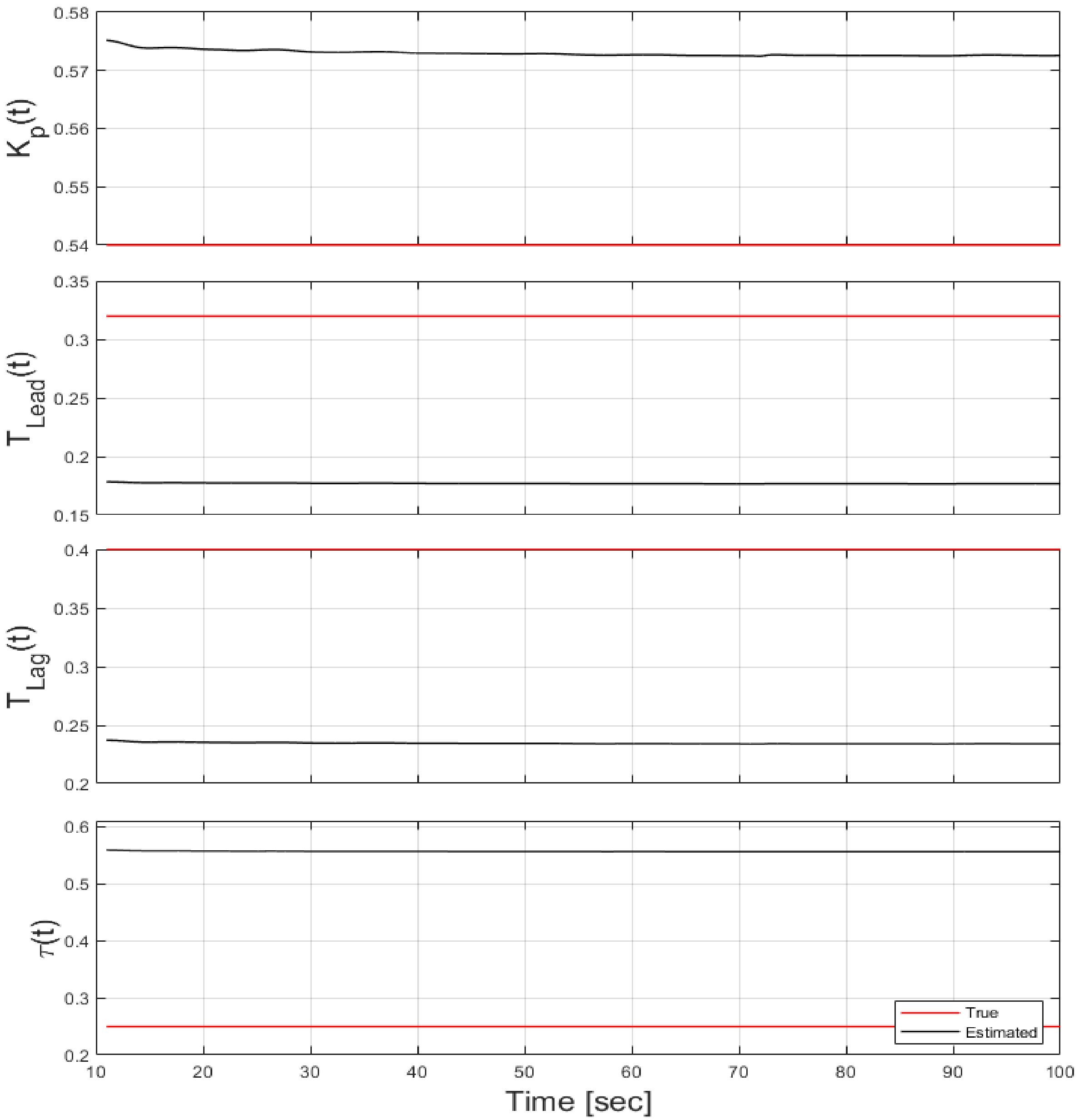


Figure 5.29 FTR-TD Case 1 Results - Pilot model parameter convergence.

The parameters may not converge to the correct values. It was already learned from the UKF results that the found solution may not always be the true solution even though it still tracks the pilot model outputs. This is shown in Figure 5.30 where $p(t)$ can be visualized. The frequency response shown in Figure 5.31 reinforces this point as well. Note that this is a model which uses the states from the PVS since FTR-TD does not estimate states.

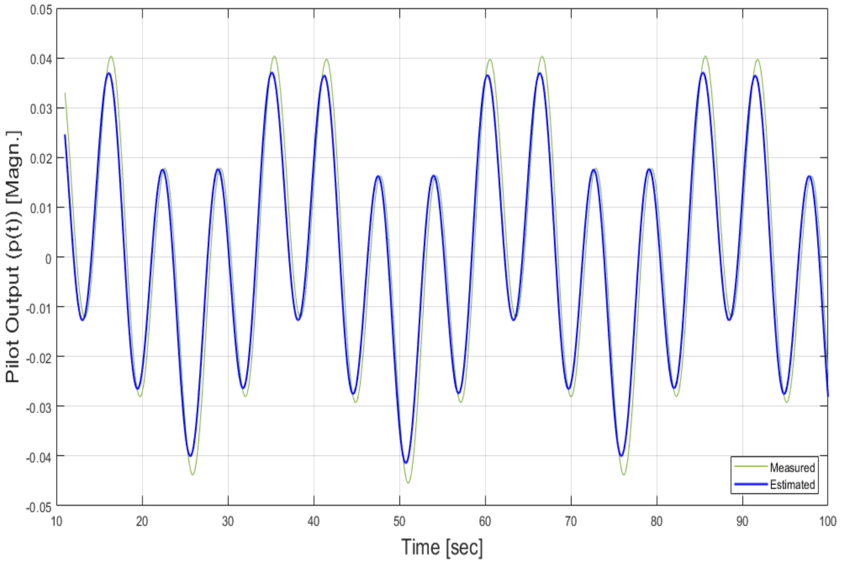


Figure 5.30 FTR-TD Case 1 Results - Pilot model output.

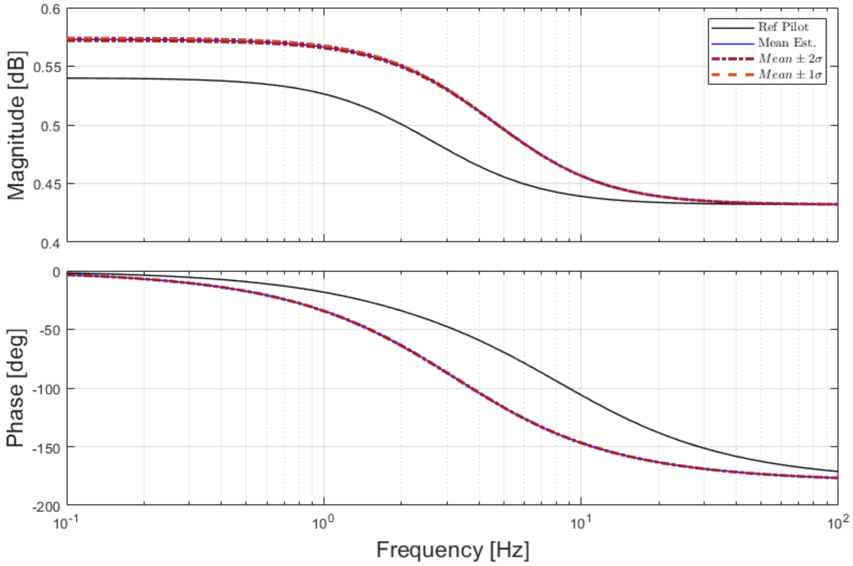


Figure 5.31 FTR-TD Case 1 Results - Frequency response bode plot.

CASE 2: One of the major downfalls of FTR-TD is in its usage of batch data. This is shown more effectively using the time-variant pilot model parameters in case 2. The pilot model parameters have much difficulty converging as can be seen in Figure 5.32. Zooming in to neglect the singularity, Figure 5.33 shows how the parameters converge up until the pilot model parameters begin varying with time.

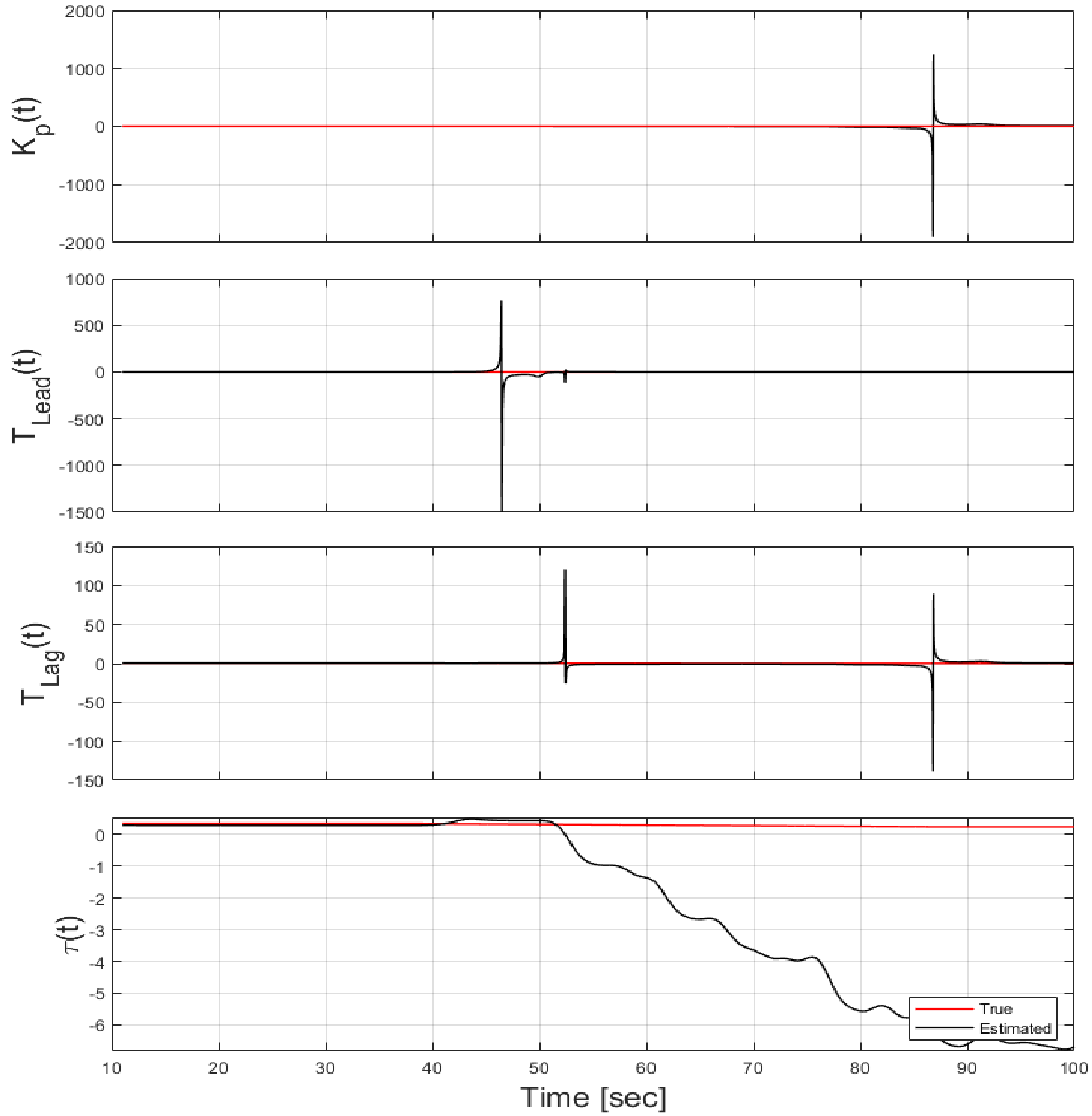


Figure 5.32 FTR-TD Case 2 Results - Pilot model parameter convergence.

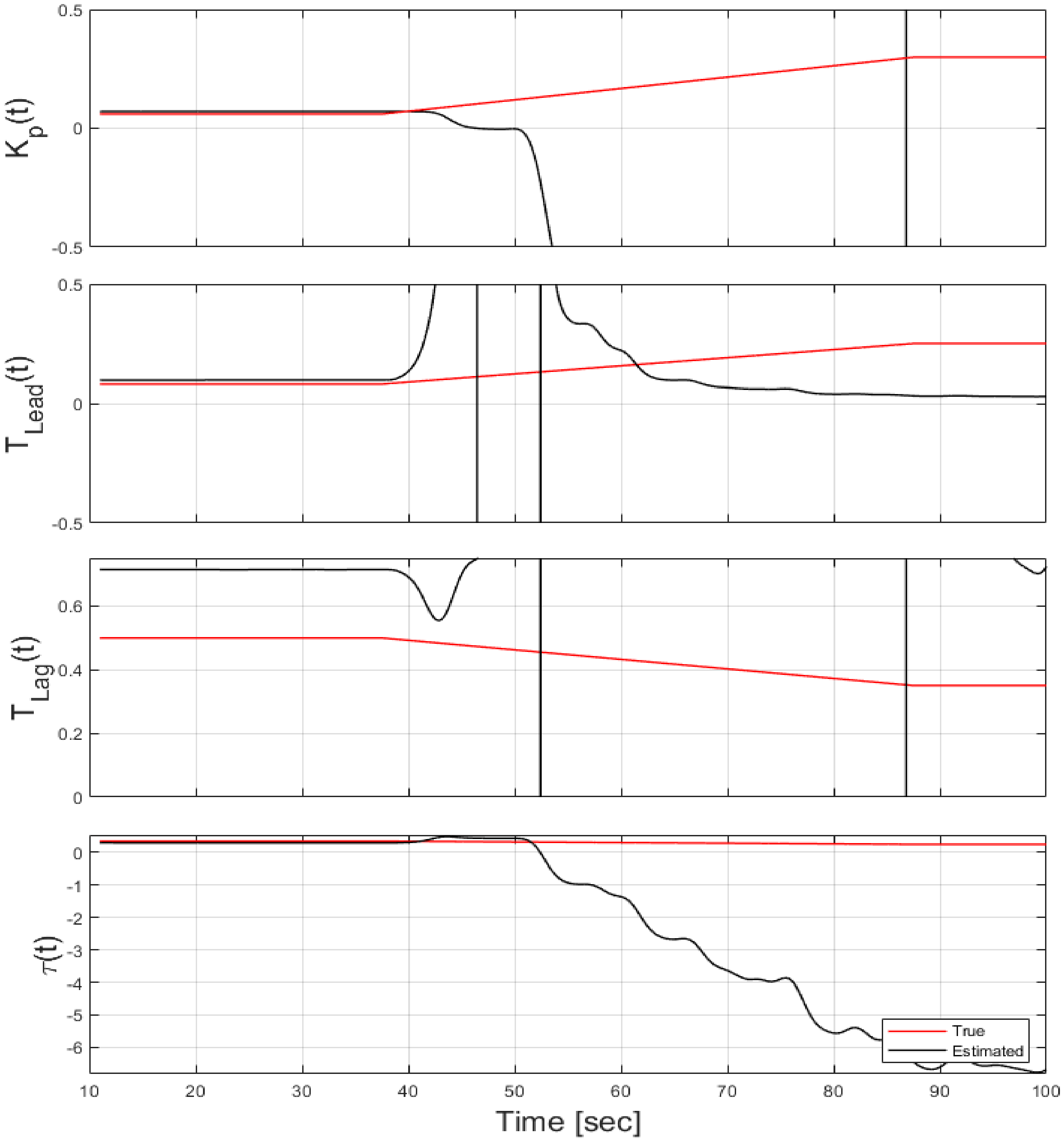


Figure 5.33 FTR-TD Case 2 Results - Pilot model parameter convergence (Zoomed in).

Given enough time once the parameters have become constant again, the solution should converge once more, but the pilot model parameters could change throughout the inputs given by the pilot Therefore this solution cannot be used in general implementation. The singularities depicted in the previous figures result in large deviation from the original system as shown in Figure 5.34.

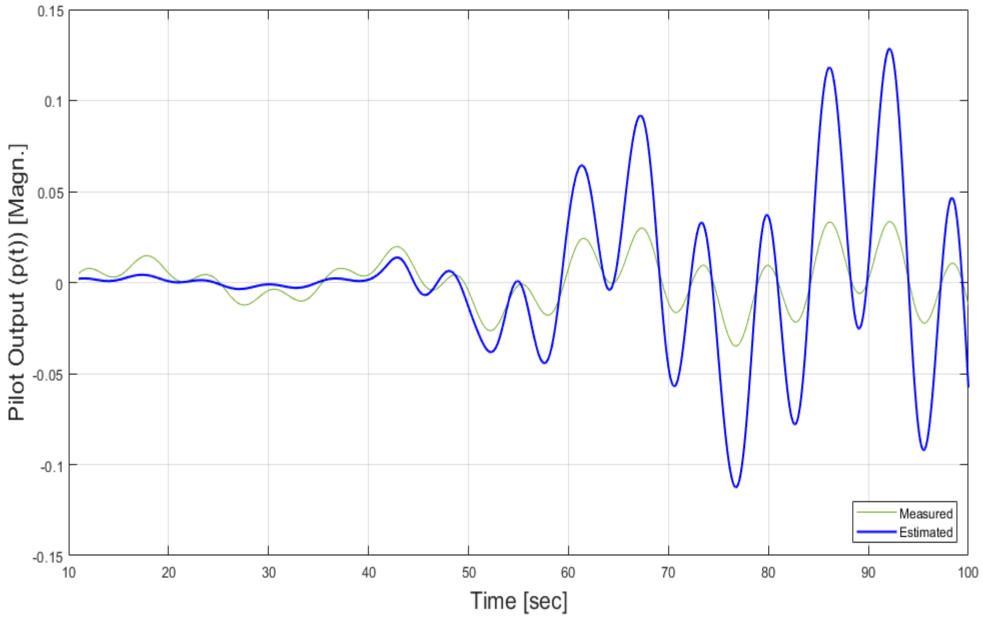


Figure 5.34 FTR-TD Case 2 Results - Pilot model output.

Figure 5.35 depicts the bode diagram from the system convergence before the pilot model variation began. The bode diagrams from after the variation are not shown as they simply show that the predicted model has high deviation from the true values.

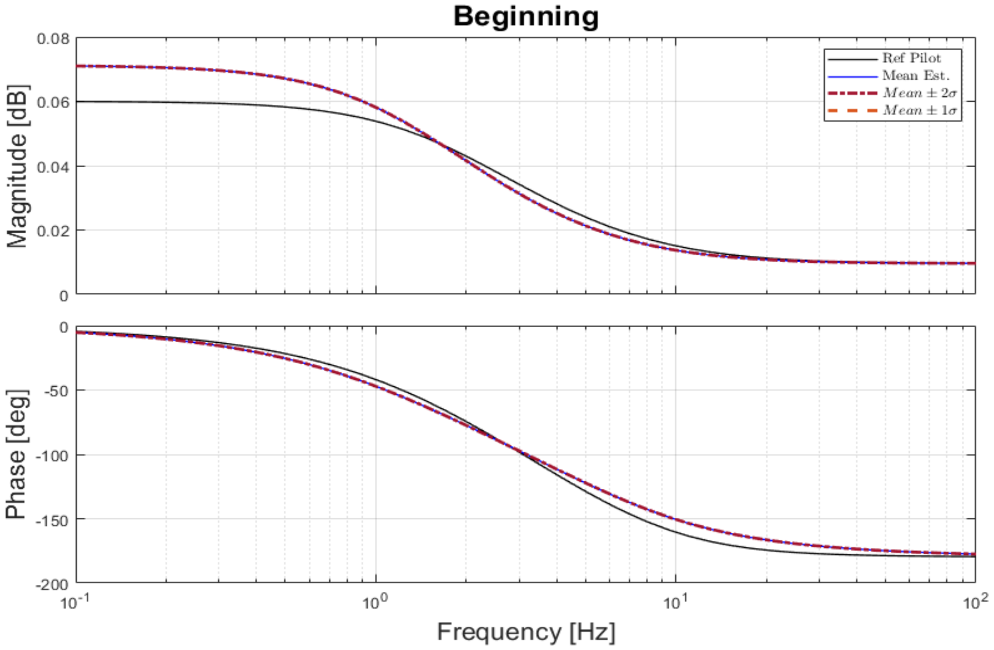


Figure 5.35 FTR-TD Case 2 Results - Frequency response bode plot.

CASE 3: Returning to the time-invariant pilot model case with a lower frequency input, the results of case 3 are shown in Figures 5.36-5.37. To begin, Figure 5.36 shows the pilot model parameter estimation.

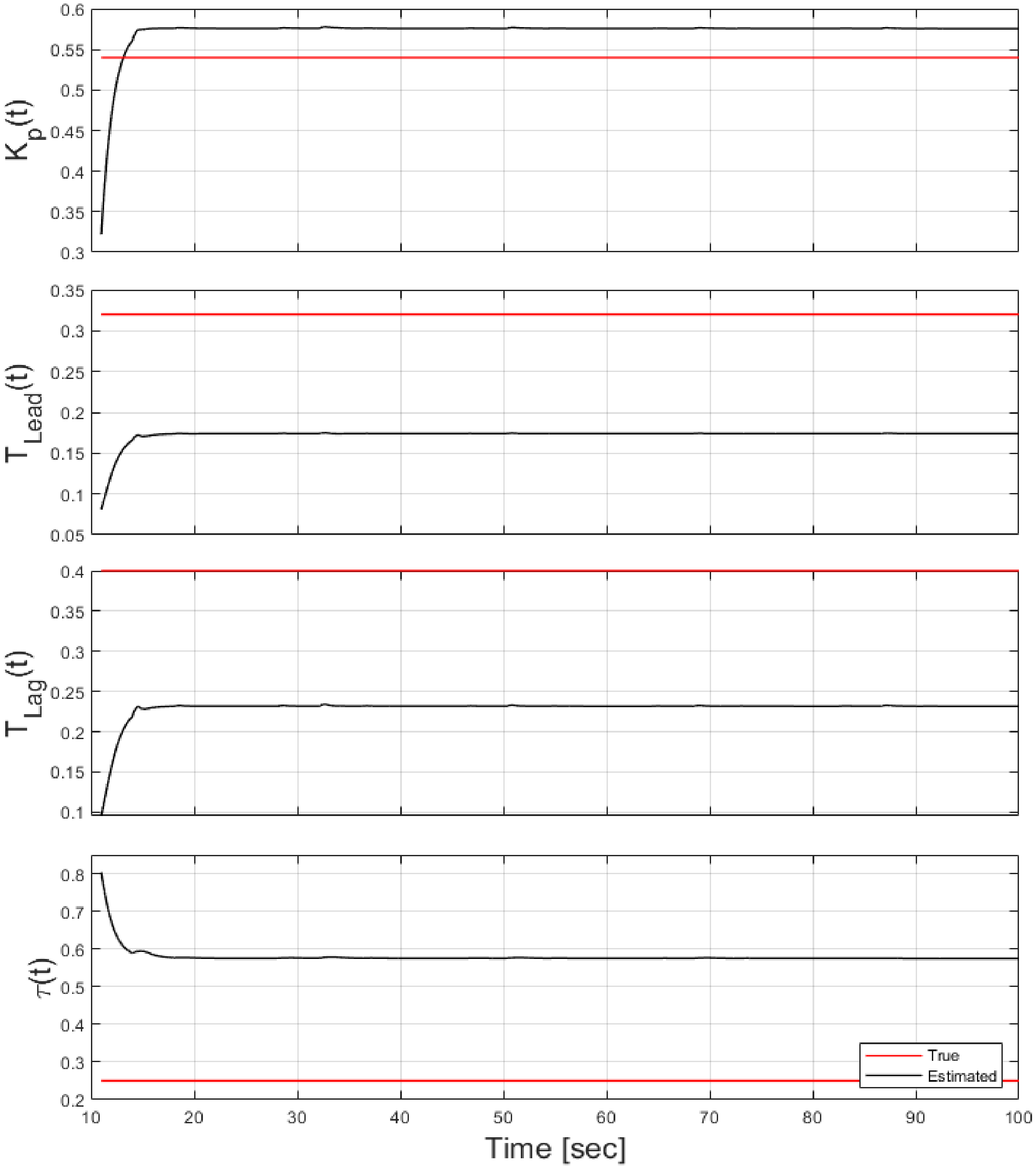


Figure 5.36 FTR-TD Case 3 Results - Pilot model parameter convergence.

Figures 5.37 and 5.38 depict the resulting pilot model output given the estimated parameters and the frequency response of the system.

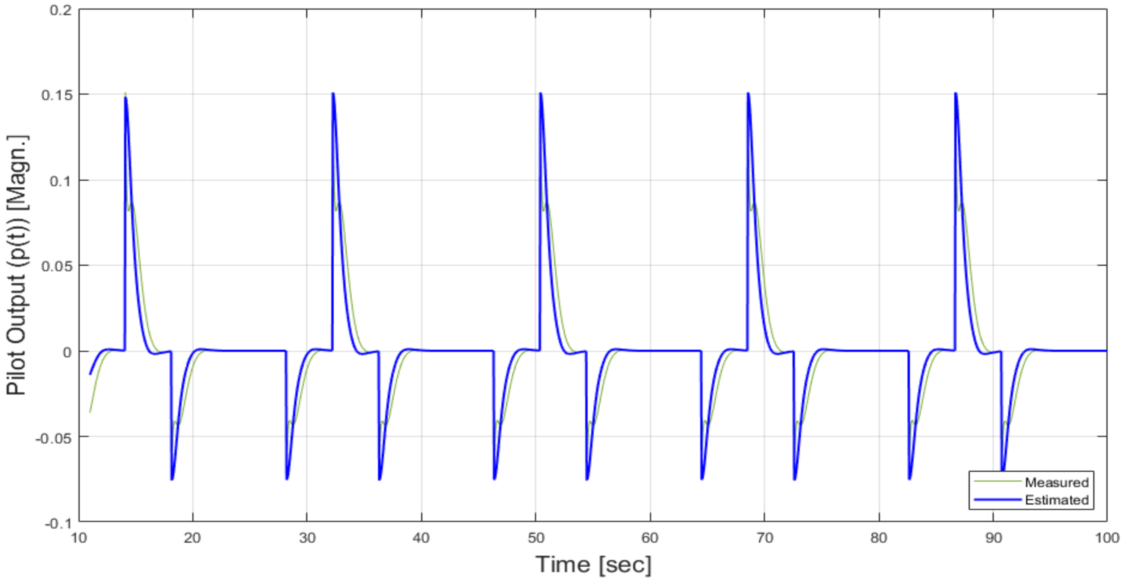


Figure 5.37 FTR-TD Case 3 Results - Pilot model output.

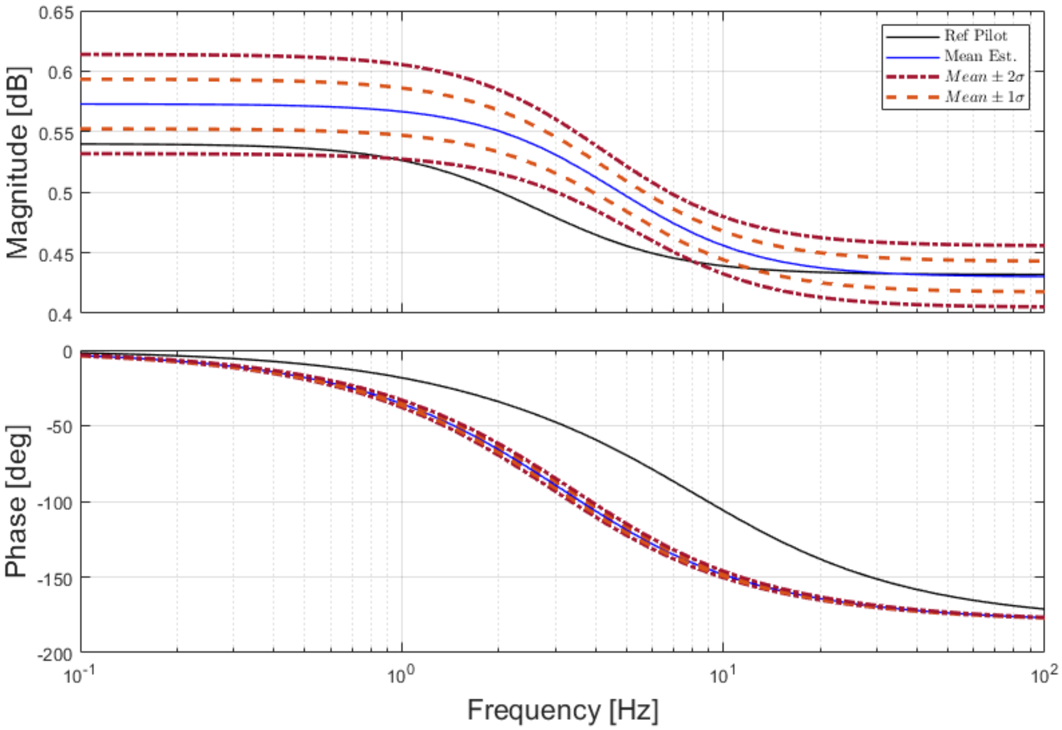


Figure 5.38 FTR-TD Case 3 Results - Frequency response bode plot.

CASE 4: Lastly, the time-variant parameters with a low-frequency doublet has pilot model parameter estimation as shown in Figure 5.39.

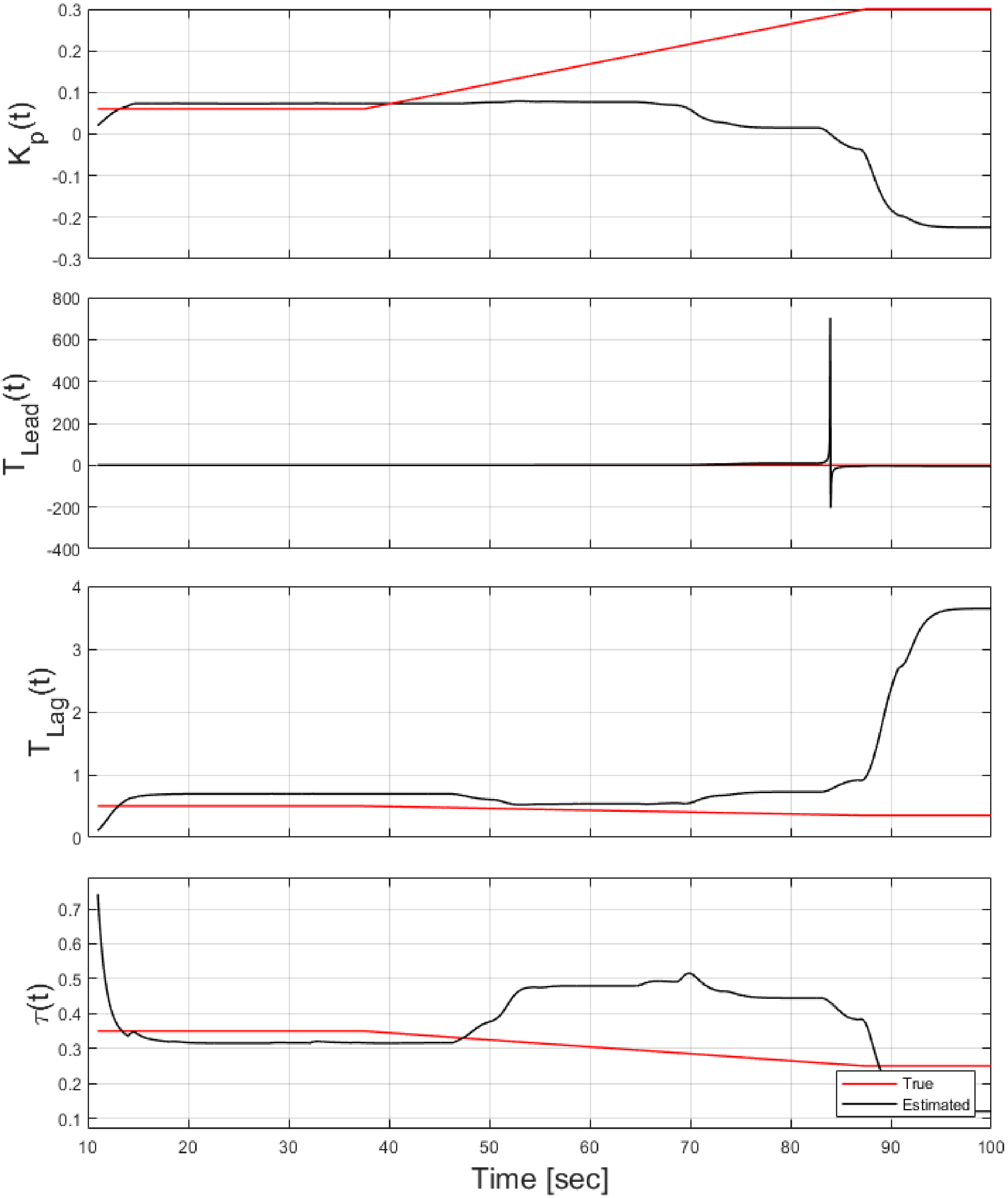


Figure 5.39 FTR-TD Case 4 Results - Pilot model parameter convergence.

The singularities depicted in the previous figures results in large deviation from the original system as shown in Figure 5.40.

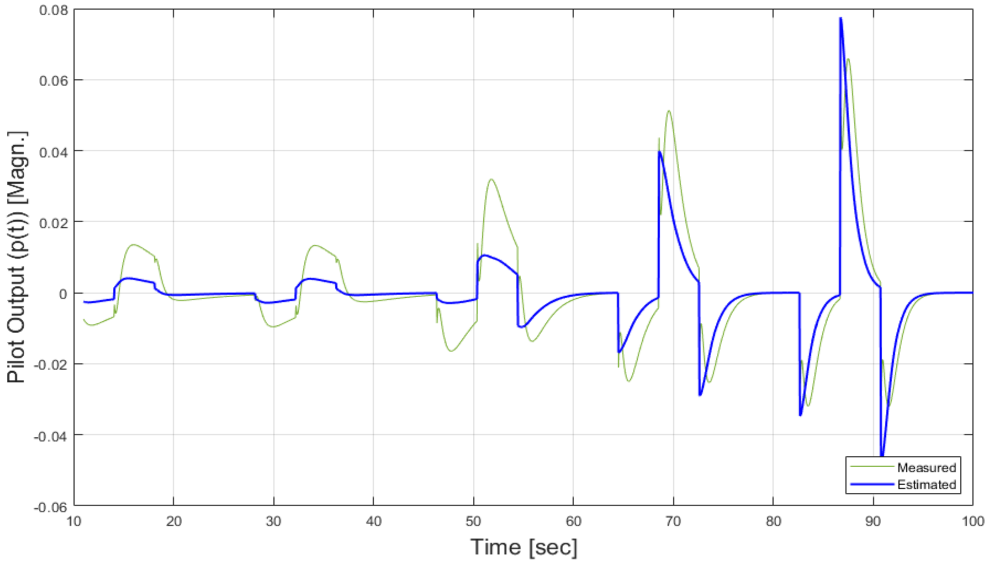


Figure 5.40 FTR-TD Case 4 Results - Pilot model output.

Figure 5.41 depicts the bode diagram from the system convergence before the pilot model variation began. The bode diagrams from after the variation are not shown as they simply show that the predicted model has high deviation from the true values.

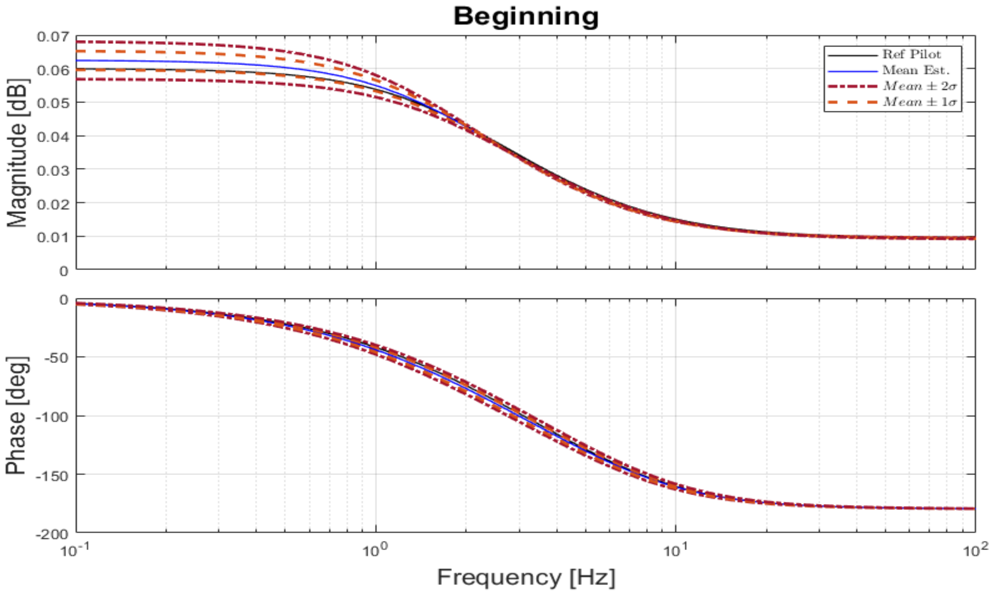


Figure 5.41 FTR-TD Case 4 Results - Frequency response bode plot.

5.3.1 FTR-TD Results Discussion

The pilot model parameter variation creates difficulty for the FTR-TD method to converge to the true values. In fact, it causes significant singularities such that the only way a valid solution can be developed is if the parameters are kept constant for an extended period of time. On the other hand, FTR-TD was capable of converging to the same constant solution every time regardless of how excited the input data was. This can be readily seen in the four bode plots where the solution found is relatively close in frequency response to the true model. Table 5.4 shows a summary of the mean squared error for each of the pilot model parameters and the pilot model outputs.

Table 5.4 FTR-TD Case Results: Table Summary of Mean Squared Errors

	Mean-Squared Error			
Parameter	Case 1	Case 2	Case 3	Case 4
K_p	0.0011	$6.1x10^2$	0.011	42.87
T_{Lead}	0.0205	$1.2x10^5$	0.0204	16.54
T_{Lag}	0.0274	$1.20x10^5$	0.0272	10.49
τ	0.0939	10.0	0.0941	3.50
$p(t)$	$6.95x10^{-4}$	0.025	$6.91x10^{-4}$	$7.1x10^{-3}$

5.4 Adaptive Neural Networks

In this section, the results of the adaptive neural networks are shown along with the respective conditions and parameters to achieve them. Note that that GMLP was used as the adaptive network in all of these cases. There are two methods of tuning used in this study. The first is a gain multiplier for the error seen by the neural network which is adjusted similarly to the learning rate of the ANN. The second is changing the initial conditions of the initial input/output weighting matrices. The latter helps improve the initial conditions to a more reasonable starting point while the former changes how quickly the system learns since the magnitude of the error seen by the loss function is increased. If the initial conditions are off, then the algorithm will take considerably longer to learn the true values. The adaptive NN feedback applies the same pilot output error to all four parameters.

CASE 1: For the high frequency and high amplitude case with time invariant parameters, the initial condition for the error gain, K_e, is 1.25. The η_V, η_W, and η_P values are learning rates and each is set to 0.45. Both work to improve the system's responsiveness. The magnitude of the initial randomized weighting matrix was set to 0.05. Figure 5.42 depicts the estimation of the pilot model parameters. Note that all four parameters have the same trend since the same error is used to update each.

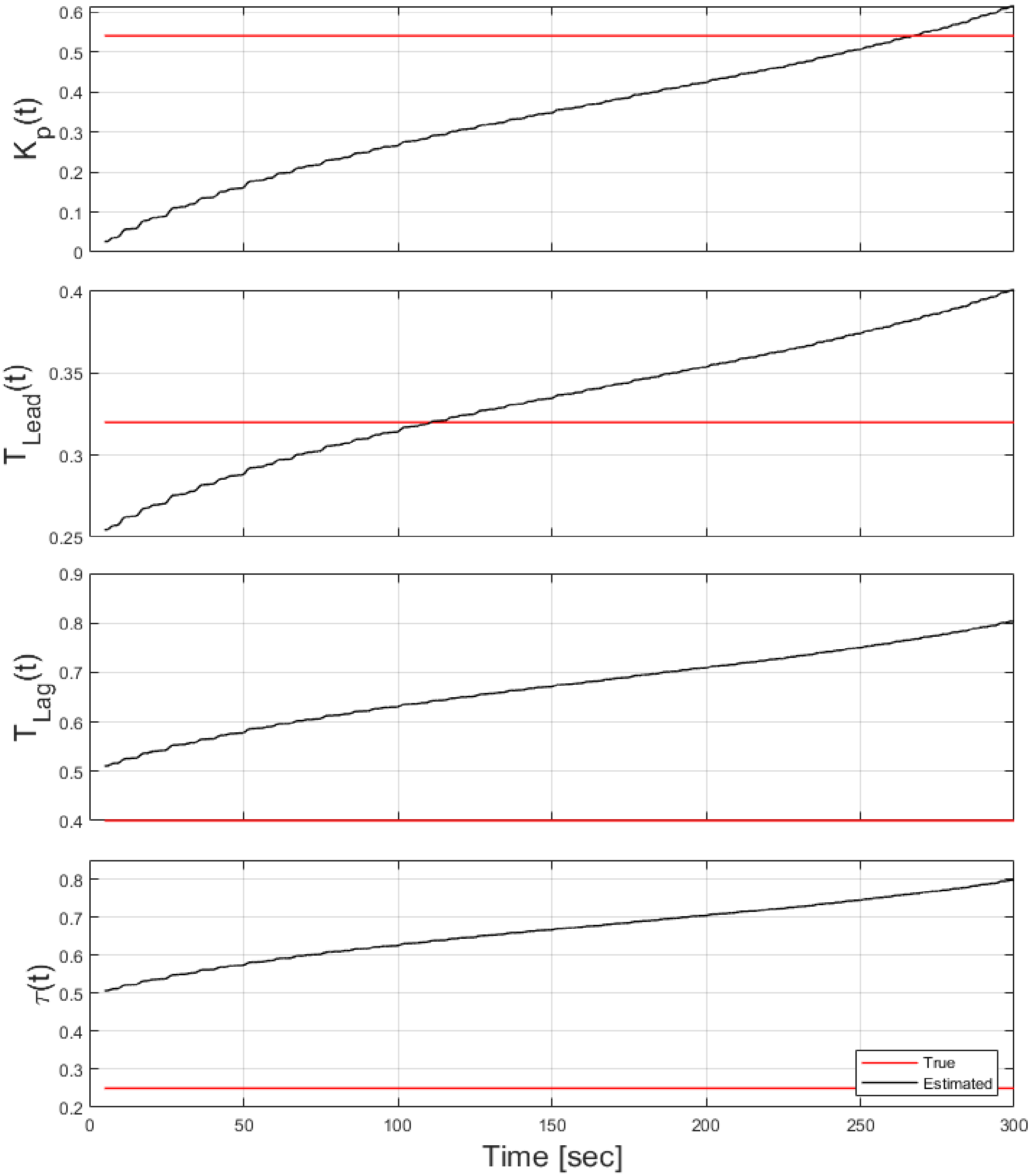


Figure 5.42 Neural Networks Case 1 Results - Pilot model parameter convergence.

Figures 5.43 and 5.44 show the resulting pilot and pitch outputs, respectively. Note that the pitch system's error is not an input to the neural network. The ANN in this case is not attempting to minimize error in the pitch system. The assumption is that if the pilot model matches, the pitch system will as well.

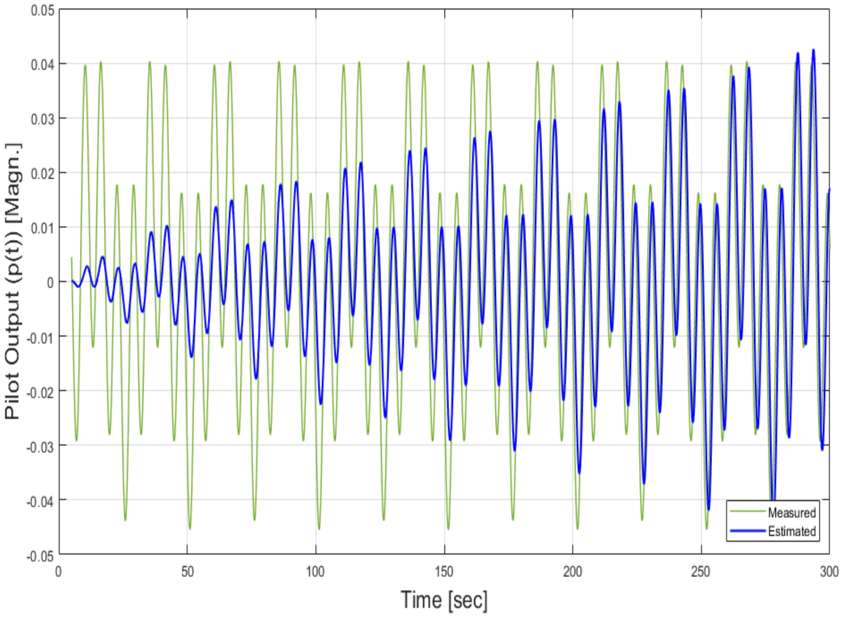


Figure 5.43 Neural Networks Case 1 Results - Pilot model output convergence.

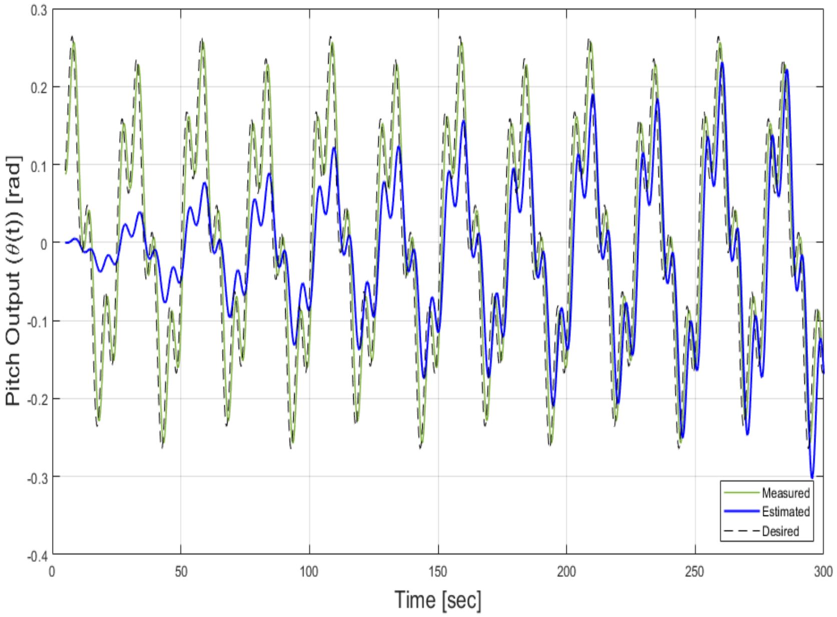


Figure 5.44 Neural Networks Case 1 Results - Pitch model output convergence.

CASE 2: Adding pilot model variation with time to case 1, the initial condition for the error gain, K_e, is 7.0. The η_V, η_W, and η_P values are learning rates and each is set to 0.30. The parameters estimated as shown in Figure 5.45.

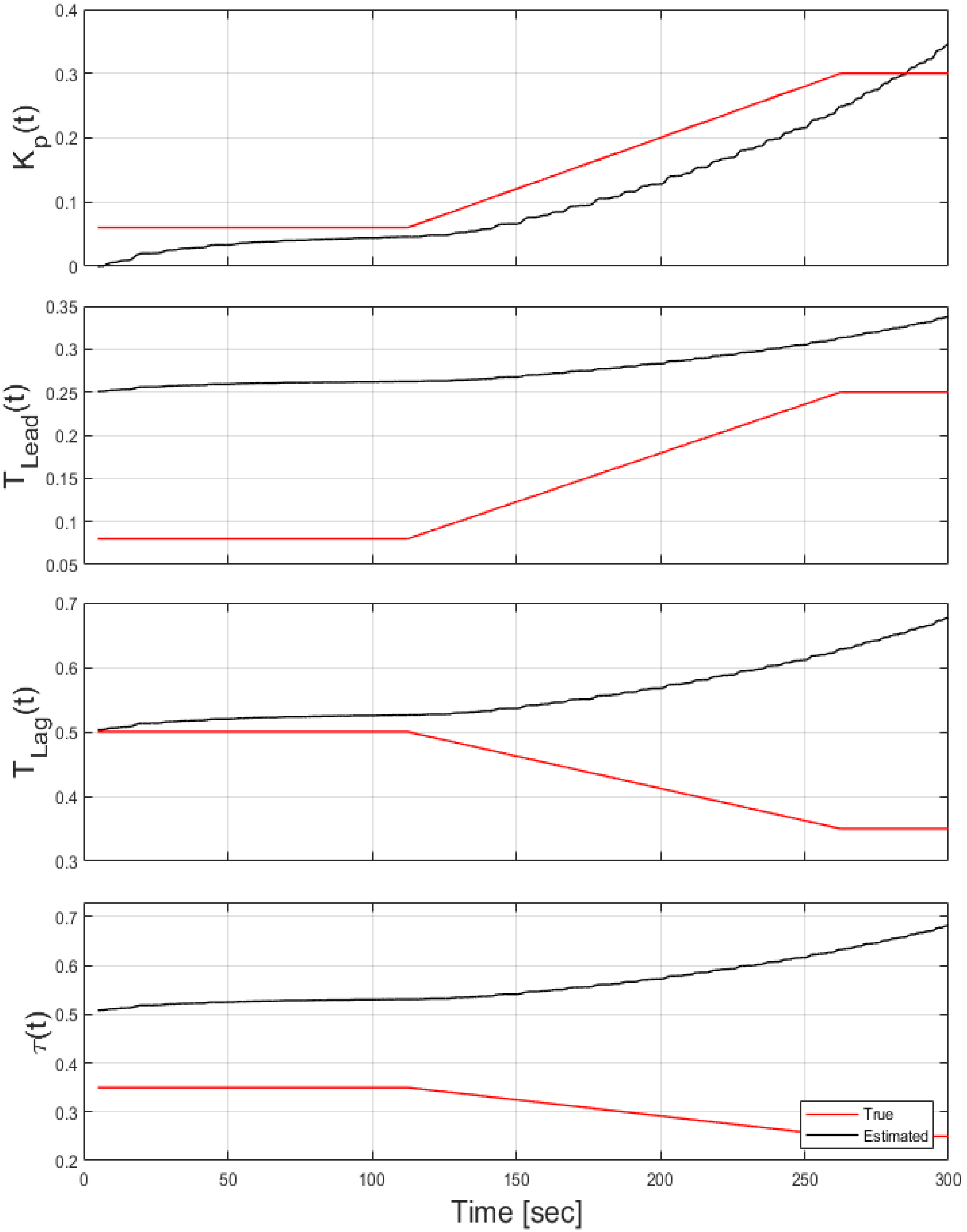


Figure 5.45 Neural Networks Case 2 Results - Pilot model parameter convergence.

Figures 5.43 and 5.44 show the resulting pilot and pitch outputs, respectively. While the pilot model parameters are not well tracked, the trend is seemingly followed and the pilot output is tracking fairly well. The bode plots offer no form of comparison due to the high magnitude of difference between the true parameters and the estimated ones.

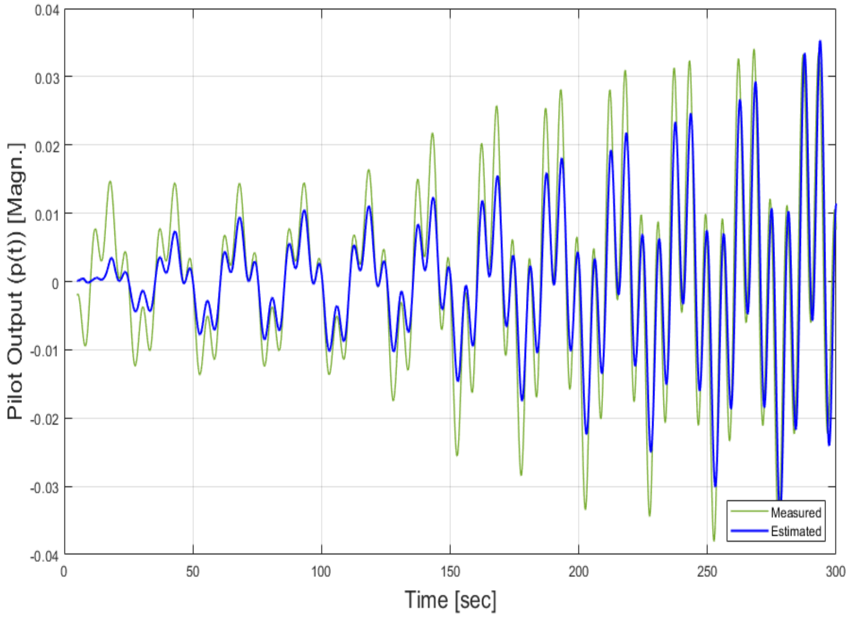


Figure 5.46 Neural Networks Case 2 Results - Pilot model output convergence.

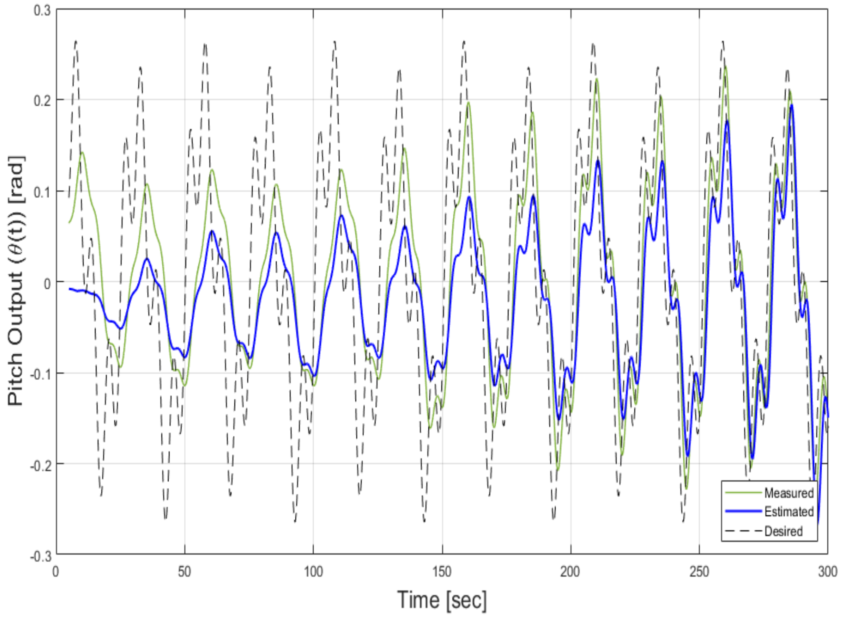


Figure 5.47 Neural Networks Case 2 Results - Pitch model output convergence.

CASE 3: Turning to a more realistic input for the pitch system, the initial condition for the error gain, K_e, is 1.00. The η_V, η_W, and η_P values are learning rates and each is set to 0.36. The parameters estimated as shown in Figure 5.48.

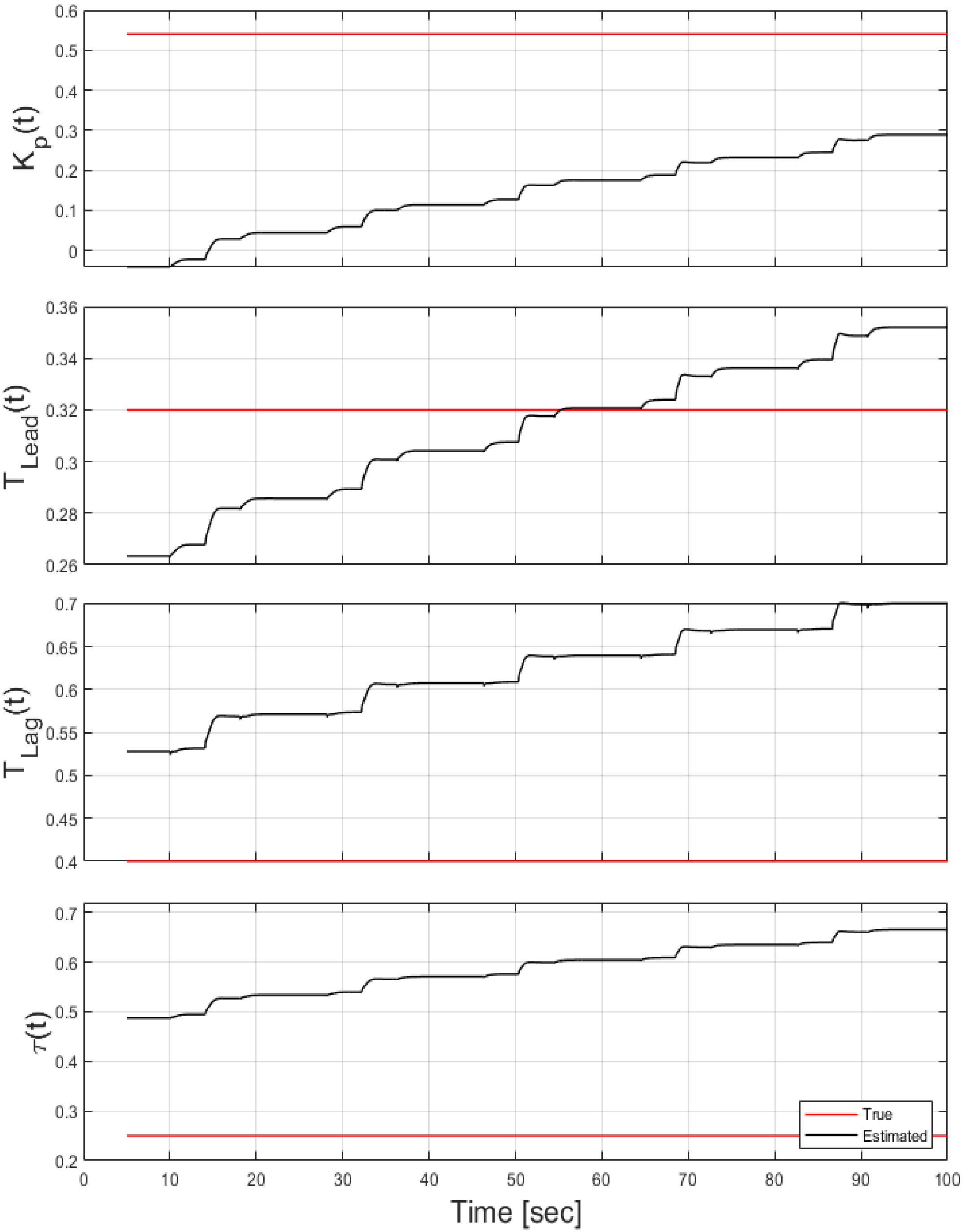


Figure 5.48 Neural Networks Case 3 Results - Pilot model parameter convergence.

Figures 5.49 and 5.50 show the resulting pilot and pitch outputs, respectively. The resulting pilot model output slowly converges to the true solution and the pitch system depicts that the trend of the pitch output is being matched, but the correct magnitude has not been achieved.

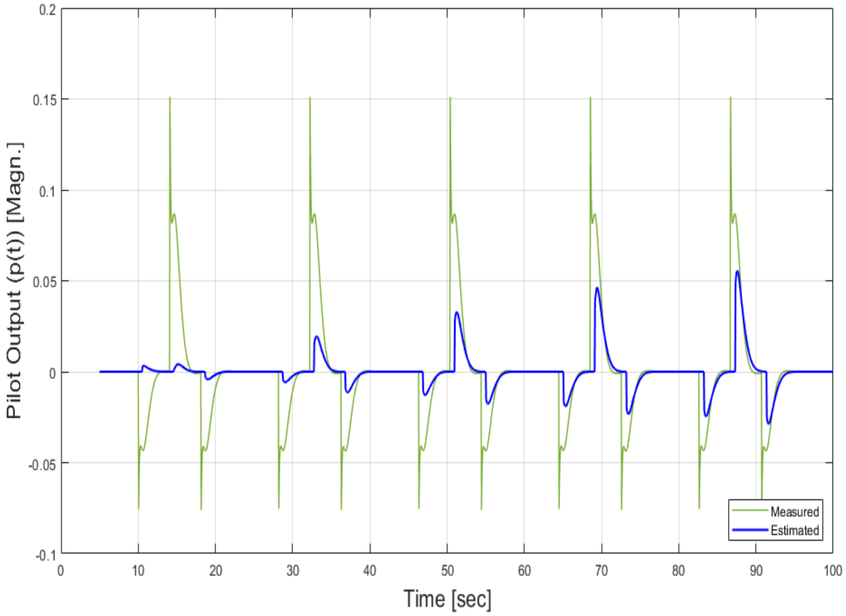


Figure 5.49 Neural Networks Case 3 Results - Pilot model output convergence.

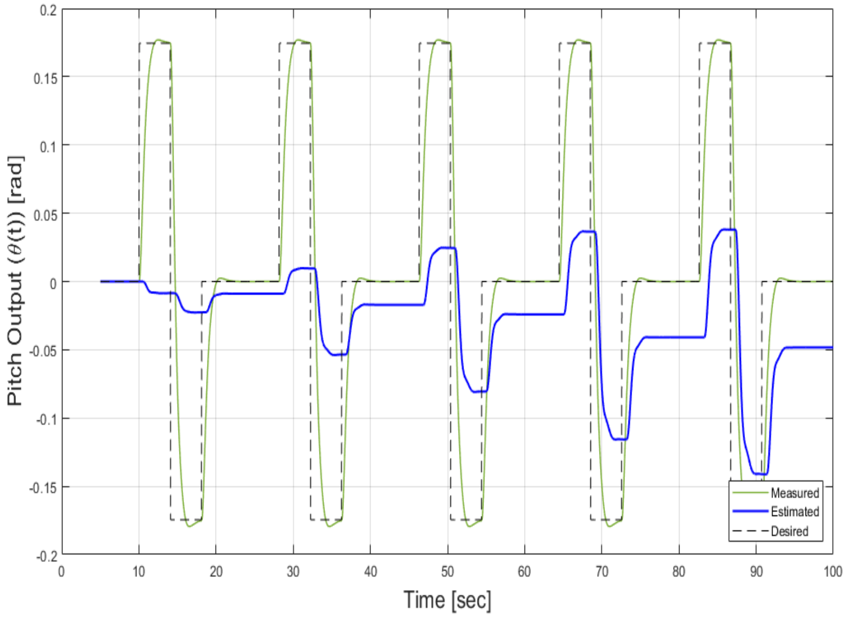


Figure 5.50 Neural Networks Case 3 Results - Pitch model output convergence.

CASE 4: Finally, applying variation in time to the pilot model parameters, the initial condition for the error gain, K_e, is 7.0. The η_V, η_W, and η_P values are learning rates and each is set to 0.3. The estimated parameters are shown in Figure 5.51.

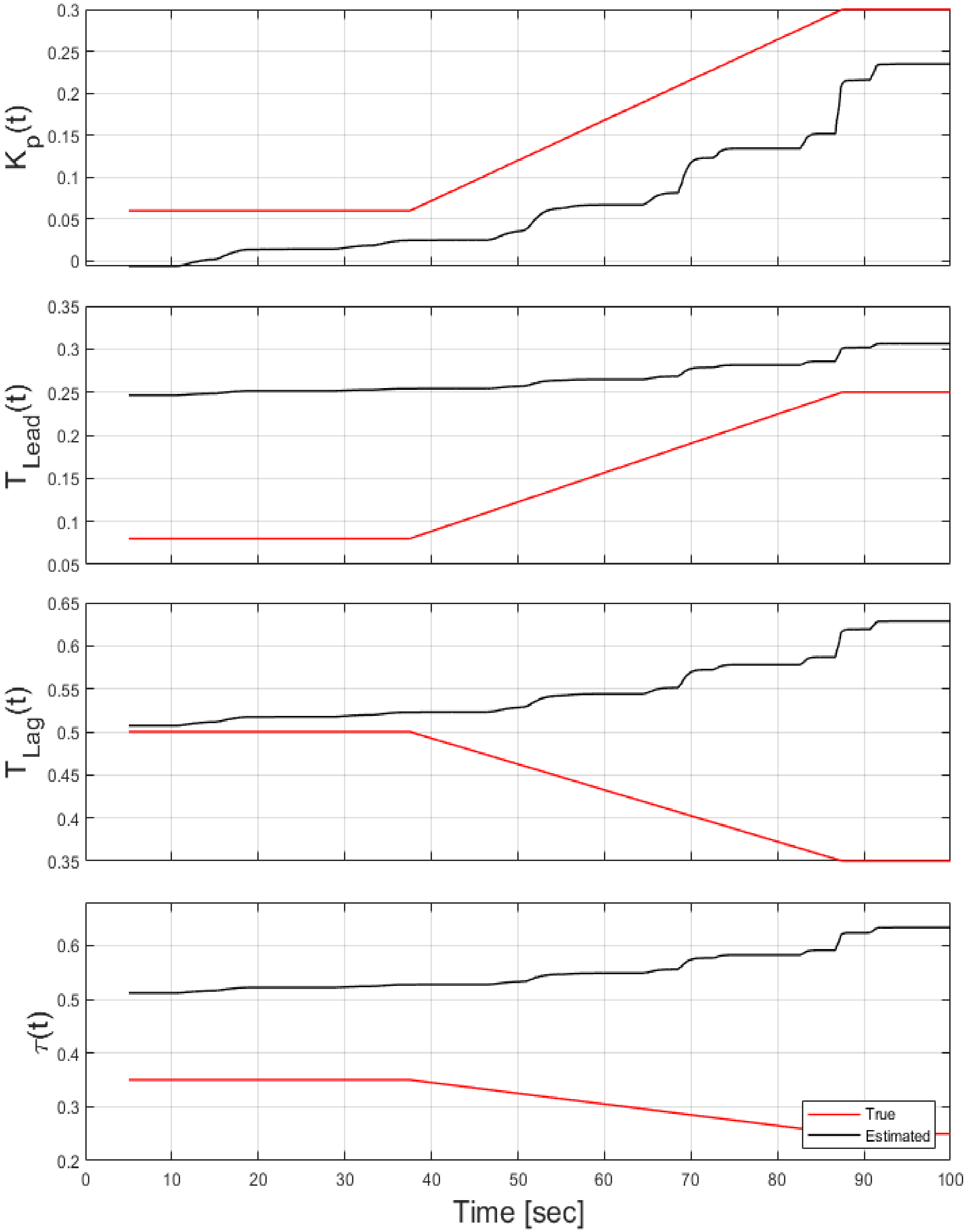


Figure 5.51 Neural Networks Case 4 Results - Pilot model parameter convergence.

Figures 5.52 and 5.53 show the resulting pilot and pitch outputs, respectively. The neural network performs more poorly with lower frequency inputs.

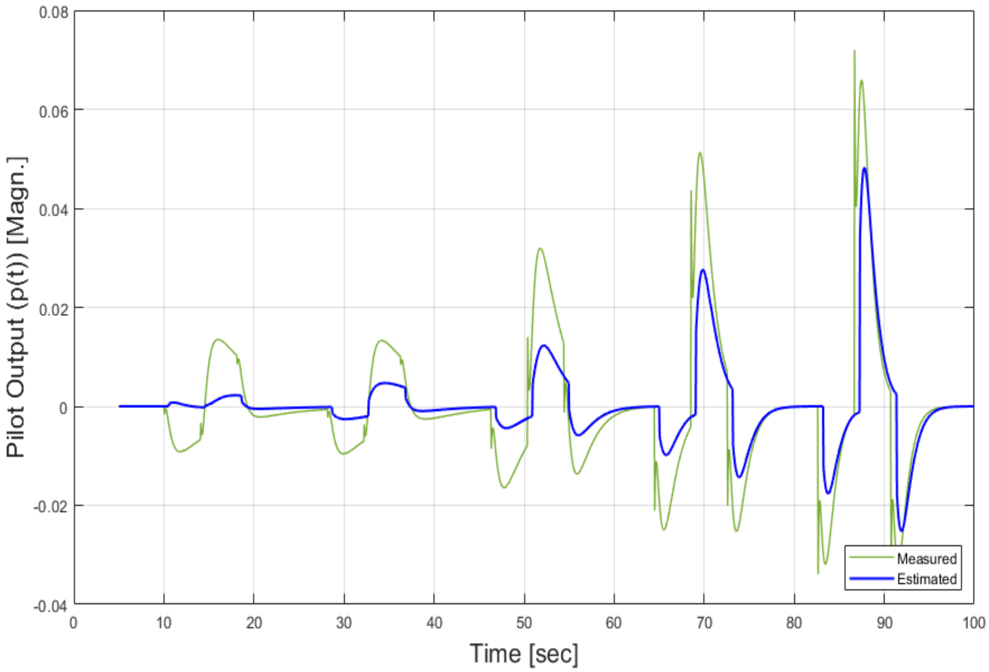


Figure 5.52 Neural Networks Case 4 Results - Pilot model output convergence.

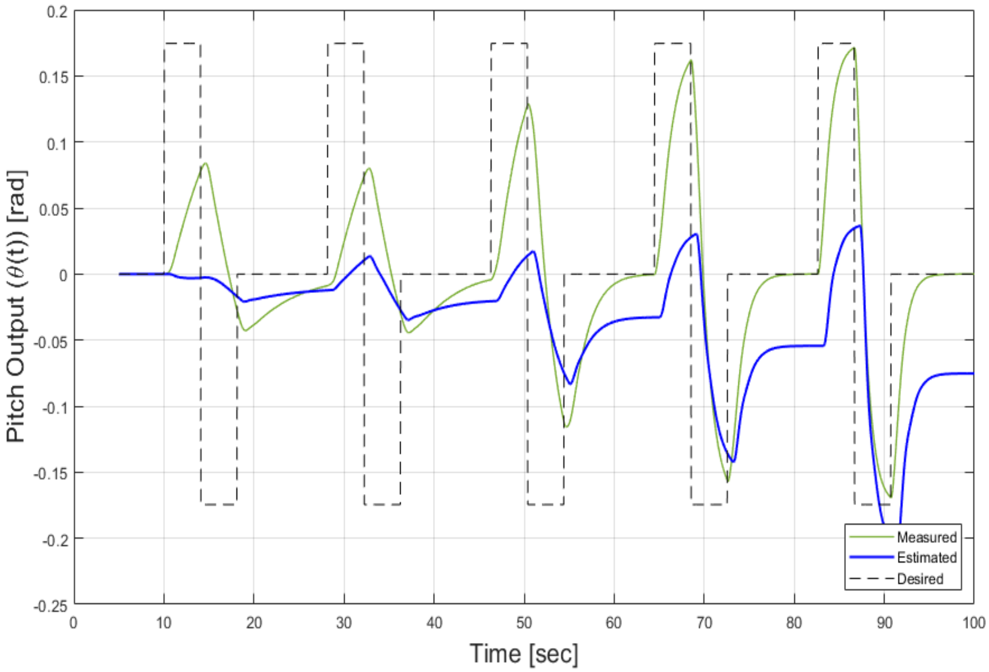


Figure 5.53 Neural Networks Case 4 Results - Pitch model output convergence.

5.4.1 Neural Networks Results Discussion

Being a stochastic estimation method, the neural network used in this problem has difficulty with the highly variable system dynamics presented. Much of what could be adjusted focused around the error of the secondary system which meant that the same error was used to update all four parameters. There was no direct method to tell the network how adjusting individual parameters could affect the results. Therefore many of the trends seen in one parameter are also seen in all others.

As stated before, the pitch output drifts away from the true values considerably throughout the estimation. This is partially due to the exclusion of the error in the pitch angle in updating the weights of the network. However, it was found that including that error had a greater potential for singularities in the estimation solution. To summarize the results from all of the cases, Table 5.5 details the mean-squared error for all cases and all parameters.

Table 5.5 Neural Networks Case Results: Table Summary of Mean Squared Errors

	Mean-Squared Error			
Parameter	Case 1	Case 2	Case 3	Case 4
K_p	0.0581	0.0021	0.1214	0.0052
T_{Lead}	0.0021	0.0184	0.0045	0.0169
T_{Lag}	0.0885	0.0236	0.0920	0.0211
τ	0.1940	0.0675	0.1680	0.0641
$p(t)$	$2.42x10^{-4}$	$3.63x10^{-5}$	0.0015	$9.63x10^{-5}$
$\theta(t)$	0.0051	0.0016	0.0085	0.0030

6 Conclusions and Future Work

Altogether, the methods presented are capable of estimating a mathematical model of the pilot when tailored to particular cases, but they cannot be applicable for robust implementation. The pilot model parameter variation problem holds many difficulties. The high nonlinearity and unpredictability of the pilot model means that an estimator which is not robust in simulation will not be able to perform well in a real world environment. Each of the methods discussed have strengths and weaknesses which help or hurt their viability in aircraft systems.

The Unscented Kalman Filter adaptation handles noise well assuming that the assumptions are correct, but the process noise covariance has to be continuously updated for each individual problem in order to converge to a true solution. The extensions of UKF are limited when one considers the fineness of the tuning required in the Q-covariance matrix in order to ensure convergence. While the UKF results turned out to be the best overall with the lowest mean squared error between all four cases discussed, the tuning required to reach that magnitude of error was extensive.

The Fourier Transform Regression with Time Domain Derivatives (FTR-TD) only works consistently when the parameters are constant. Given that this is a batch method, it is expected that the system is unable to discern a feasible model when the parameters it is estimating are changing with time. It is theorized that, with pilot-in-the-loop (PIL) simulation, the parameters could vary greatly with time. This leaves batch methods and frequency domain methods such as this one with great difficulty and inability to perform well. The tendency for singularities from the added equation solving step also reduces the algorithm's applicability to this problem.

Lastly, the Adaptive Neural Networks have potential but inability to perform well in their current state. Since they are an open stochastic estimation of the parameters, there are many adjustments that can be made to reach a viable solution. In all cases shown, even when the pilot model parameters had significant error, the pilot output still reasonably tracked the trend of the true values. This means that prediction of the pilot's future actions may be achieved using this algorithm without needing convergence of the parameters themselves. The parameters could instead reach a non-optimal solution while matching the pilot output.

6.1 Future Work

The application of these methods individually shows the start of promising solutions. In particular, between the UKF and NNs, the strengths of one algorithm are the weaknesses of the other. Neural networks are effective at tracking the trend of the pilot model output while the Unscented Kalman Filters are able to better filter out the noise and track the magnitude. In the future, it would be prudent to combine these algorithms and either use neural networks to update the Q-matrix, update the weighting of the neural network with a UKF, or augment the solution of the UKF with a neural network. Any of these options could present promising alternatives to using each of the algorithms individually.

On the other hand, one viable option to reach a more expandable solution to this problem is the application of deep machine learning or reinforcement learning. A deep learning algorithm could learn how a pilot's parameters affect their performance and learn to model and predict the actions of the pilot in real time.

Another future benefit will be the estimation of a bank of pilot profile data. Pilot model parameters which classify an attentive, alert pilot can be documented while also considering how the parameters are affected for a tired or distracted pilot. Given that these parameters can be reliably estimated in the future, the database this can produce could help simulation technologies produce more reliable Pilot-in-the-Loop flight environments without needing to extend a significant amount of expense to pilot testing in a simulator.

This also leads into the idea of predicting the onset of Pilot Induced Oscillations (PIOs). Once these methods or some altered form have matured and the pilot's actions can be readily predicted, the onset of dangerous flight scenarios or Loss-of-Control in-flight (LOC-I) can be detected and displayed as a warning to the pilot. Prevention of these scenarios will help mitigate many fatal aviation accidents for years to come.

REFERENCES

[1] Foundation, F. F. S., "Assumptions Used in the Safety Assessment Process and the Effects of Multiple Alerts and Indications on Pilot Performance," `https://flightsafety.org/asw-article/totally-relevant/`, 2015.

[2] Zaal, P., and Sweet, B., "Estimation of Time-Varying Pilot Model Parameters," *Modeling and Simulation Technologies Conference*, 2011.

[3] Gómez-Aguilar JF, C.-R. C. C.-O. I. E.-J. R. O.-P. V., Yépez-Martínez H, "Modeling of a Mass-Spring-Damper System by Fractional Derivatives with and without a Singular Kernel," 2015.

[4] Quesada, A. A., "Five Algorithms to Train a Neural Network," `https://www.neuraldesigner.com/blog/5_algorithms_to_train_a_neural_network#Levenberg-Marquardt`, 2022.

[5] McRuer, D., "Human Pilot Dynamics in Compensatory Systems. Theory, Models and Experiments with Controlled Element and Forcing Function Variations," 1965.

[6] McRuer, D., and Jex, H., "A Review of Quasi-Linear Pilot Models," *Human Factors in Electronics*, IEEE, 1967, pp. 231–249.

[7] FAA, "Fly Safe: Prevent Loss of Control Accidents," `https://www.faa.gov/newsroom/fly-safe-prevent-loss-control-accidents-0`, 2015.

[8] NTSB, "Prevent Loss of Control in Flight in General Aviation," `https://www.ntsb.gov/Advocacy/mwl/Documents/MWL_2016_factsheet06.pdf`, 2016.

[9] Mandal, T. K., and Gu, Y., "Pilot-Vehicle System Modeling Using Sub-Scale Flight Experiments," *Modeling and Simulation Technologies Conference*, 2014.

[10] Mandal, T. K., "Real-Time Monitoring and Prediction of the Pilot Vehicle System (PVS) Closed-Loop Stability," Ph.D. dissertation, West Virginia University, 2016.

[11] Phillips, K. B., “Aircraft parameter identification for application within a fault-tolerant flight control system,” Ph.D. dissertation, West Virginia University, 2011.

[12] et al, M. G. P., “On-Line Parameter Estimation Issues for the NASA IFCS F-15 Fault Tolerant Systems,” n.d.

[13] McRuer, D. T., and Krendel, E. S., “Mathematical Models of Human Pilot Behavior,” *Advisory Group For Aerospace Research Development*, Vol. 107, No. 188, 1974, pp. 22–30.

[14] Campa, G., “Mathworks File Exchange ANN - Adaptive Neural Networks,” `https://www.mathworks.com/matlabcentral/fileexchange/976-ann`, 2022.

[15] Mandal, T. K., and Gu, Y., “Online Pilot Model Parameter Estimation Using Sub-Scale Aircraft Flight Data,” *Guidance, Navigation, and Control Conference*, 2016.