A Predictive Autorotation Entry Analysis Using Bayesian Multi-Model Estimation Detection

by

Justin N. Martin

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
December 16, 2017

Keywords: Autorotation, Bayes' Theorem, Multi-Model Estimation

Approved by

Andrew J. Sinclair, Chair, Affiliates Associate Professor of Aerospace Engineering
John E. Cochran, Jr., Professor of Aerospace Engineering, Emeritus
Winfred A. Foster, Jr., Professor of Aerospace Engineering, Emeritus
David A. Cicci, Professor of Aerospace Engineering
George T. Flowers, Dean of Graduate School

Abstract

Improvements to achieving helicopter autorotation in an event of engine or driveshaft failure is an issue that has and always will be present in the engineering landscape. Questions include, “how much of a performance hit can be taken for the proper autorotation tools?” and “how much control should the pilot initially have in the event of an engine failure?” One certain aspect is that upon an engine failure in a rotorcraft, every second counts.

A prediction of future behavior of a rotorcraft can be estimated by feeding observational flight data into comparable dynamic and engine analysis models. The comparison of different hypothesis models is combined into a Bayes’ Multi-Model Estimation to evaluate the health of the rotorcraft. Interesting concepts through this work are (a) the creation of a coupled dynamic and engine model that is iterated into failure models for detecting a risk, (b) the methodology for using multiple models for reducing false positives in engine failure detection, and (c) the magnitude of the change in pilot recognition time.

The prediction of a failure in the rotorcraft model can lead to new contributions of methods into rotorcraft autorotation detection. The use of live observation data to project the state and health of the rotorcraft at a future time has been shown to be valid. The failure models were to be generated by reasonably changing states and controls in the normal functioning model. As predicted, the aircraft altitude saved due to the significant

reduction in pilot recognition time is essential for increasing the success of an autorotation landing.

Acknowledgments

The author would like to thank and appreciate his academic advisor, Dr. Andrew J. Sinclair. His leadership and guidance early in the author's graduate school career were beneficial in the completion of this dissertation. The author would to also recognize Dr. Marvin Moulton of the U.S. Army, Aerospace Engineering Directorate. Under his leadership within AED, Dr. Moulton provided the author with the guidance, base education of helicopter knowledge, and tools to investigate the topic of this dissertation.

The author would like to personally thank his wife and son, Leigh Ann and Michael Martin, along with his sister, parents, and in-laws for supporting him in every step of the doctorate program. Their love for him is the driving force he needed to complete this study.

Finally, the author would like to thank all of those who have dedicated themselves to serving in the Armed Forces and pray for those who have lost their lives doing the same. Whether this work ultimately affects Army helicopter safety or not, the author urges readers of this paper to assist the war fighter when possible.

Table of Contents

List of Figures

List of Tables

List of Variables

Variables by Letter

a_0	Main Rotor Blade Lift-Curve Slope
a_{0T}	Tail Rotor Blade Lift-Curve Slope
b	Number of Main Rotor Blades
BLCG	Buttline Center of Gravity Position
C_Q	Main Rotor Torque Coefficient
C_{TT}	Tail Rotor Thrust Coefficient
C_x, C_y, C_z	Main Rotor Force Coefficients in Shaft-Axis x-y-z
C_{ZTP}	Normalized Tailplane Force
D, E, F, G, Z	Simplified Engine Model Coefficient Matrices
$F_0^{(1)}$, $F_{1s}^{(1)}$, $F_{1c}^{(1)}$, $F_{2s}^{(1)}$,$F_{2c}^{(1)}$, $F_{1s}^{(2)}$, $F_{1c}^{(2)}$	Harmonic Comp. of Integrated Blade Aerodynamic Loads
FSCG	Fuselage Center of Gravity Position
F_T	Tail Rotor Fin Blockage Factor
g	Gravity Constant (32.174 ft/s^2)
GW	Rotorcraft Gross Weight
h_p	Pressure Altitude
h_r	Height of Main Rotor Hub Above Fuselage Ref. Point
h_T	Height of Tail Rotor Hub Above Fuselage Ref. Point
I_β	Flap Moment of Inertia
I_R	Main Rotor Moment of Inertia
I_{xx}, I_{yy}, I_{zz}	Moments of Inertia
I_{xz}	Products of Inertia
K_β	Flapping Spring at Hinge
L, M, N	External Moments on Aircraft in Body x-y-z

l_T	Tail Rotor Hub Distance Aft from Fuselage Ref. Point
l_{TP}	Tail Center of Pressure Aft from Fuselage Ref. Point
M_a	Mass of the Aircraft (683.782 slugs)
N_g	Gas Generator Speed
N_p	Power Turbine Speed
p, q, r	Aircraft Angular Rates in Body x-y-z
P_{S3}	Static Pressure at Station 3
Q_S	Engine Shaft Torque
R	Main Rotor Radius
R_T	Tail Rotor Radius
s	Main Rotor Solidity
s_T	Tail Rotor Solidity
S_{TP}	Tail Rotor Plane Area
T_{45}	Temperature at Station 4.5 (TGT)
u, v, w	Aircraft Velocity in Body x-y-z
u(k)	Engine Model Control Vector
V_T	Tangential Velocity of Tail Rotor
w_f	Engine Fuel Flow
WLCG	Waterline Center of Gravity Position
X, Y, Z	External Forces on Aircraft in Body x-y-z
x(k)	Engine Model State Vector
x(k+1)	Engine Model State Vector Advanced One Step
x_{cg}	Center of Gravity Position Forward of Fuselage Ref. Point
y(k)	Engine Model Observation Vector

Variables by Symbol

α_{TP}	Incidence for Tailplane
β_0	Main Rotor Coning Angle
β_{1cw}	1st Harmonic Cyclic Flapping Angle, Longitudinal
β_{1sw}	1st Harmonic Cyclic Flapping Angle, Lateral

γ	Lock Number
γ_s	Main Rotor Shaft Forward Tilt
κ	Non-uniform Flow and Tip Loss Factor
λ_0	Total Inflow Across Rotor Disk
λ_{1cw}	1st Harmonic Inflow Component of Cosine
λ_{1sw}	1st Harmonic Inflow Component of Sine
Ω	Main Rotor Speed
Ω_T	Tail Rotor Speed
ϕ, θ, ψ	Euler Angles
ρ	Air Density (0.0023769 slugs/ft^3)
θ_0	Main Rotor Collective Pitch
θ_{1cw}	Total Lateral Swashplate Angle
θ_{1sw}	Total Longitudinal Swashplate Angle
θ_{tw}	Tail Blade Twist
μ	Rotor Tip Speed Ratio, In-Plane
μ_z	Rotor Tip Speed Ratio, Normal to Plane

1. Introduction

Rotorcraft function by driving the main rotor(s) to push air down, and when combined with rotor blade lift, the rotorcraft can fly. In an autorotation, when either torque is no longer supplied to the main rotor or the angle of attack on the rotor blades is low, the aircraft descends and the air flow through the rotor changes direction to move upward through the rotor. The air flowing through the rotor still drives the rotor blades to continue turning. The potential energy (due to altitude) is transitioned to kinetic energy (rotor and translational speed). The rotorcraft may still be maneuvered while the rotor speed is maintained near 100%. When an engine fails, a pilot has approximately 2.0 seconds to begin actions to enter an autorotation. If the entry is delayed, the rotor speed may decay to a point that is unrecoverable given the altitude of the aircraft and time required to recover the rotor speed. An automated process could avoid this decay in rotor speed and allow a pilot more time to locate and maneuver to a safe landing site.

Autorotation can be broken into three basic phases; entry consisting of the first 3-5 seconds, descent, and flare/landing. The dangers in an engine failure are loss of aircraft control due to degradation of main rotor speed, acquisition of a safe landing site and maneuvering to the landing site with enough rotor energy to execute a proper autorotation landing.

Therefore, the air flowing upward through the rotor mixed with the airflow from the aircraft’s forward speed is the only means of driving the rotor blades to continue their motion. As shown in Figure 1, the upward flow turning the main rotor allows the rotorcraft to maneuver into a landing position and complete a controlled landing.

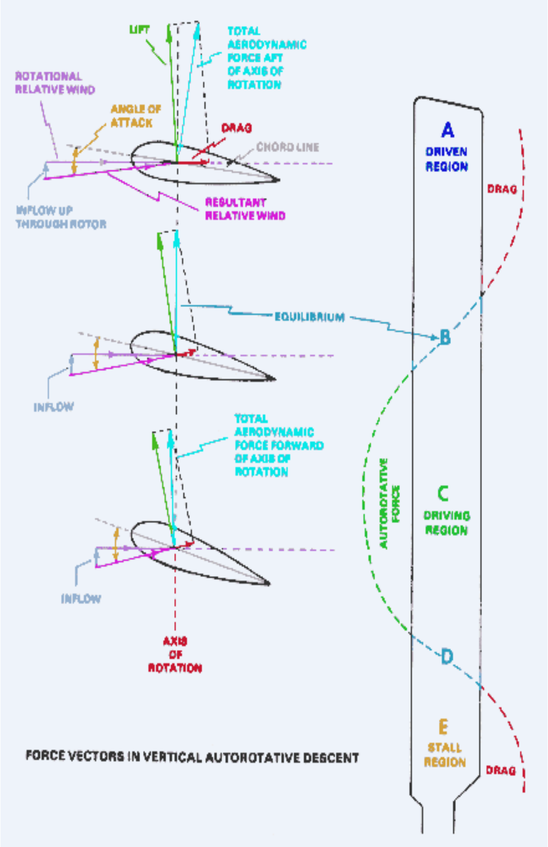


Figure 1: Rotor Force Vectors in Autorotation

Reprinted with permission from *Dynamic Flight, Inc.*, by M. J. Zeldes, 2017, Retrieved from www.dynamicflight.com/aerodynamics/autos/, Copyright 2010 by Dynamic Flight, Inc.

The diagram is an example of aerodynamic forces present on a rotor blade in autorotation. “Section A” shows the rotor tip with the resultant aerodynamic vector broken into the lift and drag vectors. The drag vector component along the tip path plane is greater than the lift vector component along the tip path plane. This results in a force working against the rotation of a torque-less rotor. The increase in total drag within the resultant vector is primarily due to the increase in speed (v = ω x r), and possibly due to twist along the span of the rotor blade increasing the drag coefficient. “Section C” is the focus of autorotation flight. If the angle of attack is reduced due to a collective stick decrease, the resultant force tilts in front of the axis of rotation. This creates a driving force causing the rotor to maintain rotational energy. The reduction of angle of attack results in the lift force component in the tip path plane being greater than the drag force component in the tip path plane. “Section E” is another area where the resultant force works against the motion of the rotor. This is caused by the very low speeds in this region. However, this region does not significantly contribute to the rotor lift or drag. The goal for a pilot is to set the collective input to a position where the total driving force is greater or equal to the total driven force.

It is debatable what the most important aspect of an autorotation is the reaction time immediately following an engine failure or the descent to a safe landing zone. Pilots are permitted approximately 2.0 seconds of reaction time immediately following an engine failure to begin recovery of the rotorcraft. Pilot models to represent these actions are intended to account for sensorial perceptions, control behaviors and command actuations [1].

From a pilot’s point of view, an autorotation is as simple as recognizing the direction air is flowing through the main rotor. However, from the early stages of pilot development in training scenarios containing known malfunctions to the later stages on battlefields and “routine” flight plan assignments, engine malfunctions followed by a poorly performed autorotation can cost the salvation of an aircraft and the lives of its crew members.

A primary failure for pilots occurs within the first 3-5 seconds of an engine failure. Physically, the engine ceases to provide torque to the rotor shaft and rotor gear boxes. With no torque being provided, drag on the main rotor blades begin to quickly decrease the rotor speed, otherwise known as rotor droop. The more collective stick applied by the pilot, the greater the angle of attack of each rotor blade causing an increase in the airfoil drag, decelerating the main rotor at a faster rate. The indicators provided to the pilot (besides cockpit bells and whistles), are in the dynamic and engine response to the reduction in power. The rotorcraft will nose down because the utility helicopter's center of gravity is aft of the main rotor position [2], yaw left due to the decrease in torque applied to the main rotor, an audible main rotor speed decrease and change in engine noise, and either loss or altitude or pilot addition of collective stick [3]. The U.S. Army has also indicated that a pilot is not to respond to the aircraft main rotor audio or multi-function/multi-purpose displays alone [3]. The U.S. Army has a mission goal for pilots to be able to decipher all the provided indicators, sort through the cockpit alerts, analyze their current flight state, and develop a plan of action, all within 2.0 seconds [4]. In some cases, that time threshold is not met, pilots continue to make errors trying to protect their aircraft and crew, and in turn, the aircraft is lost.

Previous work has been conducted to prevent the main rotor speed from decaying in the event of an engine failure. The work has ranged from increasing the moment of inertia of the rotor blades by adding masses in various positions, to sliding masses along the rotor blade to increase and decrease inertia, to even applying small rockets at the end of the rotor blades to force the rotor blade to keep spinning [5]. However, improvements proposed for autorotation performance was too great of a cost to the aircraft functionality and performance, effecting the feasibility and basic reasoning for having the aircraft in the first place [5]. An area that can still help maintaining control of a malfunctioning aircraft is by assisting the pilot to ensure the 2.0 second reaction time is upheld. Research has gone as far as giving physical and haptic cues through stick force gradients to reduce this reaction time [6].

With the focus directed to the first 3-5 seconds following an engine failure, a means to reliably and more quickly detect a failure will be developed. The areas of the aircraft to be analyzed are the dynamic responses to the rotorcraft airframe and the thermodynamic responses at sensor areas of the turboshaft engine driving the rotor system. Each model will be treated separately and then combined using a Bayes' Theorem Multiple Model Estimation (MME). The conditional probabilities are calculated, which equates the dependency between random variables to decide on the state of a single variable [7]. The early detection of an engine failure reduces the "pilot" delay time from 2.0 seconds to a moment the math models indicate a failure.

The probability analysis will require a Kalman filter for each model prior to calculating the MME. Once combined, scenarios of various engine failures in various flight states will be evaluated and reported. The groundwork will be in place to apply a software code to an aircraft in order detect a probable autorotation scenario at a pace faster than a pilot.

Chapter 2 will introduce and couple an 8-state dynamic model with a 2-state engine model. The dynamic model will consist of three body velocities, three body angular velocities, and two angular orientations using the swashplate positions as the control. The engine model defines the engine generator speed and power turbine speed with the fuel flow serving as the control. The two models will be coupled together using the mechanical gear ratio between the power turbine speed and main rotor speed.

Chapter 3 will proceed to evaluate the quality of the models used to evaluate the Bayesian estimation along with determining the estimation sensitivities due to external forces such as turbulence and internal phenomena such as instrument error. The evaluation will be over the full (10x10) normal model. Engine verification will also be completed to ensure the system is appropriate to couple with the dynamic model. The second half of Chapter 3 will be used to record and analyze the results of the Bayesian estimation. The estimation will lead to pilot reaction time analysis, and concluded with a calculation of the difference in altitude loss based upon the various pilot reaction times, Bayesian and delayed.

2. Evaluation Math Models

As described in future sections, the Bayes' Analysis will compare multiple weighted prediction models to determine the quickest means of verifying that an engine failure has occurred and the pilot should enter an autorotation to prepare for landing. The two prediction models which are capable of being modeled are the dynamic model and the engine model. The dynamic model uses the rate changes and orientation changes the helicopter undergoes to predict an engine failure. The engine model examines key parameters that are capable of being measured to predict a failure. This chapter discusses the development of the two models, the foundation of the Bayes' analysis for predicting failure, and the integration into the FLIGHTLAB® and VirtualPilot environment. FLIGHTLAB® is the component-based modeling and simulation software developed by ART, Inc. FLIGHTLAB® applies built-in solvers to progress a rotorcraft simulation model consisting of dynamic, aerodynamic, propulsion, and control systems through timesteps for model development and analysis. The VirtualPilot is an Army AED developed product which applies either flight test data in the form of stick inputs or state responses or user prescribed maneuvers to calculate and apply stick control positions to drive the rotorcraft simulation.

2.1. The Dynamic Helicopter Model

In the event of an engine failure, whether it be engine-related or aircraft damage related such as gear box or drive shaft failure, the aircraft will lose energy. The energy can be in the form of potential energy as the helicopter begins a descent, or the form of kinetic energy as the rotor speed begins to droop. Once the rotor speed begins to droop from its initial trimmed state, the aircraft position, speed, and orientation change. Using a dynamic model of the helicopter, the future states of the aircraft can be estimated and provide breadcrumbs for determining an engine failure.

The utility helicopter model is based from the UH-60 model described in work associated with the U.S. Army. For this model, NACA 0012 airfoils were applied to the aircraft main rotor. The tail rotor is modeled as a rotor disk and not with individual airfoils. The physical configuration for the utility helicopter is described in Table 1.

Variable	Value	Model Source
M_a	683.782 slugs	Aircraft Config.
I_{XX}	7344.17 slug-ft^2	Aircraft Config.
I_{YY}	46372.17 slug-ft^2	Aircraft Config.
I_{ZZ}	43384.33 slug-ft^2	Aircraft Config.
I_{XZ}	1661.08 slug-ft^2	Aircraft Config.
R	26.8 feet	FLIGHTLAB
R_T	5.5 feet	FLIGHTLAB
x_{cg}	360.0 inches	FLIGHTLAB

Table 1: Utility Helicopter Configuration

The described UH-60 configuration is to be used for every FLIGHTLAB® test. The mass characteristics will remain constant. An assumption that is to also be made is no fuel weight is to be burned and expelled in a simulation. Therefore, the UH-60 mass, inertias, and CG's will remain constant throughout.

2.1.1. Aircraft Modeling Configuration

The configuration of the utility helicopter model will remain consistent throughout the simulation and analysis of the engine failure and detection. Table 2 defines the values assigned to the variables in each equation of motion along with a substantiation report as to where the values originated. The last part is crucial for documentation, as values may have been assigned values from reference [8], the FLIGHTLAB® initial variables for a utility helicopter configuration, an updating variable linked to the simulation model at each time step, or simply calculated with an equation. Table 2 will cover the variables for all three sections covering the aircraft dynamic equations of motion.

Variable	Value	Model Source
a_0	5.73	Thanapalan Ref.
a_{0T}	5.73	Thanapalan Ref.
C_Q	rotor1_rotor_cpg_xaout_cq	FLIGHTLAB
C_{TT}	rotor2_rotor_cpg_xaout_ct	FLIGHTLAB
C_{ZTP}	0	Assumption
F_T	-0.402	Thanapalan Ref.
h_r	5.5	Aircraft Config.
h_T	6.3150	Aircraft Config.
I_β	rotor1_rotor_data_bsm	FLIGHTLAB
I_R	rotor1_rotor_data_bsm	FLIGHTLAB
K_β	rotor1_rotor_data_flapkk	FLIGHTLAB
l_T	31.17	Aircraft Config.
l_{TP}	0	Assumption
S	0.08210	Thanapalan Ref.
s_T	0.79	Aircraft Config.
S_{TP}	0	Assumption
V_T	0	Assumption
α_{TP}	0	Assumption
β_0	rotor1_rotor_cpg_xaout_a0f	FLIGHTLAB
β_{1cw}	rotor1_rotor_cpg_xaout_b1f	FLIGHTLAB
β_{1sw}	rotor1_rotor_cpg_xaout_a1f	FLIGHTLAB
γ_s	0.0523	Thanapalan Ref.
λ_0	rotor1_rotor_cpg_xaout_lambdai	FLIGHTLAB
θ_0	rotor1_rotor_cpg_xaout_theta0	FLIGHTLAB
θ_{1cw}	rotor1_rotor_cpg_xaout_a1s	FLIGHTLAB
θ_{1sw}	rotor1_rotor_cpg_xaout_b1s	FLIGHTLAB
θ_{tw}	-0.314	Thanapalan Ref.
μ	airframe_cpg_xaout_vxb / (Rω)	FLIGHTLAB
μ_z	airframe_cpg_xaout_vzb / (Rω)	FLIGHTLAB

Table 2: Equations of Motion Parameter Source

For determining both the translational and rotational velocity components, a few assumptions were made in addition to those chosen by Thanapalan [8]. First, the aircraft model is commanded to fly in coordinated flight, therefore the sideslip angle is zero degrees for straight and curved flight. Second, the tail rotor is modeled as a disk rotor, making the blade lift curve slope zero. Third, the angle of attack for the fuselage is assumed to be zero degrees.

2.1.2. Aircraft Velocity Components

The first set of equations of motion are for the aircraft velocity through the center of gravity. Note that the speeds are all body-fixed variables, where *u* is out the nose of the aircraft, *v* is out the starboard side of the aircraft, and *w* is out the bottom of the aircraft.

As defined in Aponso [9], the body-fixed velocity rate components have been described as:

$$\dot{u} = rv - qw - g\sin\theta + X / M_a \tag{1}$$

$$\dot{v} = pw - ru + g\cos\theta\sin\phi + Y / M_a \tag{2}$$

$$\dot{w} = qu - pv + g\cos\theta\cos\phi + Z / M_a \tag{3}$$

The velocity components consist of a coupled relationship between translational and angular speeds, the contribution of gravity, and external forces due primarily to aerodynamic forces applied to the fuselage and rotor. Those aerodynamic forces, as defined in Padfield [10] are:

$$X = \frac{1}{2}\rho(\Omega R)^2 \pi R^2 a_0 s \cos\gamma_s \, \frac{2C_X}{a_0 s} - \frac{1}{2}\rho(\Omega R)^2 \pi R^2 a_0 s \sin\gamma_s \, \frac{2C_Z}{a_0 s} \tag{4}$$

$$Y = \frac{1}{2}\rho(\Omega R)^2 \pi R^2 a_0 s \left(\frac{2C_Y}{a_0 s}\right) \tag{5}$$

$$\begin{aligned} Z = {} & \frac{1}{2}\rho(\Omega R)^2 \pi R^2 a_0 s \left(\sin\gamma_s \left(\frac{2C_X}{a_0 s}\right) + \cos\gamma_s \left(\frac{2C_Z}{a_0 s}\right)\right) \\ & + \frac{1}{2}\rho(\Omega R)^2 \bar{V}_T^2 S_{TP} C_{ZTP}(\alpha_{TP}) \end{aligned} \tag{6}$$

C_X, C_Y, and C_Z are the main rotor force coefficients in the shaft axis. Because of the assumption of coordinated flight, shaft axis is also representing the hub-wind axis. The rotor force coefficients are functions of the harmonics of blade aerodynamic loads and flapping. The harmonic components are defined in Equations (7-9).

$$\left(\frac{2C_X}{a_0 s}\right) = \left(\frac{F_0^{(1)}}{2} + \frac{F_{2c}^{(1)}}{4}\right)\beta_{1cw} + \frac{F_{1c}^{(1)}}{2}\beta_0 + \frac{F_{2s}^{(1)}}{4}\beta_{1sw} + \frac{F_{1s}^{(2)}}{2} \tag{7}$$

$$\left(\frac{2C_Y}{a_0 s}\right) = \left(\frac{F_0^{(1)}}{2} + \frac{F_{2c}^{(1)}}{4}\right)\beta_{1sw} - \frac{F_{1s}^{(1)}}{2}\beta_0 - \frac{F_{2s}^{(1)}}{4}\beta_{1cw} + \frac{F_{1c}^{(2)}}{2} \tag{8}$$

$$\left(\frac{2C_Z}{a_0 s}\right) = -\left(\frac{2C_T}{a_0 s}\right) = -F_0^{(1)} \tag{9}$$

where

$$F_0^{(1)} = \theta_0\left(\frac{1}{3}+\frac{\mu^2}{2}\right)+\frac{\mu}{2}\left(\theta_{1sw}+\frac{\bar{p}_w}{2}\right)+\left(\frac{\mu_Z-\lambda_0}{2}\right)+\frac{1}{4}\left(1+\mu^2\right)\theta_{tw} \qquad \textbf{(10)}$$

$$F_{1s}^{(1)} = \frac{\bar{p}-\lambda_{1sw}+\beta_{1cw}+\theta_{1sw}}{3}+\mu\left(\theta_0+\mu_Z-\lambda_0+\frac{2}{3}\theta_{tw}\right) \qquad \textbf{(11)}$$

$$F_{1c}^{(1)} = \frac{\bar{q}-\lambda_{1cw}-\beta_{1sw}+\theta_{1cw}}{3}-\frac{\mu\beta_0}{2} \qquad \textbf{(12)}$$

$$F_{2s}^{(1)} = \frac{\mu}{2}\left(\theta_{1cw}-\beta_{1sw}+\frac{\bar{q}_w-\lambda_{1cw}}{2}-\mu\beta_0\right) \qquad \textbf{(13)}$$

$$F_{2c}^{(1)} = -\frac{\mu}{2}\left(\theta_{1sw}+\beta_{1cw}+\frac{\bar{p}_w-\lambda_{1sw}}{2}+\mu\left(\theta_0+\frac{\theta_{tw}}{2}\right)\right) \qquad \textbf{(14)}$$

$$\begin{aligned}
F_{1s}^{(2)} = {} & \frac{\mu^2}{2}\beta_0\beta_{1sw}+\left(\mu_Z-\lambda_0-\frac{\mu}{4}\beta_{1cw}\right)\left(\bar{p}-\lambda_{1sw}+\beta_{1cw}\right) \\
& -\frac{\mu}{4}\beta_{1sw}\left(\bar{q}-\lambda_{1cw}-\beta_{1sw}\right)+\theta_0\left(\frac{\bar{p}-\lambda_{1sw}+\beta_{1cw}}{3}+\mu\left(\mu_Z-\lambda_0\right)-\frac{\mu^2}{4}\beta_{1cw}\right) \\
& +\left(\frac{\bar{p}-\lambda_{1sw}+\beta_{1cw}}{4}+\frac{\mu}{2}\left(\mu_Z-\lambda_0-\frac{\beta_{1c}\mu}{4}\right)\right)\theta_{tw} \qquad \textbf{(15)} \\
& +\theta_{1sw}\left(\frac{\mu_Z-\lambda_0}{2}+\mu\left(\frac{3}{8}\left(\bar{p}_w-\lambda_{1sw}\right)+\frac{\beta_{1cw}}{4}\right)\right) \\
& +\frac{\mu}{4}\theta_{1cw}\left(\frac{\bar{q}_w-\lambda_{1cw}}{2}-\beta_{1sw}-\mu\beta_0\right)-\frac{\delta\mu}{a_0}
\end{aligned}$$

$$
\begin{aligned}
F_{1c}^{(2)} &= -2\beta_0\mu\left(\mu_Z - \lambda_0 - \frac{4}{3}\mu\beta_{1cw}\right) + \left(\mu_Z - \lambda_0 - \frac{4}{3}\mu\beta_{1cw}\right)\left(\bar{q} - \lambda_{1cw} - \beta_{1sw}\right) \\
&- \frac{\mu}{4}\beta_{1sw}\left(\bar{p} - \lambda_{1sw} + \beta_{1cw}\right) + \theta_{0}\left(\frac{\bar{q} - \lambda_{1cw} - \beta_{1sw}}{3} - \frac{\mu}{2}\left(\beta_0 + \frac{\mu}{2}\beta_{1sw}\right)\right) \\
&+ \theta_{tw}\left(\frac{\bar{q} - \lambda_{1cw} - \beta_{1sw}}{4} - \mu\left(\frac{\beta_0}{3} + \frac{\mu}{8}\beta_{1sw}\right)\right) + \theta_{1cw}\left(\frac{\mu_Z - \lambda_0}{2} + \frac{\mu}{4}\left(\frac{\bar{p}_w - \lambda_{1sw}}{2} - \beta_{1cw}\right)\right) \\
&+ \frac{\mu}{4}\theta_{1sw}\left(\frac{\bar{q}_w - \lambda_{1cw}}{2} - \beta_{1sw} - \mu\beta_0\right)
\end{aligned}
\tag{16}
$$

If the aircraft model were to be simulated, the controls for longitudinal and lateral cyclic and collective stick inputs are input components to the flight control system to produce the resultant collective and cyclic pitch variables applied to the rotorcraft swashplate. However, because the flight control system of the utility rotorcraft used is proprietary, only approximate swashplate angles are to be used. The calculation of stick inputs into an approved flight control system to apply to the swashplate actuators is considered to be beyond the scope of work of this paper. Therefore, the output of the cyclic and collective swashplate angle components from FLIGHTLAB® will suffice.

2.1.3. Aircraft Rotational Rates

Like the translational velocities presented in the previous section, the aircraft rotational rate changes are defined about the center of gravity of the aircraft. The roll, pitch, and yaw rates are defined as p, q, and r, respectively, and are about the body-fixed x, y, and z axis. The body rotational rates are defined as:

$$
\begin{aligned}
\dot{p} &= \frac{\left(I_{yy} - I_{zz}\right)}{I_{xx}} qr + \frac{I_{xz}\left(I_{xx}^2 - I_{xx} I_{yy} + I_{xz}^2\right) pq + I_{xz}^2 \left(I_{yy} - I_{zz} - I_{xx}\right) qr}{I_{xx}\left(I_{xx} I_{zz} - I_{xz}^2\right)} \\
&+ \frac{I_{xz}^2 L + I_{xx} I_{xz} N}{I_{xx}\left(I_{xx} I_{zz} - I_{xz}^2\right)} + \frac{I_{xz}}{I_{xx}} pq + \frac{L}{I_{xx}}
\end{aligned}
\tag{17}
$$

$$
\dot{q} = \frac{1}{I_{yy}} \left(\left(I_{zz} - I_{xx}\right) pr - I_{xz}\left(p^2 - r^2\right)\right) + \frac{1}{I_{yy}} M \tag{18}
$$

$$
\dot{r} = \frac{\left(I_{xx}^2 - I_{xx} I_{yy} + I_{xz}^2\right) pq + I_{xz}\left(I_{yy} - I_{zz} - I_{xx}\right) qr}{I_{xx} I_{zz} - I_{xz}^2} + \frac{I_{xz} L + I_{xx} N}{I_{xx} I_{zz} - I_{xz}^2} \tag{19}
$$

The angular rate equations of motion consist of both the coupled translational and rotational effects along with the external moments applied to the aircraft by fuselage, main rotor, and tail rotor aerodynamic forces and moments. Similar to the external forces, Padfield [10] defines the external moments (L, M, N) as:

$$
\begin{aligned}
L &= \frac{-b}{2} K_{\beta} \beta_{1s} + h_R \left(\frac{1}{2} \rho \pi R^2 \left(\Omega R\right)^2 a_0 s \left(\frac{2C_Y}{a_0 s} \right) \right) \\
&+ h_T \frac{1}{2} \rho \left(\Omega_T R_T\right)^2 a_{0T} s_T \left(\pi R_T\right)^2 \left(\frac{2C_{TT}}{a_{0T} s_T} \right) F_T
\end{aligned}
\tag{20}
$$

$$
\begin{aligned}
M &= \frac{-b}{2} K_{\beta} \beta_{1s} - h_R \frac{1}{2} \rho \pi R^2 \left(\Omega R\right)^2 a_0 s \left(\cos\gamma_s \left(\frac{2C_X}{a_0 s} \right) - \sin\gamma_s \left(\frac{2C_Z}{a_0 s} \right) \right) \\
&+ x_{cg} \frac{1}{2} \rho \pi R^2 \left(\Omega R\right)^2 a_0 s \left(\sin\gamma_s \left(\frac{2C_X}{a_0 s} \right) + \cos\gamma_s \left(\frac{2C_Z}{a_0 s} \right) \right) \\
&+ \left(l_{TP} + x_{cg}\right) \frac{1}{2} \rho \left(\Omega R\right)^2 \overline{V}_T^2 S_{TP} C_{ZTP}\left(\alpha_{TP}\right)
\end{aligned}
\tag{21}
$$

$$
\begin{aligned}
N = {} & \frac{1}{2}\rho(\Omega R)^2 \pi R^3 s a_0 \left(\frac{2C_Q}{a_0 s} + \left(\frac{2I_R}{bI_\beta} \right) \frac{\overline{\Omega}'}{\gamma} \right) - x_{cg} \left(\frac{1}{2}\rho\pi R^2 (\Omega R)^2 a_0 s \left(\frac{2C_Y}{a_0 s} \right) \right) \\
& - (l_T + x_{cg}) \frac{1}{2}\rho(\Omega_T R_T)^2 a_{0T} s_T (\pi R_T)^2 \left(\frac{2C_{TT}}{a_0 s_T} \right) F_T
\end{aligned} \tag{22}
$$

The effects of the tail rotor when using a FLIGHTLAB® model are different than described. The model uses a disk rotor rather than a blade model, leading the moments from the rotor as simply a thrust output across a moment arm. Referencing Table 2, these are some of the assumptions used to zero out an aerodynamic force or moment. Similar to the external forces, the external moments are broken into the rotor dynamic equations as functions of the control parameters.

2.1.4. Aircraft Euler Angles

The aircraft Euler angles are the roll, pitch, and yaw angles of the fuselage with respect to an earth-fixed coordinate system. The equations of motion are based upon the angular rates and current measured positions.

$$\dot{\phi} = p + q \sin\phi \tan\theta + r \cos\phi \tan\theta \tag{23}$$

$$\dot{\theta} = q \cos\phi - r \sin\phi \tag{24}$$

$$\dot{\psi} = q \sin\phi \sec\theta + r \cos\phi \sec\theta \tag{25}$$

For practical purposes, the yaw angle is not a parameter of concern since the aircraft may have a heading of any direction and will not change the motion of the aircraft. However, the rate of change in yaw is of great importance when determining if an engine failure has occurred. The yaw rate will change when power is no longer provided to the tail rotor.

2.2. The Dynamic Engine Model

As the airframe and rotor system states can provide indications of an engine failure in a rotorcraft, there is also the opportunity to measure engine parameters such as temperature, pressure, torque, etc. to hypothesize on such an engine failure. For the utility helicopter in question, a basic T-701 turboshaft engine will be modeled to evaluate the engine criteria.

The Army incorporated tandem T700-GE-700 series engines in the Apache attack and Blackhawk utility helicopters [11]. The 1600-hp class turboshaft engine consists of 4 basic phases; the compressor, combustion chamber, gas generator turbine, and power turbine [12]. The compressor reduces the volume of the gas, thereby increasing the pressure. The pressurized air is sent to the combustor, where the air/fuel mixture is combusted, significantly increasing the temperature and pressure of the combusted air mixture. The high-pressurized air powers the axial spinning of the gas generator turbine, thereby powering the compressor to continue sending pressurized air to the combustor. The remainder of the energy from the combustor that is not used in the gas generator turbine is used to power the free spinning power turbine. As shown in Figure 2, the power turbine is mechanically linked to the rotor system [12].

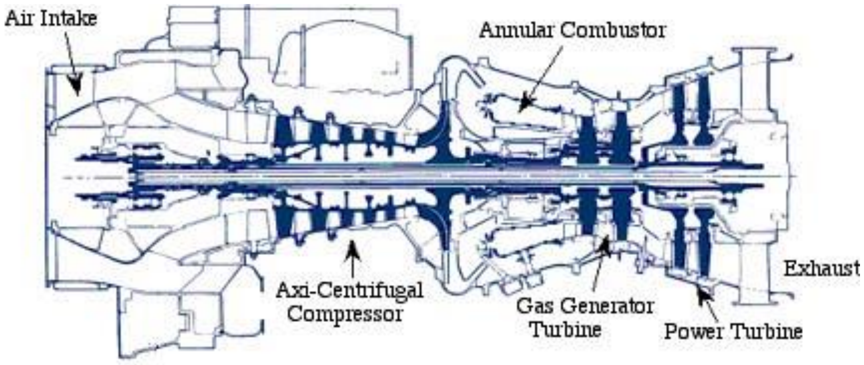


Figure 2: T701 Turboshaft Engine

A feature of rotorcraft systems is that various degrees of freedom are highly coupled. Therefore, when it comes to detecting an engine failure, multiple indicators are available for the pilot or any flight control system. Within the engine, the T45 temperature, Ng, Np, and the torque output can be monitored with sensors and will show signs of an engine failure. The T45 temperature is the air and fuel mixture temperature leaving the combustor and entering the turbine of the engine. An engine failure would result in a combustor failure and temperature drop. The gas generator speed is the resultant of the compressors and combustor. The torque from the gas generator drives the power generator turbines. Like the combustor failing to heat the air, the Ng and Np will fail to maintain speed. Finally, the torque output, related to the Np will not be significant enough to drive the rotor system. Any and all of these components are capable of being used for model hypotheses if integrated into engine model equations of motion.

The handling response of the aircraft also degrades upon an engine failure. According to pilot manuals, upon an engine failure a rotorcraft will lose rotor speed, pitch down and yaw to the left. The aircraft will pitch down due to the reduction in lift and the main rotor being slightly ahead of the aircraft CG. The left yaw will occur due to the loss of counter torque in the tail being overpowered by the friction in the main rotor (rotates CCW from a top view) [3]. These are typically signs for a pilot, along with cockpit indicators, that an engine failure has occurred. Waiting for attitude changes to detect engine failure introduces lag. The pilot is constantly making adjustments to hold pitch attitude and heading before an engine failure. Combined with a slower response due to the mass and size of a rotorcraft, valuable time is lost before a pilot reaction.

The model simulation used for the following research representing the T700 turboshaft engine is described by Ballin [12]. The author used the following assumptions for a model representing a turboshaft engine:

- Each stage of engine model applies the conservation of mass.
- Previously acquired empirical data was used to account for losses attributed to fluid dynamic or mechanical processes.
- The power turbine efficiency represented as a function of power turbine speed was ignored.
- Secondary effects such as bleed flows and variable vane geometries were represented by linear relationships and are always in the nominal position.

Using the assumptions, the model uses the inlet values of pressure and temperature to march through the various components of the engine model, calculating enthalpy, mass flows, and gas and power turbine torques along the way. The author provides multiple case studies for varying state vectors to drive the simulation. The state vector chosen consists of both the gas generator speed and power turbine speed [12]. The control input is the fuel flow introduced prior to combustion.

Additional assumptions were made by the author per the suggestions in Duyar [11]. The assumptions were applied to the model beyond the scope of the engine model developed by Ballin [12].

- The power turbine is mechanically linked to the rotor system, and the main rotor speed is a constant ratio to that of the power turbine speed.
- When operating with a healthy engine, model variables are to be adjusted to attempt to maintain a constant power turbine speed through the engine load sharing sub-function written by AED.

Ballin [12] describes multiple configurations for variable state vector sizes. The models examined use a combination of the gas generator speed (NG), the power turbine speed (NP), the total pressure at the compressor outlet (P3), the total pressure at the gas generator inlet (P41), the total pressure at the power turbine inlet (P45), and the total pressure at the gas generator input (T41). Note that station 41 is also considered the combustor outlet.

While increasing the number of parameters should lead to a more reliable engine model, Ballin presents the two degree-of-freedom model approximates the dynamics between the control volumes of the engine compressor, combustor, gas generator, and power turbine interact instantaneously, rather than implementing a time delay for fluid to pass through the engine components [12]. This paper uses the two degree-of-freedom model as opposed to those with more states because of the information available through the Army FLIGHTLAB® Integrated Quasi-Steady Engine Module (FIQSEM) model along with the age of the development of the small turboshaft paper. The Army FIQSEM model serves to bypass the modeling and dynamics for the propulsion system to better apply features such as malfunction notifications, current engine dynamics and performance, and cockpit outputs. Figure 3 shows the FIQSEM implementation into a FLIGHTLAB® model for the Army. However, the internal structure of the FIQSEM model is unknown to the author, preventing the use of engine temperatures and pressures at various stations. The age of the Ballin model also led to the reduction of variables. The paper was published in 1988. Significant advancement in engine performance, fuel performance, and material composition introduces enough discrepancy into the system that a lower number of states is preferred. The reduction in fuel flow and the known gearing ratio between the power turbine and main rotor system are appropriate to represent failure models for this research.

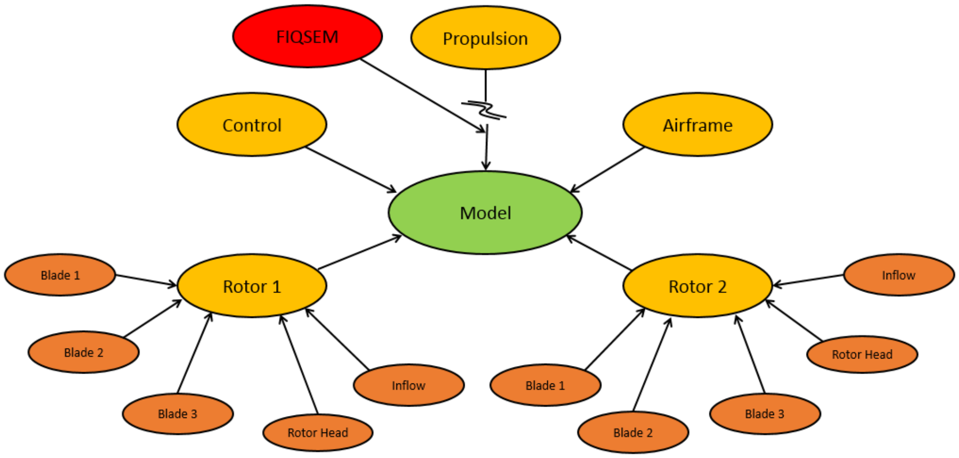


Figure 3: FLIGHTLAB® Model Components with FIQSEM Addition

Ballin [12] uses three different flight conditions in which to tune the engine models. The conditions are a high-powered flight condition, power levels associated with the utility helicopter performing around the bucket speed, and power levels that occur in a controlled descent. The second condition for operation around the bucket speed was chosen for the engine model. The model was chosen to operate at normal flight conditions. The model will only evaluate the initial moments of an engine failure and not the steady descent phase, therefore the descent condition is ignored. Table 3 outlines the variables that are to be used for the engine model along with the original source as they enter model for substantiation purposes. If the source is from the computer software, FLIGHTLAB®, the path to the variable is also provided to avoid confusion within the software package structure.

Variable	Value	Model Source
A, b	Large Matrices	Ballin Ref.
N_g	propulsion_data_FIQSEMOUTPUT_pcng1	FLIGHTLAB
N_p	propulsion_data_FIQSEMOUTPUT_pcnp1disp	FLIGHTLAB
w_f	propulsion_data_FIQSEMOUTPUT_wfpph1disp	FLIGHTLAB

Table 3: Engine Model Parameter Source

With the exception of the matrices used to simplify the engine model for bucket speed operation [12], the components in use originate from the FLIGHTLAB® model. This is to ensure there is always a consistent and non-tampered input reading into the engine model. The input variables are calculated by FLIGHTLAB®, not the engine model. The resultant engine model can be described as:

$$x(k+1) = Ax(k) + \bar{b}w_f \tag{26}$$

The state vector in Equation (26) is:

$$x(k) = \{NG \quad NP\}^T \tag{27}$$

The control variable is fuel flow and the components of Equation (26) are defined as:

$$A = \begin{bmatrix} -2.233 & 0 \\ 0.3128 & -0.4461 \end{bmatrix} \tag{28}$$

$$b = [149400 \quad 16630]^T \tag{29}$$

2.3. Coupling of Dynamic and Engine Models

The dynamic model described in Section 2.1 and the engine model described in Section 2.2 are not coupled together in their current form. It is known the engine model provides torque to the main rotor. The main rotor speed is a part of the external forces and moments that contribute to the equations of motion. The forces provide a resistance to the engine model which is accounted for with fuel flow as a control input. For coupling the dynamic model to the engine model, a mechanical linkage between the power turbine speed and main rotor speed is used.

$$\Omega_k = NP_k \left(\frac{1}{81.04} \right) \left(\frac{2\pi}{60} \right) \tag{30}$$

The gearing ratio of the power turbine to the main rotor is 81.04:1. Therefore, the dynamic model is partially a function of not only the main rotor speed, but also the power turbine speed, gas generator speed, and engine fuel flow.

Another means of further coupling the dynamic helicopter model to the engine model is through the external moment in the z-direction, N, contained within the equation for roll rate and yaw rate. The moment consists of two terms making up the torque provided to the rotor from the engine and two terms of the dynamic model acting against the motion of the main rotor. Examining the terms of the engine providing torque to the main rotor, the first represents the aerodynamic torque applied to the rotor and contains a torque coefficient which can be used to couple the dynamic and engine models. Kim [13] equates the torque coefficient (C_Q) and pressure coefficient (C_P) and defines them as:

$$C_P = C_{P_i} + C_{P_0} + C_{P_p} \tag{31}$$

The pressure coefficient is the summation of the induced power coefficient, the profile power coefficient, and the parasite power coefficient. All three coefficients are functions of the advance ratio which is a function of forward speed and main rotor speed. The induced power coefficient is also a function of the longitudinal shaft tilt in the wind axis. Therefore, the torque coefficient is a function of forward speed, main rotor speed, and pitch angle.

The second term of the Z-moment is the pseudotorque the engine provides due to the changing speed of the main rotor. The term contains a normalized rate of change of the main rotor speed. The variable is a function of the rate of change of the main rotor speed which is mechanically linked to the rate of change of the power turbine speed. Therefore, the two terms of the external moment in the Z-direction are used to couple the rate of change of power turbine speed to both the roll rate and yaw rate of the aircraft.

The parameter designation for the model matrices are matched with those in the Sioris paper [14].

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} + \Lambda\mathbf{w} \tag{32}$$

where the state and control vectors are:

$$\mathbf{x} = \{u \quad v \quad w \quad p \quad q \quad r \quad \phi \quad \theta \quad NG \quad NP\}^T \tag{33}$$

$$\mathbf{u} = \{\theta_0 \quad \theta_{1cw} \quad \theta_{1sw} \quad w_f\}^T \tag{34}$$

and the corresponding matrices are represented as:

$$A=\begin{bmatrix}
0 & r & -q & 0 & -w & v & 0 & \frac{\partial\dot{u}}{\partial\theta} & 0 & \frac{\partial\dot{u}}{\partial NP} \\
-r & 0 & p & w & 0 & -u & \frac{\partial\dot{v}}{\partial\phi} & \frac{\partial\dot{v}}{\partial\theta} & 0 & \frac{\partial\dot{v}}{\partial NP} \\
q & -p & 0 & -v & u & 0 & \frac{\partial\dot{w}}{\partial\phi} & \frac{\partial\dot{w}}{\partial\theta} & 0 & \frac{\partial\dot{w}}{\partial NP} \\
0 & 0 & 0 & \frac{\partial\dot{p}}{\partial p} & \frac{\partial\dot{p}}{\partial q} & \frac{\partial\dot{p}}{\partial r} & 0 & 0 & \frac{\partial\dot{p}}{\partial NG} & \frac{\partial\dot{p}}{\partial NP} \\
0 & 0 & 0 & \frac{\partial\dot{q}}{\partial p} & \frac{\partial\dot{q}}{\partial q} & \frac{\partial\dot{q}}{\partial r} & 0 & 0 & 0 & \frac{\partial\dot{q}}{\partial NP} \\
0 & 0 & 0 & \frac{\partial\dot{r}}{\partial p} & \frac{\partial\dot{r}}{\partial q} & \frac{\partial\dot{r}}{\partial r} & 0 & 0 & \frac{\partial\dot{r}}{\partial NG} & \frac{\partial\dot{r}}{\partial NP} \\
0 & 0 & 0 & \frac{\partial\dot{\phi}}{\partial p} & \frac{\partial\dot{\phi}}{\partial q} & \frac{\partial\dot{\phi}}{\partial r} & \frac{\partial\dot{\phi}}{\partial\phi} & \frac{\partial\dot{\phi}}{\partial\theta} & 0 & 0 \\
0 & 0 & 0 & \frac{\partial\dot{\theta}}{\partial p} & \frac{\partial\dot{\theta}}{\partial q} & \frac{\partial\dot{\theta}}{\partial r} & \frac{\partial\dot{\theta}}{\partial\phi} & \frac{\partial\dot{\theta}}{\partial\theta} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2.233 & 0 \\
\frac{\partial N\dot{P}}{\partial u} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial N\dot{P}}{\partial\theta} & \frac{\partial N\dot{P}}{\partial NG} & \frac{\partial N\dot{P}}{\partial NP}
\end{bmatrix} \tag{35}$$

$$B=\begin{bmatrix}
\frac{\partial\dot{u}}{\partial\theta_0} & \frac{\partial\dot{u}}{\partial\theta_{1cw}} & \frac{\partial\dot{u}}{\partial\theta_{1sw}} & 0 \\
\frac{\partial\dot{v}}{\partial\theta_0} & \frac{\partial\dot{v}}{\partial\theta_{1cw}} & \frac{\partial\dot{v}}{\partial\theta_{1sw}} & 0 \\
\frac{\partial\dot{w}}{\partial\theta_0} & \frac{\partial\dot{w}}{\partial\theta_{1cw}} & \frac{\partial\dot{w}}{\partial\theta_{1sw}} & 0 \\
\frac{\partial\dot{p}}{\partial\theta_0} & \frac{\partial\dot{p}}{\partial\theta_{1cw}} & \frac{\partial\dot{p}}{\partial\theta_{1sw}} & \frac{\partial\dot{p}}{\partial w_f} \\
\frac{\partial\dot{q}}{\partial\theta_0} & \frac{\partial\dot{q}}{\partial\theta_{1cw}} & \frac{\partial\dot{q}}{\partial\theta_{1sw}} & 0 \\
\frac{\partial\dot{r}}{\partial\theta_0} & \frac{\partial\dot{r}}{\partial\theta_{1cw}} & \frac{\partial\dot{r}}{\partial\theta_{1sw}} & \frac{\partial\dot{r}}{\partial w_f} \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 149400 \\
0 & 0 & 0 & \frac{\partial N\dot{P}}{\partial w_f}
\end{bmatrix} \tag{36}$$

A derivation of the Jacobian terms described in the above equations is outlined in Appendix A. As shown in the appendix, main rotor speed serves as a dependent variable for the external forces (X, Y, Z) and external moments (L, M, N). Also note that following the coupling of the dynamic and engine model, the engine parameters act upon the translational and rotation velocity equations of motion. The forward speed and pitch angle act upon the equation of motion for the power turbine speed.

2.4. Bayes' and Multiple Model Estimation

The probability of an event in a simple system is usually straight forward. The probability of rolling a 2 with a die, or the probability of a flipped coin landing on heads, or even the probability of guessing on a multiple-choice exam are examples with simple probabilities. However, most systems in the real world do not depend on a single "flip of the coin" to function. Whether it be a car backfiring due to a faulty spark plug, an incorrect fuel injection system, or an issue with fuel pumped at a gas station, or a bicycle chain malfunction due to incorrect tension, a faulty gear, or inadequate gear transfer system, multiple and independent parameters can cause the same malfunction. A probability density represents the likeliness of an outcome to occur given certain input conditions. Unlike a discrete random variable, the continuous random variables use the probability density function. The objective of a Bayes' estimation is to predict the outcome of a continuous random variable given information of a separate random variable [14].

Two dependent variables can be evaluated against each other using a conditional mean. The relationship between parameter x_i and y_j is represented as:

$$\Pr\left(x_i \middle| y_j\right) = \frac{\Pr\left(x_i, y_j\right)}{\Pr\left(y_j\right)} \tag{37}$$

$$\Pr\left(y_j \middle| x_i\right) = \frac{\Pr\left(x_i, y_j\right)}{\Pr\left(x_i\right)} \tag{38}$$

Combining Equations (37) and (38), Bayes' rule can be represented as:

$$\Pr\left(x_i \middle| y_j\right) = \frac{\Pr\left(y_j \middle| x_i\right)\Pr\left(x_i\right)}{\Pr\left(y_j\right)} \tag{39}$$

The probability of the system functioning in a specific state due to the observations of the system is a function of (a) the known observations during a desired state, (b) the probability of the system being in the desired state, and (c) the probability of observations. The equation is effective for comparing two dependent variables. However, what if there are multiple variables that can affect the probability of the value of x_i? An example is the probability that a bicycle chain may break (x_i) given the tension in the chain (y_1). The chain also has a probability to pop (x_i) given a bent tooth on a gear (y_2). For this situation, Multiple-Model Estimation (MME) can be applied. MME is a generalized principle in which many values of system parameters could be run through their own linear optimal filters, and the state estimate with the highest probability would stand as the actual value of the parameter [15]. Each filter will be assigned a confidence value. The sum of each result using a Bayes' estimation weighed with the corresponding probability confidence results in an optimal estimate.

The system also becomes a dynamic system that is continuously running for every simulation time step. Therefore, the system parameters are based upon the density function containing measured data from instrumentation, referred to as the measurement vector, along with the system states which may also vary every time step. The equation is represented as:

$$\Pr\left(\mathbf{p}_j \middle| \mathbf{z}_k\right) = \frac{pr\left(\mathbf{z}_k \middle| \mathbf{p}_j\right)\Pr\left(\mathbf{p}_j\right)_{k-1}}{\sum_{i=1}^{l}\left[pr\left(\mathbf{z}_k \middle| \mathbf{p}_i\right)\Pr\left(\mathbf{p}_i\right)_{k-1}\right]} \qquad \textbf{(40)}$$

The notation Pr( ) indicates a probability mass function while *pr*( ) is a probability density function.

The issue is a case where there are a finite number of models to evaluate with only one model being correct. Thus, the result is a discrete variable where $\Pr(\mathbf{p}_j)$ is the probability that $\mathbf{p}_j$ is correct. However, the measurement from the system can take on any value, making $\mathbf{z}_k$ a continuous variable. Therefore, $\mathrm{pr}(\mathbf{z}_k)$ is the probability density of the observation occurring.

In Equation (40), $\mathbf{p}_j$ represents the event that the j-th model is correct while $\mathbf{z}_k$ is the measurement vector of known parameters. Therefore, knowing information about $\mathbf{z}_k$ provides an estimate on model correctness variable $\mathbf{p}_j$. In short, Equation (40) estimates the probability that the model is correct.

The numerator of Equation (40), multiplies the probability that an observation vector would occur given a correct j-th model, $pr\left(\mathbf{z}_k \middle| \mathbf{p}_j\right)$, and the probability of the accuracy of the model of the previous time step given the observation vector, $\Pr\left(\mathbf{p}_j\right)_{k-1} = \Pr\left(\mathbf{p}_j \middle| \mathbf{z}_{k-1}\right)$.

The denominator is simply the summation of all the numerators representing each model available in the system. The denominator is used as a normalizing function in order to represent each model probability, $\Pr\left(\mathbf{p}_j \middle| \mathbf{z}_k\right)$, on a 0 to 1 scale. However, the density function state is unknown at this instant, therefore, the state estimate must be generated by a Kalman filter represented below [15].

$$\hat{\mathbf{x}}_k(-) = \Phi\hat{\mathbf{x}}_{k-1}(+) + \Gamma\mathbf{u}_{k-1} \qquad \textbf{(41)}$$

$$\mathbf{P}_k(-) = \Phi\mathbf{P}_{k-1}(-)\Phi^T + \Lambda\mathbf{Q}'_{k-1}\Lambda^T \qquad \textbf{(42)}$$

$$\mathbf{K}_k = \mathbf{P}_k(-)\mathbf{H}^T\left[\mathbf{H}\mathbf{P}_k(-)\mathbf{H}^T + \mathbf{R}_k\right]^{-1} \qquad \textbf{(43)}$$

$$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \mathbf{K}_k\left[\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k(-)\right] \qquad \textbf{(44)}$$

$$\mathbf{P}_k(+) = \left[\mathbf{P}_k(-)^{-1} + \mathbf{H}^T\mathbf{R}_k^{-1}\mathbf{H}\right]^{-1} \qquad \textbf{(45)}$$

The Kalman filter is broken into 5 steps for each iteration; a state estimation extrapolation (propagation), a covariance estimate extrapolation (propagation), a filter gain calculation, a state estimate update and a covariance estimate update [15]. The (-) following a parameter is the estimate that results from the propagation while the (+) indicates the estimate following an update with measured data. The state update, $\hat{\mathbf{x}}_k(+)$, is the desired output for the MME while the covariance update, $\mathbf{P}_k(+)$, tracks the uncertainty of the state estimation [16]. Equations (41-45) tracks the state estimate of the model in which it is associated.

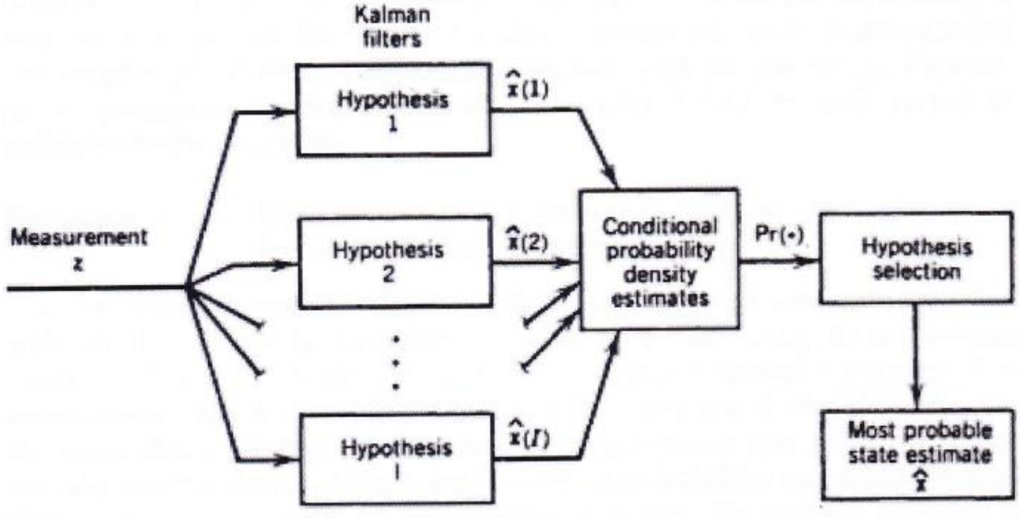


Figure 4: Bayes' Hypothesis and Probability Flow Chart with permission from [15]

The updated MME equations using Bayes' Theorem can now be represented as:

$$\Pr\left(\mathbf{p}_j \middle| \mathbf{z}_k\right) = \frac{pr\left[\mathbf{z}_k \middle| \hat{\mathbf{x}}_k\left(\mathbf{p}_j\right)\right]\Pr\left(\mathbf{p}_j \middle| \mathbf{z}_{k-1}\right)}{\sum_{i=1}^{I}\left\{pr\left[\mathbf{z}_k \middle| \hat{\mathbf{x}}_k\left(\mathbf{p}_i\right)\right]\Pr\left(\mathbf{p}_i \middle| \mathbf{z}_{k-1}\right)\right\}} \qquad \textbf{(46)}$$

Where the state vector was represented as x, the optimal estimation for the state vector given the corresponding model is correct is represented as $\hat{\mathbf{x}}_k\left(\mathbf{p}_j\right)$. The first term in the denominator may now pass through the Kalman filter to determine a propagated and updated estimate of the state. Multiple Kalman filters may run simultaneously to evaluate multiple hypotheses against each other. This scheme has been a proven method for determining the likelihood that the hypothesis is correct [17]. A Gaussian distribution is a function that indicates an observation to lie within a range of two real numbers. The distribution contains of a residual function which is the difference between a measured observation and the state propagation. A covariance matrix is also present which represents the error found in the updated state estimation [15]. The equation along with the residual and covariance matrices can be defined respectively.

$$pr\left[\mathbf{z}_k \middle| \hat{\mathbf{x}}_k\left(\mathbf{p}_i\right)\right] = \frac{1}{\left(2\pi\right)^{n/2}\left|\mathbf{S}_k\left(\mathbf{p}_i\right)\right|^{1/2}} e^{-\left(1/2\right)\mathbf{r}_k^T\left(\mathbf{p}_i\right)\mathbf{S}_k^{-1}\left(\mathbf{p}_i\right)\mathbf{r}_k\left(\mathbf{p}_i\right)} \qquad \textbf{(47)}$$

$$\mathbf{r}_k\left(\mathbf{p}_i\right) = \mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k\left[\mathbf{p}_i\left(-\right)\right] \qquad \textbf{(48)}$$

$$\mathbf{S}_k\left(\mathbf{p}_i\right) = \mathbf{H}\mathbf{P}_k\left(\mathbf{p}_i\right)\mathbf{H}^T + \mathbf{R}_k \qquad \textbf{(49)}$$

The optimal estimates for the state can now be found from the Kalman filters. Depending upon the system in which the MME Bayes' Theorem is applied will determine the coefficients for the observation and state equations.

Once the hypotheses are calculated and the probabilities from Equation (40) are determined, they can be plotted and evaluated to determine an appropriate moment that an engine failure has most likely occurred. For situations with three or more hypothesis models, an appropriate means of detecting a failure is to pinpoint the moment the "normal function" hypothesis is no longer the more probable hypothesis to occur. Figure 5 shows the comparison of three hypotheses along with the recognition point of a model surpassing the "normal" hypothesis.

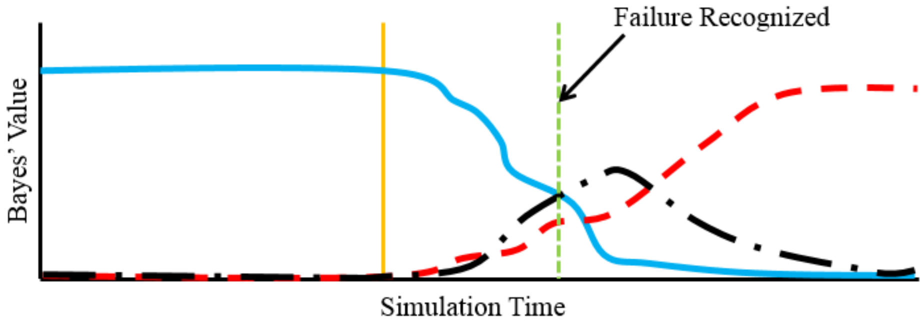


Figure 5: Failure Event Detection Scheme

Figure 5 shows and evaluation of three hypothesis models in a hypothetical situation. The solid blue line represents a model functioning with no disturbance or error. The dashed red line and the dot-dashed black line represent a model with a built in discrepancy. Therefore, in normal activity, the probability of these models being an appropriate representation of the system is near zero. The orange vertical line is the moment in time when the system goes from normal activity to broken. The green and dashed vertical line pinpoints the moment the normal model is no longer the most probable model to represent the system. This is the scheme that is to be associated with the work in future sections.

2.5. FLIGHTLAB® and VirtualPilot

The software used by the US Army to run simulations is FLIGHTLAB® produced by Advanced Rotorcraft Technology, Inc. out of Sunnyvale, CA. FLIGHTLAB® allows a user to input the physical properties of a particular rotorcraft system, design control and propulsion algorithms for parameter calculations for each simulation time step, set initial flight conditions along with aircraft configurations, trim a model and run a multi-step simulation while outputting variable values in each step. The software allows a user to control how the model is to be run. FLIGHTLAB® can be run at real-time through a helicopter simulator or non-real-time from a desktop.

The US Army has developed a VirtualPilot (VP) program which can be used to simulate a pilot-in-the-loop when running FLIGHTLAB® scripts. The virtual pilot has three basic functions; using pilot actions as a prescribed maneuver, a flight-test response, or a flight-test stick input. The VP controller is shown schematically in Figure 6 and Figure 7. The VP only controls the four stick inputs that a pilot can control, being longitudinal cyclic, lateral cyclic, collective and pedal inputs. The aircraft performance capability cannot be better than the pilot's ability to adjust with the aircraft reactions through an autorotation [18]. That is, the pilot model must have some adaptability when autorotation conditions are not optimal.

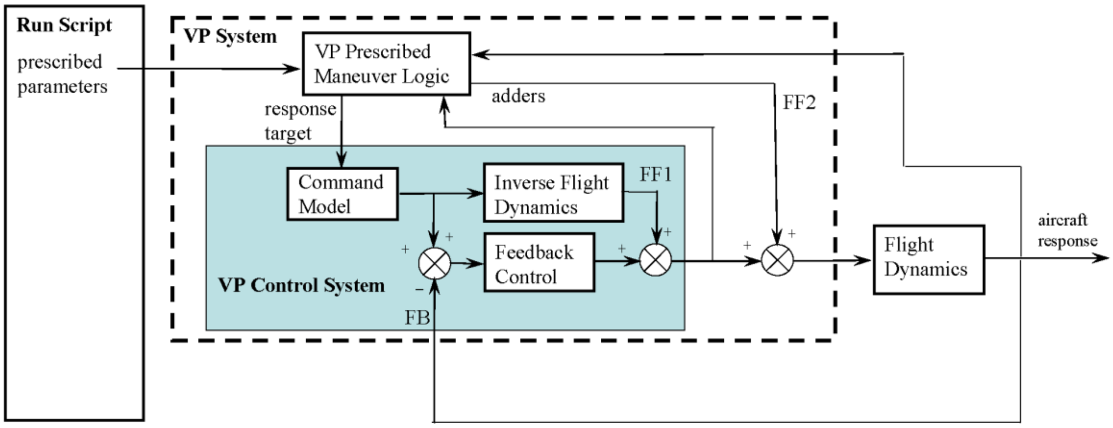


Figure 6: VirtualPilot Prescribed Maneuver Structure

The prescribed maneuver method shown in the schematic above includes a user input to perform a maneuver. The user tells the VP (a) which maneuver is to be performed whether it be forward speed control, a turn rate, heading control, etc., (b) the targets that a pilot would try to achieve such as a final forward speed or heading and (c) how aggressively to make the maneuver through gain adjustments within the VP. The commands are then fed into the command model to determine the commanded responses such as pitch, roll and yaw angle. Once the commanded responses are calculated, the stick positions to achieve the commanded responses are determined and fed into the flight dynamics model. FLIGHTLAB® takes control of the flight dynamics with its own solver functions. From that model, the aircraft state vector is determined.

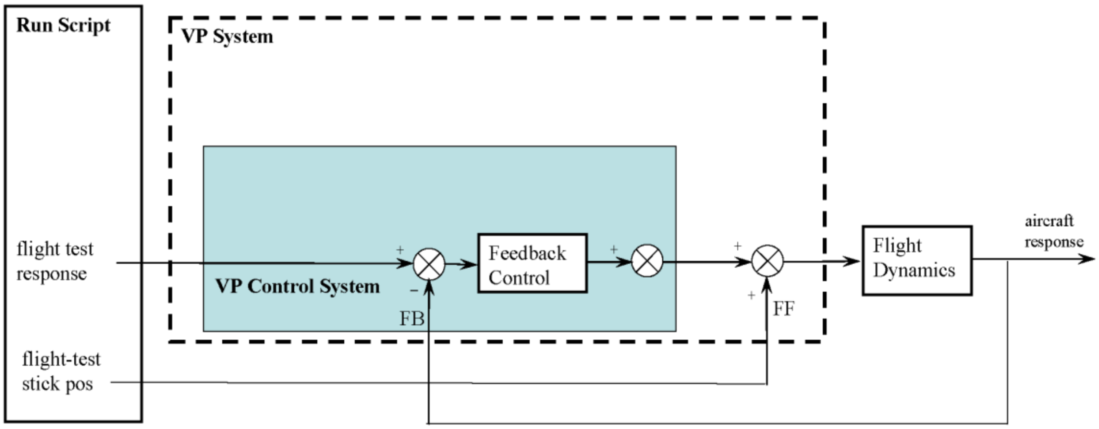


Figure 7: VirtualPilot Flight Test Data Maneuver

The final two uses for the VP as shown in the above schematic uses flight-test data as an input. For the flight-test response, the VP takes in position such as pitch, roll or yaw angle. Through the control system a stick position to achieve the state is calculated and commanded. The change in stick position is then fed into the flight dynamics model and a state vector is created. The last use for the VP is to simply input the change in stick position and feed the result into the flight dynamics model.

The VirtualPilot (VP) is broken into two user entries. The first is the commands to be run. For the longitudinal control in the cyclic, lateral control in the cyclic, collective control in the collective and directional control in the pedals, the user predefines the command type to be implemented. The second is the state targets which are used by the commands.

The command for the longitudinal cyclic is the indicated airspeed (IAS) control mode. The input is the forward indicated airspeed of the aircraft. This was selected as the primary option to allow the pilot to maintain the initial forward speed to keep the landing zone area as wide as possible. VP modes not selected for the longitudinal cyclic were pitch attitude control and forward speed control. The pitch attitude control was not used because a generic scout-class aircraft will vary forward pitch to command a forward speed with other generic utility-class vehicles, requiring the user to have extensive knowledge in the sensitivity and rigging of the aircraft in question. The IAS control mode calculates the pitch angle to be commanded and runs it through the model. The forward-speed control mode is designated for low-speed to hover applications and is therefore not used. The forward-speed control mode gains are selected as such where axial speed of the aircraft may not be the dominant speed component.

The command for the lateral cyclic has been selected down to two alternatives to be tested. The first is a roll attitude control mode. This angle can be adjusted as necessary to account for the generic model or simply remain in the trimmed roll angle to continue with straight and level flight. The other command mode which can be applied is a high-speed heading control mode. The mode essentially calculates a roll angle which can be dynamically achieved by the generic aircraft and turns until the heading is achieved. A possible down fall with this command is the turn may be taken more aggressively which will vary due to different forward speeds.

The commands in the collective and directional components will remain the same no matter the cyclic commands the user chooses. For the collective command, an altitude hold mode is to be selected until the VP recognizes and responds to an engine failure. At this point, the control mode will change to a main rotor speed control mode with an input target of 100% rotor speed. The directional component will remain in the zero-sideslip control mode. The input target will be a β-angle of zero degrees. Options not selected are heading control modes and yaw rate control modes. The modes were not selected because the optimal descent path should include zero sideslip.

3. Bayes' Analysis of Rotorcraft Failure Models

This chapter examines the UH-60 model hypothesis algorithms to test and analyze the theory from the previous chapter. In the sections to come, there will be the description of a reference model which has been calibrated to respond similarly to a U.S. Army model correlated with flight test data, and the description and calibration of the 10-State "normal" functioning model, the driveshaft failure model, and the engine failure model. The calibration will include steady flight and steady flight with speed and data discrepancies. The Bayes' analysis equations from the previous chapter will be expanded to represent a full analysis. Finally, the analysis will be run, data analyzed, and results discussed.

3.1. FLIGHTLAB® Reference Model

The Reference Model is a version of a UH-60M dynamic helicopter model created by the U.S. Army. The role of the model is to demonstrate flight performance (how well the rotorcraft flies) and handling qualities (how well the rotorcraft responds to a pilot input). The original model has gone through rigorous verification and validation testing against U.S. Army acquired flight test data.

The model used for this dissertation is an iteration of the U.S. Army model. The gross weights, moments and products of inertia, center of gravity positions, main rotor airfoil shape, and tail rotor characteristics have been altered, making the model no longer verified and validated, but approved for public release. Regression testing has been completed on the reference model by comparing it to the model used by the U.S. Army, and the author feels the results are satisfactory for continuing.

The Reference model was created using ART, Inc. software FLIGHTLAB® v3.5 within the XAnalysis tool. The model has privately designed atmosphere, airframe, main rotor, tail rotor, propulsion, flight control, and VirtualPilot sub functions being executed with each simulated time step. The engine model is external to the primary model. The model is a form of a load sharing model, in that the power used from each engine may vary. The engines are not mirrored to provide the correct power response. Finally, the model will be used to create a database to run the Bayes' Estimation as an external function. The code for the external function is provided in Appendix B and C. The model is not to be influenced by outside functions.

3.2. Model Configuration for Testing

For running all test cases, the same model configuration for the FIGHTLAB® reference model and external dynamic model will be applied. The configuration parameters can be found in Table 1. The parameters are similar to that of a real configuration of a UH-60 model, with changes applied to allow for the report to be considered for public release. The rotors applied to the model are NACA 0012 for the entire span of the rotor blade. The reference for 100% main rotor speed is 27.0 rad/sec. The reference for 100% tail rotor speed is 124.5 rad/sec. For steady flight, these values represent the rotor speed targets.

The pressure altitude allows the model to maintain the pressure and density on the aircraft for any simulated altitude. This negates the effect of changing pressure and density of the airflow across the body and through the rotor disk as the altitude changes during simulation. The ambient temperature will also remain constant at all altitudes. The model will also negate fuel burn across the simulation duration. Therefore, the weight, center of gravity, and moments/products of inertia will remain constant.

3.3. Model Verification

The sections to come will be used to compare the engine model, the 8-state dynamic model coupled with the engine model, the driveshaft failure model, and the fuel flow failure model. The initial data compared is a U.S. Army reference model which has been modified to ensure there is no classified material presented. The verification of these models will use the reference model data, propagate two seconds using the equations of motion, and compared to the model. The verification will also include analysis of the model responses to biases presented in the data along with perturbations in the physical environment. Bayesian plots will be provided to show the sensitivity of the models to the added biases, however, the full Bayes' evaluation will occur in the coming sections of this chapter.

3.3.1. Verification of Engine Model

Verification of the engine model is conducted to ensure the outputs of gas generator speed and power turbine speed respond appropriately with normal flight. The engine model has never undergone an official verification and validation; therefore, no reference model is available. However, inspection of the engine model parameters will determine the appropriateness of the current model for the work performed in this paper.

The verification will compare the steady flight of the aircraft among the varying forward speeds of 60, 80, and 100 kts indicated airspeed. The parameters evaluated are the engine fuel flow as the control variable, and the gas generator and power turbine speeds.

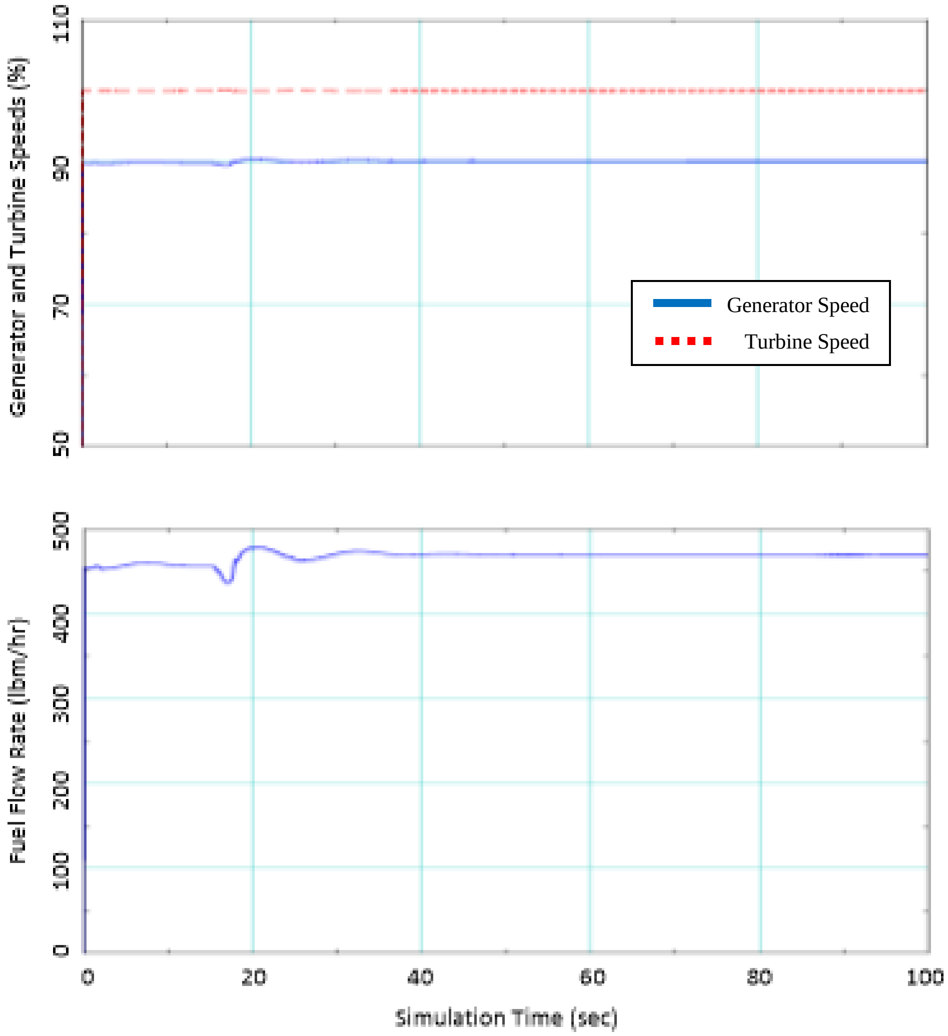


Figure 8: Engine Parameter Input and Output for 60kt Level Flight

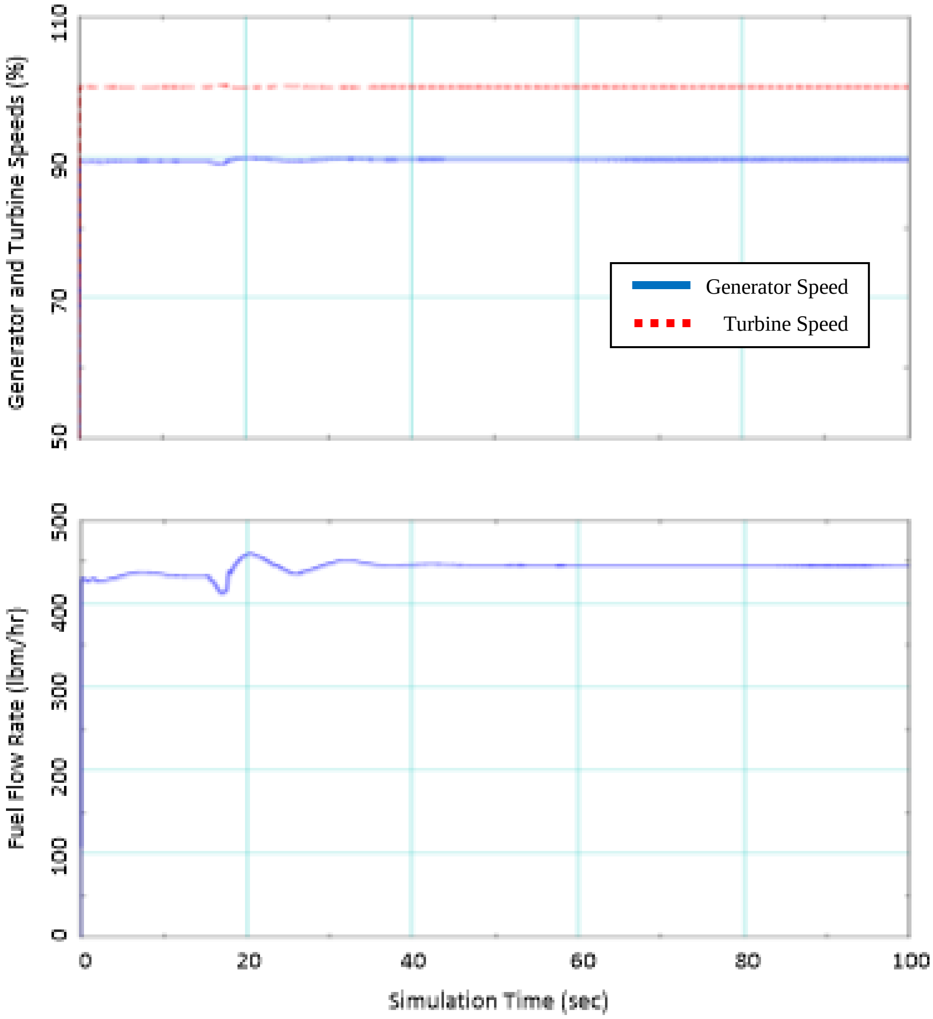


Figure 9: Engine Parameter Input and Output for 80kt Level Flight

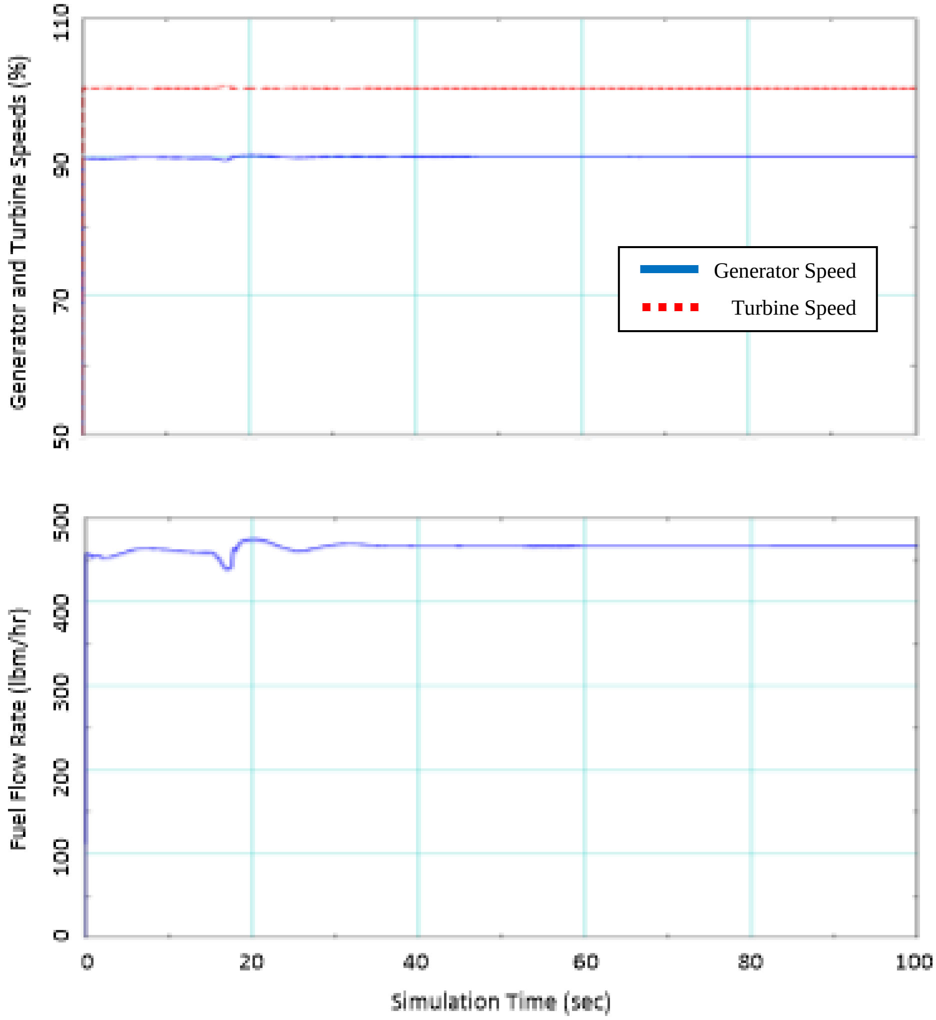


Figure 10: Engine Parameter Input and Output for 100kt Level Flight

The plots show that the gas generator speed and power turbine speed remain at an approximate value of 90% and 100% respectively no matter the airspeed. The fuel flow rate is specific for the airspeed as it is used drive the rotor and maintain a 100% rotor speed at any forward airspeed. The fuel flow will be the control when engine failure is tested. The plots indicate that the engine model used with FLIGHTLAB® input is a stable model.

3.3.2. The Coupled Dynamic and Engine Model

Comparison of the coupled dynamic and engine model to the FLIGHTLAB® reference model is used to ensure the dynamic model behaves similarly in steady flight prior to examining the model upon engine failure. The coupled model is a 10-state model consisting of body velocities, body angular velocities, pitch and roll angles, gas generator speed, and power turbine speed. The gearing ratio between the power turbine and main rotor couples the engine states to the dynamic model through the rotor speed.

Note the reference model has not undergone an official verification and validation with flight test data due to the changes made for proprietary reasons. However, the reference model has undergone regression testing with a verified and validated model. The regression testing demonstrated the expected changes in the reference model due to the changes referenced in Section 2.2. The reference model is considered appropriate for the exercises performed in this paper.

At each time step, the reference FLIGHTLAB® model data is used as the initial input into the 10-state dynamic and engine model. Using the 10-state model, the dynamic model is projected 2 seconds forward to determine how the model views the states. This should indicate stability in the system or instability. The verification will compare the steady flight of the aircraft among the varying forward speeds of 60, 80, and 100 kts indicated airspeed. The verification will also include left and right banked turns at 80 kts. The turns are to be completed with minimal sideslip. The performance parameters evaluated are the vehicle velocities in the body x, y, and z-direction, the roll, pitch, and yaw rates, and the roll and pitch angles. The verified parameters are of interest upon an engine failure, therefore it is essential that they respond like the reference model. Because the model is verified, there is a greater chance that the introduction of error in the form of an engine failure will respond similarly. Along with the aircraft performance, the swashplate longitudinal cyclic, lateral cyclic, and collective pitch angles are outputted. There is not a verified and validated engine model to compare with the dynamic engine model. Ballin [12] does provide comparison plots in the appendices to validate the simplified engine model.

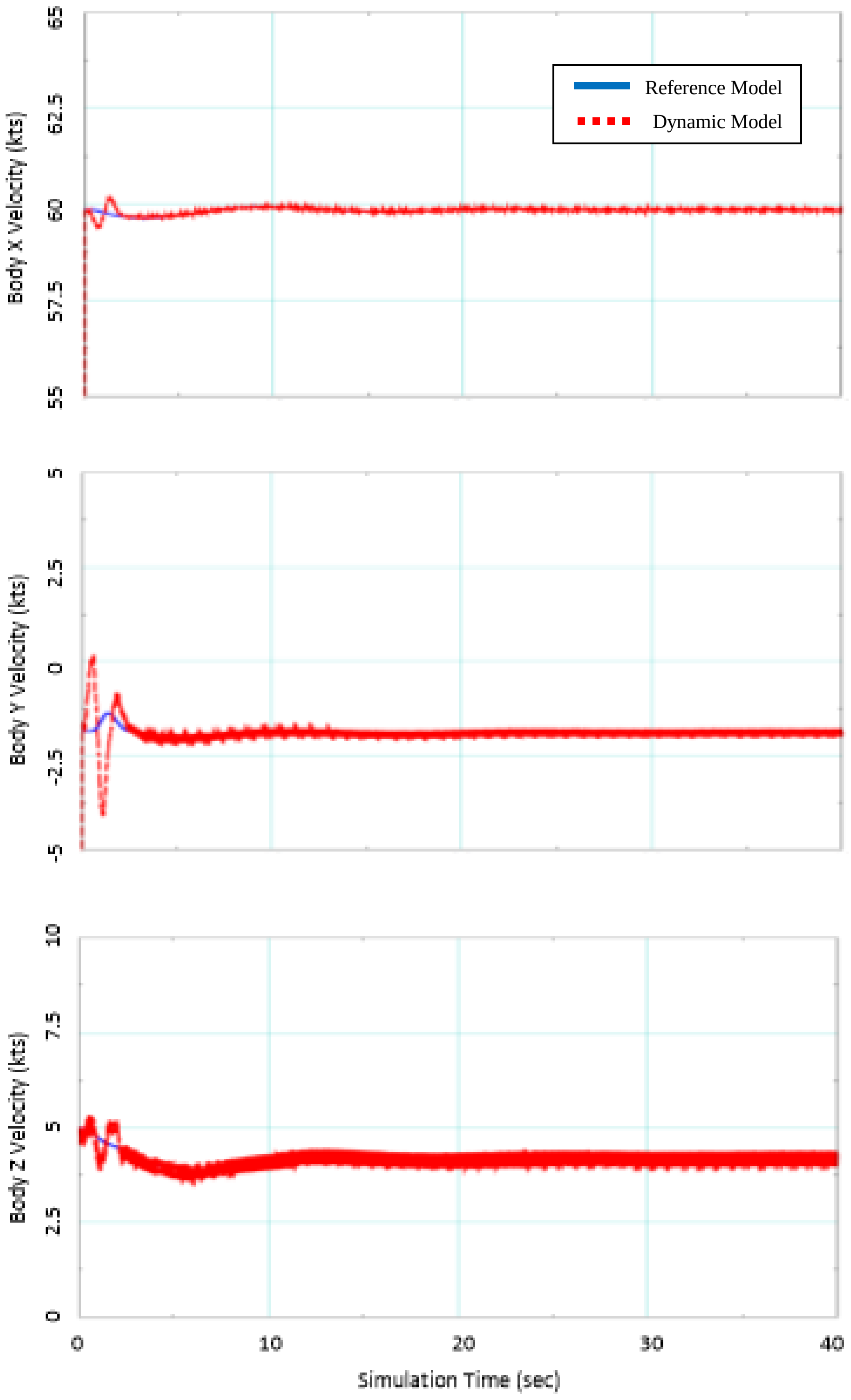


Figure 11: Body Velocities for 60kt Level Flight

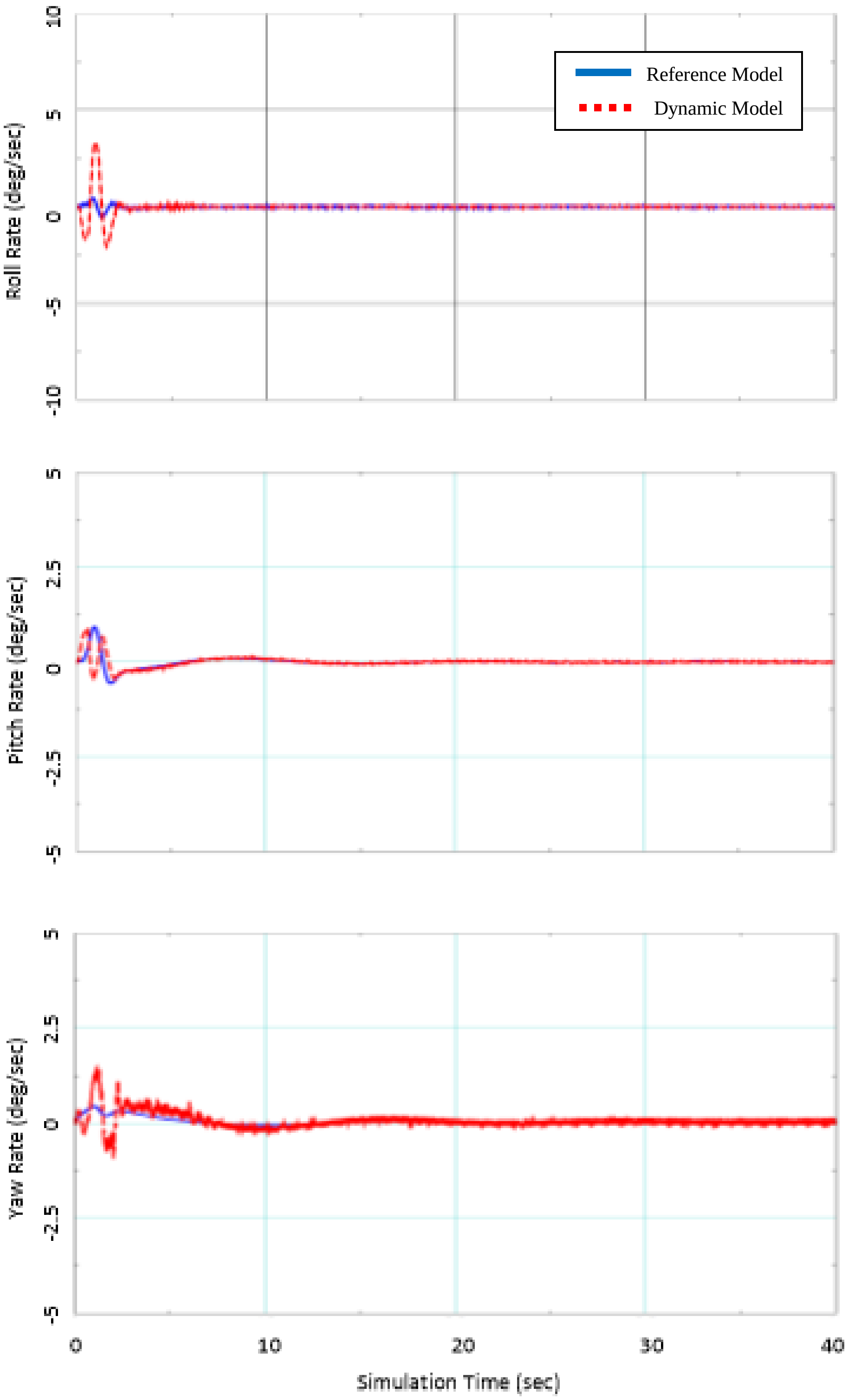


Figure 12: Body Angular Rates for 60kt Level Flight

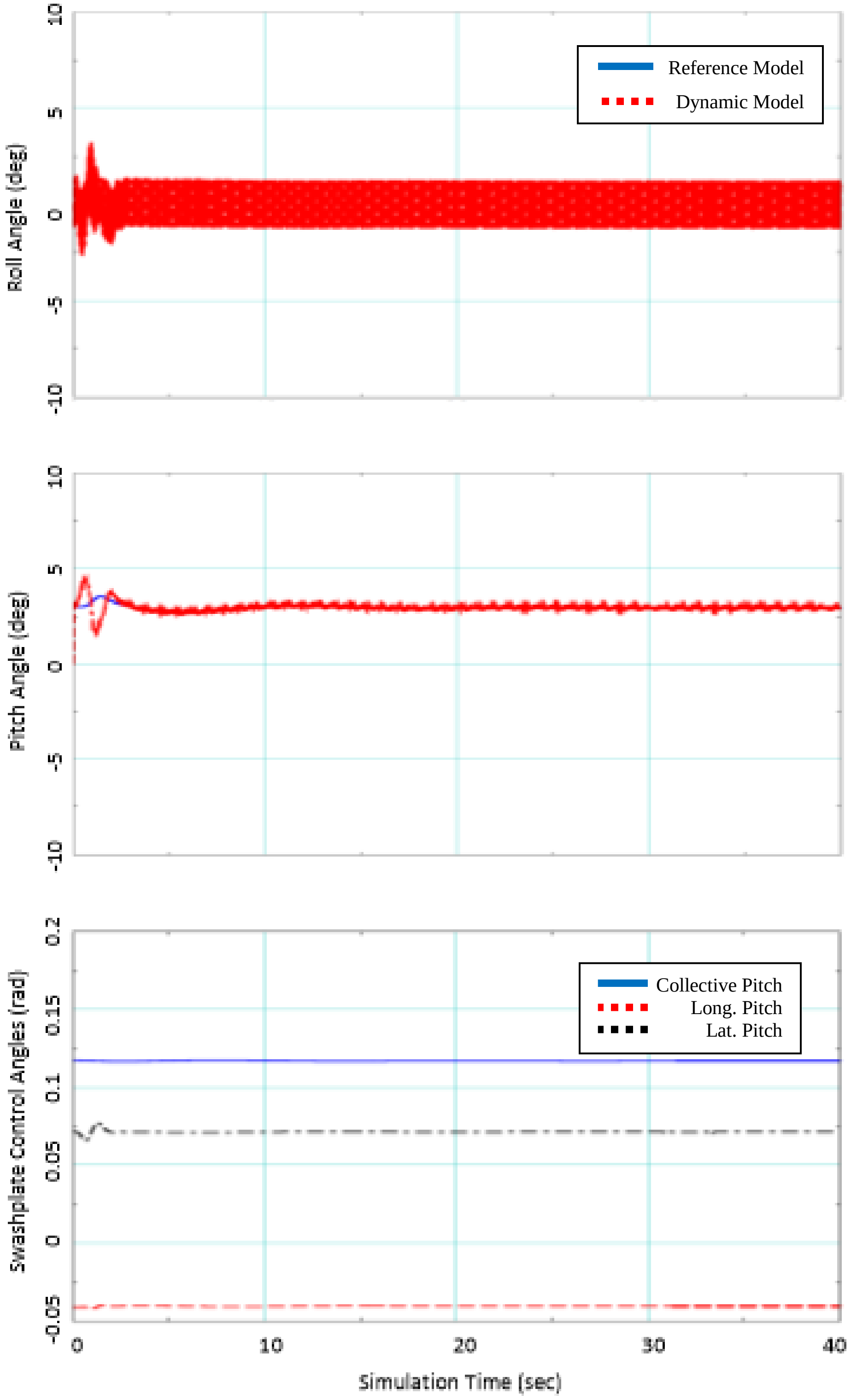


Figure 13: Orientation and Control Inputs for 60kt Level Flight

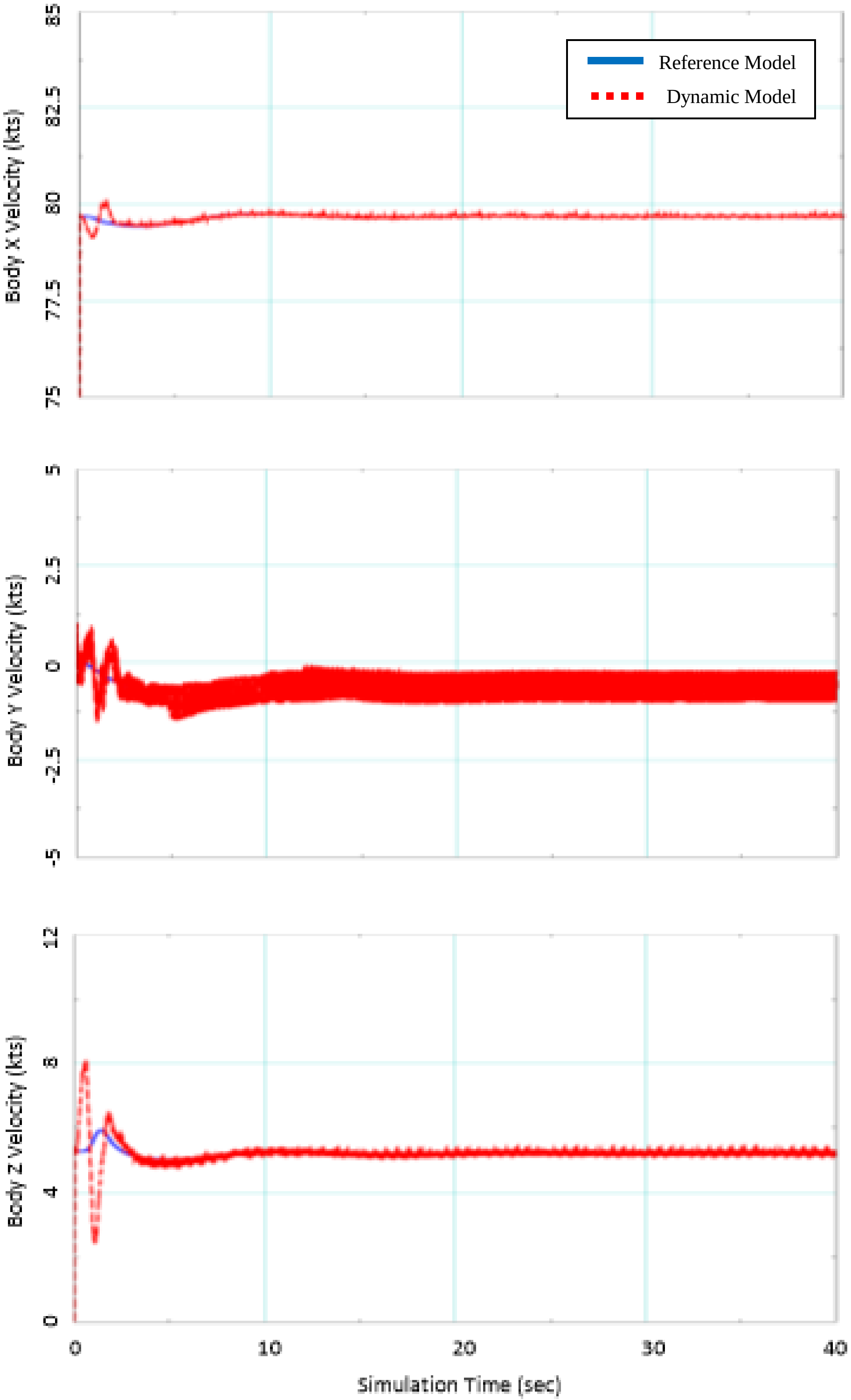


Figure 14: Body Velocities for 80kt Level Flight

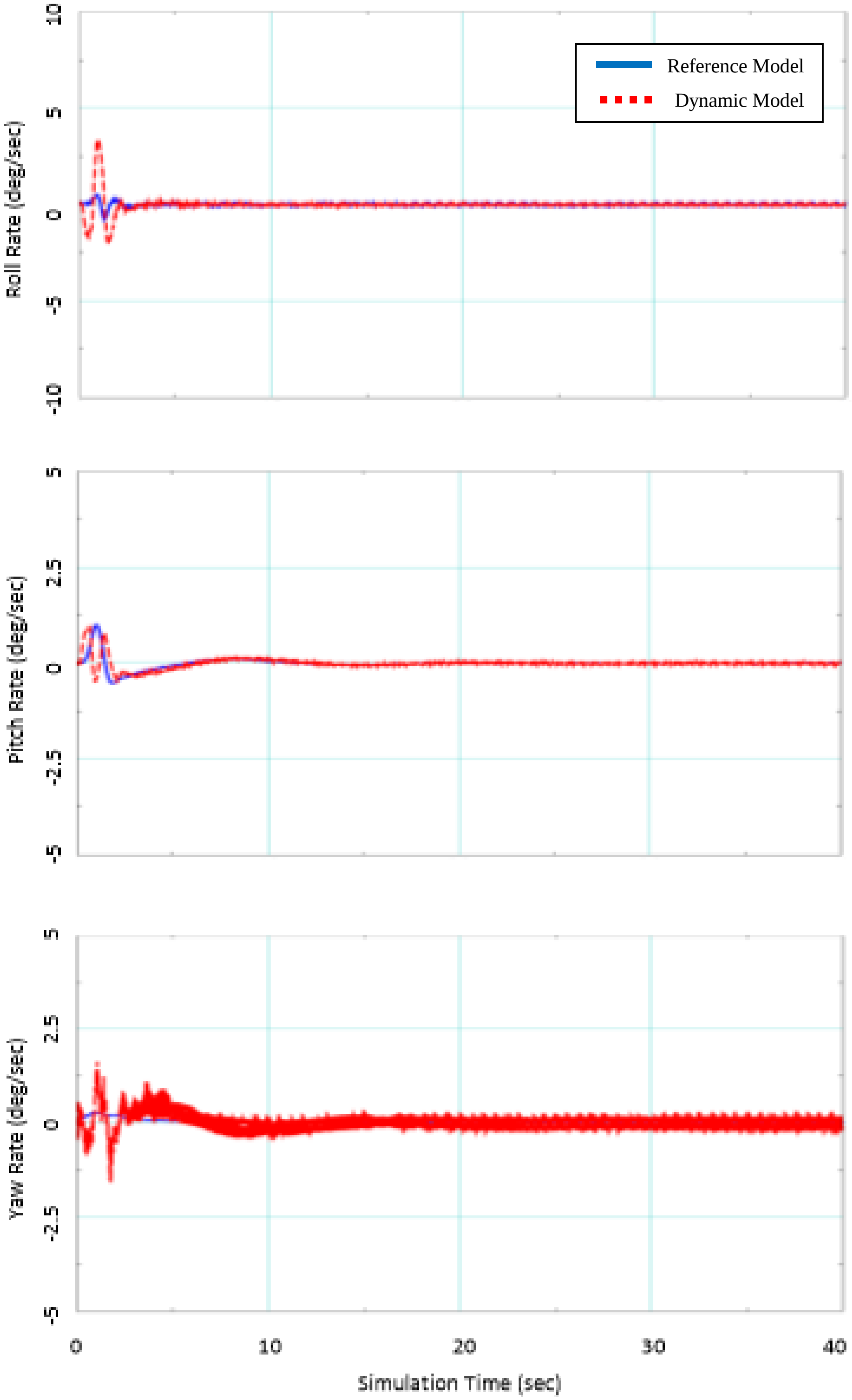


Figure 15: Body Angular Rates for 80kt Level Flight

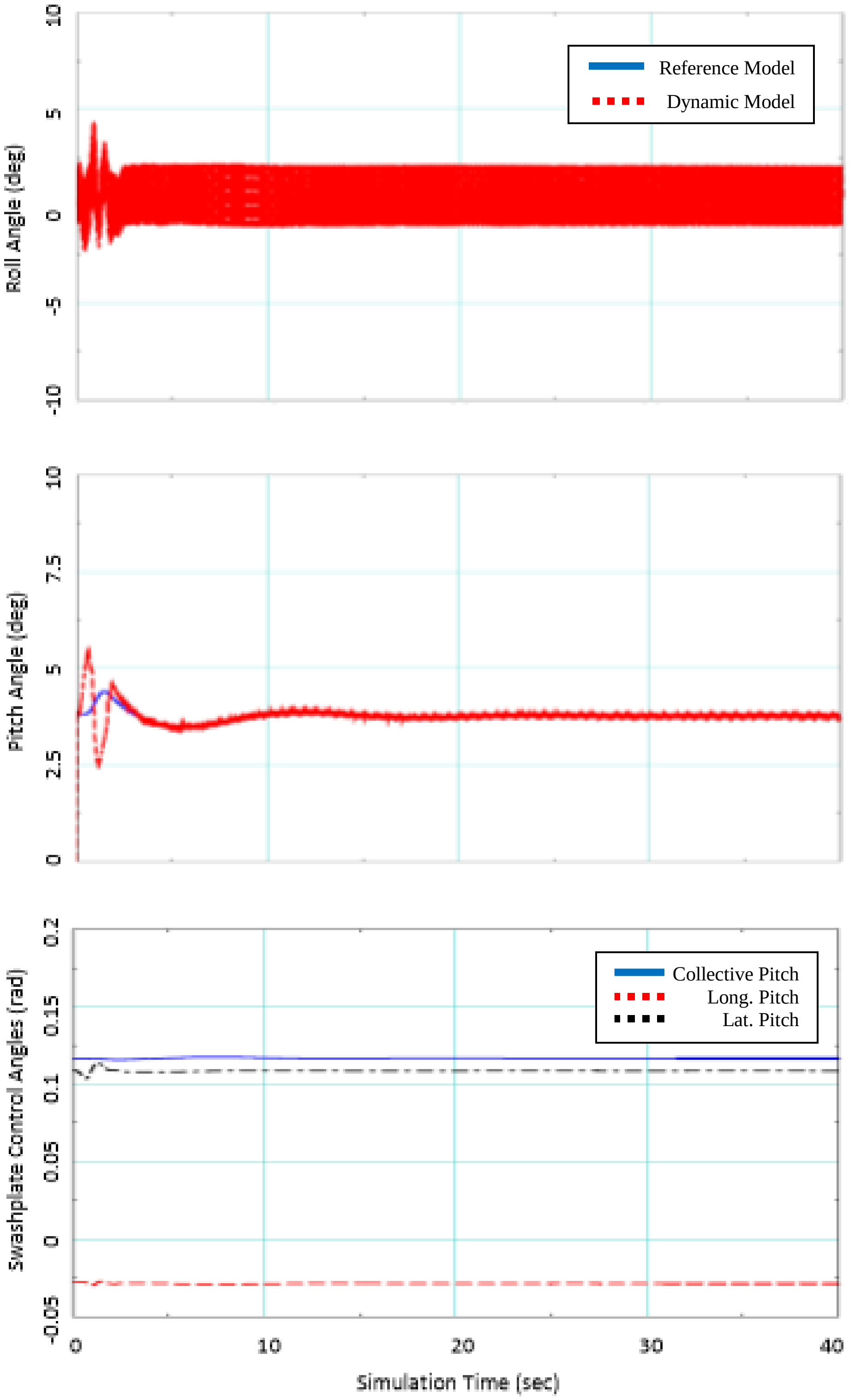


Figure 16: Orientation and Control Inputs for 80kt Level Flight

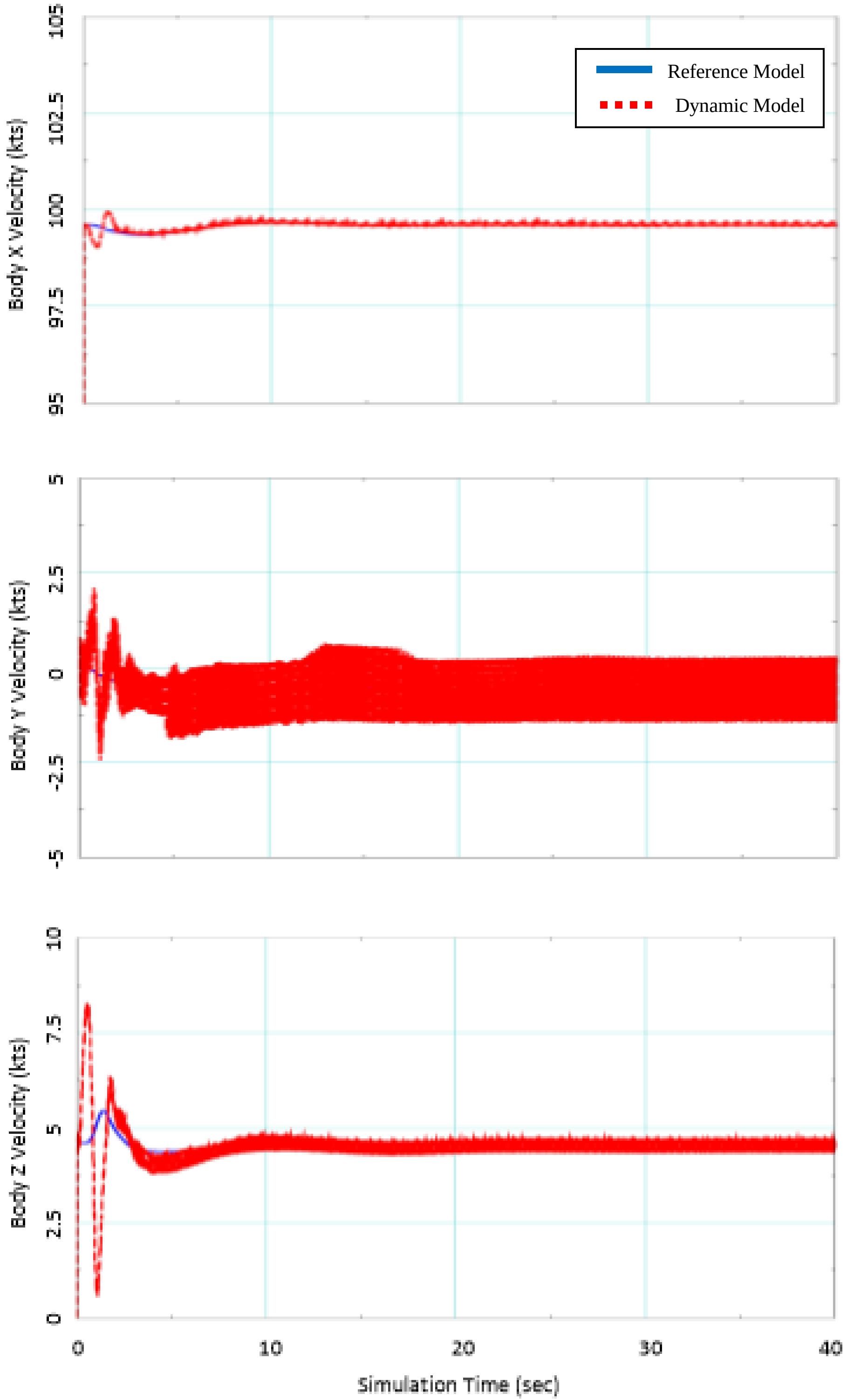


Figure 17: Body Velocities for 100kt Level Flight

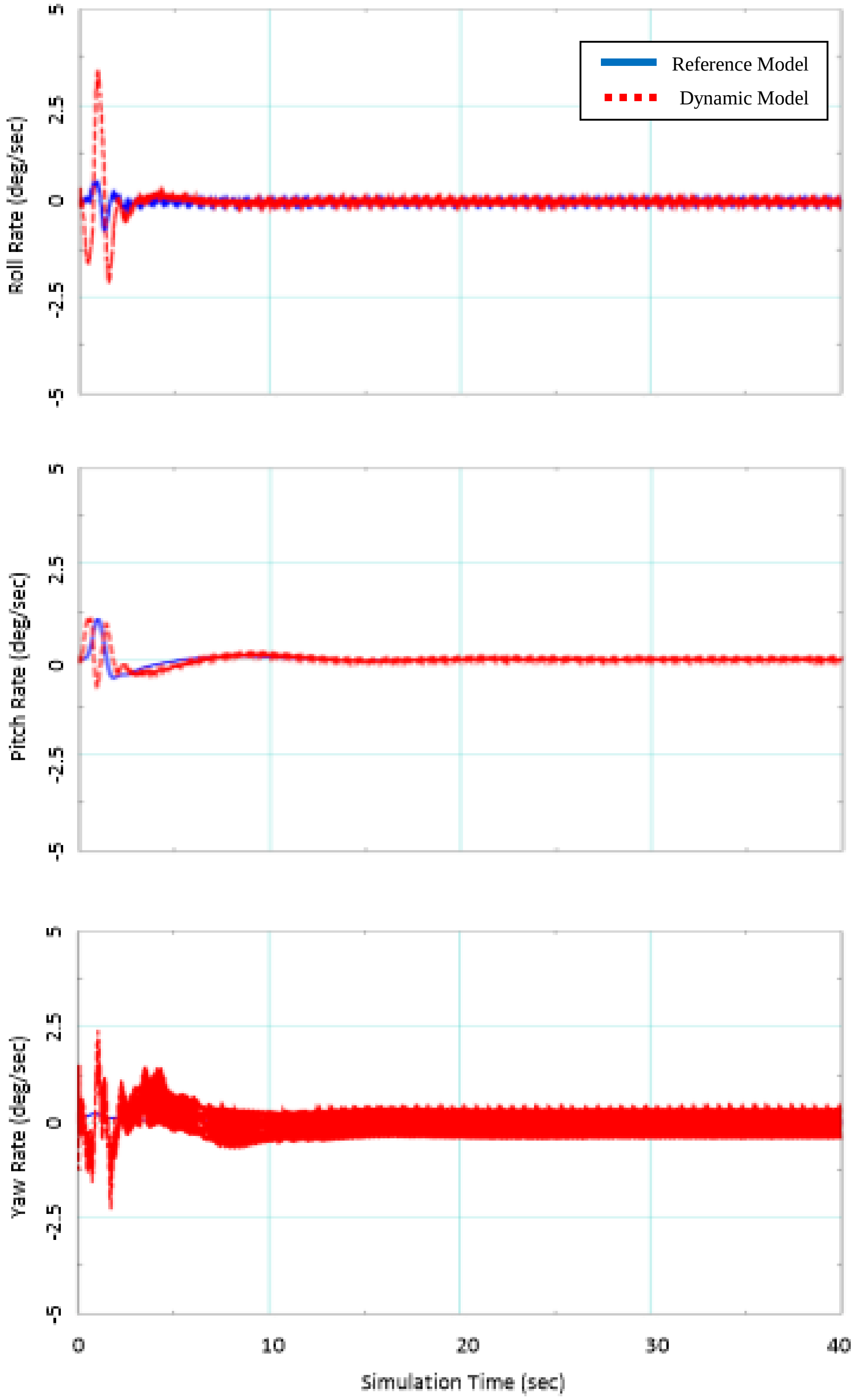


Figure 18: Body Angular Rates for 100kt Level Flight

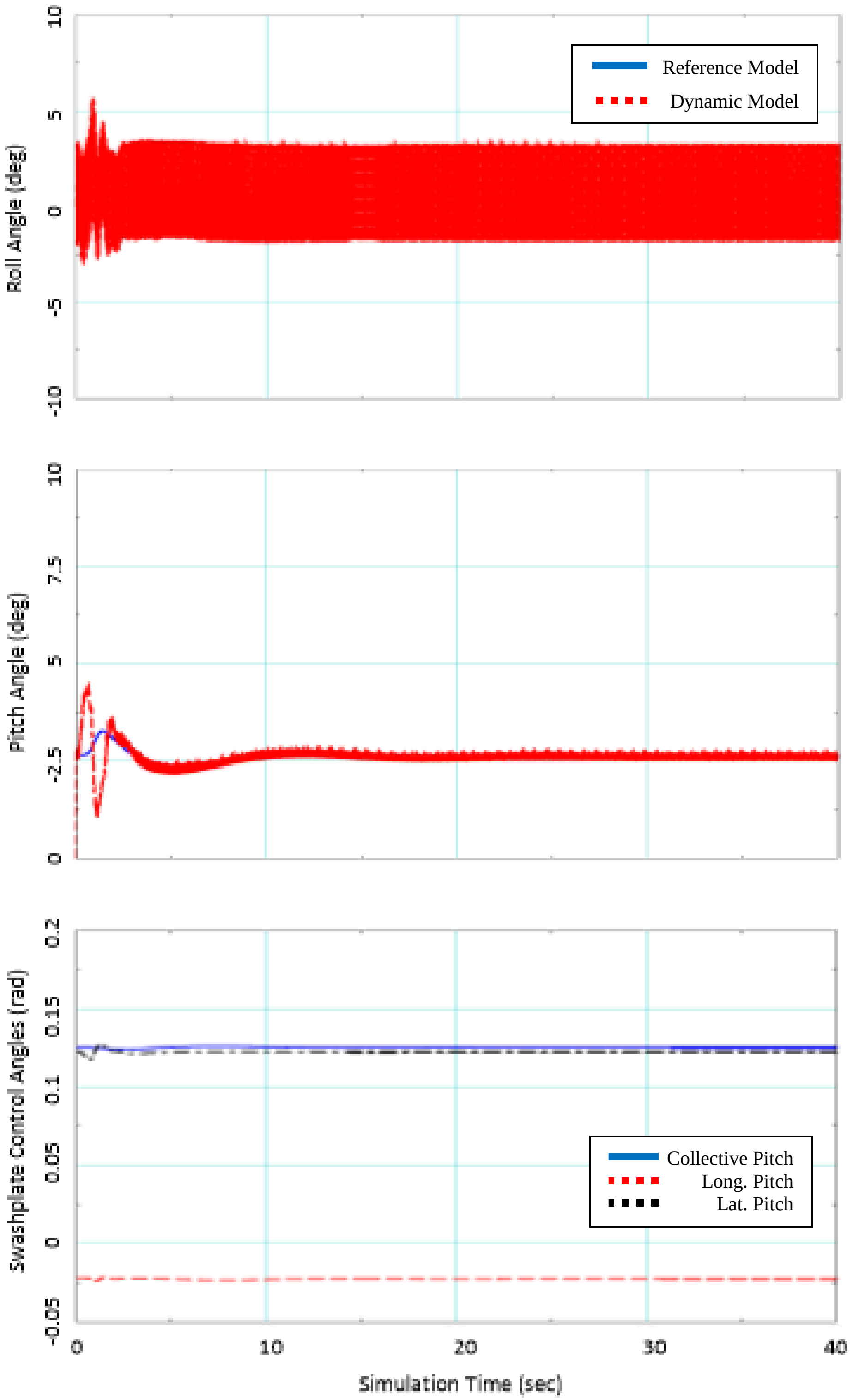


Figure 19: Orientation and Control Inputs for 100kt Level Flight

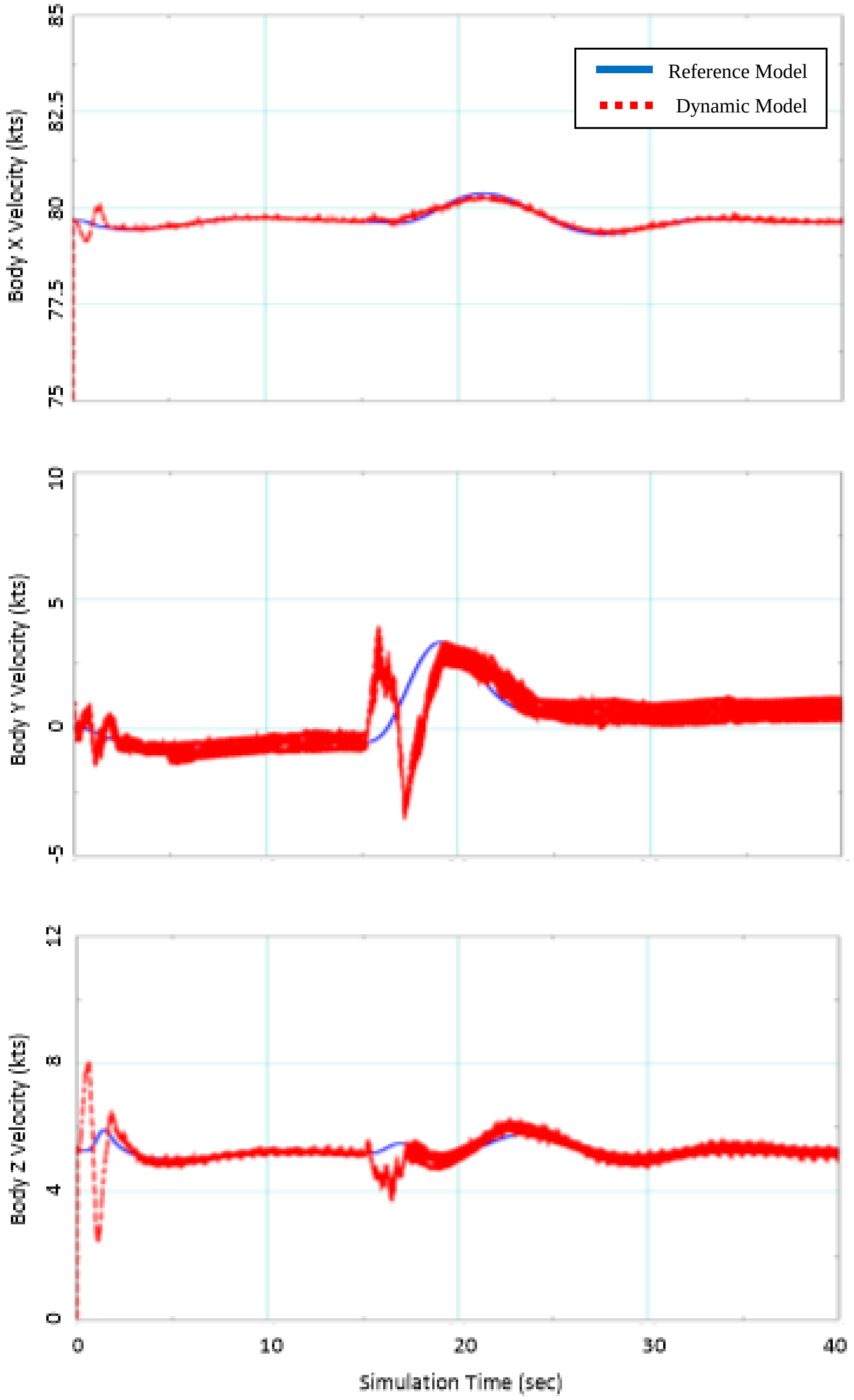


Figure 20: Body Velocities for 80kt Left Bank

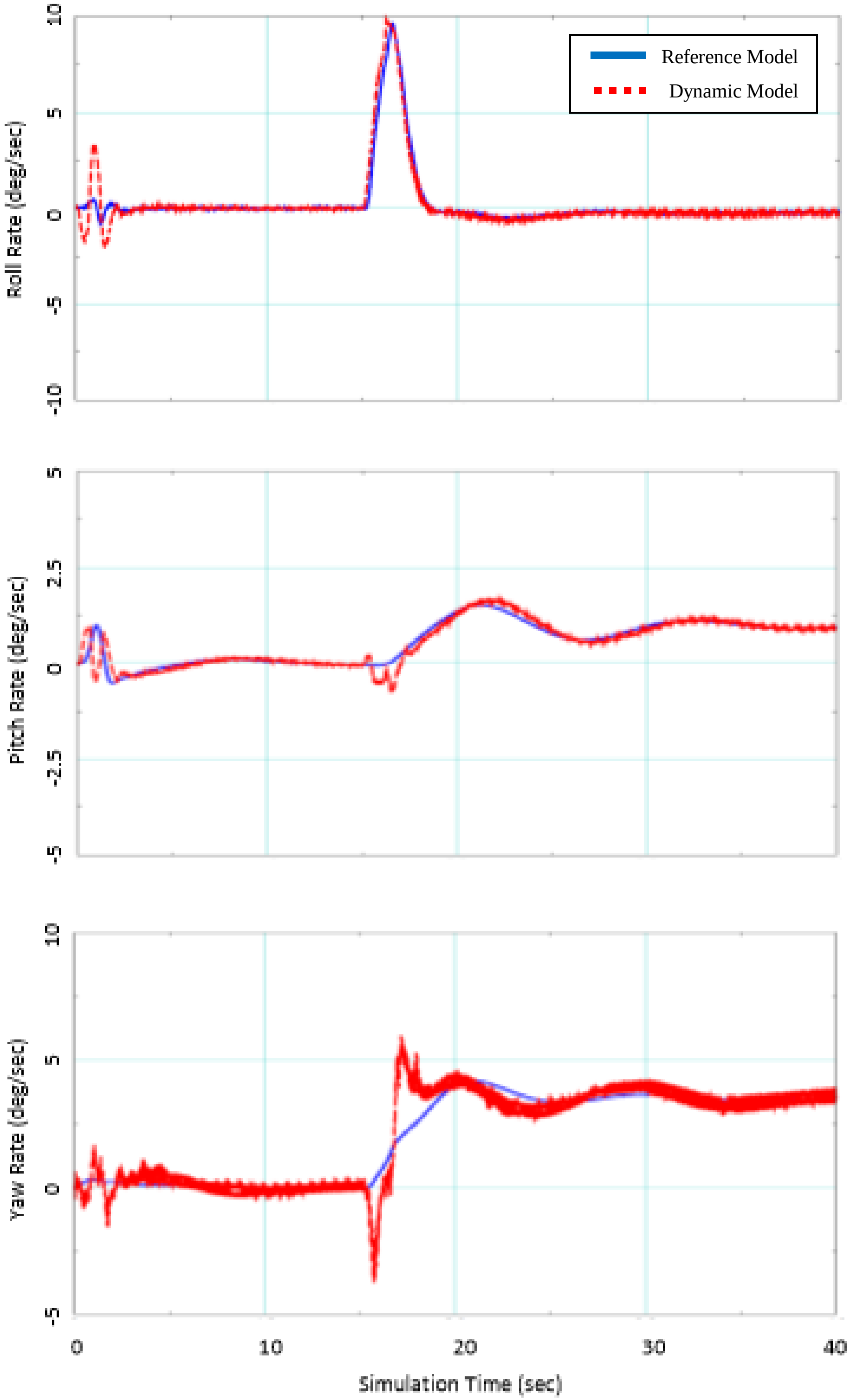


Figure 21: Body Angular Rates for 80kt Left Bank

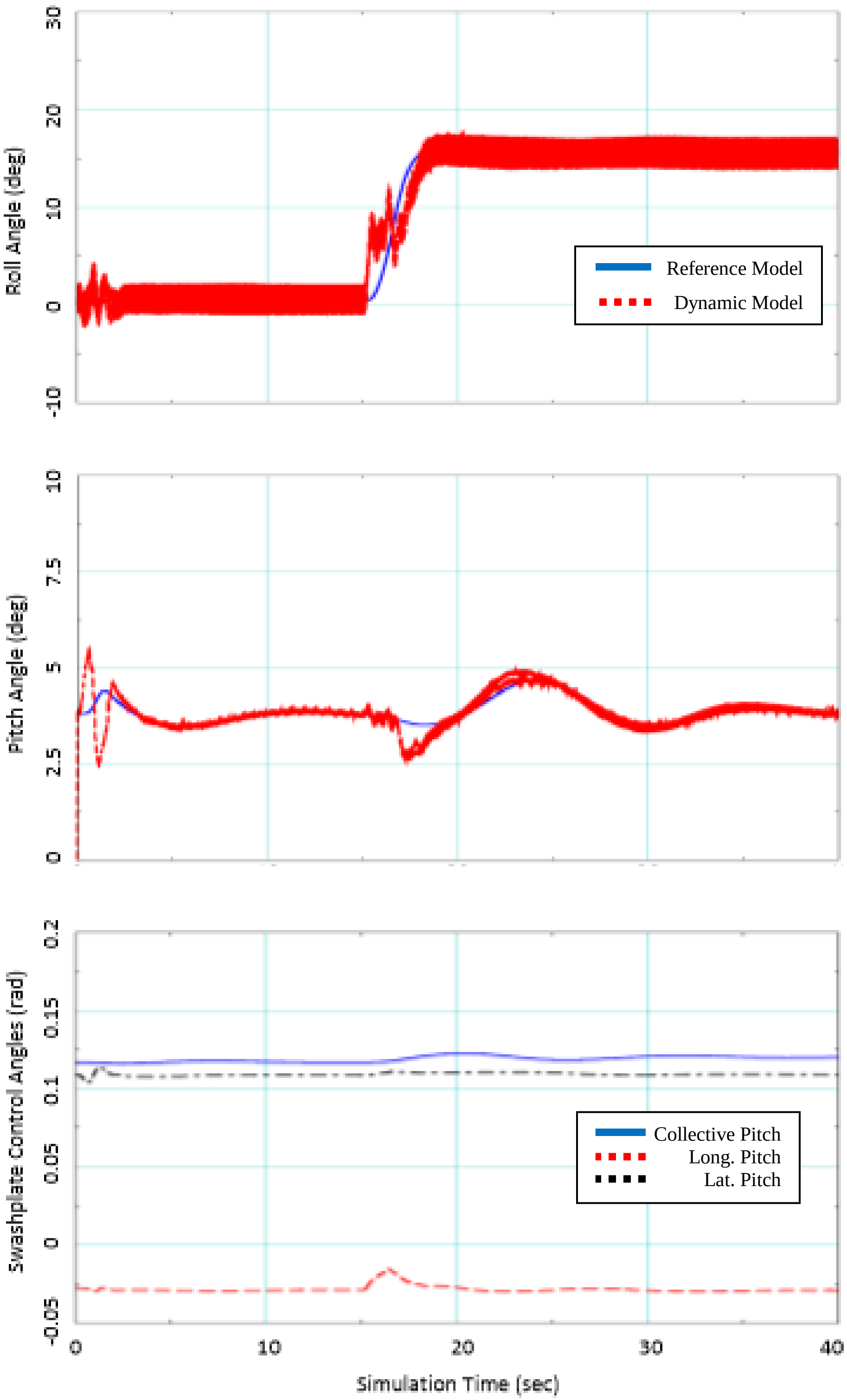


Figure 22: Orientation and Control Inputs for 80kt Left Bank

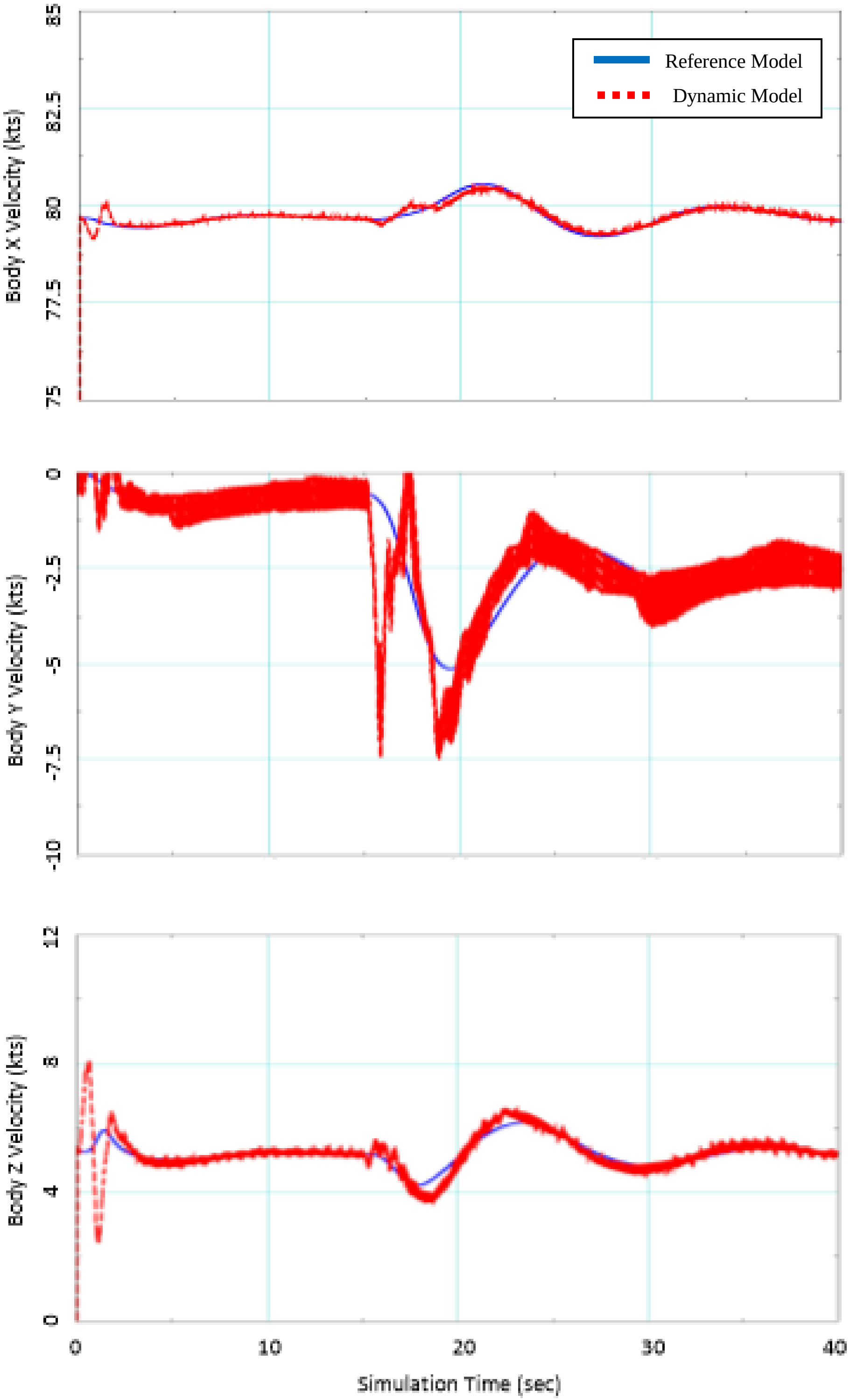


Figure 23: Body Velocities for 80kt Right Bank

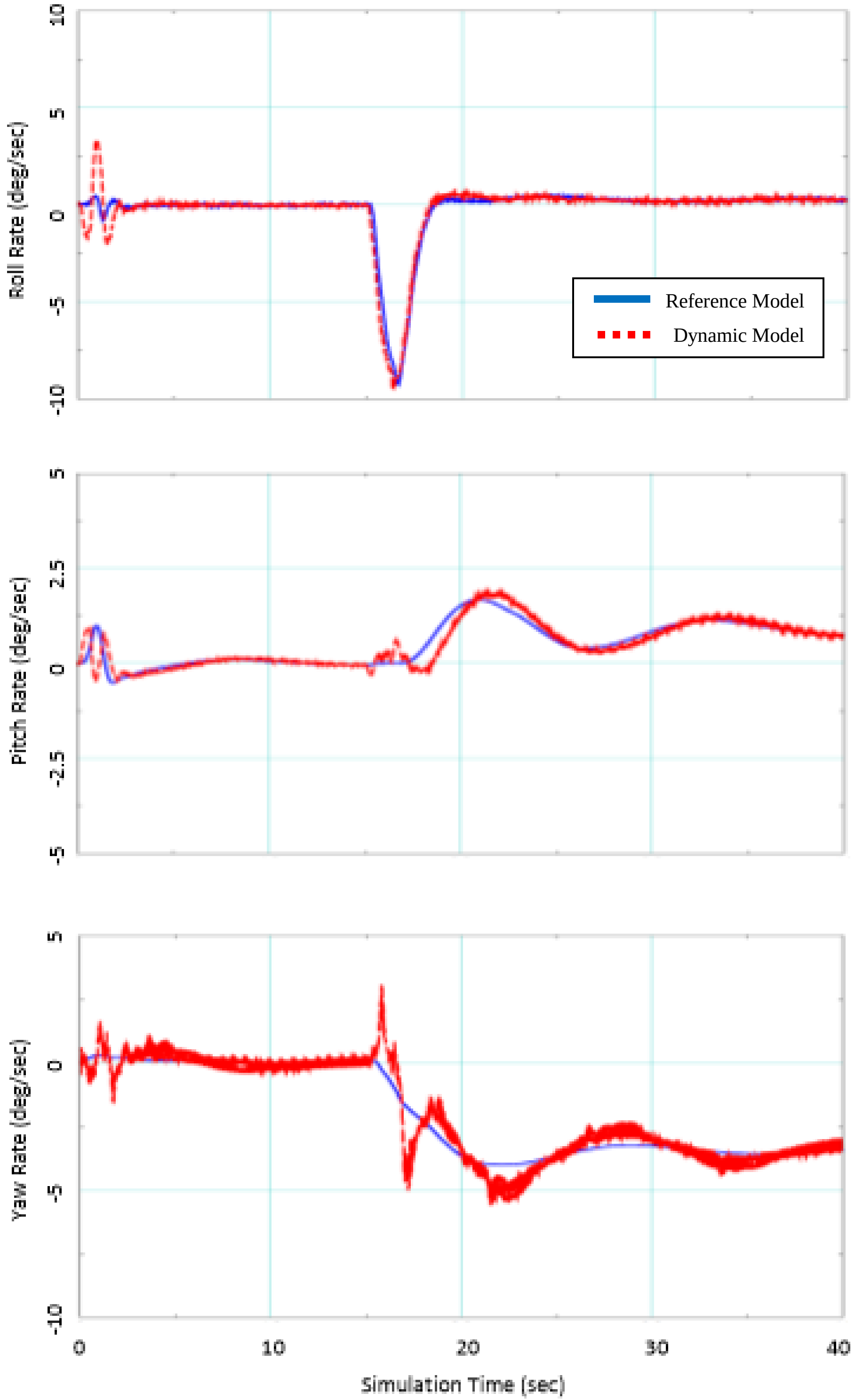


Figure 24: Body Angular Rates for 80kt Right Bank

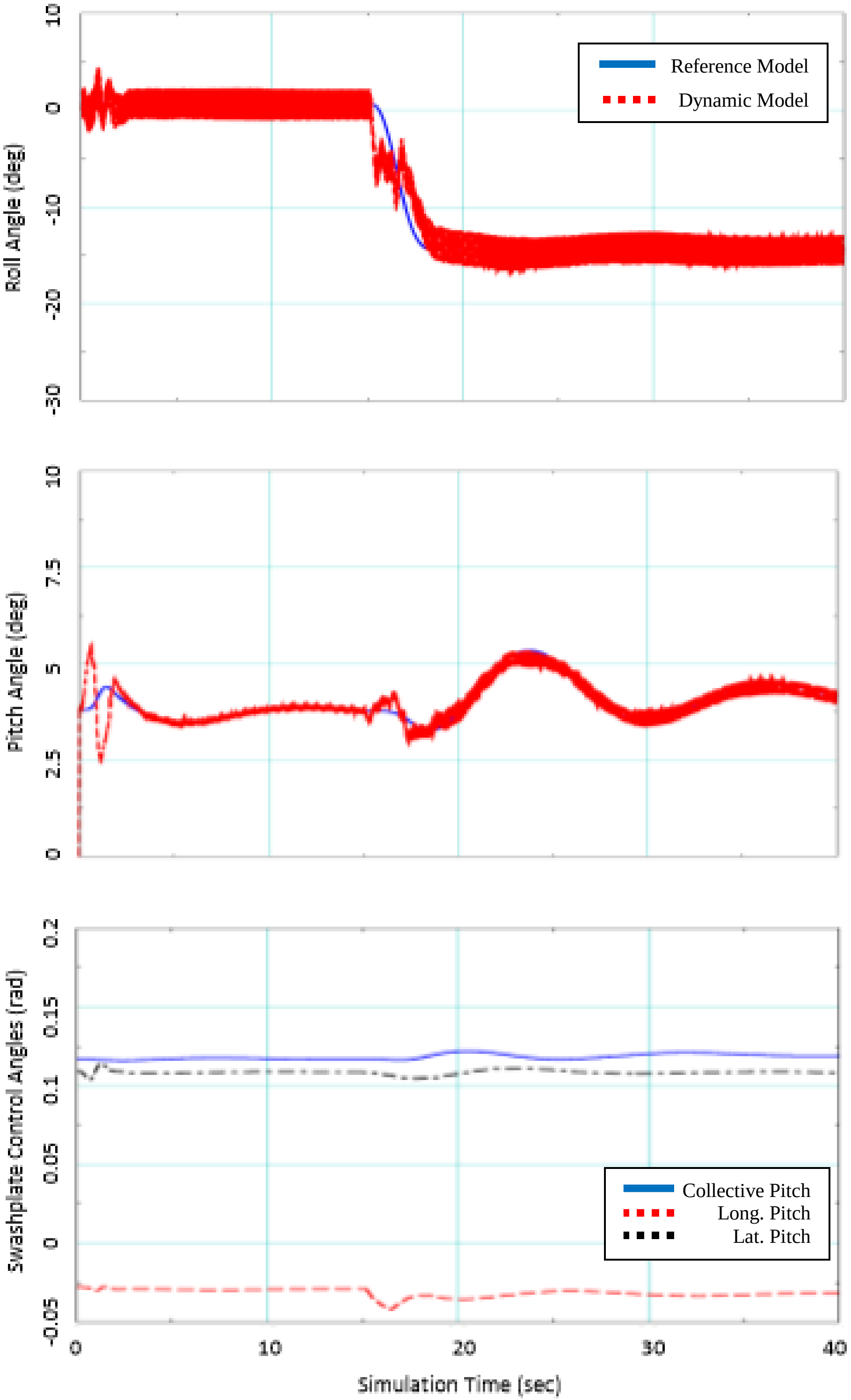


Figure 25: Orientation and Control Inputs for 80kt Right Bank

The plots demonstrate a comparison between the reference model and the dynamic model. The information from the reference model was used to project 2 seconds into the future and determine a value of the state examined. The model does struggle with the roll component as there is typically a 2-3 degree margin of error left or right, creating a noisy result.

The three indicated airspeeds examined all have a steady-state response for the 10-state vector. It should be noted for testing purposes, there is initial error in the reference model due to numerical discrepancies between the VirtualPilot and FLIGHTLAB® model. Therefore, all cases should use a 50-second runoff period to settle the model and eliminate discrepancies. Error may now be added to the system to evaluate the sensitivity of the model with biases and perturbations.

3.4. Coupled Driveshaft Failure Model

A driveshaft failure occurs when the output torque applied by the engine provides zero torque to the main rotor system. The driveshaft pushes a sprag clutch as shown in Figure 26. When the driveshaft is powered, the spokes in the clutch drive the main rotor systems. When the main rotor is turning at a greater speed than the driveshaft at the point of the sprag clutch, the spokes simply slip, providing no torque to the rotor.

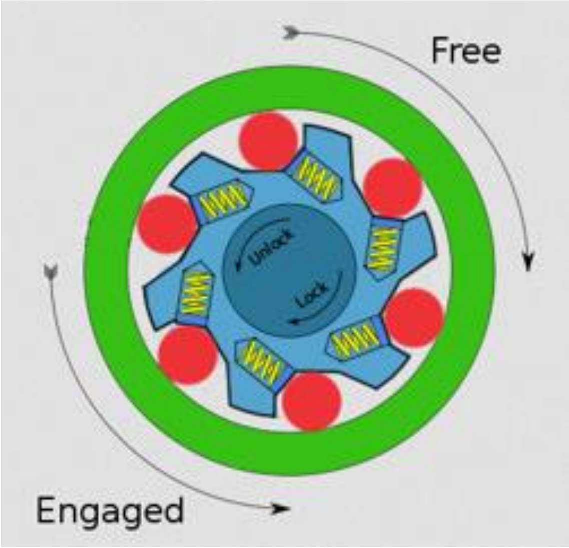


Figure 26: Sprag Clutch Components and Function
Reprinted with permission from Bert Vogel Account on *Teacher Webspace*, 2017, Retrieved from http://hrsbstaff.ednet.ns.ca/bvogel/K100/download/Starter_clutch/

Modelling of the driveshaft failure uses the concept of the sprag clutch in order to disengage the driveshaft from the main rotor. The engine model piece of the coupled dynamic model, described in Equation (26) through Equation (28), describes the forces of the rotorcraft on to the engine in the state matrix and the forces of the engine on to the rotorcraft in the control matrix. To simulate the drivetrain failure, the diagonal of the state matrix (engine component) will be set to zero, causing the engine model to spin up with no anti-torque being applied from the rotor system to the engine. This is shown below in Equation (50). Only the engine components of the state matrix is set to zero since the other states will continue to be driven. The control matrix in the engine component continues to add fuel to the system as if there is no error.

$$A_{DrivetrainFail} = \begin{bmatrix} 0 & \cdots & \cdots & 0 & \frac{\partial \dot{u}}{\partial NP} \\ \vdots & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & \cdots & \cdots & \boxed{0} & 0 \\ \frac{\partial \dot{NP}}{\partial u} & \cdots & \cdots & \frac{\partial \dot{NP}}{\partial NG} & \boxed{0} \end{bmatrix} \qquad \textbf{(50)}$$

3.5. Coupled Fuel Flow Failure Model

A fuel flow failure occurs when the engine ceases to provide torque to the driveshaft and the main rotor. When the engine does not provide the necessary torque to the main rotor, the sprag clutch becomes disengaged and the main rotor turns freely. The drag on the rotor blade will impede the rotor at a slower rate than the reduction in engine speed.

Modelling of the fuel flow failure calls upon removing the fuel flow from the engine component of the control. This is shown below in Equation (51). Every time step evaluated, the fuel flow to the engine component of the controller is set to zero, indicating that there is no assistance from the engine to the rotor system.

$$B_{FuelFail} = \begin{bmatrix} \frac{\partial \dot{u}}{\partial \theta_0} & \frac{\partial \dot{u}}{\partial \theta_{1cw}} & \frac{\partial \dot{u}}{\partial \theta_{1sw}} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \boxed{0} \\ 0 & 0 & 0 & \boxed{0} \end{bmatrix} \qquad \textbf{(51)}$$

3.5.1. Bias and Turbulence in Coupled Dynamic Model

A necessary procedure to conduct prior to examining the test matrix outlined in the coming sections is to run a sensitivity analysis on the model along with the Bayesian output. The procedure is run with the intention to prevent misleading results due to slight numerical biases within a simulation or the detection of false positives due to wind gusts or physical disturbances.

The variables analyzed will be the translational rates of the aircraft (u, v, w) from steady level flight. This will be applied in two separate manners. The first will be an additional bias added towards a steady state system. Once the system begins to calculate an approximation and multi-model estimate of a failure probability, a negative 5 knot bias to u, a positive 3 knot bias to v, and a positive 10 knot bias to w will be applied simultaneously. This will cause a small system error between the speed and the control inputs to the swashplate model. The second will be a simulated change within the FLIGHTLAB® environment. This is to ensure the model does not sense an engine failure if the VirtualPilot opts to change its trimmed flight conditions to account for an impulse input from the steady state.

Figure 27 and Figure 28 demonstrate the two methods used to test the model sensitivities and how they relate to a Bayes' Theorem result. The tests were all run at 80 kts trimmed speed. Sensitivity analysis at 60 kts and 100 kts demonstrate similar trends and results.

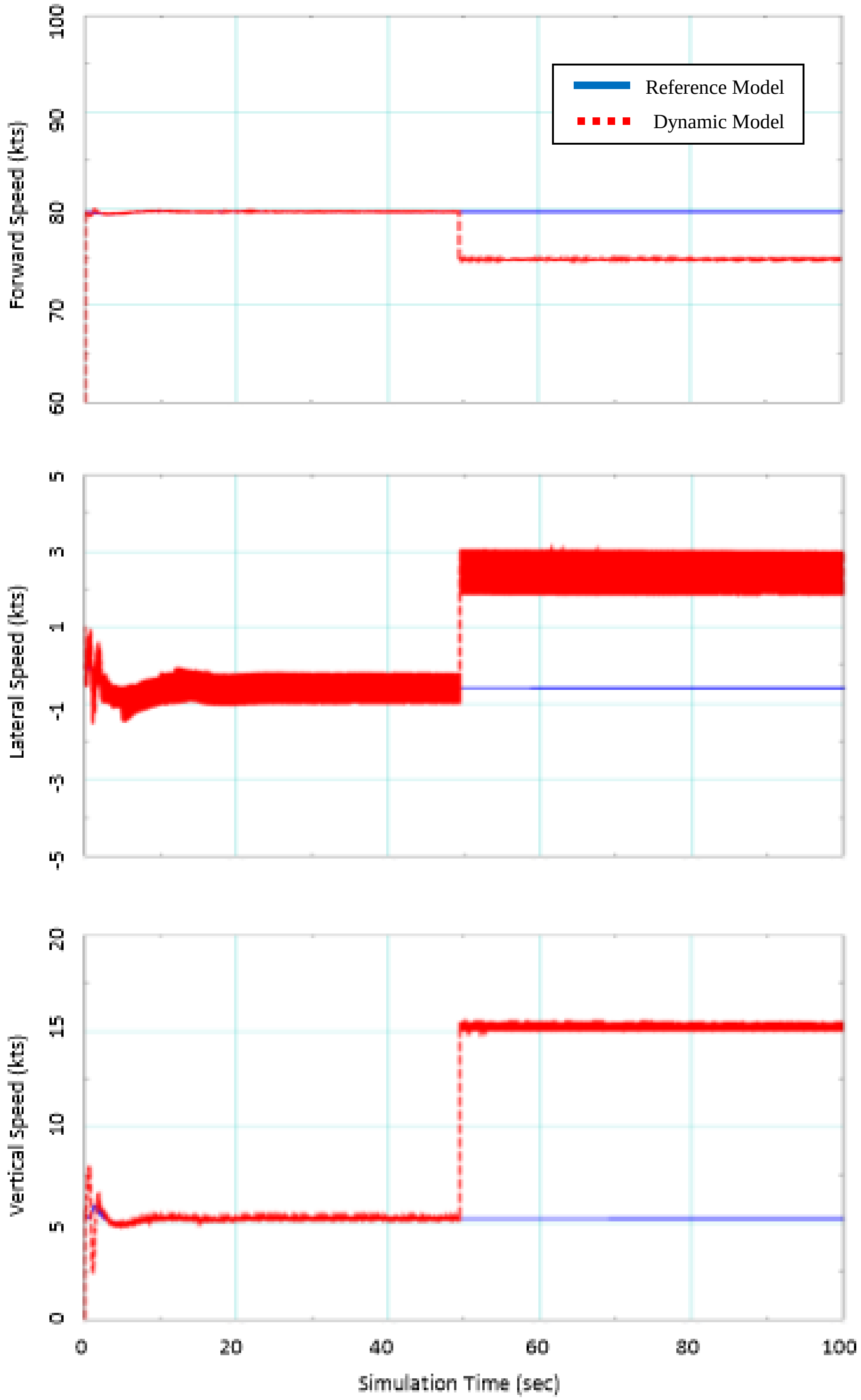


Figure 27: Applied Numerical Bias to Body Axis Velocities

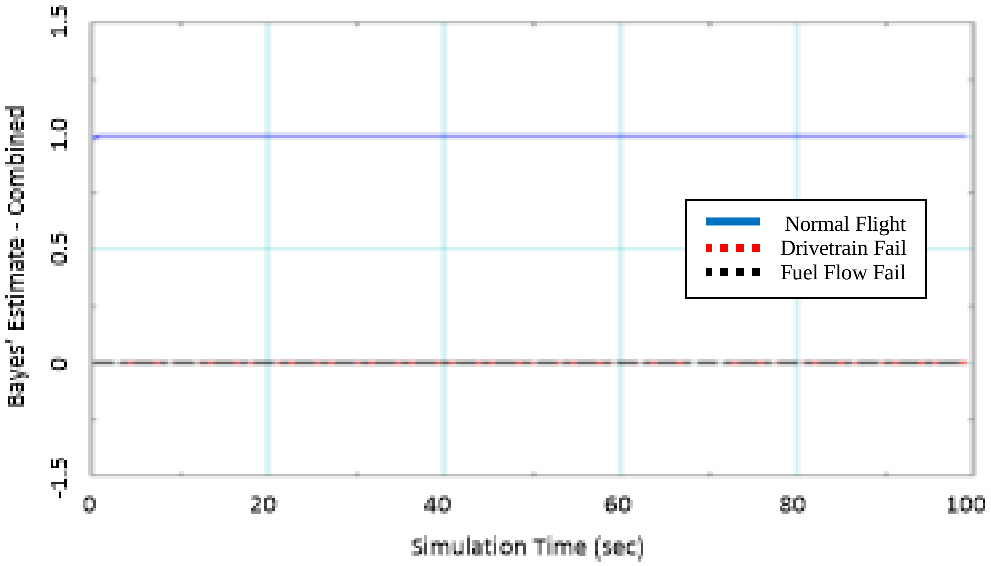


Figure 28: Bayesian Response to Body Velocity Biases

When a bias is added to either the forward, lateral, or vertical airspeed, indicating a reading error in a sensor, the Bayesian estimation did not show a noticeable change in probability of a normal, stable flight. A numerical bias to the instrumentation as an observation into the three hypothesis models does not cause the system to register false positive readings of engine or drivetrain failure.

The next set of plots represent a perturbation in the system. This is to simulate hitting a wind pocket or some type of density change. The perturbation is simulated through the VirtualPilot response in stick position to be used in the reference model. Figure 29 through Figure 31 apply a disturbance to the longitudinal, lateral, and collective axes, track the swashplate model positions, and follow the Bayesian estimation for each axes to examine the effects of the disturbance.

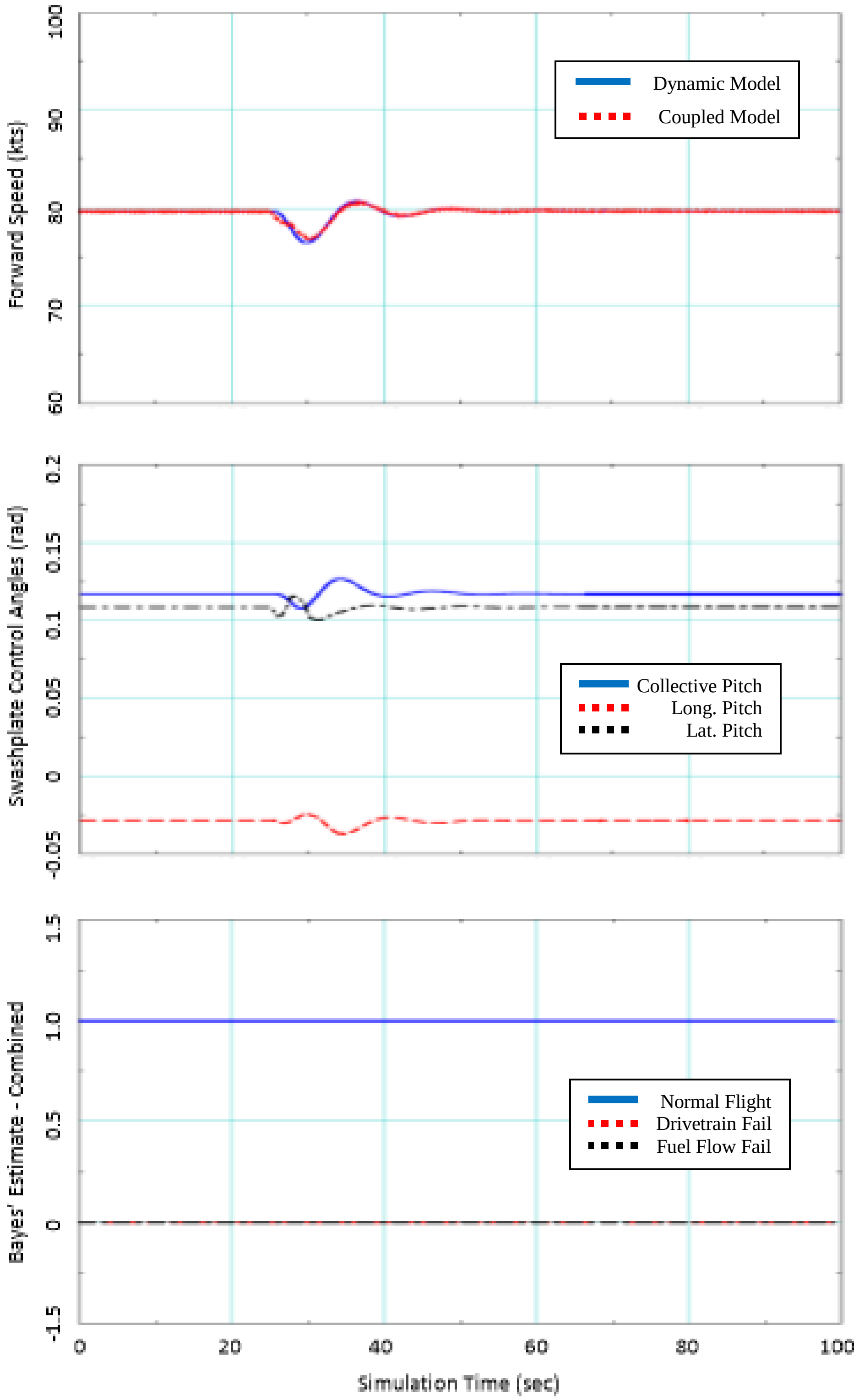


Figure 29: Model and Bayesian Response to a Longitudinal Disturbance

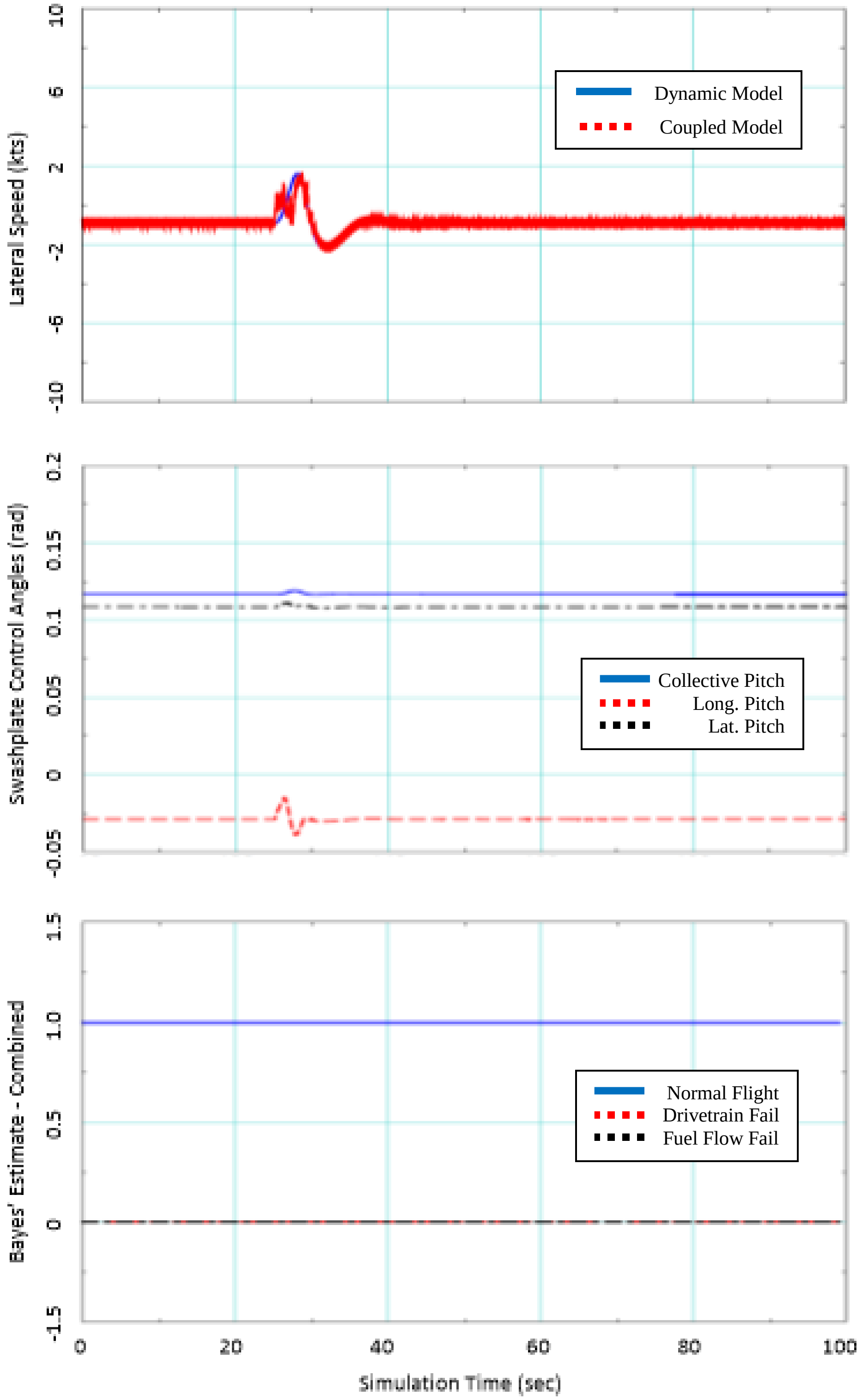


Figure 30: Model and Bayesian Response to a Lateral Disturbance

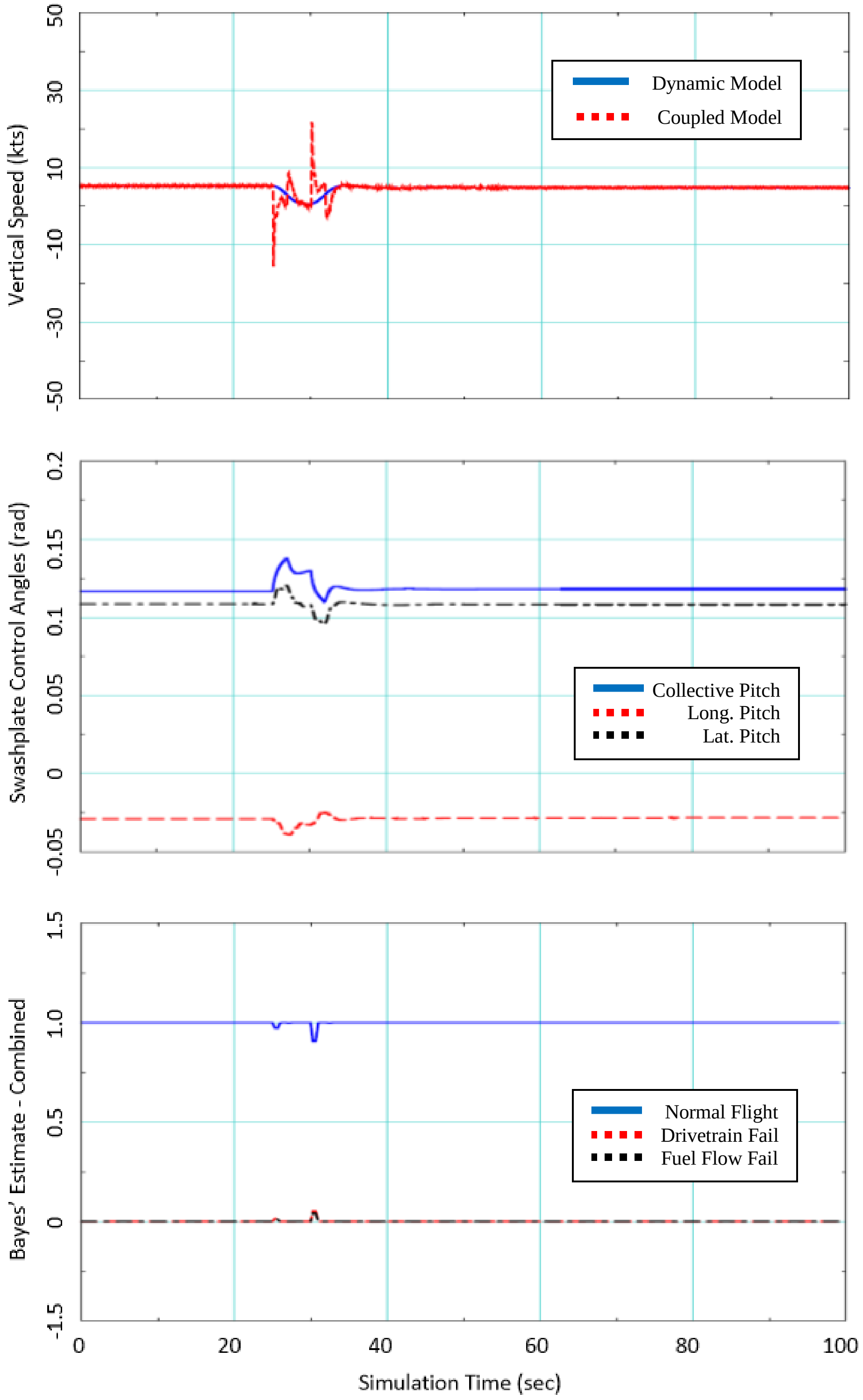


Figure 31: Model and Bayesian Response to a Collective Disturbance

The figures demonstrate the sensitivity, yet quick recovery of the dynamic model in the Bayes' analysis of the system health. The longitudinal and lateral disturbances did not register a change in the status of the normal model probability. The collective disturbance responded to the input, however, the change in the normal or failure models was not significant enough to detect a concern. In all cases with a numerical bias or disturbance in all axes, there were no significant responses to detect a failure.

3.6. Engine Failure with Non-Recovery

The engine failure analysis has two primary inputs for failure; the dynamic model failure which is triggered by a disengagement of the sprag-clutch driving the main rotor and the engine model failure which is triggered by a quick reduction in fuel flow. The section will evaluate the effects of an engine failure using both methods on the rotorcraft's dynamic states, compressor and turbine speeds, and main rotor speed along with evaluating the effects on the Bayes' analysis for detecting the failure. No pilot recovery actions are taken. The VirtualPilot model will maintain its forward speed, heading, and altitude as determined prior to the engine failure.

3.6.1. Instantaneous Engine Disconnect

In the case of a complete disconnect between the engine and main rotor, the fuel flow is instantaneously cut from the system. This is to simulate a failure between the power turbine and the main rotor, such as a sheared driveshaft. Figure 32 represents the fuel flow effects as pertaining to the main rotor. The sprag-clutch is disengaged, allowing the main rotor to act with no assistance from the engine.

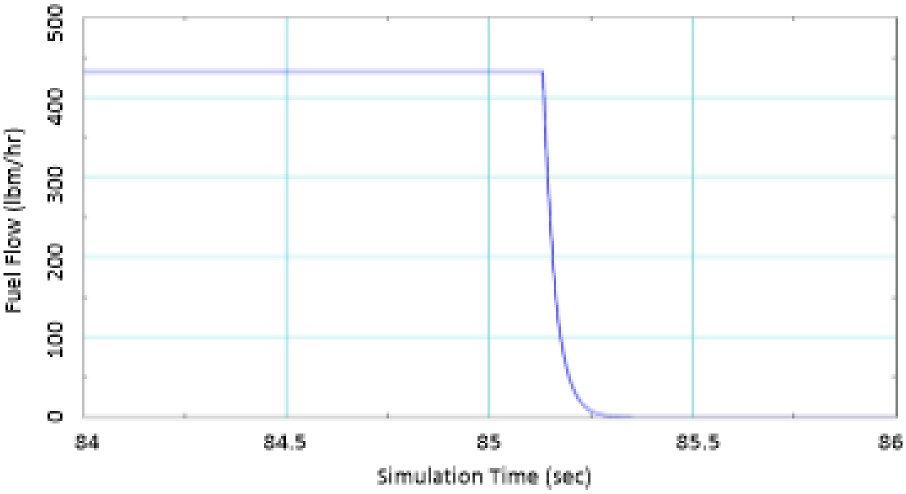


Figure 32: Main Rotor Fuel Flow for Instantaneous Failure

The next set of figures show the pilot input in order to maintain airspeed, heading, and altitude following the engine failure. Those inputs demonstrating a lack of response by a pilot to enter and autorotation are extrapolated into the dynamic response in body speeds, orientation, angular rates, and rotor speed, along with the gas generator and power turbine speeds.

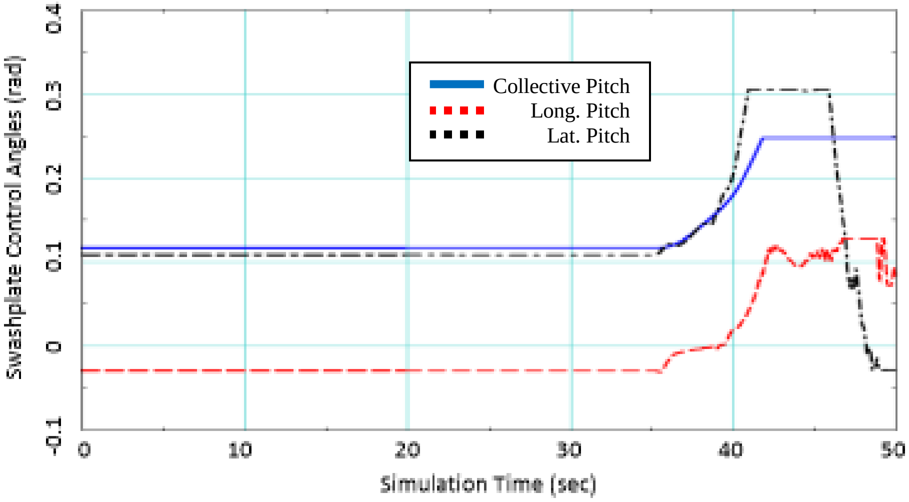


Figure 33: Model Control Inputs to Maintain Steady-State Flight

From the swashplate inputs above, the model will output other parameters for a flight with an engine failure and no pilot response. The plots are an example of how quickly an aircraft can become unstable along with the importance of pilot intervention at its earliest time step.

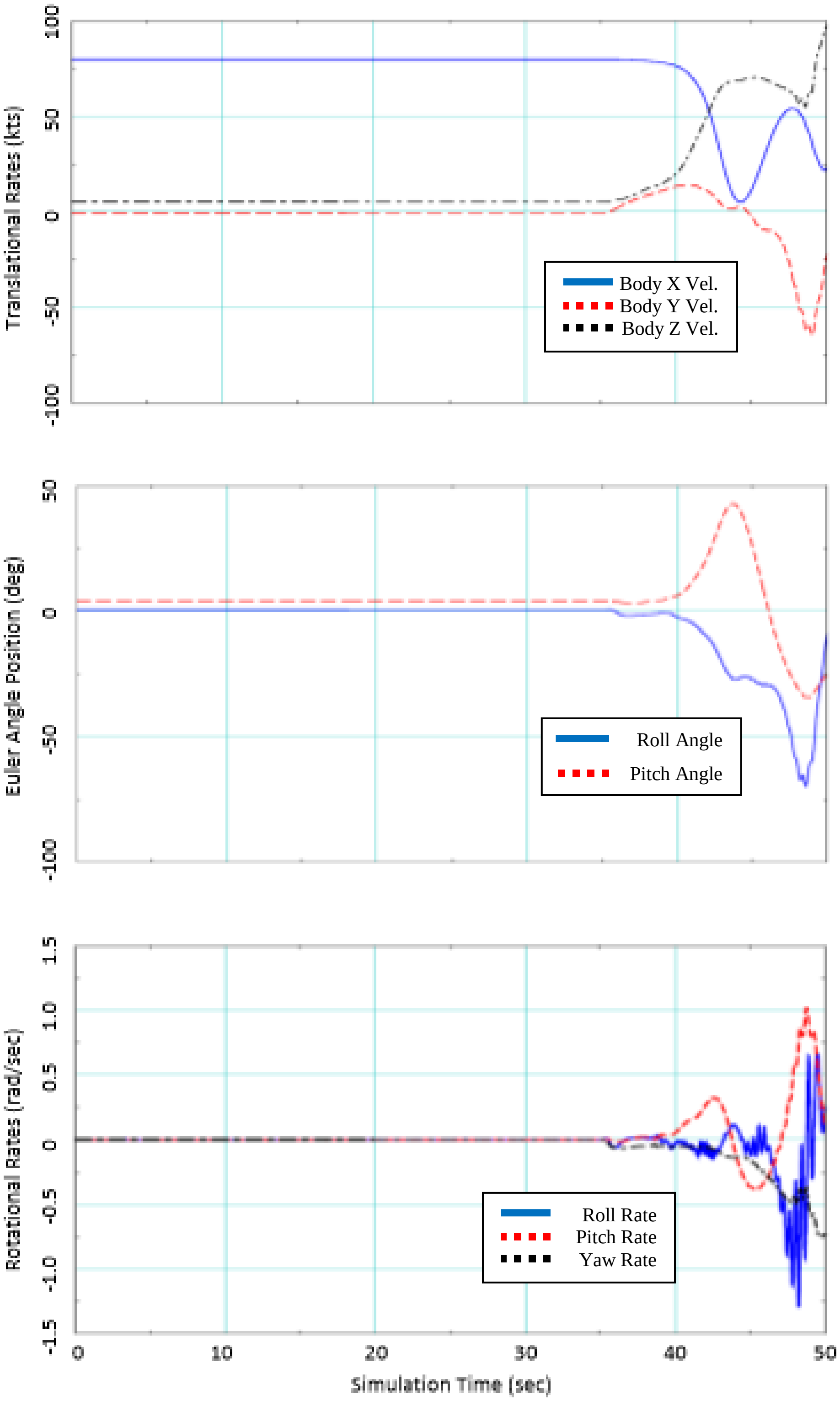


Figure 34: Angular Response to Instantaneous Failure

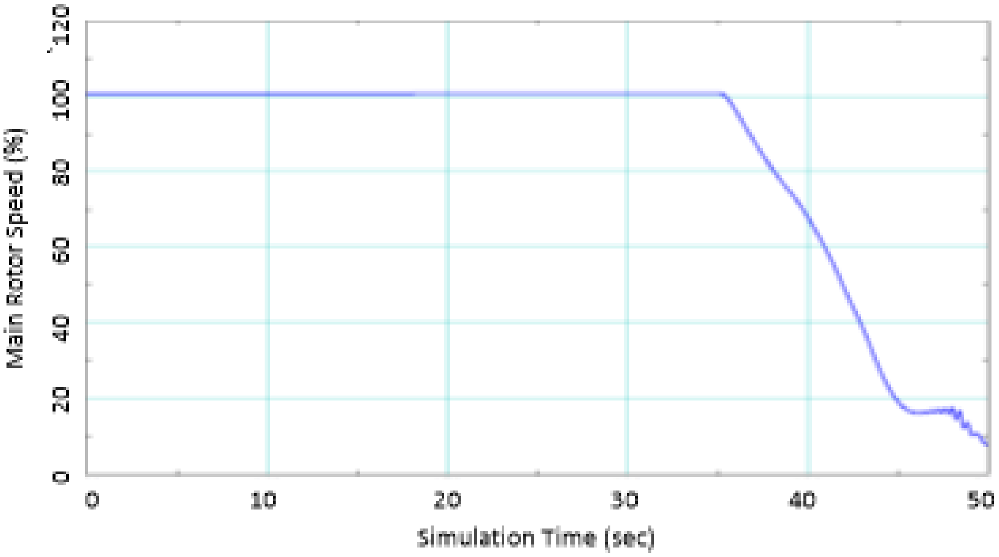


Figure 35: Rotor Speed for Instantaneous Failure

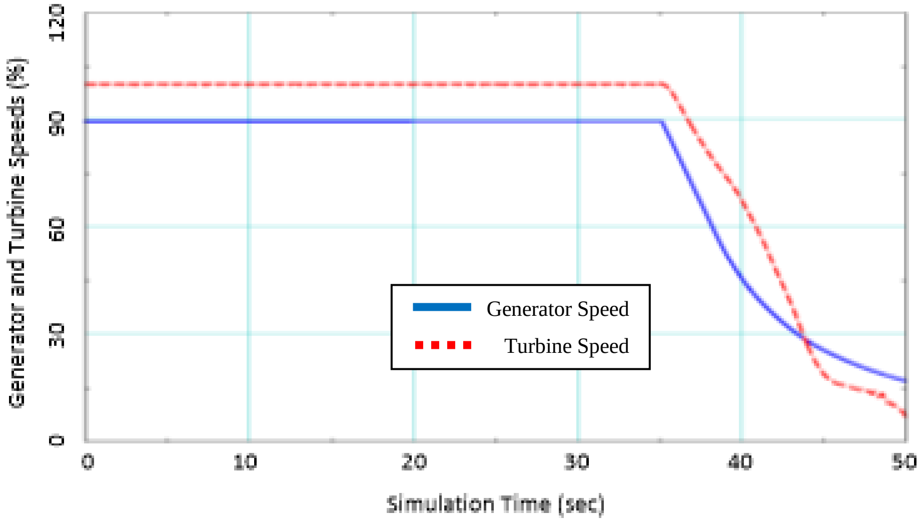


Figure 36: Generator and Turbine Speed for Instantaneous Failure

Note from Figure 34 the Euler angle along with the angular rates. A severe pitch angle followed by a tremendous roll angle have rendered the aircraft completely unstable.

The next figure portrays the quick degradation of rotor speed. Without the pilot's recognition and reaction, the rotor speed drops below 70% within approximately 5 seconds. Once below the 70% threshold, the time and altitude loss to regain enough rotor speed to land the aircraft becomes critical.

3.6.2. Engine Component Wind-Down

For a basic model of a T701 turboshaft engine, the fuel cutoff shows a similar shape to Figure 37. The two lines of interest in the figure are the "Turbine Speed Fuel Cut Off", highlighted with an orange arrow, and the "Turbine Speed Fuel Ramp Down", highlighted with a green arrow. The fuel to the turbine ramps down over a 4 second period. The dynamic model in FLIGHTLAB® will be programmed to do the same. As a result, the turbine speed slows over a 15-second period.

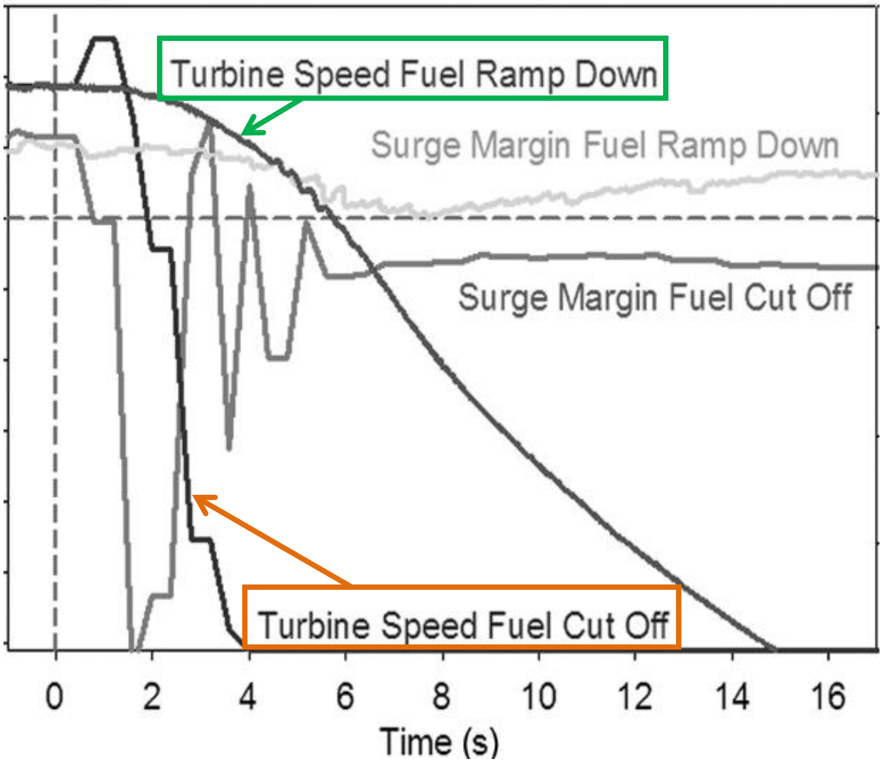


Figure 37: Engine Parameters for a Wind-Down Failure

Figure 38 shows the FLIGHTLAB® programmed fuel flow cut-off to simulate an engine failure. Similar to the previous figure, the fuel flow tapers to zero over a 4 second span. Figure 41 shows the gas generator and power turbine speed responses. As previously projected, the turbine speed slows at a comparable rate over a 15-second period.

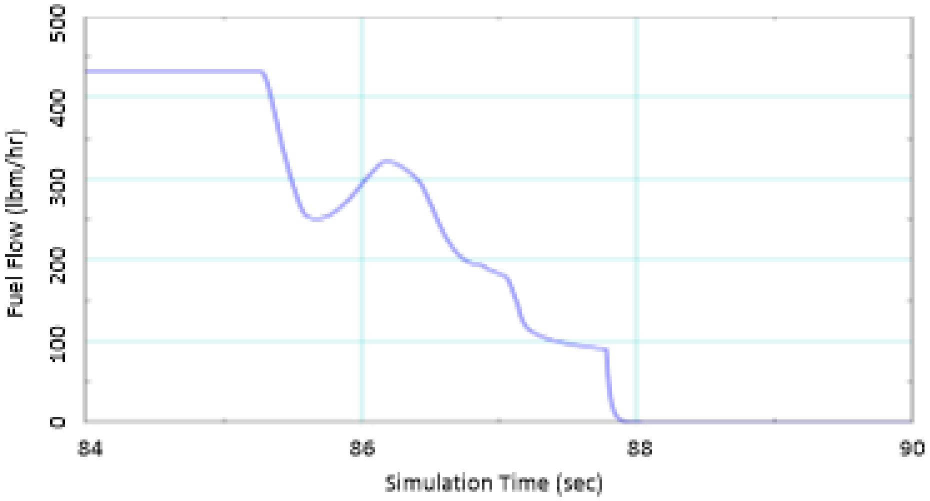


Figure 38: Engine Fuel Flow for Wind-Down Failure

The figure demonstrates the fuel flow of the rotorcraft as the throttle is reduced over a 4-second period. The model consists of two turboshaft engines using a loadsharing algorithm to ensure the proper torque is being sent to the driveshaft. The small bounces in the fuel flow curve are due to the two models arguing with each other on load distribution. However, over a 4-second time period, there is not a concern with the model performance. Figure 41 portrays the resultant gas generator and power turbine speeds based from the fuel flow in Figure 38. Similar to Figure 37, the power turbine speed powers down over a drawn-out 15-20 second duration. The model parameters to be used for a simplified Bayes' analysis for an engine appears to be appropriate.

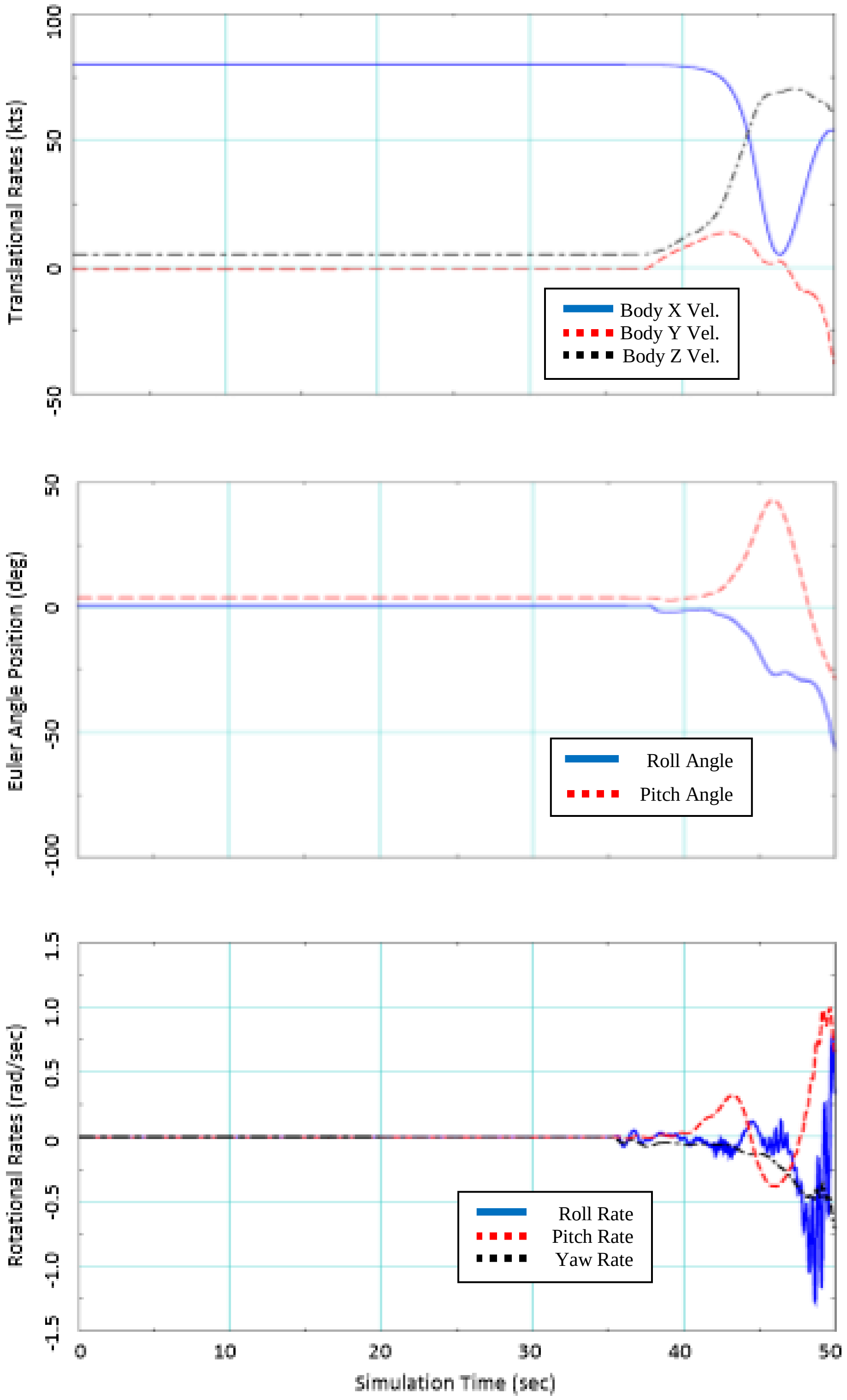


Figure 39: Angular Response for Wind-Down Failure

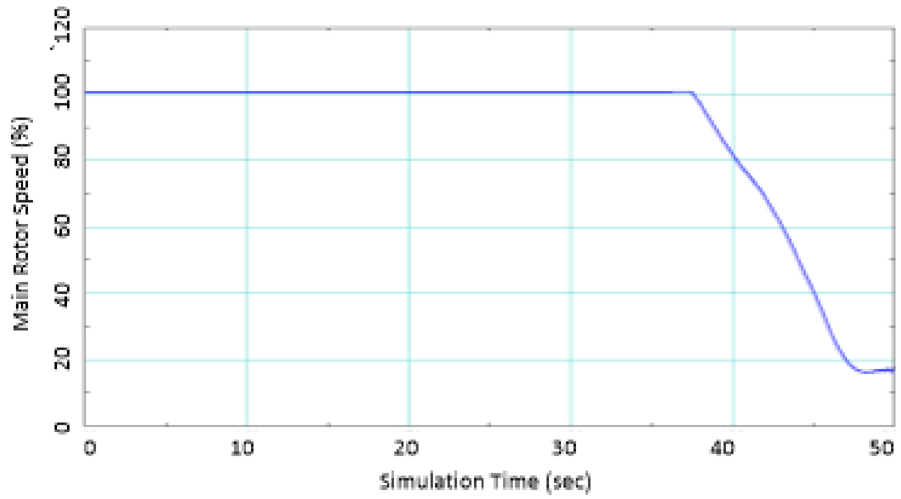


Figure 40: Main Rotor Speed for Wind-Down Failure

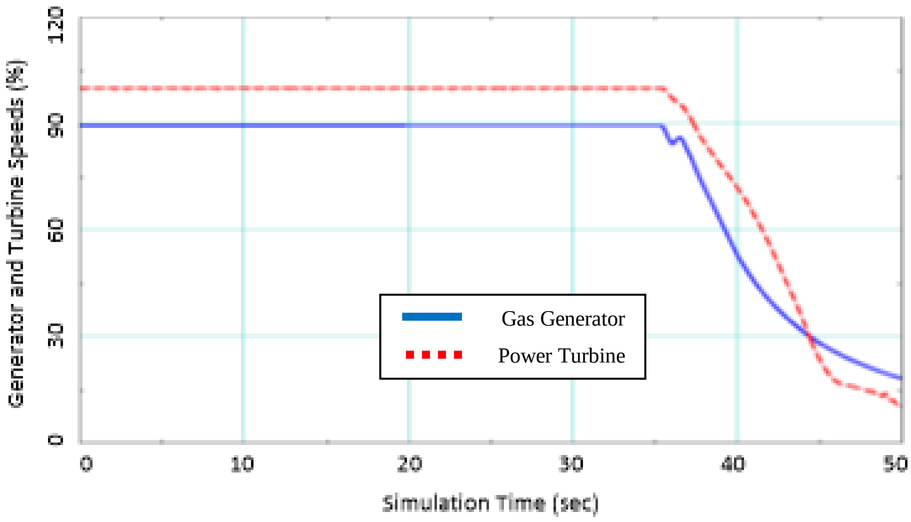


Figure 41: Generator and Turbine speeds for Wind-Down Failure

The wind-down model response to an engine failure with zero pilot response is like that of the instantaneous engine failure. There is about a second slower response, as seen in the main rotor speed in Figure 40. However, the rate of rotor speed decay appears to be similar. Dynamically, the model speeds and orientation respond in similar fashion to that of the Instantaneous engine failure.

3.7. Bayes' Analysis and Results

Figure 42 outlines the intention of the model to seek a predictive value of a performance or engine parameter. In the figure, the blue line represents the simulation data, which is run to completion. At each time step, a Kalman filter is applied to run the model a specified time forward. That value is used in the calculation for determining if the model is degrading due to an engine failure. Each time step is evaluated.

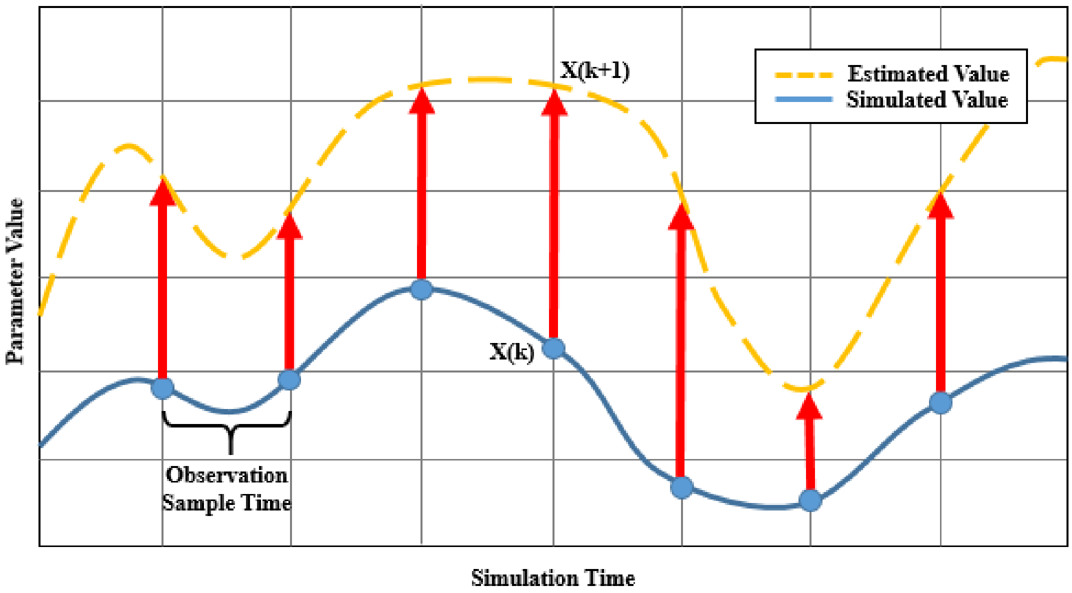


Figure 42: Kalman Filter Estimation

The methodology used in Section 1.4, Bayes' and multiple-model estimation, can be used to determine the probability of either a healthy flight or one involving an engine failure. As described in the section, Kalman filter components may be used to determine the probability of the observation vector given an updated model state vector. Equation (52) is the form the dynamic model must take to be applied in the Kalman filter [19].

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} + \Lambda\mathbf{w} \tag{52}$$

The derivation of the state matrix and control matrix is shown in Appendix A. The control vector consists of four input variables; collective pitch angle, longitudinal cyclic swashplate angle, lateral cyclic swashplate angle, and fuel flow rate. Figure 43, from [20], better demonstrates the location of the swashplate angle measurements. In an ideal situation, the control would trace back to the collective and cyclic stick inputs by the pilot. However, due to the non-consistent and proprietary flight control laws from varying helicopter platforms, along with the intense calculations between involving actuator position and dynamics and mixer dynamics, proceeding to the pilot stick inputs would distract from the scope of this paper.

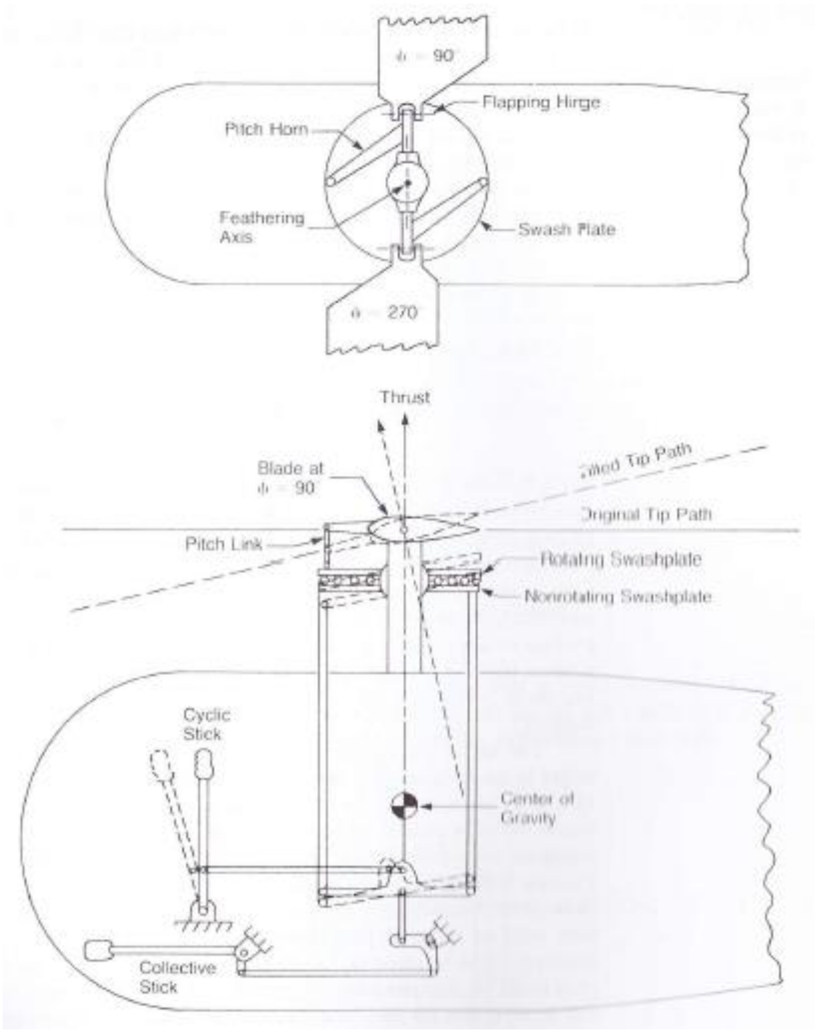


Figure 43: Swashplate Control Inputs

The following equation shows the observation matrix used in the normal model and two failure models. The variables observed are the forward speed, descent rate, roll and pitch angles, gas generator speed, and power turbine speed.

$$\mathbf{z} = \{u \quad w \quad \phi \quad \theta \quad NG \quad NP\}^T$$

Once again, the Equations (53-54) are used to compare the health of the dynamic aircraft model and the T701D engine model.

$$\Pr\left(\mathbf{p}_j \middle| \mathbf{z}_k\right) = \frac{pr\left[\mathbf{z}_k \middle| \hat{\mathbf{x}}_k\left(\mathbf{p}_j\right)\right]\Pr\left(\mathbf{p}_j \middle| \mathbf{z}_{k-1}\right)}{\sum_{i=1}^{l}\left\{pr\left[\mathbf{z}_k \middle| \hat{\mathbf{x}}_k\left(\mathbf{p}_i\right)\right]\Pr\left(\mathbf{p}_i \middle| \mathbf{z}_{k-1}\right)\right\}} \quad (53)$$

$$pr\left[\mathbf{z}_k \middle| \hat{\mathbf{x}}_k\left(\mathbf{p}_i\right)\right] = \frac{1}{(2\pi)^{n/2}\left|\mathbf{S}_k\left(\mathbf{p}_i\right)\right|^{1/2}} e^{-(1/2)\mathbf{r}_k^T(\mathbf{p}_i)\mathbf{S}_k^{-1}(\mathbf{p}_i)\mathbf{r}_k(\mathbf{p}_i)} \quad (54)$$

For the application of these equations, *I* is the number of hypothesis models. For the cases in question, the normal flight hypothesis model and two failure models are evaluated. The variable j is a single probability model and is defined as $1 \le j \le I$. The summation block upper limit, *l,* is the number of hypothesis models to be evaluated and compared to other models. All of the hypothesis models are 10-state models; the (3x1) linear velocity vector, the (3x1) angular velocity vector, the (2x1) angular position vector, and the (2x1) engine model. The above equation may be re-written as:

$$\Pr\left(\mathbf{p}_j \mid \mathbf{z}_k\right) = \frac{\frac{1}{(2\pi)^{1/2}\left|\mathbf{S}_k\left(\mathbf{p}_j\right)\right|^{1/2}} e^{-(1/2)\mathbf{r}_k^T(\mathbf{p}_j)\mathbf{S}_k^{-1}(\mathbf{p}_j)\mathbf{r}_k(\mathbf{p}_j)} \Pr\left(\mathbf{p}_j\right)_{k-1}}{A + B + C} \quad (55)$$

where

$$
\begin{aligned}
A &= \frac{1}{(2\pi)^{1/2}\left|\mathbf{S}_k(\mathbf{p}_{norm})\right|^{1/2}} e^{-(1/2)\mathbf{r}_k^T(\mathbf{p}_{norm})\mathbf{S}_k^{-1}(\mathbf{p}_{norm})\mathbf{r}_k(\mathbf{p}_{norm})} \Pr(\mathbf{p}_{norm})_{k-1} \\
B &= \frac{1}{(2\pi)^{1/2}\left|\mathbf{S}_k(\mathbf{p}_{DriveShaft})\right|^{1/2}} e^{-(1/2)\mathbf{r}_k^T(\mathbf{p}_{DriveShaft})\mathbf{S}_k^{-1}(\mathbf{p}_{DriveShaft})\mathbf{r}_k(\mathbf{p}_{DriveShaft})} \Pr(\mathbf{p}_{DriveShaft})_{k-1} \\
C &= \frac{1}{(2\pi)^{1/2}\left|\mathbf{S}_k(\mathbf{p}_{eng})\right|^{1/2}} e^{-(1/2)\mathbf{r}_k^T(\mathbf{p}_{eng})\mathbf{S}_k^{-1}(\mathbf{p}_{eng})\mathbf{r}_k(\mathbf{p}_{eng})} \Pr(\mathbf{p}_{eng})_{k-1}
\end{aligned}
\qquad (56)
$$

The resultant value is the probability of one of the three hypothesis being most correct at the given time step of k. The above equation may also be represented by Figure 44, modified from the original version provided in Figure 4.

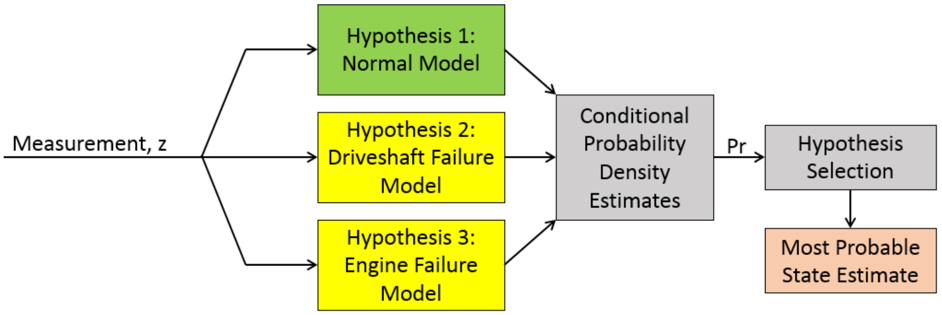

Measurement, z
Hypothesis 1: Normal Model
Hypothesis 2: Driveshaft Failure Model
Hypothesis 3: Engine Failure Model
Conditional Probability Density Estimates
Pr
Hypothesis Selection
Most Probable State Estimate

Figure 44: Hypothesis Model Flow Chart for 3-Model System

The figure shows the hypothesis models as either a normal model in which the system is healthy, or one of the two failure models where the driveshaft or engine has failed. The probability of each hypothesis model calculated from Equation (55) is evaluated to determine the hypothesis model that is most likely to be correct.

The calculation from Equations (55-56) can present many obstacles for determining the probability of an engine failure. The primary source of concern resides in the determinant of S_k, due to the size of S_k. If the model is tuned improperly, the main diagonal of S_k is liable to have some barely non-zero numbers. The first three fields in the state matrix are in knots. The next five are either radians per second or radians. The final two are in revolutions per minute.

The development of the three hypothesis models allow a user to choose the hypotheses models that is most likely to be correct and determine the probability that an event occurred given very little live flight test data.

3.7.1. Bayes' Model Selection and Threshold

As referenced in Chapter 2, three models are evaluated simultaneously using Bayes' theorem. The criteria for detecting a probable engine or driveshaft failure for a rotorcraft occurs when the normal function model no longer has the greatest probability of being the correct model. This state occurs when the driveshaft failure model which indicates the engine is functioning but the main rotor is not being driven, or the engine failure model which indicates the engine is no longer providing torque to the rotorcraft system has a greater probability of being correct.

The first case examined is the instantaneous model failure. This failure indicates a driveshaft failure where the engine and main rotor system are immediately severed from each other. The models were evaluated at 60 kts, 80 kts, and 100 kts of forward speed. Figure 45 through Figure 47 show the results of the three cases. The vertical green line indicates the instant the failure occurs. This is the instant from which the pilot reaction time is measured. The vertical orange line is the moment normal model is less probable to be correct than either failure models. This is the detection point. The horizontal green line is the probability of the of the normal model at the detection point.

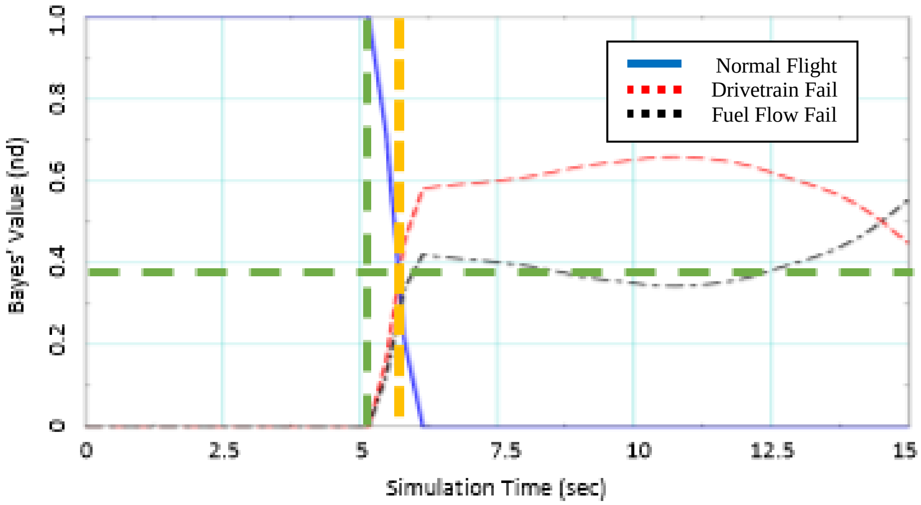


Figure 45: Failure Recognition for 60 kts Instantaneous

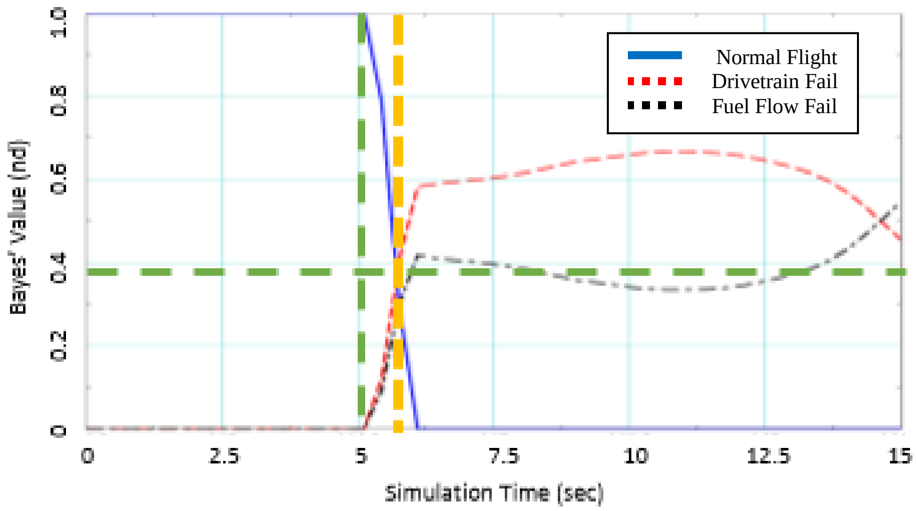


Figure 46: Failure Recognition for 80 kts Instantaneous

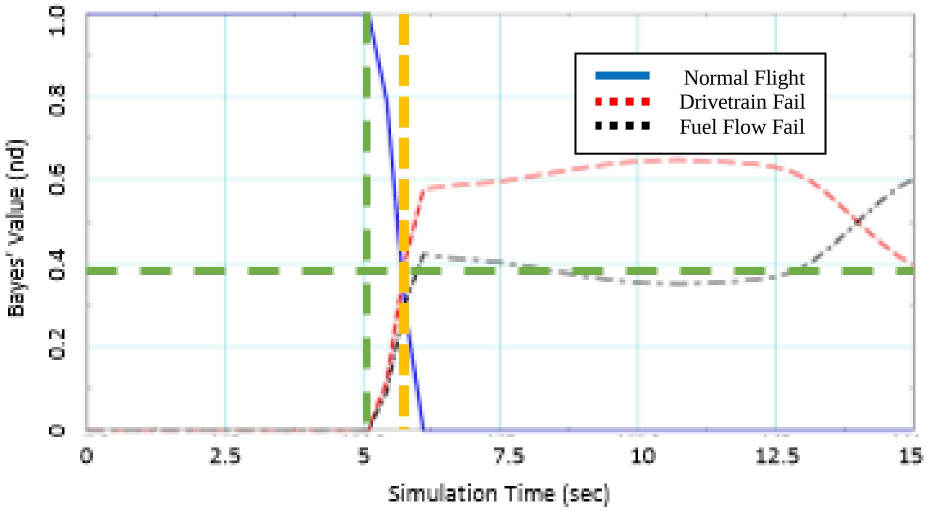


Figure 47: Failure Recognition for 100 kts Instantaneous

The three models quickly show a detection point using Bayes' theorem. In all three models, the detection came in less than 1 second of a driveshaft failure and the probability of the normal model being correct degraded quickly. Figure 48 shows the degradation of the model compared to the time since failure occurred. Note the probability is decreasing on the x-axis with time as the y-axis variable. In all three speeds tested, the failure was detected when the normal model probability degraded to approximately 35-40%, as shown by the shaded area. For all speeds, failures were detected in less than 0.6 seconds.

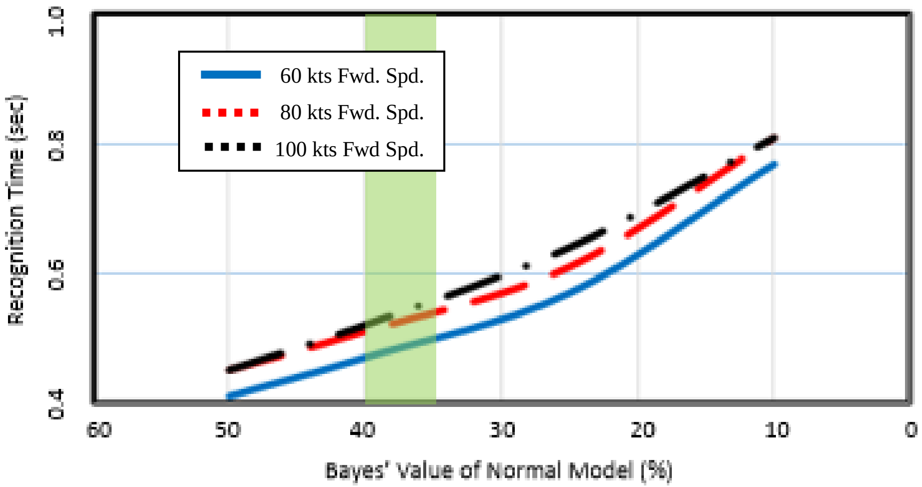


Figure 48: Degradation of Normal Probability for Instantaneous Failure

The second case examined is the engine wind-down model failure. The failure incites an engine failure where the fuel flow to the engine is immediately removed from the system. However, the engine does not separate from the main rotor system instantaneously. The remaining fuel is burned and the inertia within the engine slows the reduction in torque to the main rotor. Similarly, the models were evaluated at 60 kts, 80 kts, and 100 kts of forward speed. Figure 49 through Figure 51 show the results of the three cases.

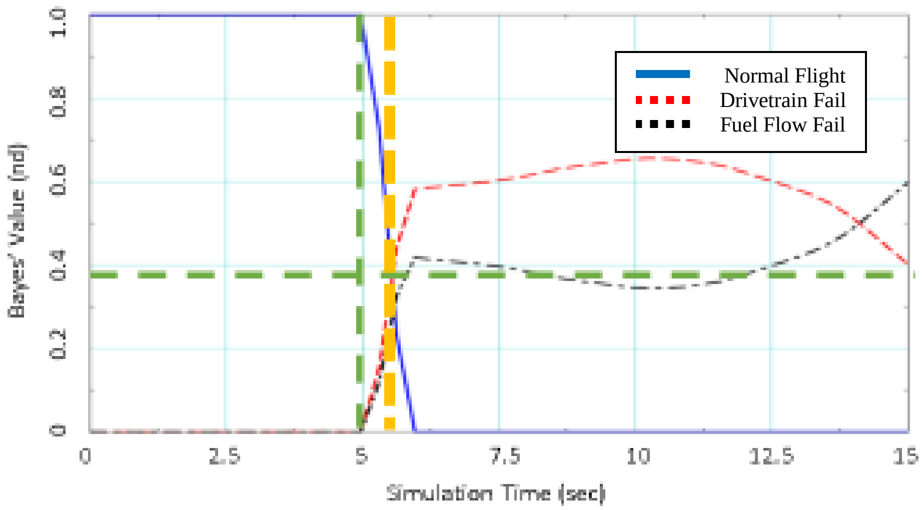


Figure 49: Failure Recognition for 60 kts Wind-Down

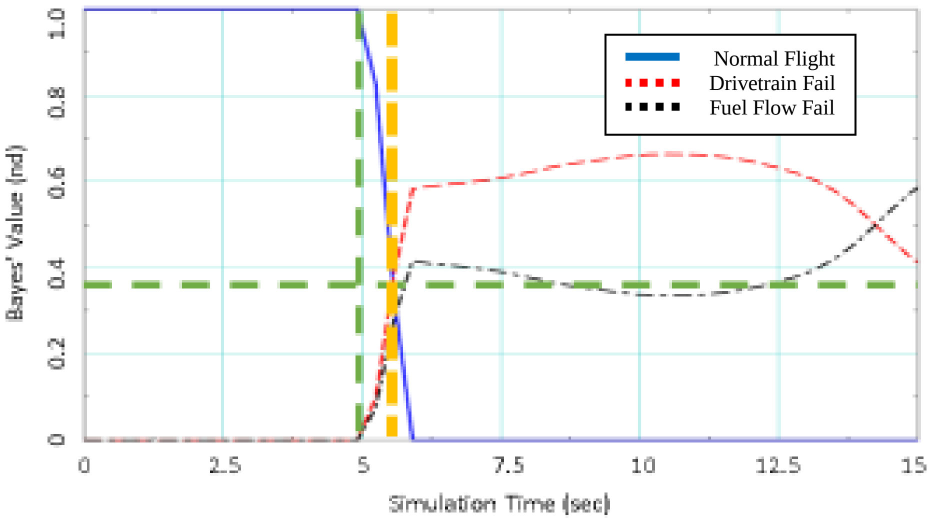


Figure 50: Failure Recognition for 80 kts Wind-Down

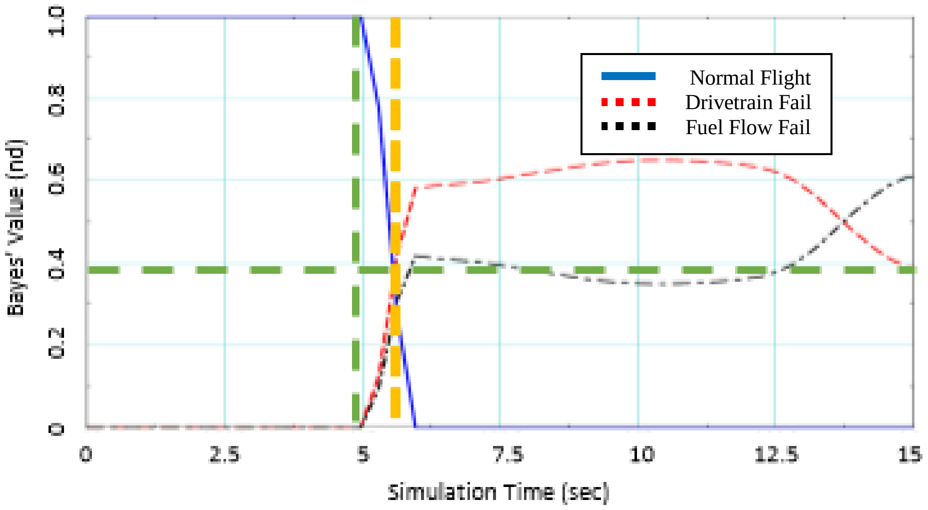


Figure 51: Failure Recognition for 100 kts Wind-Down

Similar to the instantaneous failure, the three models show a detection point using Bayes' theorem. In all three models, the detection came in less than 1 second of an engine failure and the probability of the normal model being correct degraded quickly. Figure 52 shows the degradation of the model compared to the time since failure occurred. In all three speeds tested, the failure was detected when the normal model probability degraded to approximately 35-40%, as shown by the shaded area. For all speeds, failures were detected in less than 0.8 seconds.

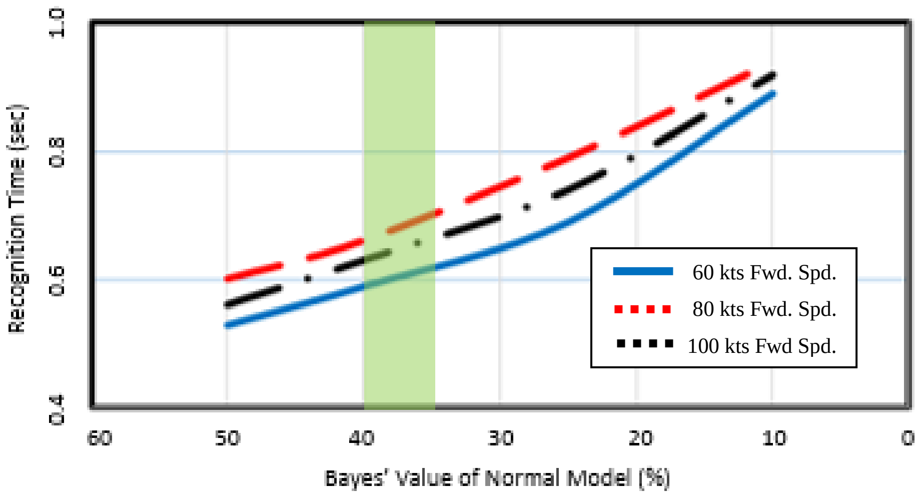


Figure 52: Degradation of Normal Probability for Wind-Down Failure

The detection of a system failure, whether it be a driveshaft or engine failure, has been found to be detected within 0.8 seconds of the event. Using the Bayes' estimation for the three hypothesis models, the normal model probability degrades to 35-40% of being the correct hypothesis. It is this moment that other failure models are more likely to be correct. The initial pilot response to an autorotation is two seconds if detected.

3.7.2. Main Rotor Recovery Response

The model response was evaluated at 60kts, 80kts, and 100kts. Remembering the purpose is to maneuver the aircraft into a steady state autorotation as quickly as possible, the pilot is responsible for maintaining control of the roll, pitch, and yaw attitude along with recovering the rotor speed. The rotor speed will be used to glide, maneuver to a landing spot, flare, and cushion to the ground. Although the pilot will lower the collective stick to zero, the position will not remain constant in order to avoid overspeeding the rotor and unnecessarily increasing the descent rate of the aircraft. Therefore, the speed that that pilot can achieve a desired and trimmed rotor speed is essential. The pilot cannot be distracted fighting to find a constant rotor speed while autorotating.

For the acquisition of a steady-state rotor speed, the threshold for error is appoximated to 3%. At 3%, there is not a need for the pilot to focus on the energy stored in the rotor. Figure 53 analyzes the rotor speeds for three different pilot delays for a driveshaft failure simulation. The delays are classified as the time of first detection as estimated by the Bayes' analysis, a standard 2-second delay for pilots in training, and a 3-second delay found as the approximate time for the rotor speed to droop to 80%.

The instant the rotor speed oscillation enters the 3% threshold and remains in a steady descent is marked for the Bayes' estimation and the 3-second delay. The time of entry is recorded and using the steady-state descent rate for each forward speed, the difference in lost altitude is calculated. For the driveshaft failure, these values are found in Table 4.

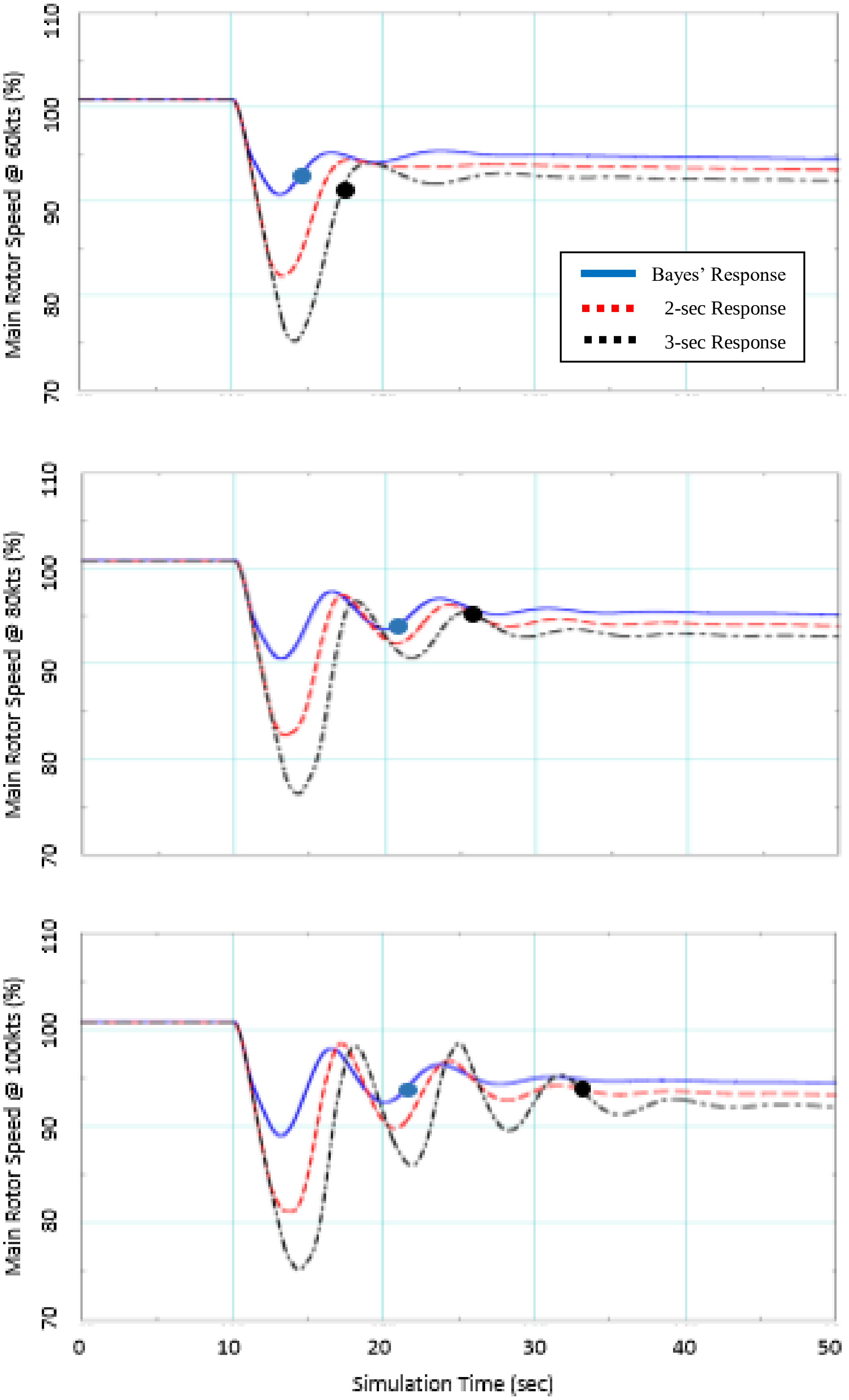


Figure 53: Rotor Speed Recovery with Instantaneous Failure

In Figure 53, the blue lines represent the rotor speed using a pilot response found from the Bayesian recognition plots in the previous section. The blue dots mark the moment the rotor speed enters and remains in a steady-state, 3% bandwidth. The red line represents the 2-second pilot delay, which is the standard for Army pilot training. The black line represents a 3-second delay which, although seemingly a short amount of time, is significant. The faster the aircraft is traveling, the more time is required to reestablish a steady-state autorotation descent.

Figure 54 examines the 10-state model with a fuel flow engine failure. The lines continue to represent the rotor speed for a Bayes' response (blue), a standard 2-second delay (red), and a 3-second delay (black). The time of entry is recorded and using the steady-state descent rate for each forward speed, the difference in lost altitude is calculated. For the engine failure, these values are found in Table 5.

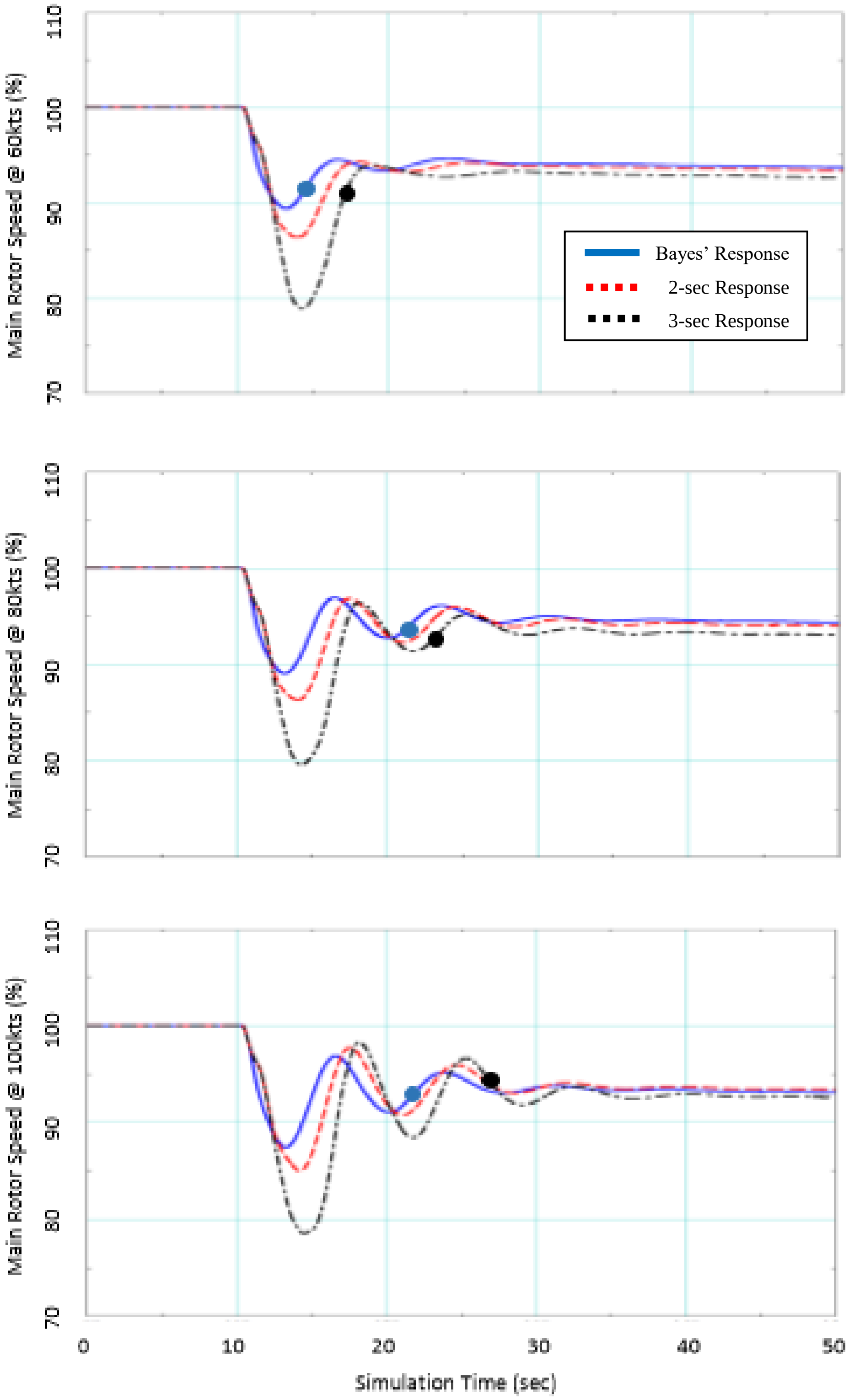


Figure 54: Rotor Speed Recovery with Wind-Down Failure

By inspection, the trend holds true that if the indicated airspeed increases from 60 knots up to 100 knots, the time of rotor speed recovery to enter and maintain a status in the 3% bandwidth increases. A more thorough analysis is exercised in the coming sections. The analysis will evaluate the recovery time, the descent rate, and translate into an altitude loss variable.

3.7.3. Altitude Loss Improvement

The VirtualPilot program is commanded to maintain 100% rotor speed following a predetermined pilot response time. The response time is based upon the Bayesian analysis from Section 3.7.1. Figure 55 is the analysis of the descent rate based upon the various response times of the VirtualPilot to reacquire the commanded rotor speed. Ultimately, no matter the pilot response, the descent rate behaves in a similar pattern for all cases. Once in a steady-state autorotation, the descent rate is precise for the initial indicated airspeed. For analyzing the effects of a Bayes' response versus a 3-second delayed pilot response, the steady state descent rate will be combined with the results from Section 3.7.2 to determine the total altitude loss difference based solely upon the pilot response time.

Figure 55 and Figure 56 demonstrate an aircraft going into a steady-state autorotation using two different engine failing procedures. The study is a rather trivial task because the steady-state autorotation descent rate should asymptote to a single value over an infinite amount of time no matter how the engine is failed. However, the task does confirm that no matter the delay, the model does maintain some type of controlled response.

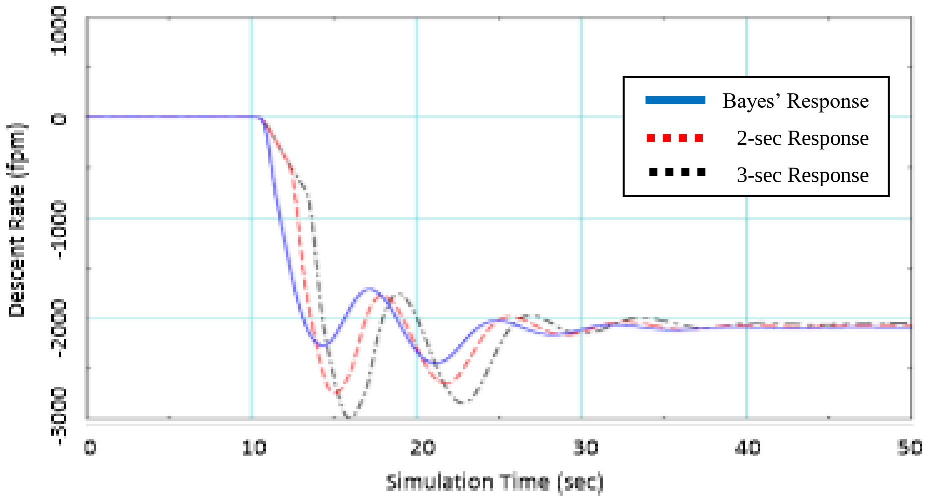


Figure 55: Steady-State Descent Rate for Instantaneous Failure

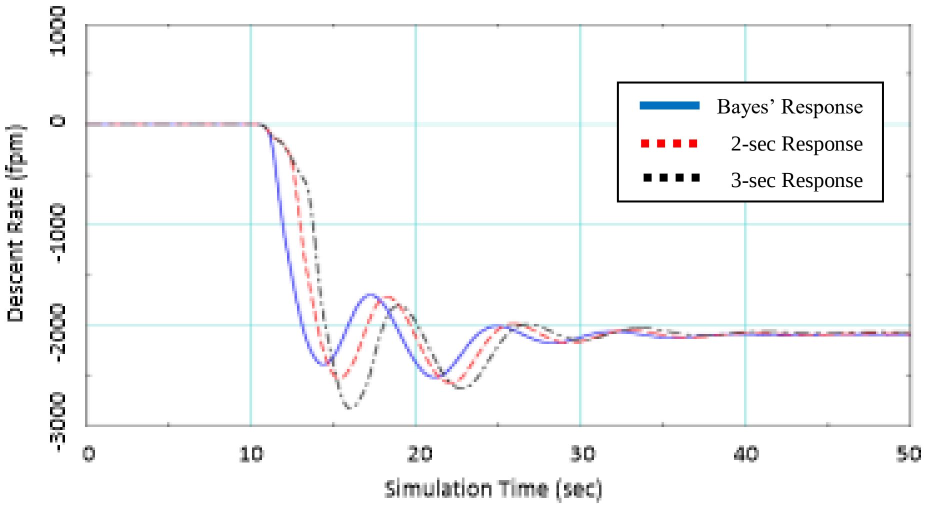


Figure 56: Steady-State Descent Rate for Wind-Down Failure

The figures confirm that over an extended period, the aircraft held at an indicated airspeed converges to a specific descent rate. The descent rate is to be used in Table 4 and Table 5 to determine altitude loss an aircraft experiences when the pilot delays 3.0 seconds versus the delay period calculated through the 10-state model Bayes' analysis. The sensitivity of Bayes' theorem with the three hypotheses lead to estimates less than a 0.6-second pilot delay for the driveshaft failure simulation and estimates less than a 0.8-second pilot delay for the engine failure when determining the main rotor speed response along with the altitude lost estimates.

Fwd. Spd. kts	Descent Rate fpm	Failure Response	MR Response Time sec	Add. Alt. Loss ft
60	-2325	3 sec. Delay	7.24	104.8
		Bayes'	4.53	
80	-2095	3 sec. Delay	15.69	162.1
		Bayes'	11.05	
100	-2175	3 sec. Delay	20.57	327.1
		Bayes'	11.54	

Table 4: Effects of Delayed Response for Instantaneous Failure

Fwd. Spd. kts	Descent Rate fpm	Failure Response	MR Response Time sec	Add. Alt. Loss ft
60	-2325	3 sec. Delay	7.72	117.6
		Bayes'	4.69	
80	-2095	3 sec. Delay	13.17	65.7
		Bayes'	11.29	
100	-2175	3 sec. Delay	16.75	185.9
		Bayes'	11.62	

Table 5: Effects of Delayed Response for Wind-Down Failure

Referencing Table 4 and Table 5 for the response, the altitude lost was determined by multiplying the rotor speed recovery time with the average descent rate. The tables compare the difference between a 3-second delay and Bayes' analysis delay rather than the entire altitude loss for each. The reasoning is the period when the engine fails through the acquisition of steady rotor speed contains significant variations in the descent rate which also varies amongst the delay lengths. Taking the difference between the two altitude losses assists in reducing the error from those periods of time.

As forward speed increases, more time is required by the pilot to establish a steady rotor speed recovery. With pilot work load being a possible hindrance for successfully completing an autorotation landing, the time saved in recovering rotor speed is essential. In normal operating speeds, using the Bayes' estimation for detecting an engine failure results in 100-350 feet of altitude preserved for the pilot to find an achievable safe landing zone. Figure 57 portrays the reduction in achievable landing space for an engine failure at 1000 ft of altitude and 80 knots forward speed.

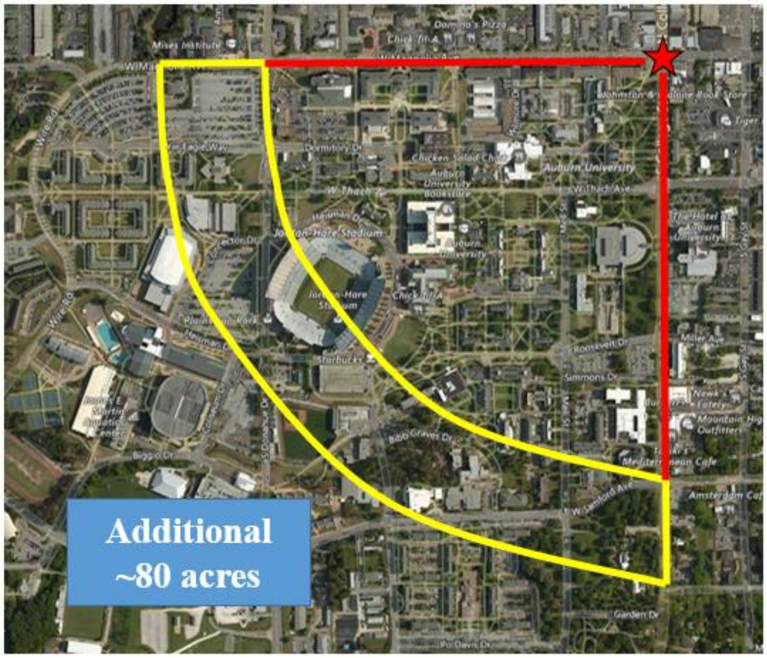


Figure 57: Landing Area Benefit for Quick Response

In graphical form, the rotor speed recovery times between a 3-second pilot delay and the less than 1-second pilot delay from the Bayesian estimation are shown in Figure 58. In the plot, the blue bars represent the time needed for the rotor speed to recover with a fast pilot reaction time. The red bars are the additional time required for the 3-second pilot delay model to recover and stabilize main rotor speed.

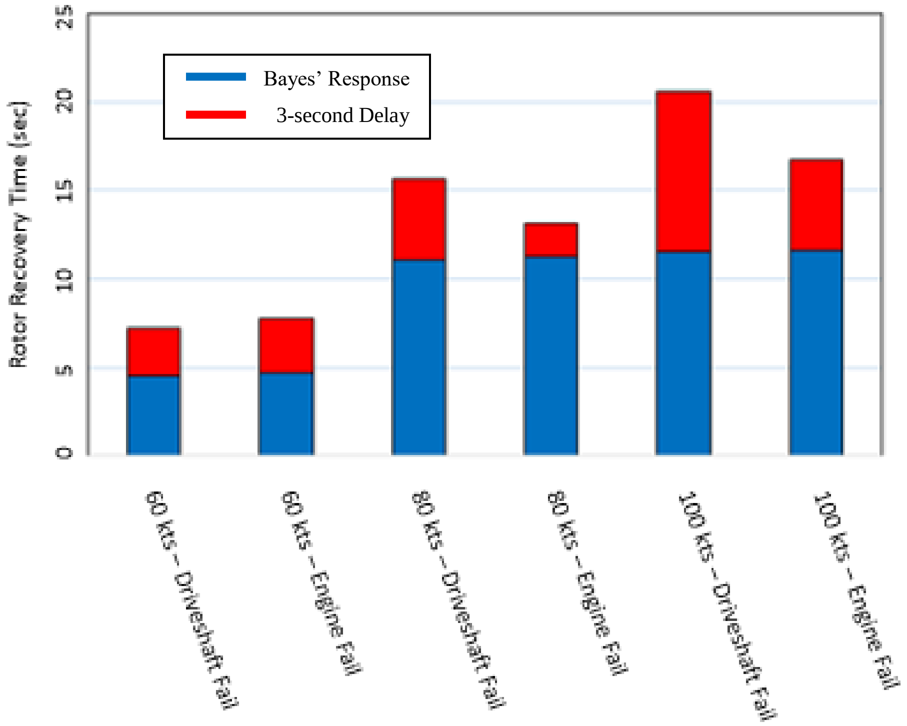


Figure 58: Rotor Speed Recovery Time Between Models

The results in Figure 58 demonstrate the toll that more than 2 seconds of pilot reaction time cost places on rotor speed recovery. The red portions of the plot represent time that a pilot can be searching for a landing area, maneuvering to a landing area, and preparing to save an aircraft with its crew, rather than deciphering the source of sirens, lights, and dynamic response and determining the correct course of action.

4. Conclusions

The creation of a model using an aircraft's current observable data to project its state vector, or vectors, to some future time and evaluate the health of the aircraft's flight condition at each time step is presented. While implementing a normal operating model, a driveshaft failure model, and an engine failure model to precisely project future states of the vehicle was difficult, probability hypotheses for each model and the failure detection criteria were developed. The models began as an 8-state dynamic rotorcraft model and a 2-state engine model. However, using the gearing ratio between the power turbine speed and the main rotor speed, the models could be coupled into a 10-state model representative of a generic utility helicopter.

Making adjustments to the engine component of the state matrix and the control variables allowed iterations of failure models to be created. By removing the state matrix components of the engine model, the torque normally generated and applied to the main rotor was no longer transferred, thereby simulating a driveshaft failure model. An engine failure was modeled by removing the fuel flow as a control.

Using the three 10-state models, the probability analysis could be calculated over a 2-second propagation of the model, using its current observable data to determine the probability of a healthy system. The Bayesian analysis determined a time when the VirtualPilot was to detect an engine failure and respond to preserve main rotor speed and successfully enter an autorotation. The response due to the Bayesian analysis was compared to a real-world pilot’s response time of 2 seconds, along with a delayed pilot reaction time of 3 seconds. The time used to recover and stabilize rotor speed was then converted to altitude lost to better compare the effectiveness of the probability analysis.

The conclusions drawn from the effort presented in this paper allow for an alternative methodology to be applied to modern rotorcraft in terms of safety analysis. The three options to protect a crew and rotorcraft in the event of a power failure are to add more devices (which impacts performance), add a means to better inform the pilot of the failure, or have the control system take the rotorcraft handling from the pilot. This work presents a baseline for combining the second and third options.

The results proved beneficial as the full rotorcraft 10-parameter dynamic and engine state vector could be iterated into two separate failure hypothesis models for analysis. From this achievement, a scheme was put in place to track the model that had the highest probability of being correct at every timestep. The models were stable enough to differentiate between a numerical bias in instrumentation and a disturbance such as wind gusts versus an actual driveshaft or engine failure. Bayes’ theorem allowed for multiple models to detect a failure as a team. From this, the system could determine when a system had failed within a single second of the failure, whether an instantaneous failure or a wind-down failure type.

Due to the success of the methods, the time to detect a failure was less than half the U.S. Army requirement. As shown in the work, a simple 3-second delay in detecting a malfunction can cost a pilot more than 20% of rotor speed, or kinetic energy, dissipated from the system. The presentation results showed the reduction time in pilot response, translated it to a reduction in rotor speed recovery time, when then reflected in the reduction of lost altitude needed to stabilize an aircraft entering autorotation.

Improvements to the process outlined in this presentation can made in the complexity of the normal model. The idea is to remove the basic assumptions used for the fuselage characteristics along with the disc model for the tail rotor. The assumptions for the fuselage were that the rotorcraft was flying maneuvers in coordinated turns and at an angle of attack of zero at all times. Allowing discrepancies in the angle of attack and sideslip will affect the external forces and moments in the equations of motion, adding more discrepancy to the model. Also, moving away from a disc rotor for the tail rotor allows for better handling and controls of the aircraft. However, this also increases the calculation time for propagating the model forward.

Finally, expansion of this work can take many directions as well. Three topics to further evaluate to improve the model for a control system is a better weighting for the covariance matrices, a higher fidelity engine model with more parameter inputs such as shaft power, pressure changes following the combustor, T45 temperature readings, torque requirements, etc., and a completion of all three legs of an autorotation to create a full autorotation simulation from failure to touchdown. The second and third expansion ideas are self-explanatory. The first expansion idea requires some detail. When calculating the probability of an engine failure using a determinant of the S_k matrix, gain sets were needed to prevent the determinant from being too close to singular. This was the case in the 8-state model where knots on the 10^2 magnitude were being grouped with angular rates and positions in radians on the 10^{-4} magnitude. Gain sets based upon airspeed schedules were used, however, it would benefit the project to have a better weighting system in place using optimal controls.

Bibliography

[1] Bottasso, C. L., Maisano, G., & Scorcelletti, F. (2009). Trajectory Optimization Procedures for Rotorcraft Vehicles Including Pilot Models, with Applications to ADS-33MTEs, Cat-A and Engin Off Landings. *American Helicopter Society 65th Annual Forum and Technology Display*. Grapevine, TX: AHS, Inc.

[2] "UH-60A Expanded Gross Weight and Center of Gravity Evaluation", USAAEFA No. 81-16, U.S. Army Aviation Engineering Flight Activity, Edwards Air Force Base, CA, August 1985.

[3] "Operator's Manual for Helicopters, Utility Tactical Transport", TM 1-1520-280-10, Department of the Army, Washington D.C., November 2007.

[4] "Operator's Manual for Army OH-58D Helicopter", TM 1-1520-248-10, Department of the Army, Washington D.C., April 2013.

[5] Graves, John D. (1983). *Methods and Devices to Improve Helicopter Autorotational Characteristics (USAAVRADCOM-TR-82-D-38)*. Culver City, CA.

[6] Keller, J. D., McKillip Jr, R. M., Horn, J. F., & Yomchinda, T. (2011). Active Flight Control and Applique Inceptor Concepts for Autorotation Performance Enhancement. *American Helicopter Society 67th Annual Forum.* Virginia Beach, VA: AHS, Inc.

[7] Bryson Jr., A. E., & Ho, Y.-C. (1975). *Applied Optimal Control: Optimization, Estimation, and Control.* Arlington, VA: Hemisphere Publishing Corporation.

[8] Thanapalan, K.K.T. *Modellingof a Helicopter System*. University College London, U.K.

[9] Aponso, B. L., Bachelder, E. N., & Lee, D. (2005). Automated Autorotation for Unmanned Rotorcraft Recovery. *The AHS International Specialists' Meeting on Unmanned Rotorcraft.* Chandler, AZ: American Helicopter Society International, Inc.

[10] Padfield, Gareth D. (2007). *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling*. Reston, VA: American Institute of Aeronautics and Astronautics, Inc.

[11] Duyar, A., Gu, Z., & Litt, J.S. (1992). A Simplified Dynamic Model of the T700 Turboshaft Engine. *48th American Helicopter Society Annual Forum*. Washington D.C.: AHS, Inc.

[12] Ballin, Mark G. (1988). *A High Fidelity Real-Time Simulation of a Small Turboshaft Engine (NASA-TM-100991)*. Moffett Field, CA: NASA.

[13] Kim, Ki C. (1999). *Analytical Cacluations of Helicopter Torque Coefficients (C_Q) and Thrust Coefficient (C_T) Values for the Helicopter Performance (HELPE) Model (ARL-TR-1986).* Aberdeen Proving Ground, MD: Army Research Laboratory.

[14] Siouris, G. M. (1996). *An Engineering Approach to Optimal Control and Estimation Theory.* New York, NY: John Wiley & Sons, Inc.

[15] Stengel, R. F. (1994). *Optimal Control and Estimation.* New York: Dover Publications, Inc., New York, NY.

[16] Stevens, B. L., & Lewis, F. L. (1992). *Aircraft Control and Simulation.* New York: John Wiley & Sons, Inc.

[17] Marschke, J. M., Crassidis, J. L., & Lam, Q. M. *Multiple Model Adaptive Estimation for Inertial Navigation During Mars Entry*. American Institute of Aeronautics and Astronautics.

[18] Ferrell, K. R. (1974). *Helicopter Performance Testing (AMCP 706-204).* Alexandria, VA: US Army Materiel Command.

[19] Tudosie, Alexandru-Nicolae. (2016). Turboshaft-Type APU for Aircraft as Controlled Object. *Scientific Research and Education in the Air Force-AFASES*, 93-100.

[20] Prouty, Raymond W. (2002). *Helicopter Performance, Stability, and Control*. Malabar, FL: Krieger Publishing Company.

Appendices

Appendix A – State Representation Form

The following Appendix outlines the development of the Kalman Filter input matrices from the equations of motion provided in the Thanapalan paper [8]. The parameter designation for the Kalman filter matrices are matched with those in the Sioris paper [14].

$$\mathbf{x_k} = \mathbf{x_{k\text{-}1}} + \Delta t\dot{\mathbf{x}}$$

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} + \Lambda\mathbf{w}$$

Continue to next page

The State Matrix component is broken into the initial values and those affected by a change in time.

$$\mathbf{x}_{k-1} = \left\{ \begin{matrix} u_{k-1} \\ v_{k-1} \\ w_{k-1} \\ p_{k-1} \\ q_{k-1} \\ r_{k-1} \\ \phi_{k-1} \\ \theta_{k-1} \\ NG_{k-1} \\ NP_{k-1} \end{matrix} \right\}$$

$$A\mathbf{x}=\begin{bmatrix} 0 & r & -q & 0 & -w & v & 0 & -g\cos\theta & 0 & \dfrac{\partial\dot{u}}{\partial NP} \\ -r & 0 & p & w & 0 & -u & g\cos\phi\cos\theta & -g\sin\phi\sin\theta & 0 & \dfrac{\partial\dot{v}}{\partial NP} \\ q & -p & 0 & -v & u & 0 & -g\sin\phi\cos\theta & -g\cos\phi\sin\theta & 0 & \dfrac{\partial\dot{w}}{\partial NP} \\ 0 & 0 & 0 & \dfrac{\partial\dot{p}}{\partial p} & \dfrac{\partial\dot{p}}{\partial q} & \dfrac{\partial\dot{p}}{\partial r} & 0 & 0 & \dfrac{\partial\dot{p}}{\partial NG} & \dfrac{\partial\dot{p}}{\partial NP} \\ 0 & 0 & 0 & \dfrac{\partial\dot{q}}{\partial p} & \dfrac{\partial\dot{q}}{\partial q} & \dfrac{\partial\dot{q}}{\partial r} & 0 & 0 & 0 & \dfrac{\partial\dot{q}}{\partial NP} \\ 0 & 0 & 0 & \dfrac{\partial\dot{r}}{\partial p} & \dfrac{\partial\dot{r}}{\partial q} & \dfrac{\partial\dot{r}}{\partial r} & 0 & 0 & \dfrac{\partial\dot{r}}{\partial NG} & \dfrac{\partial\dot{r}}{\partial NP} \\ 0 & 0 & 0 & 1 & \sin\phi\tan\theta & \cos\phi\tan\theta & (q\cos\phi-r\sin\phi)\tan\theta & (q\sin\phi+r\cos\phi)\sec^2\theta & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos\phi & -\sin\phi & r\cos\phi-q\sin\phi & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2.233 & 0 \\ \dfrac{\partial N\dot{P}}{\partial u} & 0 & 0 & 0 & 0 & 0 & 0 & \dfrac{\partial N\dot{P}}{\partial\theta} & \dfrac{\partial N\dot{P}}{\partial NG} & \dfrac{\partial N\dot{P}}{\partial NP} \end{bmatrix}\begin{Bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \phi \\ \theta \\ NG \\ NP \end{Bmatrix}$$

The Control Matrix component uses the swashplate angles and the fuel flow as inputs into the system. The fuel flow component is the primary control for the generator and power turbine speed states. The power turbine speed, when engaged with the sprag gear, is mechanically linked to the main rotor speed with a gearing ratio of 81.04:1. The torque coefficient and change in rotor speed found in the external moment about Z also are functions of various states. These relationships allow the engine model to be coupled with the dynamic (8x8) model through the main rotor speed as a non-constant.

$$B\mathbf{u} = \begin{bmatrix} \frac{\partial \dot{u}}{\partial \theta_0} & \frac{\partial \dot{u}}{\partial \theta_{1cw}} & \frac{\partial \dot{u}}{\partial \theta_{1sw}} & 0 \\ \frac{\partial \dot{v}}{\partial \theta_0} & \frac{\partial \dot{v}}{\partial \theta_{1cw}} & \frac{\partial \dot{v}}{\partial \theta_{1sw}} & 0 \\ \frac{\partial \dot{w}}{\partial \theta_0} & \frac{\partial \dot{w}}{\partial \theta_{1cw}} & \frac{\partial \dot{w}}{\partial \theta_{1sw}} & 0 \\ \frac{\partial \dot{p}}{\partial \theta_0} & \frac{\partial \dot{p}}{\partial \theta_{1cw}} & \frac{\partial \dot{p}}{\partial \theta_{1sw}} & \frac{\partial \dot{p}}{\partial w_f} \\ \frac{\partial \dot{q}}{\partial \theta_0} & \frac{\partial \dot{q}}{\partial \theta_{1cw}} & \frac{\partial \dot{q}}{\partial \theta_{1sw}} & 0 \\ \frac{\partial \dot{r}}{\partial \theta_0} & \frac{\partial \dot{r}}{\partial \theta_{1cw}} & \frac{\partial \dot{r}}{\partial \theta_{1sw}} & \frac{\partial \dot{r}}{\partial w_f} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 149400 \\ 0 & 0 & 0 & \frac{\partial N\dot{P}}{\partial w_f} \end{bmatrix} \begin{Bmatrix} \theta_0 \\ \theta_{1cw} \\ \theta_{1sw} \\ w_f \end{Bmatrix}$$

where

$$\Omega_k = NP_k\left(\frac{1}{81.04}\right)\left(\frac{2\pi}{60}\right) \therefore \frac{\partial \Omega}{\partial NP} = \left(\frac{1}{81.04}\right)\left(\frac{2\pi}{60}\right)$$

From each the state and control matrix calculated above, each has partial derivatives derived from the Taylor Series to linearize the coupled equations. Those partial derivatives are shown below. In some cases, the partial derivative is a function of more partial derivatives. Those are presented as well.

*** State Vector Partial Derivatives ***

$$\frac{\partial \dot{u}}{\partial NP} = \left(\rho\pi\Omega R^4 a_0 s\right)\left[C_X \cos\gamma_s - C_Z \sin\gamma_s\right]\frac{\partial \Omega}{\partial NP}$$

$$\frac{\partial \dot{v}}{\partial NP} = \left(\rho\pi\Omega R^4 a_0 s\right)C_Y \frac{\partial \Omega}{\partial NP}$$

$$\frac{\partial \dot{w}}{\partial NP} = \left(\rho\pi\Omega R^4 a_0 s\right)\left[C_X \sin\gamma_s + C_Z \cos\gamma_s\right]\frac{\partial \Omega}{\partial NP}$$

$$\frac{\partial \dot{p}}{\partial p} = \left(\frac{I_{xz}\left(I_{xx}^2 - I_{xx}I_{yy} + I_{xz}^2\right)}{I_{xx}\left(I_{xx}I_{zz} - I_{xz}^2\right)} + \frac{I_{xz}}{I_{xx}}\right)q$$

$$\frac{\partial \dot{p}}{\partial q} = \frac{\left(I_{yy} - I_{zz}\right)r + I_{xz}\,p}{I_{xx}} + \frac{I_{xz}^2\left(I_{yy} - I_{zz} - I_{xx}\right)r + I_{xz}\left(I_{xx}^2 - I_{xx}I_{yy} + I_{xz}^2\right)p}{I_{xx}\left(I_{xx}I_{zz} - I_{xz}^2\right)}$$

$$\frac{\partial \dot{p}}{\partial r} = \left(\frac{\left(I_{yy} - I_{zz}\right)}{I_{xx}} + \frac{I_{xz}^2 \left(I_{yy} - I_{zz} - I_{xx}\right)}{I_{xx}\left(I_{xx} I_{zz} - I_{xz}^2\right)} \right) q$$

$$\frac{\partial \dot{p}}{\partial NG} = 0.3128 I_p \frac{I_{xx} I_{xz}}{I_{xx}\left(I_{xx} I_{zz} - I_{xz}^2\right)}$$

$$\frac{\partial \dot{p}}{\partial NP} = \frac{I_{xx} I_{zz}}{I_{xx}\left(I_{xx} I_{zz} - I_{xz}^2\right)} \frac{\partial L}{\partial NP} + \frac{I_{xx} I_{xz}}{I_{xx}\left(I_{xx} I_{zz} - I_{xz}^2\right)} \frac{\partial N}{\partial NP}$$

$$\frac{\partial \dot{q}}{\partial p} = \frac{I_{zz} - I_{xx}}{I_{yy}} r - 2 \frac{I_{xz}}{I_{yy}} p$$

$$\frac{\partial \dot{q}}{\partial q} = 0$$

$$\frac{\partial \dot{q}}{\partial r} = \frac{I_{zz} - I_{xx}}{I_{yy}} p + 2 \frac{I_{xz}}{I_{yy}} r$$

$$\frac{\partial \dot{q}}{\partial NP} = \frac{1}{I_{yy}} \left[x_{cg} \rho \pi R^4 a_0 s\left(C_X \sin\gamma_s + C_Z \cos\gamma_s\right) - h_R \rho \pi R^4 a_0 s\left(C_X \cos\gamma_s - C_Z \sin\gamma_s\right) + \left(l_{TP} + x_{cg}\right) \rho R^2 C_{ZTP} \right] \Omega \frac{\partial \Omega}{\partial NP}$$

$$\frac{\partial \dot{r}}{\partial p} = \frac{\left(I_{xx}^2 - I_{xx} I_{yy} + I_{xz}^2\right)}{I_{xx} I_{zz} - I_{xz}^2} q$$

$$\frac{\partial \dot{r}}{\partial q} = \frac{I_{xz}\left(I_{yy} - I_{zz} - I_{xx}\right) r + \left(I_{xx}^2 - I_{xx} I_{yy} + I_{xz}^2\right) p}{I_{xx} I_{zz} - I_{xz}^2}$$

$$\frac{\partial \dot{r}}{\partial r} = \frac{I_{xz}\left(I_{yy} - I_{zz} - I_{xx}\right)}{I_{xx} I_{zz} - I_{xz}^2} q$$

$$\frac{\partial \dot{r}}{\partial NG} = 0.3128 I_p \frac{I_{xx}}{I_{xx} I_{zz} - I_{xz}^2}$$

$$\frac{\partial \dot{r}}{\partial NP} = \frac{I_{xz}}{I_{xx} I_{zz} - I_{xz}^2} \frac{\partial L}{\partial NP} + \frac{I_{xx}}{I_{xx} I_{zz} - I_{xz}^2} \frac{\partial N}{\partial NP}$$

$$\frac{\partial N\dot{P}}{\partial u} = \frac{I_\beta b \gamma \Omega^2}{I_R a_0 s} \left(\frac{\partial \Omega}{\partial NP}\right)^{-1} \left[-\frac{\kappa C_T}{\Omega R} \tan\left(\theta - \theta_s\right) + 1.15 C_{D_0} s \frac{u}{\Omega^2 R^2} + 1.5 \frac{f}{A} \frac{u^2}{\Omega^3 R^3} \right]$$

$$\frac{\partial N\dot{P}}{\partial \theta} = \frac{I_\beta b \gamma \Omega^2}{I_R a_0 s} \left(\frac{\partial \Omega}{\partial NP}\right)^{-1} \left[-\kappa C_T \frac{u}{\Omega R} \sec^2\left(\theta - \theta_s\right) \right]$$

$$\frac{\partial N\dot{P}}{\partial NG} = \frac{I_\beta b\gamma\Omega^2}{I_R a_0 s}\left(\frac{\partial\Omega}{\partial NP}\right)^{-1}\left[\frac{0.6256 I_p}{\rho(\Omega R)^2 \pi R^3 a_0 s}\right]$$

$$\frac{\partial N\dot{P}}{\partial NP} = -\frac{I_\beta b\gamma}{I_R a_0 s}\left[\kappa C_T\left(2\lambda\Omega - \frac{u}{R}\tan(\theta-\theta_s)\right) + \frac{C_{D_0} s\Omega}{4} - \frac{1}{2}\frac{f}{A}\frac{u^3}{\Omega^2 R^3}\right] - \frac{0.4461 I_p I_\beta \gamma b}{I_R \rho R^5 \pi a_0 s}$$

*** Control Vector Partial Derivatives ***

$$\frac{\partial\dot{u}}{\partial\theta_0} = \frac{1}{M_a}\left(\frac{1}{2}\rho(\Omega R)^2 \pi R^2 a_0 s\right)\left[\frac{\partial}{\partial\theta_0}\left(\frac{2C_X}{a_0 s}\right)\cos\gamma_s - \frac{\partial}{\partial\theta_0}\left(\frac{2C_Z}{a_0 s}\right)\sin\gamma_s\right]$$

$$\frac{\partial\dot{u}}{\partial\theta_{1cw}} = \frac{1}{M_a}\left(\frac{1}{2}\rho(\Omega R)^2 \pi R^2 a_0 s\right)\left[\frac{\partial}{\partial\theta_{1cw}}\left(\frac{2C_X}{a_0 s}\right)\cos\gamma_s - \frac{\partial}{\partial\theta_{1cw}}\left(\frac{2C_Z}{a_0 s}\right)\sin\gamma_s\right]$$

$$\frac{\partial\dot{u}}{\partial\theta_{1sw}} = \frac{1}{M_a}\left(\frac{1}{2}\rho(\Omega R)^2 \pi R^2 a_0 s\right)\left[\frac{\partial}{\partial\theta_{1sw}}\left(\frac{2C_X}{a_0 s}\right)\cos\gamma_s - \frac{\partial}{\partial\theta_{1sw}}\left(\frac{2C_Z}{a_0 s}\right)\sin\gamma_s\right]$$

$$\frac{\partial\dot{v}}{\partial\theta_0} = \frac{1}{M_a}\left(\frac{1}{2}\rho(\Omega R)^2 \pi R^2 a_0 s\right)\left(\frac{\partial}{\partial\theta_0}\left(\frac{2C_Y}{a_0 s}\right)\right)$$

$$\frac{\partial \dot{v}}{\partial \theta_{1cw}} = \frac{1}{M_a}\left(\frac{1}{2}\rho(\Omega R)^2 \pi R^2 a_0 s\right)\left(\frac{\partial}{\partial \theta_{1cw}}\left(\frac{2C_Y}{a_0 s}\right)\right)$$

$$\frac{\partial \dot{v}}{\partial \theta_{1sw}} = \frac{1}{M_a}\left(\frac{1}{2}\rho(\Omega R)^2 \pi R^2 a_0 s\right)\left(\frac{\partial}{\partial \theta_{1sw}}\left(\frac{2C_Y}{a_0 s}\right)\right)$$

$$\frac{\partial \dot{w}}{\partial \theta_0} = \frac{1}{M_a}\left(\frac{1}{2}\rho(\Omega R)^2 \pi R^2 a_0 s\right)\left[\frac{\partial}{\partial \theta_0}\left(\frac{2C_X}{a_0 s}\right)\sin\gamma_s + \frac{\partial}{\partial \theta_0}\left(\frac{2C_Z}{a_0 s}\right)\cos\gamma_s\right]$$

$$\frac{\partial \dot{w}}{\partial \theta_{1cw}} = \frac{1}{M_a}\left(\frac{1}{2}\rho(\Omega R)^2 \pi R^2 a_0 s\right)\left[\frac{\partial}{\partial \theta_{1cw}}\left(\frac{2C_X}{a_0 s}\right)\sin\gamma_s + \frac{\partial}{\partial \theta_{1cw}}\left(\frac{2C_Z}{a_0 s}\right)\cos\gamma_s\right]$$

$$\frac{\partial \dot{w}}{\partial \theta_{1sw}} = \frac{1}{M_a}\left(\frac{1}{2}\rho(\Omega R)^2 \pi R^2 a_0 s\right)\left[\frac{\partial}{\partial \theta_{1sw}}\left(\frac{2C_X}{a_0 s}\right)\sin\gamma_s + \frac{\partial}{\partial \theta_{1sw}}\left(\frac{2C_Z}{a_0 s}\right)\cos\gamma_s\right]$$

$$\frac{\partial \dot{p}}{\partial \theta_0} = \frac{1}{I_{xx}I_{zz} - I_{xz}^2}\left(\frac{1}{2}\rho(\Omega R)^2 \pi R^2 a_0 s\right)\left(I_{zz}h_R - I_{xz}x_{cg}\right)\left(\frac{\partial}{\partial \theta_0}\left(\frac{2C_Y}{a_0 s}\right)\right)$$

$$\frac{\partial \dot{p}}{\partial \theta_{1cw}} = \frac{1}{I_{xx}I_{zz} - I_{xz}^2}\left(\frac{1}{2}\rho(\Omega R)^2 \pi R^2 a_0 s\right)\left(I_{zz}h_R - I_{xz}x_{cg}\right)\left(\frac{\partial}{\partial \theta_{1cw}}\left(\frac{2C_Y}{a_0 s}\right)\right)$$

$$\frac{\partial\dot{p}}{\partial\theta_{1sw}}=\frac{1}{I_{xx}I_{zz}-I_{xz}^{2}}\left(\frac{1}{2}\rho(\Omega R)^{2}\pi R^{2}a_{0}s\right)\left(I_{zz}h_{R}-I_{xz}x_{cg}\right)\left(\frac{\partial}{\partial\theta_{1sw}}\left(\frac{2C_{Y}}{a_{0}s}\right)\right)$$

$$\frac{\partial\dot{p}}{\partial w_{f}}=16630I_{p}\frac{I_{xx}I_{xz}}{I_{xx}\left(I_{xx}I_{zz}-I_{xz}^{2}\right)}$$

$$\frac{\partial\dot{q}}{\partial\theta_{0}}=\frac{1}{I_{yy}}\left(\frac{1}{2}\rho(\Omega R)^{2}\pi R^{2}a_{0}s\right)\left[x_{cg}\left(\frac{\partial}{\partial\theta_{0}}\left(\frac{2C_{X}}{a_{0}s}\right)\sin\gamma_{s}+\frac{\partial}{\partial\theta_{0}}\left(\frac{2C_{Z}}{a_{0}s}\right)\cos\gamma_{s}\right)-h_{R}\left(\frac{\partial}{\partial\theta_{0}}\left(\frac{2C_{X}}{a_{0}s}\right)\cos\gamma_{s}-\frac{\partial}{\partial\theta_{0}}\left(\frac{2C_{Z}}{a_{0}s}\right)\sin\gamma_{s}\right)\right]$$

$$\frac{\partial\dot{q}}{\partial\theta_{1cw}}=\frac{1}{I_{yy}}\left(\frac{1}{2}\rho(\Omega R)^{2}\pi R^{2}a_{0}s\right)\left[x_{cg}\left(\frac{\partial}{\partial\theta_{1cw}}\left(\frac{2C_{X}}{a_{0}s}\right)\sin\gamma_{s}+\frac{\partial}{\partial\theta_{1cw}}\left(\frac{2C_{Z}}{a_{0}s}\right)\cos\gamma_{s}\right)-h_{R}\left(\frac{\partial}{\partial\theta_{1cw}}\left(\frac{2C_{X}}{a_{0}s}\right)\cos\gamma_{s}-\frac{\partial}{\partial\theta_{1cw}}\left(\frac{2C_{Z}}{a_{0}s}\right)\sin\gamma_{s}\right)\right]$$

$$\frac{\partial\dot{q}}{\partial\theta_{1sw}}=\frac{1}{I_{yy}}\left(\frac{1}{2}\rho(\Omega R)^{2}\pi R^{2}a_{0}s\right)\left[x_{cg}\left(\frac{\partial}{\partial\theta_{1sw}}\left(\frac{2C_{X}}{a_{0}s}\right)\sin\gamma_{s}+\frac{\partial}{\partial\theta_{1sw}}\left(\frac{2C_{Z}}{a_{0}s}\right)\cos\gamma_{s}\right)-h_{R}\left(\frac{\partial}{\partial\theta_{1sw}}\left(\frac{2C_{X}}{a_{0}s}\right)\cos\gamma_{s}-\frac{\partial}{\partial\theta_{1sw}}\left(\frac{2C_{Z}}{a_{0}s}\right)\sin\gamma_{s}\right)\right]$$

$$\frac{\partial\dot{r}}{\partial\theta_{0}}=\frac{1}{I_{xx}I_{zz}-I_{xz}^{2}}\left(\frac{1}{2}\rho(\Omega R)^{2}\pi R^{2}a_{0}s\right)\left(I_{xz}h_{R}-I_{xx}x_{cg}\right)\left(\frac{\partial}{\partial\theta_{0}}\left(\frac{2C_{Y}}{a_{0}s}\right)\right)$$

$$\frac{\partial\dot{r}}{\partial\theta_{1cw}}=\frac{1}{I_{xx}I_{zz}-I_{xz}^{2}}\left(\frac{1}{2}\rho(\Omega R)^{2}\pi R^{2}a_{0}s\right)\left(I_{xz}h_{R}-I_{xx}x_{cg}\right)\left(\frac{\partial}{\partial\theta_{1cw}}\left(\frac{2C_{Y}}{a_{0}s}\right)\right)$$

$$\frac{\partial \dot{r}}{\partial \theta_{1sw}} = \frac{1}{I_{xx} I_{zz} - I_{xz}^2} \left(\frac{1}{2} \rho (\Omega R)^2 \pi R^2 a_0 s \right) \left(I_{xz} h_R - I_{xx} x_{cg} \right) \left(\frac{\partial}{\partial \theta_{1sw}} \left(\frac{2C_Y}{a_0 s} \right) \right)$$

$$\frac{\partial \dot{r}}{\partial w_f} = 16630 I_p \frac{I_{xx}}{I_{xx} I_{zz} - I_{xz}^2}$$

$$\frac{\partial N\dot{P}}{\partial w_f} = \frac{I_\beta b \gamma \Omega^2}{I_R a_0 s} \left(\frac{\partial \Omega}{\partial NP} \right)^{-1} \left[\frac{33260 I_p}{\rho (\Omega R)^2 \pi R^3 a_0 s} \right]$$

*** External Moments Partial Derivatives ***

$$\frac{\partial L}{\partial NP} = 2 h_R \rho \pi R^4 \Omega C_Y \frac{\partial \Omega}{\partial NP}$$

$$\frac{\partial N}{\partial NP} = -0.4461 I_p - 2 x_{cg} \rho \pi R^4 \Omega C_Y \frac{\partial \Omega}{\partial NP}$$

*** External Force Coefficient Partial Derivatives ***

$$\frac{\partial}{\partial\theta_0}\left(\frac{2C_X}{a_0 s}\right)=\left(\frac{1}{6}-\frac{3\mu^2}{8}\right)\beta_{1cw}+\frac{\alpha_{sw}}{6}+\frac{\mu}{2}\left(\mu_z-\lambda_0\right)$$

$$\frac{\partial}{\partial\theta_{1cw}}\left(\frac{2C_X}{a_0 s}\right)=\left(\frac{1}{6}-\frac{\mu^2}{8}\right)\beta_0+\frac{\mu}{16}\left(\bar{q}_w-\lambda_{1cw}\right)$$

$$\frac{\partial}{\partial\theta_{1sw}}\left(\frac{2C_X}{a_0 s}\right)=\frac{3\mu}{16}\beta_{1cw}+\frac{\mu_z-\lambda_0}{4}+\frac{3\mu}{32}\left(\bar{p}_w-\lambda_{1sw}\right)$$

$$\frac{\partial}{\partial\theta_0}\left(\frac{2C_Y}{a_0 s}\right)=\left(\frac{1}{6}-\frac{3\mu^2}{8}\right)\beta_{1sw}-\frac{3\mu}{4}\beta_0+\frac{\alpha_{cw}}{6}$$

$$\frac{\partial}{\partial\theta_{1cw}}\left(\frac{2C_Y}{a_0 s}\right)=-\frac{\mu}{4}\beta_{1cw}+\frac{\mu_z-\lambda_0}{4}+\frac{\mu}{16}\left(\bar{p}_w-\lambda_{1sw}\right)$$

$$\frac{\partial}{\partial\theta_{1sw}}\left(\frac{2C_Y}{a_0 s}\right)=-\left(\frac{1}{6}+\frac{\mu^2}{8}\right)\beta_0+\frac{\mu}{16}\left(\bar{q}_w-\lambda_{1cw}\right)$$

$$\frac{\partial}{\partial\theta_0}\left(\frac{2C_Z}{a_0 s}\right)=-\frac{1}{3}-\frac{\mu^2}{2}$$

$$\frac{\partial}{\partial \theta_{1cw}}\left(\frac{2C_Z}{a_0 s}\right) = 0$$

$$\frac{\partial}{\partial \theta_{1sw}}\left(\frac{2C_Z}{a_0 s}\right) = -\frac{\mu}{2}$$

Appendix B – XAnalysis Code

********** Bayesian Estimation for Dynamic Model **********

```
//***************** F*x(k-1) information ******************//

if(counter == 1)
       wk0 = [0 0 0 0 0 0 0 0 0 0]';
       vk = [0 0 0 0 0 0]';  // Htot is 6x10, vk is 6x1
       xkm1 = [ukm1 vkm1 wkm1 pkm1 qkm1 rkm1 phikm1 thetakm1 NGkm1 NPkm1]'; // NG and NP
              are rpm - input rpm
       xk = [u v w p q r phi theta NG NP]';
       xk0 = xk;
       zk = [u w phi theta NG NP]';
       zk0 = zk;
       ucontrolkm1 = [the0km1 the1cwkm1 the1swkm1 wfkm1]';
       ucontrolk = [the0 the1cw the1sw wf]';
end
ucontrolkm1(4,1) = wfkm1;
ucontrolk(4,1) = wf;

dpdp = (Ixz*(Ixx^2-Ixx*Iyy+Ixz^2)/(Ixx*(Ixx*Izz-Ixz^2))+Ixz/Ixx)*q;
dpdq = ((Iyy-Izz)*r+Ixz*p)/Ixx + (Ixz^2*(Iyy-Izz-Ixx)*r+Ixz*(Ixx^2-
Ixx*Iyy+Ixz^2)*p)/(Ixx*(Ixx*Izz-Ixz^2));
dpdr = ((Iyy-Izz)/Ixx + Ixz^2*(Iyy-Izz-Ixx)/(Ixx*(Ixx*Izz-Ixz^2)))*q;
dqdp = (Izz-Ixx)*r/Iyy - 2*Ixz*p/Iyy;
dqdq = 0;
dqdr = (Izz-Ixx)*p/Iyy + 2*Ixz*r/Iyy;
drdp = (Ixx^2-Ixx*Iyy+Ixz^2)/(Ixx*Izz-Ixz^2)*q;
drdq = (Ixz*(Iyy-Izz-Ixx)*r + (Ixx^2-Ixx*Iyy+Ixz^2)*p)/(Ixx*Izz-Ixz^2);
drdr = Ixz*(Iyy-Izz-Ixx)/(Ixx*Izz-Ixz^2)*q;

om = (xk(10,1)/81.04)*(2*3.14/60);  // NP is rpm, om is rad/sec
domdNP = (2*3.14/60)/81.04;

F01 = the0*(1/3 + mu^2/2) + mu/2*(the1sw + pw/2) + (muz/2 - lam0/2) + thetw*(1+mu^2)/4;
F2c1 = -mu*(the1sw + beta1cw + (pw-lam1sw)/2 + mu*(the0 + thetw/2))/2;
F1c1 = (alphacw + the1cw)/3 - mu*beta0/2;
F1s1 = (alphasw + the1sw)/3 + mu*(the0 + muz -lam0 + 2*thetw/3);
F2s1 = mu*(the1cw - beta1sw + (qw-lam1cw)/2 - mu*beta0)/2;
F1c2 = -2*beta0*mu*(muz-lam0-4*mu*beta1cw/3) + (muz-lam0-4*mu*beta1cw/3)*alphacw -
mu*beta1sw*alphasw/4 + the0*(alphacw/3-mu*(beta0+mu*beta1sw/2)/2) + thetw*(alphacw/4-
mu*(beta0/3+mu*beta1sw/8)) + the1cw*((muz-lam0)/2+mu*((alphasw-beta1cw)/2-beta1cw)/4) +
mu*the1sw*((alphacw-beta1sw)/2-beta1sw-mu*beta0)/4;
F1s2 = mu^2*beta0*beta1sw/2 + (muz-lam0-mu*beta1cw/4)*alphasw - mu*beta1sw*alphacw/4 +
the0*(alphasw/3+mu*(muz-lam0)-mu^2*beta1cw/4) + thetw*(alphasw/4+mu*(muz-lam0-
beta1cw*mu/4)/2) + the1sw*((muz-lam0)/2+mu*(3*(alphasw-beta1cw)/8+beta1cw/4)) +
mu*the1cw*((alphacw-beta1sw)/2-beta1sw-mu*beta0)/4;
Cx = (F01/2 + F2c1/4)*beta1cw + F1c1*beta0/2 + F2s1*beta1sw/4 + F1s2/2;  // 2Cx/a0s
Cy = (F01/2 + F2c1/4)*beta1sw - F1s1*beta0/2 - F2s1*beta1cw/4 + F1c2/2;  // 2Cy/a0s
Cz = -F01;  // 2Cz/a0s
dLdNP = 2*hR*rho*3.14*MRR^4*om*Cy*domdNP;
dNdNP = -0.4461*Ip*-2*xcg*rho*3.14*MRR^4*om*Cy*domdNP;

duddNP = (1/Ma)*((rho*3.14*MRR^4*a0*s)*(Cx*cos(gs)-Cz*sin(gs))*om*domdNP);
dvddNP = (1/Ma)*((rho*3.14*MRR^4*a0*s)*Cy*om*domdNP);
dwddNP = (1/Ma)*((rho*3.14*MRR^4*a0*s)*(Cx*sin(gs)+Cz*cos(gs))*om*domdNP);
```

```
dpddNG = 0.3128*Ip*Ixx*Izz/(Ixx*(Ixx*Izz-Ixz^2));
dpddNP = Ixx*Izz*dLdNP/(Ixx*(Ixx*Izz-Ixz^2)) + Ixx*Ixz*dNdNP/(Ixx*(Ixx*Izz-Ixz^2));
dqddNP = (1/Iyy)*(xcg*rho*3.14*MRR^4*a0*s*(Cx*sin(gs)+Cz*cos(gs)) -
hR*rho*3.14*MRR^4*a0*s*(Cx*cos(gs)-Cz*sin(gs)) + lTPxCG*rho*MRR^2*CZTPval)*om*domdNP;
drddNG = 0.3128*Ip*Ixx/(Ixx*Izz-Ixz^2);
drddNP =Ixz*dLdNP/(Ixx*Izz-Ixz^2) + Ixx*dNdNP/(Ixx*Izz-Ixz^2);
dNPddu = Ibeta*b*LockNum*om^2/(IR*a0*s*domdNP)*(-k*CT*tan(x(8)-
3*3.14/180)/(om*MRR)+1.15*CD0*s*u/(om*MRR)^2+1.5*fA*u^2/(om*MRR)^3;
dNPddtheta = Ibeta*b*LockNum*om^2/(IR*a0*s*domdNP))*(-k*CT*u/(om*MRR*cos(x(8)-
3*3.14/180)^2);
dNPddNG = Ibeta*b*LockNum*om^2*0.6256*Ip/(IR*a0*s*domdNP*rho*(om*Mrr)^2*3.14*MRR^3*a0*s);
dNPddNP = (Ibeta*b*LockNum/(IR*a0*s))*(k*CT*(2*lam0*om-x(1)*tan(x(8)-
3*3.14/180)/MRR))+CD0*s*om/4-0.5*fA*u^3/(om^2*MRR^3))-
0.4461*Ip*Ibeta*LockNum*b/(IR*rho*MRR^5*3.14*a0*s);

A11 = 0;
A12 = r;
A13 = -q;
A14 = 0;
A15 = -w;
A16 = v;
A17 = 0;
A18 = -grav*cos(theta);
A19 = 0;
A110 = duddNP;
A21 = -r;
A22 = 0;
A23 = p;
A24 = w;
A25 = 0;
A26 = -u;
A27 = grav*cos(phi)*cos(theta);
A28 = -grav*sin(phi)*sin(theta);
A29 = 0;
A210 = dvddNP;
A31 = q;
A32 = -p;
A33 = 0;
A34 = -v;
A35 = u;
A36 = 0;
A37 = -grav*sin(phi)*cos(theta);
A38 = -grav*cos(phi)*sin(theta);
A39 = 0;
A310 = dwddNP;
A41 = 0;
A42 = 0;
A43 = 0;
A44 = dpdp;
A45 = dpdq;
A46 = dpdr;
A47 = 0;
A48 = 0;
A49 = dpddNG;
A410 = dpddNP;
A51 = 0;
A52 = 0;
A53 = 0;
A54 = dqdp;
A55 = dqdq;
A56 = dqdr;
A57 = 0;
A58 = 0;
A59 = 0;
A510 = dqddNP;
A61 = 0;
A62 = 0;
A63 = 0;
A64 = drdp;
```

```
A65 = drdq;
A66 = drdr;
A67 = 0;
A68 = 0;
A69 = drddNG;
A610 = drddNP;
A71 = 0;
A72 = 0;
A73 = 0;
A74 = 1;
A75 = sin(phi)*tan(theta);
A76 = cos(phi)*tan(theta);
A77 = (q*cos(phi)-r*sin(phi))*tan(theta);
A78 = (q*sin(phi)+r*cos(phi))/(cos(theta)*cos(theta));
A79 = 0;
A710 = 0;
A81 = 0;
A82 = 0;
A83 = 0;
A84 = 0;
A85 = cos(phi);
A86 = -sin(phi);
A87 = -(q*sin(phi)-r*cos(phi));
A88 = 0;
A89 = 0;
A810 = 0;
A91 = 0;
A92 = 0;
A93 = 0;
A94 = 0;
A95 = 0;
A96 = 0;
A97 = 0;
A98 = 0;
A99 = -2.233;
A910 = 0;
A101 = dNPddu;
A102 = 0;
A103 = 0;
A104 = 0;
A105 = 0;
A106 = 0;
A107 = 0;
A108 = dNPddtheta;
A109 = dNPddNG;
A1010 = dNPddNP;

Atot = [A11 A12 A13 A14 A15 A16 A17 A18 A19 A110;
        A21 A22 A23 A24 A25 A26 A27 A28 A29 A210;
        A31 A32 A33 A34 A35 A36 A37 A38 A39 A310;
        A41 A42 A43 A44 A45 A46 A47 A48 A49 A410;
        A51 A52 A53 A54 A55 A56 A57 A58 A59 A510;
        A61 A62 A63 A64 A65 A66 A67 A68 A69 A610;
        A71 A72 A73 A74 A75 A76 A77 A78 A79 A710;
        A81 A82 A83 A84 A85 A86 A87 A88 A89 A810;
        A91 A92 A93 A94 A95 A96 A97 A98 A99 A910;
        A101 A102 A103 A104 A105 A106 A107 A108 A109 A1010];

I1010 = [1 0 0 0 0 0 0 0 0 0;
         0 1 0 0 0 0 0 0 0 0;
         0 0 1 0 0 0 0 0 0 0;
         0 0 0 1 0 0 0 0 0 0;
         0 0 0 0 1 0 0 0 0 0;
         0 0 0 0 0 1 0 0 0 0;
         0 0 0 0 0 0 1 0 0 0;
         0 0 0 0 0 0 0 1 0 0;
         0 0 0 0 0 0 0 0 1 0;
         0 0 0 0 0 0 0 0 0 1];
```

```
//***************** G*u(k) information *****************//

dcxdt0 = beta1cw/3 + (pw-lam1sw)/6 + mu*(muz-lam0)/2;
dcxdt1cw = (1/6-mu^2/8)*beta0 + mu*(qw-lam1cw)/16;
dcxdt1sw = mu*beta1cw/4 + (muz-lam0)/4 + 3*mu*(pw-lam1sw)/16;
dcydt0 = (qw-lam1cw)/6 - 3*mu*beta0/8;
dcydt1cw = -mu*beta1cw/4 + (muz-lam0)/4 + mu*(pw-lam1sw)/16;
dcydt1sw = -(1/6+mu^2/8)*beta0 + mu*(qw-lam1cw)/16;
dczdt0 = -1/3 - mu^2/2;
dczdt1cw = 0;
dczdt1sw = -mu/2;

common = 0.5*rho*(om*MRR)^2*3.14*MRR^2*a0*s;
dudt0 = (1/Ma)*common*(dcxdt0*cos(gs)-dczdt0*sin(gs));
dudt1cw = (1/Ma)*common*(dcxdt1cw*cos(gs)-dczdt1cw*sin(gs));
dudt1sw = (1/Ma)*common*(dcxdt1sw*cos(gs)-dczdt1sw*sin(gs));
dvdt0 = (1/Ma)*common*dcydt0;
dvdt1cw = (1/Ma)*common*dcydt1cw;
dvdt1sw = (1/Ma)*common*dcydt1sw;
dwdt0 = (1/Ma)*common*(dcxdt0*sin(gs)+dczdt0*cos(gs));
dwdt1cw = (1/Ma)*common*(dcxdt1cw*sin(gs)+dczdt1cw*cos(gs));
dwdt1sw = (1/Ma)*common*(dcxdt1sw*sin(gs)+dczdt1sw*cos(gs));
dpdt0 = (common/(Ixx*Izz-Ixz^2))*(Izz*hR-Ixz*xcg)*dcydt0;
dpdt1cw = (common/(Ixx*Izz-Ixz^2))*(Izz*hR-Ixz*xcg)*dcydt1cw;
dpdt1sw = (common/(Ixx*Izz-Ixz^2))*(Izz*hR-Ixz*xcg)*dcydt1sw;
dpdwf = 16630*Ip*Ixx*Ixz/(Ixx*(Ixx*Izz-Ixz^2));
dqdt0 = (common/Iyy)*(xcg*(dcxdt0*sin(gs)+dczdt0*cos(gs))-hR*(dcxdt0*cos(gs)-
dczdt0*sin(gs)));
dqdt1cw = (common/Iyy)*(xcg*(dcxdt1cw*sin(gs)+dczdt1cw*cos(gs))-hR*(dcxdt1cw*cos(gs)-
dczdt1cw*sin(gs)));
dqdt1sw = (common/Iyy)*(xcg*(dcxdt1sw*sin(gs)+dczdt1sw*cos(gs))-hR*(dcxdt1sw*cos(gs)-
dczdt1sw*sin(gs)));
drdt0 = common/(Ixx*Izz-Ixz^2)*(Ixz*hR-Ixx*xcg)*dcydt0;
drdt1cw = common/(Ixx*Izz-Ixz^2)*(Ixz*hR-Ixx*xcg)*dcydt1cw;
drdt1sw = common/(Ixx*Izz-Ixz^2)*(Ixz*hR-Ixx*xcg)*dcydt1sw;
drdwf = 16630*Ip*Ixx/(Ixx*Izz-Ixz^2);
dNPdwf = Ibeta*b*LockNum*om^2*33260*Ip/(IR*a0*s*rho*om^2*MRR^5*3.14*a0*s*domdNP);

B11 = dudt0;
B12 = dudt1cw;
B13 = dudt1sw;
B14 = 0;
B21 = dvdt0;
B22 = dvdt1cw;
B23 = dvdt1sw;
B24 = 0;
B31 = dwdt0;
B32 = dwdt1cw;
B33 = dwdt1sw;
B34 = 0;
B41 = dpdt0;
B42 = dpdt1cw;
B43 = dpdt1sw;
B44 = dpdwf;
B51 = dqdt0;
B52 = dqdt1cw;
B53 = dqdt1sw;
B54 = 0;
B61 = drdt0;
B62 = drdt1cw;
B63 = drdt1sw;
B64 = drdwf;
B71 = 0;
B72 = 0;
B73 = 0;
B74 = 0;
B81 = 0;
B82 = 0;
B83 = 0;
```

```
B84 = 0;
B91 = 0;
B92 = 0;
B93 = 0;
B94 = 149400;
B101 = 0;
B102 = 0;
B103 = 0;
B104 = dNPdwf;

Btot = [B11 B12 B13 B14;
        B21 B22 B23 B24;
        B31 B32 B33 B34;
        B41 B42 B43 B44;
        B51 B52 B53 B54;
        B61 B62 B63 B64;
        B71 B72 B73 B74;
        B81 B82 B83 B84;
        B91 B92 B93 B94;
        B101 B102 B103 B104];

//***************** xkp1 Calculation *****************//

F = I1010 + dt*Atot;
G = dt*Btot;

wk = wk0 - dt*Atot*xkm1 - dt*Btot*ucontrolkm1;
xkp1 = F*xk + G*ucontrolk + wk;
wk = xkp1 - xk0;

Qk = [wk(1,1)^2 0 0 0 0 0 0 0 0 0;
      0 wk(2,1)^2 0 0 0 0 0 0 0 0;
      0 0 wk(3,1)^2 0 0 0 0 0 0 0;
      0 0 0 wk(4,1)^2 0 0 0 0 0 0;
      0 0 0 0 wk(5,1)^2 0 0 0 0 0;
      0 0 0 0 0 wk(6,1)^2 0 0 0 0;
      0 0 0 0 0 0 wk(7,1)^2 0 0 0;
      0 0 0 0 0 0 0 wk(8,1)^2 0 0;
      0 0 0 0 0 0 0 0 wk(9,1)^2+0.00000000000000000001 0;
      0 0 0 0 0 0 0 0 0 wk(10,1)^2+0.00000000000000000001];

if(counter==1)
        alpha = 10;
        Pk = alpha^2*Qk;  //Durrant-Whyte Reference states initial Q * alpha^2 with alpha
                            about 10 is a good first guess
                          //The 100 can drop significantly lower if need be
end
Pkp1 = Atot*Pk*Atot' + Qk;  // Sioris paper Eq 4.7-43

//***************** zkp1 Calculations *****************//

Htot = [1 0 0 0 0 0 0 0 0 0;
        0 0 1 0 0 0 0 0 0 0;
        0 0 0 0 0 0 1 0 0 0;
        0 0 0 0 0 0 0 1 0 0;
        0 0 0 0 0 0 0 0 1 0;
        0 0 0 0 0 0 0 0 0 1];

zkp1 = Htot*xkp1 + vk; // vk is previously defined
vkp1 = zkp1 - zk0;

Rk = [vkp1(1,1)^2 0 0 0 0 0;
      0 vkp1(2,1)^2 0 0 0 0;
      0 0 vkp1(3,1)^2 0 0 0;
      0 0 0 vkp1(4,1)^2 0 0;
      0 0 0 0 vkp1(5,1)^2+0.00000000000000000001 0;
      0 0 0 0 0 vkp1(6,1)^2+0.00000000000000000001];  // weights found in other scripts
                                                         from rev0
```

```
Kk = Pkp1*Htot'*inv(Htot*Pkp1*Htot' + Rk);
xkp1kp1 = xkp1 + Kk*(zkp1 - Htot*xkp1);
Pkp1kp1 = inv(inv(Pkp1) + Htot'*inv(Rk)*Htot);
Sk = Htot*Pkp1*Htot' + Rk;
rkatp1 = zkp1 - Htot*xkp1;
detSk = det(Sk);
PROBzkxk1 = (1/((2*3.14)^(1/2)*detSk^0.5))*exp((-0.5)*rkatp1'*inv(Sk)*rkatp1);

// Reset variables
xkm1 = xk;
xk = xkp1kp1;
zk = zkp1;
wk0 = wk;
vk = vkp1;
Pk = Pkp1kp1;

u = xkp1kp1(1,1);
v = xkp1kp1(2,1);
w = xkp1kp1(3,1);
p = xkp1kp1(4,1);
q = xkp1kp1(5,1);
r = xkp1kp1(6,1);
phi = xkp1kp1(7,1);
theta = xkp1kp1(8,1);
NG = xkp1kp1(9,1);
NP = xkp1kp1(10,1);

//*********** End of Probability Calc ***********//
```

********** Bayesian Estimation for Driveshaft Failure Model **********

```
//***************** F*x(k-1) information ******************//

if(counterAdiagzero == 1)
        wk0AdiagzeroNEW = [0 0]';
        wk0Adiagzero = [0 0 0 0 0 0 0 0 0 0]';
        vkAdiagzeroNEW = [0 0]';  // Htot is 6x10, vk is 6x1
        vkAdiagzero = [0 0 0 0 0 0]';
        xkm1Adiagzero = [ukm1Adiagzero vkm1Adiagzero wkm1Adiagzero pkm1Adiagzero
qkm1Adiagzero rkm1Adiagzero phikm1Adiagzero thetakm1Adiagzero NGkm1Adiagzero
NPkm1Adiagzero]'; // NG and NP are rpm - input rpm
        xkAdiagzero = [uAdiagzero vAdiagzero wAdiagzero pAdiagzero qAdiagzero rAdiagzero
phiAdiagzero thetaAdiagzero NGAdiagzero NPAdiagzero]';
        xkAdiagzero0 = xkAdiagzero;
        zkAdiagzero = [uAdiagzero wAdiagzero phiAdiagzero thetaAdiagzero NGAdiagzero
NPAdiagzero]';
        zkAdiagzero0 = zkAdiagzero;
        ucontrolkm1Adiagzero = [the0km1 the1cwkm1 the1swkm1 wfkm1]';
        ucontrolkAdiagzero = [the0 the1cw the1sw wf]';
end
ucontrolkm1Adiagzero(4,1) = wfkm1;
ucontrolkAdiagzero(4,1) = wf;

dpdp = (Ixz*(Ixx^2-Ixx*Iyy+Ixz^2)/(Ixx*(Ixx*Izz-Ixz^2))+Ixz/Ixx)*q;
dpdq = ((Iyy-Izz)*r+Ixz*p)/Ixx + (Ixz^2*(Iyy-Izz-Ixx)*r+Ixz*(Ixx^2-
Ixx*Iyy+Ixz^2)*p)/(Ixx*(Ixx*Izz-Ixz^2));
dpdr = ((Iyy-Izz)/Ixx + Ixz^2*(Iyy-Izz-Ixx)/(Ixx*(Ixx*Izz-Ixz^2)))*q;
dqdp = (Izz-Ixx)*r/Iyy - 2*Ixz*p/Iyy;
dqdq = 0;
dqdr = (Izz-Ixx)*p/Iyy + 2*Ixz*r/Iyy;
drdp = (Ixx^2-Ixx*Iyy+Ixz^2)/(Ixx*Izz-Ixz^2)*q;
drdq = (Ixz*(Iyy-Izz-Ixx)*r + (Ixx^2-Ixx*Iyy+Ixz^2)*p)/(Ixx*Izz-Ixz^2);
```

```
drdr = Ixz*(Iyy-Izz-Ixx)/(Ixx*Izz-Ixz^2)*q;

om = (xkAdiagzero(10,1)/81.04)*(2*3.14/60);  // NP is rpm, om is rad/sec
domdNP = (2*3.14/60)/81.04;

F01 = the0*(1/3 + mu^2/2) + mu/2*(the1sw + pw/2) + (muz/2 - lam0/2) + thetw*(1+mu^2)/4;
F2c1 = -mu*(the1sw + beta1cw + (pw-lam1sw)/2 + mu*(the0 + thetw/2))/2;
F1c1 = (alphacw + the1cw)/3 - mu*beta0/2;
F1s1 = (alphasw + the1sw)/3 + mu*(the0 + muz -lam0 + 2*thetw/3);
F2s1 = mu*(the1cw - beta1sw + (qw-lam1cw)/2 - mu*beta0)/2;
F1c2 = -2*beta0*mu*(muz-lam0-4*mu*beta1cw/3) + (muz-lam0-4*mu*beta1cw/3)*alphacw -
mu*beta1sw*alphasw/4 + the0*(alphacw/3-mu*(beta0+mu*beta1sw/2)/2) + thetw*(alphacw/4-
mu*(beta0/3+mu*beta1sw/8)) + the1cw*((muz-lam0)/2+mu*((alphasw-beta1cw)/2-beta1cw)/4) +
mu*the1sw*((alphacw-beta1sw)/2-beta1sw-mu*beta0)/4;
F1s2 = mu^2*beta0*beta1sw/2 + (muz-lam0-mu*beta1cw/4)*alphasw - mu*beta1sw*alphacw/4 +
the0*(alphasw/3+mu*(muz-lam0)-mu^2*beta1cw/4) + thetw*(alphasw/4+mu*(muz-lam0-
beta1cw*mu/4)/2) + the1sw*((muz-lam0)/2+mu*(3*(alphasw-beta1cw)/8+beta1cw/4)) +
mu*the1cw*((alphacw-beta1sw)/2-beta1sw-mu*beta0)/4;
Cx = (F01/2 + F2c1/4)*beta1cw + F1c1*beta0/2 + F2s1*beta1sw/4 + F1s2/2;  // 2Cx/a0s
Cy = (F01/2 + F2c1/4)*beta1sw - F1s1*beta0/2 - F2s1*beta1cw/4 + F1c2/2;  // 2Cy/a0s
Cz = -F01;  // 2Cz/a0s
dLdNP = 2*hR*rho*3.14*MRR^4*om*Cy*domdNP;
dNdNP = -0.4461*Ip*-2*xcg*rho*3.14*MRR^4*om*Cy*domdNP;

duddNP = (1/Ma)*((rho*3.14*MRR^4*a0*s)*(Cx*cos(gs)-Cz*sin(gs))*om*domdNP);
dvddNP = (1/Ma)*((rho*3.14*MRR^4*a0*s)*Cy*om*domdNP);
dwddNP = (1/Ma)*((rho*3.14*MRR^4*a0*s)*(Cx*sin(gs)+Cz*cos(gs))*om*domdNP);
dpddNG = 0.3128*Ip*Ixx*Izz/(Ixx*(Ixx*Izz-Ixz^2));
dpddNP = Ixx*Izz*dLdNP/(Ixx*(Ixx*Izz-Ixz^2)) + Ixx*Ixz*dNdNP/(Ixx*(Ixx*Izz-Ixz^2));
dqddNP = (1/Iyy)*(xcg*rho*3.14*MRR^4*a0*s*(Cx*sin(gs)+Cz*cos(gs)) -
hR*rho*3.14*MRR^4*a0*s*(Cx*cos(gs)-Cz*sin(gs)) + lTPxCG*rho*MRR^2*CZTPval)*om*domdNP;
drddNG = 0.3128*Ip*Ixx/(Ixx*Izz-Ixz^2);
drddNP =Ixz*dLdNP/(Ixx*Izz-Ixz^2) + Ixx*dNdNP/(Ixx*Izz-Ixz^2);
dNPddu = Ibeta*b*LockNum*om^2/(IR*a0*s*domdNP)*(-k*CT*tan(x(8)-
3*3.14/180)/(om*MRR)+1.15*CD0*s*u/(om*MRR)^2+1.5*fA*u^2/(om*MRR)^3;
dNPddtheta = Ibeta*b*LockNum*om^2/(IR*a0*s*domdNP))*(-k*CT*u/(om*MRR*cos(x(8)-
3*3.14/180)^2);
dNPddNG = Ibeta*b*LockNum*om^2*0.6256*Ip/(IR*a0*s*domdNP*rho*(om*Mrr)^2*3.14*MRR^3*a0*s);
dNPddNP = (Ibeta*b*LockNum/(IR*a0*s))*(k*CT*(2*lam0*om-x(1)*tan(x(8)-
3*3.14/180)/MRR))+CD0*s*om/4-0.5*fA*u^3/(om^2*MRR^3))-
0.4461*Ip*Ibeta*LockNum*b/(IR*rho*MRR^5*3.14*a0*s);

A11 = 0;
A12 = r;
A13 = -q;
A14 = 0;
A15 = -w;
A16 = v;
A17 = 0;
A18 = -grav*cos(theta);
A19 = 0;
A110 = duddNP;
A21 = -r;
A22 = 0;
A23 = p;
A24 = w;
A25 = 0;
A26 = -u;
A27 = grav*cos(phi)*cos(theta);
A28 = -grav*sin(phi)*sin(theta);
A29 = 0;
A210 = dvddNP;
A31 = q;
A32 = -p;
A33 = 0;
A34 = -v;
A35 = u;
A36 = 0;
A37 = -grav*sin(phi)*cos(theta);
```

```
A38 = -grav*cos(phi)*sin(theta);
A39 = 0;
A310 = dwddNP;
A41 = 0;
A42 = 0;
A43 = 0;
A44 = dpdp;
A45 = dpdq;
A46 = dpdr;
A47 = 0;
A48 = 0;
A49 = dpddNG;
A410 = dpddNP;
A51 = 0;
A52 = 0;
A53 = 0;
A54 = dqdp;
A55 = dqdq;
A56 = dqdr;
A57 = 0;
A58 = 0;
A59 = 0;
A510 = dqddNP;
A61 = 0;
A62 = 0;
A63 = 0;
A64 = drdp;
A65 = drdq;
A66 = drdr;
A67 = 0;
A68 = 0;
A69 = drddNG;
A610 = drddNP;
A71 = 0;
A72 = 0;
A73 = 0;
A74 = 1;
A75 = sin(phi)*tan(theta);
A76 = cos(phi)*tan(theta);
A77 = (q*cos(phi)-r*sin(phi))*tan(theta);
A78 = (q*sin(phi)+r*cos(phi))/(cos(theta)*cos(theta));
A79 = 0;
A710 = 0;
A81 = 0;
A82 = 0;
A83 = 0;
A84 = 0;
A85 = cos(phi);
A86 = -sin(phi);
A87 = -(q*sin(phi)-r*cos(phi));
A88 = 0;
A89 = 0;
A810 = 0;
A91 = 0;
A92 = 0;
A93 = 0;
A94 = 0;
A95 = 0;
A96 = 0;
A97 = 0;
A98 = 0;
A99 = 0;  // set to zero to remove load from engine
A910 = 0;
A101 = dNPddu;
A102 = 0;
A103 = 0;
A104 = 0;
A105 = 0;
A106 = 0;
```

```
A107 = 0;
A108 = dNPddtheta;
A109 = dNPddNG;
A1010 = 0;  // set to zero to remove load from engine

AtotAdiagzero = [A11 A12 A13 A14 A15 A16 A17 A18 A19 A110;
        A21 A22 A23 A24 A25 A26 A27 A28 A29 A210;
        A31 A32 A33 A34 A35 A36 A37 A38 A39 A310;
        A41 A42 A43 A44 A45 A46 A47 A48 A49 A410;
        A51 A52 A53 A54 A55 A56 A57 A58 A59 A510;
        A61 A62 A63 A64 A65 A66 A67 A68 A69 A610;
        A71 A72 A73 A74 A75 A76 A77 A78 A79 A710;
        A81 A82 A83 A84 A85 A86 A87 A88 A89 A810;
        A91 A92 A93 A94 A95 A96 A97 A98 A99 A910;
        A101 A102 A103 A104 A105 A106 A107 A108 A109 A1010];

I1010 = [1 0 0 0 0 0 0 0 0 0;
         0 1 0 0 0 0 0 0 0 0;
         0 0 1 0 0 0 0 0 0 0;
         0 0 0 1 0 0 0 0 0 0;
         0 0 0 0 1 0 0 0 0 0;
         0 0 0 0 0 1 0 0 0 0;
         0 0 0 0 0 0 1 0 0 0;
         0 0 0 0 0 0 0 1 0 0;
         0 0 0 0 0 0 0 0 1 0;
         0 0 0 0 0 0 0 0 0 1];

//***************** G*u(k) information *****************//

dcxdt0 = beta1cw/3 + (pw-lam1sw)/6 + mu*(muz-lam0)/2;
dcxdt1cw = (1/6-mu^2/8)*beta0 + mu*(qw-lam1cw)/16;
dcxdt1sw = mu*beta1cw/4 + (muz-lam0)/4 + 3*mu*(pw-lam1sw)/16;
dcydt0 = (qw-lam1cw)/6 - 3*mu*beta0/8;
dcydt1cw = -mu*beta1cw/4 + (muz-lam0)/4 + mu*(pw-lam1sw)/16;
dcydt1sw = -(1/6+mu^2/8)*beta0 + mu*(qw-lam1cw)/16;
dczdt0 = -1/3 - mu^2/2;
dczdt1cw = 0;
dczdt1sw = -mu/2;

common = 0.5*rho*(om*MRR)^2*3.14*MRR^2*a0*s;
dudt0 = (1/Ma)*common*(dcxdt0*cos(gs)-dczdt0*sin(gs));
dudt1cw = (1/Ma)*common*(dcxdt1cw*cos(gs)-dczdt1cw*sin(gs));
dudt1sw = (1/Ma)*common*(dcxdt1sw*cos(gs)-dczdt1sw*sin(gs));
dvdt0 = (1/Ma)*common*dcydt0;
dvdt1cw = (1/Ma)*common*dcydt1cw;
dvdt1sw = (1/Ma)*common*dcydt1sw;
dwdt0 = (1/Ma)*common*(dcxdt0*sin(gs)+dczdt0*cos(gs));
dwdt1cw = (1/Ma)*common*(dcxdt1cw*sin(gs)+dczdt1cw*cos(gs));
dwdt1sw = (1/Ma)*common*(dcxdt1sw*sin(gs)+dczdt1sw*cos(gs));
dpdt0 = (common/(Ixx*Izz-Ixz^2))*(Izz*hR-Ixz*xcg)*dcydt0;
dpdt1cw = (common/(Ixx*Izz-Ixz^2))*(Izz*hR-Ixz*xcg)*dcydt1cw;
dpdt1sw = (common/(Ixx*Izz-Ixz^2))*(Izz*hR-Ixz*xcg)*dcydt1sw;
dpdwf = 16630*Ip*Ixx*Ixz/(Ixx*(Ixx*Izz-Ixz^2));
dqdt0 = (common/Iyy)*(xcg*(dcxdt0*sin(gs)+dczdt0*cos(gs))-hR*(dcxdt0*cos(gs)-
dczdt0*sin(gs)));
dqdt1cw = (common/Iyy)*(xcg*(dcxdt1cw*sin(gs)+dczdt1cw*cos(gs))-hR*(dcxdt1cw*cos(gs)-
dczdt1cw*sin(gs)));
dqdt1sw = (common/Iyy)*(xcg*(dcxdt1sw*sin(gs)+dczdt1sw*cos(gs))-hR*(dcxdt1sw*cos(gs)-
dczdt1sw*sin(gs)));
drdt0 = common/(Ixx*Izz-Ixz^2)*(Ixz*hR-Ixx*xcg)*dcydt0;
drdt1cw = common/(Ixx*Izz-Ixz^2)*(Ixz*hR-Ixx*xcg)*dcydt1cw;
drdt1sw = common/(Ixx*Izz-Ixz^2)*(Ixz*hR-Ixx*xcg)*dcydt1sw;
drdwf = 16630*Ip*Ixx/(Ixx*Izz-Ixz^2);
dNPdwf = Ibeta*b*LockNum*om^2*33260*Ip/(IR*a0*s*rho*om^2*MRR^5*3.14*a0*s*domdNP);

B11 = dudt0;
B12 = dudt1cw;
B13 = dudt1sw;
B14 = 0;
```

```
B21 = dvdt0;
B22 = dvdt1cw;
B23 = dvdt1sw;
B24 = 0;
B31 = dwdt0;
B32 = dwdt1cw;
B33 = dwdt1sw;
B34 = 0;
B41 = dpdt0;
B42 = dpdt1cw;
B43 = dpdt1sw;
B44 = dpdwf;
B51 = dqdt0;
B52 = dqdt1cw;
B53 = dqdt1sw;
B54 = 0;
B61 = drdt0;
B62 = drdt1cw;
B63 = drdt1sw;
B64 = drdwf;
B71 = 0;
B72 = 0;
B73 = 0;
B74 = 0;
B81 = 0;
B82 = 0;
B83 = 0;
B84 = 0;
B91 = 0;
B92 = 0;
B93 = 0;
B94 = 149400;
B101 = 0;
B102 = 0;
B103 = 0;
B104 = dNPdwf;

BtotAdiagzero = [B11 B12 B13 B14;
       B21 B22 B23 B24;
       B31 B32 B33 B34;
       B41 B42 B43 B44;
       B51 B52 B53 B54;
       B61 B62 B63 B64;
       B71 B72 B73 B74;
       B81 B82 B83 B84;
       B91 B92 B93 B94;
       B101 B102 B103 B104];

//***************** xkp1 Calculation *****************//
FAdiagzero = I1010 + dt*AtotAdiagzero;
GAdiagzero = dt*BtotAdiagzero;

wkAdiagzero = wk0Adiagzero - dt*FAdiagzero*xkm1Adiagzero –
       dt*BtotAdiagzero*ucontrolkm1Adiagzero;
xkp1Adiagzero = FAdiagzero*xkAdiagzero + GAdiagzero*ucontrolkAdiagzero + wkAdiagzero;
wkAdiagzero = xkp1Adiagzero - xkAdiagzero0;

QkAdiagzero = [wk(1,1)^2 0 0 0 0 0 0 0 0 0;
      0 wkAdiagzero(2,1)^2 0 0 0 0 0 0 0 0;
      0 0 wkAdiagzero(3,1)^2 0 0 0 0 0 0 0;
      0 0 0 wkAdiagzero(4,1)^2 0 0 0 0 0 0;
      0 0 0 0 wkAdiagzero(5,1)^2 0 0 0 0 0;
      0 0 0 0 0 wkAdiagzero(6,1)^2 0 0 0 0;
      0 0 0 0 0 0 wkAdiagzero(7,1)^2 0 0 0;
      0 0 0 0 0 0 0 wkAdiagzero(8,1)^2 0 0;
      0 0 0 0 0 0 0 0 wkAdiagzero(9,1)^2 0;
      0 0 0 0 0 0 0 0 0 wkAdiagzero(10,1)^2];

QkAdiagzeroNEW = [wkAdiagzero(9,1)^2 0;0 wkAdiagzero(10,1)^2];
```

```
if(counterAdiagzero==1)
       alpha = 10;
       PkAdiagzero = alpha^2*QkAdiagzero;  //Durrant-Whyte Reference states initial Q *
                     alpha^2 with alpha about 10 is a good first guess
                     //The 100 can drop significantly lower if need be
       PkAdiagzeroNEW = alpha^2*QkAdiagzeroNEW;  //Durrant-Whyte Reference states initial
                     Q * alpha^2 with alpha about 10 is a good first guess
                     //The 100 can drop significantly lower if need be
end

Pkp1Adiagzero = AtotAdiagzero*PkAdiagzero*AtotAdiagzero' + QkAdiagzero;  // Sioris paper
Eq 4.7-43
AtotAdiagzeroNEW = [A99 A910;A109 A1010];
Pkp1AdiagzeroNEW = AtotAdiagzeroNEW*PkAdiagzeroNEW*AtotAdiagzeroNEW' + QkAdiagzeroNEW;
// Sioris paper Eq 4.7-43

//***************** zkp1 Calculations *****************//

HtotAdiagzero = [1 0 0 0 0 0 0 0 0 0 0;
        0 0 1 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 1 0 0 0 0;
        0 0 0 0 0 0 0 1 0 0 0;
       0 0 0 0 0 0 0 0 0 1 0;
       0 0 0 0 0 0 0 0 0 0 1];
HtotAdiagzeroNEW = [1 0;0 1];

zkp1Adiagzero = HtotAdiagzero*xkp1Adiagzero + vkAdiagzero; // vk is previously defined
vkp1Adiagzero = zkp1Adiagzero - zkAdiagzero;
NGAdiagzero NPAdiagzero]';

RkAdiagzero = [vkp1Adiagzero(1,1)^2 0 0 0 0 0;
      0 vkp1Adiagzero(2,1)^2 0 0 0 0;
      0 0 vkp1Adiagzero(3,1)^2 0 0 0;
      0 0 0 vkp1Adiagzero(4,1)^2 0 0;
      0 0 0 0 vkp1Adiagzero(5,1)^2+0.00000000000000000001 0;
      0 0 0 0 0 vkp1Adiagzero(6,1)^2+0.00000000000000000001];  // weights found in other
scripts from rev0

RkAdiagzeroNEW = [vkp1Adiagzero(5,1)^2 0;0 vkp1Adiagzero(6,1)^2];
KkAdiagzero = Pkp1Adiagzero*HtotAdiagzero'*inv(HtotAdiagzero*Pkp1Adiagzero*HtotAdiagzero'
+ RkAdiagzero);
xkp1kp1Adiagzero = xkp1Adiagzero + KkAdiagzero*(zkp1Adiagzero -
HtotAdiagzero*xkp1Adiagzero);
Pkp1kp1Adiagzero = inv(inv(Pkp1Adiagzero) +
HtotAdiagzero'*inv(RkAdiagzero)*HtotAdiagzero);

SkAdiagzero = HtotAdiagzeroNEW*Pkp1AdiagzeroNEW*HtotAdiagzeroNEW' + RkAdiagzeroNEW;
rkatp1Adiagzero = zkp1Adiagzero - HtotAdiagzero*xkp1Adiagzero;
rkatp1AdiagzeroNEW = [rkatp1Adiagzero(5,1);rkatp1Adiagzero(6,1)];
rkatp1AdiagzeroNEW = 1e-10*rkatp1AdiagzeroNEW;
detSkAdiagzero = det(SkAdiagzero);
PROBzkxk1Adiagzero = (1/((2*3.14)^(1/2)*detSkAdiagzero^0.5))*exp((-
0.5)*rkatp1AdiagzeroNEW'*inv(SkAdiagzero)*rkatp1AdiagzeroNEW);

// Reset variables
xkm1Adiagzero = xkAdiagzero;
xkAdiagzero = xkp1kp1Adiagzero;
zkAdiagzero = zkp1Adiagzero;
wk0Adiagzero = wkAdiagzero;
vkAdiagzero = vkp1Adiagzero;
PkAdiagzero = Pkp1kp1Adiagzero;

uAdiagzero = xkp1kp1Adiagzero(1,1);
vAdiagzero = xkp1kp1Adiagzero(2,1);
wAdiagzero = xkp1kp1Adiagzero(3,1);
pAdiagzero = xkp1kp1Adiagzero(4,1);
qAdiagzero = xkp1kp1Adiagzero(5,1);
rAdiagzero = xkp1kp1Adiagzero(6,1);
```

```
phiAdiagzero = xkp1kp1Adiagzero(7,1);
thetaAdiagzero = xkp1kp1Adiagzero(8,1);
NGAdiagzero = xkp1kp1Adiagzero(9,1);
NPAdiagzero = xkp1kp1Adiagzero(10,1);

//*********** End of Probability Calc ***********//
```

********** Bayesian Estimation for Engine Failure Model **********

```
//***************** F*x(k-1) information ******************//

if(counterwfzero == 1)
        wk0wfzeroNEW = [0 0]';
        wk0wfzero = [0 0 0 0 0 0 0 0 0 0]';
        vkwfzeroNEW = [0 0]';  // Htot is 6x10, vk is 6x1
        vkwfzero = [0 0 0 0 0 0]';
        xkm1wfzero = [ukm1wfzero vkm1wfzero wkm1wfzero pkm1wfzero qkm1wfzero rkm1wfzero
phikm1wfzero thetakm1wfzero NGkm1wfzero NPkm1wfzero]'; // NG and NP are rpm - input rpm
        xkwfzero = [uwfzero vwfzero wwfzero pwfzero qwfzero rwfzero phiwfzero thetawfzero
NGwfzero NPwfzero]';
        xkwfzero0 = xkwfzero;
        zkwfzero = [uwfzero wwfzero phiwfzero thetawfzero NGwfzero NPwfzero]';
        zkwfzero0 = zkwfzero;
        ucontrolkm1wfzero = [the0km1 the1cwkm1 the1swkm1 wfkm1]';
        ucontrolkwfzero = [the0 the1cw the1sw wf]';
end
ucontrolkm1wfzero(4,1) = 0;
ucontrolkwfzero(4,1) = 0;

dpdp = (Ixz*(Ixx^2-Ixx*Iyy+Ixz^2)/(Ixx*(Ixx*Izz-Ixz^2))+Ixz/Ixx)*q;
dpdq = ((Iyy-Izz)*r+Ixz*p)/Ixx + (Ixz^2*(Iyy-Izz-Ixx)*r+Ixz*(Ixx^2-
Ixx*Iyy+Ixz^2)*p)/(Ixx*(Ixx*Izz-Ixz^2));
dpdr = ((Iyy-Izz)/Ixx + Ixz^2*(Iyy-Izz-Ixx)/(Ixx*(Ixx*Izz-Ixz^2)))*q;
dqdp = (Izz-Ixx)*r/Iyy - 2*Ixz*p/Iyy;
dqdq = 0;
dqdr = (Izz-Ixx)*p/Iyy + 2*Ixz*r/Iyy;
drdp = (Ixx^2-Ixx*Iyy+Ixz^2)/(Ixx*Izz-Ixz^2)*q;
drdq = (Ixz*(Iyy-Izz-Ixx)*r + (Ixx^2-Ixx*Iyy+Ixz^2)*p)/(Ixx*Izz-Ixz^2);
drdr = Ixz*(Iyy-Izz-Ixx)/(Ixx*Izz-Ixz^2)*q;

om = (xkwfzero(10,1)/81.04)*(2*3.14/60);  // NP is rpm, om is rad/sec
domdNP = (2*3.14/60)/81.04;

F01 = the0*(1/3 + mu^2/2) + mu/2*(the1sw + pw/2) + (muz/2 - lam0/2) + thetw*(1+mu^2)/4;
F2c1 = -mu*(the1sw + beta1cw + (pw-lam1sw)/2 + mu*(the0 + thetw/2))/2;
F1c1 = (alphacw + the1cw)/3 - mu*beta0/2;
F1s1 = (alphasw + the1sw)/3 + mu*(the0 + muz -lam0 + 2*thetw/3);
F2s1 = mu*(the1cw - beta1sw + (qw-lam1cw)/2 - mu*beta0)/2;
F1c2 = -2*beta0*mu*(muz-lam0-4*mu*beta1cw/3) + (muz-lam0-4*mu*beta1cw/3)*alphacw -
mu*beta1sw*alphasw/4 + the0*(alphacw/3-mu*(beta0+mu*beta1sw/2)/2) + thetw*(alphacw/4-
mu*(beta0/3+mu*beta1sw/8)) + the1cw*((muz-lam0)/2+mu*((alphasw-beta1cw)/2-beta1cw)/4) +
mu*the1sw*((alphacw-beta1sw)/2-beta1sw-mu*beta0)/4;
F1s2 = mu^2*beta0*beta1sw/2 + (muz-lam0-mu*beta1cw/4)*alphasw - mu*beta1sw*alphacw/4 +
the0*(alphasw/3+mu*(muz-lam0)-mu^2*beta1cw/4) + thetw*(alphasw/4+mu*(muz-lam0-
beta1cw*mu/4)/2) + the1sw*((muz-lam0)/2+mu*(3*(alphasw-beta1cw)/8+beta1cw/4)) +
mu*the1cw*((alphacw-beta1sw)/2-beta1sw-mu*beta0)/4;
Cx = (F01/2 + F2c1/4)*beta1cw + F1c1*beta0/2 + F2s1*beta1sw/4 + F1s2/2;  // 2Cx/a0s
Cy = (F01/2 + F2c1/4)*beta1sw - F1s1*beta0/2 - F2s1*beta1cw/4 + F1c2/2;  // 2Cy/a0s
Cz = -F01;  // 2Cz/a0s
dLdNP = 2*hR*rho*3.14*MRR^4*om*Cy*domdNP;
dNdNP = -0.4461*Ip*-2*xcg*rho*3.14*MRR^4*om*Cy*domdNP;

duddNP = (1/Ma)*((rho*3.14*MRR^4*a0*s)*(Cx*cos(gs)-Cz*sin(gs))*om*domdNP);
dvddNP = (1/Ma)*((rho*3.14*MRR^4*a0*s)*Cy*om*domdNP);
dwddNP = (1/Ma)*((rho*3.14*MRR^4*a0*s)*(Cx*sin(gs)+Cz*cos(gs))*om*domdNP);
```

```
dpddNG = 0.3128*Ip*Ixx*Izz/(Ixx*(Ixx*Izz-Ixz^2));
dpddNP = Ixx*Izz*dLdNP/(Ixx*(Ixx*Izz-Ixz^2)) + Ixx*Ixz*dNdNP/(Ixx*(Ixx*Izz-Ixz^2));
dpddNP = ((Ixz^2+(Ixx*Izz-Ixz^2))/(Ixx*(Ixx*Izz-Ixz^2))*hR*rho*3.14*MRR^4*a0*s*Cy +
Ixz/(Ixx*Izz-Ixz^2)*rho*3.14*MRR^4*a0*s*(MRR*CQval-xcg*Cy))*om*domdNP;
dqddNP = (1/Iyy)*(xcg*rho*3.14*MRR^4*a0*s*(Cx*sin(gs)+Cz*cos(gs)) -
hR*rho*3.14*MRR^4*a0*s*(Cx*cos(gs)-Cz*sin(gs)) + lTPxCG*rho*MRR^2*CZTPval)*om*domdNP;
drddNG = 0.3128*Ip*Ixx/(Ixx*Izz-Ixz^2);
drddNP =Ixz*dLdNP/(Ixx*Izz-Ixz^2) + Ixx*dNdNP/(Ixx*Izz-Ixz^2);
dNPddu = Ibeta*b*LockNum*om^2/(IR*a0*s*domdNP)*(-k*CT*tan(x(8)-
3*3.14/180)/(om*MRR)+1.15*CD0*s*u/(om*MRR)^2+1.5*fA*u^2/(om*MRR)^3;
dNPddtheta = Ibeta*b*LockNum*om^2/(IR*a0*s*domdNP))*(-k*CT*u/(om*MRR*cos(x(8)-
3*3.14/180)^2);
dNPddNG = Ibeta*b*LockNum*om^2*0.6256*Ip/(IR*a0*s*domdNP*rho*(om*Mrr)^2*3.14*MRR^3*a0*s);
dNPddNP = (Ibeta*b*LockNum/(IR*a0*s))*(k*CT*(2*lam0*om-x(1)*tan(x(8)-
3*3.14/180)/MRR))+CD0*s*om/4-0.5*fA*u^3/(om^2*MRR^3))-
0.4461*Ip*Ibeta*LockNum*b/(IR*rho*MRR^5*3.14*a0*s);

A11 = 0;
A12 = r;
A13 = -q;
A14 = 0;
A15 = -w;
A16 = v;
A17 = 0;
A18 = -grav*cos(theta);
A19 = 0;
A110 = duddNP;
A21 = -r;
A22 = 0;
A23 = p;
A24 = w;
A25 = 0;
A26 = -u;
A27 = grav*cos(phi)*cos(theta);
A28 = -grav*sin(phi)*sin(theta);
A29 = 0;
A210 = dvddNP;
A31 = q;
A32 = -p;
A33 = 0;
A34 = -v;
A35 = u;
A36 = 0;
A37 = -grav*sin(phi)*cos(theta);
A38 = -grav*cos(phi)*sin(theta);
A39 = 0;
A310 = dwddNP;
A41 = 0;
A42 = 0;
A43 = 0;
A44 = dpdp;
A45 = dpdq;
A46 = dpdr;
A47 = 0;
A48 = 0;
A49 = dpddNG;
A410 = dpddNP;
A51 = 0;
A52 = 0;
A53 = 0;
A54 = dqdp;
A55 = dqdq;
A56 = dqdr;
A57 = 0;
A58 = 0;
A59 = 0;
A510 = dqddNP;
A61 = 0;
A62 = 0;
```

```
A63 = 0;
A64 = drdp;
A65 = drdq;
A66 = drdr;
A67 = 0;
A68 = 0;
A69 = drddNG;
A610 = drddNP;
A71 = 0;
A72 = 0;
A73 = 0;
A74 = 1;
A75 = sin(phi)*tan(theta);
A76 = cos(phi)*tan(theta);
A77 = (q*cos(phi)-r*sin(phi))*tan(theta);
A78 = (q*sin(phi)+r*cos(phi))/(cos(theta)*cos(theta));
A79 = 0;
A710 = 0;
A81 = 0;
A82 = 0;
A83 = 0;
A84 = 0;
A85 = cos(phi);
A86 = -sin(phi);
A87 = -(q*sin(phi)-r*cos(phi));
A88 = 0;
A89 = 0;
A810 = 0;
A91 = 0;
A92 = 0;
A93 = 0;
A94 = 0;
A95 = 0;
A96 = 0;
A97 = 0;
A98 = 0;
A99 = -2.233;
A910 = 0;
A101 = dNPddu;
A102 = 0;
A103 = 0;
A104 = 0;
A105 = 0;
A106 = 0;
A107 = 0;
A108 = dNPddtheta;
A109 = dNPddNG;
A1010 = dNPddNP;

Atotwfzero = [A11 A12 A13 A14 A15 A16 A17 A18 A19 A110;
        A21 A22 A23 A24 A25 A26 A27 A28 A29 A210;
        A31 A32 A33 A34 A35 A36 A37 A38 A39 A310;
        A41 A42 A43 A44 A45 A46 A47 A48 A49 A410;
        A51 A52 A53 A54 A55 A56 A57 A58 A59 A510;
        A61 A62 A63 A64 A65 A66 A67 A68 A69 A610;
        A71 A72 A73 A74 A75 A76 A77 A78 A79 A710;
        A81 A82 A83 A84 A85 A86 A87 A88 A89 A810;
        A91 A92 A93 A94 A95 A96 A97 A98 A99 A910;
        A101 A102 A103 A104 A105 A106 A107 A108 A109 A1010];

I1010 = [1 0 0 0 0 0 0 0 0 0;
         0 1 0 0 0 0 0 0 0 0;
         0 0 1 0 0 0 0 0 0 0;
         0 0 0 1 0 0 0 0 0 0;
         0 0 0 0 1 0 0 0 0 0;
         0 0 0 0 0 1 0 0 0 0;
         0 0 0 0 0 0 1 0 0 0;
         0 0 0 0 0 0 0 1 0 0;
         0 0 0 0 0 0 0 0 1 0;
         0 0 0 0 0 0 0 0 0 1];

//***************** G*u(k) information *****************//

dcxdt0 = beta1cw/3 + (pw-lam1sw)/6 + mu*(muz-lam0)/2;
dcxdt1cw = (1/6-mu^2/8)*beta0 + mu*(qw-lam1cw)/16;
dcxdt1sw = mu*beta1cw/4 + (muz-lam0)/4 + 3*mu*(pw-lam1sw)/16;
dcydt0 = (qw-lam1cw)/6 - 3*mu*beta0/8;
dcydt1cw = -mu*beta1cw/4 + (muz-lam0)/4 + mu*(pw-lam1sw)/16;
dcydt1sw = -(1/6+mu^2/8)*beta0 + mu*(qw-lam1cw)/16;
dczdt0 = -1/3 - mu^2/2;
dczdt1cw = 0;
dczdt1sw = -mu/2;

common = 0.5*rho*(om*MRR)^2*3.14*MRR^2*a0*s;
dudt0 = (1/Ma)*common*(dcxdt0*cos(gs)-dczdt0*sin(gs));
dudt1cw = (1/Ma)*common*(dcxdt1cw*cos(gs)-dczdt1cw*sin(gs));
dudt1sw = (1/Ma)*common*(dcxdt1sw*cos(gs)-dczdt1sw*sin(gs));
dvdt0 = (1/Ma)*common*dcydt0;
dvdt1cw = (1/Ma)*common*dcydt1cw;
dvdt1sw = (1/Ma)*common*dcydt1sw;
dwdt0 = (1/Ma)*common*(dcxdt0*sin(gs)+dczdt0*cos(gs));
dwdt1cw = (1/Ma)*common*(dcxdt1cw*sin(gs)+dczdt1cw*cos(gs));
dwdt1sw = (1/Ma)*common*(dcxdt1sw*sin(gs)+dczdt1sw*cos(gs));
dpdt0 = (common/(Ixx*Izz-Ixz^2))*(Izz*hR-Ixz*xcg)*dcydt0;
dpdt1cw = (common/(Ixx*Izz-Ixz^2))*(Izz*hR-Ixz*xcg)*dcydt1cw;
dpdt1sw = (common/(Ixx*Izz-Ixz^2))*(Izz*hR-Ixz*xcg)*dcydt1sw;
dpdwf = 16630*Ip*Ixx*Ixz/(Ixx*(Ixx*Izz-Ixz^2));
dqdt0 = (common/Iyy)*(xcg*(dcxdt0*sin(gs)+dczdt0*cos(gs))-hR*(dcxdt0*cos(gs)-
dczdt0*sin(gs)));
dqdt1cw = (common/Iyy)*(xcg*(dcxdt1cw*sin(gs)+dczdt1cw*cos(gs))-hR*(dcxdt1cw*cos(gs)-
dczdt1cw*sin(gs)));
dqdt1sw = (common/Iyy)*(xcg*(dcxdt1sw*sin(gs)+dczdt1sw*cos(gs))-hR*(dcxdt1sw*cos(gs)-
dczdt1sw*sin(gs)));
drdt0 = common/(Ixx*Izz-Ixz^2)*(Ixz*hR-Ixx*xcg)*dcydt0;
drdt1cw = common/(Ixx*Izz-Ixz^2)*(Ixz*hR-Ixx*xcg)*dcydt1cw;
drdt1sw = common/(Ixx*Izz-Ixz^2)*(Ixz*hR-Ixx*xcg)*dcydt1sw;
drdwf = 16630*Ip*Ixx/(Ixx*Izz-Ixz^2);
dNPdwf = Ibeta*b*LockNum*om^2*33260*Ip/(IR*a0*s*rho*om^2*MRR^5*3.14*a0*s*domdNP);

B11 = dudt0;
B12 = dudt1cw;
B13 = dudt1sw;
B14 = 0;
B21 = dvdt0;
B22 = dvdt1cw;
B23 = dvdt1sw;
B24 = 0;
B31 = dwdt0;
B32 = dwdt1cw;
B33 = dwdt1sw;
B34 = 0;
B41 = dpdt0;
B42 = dpdt1cw;
B43 = dpdt1sw;
B44 = dpdwf;
B51 = dqdt0;
B52 = dqdt1cw;
B53 = dqdt1sw;
B54 = 0;
B61 = drdt0;
B62 = drdt1cw;
B63 = drdt1sw;
B64 = drdwf;
B71 = 0;
B72 = 0;
B73 = 0;
B74 = 0;
B81 = 0;
```

```
B82 = 0;
B83 = 0;
B84 = 0;
B91 = 0;
B92 = 0;
B93 = 0;
B94 = 149400;
B101 = 0;
B102 = 0;
B103 = 0;
B104 = dNPdwf;

Btotwfzero = [B11 B12 B13 B14;
        B21 B22 B23 B24;
        B31 B32 B33 B34;
        B41 B42 B43 B44;
        B51 B52 B53 B54;
        B61 B62 B63 B64;
        B71 B72 B73 B74;
        B81 B82 B83 B84;
        B91 B92 B93 B94;
        B101 B102 B103 B104];

//***************** xkp1 Calculation *****************//
Fwfzero = I1010 + dt*Atotwfzero;
Gwfzero = dt*Btotwfzero;

wkwfzero = wk0wfzero - dt*Fwfzero*xkm1wfzero - dt*Btotwfzero*ucontrolkm1wfzero;
xkp1wfzero = Fwfzero*xkwfzero + Gwfzero*ucontrolkwfzero + wkwfzero;
wkwfzero = xkp1wfzero - xkwfzero0;

Qkwfzero = [wk(1,1)^2 0 0 0 0 0 0 0 0 0;
      0 wkwfzero(2,1)^2 0 0 0 0 0 0 0 0;
      0 0 wkwfzero(3,1)^2 0 0 0 0 0 0 0;
      0 0 0 wkwfzero(4,1)^2 0 0 0 0 0 0;
      0 0 0 0 wkwfzero(5,1)^2 0 0 0 0 0;
      0 0 0 0 0 wkwfzero(6,1)^2 0 0 0 0;
      0 0 0 0 0 0 wkwfzero(7,1)^2 0 0 0;
      0 0 0 0 0 0 0 wkwfzero(8,1)^2 0 0;
      0 0 0 0 0 0 0 0 wkwfzero(9,1)^2 0;
      0 0 0 0 0 0 0 0 0 wkwfzero(10,1)^2];

QkwfzeroNEW = [wkwfzero(9,1)^2 0;0 wkwfzero(10,1)^2];

if(counterwfzero==1)
        alpha = 10;
        Pkwfzero = alpha^2*Qkwfzero;  //Durrant-Whyte Reference states initial Q * alpha^2
                                        with alpha about 10 is a good first guess
                                      //The 100 can drop significantly lower if need be
        PkwfzeroNEW = alpha^2*QkwfzeroNEW;  //Durrant-Whyte Reference states initial Q *
                                      alpha^2 with alpha about 10 is a good first guess
                                      //The 100 can drop significantly lower if need be
end

Pkp1wfzero = Atotwfzero*Pkwfzero*Atotwfzero' + Qkwfzero;  // Sioris paper Eq 4.7-43
AtotwfzeroNEW = [A99 A910;A109 A1010];
Pkp1wfzeroNEW = AtotwfzeroNEW*PkwfzeroNEW*AtotwfzeroNEW' + QkwfzeroNEW;  // Sioris paper
Eq 4.7-43

//***************** zkp1 Calculations *****************//

Htotwfzero = [1 0 0 0 0 0 0 0 0 0;
        0 0 1 0 0 0 0 0 0 0;
        0 0 0 0 0 0 1 0 0 0;
        0 0 0 0 0 0 0 1 0 0;
        0 0 0 0 0 0 0 0 1 0;
        0 0 0 0 0 0 0 0 0 1];
HtotwfzeroNEW = [1 0;0 1];
```

```
zkp1wfzero = Htotwfzero*xkp1wfzero + vkwfzero; // vk is previously defined
vkp1wfzero = zkp1wfzero - zkwfzero;

Rkwfzero = [vkp1wfzero(1,1)^2 0 0 0 0 0;
      0 vkp1wfzero(2,1)^2 0 0 0 0;
      0 0 vkp1wfzero(3,1)^2 0 0 0;
      0 0 0 vkp1wfzero(4,1)^2 0 0;
      0 0 0 0 vkp1wfzero(5,1)^2+0.00000000000000000001 0;
      0 0 0 0 0 vkp1wfzero(6,1)^2+0.00000000000000000001];  // weights found in other
scripts from rev0

RkwfzeroNEW = [vkp1wfzero(5,1)^2 0;0 vkp1wfzero(6,1)^2];
Kkwfzero = Pkp1wfzero*Htotwfzero'*inv(Htotwfzero*Pkp1wfzero*Htotwfzero' + Rkwfzero);
xkp1kp1wfzero = xkp1wfzero + Kkwfzero*(zkp1wfzero - Htotwfzero*xkp1wfzero);
Pkp1kp1wfzero = inv(inv(Pkp1wfzero) + Htotwfzero'*inv(Rkwfzero)*Htotwfzero);

Skwfzero = HtotwfzeroNEW*Pkp1wfzeroNEW*HtotwfzeroNEW' + RkwfzeroNEW;
rkatp1wfzero = zkp1wfzero - Htotwfzero*xkp1wfzero;
rkatp1wfzeroNEW = [rkatp1wfzero(5,1);rkatp1wfzero(6,1)];
rkatp1wfzeroNEW = 1e-8*rkatp1wfzeroNEW;
detSkwfzero = det(Skwfzero);
PROBzkxk1wfzero = (1/((2*3.14)^(1/2)*detSkwfzero^0.5))*exp((0.5)*rkatp1wfzeroNEW'*
       inv(Skwfzero)*rkatp1wfzeroNEW);

// Reset variables
xkm1wfzero = xkwfzero;
xkwfzero = xkp1kp1wfzero;
zkwfzero = zkp1wfzero;
wk0wfzero = wkwfzero;
vkwfzero = vkp1wfzero;
Pkwfzero = Pkp1kp1wfzero;

uwfzero = xkp1kp1wfzero(1,1);
vwfzero = xkp1kp1wfzero(2,1);
wwfzero = xkp1kp1wfzero(3,1);
pwfzero = xkp1kp1wfzero(4,1);
qwfzero = xkp1kp1wfzero(5,1);
rwfzero = xkp1kp1wfzero(6,1);
phiwfzero = xkp1kp1wfzero(7,1);
thetawfzero = xkp1kp1wfzero(8,1);
NGwfzero = xkp1kp1wfzero(9,1);
NPwfzero = xkp1kp1wfzero(10,1);

//*********** End of Probability Calc ***********//
```

Appendix C – VirtualPilot Command Table

The VirtualPilot uses commanded targets for each of the 4 axes of control; Longitudinal Cyclic, Lateral Cyclic, Collective, and Pedals. One feature of the VirtualPilot is to set the modes and targets against the simulation time scale. The first plot demonstrates the input mode and target value to put a helicopter into a steady-state level flight. At some initiation point, the lateral cyclic command adjusts the roll angle to put the aircraft in a banked turn. The other targets remain constant. The "Hold Inputs" for the first 0.3 seconds is used to zero out any errors from a simulation fresh start.

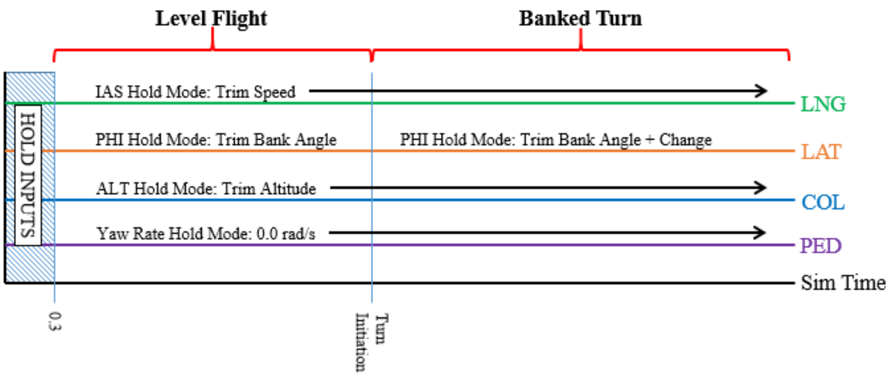
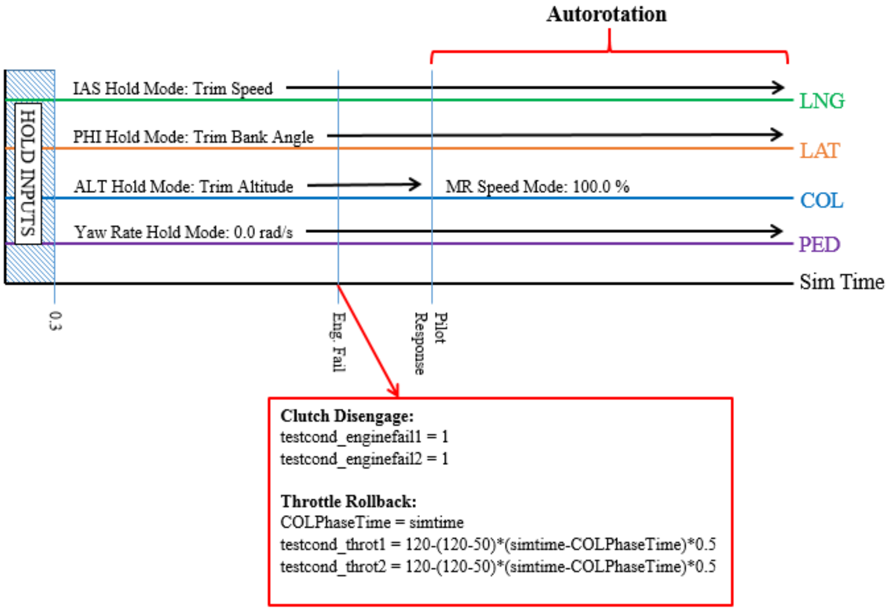

Level Flight
Banked Turn
HOLD INPUTS
IAS Hold Mode: Trim Speed
PHI Hold Mode: Trim Bank Angle
PHI Hold Mode: Trim Bank Angle + Change
ALT Hold Mode: Trim Altitude
Yaw Rate Hold Mode: 0.0 rad/s
LNG
LAT
COL
PED
Sim Time
0.3
Turn Initiation

Figure 59: VirtualPilot Level Flight and Banked Turn Scheme

The second plot evaluates the pilot maneuvers to transition from a steady state level flight to an autorotation. The first vertical line at some time is the point of the engine failure. There are two method to “fail” the engine in the simulation model. The first method is to disengage the sprag clutch. This immediately disconnects the main rotor and tail rotor from the driveshaft. The second method is to roll back the throttle from the cockpit to simulate a 4-second fuel flow decrease. A throttle position of 120 is the normal operating value. A throttle position of 50 is the point where the aircraft with the provided configuration is not capable of maintaining the fuel flow required for steady and level flight. For each time step, the value of (simtime – COLPhaseTime) increases linearly. This reduces the throttle position for each time step. Once the engine begins its failure state, a specified time later, the pilot reacts to the engine failure. In this scenario, the pilot correctly reduces the collective and drives the rotor speed back to 100%. The aircraft at that point is in a steady autorotation.


Autorotation
HOLD INPUTS
IAS Hold Mode: Trim Speed
PHI Hold Mode: Trim Bank Angle
ALT Hold Mode: Trim Altitude
MR Speed Mode: 100.0 %
Yaw Rate Hold Mode: 0.0 rad/s
LNG
LAT
COL
PED
Sim Time
0.3
Eng. Fail
Pilot Response

Clutch Disengage:
testcond_enginefail1 = 1
testcond_enginefail2 = 1

Throttle Rollback:
COLPhaseTime = simtime
testcond_throt1 = 120-(120-50)*(simtime-COLPhaseTime)*0.5
testcond_throt2 = 120-(120-50)*(simtime-COLPhaseTime)*0.5

Figure 60: VirtualPilot Autorotation Scheme