A VIRTUAL PILOT ALGORITHM FOR SYNTHETIC HUMS DATA GENERATION

A Thesis
Presented to
The Academic Faculty

By

LEE EVERETT FOWLER

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science in Mechanical Engineering

Georgia Institute of Technology

December, 2015

A VIRTUAL PILOT ALGORITHM FOR SYNTHETIC HUMS DATA GENERATION

Approved by:

Dr. Jonathan Rogers, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Aldo Ferri
School of Mechanical Engineering
Georgia Institute of Technology

Dr. William Singhose
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: November 23, 2015

DEDICATION

To my Father, for encouraging me to press hard into the best of my capabilities, to my Mother, for teaching me to how to learn everything I know, and to my Brothers, for always being there for me.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Jonathan Rogers. From the first opportunity to build, test, and fly aircraft to all the guidance and encouragement given throughout each of our research projects, he gave me a direction to go in engineering that I can be passionate about. I also thank my committee members for their time and feedback during this project.

Thank you to Andrew Brown, Jonathan Warner, and all of the members of the iREAL lab for allowing us to have a fun place to bounce ideas back and forth. To Nate Miller and Bob Long for showing me how to take initiative, and to Zach Sunberg and Brad Taylor for being awesome role models in engineering. I would not have seen how rewarding graduate school can be without you all.

Last, I would like to thank the Georgia Tech Rowing Team for being a family to me throughout a very stressful time in life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

x, y, z	Inertial position coordinates
Φ, θ, Ψ	Roll, pitch, and yaw Euler angles
u, v, w	Body-frame velocity components
p, q, r	Body-frame angular velocities
β_0	Blade coning angle measured from the rotor hub plane
β_{1c}, β_{1s}	Longitudinal and lateral first harmonic blade flapping coefficients
Ω	Rotor angular velocity
Ψ_{MR}	Azimuth angle of the rotor blade
$\lambda_0, \lambda_s, \lambda_c$	Component states of induced inflow distribution
λ_i	Main rotor induced inflow
e	Rotor blade hinge offset from the hub center
W	Helicopter gross weight
m	Helicopter mass
I_B	Blade flap-wise intertia
r	Radial distance from rotor hub
R	Main rotor radius
θ_0	Main rotor collective blade pitch
θ_{1c}	Main rotor lateral cyclic blade pitch
θ_{1s}	Main rotor longitudinal cyclic blade pitch
θ_{tr}	Tail rotor collective blade pitch

SUMMARY

Regime recognition is an important tool used in creation of usage spectra and fatigue loads analysis. While a variety of regime recognition algorithms have been developed and deployed to date, verification and validation (V&V) of such algorithms is still a labor intensive process that is largely subjective. The current V&V process for regime recognition codes involves a comparison of scripted flight test data to regime recognition algorithm outputs. This is problematic because scripted flight test data is expensive to obtain, may not accurately match the maneuver script, and is often used to train the regime recognition algorithms and thus is not appropriate for V&V purposes. In this paper, a simulation-based virtual pilot algorithm is proposed as an alternative to physical testing for generating V&V flight test data. A "virtual pilot" is an algorithm that replicates a human's piloting and guidance role in simulation by translating high level maneuver instructions into parameterized control laws. Each maneuver regime is associated with a feedback control law, and a control architecture is defined which provides for seamless transitions between maneuvers and allows for execution of an arbitrary maneuver script in simulation. The proposed algorithm does not require training data, iterative learning, or optimization, but rather utilizes a tuned model and feedback control laws defined for each maneuver. As a result, synthetic HUMS data may be generated and used in a highly automated regime recognition V&V process. In this thesis, the virtual pilot algorithm is formulated and the component feedback control laws and maneuver transition schemes are defined. Example synthetic HUMS data is generated using a simulation model of the SH-60B, and virtual pilot fidelity is demonstrated through both conformance to the ADS-33 standards for selected Mission Task Elements and comparison to actual HUMS data.

CHAPTER 1 : INTRODUCTION

Structural usage monitoring systems have become an important element in lifecycle management for military helicopters. Over the past several decades, the US Navy has developed a suite of analysis tools for performing so-called regime recognition based on post-flight data processing, summarized in References [1-8]. Regime recognition is a problem of high-dimensional classification – given a set of observations (flight data) over a specific time interval, the observations are classified as belonging to a certain flight regime. Sets of post-flight data can then be batch processed to determine how much total time was spent in each regime, and fatigue/lifecycle penalties can be computed accordingly. The regime recognition problem is inherently high-dimensional due to the relatively large number of possible regimes (100+) and the number of measured parameters (20+) provided by Health and Usage Monitoring Systems (HUMS).

Verification and validation (V&V) of regime recognition codes is a challenging task and usually relies on scripted flight test data. In these tests, a pilot flies a particular flight test card and records the times at which maneuvers begin and end. This scripted test data is then used for V&V by comparing the outputs of a regime recognition code at a given time to the maneuver reported by the pilot. Such a process is not only labor intensive but also error prone due to inconsistencies in pilot technique, inaccurate maneuver start and stop times, and other issues. Regime recognition software may identify mixed regimes (i.e., climbing turn) when the pilot intended to fly a simpler maneuver (i.e., turn at constant altitude). As a result, there is increasing interest in the use of simulated, or synthetic, flight test data for use in the V&V process. Potentially, a maneuver script could be flown in simulation with the resulting simulator outputs used to perform V&V. Such a scheme would lower costs significantly, remove inconsistencies in the V&V data, and facilitate an automated V&V process which could leverage a large amount of data. This notional automated process is depicted in Figure 1.1.

A critical component of this process is a control algorithm that can execute arbitrary maneuver scripts in simulation in a similar manner to a human pilot. Standard helicopter autopilots are not sufficient for this purpose in that they are designed primarily to operate in, or between, trim flight conditions and cannot perform a sequence of arbitrary maneuvers such as symmetric pullouts, autorotations, sideslips, etc. In this work, a so-called "virtual pilot" control algorithm is proposed which allows execution of a maneuver sequence through definition of a feedback control law associated with each maneuver. Simulation results demonstrate that synthetic HUMS data produced by the virtual pilot conforms to the standards for selected Mission Task Elements in ADS-33 and also matches flight data produced in actual piloted flight tests of the SH-60B.

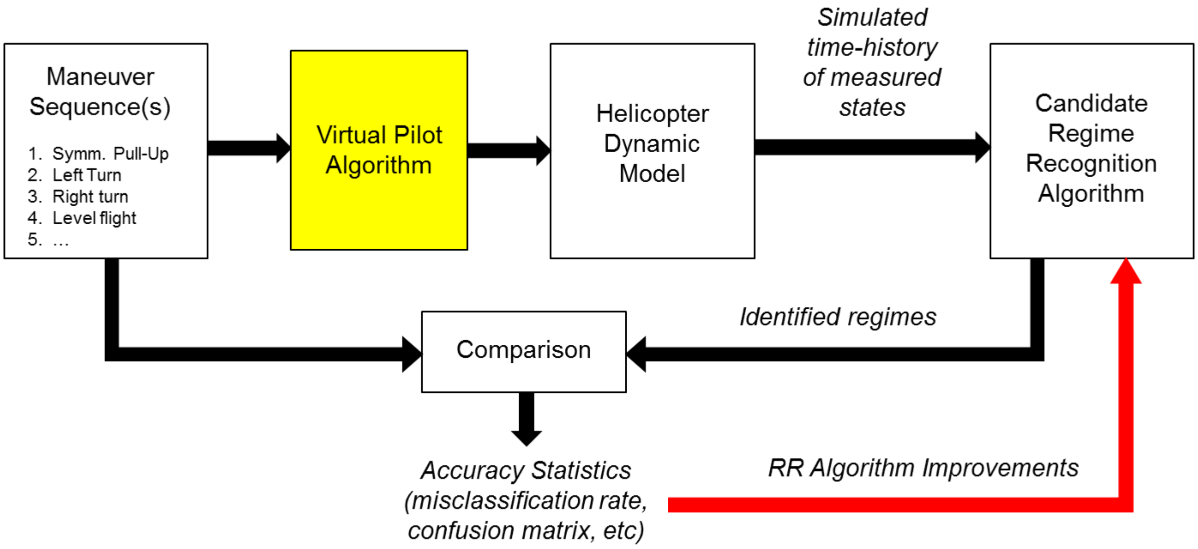
Maneuver Sequence(s)
1. Symm. Pull-Up
2. Left Turn
3. Right turn
4. Level flight
5. ...
Virtual Pilot Algorithm
Helicopter Dynamic Model
Simulated time-history of measured states
Candidate Regime Recognition Algorithm
Comparison
Identified regimes
Accuracy Statistics (misclassification rate, confusion matrix, etc)
RR Algorithm Improvements

Figure 1.1. Automated Verification and Validation Process for Maneuver Recognition

The use of preprogrammed maneuvers in autonomous flight control is not new and has been explored in the literature. Many control systems with greater autonomy are really path planning techniques that focus on positioning, such as waypoint following and obstacle avoidance. In contrast, the virtual pilot algorithm used for regime recognition V&V would be relatively unconcerned with particular geometric trajectories or final positions. Although the goal of a virtual pilot algorithm would be to simply emulate the flight characteristics of some maneuver, there has been much previous work regarding flight control for individual maneuvers that are relevant to the development of a virtual pilot. It is common to define maneuvers with timed intervals during which a controller tracks set rates about each control axis, as in [9]. Maneuvers such as these often mimic human pilot control inputs. Canned maneuvers can be included in path planning if the resultant changes in the vehicle state can be predicted for each maneuver. Such a technique was implemented in [10].

In [11], maneuvers of the same type were defined for discrete increments of varying magnitude, and interpolated so that smooth path optimization could be accomplished using maneuvers of any scale. By combining potentially complex control sequences into a predefined maneuver, the dimensionality of path planning is greatly reduced. The design of a virtual pilot for the purposes of generating recognizable flight regimes will not require predictive capability because the final trajectory or position of the aircraft is not important for regime recognition. However, defining and executing maneuvers includes a more sophisticated degree of autonomy than simple aircraft stabilization. Stabilization is an important capability for automatically generating human pilot-like trajectories, but creating smooth transitions between maneuvers is also important to create flight data that gradually fades from one identifiable regime to another.

Creating flight plans that consist of sequenced maneuvers falls under the category of mid-level decision making according to the framework for categorizing levels of autonomy set forth in [12]. It makes sense to emulate the human thought process of decision making especially when the goal is to generate identifiable flight regimes traditionally flown and labeled by human pilots. Sectioning trajectories into pre-defined maneuvers is a sensible way to do this. Further separating maneuvers into timed intervals of tracking certain set-points on the state, which could be variables like altitude, altitude rate, angle of bank, roll rate, etc.., follows the intuitive pattern of human behavior ideal for the design of a virtual pilot. In the creation of the virtual pilot algorithm, many considerations from these aforementioned autonomous rotorcraft control algorithms were implemented where relevant, such as the definition of discrete maneuvers using desired rate tracking and smooth blending between the active control laws.

The goals of this research are therefore defined as follows:

1. Create a virtual pilot algorithm that can create control inputs for a helicopter simulation in a manner that produces flight data similar to human piloted test flights. This algorithm will take as input a pre-defined flight test card.
2. The resulting prototype virtual pilot should be able to generate 15-20 distinct maneuvers and should ensure smooth transitions between each flight regime.
3. The virtual pilot algorithm should be validated in simulation in comparison with actual experimental HUMS flight data and other rotorcraft piloting standards as available.

The remainder of this thesis describes the virtual pilot design and preliminary simulation results that satisfy the goals outlined above.

CHAPTER 2 : VIRTUAL PILOT ALGORITHM

The work presented here focuses on the transformation of high level commands into feedback control laws. This essentially is taking the names of maneuvers like those written on a flight test card and converting them to meaningful instructions for conventional feedback control.

2.1 Virtual Pilot Architecture

The structure of the virtual pilot can be divided into three parts: parametrized maneuver scripting, interpretation of input maneuvers, and integration of tracking setpoints within a set of feedback loops. The first section of the virtual pilot is simply a user defined script which defines the actions that should be taken. This input script to the virtual pilot consists of a list of prescribed timed maneuvers to be executed in sequential order, and a corresponding list of numerical parameters for each maneuver. These user defined numerical values vary from forward speed or angle of bank depending on the nature of the maneuver. For example, a banked turn command will require an associated bank angle. Sometimes a maneuver may have several options for parametrizing the action. A vertical climb may be defined by a climb rate or a target altitude, while a lateral and longitudinal position must also be specified. Simple maneuver types are easily combined into "compound" maneuvers. For instance, a hover turn may be commanded at the same time as a vertical descent. The semantics of the user-defined input sequence are designed to be as simple as possible while including maximum functionality. In this manner, the regime recognition algorithm designer may provide the same flight test script to the virtual pilot as would be given to an actual test pilot performing a scripted flight test.

Table 2.1 List of Virtual Pilot Maneuvers

	Numerical Parameters	Default Values
Hover Maneuvers		
Hover	Position, Forward Speed	Current position, 0 ft/s
Hover Turn (Rate)	Rate of Heading Change	25 deg/s
Hover Turn (Heading)	Final Heading	Current + 45 deg
Axial Climb/Descent	Rate of Ascent/Descent	Climb: 8 ft/s Descent: 5 ft/s
Forward Flight Maneuvers		
Forward Flight	Forward Speed	Last speed
Sideslip	Angle of Sideslip	20 deg
Banked Turn	Bank Angle	35 deg
Heading Turn	Final Heading	Current + 90 deg
Level Climb	Forward Speed, Rate of Climb	Last speed, 35 ft/s
Level Descent	Forward Speed, Rate of Climb	Last speed, 25 ft/s
Landing	Forward Speed(s), Rate(s) of Descent, Transition Altitude	13 ft/s to 1 ft/s, 5 ft/s to 2 ft/s, transition at 45 ft
Autorotation	Forward Speed, Desired Rotor Velocity	80 knots, 105%
Open Loop Maneuvers		
Lateral Doublet	Duration, Control Deflection	2 sec, 0.5 deg
Collective Doublet	Duration, Control Deflection	2 sec, 3 deg
Roll Frequency Sweep	Duration, Max Deflection	3 sec, 1 deg
Pitch Frequency Sweep	Duration, Max Deflection	3 sec, 1 deg
Yaw Frequency Sweep	Duration, Max Deflection	3 sec, 1 deg
Symmetric Pullout	Duration, Control Deflection	4 sec, 1.5 deg
Symmetric Pushover	Duration, Control Deflection	4 sec, 3 deg

A list of maneuvers currently achievable by the prototype virtual pilot is shown in Table 2.1. Additional maneuvers may be defined in future work using the same implementation structure as this basic set. Some maneuvers can be combined as long as their control parameters and feedback loop structure do not interfere with the each other. The various feedback strategies are described in later sections. While the main numerical parameters for each maneuver are listed, more may be optionally entered. If optional parameters are not enumerated in the maneuver script, the next maneuver will inherit whatever value the optional parameter was in the last maneuver. For example, the forward velocity does not need to be explicitly re-entered for a banked turn maneuver coming from a forward flight maneuver with a specified velocity. This system is intuitive and works well for defining maneuvers as simply as possible, since only changes are to be specified. Default values for each maneuvers exist as seen in Table 2.1, so if a user calls for a level climb without specifying the climb rate, the default climb rate is used. In the cases where the new maneuver calls for a parameter which was previously unneeded or unspecified, the setpoint becomes the current state. For example, following a maneuver script going from forward flight to hover, the heretofore unused hover setpoint can either be specified by the script in absolute coordinates, coordinates relative the helicopter body, or left unspecified leaving the default to be the current position.

The second section of the virtual pilot consists of interpretation of the maneuver and its parameters into commanded values, or setpoints, which the feedback control loops can track. This section of the virtual pilot must methodically manage which control feedback loops should be active and manage the transitions between maneuvers. The third section of the virtual pilot directs the set-points to the appropriate feedback controller, implements control blending if necessary during a transition between controllers, and implements blending going to and from open loop control. These two elements of the virtual pilot used in conjunction make up the entire closed-loop command set used to produce final control outputs to the actuators. Each control update is able to be computed quickly at each time step after referencing the input script for the current maneuver.

Interpreting a maneuver consists of parsing the input script from recognizable named maneuvers, tracking parameters, and start-stop times. Different maneuvers are accomplished by activating and deactivating different feedback loops. The structure of the feedback control loops is shown in Figure 2.1. The layered feedback loop structure facilitates the changing control structure of the virtual pilot depending on the maneuver. For example, the outermost loop of position feedback is only active while in hovering maneuvers. Most maneuvers may share two or three of the same active control loops, especially the innermost loops. Only the necessary control loops will be activated by the interpreter during a particular maneuver, which practically means that only one controller in each layer can be active at a time. The outermost active control loop tracks a setpoint provided by the interpreter. The virtual pilot will simply define which setpoints are active and what values they should be.

The start and end times of each maneuver are kept track of within the interpretation section of the virtual pilot, so that changes can be anticipated without creating discontinuities while switching control laws. The authors found that effective management of the transition of one maneuver to another was essential for preventing unwanted transients in the control output as the controllers changed. Such discontinuities in control could cause large disturbances and oscillations in the aircraft dynamic response. The virtual pilot not only ensures continuity in all control axes, but also ensures first derivative continuity using cubic splines near the transition points. Control set-points are preserved across maneuvers if they use that particular controller in common. When a new controller starts without a previous reference set-point, an estimate can be made from the last physical state of the helicopter. This comes into play particularly when moving from rate commands to position commands, such as a level climb transitioning to straight and level flight. When a set-point changes, the difference is gradually faded in over time, again with a cubic spline for first derivative continuity. The time allowed for a transition is 4 sec at the beginning and end of each maneuver, with more time allowed for changes in airspeed or a flight mode switch.

Open loop maneuvers such as a symmetric or doublet maneuvers are also achievable with the virtual pilot. Open loop commands can also be linearly blended with closed loop commands in order to produce smooth transitions between these types of maneuvers.

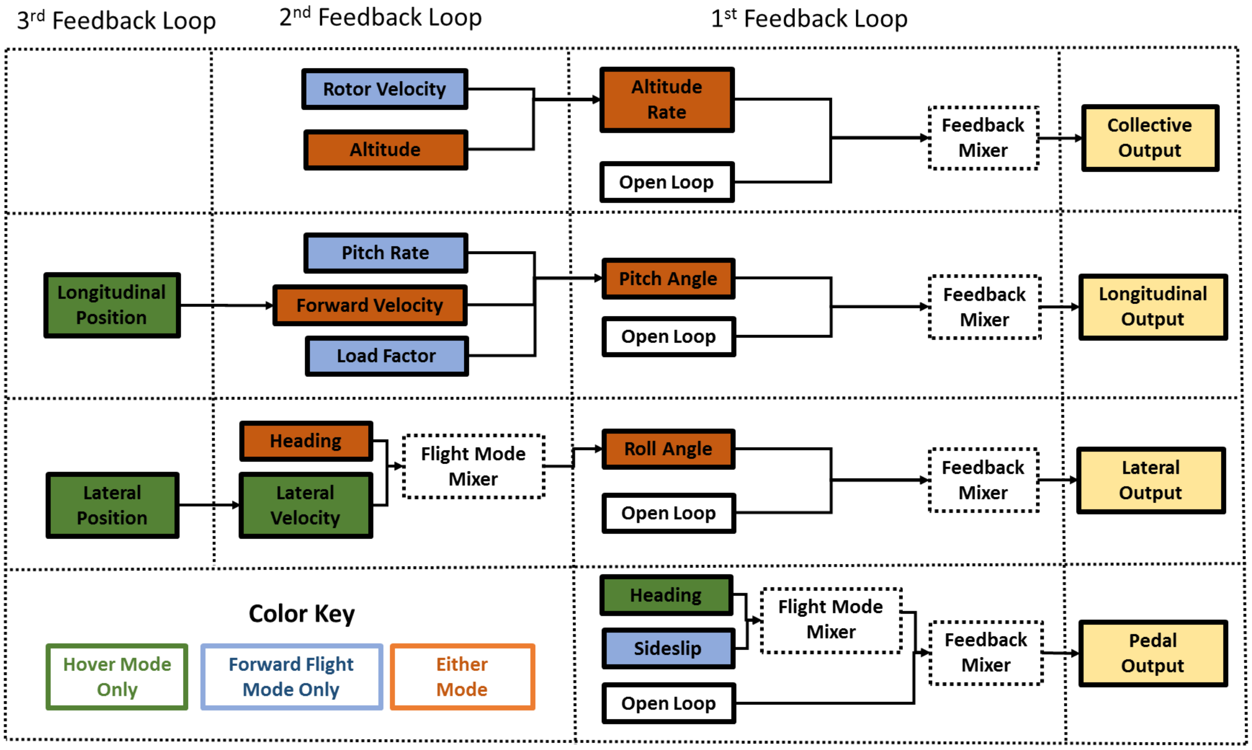

Figure 2.1: Structure of Feedback Loops within the Virtual Pilot

The third section of the virtual pilot is simply responsible for routing the commanded set-points to the appropriate controller. Notice from Figure 2.1 that the inner control loops are the same in all cases except for the pedal. The outer controllers must produce a set-point that can be used by the next inner controller. This layered structure makes it easy to tune proportional-integral-derivative gains from the inside out, as the performance of the inner loop can be used by an outer loop to achieve a more complex set-point.

Flight mode mixing combines the commands of two controllers during a flight mode transition using a fuzzy logic scheme. The purpose of the linear blend of flight modes is mainly for continuity and should only take a few seconds so that the two controllers do not fight each other. The flight mode transition can also be made immediate if desired. Flight maneuvers were categorized as belonging to either “forward flight” or “hover” modes. The flight mode determines which set of controllers should be active in the feedback loops, much like activated set-points but with the ability to compute both at once and blend them. Only the lateral controller and pedal controller need flight mode mixing. In forward flight, the heading is achieved by rolling through the lateral axis while the tail rotor minimizes sideslip. Conversely, hover mode uses position feedback to control the lateral axis and the tail to track a commanded heading. The differing structures of the pedal feedback controller in hover and forward flight (regulating heading error vs regulating sideslip) necessitate this fuzzy blending scheme so that the controller structure itself is not subject to a sudden step change.

2.2 Forward Flight Maneuvers

The assumption that coordinated flight is desirable is the main distinction between the categorization of forward flight maneuvers and hover maneuvers. Where hover mode may allow pure lateral motion with a fixed heading, forward flight mode minimizes sideslip at all times (unless in a purposeful sideslip maneuver). In forward flight, the roll controller is used to achieve smooth, coordinated banking turns for any desired change in heading. Within the forward flight mode, changes can be made to the desired climb rate, bank angle, and even desired sideslip without changing any of the controller structure. The feedback loop structures for forward flight mode are shown below. Note that setpoints can take the place of calculated PID values in the inner loops, which corresponds to more direct control for the virtual pilot. This is used to implement maneuvers where outer loops are removed and intermediate loop setpoints are injected directly.

$$e_u = u_{desired} - u \tag{1}$$

$$e_\theta = (\theta_{desired} - \theta) \ or \ k_{p\_pitch\_outer} * e_u + k_{i\_pitch\_outer} * \int e_u + k_{d\_pitch\_outer} * \dot{u} \tag{2}$$

$$\theta_{1s} = k_{p\_pitch\_inner} * e_\theta + k_{i\_pitch\_inner} * \int e_\theta + k_{d\_pitch\_inner} * \dot{\theta} \tag{3}$$

The equations shown above express the layered PID feedback loops for longitudinal control. The block diagram related the longitudinal control is shown below. The innermost loop governs the longitudinal orientation, or pitch angle. The outer loop regulates the longitudinal speed.

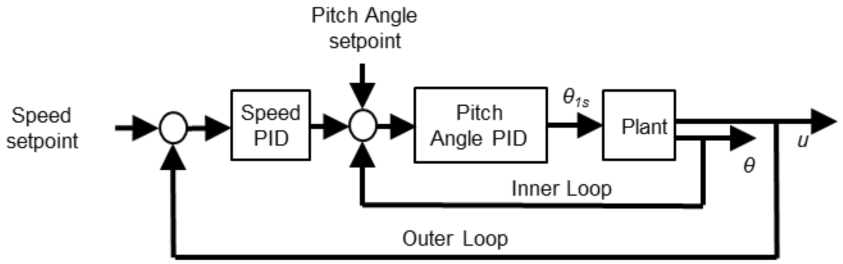

Figure 2.2: Forward Flight Longitudinal Cyclic Control

$$e_\Psi = \Psi_{desired} - \Psi \tag{4}$$

$$e_\varphi = (\varphi_{desired} - \varphi) \ or \ k_{p\_roll\_outer} * e_\Psi + k_{i\_roll\_outer} * \int e_\Psi + k_{d\_roll\_outer} * \dot{\Psi} \tag{5}$$

$$\theta_{1c} = k_{p\_roll\_inner} * e_\varphi + k_{i\_roll\_inner} * \int e_\varphi + k_{d\_roll\_inner} * \dot{\varphi} \tag{6}$$

The equations shown above express the layered PID feedback loops for lateral control. The block diagram related the lateral control is shown below. The innermost loop governs the lateral orientation, or roll angle. The outer loop regulates the lateral speed.

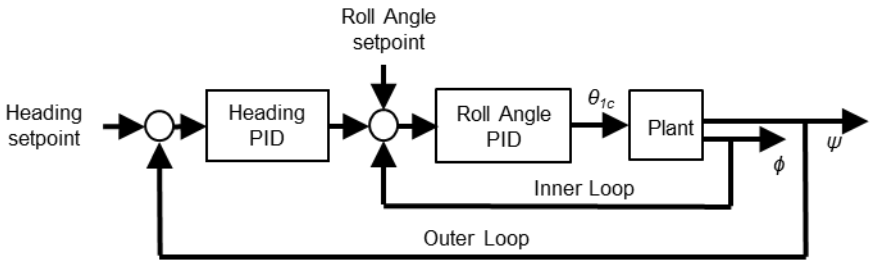

Figure 2.3: Forward Flight Lateral Cyclic Control

$$e_z = z_{desired} - z \quad \textbf{7}$$

$$e_{\dot{z}} = (\dot{z_{desired}} - \dot{z}) \ \ or \ \ k_{p\_col\_outer} * e_z + k_{i\_col\_outer} * \int e_z + \ k_{d\_col\_outer} * \dot{z} \quad \textbf{8}$$

$$\theta_0 = k_{p\_col\_inner} * e_{\dot{z}} + k_{i\_col\_inner} * \int e_{\dot{z}} + \ k_{d\_col\_inner} * \ddot{z} \quad \textbf{9}$$

The equations shown above express the layered PID feedback loops for collective control. The block diagram related the heave control is shown below. The innermost loop governs the vertical velocity, while the outer loop regulates the vertical position, or altitude.

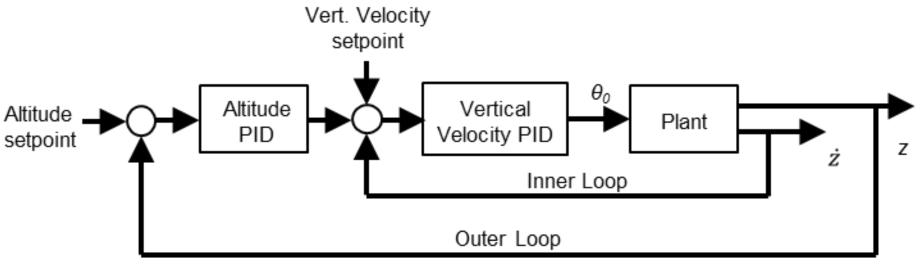

Figure 2.4: Forward Flight Collective Control

$$e_\beta = \beta_{desired} - \beta \quad \textbf{10}$$

$$\theta_{0TR} = \ k_{p\_rud\_inner} * e_\beta + k_{i\_rud\_inner} * \int e_\beta + \ k_{d\_rud\_inner} * \dot{\beta} \quad \textbf{11}$$

The equations shown above express the PID feedback loop for rudder control. The block diagram related the rudder control is shown below. This is a single loop regulating the angle of sideslip. It is important to note that when an outer loop is active, it completely provides the setpoint for the inner loop, and when an inner loop setpoint is provided, the outer loop is not active.

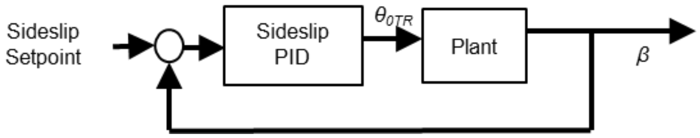

Figure 2.5: Forward Flight Rudder Control

2.3 Hover Flight Maneuvers

The hover mode adds another layer to the cyclic controls for position feedback and uses a pedal control that tracks heading rather than sideslip. The collective controller remains the same. Maneuvers in the hover regime include hover, hover turns, vertical climbs and descents, and slow speed uncoordinated repositioning.

$$e_x = (x_{desired} - x) \qquad \textbf{12}$$

$$e_u = (u_{desired} - u) \; or \; k_{p\_pitch\_outer} * e_x + k_{i_{pitch_{outer}}} * \int e_x + k_{d_{pitch_{outer}}} * \dot{x} \qquad \textbf{13}$$

$$e_\theta = (\Theta_{desired} - \Theta) \; or \; k_{p\_pitch\_mid} * e_u + k_{i\_pitch\_mid} * \int e_u + k_{d\_pitch\_mid} * \dot{u} \qquad \textbf{14}$$

$$\theta_{1s} = k_{p\_pitch\_inner} * e_\theta + k_{i\_pitch\_inner} * \int e_\theta + k_{d\_pitch\_inner} * \dot{\theta} \qquad \textbf{15}$$

These equations describe the layered PID feedback control loops for the longitudinal control. Note that they are just like the forward flight longitudinal control loops, with the addition of another outer loop that regulates the longitudinal position of the helicopter. The same is true for the lateral control axis. Here, the outermost loop will convert a position error into a speed setpoint for the middle loop. The block diagram for the longitudinal control is shown below.

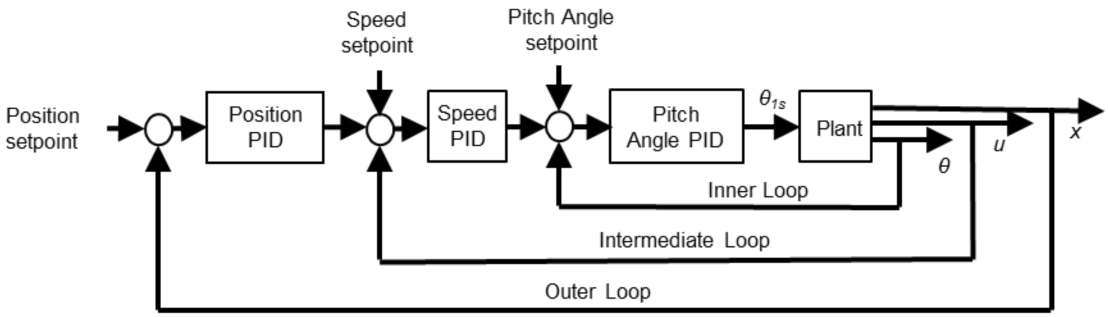

Figure 2.6: Hover Longitudinal Cyclic Control

$$e_y = y_{desired} - y \qquad \textbf{16}$$

$$e_v = (v_{desired} - v) \ or \ k_{p\_roll\_outer} * e_y + k_{i\_roll\_outer} * \int e_y + k_{d\_roll\_outer} * \dot{y} \qquad \textbf{17}$$

$$e_\varphi = (\varphi_{desired} - \varphi) \ or \ k_{p\_roll\_outer} * e_v + k_{i\_roll\_outer} * \int e_v + k_{d\_roll\_outer} * \dot{v} \qquad \textbf{18}$$

$$\theta_{1c} = k_{p\_roll\_inner} * e_\varphi + k_{i\_roll\_inner} * \int e_\varphi + k_{d\_roll\_inner} * \dot{\varphi} \qquad \textbf{19}$$

These equations describe the layered PID feedback control loops for the lateral control. The block diagram for the lateral control is shown below.

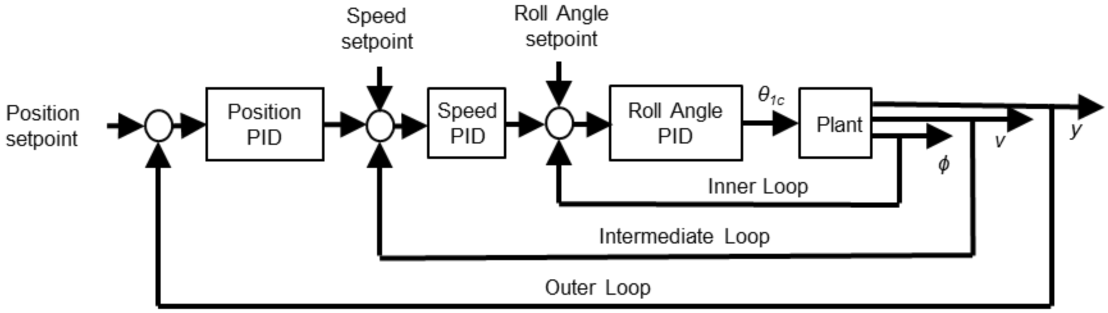

Figure 2.7: Hover Lateral Cyclic Control

$$e_\Psi = (\Psi_{desired} - \Psi) \qquad \textbf{20}$$

$$e_{\dot{\Psi}} = (\dot{\Psi}_{desired} - \dot{\Psi}) \ or \ k_{p\_rud\_outer} * e_\Psi + k_{i\_rud\_outer} * \int e_\Psi + k_{d\_rud\_outer} * \dot{\Psi} \qquad \textbf{21}$$

$$\theta_{0TR} = k_{p\_rud\_inner} * e_{\dot{\Psi}} + k_{i\_rud\_inner} * \int e_{\dot{\Psi}} + k_{d\_rud\_inner} * \ddot{\Psi} \qquad \textbf{22}$$

The equations above describe the layered PID feedback loops used for the rudder control. These take on a form similar to the collective controller, with the rate controller in the inner loop. Here, a yaw angle error will be converted into a yaw rate setpoint. The block diagram for the rudder control in hover is shown below.

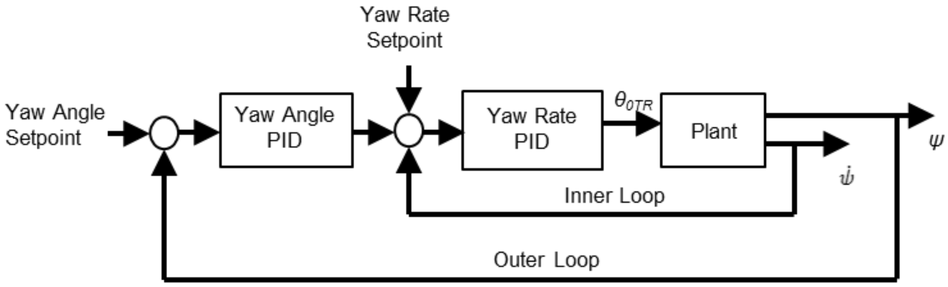

Figure 2.8: Hover Rudder Control

2.4 Open Loop Maneuvers

A category of open loop maneuvers was created for actions that may not require feedback control, such as symmetric pullups, pushovers, and doublets. Feedback loops may still be active for some control inputs in order to maintain the forward flight trim condition. Otherwise, open loop maneuvers hold the last control input from the end of the last maneuver and add piecewise constant deflections to achieve the desired effect.

All control axes in the autopilot architecture considered here have feedback loops that are structured with position feedback in the outer loops and rate feedback in the inner loops. This is a traditional setup and works well as a layered PID system. An explicit list of which control loops are active for each maneuver is shown in Table 2.2 below. Note that the following abbreviations are used in the table: SS for sideslip, OL for open loop, LF for load factor, as well as three letter abbreviations for position, velocity, orientation angle, and heading.

Table 2.2 Maneuvers and their active feedback control loops

Hover Maneuvers	**Collective Outer**	**Collective Inner**	**Long. Cyc. Outer**	**Long. Cyc. Mid.**	**Long. Cyc. Inner**	**Lat. Cyc. Outer**	**Lat. Cyc. Mid.**	**Lat. Cyc. Inner**	**Pedal Outer**	**Pedal Inner**
Hover	Pos.	Vel.	Pos.	Vel.	Ang.	Pos.	Vel.	Ang.	Hdg.	Rate
Hover Turn (Rate)	Pos.	Vel.	Pos.	Vel.	Ang.	Pos.	Vel.	Ang.		Rate
Hover Turn (Heading)	Pos.	Vel.	Pos.	Vel.	Ang.	Pos.	Vel.	Ang.	Hdg.	Rate
Axial Climb/Descent		Vel.	Pos.	Vel.	Ang.	Pos.	Vel.	Ang.	Hdg.	Rate
Forward Flight Maneuvers										
Forward Flight	Pos.	Vel.		Vel.	Ang.		Hdg.	Ang.		SS
Sideslip	Pos.	Vel.		Vel.	Ang.		Hdg.	Ang.		SS
Banked Turn	Pos.	Vel.		Vel.	Ang.			Ang.		SS
Heading Turn	Pos.	Vel.		Vel.	Ang.		Hdg.	Ang.		SS
Level Climb		Vel.		Vel.	Ang.		Hdg.	Ang.		SS
Level Descent		Vel.		Vel.	Ang.		Hdg.	Ang.		SS
Landing		Vel.		Vel.	Ang.		Hdg.	Ang.		SS
Autorotation	Rotor Rpm	Vel.		Vel.	Ang.		Hdg.	Ang.		SS
Open Loop Maneuvers										
Lateral Doublet	Pos.	Vel.		Vel.	Ang.			OL		SS
Collective Doublet		OL		Vel.	Ang.		Hdg.	Ang.		SS
Roll Frequency Sweep	Pos.	Vel.		Vel.	Ang.		Hdg.	Ang. + OL		SS
Pitch Frequency Sweep	Pos.	Vel.		Vel.	Ang. + OL		Hdg.	Ang.		SS
Yaw Frequency Sweep	Pos.	Vel.		Vel.	Ang.		Hdg.	Ang.		SS + OL
Symmetric Pullout		OL		LF	Ang.		Hdg.	Ang.		SS
Symmetric Pushover		OL		LF	Ang.		Hdg.	Ang.		SS

An axial climb maneuver will be used as an example to understand the above table. This maneuver’s specific feedback loops include three layers on lateral and longitudinal cyclic for position hold, one layer in the collective feedback loop to track vertical velocity, and two layers in the yaw feedback loop. This structure is most simple in order to accomplish the goals of an axial climb: maintain lateral and longitudinal position, maintain heading, and track a vertical velocity. These loops are explicitly shown in Figure 2.9 as an example of a single maneuver’s structure.

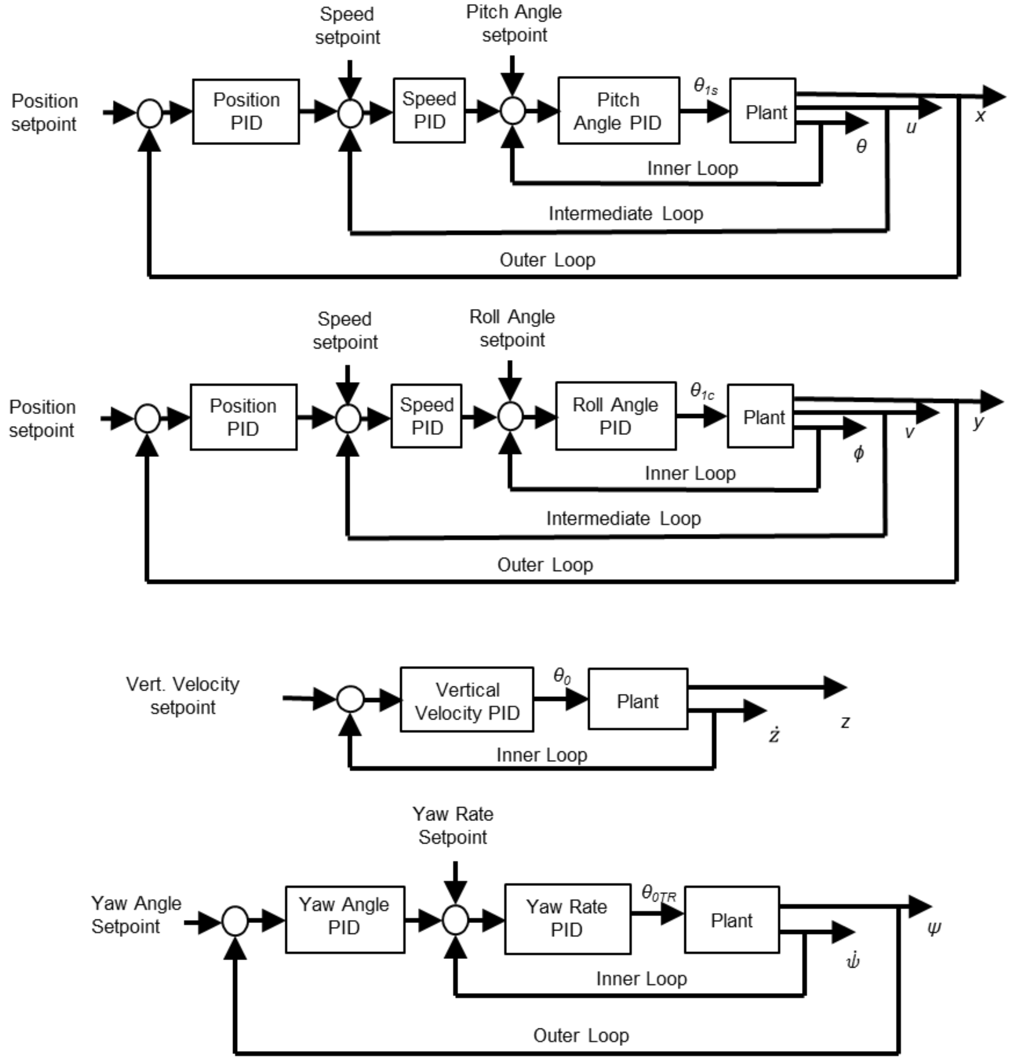

Figure 2.9: Example of the set of feedback loops for axial climb

2.5 Maneuver Transitions

A great deal of effort was taken to make the transitions between scripted maneuvers as smooth as possible. A control system should smoothly manage a transition from one simple maneuver to another in order to avoid transient responses, which result in momentary control saturation and large forces or moments applied to the dynamic system. Several techniques have been developed by previous researchers that mitigate these kinds of effects when switching controllers. Variable Structure Control makes use of carefully designed switching functions that enact changes to a linear control system to ensure stability [13]. Robotic manipulators that utilize separate feedback from both position and force can use hybrid control, which links position and force constraints in separate controllers and uses corrector functions to mix the output of the two feedback loops [14]. However, the most applicable of industrial methods to the work presented here are concepts from a control technique called the Bumpless Transfer Method. In this method, auxiliary control inputs are generated at the transition between control laws called realizable references [15]. Realizable references are calculated in order to minimize the difference in control output using the new controller from the old controller. This reference is then used as a starting input when the new controller takes effect. This basic idea is used during the transition between layers of the cascade style PID feedback loops in the virtual pilot control structure. Bumps in the control output are minimized by ramping in the new desired setpoint from a "realizable reference". For example, when changing from forward flight to a level climb, the collective controller loses the outer loop altitude feedback. The new altitude rate setpoint is then initialized at the current rate of change of altitude. Therefore, all states and their derivatives that might be useful as an initial condition for the next maneuver are stored during a transition.

A timed system of fading in new setpoint values was implemented in order to further smooth the maneuver transitions. A cubic curve ramp was used instead of a linear ramp in order to preserve first derivative continuity for the commanded setpoints. This ramp can be spread over a desired fade-in time, which can be altered to suit the expected magnitude of the change. The equation for this cubic ramp is based on a linear ramp, as shown below.

$$r_{lin} = \frac{t_{elapsed}}{t_{ramp}} \tag{23}$$

$$r_{cubic} = -2(r_{lin})^3 + 3\,(r_{lin})^2 \tag{24}$$

where t_{ramp} is the desired duration of the ramp, and $t_{elapsed}$ is the time since the start of the maneuver change. The time for the ramp is 5 seconds, with proportionally more time allowed in a change in forward velocity is commanded. This is to allow larger changes in setpoints to have more time to be implemented. If the time allotted for a particular maneuver is shorter than the recommended ramp time, the whole maneuver time is used for the ramp and a warning is issued. These cubic ramps are also used at the end of a maneuver when a return to the previous condition is expected, such as when transitioning from level flight to a sideslip maneuver and then back to level flight.

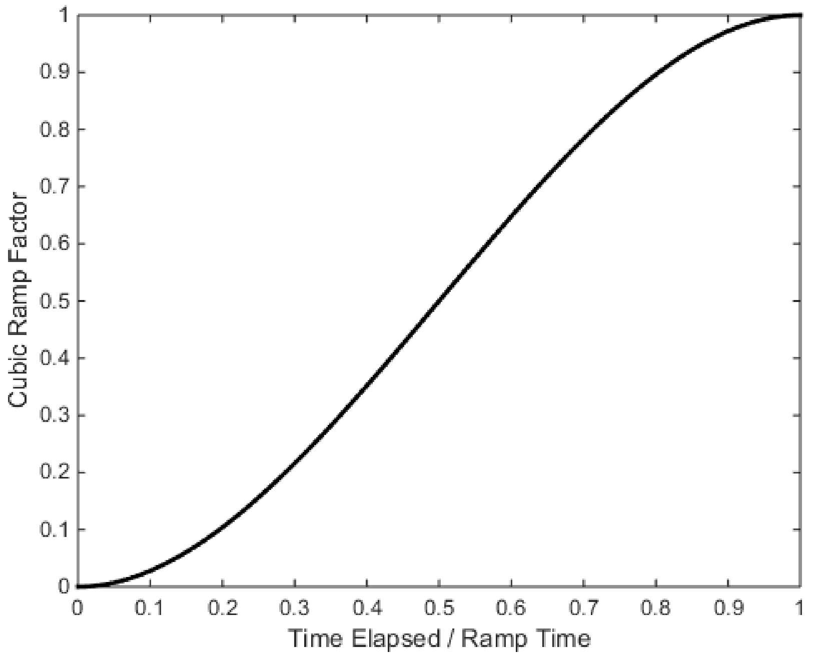

Figure 2.10: Cubic ramp curve used when changing controller setpoints.

Similar ramps can also be used for fuzzy control transitions between the hover and forward flight modes. A cubic ramp is used to calculate the proportional blending values assigned to the two simultaneously active controllers. An equation showing the proportional blending scheme is shown below.

$$control_{fuzzy} = r_{cubic} * control_{hover} + (1 - r_{cubic}) * control_{forward\ flight}$$

Typically, these transitions should be very short in order to avoid the two controller types fighting for control in order to achieve slightly different goals. In the results shown in the this thesis, no fuzzy control is used during the hover to forward flight transitions, but the functionality remains if a user should decide to utilize it.

CHAPTER 3 : SIMULATION MODEL

A high-fidelity helicopter simulation model was used to validate the efficacy of the proposed virtual pilot. A detailed model is required in order to produce flight data that accurately represents the various flight regimes. The output of the system should be similar to the real flight characteristics of the helicopter in order to produce valid training data for the candidate regime recognition algorithm. Matching control inputs to the real world data is not as critical since recognition algorithms use only the recorded output states of the helicopter. The model used is similar to the ARMCOP model develop by Talbot and Chen [13] [14] [15], with improvements made to the main rotor model incorporating dynamic inflow, ground effect, and blade stall as implemented by Sunberg *et al* in [16]. There are twelve components in the state vector used to describe rigid body motion of the helicopter, six components for blade flapping, and three components for dynamic inflow. The state is integrated using a fourth order Runga-Kutta method with a timestep of 0.001 seconds. These 22 states are

$$\text{state } \vec{\mathrm{x}} = [x, y, z, \varphi, \theta, \Psi, u, v, w, p, q, r, \Omega, \beta_0, \beta_{1s}, \beta_{1c}, \dot{\beta}_0, \dot{\beta}_{1s}, \dot{\beta}_{1c}, \lambda_0, \lambda_s, \lambda_c] \qquad \mathbf{3.1}$$

Orientation is represented by a 3-2-1 Euler angle rotation sequence through Ψ, θ, and Φ. The velocities and angular rates represented in the body frame are called *u*, *v*, *w*, *p*, *q*, and *r*. The rigid body equations of motion are presented in equations 3.2-3.5. Note the use of shorthand notation $c_{(\cdot)}$ for *cos*(·) and $s_{(\cdot)}$ for *sin*(·)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad \textbf{3.2}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{c_\theta} \begin{bmatrix} c_\theta & s_\phi s_\theta & c_\phi s_\theta \\ 0 & c_\phi c_\theta & -s_\phi c_\theta \\ 0 & s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad \textbf{3.3}$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} - \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad \textbf{3.4}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = [I_c]_B^{-1} \left\{ \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} - \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} [I_c]_B \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right\} \quad \textbf{3.5}$$

In these equations, the total helicopter mass if represented by *m*, and the body frame forces and moments are represented by $F_{x,y,z}$ and $M_{x,y,z}$, which act about the helicopter mass center and include contributions from the helicopter weight, main and tail rotors, and fuselage and empennage aerodynamics.

3.1 Rotor Dynamics

A detailed rotor model is central to the design of a physics-based helicopter simulation. The rotors provide the largest aerodynamic forces and moments, including the main source of lift for the aircraft, and handle all of the control actuation. These rotor dynamics are modeled with a numerical blade element approach. Each rotor blade has 15 stations along its length for which 2D aerodynamic analysis is performed at 30 rotational stations within a complete revolution of the rotor head. Based on the velocity of the body and the induced inflow, the forces on the blade element are calculated using a lookup table for the airfoil aerodynamics. The tables include wind tunnel data for angles of attack through 360°, so static stall behavior is implicitly modeled. The forces are then summed and normalized in order to obtain the aerodynamic forces exerted by the whole rotor on the rest of the aircraft. These forces also affect the rotational velocity of the main rotor Ω when simulating a power-off autorotation maneuver.

3.2 Blade Flapping

Determining the flapping motion of the rotor blades is critical for the accuracy of the above force and moment calculations. First harmonic blade flapping effects are included in this model. Higher-harmonic flapping dynamics are generally very small compared to the first harmonic components and are therefore neglected. The equation including first harmonic effects for the flapping angle as a function of the azimuth angle of the rotor blade is given by

$$\beta(\Psi_{MR}) = \beta_0 + \beta_{1c}\cos(\Psi_{MR}) - \beta_{1s}\sin(\Psi_{MR}) \qquad \textbf{3.6}$$

where a positive β represents downward flapping. The flapping component β_0 represents the coning angle, and β_{1c} and β_{1s} are defined as the longitudinal and lateral flapping angles, respectively. The differential equation that governs flapping is given by

$$\ddot{\beta} + \omega_N{}^2\beta = M_F \qquad \textbf{3.7}$$

where

$$\omega_N = \Omega\sqrt{\frac{I_B + \frac{meR}{2}}{I_B}} \qquad \textbf{3.8}$$

In the above equations, the blade mass is represented by *m,* the blade radius is *R*, the flap hinge offset is *e*, and the blade flap-wise inertia is I_B. M_F in equation 3.8 is the total moment acting about the blade flapping hinge calculated through blade element theory and inertial moments. See references [14] and [15] for additional details.

3.3 Dynamic Inflow

Three additional states, λ_0, λ_s, and λ_c are used in order to describe the induced inflow ratio distribution over the rotor disk. This model was first described by Peters and HaQuang [17], which is based on the Pitt and Peters model [18]. This industry standard model for dynamic inflow for single rotor helicopters is widely used and has been experimentally verified [19]. The dynamic inflow at a given radius and azimuth angle is given by

$$\lambda_i(r, \Psi_{MR}) = \lambda_0 + \lambda_s \frac{r}{R}\sin(\Psi_{MR}) + \lambda_c \frac{r}{R}\cos(\Psi_{MR}) \qquad \textbf{3.9}$$

These states are propagated according to the dynamic equation

$$[M]\begin{bmatrix} \dot{\lambda}_0 \\ \dot{\lambda}_s \\ \dot{\lambda}_c \end{bmatrix} + [\hat{L}]^{-1}\begin{bmatrix} \lambda_0 \\ \lambda_s \\ \lambda_c \end{bmatrix} = \boldsymbol{C} \qquad \textbf{3.10}$$

In equation 3.10, $\boldsymbol{C}$ is a vector of force and moment coefficients calculated using the blade element approach based on the stations described in Section 3.1. The matrix $[\hat{L}]$ is consists of elements related to the sideslip angle and wake angle, and matrix $[M]$ is based on the mass of air near the rotor. Additional details regarding this model can be found in Reference [17].

A simple ground effect correction is applied to the dynamic inflow model when the rotor is within two rotor diameters of the ground. A more detailed description of this correction can be found in [16].

3.4 Tail Rotor, Fuselage, Empennage, Stabilizers

For the tail rotor, we assume uniform inflow and near-steady state blade flapping because the dynamics for the tail rotor are fast enough to be neglected for this level of control and handling qualities research [13]. Newton-Raphson iteration is used to calculate uniform tail rotor inflow, and the blade flapping dynamic equations are solved for steady state at the current conditions ($\ddot{\beta} = \dot{\beta} = 0$). The other components of the helicopter model apply body-frame forces and moments using standard methods for modeling fuselage, empennage, and stabilizer aerodynamics [13].

3.5 Actuators

A simple rate limiter is applied to the actuator dynamics and a maximum and minimum range of actuator deflection is enforced. For the main rotor swash plate of an SH-60B, the range limits on the collective are from 9.9° to 25.9°, the lateral cyclic from +/- 8°, and the longitudinal cyclic from -12.3° to 16.5°. The maximum actuator rate is assumed to be 40° /s. Control from the virtual pilot is updated at 20 Hz. The actuators will respond as quickly as possible without exceeding the maximum rate.

CHAPTER 4 : SIMULATION RESULTS

The virtual pilot has been tested using the above nonlinear 6-degree-of-freedom helicopter model and compared to actual HUMS data from the SH-60B. First, we demonstrate the capability of the virtual pilot algorithm to generate completely new flight data sets following new scripted maneuver sequences. Secondly, the virtual pilot was tasked with executing standard maneuvers within certain tolerances defined in ADS-33 [20]. Various Mission Task Elements are shown to be within the adequate, and often desirable, execution criteria. Finally, maneuver sequences are written to match onboard HUMS flight data from actual flight tests in an SH-60B. In this manner, a direct comparison of the virtual pilot and a real pilot running the same maneuvers can be made. Two such matching sequences are presented here, demonstrating the capability of the virtual pilot to carry out the same maneuvers in a very similar manner to a real pilot. Table 4.1 lists some of the important model parameters assumed in the following simulations.

Table 4.1: Sikorsky SH-60B Model Parameters

Parameters	Symbol	Value
Helicopter gross weight	W	16,000 lbs
Number of main rotor blades	N_b	4
Main rotor blade chord	c	1.73 ft
Main rotor radius	R	26.83 ft
Main rotor blade moment of inertia	I_B	1491 slug ft^2
Main rotor height above ground (water line)	WL_{MR}	12 ft
Main rotor normal operating speed	Ω_{normal}	27.0 rad/s
Main rotor blade airfoil used for simulation		SC 1095
Actuator max rate	$\dot{\delta}_{max}$	40 deg/s
Controller update rate		20 Hz

4.1 Example Maneuver Sequences

For the following plots of state time histories, the input to the virtual pilot is provided in preceding tables. These maneuver scripts are the entirety of the guidance information given to the virtual pilot in order to produce the following trajectories. Numerical parameters detailing speeds or heading changes are provided for each listed maneuver.

Several examples of virtual pilot flight data generation will be given as samples of what the algorithm is capable of. The first example maneuver sequence consists of gentle forward flight and 90 degree turns, followed by a level climb.

Table 4.2 Example Maneuver Sequence 1

Time [s]	*Maneuver*
0	Forward Flight 0.75Vh
15	Right Heading Turn, 90°
30	Forward Flight 0.75Vh
40	Left Heading Turn, 90°
55	Forward Flight 0.75Vh
65	Left Heading Turn, 180°
95	Level Climb, 35 ft/s

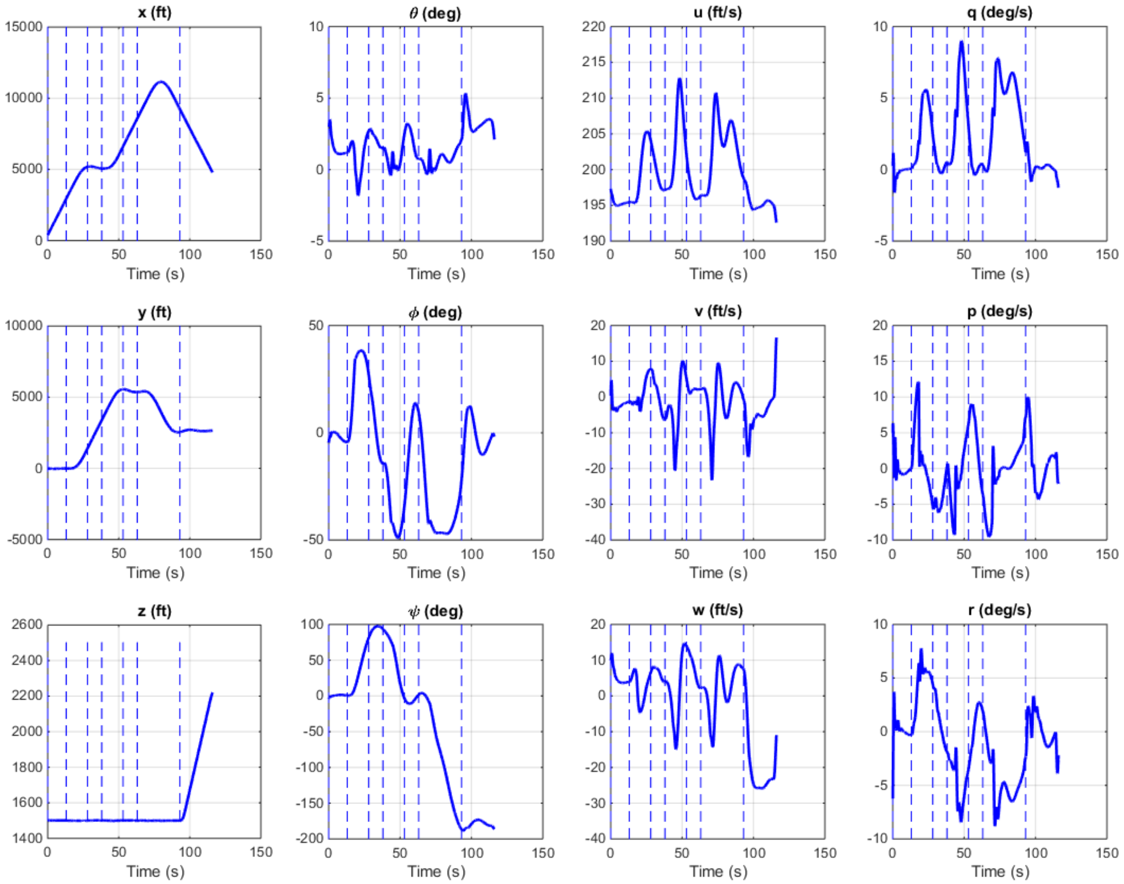

Figure 4.1: Time history of a simple maneuver sequence with turns and a level climb.

Figure 4.1 shows the full 6DOF state of the SH-60B, in which smooth changes in heading can be observed as the 90 degree turns are taken, as well as the corresponding changes in bank angle. Vertical dashed lines in this figure denote maneuver switches. From the overhead view of this trajectory in Figure 4.2, the gentle nature of the turns and transitions is evident as setpoints and control structure changes in between the maneuvers.

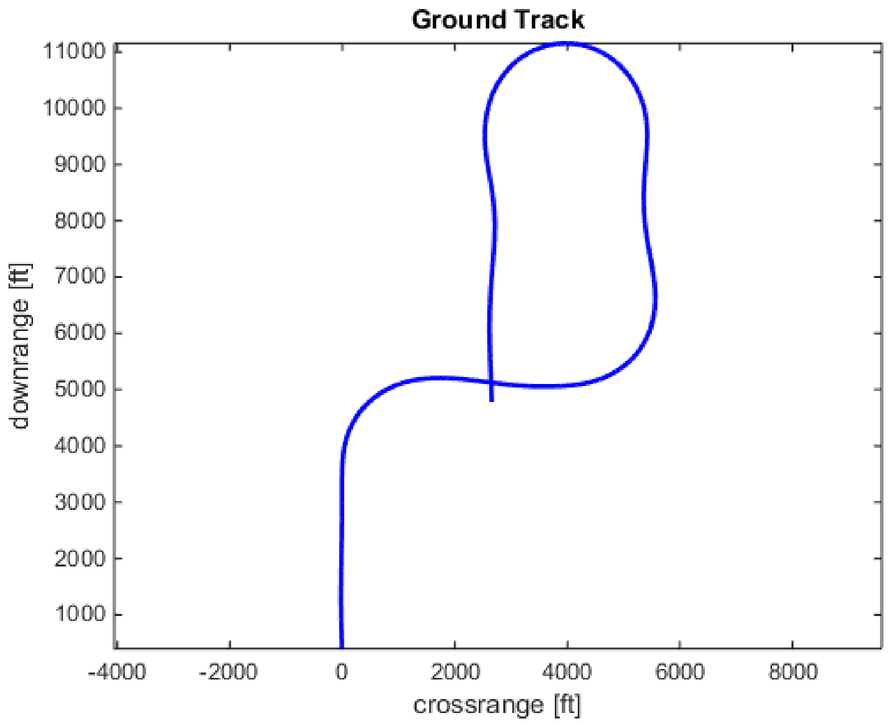

Figure 4.2: The overhead view of the trajectory taken for the first example sequence

For the next example maneuver sequence, the sideslip maneuver is demonstrated along with a level descent and a velocity increase.

Table 4.3 Example Maneuver Sequence 2

Time [s]	*Maneuver*
0	Forward Flight 0.4Vh
20	Left Sideslip 25°
42	Forward Flight 0.4Vh
52	Right Sideslip 15°
74	Forward Flight 0.4Vh
84	Level Descent 25 ft/s
100	Forward Flight 0.6Vh

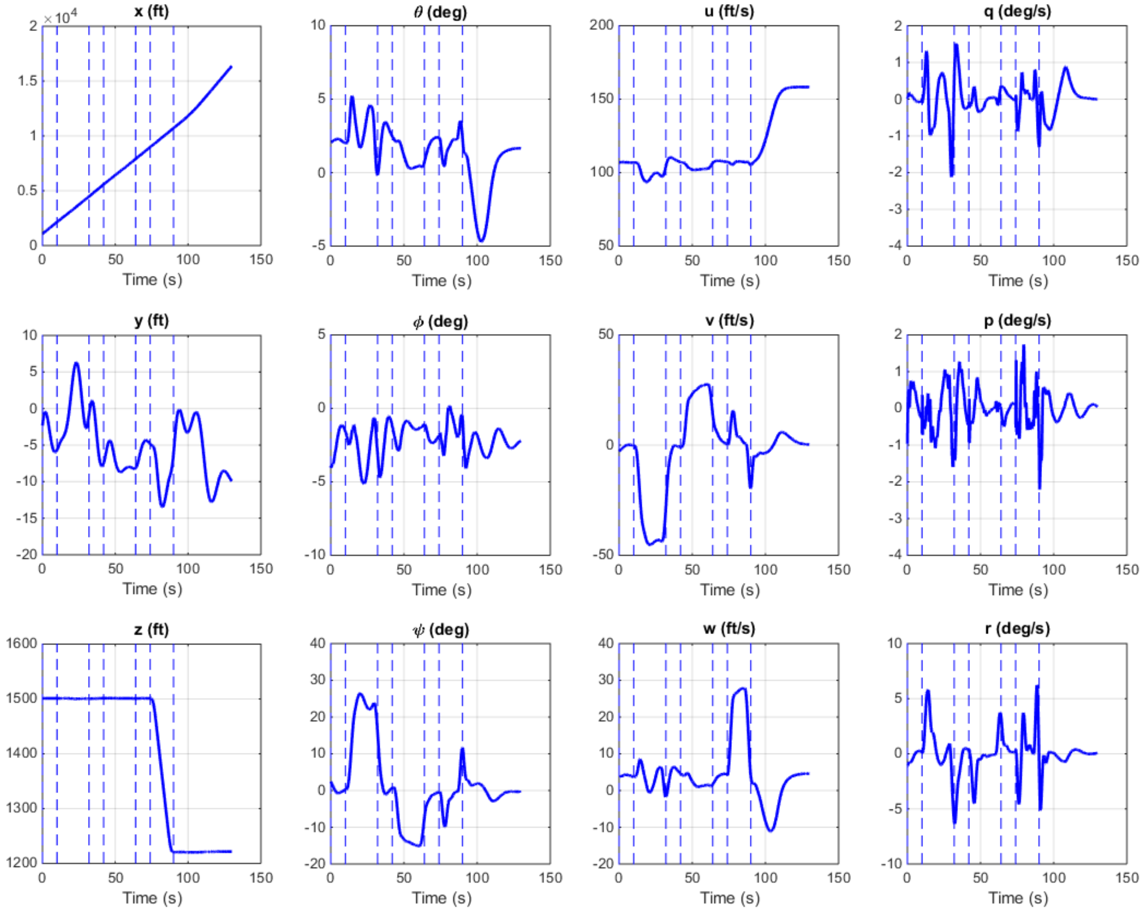

Figure 4.3: Time history of an example maneuver sequence with sideslip and a level descent

Notice from the heading angle in Figure 4.3 that two gradations of sideslip in opposite directions was achieved as prescribed in the maneuver script. Motion in the crossrange direction (y) is kept within 20 ft as all these maneuvers are accomplished. Smooth increases and decreases in vertical velocity are observed during the transitions to and from level descent, as well as a smooth increase in forward velocity after that.

The next maneuver sequence demonstrates hover maneuvers, transition to forward flight, and an autorotation.

Table 4.4 Example Maneuver Sequence 3

Time [s]	*Maneuver*
0	Hover
7	Hover Turn Heading Left, 135°
20	Hover
35	Hover Turn Rate Right, 25°/s
50	Hover
55	Axial Climb, 8 ft/s
70	Hover
80	Transition to Forward Flight, 0.4Vh
95	Level Climb, 35 ft/s
110	Forward Flight, 0.4Vh
120	Autorotate, 80knots, 105% rpm
135	Autorotation Recovery
140	Forward Flight, 0.4Vh

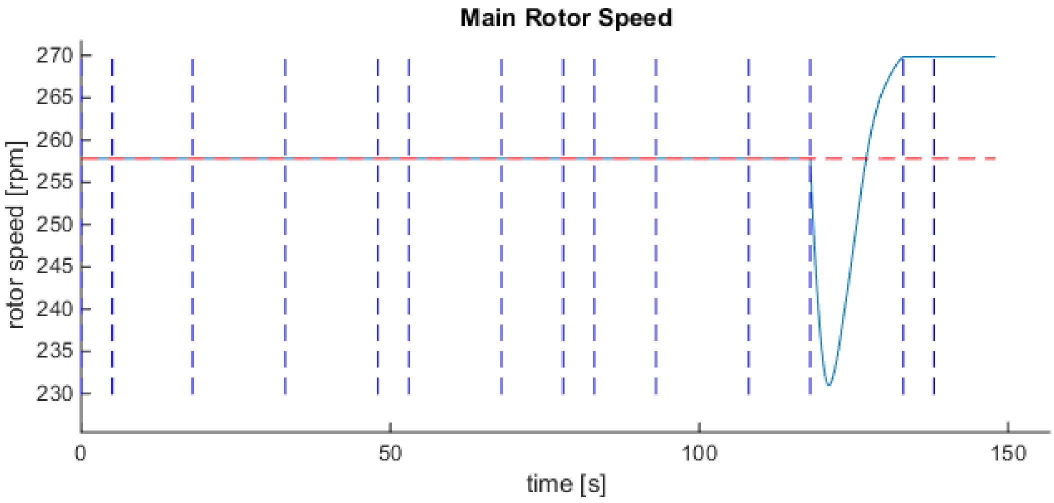

Figure 4.4: Rotor rpm over time for example sequence 3, with autorotation and recovery

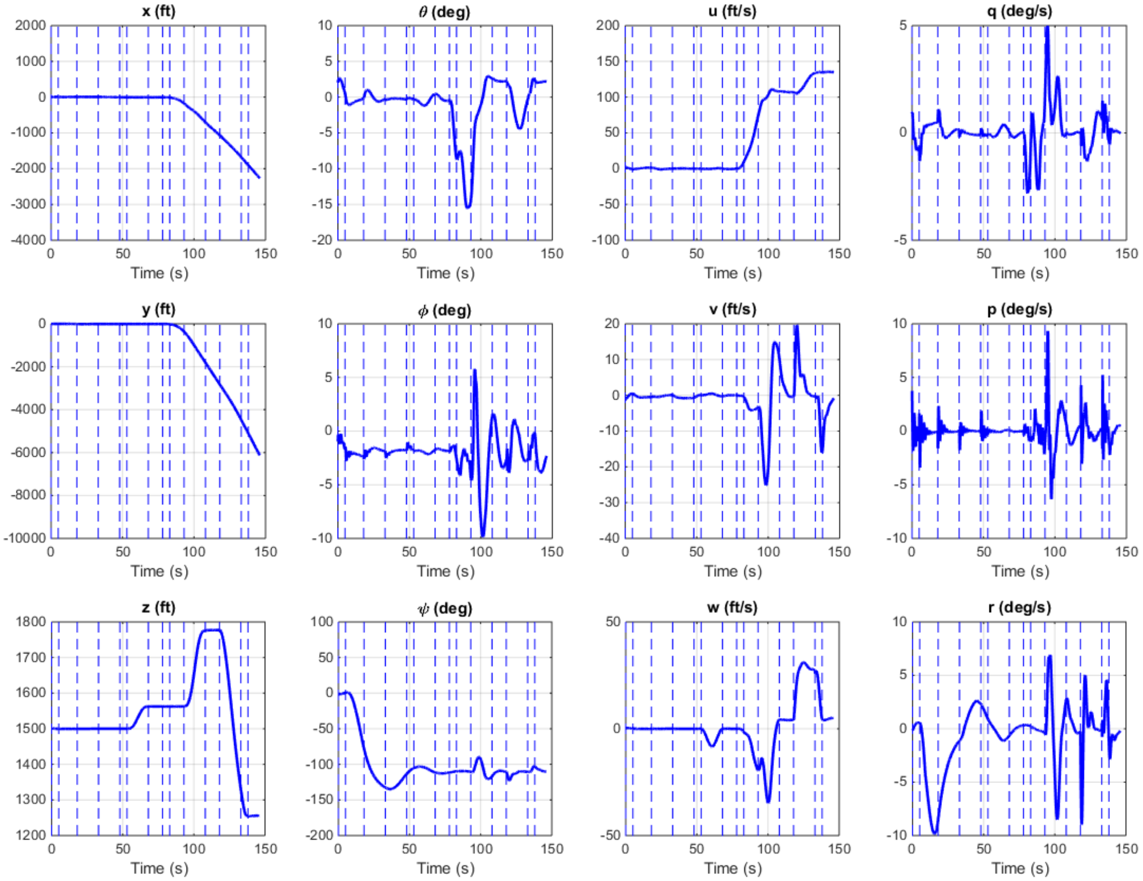

Figure 4.5: Time history of an example maneuver sequence with hover maneuvers and autorotation

In Figure 4.5, the helicopter remains level during hover maneuvers and turns in place with two types of controllers. The left turn is made with a commanded heading, and the right turn tracks a turn rate which fades smoothly in and out. During autorotation, the rotor rpm dips slightly as the rpm feedback loop begins to command a decrease in collective, and the rpm comes back up to 105% of the original (which is the desired rpm value during autorotation). Power is returned and the sink rate is smoothly arrested as forward flight resumes in autorotation recovery.

4.2 Conformance to ADS-33 Standard

The ADS-33 standard describes several “Mission Task Elements” or MTEs that are commonly used to evaluate the basic controllability of a helicopter [23]. Each MTE contains a description of a particular maneuver, and several requirements that may impose a range of allowable values for a particular state (or in some cases settling time requirements). ADS-33 enumerates requirements for two performance standards: “desired” and “adequate”. The requirements stated in the description of each maneuver are for the “desired” performance unless specified otherwise. The virtual pilot is able to meet most of the selected maneuvers with “desired” performance, while all satisfy the “adequate” standard. Desired performance is indicated on the relevant results with a green dotted line, and adequate performance is indicated with magenta.

The hover MTE requires the helicopter to be moving at least 6 knots, located south-west of the target hover point, and then initiate a hover stop to settle at that point. Both “adequate” and “desirable” tolerances are given for the altitude, heading, and for the final hover position, which must remain in the specified range. The tolerances are shown on the simulation plots below as pink and green dashed lines. In this case, altitude and heading remain well within the desired range, and hover position remains well within the adequate range and almost within the desired range.

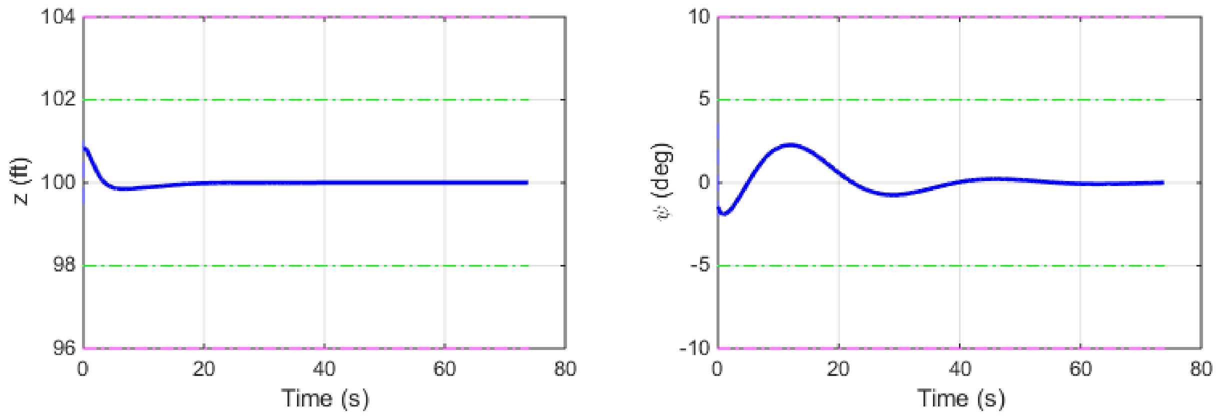

Figure 4.6: Hover Mission Task Element Altitude and Heading

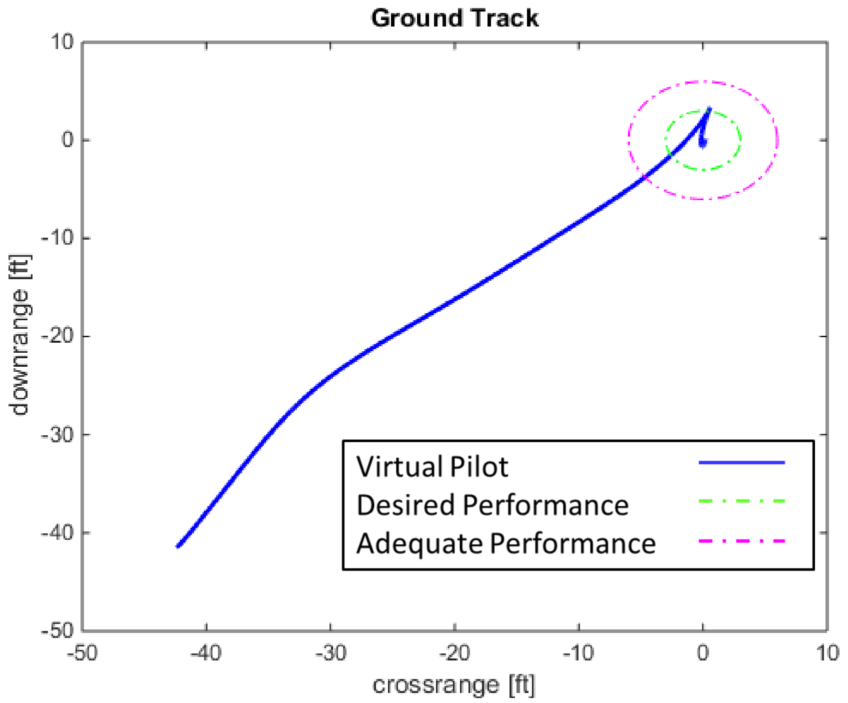

Figure 4.7: Hover Mission Task Element Ground Track

The purpose of a pullup/pushover MTE according to ADS-33 is to check the handling qualities of the helicopter at elevated and reduced load factors, check for undesirable coupling between pitch, roll, and yaw in forward flight, and to evaluate the ability of the helicopter to avoid obstacles during high-speed operations. The maneuver is defined as attaining a sustained positive load factor for 2 seconds in a symmetric pullup from level flight at 120 knots (200 ft/s), then transition with a symmetric pushover to a sustained load factor less than 1 within 2 seconds and maintain load factor less than 1 for 2 seconds. Angular deviations in roll and yaw must stay within +/- 10° from the initial level flight condition. A state time history for the virtual pilot performing this maneuver is shown in Figure 4.8 below, along with the load factor in Figure 4.9. Again, adequate and desired tolerances are shown in magenta and green dotted lines in the plots for bank angle and heading. Note that in this case the virtual pilot meets the desired performance requirements easily.

Table 4.5 Pullup/Pushover Maneuver Sequence

Time [s]	*Maneuver*
0	Forward Flight 0.75Vh
15	Symmetric Pullout,
17	Symmetric Pushover
22	Level Descent, 25 ft/s
25	Forward Flight, 0.75Vh

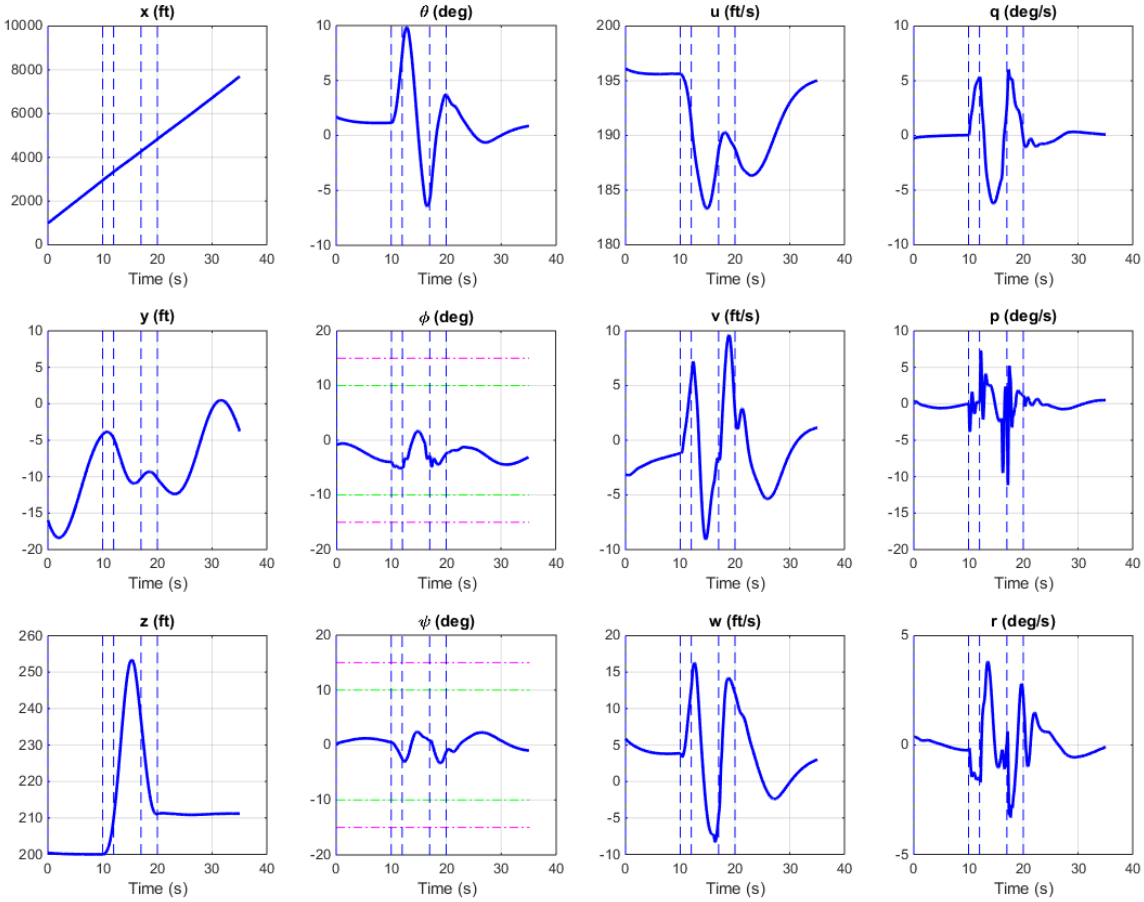

Figure 4.8: Time history of a symmetric pullup/pushover maneuver

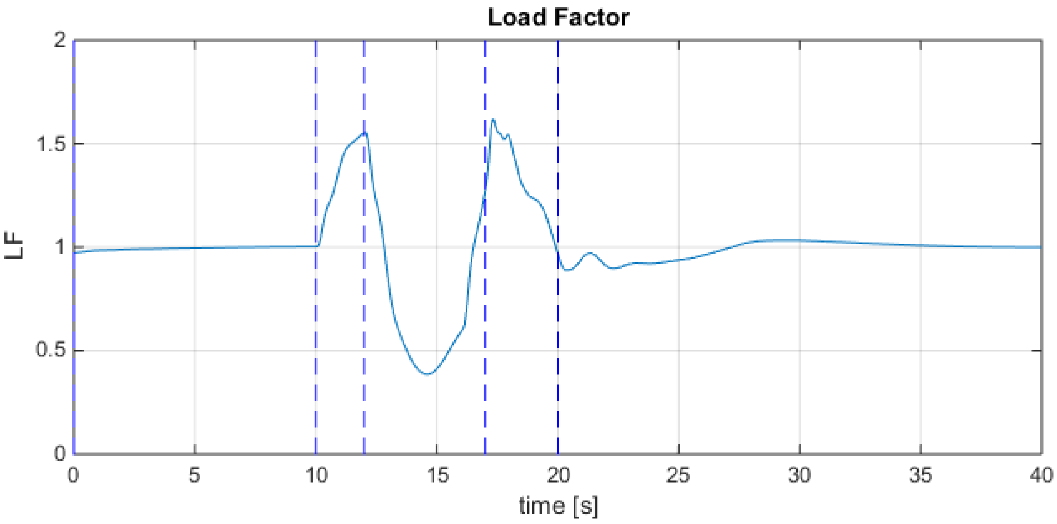

Figure 4.9: Load Factor during Symmetric Pullup/Pushover Maneuver

The execution of a landing maneuver requires finely tuned altitude response characteristics. ADS-33 prescribes a smooth continuous descent with no objectionable oscillations, touchdown within 10 seconds of passing below 10 ft of altitude, and heading maintained within +/- 5° of the reference. The virtual pilot divides landing into two phases with different prescribed forward and descending velocities which automatically transition at an altitude of 45 ft as shown in Table 4.6. Figure 4.10 shows the results of the virtual pilot performing the landing MTE, where it can be seen that the virtual pilot satisfies the MTE requirements as stated above.

Table 4.6 Maneuver Sequence for Normal Landing

Time [s]	*Maneuver*
0	Hover 0.1Vh
25	Normal Landing

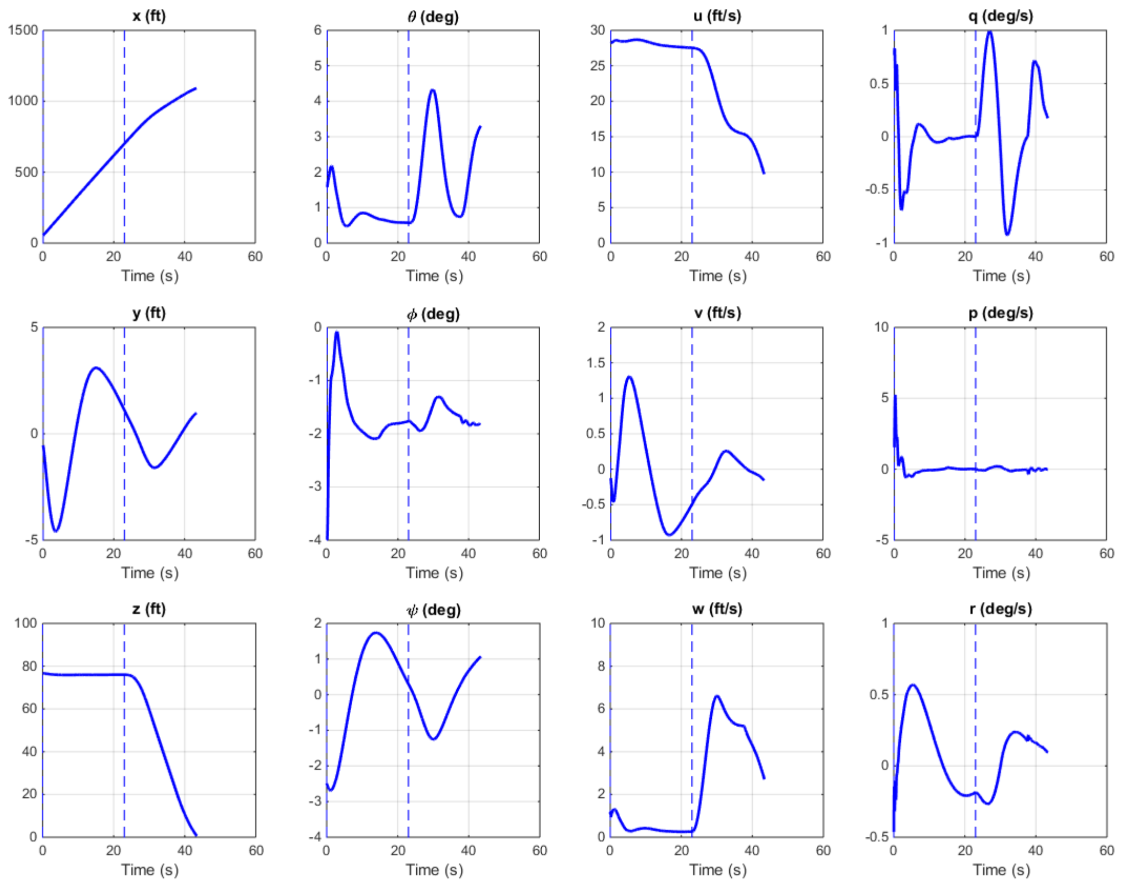

Figure 4.10: Time history of a landing maneuver

A vertical maneuver is described in ADS-33 as a rapid unmask and remask maneuver, with a short pause at the peak altitude to simulate an aiming task. Performing this rapid up and down maneuver displays the controller's ability to precisely start and stop a vertical rate and exposes any existing coupling between collective and the other control axes. The longitudinal and lateral position as well as the starting and final altitude should be held within 3 ft and the heading within 5 degrees. The whole maneuver should be completed within 13 seconds for desired performance, or 18 seconds for adequate performance. In this case, the virtual pilot met all desired performance standards except total completion time, which was adequate. The maneuver script was designed with simple climbing, descending, and hovering maneuvers with special attention paid to timing and rate parameters in order to meet the requirements. The state time history for this maneuver is shown in Figure 4.11.

Table 4.7 Maneuver Sequence for the Vertical Maneuver

Time [s]	*Maneuver*
0	Hover
5	Axial Climb
10.8	Hover
14	Axial Descent
19.9	Hover

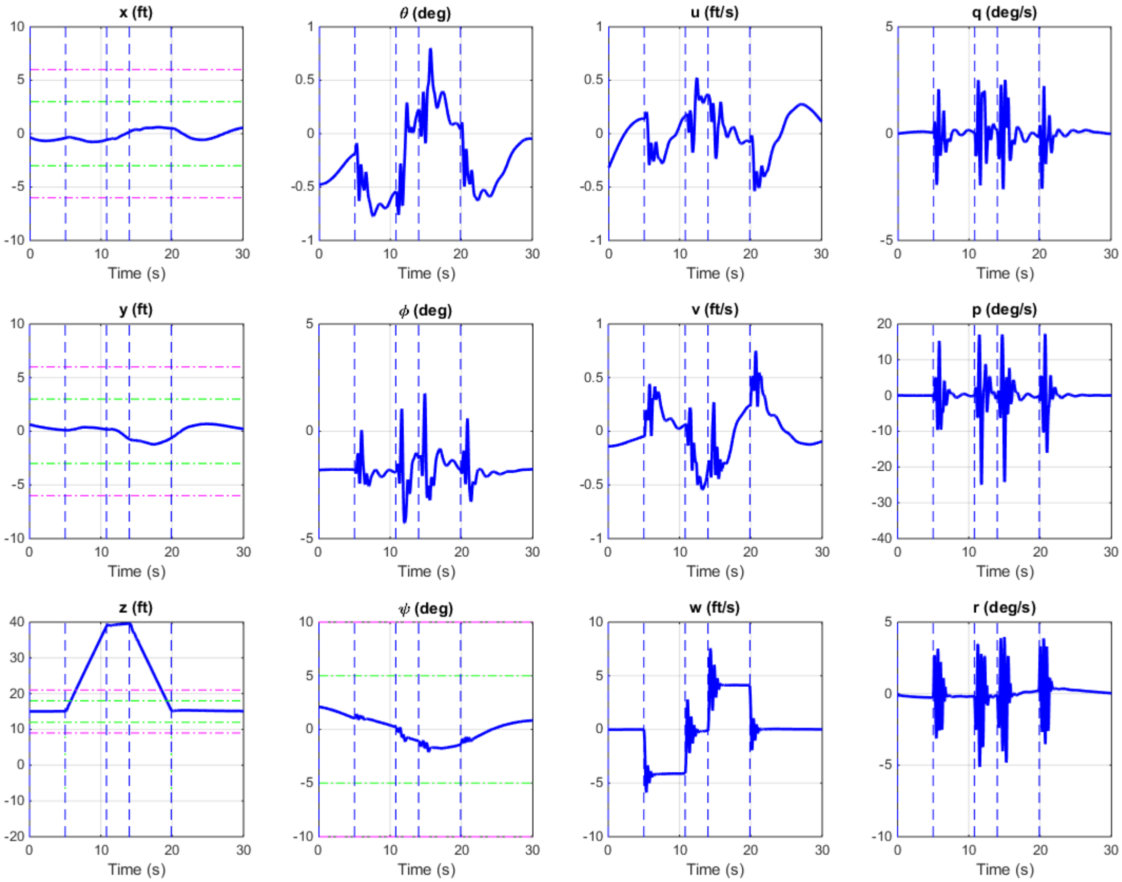

Figure 4.11: Time history for the Vertical Maneuver Mission Task Element

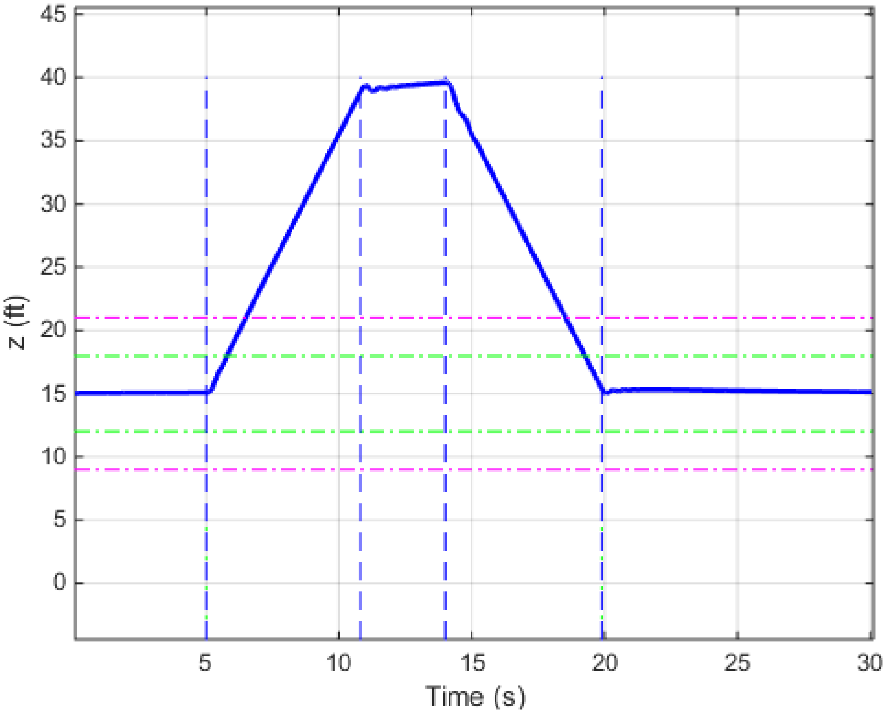

Figure 4.12: Altitude vs time for the Vertical Maneuver Mission Task Element

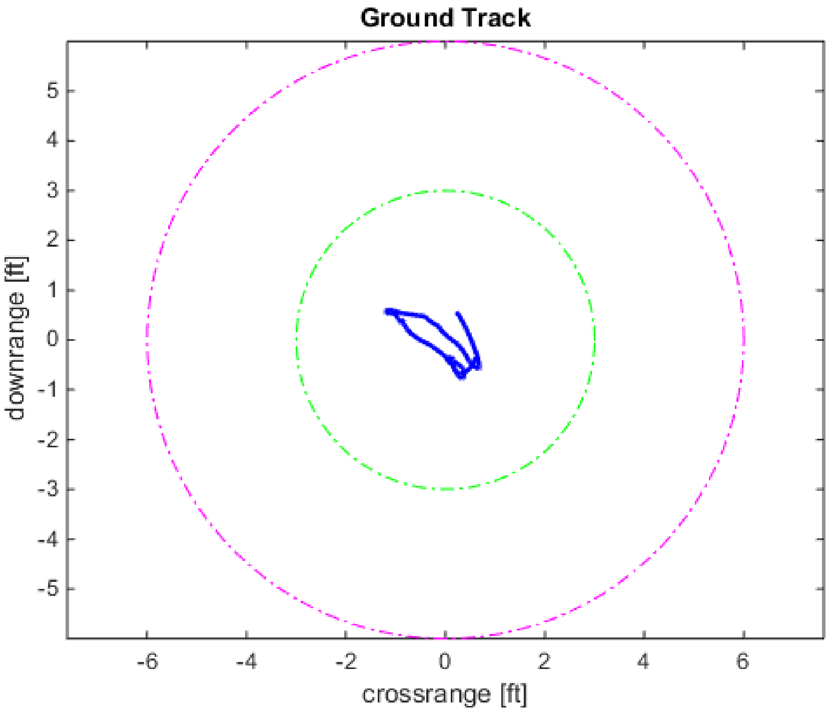

Figure 4.13: Ground Track for the Vertical Maneuver Mission Task Element

4.3 HUMS Tracking Maneuver Sequences

In this section, the virtual pilot is shown to be a producer of viable flight data for regime recognition purposes through comparison to real SH-60B HUMS flight data. The HUMS data used for this comparison is from a scripted flight test and thus is labeled with the intended flight regime at each timestep, so creating a maneuver sequence for the virtual pilot to match it was nearly trivial. There were large gaps in the HUMS dataset, so some guesses as to when maneuver transitions occurred had to be made. HUMS data is shown in the following plots as black, while the virtual pilot state is represented in blue. Some of the twelve rigid body states were not directly recorded in the HUMS dataset. Some of the missing data could be calculated from other information, but position information for HUMS was not recorded. Large bias error was evident in categories like the angular rate, while other categories like body frame vertical velocity needed heavy filtering. Overall, general comparisons can be made between the HUMS data and the virtual pilot simulation data.

The maneuvers sent to the virtual pilot in these instances were designed to follow the HUMS data as closely as possible in order to demonstrate the powerful simplicity of the basic maneuver scripting concept. The first maneuver sequence is enumerated in Table 4.8, and consists of a gradual climb, descent, and reduction in forward velocity. This simple maneuver sequence illustrates the ability of the virtual pilot to create continuous flight data using an incomplete reference set. This sort of high level “interpolation” demonstrates the quick generation of flight data by simply using the name of a flight regime. Results are depicted in Figure 4.14, where vertical lines indicate the time of the start of a new maneuver.

Table 4.8 First HUMS Tracking Maneuver Sequence

Time [s]	*Maneuver*
0	Forward Flight 0.7Vh
15	Level Climb
102	Level Descent
170	Forward Flight 0.4Vh

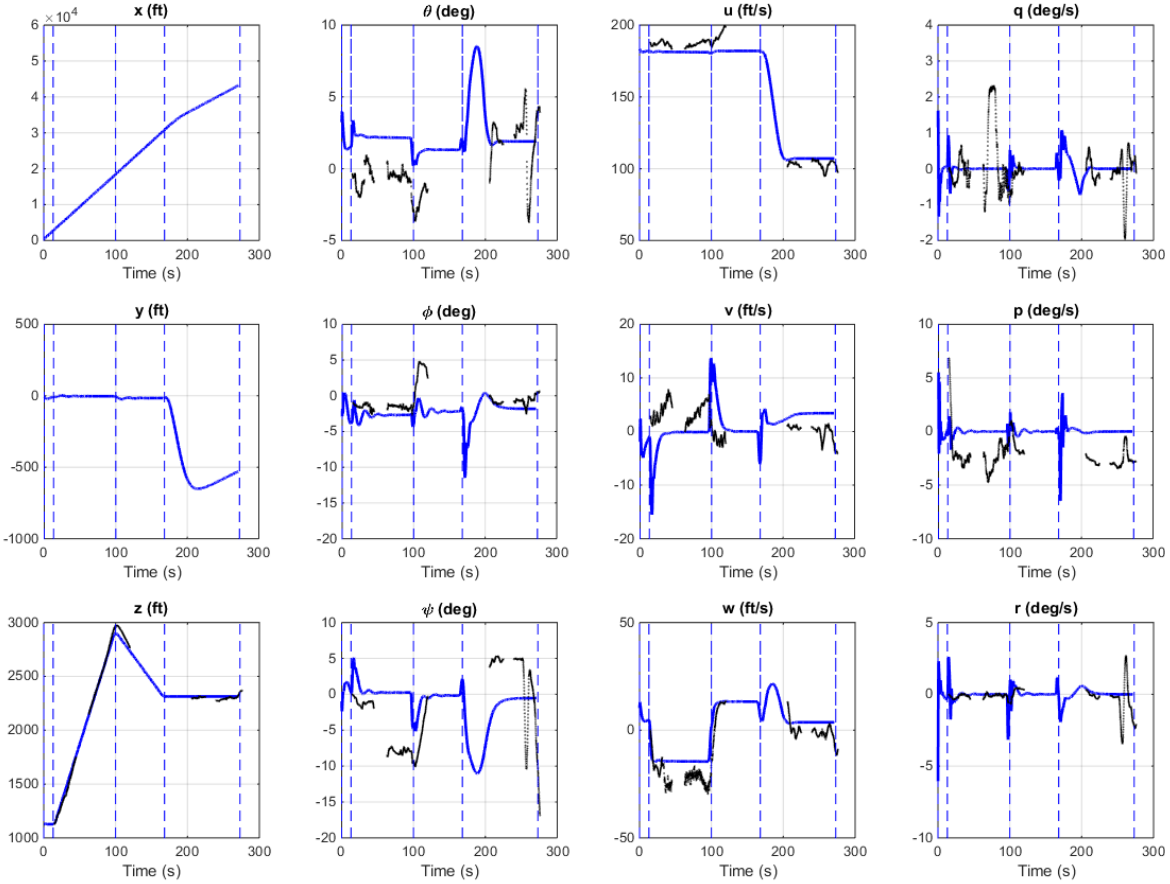

Figure 4.14: Time history of a maneuver sequence matching a typical HUMS data set with numerous gaps (Example 1)

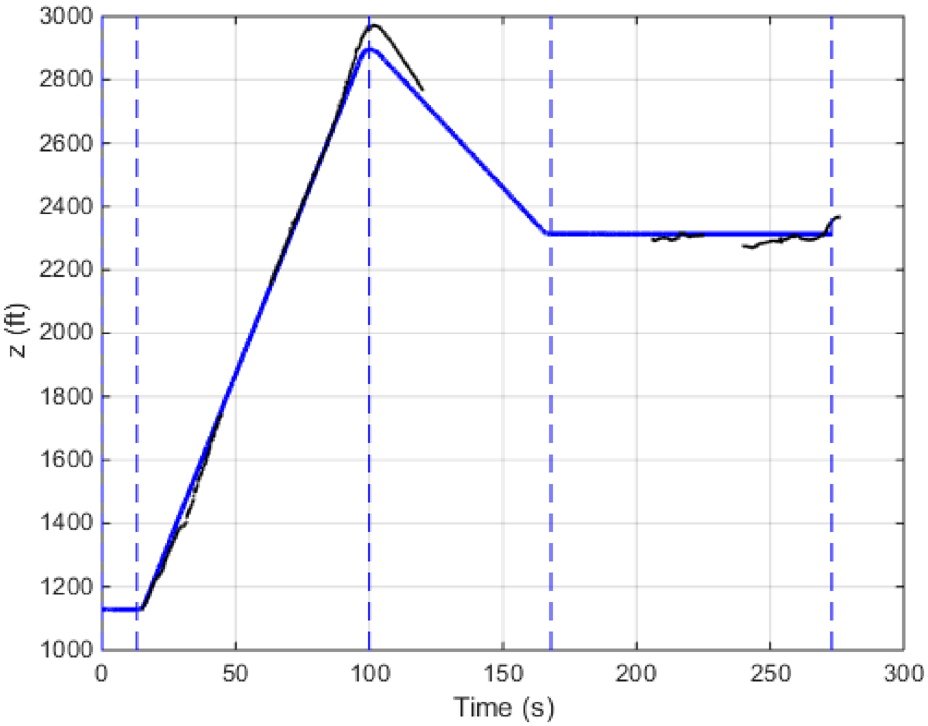

Figure 4.15: Altitude over time for the first HUMS matching maneuver sequence (Example 1)

The next HUMS data set for comparison consists of incrementally steeper banked turns followed by a climbing turn and descending turn. These maneuvers are executed in simulation with striking similarity to the real data set. Again, all that was given to the virtual pilot was the list of maneuvers and their respective numerical parameters, designed to approximately match the original maneuvers.

Table 4.9: Second HUMS tracking maneuver sequence

Time [s]	*Maneuver*
0	Forward Flight 0.5Vh
20	Right Banked Turn 35°
67	Right Banked Turn 41°
114	Left Banked Turn 15°
152	Left Banked Turn 28°

Table 4.9 (continued)	
190	Left Banked Turn 40°
250	Forward Flight 0.5Vh
270	Climbing Turn, +180°, 27 ft/s
410	Forward Flight 0.5Vh
446	Descending Turn, -360°, -42 ft/s
540	Forward Flight 0.5Vh

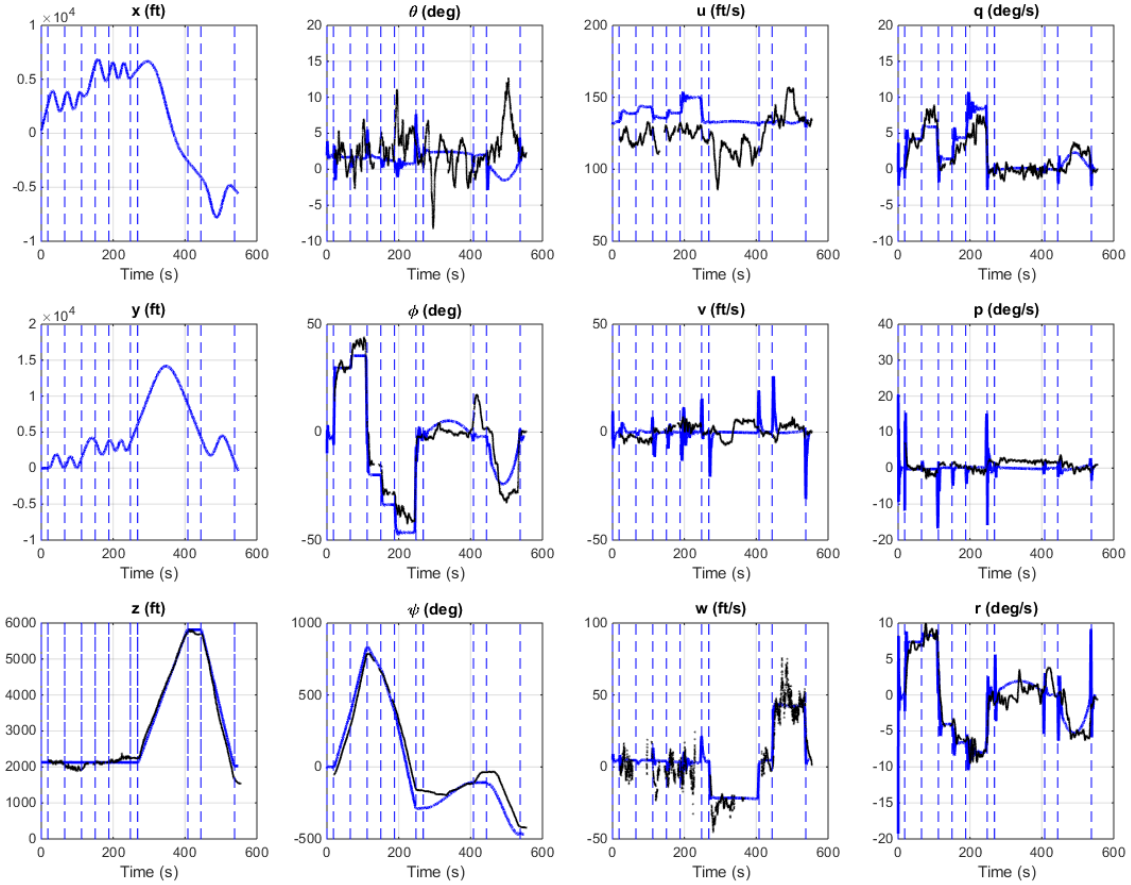

Figure 4.16: HUMS tracking sequence time history, Example 2.

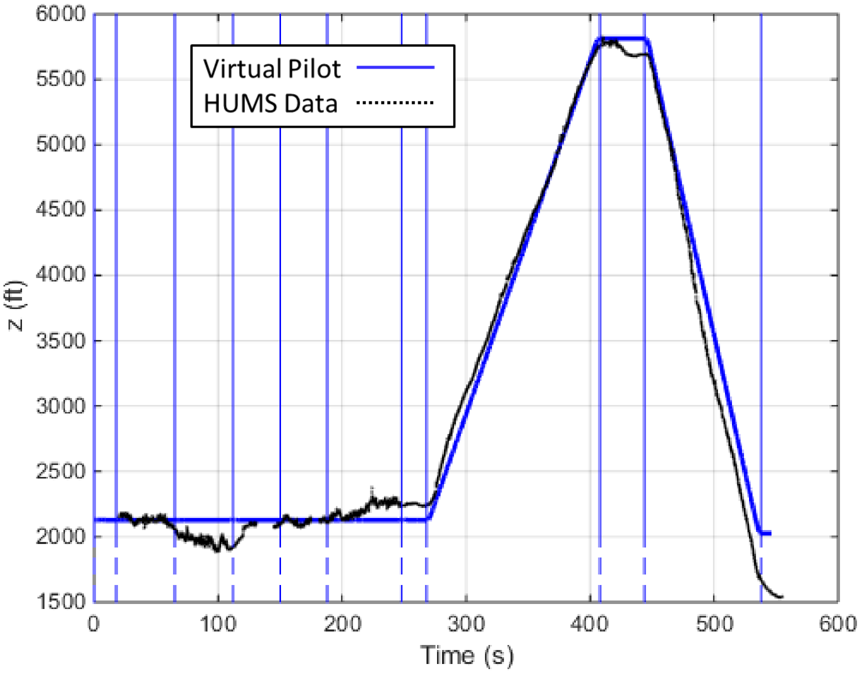

Figure 4.17: Altitude vs Time for HUMS Tracking Sequence Example 2

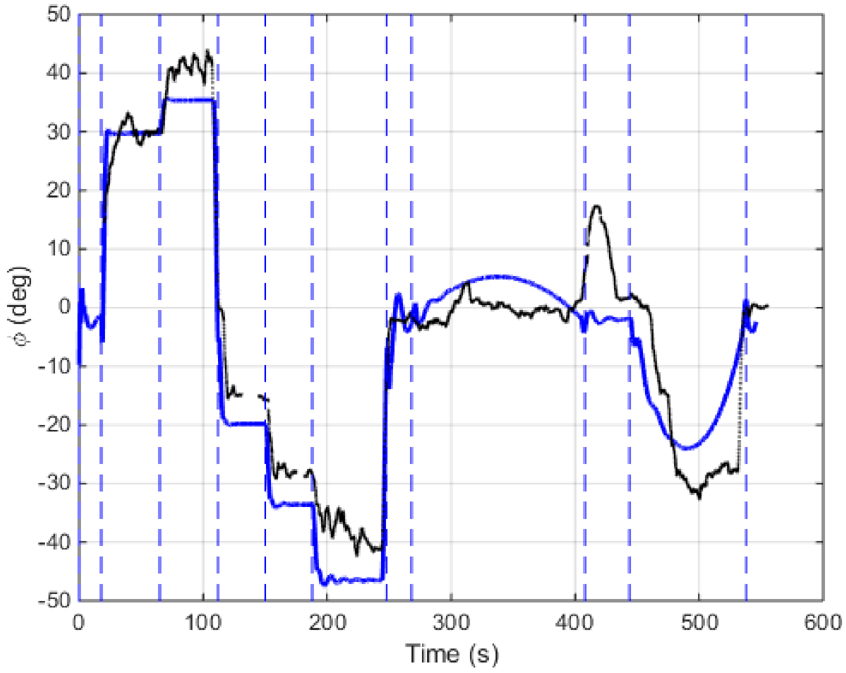

Figure 4.18: Bank Angle vs Time for HUMS Tracking Sequence Example 2

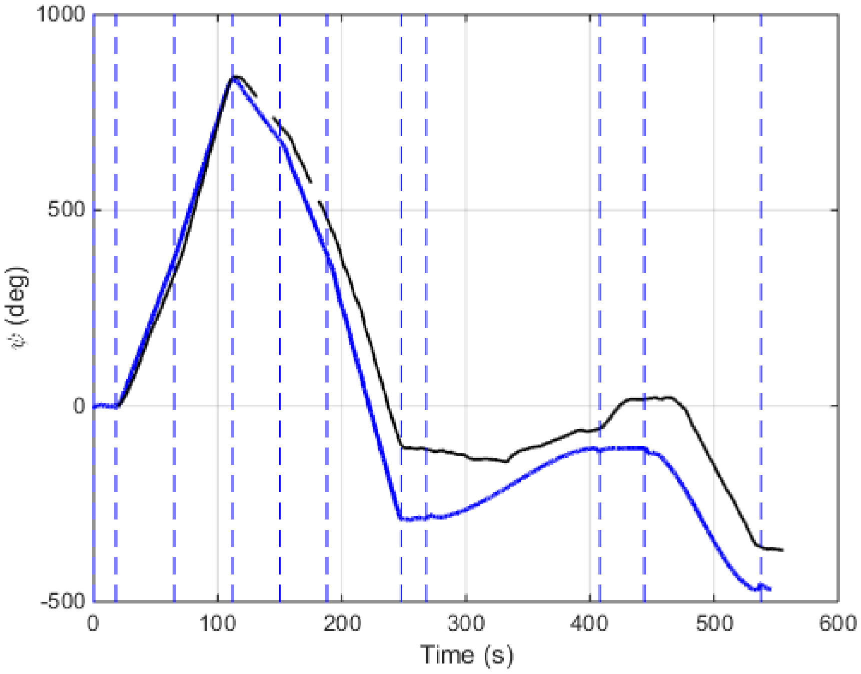

Figure 4.19: Heading vs Time for HUMS Tracking Sequence Example 2

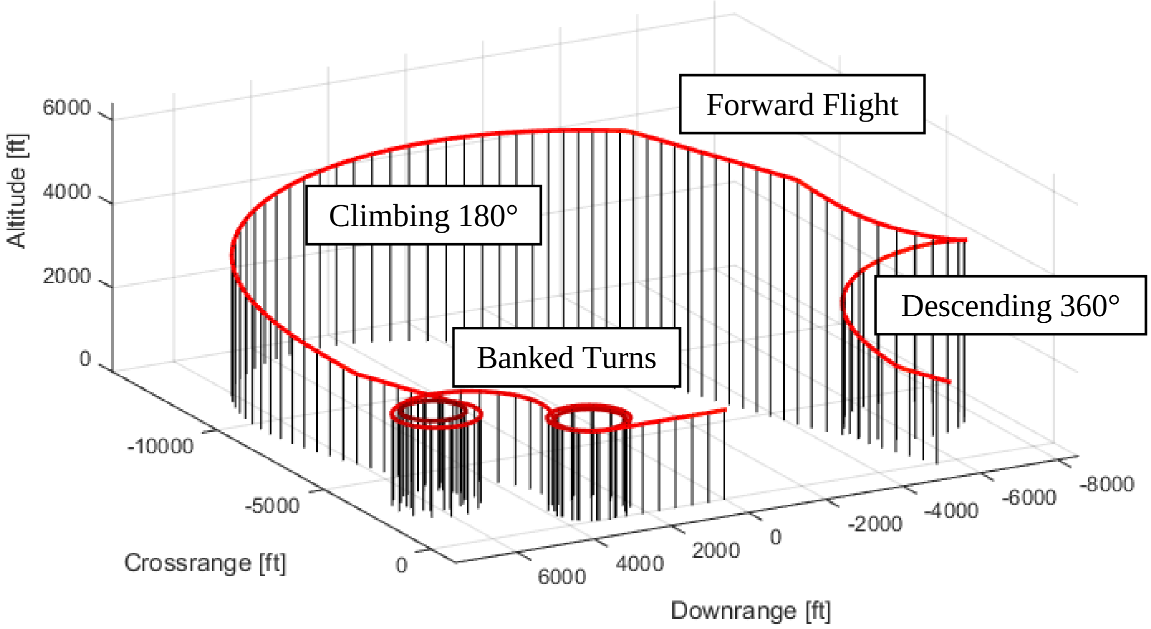

Figure 4.20: Trajectory of Virtually Piloted Maneuver Sequence, Example 2.

These two comparisons against actual SH-60B HUMS data clearly show that the virtual pilot is able to execute the same type of maneuvers found in the HUMS dataset with similar timing and results. Since the goal of the virtual pilot is to produce similar flight data, it is more important that the same steady state dynamics for a maneuver are achieved rather than exact position tracking. Even so, the integrated error over time in heading in Figure 4.19 is comparatively low. Figure 4.20 shows the trajectory of this maneuver sequence through space, which is a good visualization of the turns happening at the same time as the climbing and descending, with distinct but smooth transitions visible between each maneuver.

CHAPTER 5 : LIMITATIONS AND FUTURE WORK

As of right now, the virtual pilot offers a sparse, basic set of flight maneuvers, while regime recognition algorithms require training data for as many as 300 flight regimes. Additional maneuvers or combinations of maneuvers can easily be integrated into the existing virtual pilot structure to allow for the emulation of even more flight regimes. It is clear that while the virtual pilot meets many of the desired performance qualities mentioned in ADS-33, the transitions between flight control laws can still use improvement. PID gain tuning can be improved, although a more formal implementation of the bumpless transfer method for switching control laws would offer better progress towards seamless control law switching. Finally, additional studies of virtual pilot performance in normal and gusting wind conditions with sensor noise should be evaluated. However, the a turbulence-free simulation environment may provide more clear representations of pure flight regime behavior. Since real flight data will always contain these disturbance elements, performance analysis of regime recognition algorithms trained on both kinds of results should be evaluated. An extensive study of human factors and pilot behavior regarding maneuver execution would also be useful information in drafting control laws that attempt to emulate human pilots.

The next steps in this project should include the enhancements to the virtual pilot listed above, as well as integration of the virtual pilot within the automated regime recognition verification and validation pipeline.

CHAPTER 6 : CONCLUSIONS

A virtual pilot algorithm was presented for the purposes of synthetic HUMS data generation for use in V&V for regime recognition algorithms. It is clear that the virtual pilot performance compares favorably with that achieved by actual pilots with regard to maneuver consistency, timing, transitioning, and handling qualities specifications. The layered and switching feedback loop structure proved to be a viable control system architecture. The parameter tracking scheme creates enough flexibility to achieve various maneuvers with the same feedback loop structure. The result is a very powerful scripted flight planner for simulated rotorcraft. This quick method of generating flight data enables rapid iteration for the training and verification of regime recognition algorithms.

CHAPTER 7 REFERENCES

[1] G. Barndt and S. Moon, "Development of Fatigue Tracking Program for Navy Rotary Wing Aircraft," in *American Helicopter Society 50th Annual Forum*, Washington, D.C., May 1994.

[2] S. Moon and N. Phan, "Rotary Wing Aircraft Regime Recognition Algorithm Development & Validation," in *American Helicoper Society 64th Annual Forum*, Montreal, Canada, May 9-11, 2008.

[3] D. Hass, J. Walker and L. Kough, "Using Flight Data to Improve Operational Readiness in Naval Aviation," in *American Helicopter Society*, Montreal Canada, April 29 - May 1, 2008.

[4] R. Teal, J. Evernham, T. Larchuck, G. Miller, D. Marquith, F. White and D. Deibler, "Regime Recognition for MH-47E Structural Usage Monitoring," in *American Helicopter Society 53rd Annual Forum*, Virginia Beach, VA, April 29 - May 1, 1997..

[5] Y. Lu, R. Chris, T. Puckett, R. Teal and B. Thompson, "AH-64 Apache Longbow Structural Usage Monitoring System," in *American Helicopter Society 58th Annual Forum*, Montreal, Canada, June 11-13, 2002.

[6] G. Barndt, S. Sarkhar and G. Miller, "Maneuver Regime Recognition Development Validation for H-60 Structural Monitoring," in *American Helicopter Society 63rd Annual Forum*, Virgina Beach, VA, May 1-3, 2007.

[7] S. Wu, D. He and E. Bechhoefer, "A Practical Regime Recognition Approach for HUMS Applications," Virginia Beach, VA, May 1-3, 2007..

[8] D. He, S. Wu and E. Bechhoefer, "A Regime Recognition System for Helicopter Usage Monitoring," in *Aerospace Technologies Advancements*, Croatia, INTECH Publishing, 2010.

[9] V. Gavrilets, L. Martinos, B. Mettler and E. Feron, "Human-Inspired Control Logic for Automated Maneuvering of Minature Helicopter," *Journal of Guidance, Control, and Dynamics Vol. 27, No. 5,* September-October 2004.

[10] T. Schouwennars, B. Mettler and E. Feron, "Hybrid Architecture for Full-Envelope Autonomous Rotorcraft Guidance," in *American Helicopter Society 59th Annaul Forum,* Phoenix, Arizona, May 6-8, 2003.

[11] C. Dever, B. Mettler, E. Feron and F. Popovic, "Nonlinear Trajectory Generation for Autonmous Vehicles via Parameterized Maneuver Classes," in *AIAA Guidance Navigation and Control Conference*, Providence, RI, August 16-19, 2004.

[12] F. Kendoul, "Survey of Advances in Guidance, Navigation, and Control of Unmanned Rotorcraft Systems Vol. 29," *Journal of Field Robotics,* pp. 315-378, 2012.

[13] J. Y. Hung, W. Gao and J. C. Hung, "Variable Structure Control: A Survey," *IEEE Transactions on Industrial Electronics,* vol. 40, no. 1, February 1993.

[14] M. H. Raibert and J. J. Craig, "Hybrid Position/Force Control of Manipulators," *Journal of Dynamic Systems, Measurement and control,* vol. 102, pp. 126-133, June 1981.

[15] R. Hanus, M. Kinnaert and J. L. Henrotte, "Conditioning Technique, a General Anti-windup and Bumpless Transfer Method," *Automatica,* vol. 24, no. No. 6, pp. pp 729-739, 1987.

[16] P. D. Talbot, B. E. Tinling, W. A. Decker and R. T. N. Chen, "A Mathematical Model of Single Main Rotor Helicopter for Piloted Simulation," NASA Tech. Rep. TM-84281, Ames Research Center, Moffett Field, CA, 1982.

[17] R. T. N. Chen, "A Simplified Rotor System Mathematical Model for Piloted Flight Dynamics Simulation," NASA Tech. Rep. TM-78575, Ames Research Center, Moffett Field, CA, May, 1979.

[18] R. T. N. Chen, "Effects of primary rotor parameters on flapping dynamics," NASA Tech. Rep. TM-1431, Ames Research Center, Moffett Field, CA, January 1980.

[19] Z. Sunberg, "A Real Time Expert Control System for Helicopter Autorotation," Master's Thesis, Texas A&M University, College Station, Texas, 2013.

[20] D. A. Peters and N. HaQuang, "Dynamic Inflow for Practical Applications," *Journal of the American Helicopter Society*, pp. 64-68, October, 1998.

[21] D. M. Pitt and D. A. Peters, "Theoretical prediction of dynamic-inflow derivatives," *Vetica*, vol. 6, pp. 21-34, 1981.

[22] D. A. Peters and G. H. Gaonkar, "Effectiveness of Current Dynamic-Inflow Models in Hover and Forward Flight," *Journal of the American Helicopter Society*, pp. 47-57, April, 1986.

[23] "Handling qualities requirements for military rotorcraft," in *Aeronautical Design Standard Performance Specification ADS-33E-PRF*, 2000.

[24] B. Taylor, "Experimental Investigation of Helicopter Weight and Mass Center Estimation," Master's Thesis, Texas A&M University, College Station, Texas, 2013.