VELOCITY AND ALTITUDE CONTROL OF AN ORNITHOPTER MICRO AERIAL VEHICLE

by

Katherine Sarah Shigeoka

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Department of Electrical and Computer Engineering

The University of Utah

May 2007

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

SUPERVISORY COMMITTEE APPROVAL

of a thesis submitted by

Katherine Sarah Shigeoka

This thesis has been read by each member of the following supervisory committee and by majority vote has been found to be satisfactory.

Chair: Mark A. Minor

Marc Bodson

Stacy J. Morris Bamberg

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

FINAL READING APPROVAL

To the Graduate Council of the University of Utah:

I have read the thesis of Katherine Sarah Shigeoka in its final form and have found that (1) its format, citations, and bibliographic style are consistent and acceptable; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the Supervisory Committee and is ready for submission to The Graduate School.

Date

Mark A. Minor
Chair, Supervisory Committee

Approved for the Major Department

Marc Bodson
Chair/Dean

Approved for the Graduate Council

David S. Chapman
Dean of The Graduate School

ABSTRACT

This thesis is concerned with analyzing stability and controllability of nonlinear mechanical systems subject to forcing inputs with multiple sinusoidal components. Analysis is applied, in particular, to that of an ornithopter micro aerial vehicle, an aircraft that utilizes flapping motion to fly. The dynamic complexity of flapping wing flight generates the desire for semiautonomous flight control to reduce the load on the pilot, as well as to gain stability, maneuverability, and efficiency. Micro Aerial/Air Vehicles size and weight constraints limit the available computational power and generate the need for simple linear feedback control. However, the existence of multiple sinusoidal terms in the forcing input created by the wings, along with the nonlinearities in the state-space model, make control challenging.

Controller design is studied via analysis, modeling, and simulation and is based on research which indicates stability of nonlinear systems subject to vibrational and oscillatory forcing. First, acceleration data are analyzed to characterize the forcing input as a function of flapping frequency. Then, linear-motion data are used to identify a one-dimensional ornithopter system transfer function. A feedback path filter is designed based on the frequency spectrum of the forcing input and for limited delay. Velocity control is achieved with a proportional-integral controller based on the identified transfer function and an approximation of the mean velocity found from filtering. Cases for constant horizontal velocity and constant hover are considered. Lastly, a state-space model of the ornithopter is created by considering horizontal, vertical, and angular equations of motion and is used to represent the ornithopter in hover and in steady horizontal flight. For these cases, feedback gains are found to control the velocity and altitude and integral action is included for robust tracking and disturbance rejection.

Though control is obtained for the instances considered in this thesis, it is found that a better system model is required to successfully proceed with further altitude and velocity control. Expressly, a model of lift, which in this case is generated mainly from the flapping of the wings, is required as a function of angle of attack. This work serves as a basis for semiautonomous flight control of an ornithopter MAV.

To the memory of my mom, Ann O'Neill Shigeoka, M.D.

To my dad and family.

CONTENTS

CHAPTERS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my advisor, Prof. Mark A. Minor, who was instrumental in getting the ornithopter project started. His active support, enthusiasm, and guidance were of great help in completing this thesis.

I would also like to thank my committee members: Prof. Marc Bodson for his valuable suggestions and use of his lab and Prof. Stacy J. Morris Bamberg for just being her sunny and supportive self.

Much appreciation and gratitude also go to Xiuyan Guo for his help in working through new and advanced theory, as well as with the thesis track and Eric A. Johnson for his contributions and cooperation on the ornithopter project.

CHAPTER 1

INTRODUCTION

1.1 Application and Motivation

An ornithopter is an aircraft that uses flapping wing motion to fly. This type of flight offers potential advantages over fixed-wing flight, such as maneuverability [1], at slow speeds (1-40m/s). Natural ornithopters range in size from small flying insects to large birds, as noted in [1], and flap their wings from about 5 to 200Hz, as noted in [2]. The Defense Advanced Research Projects Agency (DARPA) largely motivates Micro Aerial/Air Vehicle development in military application for reconnaissance missions in confined spaces or under dangerous circumstances. The discreetness of a flapping wing MAV adds appeal for these types of covert operations. Employment in civilian search and rescue missions under dangerous or questionable circumstances such as fire or earthquake also fuels MAV development. Inspiration to mimic insect flight, which combines oscillating and rotating wings, is detailed in [2] and points to the adeptness that can be obtained through this type of flight. Another important insight from [2] is the focus on simple control loops in place of the complex control loops prevalent in modern aircraft. This focus spawns from the combination of limited on-board processing power with the need for many sensors and high-frequency update rates.

The ornithopter under study was developed as a Mechanical Engineering Senior Design Project.[1] Basic open-loop flight control is possible through a radio frequency remote controller with joystick. Closed-loop control is desired to make the ornithopter easier to maneuver and control for the average user. Because MAVs are restricted both in size and weight, an embedded system used to control the ornithopter must be low-power and lightweight. This limits the processing power available for control calculations and generates the need for simple control loops. Initial work to create an embedded control

[1] Details on the ornithopter design and the Ornithopter Design Group can be found in Appendix A.

system to meet these requirements was done as an Electrical Engineering Senior Project.[2] Before choosing a final micro-controller, however, processor speed and other requirements (input-output capability, floating point unit requirement, etc.) must be specified. This requires the design of a controller and inertial measurement unit.

1.2 Challenges

There are many challenges involved in the study of flapping wing flight. At low speeds, there is a lack of significant lift generated from oncoming airflow. Hence, lift and thrust must be generated predominantly from flapping. Besides aerodynamic challenges, there are also those involving control issues. As noted in the previous section, limited on-board processing power creates the desire for a low complexity controller. However, flapping-wing flight introduces oscillations to the system that appear as large sinusoidal disturbances. As the frequency of flapping changes, both the mean of the forcing input from the wings and the amplitude, frequency, and phase of individual sinusoidal components of the forcing input from the wings vary nonlinearly. This makes tracking hard and control challenging. Use of traditional disturbance rejection techniques to cancel these sinusoidal disturbances is not suitable as they are components of the forcing input exciting the system.

1.3 Related Ornithopter Research

Scientific exploration into the aerodynamics of flapping-wing flight is limited, but has recently been on the rise. [1], [3], [4], and [5] study mechanical design of an ornithopter MAV. Trade-offs between ornithopter weight and wing length, mass and speed, and wing designs are identified in [1]. Micro-electro-mechanical systems wing technology for a battery-powered ornithopter was also studied and created in [1]. A study of the unsteady aerodynamics of a flapping wing was done in [3] for a flapping wing MAV in hover. A flapping wing MAV was built and studied in [4] in order to maximize flap efficiency. A wing's force and flow structures were studied in [5] for a simplified flapping motion similar to that of an insect.

After developing a competitive inch-size flapping wing MAV, averaging is used to show controllability of the Micromechanical Flying Insect being developed at UC Berkeley. This is important in that it offers insight into the controllability of the ornithopter being

[2] Work done by Katherine Shigeoka

studied in this thesis. In designing the MFI, work was first done to investigate rotational wing movement [6]. Flight force measurements and simulations were then generated in [7]. Wing generated forcing and controllability issues were assessed in [8]. With this work, a force map characterization was completed in [9]. This force map characterization correlates the sinusoidal voltage amplitude input and flapping and rotation angle phase differences of the wings to the generated thrust and lift. Stable hovering motion via control of wing movement is simulated for a dynamic model of the MFI thorax with approximate wing movements in [10]. As noted, one drawback to these simulations is the approximation of the nonlinear periodic forcing input generated by the wings. Attitude control along the z-axis is then shown using ocelli (e.g. light sensors) and halteres (e.g. rotational velocity detectors) in [11]. Most importantly, a detailed controllability analysis of flapping wing flight is presented in [12]. Here, high frequency control theory is used to apply averaging to the controllability of a MAV with wings each limited to a single degree of freedom and passive rotation. Controllability is then extended to the case including the wing thorax dynamics and a foundation for use of simple linear feedback laws is made. The simplified ornithopter MAV model in [12] is quite similar to the one studied here and the research offers a basis on which to proceed with simple feedback control.

1.4 Related and Applicable Theory

The nonlinearity of the ornithopter system, as well as forcing input to the system, create difficulty for stability and control of the ornithopter. The forcing input created by a wing's flapping motion is also vibrational and semiperiodic. However, stability and control of nonlinear systems via vibrational, oscillatory, or periodic inputs has been intensively researched.

Stability of nonlinear systems though vibrational control was heavily researched by R. E. Bellman, J. Bentsman, and S. M. Meerkov in the 1980s as an alternative where classical control methods fail. Criteria for stability via nonlinear multiplicative vibrations, quasi-periodic forcing, and vector additive vibrations is presented in [13]. Controllability is then shown in [14]. Gurvits extends the use of oscillatory inputs to nonholonomic motion planning via averaging in [15].

Open-loop, high-frequency oscillatory forcing of mechanical control systems was studied by J. Baillieul and S. Weibel beginning in the mid-1990s. In this research, the Method of Averaging and the Method of Multiple Scales offer solutions in the form of an "averaged potential." The use of open-loop, high frequency oscillatory forcing to bring a nonlinear mechanical system to rest at fixed points of the averaged system, which are not equilibria of the nonautonomous or forced system, is researched in [16]. In the 1998 publication of [17], J. Ballieul and S. Weibel summarized the developments of oscillatory control of mechanical systems up to that point and looked at the nonlinear effects of oscillatory forcing. Ballieul goes on to summarize the response of under-actuated systems to oscillatory forcing via the averaged potential of the forcing input in [18].

Periodic forcing and oscillatory control are studied in [19],[20],[21],[22], and [23] though averaging theory. Application of averaging theory to prove stability and controllability require a relatively robust dynamic model. Though one is not currently available for this project, these theories offer insight into controllability of the ornithopter. A link between averaging and controllability theory is found in [19] where mechanical systems subject to high amplitude high frequency inputs are studied. An under-actuated spacecraft with small-amplitude, periodically time-varying forcing is studied in [20], open-loop attitude control via periodic forcing is studied in [21], and control via large amplitude high frequency oscillatory inputs is studied in [22]. An autonomous undersea vehicle with oscillating fins is studied in [23] where a Fourier expansion is used to characterize the force and moment signals created by the fins. This body of work, along with that done in [12] on the MFI, support the use of averaging to show controllability of the ornithopter and for stabilization through basic control laws.

1.5 Goals and Contributions

With evidence showing controllability of systems subject to vibrational, periodic, and oscillatory inputs through averaging analyses and evidence of superior low-speed flight capability with flapping flight, work is now done to control the ornithopter. The objective of this research is to:

- Analyze motion data of an ornithopter MAV in order to develop a system transfer function

- Examine the frequency spectrum of the forcing input created by an ornithopter's flapping wings

- Characterize and model the forcing input created by the wings

- Generate a time-varying forcing input in order to demonstrate closed-loop control through simulations
- Develop a feedback path filter to filter out turbulence in the motion data based on frequency spectrum analysis
- Demonstrate controllability of an ornithopter MAV via simple feedback control
- Demonstrate controllability of nonlinear systems subject to semi-periodic, forcing inputs with multiple large amplitude sinusoidal terms based on that demonstrated with the ornithopter MAV
- Develop velocity and altitude control in order to gain a basis for semiautonomous control of the ornithopter.

1.6 Research Approach

The design process of the ornithopter is very fluid. This is because the ornithopter has been redesigned multiple times since the commencement of this project and is currently out of commission. Given these design iterations and the difficulty involved in producing an analytical model of the flapping forces, an exact model of the ornithopter is unavailable. In this thesis, simple experimental motion data are analyzed and used to obtain a rudimentary model of the ornithopter sufficient for simulation based controller evaluation. Control of the ornithopter is first implemented for 2 one-dimensional models. The one-dimensional models consist of velocity control for the ornithopter constrained to 1) forward horizontal movement along a line and 2) vertical movement for hover. In the previous case, the ornithopter is faced in a horizontal stance, and in the later case, the ornithopter is faced in a vertical stance. In both of these cases the ornithopter is subject to a forcing input from the wings only. Velocity and altitude control are then implemented for 2 two-dimensional longitudinal models of the ornithopter in 1) hover and 2) horizontal flight. In these cases, forcing input comes from the wings and tail. Research being done in parallel with this thesis includes the development of an IMU by Eric A. Johnson.[3]

[3] Paper to be submitted to IEEE: “An Adaptive Filter for Rejecting Noise and Tracking Bias in Inertial Sensors.”

CHAPTER 2

ONE-DIMENSIONAL CONTROL

In this chapter one-dimensional models of the ornithopter are derived and validated according to motion sensor data.

2.1 Data Acquisition

Data was collected from the ornithopter while traveling down a linear bearing rail by Eric A. Johnson[1] and Charles Fisher. The ornithopter was strapped to a slider on the rail, restricting the ornithopter to movement in the forward horizontal direction. Acceleration and position sensor information were collected at different throttles. Here, throttle refers to the position of the joystick on the remote control device. This is approximately proportional to speed of the brushless DC motor sourcing the wings.[2] Acceleration data was measured by an accelerometer strapped to the body of the ornithopter while submillimeter position data were collected with a VICON Motion Tracking System.[3] The bias level of the accelerometer data was adjusted until it agreed with the second derivative of the position data. In this way, the sensor drift of the accelerometer is accounted for. Approximate velocity data were found from the integral of the tuned acceleration data. Figures 2.1, 2.2, and 2.3 show the acceleration data filtered with a low pass filter with cutoff frequency of 40Hz, velocity, and position data found from the accelerometer.

[1] The data used here were collected and analyzed to aid in development of an IMU. Paper to be submitted to IEEE: “An Adaptive Filter for Rejecting Noise and Tracking Bias in Inertial Sensors,” by Eric A. Johnson.

[2] Voltage applied to the brushless motor is a more suitable and accurate measurement of changes in the forcing input. This information is not available; hence approximate throttle value is used instead.

[3] The VICON motion tracking system was developed by and belongs to The Motion Analysis Core Facility (http://mocap.uofupt.com/index.html). Personal help obtained from John Droge.

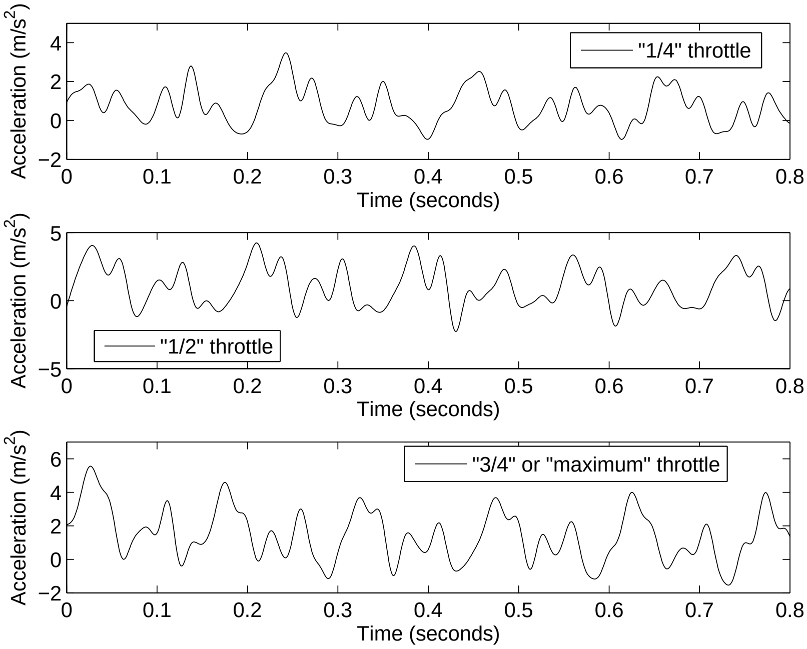


Figure 2.1. Filtered Acceleration Data

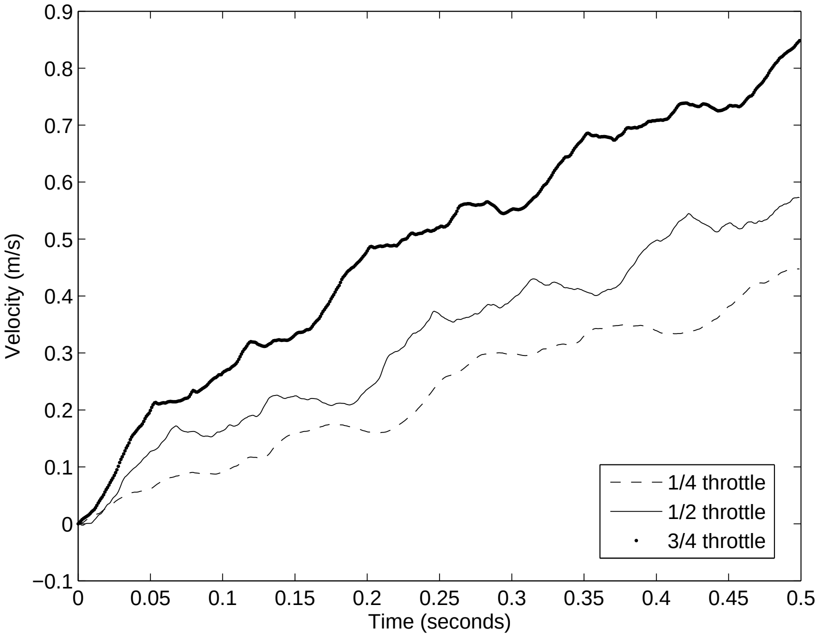


Figure 2.2. Linear Velocity Data

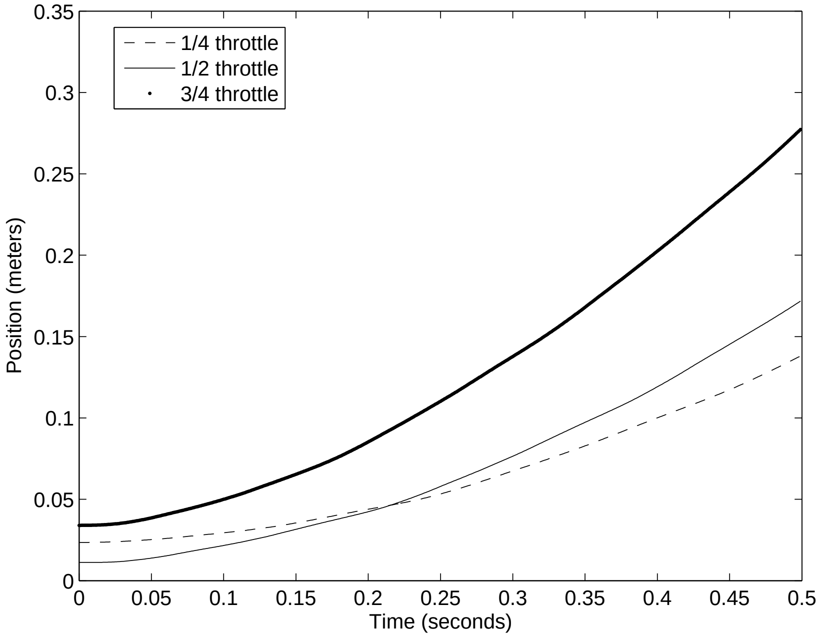


Figure 2.3. Position Data

2.2 Data Analysis and Modeling

This section describes data analysis and modeling of the ornithopter. First, a basic model of the ornithopter is derived using equations of motion. Then, the net force driving the ornithopter is characterized based on the acceleration data. Work is then done to find a specific transfer function of the ornithopter based on the basic model and motion data. The model is then validated through simulation.

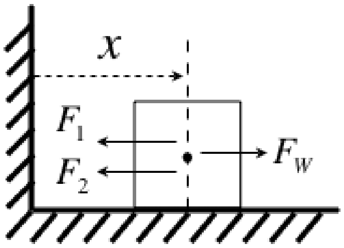
2.2.1 A Basic Model

In the case of the ornithoper constrained to one-dimensional horizontal motion, a simplified model can be expressed as a mass with damping subject to a forcing input. The following equation of motion describes the time-domain model of the ornithopter, which is pictured in Figure 2.4:

$$f(t) = m\ddot{x} + B\dot{x} = m\dot{v} + Bv.$$

From this equation, a transfer function from forcing input to velocity output can be obtained through a Laplace Transform as:

$$F(s) = msV(s) + BV(s) = (ms + B)V(s).$$



$$F_1 = b_X \dot{x}$$

$$F_2 = m\ddot{x}$$

$$\sum F_X = F_W - F_1 - F_2$$

Figure 2.4. One-Dimensional Free Body Diagram of Ornithopter: Horizontal Movement

Giving the transfer function:

$$P(s) = \frac{V(s)}{F(s)} = \frac{1}{ms + B} = \frac{1/m}{s + B/m}.$$

We can also obtain the transfer function from acceleration to velocity as:

$$\frac{V(s)}{F(s)} = \frac{V(s)}{mA(s)} = P(s)$$

$$\frac{V(s)}{A(s)} = mP(s) = \frac{1}{s + B/m}. \tag{2.1}$$

Thus, a first-order transfer function from acceleration input to velocity output can be determined to match the gain and rise-time of the measured velocities. However, the acceleration must first be characterized in order to simulate it as an input to the transfer function.

2.2.2 Frequency Analysis and Simulation of Acceleration

The frequency spectrum of the filtered acceleration data, shown in Figure 2.5, reveals some important things about the system under study. The frequency of flapping (or forcing) is determined to be the smallest non-DC frequency component of the acceleration. Table 2.1 summarizes the estimated flapping frequency found at each throttle. These values change slightly, by about $\pm 0.1 Hz$, depending on the period of the data examined.

Each acceleration signal can be summarized through a Fourier series expansion, as:

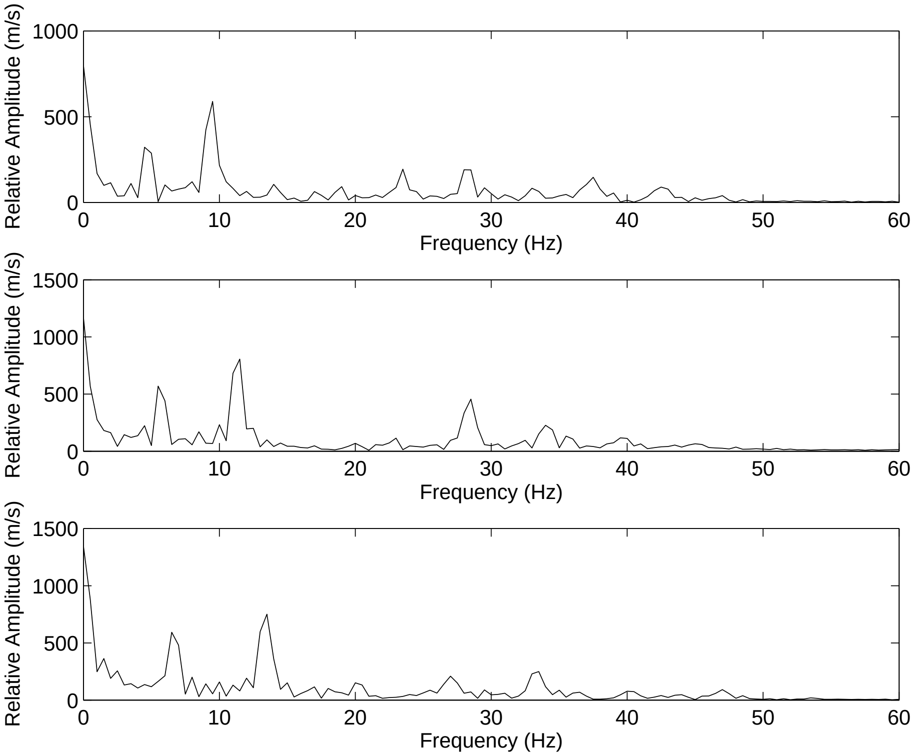


Figure 2.5. Frequency Spectrum of Filtered Acceleration Data

Table 2.1. Throttle vs. Estimated Frequency of Flapping

Throttle	Frequency of Flapping
1/4	4.6Hz
1/2	5.6Hz
3/4	6.6Hz

$$a(t) = \sum_{i=0}^{n} \{M_i \cos(\omega_i t + \phi_i)\}$$

Where n is the number of components (or harmonics) used to represent the signal. In this way, the acceleration can be simulated using the frequency, amplitude, and phase of the most prominent components. For instance, Table 2.2 summarizes this information for the five most prominent components of the acceleration at 1/2 throttle.

Each acceleration signal is cut to one representative period in order to allow for simulation over an unspecified amount of time and to avoid windowing problems. Windowing problems occur when frequency components exist which correspond to the period of the window of time in which the data is taken. Because data is available for three different throttles, a third-order polynomial can be fit to the three Fourier series expansions to approximate acceleration at other throttles. Also, to avoid the ambiguity of approximate "throttle," the estimated frequency of flapping is used instead as the input. Figure 2.6 compares the cut acceleration (single period representation of the actual data repeated over time) next to that simulated via MATLAB at a flapping frequency of 5.6Hz. Here, n, the number of components, is equal to 10. From this figure, one can see that the simulation is relatively accurate, but that the simulated signal does not have quite the same peak-to-peak or DC magnitude of the sensor data.

To allow simulation of the acceleration in a SIMULINK model, a SIMULINK s-function is made. This allows for future use in real-time control simulations. Gains are added to adjust the DC and peak-to-peak magnitude for a better approximation. Though one could theoretically duplicate the acceleration exactly by increasing the number of frequency components used to approximate the signal, this requires a large increase in computational power. Adding simple fixed gains to the simulation achieves the same goal of more accurately modeling the acceleration signal, but does not significantly increase the required computational power. Figure 2.7 shows the SIMULINK model with s-function and gains. The s-function code which makes explicit calls to generate the acceleration

Table 2.2. Summary of Amplitude, Frequency, and Phase of the Five Most Prominent Components in Acceleration at 1/2 Throttle

Component	1	2	3	4	5
Relative Amplitude (m/s)	594.2	230.9	511.1	25.0	25.0
Frequency (hertz)	0	5.6	11.3	16.9	22.5
Phase (radians)	0	1.7	2.9	2.4	-2.0

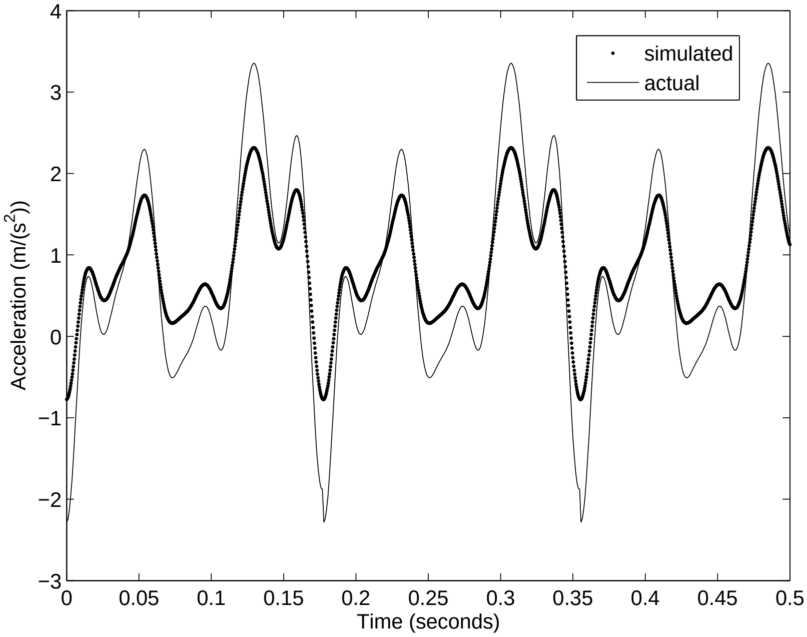


Figure 2.6. Comparison of MATLAB Simulated and Cut Acceleration (Flapping Frequency = 5.6 Hz)

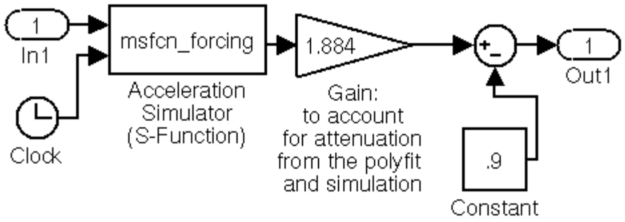


Figure 2.7. SIMULINK Acceleration Simulator S-Function with Gains

through use of polynomial coefficients can be found in Appendix C. Figure 2.8 shows the cut acceleration along with that simulated in SIMULINK, again, at a flapping frequency of 5.6Hz.[4] Here, one can see a significant increase in the accuracy of the simulated acceleration due to the addition of gain blocks.

In order to better visualize the acceleration as a function of flapping frequency, it is simulated over flapping frequencies which can reasonably be estimated from the

[4] Plots of simulations at other frequencies can be found in Appendix B.

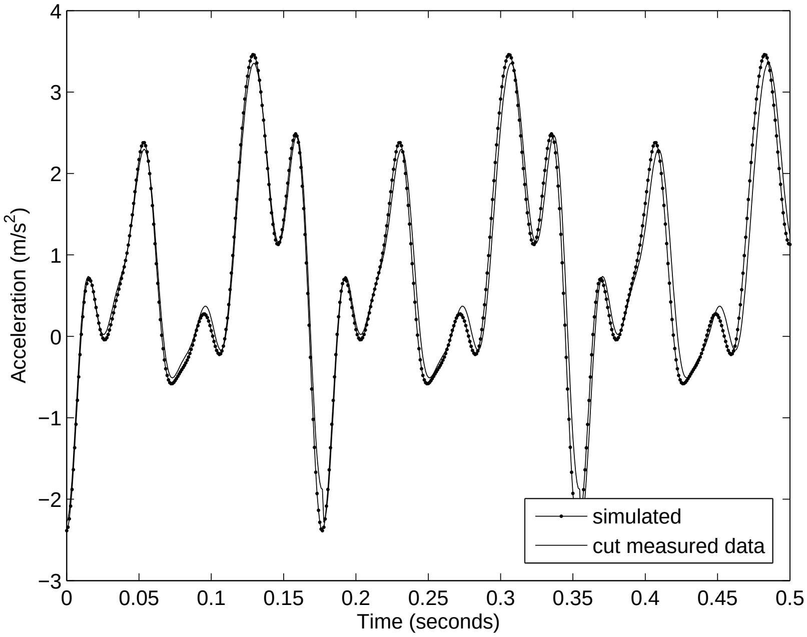


Figure 2.8. Comparison of SIMULINK Simulated and Cut Acceleration (Flapping Frequency = 5.63 Hz)

polynomial fit. Figure 2.9 shows the mean acceleration simulated from 0 to 10Hz. The sensor data at 6.6Hz (or 3/4 throttle) is estimated to be close to the maximum achievable acceleration. Therefore the upper bound of the frequency of flapping or input to the acceleration will be limited to 7Hz in subsequent simulations. This range is further restricted, for the same reason, to a minimum bound of 4 Hz and since the simulated acceleration loses fidelity below this point. In future work on the ornithopter, more data should be collected to better approximate the acceleration at other flapping frequencies. The RMS (root mean squared) acceleration is also found and plotted for given frequency of flapping in Figure 2.10.

In addition to limiting the input range, the acceleration is modified to account for slowly time-varying inputs. This allows one to change the frequency of flapping during simulation without incongruities in the acceleration. This can be done by incorporating changes in frequency into the phase, since phase is the time integral of frequency. In this case one component of the acceleration signal is expressed as:

$$S_{n+1} = A_{n+1} \cos(\phi_{n+1})$$

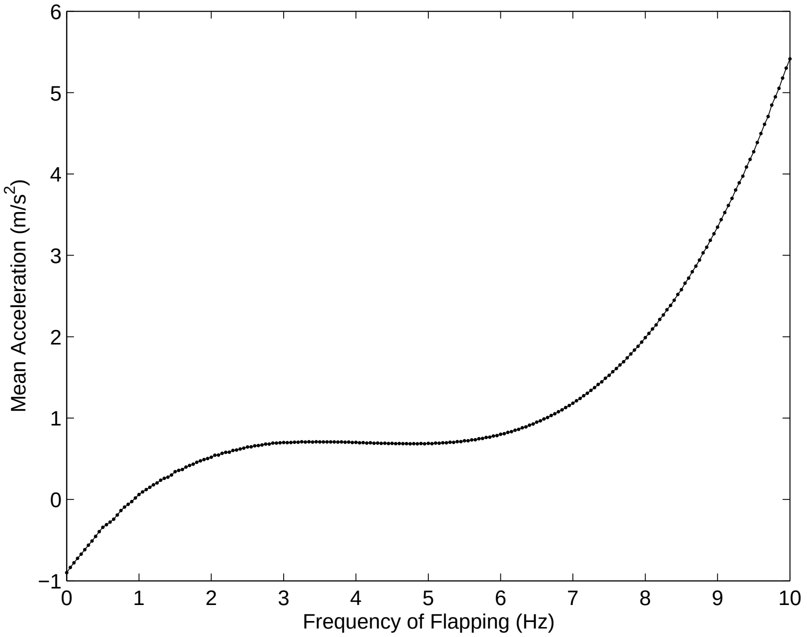


Figure 2.9. Frequency of Flapping vs. Simulated Mean Acceleration

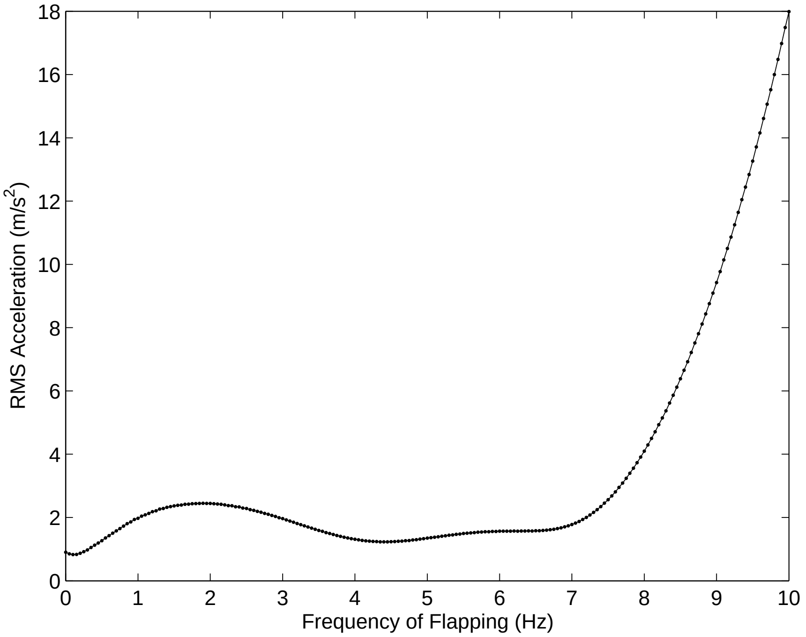


Figure 2.10. Frequency of Flapping vs. Simulated RMS Acceleration

for

$$\phi_{n+1} = \phi_n + \omega_n dt$$

where S is the time-varying signal, A is the amplitude of the signal, ω is the frequency (in radians), ϕ is the phase, and n is the sample number. Figure 2.11 shows the time-varying acceleration generated for input flapping frequencies from 4Hz to 7Hz. The s-function code updated to account for time-varying inputs can be found in Appendix C. With the ability to simulate the acceleration, a transfer function from acceleration input to velocity output can now be found.

2.2.3 Modeling of Ornithopter Transfer Function and Model Validation

A first-order transfer function of the ornithopter, similar to the one found from the equations of motion, can be found by measuring the gain and rise-time of the velocity. However, the velocity information must first be extrapolated to find its steady-state value since the measured data is restricted to about one second and does not reach a steady-state value. The restriction in data come from the fact that the linear bearing rail used to confine the ornithopter to one-dimensional horizontal movement is only about

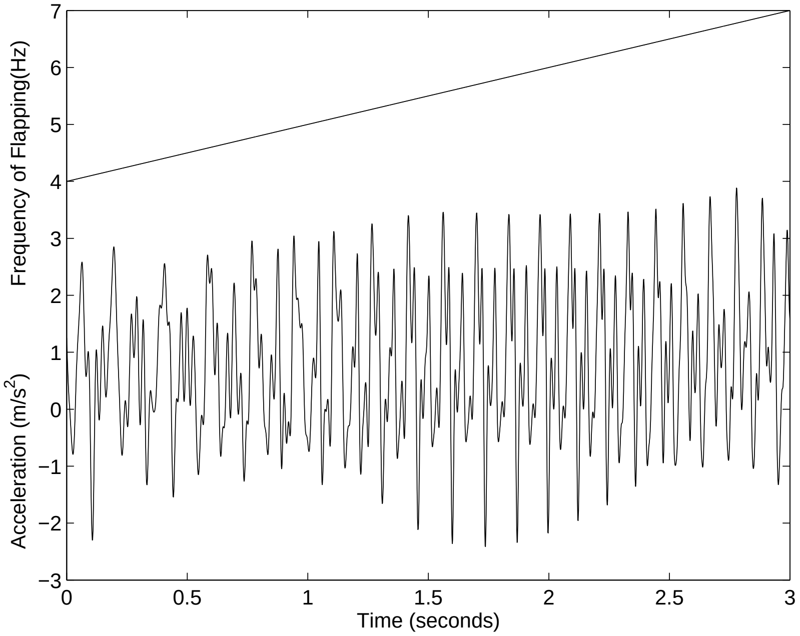


Figure 2.11. Time Varying Input and Simulated Acceleration

0.75 meters long. Figure 2.12 shows an extrapolation of the velocity for a frequency of flapping of 5.63Hz. The exponential equations expressing the extrapolated velocities are $v(t) = 1(1 - e^{-1.15t})$, $v(t) = 1.5(1 - e^{-1.2t})$, and $v(t) = 1.6(1 - e^{-1.5t})$ for flapping frequencies of 4.6Hz, 5.6Hz, and 6.6Hz, respectively. The average exponential velocity fit, $v(t) = 1.33(1 - e^{-1.28t})$, can be used as an approximation of the system. Thus, for an input, A(s), output, V(s), and 1st-order transfer function, G(s), we have:

$$\frac{V(s)}{A(s)} = G(s)$$

$$V(s) = A(s)G(s) = A(s)\frac{K}{(s+a)}.$$

If we assume a step input (seen as the DC component of the acceleration), we now have:

$$V(s) = \frac{A_0}{s}\frac{K}{(s+a)} = \frac{A_0K/a}{s} - \frac{A_0K/a}{(s+a)}.$$

Using the Final Value Theorem, the steady-state velocity, v_{SS}, can be found as:

$$v_{SS} = \lim_{t\to\infty} v(t) = \lim_{s\to 0} sV(s) = \frac{A_0K}{a}.$$

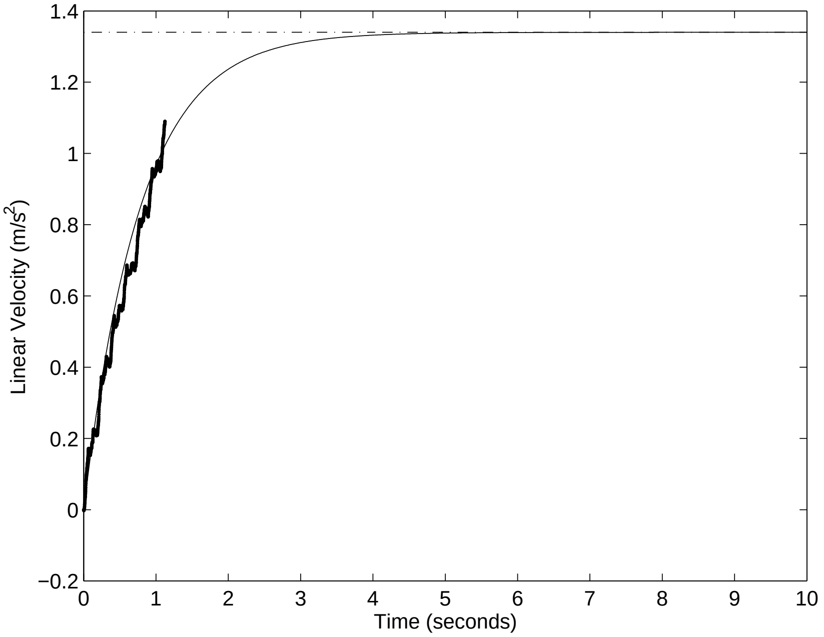


Figure 2.12. Extrapolation of Linear Velocity Data to Steady State (Flapping Frequency = 5.63Hz)

This gives the time domain response:

$$v(t) = v_{SS}(1 - e^{-at}) = (\frac{A_0 K}{a})(1 - e^{-at}).$$

From the average exponential velocity fit, we can assume

$$\frac{A_0 K}{a} = 1.33$$

and

$$a = 1.28.$$

Therefore

$$A_0 K = 1.71,$$

which gives

$$G(s) = \frac{K}{s+a} = \frac{1.71/A_0}{s+1.28}.$$

From Figure 2.9, an estimated average acceleration is $A_0 = 0.73$. The transfer function now becomes

$$G(s) = \frac{2.34}{s+1.28}. \tag{2.2}$$

From this transfer function, we can also find b_x, the coefficient of viscous friction (or damping). From equation 2.1 and 2.2 we find that

$$B/m = a$$

and

$$B = am = 1.28(1/s) * 0.096(kg) = 0.12(Ns/m).$$

Acceleration, velocity, and position are simulated in order to validate the estimated transfer function model. The open-loop model used for simulation and created in SIMULINK is shown in Figure 2.13. The “f_flap” block specifies the flapping frequency input in Hertz and the “Acceleration Function Subsystem” is the acceleration simulator described in Section 2.2.2 and shown in Figure 2.7. The “accel”, “vel”, and “pos” blocks output the simulated data to the workspace in MATLAB. Figure 2.14 compares the simulated and actual open-loop velocity at a flapping frequency of 5.6Hz. The simulated velocity is extended to 10 seconds. The flapping frequencies and corresponding simulated linear velocities are summarized at the sensor data points in Table 2.3. In order to demonstrate the validity of these linear velocities, they are compared to the final slope

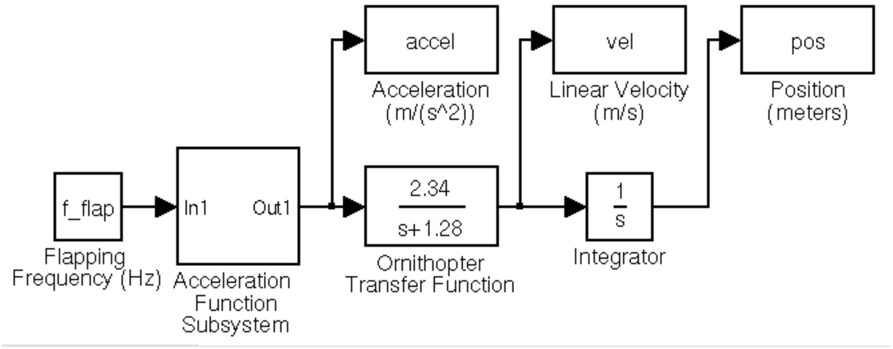


Figure 2.13. One-Dimensional Horizontal SIMULINK Model of Ornithopter

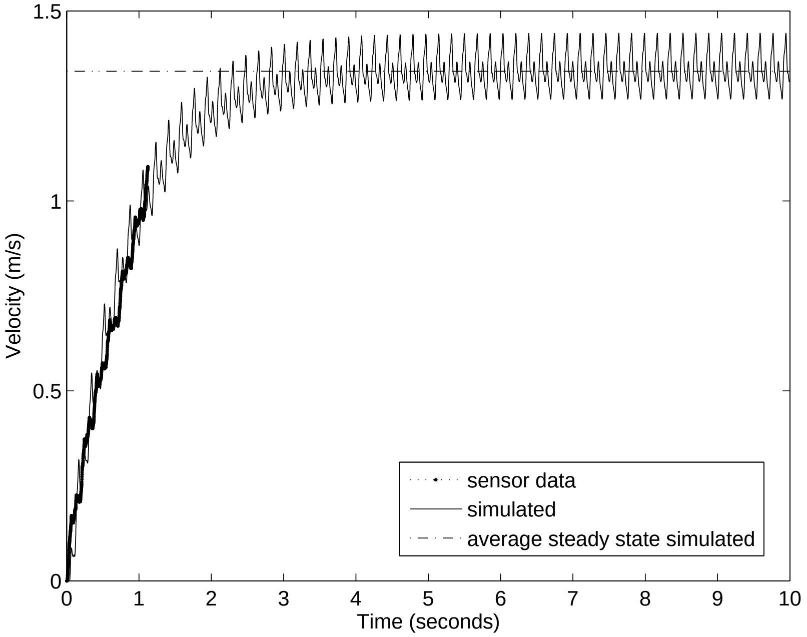


Figure 2.14. Comparison of Simulated and Actual Linear Velocity (Frequency of Flapping = 5.63Hz)

Table 2.3. Frequency of Flapping vs. Simulated and Estimated Steady-State Linear Velocity

Frequency of Flapping	Estimated Steady-State Linear Velocity		
	Simulated	Based on Position Data	Extrapolated
4.6 Hz	1.26 m/s	.93 -1.17 m/s	1 m/s
5.6 Hz	1.34 m/s	1.33 -1.67 m/s	1.5 m/s
6.6 Hz	1.80 m/s	1.57 -1.97 m/s	1.6 m/s

of the corresponding position data. If we assume that the velocity at these points are at 60% to 75% of their steady-state values, we find that the simulated linear velocities are reasonable approximations. The steady-state velocities estimated from the position data and the extrapolated velocities are also summarized in Table 2.3. The velocity output is integrated to find position and a comparison with the position data is shown in Figure 2.15. Figure 2.16 shows the estimated linear velocity from 4Hz to 7Hz.

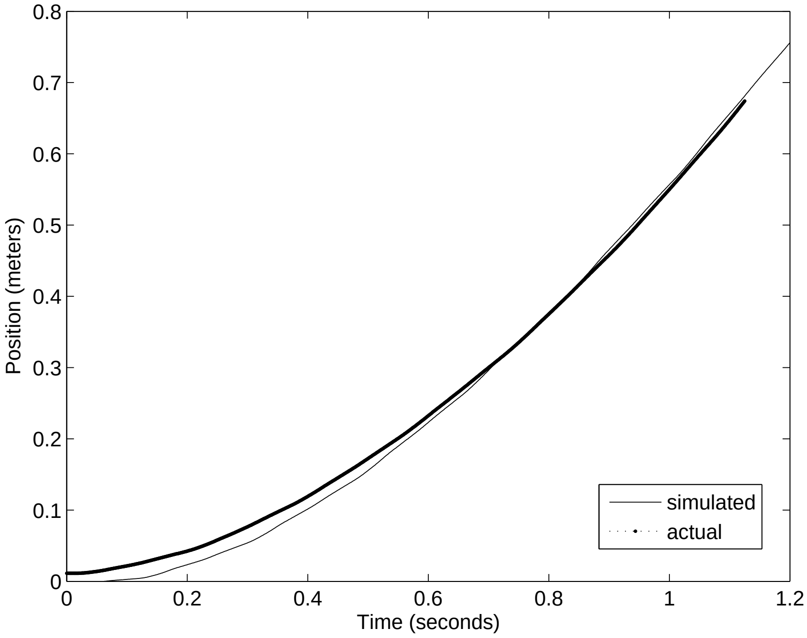


Figure 2.15. Comparison of Simulated and Actual Position (Frequency of Flapping = 5.63Hz)

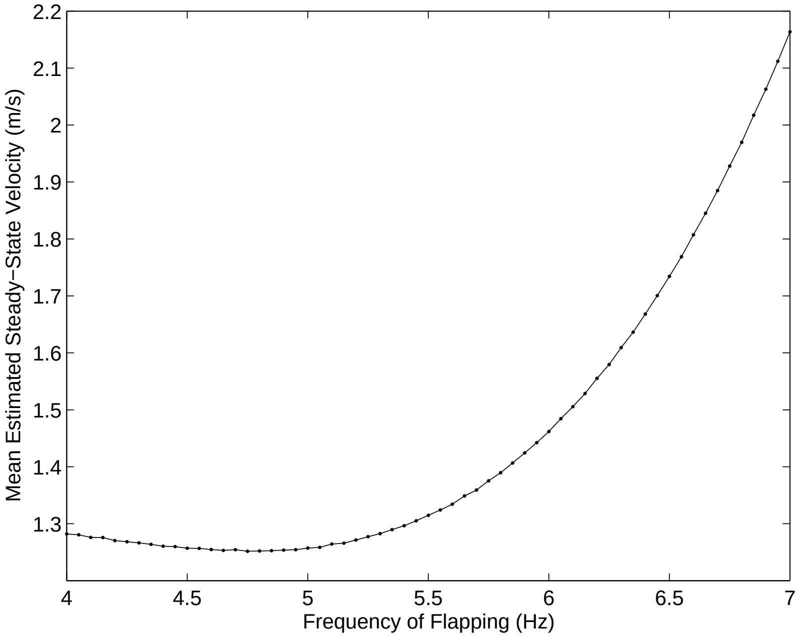


Figure 2.16. Frequency of Flapping vs. Simulated Linear Velocity

2.2.4 Model Assumptions and Corrections

While verifying the model transfer function in the previous section, the acceleration input is assumed to be a step input where the magnitude of the step is the DC component of the acceleration. This assumption is verified by separating the DC and non-DC components of the acceleration and viewing the non-DC components as a disturbance to the system:

$$\begin{aligned} a(t) &= \sum_{i=0}^{n}\{M_i \cos(\omega_i t + \phi_i)\} \\ &= M_0 + \sum_{i=1}^{n}\{M_i \cos(\omega_i t + \phi_i)\} \\ A(s) &= \frac{M_0}{s} + \sum_{i=1}^{n} M_i\{\frac{s * cos(\phi_i) - \omega_i * sin(\phi_i)}{s^2 + \omega_i^2}\} \end{aligned}$$

A model of the ornithopter with transfer function from acceleration to velocity and from forcing to velocity is shown in Figure 2.17. Here, the input $u(s) = \frac{M_0}{s}$ and the disturbance $d(s) = \sum_{i=1}^{n} M_i\{\frac{s*cos(\phi_i)-\omega_i*sin(\phi_i)}{s^2+\omega_i^2}\}$. In an ideal case each sinusoidal term is symmetric about zero and each will have a net effect in acceleration of zero over time. Because we are

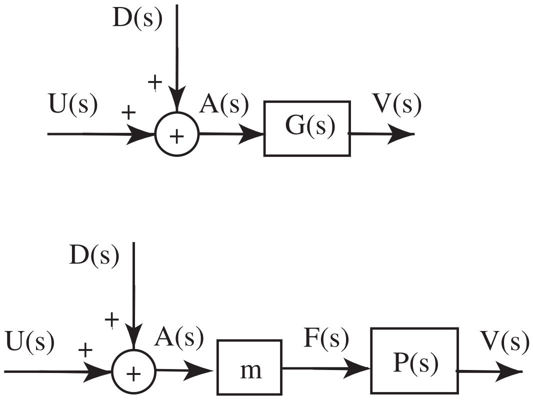


Figure 2.17. One-Dimensional Horizontal Model of Ornithopter with Disturbance

assuming a linear model of the ornithopter, the velocity output from each pure sinusoidal input term will include a magnitude and phase change, but will still be symmetric about zero. Because we do not have an ideal case and the ornithopter is not completely linear, some error in the velocity output is expected to exist.

Because the modeling done in the previous sections has been concerned with the experimental data, it reflects the motion of the total mass of the ornithopter and slider used to attach it to the linear bearing rail. If we assume that the same force is generated by the ornithopter with and without the mass of the slider included, we can make the following calculations for the ornithopter's actual acceleration:

$$F = ma$$

$$m_{test}a_{test} = m_{ornithopter}a_{new}$$

$$a_{new} = \frac{m_{test}}{m_{ornitopter}}a_{test} = \frac{0.2kg}{0.096kg}a_{test}$$

$$a_{new} \approx 2a_{test}$$

Therefore, the acceleration of the ornithopter alone is twice that of the sensor data acceleration.[5] Following simulations will reflect this increase in acceleration.

[5] Other causes of attenuation in acceleration may exist due to the friction between the slider and linear bearing rail and/or torques on the ornithopter.

2.3 One-Dimensional Horizontal Closed-Loop System

The following sections address challenges created by flapping of the wings, controllability and stability of the one-dimensional ornithopter model, and design of a controller.

2.3.1 Feedback Path Filter

A filter is used in the feedback path of the closed-loop system to avoid tracking of fluctuations in velocity created by the flapping of the wings. Though the ornithopter itself acts as a low pass filter (LPF), which filters the forcing input and has a cutoff frequency at about 1rad/sec (based on the transfer function, Equation 2.2, found in Section 2.2.3), fluctuations still exist in the velocity as is evident upon examination of the velocity data. This is problematic as the frequency of flapping is close to that of the oscillations one might see in velocity about steady-state. A LPF can be used to eliminate the majority of the unwanted frequency components and to facilitate tracking of the mean velocity. A LPF of the following form was chosen:

$$H(s) = \frac{a}{s+a}$$

for $a < 0$.

In choosing the placement of the filter pole one must consider the tradeoff between the delay of the filter and its filtering capability. As the filter pole decreases in magnitude, the filtering capability of lower frequencies increases. Because the contribution of the filtered frequency components is attenuated, the magnitude of the filtered signal is smaller and takes longer to reach a steady-state value. This is seen as a delay in the filter output and so it takes longer to track the actual velocity. A unity gain filter with good filtering capability (based on the frequency spectrum seen in Figure 2.5) and limited delay was chosen to filter frequencies above 0.5 Hz:

$$H(s) = \frac{2\pi * 0.5}{s + (2\pi * 0.5)} = \frac{\pi}{s+\pi}.$$

The simulated and filtered velocities are shown in Figure 2.18 and a Bode plot of the filter is shown in Figure 2.19.

2.3.2 Controllability, Stability, and Controller Choice for One-Dimensional Horizontal System

The simplified transfer function of the ornithopter from forcing input to velocity output is a first order transfer function consisting of a pole in the left-hand plane around

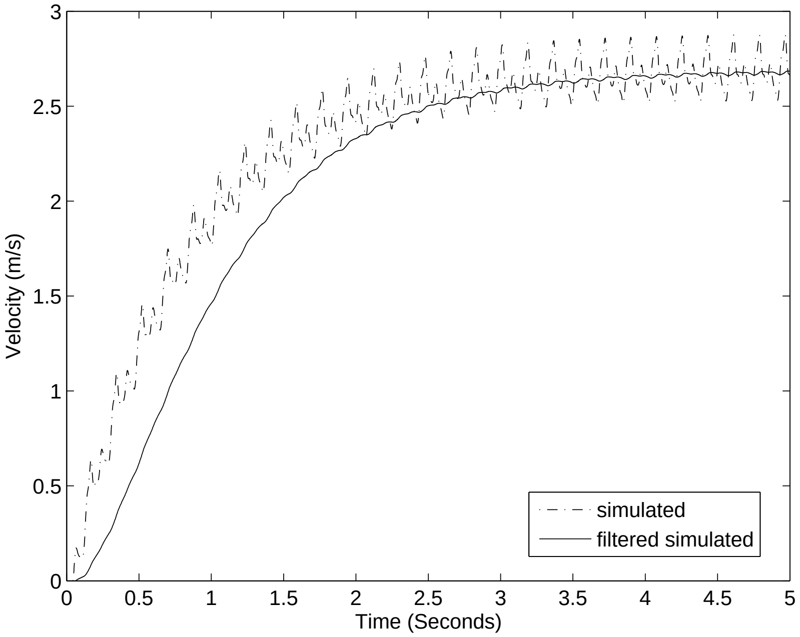


Figure 2.18. Simulated and Filtered Simulated Velocities

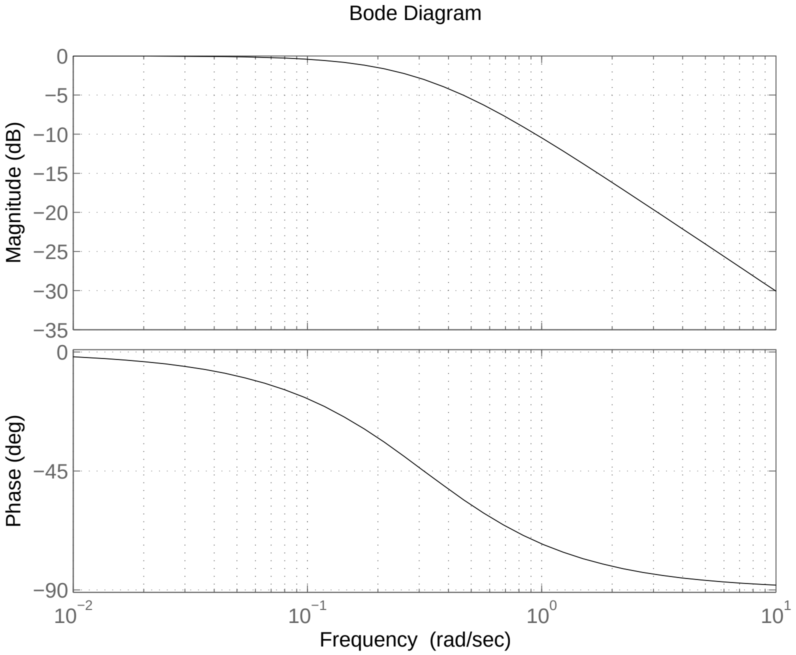


Figure 2.19. Bode Plot of Feedback Path Filter

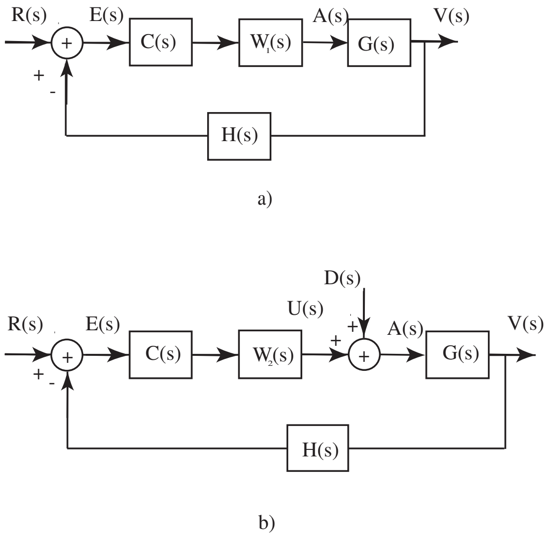
$s = -1$ (based on the transfer function, Equation 2.2, found in Section 2.2.3). Another pole and zero are added when the filter is included in the system. Stability of this simplified model can be assumed, since all of the poles are in the LHP. A controller can be chosen by looking at the closed-loop system again with the forcing input separated into a constant (DC) term plus sinusoidal terms. Figure 2.20 shows the system with input, U(s), and disturbance, D(s). The error, E(s), and output, V(s), can then be found as: $E(s) = R(s) - H(s)V(s)$ and $V(s) = G(s)D(s) + G(s)F(s)E(s)$, where $F(s) = C(s)W_2(s))$. We can then look at the total error:

$$\begin{aligned} E(s) &= R(s) - H(s)[G(s)D(s) + G(s)F(s)E(s)] \\ &= R(s) - H(s)G(s)D(s) - H(s)G(s)F(s)E(s) \end{aligned}$$

$$\begin{aligned} E(s)[1 + H(s)G(s)F(s)] &= R(s) - H(s)G(s)D(s) \\ E(s) &= \underbrace{\frac{R(s)}{1 + H(s)G(s)F(s)}}_{error\,due\,to\,input} - \underbrace{\frac{H(s)G(s)D(s)}{1 + H(s)G(s)F(s)}}_{error\,due\,to\,disturbance} \end{aligned}$$

which are the error due to the input and disturbance, respectively. We want zero error at steady-state which can be found using the Final Value Theorem:

$$\lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s) = 0$$



a)

b)

Figure 2.20. Closed-Loop System a) Shown as Originally Modeled b) Shown Modeling Sinusoidal Part of Acceleration as a Disturbance

Applying this to the error due to the input alone (for a step input):

$$\begin{aligned}
\lim_{s\to 0} sE(s) &= \lim_{s\to 0} \frac{sR(s)}{1+H(s)G(s)F(s)} \\
&= \lim_{s\to 0} \frac{s(\frac{1}{s})}{1+(\frac{n_H}{d_H})(\frac{n_G}{d_G})(\frac{n_F}{d_F})} \\
&= \lim_{s\to 0} \frac{d_H d_G d_F}{d_H d_G d_F + n_H n_G n_F}
\end{aligned}$$

One can see that if the denominator of F(s), d_F, becomes zero as $s \to 0$, then the error due to the input will also be zero. This requires a pole at 0. Since $F(s) = C(s)W_2(s)$ and $W_2(s)$ cannot be changed, the denominator of the controller d_C must have a pole at zero. We now look at the error due to the disturbance alone (again, for a step input in disturbance):

$$\begin{aligned}
\lim_{s\to 0} sE(s) &= \lim_{s\to 0} -\frac{sH(s)G(s)D(s)}{1+H(s)G(s)F(s)} \\
&= \lim_{s\to 0} -\frac{s(\frac{n_H}{d_H})(\frac{n_G}{d_G})(\frac{n_D}{d_D})}{1+(\frac{n_H}{d_H})(\frac{n_G}{d_G})(\frac{n_F}{d_F})} \\
&= \lim_{s\to 0} -\frac{s n_H n_G n_D \frac{d_F}{d_D}}{d_H d_G d_F + n_H n_G n_F}
\end{aligned}$$

We see again that if d_F has a pole at 0, the error due to the disturbance will also approach zero as $s \to 0$. This again implies that d_C must have a pole at zero. A proportional-integral (PI) controller can then be chosen to eliminate error in the output at steady-state due to a step in the input and disturbance. Adding derivative action to the controller is undesirable for the same reasons the filter was added. We do not want the controller to track velocity fluctuations. Because the actual system is subject to more than a step input disturbance, large feedback gains can theoretically be used to reduce the error caused by the sinusoidal terms.

2.3.3 One-Dimensional Horizontal Closed-Loop Simulations and Controller Design

Control of the one-dimensional horizontal system is considered and simulated in this section. Given that C(s), $W_2(s)$, G(s), and H(s) from Figure 2.20 are of the forms: $C(s) = \frac{K(s+z_C)}{s}$, $W_2(s) = C_{w2}$, $G(s) = \frac{V(s)}{A(s)} = \frac{2.34}{s+1.28}$, and $H(s) = \frac{\pi}{s+\pi}$, where C_{w2} is a constant, the simplified loop-gain of the system is:

$$C(S)W_2(s)G(s)H(s) = \frac{2.34C_{w2}K(s+z_C)\pi}{s(s+1.28)(s+\pi)}.$$

The root locus of this simplified system does not consider the effects of the sinusoidal components and so is incomplete, however it does offer insight into placement of the controller zero. Simulations were used to test the system with zeros at different locations and to find and adjust gains. The following PI controller was found:

$$C(S) = \frac{K_P(s + \frac{K_I}{K_P})}{s} = \frac{K(s + 0.05)}{s}$$

with K = 40. Figure 2.21 shows a closed-loop SIMULINK model of the ornithopter. Important to note are the Relay, Frequency of Flapping Limiter, and Product blocks. Together, this combination effectively limits the allowed flapping frequencies, while enabling the ornithopter to turn off when the control signal reaches a certain point. Currently, the relay will turn off after the control signal goes below $-40Hz$. This is an arbitrary point used to avoid having the ornithopter turn off during initial testing and simulations. However, this point can be changed to allow thee ornithopter to turn off. The ability of this component will be considered more in Section 2.3.4.

The desired and actual velocities of the simulated closed-loop system are shown in Figure 2.22. The controller output is seen in Figure 2.23 along with the limited control signal (limited from 4-7 Hz as noted in Section 2.2.2). Upon examination of the controller signals, one can see that the controller is tracking the fluctuations caused by the flapping of the wings despite attempts to filter them. To curb this problem, the controller's proportional gain is reduced from $K_P = 40$ to $K_P = 10$ once the error between the desired and actual velocities reduces to below $0.1m/s$. Figure 2.24 and 2.25 show the velocity and control signals for the improved controller.

2.3.4 One-Dimensional Horizontal Closed-Loop System Additions

As noted in Section 2.3.3, there exists a Relay, Frequency of Flapping Limiter, and Product block in the SIMULINK Model of the Horizontal Closed-Loop System seen in Figure 2.21. Not only does this allow the ornithopter to turn off at some point, but it also allows the ornithopter to achieve a flapping-gliding behavior to reach slower speeds. Though this is not a main concern in this thesis, it is worth noting for future developments of the controller design. To demonstrate the ability of this part of the controller, simulations of the ornithopter at velocities in and below the range for continuous flapping are shown in Figure 2.26. Control signals can be seen in Figure 2.27 where the relay turns the flapping off when the (unlimited) control signal goes below $-4Hz$.

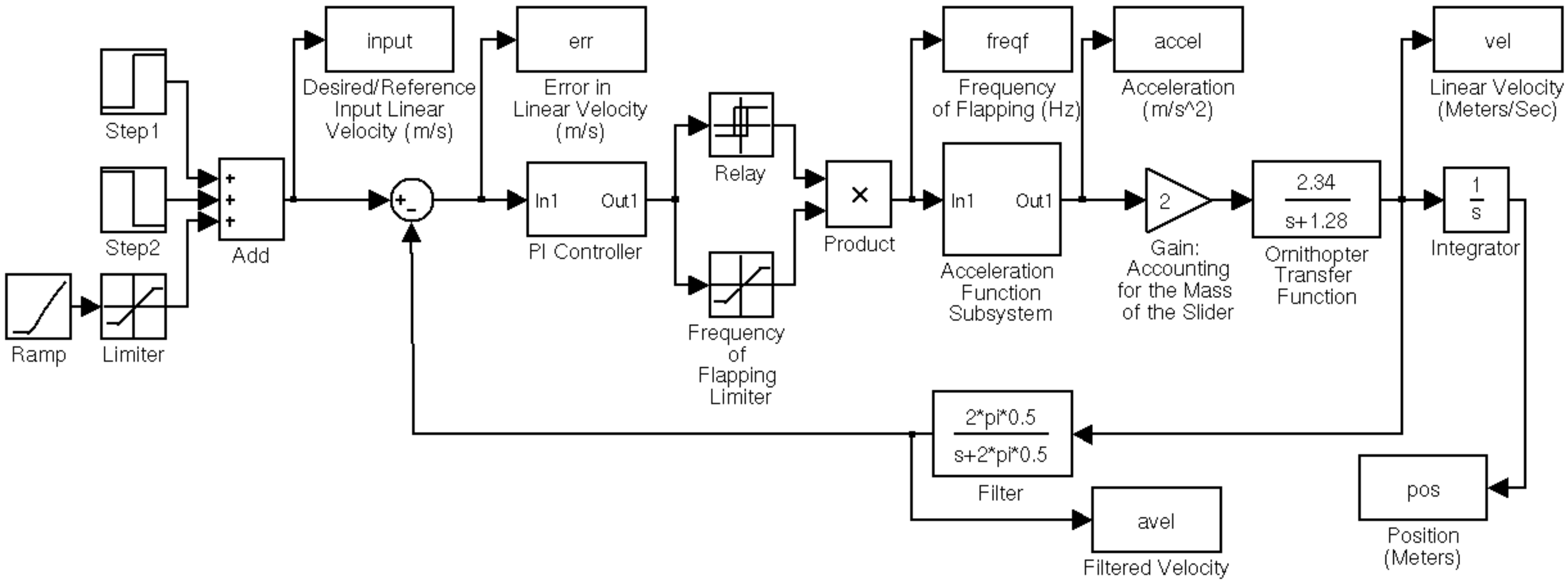

input
Desired/Reference Input Linear Velocity (m/s)
err
Error in Linear Velocity (m/s)
freqf
Frequency of Flapping (Hz)
accel
Acceleration (m/s^2)
vel
Linear Velocity (Meters/Sec)
Step1
Step2
Ramp
Limiter
+ + +
Add
+ -
In1 Out1
PI Controller
Relay
Frequency of Flapping Limiter
×
Product
In1 Out1
Acceleration Function Subsystem
2
Gain: Accounting for the Mass of the Slider
2.34 / s+1.28
Ornithopter Transfer Function
1 / s
Integrator
2*pi*0.5 / s+2*pi*0.5
Filter
avel
Filtered Velocity
pos
Position (Meters)

Figure 2.21. SIMULINK Model of One-Dimensional Horizontal Closed-Loop System

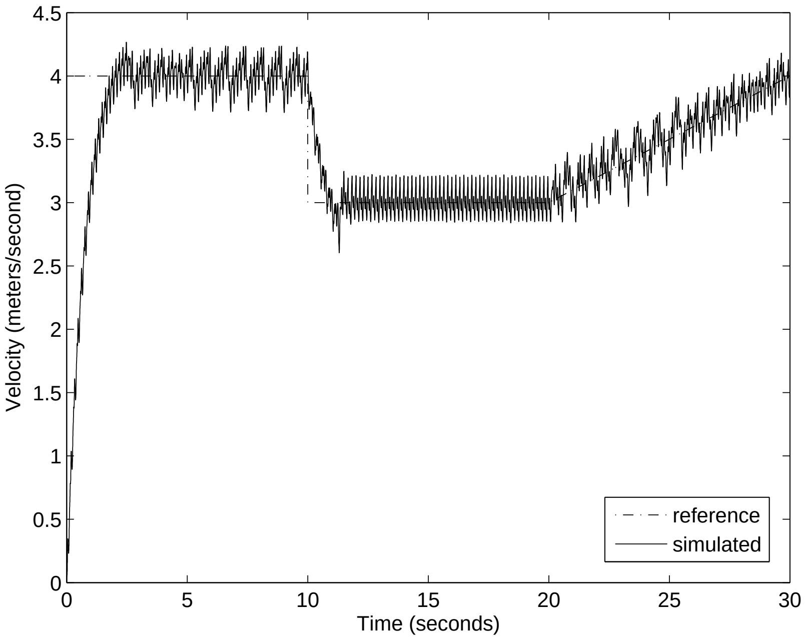


Figure 2.22. Simulated One-Dimensional Horizontal Closed-Loop Velocity with $K_P = 40$ and $K_I = 2$

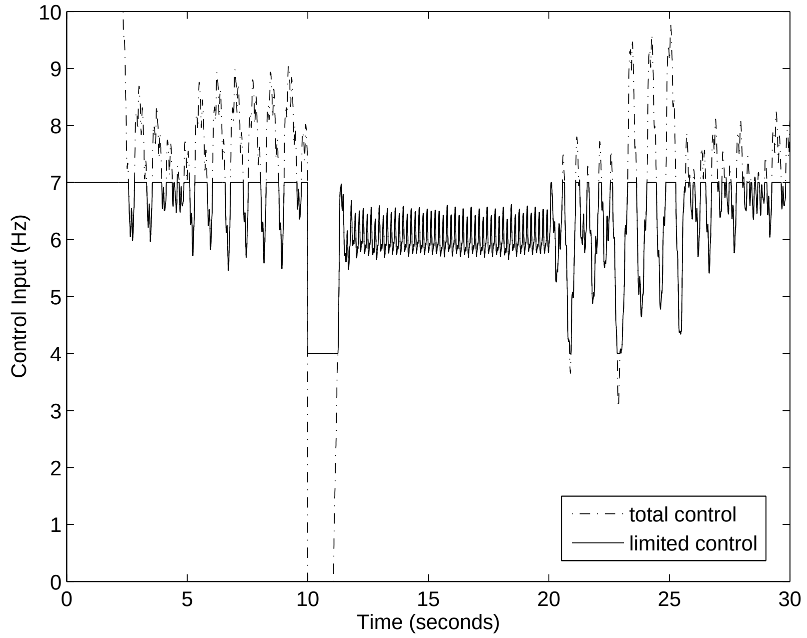


Figure 2.23. Control Signals of Simulated One-Dimensional Horizontal Closed-Loop System with $K_P = 40$ and $K_I = 2$

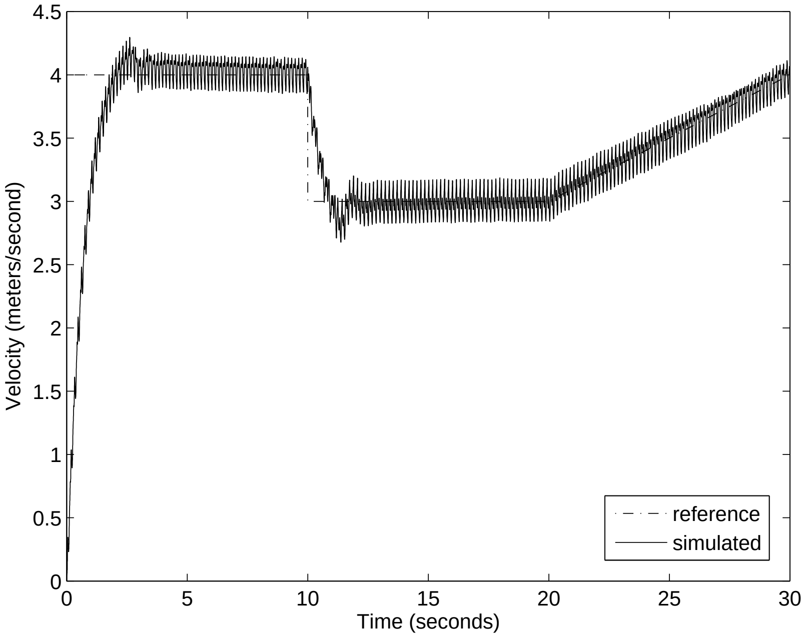


Figure 2.24. Simulated One-Dimensional Horizontal Closed-Loop Velocity (Updated Controller) with $K_P = 40$ and $K_I = 2$)

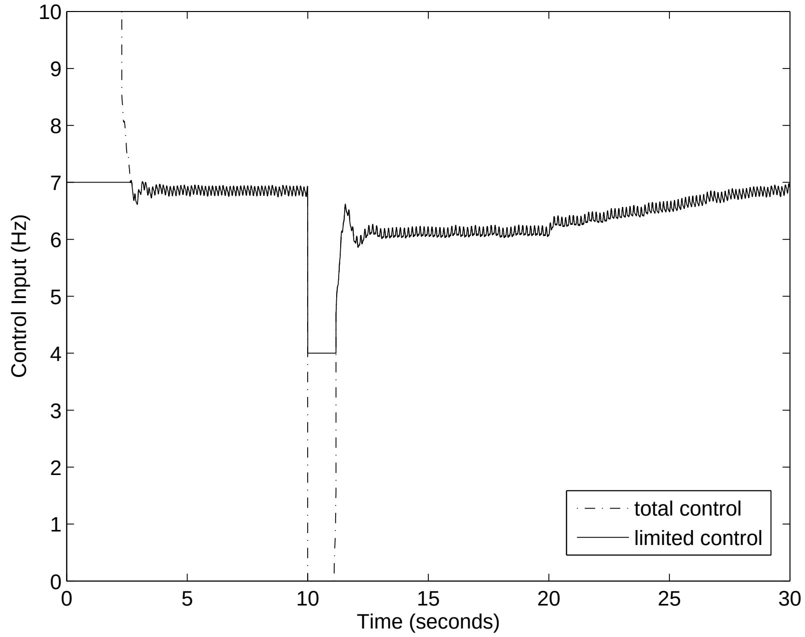


Figure 2.25. Control Signals of Simulated One-Dimensional Horizontal Closed-Loop System (Updated Controller) with $K_P = 40$ and $K_I = 2$

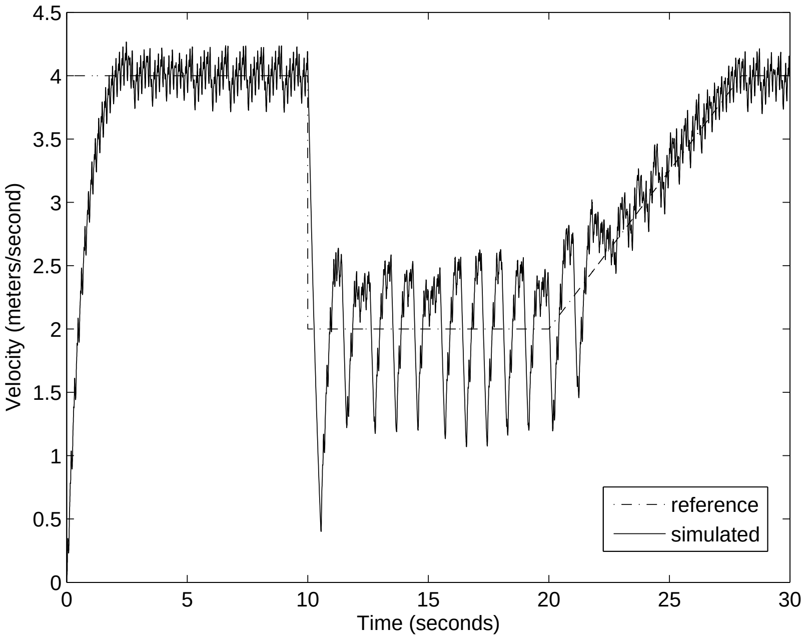


Figure 2.26. Simulated One-Dimensional Horizontal Closed-Loop Velocity with $K_P = 40$ and $K_I = 2$ Showing Flapping-Gliding Behavior

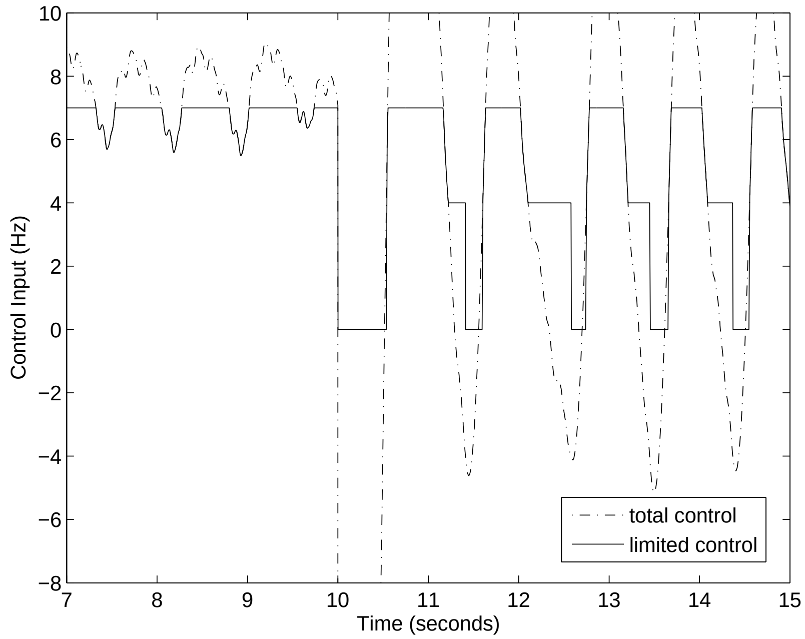


Figure 2.27. Close-up View of Control Signals of Simulated One-Dimensional Horizontal Closed-Loop System with $K_P = 40$ and $K_I = 2$ Showing Flapping-Gliding Behavior

2.4 One-Dimensional Vertical Closed-Loop System

In order to gain more insight into the ornithopter system, analysis of a one-dimensional system of the ornithopter in hover is considered in this section. The information gained here will help in developing a state-space model, which will be considered in the next section. Here, the ornithopter is in a completely vertical facing stance. This allows the ornithopter to direct all of its thrust upward, to counteract gravity, while limiting its horizontal movement. An equation describing this motion is:

$$f(t) - mg = m\ddot{y} + B\dot{y} = m\dot{v} + Bv.$$

Hence, the same transfer function will appear for the vertical case, along with a gravitational term subtracted from the forcing term. Figure 2.28 show a free body diagram of this system. Because the maximum acceleration of the ornithopter is around $2.4m/s^2$, the current design of the ornithopter cannot overcome gravity (at $9.81m/s^2$). If we assume a gain in thrust ($thrust = force = ma$) is possible through physical reassessment and design of the ornithopter, a gain block can be added in series with the acceleration s-function block to reflect this adjustment. Though lacking in precision, this approach may lead to insight into controllability of the system in hover.

Open-loop simulations of the system lead to the conclusion that a minimum gain in acceleration of $G = 4.6$ is required for hover at a flapping frequency of 7Hz. This is slightly

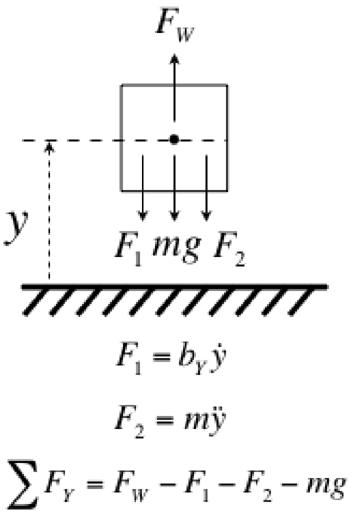


Figure 2.28. One-Dimensional Free Body Diagram of Ornithopter: Vertical Movement

above the required gain of about 4.1 when considering the estimated acceleration at 7Hz in Figure 2.9 (remember that the data in this figure must be multiplied by a gain of 2 in order to compensate for the mass of the slider). Simulations of the closed-loop system lead to a small change in the controller from that found for the horizontal case. Although a gain of $K = 40$ will still work, the zero is moved from $z = -0.05$ to $z = -1$. This amounts to an increase in the integral component of the control signal since $\frac{K_I}{K_P} = -z$. Because the gain in acceleration increases the magnitude of the fluctuations in velocity, the updated controller of Section 2.3.3 was of no help. Simulations of the closed-loop system velocity and position are seen in Figure 2.29 and 2.30 with a gain in acceleration of $G = 5$ for good measure. The reference input for velocity dictates a constant step, step down to zero, and then gradual ramp. This allows the integrator to accumulate enough gain to have an effect on the system at hover and shows the capability of the system beyond hovering.

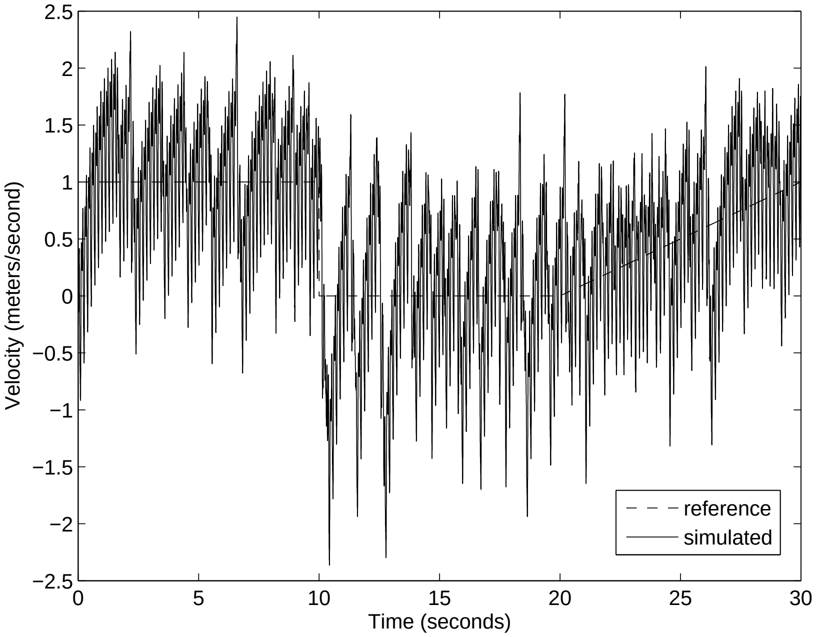


Figure 2.29. Simulated One-Dimensional Vertical Closed-Loop Velocity with $K_P = 40$ and $K_I = 40$

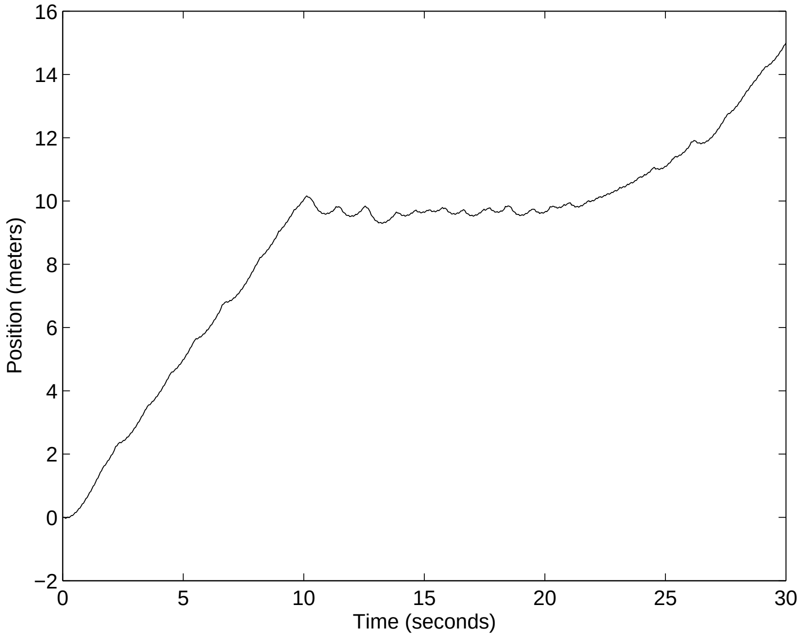


Figure 2.30. Simulated One-Dimensional Vertical Closed-Loop Position with $K_P = 40$ and $K_I = 40$

CHAPTER 3

TWO-DIMENSIONAL CONTROL

In this chapter, a state-space model of the ornithopter is derived to allow for velocity and altitude control. Equilibrium points are analyzed for vertical hover and for steady horizontal flight. Feedback gains are found for both cases to implement control.

3.1 A General Two-Dimensional Model of the Ornithopter

A general free body diagram of the ornithopter in flight is shown in Figure 3.1. The following notation applies to Figure 3.1 and following free body diagrams:

x the horizontal axis.

y the vertical axis.

x̂ the horizontal body axis.

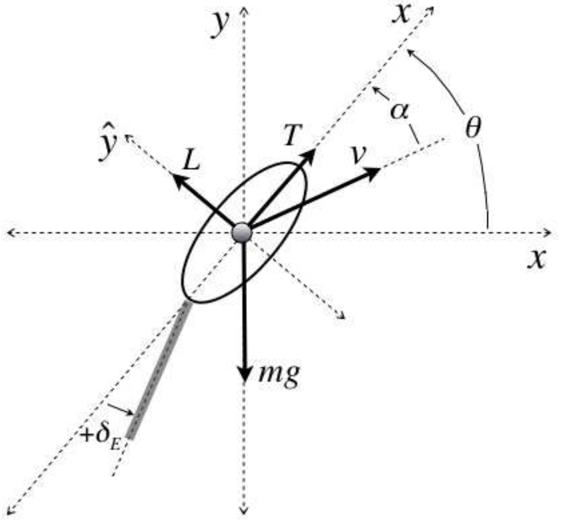


Figure 3.1. General Two-Dimensional Free Body Diagram of Ornithopter in Flight

$\hat{\mathbf{y}}$ the vertical body axis.

θ the angular position of the ornithopter.

α the angle of attack of the ornithopter and angle between the velocity vector and horizontal body axis.

$\delta_{\mathbf{E}}$ the angle of the tail (or elevator).

m the mass of the ornithopter.

g Earth's gravity: g $\approx 9.81m/s^2$.

v the velocity of the ornithopter: $v = \sqrt{v_x^2 + v_y^2} = \sqrt{\dot{x}^2 + \dot{y}^2}$.

T the thrust produced by the wings, which is assumed to be parallel to the body axis.

L the lift generated by oncoming air flow.

The equations of motion for this case can be summarized as:

$$\begin{array}{rcl} m\ddot{x} + b_x\dot{x} & = & T\cos\theta - L\sin\theta \\ m\ddot{y} + b_y\dot{y} & = & T\sin\theta + L\cos\theta - mg \\ \ddot{\theta}I + b_\theta\ddot{\theta} & = & -K_t\delta_E \end{array}$$

where b_x, b_y, and b_θ are horizontal, vertical, and rotational damping of the ornithopter, I is the moment of inertia of the ornithopter, and K_t is a torque constant relating the total torque, of which the ornithopter is subject, to the angle of the tail. From the free body diagram, we can also conclude that $\dot{x} = v\cos(\theta - \alpha)$ and $\dot{y} = v\sin(\theta - \alpha)$.

3.2 Two-Dimensional Closed-Loop System with Vertical Equilibrium

The system in hover described in the Section 2.4 is extended here to state-space representation. This allows for a more realistic model of the ornithopter and is helpful as a precursor to velocity and altitude control for steady horizontal flight. Figure 3.2 shows a free body diagram of this system. The notation from Section 3.1 applies to this section as well. Here, the lift, L, is approximately equal to zero. Hence, the equations of motion used for the ornithopter in vertical hover can be summarized as:

$$m\ddot{x} + b_x\dot{x} = T\cos\theta$$

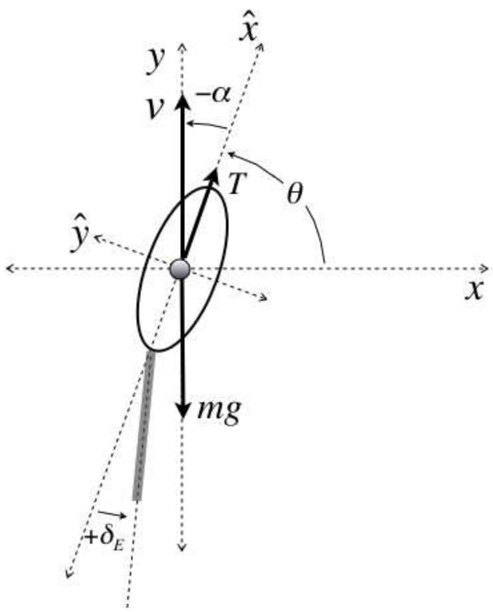


Figure 3.2. Two-Dimensional Free Body Diagram of Ornithopter in Hover

$$\begin{aligned} m\ddot{y} + b_y\dot{y} &= T\sin\theta - mg \\ I\ddot{\theta} + b_\theta\dot{\theta} &= -K_t\delta_E. \end{aligned}$$

An approximation of the moment of inertia is made using the following equation for the moment of inertia for a solid cuboid:

$$I = \frac{1}{12}m(h^2 + l^2)$$

where $h = height$ and $l = length$. For $h \approx 4\,in. = 0.1016\,m$, $l \approx 6\,in. = 0.1524\,m$, and $m = 0.096\,kg$, we find that $I \approx 2.6839E - 4\,kg \cdot m^2$. An estimate of the rotational damping and torque constant can be made by considering two cases for the equation $I\ddot{\theta} + b_\theta\dot{\theta} = -K_t\delta_E$. In the first case, the ornithopter is at rest and a maximum step in angular acceleration, $\ddot{\theta}_{max}$, is applied to the system. At this point $\dot{\theta} = 0$ and $\delta_E = \delta_{E,max}$. From observation, $\delta_{E,max} \approx \pm 45°$ and $\ddot{\theta}_{max} \approx \pm 540°/s^2 \approx \pm 9.43\,rad/s^2$.[1] We can then solving for $K_t \approx 0.0032$. In the second case, the ornithopter is in steady rotation. Here $\dot{\theta} = \dot{\theta}_{max} = 180°/s$ and $\ddot{\theta} = 0$.[2] We can then solve for $b_\theta \approx 8.0516E - 4\,N \cdot s/m$.

[1] The approximation for $\ddot{\theta}_{max}$ is a ballpark figure and should be validated in future work.

[2] The approximation for $\dot{\theta}_{max}$ is also a ballpark figure and should be validated in future work.

Now that we have the required information for the equations of motion, we can continue with analysis of the system. The state-space equations for this system become:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} \; and \; \dot{x} = \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \\ \dot{x_5} \\ \dot{x_6} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{u_1}{m}\cos x_5 - \frac{b_x}{m}x_2 \\ x_4 \\ \frac{u_1}{m}\sin x_5 - \frac{b_y}{m}x_4 - g \\ x_6 \\ -\frac{K_t}{I}u_2 - \frac{b_\theta}{I}x_6 \end{bmatrix}$$

with input

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} T \\ \delta_E \end{bmatrix}.$$

This gives a set of nonlinear differential equations of the form

$$\dot{x}(t) = h(x(t), u(t), t) \tag{3.1}$$

A linearized system can be obtained, as an estimate of this system, by evaluating the state-space equations at an equilibrium point. If the system is linearized around an equilibrium point with initial state, x_0, and input, u_0, which is subject to small perturbations, 3.1 become:

$$\begin{aligned} \dot{x_0}(t) + \dot{\overline{x}}(t) &= h(x_0(t) + \overline{x}(t), u_0(t) + \overline{u}(t), t) \\ &= h(x_0(t), u_0(t), t) + \frac{\partial h}{\partial x}\overline{x} + \frac{\partial h}{\partial u}\overline{u} + \cdots, \end{aligned}$$

which reduces to

$$\dot{\overline{x}}(t) = \frac{\partial h}{\partial x}\overline{x} + \frac{\partial h}{\partial u}\overline{u} = A(t)\overline{x} + B(t)\overline{u}$$

(where $A(t) := \frac{\partial h}{\partial x}$ and $B(t) := \frac{\partial h}{\partial u}$ are Jacobians) [24]. For the ornithopter in vertical hover, θ is approximately equal to 90° and the angle of attack, α, is approximately equal to zero. The equilibrium is expressed as:

$$x_0 = \begin{bmatrix} 0 \\ 0 \\ C_1 \\ 0 \\ C_2 \\ 0 \end{bmatrix} \; and \; \dot{x}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where C_1 = constant altitude, $C_2 = 90°$ = constant angular position, and we assume that the system experiences no rotational torque at its equilibrium. This requires an initial input of:

$$u_0 = \begin{bmatrix} mg \\ 0 \end{bmatrix}.$$

The Jacobian matrices, A and B, can then be solved for as:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{b_x}{m} & 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{b_y}{m} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\frac{b_\theta}{I} \end{bmatrix} \; and \; B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m} & 0 \\ 0 & 0 \\ 0 & -\frac{K_t}{I} \end{bmatrix}.$$

Because the rank of the controllability matrix, $C = [B\, AB\, A^2B \, \cdots \, A^{n-1}B]$, is of full rank ($rank(C) = 6$), the system is assumed to be controllable. This linearized system can then be used to design state-feedback gains, which in turn can be used to control the nonlinear system. Closed-loop system gains are found for the linearized system to place the desired closed-loop system poles in a Butterworth pattern below the $\zeta = 0.707$ damping line (the damping, $\zeta > 0.707$) [24]. By varying the radius of the closed-loop poles, gains can be found which yield the best response. For a system of the following form:

$$\dot{x}(t) = Ax + Bu$$

with $u = u_0 - \bar{u}$, $\bar{u} = K\bar{x}$, and $\bar{x} = x - x_0$, the closed-loop system becomes:

$$\dot{x}(t) = (A - BK)x + Bu_0 + BKx_0.$$

For a radius of $R = 15$, the gain matrix is

$$K = \begin{bmatrix} -390.7551 & -61.4923 & 19.9253 & 2.5061 & 41.4851 & 1.0601 \\ 441.5719 & 92.4694 & 1.2553 & 0.0886 & -92.2723 & -4.1380 \end{bmatrix}.$$

Low-pass filters, each with a pole at $p = 2\,Hz$ are used to filter the $\dot{x}$ and $\dot{y}$ states before feeding them back. The filter poles in this system have been increased from those in the one-dimenasional case to reduce the delay. Poles at $p = 0.5\,Hz$ add too much delay for this system to remain stable.

In addition to state-feedback, integral control is added for robust tracking and disturbance rejection [24]. This addition aids in control where disturbances exist (from the flapping of the wings). It is also needed due to a constant error in the mean steady-state altitude, y, for the system with feedback gains alone. An augmented state variable, x_a, is added. We now have:

$$u = u_0 - \bar{u} + \tilde{u} = u_0 - K\bar{x} + K_a x_a$$

with

$$\dot{x}_a = r - y = r - Cx$$

where

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \; y = \begin{bmatrix} \dot{x} \\ y \\ \theta \end{bmatrix}, \; r = \begin{bmatrix} \dot{x}_d \\ y_d \\ \theta_d \end{bmatrix},$$

and

$$K_a = \begin{bmatrix} K_{a_{11}} & K_{a_{12}} & K_{a_{13}} \\ K_{a_{21}} & K_{a_{22}} & K_{a_{23}} \end{bmatrix} = \begin{bmatrix} 30 & 30 & 30 \\ 0 & 0 & 0 \end{bmatrix}.$$

Here, the subtext, d, stands for the desired state. Simulation results for the altitude and vertical velocity are shown in Figures 3.3 and 3.4. Simulation results for the horizontal and angular position are shown in Figure 3.5 and 3.6.

3.3 Two-Dimensional Closed-Loop System with Horizontal Equilibrium

In this section, a longitudinal state-space representation is found for the ornithopter in steady horizontal flight. In this case, the ornithopter is subject to a lift force, L. Because we do not know the exact lift model for the ornithopter, a linear function is used to approximate lift generation based on observations obtained from nominal flight under manual control. Hence, we assume a linearized model of the lift to be proportional to velocity, v. The equations of motion for this case can be summarized as:

$$\begin{aligned} m\ddot{x} + b_x\dot{x} &= T\cos\theta - L\sin\theta \\ m\ddot{y} + b_y\dot{y} &= T\sin\theta + L\cos\theta - mg \\ \ddot{\theta}I + b_\theta\ddot{\theta} &= -K_t\delta_E \end{aligned}$$

With notation defined in Section 3.1. Since lift is assumed to be proportional to velocity, it can be expressed as $L = K_l v = \sqrt{v_x^2 + v_y^2}$ (where K_l is a lift constant). A free body diagram of this system is included in Figure 3.7 to aid in visualization. The state-space representation then becomes

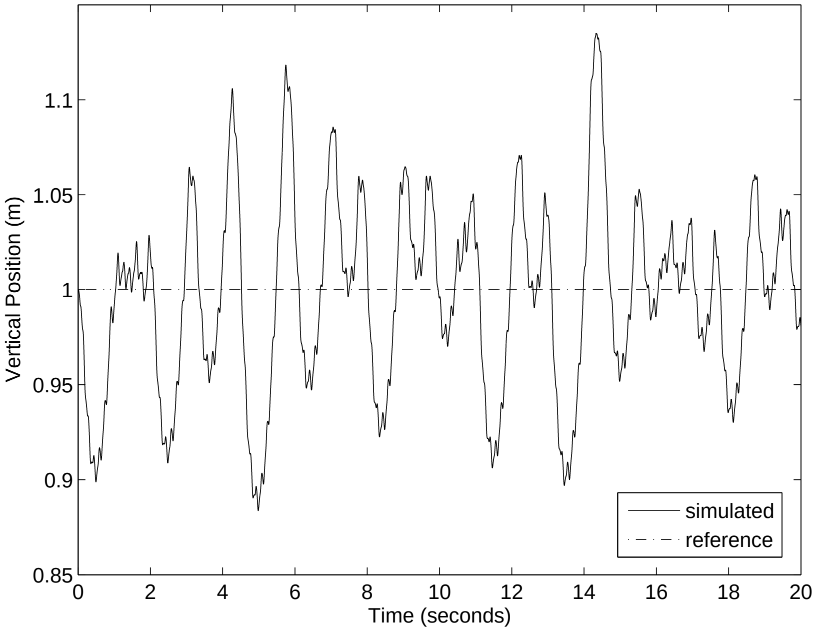


Figure 3.3. Simulated Two-Dimensional Closed-Loop System in Vertical Hover: Vertical Position

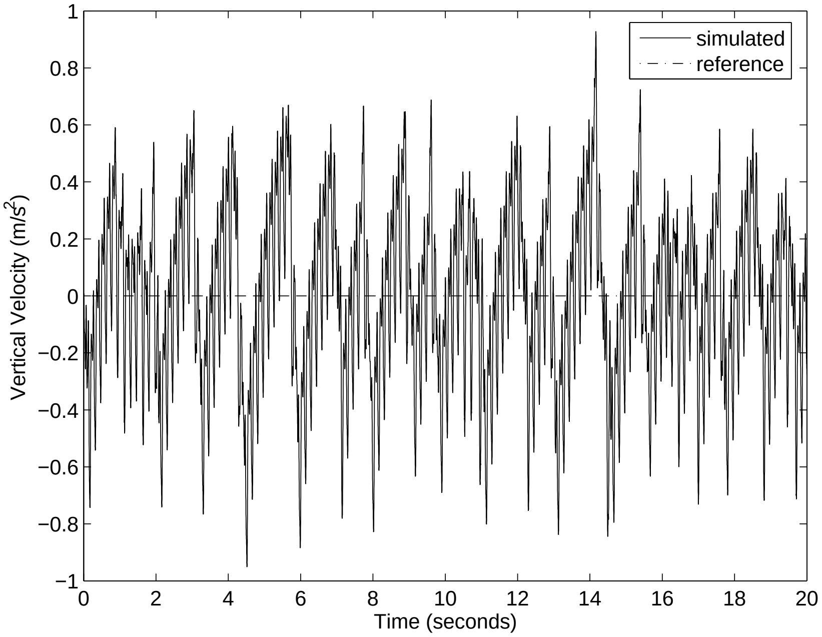


Figure 3.4. Simulated Two-Dimensional Closed-Loop System in Vertical Hover: Vertical Velocity

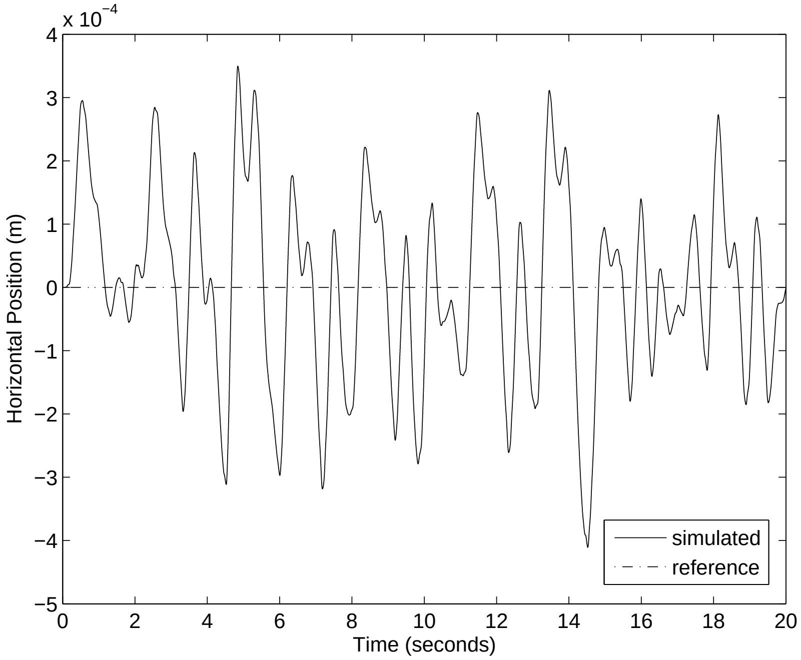


Figure 3.5. Simulated Closed-Loop System in Vertical Hover: Horizontal Position

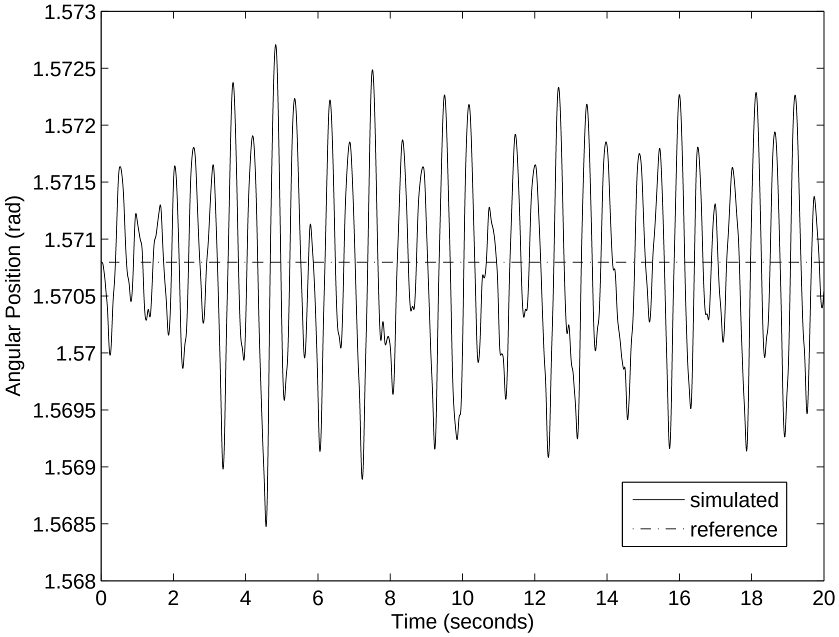


Figure 3.6. Simulated Closed-Loop System in Vertical Hover: Angular Position

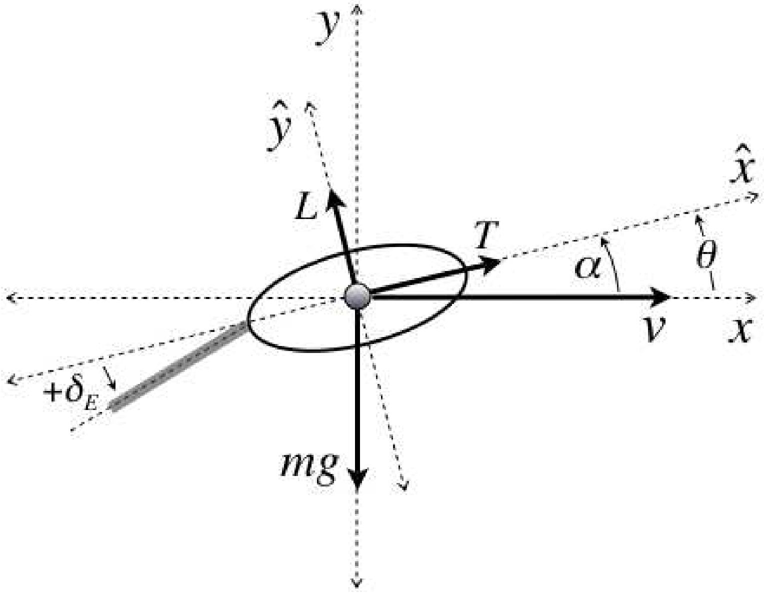


Figure 3.7. Two-Dimensional Free Body Diagram of Ornithopter in Horizontal Flight

$$\dot{x} = \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \\ \dot{x_5} \\ \dot{x_6} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{u_1}{m}\cos x_5 - \frac{K_l}{m}\sqrt{x_2^2 + x_4^2}\cos\left(90^\circ - x_5\right) \ - \frac{b_x}{m}x_2 \\ x_4 \\ \frac{u_1}{m}\sin x_5 + \frac{K_l}{m}\sqrt{x_2^2 + x_4^2}\sin\left(90^\circ - x_5\right) - \frac{b_y}{m}x_4 - g \\ x_6 \\ -\frac{K_t}{I}u_2 - \frac{b_\theta}{I}x_6 \end{bmatrix}.$$

A horizontal equilibrium can then be found for the ornithopter in steady horizontal flight. Here, the ornithopter is flying with a constant horizontal velocity, $\dot{x}_0 = C_1$, at a constant altitude, $y_0 = C_2$, and at a constant angular position, $\theta_0 = C_3$. At this equilibrium point, the ornithopter experiences no increase in velocity or altitude and experiences no rotational torque. The initial states for this equilibrium are

$$x_0 = \begin{bmatrix} 0 \\ C_1 \\ C_2 \\ 0 \\ C_3 \\ 0 \end{bmatrix} and\, \dot{x} = \begin{bmatrix} C_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

From observation, $\theta_0 \approx 5^\circ$. Since the velocity vector is aligned with the x-axis, we also find that $\alpha \approx 5^\circ$. Therefore, $\dot{x} = vcos(\theta - \alpha) = v$ and $\dot{y} = vsin(\theta - \alpha) = 0$ as desired. The initial input, u_1, and lift constant, K_l, required for equilibrium can be solved for from the equations of motion via matrix algebra as $Ax = B$, where A and B (here A and B are different from the Jacobian matrices A and B) are:

$$A = \begin{bmatrix} \cos\theta & -\sqrt{x_2^2 + x_4^2}\sin\theta \\ \sin\theta & \sqrt{x_2^2 + x_4^2}\cos\theta \end{bmatrix} \text{ and } B = \begin{bmatrix} m\ddot{x} + b_x\dot{x} \\ m\ddot{y} + b_y\dot{y} + mg \end{bmatrix}.$$

We can then solve for $x = A^{-1}B = \begin{bmatrix} T_0 \\ K_l \end{bmatrix}$. We now have $u_0 = \begin{bmatrix} u_{1,0} \\ u_{2,0} \end{bmatrix} = \begin{bmatrix} T_0 \\ 0 \end{bmatrix}$. With these initial conditions, the Jacobian matrices, A and B, become:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & a_{2,2} & 0 & a_{4,2} & a_{5,2} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & a_{2,4} & 0 & a_{4,4} & a_{5,4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & a_{6,6} \end{bmatrix} \text{ and } B = \begin{bmatrix} 0 & 0 \\ \frac{1}{m}\cos x_5 & 0 \\ 0 & 0 \\ \frac{1}{m}\sin x_5 & 0 \\ 0 & 0 \\ 0 & -\frac{K_t}{I} \end{bmatrix},$$

where:

$$\begin{aligned} a_{2,2} &= -\frac{1}{2}\left(\frac{K_l}{m}\right)(x_2^2 + x_4^2)^{-\frac{1}{2}}(2x_2)\sin x_5 - \frac{b_x}{m} \\ a_{2,4} &= \frac{1}{2}\left(\frac{K_l}{m}\right)(x_2^2 + x_4^2)^{-\frac{1}{2}}(2x_2)\cos x_5 \\ a_{4,2} &= -\frac{1}{2}\left(\frac{K_l}{m}\right)(x_2^2 + x_4^2)^{-\frac{1}{2}}(2x_4)\sin x_5 \\ a_{4,4} &= \frac{1}{2}\left(\frac{K_l}{m}\right)(x_2^2 + x_4^2)^{-\frac{1}{2}}(2x_4)\cos x_5 - \frac{b_y}{m} \\ a_{5,2} &= -\frac{u_1}{m}\sin x_5 - \left(\frac{K_l}{m}\right)\sqrt{x_2^2 + x_4^2}\cos x_5 \\ a_{5,4} &= \frac{u_1}{m}\cos x_5 - \left(\frac{K_l}{m}\right)\sqrt{x_2^2 + x_4^2}\sin x_5 \\ a_{6,6} &= -\frac{b_\theta}{I} \end{aligned}$$

and are evaluated at the initial states. As was done in Section 3.2, the controllability matrix is determined to be of full rank and then gains are found and tested in simulation. Simulation worked well for the nonlinear system without flapping input for almost any closed-loop pole radius, R. There was also a noticeable increased in responsiveness with an increase in R (or gain, K). However, once the flapping is included the system becomes unstable. Feedback gains alone are not sufficient and the robust tracking implemented in Section 3.2 must be used to maintain stability. Control is implemented for a closed-loop pole radius, $R = 10$, and a corresponding gain matrix of

$$K = \begin{bmatrix} -29.2864 & 1.0324 & 39.5990 & 15.4548 & 9.6418 & 0.4829 \\ 51.3869 & 1.5476 & -9.9783 & -14.2381 & -20.8977 & -1.9712 \end{bmatrix}.$$

Unity feedback is implemented with:

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \; y = \begin{bmatrix} \dot{x} \\ y \\ \theta \end{bmatrix}, \; r = \begin{bmatrix} \dot{x}_d \\ y_d \\ \theta_d \end{bmatrix},$$

and

$$K_a = \begin{bmatrix} K_{a_{11}} & K_{a_{12}} & K_{a_{13}} \\ K_{a_{21}} & K_{a_{22}} & K_{a_{23}} \end{bmatrix} = \begin{bmatrix} 100 & 100 & 100 \\ 0 & 0 & 0 \end{bmatrix}.$$

Simulation results for horizontal position and velocity, vertical position, and angular position are shown for $\theta_0 = 5°$ in Figures 3.8-3.11.

Though gains for the augmented states can be increased, within reason, without detrimental effect to the system, simulations become unstable for state feedback gains which cause the closed-loop pole radii to be larger then $R = 15$. In fact, the system is quite sensitive to large gains. The acceleration gain of $G = 5$ added for simulation of the ornithopter in hover (in Section 3.2) makes the system in steady horizontal flight unstable. When the gain is reduced to $G = 3$, the system works quite well. This behavior may be ascribable to an insufficiency in the modeling of the ornithopter and in particular to that

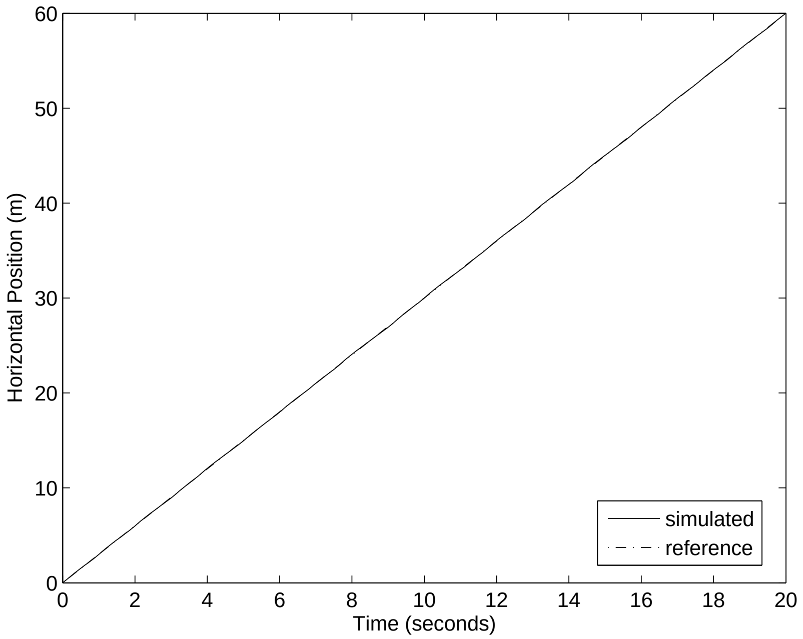


Figure 3.8. Simulated Two-Dimensional Closed-Loop for Horizontal Flight: Horizontal Position

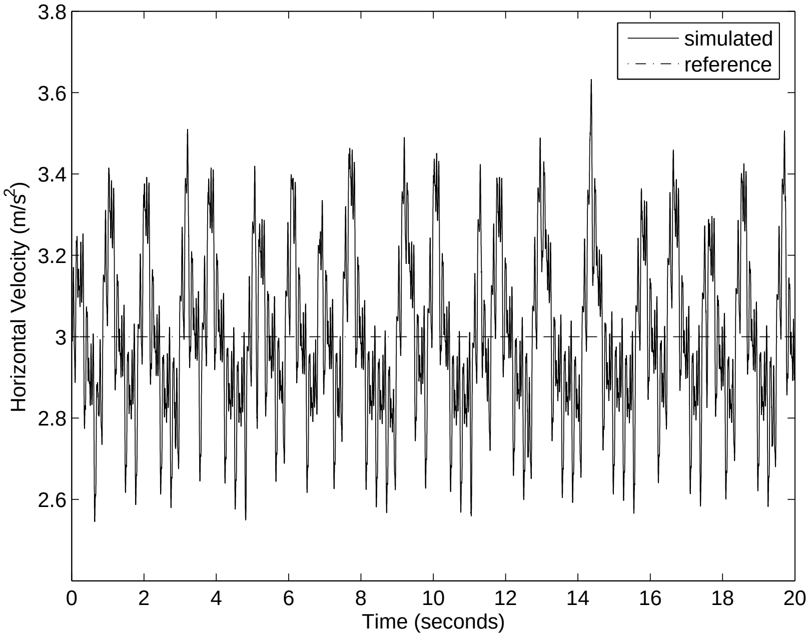


Figure 3.9. Simulated Two-Dimensional Closed-Loop System for Horizontal Flight: Horizontal Velocity

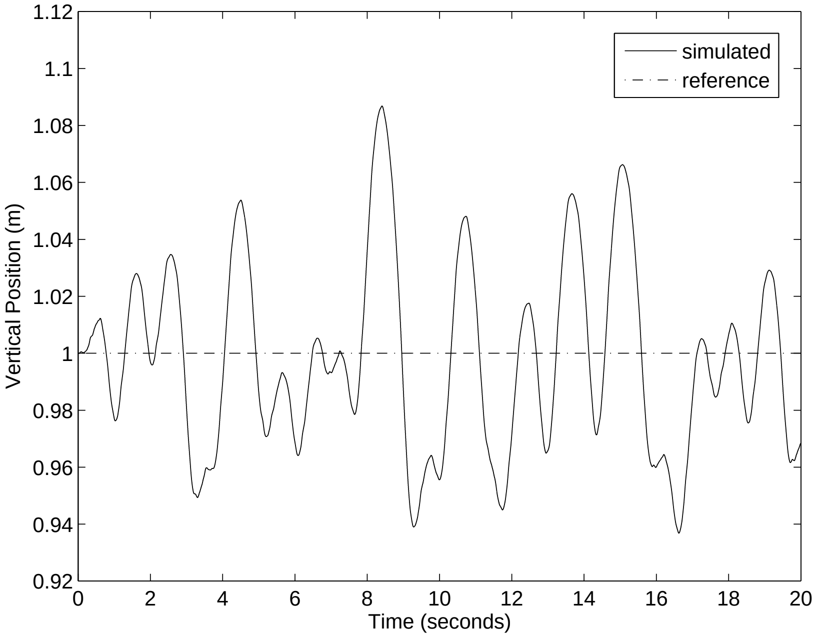


Figure 3.10. Simulated Two-Dimensional Closed-Loop System for Horizontal Flight: Vertical Position

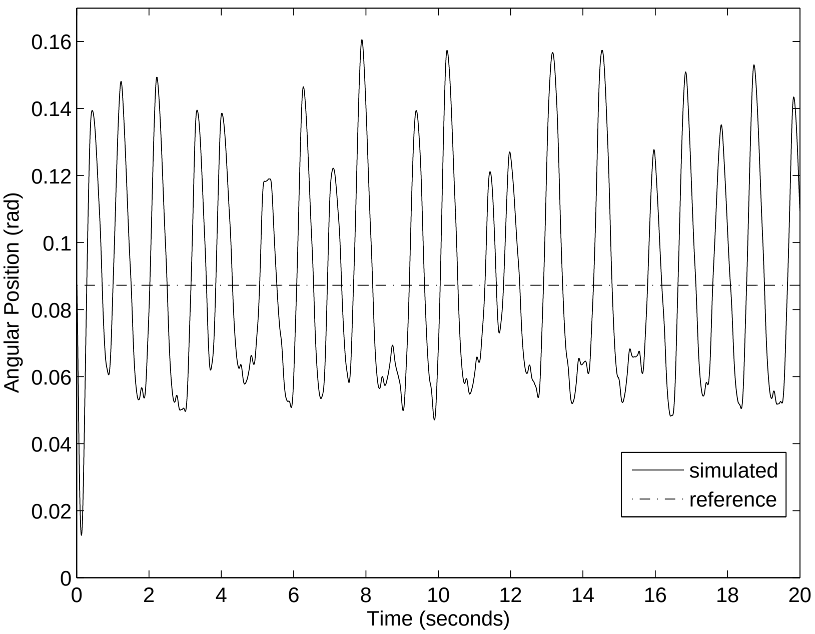


Figure 3.11. Simulated Two-Dimensional Closed-Loop System for Horizontal Flight: Angular Position

concerning angular motion and lift. Experimental analysis of the system's rotational damping, moment of inertia, and elevator (tail) effectiveness would help improve the system model. More accurate modeling of the net torque can be achieved by looking at the moment arms of individual parts of the ornithopter. Accurate determination of the lift and pitching moment of the ornithopter also seem to be required in order to achieve stability in horizontal flight. A more realistic model of the lift can be found by analyzing the lift coefficient versus angle of attack. In the future, drag also needs to be factored into the system.

If instability at high gains is indeed observable upon improvement of the ornithopter model and/or in the real system, doubts concerning the ability of the ornithopter to hover may arise. The large magnitude thrust required to maintain hover may instigate instability. It may also point to a trade-off in thrust and/or lift versus stability, which was observed and noted in [12].

To test the controller for robustness, the initial conditions were perturbed by incremental amounts until the system became unstable. Maximum deviation tolerances are summarized in Table 3.1, which show the controller has a satisfactory aptitude for recov-

Table 3.1. Summary of Controller's Maximum Deviation Tolerance

State	Maximum Deviation from Initial Conditions
$x_1 = x_0$	$0.2m \approx 7.87in.$
$x_2 = \dot{x}_0$	$2m/s \approx 78.74in./s$
$x_3 = y_0$	$0.6m \approx 23.62in./s$
$x_4 = \dot{y}_0$	$0.5m/s \approx 19.79in./s$
$x_5 = \theta_0$	$1.05rad \approx 60°$
$x_6 = \dot{\theta}_0$	$12.5664rad/s \approx -720°/s$

ery. Figures 3.12-3.17 show the ornithopter recovering from the following displacements in initial conditions: $x = x_0, \dot{x}_0 = \dot{x}_0 + 2m/s, y_0 = y_0 - 0.5m, \dot{y}_0 = \dot{y}_0 + 0.5m/s, \theta_0 = \theta_0 + 60°, \dot{\theta}_0 = \dot{\theta}_0 - 720°/s$.

One observation found while running simulations which deserves attention is the fact that once filters are added to the closed-loop system *without* the flapping included, the system becomes unstable. Though this is expected, since the addition of filters effectively changes the loop gain of the system, this does not happen to the system *with* flapping input. Perhaps the flapping force possesses a restorative attribute. More research must be done to investigate this phenomena.

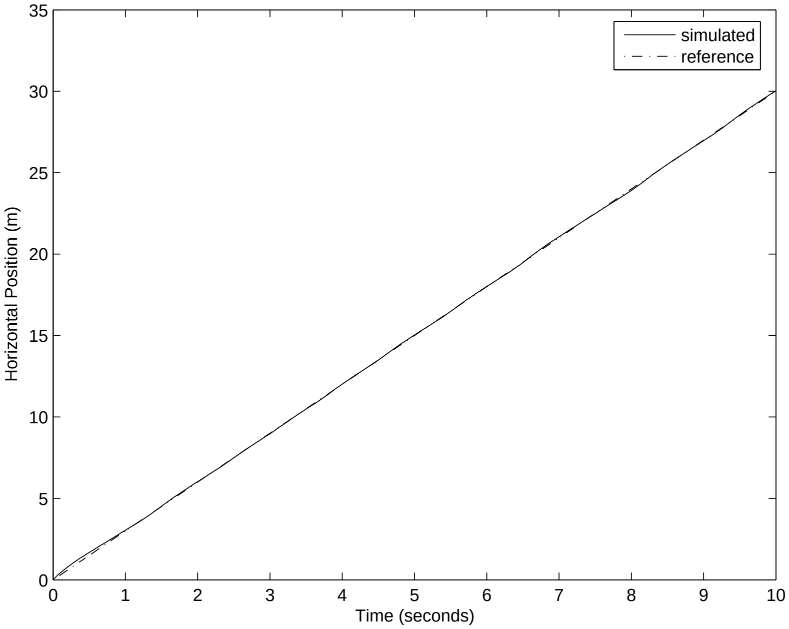


Figure 3.12. Simulation Showing Perturbations in Two-Dimensional Horizontal Closed-Loop Position for Horizontal Flight

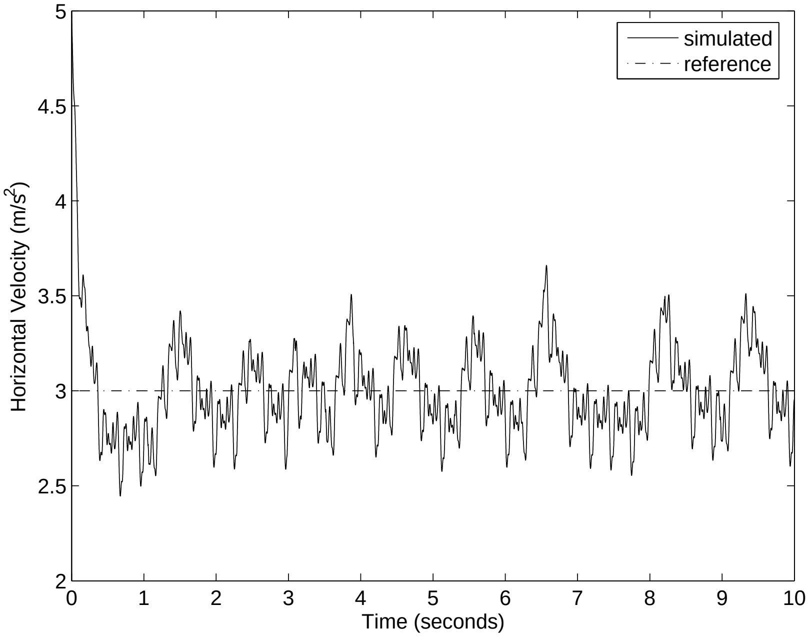


Figure 3.13. Simulation Showing Perturbations in Two-Dimensional Closed-Loop System for Horizontal Flight: Horizontal Velocity

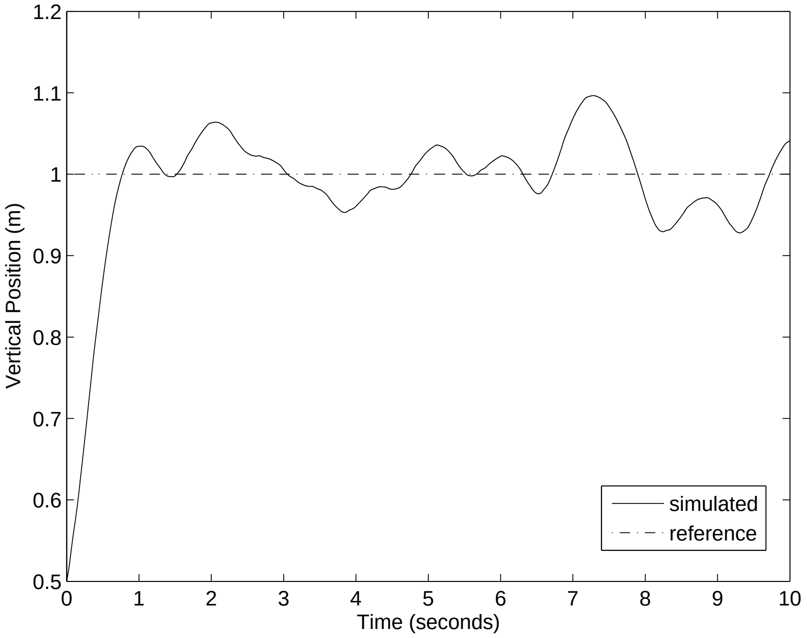


Figure 3.14. Simulation Showing Perturbations in Two-Dimensional Closed-Loop System for Horizontal Flight: Vertical Position

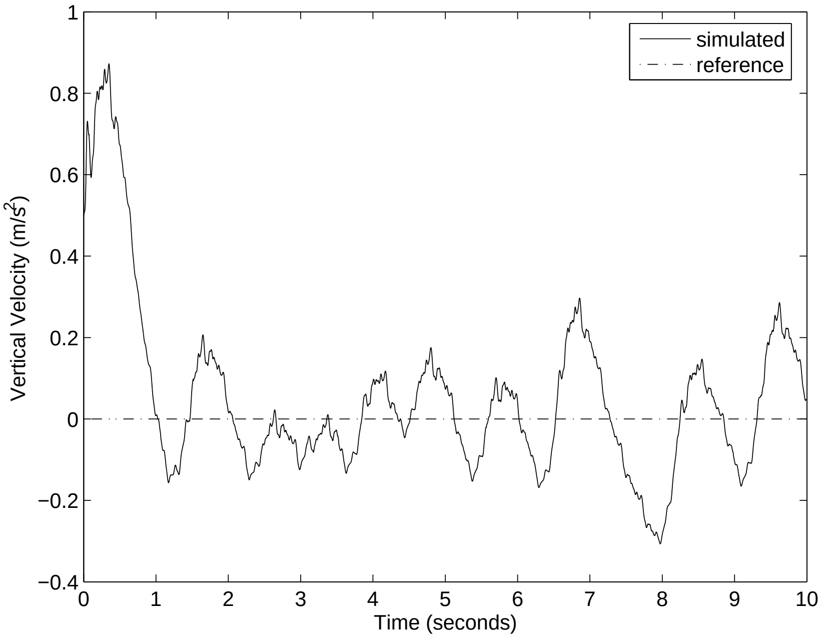


Figure 3.15. Simulation Showing Perturbations in Two-Dimensional Horizontal Closed-Loop Vertical Velocity for Horizontal Flight

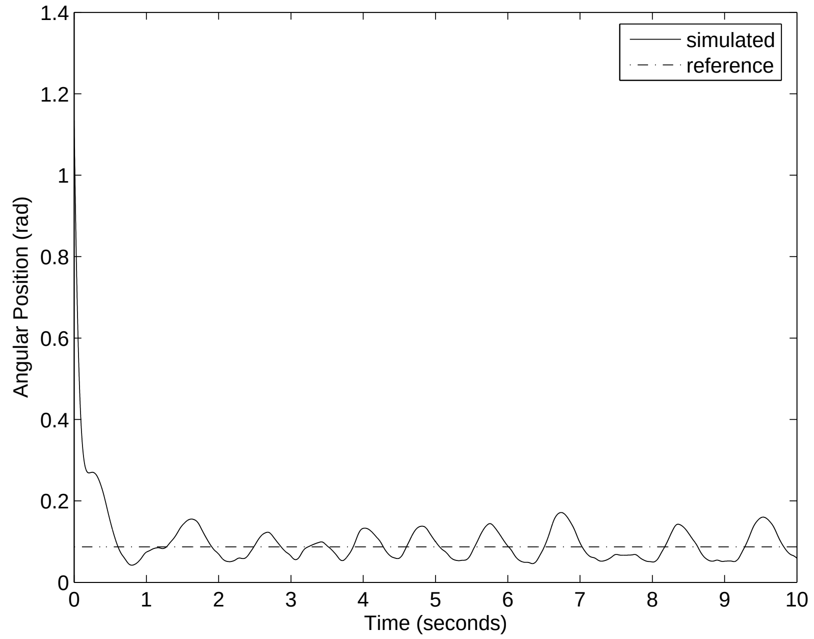


Figure 3.16. Simulation Showing Perturbations in Two-Dimensional Closed-Loop System for Horizontal Flight: Angular Position

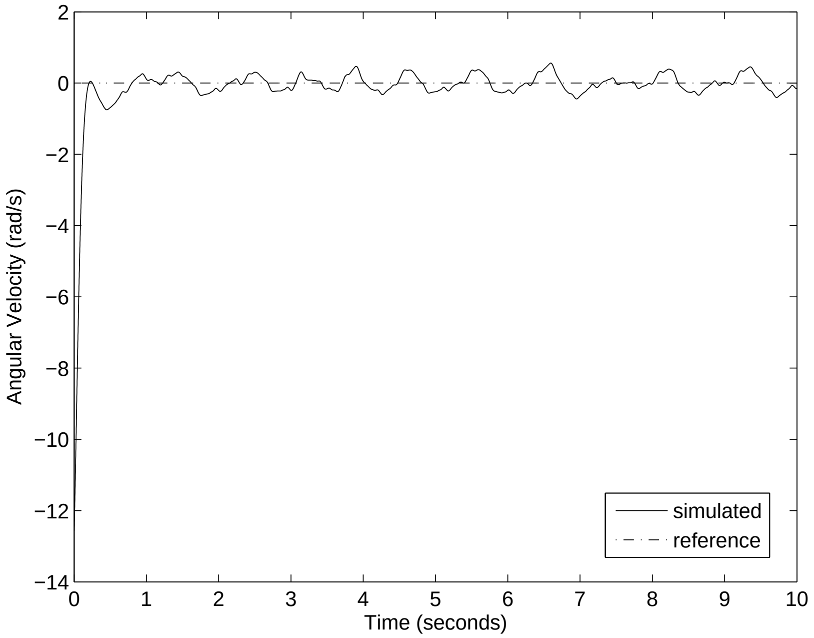


Figure 3.17. Simulation Showing Perturbations in Two-Dimensional Horizontal Closed-Loop Angular Velocity for Horizontal Flight

CHAPTER 4

CONCLUSIONS

4.1 Results

The main objective of this work was to analyze and develop control of an ornithopter MAV. This work has broader implications to control of nonlinear systems subject to nonlinear forcing inputs with multiple sinusoidal components. In this work, a wing-force characterization for the ornithopter was done based on the amplitude, frequency, and phase information found from acceleration data. This lead to the ability to extrapolate the acceleration and mean acceleration at other flapping frequencies; to the design of a transfer function describing a one-dimensional model of the ornithopter; and to design of feedback path filters based on the frequency spectrum of the forcing input. The ability to recreate a time-varying acceleration input was completed to allow for real-time simulation and control. Therefore insight into the aerodynamics and controllability of flapping wing flight is gained through explicit modeling of wings forcing input.

A nominal state-space LTI (linear time-invariant) model of the ornithopter was created for hover and steady longitudinal flight based on linearization techniques and by first studying more simple one-dimensional models. Control of these systems was completed using a simple feedback gain scheme and placing the closed-loop poles in a Butterworth pattern. Integral control was added to achieve zero steady-state error and tracking ability. Though control was obtained for both the system in hover and in steady horizontal flight, the later showed instability at high gains. This indicated a need for a more accurate model of the ornithopter and suggested a trade-off between lift and controllability.

4.2 Discussion

Our findings support those of [12] where controllability of a flapping wing MAV with wings limited to a single degree of freedom and passive rotation was analyzed by use of averaging and high frequency control. Here, controllability was shown to be ensured and a basis for use of simple linear feedback laws was posed. In this thesis, the use of simple control laws to design a controller was achieved through use of rigid body-motion equations and by analysis of the mean forcing input and torque related to the center of mass. As noted in [12], our work also suggests tradeoffs between controllability and lift generation ability.

Flapping wing MAVs offer the ability for maneuverability at low speeds, as well as to hover. It was observed in this thesis that the ornithopter did not generate enough lift to hover. Though the ornithopter mechanism had begun a steady decline in condition by the time that sensor motion data was collected, exploration summarized in [8] suggests that this indeed the case. Here, the success of a flapping wing MAV was concluded to depend on three lift generating mechanisms called *delayed stall, rotational circulation,* and *wake capture.* Similar findings are observed and studied in [3] and [5]. This required the ability of the wings to rotate: something of which our ornithopter is not capable. However, if hovering is not an end objective for this ornithopter, something of which not all natural ornithopters are capable, forward velocity was observed, both in real life and in simulations, to provide enough lift for flight.

4.3 Future Work

As noted, in future work a more precise model of the ornithopter system is needed. This includes acquiring better steady-state velocity information and experimental analysis of elevator/tail effectiveness, pitching moment, and lift. This will enable design of a better and more precise controller. Estimates of angular damping, moment of inertia, and torque constants must also be revised. A better model of the ornithopter will also include varying the angle of attack, α. Other things to be considered in the model include experimental analysis of lift-induced and parasitic drag, the coefficient of lift, and total lift. If hovering is desired, perhaps the physical design of the ornithopter must also be improved to gain more acceleration.

Once a new controller is found for the new system model, work must also be done to validate the controllers' ability on the actual ornithopter and with the integration of sensors.[1] From there, lateral aerodynamics, which include roll and yaw movements can be analyzed and incorporated into a three-dimensional system. Lateral and longitudinal

[1] Validation of the current controllers' abilities was not done in this thesis due to disrepair of the ornithopter and a lack of complete embedded control system including sensors.

control can then be the subject of future work. Beneficial insight can be gleaned from the approaches taken in [10] and [11].

APPENDIX A

ORNITHOPTER DESIGN

The Ornithopter Design Group includes Eric A Johnson, Ari Fershtut, Charles Fisher, Jared Terry, and Brent Orme. This ornithopter was developed as a Senior Design Project at the University of Utah. A design report on the ornithopter was submitted in 2006 to the 10th Annual MAV Competition at Brigham Young University (see http://www.et.byu.edu/groups/wwwmav/Tenth_MAV_Site/Home/home.htm) and is called "University of Utah MAV Ornithopter Design." Figures A.1 show a head on view of the ornithopter drawn in Pro/ENGINEER with wings fully closed and opened.

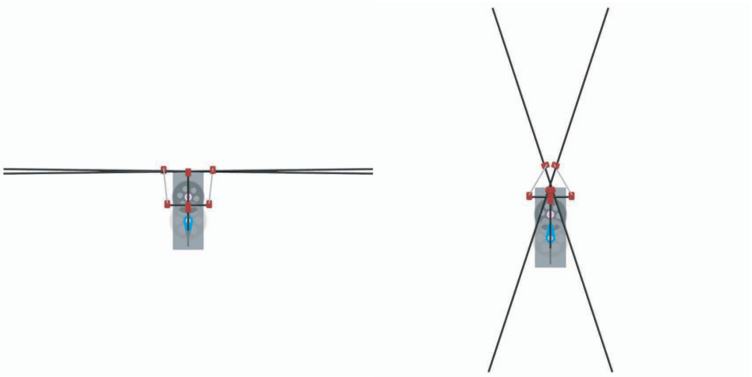


Figure A.1. Head On View of Ornithopter with Wings Closed and Opened

APPENDIX B

SIMULATION AND DATA PLOTS

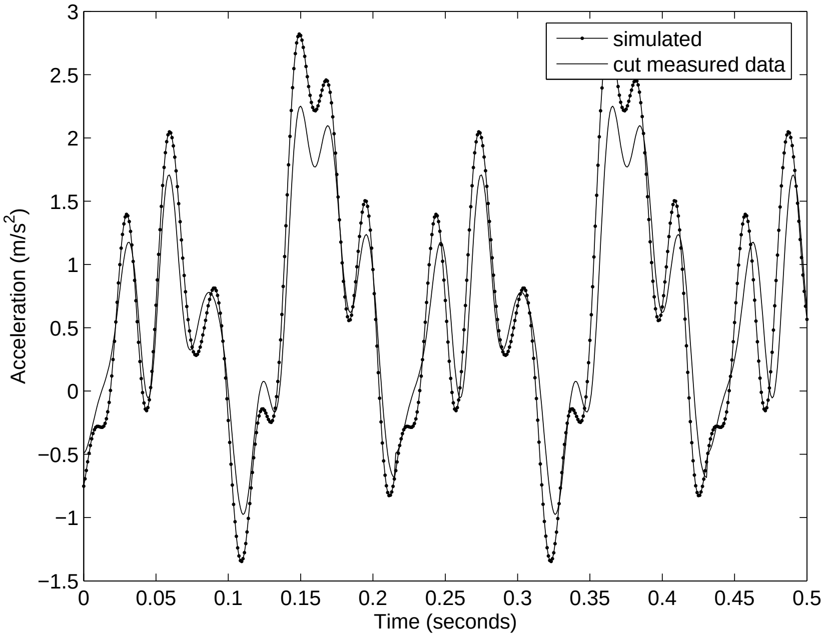


Figure B.1. Comparison of SIMULINK Simulated and Cut Acceleration (Flapping Frequency = 4.64 Hz)

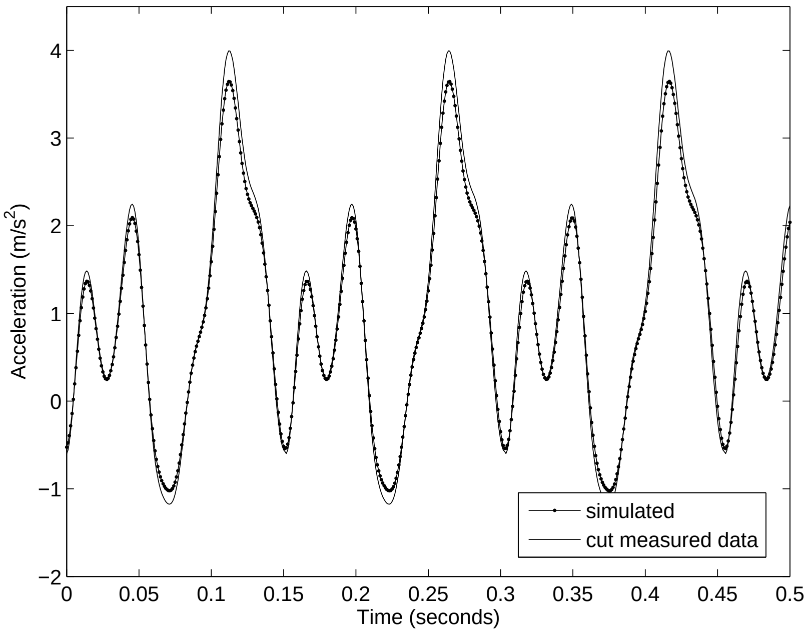


Figure B.2. Comparison of SIMULINK Simulated and Cut Acceleration (Flapping Frequency = 6.59 Hz)

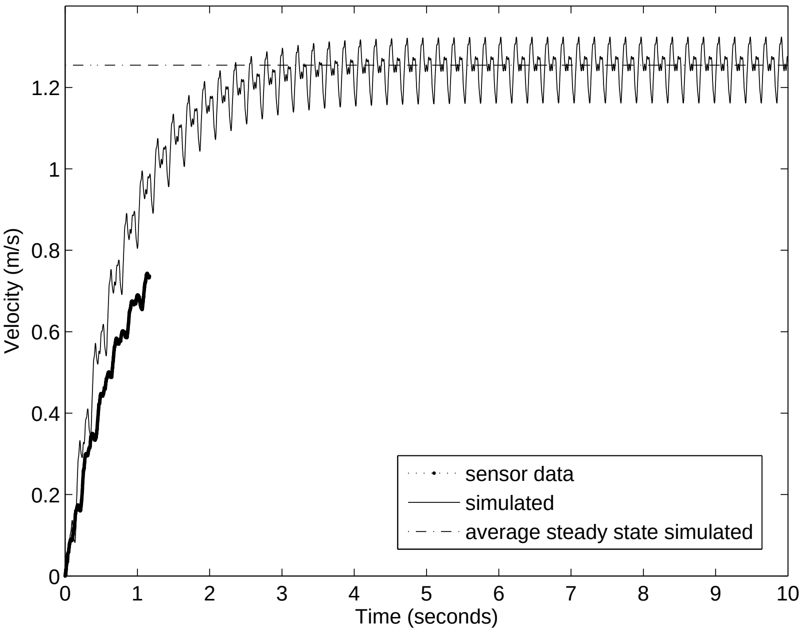


Figure B.3. Comparison of Simulated and Actual Linear Velocity (Frequency Flapping = 4.64Hz)

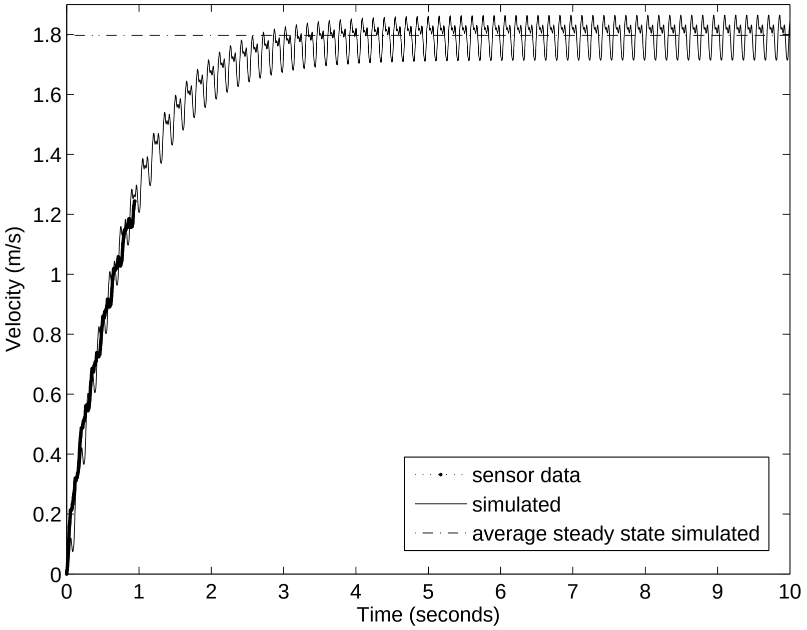


Figure B.4. Comparison of Simulated and Actual Linear Velocity (Frequency Flapping = 6.59Hz)

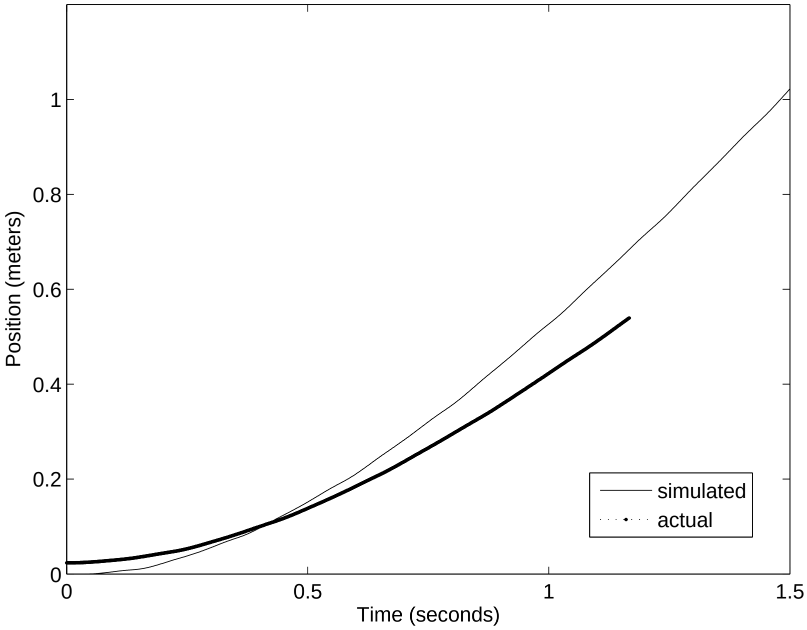


Figure B.5. Comparison of Simulated and Actual Position (Frequency Flapping = 4.64Hz)

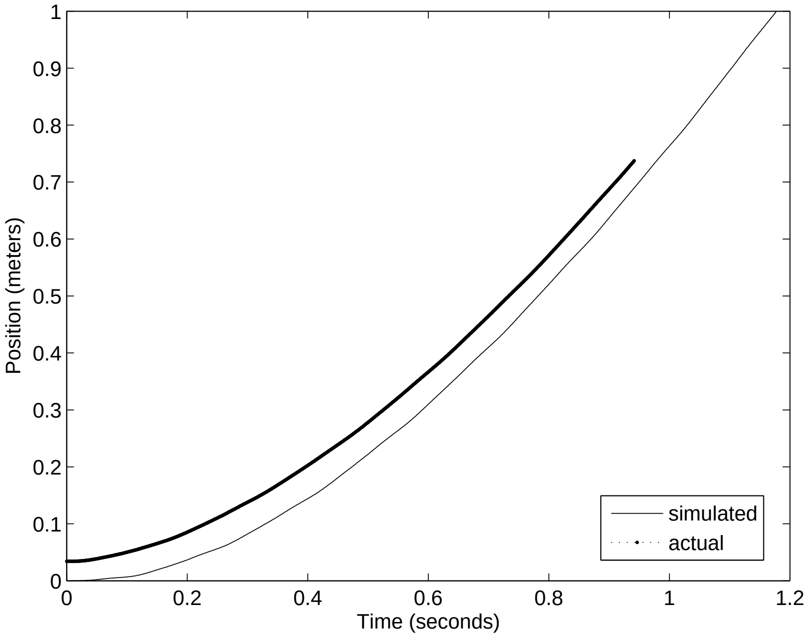


Figure B.6. Comparison of Simulated and Actual Position (Frequency Flapping = 6.59Hz)

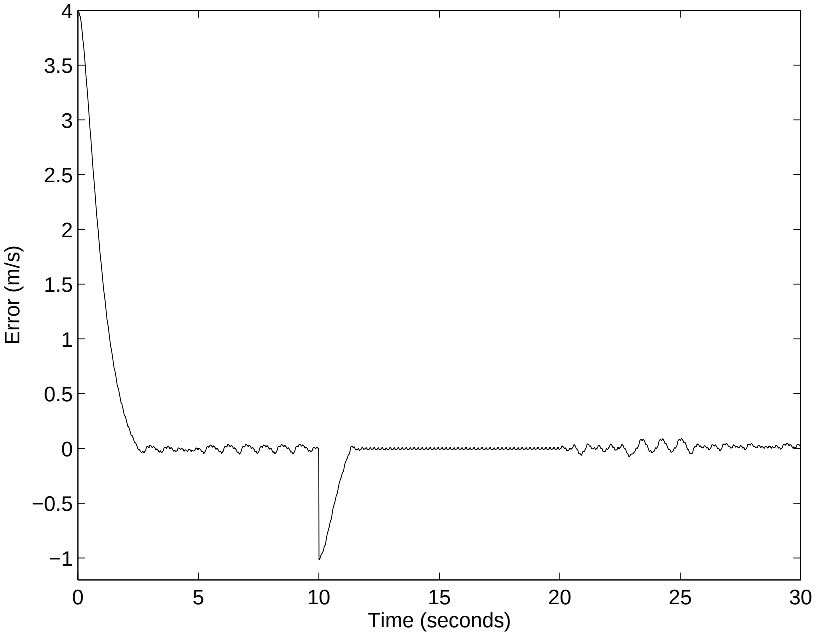


Figure B.7. Error Signal of Simulated One-Dimensional Horizontal Closed-Loop System with $K_P = 40$ and $K_I = 2$

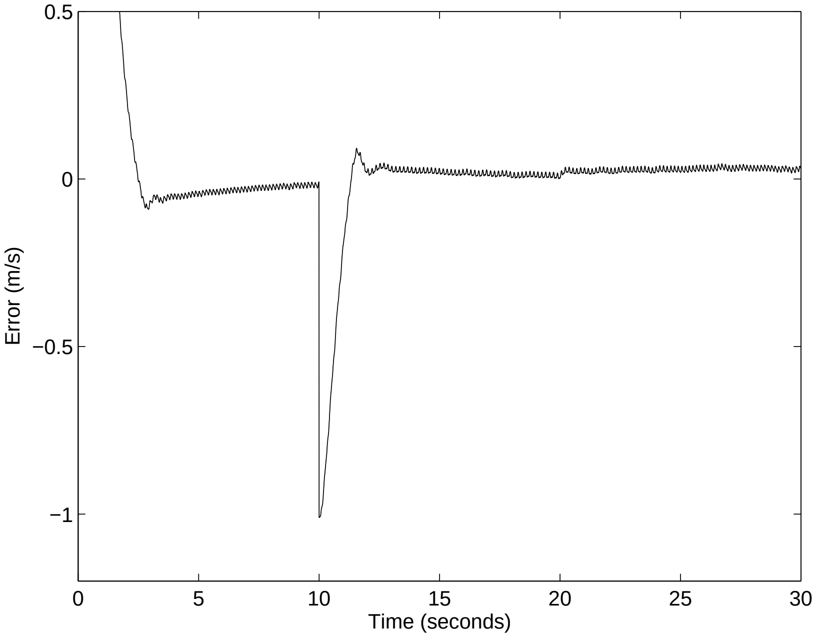


Figure B.8. Error Signal of Simulated One-Dimensional Horizontal Closed-Loop System (Updated Controller) with $K_P = 40$ and $K_I = 2$

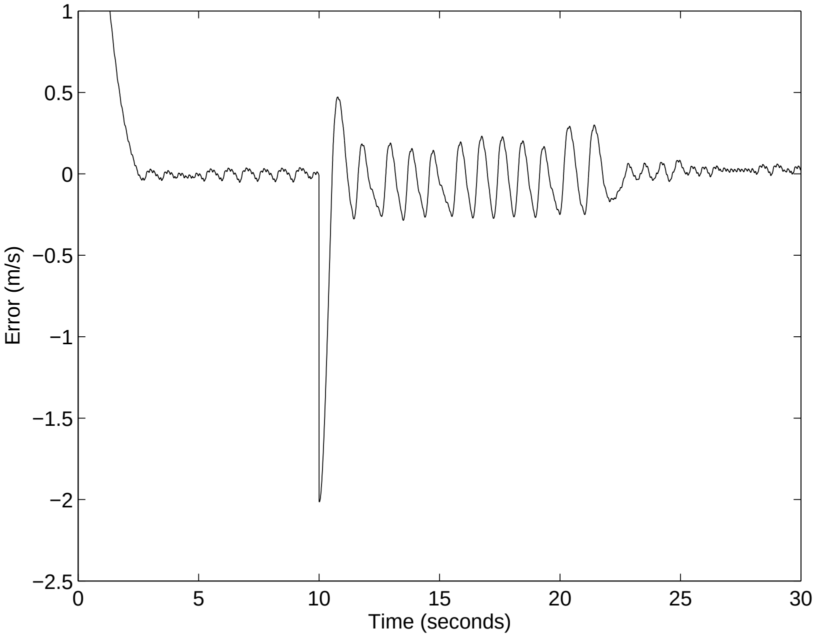


Figure B.9. Error Signal of Simulated One-Dimensional Horizontal Closed-Loop System (Showing Flapping-Gliding Behavior) with $K_P = 40$ and $K_I = 2$

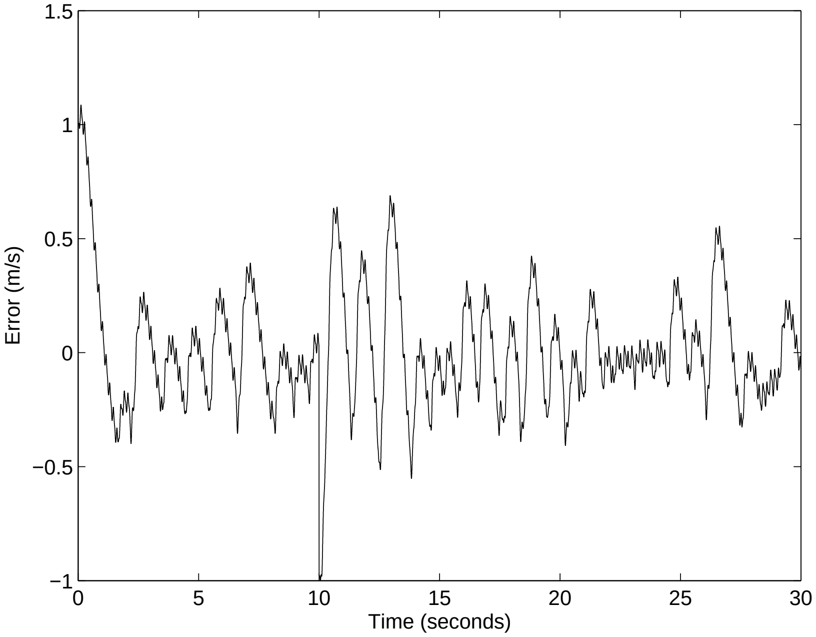
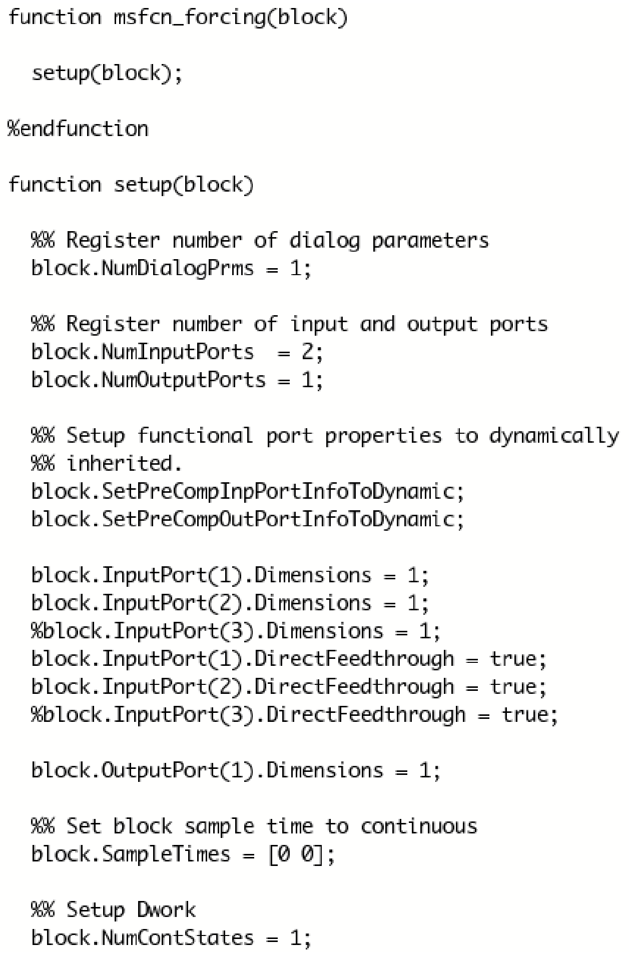


Figure B.10. Error Signal of Simulated One-Dimensional Vertical Closed-Loop System with $K_P = 40$ and $K_I = 40$

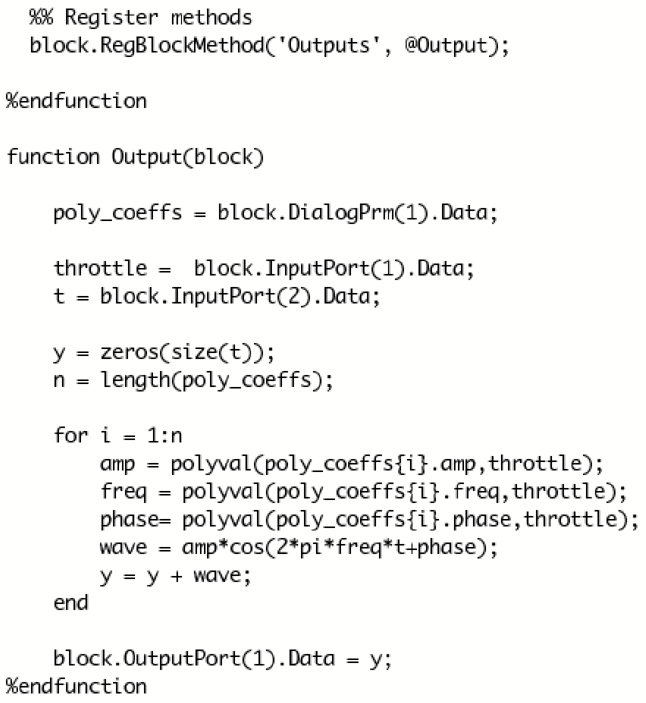
APPENDIX C

SIMULINK CODE AND FIGURES



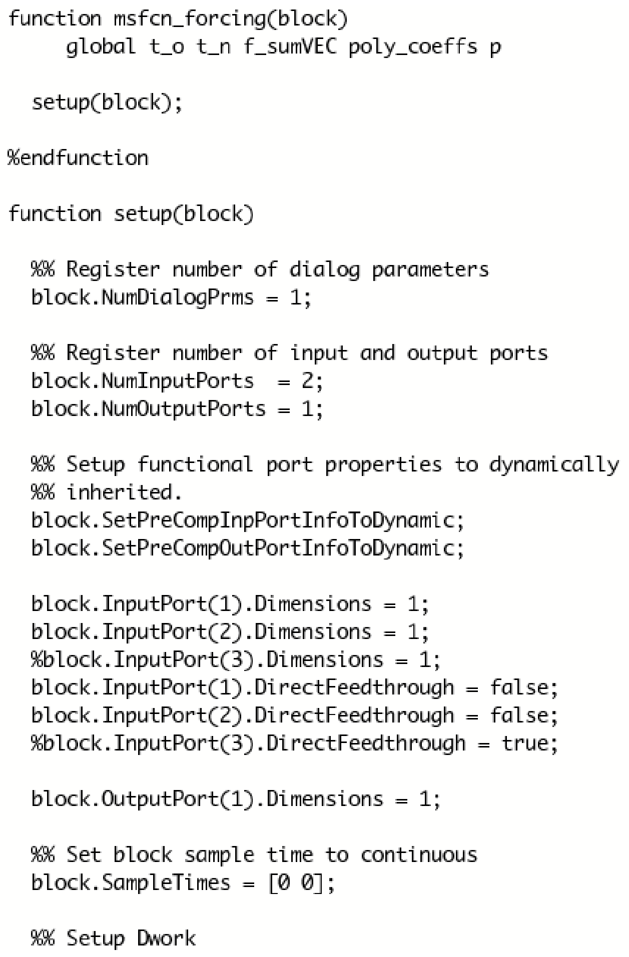
```
function msfcn_forcing(block)

  setup(block);

%endfunction

function setup(block)

  %% Register number of dialog parameters
  block.NumDialogPrms = 1;

  %% Register number of input and output ports
  block.NumInputPorts  = 2;
  block.NumOutputPorts = 1;

  %% Setup functional port properties to dynamically
  %% inherited.
  block.SetPreCompInpPortInfoToDynamic;
  block.SetPreCompOutPortInfoToDynamic;

  block.InputPort(1).Dimensions = 1;
  block.InputPort(2).Dimensions = 1;
  %block.InputPort(3).Dimensions = 1;
  block.InputPort(1).DirectFeedthrough = true;
  block.InputPort(2).DirectFeedthrough = true;
  %block.InputPort(3).DirectFeedthrough = true;

  block.OutputPort(1).Dimensions = 1;

  %% Set block sample time to continuous
  block.SampleTimes = [0 0];

  %% Setup Dwork
  block.NumContStates = 1;
```

Figure C.1. SIMULINK S-Function Code for Acceleration (Part I)



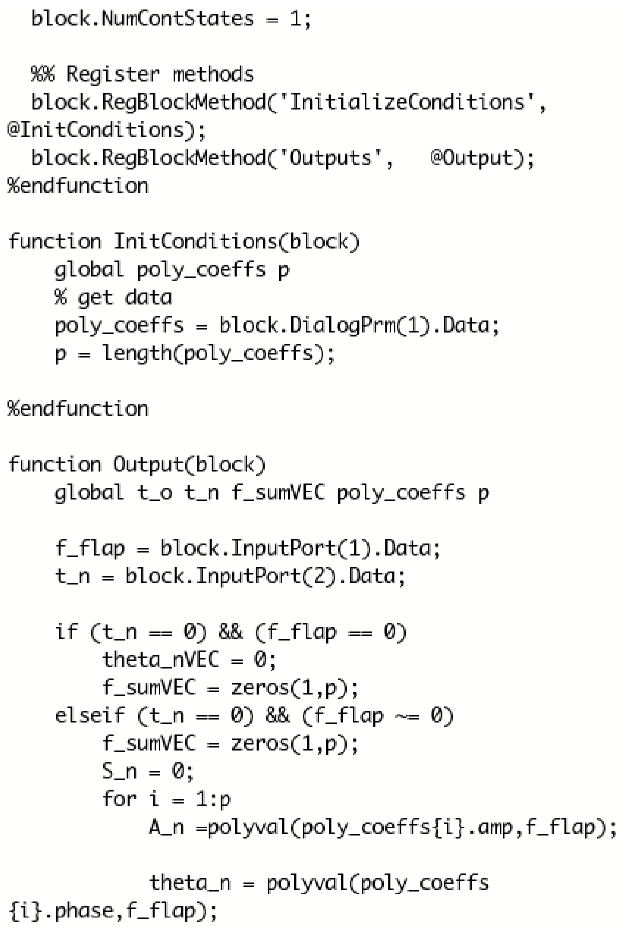
```
  %% Register methods
  block.RegBlockMethod('Outputs', @Output);

%endfunction

function Output(block)

    poly_coeffs = block.DialogPrm(1).Data;

    throttle =  block.InputPort(1).Data;
    t = block.InputPort(2).Data;

    y = zeros(size(t));
    n = length(poly_coeffs);

    for i = 1:n
        amp = polyval(poly_coeffs{i}.amp,throttle);
        freq = polyval(poly_coeffs{i}.freq,throttle);
        phase= polyval(poly_coeffs{i}.phase,throttle);
        wave = amp*cos(2*pi*freq*t+phase);
        y = y + wave;
    end

    block.OutputPort(1).Data = y;
%endfunction
```

Figure C.2. SIMULINK S-Function Code for Acceleration (Part II)



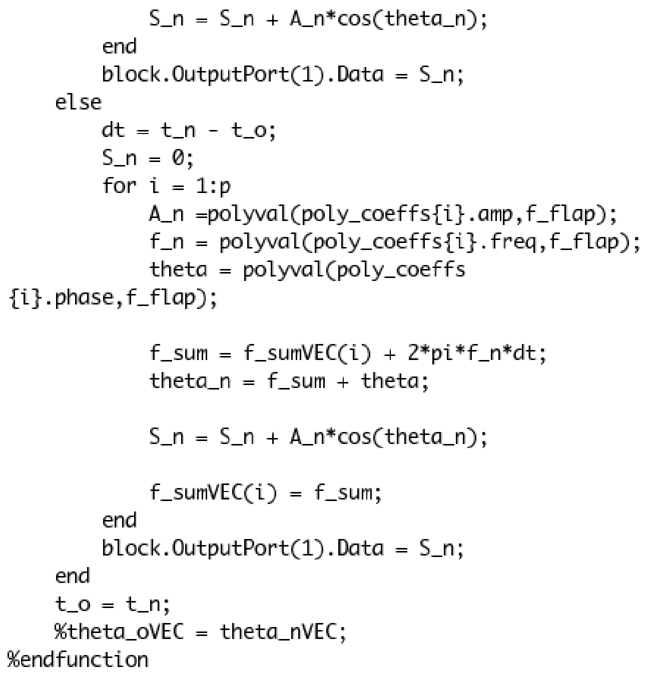
```
function msfcn_forcing(block)
     global t_o t_n f_sumVEC poly_coeffs p

  setup(block);

%endfunction

function setup(block)

  %% Register number of dialog parameters
  block.NumDialogPrms = 1;

  %% Register number of input and output ports
  block.NumInputPorts  = 2;
  block.NumOutputPorts = 1;

  %% Setup functional port properties to dynamically
  %% inherited.
  block.SetPreCompInpPortInfoToDynamic;
  block.SetPreCompOutPortInfoToDynamic;

  block.InputPort(1).Dimensions = 1;
  block.InputPort(2).Dimensions = 1;
  %block.InputPort(3).Dimensions = 1;
  block.InputPort(1).DirectFeedthrough = false;
  block.InputPort(2).DirectFeedthrough = false;
  %block.InputPort(3).DirectFeedthrough = true;

  block.OutputPort(1).Dimensions = 1;

  %% Set block sample time to continuous
  block.SampleTimes = [0 0];

  %% Setup Dwork
```

Figure C.3. SIMULINK S-Function Code for Time-Varying Acceleration (Part I)



```
  block.NumContStates = 1;

  %% Register methods
  block.RegBlockMethod('InitializeConditions',
@InitConditions);
  block.RegBlockMethod('Outputs',   @Output);
%endfunction

function InitConditions(block)
    global poly_coeffs p
    % get data
    poly_coeffs = block.DialogPrm(1).Data;
    p = length(poly_coeffs);

%endfunction

function Output(block)
    global t_o t_n f_sumVEC poly_coeffs p

    f_flap = block.InputPort(1).Data;
    t_n = block.InputPort(2).Data;

    if (t_n == 0) && (f_flap == 0)
        theta_nVEC = 0;
        f_sumVEC = zeros(1,p);
    elseif (t_n == 0) && (f_flap ~= 0)
        f_sumVEC = zeros(1,p);
        S_n = 0;
        for i = 1:p
            A_n =polyval(poly_coeffs{i}.amp,f_flap);

            theta_n = polyval(poly_coeffs
{i}.phase,f_flap);
```

Figure C.4. SIMULINK S-Function Code for Time-Varying Acceleration (Part II)



```
            S_n = S_n + A_n*cos(theta_n);
        end
        block.OutputPort(1).Data = S_n;
    else
        dt = t_n - t_o;
        S_n = 0;
        for i = 1:p
            A_n =polyval(poly_coeffs{i}.amp,f_flap);
            f_n = polyval(poly_coeffs{i}.freq,f_flap);
            theta = polyval(poly_coeffs
{i}.phase,f_flap);

            f_sum = f_sumVEC(i) + 2*pi*f_n*dt;
            theta_n = f_sum + theta;

            S_n = S_n + A_n*cos(theta_n);

            f_sumVEC(i) = f_sum;
        end
        block.OutputPort(1).Data = S_n;
    end
    t_o = t_n;
    %theta_oVEC = theta_nVEC;
%endfunction
```

Figure C.5. SIMULINK S-Function Code for Time-Varying Acceleration (Part III)

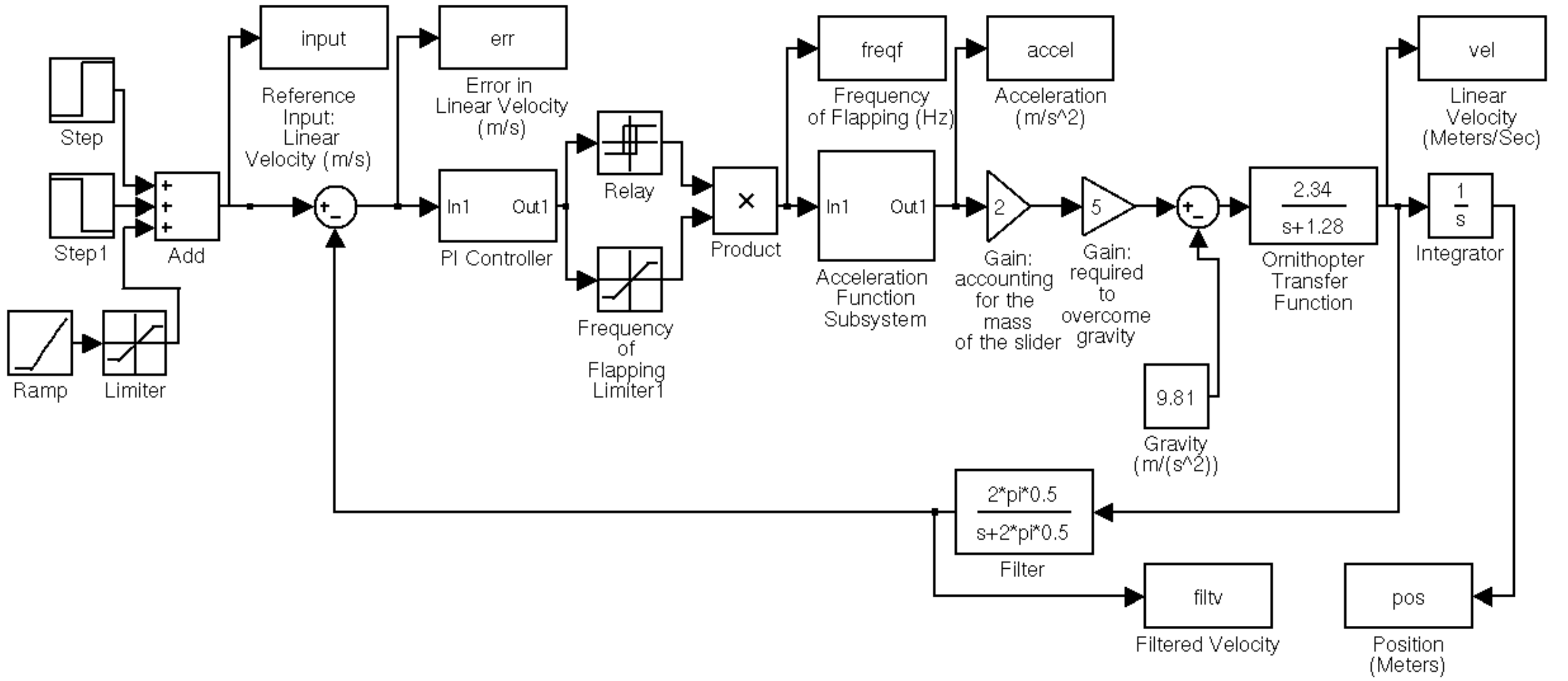

Step
Step1
Ramp
Limiter
+
+
+
Add
input
Reference Input: Linear Velocity (m/s)
+
-
err
Error in Linear Velocity (m/s)
In1
Out1
PI Controller
Relay
Frequency of Flapping Limiter1
×
Product
freqf
Frequency of Flapping (Hz)
In1
Out1
Acceleration Function Subsystem
accel
Acceleration (m/s^2)
2
Gain: accounting for the mass of the slider
5
Gain: required to overcome gravity
+
-
9.81
Gravity (m/(s^2))
2.34
s+1.28
Ornithopter Transfer Function
vel
Linear Velocity (Meters/Sec)
1
s
Integrator
2*pi*0.5
s+2*pi*0.5
Filter
filtv
Filtered Velocity
pos
Position (Meters)

Figure C.6. SIMULINK Model of Vertical Closed-Loop System

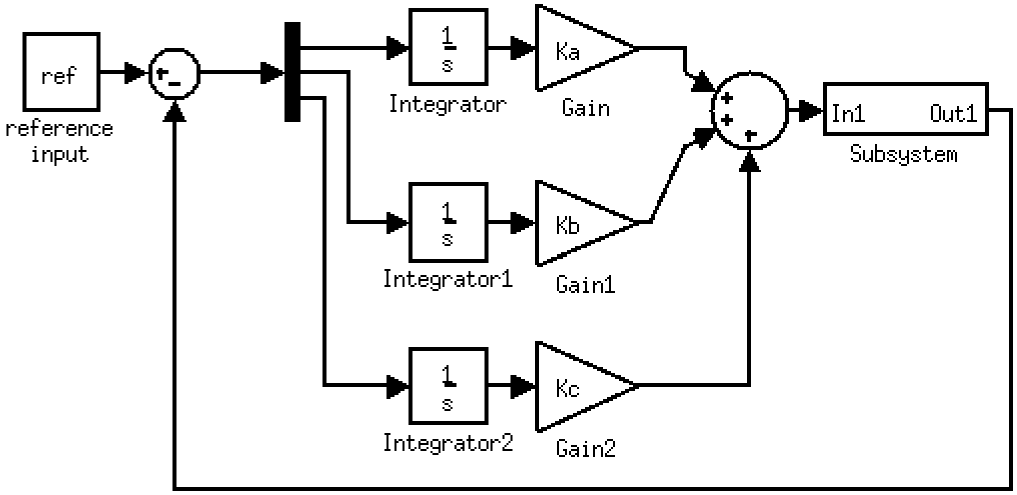


Figure C.7. SIMULINK Model of Closed-Loop State-Space System

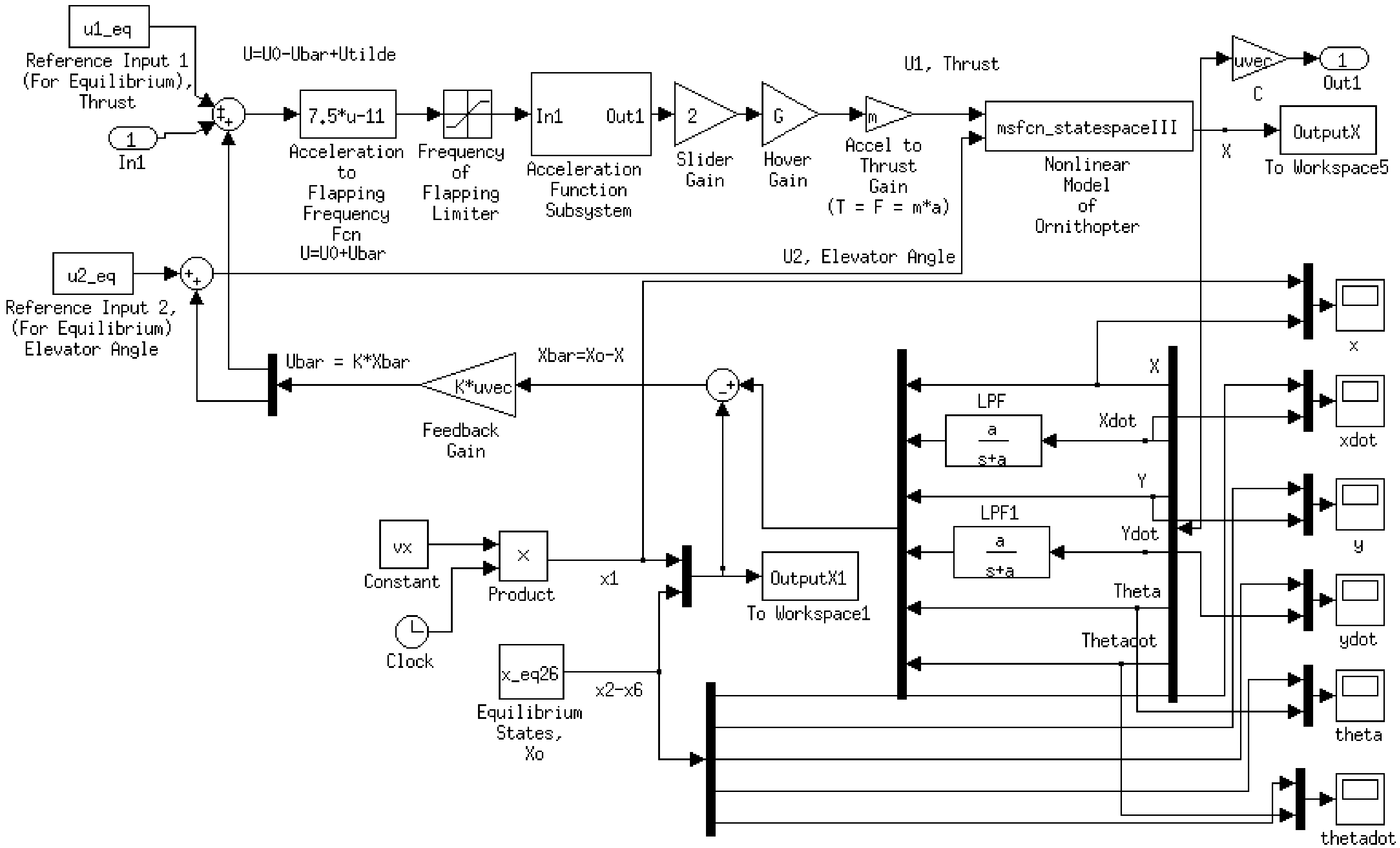


Figure C.8. SIMULINK Subsystem Model of Closed-Loop State-Space System

REFERENCES

[1] T. N. Pornsin-Sirirak, S. W. Lee, H. Nassef, J. Grasmeyer, Y. C. Tai, C. M. Ho, and M. Keennon, "Mems wing technology for a battery-powered ornithopter," 2000, pp. 799–804.

[2] R. Zbikowski, "Sensor-rich feedback control: a new paradigm for flight control inspired by insect agility," *Instrumentation & Measurement Magazine, IEEE*, vol. 7, no. 3, pp. 19–26, 2004.

[3] Z. A. Khan and S. K. Agrawal, "Force and moment characterization of flapping wings for micro air vehicle application," 2005, pp. 1515–1520 vol. 3.

[4] R. Madangopal, Z. A. Khan, and S. K. Agrawal, "Energetics based design of small flapping wing air vehicles," vol. 3, 2004, pp. 2367–2372 Vol.3.

[5] J. Tang and M. Sun, "Force and flow structures of a wing performing flapping motion at low reynolds number," *Acta Mechanica*, vol. 152, no. 1-4, pp. 35–48, 2001.

[6] R. S. Fearing, K. H. Chiang, M. H. Dickinson, D. L. Pick, M. Sitti, and J. Yan, "Wing transmission for a micromechanical flying insect," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 2, 2000, pp. 1509–1516.

[7] R. J. Wood and R. S. Fearing, "Flight force measurements for a micromechanical flying insect," vol. 1, 2001, pp. 355–362 vol.1.

[8] J. Yan, R. J. Wood, S. Avadhanula, M. Sitti, and R. S. Fearing, "Towards flapping wing control for a micromechanical flying insect," vol. 4, 2001, pp. 3901–3908 vol.4.

[9] J. Yan and R. S. Fearing, "Wing force map characterization and simulation for the micromechanical flying insect," vol. 2, 2003, pp. 1343–1349 vol.2.

[10] X. Deng, L. Schenato, and S. S. Sastry, "Model identification and attitude control for a micromechanical flying insect including thorax and sensor models," vol. 1, 2003, pp. 1152–1157 vol.1.

[11] L. Schenato, W. C. Wu, and S. Sastry, "Attitude control for a micromechanical flying insect via sensor output feedback," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 1, pp. 93–106, 2004.

[12] L. Schenato, D. Campolo, and S. Sastry, "Controllability issues in flapping flight for biomimetic micro aerial vehicles (mavs)," vol. 6, 2003, pp. 6441–6447 Vol.6.

[13] R. Bellman, J. Bentsman, and S. Meerkov, "Vibrational control of nonlinear systems: Vibrational stabilizability," *Automatic Control, IEEE Transactions on*, vol. 31, no. 8, pp. 710–716, 1986.

[14] J. Bentsman, "Vibrational control of a class of nonlinear systems by nonlinear multiplicative vibrations," *Automatic Control, IEEE Transactions on*, vol. 32, no. 8, pp. 711–716, 1987.

[15] L. Gurvits, "Averaging approach to nonholonomic motion planning," 1992, pp. 2541–2546 vol.3.

[16] S. Weibel, J. Baillieul, and T. J. Kaper, "Small-amplitude periodic motions of rapidly forced mechanical systems," vol. 1, 1995, pp. 533–539.

[17] J. Baillieul and S. Weibel, "Scale dependence in the oscillatory control of micromechanisms," in *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, vol. 3, 1998, pp. 3058–3063 vol.3.

[18] J. Baillieul, "Averaging methods for force controlled and acceleration controlled lagrangian systems," vol. 2. Institute of Electrical and Electronics Engineers Inc., Dec 2000, pp. p 1266–1272.

[19] F. Bullo, "Averaging and vibrational control of mechanical systems," *SIAM Journal on Control and Optimization*, vol. 41, no. 2, pp. p 542–562, 2003.

[20] N. E. Leonard, "Periodic forcing, dynamics and control of underactuated spacecraft and underwater vehicles," vol. 4, 1995, pp. 3980–3985 vol.4.

[21] N. E. Leonard and P. S. Krishnaprasad, "Averaging for attitude control and motion planning," 1993, pp. 3098–3104 vol.4.

[22] S. Martinez, J. Cortes, and F. Bullo, "Analysis and design of oscillatory control systems," *Automatic Control, IEEE Transactions on*, vol. 48, no. 7, pp. 1164–1177, 2003.

[23] S. N. Singh, A. Simha, and R. Mittal, "Biorobotic auv maneuvering by pectoral fins: inverse control design based on cfd parameterization," *Oceanic Engineering, IEEE Journal of*, vol. 29, no. 3, pp. 777–785, 2004.

[24] C. Chen, *Linear System Theory and Design*, 3rd ed. New York, New York: Oxford University Press, 1999.