A System for Landing an Autonomous Radio Controlled Helicopter on Sloped Terrain

by

Jacqueline M. Moore

S.B., Aeronautical and Astronautical Engineering
Massachusetts Institute of Technology, 1992

Submitted to the
Department of Aeronautics and Astronautics in
Partial Fulfillment of the Requirements for the Degree of

Master of Science
in Aeronautical and Astronautical Engineering
at the

Massachusetts Institute of Technology

February 1994

Signature of Author _____
Department of Aeronautics and Astronautics
11 January 1994

Certified by _____
Associate Professor Christopher G. Atkeson
Thesis Supervisor

Certified by _____
Associate Professor John Hansman
Department of Aeronautics and Astronautics Reader

Accepted by _____
Professor Harold Y. Wachman
Chairman, Department Graduate Committee

A System for Landing an Autonomous Radio Controlled Helicopter on Sloped Terrain

by

Jacqueline M. Moore

Submitted to the Department of Aeronautics and Astronautics on 11 January 1994
in partial fulfillment of the requirements for the Degree of
Master of Science in Aeronautical and Astronautical Engineering

ABSTRACT

An autonomous landing system is designed to land a helicopter-type Unmanned Aerial Vehicle (UAV) on a target on a sloped surface. The UAV is a Radio Controlled (RC) model helicopter. In flight, a Linear Quadratic Gaussian (LQG) compensator is used to stabilize the helicopter in hover.

The landing system lowers the helicopter to the target from directly above the target by following a collective command input which reduces the thrust of the helicopter. A multiple model adaptive controller (MMAC) is activated when the helicopter is in hover and adds an additional cyclic input proportional to the collective command when the residuals of the rate gyro states in the estimators indicate that the helicopter is probably landing on a slope to prevent reaction torques that would drive the helicopter into the slope.

Comparing the landing accuracy of the MMAC landing system to the landing accuracy of the LQG hover compensator following the same collective command input over a variety of slopes, the MMAC reverts to the LQG hover compensator on slopes less than 2° in magnitude and performs about the same as the hover compensator on slopes less than 6°. For slopes greater than or equal to 12°, both the hover compensator and the MMAC saturate the cyclic command. The MMAC lands closer to the desired target position than the LQG hover compensator on slopes ranging from 6° to 12° in magnitude. So if only slopes less than 6° are encountered, the LQG hover compensator is recommended because it yields the same performance and is easier to implement. If implemented properly, the MMAC yields better performance at the steeper slopes while maintaining the performance of the LQG hover compensator at the slopes under 6°.

Acknowledgments

Ben, a best friend and arch nemesis, thanks for everything. Ben performed much of the sensor and vehicle research and instructed me on the repair of the helicopters. He brought me into the project and managed the entire Helicopter Project at the Artificial Intelligence Lab. He also served as the test pilot throughout all of the experiments and saved many vehicles from demise.

Elaine, Heidi, Matt, Pete, Monte and Mark, thanks for keeping in touch. I'm finally getting out of this place!

Mike O'C. and Joe, thanks for putting up with me and my frustrations and reminding me of the other sides of life.

Thanks to Prof. Chris Atkeson, the character who funded my graduate school endeavor and gave Ben and I a lot of room to do what we wanted with the project. Although I don't know if we ever fully understood each other, he was always there to provide useful suggestions when I was really flailing.

Thanks to Prof. John Hansman for reading my thesis and providing some of the technical terms for what I was trying to explain.

Dave, thanks for letting me use your printer!

Mike S., Anne, Scott, Jerry, and Chris B., your enthusiasm is contagious. Special thanks to Anne who shared similar frustrations and kept me from going bonkers. She's also given the 68332, the color vision system, and computer science in general personalities in her explanations that I'll never forget.

Mom, Dad, Harry, Teresa and Tom, thanks for being a family of which I'm happy to be a part.

Table of Contents

List of Figures

1 Introduction and Motivation of Thesis

Unmanned aerial vehicles (UAV's) have been developed for missions deemed too hazardous for human pilots because UAV's are considered disposable. Therefore, UAV's should be expected to handle more rugged conditions than human piloted vehicles can. Both human piloted and autonomous vehicles have difficulty landing on uneven surfaces. Human pilots can remotely land radio controlled (RC) helicopters on hills with a ±12° slope; however, in a hazardous situation the range of the radio system and/or the monitoring system may be too short for the safety of the human operator. Consequently, a completely autonomous UAV which is able to land successfully in such conditions is desired.

The helicopter project at the Artificial Intelligence lab at MIT is developing relatively low cost platforms (compared to Canadair's CL-227 Sentinel® and Sikorsky's Cypher™) for testing learning algorithms. The vehicle is a hobbyist radio controlled (RC) model helicopter under computer control. This thesis studies different strategies for autonomously landing the helicopter on slopes ranging from 0°-12° using standard control techniques.

1.1 Dynamics of a Sloped Landing

Figure 1 shows four different phases when landing a helicopter on a sloped surface. The *hover phase* is when the helicopter is hovering above the target. During the *descent phase*, the helicopter moves from the hover position above the ground down into ground effect until the helicopter just touches the ground. The next phase is the *settle phase* which covers the time from when the helicopter first touches the ground until the helicopter completely rests on the ground. When landing on level ground, ideally all points of the gear touch at about the same time and the settle phase is relatively short. However, when landing on a sloped surface, one part of the gear touches the high side of

the slope before the rest of the gear comes into contact with the slope. In the case of a sloped surface, the settle phase begins when any part of the gear initially touches the ground and ends when the entire gear is in contact with the ground. Even when the gear is in full contact with the ground, the helicopter blades are still spinning but they are not producing enough lift to overcome the weight of the vehicle. During *spin down,* the thrust is reduced until the motor is shut off.

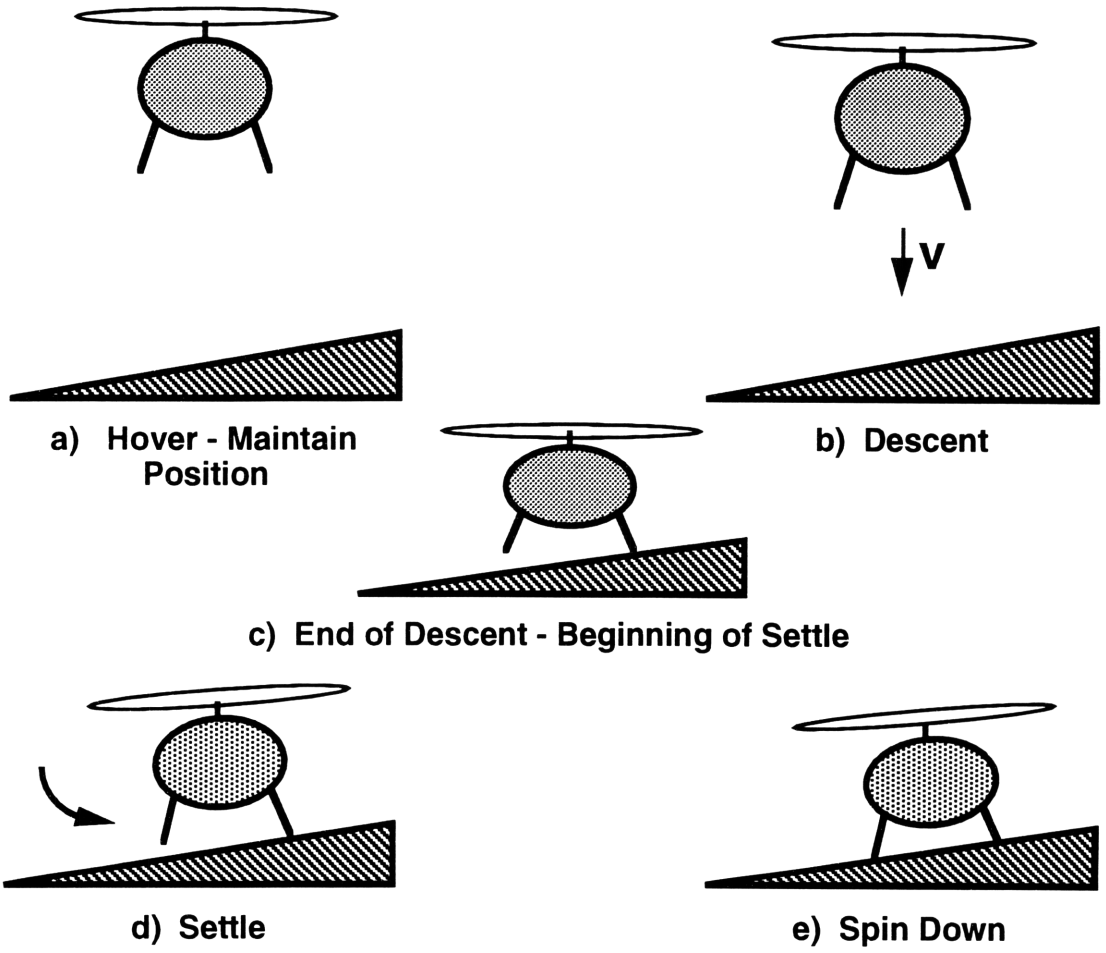


Figure 1
Landing Phases

During the descent, there are a variety of ways the vehicle can approach the ground for a landing. One possible approach is that the helicopter approaches the desired target from a downhill location with predominately a horizontal velocity and very little vertical velocity. Figure 2a shows that the ground would touch the gear from the side

with a reaction force equivalent to the force with which the helicopter hit the slope. Since the velocity of the helicopter was predominantly horizontal, the reaction force is also predominantly horizontal and acting at the point of contact and not through the center of mass of the helicopter. This produces an external torque on the helicopter about the point of contact which brings the helicopter closer to the slope. Figure 2b shows that if the rotation of thehelicopter is too large, the force due to gravity no longer acts through the base of the helicopter. The helicopter tips over, and this situation is known as dynamic rollover. Dynamic rollover happens anytime the net torque acting on the vehicle forces the helicopter to tip over, and it is not limited to a horizontal approach. However, the horizontal approach creates a torque in a undesirable direction and aggravates an already delicate balance.

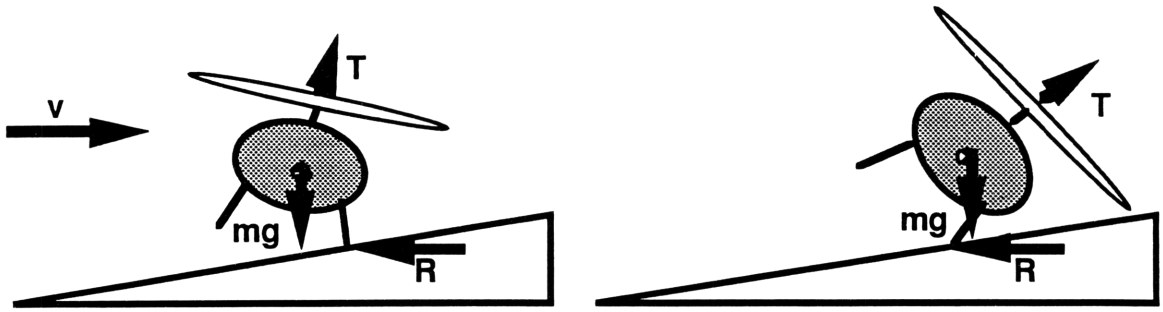
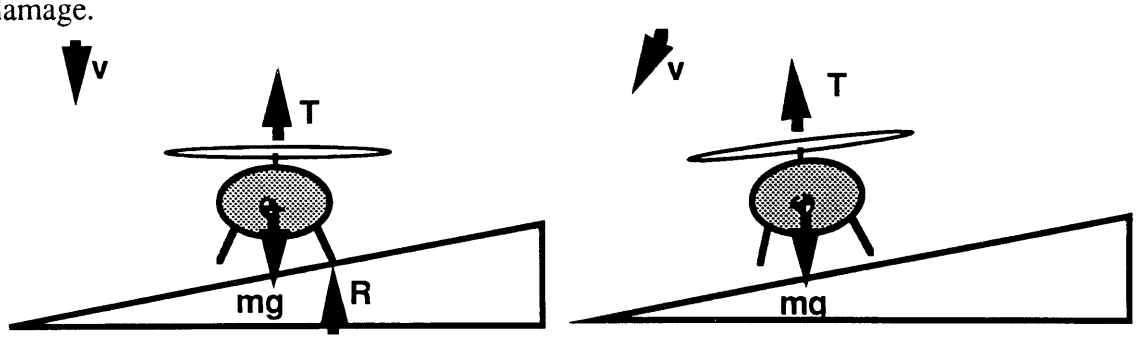


Figure 2
Horizontal Approach and Dynamic Roll Over

The RC helicopter used in this study can not statically take a hill from the side any larger than 45° without the blades hitting the ground and destroying the vehicle. However, the limitations preventing a dynamic rollover situation depend on the geometry of the landing gear. The original gear that comes with the X-Cell 60 can take a maximum attitude roll angle of 45°, but the modified gear (see chapter 2) can take an attitude angle of 60°. Too much overshoot in a controller using a side approach can cause dynamic roll over and thus destroy the vehicle.

In a different approach shown in Figure 3, the helicopter descends with virtually no horizontal velocity so that the bottom of the gear touches the ground first. If the

helicopter were landing on a sloped surface, the gear would touch the high side of the ground first and the resulting torque would lead to a rotation of the vehicle and a horizontal component of the thrust vector. With preservation of the vehicle being a higher priority than maintaining target position, the vertical landing strategy is used in this study because the natural tendency is to move away from the hill preventing possible damage.



a) Reaction Force Causes Torque Downhill **b) Loss of Target Position**

Figure 3
Vertical Approach

Assuming a robust hover controller, the helicopter is able to maintain position over its target given typical disturbances. When the vehicle touches the ground during the settle phase, the external torques applied to the helicopter create angular accelerations that are not predicted by the model of the helicopter in hover. These angular accelerations tilt the thrust of the helicopter. The component of the thrust in the horizontal direction accelerates the helicopter in the horizontal direction which integrates twice to a loss of desired position. Once the skids of the vehicle are in full contact with the ground, the controllability of the x and y position of the vehicle is severely attenuated. This study focuses on making corrections to achieve the desired final target position during the settle phase of landing.

The velocity with which the helicopter descends onto the slope, the characteristics of the ground, and the helicopter suspension system determine whether the helicopter bounces, slides or comes to a rest on the slope. If the collision between the helicopter and

the slope upon landing is perfectly elastic, a bounce-free landing would be virtually impossible. However, the collision is not perfectly elastic, so some energy is absorbed in the landing. The kinetic energy of the helicopter just prior to landing is

$$KE = \frac{1}{2} m v_{descent}^2 \qquad [1]$$

Reducing the descent velocity reduces the bouncing effect of the landing of the helicopter. A harder surface such as concrete has a more elastic quality than a softer surface such as grass, so it is important to descend more slowly upon a harder surface than necessary for a softer surface.

On a sloped surface, the helicopter may also slip down the hill depending on the friction characteristics between the ground and the helicopter gear. Figure 4 shows the forces acting on a mass resting on a slope. When a helicopter lands on a slope (during the spin down phase), normal force acting on the helicopter is changing as the thrust vector of the helicopter is reduced.

$$N = mg\cos\alpha - T\cos\theta \qquad [2]$$

where α is the angle between level ground and the slope and θ is the angle of the rotor plane with respect to the helicopter vertical (see Figure 5). As long as the frictional force is greater than the force due to the acceleration pulling the helicopter down the hill

$$\mu N \geq mg\sin\alpha - T\sin\theta \qquad [3]$$

where μ is the coefficient of friction, no slip occurs. The helicopter should not try to land on slopes with low coefficients of friction (like ice, wet grass or mud) on which it cannot sit at rest.

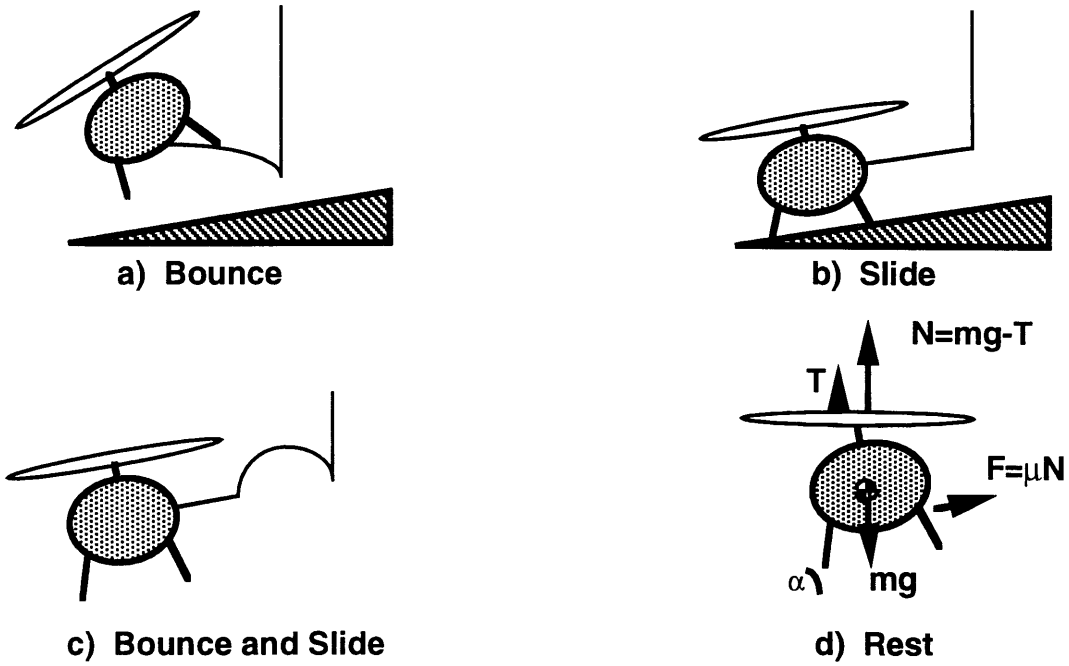


Figure 4
Ground Interaction

During the settle phase, the dynamics between attitude rates and angles differ from the dynamics in the air. Figure 5 shows that an angular acceleration to the right does not carry the vehicle to the right as it would in the air because the helicopter hits the slope. Angular accelerations away from the hill still correspond to a translation in position away from the hill, but the helicopter simply cannot fly through the ground. Trying to maintain the vehicle's position, a controller (human or computer) tends to steer into the slope. As the collective pitch is reduced, the vehicle tends towards a resting point on the slope. To prevent loss in position, roll cyclic is increased to compensate for the rotating thrust vector. However, the cyclic command is physically limited to prevent the separation of the rotor head from the body of the vehicle. When the cyclic is held at this limit, a condition known as mast bumping or droop stop pounding[1] results which causes severe vibrations on the vehicle. When this situation occurs, the collective should

[1]Raymond W. Prouty. Helicopter Performance, Stability, and Control, p. 653.

be increased and a more suitable landing location with a less severe grade should be found.

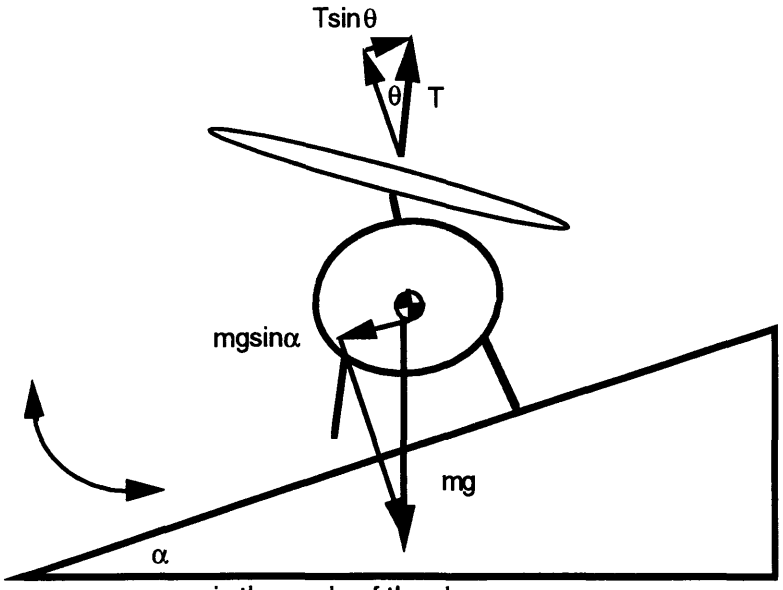


Figure 5
Mast Bumping

1.2 Specialization of the Problem

Given the dynamics of landing on a sloped surface, this thesis specializes the problem into landing on smooth, sloped surfaces without wind disturbances using a vertical approach. The intent is to study the effects of a grade on the ability of the vehicle to land on its target. Smooth is a necessary requirement since an uneven surface may mask the direction of the slope by initially perturbing the helicopter in a direction different than that of the slope. Also, since the experimental setup is in a controlled indoor environment, effects due to wind gusts are not considered.

As discussed in section 1.1, the loss of position is the result of the unaccounted for external forces due to the ground interaction during the settle phase of landing. There are a wide variety of possible solutions to correct this loss of target position. One possible

dynamic solution is if the slope and friction of the slope are known, the helicopter targets a position slightly uphill of the desired position during the descent phase and slide into the target position during the settle phase of the landing. Since the specific characteristics of the terrain are generally unknown, this study has limited its scope to strategies which can land the helicopter on a variety of slopes without prior knowledge of the terrain. Another, simpler solution is to simply shutoff of the collective at the beginning of the settle phase thus eliminating the thrust which gives rise to the horizontal accelerations. The problem with this method is when the thrust is cut off, the helicopter on a sloped surface drops abruptly to the ground which may cause the vehicle to bounce or may damage the helicopter or its sensors.

Another strategy is to identify a new model that takes into account the ground interaction of the vehicle on a slope. Ideally, this model characterizes the dynamics from cyclic and collective inputs to attitude rates through attitude angles, velocities and position states. The vehicle in this regime is highly non-linear and uphill motions do not mirror downhill motions. Using a compensator based on this model, multiple model techniques can generate the appropriate commands based on the probability of the vehicle being in the valid region of a particular model. The advantage is that the model would take into account the interaction of the helicopter with the ground and better performance would be expected. The disadvantage is the overhead involved in identifying this model.

The solution studied is to correct for loss of position during the settle phase when landing with a predominantly vertical velocity on an unknown slope. However, it is assumed that the slope is capable of being landed on which means that it can have a grade less than $\pm 12°$. Only slopes to the left or right of the helicopter are studied since any algorithms developed for roll cyclic control can be extended to the pitch cyclic control as long as the pitch cyclic controller has an additional constraint which prevents a tail boom strike.

2 Experimental Setup and Hover Compensator

An overall view of the helicopter testing area is shown in Figure 6. In the center of the area is the RC model helicopter. A tether carries power from the ground up to the helicopter while a data tether carries information from on board sensors to processors on the ground which handle all control calculations of the helicopter. A steel safety cable attached to the floor prevents the helicopter from hitting the ceiling should it ever go out of control. This setup requires two people to operate it: one person is the test pilot who has a manual override and operates the power switch to the motor while the second person issues commands like take-off, land, and record data to the computer and monitors the sensor operation.

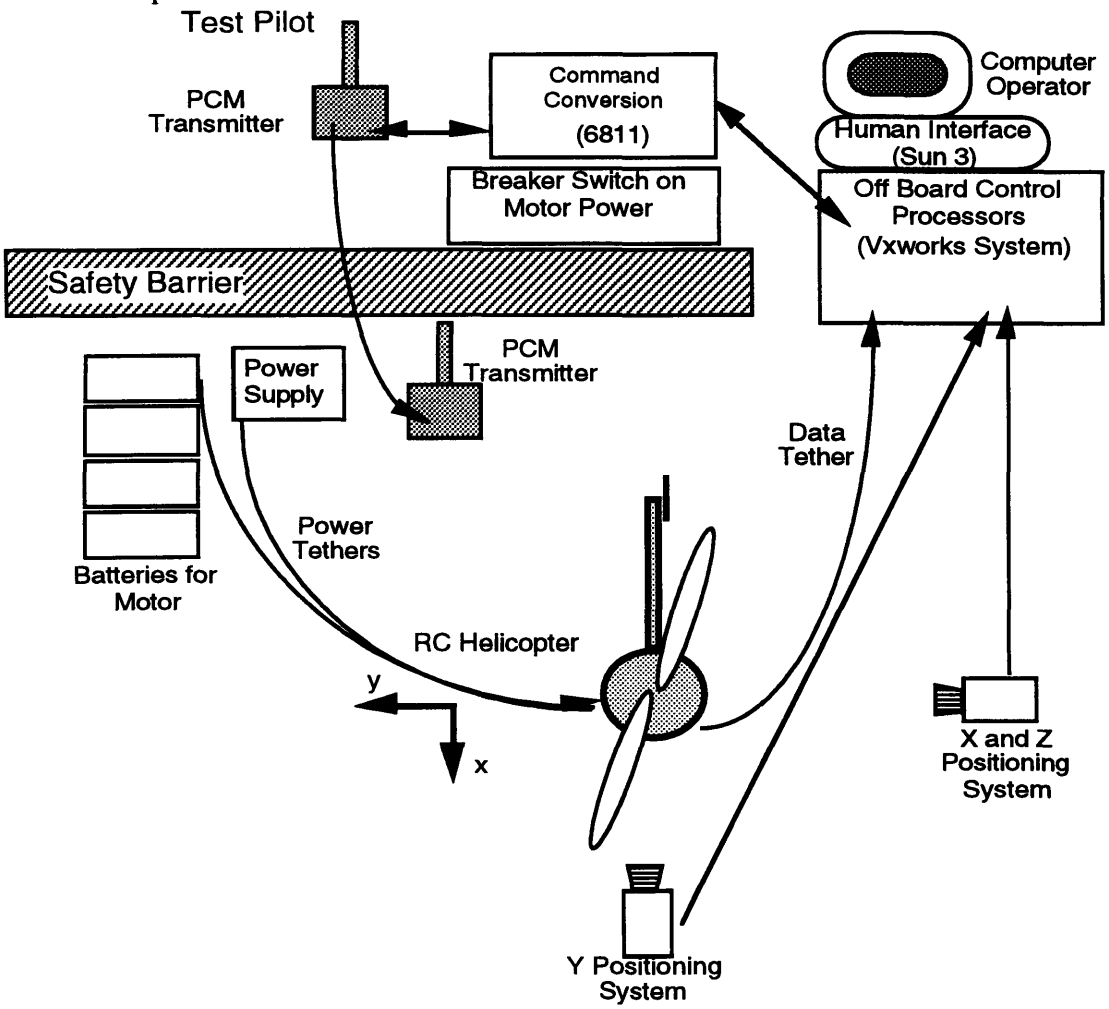


Figure 6
Top View of Experimental Setup

2.1 Landing Platform

The landing platform has an adjustable slope and enough friction for the helicopter to sit on the slope without sliding when no external accelerations other than gravity are acting on the vehicle. The landing platform is a 4' x 8' x 3/4" piece of plywood reinforced with a framework of 2" x 4" boards to add stiffness to the platform. The low end of the slope is created by a hinged attachment to two floor panels on one side of the plywood. The other side rests on two adjustable posts hinged to two more floor panels. The length of the rods adjust the slope of the landing platform.

When the test pilot attempted to land the helicopter on the plywood platform, the helicopter slid to the bottom of the hill on slopes more than about 5°. Carpeting was added to the plywood to increase the friction of the platform so that the helicopter at rest remains still at the upper slopes of the platform. The safety cable for the helicopter is attached to the floor and runs through a hole in the platform to the helicopter.

2.2 RC Helicopter and Sensor Platform

The vehicle for this thesis is a radio controlled model helicopter, the Custom X-cell-60. The X-Cell has a 1.4 m rotor diameter and comes with a 2 hp, two cycle engine. The gas engine has been replaced with a 2 kW DC electric motor by Hecktoplekt. An Astroflight Speed Controller is used to maintain a constant rotor rpm during steady state hover conditions.[1] A human pilot can fly the electric setup the same as he/she flies a gas setup. In addition, a fan has been added to cool the motor.

The landing gear of the standard RC helicopter is replaced with the sensor platform shown in Figure 7. The sensor platform is a section of plywood with non-stress bearing sections removed to reduce the weight. The plywood is reinforced with

[1] Ben Weintraub provided this information as he implemented much of the modifications to the helicopters in the AI Lab. More details can be found in his master's thesis: Learning Control Applied to a Model Helicopter, MIT: Feb. 1994.

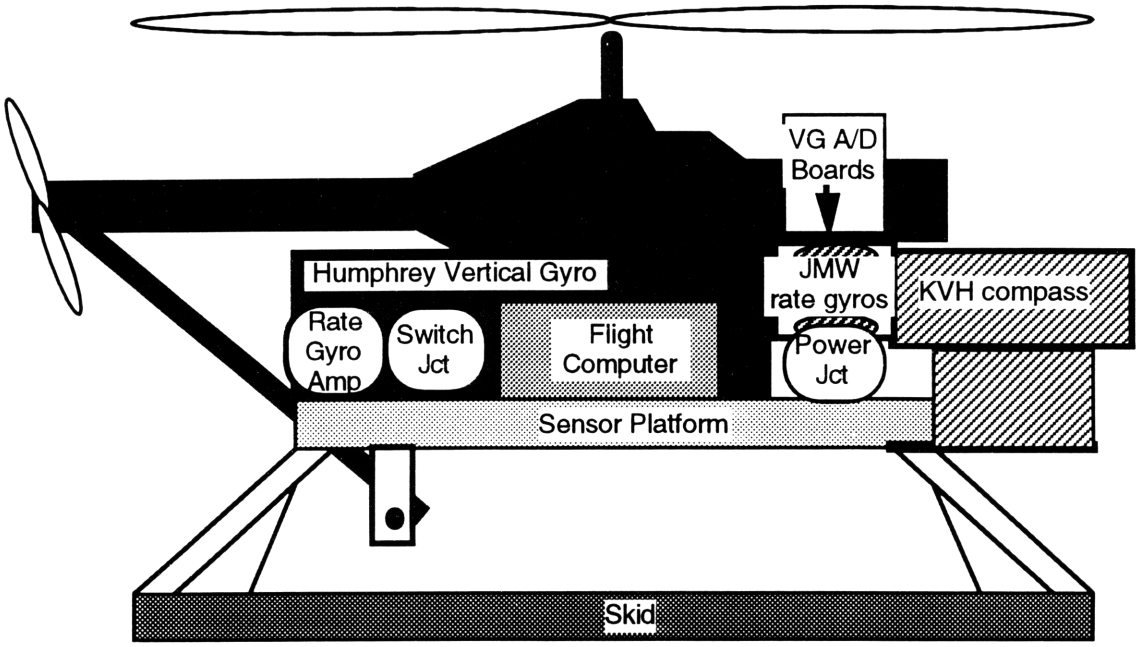
fiberglass and mounted to a 35.56 cm (14 inch) wider base. The wider base gives the helicopter a larger margin to avoid dynamic rollover. The sensor platform is shock mounted to the helicopter to reduce structural vibrations of the sensor platform which would appear in the sensor output.

Mounted to the sensor platform are three JMWIII Gyrosensor rate gyros oriented to measure angular rates about the roll, pitch and yaw axes. A Humphrey VG24-3301-1 vertical gyro used in actual aircraft measures the roll and pitch attitude angles. The roll and pitch range of vertical gyro are $\pm 60^{\circ}$ pitch and $\pm 90^{\circ}$ roll.

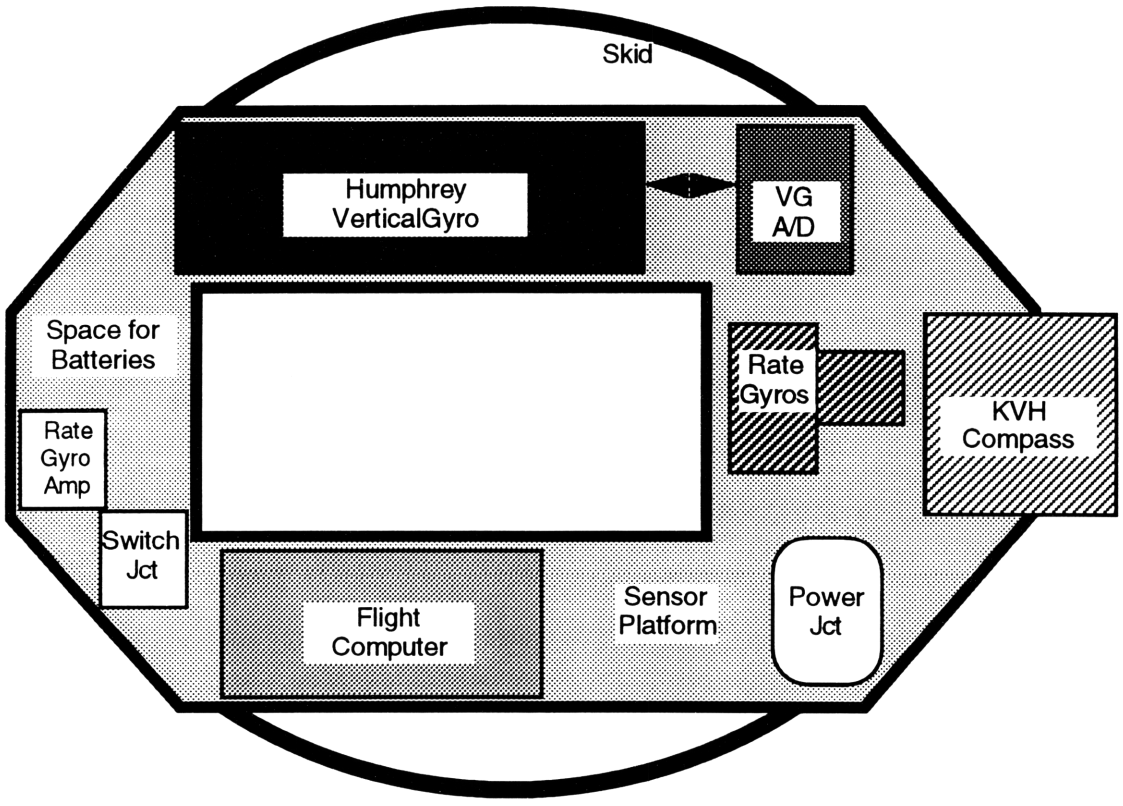
The KVH Heading Sensor ROV-1000, an electromagnetic compass, measures the heading angle which is the yaw angle in level flight. The analog output of the compass has a wrap point when it goes from 359° to 0° in heading. Since heading is maintained constant in this experiment, the analog offset was adjusted so that the wrap point was at the rear of the vehicle. This sensor must be mounted at least 12 inches forward of the electric motor to minimize interference from the magnets in the motor.

The flight computer, also mounted on the sensor platform, is a Motorola 68332 which collects sensor information and sends it to the ground computer via a data tether. The power junction box routes power from either batteries or the power tether to all the components on the sensor platform as well as the radio receiver and servos on the helicopter.

The center of mass of the helicopter should be aligned with the shaft that spins the main rotor blades. The helicopter was suspended from the ceiling by its blades, and the sensors were placed, given the 12 inch forward requirement on the compass, so that the helicopter platform was as level as possible thus indicating that the sensor weight is equally distributed. Figure 7 shows the layout of the platform.



a) Side View



b) Top View

Figure 7
Schematic of Sensor Platform Layout

2.3 Ground Based Equipment

Due to the vehicle payload limit of 12-15 lb., the power for the vehicle and sensor platform components are supplied by tether cables. The electric motor is supplied with 42 volts DC power. A power tether connects the electric motor to the car batteries on the ground (three 12 volt batteries and one 6 volt battery in series) which can supply the necessary voltage for extended periods of time without recharging. The components on the sensor platform are supplied through a second tether with +5v and ± 15v from a DC power supply located on the ground.

The flight computer, a 68332 processor, collects the sensor information from the platform and sends the information to the off board control processors via a data tether cable. All three tethers (2 power and 1 data) are depicted in Figure 6.

Commands to the helicopter use the standard PCM transmitters used with radio controlled hobby vehicles. The PCM transmitter is connected to the off board control processors via analog to digital and digital to analog converters on a 6811 board. The PCM radio has been modified with a switch which allows the test pilot to manually override the computer control. To eliminate radio interference, this transmitter is connected via trainer cord to a second transmitter which contains the transmitter module and active antenna. Without this setup, the servos on the helicopter receive noisy radio signals because the wire connecting the transmitter to the computer system was acting as an additional antenna and interfering with the desired signals.

The off board control processors are a Vxworks real time operating system which gathers information through its serial ports from the onboard flight computer, the PCM radio transmitter system, and the two ground based positioning systems. The off board control processors then process the information and calculate the commands which are sent to the PCM transmitter. A Sun 3 workstation is connected to the off board control processors for human interaction and for recording data files.

The ground based positioning system is a color vision tracking system which tracks a bright orange canopy covering the helicopter platform. Figure 8a shows a schematic of the setup of the positioning system. The video camera image is thresholded to show only orange objects in the image and a schematic of the image is shown in Figure 8b. The helicopter canopy is the largest and usually the only object in the image. The pixel location of the centroid of the largest object is used as the position of the helicopter.[1] The pixel location is converted to x, y and z coordinates by the off board control processors.

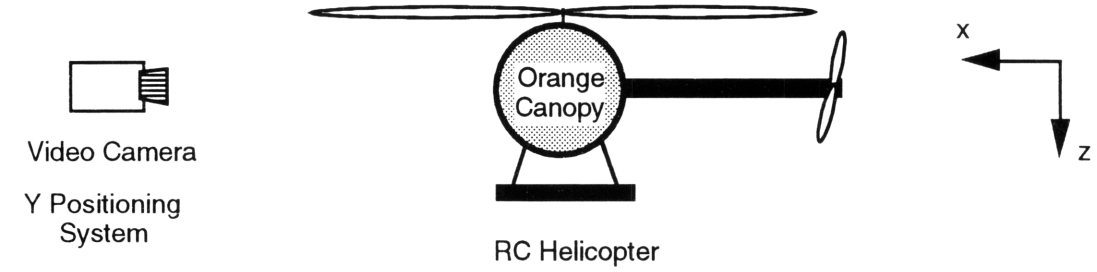


Figure 8a
Schematic of Positioning System Setup

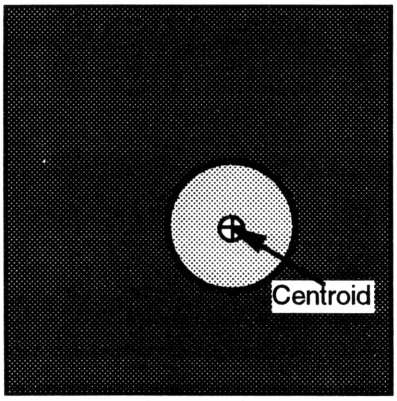


Figure 8b
Schematic of Thresholded Image of Positioning System

[1]More details can be found in the paper by Anne Wright titled "A High-speed, Low-latency Portable Visual Sensing System", 1993 SPIE Proceedings, 1993.

3 Design of the LQG Hover Compensator

The landing strategy in this thesis starts from a stabilized hover. Computer control of hover is achieved on this experimental setup using a Linear Quadratic Gaussian (LQG) estimator and controller. Since the LQG compensator is a model based compensator, a model of the helicopter in hover was identified using knowledge of the dynamics and system identification techniques.

3.1 Explanation of Controls

The controls of the RC helicopter are referred to as roll cyclic, pitch cyclic, rudder and collective. The roll and pitch cyclic commands change the pitch of the main blades as a function of the blade's position in a revolution. The cyclic commands control the tilt of the rotor plane and thus the tilt of the thrust vector of the helicopter. [1]

The rudder command controls the collective pitch of the tail rotor. The rudder command is mixed in the transmitter with the collective command to cancel out the torque on the body due to the spinning of the main rotor blades. The rudder command adds or subtracts from this collective blade pitch of the tail rotor to produce a torque on the body about the z axis and hence allow the vehicle to change it's heading.

The collective command controls the total lift on the helicopter by controlling the pitch of both main rotor blades.

3.2 Dynamics of Helicopter in Hover

For the purposes of model identification, the controls are assumed to decouple as roll (or roll cyclic) controlling roll and y motion of the vehicle, pitch (or pitch cyclic) controlling pitch and x motion of the vehicle, rudder controlling yaw motion of the vehicle, and collective controlling z motion of the vehicle.

Since cyclic commands create a differential in the force acting on the blades, a net torque acts on the body of the helicopter which is related to an angular acceleration on the vehicle by

---

[1] Wayne Johnson. Helicopter Theory. Princeton: Princeton University Press, 1980. p. 159.

$$\tau = I\ddot{\theta} \qquad [4]$$

where I is the moment of inertia of the helicopter about the axis of rotation. This means that the cyclic commands should integrate into an angular rate. The commands are known and the rate gyros provide a measurement of the angular rate of the vehicle, so an input-output relationship can be found by fitting a controlled autoregressive model (ARX) model to the data.[1]

Angular rates integrate to an angle and an input-output relation can be found by comparing the angular measurement to the angular rate measurement. Any non-zero body angle tilts the thrust vector which then has a horizontal component of force. This horizontal force is related to the horizontal acceleration by the mass of the vehicle (F = ma) and integrates to the horizontal velocity of the vehicle. Only horizontal position is measured, but since the system identification is not in real time, the horizontal velocity is estimated by taking the difference in position measurements and scaling by the sample time. An input-output relation is determined again by fitting an ARX model to the body angle measurement and the velocity estimates. The relationship between the velocity estimates and the position data is known.

Rudder commands create a torque on the body which is related to the yaw angular acceleration by the inertia of the vehicle. The angular acceleration is integrated to a yaw rate which is measured by the yaw rate gyro. The yaw rate integrates to yaw position which is measured by the compass heading sensor. Input-output relations from command to rate gyro and from rate gyro to compass form the yaw model.

The collective command creates a force on the blades (thrust) which is related to the vertical acceleration acting on the vehicle by the mass of the vehicle. (In hover, the thrust exactly cancels the force due to gravity acting on the vehicle.) This acceleration

[1]The system identification process is described in more detail in the Appendix. The Appendix also gives the form of the ARX model.

integrates to vertical velocity of the helicopter. Again the vertical velocity is created as the horizontal velocities were by taking differences of the vertical position measurements.

3.3 Parameter Identification

Since the helicopter is an inherently unstable vehicle, the standard system identification method of driving the plant with an impulse command and recording the response is not feasible since the helicopter might be damaged in the response. An alternative method is used where a test pilot flies the helicopter in hover and the computer adds pseudo-random inputs to the pilot's commands.[1] These pseudo-random inputs are scaled so that the pilot can recover the vehicle from the worst command generated. The pilot can only compensate for very slow offsets and does not react to the quick series of pulses, which prevents identification of the dynamics of the test pilot.

After taking several perturbation data sets for each of the commands, means of each measured state are removed for system identification. For each input-output relation, first compare the spectral frequency analysis of the data sets. In this experimental setup, data is sampled at 50 Hz. The main rotor blades spin at 1800 rpm which results in a 30 Hz noise source. The data is not sampled fast enough to eliminate aliasing of this frequency so models identified cannot be accurate around 30 Hz $\approx$ 188 rad/sec. This is seen in the spectral analysis plot for the relationship for the roll cyclic and the rate gyro of Figure 9 as the roughness of the data plots above 20 rad/sec. A useful model captures the low frequency dynamics, so the data from which the model is identified should be consistent through at least 1.5 Hz $\approx$ 10 rad/sec. When searching for the model structure, post-processing the data with a zero phase, 5 Hz cutoff low pass filter attenuates high frequency information in the data so that the model doesn't attempt to fit high frequency data (above 20 rad/sec) which is inconsistent from data set to data set.

---

[1] See Torsten Söderström. System Identification. New York: Prentice Hall, 1989. pp. 96-112. for more information of pseudo-random inputs for system identification.

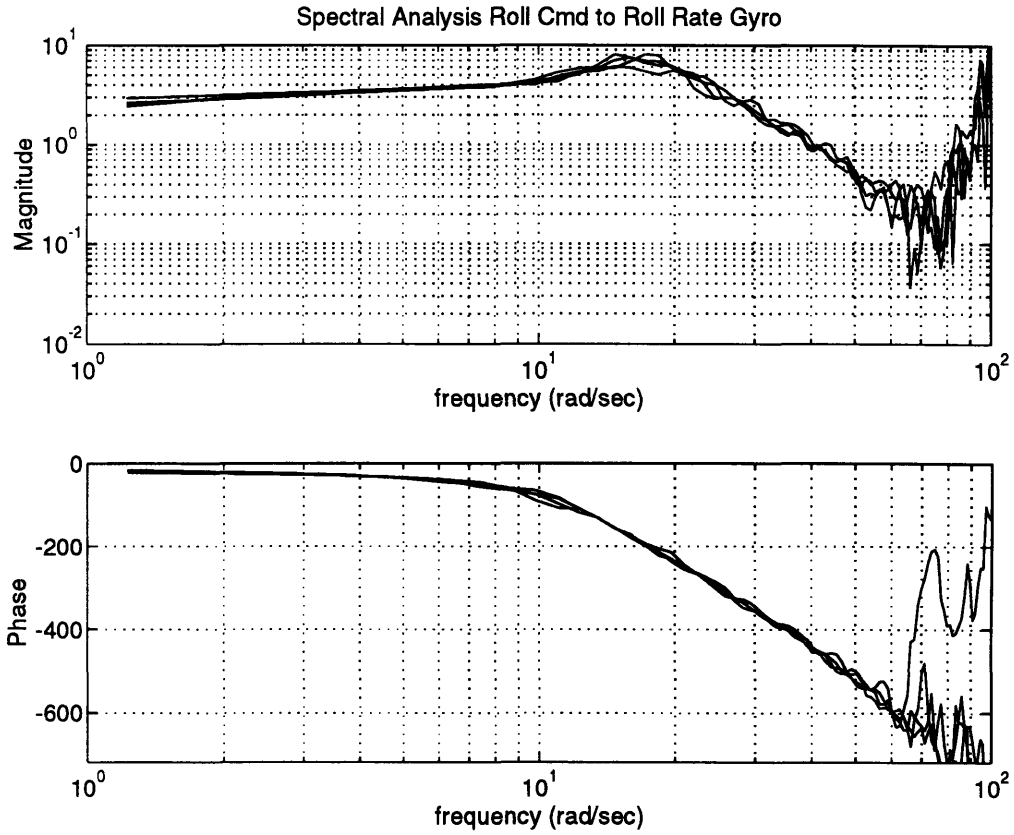


Figure 9
Spectral Analysis of Roll Command to Roll Rate Gyro

Both one step and multi-step simulations were used to judge the quality of the model. Figure 10a shows a simulation of the identified relationship between roll command and rate gyro on the same data from which the relationship was identified. The first plot shows a single step ahead prediction where the match is almost exact. The second plot is the identified relationship driven only by the command compared to the measurement, and this shows that the although the model is not exact it still gives a reasonable prediction of the measurement. Since this is the data from which the model was identified, a close fit is expected. Figure 10b shows a cross validation by running the same simulation of the same identified model driven by a different set of data. Since the multi-step simulation still reasonably predicts the measurements, the relationship is considered valid.

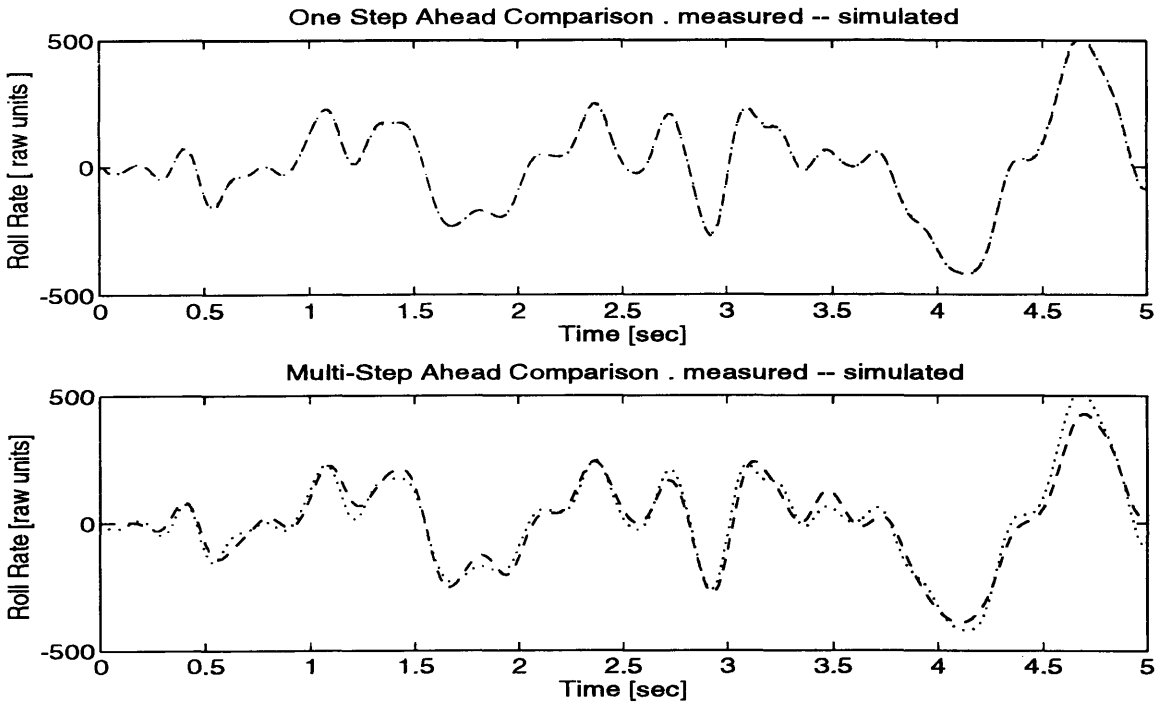


Figure 10a
Simulation of Command to Rate Gyro Model on ID Data

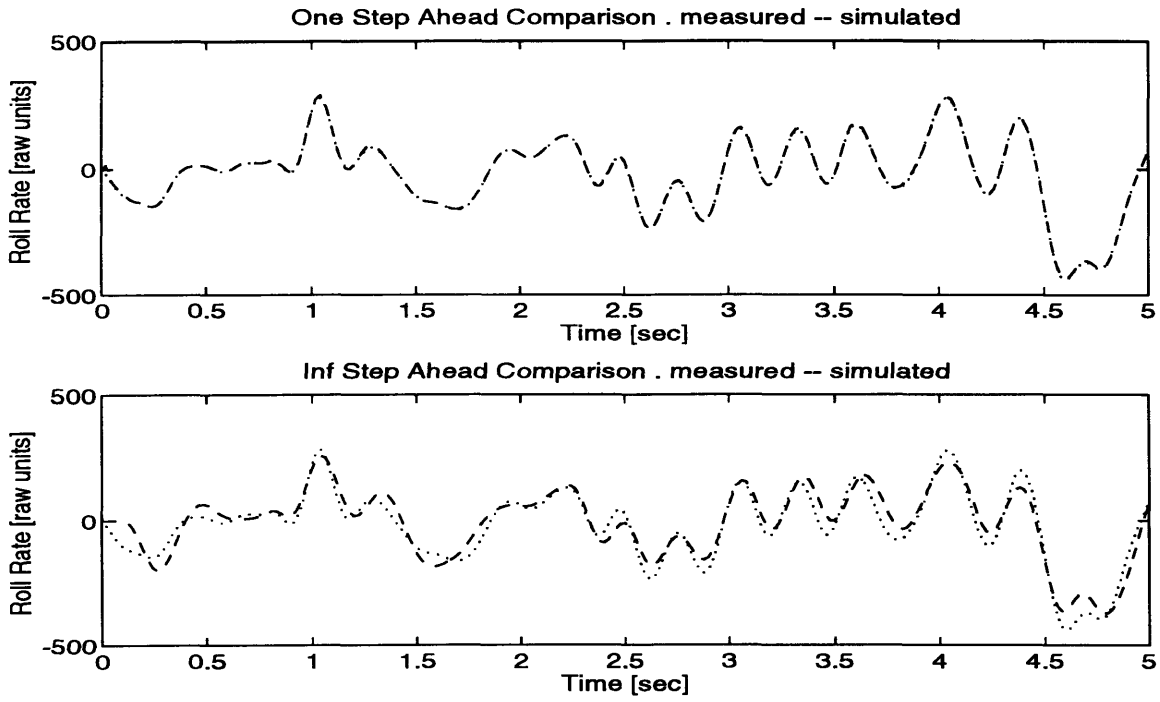


Figure 10b
Simulation of Command to Rate Gyro Model Cross Verification

Once a structure and its associated parameters were identified for each input-output relation, a state space description of the system was assembled. The state space description of the roll controller is as follows:

$$\begin{bmatrix} cmd1 \\ cmd2 \\ cmd3 \\ cmd4 \\ cmd5 \\ cmd6 \\ cmd7 \\ \dot{\phi} \\ \dot{\phi}1 \\ \phi \\ \dot{y} \\ y \end{bmatrix}_{i+1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a & b & c & d & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e & 0 & f & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g & h & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & dt & 1 \end{bmatrix} \begin{bmatrix} cmd1 \\ cmd2 \\ cmd3 \\ cmd4 \\ cmd5 \\ cmd6 \\ cmd7 \\ \dot{\phi} \\ \dot{\phi}1 \\ \phi \\ \dot{y} \\ y \end{bmatrix}_i + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u_i \qquad [5]$$

where the numbered command and angular rate states indicate delays and the dotted states indicate rates. The delayed command states represent the time delay from when the command is issued from the computer until the effect is first visible in the measurements. More details on the system identification of the LQG hover compensator can be found in the Appendix.

3.4 LQG Controller and Estimator Gains

Using these state-space descriptions and the recursive Ricatti equation algorithm[1], controller gains for full-state feedback are determined. Since the matrices Q and R define the cost equation of the linear quadratic regulator (LQR), R, the penalty on the command was identity and Q, the state weighting matrix, had only a value for the position state since the goal is to maintain position. All of the command states are penalized in R and non zero values in the other states create a non zero value in the position state anyway. The values of Q (or R) are varied because the exact noise of the sensors and the frequencies of the unmodeled dynamics are unknown.

[1]Gene F. Franklin, J. David Powell, Michael L. Workman. Digital Control of Dynamic Systems. Reading, MA: Addison-Wesley Publishing, 1990. p. 427.

The translational velocities were created in post processing, so a Kalman filter using the model identified above is used to estimate this state. It also smoothes existing measurements. Kalman filter gains are created by estimating the noise on each sensor to determine how much to trust it.

The LQG controller and estimator parameters Q and R and hence the controller and estimator gains were experimentally determined to achieve a balance between position drift and excitation of unmodeled dynamics. A description of the process as well as the actual gain values are given in the Appendix.

3.5 Standardized Collective Input

Since many factors are involved in how accurately the helicopter can land on a target, the same standardized collective input is flown by each of the controllers and the difference between the final position and the targeted position serves as a measure for how well each controller performed. The helicopter lands so that the heading is perpendicular to the gradient of the slope. Since only roll cyclic is modified in this experiment, the heading perpendicular to the gradient implies that the downhill motion of the helicopter can be completely characterized by the y position measurement of the helicopter (see Figure 11).

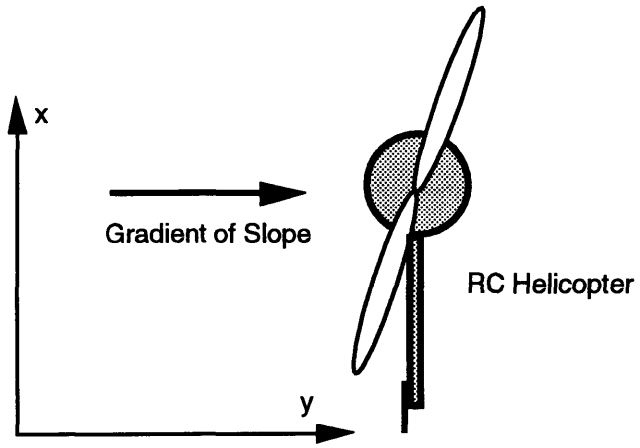


Figure 11
Heading of Helicopter Relative to the Slope Gradient

Each of the controllers modified the cyclic command if they modified anything at all, but slight variations in the collective command can drastically change the outcome

of the landing by causing the helicopter to bounce or slide regardless of the applied cyclic command. To eliminate this uncertainty, the collective command is generated by the state machine shown in Table 1. Since the collective to z dynamics are stable, an open loop collective command is used for take off and landing the helicopter. The first three states take the helicopter to hover which serves as the starting point of this experiment.

Table 1
Z State Machine

State	Tasks	State Change Conditions
ON GROUND	Collective off	Operator gives take off cmd
SPIN UP	Increase collective pitch Decrement collective cmd	Collective can lift vehicle
TAKE OFF	Increase collective pitch Decrement collective cmd	Collective at hover level
HOVER	Maintain altitude	Operator gives land cmd
LAND	Decrease collective pitch Increment collective cmd	Helicopter touches down
SETTLE	Decrease collective pitch Increment collective cmd	Collective can't lift vehicle
SPIN DOWN	Decrease collective pitch Increment collective cmd Shutoff	Collective off Switch to ON GROUND

Figure 12 shows an example of the collective command used in the experiments. The standardized collective input starts from when the helicopter is in hover and the data storage mechanism and the landing trajectory are triggered at the same time so that information from the data sets may be compared. The first part of the standardized collective input is given by the land state in the z state machine and corresponds to the descent phase illustrated in Figure 1. The descent of the vehicle from the hover altitude to the ground which can be seen in the z position measurement in Figure 12 as an increase in the z position since z is positive downward. The rate of descent (i.e., the

slope of the collective command) is chosen so that the helicopter does not bounce when it touches down. Also notice the spike in the roll rate gyro measurement which indicates when the helicopter first touched the ground. The second phase is the settle phase in both the state machine in Table 1 and the landing phases in Figure 1 which is the area of decreased slope in the collective command of Figure 12. Note that the vertical roll gyro measurement moves from a zero altitude to the slope of the hill during this phase. The third phase seen in the standardized collective command input is the spin down phase which has a steeper slope and finally a cutoff in the collective command. The cutoff is delayed slightly to ensure that the vehicle skids are in full contact with the ground. A premature shutoff leads to the vehicle dropping abruptly to the ground if the changeover point between settle and spin down was inappropriately chosen. The shutoff is necessary to reduce the amount of low frequency vibration on the helicopter as the blades are spinning down. The roll rate gyro in Figure 12 shows some vibration even though the collective command is shutoff.

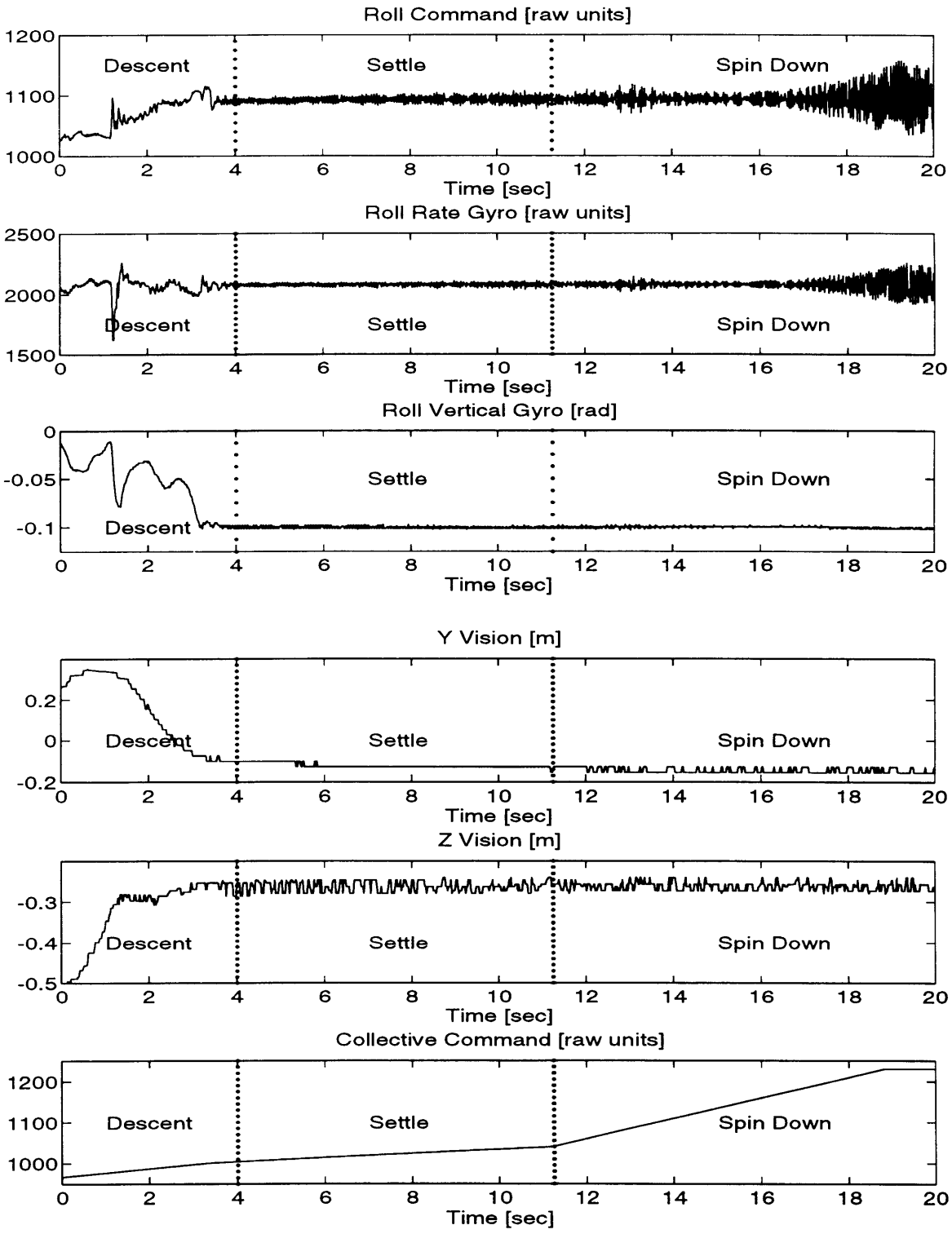


Figure 12
Example of Measurements Following Collective Command Trajectory on 5°

3.6 Measurement of Landing Accuracy

The landing accuracy defined as the difference between the final landing position and the desired landing position is used to judge how each of the controllers achieves the final goal of landing on a specified target. Each controller is compared to the LQG hover compensator following the standardized collective input. Controllers performing significantly worse than the LQG hover compensator are eliminated from further consideration. The helicopter using the LQG hover compensator is capable of landing on the hill without sliding to the bottom for slopes less than 12°. Controllers which cannot keep the helicopter on the hill are considered significantly worse than the LQG hover compensator. Those that can keep the helicopter on the landing platform fly each hill under consideration four times.

For each flight, the desired target position is considered the steady-state position that the helicopter hovers over prior to starting the standardized collective input. The desired target position is calculated as an average of the horizontal position of the first 100 data points before the helicopter has interacted with the ground. The final position for each data set is calculated similarly using the last data points of the data set when the helicopter has come to full rest upon the slope. The landing accuracy for a flight is simply the difference between the final position and the desired position of the flight.

Since each controller makes 4 flights on each slope, the mean landing accuracy of a particular controller at a particular slope is calculated as the average over the four flights.

The sample standard deviation is calculated as follows:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{4} (Y_i - \overline{Y})^2}{4-1}} \qquad [6]$$

where Y_i is the landing accuracy of a particular flight and $\overline{Y}$ is the sample mean accuracy of a particular slope for each controller. The sample standard deviations of the landing

accuracy are presented in the result tables, and the error bars in the presentation of the final results indicate the minimum and maximum flight landing accuracy at each slope.

4 Experiment

Three strategies are compared to the LQG hover compensator. Since the LQG hover compensator focuses on maintaining position, the first strategy, the altered weight LQG hover compensator, adds penalties to the roll rate and attitude states in the LQR controller cost function. The second strategy, the open loop cyclic controller, attempts to mimic the human test pilot's cyclic input which is proportional to the standardized collective input. The final strategy, the multiple model adaptive controller (MMAC), is a combination of the LQG hover compensator and the open loop cyclic controller strategies. The design and performance of each of these strategies follows.

4.1 LQG Hover Compensator

4.1.1 Design of LQG Hover Compensator

This strategy required no modifications since it is assumed that the helicopter has an existing hover controller. The standardized collective input is used to land the helicopter. Since the LQG hover compensator is robust enough to handle some disturbances, the dynamics of the ground are simply treated as an external disturbance and the LQG hover compensator attempts to recover from an external torque applied by the ground. The control law is given by

$$u = -K_{hover}\hat{x}_{hover} \qquad [7]$$

where K_{hover} are the roll cyclic LQG hover compensator gains used to maintain roll and y position in hover (acutual values are given in the Appendix) and $\hat{x}_{hover}$ are the estimated states of the LQG roll cyclic control axis. Since no adaptation of the control from a hovering vehicle is needed, experiments using this controller serve as the "control" experiment to which the other strategies are compared.

This method succeeds in landing the vehicle on slopes with sufficient friction, but experiences some loss in landing accuracy due to the fact that the LQG controller is a model based compensator which has no knowledge of the existence of the ground which produces external forces on the helicopter. The controller only corrects

for errors in position after the fact and does not produce enough of a command to fully counteract the torque generated by the external input.

4.1.2 Performance of the LQG Hover Compensator

Table 2 shows the mean and the standard deviation of the landing accuracy for the given ground slopes. Since a greater slope allows for a greater rotation and thus a larger horizontal component of the thrust leading to a larger position loss, the results in Table 2 reflect that the steeper the slope, the greater the magnitude of the mean landing accuracy. A negative mean landing accuracy on a positive ground slope indicates that the helicopter landed downhill of the desired target and vice versa for a negative ground slope. The vehicle does remain on the hill because once the lift is reduced enough to generate enough frictional force (via the Normal force) the motion is stopped. Looking at Figure 12, the roll command applies a command into the slope based on the displacements in the roll angle and y position states, and this stops the loss of position down the slope. However, this method does not anticipate the slope, so it cannot prevent the loss of position, but it can stop further loss.

Table 2
Mean Landing Accuracy of the LQG Hover Compensator

Slope	Mean Landing Accuracy	Standard Deviation of Landing Accuracy
-4°	0.0367 m	0.1216 m
0°	-0.0566 m	0.0055 m
2°	-0.2025 m	0.0325 m
5°	-0.3793 m	0.0461 m
6°	-0.4804 m	0.0375 m
8°	-0.8044 m	0.0869 m
10°	-0.9060 m	0.1090 m
12°	-0.9358 m	0.1447 m

4.2 Attitude Weighted LQG Hover Compensator

Interviewing the test pilot revealed that when the helicopter is in contact with the ground, he concentrates on maintaining a level attitude while throttling down to keep from losing position. When the LQG hover compensator was created, the focus was on maintaining position, and so the penalties in the cost function were applied primarily on the position states. Incorporating the human pilot's strategy, increasing the cost on the attitude angles and rates during the settle phase can yield better performance. Some penalty on position is necessary to maintain the target position achieved by the original LQG hover compensator. This method still has no knowledge of the ground interaction, but it should respond sooner than the LQG hover compensator because it does not have to wait for attitude errors to integrate into position errors. The control law is given by

$$u = -K_{AWLQG}\hat{x}_{hover} \qquad [8]$$

where K_{AWLQG} is similar to K_{hover} except with higher gain values on the attitude rate and angle states.

In reality, this method encounters problems with the accuracy of the model. The original LQG hover compensator gains were chosen to be as high as possible without significantly exciting high frequency unmodeled dynamics. The end result is that the gains for the attitude weighted LQG hover compensator cannot be raised much from the values of the original LQG hover compensator without exciting these dynamics which cause severe oscillations of the helicopter (shown in the region between the dotted lines on Figure 13), and so this method is eliminated from further consideration.

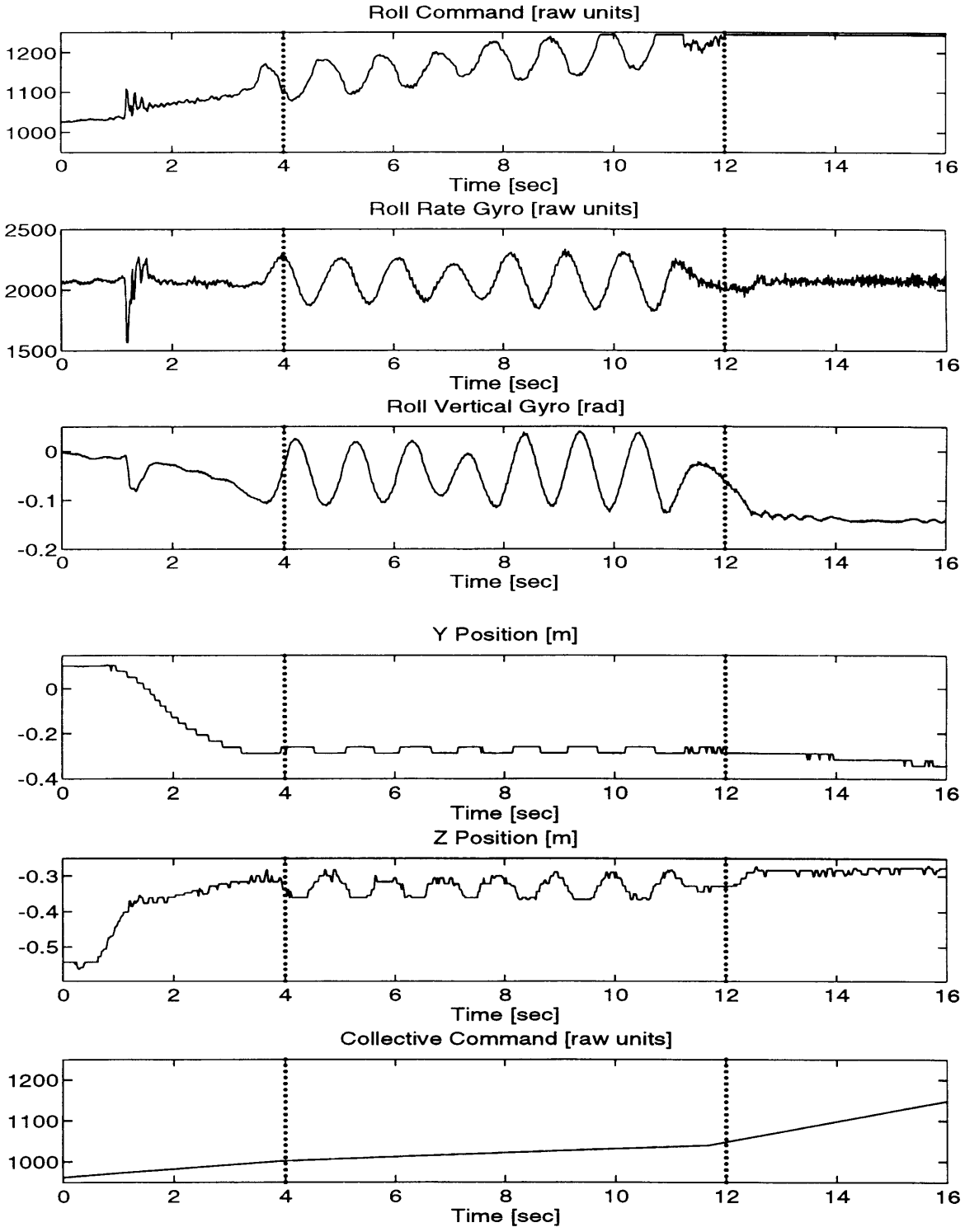


Figure 13
Attitude Weighted LQG Hover Compensator on 5 ° Slope

4.3 Open Loop Cyclic Controller

Observing how the test pilot landed on the slope revealed that once the helicopter was in contact with the ground, the test pilot applied a cyclic command in the uphill direction proportional to the collective command. Since the test pilot achieved a better target position than the LQG hover compensator, a controller of the form

$$u = k \cdot collective\_cmd \qquad [9]$$

based on a model of how the collective command affects the attitude and position of a helicopter should improve the target landing performance.

4.3.1 Design of the Open Loop Cyclic Controller

Although it is desirable to identify a complete model of both the cyclic and collective inputs to the behavior of the attitude and position of the helicopter while it is in contact with the ground using the system identification methods that were used to identify the model of the helicopter in hover, this is impractical since the helicopter on the ground undergoing pseudo-random inputs large enough to identify a model might damage the helicopter. Therefore, instead of perturbations, observations of the human test pilot is used to create a model of how the cyclic command is related to the collective command. The test pilot controlled the cyclic and rudder control axes while the computer ran the collective trajectory.

An 8° slope was used because the slope was as different from the level surface as possible without the human pilot saturating the roll cyclic. Taking a linear regression between the cyclic command and collective command during the settle phase to identify k in equation [9] showed that the pilot was consistent in the gain applied on the collective command to generate a cyclic command on the same slope. Figure 14 shows the roll command in a dashed line versus a roll command produced by the identified gain on collective command in a solid line for three different data sets taken of the test pilot landing on an 8° slope to the right of the helicopter.

The method is an open loop roll command applied once the helicopter is in contact with the ground. This method only works if the helicopter is stable in roll during this phase. The model used in the LQG hover compensator is unstable (the plant is unstable), and the ground probably does not stabilize roll motions until the skids are in full contact with the ground and the thrust is less than the weight of the vehicle. This is not an effective landing method as discussed in the next section.

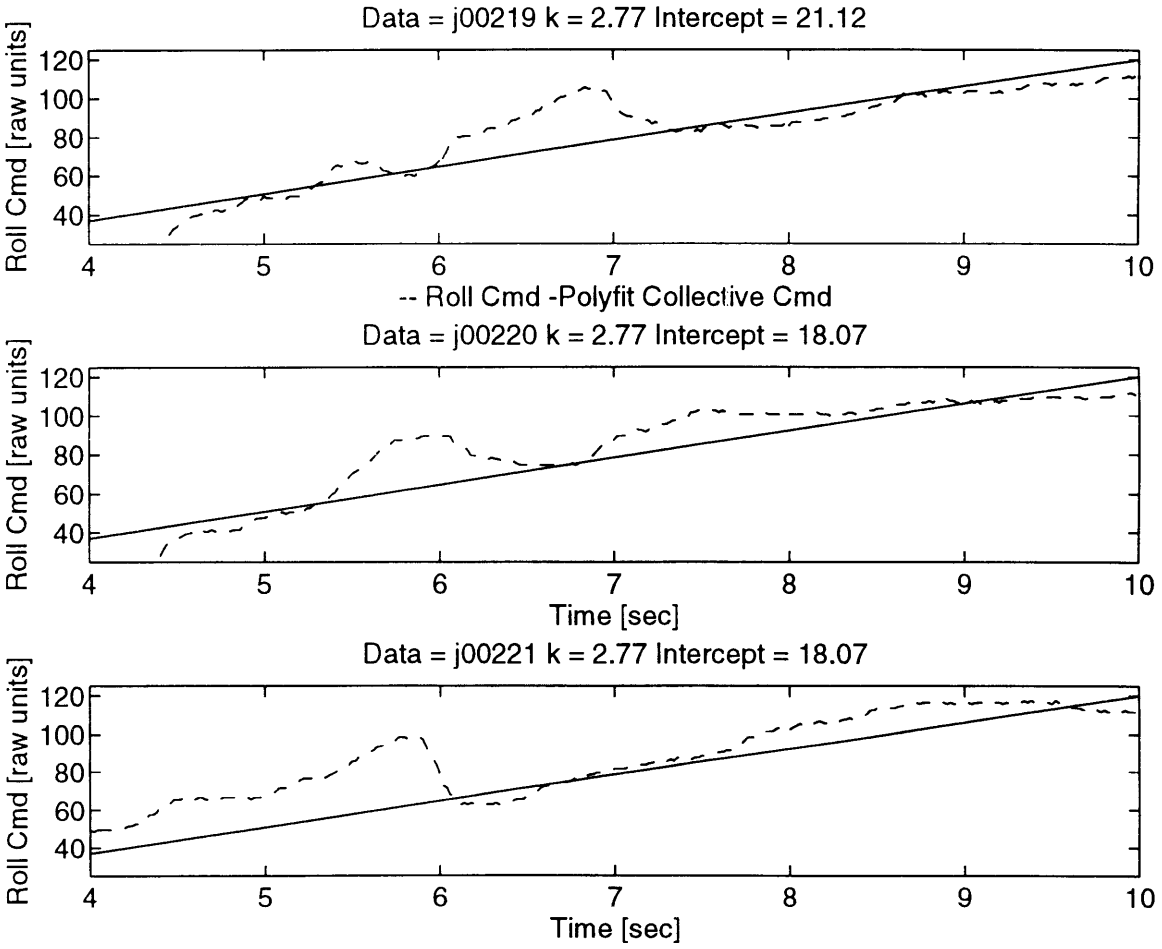


Figure 14
Open Loop Cyclic Controller Simulated vs. Measured Roll Cyclic on 8° Slope

4.3.2 Performance of Open Loop Cyclic Controller

Figure 15 shows data from a flight using open loop cyclic controller. Although the roll command was applied in the correct direction, the helicopter as seen in

the y measurement ended up at the bottom of the hill approximately 1.0 m to the left of the desired position (corresponding to a downhill motion). This loss in position can be attributed to slight errors in the state when the controller switched from the LQG hover compensator to the open loop cyclic controller. The open loop cyclic controller cannot correct for these errors without feedback. Since this strategy is considered significantly off target, it is eliminated from further consideration.

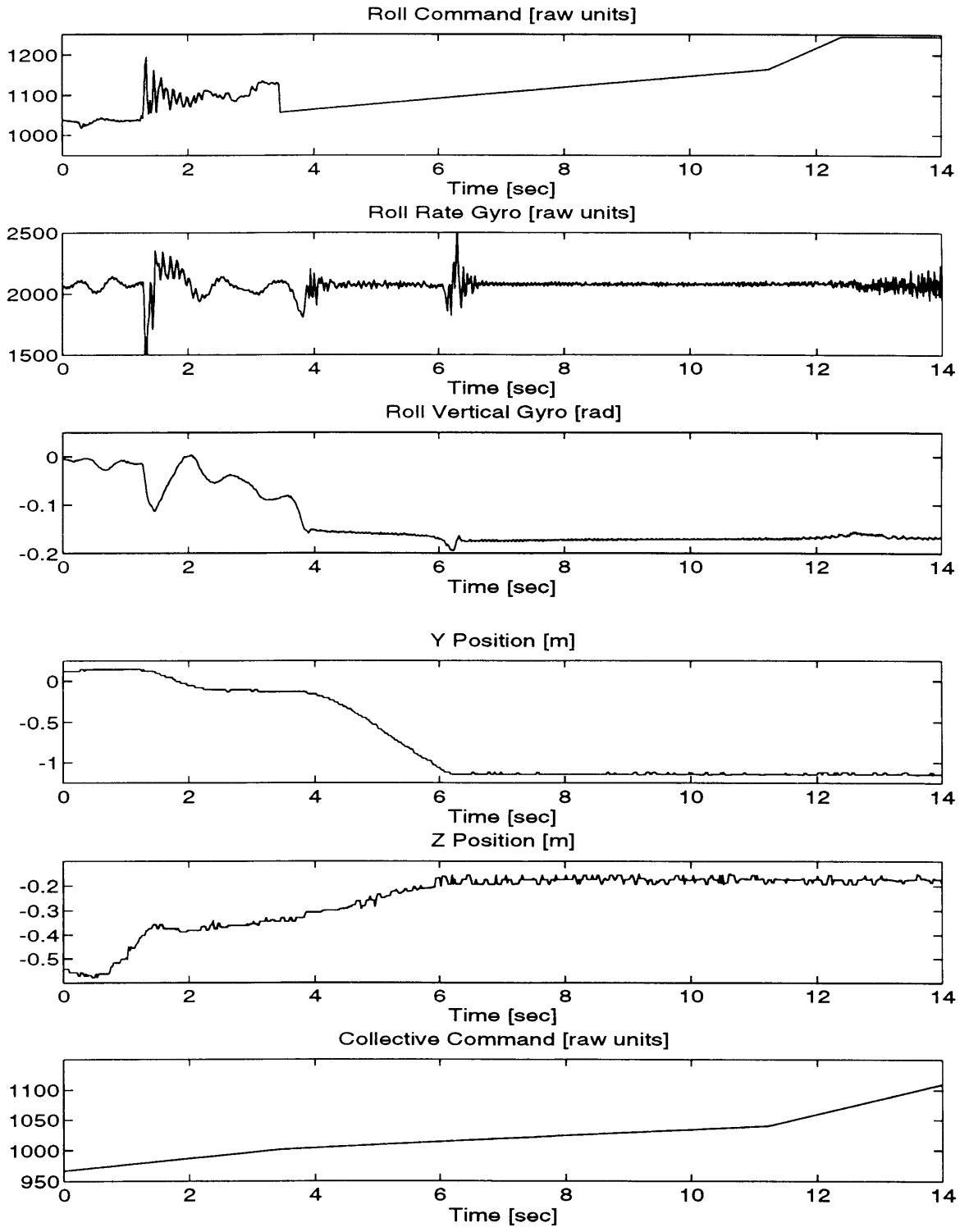


Figure 15
Open Loop Cyclic Controller on 8° Slope

4.4 Multiple Model Adaptive Controller (MMAC)

The final strategy considered combines the LQG hover compensator, which corrects for position errors, with the open loop cyclic controller, which accounts for the constant disturbance in the attitude rate due to the ground. This method is more powerful than the LQG hover compensator because the reaction force due to the ground is included. It also works better than the open loop cyclic controller because it has a feedback component which allows for corrections to errors in the roll attitude rate and angle and y position states.

4.4.1 Design of Multiple Model Adaptive Controller

Part of the cyclic command is generated by the LQG hover compensator and part of the command is related to the collective input as in the open loop cyclic controller. Since the sign of the component of the roll cyclic control from the open loop cyclic controller depends on the direction of the slope, multiple model adaptive control[1] is used to determine the sign of the additional cyclic command.

Multiple model adaptive control (MMAC) uses the model residuals and prior probabilities to determine the probability that the vehicle is in the regime predicted by that particular model and then uses the probability times the control generated by each individual compensator to generate the total command. In this application three models were used for the roll cyclic controller: uphill to the right of the helicopter, level, and uphill to the left of the helicopter.

Using the information discussed in section 4.3, the open loop cyclic command was obtained by *adding* a gain times the collective value to the cyclic command to compensate for a hill on the right. This implies that a helicopter without this compensation behaves as it does in hover except that a component equivalent to the gain times collective value is *subtracted* from the roll cyclic command value. Similarly, the

[1]Multiple model adaptive control has been used with sensor and actuator failures as explained in paper by Peter S. Maybeck and Donald L. Pogoda, "Multiple Model Adaptive Controller of the STOL F-15 with Sensor/Actuator Failures," Proceedings of 28th IEEE Conference on Decision and Control. Dec. 1989. pp. 1566-1572.

model for a hill on the left was created by reversing the appropriate signs in the hill on the right model. The level model is exactly the model of the helicopter in hover.

The difference between the models can be seen in the residuals of the roll rate gyro state where the residual is the difference between the measured rate gyro and the estimated rate gyro state before the observation update. Since all other states involve at least one integration from the rate gyro state, the corresponding residuals of those states contain little information regarding the probability of the model because the estimates have already been updated by the measurements during the integration process.

The calculation of the control uses the probabilities that the helicopter is described by a particular model based on the measurements and probability history. The probability density function that the current measurements indicate a particular model given the previous measurements is given by

$$f(z_i|a_m, z_{i-1}) = \frac{1}{\sqrt{2\pi}\sigma_{rg}} \exp\left\{-\frac{r_m^2(t_i)}{2\sigma_{rg}^2}\right\} \qquad [10]$$

where z is the rate gyro output, a indicates which model describes the helicopter, m indexes the models, σ_{rg} is the standard deviation of the rate gyro measurement in hover, r_m is the rate gyro residual of the m_{th} model and t_i indicates the current time. The variable z can be expanded into a vector to include other measurements if the model changes in other states are known.

The probability that the helicopter is described by a particular model given the measurements and previous probability is calculated from

$$p_m(t_i) = \frac{f(z_i|a_m, z_{i-1}) \cdot p_m(t_{i-1})}{\sum_{j=1}^{3} f(z_i|a_j, z_{i-1}) \cdot p_j(t_{i-1})} \qquad [11]$$

where f is the probability density function described in equation [10]. The probabilities are bounded from below by 0.01 to prevent the MMAC method from fixating on a particular model. As seen in equation [11], this situation occurs once a probability becomes zero because no future probabilities for that model can be non-zero. Also in the

implementation it is necessary to ensure that the denominator of equation [11] does not go to zero causing the function to be undefined, if this situation occurs, the previous probability is maintained as the current probability. The probability is then used to compute the MMAC cyclic control by

$$u_{MMAC}(t_i) = \sum_{m=1}^{3} u_m(t_i) \cdot p_m(t_i) \qquad [12]$$

where u_m is the cyclic command generated by each of the models as shown in Table 3.

Table 3
Cyclic Command Generated by Individual Models

Model	Cyclic Command
Level Ground	$u_1 = -K_{hover} \cdot \hat{x}_{hover}$
Hill on Right	$u_2 = -K_{hover} \cdot \hat{x}_{hill\_right} + k \cdot collective\_cmd$
Hill on Left	$u_3 = -K_{hover} \cdot \hat{x}_{hill\_left} - k \cdot collective\_cmd$

The k in Table 3 indicates the gain found for the open loop cyclic controller. K_{hover} are the LQG hover compensator controller gains given in the Appendix. The $\hat{x}$'s in Table 3 indicate the state estimates given by the respective models. This collective gain is multiplied by a processed version of the collective command which is zero before the helicopter comes in contact with the ground and is maintained constant at the level when collective cannot raise any part of the gear off of the ground. The zeroing is done to ensure that an additional command is not inadvertently added while the helicopter is in the air. The clipping of the collective when the helicopter is heavy limits the additional command due to the collective input to prevent the additional value from saturating the roll cyclic command for an extended period of time. If the roll cyclic is held at its maximum value for an extended period of time while the blades are still producing lift, the head mechanics of the helicopter may be damaged.

Since the MMAC is intended to make corrections during landing, this compensator is turned on once the helicopter has enter the HOVER state in the Z State Machine (see Table 1). The model probabilities are initialized as the level/hover model having a probability of 0.90 while each of the hill models have a probability of 0.05 since

it is known that the helicopter is in hover. One difficulty with this method is that when landing on level ground, any slight disturbance which causes one side of the gear to hit before the other may cause the helicopter to choose one of the hill models instead of the level model. The only effect this has is exerting more cyclic than necessary to prevent further loss of the targeted position.

Figure 16 shows the residuals of each of the models and the corresponding probabilities as the helicopter is landing on an 10° slope to the right. Note that the model for the hill on the right has the smallest residuals which implies that the model is the most accurate predictor of the given models. By equation [10], the smaller residual has a larger probability density function which increases the probability associated with that model. In all trials of the MMAC the controller settled to a probability of 0.98 on one of the models without difficulty which indicates that the dynamics of these three models are different enough to reliably select one of the models[1].

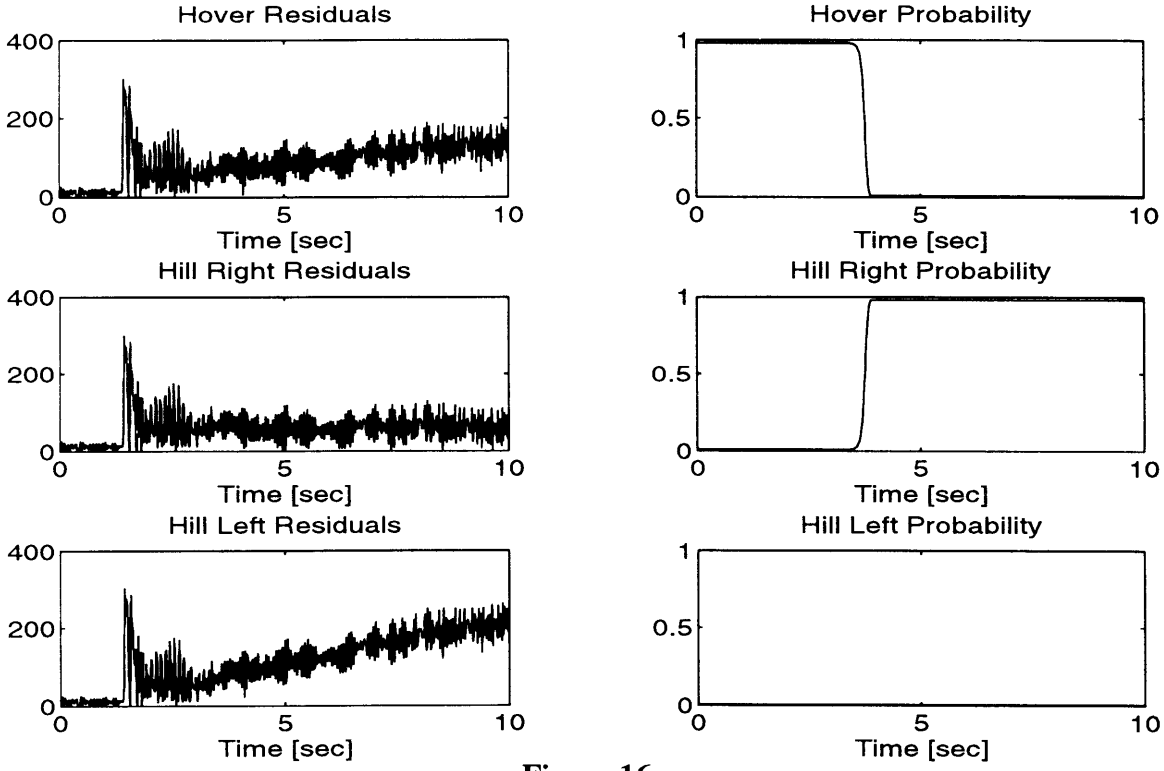


Figure 16
MMAC Residuals and Probabilities for Three Models on 10° Slope

[1]Maybeck paper discusses possible solutions if the adaptation mechanism has difficulty in determining which controller to weigh most heavily.

4.4.2 Performance of Multiple Model Adaptive Controller

Table 4 shows the mean and standard deviation of the landing accuracy for the given ground slopes. As with the LQG hover compensator, note that the farther away from level the slope is, the greater the position loss is. Figure 17 shows that the roll command does apply a command into the slope based on the displacements in the roll angle and y position states as well as the collective command, and this stops the loss of position down the slope.

Table 4
Mean Landing Accuracy for MMAC

Slope	Mean Landing Accuracy	Standard Deviation of Landing Accuracy
-4°	0.0153 m	0.0641 m
0°	-0.0580 m	0.0459 m
2°	-0.1454 m	0.0273 m
5°	-0.3433 m	0.0333 m
6°	-0.4746 m	0.0374 m
8°	-0.5385 m	0.0494 m
10°	-0.4923 m	0.0486 m
12°	-0.7996 m	0.1929 m

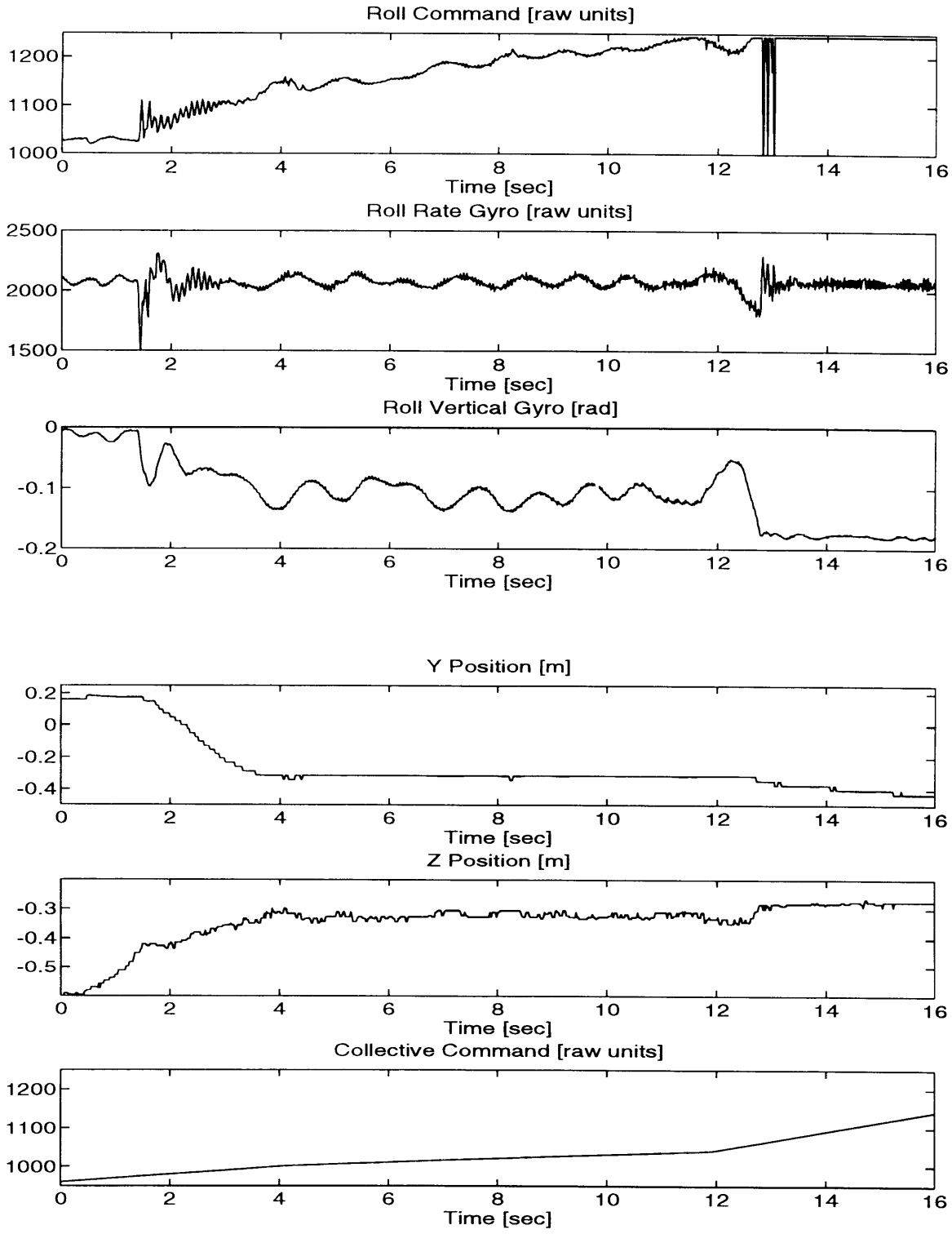


Figure 17
Multiple Model Adaptive Controller (MMAC) on 10° Slope

Figure 18 shows the MMAC controller landing on a 12° slope to the right compared to the LQG hover compensator landing on the same slope. The roll cyclic command generated by these two methods are at the limit of the roll cyclic command authority but the slope of the cyclic during the settle phase of the MMAC controller is steeper than that of the LQG hover compensator. Since both controllers are at the edge of the authority, neither controller has a very good performance as can be seen by the loss of position of nearly -0.6 m for both controllers. If there is a discernible difference between the performance of the controllers, the MMAC should do slightly better because it stops the loss of position sooner than the LQG hover compensator due to the faster cyclic control input.

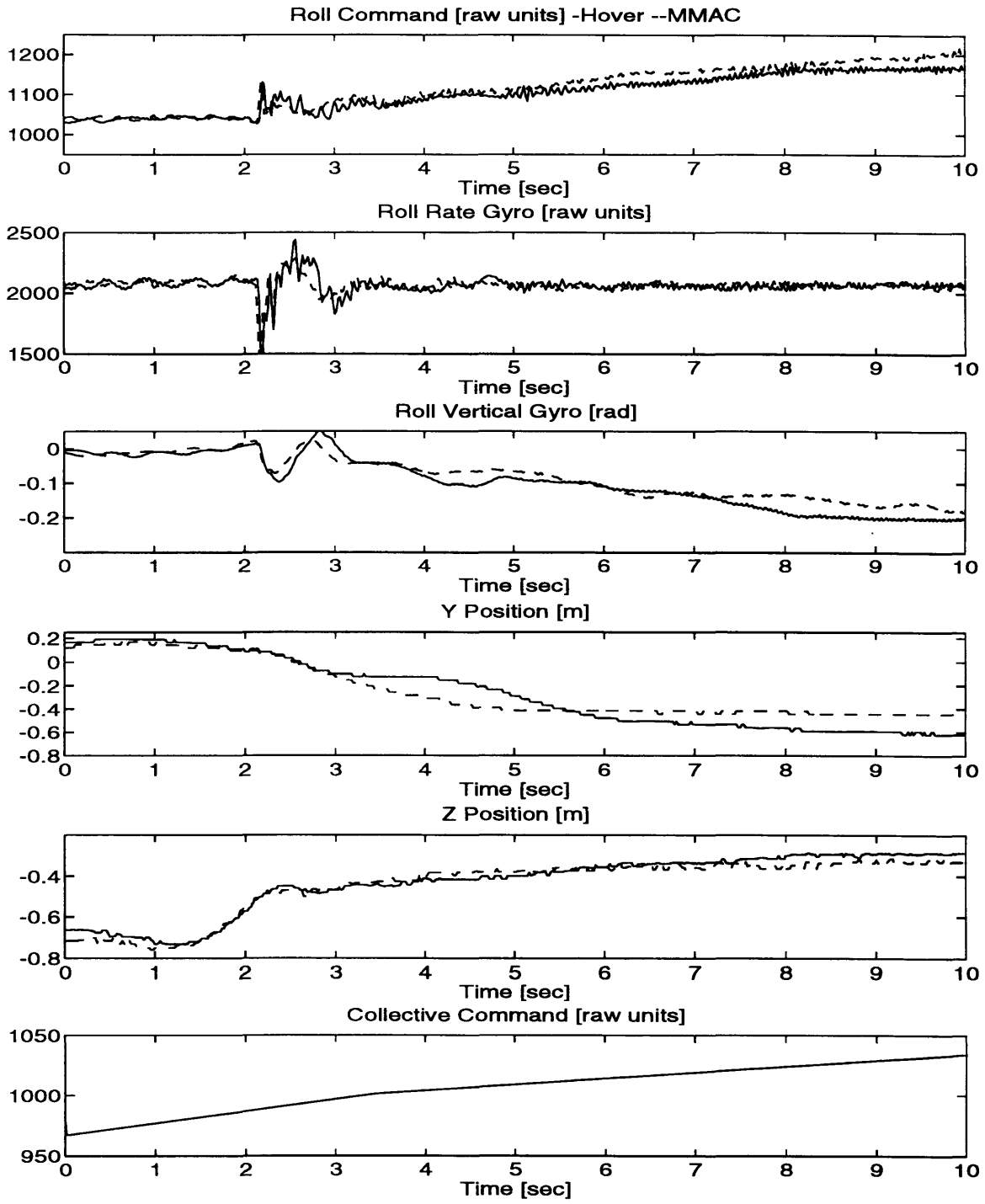


Figure 18
Comparison of LQG Hover Compensator and MMAC on 12° Slope

Figure 19 compares the performance of the LQG hover compensator with the performance of the MMAC controller using the mean landing accuracies. The error

bars indicate the maximum and minimum valued flight landing accuracies. The ideal performance is if the landing accuracy is zero across all ground slopes, so by looking at the best fit lines through the averages, the shallower MMAC line indicates better performance. Note that in the region less than 6°, the error bars overlap and the mean landing accuracies are nearly equal. The MMAC compensator only reverted to the level (hover) model for slopes less than or equal 2° in magnitude, so not much improvement is gained by the MMAC compensator in this range. Improvement in the MMAC compensator over the LQG hover compensator is seen for slopes between 6° and 12°; however, the cyclic command is saturated for 12° slopes and both compensators perform poorly and again the error bars overlap. For this setup, the MMAC controller landed the helicopter approximately 41 cm closer to the target than the LQG hover compensator on a 10° slope and approximately 26 cm closer to the target on an 8° slope.

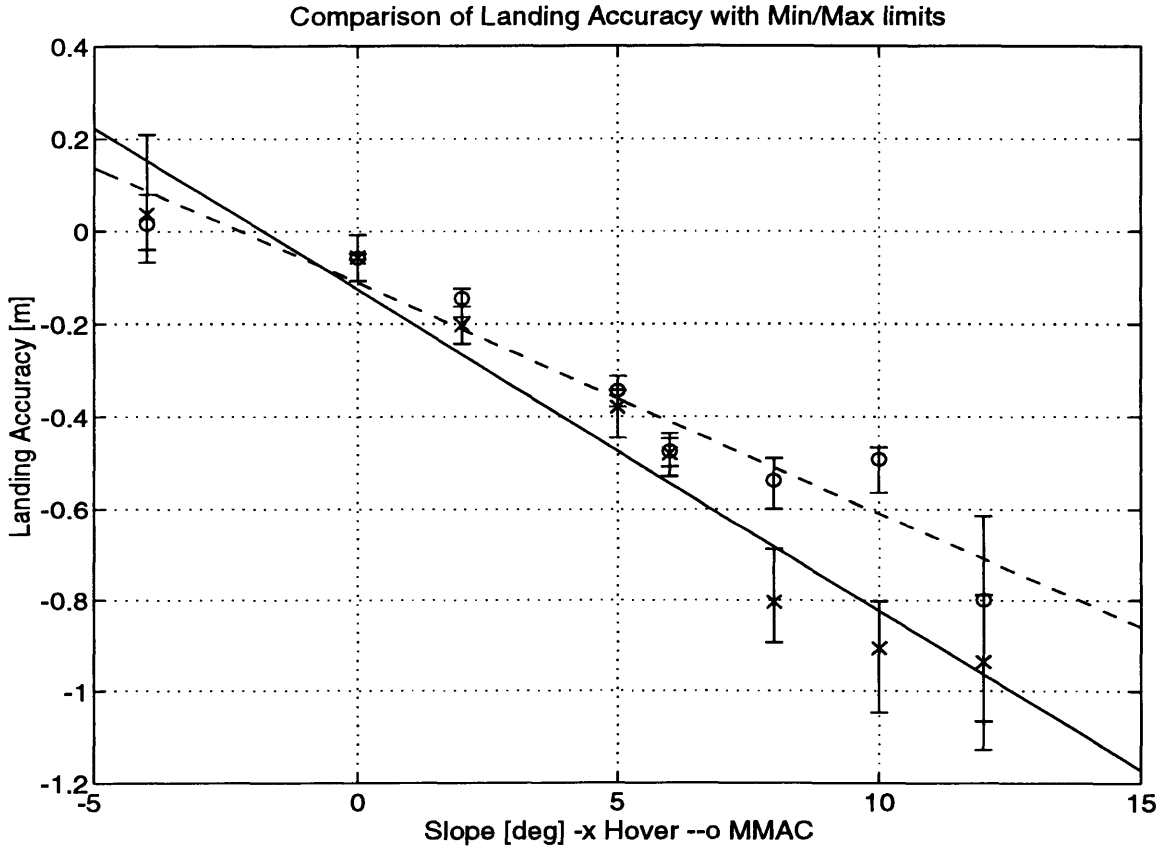


Figure 19
Performance of LQG Hover Compensator and MMAC vs. Slope

5 Conclusions

A model RC helicopter modified with a wider landing gear to increase the dynamic rollover margin landed on a series of slopes on an indoor platform. The goal was to land the helicopter on an unknown slope less than 12°. Since the specific characteristics of the slope are unknown, all strategies considered maintained the LQG hover compensator until the helicopter came into contact with the ground (i.e., no anticipatory strategies were considered). The helicopter descended upon a target position on the slope from directly over the target position such that its heading was perpendicular to the gradient of the slope. This means that all downhill movement of the helicopter is limited to the y-z body axis plane; therefore, each strategy only modified the roll cyclic command.

When attempting to land on a smooth, sloped surface without wind disturbances, the open loop cyclic command controller strategy was eliminated from consideration because it failed to stabilize the plant. The attitude weighted LQG hover compensator was eliminated because it excited unmodeled dynamics which caused oscillations. The two remaining strategies were either to adapt nothing and use the LQG hover compensator or to create a multiple model adaptive controller (MMAC) which uses the LQG hover compensator with an additional cyclic input proportional to the standardized collective input.

For the standardized collective command input with the descent rate adjusted to prevent the helicopter from bouncing upon touchdown, the LQG hover compensator and the MMAC compensator land approximately the same distance from a desired target on unknown slopes between ±6°. When the unknown slope is 12° or greater in magnitude, both the hover and the MMAC controller have saturated the roll cyclic. However, the MMAC compensator yields improvement in the mean landing accuracy on unknown slopes between 6° and 12° which is approximately 41 cm improvement on a 10° slope in this experimental setup.

Therefore, if it is known that all slopes encountered by the helicopter will be less than 6° in magnitude, then only the LQG hover compensator is necessary. However, if the slopes encountered are less than 12° in magnitude, then the MMAC compensator will provide improved performance at the steeper slopes (greater than 6°) while maintaining the hover compensator performance on slopes less than 6°.

Appendix Hover Model System Identification

The model of the RC helicopter in hover is decoupled into four single input models. The roll cyclic controls the roll rate and attitude and the translational velocity and position along the helicopter's body y-axis. Similarly, the pitch cyclic controls the pitch rate and attitude and the translational velocity and position along the helicopter's body x-axis. The rudder controls the yaw rate and attitude of the helicopter while the collective command controls the altitude rate and position of the helicopter. The following describes the model identification and gain selection process and includes the scripts for Matlab™ used in this process.

A.1 Spectral Analysis

The controlled autoregression model (ARX) assumes the input-output relationship is linear with a white noise disturbance. The identified model will be of the form

$$\begin{aligned} y(t) + a_1 y(t-1) + \ldots + a_{na} y(t-na) = \\ b_1 u(t-nk) + b_2 u(t-nk-1) + \ldots + b_{nb} u(t-nk-nb+1) + e(t) \end{aligned} \quad [13]$$

where *na* is the number of past output states used to calculate the output *y(t)*, *nb* is the number of control terms used, and *nk* indicates the number of delays before the input *u(t-nk)* effects the output *y(t)*. The identification process needs to identify the order of the model and the number of delays in the model as well as the actual parameter values.

Four different sets of pseudo-random perturbations filtered at 10 Hz were fed into each command input as the test pilot attempted to hold the helicopter in a hover. These ascii data sets were loaded into Matlab™ by scripts like the excerpt from load_roll.m shown here:

```
LOAD_ROLL.M
% ******* WARNING - the index numbers used here are 1 higher than the number
%         in vxworks
%*** Data files********
data1 = 'j00052'
data2 = 'j00053'
data3 = 'j00054'
```

```
data4 = 'j00051'

%%%%%%%%%% Load first data1 set %%%%%%%%%%
n_channels = 200;
%file_name = input( 'Type file name to load: ', 's' );
file_name = data1


open_err = sprintf( 'Could not fopen file %s', file_name );
file_size = 1000*n_channels;

fp = fopen( file_name, 'r' );

if fp == -1 | fp == 2
  open_err
end

% Now load in data1 set

[dat1, count] = fread( fp, [n_channels, 1000], 'float');
dat1 = dat1';

if count ~= file_size
  read_err = sprintf( 'Error reading file %s, count = %d', file_name, count )
else
  check = sprintf( 'File %s read successfully', file_name )
end

fclose(fp);

% ************    COMMANDS    ***************
rc1 = dtrend(dat1(:,6)) ;

%**********    MEASUREMENTS    **********************

vg1 = dtrend(dat1(:,53));
rg1 = dtrend(dat1(:,14));
yv1 = dtrend(dat1(:,43));
raw_vg1 = dtrend(dat1(:,17));

clear dat1

%% This process repeats for each data set loaded.
```

The identification process starts with a spectral analysis of the data which shows the consistency of the data and if filtering is necessary. Figure 20 shows a spectral analysis of four different data sets for the roll cyclic command to roll rate gyro. Note the low frequency consistency between the data sets and the resonance at 1.5 rad/sec. Also note that above 50 rad/sec both the magnitude and phase plots are no longer consistent. This is because there was very little information available in the data at these frequencies (almost none since the input was filtered at 10 Hz). To keep the model from attempting

to identify these higher frequency characteristics that may not be accurate due to the lack of data, a 5th order, zero phase filter is used to remove this information from the data.

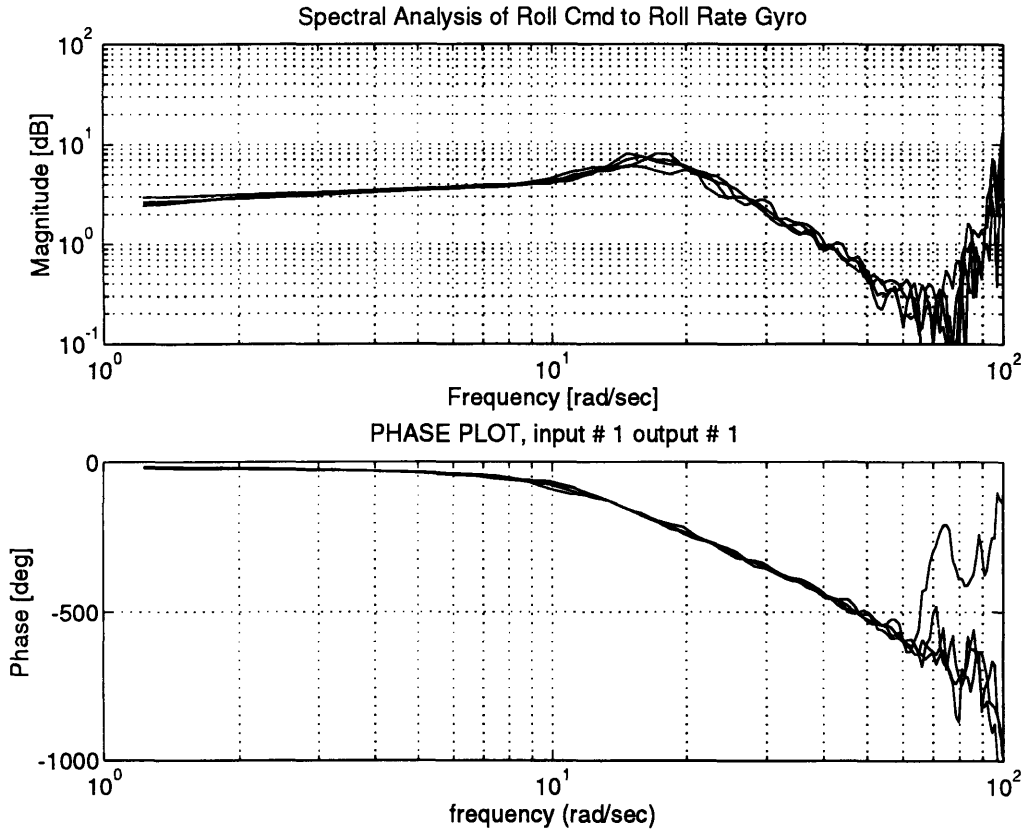


Figure 20
Spectral Analysis of the Roll Command to the Roll Rate Gyro

Figure 21 shows a comparison of the spectral analysis of the filtered data compared to the unfiltered data. Note that the high frequency information has been removed (indicated by the flat region at 0 dB at the higher frequencies). The rcg_spa.m script follows which peformed this analysis and generated the spectral analysis plots. Similar scripts were written for each input-output relationship identified, and Table 5 shows the filtering frequencies used for each of the input-output relationships.

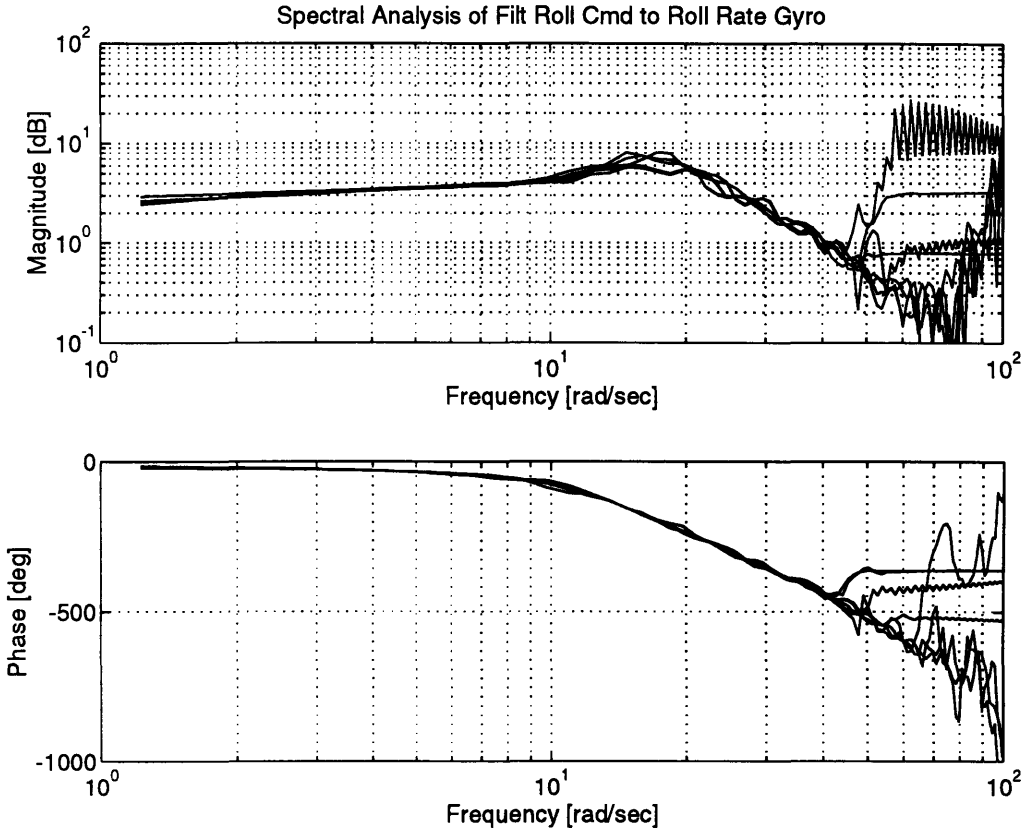


Figure 21
Spectral Analysis Comparing Filtered to Unfiltered Data

RCG_SPA.M

```
% Matlab script which performs spectral analysis

% Setup output- input vectors
z1 = [rg1 rc1];
z2 = [rg2 rc2];
z3 = [rg3 rc3];
z4 = [rg4 rc4];

% Spectral Analysis on unfiltered data
spa1 = spa(z1,128,-1,-1,0.02);
spa2 = spa(z2,128,-1,-1,0.02);
spa3 = spa(z3,128,-1,-1,0.02);
spa4 = spa(z4,128,-1,-1,0.02);

% Filter Data with Zero Phase Filter
filt_ord = 5;
%wn = [1 6]/25;    %cutoff frequency in fractions of Nyquist freq
wn = 6/25;         %cutoff frequency in fractions of Nyquist freq

cf1 = idfilt(rc1,filt_ord,wn);
cf2 = idfilt(rc2,filt_ord,wn);
cf3 = idfilt(rc3,filt_ord,wn);
cf4 = idfilt(rc4,filt_ord,wn);

gf1 = idfilt(rg1,filt_ord,wn);
gf2 = idfilt(rg2,filt_ord,wn);
```

```
gf3 = idfilt(rg3,filt_ord,wn);
gf4 = idfilt(rg4,filt_ord,wn);

% Setup output- input filtered vectors
zf1 = [gf1 cf1];
zf2 = [gf2 cf2];
zf3 = [gf3 cf3];
zf4 = [gf4 cf4];

spaf1 = spa(zf1,128,-1,-1,0.02);
spaf2 = spa(zf2,128,-1,-1,0.02);
spaf3 = spa(zf3,128,-1,-1,0.02);
spaf4 = spa(zf4,128,-1,-1,0.02);


% do some plotting
figure(1)
bodeplot([spa1 spa2 spa3 spa4])
subplot(211)
title('Spectral Analysis of Roll Cmd to Roll Rate Gyro')
ylabel('Magnitude [db]')
xlabel ('Frequency [rad/sec]')
grid on
subplot(212)
grid on
ylabel('Phase [deg]')

figure(2)
bodeplot([spaf1 spaf2 spaf3 spaf4])
subplot(211)
title('Spectral Analysis of Filt Roll Cmd to Roll Rate Gyro')
ylabel('Magnitude [db]')
xlabel ('Frequency [rad/sec]')
subplot(212)
grid on
ylabel('Phase [deg]')
```

Table 5
Filtering Frequencies for Each Input-Output Relationship

Input-Output Relation	Cutoff Frequency of Low Pass Filter ω_n
Roll Cmd-Roll Rate	6 Hz
Roll Rate-Roll Angle	7.5 Hz
Roll Angle-Y Velocity	3 Hz
Pitch Cmd-Pitch Rate	3 Hz
Pitch Rate-Pitch Angle	5 Hz
Pitch Angle-X Velocity	3 Hz
Rudder Cmd-Yaw Rate	5 Hz
Yaw Rate-Yaw Angle	1 Hz
Collective Cmd - Z Velocity	5 Hz

The following are the full series of filtered spectral analysis plots for each identified input-output relationship.

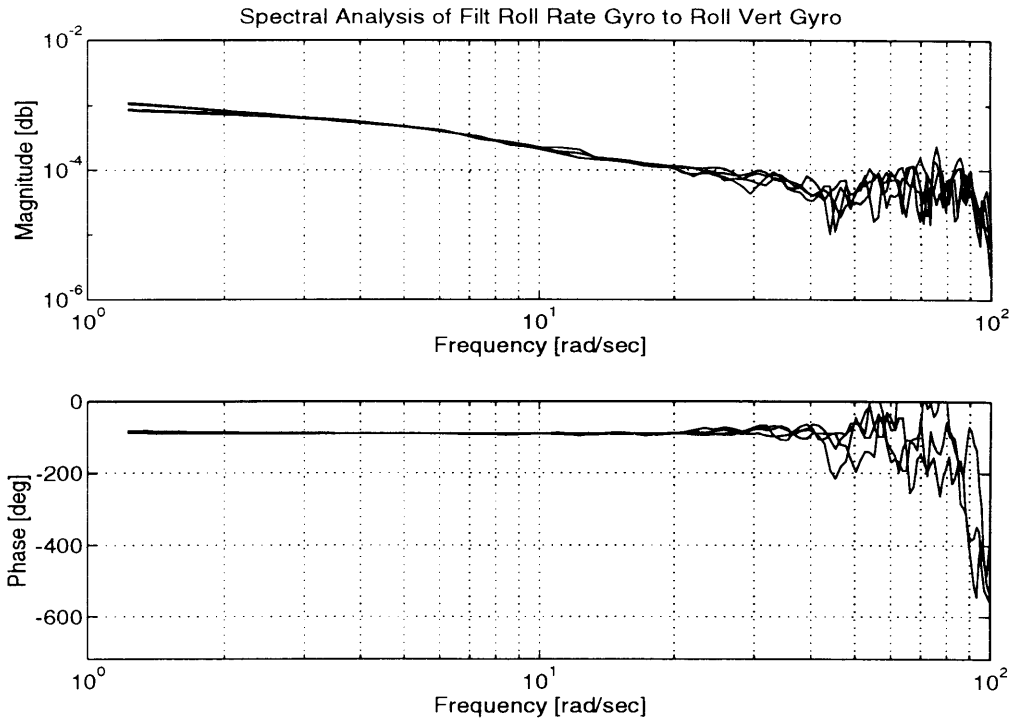


Figure 22
Spectral Analysis of Roll Rate to Roll Angle

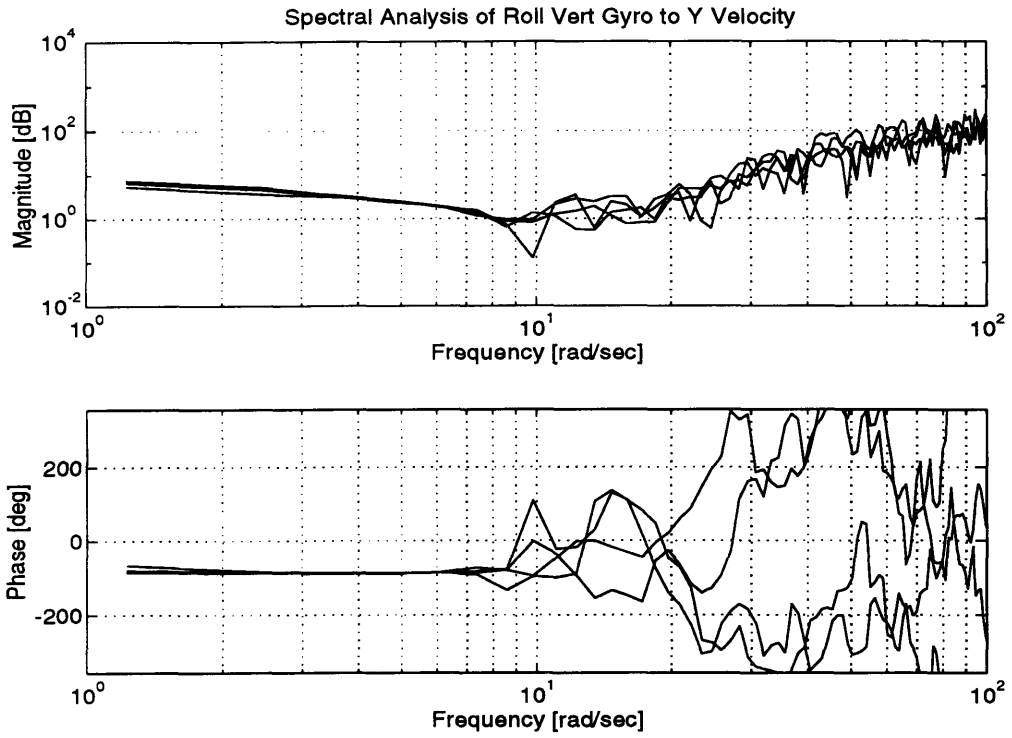


Figure 23
Spectral Analysis of Roll Angle to Y Velocity

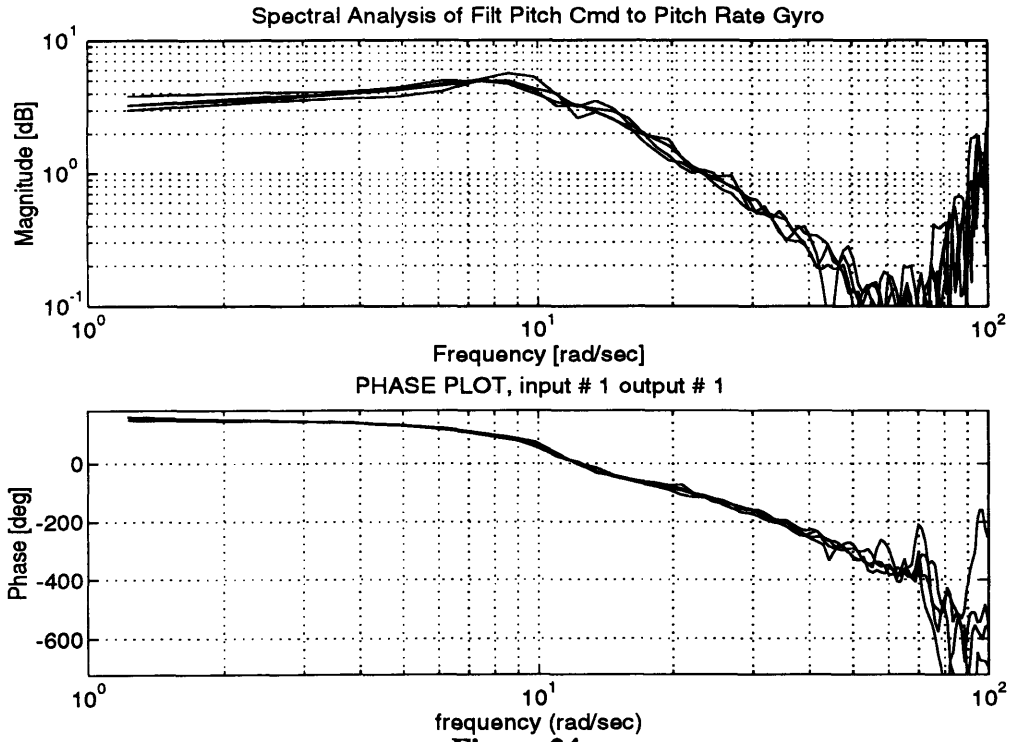


Figure 24
Spectral Analysis of Pitch Command to Pitch Rate

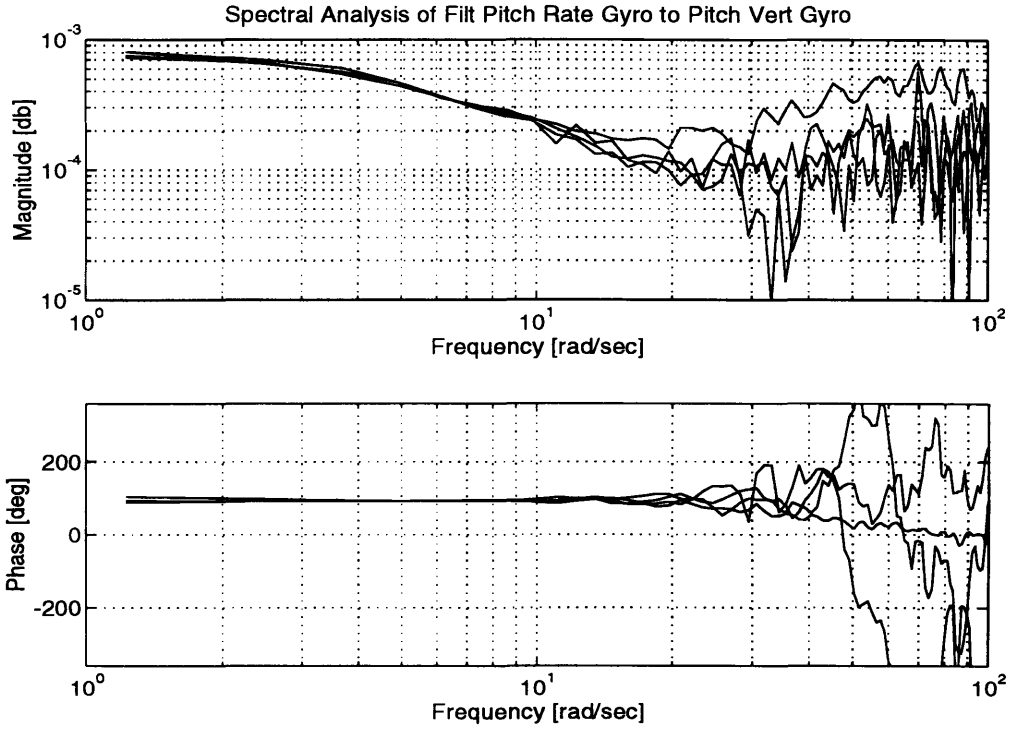


Figure 25
Spectral Analysis of Pitch Rate to Pitch Angle

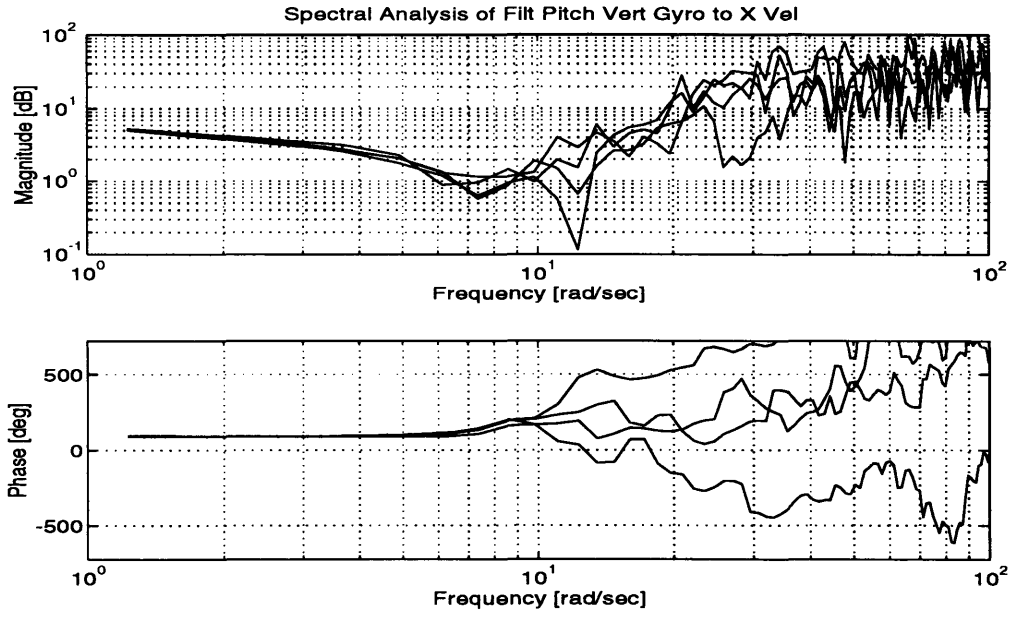


Figure 26
Spectral Analysis of Pitch Angle to X Velocity

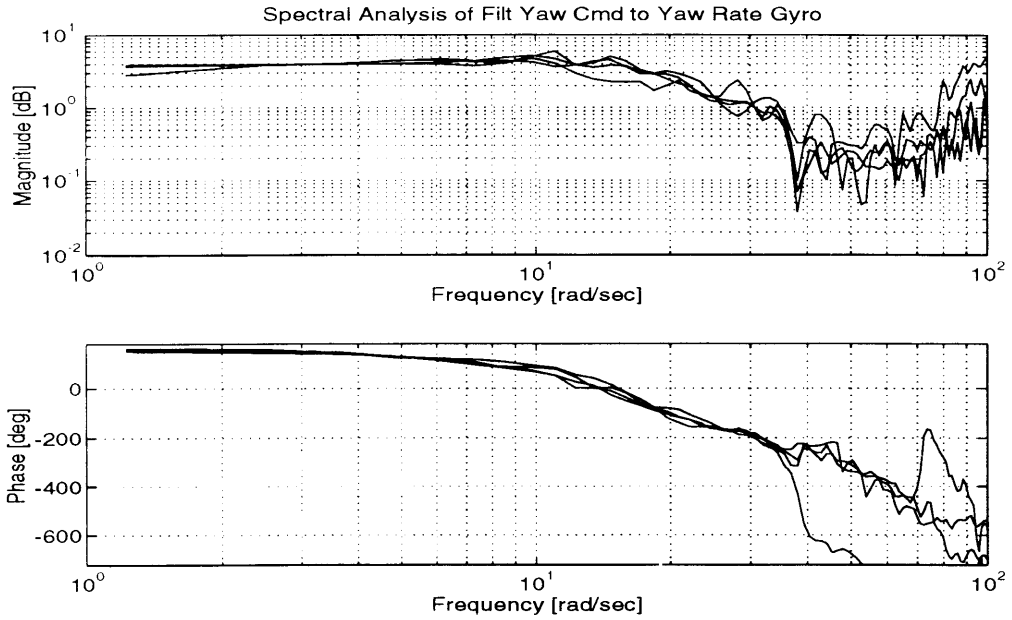


Figure 27
Spectral Analysis of Rudder Command to Yaw Rate

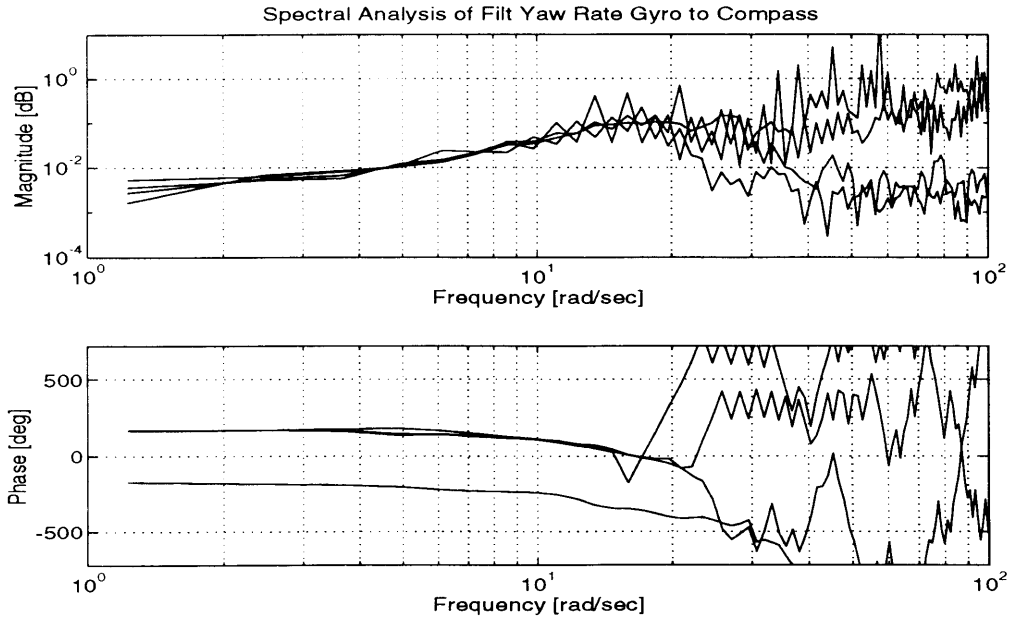


Figure 28
Spectral Analysis of Yaw Rate to Yaw Angle

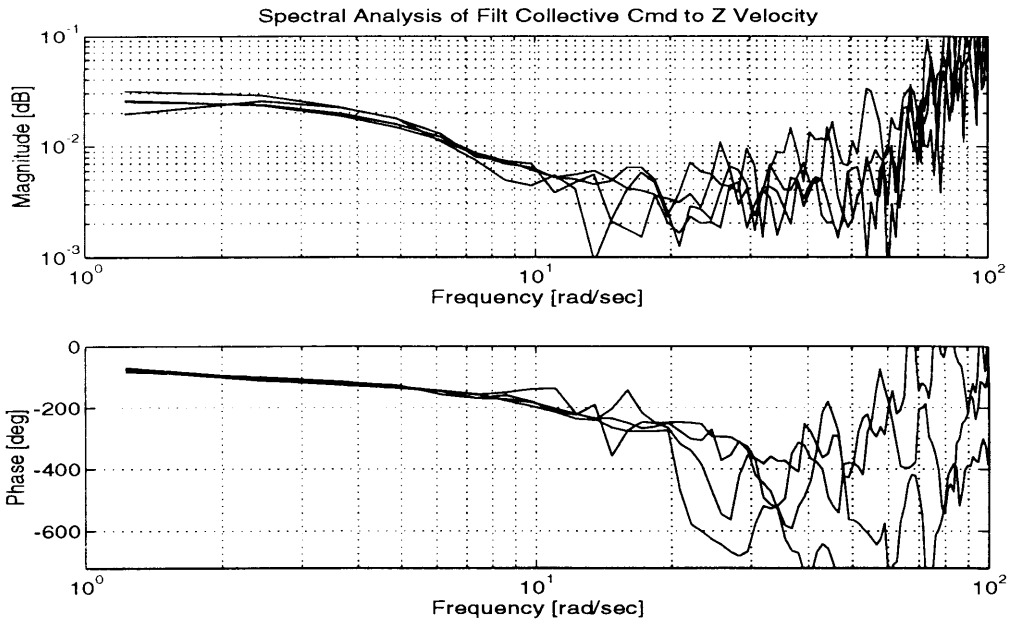


Figure 29
Spectral Analysis of Collective Command to Z Velocity

A.2 Model Structure

The next step in the system identification process is to select a model structure which is the number of delays and number of input and past output terms that will satisfactorily predict the behavior of the system. The script rcg_model.m shows part of the model structure selection procedure for the roll command to roll rate gyro model. First, the data is filtered at the frequencies selected in spectral analysis, but note that this entire process is iterative and that if a suitable model is not attained, the filter cutoff frequency may be changed and the spectral analysis and model structure selection processes repeated. The Matlab™ command arxstruc creates a series of ARX models based on one set of data and calculates a sum of the squared prediction errors (a loss function) generated in cross-validation. The Matlab™ command selstruc selects the "best" model structure based on minimization of the prediction errors and the order of the model[1].

```
RCG_MODEL.M
% this code creates the arx  model
% run rcg_spa  first to specify variable names

filt_ord = 5;
wn = 6/25;          %cutoff frequency in fractions of Nyquist freq

cf1 = idfilt(rc1,filt_ord,wn);
gf1 = idfilt(rg1,filt_ord,wn);

zf1 = [gf1 cf1];

% testing use of arxstruc
na = 1:4;
nb1 = 1:4;
nk1 = 1:8;

NN = make_nn(na,nb1,nk1);

V1 = arxstruc(zf1,z2,NN);
nn1 = selstruc(V1,0)

V1a = arxstruc(z1,z2,NN);
nn1a = selstruc(V1a,0)


V2 = arxstruc(zf1,z3,NN);
nn2 = selstruc(V2,0)

V2a = arxstruc(z1,z3,NN);
nn2a = selstruc(V2a,0)
```

[1]For details on the Matlab™ commands, see the System Identification Toolbox for Use with Matlab™.

```
V3 = arxstruc(zf1,z4,NN);
nn3 = selstruc(V3,0)

V3a = arxstruc(z1,z4,NN);
nn3a = selstruc(V3a,0)

% create other arx models for comparison
th1 = arx( zf1, nn1);
th1 = sett(th1,0.02);

th2 = arx( zf1, nn1a);
th2 = sett(th2,0.02);

th3 = arx( zf1, nn2);
th3 = sett(th3,0.02);

th4 = arx( zf1, nn2a);
th4 = sett(th4,0.02);

th5 = arx( zf1, nn3);
th5 = sett(th5,0.02);

th6 = arx( zf1, nn3a);
th6 = sett(th6,0.02);

% Generate frequency response
fth1 = trf(th1);
fth2 = trf(th2);
fth3 = trf(th3);
fth4 = trf(th4);
fth5 = trf(th5);
fth6 = trf(th6);

figure(2)
clf
bodeplot([spa1 spa2 spa3 spa4 fth1 fth2 fth3 fth4 fth5 fth6]);
grid on
subplot(211)
grid on
title('Spa and ARX Frequency Response')
ylabel('Magnitude [dB]')
xlabel('Frequency [rad/sec]')
subplot(212)
title('')
ylabel('Phase [deg]')
xlabel('Frequency [rad/sec]')
```

For each of the structures, the frequency response of the each of the selected models is compared to the spectral analysis of the data sets as shown in Figure 30. Note that some of the models do not capture the desired frequency characteristics. Since selstruc only selects one model, it is sometimes useful to skim off structures generated by the arxstruc command that are close to the loss function value of the "best" model selected and look for a structure that is consistent across all three cross-validations. A model that may be the best on one cross-validation may not perform as well on a different

cross-validation, and so it is better to choose a model structure that appears consistently through all of the cross-validations.

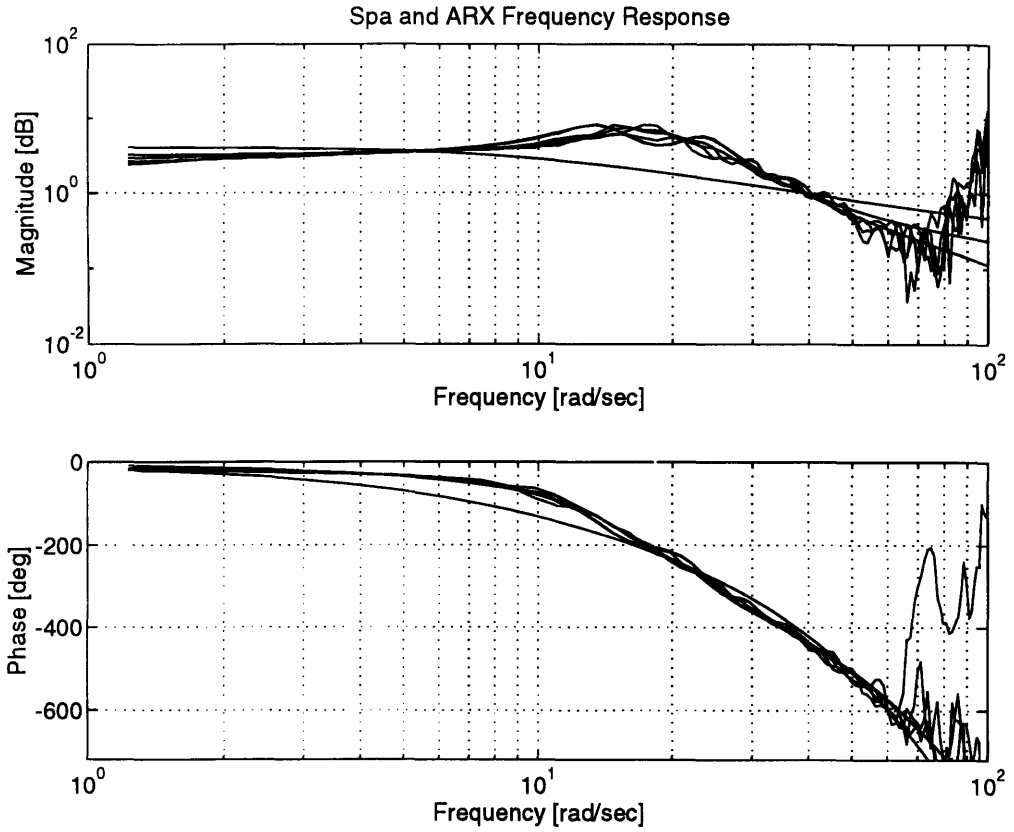


Figure 30
Spectral Analysis and ARX Frequency Response of Roll Command to Roll Rate

The time simulations generated by rcg_sim.m for the roll command to roll rate gyro relationship aid in the model selection process. Since it is difficult to tell the difference in performance based on a single step ahead prediction with output (rate gyro measurement) corrections at each time step, the Matlab™ command compare.m is used to obtain the pure simulation based on the input (roll command) vector only. The difference between the measured output (roll rate gyro) and the predicted output is used to calculate the fit[1]. The first plot shown in Figure 31 shows three different model structure simulations with the filtered measured output from which the models were created. The

[1] See Matlab™ documentation for details.

fits of each model are indicated above the plot. The three remaining plots compare how these models work on different data sets in cross validation. Although the [2 2 6] model structure may not give the best fit across the data sets, this structure has been selected for the command to attitude rate relationship since it consistently appears close in the sense of fit and loss function values but is generally discounted by the selstruc command because of its higher order. During implementation, the [2 2 6] model structure for the command to rate gyro on both the roll and pitch axes has proved to consistently work where some of the lower order structures have failed. Table 6 shows the structures selected for each of the input-output relationships.

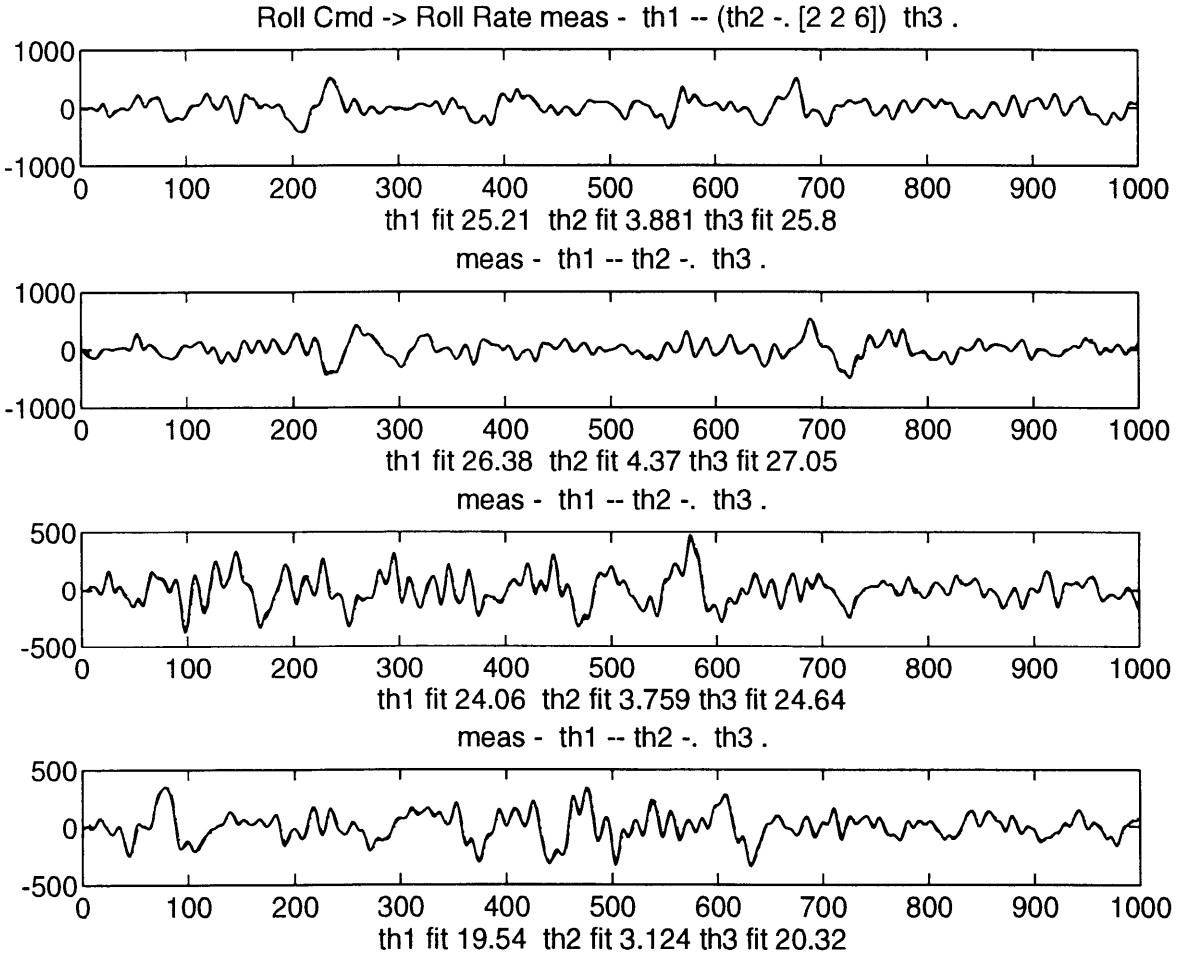


Figure 31
Time Simulations of Roll Command to Roll Rate Gyro

Table 6
Model Structures for Each Input-Output Relation

Input-Output Relation	[na nb nk]
Roll Command - Roll Rate	[2 2 6]
Roll Rate - Roll Angle	[1 1 1]
Roll Angle - Y Velocity	[1 1 1]
Y Velocity - Y Position	[1 1 1]
Pitch Command - Pitch Rate	[2 2 6]
Pitch Rate - Pitch Angle	[1 1 1]
Pitch Angle - X Velocity	[1 1 1]
X Velocity - X Position	[1 1 1]
Rudder Command - Yaw Rate	[2 1 8]
Yaw Rate - Yaw Angle	[1 1 1]
Collective Command - Z Velocity	[1 1 10]
Z Velocity - Z Position	[1 1 1]

```
RCG_SIM.M
% Choose models for comparison
% Roll Command to Roll Rate Gyro Model
th1 = arx( zf1, [1 3 6]);
th2 = arx( zf1, [2 2 6]);
th3 = arx( zf1, [1 1 7]);

th1 = sett(th1,0.02);
th2 = sett(th2,0.02);
th3 = sett(th3,0.02);

figure(4)
clf
subplot(411)
% Time Simulation and Fit Calculation
[yh1,fit1] = compare(zf1,th1,1);
[yh2,fit2] = compare(zf1,th2,1);
[yh3,fit3] = compare(zf1,th3,1);

plot(zf1(:,1),'y-')
hold on
plot(yh1,'m--')
plot(yh2,'c-.')
plot(yh3,'g:')
title('Roll Cmd -> Roll Rate meas -  th1 -- (th2 -. [2 2 6])  th3 . ');
xlabel(['th1 fit ',num2str(fit1),'  th2 fit ',num2str(fit2),' th3 fit ',num2str(fit3)]);
hold off

subplot(412)
[yh1,fit1] = compare(zf2,th1,1);
[yh2,fit2] = compare(zf2,th2,1);
[yh3,fit3] = compare(zf2,th3,1);

plot(zf2(:,1),'y-')
hold on
plot(yh1,'m--')
plot(yh2,'c-.')
plot(yh3,'g:')
title('meas -  th1 -- th2 -.  th3 .');
xlabel(['th1 fit ',num2str(fit1),'  th2 fit ',num2str(fit2),' th3 fit ',num2str(fit3)]);
hold off
```

```
subplot(413)
[yh1,fit1] = compare(zf3,th1,1);
[yh2,fit2] = compare(zf3,th2,1);
[yh3,fit3] = compare(zf3,th3,1);

plot(zf3(:,1),'y-')
hold on
plot(yh1,'m--')
plot(yh2,'c-.')
plot(yh3,'g:')
title('meas -  th1 -- th2 -.  th3 .');
xlabel(['th1 fit ',num2str(fit1),'  th2 fit ',num2str(fit2),' th3 fit ',num2str(fit3)]);
hold off

subplot(414)
[yh1,fit1] = compare(zf4,th1,1);
[yh2,fit2] = compare(zf4,th2,1);
[yh3,fit3] = compare(zf4,th3,1);

plot(zf4(:,1),'y-')
hold on
plot(yh1,'m--')
plot(yh2,'c-.')
plot(yh3,'g:')
title('meas -  th1 -- th2 -.  th3 .');
xlabel(['th1 fit ',num2str(fit1),'  th2 fit ',num2str(fit2),' th3 fit ',num2str(fit3)]);
hold off

figure(3)
clf
fth1 = frf(th1);
fth2 = frf(th2);
fth3 = frf(th3);

bodeplot([spa1 spa2 spa3 spa4 fth1 fth2 fth3 ]);
grid on
subplot(211)
grid on
title('Spa and ARX Frequency Response')
ylabel('Magnitude [dB]')
xlabel('Frequency [rad/sec]')
axis([1e0 1e2 1e-2 1e2])
subplot(212)
title('')
ylabel('Phase [deg]')
xlabel('Frequency [rad/sec]')
axis([1e0 1e2 -720 0])
```

The entire series of time simulation plots is shown here.

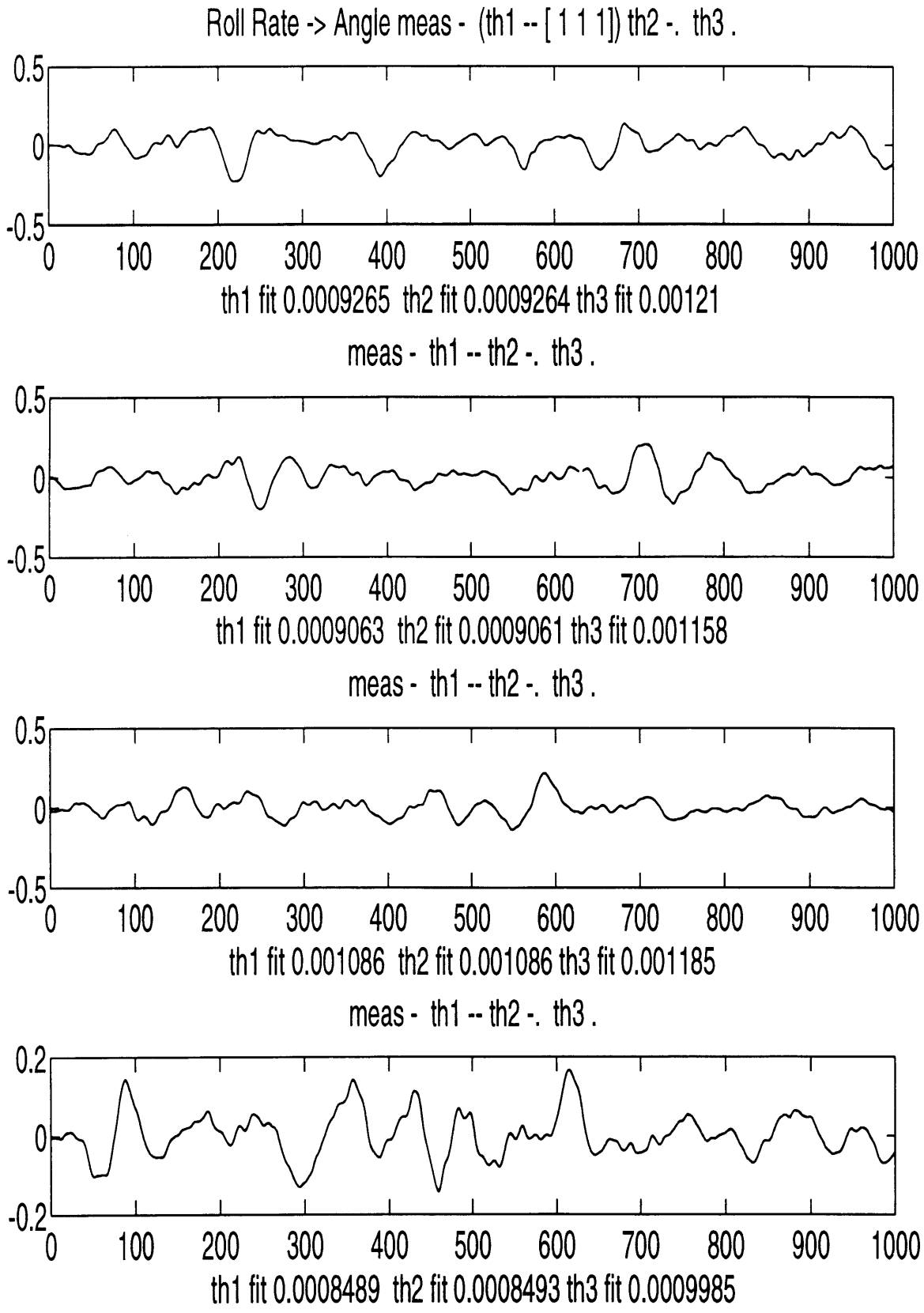


Figure 32
Time Simulation of Roll Rate to Roll Angle

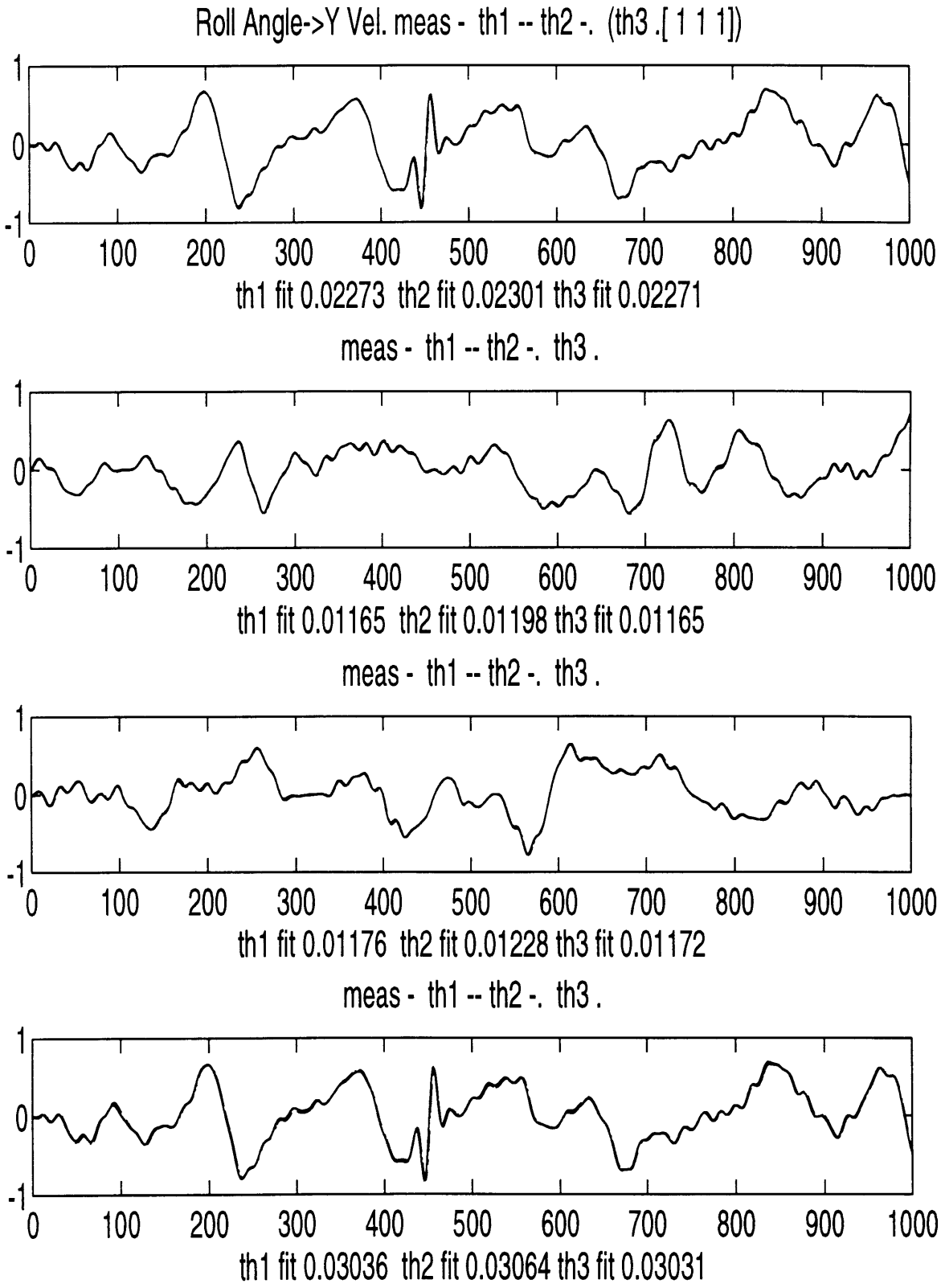


Figure 33
Time Simulation of Roll Angle to Y Velocity

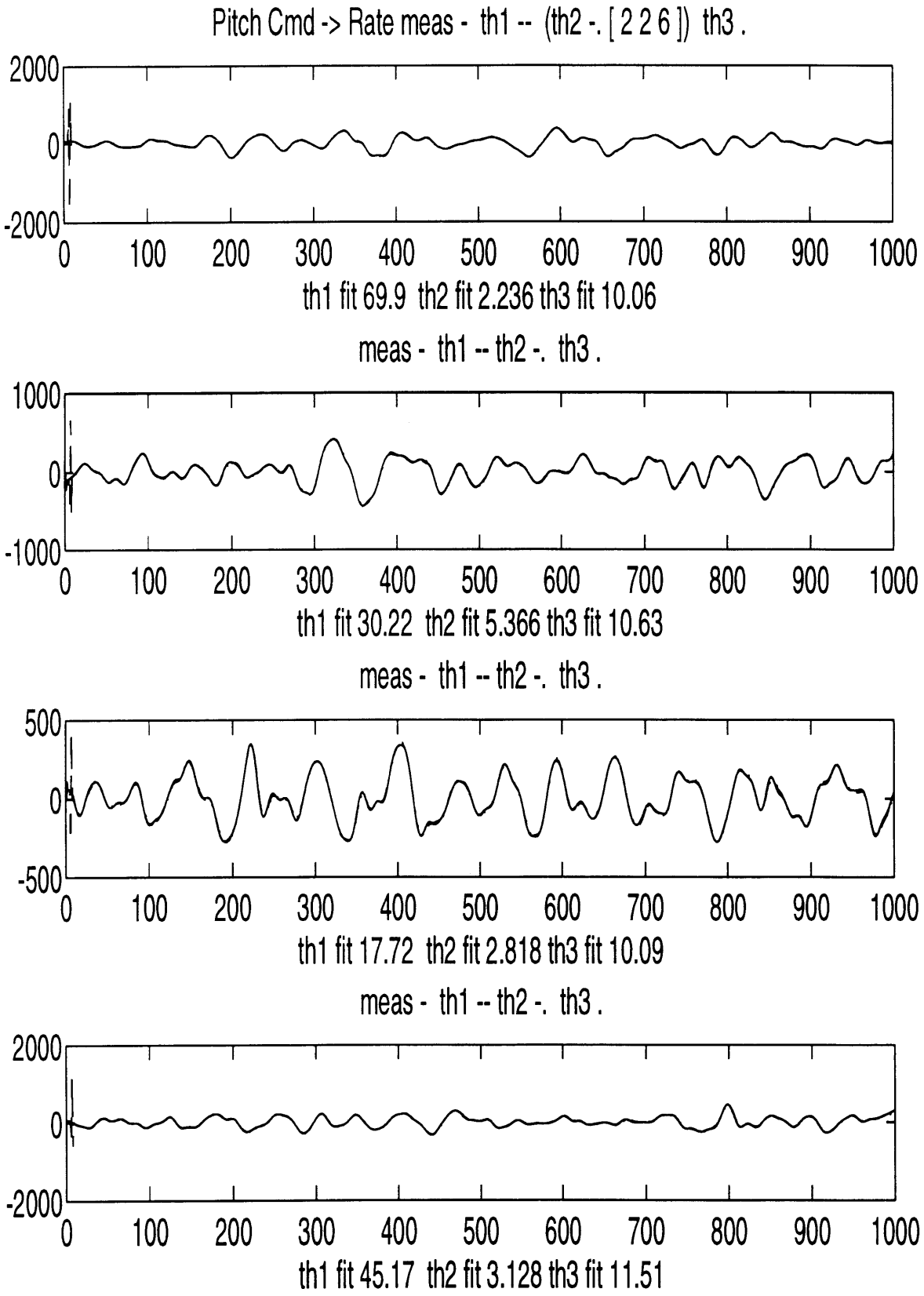


Figure 34
Time Simulation of Pitch Command to Pitch Gyro

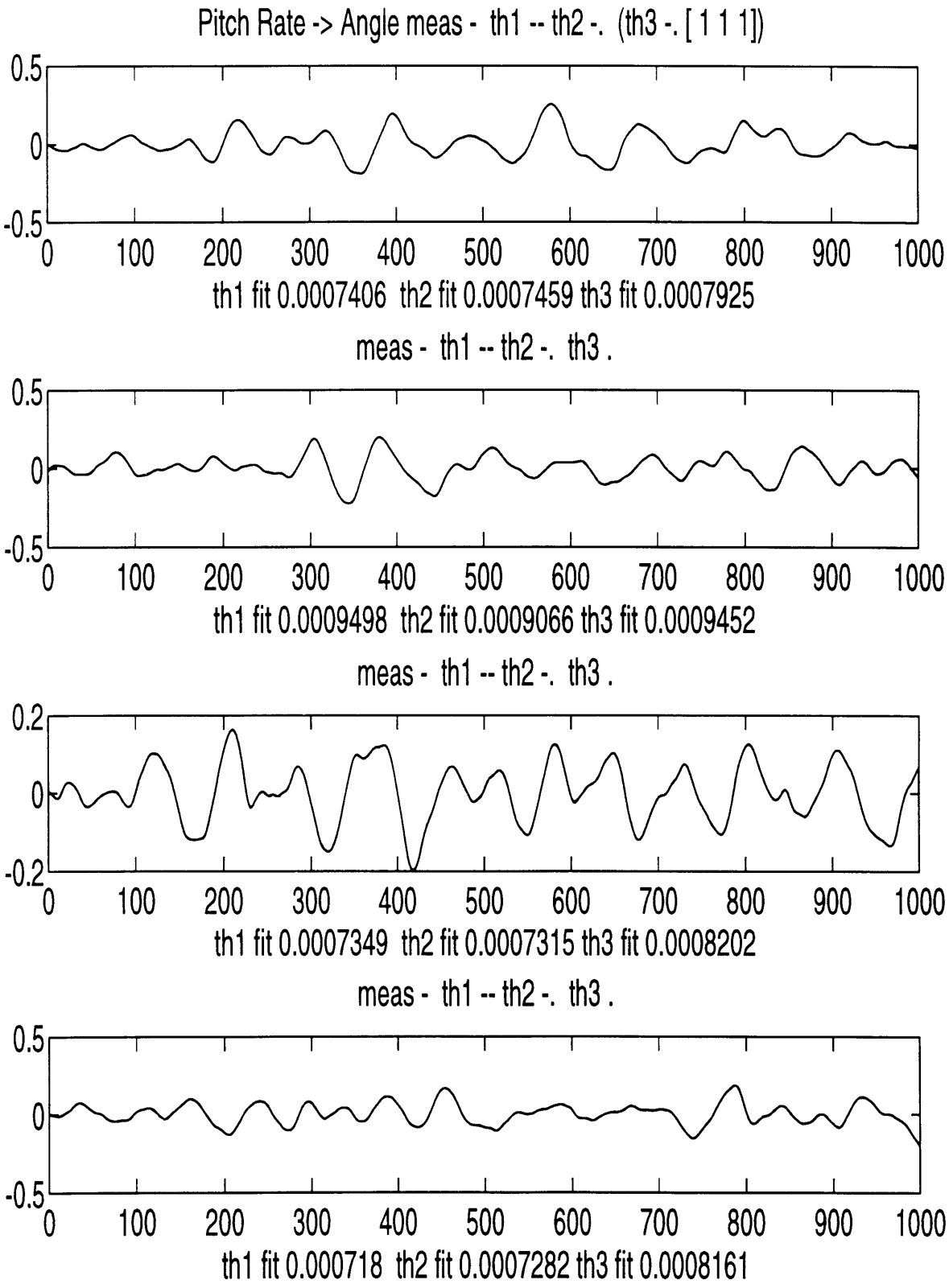


Figure 35
Time Simulation of Pitch Rate to Pitch Angle

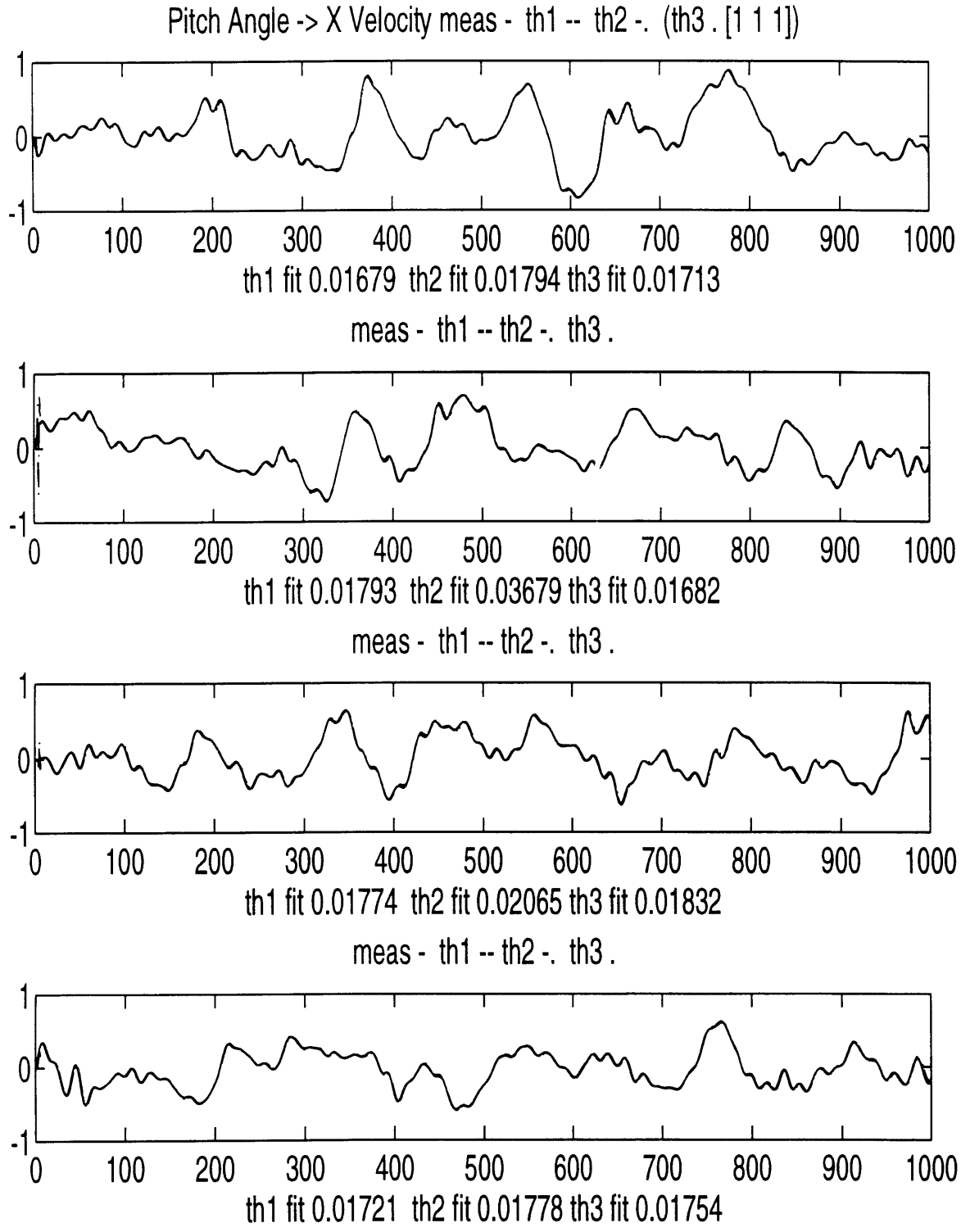


Figure 36
Time Simulation of Pitch Angle to X Velocity

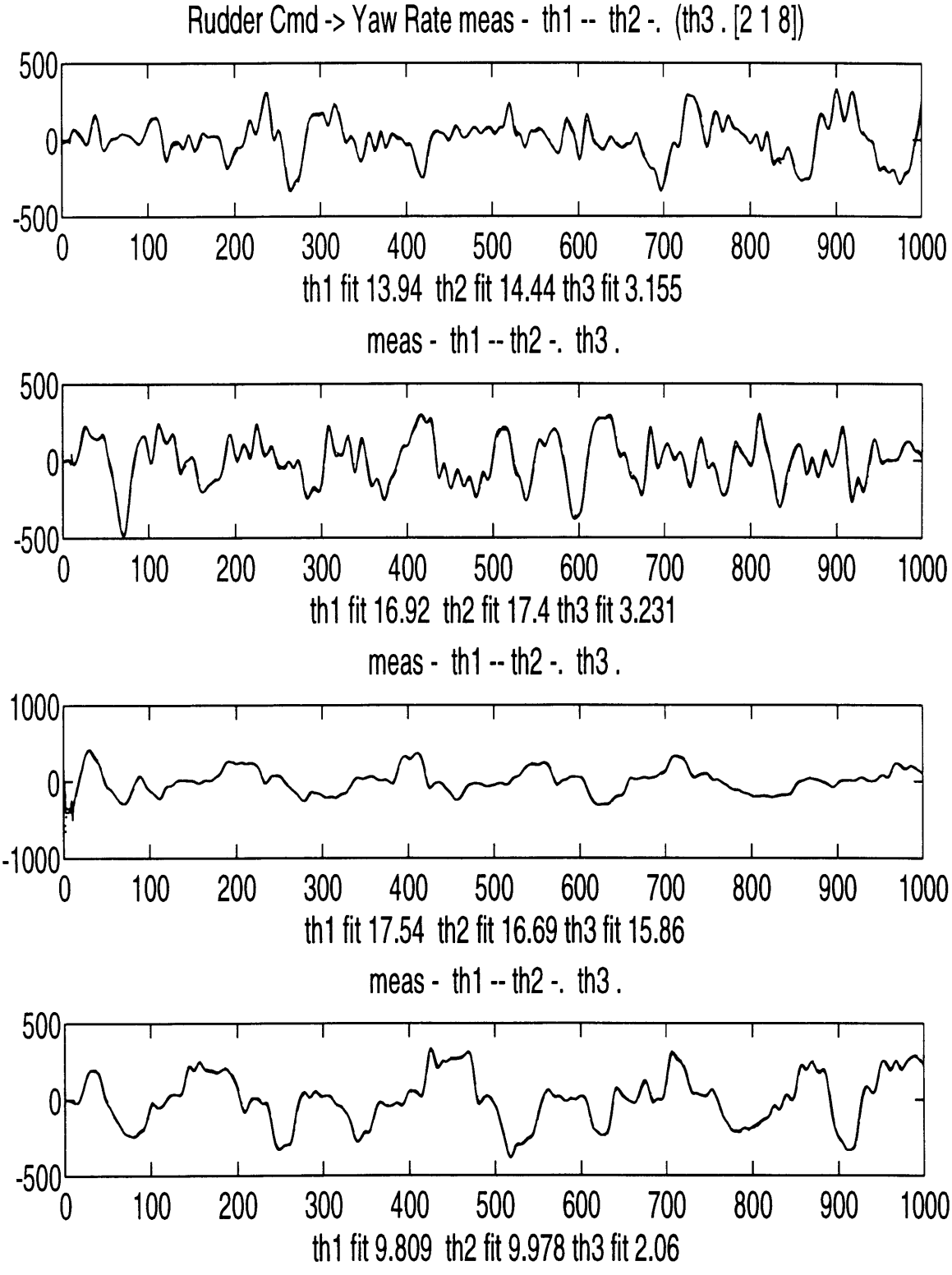


Figure 37
Time Simulation of Rudder Command to Yaw Rate

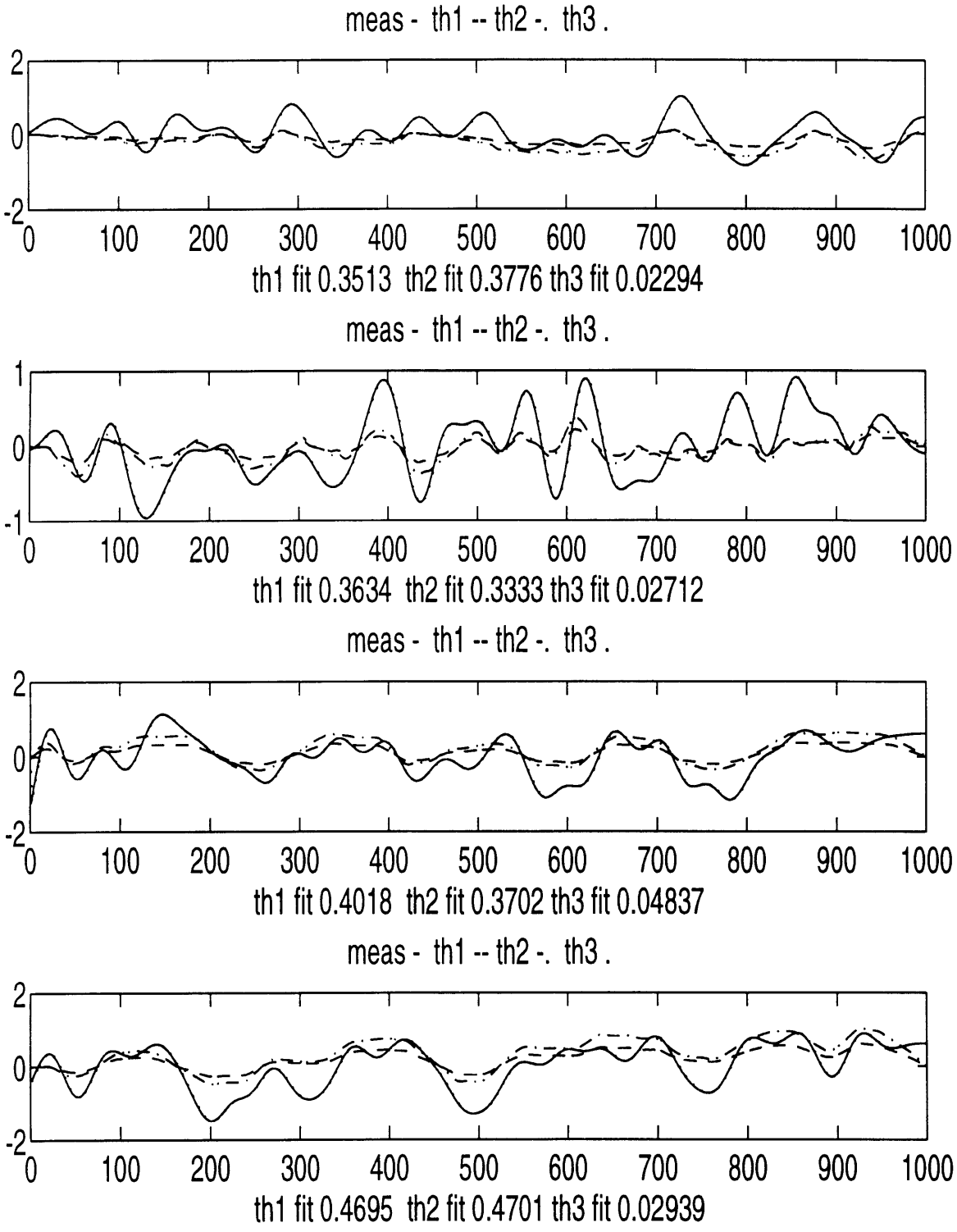


Figure 38
Time Simulation of Yaw Rate to Yaw Angle

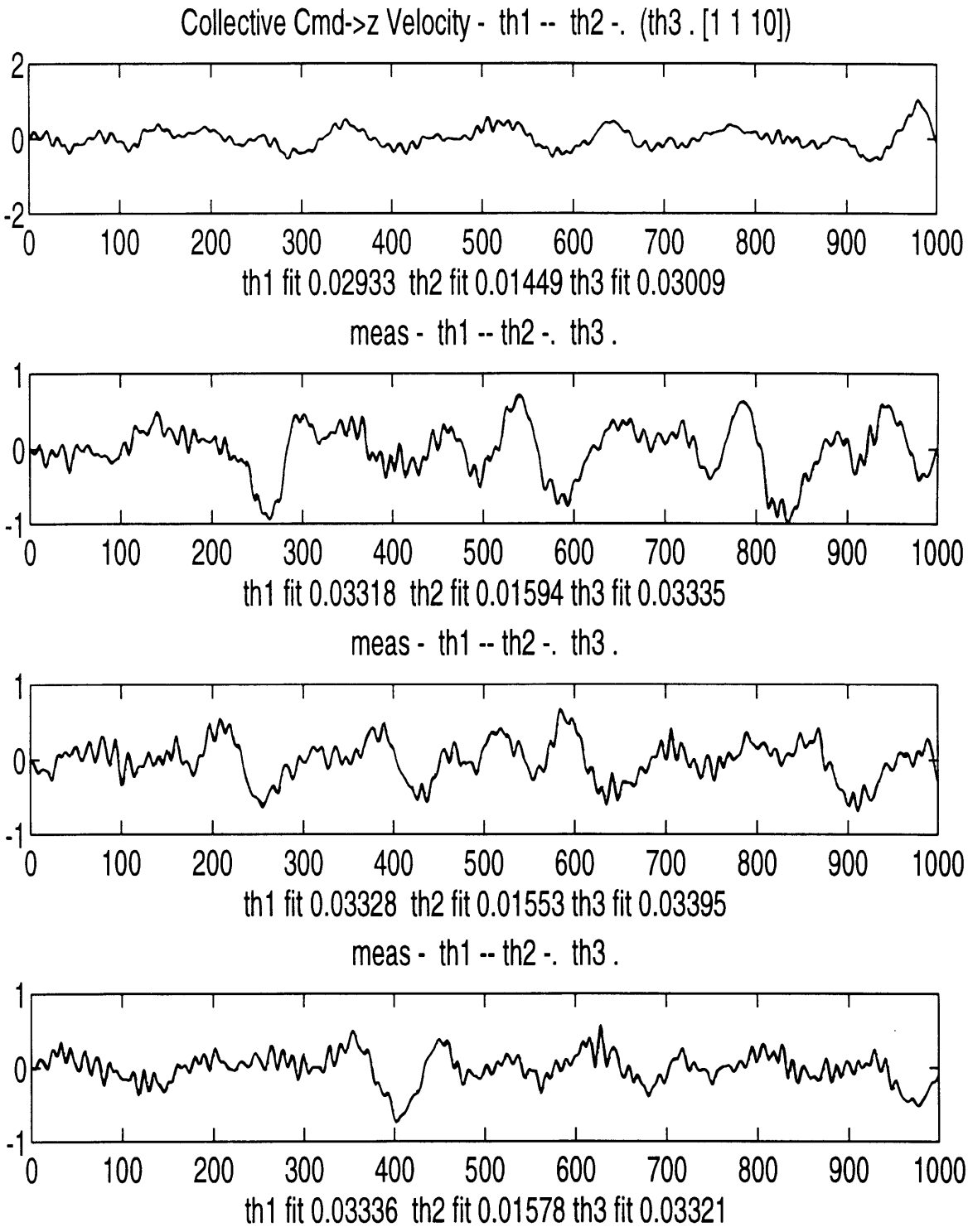


Figure 39
Time Simulation of Collective Command to Z Velocity

A.3 State Space Description

Once the model structure and parameters have been identified, a state space description of the control axis is compiled of the form:

$$x_{k+1} = Ax_k + Bu_k + Gw_k$$
$$y_k = Cx_k + Du_k + v_k \qquad [14]$$

Where x is the state vector, y is the measurement vector, w is the process noise and v is the measurement noise. The script make_r_abcd.m shows how the state space description was generated for the roll control axis. The Matlab™ command polyform pulls the parameter values of the a's and b's of equation [13] out of each of the input-output relationships identified above. Since the y velocity was estimated from the y position by differences of the y position data, the relationship between the y velocity and y position is set as a discrete time forward Euler integration instead of using values identified in the identification process. (Actually, the values generated by the identification process should be very close to those used in the forward Euler integration, and this can be used to check if the identification procedure is being performed properly.)

```
MAKE_R_ABCD.M
% creates the A,B,C,D matrices for the roll model
% uses thrcg and thrvxd
% currently assumes that vision yd is taken at 50 Hz.

%echo on
% Model structures
nnrg = [ 2 2 6];
nnvg = [ 1 1 1];
nnyd = [ 1 1 1];
nny  = [ 1 1 1];


% Parameters of current roll_model
nc = nnrg(3) + nnrg(2) - 1;
ng = max([nnrg(1) nnvg(2) nnvg(3)]);
nr = max([nnvg(1) nnyd(2) nnyd(3)]);
nyd = max([nnyd(1) nny(2) nny(3)]);
ny = nny(1);

state_length = nc + ng + nr + nyd + ny;

gnc = nnrg(3) + nnrg(2);
gng = nnrg(1);

rng = nnvg(3) + nnvg(2);
rnr = nnvg(1);
```

```
ydnr = nnyd(3) + nnyd(2);
ydnyd = nnyd(1);

ynyd = nny(3) + nny(2);
yny = nny(1);

% crank on A,B,C,D

[gg,cg] = polyform(thrcg);

[rr,gr] = polyform(thrgr);

[ydyd,ryd] = polyform(thrvyd);

[yy,ydy] = polyform(thydy)
% perfect integration:
yy = [     1.0000e+00          -1.00 ];
ydy = [ 0              0.0200    ];

A = zeros(state_length);

for i = 2:state_length,
          A(i,i-1) = 1;
end

%gyro equation:

% takes out delayed term
A(nc+1,nc) = 0;

for i = 2:(gnc),
          A(nc+1,i-1) = cg(i);
end

for i = 2:(gng + 1),
          A(nc+1,nc+i-1) = -gg(i);
end

%roll equation:

%takes out delayed term
A(nc+ng+1,nc+ng) = 0;

for i = 2:(rng),
          A(nc+ng+1,nc+i-1) = gr(i);
end

for i = 2:(rnr + 1),
          A(nc+ng+1,nc+ng+i-1) = -rr(i);
end

% for now just use this for gyro -> rollv
%A(10,10) = 1;
%A(10,9) = -7.0109e-05;

%yd equation:

%takes out delayed term
A(nc+ng+nr+1,nc+ng+nr) = 0;

for i = 2:(ydnr),
          A(nc+ng+nr+1,nc+ng+i-1) = ryd(i);
end
```

```
for i = 2:(ydnyd + 1),
        A(nc+ng+nr+1,nc+ng+nr+i-1) = -ydyd(i);
end

%Y equation:

% takes out delayed term
A(nc+ng+nr+nyd+1,nc+ng+nr+nyd) = 0;

% ydy(i-1) is questionable - ydy was only a 1 x 1
if ynyd < 2
 for i = 2:(ynyd),
        A(nc+ng+nr+nyd+1,nc+ng+nr+i-1) = ydy(i-1);
 end
else
 for i = 2:(ynyd),
        A(nc+ng+nr+nyd+1,nc+ng+nr+i-1) = ydy(i);
 end
end

for i = 2:(yny + 1),
        A(nc+ng+nr+nyd+1,nc+ng+nr+nyd+i-1) = -yy(i);
end

B = zeros(state_length,1);
B(1) = 1;

C = zeros(1,state_length);
%gyro version: C(nc+1) = 1;
%Y version:
C(nc + ng + nr + nyd + 1) = 1;

D = [0];
```

The first 6 to 8 states of each of the models indicate delayed command states which correspond to the time delay from when the command is issued until it is seen in a measured state. There are two rate gyro states in the roll and pitch models because the model structure selected to predict the rate gyro state uses two back states. The A matrices for each of the control axes follow. All of the B matrices are a vector of zeros the length of the A matrix except that the first entry is a one. All of the C matrices pull out the states that are measured and each of the D matrices are zeros. The G matrices are discussed in the next section.

A MATRICES

A_roll =

0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0.1270	0.2622	1.7613	-0.8761	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0.9947	0	0
0	0	0	0	0	0	0	0	0	0.2300	0.9947	0
0	0	0	0	0	0	0	0	0		0.02	1

A_pitch =

0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	-0.1625	0.0479	1.8593	-0.8911	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	-0.0000	0.9956	0	0
0	0	0	0	0	0	0	0	0	-0.1869	0.9963	0
0	0	0	0	0	0	0	0	0	0	0.02	1

A_yaw =

0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	-0.2315	1.7960	-0.8601	0
0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	-0.0001	0	1

A_z =

0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0.0013	0.9807	0
0	0	0	0	0	0	0	0	0	0	0.02	1

A.4 Linear Quadratic Controller and Estimator Gain Selection

Since each of the A matrices above are singular, the standard Matlab™ lqr.m and lqe.m commands cannot be used to calculate optimal controller and estimator gains. The LQG gains are computed using the recursive Ricatti equation outlined in Franklin, Powell and Workman p. 427 implemented in the script ric_lqr.m which follows and the estimator gains are calculated similarly using the principle of duality.

```
RIC_LQR.M
% Called by [K1,K, S] = ric_lqr(A,B,Q,R);
% where K is a 1000 x # of states matrix
%
% K1 returns the lqr results
% K can be plotted to check convergence of gains
% format of K is: K = [ ..
%                       ku1 ku2 ku3 .. (ith run)
%                       ku1 ku2 ku3 ... ((i-1)th run)
%                       .. ]
% If the gains do not converge, copy ric_lqr.m over and increase
% the number of iterations N.
% S is the steady state solution to the Ricatti equations
%
% This function Computes "optimal" time-varying feedback gains
% according to Ricatti equations. Q1 and Q2 are weighting matrices
% chosen by the designer but must be nonnegative definite.
% from p. 427 of Dig Cont. Sys - Franklin et al.
% Capable of handling a singular A matrix - handy for delays!

function [K1,K, S] = ric_lqr(A,B,Q1,Q2)

% determine the number of control's (u) in B
u = size(B,2);

states = length(A);

N = 1000;

S = Q1;
Kk = zeros(u,states);
K = zeros(N,u*states);

for i = N:-1:u+1

 M = S - S * B * (inv(Q2 + B'*S*B)) * B' * S;
 Kk = (inv(Q2 + B'*S*B)) * B' * S * A;
 for j = 1:u
  K(i,[1+(j-1)*(states):j*states]) = Kk(j,:);
 end
 S = A' * M * A + Q1;

end

K1 = Kk;
```

Estimator gains L of the discrete Kalman filter

$$\hat{x}_k = \bar{x}_k + L(y_k - C\bar{x}_k)$$
$$\bar{x}_{k+1} = A\hat{x}_k + Bu_k \qquad [15]$$

were selected to minimize the estimation error with respect to the process (w) and measurement (v) noise of the of the system represented in [14]. Although the exact noises are not known, an estimate of the variances (and possibly covariances) as well as the matrix G is necessary to compute the estimator gains. Since the computer generates the commands, it is a fair assumption that there is no process noise in the command or delayed command states. Process noise will be accounted for in the initial rate gyro state but not in any delayed states. Therefore, the G matrix for each of the control axes are

$$G_{roll,pitch} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad G_{yaw} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad G_{coll} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

When estimating the process covariance matrix Q and the noise matrix R to determine the estimator gains, a good starting point is to normalize all of the measurements which is what the xxx_scale factors in the roll_obs.m script are intended to do. Then, the xxx_noise factors can be estimated as a percentage of the full measurement scale of the average magnitude of the noise during a typical hover flight. For example, the full scale of the y measurement is about a meter, but "noise" on the y position is on the order of 1 cm during an average hover flight, so an estimate for y_noise is 0.01 while the y_scale = 1.0 m. To get an estimate of the xx_proc scale factor, let the xxx_proc be a scale on how much trust to place in the identified input-output relationship identified in

section A.1. For example, the relationship between velocity and position is known to be a direct integration, while the relationship between the roll angle and y velocity is more susceptible to other unaccounted for factors like roll-pitch coupling that are not accounted for in the relationship. Therefore, the y_proc factor was selected to be 0.004 and the yd_proc factor indicating the process noise in going from the roll angle to the y velocity was selected to be 1 indicating much less trust in this relationship.

After a set of estimator gains are produced, simulation of the estimator using inputs and measurements from an actual flight is used to determine the feasibility of the gains. If the estimator does not produce estimates close to the measurements, the gains are discarded and the xxx_proc and xxx_noise factors are adjusted accordingly and the process is repeated. If the estimates seem feasible, another check is to integrate the velocity estimate and see how close it comes to the position measurement. Finally, the estimator is implemented in the actual helicopter flight controller and the helicopter is flown under the test pilot's control while the estimator runs off of the test pilot's command and the measurements in real time. If the estimates follow the desired characteristics of the measurements with the desired smoothing, the next step is to calculate the controller gains.

The final estimator gains and estimator Q and R matrices as well as the code used in the roll estimator gain design process is included here.

Table 7
Hover Compensator Estimator Gains L and Covariance Matrices Q and R

	Roll	Pitch	Yaw	Collec-tive
Q*	0.6000 0.0003 1.0000 0.0040	0.1320 0.0001 0.0022 0.0009	60.0000 0.1000	10.000 0.050
R*	0.6000 0.0001 0.2000	0.2100 0.0001 0.0050	750 700	1
L	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.7335 0.0168 -0.0000 0.2791 0.0289 -0.0000 0.0000 0.8282 0.0000 -0.0000 0.0389 1.8262 -0.0000 0.0001 0.2784	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.6998 -0.0155 -0.0000 0.3082 -0.0234 -0.0000 -0.0000 0.5951 -0.0000 -0.0000 -0.0743 0.4694 -0.0000 -0.0005 0.3583	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.4377 -0.0000 0.2979 -0.0000 -0.0000 0.0119	0 0 0 0 0 0 0 0 0 0 2.2706 0.3318

*Only the diagonal elements of the Q and R matrices are presented here.

ROLL_OBS.M

```
% Creates the pitch observer gain matrix according to
% sensor and plant noise parameters.

% load up the model
%load hover_r1

% make correct C and D for creating observer gains
%CC = [ eye(10)              zeros(10,2)
%       zeros(1,11)          1 ];
%DD = zeros(11,1);

% Output sensor rate gyro,vertical gyro, y position
CC = zeros(3,12);
CC(1,8) = 1;
CC(2,10) = 1;
CC(3,12) = 1;

% from these equations, create the 'noise covariance matrices for the system.
% x[n+1] = Ax[n] + Bu[n] + Gw[n]   {State equation}
% z[n]   = Cx[n] + Du[n] + v[n]    {Measurements}
% E(ww') = Q -> process noise ,   E(vv') = R -> measurement noise;

% noise must be scaled - following scales represent max range
% of each sensor

y_scale = 1;
yd_scale = 1;
```

```
r_scale = 0.15;
rd_scale = 300;
comm_scale = 150;

% for now, assume process noise Q, looks like a command disturbance
%G = B;
%Q = 1;

% In this case, assume process noise looks like 1/2 a state vector
G = [   0 0 0 0
        0 0 0 0
        0 0 0 0
        0 0 0 0
        0 0 0 0
        0 0 0 0
        0 0 0 0
        1 0 0 0
        0 0 0 0
        0 1 0 0
        0 0 1 0
        0 0 0 1 ];

% process noises
%rd_proc = 0.1;
rd_proc = 0.002;
r_proc = 0.002;
%yd_proc = 25.0;
yd_proc = 1;
y_proc = 0.004;

% Q is then 4x4
Q = eye(4);
Q(1,1) = rd_proc*rd_scale;
Q(2,2) = r_proc*r_scale;
Q(3,3) = yd_proc*yd_scale;
Q(4,4) = y_proc*y_scale;


% noise must be scaled
% These numbers represent noise magnitudes relative to full scale
% Sensor noise
y_noise =  0.02;
yd_noise =  0.01;
r_noise =  0.0005;
rd_noise =  0.002;
comm_noise =  0.001;

R = eye(3);
R(3,3) = (y_scale*y_noise);
R(2,2) = (r_scale*r_noise);
R(1,1) = (rd_scale*rd_noise);

mu = 1e1;
R = mu*R;


% create Kalman Gain Matrix L
L = ric_lqe(A,G,CC,Q,R)

% NOTE , if two WARNING's of singularity will be displayed,
% you can ignore them since they result from a calculation of the estimate
% error covariance, and do not affect L.

% display Q and R
```

```
for i = 1:length(Q)
        q(i) = Q(i,i);
end

for i = 1:length(R)
        r(i) = R(i,i);
end

%l = [   L(8,8)
%        L(9,9)
%        L(10,10)
%        L(12,11) ];

q = q
r = r'
%l = l'

%sim

L2 = L;
L = zeros(size(A));
L(:,8) = L2(:,1);
L(:,10) = L2(:,2);
L(:,12) = L2(:,3);


SIM.M
% Create estimator
dt = .02;

known = 1;
sensors = [1 2 3];
[Ae,Be,Ce,De] = destim(A,[B G],CC,zeros(3,5),L,sensors,known);

[y1,x1] = dlsim(Ae,Be,Ce,De,[rc1 rg1 vg1 yv1]);
[y2,x2] = dlsim(Ae,Be,Ce,De,[rc2 rg2 vg2 yv2]);
[y3,x3] = dlsim(Ae,Be,Ce,De,[rc3 rg3 vg3 yv3]);
[y4,x4] = dlsim(Ae,Be,Ce,De,[rc4 rg4 vg4 yv4]);

figure(1)
clf
subplot(411)
plot(rc1)
title('roll command')

subplot(412)
plot(rg1)
hold on
plot(y1(:,1),'m')
hold off
axis([0 1000 -450 450])
title('roll rate gyro')

subplot(413)
plot(vg1)
hold on
plot(y1(:,2),'m')
hold off
title('y- vgr m-- roll estimate ')
axis([0 1000 -.2 .2])

subplot(414)
plot(yv1)
hold on
```

```
plot(y1(:,3),'m')
hold off
title('y- y vision m-- y estimate ')
%axis([0 1000 -.2 .2])

figure(2)
clf
subplot(411)
plot(rc2)
title('roll command')

subplot(412)
plot(rg2)
hold on
plot(y2(:,1),'m')
hold off
axis([0 1000 -450 450])
title('roll rate gyro and est')

subplot(413)
plot(vg2)
hold on
plot(y2(:,2),'m')
hold off
title('y- vgr m-- roll estimate')
axis([0 1000 -.2 .2])


subplot(414)
plot(yv2)
hold on
plot(y2(:,3),'m')
hold off
title('y- y vision m-- y estimate')
%axis([0 1000 -.2 .2])


figure(3)
clf
subplot(411)
plot(rc3)
title('roll command')

subplot(412)
plot(rg3)
hold on
plot(y3(:,1),'m')
hold off
title('roll rate gyro and est')
axis([0 1000 -450 450])

subplot(413)
plot(vg3)
hold on
plot(y3(:,2),'m')
hold off
title('y-vgr m- roll estimate ')
axis([0 1000 -.2 .2])

subplot(414)
plot(yv3)
hold on
plot(y3(:,3),'m')
hold off
```

```
title('y- y vision m- y estimate ')
%axis([0 1000 -.2 .2])

figure(4)
clf
subplot(411)
plot(rc4)
title('roll command')

subplot(412)
plot(rg4)
hold on
plot(y4(:,1),'m')
hold off
title('roll rate gyro')
axis([0 1000 -450 450])

subplot(413)
plot(vg4)
hold on
plot(y4(:,2),'m')
hold off
title('y- vgr m-- roll estimate')
axis([0 1000 -.2 .2])

subplot(414)
plot(yv4)
hold on
plot(y4(:,3),'m')
hold off
title('y- y vision m-- y estimate')
%axis([0 1000 -.2 .2])
L2 = L;
L = zeros(size(A));
L(:,8) = L2(:,1);
L(:,10) = L2(:,2);
L(:,12) = L2(:,3);
```

Controller gains K of the control law

$$u_k = -K\hat{x}_k \qquad [16]$$

were generated for the cost function

$$J = \int \left[x^T Q x + u^T R u \right] dt \qquad [17]$$

using a unit penalty on the command (R = 1) and varying the penalty on the state. Since the goal of this experiment is to maintain position, the controller Q matrix is zero everywhere except in the position state. (In the yaw model, the Q is zeros everywhere except in the yaw angle state.) Since only the ratio of Q to R matters in minimization of the cost function, either the Q or the R value may be varied to change the gain values. Several gains were generated and gains that would reach the limiting values of the control input for a nominal disturbance in the state are discarded from further consideration. For example, the roll cyclic range is 400 units. If the gain on the y position state is 2000, a 10 cm offset in y position will saturate the command. The final selection of gains comes from flying estimator and controller gains (the estimator needs to be on to get a position velocity state). Each axis is checked individually by having the test pilot fly the remaining control axes. During the flying phase of gain selection, low controller gains were used initially. These low controller gains stabilize the helicopter as expected because the LQR gains must stabilize the modeled plant. Although these low controller gains stabilize the plant, the helicopter still drifts in position. Increasing the controller gains will eliminate this drift but may excite the higher frequency unmodeled dynamics of the plant. This can be seen when the helicopter begins to shake.

After gains are selected for each controller axis, the roll and pitch controllers are flown together while the test pilot flies the other two control axes. The gains selected when observing the single computer controlled axis may need to be reduced if the helicopter's behavior is not acceptable. For instance, in this setup the roll and pitch controller gains needed to be reduced to eliminate a "wobbling" effect which was the

result of the roll-pitch coupling which is not accounted for in the decoupled controller axis model used here. Once the roll and pitch controller work together, the yaw controller is activated, and finally, the Z State Machine with the z hover controller is activated. Table 8 shows the final values of Q and R and the corresponding controller gains used for each control axis in the hover controller.

Table 8

Hover Compensator Control Gains K and Penalty Values Q and R

	Roll	**Pitch**	**Yaw**	**Collective**
Q*	1e4	1e4	1	1
R*	1	1	1e-4	1e-5
$\mathbf{K^T}$	0.0716	0.0694	0.0219	0.0677
	0.0742	0.0717	0.0219	0.0698
	0.0767	0.0742	0.0220	0.0719
	0.0793	0.0766	0.0219	0.0740
	0.0820	0.0791	0.0218	0.0760
	0.0846	0.0816	0.0216	0.0780
	0.0582	-0.0337	0.0214	0.0800
	0.2291	-0.7252	0.0212	0.0819
	-0.1946	0.6265	-0.0906	0.0837
	629.4341	618.1067	0.0787	0.0856
	97.3054	-109.3495	98.9107	69.1003
	96.4816	-96.5905		136.7137

*Only non-zero value in Q and R is given.

References

Athans, Michael. "Lecture Notes for Multivariable Control Systems I and II." 1993.

Bramwell, A.R.S. Helicopter Dynamics. London: Edward Arnold Publishers Ltd., 1976.

Brown, Robert Grover and Patrick Y.C. Hwang. Introduction to Random Signals and Applied Kalman Filtering, 2nd ed. New York: John Wiley and Sons, Inc., 1992.

Franklin, Gene F., J. David Powell and Michael L. Workman. Digital Control of Dynamic Systems, 2nd ed. Reading, MA: Addison-Wesley Publishing Company, 1990.

Johnson, Wayne. Helicopter Theory. Princeton, NJ: Princeton University Press, 1980.

Ljung, Lennart. System Identification Toolbox for Use with Matlab™. Natick, MA: The Mathworks, Inc., 1992.

Maybeck, Peter S. and Donald L. Pogoda. "Multiple Model Adpative Controller for the STOL F-15 with Sensor/Actuator Failure," Proceedings 28th IEEE Conference on Decision and Control. Dec 1989, pp. 1566-1572.

Prouty, Raymond W. Helicopter Performance, Stability, and Control. Malabar, FL: Robert E. Krieger Publishing Company, 1990.

Söderström, Torsten and Petre Stoica. System Identification. New York: Prentice Hall, 1989.

Spiegel, Murray R. Theory and Problems of Probablity and Statistics. New York: McGraw-Hill, Inc., 1992.

Wright, Anne. "A High-speed, Low-latency Portable Visual Sensing System," 1993 SPIE Proceedings, 1993.