CDS

TECHNICAL MEMORANDUM NO. CIT-CDS 96-009
June, 1996

"The Caltech Helicopter Control Experiment"

Xiaoyun Zhu and Michiel van Nieuwstadt

Control and Dynamical Systems
California Institute of Technology
Pasadena, CA 91125

THE CALTECH HELICOPTER CONTROL EXPERIMENT

XIAOYUN ZHU AND MICHIEL VAN NIEUWSTADT

Division of Engineering and Applied Science
California Institute of Technology
Pasadena, CA 91125

CDS Technical Report 96-009
June 19, 1996

ABSTRACT. This report describes the Caltech helicopter control experiment. The experiment consists of an electric model helicopter interfaced to and controlled by a PC. We describe the hardware and software. A state-space model for the angular position is identified from experimental data near hover, using the prediction error method. An LQR controller with integrators for set point tracking is designed for the system. We also undertake a separate identification and loop shaping control for the yaw dynamics.

1. INTRODUCTION

This report describes the model helicopter experiment at the California Institute of Technology. The purpose of this report is to document our experience with the experiment, to facilitate the use in future projects. The Caltech model helicopter is a testbed for advanced linear and nonlinear control methodologies. Our primary objective is not to get a completely autonomous aerial vehicle, but rather study the control issues involved in this endeavor. We describe the hardware, the software, the identification procedure and a simple controller design to introduce the novice user to the various aspects involved in operating the helicopter. It is important to automate the identification, since every change in parts or setting will necessitate a new identification

Key words and phrases. model helicopter, identification, real-time control, LQR, loop shaping.

Research supported in part by NASA, by NSF Grant CMS-9502224 and AFOSR Grant F49620-95-1-0419 .

session. Once a good model structure is determined, this identification can be performed efficiently, so that replacing parts will not necessarily result in time consuming system ID. With the matlab software developed for this project, this is readily accomplished. We also obtained some experience with control design. We are not trying to push a particular methodology, but would like to convey some design structure that has been useful for us. The controller presented in this report can serve as a first cut for future iterations. The current setup of the experiment evolved from an earlier setup described in [8, 9, 7, 11]. More details about helicopter dynamics can be found in [1, 4].

For the most up-to-date information on the helicopter experiment, consult the web page at URL *http://avalon.caltech.edu/~heli.*

2. Hardware

2.1. **Overview.** The hardware platform of the helicopter experiment is comprised of a Kyosho EP concept 30 electric model helicopter [3], a PWM IO board, a Polhemus inside track sensor [10], a Pentium 100 PC, and SUN workstations. The 30-inch main rotor of the helicopter is powered up by an Astroflight Cobalt 05 helicopter motor. The PWM IO board and the Polhemus Inside Track make up the interface between the physical system and the computer. The real-time control and data acquisition run on the PC, which hosts the PWM IO board and the Polhemus board. The PC is connected to SUN workstations through the PC-NFS network file system. The workstations are employed for off-line computations, such as experimental data analysis, system identification, controller synthesis and simulation. The experimental setup is depicted in Figure 1.

2.2. **PWM IO Board.** The PWM IO board reads PWM input from the pilot and writes PWM output to the servo motors. The PWM signals are TTL (5 Volt) signals with a pulse width of between 1 and 2 ms, and a period of about 20 ms. The length of the pulse encodes the magnitude of the signal.

The pilot gives input through an RC transmitter, whose signals are captured by an RC receiver, that generates PWM signals. The receiver is connected to the PWM board. There are 6 PWM input channels. Of these, 4 regulate the attitude and elevation, 1 regulates throttle and 1 is an on-off switch that the pilot can use as an emergency stop.

The control signals computed by the PC are output to the servos through the PWM board. There are 4 servos on the helicopter: aileron (left-right cyclic), elevator (fore-aft cyclic), rudder (directional) and

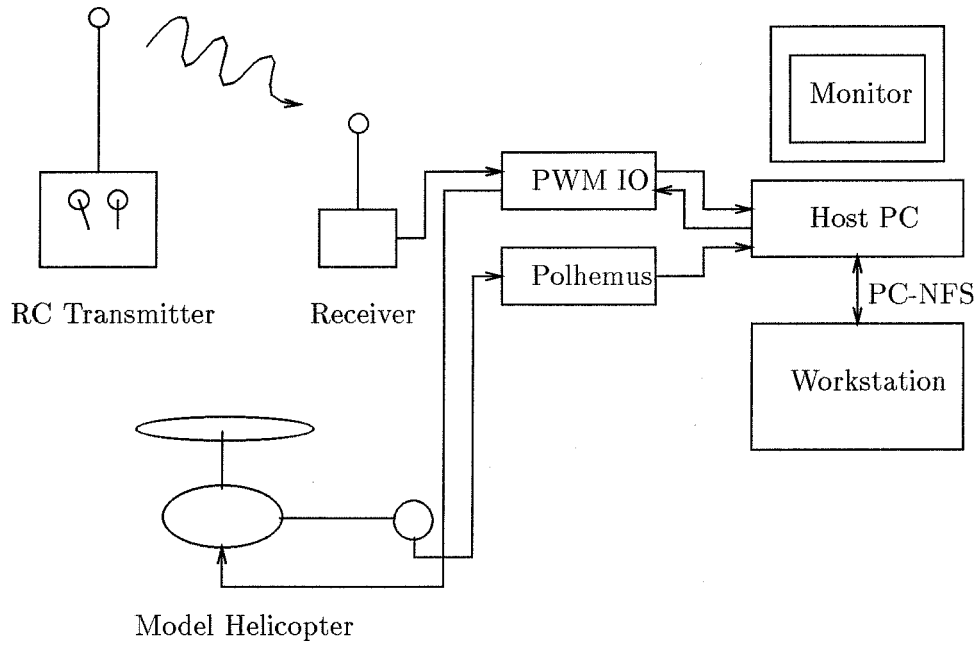

FIGURE 1. Setup of the Helicopter Experiment

collective. The servos are high speed coreless Futaba S9601 servos, which output 2.4 kg.cm of torque, and move at 400 degrees per second.

The thrust on the helicopter is provided by an Astroflight Cobalt 40 electric motor. This motor is driven by a Copley current amplifier that receives its input voltage from the PC. The input voltage is computed from the pilot PWM throttle according to:

$$V_{out} = k_m \times (\text{throttle} - 1.0), \tag{1}$$

where k_m is a constant scaled to cover the full range of the motor. Currently, $k_m = 7$.

The PWM board was custom designed and built by the authors. It is based on 6 Intel 8254 timer chips. Each chip houses 3 independent 16-bit counters. We use an on board 2 MHz crystal to generate a 50 Hz base signal for the PWM output with one of the counters. This on board timing makes the board immune to changes in ISA bus speed. Of the remaining 17 counters, 8 are configured as PWM input and 8 as PWM output, while 1 is not used.

The output counters are run in mode 1, the input counters in mode 2. The PWM input signal is connected to the gate of the counter. To get the right pulse width reading, we must make sure that the gate is low during a read. We check this by 2 subsequent readings of the

counter value. If the counter value changes, the gate is still high and we return the previous count. If the counter value is the same between the 2 readings, we assume the gate is low, the input pulse has ended, and therefore the reading is valid. This of course assumes that a clock pulse has occurred between 2 readings, which means that for a 2 MHz crystal, a read has to take at least 500 ns. This is no problem with the Pentium 100. We also need to make sure that we don't try to read the counter every time when the gate is high. Since the input PWM signal has a frequency of about 50 Hz, we can achieve this by sampling faster than 50 Hz. In practice we can sample at 50 Hz since the phase of the PWM signal changes over time. The schematic of the PWM board is shown is Figure 2.

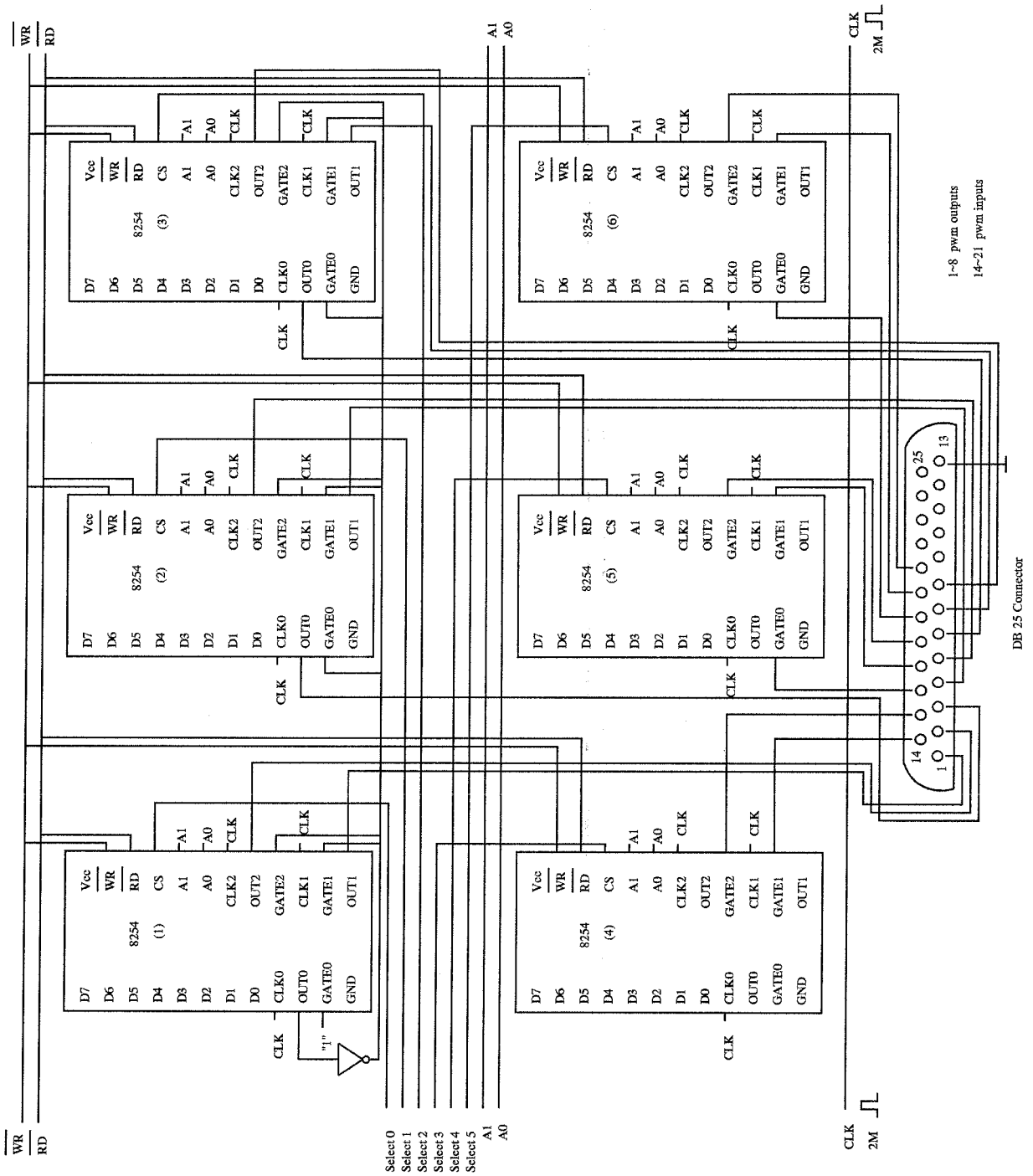

FIGURE 2. Schematic of the PWM I/O Board. All pins A0, A1 and CLK are connected in this schematic.

2.3. **Polhemus Inside Track.** The Polhemus Inside Track is an inexpensive magnetic pulsed DC sensor that provides full 6 degree-of-freedom position and attitude information. The sensor consists of a ground based transmitter and a receiver mounted on the helicopter. Both are connected to a C40 based ISA slot processing board. The board computes (x, y, z) position and azimuth, elevation and roll (ZYX Euler angles) based on the phase and magnitude difference between transmitted and received pulse. The sensor data is corrupted by large metal objects, and RF noise in the vicinity of the receiver, since this affects the magnetic field. The magnetic pulses are sent at a 10 MHz frequency, so that it is especially sensitive to switched power supplies and amplifiers that typically operate at that frequency. PCs, electric motors and speed controllers are serious sources of RF noise. We mounted the sensor on the landing frame under the tail boom, as far away as possible from the motor, to mitigate noise perturbations. The distance between sensor and center of mass increases the vibrational noise on the sensor, but this is high frequency and can be filtered out. We use a floor-mounted power amplifier instead of a helicopter-mounted speed controller to eliminate noise from the latter. In its standard configuration the sensor has a range of 6 feet. Polhemus sells an amplification module that increases this range to 18 feet. Accuracy is linear in the distance between transmitter and sensor, and is 0.0003 mm per mm range, and 0.03 degrees. At a distance of 1 meter, this results in an accuracy of 0.3 mm. We run the Polhemus board in the software trigger mode, i.e. a measurement is made only when requested by the real-time code. The Polhemus board takes about 20 ms to convert a measurement into angles and position. It sets a data ready flag when a full record of position data (3 positions and 3 angles) is produced. The device driver checks for this flag and reads the new record if it is available. It keeps the old position data if the new record is not yet available. This allows the real-time code to be run at sampling rates higher than 50 Hz for faster reading and writing of other devices. The position data is still essentially updated at 50 Hz. See [10] for more details.

2.4. **Rigid body transformations.** The Polhemus board measures ZYX Euler angles $(\psi_1, \theta_1, \phi_1)$. The relation between the spatial (subscript s) and body (subscript b) representation of a vector v then becomes:

$$\begin{aligned} v_s =& R_z(\psi_1)R_y(\theta_1)R_x(\phi_1) \\ =& \begin{pmatrix} \cos\psi_1 & -\sin\psi_1 & 0 \\ \sin\psi_1 & \cos\psi_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_1 & 0 & \sin\theta_1 \\ 0 & 1 & 0 \\ -\sin\theta_1 & 0 & \cos\theta_1 \end{pmatrix} \\ & \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_1 & -\sin\phi_1 \\ 0 & \sin\phi_1 & \cos\phi_1 \end{pmatrix} v_b \end{aligned} \tag{2}$$

For some applications it is more convenient to have Euler XYZ angles available. For example, the main rotor thrust and tail rotor torque are aligned with the body z-axis. If we use XYZ Euler angles to transform the body z-axis to its spatial representation, we eliminate the Z-angle ψ_1 from the expression. If $(\phi_2, \theta_2, \psi_2)$ are Euler XYZ angles, the relation between a vector in spatial and body coordinates is:

$$v_s = R_x(\phi_2)R_y(\theta_2)R_z(\psi_2)v_b \tag{3}$$

If we measure ZYX angles $(\psi_1, \theta_1, \phi_1)$ we can find the corresponding XYZ angles $(\phi_2, \theta_2, \psi_2)$ by calculating $R = R_z(\psi_1)R_y(\theta_1)R_x(\phi_1)$ and:

$$\begin{aligned} \theta_2 &= \arcsin(r_{13}) \\ \psi_2 &= -\arcsin\frac{r_{12}}{\cos\theta_2} \\ \phi_2 &= -\arcsin\frac{r_{23}}{\cos\theta_2} \end{aligned} \tag{4}$$

where r_{ij} is the entry in the i-th row and j-th column of R. Conversely, if we measure XYZ angles $(\phi_2, \theta_2, \psi_2)$ we can find the corresponding ZYX angles $(\psi_1, \theta_1, \phi_1)$ by calculating $R = R_x(\phi_2)R_y(\theta_2)R_z(\psi_2)$ and:

$$\begin{aligned} \theta_1 &= -\arcsin(r_{31}) \\ \phi_1 &= \arcsin\frac{r_{32}}{\cos\theta_1} \\ \psi_1 &= \arcsin\frac{r_{21}}{\cos\theta_1} \end{aligned} \tag{5}$$

2.5. **Helicopter Maintenance.** It is crucial to keep the helicopter in good condition. Check nuts and bolts thoroughly, make sure all linkages are properly attached. A small defect may result in a large crash.

Tracking the blades is of vital importance. Due to asymmetry in the blades, they will not automatically rotate in the same plane. We therefore need to adjust the pitch angle to establish this. Asymmetric blades will result in an imbalance that cannot be compensated for by the aerodynamic controls. To track the blades, mark one blade with a piece of colored tape and turn up the throttle until the helicopter is just about to take off, but still stays on the ground. Observe which blade is higher. Adjust the push rod of one of the blades. turning the ball joint in (clockwise) will increase the blade pitch, turning it out (counter clockwise) will decrease the blade pitch. Repeat the procedure until good tracking is achieved. The collective blade pitch right before take off should be 10 degrees.

When the tail boom is damaged to the point where it touches the timing belt, a replacement is needed. Make sure that the timing belt which connects the front drive sprocket and the tail drive sprocket should be twisted in a proper way such that the when the main rotor rotates counter clockwise, the tail rotor rotates correspondingly. The main rotor rotates clockwise in normal operation, and the tail rotor rotates up in the down wash of the main rotor.

Due to the long push rod connecting the rudder servo to the tail rotor, the rudder actuation is prone to high friction, stiction, and back lash. Make sure the servo horn and the tail rotor bracket are mounted perpendicular to the push rod to minimize asymmetry, and keep the sleeve around the push rod well lubricated.

2.6. **Stand.** All experiments described in this report are done with the helicopter mounted on a stand. This is mainly for safety reasons, since we fly the helicopter in the lab. The stand is 6 degrees of freedom, but we clamp the 3 lateral motions so that we only have yaw, pitch and roll motion. In later experiments we plan to use the full 6 degree of freedom motion. The stand is made of aluminum and adds a payload of 0.5 kg to the helicopter, when full 6 dof motion is allowed. The previous version of this experiment, described in [8] had a 3 degree of freedom wrist attached to the stand, to allow yaw pitch and roll motion and measurement with encoders. This wrist added some 2.5 kg to the payload of the helicopter, which could not take off. The load was reduced with springs at the elbow joint of the stand, which resulted in strongly nonlinear heave dynamics, as the payload increased with increasing altitude.

3. Software

Data processing and controller design are done on SUN workstations with the Matlab Toolboxes, such as System ID Toolbox, Signal Processing Toolbox, and Control System Toolbox. The real-time software runs on the PC and interfaces between the data in Matlab and the experimental data captured by the IO boards. The tasks the real-time software accomplishes include device initialization, data acquisition, real-time computation, dynamic displaying, and online user interaction. The real-time kernel of the software is *Sparrow*, which was developed at Caltech by Richard Murray and his group. Documentation on Sparrow can be found on the web at URL *http://avalon.caltech.edu/~murray/sparrow*. Two custom device drivers have been written for the PWM IO board and Polhemus Inside Track. All sensor input and actuator output can be displayed dynamically on the PC monitor. The display also allows interactive modification of control variables, like controller selection and flight mode.

4. Linear Identification

We need a linear time-invariant (LTI) model to capture the main dynamical features of the model helicopter around hover. Since it is hard to maintain the helicopter at hover when it flies freely, we mount it on a 3 degree-of-freedom (DOF) wrist on a stand initially. Thus the helicopter cannot move in the lateral directions but only in the rotational directions, i.e. in roll (ϕ), pitch (θ) and yaw (ψ) axes. There is significant coupling between yaw and heave, so this does not give good models for free flight, but it serves as a start. Hence the model we want to identify is three-input three-output, with the three inputs being aileron (ξ), elevator (η), and rudder (ζ) (See Figure 3).

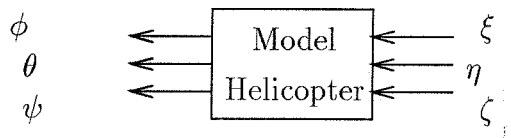

Figure 3. Linear MIMO Helicopter Model

The data was gathered as follows. After the motor was powered up, a human pilot manipulated the transmitter to provide a trim signal to keep the helicopter flying near hover, on top of which a sequence of small square pulses for aileron, elevator and rudder respectively were

superimposed. The small excitations were given by the real-time software. Their amplitudes were properly chosen so that the helicopter could be kept stable around hover. Thus the slowly time-varying trim signals plus a set of small signal excitations composed the inputs to the system. The angular positions were measured by the Polhemus sensor. The data was captured by the real-time software and dumped into a data file on the PCs hard disk. A couple of different data sets were taken to check consistency.

The data analysis and system identification was done with Matlab. First we filtered both input and output data through a band pass filter to reduce the effect of the low frequency trim signal and the high frequency noise. The plot of all three angles together shows that they are not decoupled. Thus a state-space MIMO structure is chosen and the Prediction Error Method (PEM) in state-space is employed for identification, [5, 6]. The discrete-time LTI model we use is simply given by:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{6}$$

where $x = [\phi,\ \theta,\ \psi,\ p,\ q,\ r]^T$, with p, q, r being the roll, pitch and yaw rates respectively, $u = [\xi,\ \eta,\ \zeta]^T$ (incremental aileron, elevator and rudder respectively) is the control input, and y is the measured angular position.

The structure we use for A, B, C is as follows.

$$A = \begin{bmatrix} 1 & 0 & 0 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 & T_s & 0 \\ 0 & 0 & 1 & 0 & 0 & T_s \\ a_{41} & a_{42} & 0 & a_{44} & 0 & a_{46} \\ 0 & a_{52} & 0 & 0 & a_{55} & a_{56} \\ 0 & 0 & a_{63} & a_{64} & a_{65} & a_{66} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix} \tag{7}$$

$$C = [I_3\ 0_3],$$

where $T_s = 0.02$ seconds is the sampling period. The structures of B and C matrices are fixed. The above structure for the A matrix was arrived at by trial and error. Identification of these coefficients was done on different data sets and checked by two criteria. First, a

coefficient which is highly repeatable on all the data sets is determined to be good. Second, a coefficient with a high percent standard deviation (above 20%) is considered to be bad and eliminated from the structure.

Model validation was done on several other data sets to check the prediction capability of this model. The prediction is fairly good. Figures 4, and 5 show simulations of the model on the identification data and a validation set respectively. The simulation is plotted with a dashed line while the experimental data is displayed with a solid line.

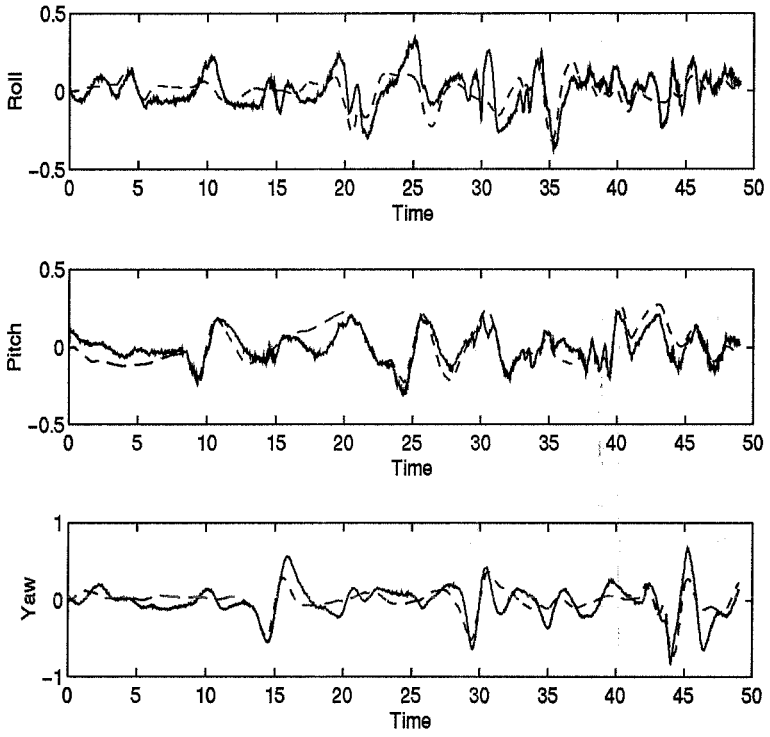

FIGURE 4. Full 3dof identification. Experimental data is solid, identified model is dashed.

4.1. **Identification of yaw dynamics.** It is hard to obtain meaningful data on the yaw dynamics for various reasons. Since the tail rotor is coupled to the main rotor, a variation in disk load will change the tail rotor speed and therefore the torque exerted by the tail rotor. Changes in roll and pitch therefore affect yaw in a nonlinear fashion: equal but opposite changes in roll will have the same effect on yaw, and similarly for pitch. Due to the length of the push rod from the rudder servo to the tail rotor, there is significant friction in the rudder actuation. Due

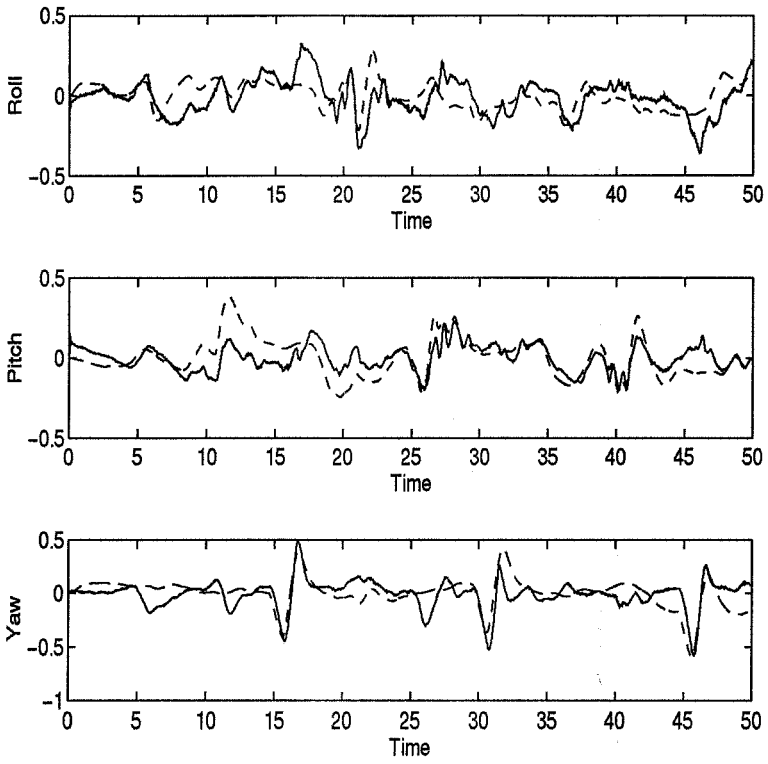

FIGURE 5. Full 3dof validation. Experimental data is solid, identified model is dashed.

to the effect of the main rotor, it is much easier to spin in the counter clockwise direction than in the clockwise direction, so rudder actuation is strongly nonlinear. Since the dynamics are invariant under yaw, it is hard to keep the yaw angle close to zero. Small disturbances will cause a drift in yaw that is not automatically compensated for, and is not easily observable to the pilot. It is therefore useful to do an identification from rudder to yaw only, to gain better insight in the yaw dynamics. We took data with rudder excitation only, with the pilot trying to keep roll and pitch equal to zero. The excitation consisted of a series of doublets with increasing pulse width. The model structure used for identification of the yaw dynamics is a subset of (7):

$$
\begin{aligned}
A &= \begin{bmatrix} 1 & T_s \\ a_{63} & a_{66} \end{bmatrix} \\
B &= \begin{bmatrix} 0 \\ b_3 \end{bmatrix} \\
C &= [1\ 0].
\end{aligned}
\tag{8}
$$

The matlab PEM routine run on the above model structure with $T_s = 0.02$ resulted in $a_{63} = -0.1376$, $a_{66} = 0.8947$, $b_3 = -2.0269$. Figure 6 shows the data range used for identification, and Figure 7 the range used for validation. Note that the prediction on the validation data is quite good.

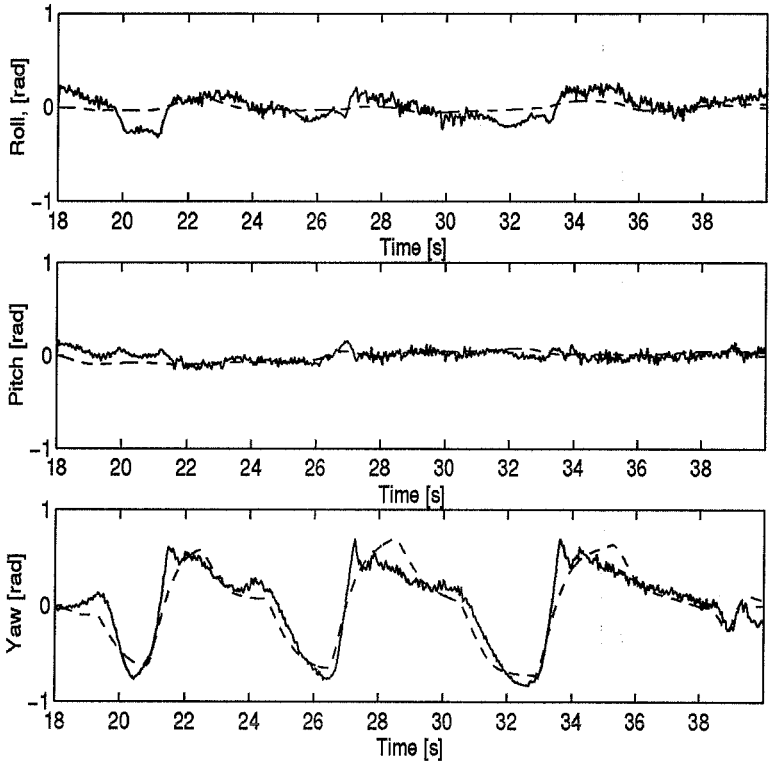

FIGURE 6. Yaw identification. Experimental data is solid, identified model is dashed.

The identified model for the yaw dynamics turns out to be significantly different than the one obtained by looking at the subsystem for yaw in (7), thus justifying the separate identification of yaw. We splice the separately identified yaw model into the total model.

4.2. **Summary.** To summarize our identification procedure, we list the steps below.

1. Take data with ample roll and pitch excitations.
2. Use PEM to estimate a 3 degree of freedom model of the form (7).
3. Take data with ample yaw excitations.

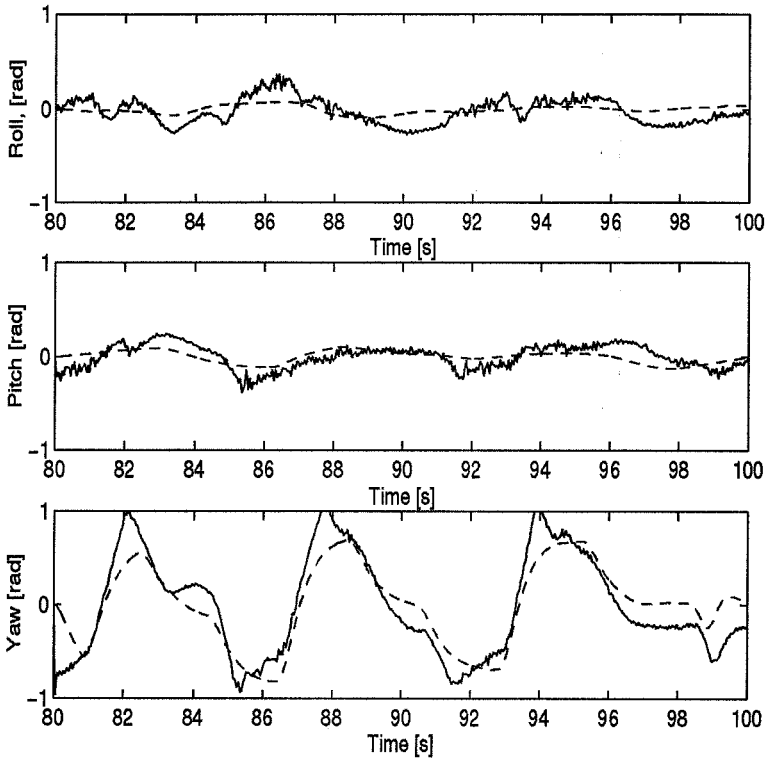

FIGURE 7. Yaw validation. Experimental data is solid, identified model is dashed.

4. Use PEM to estimate a 1 degree of freedom model from rudder to yaw of the form (8).
5. Splice the yaw model into the 3 dof model.

We do not suggest that this is the only legitimate way to obtain a model, but it worked for us and might serve as a first cut for future experiments.

5. LQR CONTROLLER SYNTHESIS

Our objective is to design an LQR controller to stabilize the helicopter at hover. The structure of the closed-loop system is shown in Figure 8. The pilot command enters at the output of the helicopter for set point tracking. The integrators are added to ensure zero steady-state tracking error. The actuator model is simply a saturation with a maximum absolute value of 0.5 ms.

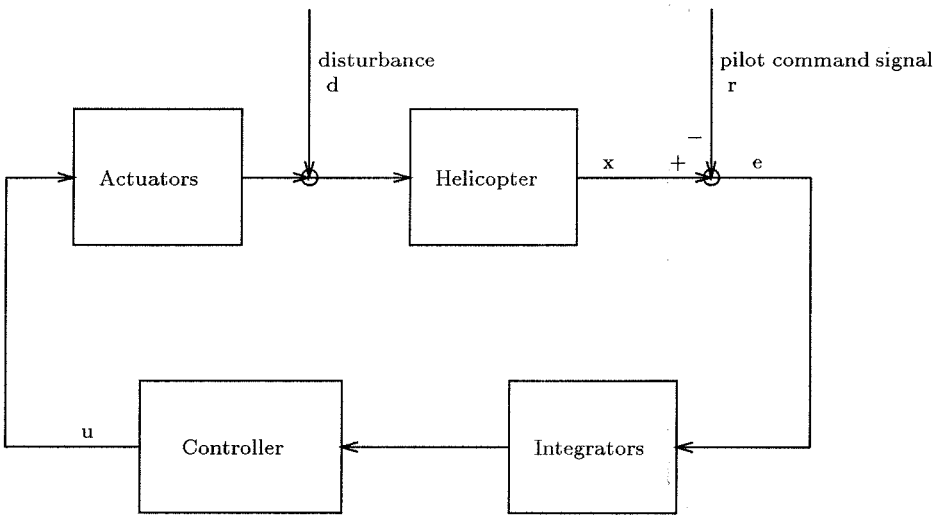

FIGURE 8. Closed-loop System with LQR Controller

The performance measure we try to optimize is the standard quadratic function:

$$J = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} x(t)^T Q x(t) + u(t)^T R u(t), \tag{9}$$

where x is the augmented state vector including three angular positions, three angular rates and three integrator states, and u is the control effort. We do the controller design with two different schemes. One is full information feedback in which the C matrix is a 6×6 identity matrix such that we have both angular positions and velocities feedback. The velocities are computed real time by a digital filter on the angles. A controller K is designed by the matlab routine `dlqr` [2]. The step response of the closed-loop system is simulated with `simulink`. We choose the Q and R matrices in the LQR design to be diagonal and tune the coefficients to get satisfactory (overdamped) step responses on roll, pitch and yaw as well as zero steady-state error and no control saturation. The other scheme is output feedback in which we do the same `dlqr` routine but with a different C matrix only measuring the angles. Then the state estimator L is designed with the `dlqe` routine. Finally the routine `dreg` is used to integrate the LQR controller K and the estimator L into a single LQR controller. For both designs we need to wrap the integrators back into the controller. The entire design is done in discrete time so we eliminate inaccuracies resulting from conversion to continuous time and back. We tested the disturbance rejection of the controller by tapping on the landing gear in different

directions. The results are presented in Figure 9. The times at which the disturbance was applied are indicated by vertical lines. Disturbance rejection for roll and pitch is quite good, but the disturbance rejection for yaw is very poor. The controller does not compensate for the large excursions yaw, as can be seen in the figure. Note the high frequency noise in roll and pitch due to the vibration caused by the main rotor. The sensor is mounted far away from the main shaft, to minimize the RF interference form the motor. Therefore the vibrations at the sensor location are much higher than at the main helicopter body.

The step responses of roll, pitch and yaw angles in simulation for the full information design are shown in Figure 10. Again, roll and pitch are much better than yaw.

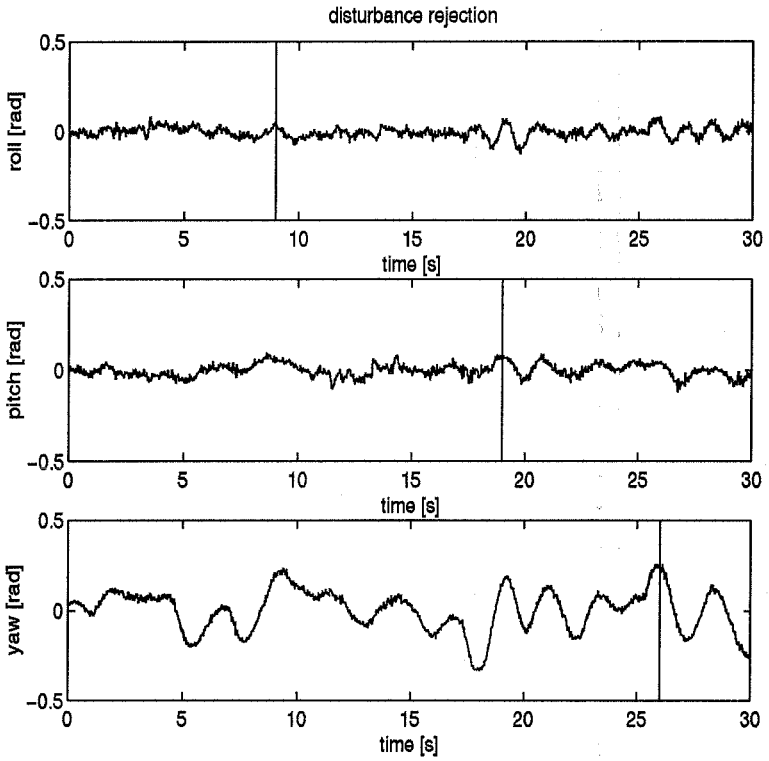

FIGURE 9. Disturbance rejection with LQR controller. Disturbances indicated with vertical lines.

5.1. **Loop shaping design for yaw dynamics.** The LQR design for the yaw dynamics resulted in poor disturbance rejection and step response for yaw. After increasing the gains on yaw, yaw rate, and integrated yaw as much as possible without causing instability, we still

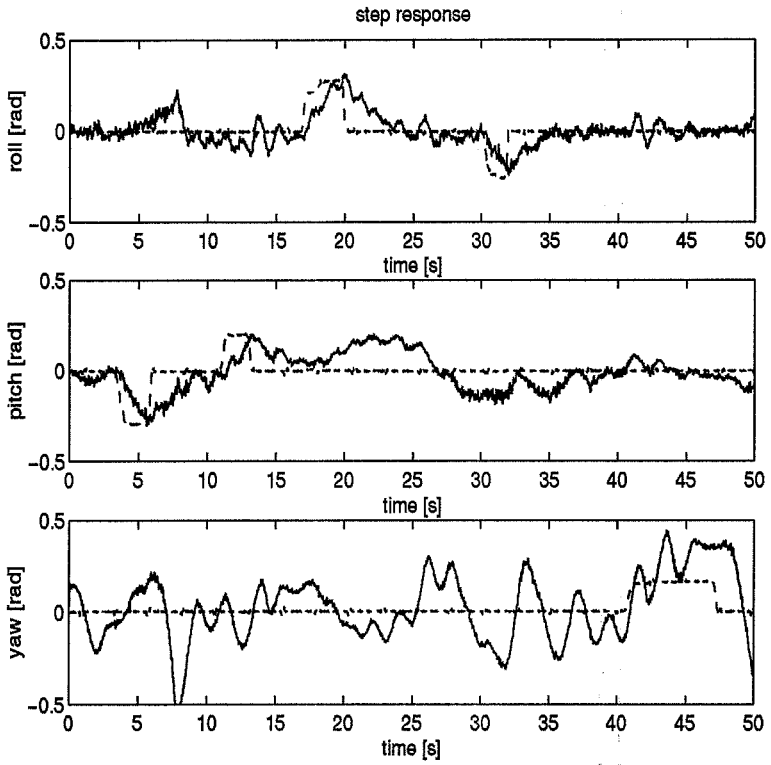

FIGURE 10. Step responses with LQR controller. Reference signal is dashed, measured response is solid.

could not obtain good yaw performance. We therefore did a separate loop shaping design for the SISO system from rudder to yaw angle that was identified in subsection 4.1. This model is first converted to continuous time, since loop shaping is easier in continuous time. The loop shaping controller has the form

$$K(s) = k\frac{\Pi(s+z_i)}{\Pi(s+p_i)} \tag{10}$$

with gain $k = 0.04$, zeros $z = (-0.5, -2.5, -2.9)$, and poles $p = (0, -10, -20, -20)$. The zeros at $(-2.5, -2.9)$ cancel the stable poles of the rudder to yaw transfer function, the pole at 0 achieves zero steady state error, and the remaining zero and poles are used to obtain sufficient gain and phase margin for the open loop. The controller was converted to discrete time and was tested with the roll and pitch axes clamped, and then spliced into the LQR controller for pitch and roll.

The resulting disturbance rejection is shown in Figure 11, and the step response in Figure 12. Pitch and roll show slight improvement over the full state LQR controller, and the improvement in yaw response and

disturbance rejection is dramatic. Compare with Figures 9, and 10 of the full LQR controller.

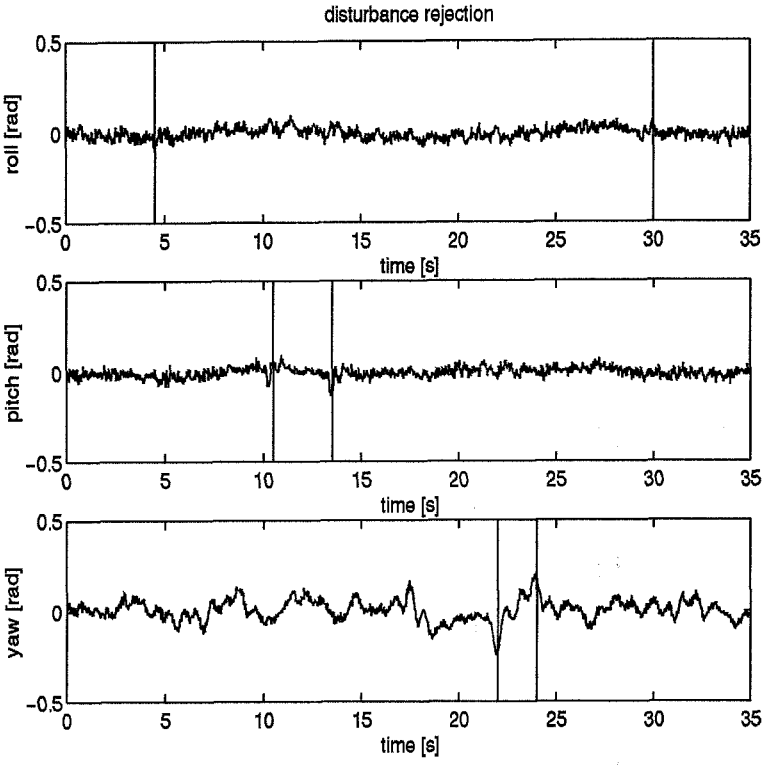

FIGURE 11. Disturbance rejection with LQR/LS controller. Disturbances indicated with vertical lines.

6. CONCLUSION

We described the hardware and software of the Caltech helicopter experiment. We summarized our experience with the maintenance. We described the identification procedure and designed a controller stabilizing hover. LQR control of pitch and roll was successful. Control of the yaw angle proved problematic due to asymmetry in the actuation and high friction and stiction in the rudder actuation assembly. We performed a separate identification of the SISO system from rudder to yaw, and designed a loop shaping controller for this system. Combining this controller with the LQR controller for pitch and roll resulted in a dramatic improvement in yaw response. We do not suggest that these are the only control methodologies that would work, but they can serve as an initial cut to get things to work.

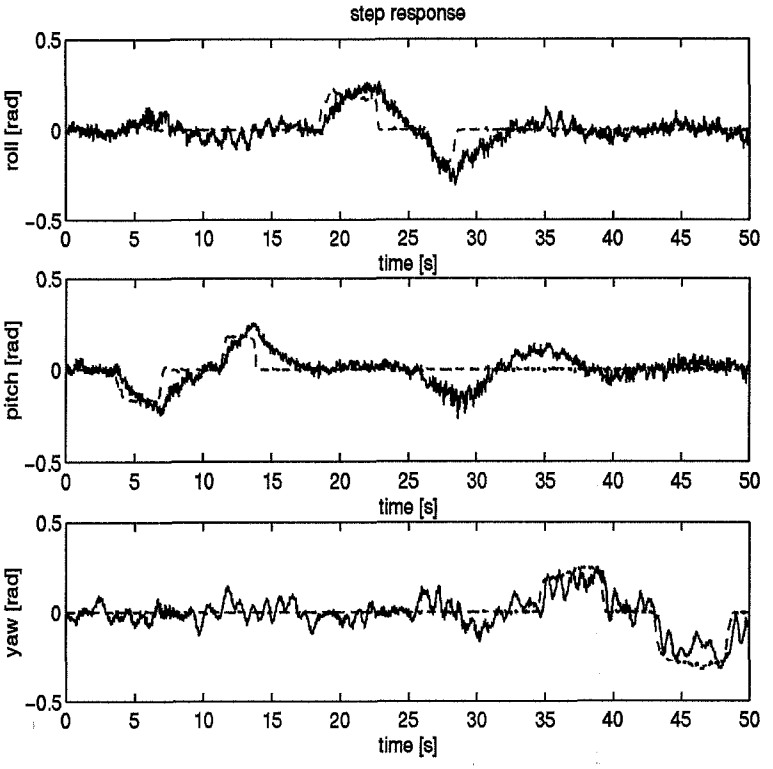

FIGURE 12. Step responses with LQR/LS controller. Reference signal is dashed, measured response is solid.

REFERENCES

[1] G. Myers A. Gessow. *Aerodynamics of the Helicopter.* Frederick Ungar Publishing Company, 1952.

[2] A. Grace, A.J. Laub, J.N. Little, and C.M. Thompson. *Control Systems Toolbox,* July 1992.

[3] R. Hostetler. *Ray's Complete Helicopter Manual.* R/C Modeler Corporation, 1990. 2nd Edition.

[4] W. Johnson. *Helicopter Theory.* Princeton University Press, 1980.

[5] L. Ljung. *System Identification, Theory for the User.* Prentice Hall, 1987.

[6] L. Ljung. *System Identification Toolbox.* MathWorks, 1991.

[7] J. Morris and P. Bendotti. Robust hover control for a model helicopter. In *Proceedings of the American Control Conference,* 1995.

[8] J. Morris and M. van Nieuwstadt. Experimental platform for real-time control. In *Proceedings of the ASEE Annual Conference,* 1994.

[9] J. Morris, M. van Nieuwstadt, and P. Bendotti. Identification and control of a model helicopter in hover. In *Proceedings of the American Control Conference,* 1994.

[10] Polhemus Incorporated. *Polhemus 3SPACE User's Manual,* December 1993.

[11] M. van Nieuwstadt and J.Morris. Control of rotor speed for a model helicopter: a design cycle. In *Proceedings of the American Control Conference*, 1995.