AIAA Guidance, Navigation, and Control Conference and Exhibit
14-17 August 2000 Denver, CO

A00-37046

AIAA-2000-4057

HIERARCHICAL CONTROL SYSTEM SYNTHESIS FOR ROTORCRAFT-BASED UNMANNED AERIAL VEHICLES

David Hyunchul Shim*, Hyoun Jin Kim*, Shankar Sastry**

ABSTRACT

This paper introduces the development of multiple number of Unmanned Arial Vehicle (UAV) system as a part of *BErkeley AeRobot* (BEAR) project, highlighting the recent achievements in the design and implementation of rotorcraft-based UAV (RUAV) control system. Based on the experimental flight data, linear system model valid near hover condition is found by applying time-domain numerical methods to experimental flight data. The acquired linear model is used to design feedback controller consisting of inner-loop attitude feedback control, mid-loop velocity feedback control and the outer-loop position control. The proposed vehicle-level controller is implemented and tested in Berkeley UAV, *Ursa Magna 2*, and shows superior hovering performance. The vehicle level controller is integrated with higher-level control using a script language framework to command UAV.

1. Introduction

The rapid development of the sensor, communication and control technology in the last few decades has made autonomous vehicles smaller and more powerful. And the need to alleviate the actual human participation in order to save human efforts and avoid hazards has been ever increasing in numerous fields. As a result, autonomous vehicles are about to become part of reality in many applications.

BErkeley AeRobot project aims to organize multiple number of autonomous agents into integrated and intelligent systems with reduced cognition and control complexity, fault-tolerance, adaptivity to changes in task and environment, modularity and scalability to perform complex missions efficiently. Figure 1 illustrates one scenario in which a fleet of low altitude Unmanned Arial Vehicle fly over a suspected area looking for hiding ground-based enemies, possibly Unmanned Ground Vehicles (UGVs). UAV at higher altitude coordinates UAVs at lower altitude and reports the current status to the remote base via wireless communication media. The remote base receives the mission states and issues the following-up commands.

To realize this scenario, we need a number of cooperating UAVs that are able to navigate the area following waypoints commanded by the higher-level guidance system. They should be equipped with various sensors, which provide estimates of state variables and detect objects of interest, and controllers which generate the actuator signals required to follow the reference attitude and trajectory properly. And the guidance system needs to be able to process reference trajectory suitable for the mission. As for the ground-based agents, similar automated vehicles with navigation capability are required and these are available commercially.

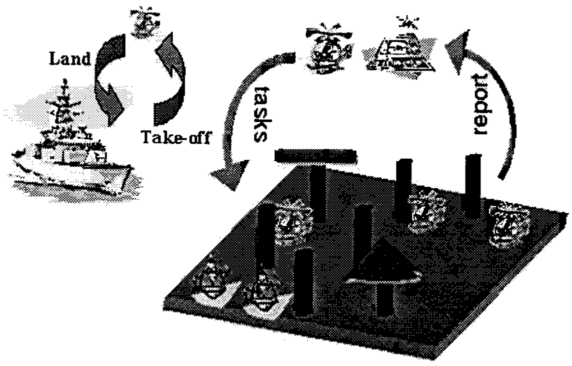


Figure 1. Mission coordination of multiple agents

The rotorcraft suits our scenarios particularly well due to its versatile maneuver-ability such as vertical take-off/landing, hovering, sideslip, pirouette and so on. The rotorcraft based UAV can be operated in relatively smaller space by vertical take-off and landing. It is also able to hover over or track a target agent below at matching low speed.

*Graduate Students, Department of Mechanical Engineering, University of California at Berkeley
**Professor, Department of Electrical Engineering and Computer Science, University of California at Berkeley
{hcshim,jin,sastry}@robotics.eecs.berkeley.edu

Compared with the mission-level guidance tasks, the actual UAV navigation and control problem is not fully investigated and the diverse research activities are ongoing[2,3,4,5,6]. One of the daunting tasks in constructing a UAV system is to obtain high fidelity model to design flight controllers upon. The traditional approach of borrowing the model developed for full-size helicopters often fails to account for the unique dynamics of RUAVs, which are commonly equipped with the servorotor system. Recently, Mettler[2] suggested a very effective model to account for the dynamics of servorotor. In this research, this model is adopted for the identification of the hovering and instead of frequency-domain identification, time-domain method is applied. The obtained model is used for multi-loop classical hovering control design and full-model linear robust control synthesis.

Vehicle Control Language(VCL) is also proposed in this paper as an application and development tool for RUAV systems. This script language system, operating in a hierarchical client-server environment, integrates the vehicle-level control and the higher-level motion command.

The organization of the paper is as follows. Hierarchical architecture and configuration of Berkeley UAV test bed system are explained in Sec. II. In Sec. III, we present the dynamic model identified from Berkeley RUAVs, highlighting the effect of servorotor. In Sec. IV, we discuss how we design stabilizing control law and present the experiment results. In Sec V, the novel approach of VCL is introduced and examples are given.

2. Berkeley UAV System Architecture

Berkeley UAV research test bed is established in the context of multiple number of UAV and/or UGV agents operations. For base airframe, BEAR has chosen four different sizes of model helicopters. Two Kyosho Concept helicopters were built and equipped with navigation and flight control systems. These 60 class helicopters are powered by 0.90 cubic-inch two-cycle glow-plug engine, which can carry up to 5 kg as payload. At the other end of the UAV size spectrum, two Yamaha agricultural helicopters, Yamaha RMAXs, are adopted for their sufficient payload (>30kg) and reliable, consistent performance. Yamaha R-50, the predecessor of RMAX, is slightly smaller version and is powered by single cylinder water-cooled two-stroke gasoline engine and can carry up to 20 kg. As the intermediate size, Bergen Industrial Twin helicopters are chosen. They are powered by customized two cylinder four stroke gasoline engine and offers 10 kg payload, which is ideal for carrying the basic navigation/ control package and some other extra equipment such as vision processing board, camera system and so on.

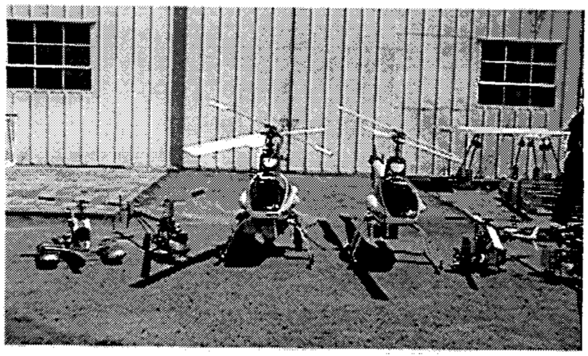


Figure 2. UAV fleet at UC Berkeley

Kyosho Concept 60 Graphite has been used as the valuable testbed for prototyping the flight system design[6]. Classical SISO position/velocity/attitude controller has been designed and tested successfully. In this paper, similar approach is applied to Yamaha R-50 helicopter.

A UAV is a vehicle integrated with mechanical and electronic components such as airframe, navigation sensors, computers, batteries and other sensors, performing autonomous tasks desirably with minimal intervention by remote human operator. The onboard components can be categorized into the following: 1) flight control computer (FCC), 2) navigational sensors, 3) communication module, and 4) onboard power pack.

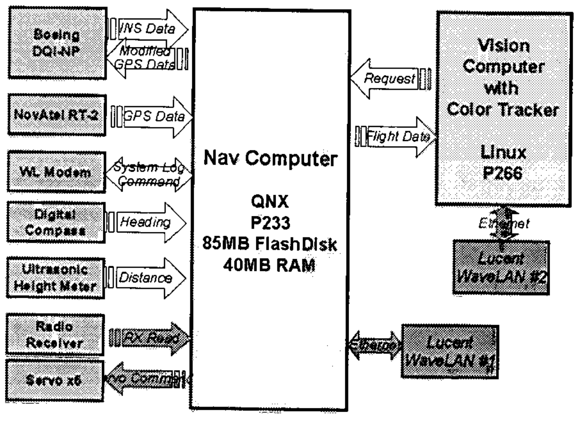


Figure 3. Onboard flight control system structure

Table 1. Berkeley UAV physical specification

Name	Length	Height	Weight	Engine	Autonomy
Kyosho Concept 60	1.4 m	0.47m	9.3kg (4.5kg+4.8kg avionics)	OS FX91 2.8bhp	Boeing DQI NovAtel RT-2 MediaGX233
Bergen Industrial Twin	1.8 m	0.6m	7kg dry weight	Twin Genoa Gasoline engine	N/A
Yamaha R-50	3.5 m	1.08m	54kg (44kg+10kg avionics)	Water cooled 2stroke 1 cylinder gasoline engine	Boeing DQI NovAtel RT-2 Pentium233 4 ultrasonic altimeter Digital compass Vision processor
Yamaha RMAX	3.63 m	1.2m	60kg dry weight	Water cooled 2 stroke 2 cylinder gasoline engine	N/A

Flight control computer is constructed using PC104-compatible boards due to their industry-grade reliability, compactness and expandability. The main board is powered by Pentium 233MHz MMX CPU with 64MB RAM and 72MB FlashRAM. Serial port expansion board, counter/timer board, custom take-over board (TOB), and DC-DC conversion power supply board are "stacked up" on the CPU board via PC104 for communication, servo control and power supply, respectively. As the heart of the navigation sensor, Boeing DQI-NP inertial navigation system (INS) is adopted. DQI-NP consists of a pack of solid state inertial sensors and digital signal processors with serial port. DQI-NP outputs the navigation solutions in proprietary message format via serial port. It needs periodic position update from external sensor to correct the position estimation error.

Figure 4. One of Berkeley UAVs: *Ursa Magna 2*

The global positioning system (GPS) used in this research is NovAtel RT-2, which has remarkable accuracy of 2cm. The flight control computer acquires the position, linear/angular velocity and attitude from DQI-NP, and high accuracy position estimate from NovAtel RT-2 via RS-232. It also relays the converted position estimate message packet from GPS to DQI-NP every second. Based on the acquired navigation data, FCC computes the control output for four channels: main rotor collective pitch, tail rotor collective pitch, main rotor longitudinal cyclic pitch, and lateral cyclic pitch. These control surfaces are actuated by commercially available servomotors, which accept PWM signal (14-21 ms period, 0.8-2.4 ms on duty) as the reference command. The output angle of servo rotor is proportional to the duty-on duration. The PWM signal is generated by Intel 8254 counter/timer chip. To ensure safety, a special circuit is added on the TOB to switch from FCC control to human pilot command by a toggle switch on the radio transmitter. Other channels of counter/timer board read the radio receiver output to log the human pilot's command, which has been proven to be extremely valuable for system identification and feedback-assisted flight.

The communication module contains two 900MHz wireless modem cards and one 2.4GHz wireless Ethernet card. The type of communication device is chosen based on the mission type. The wireless modems are preferred for long range mission because of their superior range up to 20 miles. The drawback is the relatively slow throughput (<11.5kbps). One wireless modem is used for data communication and the other is used for reading the differential GPS broadcast data. In normal situation, 2.4GHz wireless LAN is preferred because of their high bandwidth (up to 11Mbps), versatility, and low power output minimizing the potential interference with sensitive GPS operation. Currently, wireless Ethernet system is used as the backbone of multi-agent system consisting of multiple number of UAV, UGV, the ship motion simulator and the ground station.

Ground station consists of a GPS base station and a portable computer connected with a communication device such as wireless modem or wireless Ethernet. Ground station monitors and stores the flight data of the UAV and also sends the navigation commands comprised of vehicle control language as explained in Sec. V.

Larger UAVs are equipped with an onboard vision processing unit (VPU) and a camera actuated by pan-tilt-zoom platform. VPU can track a target object with certain color and computes its coordinate based on the navigation data received from the onboard FCC via serial link. It can be accessed by its independent wireless Ethernet for monitoring and debugging purposes. VPU serves the vital role for vision-based landing, ground object detection and map building.

3. System Identification

The acquisition of high fidelity system model of target UAV is a crucial step towards the successful design of high-performance flight control system. In general, however, it is often a challenging task to perform system identification of a rotorcraft based UAV system due to its multi-input multi-output (MIMO), nonlinear characteristics, severe noise and disturbance, and wide flight envelop.

As our first step, we attempted to obtain a linear, time-invariant model valid in near-hover conditions. A 6-degrees-of-freedom linear rigid body helicopter model augmented with first-order approximation of servorotor dynamics is given by a differential equation

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{1}$$

where

$$\mathbf{x} = [u \; v \; p \; q \; \Phi \; \Theta \; a_{1s} \; b_{1s} \; w \; r \; r_{fb}]^T \tag{2}$$

$$\mathbf{u} = [u_{a_{1s}} \; u_{b_{1s}} \; u_{\theta_M} \; u_{r_{ref}}]^T \tag{3}$$

u, v, w: body-coordinate velocity
Φ, Θ, Ψ: roll, pitch, yaw angle, respectively
p, q, r: roll, pitch, yaw rate, respectively
a_{1s}, b_{1s}: flapping angle

$$\mathbf{A} = \begin{bmatrix}
-0.0954 & 0 & 0 & 0 & 0 & -g & -g & 0 & 0 & 0 & 0 \\
0 & -0.2221 & 0 & 0 & g & 0 & 0 & g & 0 & 0 & 0 \\
-0.2047 & 0.1521 & 0 & 0 & 0 & 0 & 22.140 & 32.995 & 0 & 0 & 0 \\
-0.0836 & -0.0514 & 0 & 0 & 0 & 0 & 67.74 & 142.50 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & -2.6645 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0.5543 & -2.6645 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -28.850 & -121.20 & -0.5377 & 5.7974 & 0 \\
0 & 0 & -0.0178 & 0 & 0 & 0 & 0 & 0 & 0.0746 & -4.4017 & -46.959 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.3394 & -5.4830
\end{bmatrix} \tag{4}$$

$$\mathbf{B} = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
-0.5912 & 1.9729 & 0 & 0 \\
-2.4055 & -0.0993 & 0 & 0 \\
0 & 0 & 116.9520 & 0 \\
0 & 0 & -46.9690 & 15.2454 \\
0 & 0 & 0 & 0
\end{bmatrix}$$

Table 2. Eigenvalues of the identified helicopter system

Mode	Value
Phugoid 1	-0.4419±0.0859j
Phugoid 2	0.2796±0.0870j
Roll	-1.3608±11.7676j
Pitch	-1.3001±8.2726j
Yaw	-4.9507±10.4501j
Heave	-0.5210

r_{fb} : feedback gyro system state .

One candidate model of eleventh order takes the form of Eq. (4) as suggested by Mettler *et al* [2]. One distinction of this model is the explicit account for the *servorotor*, which modifies the helicopter dynamics significantly. The most important role of servorotor is to slow down the roll and pitch response so that human pilot on the ground can control the helicopter with a remote controller. Heave dynamics is approximated by first order quasi-static model. This model yields linear, low-order approximation of the nonlinear high-order heave dynamics. For higher bandwidth controllers, the third or fourth order of model containing inflow and flapping dynamics should be used [7]. Yaw dynamics is inherently stable and modeled as first order system with reasonable fidelity. One special feature of the yaw dynamics is the built-in feedback action of yaw rate in the loop, which is provided by the built-in rate gyro amplifier/mixer. Even though uncompensated yaw dynamics is stable, the variation of the anti-torque of the main rotor continually perturbs the heading of the helicopter. The yaw rate feedback counteracts the torque by compensating the tail rotor collective pitch and left in the UAV system in case of manual flights. The gyro system model suggested by Mettler[2] is effective to account for the yaw dynamics. One deficiency of the model (4) is the absence of the cross coupling from yaw to sideslip and roll. It can be additionally parametrized in the model, but it turned out during the numerical process that the additional parameters are cumbersome to find because they appear as a product of parameters and the numerical process becomes singular.

Since the model is treated as a linear model, the nonlinearity of helicopter aerodynamics should not be excited by the excessive amount of control action. Hence, the appropriate design of control input signals for flight test is extremely important for the identification of target flight dynamics. A number of experimental flights have been made to collect the flight data and the pilot input at 50Hz sampling rate. The control input consists of the combination of frequency-sweeping and random signals in longitudinal, lateral, yaw and heave channels in turn. At the first stage, the control in longitudinal and lateral channels are given simultaneously to capture the coupling between these axes while other channels are controlled to maintain constant altitude and heading. In the next stage, main rotor collective pitch or tail collective pitch is perturbed. Finally, control signals are issued in all channels to capture the cross-coupling term.

Before processing the data, the angular rate signals are filtered by zero-phase noncausal discrete-time filters to filter out high frequency noise without introducing phase delay. The experiment result is processed using time-domain output-error minimization tool from MatLAB™ System Identification Toolbox™. The prediction error method (PEM) is an estimation algorithm, which seeks to a set of parameters minimizing the quadratic error between the predicted output and experiment data[8,9]. It should be noted that this method is extremely sensitive to the initial guess of the parameters and easily trapped in local minima of the parameter hypersurface. To obtain meaningful results other than some parameter set that blindly matches the time history, the following technique is devised. First, the angular dynamics augmented with rotor dynamics is identified using initial guess. Since the angular rate/rotor dynamics is known to be stable and small number of parameters are involved, the numerical solution converges to consistent solutions. Then the horizontal dynamics, i.e., the longitudinal and lateral dynamics with linear velocity terms u and v, are identified while fixing the angular dynamics parameters. This stage is rather challenging due to the unstable linear velocity dynamics. Shorter length of experiment data should be used to avoid the instability of the predictor and the divergence of the prediction error with the small mismatch of the initial condition and parameters. The solution is found after a large number of iterations using the experimental data from different time intervals. Separate from the longitudinal and lateral dynamics, the heave and yaw dynamics is identified in similar manner. The inherent stability of yaw and heave allows nice convergence of the parameters. Once these two subsystems are identified, they are combined as the full-model dynamics and then the cross-coupling terms are estimated. Finally, a small number of iteration is performed to recalibrate the parameters in the subsystems.

4. Controller Design and Experiments

Based on the identified model in Sec. 3, stabilizing control law is designed. As our first approach, we employ single-input, single-output (SISO) control structure. Classic control design approach has a number of advantages such as simple structure, straightforward design process and low computing load imposed on the FCC. On the contrary, it is limited by a number of disadvantages: it does not provide a systematic way to account for uncertainty, disturbance and/or noise. Moreover, it has limited capability to alleviate the coupling among channels.

The proposed control system consists of three loops: innermost attitude controller, mid-loop linear velocity controller, and the outer loop position controller, as shown in Figure 5.

The attitude dynamics of RUAV with servorotor mechanism has a unique dynamics, which is significantly different from the full-size helicopter without servorotor system[1]. The attitude dynamics of RUAV can be considered marginally stable if translational velocities are fixed at zero. On the contrary, the full-size helicopter without servorotor exhibits unstable attitude dynamics, which must be stabilized by angular feedback. The attitude controller proposed in this research only feed back the deviation of the roll and pitch angles from the trim condition and does not feed back the noisy angular rates p and q measured by solid state rate gyros. This approach yields a controller that is simpler and more robust to mechanical vibration. The adequate angular feedback gains for roll and pitch channels are determined to have acceptable response speed and damping by using root locus and step response.

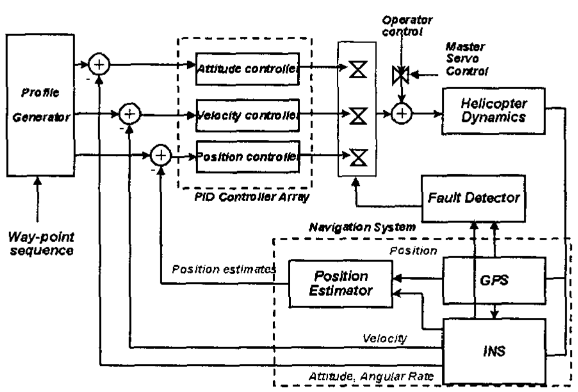


Figure 5. The proposed controller architecture using SISO multi-loop controllers

The translational velocity dynamics of RUAV is almost identical to that of a full-size helicopter. This part of dynamics is unstable even with attitude feedback and requires velocity feedback. It can be stabilized with velocity feedback with constant gain, which is again found by conventional root locus and step response method.

For position controller design, the system model (2) is augmented with three parallel integrators, which integrate translational velocities of x, y, z direction, respectively. The actual position integration, however, involves the coordinate transformation using Euler angles, which cannot be described within linear equations. Hence, the position control involves internal coordinate transformation to compensate for the heading change. The position gains are found by applying the similar methods described above to the augmented RUAV dynamics with velocity and attitude feedback.

Apart from x-y directional dynamics, the vertical and the heading control require their own controllers. Although vertical and heading dynamics shows considerable coupling, conventional SISO control approach is maintained. These sub-dynamics show inherently stable response due to the natural equilibrium of the change of inflow and the generated lift. The vertical response can be further improved by introducing artificial damping by negative velocity feedback. For yaw rate response, the dynamics is already compensated enough by built-in gyro system. Still, the yaw angle response is marginally stable and the yaw tracking system can be built by simple heading error feedback with constant gain in its simplest form. The proposed controller based on fully decoupled SISO model is then tested on the full model, investigating the level of coupling when the loop is closed. The simulation showed that the closed loop system shows satisfactory performance.

This simple structure of classical approach enables simple, but very effective control algorithm. In cruise mode, the velocity and attitude loops are closed for stabilization and tracking. When hovering over a certain spot is required, the outmost loop for position feedback is closed along with the inner loops. This algorithm is implemented in FCC and showed this idea actually works for real situation.

The proposed control algorithm is implemented on the FCC. FCC is running on QNX real-time operating system. The onboard navigation and control software has two main processes and two auxiliary processes running concurrently, interacting with INS, GPS, servos, ultrasonic sensors, and vision computer. Process *DQIGPS* reads the GPS information and updates the INS using the GPS position estimate at 1Hz. Process *DQICONT* reads in the INS measurements at 100Hz. These INS and GPS measurements are stored in the shared memory space accessible to client processes. The control output is calculated using INS and GPS measurement at 46Hz and then sent to the registers of counter/timer chip, which generates a set of five PWM signals. The output to five servos can be switched from the signal from radio receiver or the computer generated PWM signal transmitted over optoisolators to guarantee stable operation of built-in Yamaha controllers and onboard computer systems.

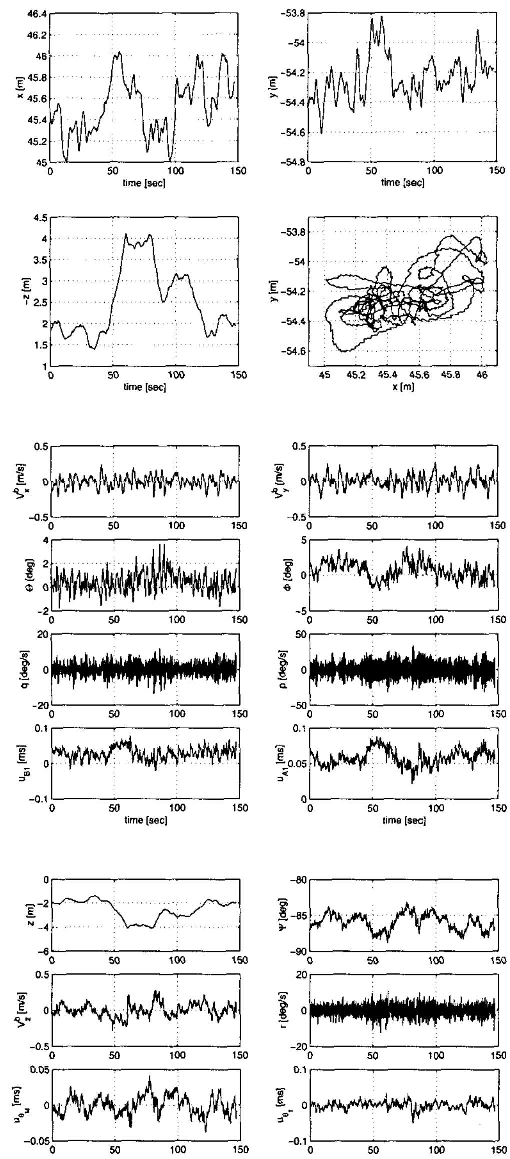


Figure 6. Experiment result of autonomous hovering

A series of experiments has been performed using the proposed controller on Yamaha R-50. The experiment is performed as described here: The RUAV is first placed at the test field, preferably flat location, and then the flight system and radio/servo system are turned on. The GPS system automatically starts tracking the GPS satellite signals to compute the current location coordinates. The DGPS broadcast are used to achieve the ultimate accuracy of 2cm. Once GPS is completely locked on, the DQI-NP system is initialized and sequenced into the *fine-alignment mode.* During this process, the DQI system should be left undisturbed to correctly estimate the scale and bias factors of the six solid state inertial sensors. After these two navigation sensors are initialized, the engine is started manually and the RPM is quickly brought up to 90 % of the hovering RPM to avoid any low frequency mechanical resonance harmful to proper INS/GPS operation.

During the repeated experiments, the attitude/velocity controller has shown stable operation even when the helicopter stays on the ground. Therefore, more accurate take-off and landing can be achieved by activating the attitude/velocity controller even before the helicopter takes off from the ground. When operated manually, the pilot engages the attitude/velocity controller using a switch on the transmitter and then takes the helicopter off from the ground. At this time, only steady heave reference command is given. Once the helicopter reaches the desired altitude, the hovering controller, i.e., the position/velocity/attitude loop controller is activated.

Figure 6 shows the experiment result of hovering controller tested on R-50 UAV. The RUAV showed a stable response over two minutes with ±0.5m accuracy in x and y direction. Roll, pitch, translational velocity in x and y directions are regulated very well altogether. The altitude regulation shows rather large variation because the engine was not regulated to a constant RPM. A simple proportional-integral(PI) controller will be added in system for more stable response.

5. Vehicle Control Language Framework

Once the regulation layer for the vehicle control is designed, then supervising control logic should be integrated with the vehicle regulation layer to guide the RUAV along the desired trajectory. This layer plays an important role to relay the mission control layer and the vehicle-level control layer by generating appropriate reference trajectories and then injecting them into the proper controller, which is activated accordingly depending on the flight status and the target mode.

Vehicle Control Language, or *VCL*, is an application and development tool for UAV systems, operating in a hierarchical structure as shown in Figure 7. This approach allows to program complex behavior of the UAV adaptively without rewriting program as the mission changes. The sequence of motion commands is described in a script language form understandable to human. VCL module consists of user interface part on ground station, language interpreter and sequencer on the UAV side.

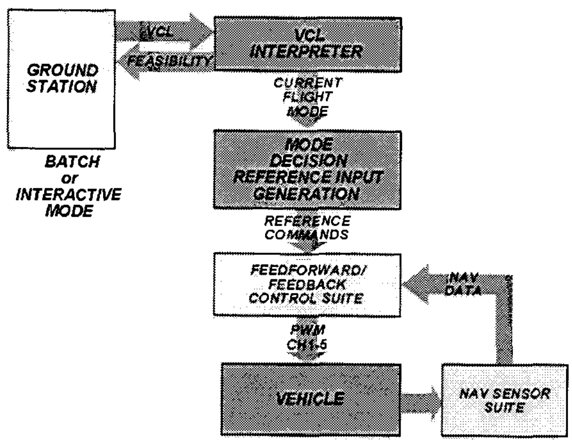


Figure 7. Hierarchical structure of VCL processing

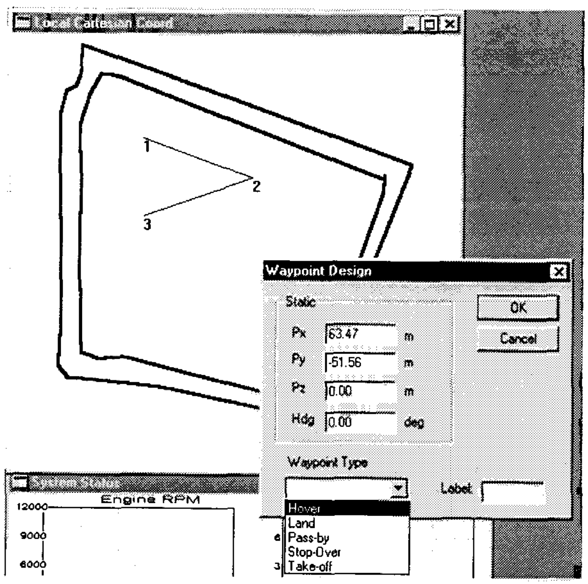


Figure 8. Waypoint generation using graphical user interface

When a mission is given, the ground operator generates a sequence of waypoints with their attributes such as the type of waypoint, heading, velocity and so on as shown in Figure 8. When finished, the VCL is uploaded to the RUAV control system and then executed in sequential manner as shown in Figure 9. The VCL execution module (VCLEM) selects the proper controller depending on the flight mode and generates the reference command. VCLEM monitors the vehicle trajectory and determines if one sequence is finished or not. It also monitors the vehicle status for possible troubles in sensor or the vehicle itself. If error detected, the fault detection algorithm shown in Figure 5 is activated and proper error handling measure is executed. In the worst case, the VCL releases the automatic vehicle control mode and return the control to the ground-based test pilot. This routine is repeated until the end of VCL command script is reached and the RUAV returns to its default flight mode. Finally, sample VCL codes are given in Figure 10.

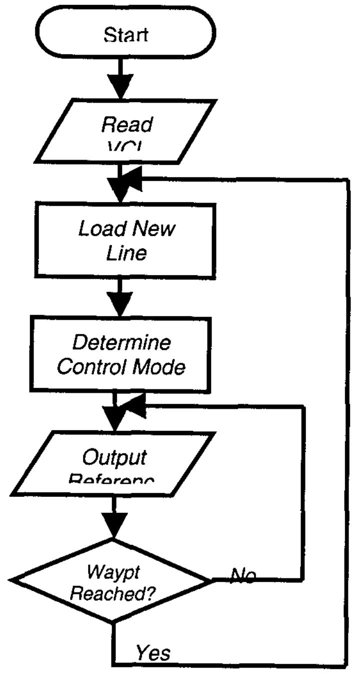
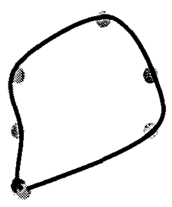


Figure 9. Flowchart of VCL execution module

6. Conclusion

This paper introduced the development of hierarchical RUAV controller design conducted at University of California at Berkeley. The vehicle dynamics model is identified using the parametric model suggested by Mettler. A different approach based on the time-domain analysis tool is applied to the experiment data and linear time domain model is obtained. The low-level vehicle control layer based classical SISO approach and implemented on a Berkeley RUAV, Yamaha R-50. The proposed controller showed satisfactory autonomous hover flight capability. The controller, or any other vehicle level controller, is integrated with higher level behavioral control logic called VCL. Currently, the proposed paradigm is being implemented on Berkeley UAVs and will be tested in near future.



```
GO AUTO
WaitFor Go
TakeoffTo(0,0,-2)abs for 30sec
FlyTo (5,10,0)rel vel 5mps
passby autoheading
FlyTo (0,5,0)rel vel 10mps
stopover autoheading
Hover (0,0,0)rel for 20sec
FlyTo (8,-5,0)rel vel 5mps
passby autoheading
FlyTo (-5,-10,0)rel vel 5mps
passby autoheading
FlyTo (-5,0,0)rel vel 5mps
stopover autoheading
FlyTo (0,0,0)abs vel 5mps
passby autoheading
Hover (0,0,0)rel for 40sec
Land
GO MANUAL
```

Figure 10. Sample VCL code for free-style waypoints

7. Acknowledgement

This research was supported in part by the Army Research Office DAAH04-96-1-0341 and by the Office of Naval Research under Grants N00014-97-1-094.

8. Reference

[1] R.W. Prouty, *Helicopter Performance, Stability and Control*, Krieger Publishing Company, 1995.

[2] M. B. Tischler, "System Identification of Small-Size Unmanned Helicopter Dynamics," *American Helicopter Society 55th Forum*, Montreal, Quebec, Canada, May 1999.

[3] H. Shim, T. J. Koo, F. Hoffmann, S. Sastry, "A Comprehensive Study of Control Design for an Autonomous Helicopter," *37th IEEE Conference on Decision and Control*, pp. 3653-3658, 1998.

[4] C. P. Sanders, P. A. DeBitetto, E. Feron, H. F. Vuong, N. Leveson, "Hierarchical Control of Small Autonomous Helicopters," *37th IEEE Conference on Decision and Control*, pp. 3629-3634, 1998.

[5] J. E. Corban, A. J. Calise, J. V. R. Prasad, "Implementation of Adaptive Nonlinear Control for Flight Test on an Unmanned Helicopter," *37th IEEE Conference on Decision and Control*, pp. Pp. 3641-3646, 1998.

[6] D. H. Shim, H. J. Kim, S. Sastry, "Control System Design for Rotorcraft-based Unmanned Aerial Vehicles using Time-domain System Identification," *IEEE Conference on Control Applications*, Anchorage, Alaska, 2000.

[7] R. T. N. Chen, M. B. Tischler, "The Role of Modeling and Flight Testing in Rotorcraft Parameter Identification," The 1986 AHS Forum, System Identification Session, Washington, D.C., 1986.

[8] L. J. Ljung, *Matlab System Identification Toolbox User's Guide* , The Math Works, Inc.,1997.

[9] L. J. Ljung and E. J. Ljung, *System Identification: Theory for the User*, 2nd ed., Prentice-Hall, 1998.