Hardware-In-Loop Simulator for autonomous Navigation of Mini Aerial Vehicle

Vishisht V. Gupta and Prasanna S. Gandhi

gandhi@me.iitb.ac.in

Department of Mechanical Engineering

H. Arya and K. Sudhakar

arya@aero.iitb.ac.in, sudhakar@aero.iitb.ac.in

Department of Aerospace Engineering

Indian Institute of Technology Bombay

Powai, Mumbai-400076, INDIA

Abstract

Hardware-In-Loop Simulation (HILS) is increasingly being used in design and development of control systems. In HILS some of the control-loop components are real hardware, and some are simulated. This paper describes the steps that have been taken for the implementation of Hardware-In-the-Loop Simulator for mini air vehicle. Flight dynamics simulation (FDS) has been done for the aircraft in real-time operating system RTLinux. Input/Output cards have been used for A/D and D/A conversion for giving the sensor output in the same form as in the real scenario. These input/output cards have been configured in RTLinux using Comedi drivers for data acquisition. Way-point navigation has been used. Straight Line Guidance and Line of sight Guidance are two guidance strategies used to guide the vehicle between two way-points. Longitudinal and directional control loops control altitude, air speed and heading of the aircraft. This NGC strategy was implemented on Motorola micro-controller 68332. Off-line simulations (where FDS and controller are in the same computer) have been done to find out approximate gains for controller to be used in HILS. NGC strategy is verified using off-line simulation. Finally, full HILS were performed by putting the controller and actuators in loop with flight dynamics simulator. Trajectories obtained by HILS were compared with those obtained from off-line simulations. Hardware-in-the-loop simulator is verified for implementation.

1 Introduction

Modeling and simulation technologies have helped in system development and successfully shortened design and development cycle of complex systems. Full software simulation of a system requires a high fidelity model of the complete system. In case a subsystem which cannot be adequately characterized by mathematical models, it is safe to embed it, as it is, into the simulation e.g. actuator dynamics is often difficult to model and embedded microcontroller with its resident code is hard to take into account. Simulation consisting of actual components is known as Hardware-In-Loop Simulation (HILS). Use of HILS, where actual hardware is embedded into the simulation started in aircraft and space applications and is slowly percolating to other industries. When hardware or human is embedded into a simulation, the simulation of all its components has to proceed in “real-time”.

A computer software with real-time simulation capabilities and a computer with necessary communication abilities (A/D, D/A converters for communications with analog signals and digital ports for communication with digital signals) is necessary to perform hardware-in-the-loop simulation. In this paper such a system for Mini Air Vehicle (MAV) development is presented.

2 Proposed System

In the present work a system is developed to enhance the capability of a radio control aircraft as an autonomous vehicle. Radio control aircraft are low cost, easily available and a good resource for an educational institute for development of flight mechanics, navigation guidance and control studies. The proposed autonomous aircraft should be able to complete an autonomous mission through a number of given
way-points. It should also be able to perform some predetermined maneuvers on reaching the destination way-points, e.g. horizontal circle, figure of eight etc. A HILS facility is envisaged to support the development of autonomous capability. Hardware which can be part of the simulation loop are on-board computer, actuators, transmitter and receiver. The scheme of such a system is shown in Figure 1. The steps involved in systematic development are:

1) Development of full 6 DOF non-linear Flight Dynamics Simulation (FDS) and its verification
2) Development of sensor models and integrating with FDS
3) Simulating sensor output using Data Acquisition Cards (DAC) and other hardware
4) Integrating above for real-time simulation
5) Identification and development of Navigation, Guidance and Control (NGC) algorithm
6) Integrating NGC with FDS, evaluate control update rate by offline simulation
7) Identifying the on board computer to execute NGC algorithm in the time step arrived by above
8) Integrating actuators with onboard computer
9) Integrating transmitter and receiver with onboard computer
10) Feedback from the actuators to FDS
11) Conduct simulation for the various NGC algorithm, mission etc.

In the present work all except 10 have been completed. NGC algorithm is fashioned along that of a proven autonomous vehicle [1]. There are two control loops called longitudinal and directional control loops. Longitudinal control loop controls altitude and velocity of the aircraft as shown in Figure 2 and directional control loop controls the heading of the aircraft as shown in Figure 3. A yaw rate damper acts as stability augmentation system for the directional loop. The emphasis in the present work is the integration of the system and conducting few studies on autonomous mission. In the following sections details about the various systems and justification for selection is presented. FDS code is in 'C' and executes in real time. Fourth order fixed time step Runge Kutta method was used for time integration. FDS is verified offline with Matlab add-on, FDC [2]. In the present study for all the simulation Beaver aircraft data is used which is part of the FDC. Simulation results obtained from FDS and FDC were having error less then 0.001 %. NGC of reference 1 uses vertical gyro for pitch and roll angle, yaw rate gyro for damping directional motion. Altitude and air-
speed sensor for height and indicated airspeed. Linear models for these sensors are used in the present work.

![Figure 2: Longitudinal Control Loop.](image)

For simulation of analog sensors 12 bit DAC card PCI-DDA12 is used. The selection of this card is based on the ease of its use with RTLinux. It has eight DAC channels and this can simulate eight analog sensors, in the present case five are used.

Real time operating systems are expensive and the support for interface cards will add more cost to it. A shareware OS RTLinux, which supports the above interface cards is used in the present study.

On-board computer selection is based on the low weight, low power consumption and adequate computational capability. From literature it was observed that Motorola 68332 processor has been used in autonomous mission of MAV [3]. Special feature of this processor is that it can be easily configured for interfacing with radio control servos using its time processing unit. RC servos require pulse width modulation for angular position control. Same servos which will be used in actual mission are part of the simulation. For feed back to FDS signal from the internal potentiometer of the servo is used. It has eight 12 bit ADC @ 100 kHz for sensor input.

Various issues involved in the development of the setup are given below:

Real time implementation :- RTLinux is running in two modes that is periodic thread and interrupt handler (Figure 4).

RT-TASK 1 is a periodic real-time task with a period equal to the frame time of the FDS i.e. 1 msec. Its job is to take the position of control actuators (elevator, aileron, throttle and rudder) though I/O cards at the start of every time step, do all the calculations and obtain the state of the aircraft at the next time step. At the end of time step it gives sensor output. When the real-time simulation is started this process also does a one time job of reading the aircraft and engine data and initial states from files by communicating with a user level task through RT-FIFOs. The RT-TASK 1 creates the real-time FIFOs for the user level task to put data into them and then reads the RT-FIFOs to get the required data. This process is the main thread in our application. It gets second priority after the RT-TASK 2. The approximate time it takes to execute has been measured to be 270µsecs.

RT-TASK 2 is an interrupt based task. In each time step RT-TASK 1 stores data output from the heading sensor inside proper data-structure. When microcontroller interrupts the CPU using IRQs requesting heading data input, this process wakes up and supplies data to the micro-controller in proper format. This task is given the first priority, so the micro-controller request for data gets immediate attention by the FDS.

RT-TASK 2 is useful for simulation of sensors with IIC interface. Time taken for such communication is much less than 100 microsecond and this can be easily accommodated in the spare time available. In the present sensor set it is not required. Independent study was carried out to understand the behavior of system when high priority interrupts are raised.

Comedi drivers for interfacing cards: Comedi (Control and Measurement Device Interface) for Linux is a collection of driver for a variety of common data acquisition plug-in boards. In the present PCI-DAS card, the outputs from ADC were not settling with the default Comedi setup, delay of 5 micro second was introduced to overcome this problem.

Feed back to FDS can be given by connecting a potentiometer and the servo face to face. Mechanical coupling between the two is an important issue. Second option is to tap the signal from the internal
potentiometer of the servo. In the present setup initially mechanical coupling was used but it was not giving satisfactory repeatability. Later direct internal signal was used for observing servo movement. Figure 5 shows the Voltage PWM characteristic of servo for clockwise and anticlockwise motion.

DAC card was giving high frequency (> 1KHz) noise and this was suppressed using hardware filter (15 Hz) between the onboard computer and simulation computer. The output was within 5% of the desired value.

Off line simulation: Offline simulation was done using 'C' code and the results were compared with the data available in the literature. After verifying the flight dynamics, NGC algorithm was added for autonomous mission simulation. Stability augmentation system for yaw motion is used.

NGC algorithm: The Navigation strategy is waypoint navigation. In this navigation scheme aircraft passes through various point in 3D space (waypoints). This is achieved by implementing two control functions. i) Altitude and speed hold ii) Heading and track hold. Block diagrams of these control functions are given in Figure 6 and 7. Navigation is based on Air Data Dead Reckoning (ADDR) using Indicated Air Speed (IAS) and heading. The navigation error is corrected by GPS data which is available every second. For guidance two strategies are used i) Straight Line Guidance (SLG): In this strategy aircraft tries to follow the straight line joining two waypoints. Guidance is achieved by reducing cross track error as shown in Figure 6. ii) Line of Sight Guidance (LOSG): In this strategy, at any instant the aircraft tries to follow the line joining its position at that instant and the waypoint it is headed towards as shown in Figure 7. Reference heading i.e. heading of line joining aircraft and next waypoint is the guidance criterion. Choice of guidance strategy depends on the mission specifi-
cation. In case of SLG aircraft will follow a corridor along the line joining two waypoints. Whereas in case of LOSG it is only guaranteed that aircraft will pass the waypoint. In the present case when the aircraft reaches within 100 m radius of target waypoint, the waypoint is assumed to be reached and it starts heading towards the next waypoint.

\[ \text{REF. HEADING} = (\text{CONSTANT} \times \text{YERROR}) \]

![Figure 6: Straight Line Guidance.](image)

Onboard software: Onboard software was developed in the 'C' using the Tattletale development environment. Present micro-controller is a 32 bit architecture and without floating point unit, each trigonometric calculation takes about 2 msec. This was reduced by using lookup table to about 0.6 msec.

Real-time simulation: Real-time simulation was done in asynchronous mode. This means that there is no communication between FDS and on-board computer to synchronise events. Both are running at their own real time clock and it is assumed that both are working with wall clock with different off-set. Flight dynamics time step is 1 msec and controller time step is 15 msec. The mismatch in the sensor output and on-board computer sensing can be maximum of 1 msec. Advantage of asynchronous mode simulation is that the on-board computer can be used after testing without any modification.

<table>
<thead>
<tr>
<th>Gain Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altitude Proportional Gain</td>
<td>0.008</td>
</tr>
<tr>
<td>Altitude Integral Gain</td>
<td>0.0002</td>
</tr>
<tr>
<td>Pitch Angle Proportional Gain</td>
<td>1</td>
</tr>
<tr>
<td>Pitch Angle Integral Gain</td>
<td>0.02</td>
</tr>
<tr>
<td>IAS Proportional Gain</td>
<td>0.04</td>
</tr>
<tr>
<td>IAS Integral Gain</td>
<td>0.0007</td>
</tr>
<tr>
<td>YError Proportional Gain</td>
<td>0.0015</td>
</tr>
<tr>
<td>Roll Gain</td>
<td>2</td>
</tr>
<tr>
<td>Roll Angle Proportion Gain</td>
<td>1</td>
</tr>
<tr>
<td>Roll Angle Integral Gain</td>
<td>0.004</td>
</tr>
<tr>
<td>Yaw Gain</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 1: Gains for HILS Simulations.

3 Results and Discussion

Offline simulation were done to compute the first guess of the gains for the control strategy. Later these gains were modified in HILS. Table 1 shows the gains used during these simulations. Simulations were carried out for SLG and LOSG under different wind conditions using only ADDR and with GPS assisted corrections. It was also noted during off-line simulations that GPS assisted navigation gives path closer to the intended path especially in presence of winds.

Test case simulations for autonomous missions consisting of way-points in vertical and horizontal plane were carried out. Figure 8 shows a mission consisting of six waypoints in vertical plane. Corresponding throttle servo movement is shown in Figure 9.

![Figure 8: Simulation in Vertical Plane.](image)
Another test case consisting of four waypoints in horizontal plane with an inter-waypoint distance of ≈ 1400 meters as shown in Figure 10. Corresponding aileron servo movement is shown in Figure 11.

Using HILS setup complex missions consisting of 10 way-points as shown in Figure 12 were carried out. The trajectories from HILS compare well with those of off-line simulations but for small deviations. In Off-line simulations actuator dynamics and interface issues (noise, filter delays etc.) are not included and these are difficult to model. The difference in trajectories obtained using off-line and HILS can be attributed to these. Study was also carried out to see the effect of time step of the on-board controller and good path following was observed over a range of 10 msec to 250 msec.

In the present work balance between cost and the efforts required for development of the system were considered. Decision on selection of DAC/ADC cards was based on the support available in RTLinux to reduce the cost of drivers. Use of higher language in embedded system is common and ‘C’ development environment for this became handy in the present case.

Acknowledgements

Authors wish to thank AR&DB for the financial support in execution of the project.

References

Information about Authors

Vishisht V. Gupta
Graduate student
Department of Mechanical Engineering
Indian Institute of Technology Bombay
Powai, Mumbai-400076
INDIA

Prasanna S. Gandhi
gandhi@me.iitb.ac.in
Assistant Professor
Department of Mechanical Engineering
Indian Institute of Technology Bombay
Powai, Mumbai-400076
INDIA
Phone : + 91-22-25767519 (o),
FAX : + 91-22-25726875

H. Arya
arya@aero.iitb.ac.in
Assistant Professor
Department of Aerospace Engineering
Indian Institute of Technology Bombay
Powai, Mumbai-400076
INDIA
Phone : + 91-22-25767118 (o),
FAX : + 91-22-25767840

K. Sudhakar
sudhakar@aero.iitb.ac.in
Professor
Department of Aerospace Engineering
Indian Institute of Technology Bombay
Powai, Mumbai-400076
INDIA
Phone : + 91-22-25767111 (o),
FAX : + 91-22-25767840