

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



The Development of a Hardware-in-the-Loop Simulation System for Unmanned Aerial Vehicle Autopilot Design Using LabVIEW

Yun-Ping Sun

*Department of Mechanical Engineering, Cheng Shiu University,
Taiwan*

1. Introduction

This chapter describes a continuing research on design and verification of the autopilot system for an unmanned aerial vehicle (UAV) through hardware-in-the-loop (HIL) simulation. UAVs have the characteristics of small volume, light weight, low cost in manufacture, high agility and high maneuverability without the restriction of human body physical loading. Equipped with the on-board autopilot system an UAV is capable of performing out-of-sight missions that inspires scientists and engineers with a lot of innovative applications. Not only in the military but also in the civil, the applications of UAVs are in full bloom. Apparently one of the key challenge of UAV research and development is its autopilot system design.

For the purpose of designing an autopilot, the technology of hardware-in-the-loop simulation plays an important role. The concept of HIL simulation is that a stand-alone personal computer is used to simulate the behavior of the plant, several data-acquisition devices are exploited to generate the real signals, and the prototype controller can be tested in real-time and in the presence of real hardware. HIL simulation presents a new challenge of control engineering developers as the “correctness” of a real-time model not only depends upon the numerical computation, but the timelines with which the simulation model interacts with external control equipment that is the major difference between HIL simulation and numerical simulation. Due to the useful feature, HIL simulation is applicable to solve many problems in engineering and sciences effectively (Shetty & Kolk, 1997; Ledin, 2001).

HIL simulation provides an effective technique for design and test of autopilot systems. Using HIL simulation the hardware and software at subsystem level perform the actual input/output signals and run at real time so that the test target (e.g. prototype controller) is working as if in real process. This provides the ability to thoroughly test subsystems under different working loads and conditions; therefore, engineers can correct and improve their original designs early in the development process. The advantages of HIL simulation are reducing the risk in test and shortening the development time. Especially HIL simulation is suitable for critical or hazardous applications.

(Cosic et al., 1999) used TMS320C40 DSP to set up a HIL simulation platform for a semi-automatic guided missile system. (Carrijo et al., 2002) applied HIL simulation to test the on-board computer on a satellite launcher vehicle for motion and attitude control. (Sun et al., 2006; Sun et al., 2008a; Sun et al., 2009; Sun et al., 2010) developed the HIL simulation system to evaluate the performance of UAV autopilot that was employed different control laws.

Some valuable applications in traffic control (Bullock et al., 2004) and UAV design (Cole et al., 2006; Salman et al., 2006) using HIL simulation can be found.

The hardware arrangement of HIL simulation includes a personal computer used to simulate the behavior of an UAV plant, several plug-in data-acquisition (DAQ) devices used to acquire/generate the specific real input/output signals, and the embedded control system used to execute the control laws and send control signals to real hardware.

The software development environment for HIL simulation in this work is LabVIEW. LabVIEW is a graphical programming language widely adopted throughout industry and academia as a standard for data acquisition and instrument control software. LabVIEW provides an intuitive graphical programming style to create programs in a pictorial form called a block diagram that allows users to focus on flow of data within applications and makes programming easy and efficiency. Additionally, LabVIEW can command plug-in DAQ devices to acquire or generate analog/digital signals. In combination of LabVIEW and DAQ devices, a PC-based or embedded control system can communicate with the outside real world, e.g., take measurements, talk to an instrument, send data to another subsystem. These features are very helpful in building a HIL simulation system for UAV autopilot design. As a matter of fact, LabVIEW is not the only software development environment for HIL simulation, but based on the enumerated advantages it is certainly a nice choice.

In this chapter we are going to apply different control methods to accomplish the design of UAV autopilot, and compare their results in HIL simulations using LabVIEW. The chapter is organized as follows. Section 2 describes the dynamic model of an UAV. Section 3 outlines the HIL simulation system architecture and introduces hardware and software development environment LabVIEW. Section 4 focuses on stability augmentation system design. Section 5 focuses on autopilot system design including pitch attitude hold mode, velocity hold mode, roll attitude hold mode and heading angle hold mode. Section 6 concludes this chapter.

2. Dynamics model of an UAV

The MP2000UAV (Fig. 1) is a low-cost and small-volume unmanned aerial vehicle (UAV). Its physical properties are wingspan 1.75 m, length 1.4 m, wing area 0.5116 m², wing chord 0.29 m, and weight 3.84 kg. MP2000UAV equipped with a 40 in³, 1 HP, 2 cycle glow engine and a miniature autopilot is capable to carry out autonomous operations.



Fig. 1. Unmanned aerial vehicle MP2000UAV

The dynamics of an aircraft obey the equations of motion derived by Newton's second law. The forces and moments resulting from lift and drag, the control surface, the propulsion system, and gravity govern the aircraft. A body-fixed coordinate system shown in Fig. 2 that is fixed to center of mass and rotating with the aircraft is used to express these forces and moments. The rigid body equations of motion in body-fixed coordinates can be derived as: (Bryson, 1994; Cook, 2007)

$$\begin{aligned} m(\dot{U} - RV + QW) &= X - mg \sin \theta + T \cos \kappa \\ m(\dot{V} + RU - PW) &= Y + mg \sin \phi \cos \theta \\ m(\dot{W} - QU + PV) &= Z + mg \cos \theta \cos \phi - T \sin \kappa \end{aligned} \quad (1)$$

$$\begin{aligned} I_{xx}\dot{P} - (I_{yy} - I_{zz})QR - I_{xz}(\dot{R} + QP) &= L \\ I_{yy}\dot{Q} - (I_{zz} - I_{xx})PR + I_{xz}(P^2 - R^2) &= M \\ I_{zz}\dot{R} - (I_{xx} - I_{yy})PQ + I_{xz}(QR - \dot{P}) &= N \end{aligned} \quad (2)$$

where

- m = mass of the aircraft,
- g = gravitational acceleration per unit mass,
- I_{ii} = moments and products of inertia in body-fixed axes,
- U, V, W = components of the velocity of c.m. (center of mass) in body-fixed frame,
- P, Q, R = components of the angular velocity of the aircraft in body-fixed frame,
- X, Y, Z = components of the aerodynamic force about the c.m. in body-fixed frame,
- L, M, N = components of the aerodynamic moment in body-fixed frame,
- T = thrust force,
- θ, ϕ = Euler pitch and roll angles of the body axes with respect to horizontal,
- κ = angle between thrust and body x-axis,

The equations of motion derived for a body-fixed coordinate system are nonlinear that is difficult for analysis and design. In order to proceed for analysis and design they have to be linearized using the small-disturbance theory (Nelson, 1998; Roskam, 2007). On the assumption that the motion of the aircraft consists of small deviations about a steady flight condition, all the variables in the equations of motion are replaced by a perturbation.

The linearized equations of motion for an aircraft that describe small deviations from constant speed, straight and level flight can be divided into two fourth-order uncoupled sets representing the perturbations in longitudinal and lateral motion as follows: (Nelson, 1998)

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} Y_v & Y_p & -(u_0 - Y_r) & g \\ L_v & L_p & L_r & 0 \\ N_v & N_p & N_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} + \begin{bmatrix} 0 & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{j} \end{bmatrix} = \begin{bmatrix} Y_v & Y_p & -(u_0 - Y_r) & g \\ L_v & L_p & L_r & 0 \\ N_v & N_p & N_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ j \end{bmatrix} + \begin{bmatrix} 0 & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (4)$$

where

- u, v, w = perturbations of the linear velocity in body-fixed frame (unit: m/s),
 p, q, r = perturbations of the angular velocity in body-fixed frame (unit: deg/s),
 δ_e = deflection of control surface, elevator (unit: deg),
 δ_a = deflection of control surfaces, aileron (unit: deg),
 δ_r = deflection of control surfaces, rudder (unit: deg),
 δ_{th} = throttle setting,

and there are 13 dimensional stability derivatives in longitudinal model, including $X_u, X_w, X_{\delta_e}, X_{\delta_{th}}, Z_u, Z_w, Z_{\delta_e}, Z_{\delta_{th}}, \bar{M}_u, \bar{M}_w, \bar{M}_q, \bar{M}_{\delta_e}, \bar{M}_{\delta_{th}}$, and 14 dimensional stability derivatives in lateral model, including $Y_v, Y_p, Y_r, Y_{\delta_r}, L_v, L_p, L_r, L_{\delta_a}, L_{\delta_r}, N_v, N_p, N_r, N_{\delta_a}, N_{\delta_r}$ to be determined (Nelson, 1998; Roskam, 2007).

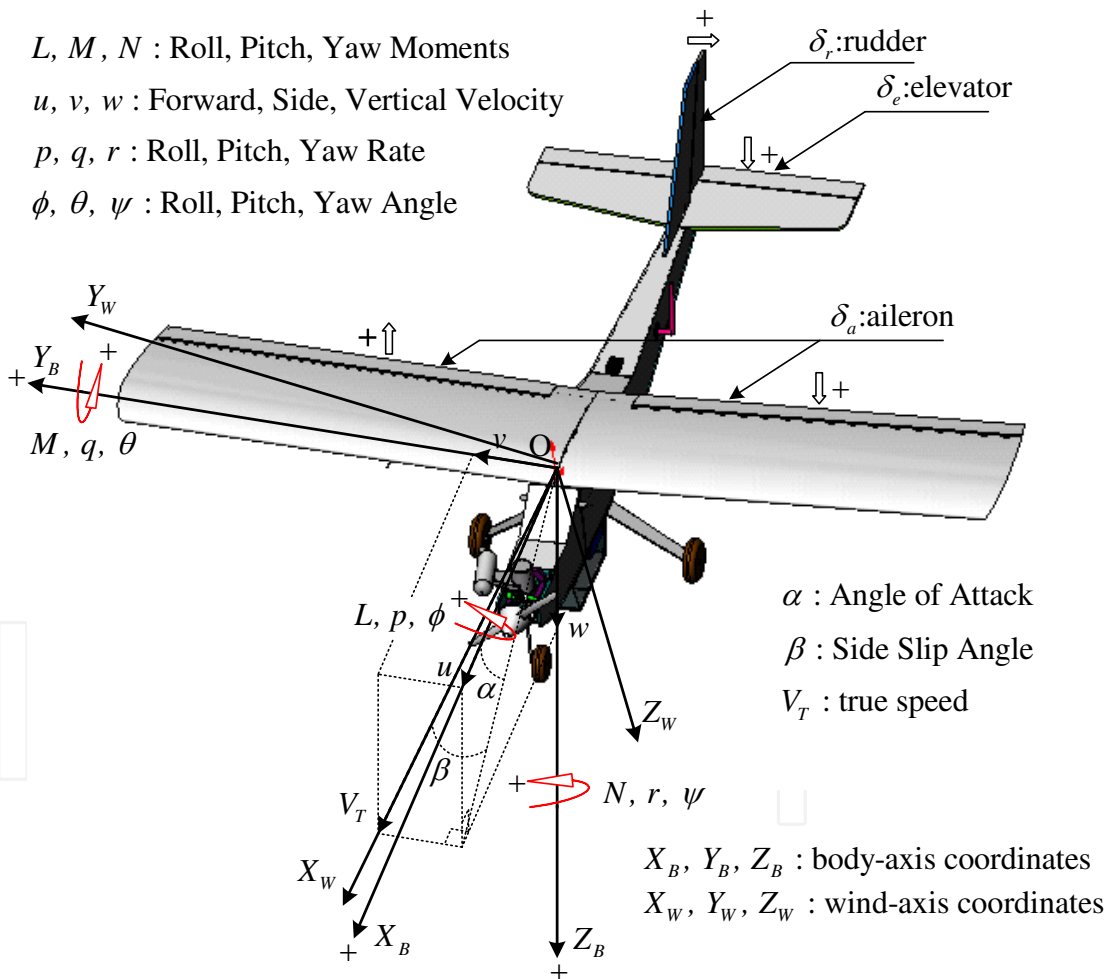


Fig. 2. Definition of aircraft coordinates

These stability derivatives of MP2000UAV are estimated from flight test data in a straight and horizontal flight condition, $u_0 = 16$ m/s at 80 m altitude and shown in Table 1 (Sun et al., 2008b).

X_u	X_w	X_{δ_e}	$X_{\delta_{th}}$	Z_u	Z_w	Z_{δ_e}	$Z_{\delta_{th}}$	\bar{M}_u
-6.114	0.789	-0.273	2.936	8.952	-9.220	3.919	143.24	1.275
\bar{M}_w	\bar{M}_q	\bar{M}_{δ_e}	$\bar{M}_{\delta_{th}}$	Y_v	Y_p	Y_r	Y_{δ_r}	L_v
-1.291	-1.366	-1.699	-64.192	-0.374	-5.113	0.764	-1.264	-2.136
L_p	L_r	L_{δ_a}	L_{δ_r}	N_v	N_p	N_r	N_{δ_a}	N_{δ_r}
-2.656	-5.414	0.967	5.974	0.584	1.250	1.307	-0.191	-4.969

Table 1. Estimation results of dimensional stability derivatives

As a result, for the unmanned aerial vehicle MP2000UAV, the longitudinal equations of motion in state-space form are:

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -6.113 & 0.7889 & 0 & -9.8 \\ 8.952 & -9.220 & 16 & 0 \\ 1.2746 & -1.290 & -1.3656 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} -0.273 & 2.9361 \\ 3.9191 & 143.24 \\ -1.699 & -64.192 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_{th} \end{bmatrix} \quad (5)$$

and the lateral equations of motion in state-space form are:

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -0.373 & -5.113 & -15.236 & 9.8 \\ -2.135 & -2.6564 & -5.4143 & 0 \\ 0.5837 & 1.2497 & 1.3069 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} + \begin{bmatrix} 0 & -1.2636 \\ 0.9666 & 5.9744 \\ -0.191 & -4.9689 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (6)$$

where the maximum deflection angles of elevator, aileron, and rudder are $\pm 15^\circ$, $\pm 6^\circ$, and $\pm 10^\circ$.

3. HIL simulation system

Hardware-in-the-Loop (HIL) simulation is a kind of real-time simulation that the input and output signals shows the same time dependent values as the real process. It is usually used in a laboratory environment on the ground to test the prototype controller under different working loads and conditions conveniently and safely. Compared with numerical simulation, HIL simulation is more reliable and credible because numerical simulation is often operated in ideal circumstances, without considering noise, disturbance, and some practical problems often ignored which might result in fatal failure. HIL simulation has the advantages of reducing the risk, shortening the developing time, and being well suitable for critical and hazardous applications.

The concept of HIL simulation is described by Fig. 3. A typical HIL simulation system for UAV autopilot design is composed of a stand-alone embedded real-time control system, a personal computer, a servo unit, and a host PC. The hardware arrangement of HIL simulation system is shown in Fig. 4. The graphical programming language LabVIEW is the software development environment for data acquisition (DAQ) and instrument control in performing HIL simulation.

3.1 Embedded real-time control system (prototype controller)

The prototype controller is implemented by the National Instruments (NI) PXI real-time embedded control system which includes:

Real-time Controller NI PXI-8184 RT: It is a stand-alone embedded real-time control system. This system real-time computes the control output according to the implemented control law and the error between command signal and feedback signal from sensor.

Multifunction DAQ device NI PXI-6259: It takes the responsibility on (1) acquiring the analog feedback signal from the plant PC, (2) sending the analog control signal to the plant PC and the host PC, (3) receiving the digital signal from the host PC to start/stop the PXI system.

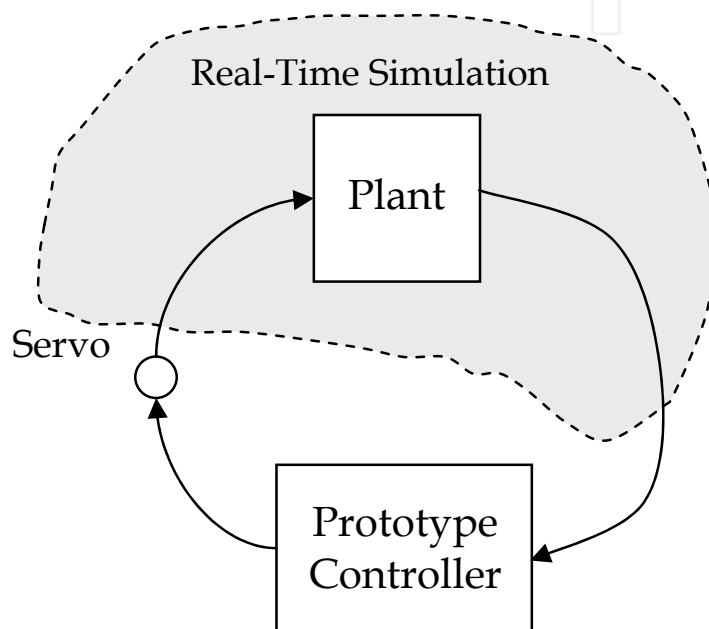


Fig. 3. The concept of HIL simulation

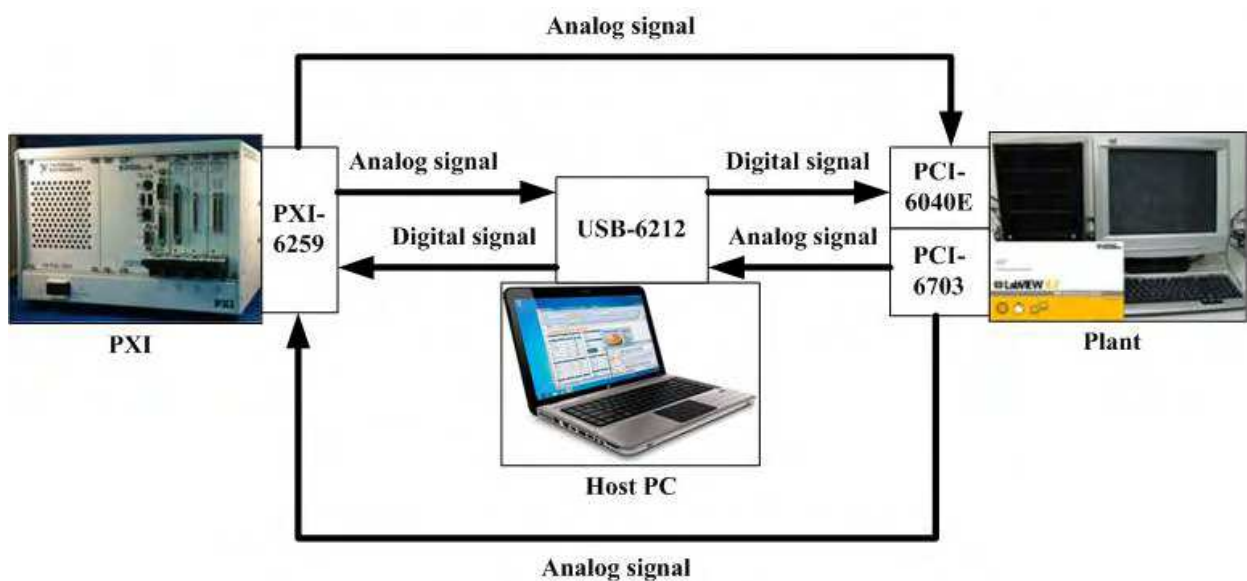


Fig. 4. The hardware layout of a typical HILS system

3.2 Personal computer (Plant)

The personal computer (PC) is used to simulate the dynamics of plant (UAV). The major specifications are Intel Pentium 4-3.0 GHz CPU and 1 GB SDRAM. The I/O interfaces include:

Multifunction DAQ device NI PCI-6040E: It takes the responsibility on (1) acquiring the analog control signal from the PXI system or from the potentiometer on the servo unit that represents the control surface angle, (2) receiving the digital signal from the host PC to start/stop the plant PC.

Multifunction DAQ device NI PCI-6703: It takes the responsibility on sending the analog state signal to the PXI system and the host PC.

3.3 Servo unit

Each servo unit contains two sets of servos, Futaba S3001, which is used to actuate the control surface in UAV to generate the corrective torque in order to keep the desired attitude. The rotational angle of servo is measured by potentiometer, model no. J50S, manufactured by Copal Electronics Co., Ltd. in Japan.

3.4 Host PC

The host PC or notebook is used to monitor, synchronously digitally trigger, and display the virtual aircraft instruments of UAV. The major specifications are Intel Core 2 Duo P8400-2.26 GHz CPU and 2 GB SDRAM. The I/O interfaces include:

Multifunction DAQ device NI USB-6212: It takes the responsibility on (1) acquiring the analog state signal from the plant PC, (2) acquiring the analog control signal from the PXI system, (3) and sending the digital signal to synchronously trigger the start/stop action of the plant PC and PXI system.

3.5 Software development environment LabVIEW

To develop a HIL simulation system is not an easy work. The graphical programming software LabVIEW streamlines the system building with a convenient environment to integrate hardware and software seamlessly. LabVIEW is a graphical programming language that has been widely adopted throughout industry and academia as the standard for data acquisition and instrument control software. While other text-based languages create lines of codes, LabVIEW provides an intuitive graphical programming style to create programs in a pictorial form called a block diagram. Graphical programming allows users to focus on flow of data within the applications that makes programming easy and efficiency.

A LabVIEW program, called virtual instrument (VI), has two main parts: a front panel and a block diagram. The front panel is the interactive user interface of a VI that the appearance and operation often imitates actual physical instruments. The block diagram is the VI's source code and is the actual executable program. The components of a block diagram are lower-level VIs, functions, constants, and program execution control structures. In addition, LabVIEW can command plug-in DAQ devices to acquire or generate analog and digital signals. You might use DAQ devices and LabVIEW to hook your computer up to the real world, for example, to take measurements, talk to an instrument, send data to another computer. It is very convenient to construct HIL simulation systems using LabVIEW. A lot of valuable books (Larsen, 2011; Travis & Kring, 2007) provide more foundations, programming skills, and applications for LabVIEW.

4. Stability augmentation system

In order to perform missions, the UAV has to be hold on or maneuvered to a specified cruising speed, altitude, and attitude. A typical feedback system providing desirable handling qualities for pilot/autopilot commands is called stability augmentation system (SAS). Especially for an aircraft flying throughout an extended flight envelope, the stability derivatives in Eq. (3) and (4) are expected to vary significantly. Because of the variation of stability derivatives, the handling qualities also are going to change. SAS can be designed to improve the handling qualities over its entire operational envelope (Nelson, 1998; Roskam, 2003; Kayton & Fried, 1969).

The stability augmentation system (SAS) is the inner loop of the flight control system (FCS) that provides the desirable handling characteristics for pilots. The design parameter in the SAS is the feedback gain that is determined by applying the root locus technique. The root locus technique is a simple and powerful tool in classical control theory for determining, by graphical plot, the detailed information about the stability and performance of a closed-loop system knowing only the open-loop transfer function. The root locus plot shows the poles of the closed-loop system in complex plane for every value of the feedback gain. The location of the closed-loop system poles determine two important quantities, damping ratio and natural frequency, that are closely related to the time-domain performance specifications (Cook, 2007; Nise, 2008). According to the linear mathematical models of longitudinal dynamics for the MP2000UAV, the SAS for pitch attitude control are at first designed by the root locus technique in MATLAB and then are tested in HIL simulation.

4.1 Design of SAS

The pitch angle, one of the state variables in Eq. (3), is an appropriate output variable for attitude control. Therefore the pitch angle is used as a feedback in SAS; it can be measured by vertical gyro or attitude and heading reference system (AHRS).

The negative pitch-attitude-to-elevator transfer function from Eq. (2) for the MP2000UAV is:

$$\frac{-\theta(s)}{\delta_e(s)} = \frac{1.699(s + 4.7308)(s + 13.7827)}{(s + 0.0106)(s + 8.854)(s + 3.917 \pm j2.4488)} \quad (7)$$

The open-loop system has two real poles at -0.0106 , -8.854 , and a pair of stable complex poles at $-3.917 \pm j2.4488$. The corresponding damping ratios and natural frequencies are given in Table 2.

Poles	Natural frequency	Damping ratio
-0.0106	0.0106	1
-3.917+j2.4488	4.6195	0.8479
-3.917-j2.4488	4.6195	0.8479
-8.854	8.8540	1

Table 2. The natural frequency and damping ratio for open-loop poles

From the step response as shown in Fig. 5, it is obviously that the settling time is too long, approximately 300 sec. As a result, the maneuverability is very poor. It is necessary to design a SAS to enhance the handling quality. The block diagram for pitch-angle to elevator feedback of SAS is shown in Fig. 6. The elevator deflection is produced in proportion to the pitch angle and adding it to the pilot's control input as:

$$\delta_e = \delta_{e_pilot} + K\theta \tag{8}$$

where δ_{e_pilot} is that part of the elevator deflection created by the pilot. The gain K is the design parameter.

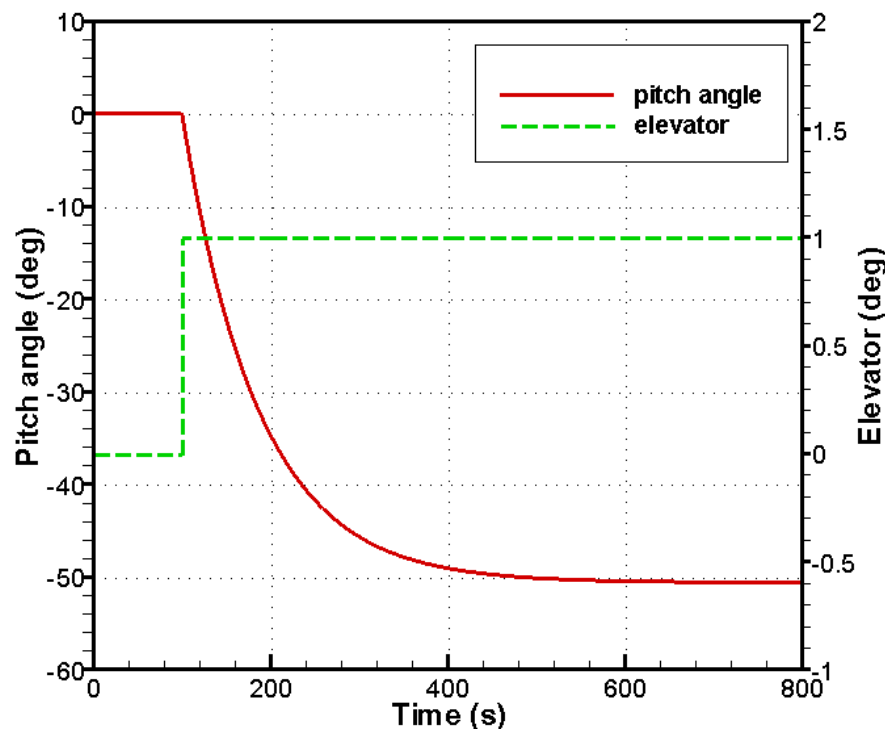


Fig. 5. Step response of open-loop system

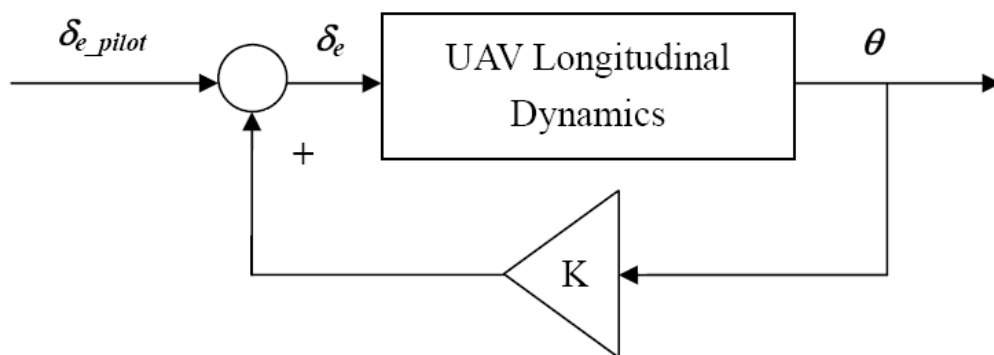


Fig. 6. The block diagram of SAS for pitch angle control

From classical control theory, the system response is determined by the pole location in complex plane. The time-domain performance specifications such as peak time, percent overshoot and settling time are functions of damping ratio and natural frequency of the dominant poles. The duty of SAS is, therefore, to modify the damping ratio and natural frequency by adjusting the value of feedback gain K so that the UAV is easier to control. Studies have shown that to obtain a desirable transient performance such as a smaller overshoot and a shorter settling time, the design specifications for damping ratio is approximately 0.6-0.7 and natural frequency is larger than 2 rad/s.

The root locus technique permits the designer to view the trajectories of the close-loop system poles as the design parameter (feedback gain K) is varied. It is very convenient to determine the value of K in SAS by applying the root locus technique. The root locus plot constructed by using MATLAB for the pitch-attitude-to-elevator transfer function, Eq. (7), is shown in Fig. 7.

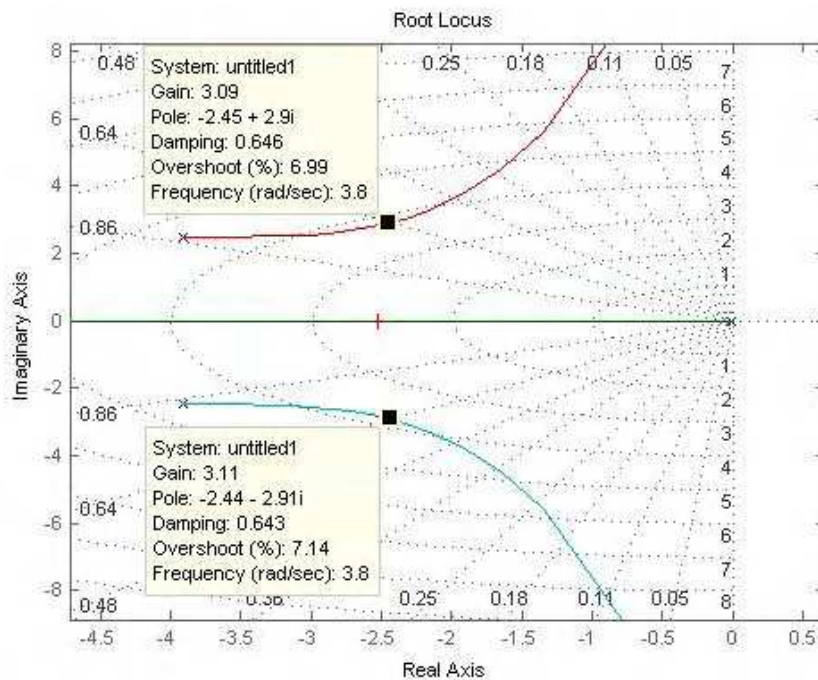


Fig. 7. Root locus plot for SAS

It is noted that at $K = 3$ the damping ratio and natural frequency of the dominant poles meets the specifications while the damping ratio is 0.6564 and natural frequency is 3.7915. All the closed-loop system poles and the corresponding damping ratios and natural frequencies are given in Table 3.

Poles	Natural frequency	Damping ratio
-2.5306	-2.5306	1
-2.489+j2.86	3.7915	0.6564
-2.489-j2.86	3.7915	0.6564
-9.1904	9.1904	1

Table 3. The natural frequency and damping ratio for closed-loop poles

The dc gain (steady-state gain) of closed-loop system is -0.3313. It indicates that at steady state the ratio of pitch angle to pilot’s control input is approximately one-third:

$$\frac{\theta(\infty)}{\delta_{e_pilot}(\infty)} \approx -\frac{1}{3} \tag{9}$$

In Fig. 8 the transient response shows a significant improvement that the settling time is greatly reduced to 1.2 sec by employing SAS. At this stage, the performance of SAS is verified by computer simulation that the desired handling quality for pilot command is achieved.

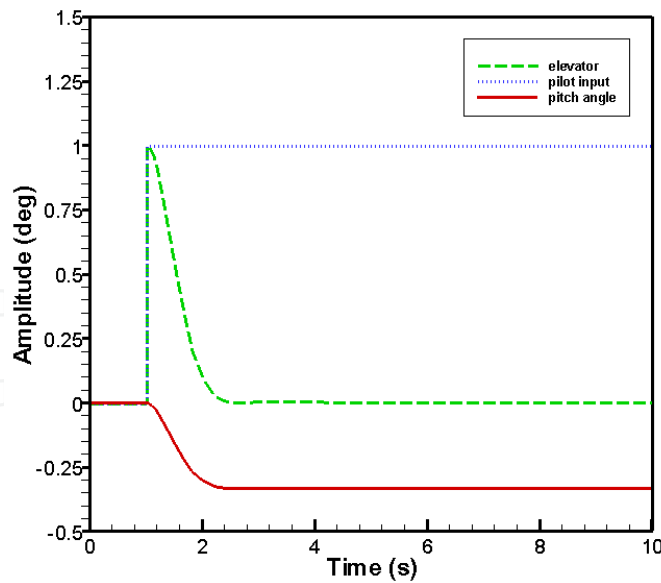


Fig. 8. Step response of closed-loop system

4.2 HIL simulation results

In this section the prototype SAS is realized and implemented by the PXI real-time control system, and the performance of SAS is examined by HILS experiment in real time and real signal.

4.2.1 Pilot input on time-table experiment

The pilot control input according to schedule is defined as:

$$\delta_{e_pilot}(t) = \begin{cases} 0, & 0 \leq t < 1 \\ 5, & 1 \leq t < 2 \\ 0, & t \geq 2 \end{cases} \quad (10)$$

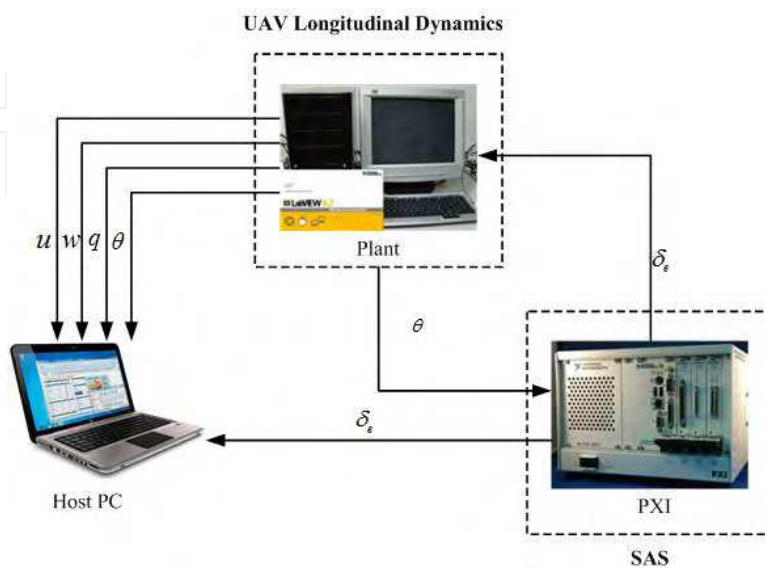


Fig. 9. The signal flow diagram of HIL simulation

The signal flow diagram of HILS system is shown in Fig. 9. The LabVIEW virtual instruments of the controller, plant, and host PC used to perform HILS experiment are presented in Figs. 10-14. The sampling rate is 50 Hz.

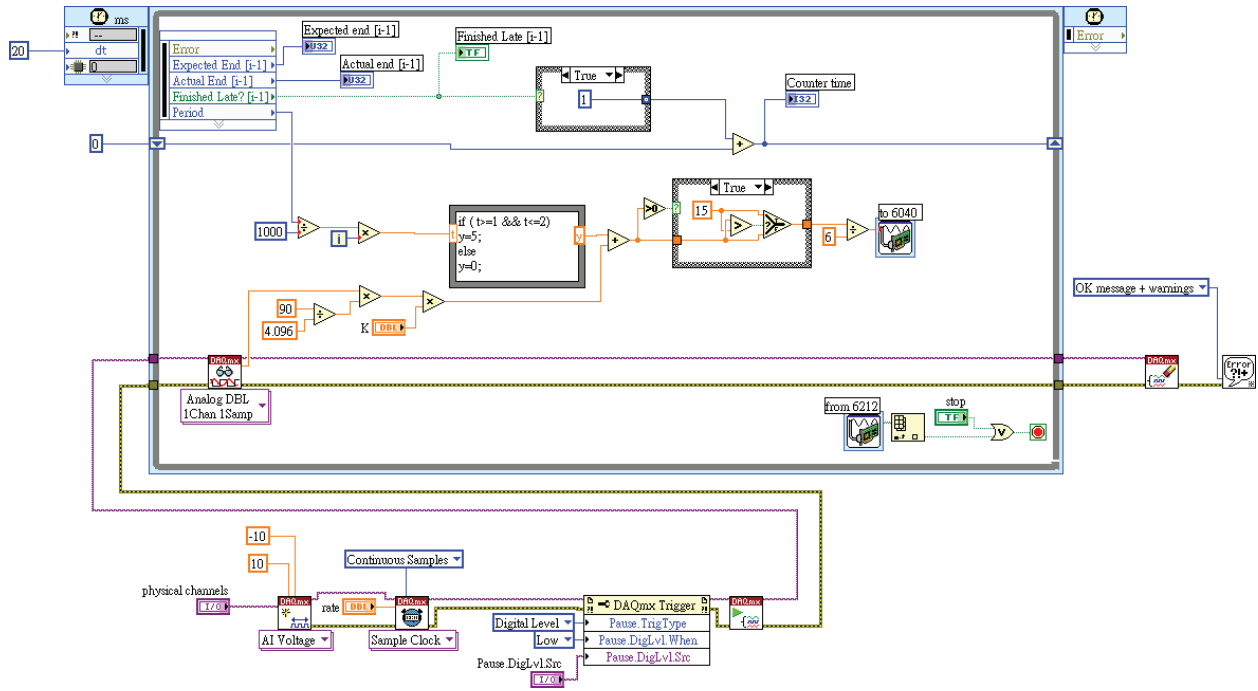


Fig. 10. The LabVIEW block diagram of controller.

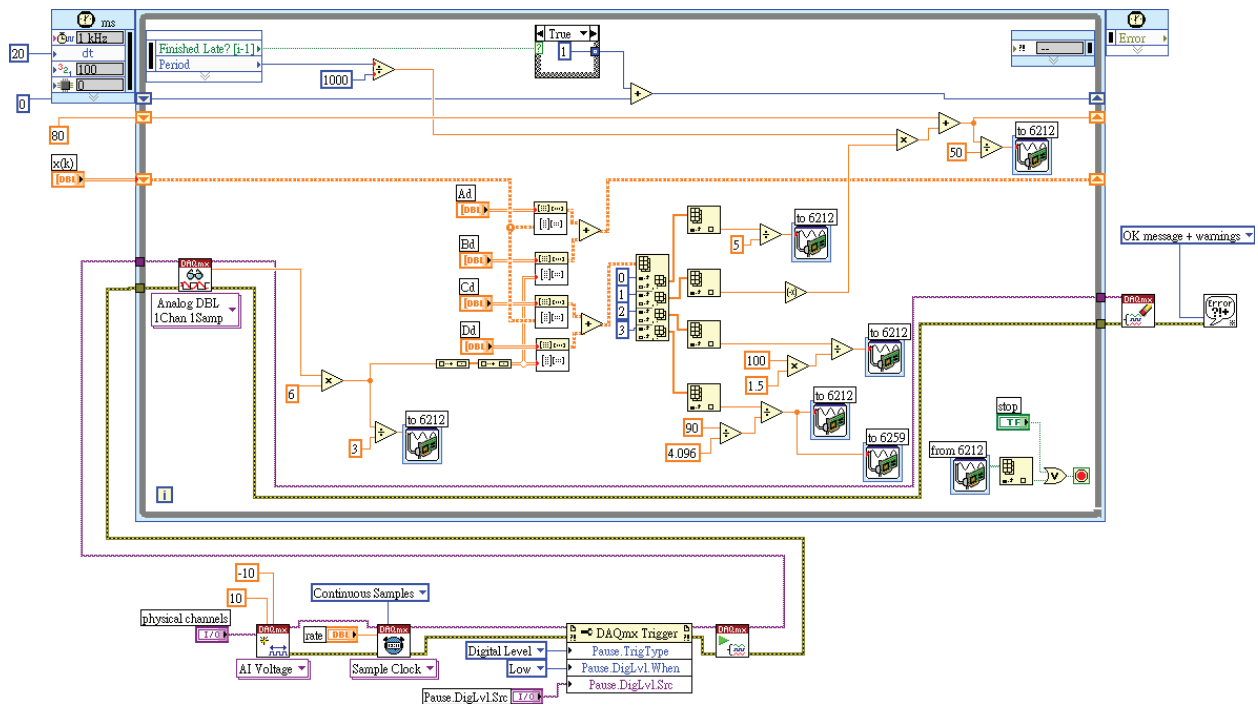


Fig. 11. The LabVIEW block diagram of plant

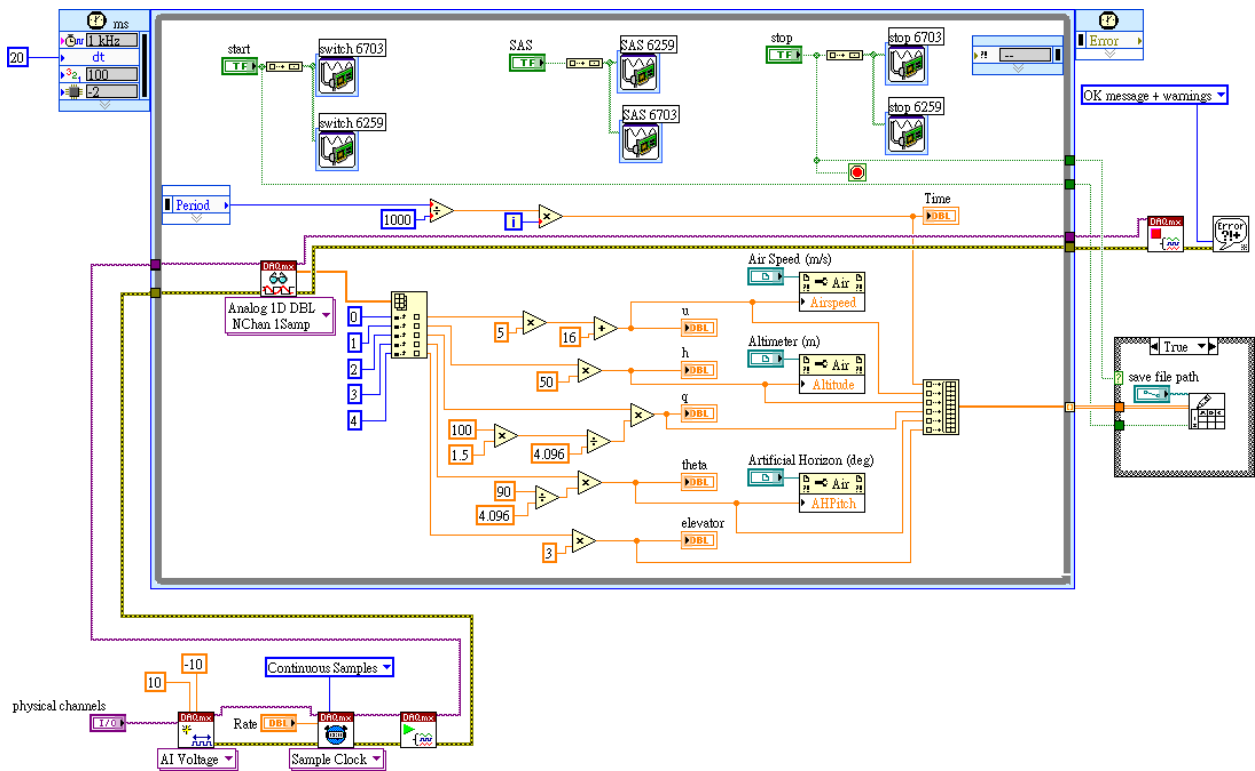


Fig. 12. The LabVIEW block diagram of host PC

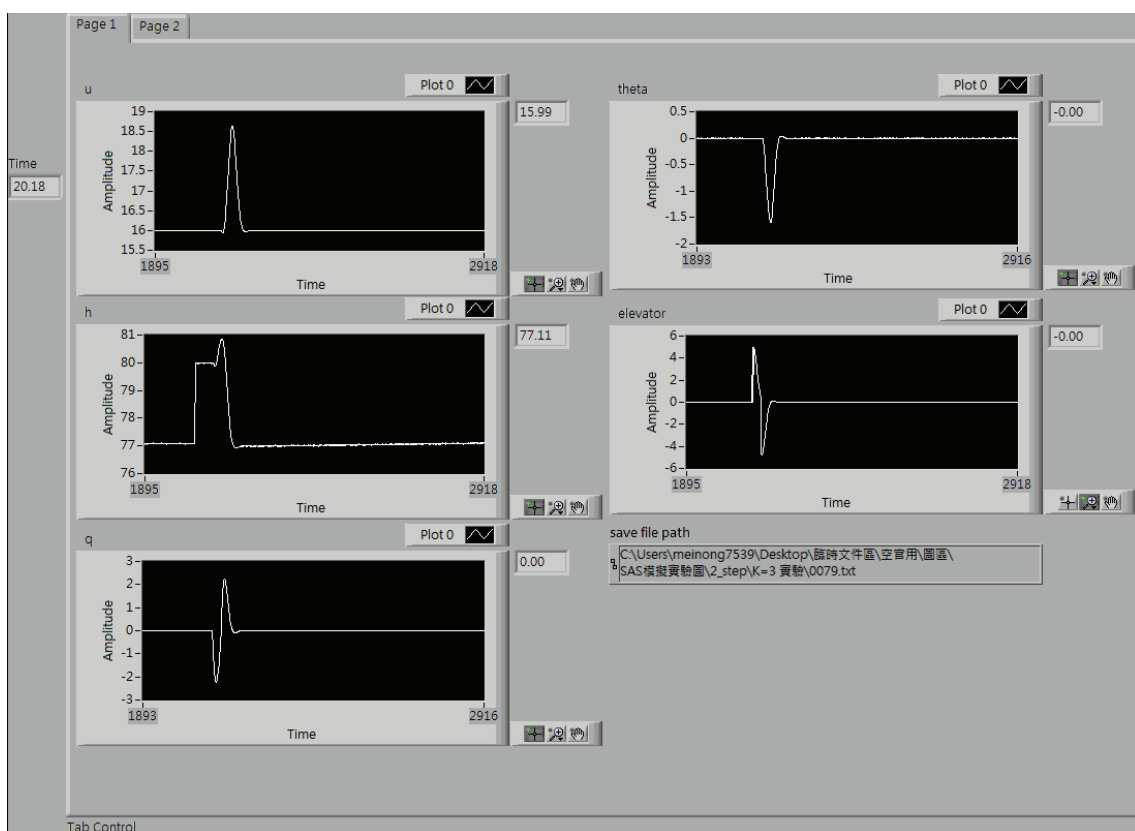


Fig. 13. The LabVIEW front panel of host PC (flight data)



Fig. 14. The LabVIEW front panel of host PC (aircraft instruments)

4.2.2 Pilot-in-the-loop experiment

The pilot-in-the-loop (PIL) experiment is developed to test the performance of SAS when the human pilot joins in the control loop. The pilot's task is to control UAV pitch attitude to a desired angle. The pilot looks at the virtual aircraft instruments displayed in the host PC screen and tries to correct the error between the desired pitch angle and the actual pitch angle. This process of pilot in control can be expressed as follows: (1) the pilot's eyes watch the pitch angle on the screen, (2) the pilot's brain figures out the magnitude of the error, (3) the pilot's brain sends a signal to actuate the hand of pilot, (4) the pilot's hand manipulates the control stick to move the elevator, (5) the deflection of elevator changes the pitch angle of the UAV. The operation of PIL experiment depicted in Fig 15 describes a compensatory control situation: the pilot tries to maintain the desired pitch attitude by driving the pitch angle error to zero with the assistance of SAS.

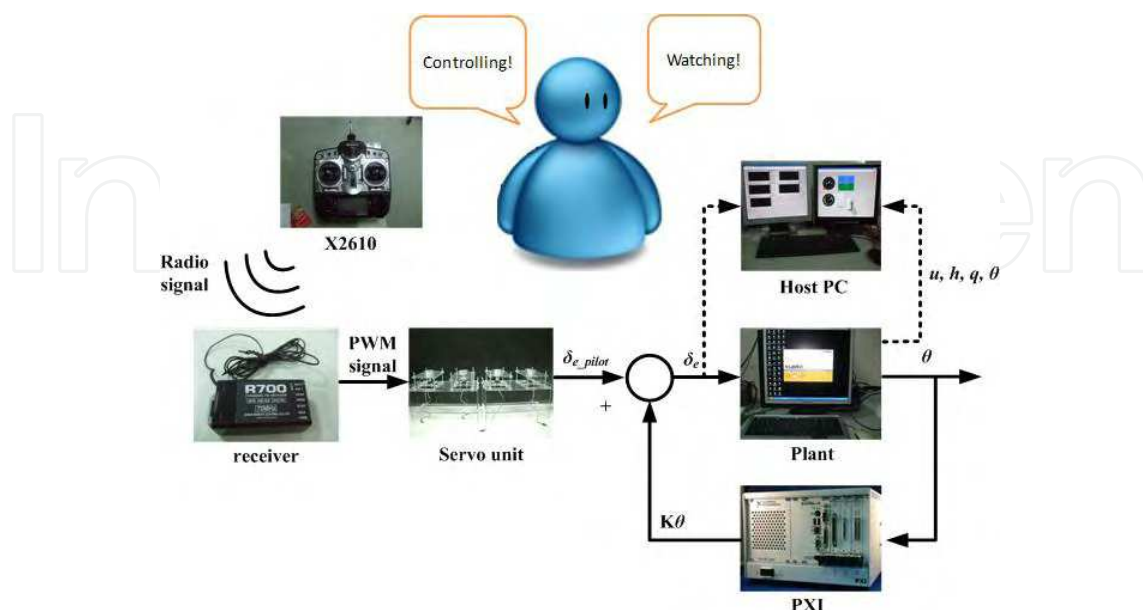


Fig. 15. The operation of pilot-in-the-loop experiment

From the HIL simulation results of PIL experiment in Fig. 16-17, pilot control using SAS demonstrates an excellent handling performance than that without using SAS. The effect of SAS on improving the handling qualities of UAV is confirmed.

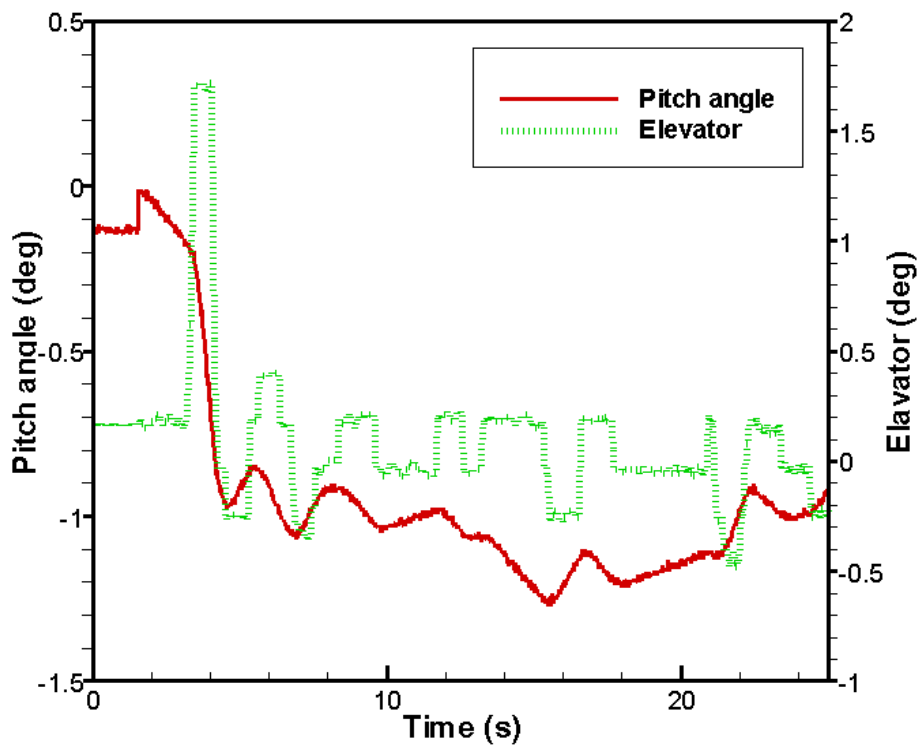


Fig. 16. PIL experiment without SAS

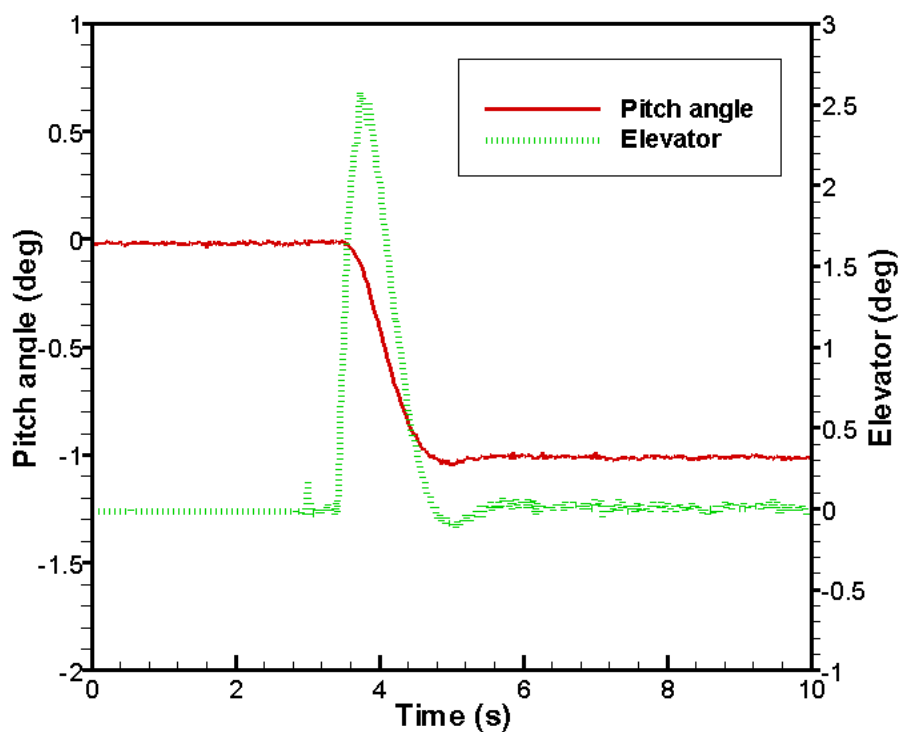


Fig. 17. PIL experiment with SAS

5. Autopilot system

The function of stability augmentation system is to improve the flying qualities of an airplane for pilot manual control. To lower pilot workload, particularly on long-range flights or out-of-sight flights of UAV, most airplanes or UAVs are equipped with automatic flight control systems or autopilots. The basic autopilot modes include pitch attitude hold mode, airspeed hold mode, bank angle (roll attitude) hold mode and heading angle hold mode.

Normally pitch attitude is controlled by the elevator, airspeed is controlled by the engine throttle, roll attitude is controlled by the aileron, and heading is controlled by the rudder. Thus there are four feedback loops to achieve four different autopilot modes. Fig. 18 shows the block diagram of pitch attitude hold mode in HILS experiment.

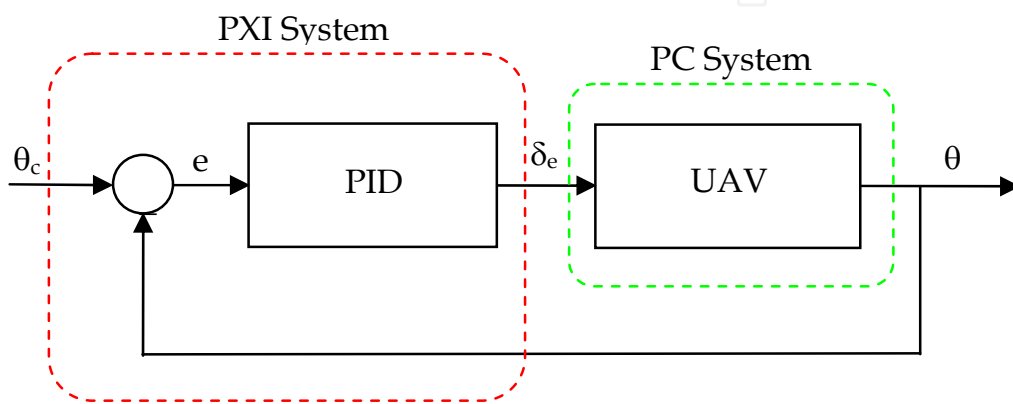


Fig. 18. The block diagram of pitch attitude control for autopilot in HILS experiment

5.1 PID controller design

The proportional-integral-derivative (PID) controllers are frequently used in practical control systems owing to their simple structures and quite clear physical senses. Usually the PID controller is described by the transfer function:

$$K_{PID}(s) = K_p + \frac{K_i}{s} + K_d s \quad (11)$$

where K_p , K_i , and K_d are gains to be determined to meet design requirements. Specifically, the PID controller can be expressed by the form in time domain as:

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (12)$$

where u is the controller output, e is the error between desired and actual output, T_i is integral time, and T_d is derivative time. Three important characteristics of PID controller are described as follows: it provides feedback; it eliminates steady-state error through integral action; it improves transient response by anticipating the future through derivative action. In order to meet the design specifications on transient and steady-state response, PID control is an ideal choice for autopilot design.

The PID controllers are designed by two phases. At first, the coarse PID gains are obtained according to the well-known Ziegler Nichols tuning rules that provide an acceptable closed-

loop response. Next, let the coarse PID gains be the initial guess, the Nonlinear Control System Toolset in MATLAB/Simulink is applied to optimally determine the fine PID gains to meet time domain specifications such as rise time, overshoot, settling time, steady state error, and actuator constrain. The resulting PID gains in four feedback loops of autopilot are listed in Table 4.

	K_p	T_i	T_d
Pitch Altitude Loop	-11.99	0.001118	0.005657
Velocity Loop	0.07506	0.03649	0.004805
Roll Attitude Loop	0.2571	0.003650	0.01531
Heading Loop	-1.9058	0.07574	-0.04379

Table 4. The PID Gains in Four Feedback Control Loops of UAV Autopilot.

5.2 HIL simulation results

In this section the prototype PID controller for each mode of autopilot is implemented by the PXI real-time control system, and the performance is explored in HIL simulation. The hardware arrangement of HIL simulation system is shown in Fig. 19. Figs. 20 and 21 represent the LabVIEW front panel and block diagram of UAV plant.

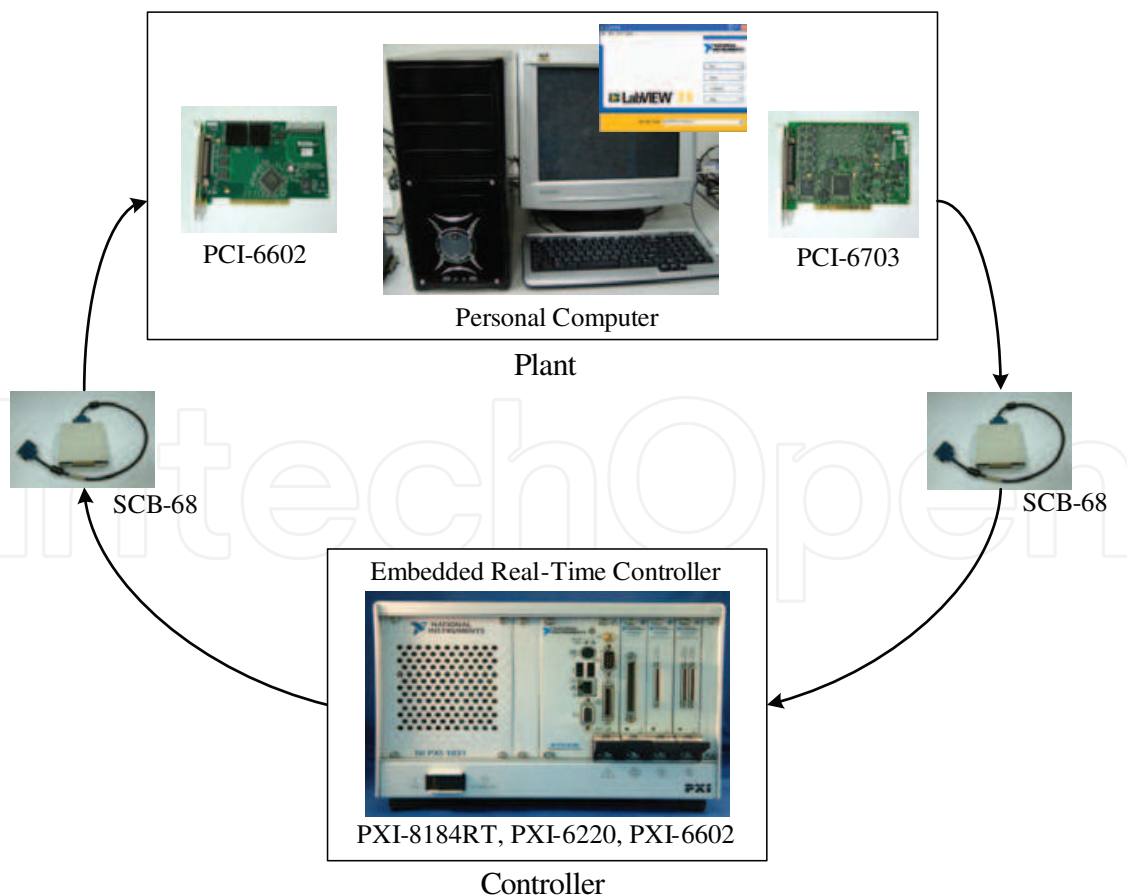


Fig. 19. The Hardware Setup of HIL simulation for UAV autopilot

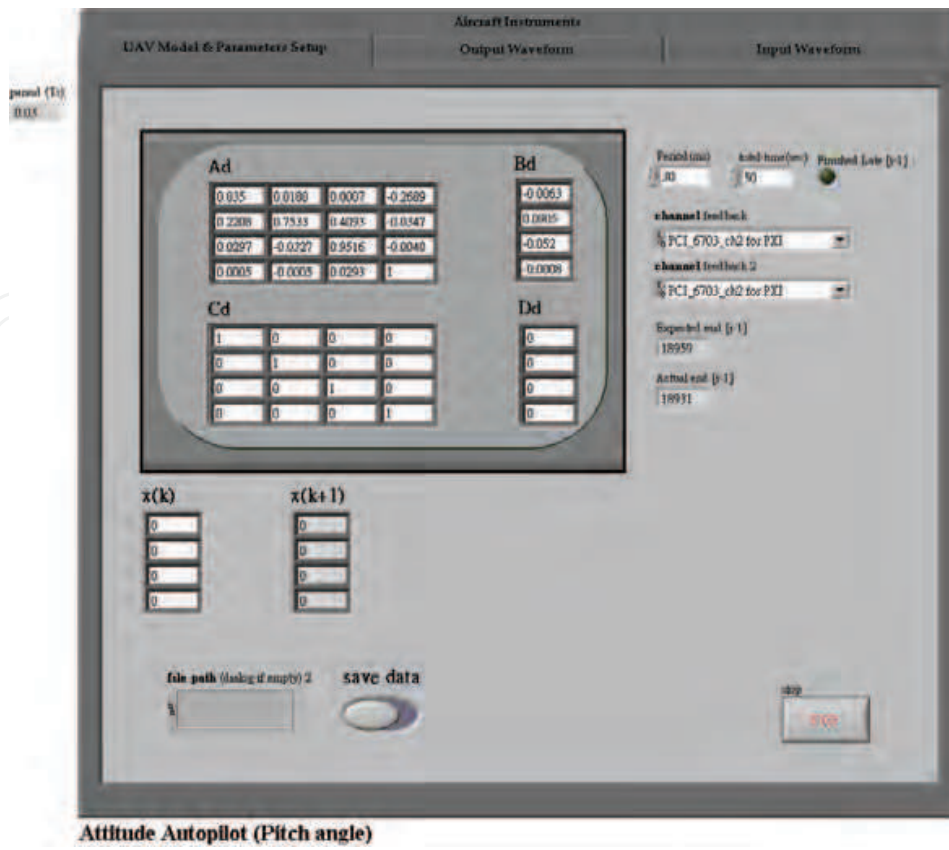


Fig. 20. LabVIEW front panel of UAV plant

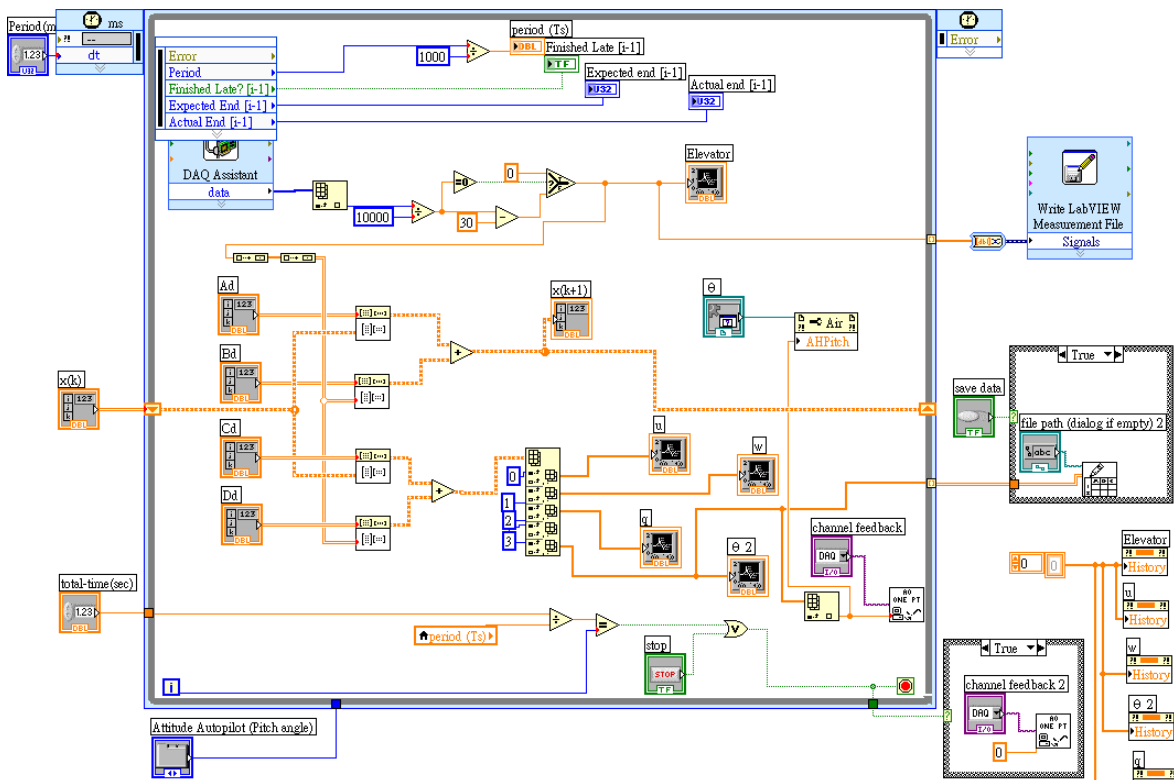


Fig. 21. LabVIEW block diagram of UAV plant

The following LabVIEW front panel and block diagram windows shown in Fig. 22 represent the prototype PID controller in pitch attitude hold mode of autopilot. It is noticed that the values of PID gains in front panel are from Table 4 where the integral time T_i and the derivative time T_d are in minute available for LabVIEW PID function.

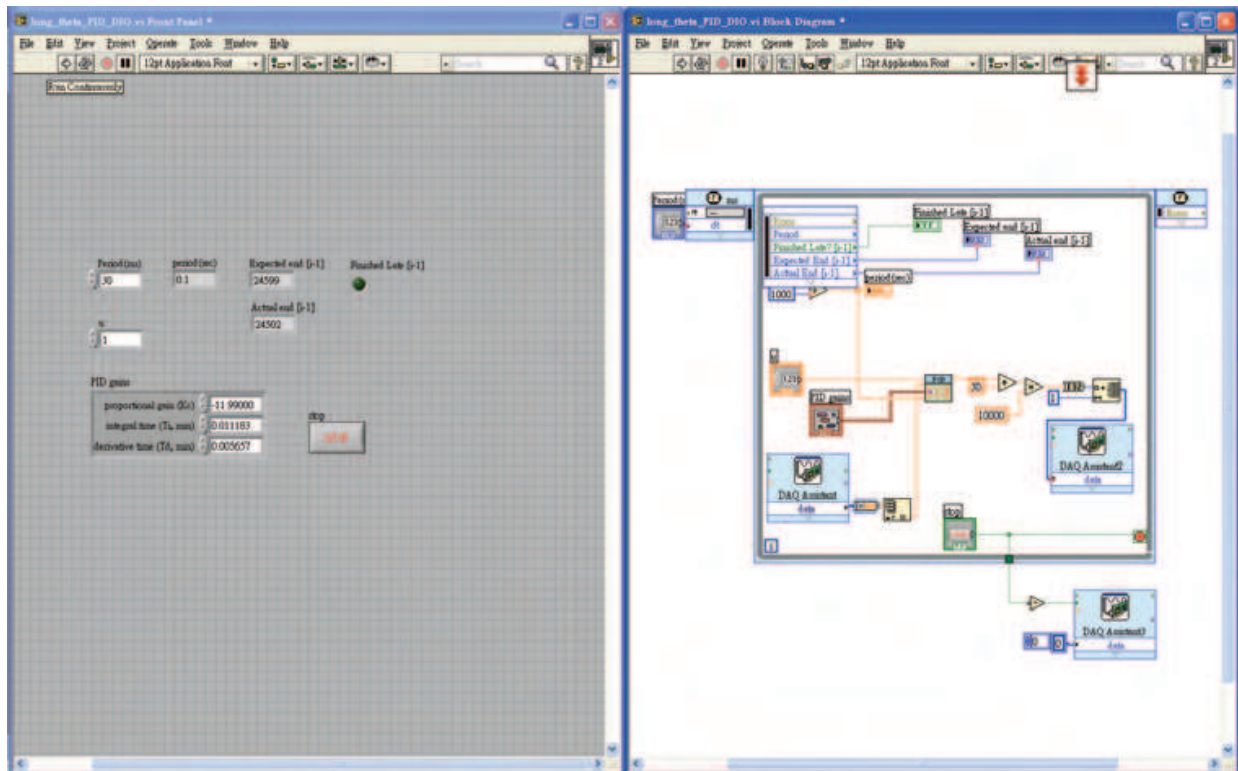


Fig. 22. LabVIEW front panel and block diagram window of PID controller

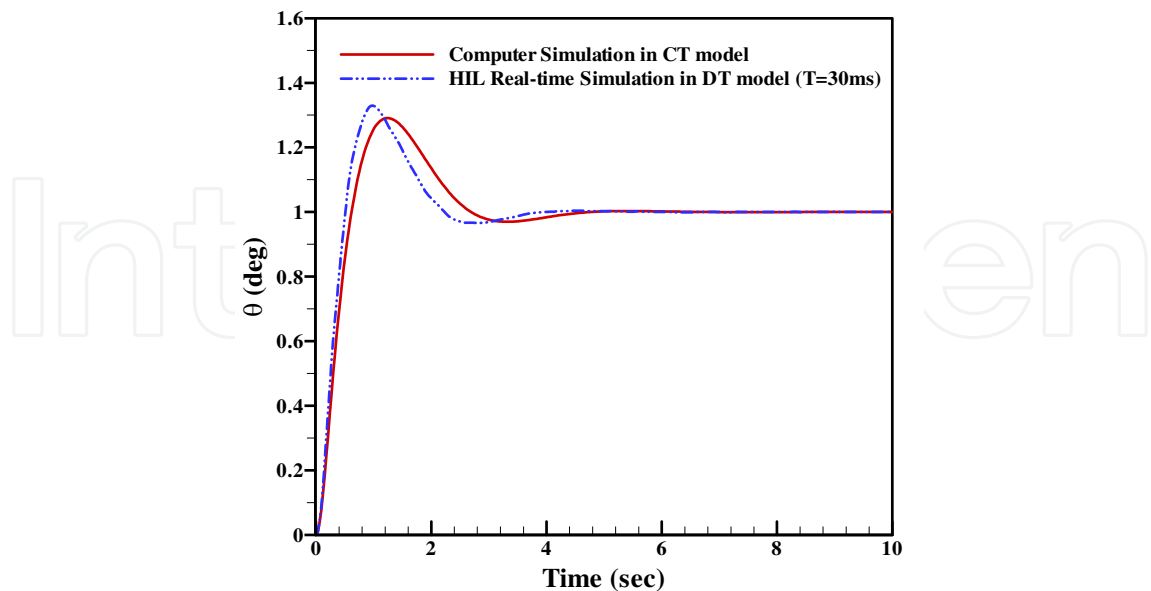


Fig. 23. The unit-step response of PID controller in pitch attitude hold mode

Figs. 23-26 show the closed loop unit-step time response of PID controllers in pitch attitude hold mode, velocity hold mode, bank angle hold mode and heading angle hold mode for

UAV autopilot. Apparently, the results of HIL simulation and computer simulation are very close to each other. Fig. 23 exhibits an underdamped response in pitch control with 4 sec. settling time, 30% overshoot, and no steady-state error; Fig. 24 also exhibits an underdamped response (a little increase in damping ratio) in velocity control with 6 sec. settling time, 35% overshoot, and no steady-state error; Fig. 25 also exhibits an underdamped response (a bigger time constant) in roll control with over 25 sec. settling time, 5% overshoot, and still no steady-state error; Fig. 26 exhibits an overdamped response in heading control with 14 sec. settling time, no overshoot, and no steady-state error. From the results in HIL simulation, PID controllers demonstrate very good performance.

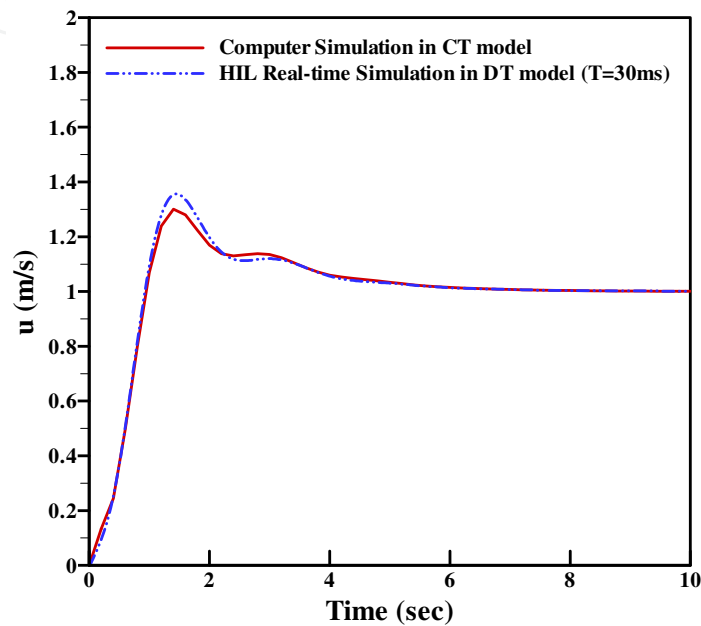


Fig. 24. The unit-step response of PID controller in airspeed hold mode

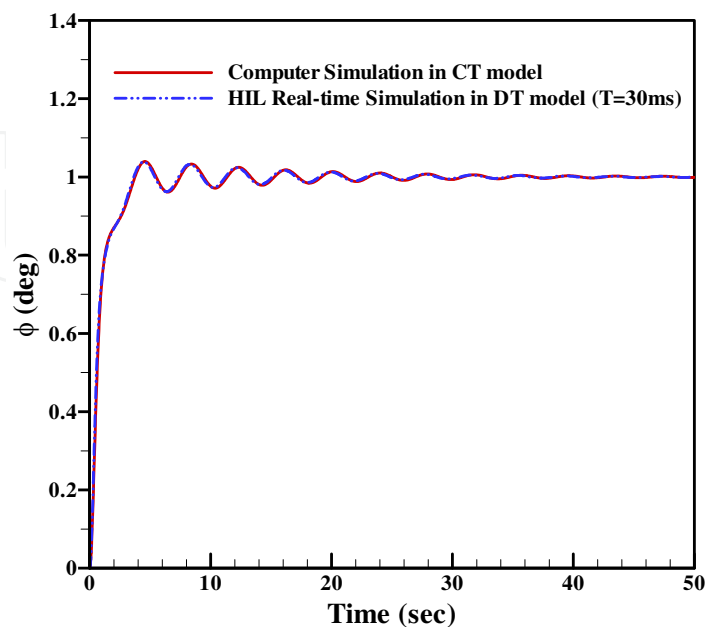


Fig. 25. The unit-step response of PID controller in bank angle hold mode

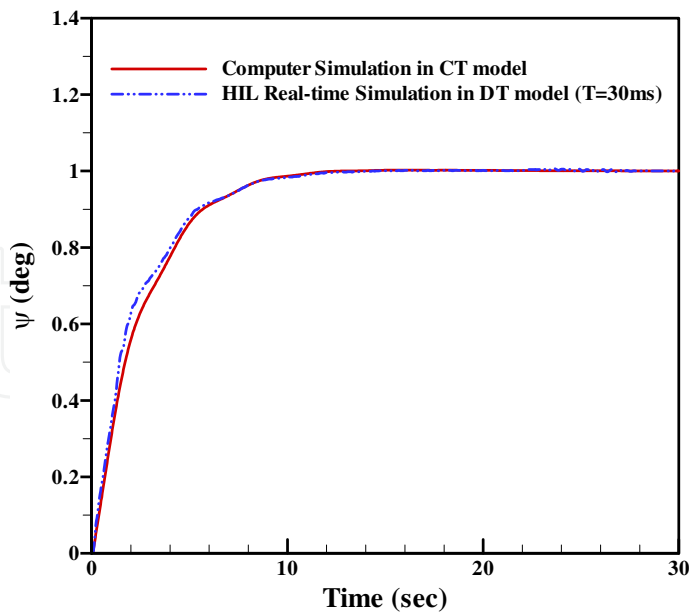


Fig. 26. The unit-step response of PID controller in heading angle hold mode

5.3 HIL simulation system including a servo unit

In this subsection the HIL simulation system with servo unit is taken into consideration. The servo unit is described in section 3.3. The HIL simulation system with servo unit is composed of three major parts: a Pentium 4 desktop personal computer (PC) system, a National Instrument (NI) real-time PXI system, and a servo unit. The hardware arrangement is shown in Fig. 27.

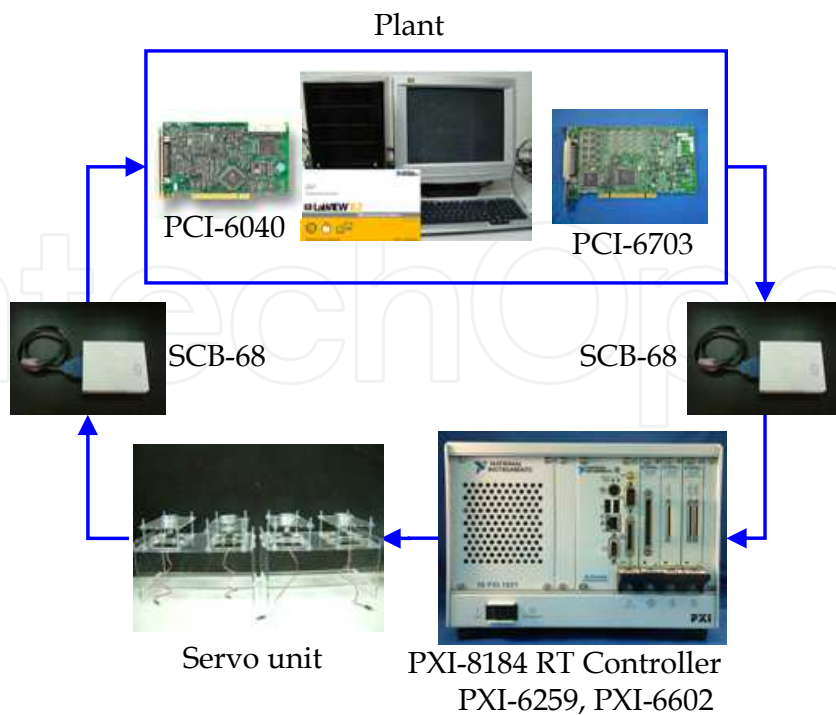


Fig. 27. The hardware arrangement of HIL simulation system with a servo unit

After receiving feedback signals by DAQ PXI-6259 from plant (PC system) that represent UAV actual states, the real-time PXI system carries out the proportional-integral-derivative (PID) algorithm, computes the control effort (control surface deflection angle and throttle setting) for UAV flight control, and then generates pulse-width-modulation (PWM) signals by DAQ PXI-6602 to control a real servo. Each servo unit consists of two Futuba-S3001 servos and two accurate potentiometers. The servo receives PWM signals from PXI system and rotates to a specific angle. The resulting angle is measured by a potentiometer and fed back to the plant through DAQ PCI-6040. Finally the PC system computes the dynamical states of UAV based on the state-space model and outputs these analog signals to the PXI system by DAQ PCI-6703. As a whole the PC system, PXI system, and servo unit constitute a real-time closed-loop control system for UAV autopilot HIL simulation.

The HIL simulation results of PID controller for system including servo is denoted by the solid line in Fig. 28. Apparently the performance becomes a little worse because the real servo and potentiometer involved in HILS system result in so called unmodelled dynamics that is not taken into consideration in controller design. This deterioration of controller performance is revealed by HIL simulation not by numerical simulation. The PID controller has to be tuned to obtain an acceptable performance. It clearly demonstrates that HIL simulation is indispensable to controller design and verification.

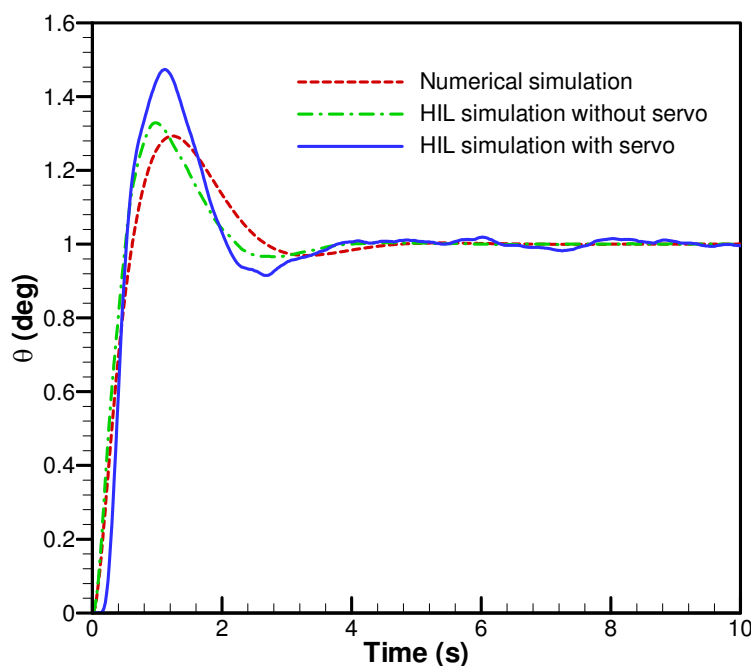


Fig. 28. The performance of PID controller for HIL simulation system including servo

6. Conclusion

With the growing importance of autonomous vehicles for industry, science, aerospace and defense applications, engineers are encouraged to use HIL simulation methodology to shorten development cycle, lower total cost and improve functional performance of the vehicles. LabVIEW features an easy-to-use graphical programming environment and an intuitive data flow programming style that makes the work of HIL simulation to be easier and more efficiency. This chapter not only provides LabVIEW solutions to HIL simulation

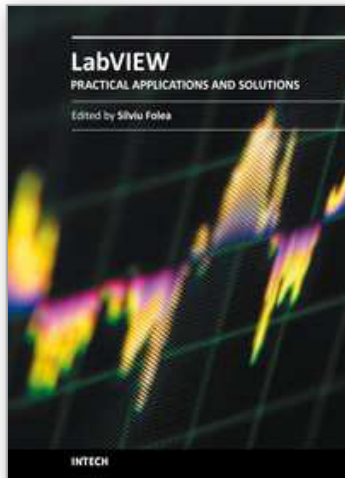
but also presents a complete analysis, design and HILS verification of UAV stability augmentation system and autopilot.

7. References

- Bullock, D., Johnson, B., Wells, R. B., Kyte, M. & Li, Z. (2004). Hardware-in-the-Loop Simulation. *Transportation Research Part C: Emerging Technologies*, Vol. 12, No. 1, (February 2004), pp. 73-89, ISSN 0968-090X
- Bryson, A. E., Jr. (1994). *Control of Spacecraft and Aircraft*, Princeton University Press, ISBN 0-691-08782-2, Princeton, New Jersey, USA
- Carrijo, D. S., Oliva, A. P. & W. de Castro Leite Filho (2002). Hardware-in-the-Loop Simulation development. *International Journal of Modeling and Simulation*, Vol. 22, No. 3, pp. 167-175, (July 2002), ISSN 0228-6203
- Cole, D. T., Sukkarieh, S. & Goktogan, A. H. (2006). System Development and Demonstration of a UAV Control Architecture for Information Gathering Missions. *Journal of Field Robotics*, Vol. 26, No. 6-7, (June-July 2006), pp. 417-440, ISSN 1556-4967
- Cook, M. V. (2007). *Flight Dynamics Principles* (2nd Ed.), Elsevier, ISBN 978-0-7506-6927-6, Oxford, UK
- Cosic, K., Kopriva, I., Kostic, T., Samic, M. & Volareic, M. (1999), Design and Implementation of a Hardware-in-the-Loop Simulator for a Semi-Automatic Guided Missile System. *Simulation Practice and Theory*, Vol. 7, No. 2, (April 1999), pp. 107-123, ISSN 1569-190X
- Kayton, M. & Fried, W. R. (Editors). (1969). *Avionics Navigation Systems*, John Wiley & Sons, ISBN 471-46180-6, New York, USA
- Larson, R. W. (2011). *LabVIEW for Engineers*, Prentice-Hall, ISBN-13 978-0-13-609429-6, New Jersey, USA
- Ledin, J. (2001). *Simulation Engineering*, CMP Books, ISBN 157-820-0806, Lawrence, USA
- Nelson, R. C. (1998). *Flight Stability and Automatic Control* (2nd Ed.), McGraw-Hill, ISBN 0-07-115838-3, Boston, Massachusetts, USA
- Nise, N. S. (2008). *Control Systems Engineering* (5th Ed.), John Wiley & Sons, ISBN 978-0-470-16997-1, New Jersey, USA
- Roskam, J. (2007). *Airplane Flight Dynamics and Automatic Flight Controls, Part I*, DARcorporation, ISBN-13 978-1-884885-17-4, Kansas, USA
- Roskam, J. (2003). *Airplane Flight Dynamics and Automatic Flight Controls, Part II*, DARcorporation, ISBN 1-884885-18-7, Kansas, USA
- Salman, S. A., Puttige, V. R. & Anavatti, S. G. (2006). Real-Time Validation and Comparison of Fuzzy Identification and State-Space Identification for a UAV Platform. *Proceedings of the 2006 IEEE International Conference on Control Applications*, pp. 2138-2143, ISBN 0-7803-9797-5, Munich, Germany, October 4-6, 2006
- Shetty, D. & Kolk, R. A. (1997). *Mechatronics System Design*, PWS Publishing Company, ISBN 053-495-2852, Boston, USA
- Sun, Y. P., Wu, L. T. & Liang, Y. C. (2006). System Identification of Unmanned Air Vehicle and Autopilot Verification via Hardware-in-the-Loop Real-time Simulation. *Proceedings of International Forum on Systems and Mechatronics*, ISBN 986-688-904-1, Tainan, Taiwan, December 6-8, 2006

- Sun, Y. P., Chu, C. Y., & Liang, Y. C. (2008a). Using Virtual Instruments to Develop an Actuator-Based Hardware-in-the-Loop Test-Bed for Autopilot of Unmanned Aerial Vehicle. *Proceedings of SPIE-Fourth International Symposium on Precision Mechanical Measurements*, Vol. 7130, Part I, pp. 71301J-1-6, ISBN 9780819473646, Anhui, China, August 25-29, 2008
- Sun, Y. P., Wu, L. T. & Liang, Y. C. (2008b). Stability Derivatives Estimation of Unmanned Aerial Vehicle. *Key Engineering Materials*, Vol. 381-382, pp. 137-140, ISSN 1013-9826
- Sun, Y. P., Tsai, C. H. & Liang, Y. C. (2009). Fuzzy Logic Control Design and Verification of Unmanned Aerial Vehicle Autopilot via Hardware-in-the-Loop Simulation. *Proceedings of the 2009 International Symposium on Mechatronic and Biomedical Engineering and Applications*, pp. 156-164, ISBN 978-986-7339-508 , Kaohsiung, Taiwan, November 5, 2009
- Sun, Y. P., Tsai, C. H. & Liang, Y. C. (2010). Design and Implementation of a Stability Augmentation System for an Unmanned Aerial Vehicle Using Hardware-in-the-Loop Simulation. *Proceedings of the 2010 International Symposium on Mechatronic and Biomedical Engineering and Applications*, pp. 183-193, ISBN 978-986-7339-62-1, Kaohsiung, Taiwan, November 9, 2010
- Travis, J. & Kring, J. (2007). *LabVIEW for Everyone: Graphical Programming Made Easy and Fun* (3rd Ed.), Prentice Hall, ISBN 0-13-185672-3, New Jersey, USA

IntechOpen



Practical Applications and Solutions Using LabVIEW™ Software

Edited by Dr. Silviu Folea

ISBN 978-953-307-650-8

Hard cover, 472 pages

Publisher InTech

Published online 01, August, 2011

Published in print edition August, 2011

The book consists of 21 chapters which present interesting applications implemented using the LabVIEW environment, belonging to several distinct fields such as engineering, fault diagnosis, medicine, remote access laboratory, internet communications, chemistry, physics, etc. The virtual instruments designed and implemented in LabVIEW provide the advantages of being more intuitive, of reducing the implementation time and of being portable. The audience for this book includes PhD students, researchers, engineers and professionals who are interested in finding out new tools developed using LabVIEW. Some chapters present interesting ideas and very detailed solutions which offer the immediate possibility of making fast innovations and of generating better products for the market. The effort made by all the scientists who contributed to editing this book was significant and as a result new and viable applications were presented.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yun-Ping Sun (2011). The Development of a Hardware-in-the-Loop Simulation System for Unmanned Aerial Vehicle Autopilot Design Using LabVIEW, Practical Applications and Solutions Using LabVIEW™ Software, Dr. Silviu Folea (Ed.), ISBN: 978-953-307-650-8, InTech, Available from:

<http://www.intechopen.com/books/practical-applications-and-solutions-using-labview-software/the-development-of-a-hardware-in-the-loop-simulation-system-for-unmanned-aerial-vehicle-autopilot-de>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen