Journal of Mechatronics, Electrical Power, and Vehicular Technology 8 (2017) 60–69

Journal of Mechatronics, Electrical Power, and Vehicular Technology

e-ISSN: 2088-6985
p-ISSN: 2087-3379

www.mevjournal.com

Design and implementation of hardware in the loop simulation for electric ducted fan rocket control system using 8-bit microcontroller and real-time open source middleware

Reza Aulia Yulnandi, Carmadi Machbub *, Ary Setijadi Prihatmanto, Egi Muhammad Idris Hidayat

School of Electrical Engineering and Informatics, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung 40132, Indonesia

Received 28 April 2017; received in revised form 22 June 2017; accepted 02 July 2017
Published online 31 July 2017

Abstract

Hardware in the Loop Simulation (HILS) is intended to reduce time and development cost of control system design. HILS systems are mostly built by integrating both controller hardware and simulator software where the software is not an open source. Moreover, implementing HILS by using manufactured system is costly. This paper describes the design and implementation of HILS for Electric Ducted Fan (EDF) rocket by using open-source platform for development with middleware. This middleware system is used to bridge the data flow between controller hardware and simulator software. A low-cost ATMEGA 2560 8-bit microcontroller is used to calculate rocket's attitude with Direction Cosine Matrix (DCM) algorithm and PID controller is employed to regulate rocket's dynamics based on desired specifications. X-Plane 10 simulator software is used for generating simulated sensory data. The test results validate that HILS design meets the defined specifications, i.e. angle difference of 0.3 degrees and rise time of 0.149 seconds on pitch angle.

Keywords: HILS; DCM; open-source platform; X-Plane; middleware; EDF rocket

I. Introduction

Electric-Ducted Fan (EDF) rocket is one class of rocket that uses electric thruster as its main actuator [1]. It gains popularity for education and research purposes due its low cost and versatility. One important subsystem of an EDF rocket is the autopilot system [2]. Testing an autopilot system in a real experiment directly would lead to a high risk of failure due to its low durability and a high cost.

Computer's system evolution creates the development of simulation method by using the virtual environment to imitate real situation became more preferable [3]. However, simulation result does not always give the same result as the real implementation process due to the uncertainty in electronic hardwares. Therefore, Hardware in the Loop Simulation (HILS) approach is used. HILS is the simulation system that integrates hardware system to simulation process. HILS have been carried out to improve the accuracy of results [4, 5, 6]. Most of the development processes use costly controller hardware such as Motorola's MPC5554-MCU [7].

This hardware is not suitable for simple autonomous system such as Proportional Integral Derivative (PID) algorithm due to the high cost which is not in accordance with the achieved performance. PID algorithm does not need to have a very high-speed processing unit to produce a control signal. Hardware with average speed processing will be sufficient to handle the control process. It also does not need a large memory capacity to store the program code. Moreover, most of the HILS use integrated hardware and software system which is not an open source, not a re-programmable nor having non-free license [8, 9, 10].

* Corresponding Author. Tel: +62 22 250 0960
E-mail address: carmadi@stei.itb.ac.id

doi: https://dx.doi.org/10.14203/j.mev.2017.v8.60-69

In this paper, the design and implementation of HILS system is presented, with the main contribution of exposing three main subsystems which are middleware design to ease bridge process, controller hardware, and simulator software. The middleware performance will be validated through comparison between sensor from simulator software and sensor from sensory hardware with attitude control mechanism to imitate simulator software.

II. Component of HILS

The HILS system consists of three subsystems, i.e. controller hardware, simulator software, and middleware system.

- *Controller Hardware* acts as the main control unit in HILS. This subsystem also controls sensory data retrieval both from sensory hardware and simulator software.
- *Simulator Software* generates sensory data as replacement of sensory hardware, establish a virtual environment to test and provide the simulated dynamics of EDF Rocket.
- *Middleware system* bridges the data flow between controller hardware and simulator software. The middleware system is essential since hardware and software have different communication protocols. Microcontroller hardware uses Universal Asynchronous Receive Transmit (UART) protocol and simulator software uses User Datagram Protocol (UDP).

Illustration of each subsystem's tasks is shown in Figure 1.

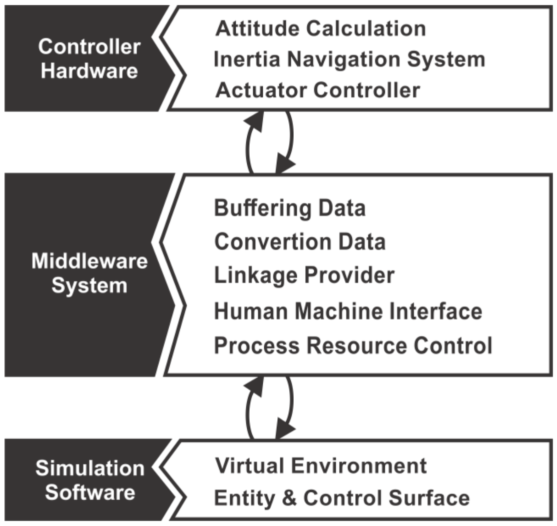


Figure 1. Functional subsystems of HILS

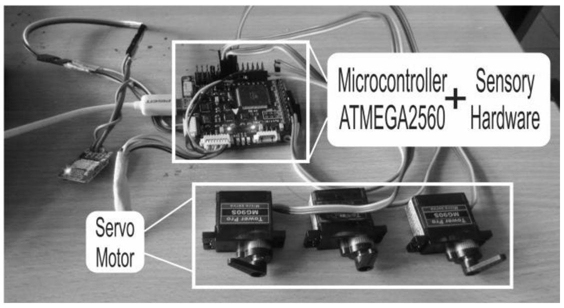


Figure 2. Realization of HILS

III. Design specification

The HILS system for EDF rocket control system uses an 8-bit microcontroller as a main processing unit. The HILS system is designed to meet the following specifications:

- The minimum sampling rate of attitude data from simulator software to controller hardware with 100 samples/second due to the middleware has to provide data with sampling rate that approach the sampling rate from sensory hardware.
- The error of simulated sensor from software simulation to sensory hardware does not exceed 2° in steady-state condition.
- The performance difference of rise time in control process with the same parameter of PID between the simulated sensor and sensory hardware from an initial condition to reach the reference point does not exceed 1 second and steady-state error $< 2\%$.

IV. Implementation

A. Controller hardware

Controller hardware is assigned to process the sensory data and generate the control signal. Typically, HILS is built with 64/32-bit single- board computer as the main processing-unit. This kind of board costs relatively high and is not preferred for trivial computation.

In this work the 8-bit microcontroller of ATMEGA2560 is used to achieve the desired design specifications. This processing unit should be sufficient to meet the specifications based on the following features:

- It has 16 MHz clock speed to calculate attitude and control algorithm which takes 100-120µs.
- It has UART protocol for data transfer from microcontroller to personal computer through serial port (RS232) with transfer rate up to 250Kbps that can achieve minimum sampling rate of 100 samples/second according to design specification.
- It has 400 KHz inter integrated circuit (I2C) communication protocol to send sensory data from IMU to microcontroller in a compact module as shown in Figure 2.
- It has 256 KB flash memory to store program data and 1KB EEPROM should be adequate for calibration data storage.
- It has interrupt features for pulse generator to drive motor servo that will be used to change rocket's control surface position as shown in Figure 2.

The sensory system of our EDF rocket consists of few attitude sensors:

- MPU6050 as accelerometer and gyroscope with 50 Hz data sampling rate and 16-bit word length.
- HMC5883 as magnetometer with 10 Hz data sampling rate and 12-bit word length.
- MS561101BA as barometer with 10 Hz data sampling rate and 24-bit word length.

Gyroscope sensor produces angular velocity to directly calculate the plant's orientation. However, gyroscope cannot be used for long term measurement because it has drift in sensing process. Accelerometer and magnetometer can be used for long-term measurement but they are slower in generating data. Therefore, those cannot be used for short term calculation.

1) Attitude calculation algorithm and sensor fusion

Direction cosine matrix method is used to compute the Euler angles using information from the gyro, accelerometer, magnetic compass and/or GPS [11]. Advantages of this algorithm are its implementation simplicity which can be linearly calculated. Hence, it requires light and short-duration computation.

Vectors i, j, k represent vector body and vectors I, J, K represent vector global as shown in Figure 3. The vector can be described as follow:

$$i^B = \begin{bmatrix}1\\0\\0\end{bmatrix}, j^B = \begin{bmatrix}0\\1\\0\end{bmatrix}, k^B = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

$$I^G = \begin{bmatrix}1\\0\\0\end{bmatrix}, J^G = \begin{bmatrix}0\\1\\0\end{bmatrix}, K^G = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

The projection of vector i toward the X-axis of global coordinate frame can be described as

$$i_x^G = |i|cos(X,i) = cos(I,i) \tag{1}$$

where $|i|$ is normalization of vector i and $cos(I,i)$ is cosine angle that created by vector I and vector i. When $|I| = 1$ dan $|i|$ =1, equation (1) can be written as $i_x^G = cos(I,i) = \frac{I.i}{|I||i|}$. Since $cos(a,b) = \frac{a.b}{|a||b|}$, then

$$i_x^G = |I||i|cos(I,i) = I.i \tag{2}$$

Similarly, the projections of vector i toward the other axes are given as follow:

$$i_y^G = |J||i|cos(J,i) = J.i \tag{3}$$

$$i_z^G = |K||i|cos(K,i) = K.i \tag{4}$$

Then equations (2), (3) and (4) can be written as

$$i^G = \begin{bmatrix}i_x^G\\i_y^G\\i_z^G\end{bmatrix} = \begin{bmatrix}I.i\\J.i\\K.i\end{bmatrix}, j^G = \begin{bmatrix}j_x^G\\j_y^G\\j_z^G\end{bmatrix} = \begin{bmatrix}I.j\\J.j\\K.j\end{bmatrix},$$

$$k^G = \begin{bmatrix}k_x^G\\k_y^G\\k_z^G\end{bmatrix} = \begin{bmatrix}I.k\\J.k\\K.k\end{bmatrix}$$

Vector i, j, k in global coordinate can be described as follow:

$$\begin{aligned}[i^G \quad j^G \quad k^G] &= \begin{bmatrix}I.i & I.j & I.k\\J.i & J.j & J.k\\K.i & K.j & K.k\end{bmatrix}\\ &= \begin{bmatrix}cos(I,i) & cos(I,j) & cos(I,k)\\cos(J,i) & cos(J,j) & cos(J,k)\\cos(K,i) & cos(K.j) & cos(K,k)\end{bmatrix}\\ &= DCM^G\end{aligned}$$

Vector I, J, K in body coordinate can be described as follow:

$$\begin{aligned}[\mathrm{I^B} \quad \mathrm{J^B} \quad \mathrm{K^B}] &= \begin{bmatrix}\mathrm{I.i} & \mathrm{J.i} & \mathrm{K.i}\\\mathrm{I.j} & \mathrm{J.J} & \mathrm{K.j}\\\mathrm{I.k} & \mathrm{J.J} & \mathrm{K.k}\end{bmatrix}\\ &= \begin{bmatrix}\cos(\mathrm{I,i}) & \cos(\mathrm{J,i}) & \cos(\mathrm{K,i})\\\cos(\mathrm{I,j}) & \cos(\mathrm{J,j}) & \cos(\mathrm{K,j})\\\cos(\mathrm{I,k}) & \cos(\mathrm{J,k}) & \cos(\mathrm{K,k})\end{bmatrix}\\ &= \mathrm{DCM^B}\end{aligned}$$

DCM defines the rotation of plant body toward other coordinate. DCM also can be used to determine vector coordinates global of motion in case that the vector coordinates body of motion is known (vice versa).

According to Figure 4, the basic equation of rotation vector on axis are $r = r(t), r' = r(t+dt), dr = r' - r$ where, vector r' is rotation vector with angle vector $d\theta$ in interval time dt on axis vector u. Then vector u is cross product for r and r'

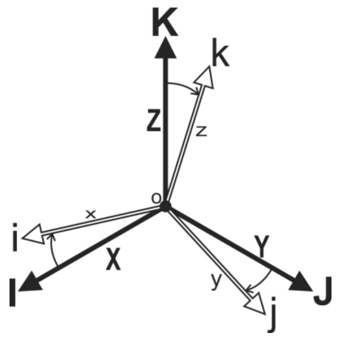


Figure 3. Vector body (white) and vector global (black)

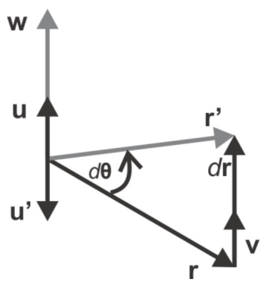


Figure 4. Rotating vector on axis

$$u = \frac{r \times r'}{|r \times r'|} = \frac{r \times r'}{|r||r'|\sin(d\theta)} = \frac{r \times r'}{|r|^2 \sin(d\theta)} \quad (5)$$

Where, $|r| = |r'|$ since rotation process does not alter length of vector. The linear velocity of vector r can be defined as vector

$$v = \frac{dr}{dt} = \frac{r' - r}{dt} \quad (6)$$

Angular rate of displacement vector r can be described as follow:

$$w = \frac{d\theta}{dt} u \quad (7)$$

The equation (8) is obtained by substituting equation (5) to equation (7).

$$w = \frac{d\theta}{dt} u = \frac{d\theta}{dt} \frac{r \times r'}{|r|^2 \sin(d\theta)} = \frac{r \times r'}{|r|^2 dt} \quad (8)$$

Hence for small $d\theta$, $\sin(d\theta) \approx d\theta$, the equation (9) is obtained as follow:

$$w = \frac{r \times r'}{|r|^2 dt} = \frac{r \times (r + dr)}{|r|^2 dt} = \frac{(r \times r) + (r \times dr)}{|r|^2 dt}$$

$$w = \frac{r \times dr}{|r|^2 dt} = \frac{r \times v}{|r|^2} \quad (9)$$

Vector v is described by deriving the equation as follow:

$$w \times r = \frac{(r \cdot r)v + (v \cdot r)r}{|r|^2} = \frac{|r|^2 \cdot v}{|r|^2} + \frac{0}{|r|^2} = \frac{|r|^2 \cdot v}{|r|^2} = v$$

Therefore:

$$\mathrm{dr_x = dt\, v_x = w_x \times r\, dt}$$

$$\mathrm{dr_y = dt\, v_y = w_y \times r\, dt}$$

$$\mathrm{dr_z = dt\, v_z = w_z \times r\, dt}$$

Then

$$v = \frac{dr}{dt} = (w_x + w_y + w_z) \times r$$

DCM equation can be used to define a sensory system to implement sensors fusion for calculating attitude of the plant. As mentioned to previous section, each of sensory hardware's measurement data can be described as follow:

$A = \{A_x, A_y, A_z\}$ is accelerometer sensor data

$M = \{M_x, M_y, M_z\}$ is magnetometer sensor data

$G = \{G_x, G_y, G_z\}$ is gyroscope sensor data

The measurement results can be presented into global coordinates as shown in Figure 5 with $DCM^G = (DCM^B)^T = [I^B \quad J^B \quad K^B]^T$.

Accelerometer can sense earth's gravity which can become bottom reference. It can be described as follow:

$$K^B = -A \quad (10)$$

Magnetometer can sense earth's magnetic north as reference. It can be described as follow:

$$I^B = M \quad (11)$$

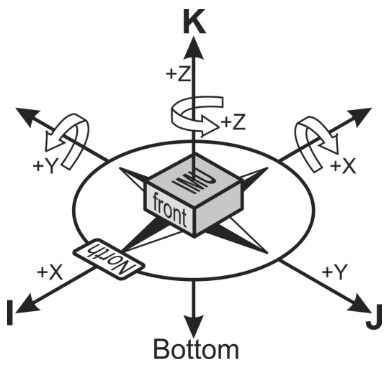


Figure 5. Illustration of sensor orientation on global coordinate frame

With K^B and I^B are known, J^B can be calculated by using equation $J^B = K^B \times I^B$ with the rule of right handed coordinate. While w_a as angular velocity for orientation calculation that affected by accelerometer can be described as

$$w_a = \frac{K_0^B \times v_a}{|K_0^B|^2} \text{ where } v_a = \frac{dK^B}{dt} = \frac{(K_A^B - K_0^B)}{dt}$$

Then

$$\mathrm{d\theta_a = w_a dt}$$
$$\mathrm{= (K_0^B \times v_a)dt}$$
$$\mathrm{= \left(K_0^B \times \left(K_A^B - K_0^B\right)/dt\right)dt}$$
$$\mathrm{= K_0^B \times \left(K_A^B - K_0^B\right)}$$

In order to improve accuracy of attitude calculation, complementary filter can be used. The complementary filter is used to fuse the sensors for getting reliable attitude information. The basic principle of complementary filter is to weight more on the trusted sensor [12].

$$d\theta = \frac{(s_a\, d\theta_a + s_g\, d\theta_g)}{s_a + s_g}$$

with s_a and s_g are respectively weighted accelerometer and gyroscope sensors.

Following same logic as accelerometer, we can determine angular rate with magnetometer described as follow

$$d\theta_m = dt\, w_m = I_0^B \times (I_M^B - I_0^B)$$

To know whether zenith (K) vector using gyroscope in short term condition, it can be described as follow:

$$K_G^B = K_0^B + dt\, v$$

$$K_G^B = K_0^B + (d\theta_g \times K_0^B) \quad (12)$$

With $d\theta_g = w_g\, dt$ and $w_g = G$ because of gyroscope can directly measure angular speed. By using same logic in equation (12), it is known that

$$I_1^B = I_0^B + (d\theta \times I_0^B) \quad (13)$$

$$J_1^B = J_0^B + (d\theta \times J_0^B) \quad (14)$$

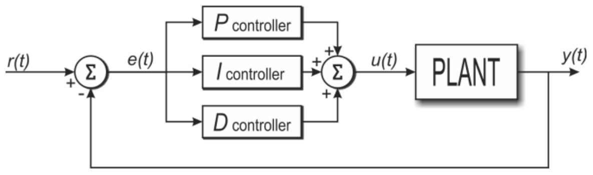


Figure 6. Closed-loop PID

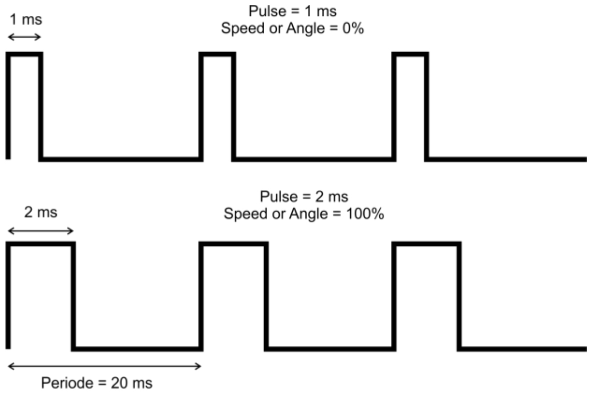
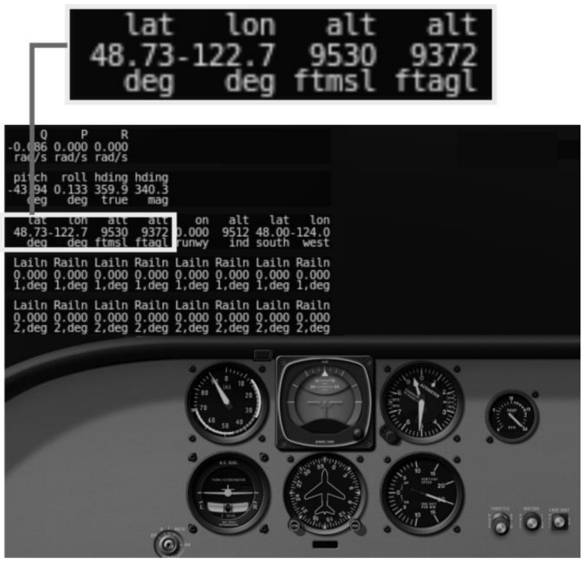


Figure 7. PPM signal



```
  lat     lon    alt    alt
48.73-122.7   9530   9372
  deg     deg  ftmsl  ftagl
```

```
     Q      P      R
-0.086  0.000  0.000
 rad/s  rad/s  rad/s

 pitch   roll  hding  hding
-43.94  0.133  359.9  340.3
   deg    deg   true    mag

   lat    lon    alt    alt     on    alt    lat    lon
48.73-122.7   9530   9372  0.000   9512 48.00-124.0
   deg    deg  ftmsl  ftagl  runwy    ind  south   west

 Lailn  Railn  Lailn  Railn  Lailn  Railn  Lailn  Railn
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 1,deg  1,deg  1,deg  1,deg  1,deg  1,deg  1,deg  1,deg

 Lailn  Railn  Lailn  Railn  Lailn  Railn  Lailn  Railn
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 2,deg  2,deg  2,deg  2,deg  2,deg  2,deg  2,deg  2,deg
```

Figure 8. X-PLANE 10 Unit

2) Control algorithm

Proportional Integral Derivative (PID) is relatively simple control algorithm that is suitable for controlling simple UAV. This control method can be formed by linear calculation which can be handled by 8-bit microcontroller. PID will give feedback control signal from error in order to control the plant in a closed-loop system as shown in Figure 6.

Following Figure 6, PID consists of three main component calculations of the control signal $u(t)$ as given in equation (15):

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \qquad (15)$$

Where K_p , K_i and K_d describe proportional, integral, derivative gains respectively. Each of them has different responses when step input is applied. The response is shown in Table 1. Ziegler–Nichols tuning rule method is used to tune the PID gains.

3) Actuator controller

Servo motors are handling control surface mechanism to generate control motion. Servo motor needs a particular signal for generating control motion which is a pulse with 50 Hz frequency. Most of servo motor is using 1000-2000 microsecond pulse width as depicted in Figure 7. Therefore, ATMEGA's interrupt feature is a great help to resolve amount pulse generator problem.

B. Simulation software

Simulation software is assigned to generate manipulated virtual environment. This environment can generate a streaming simulated sensory data to controller hardware via middleware.

X-Plane 10 (X-Plane version 10) is used in this work for simulation software. According to Figure 8,

Byte								
1	2	3	4	5	6	7	8	9
HEADER					Type Data			
STRING 4 Byte					4-byte unsigned Integer			
Byte								
10	11	12	13		14	15	16	17
Data 1					Data 2			
4-Byte Sign Float					4-Byte Sign Float			
Byte								
...	...	...	...		38	39	40	41
Data ...					Data 8			
4-Byte Sign Float					4-Byte Sign Float			

Figure 9. X-Plane 10 UDP format data

Byte									
1	2	3	4	5	...	...	...	n+2	n+3
Header			Data – 1	Data - 2	...	...	...	Data - n	Check sum

Figure 10. Format data protocol UART

Table 1.
Step-input PID control responses

Increase gain	Rise time	Overshoot	Settling time	Steady-state error
Propor-tional	Decrease	Increase	Increase little	decrease
Integral	Decrease	Increase	Increase	eliminate
Derivative	Alter little	decrease	decrease	Alter little

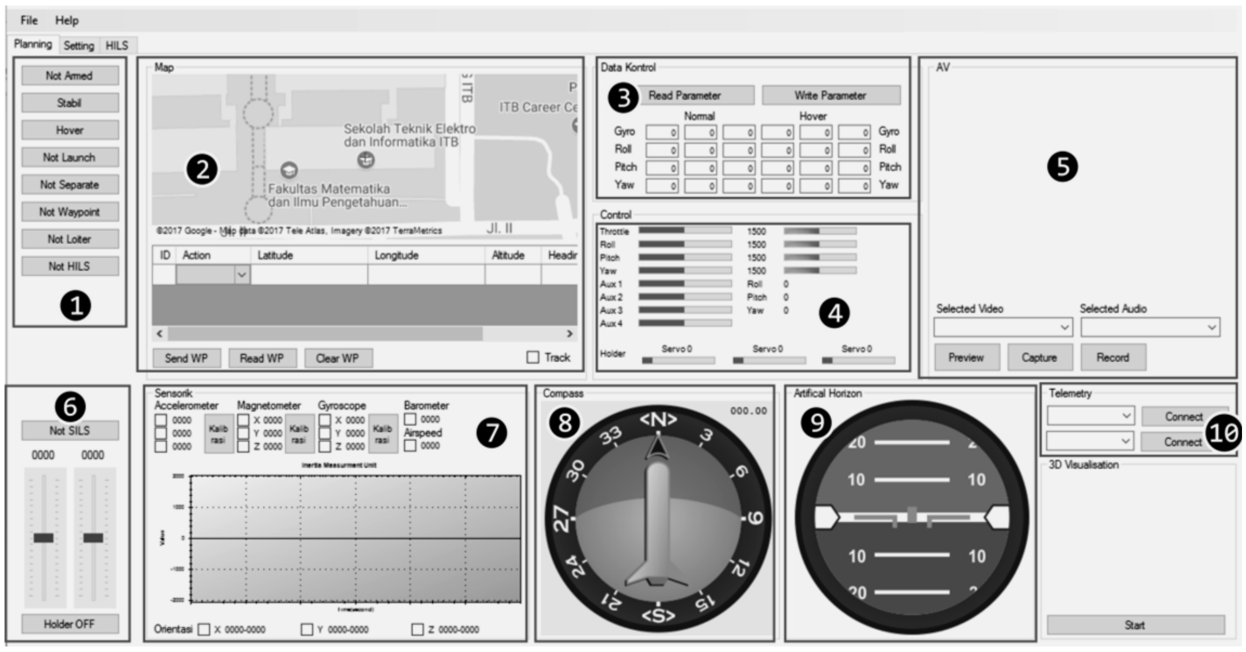


Figure 11. Designed middleware

most of unit on X-Plane 10 are an imperial unit. Therefore, the conversion to metric unit is needed and this is one of tasks assigned to the middleware. X-Plane 10 uses UDP protocol to transfer attitude data. X-Plane 10's format data UDP is described in Figure 9.

X-Plane 10's data transfer rate can be increased up to 100 samples/second. It can be assumed that it can produce simulated sensory data to replace sensory hardware with fastest sampling sensory data which is 50 samples/second from gyroscope and accelerometer.

C. Middleware system

Middleware is assigned to provide demands needed by controller hardware which is sensory data and simulator software which is feedback control from generated sensory data. As mentioned in the previous section, the main concern of middleware system is bridging the difference of communication protocols between UDP, used in simulation software, and UART, used in controller hardware. The middleware system is designed by using Visual Studio 2012 Express. Middleware's assignment will consists in several parts such as follow:

- Setting simulator software to send data (in UDP) every 10 ms or 100 samples/second to avoid a bottleneck in sensory data.
- Creating format data with error checking as shown in Figure 10, to handle UART inability to detect occurrence of data error. That also prevents glitch data which can obstruct controlling process. Header data and checksum will be good solution to prevent data error.
- Bridging between UDP protocol and UART protocol. C# language with framework 4.5 can provide UDP and UART communication protocol.
- Implementing real-time simulation system as well as providing Human Machine Interface (HMI) about condition of controlled plant, as shown in Figure 11, which contain:
  1. Controller Mode to set condition of plant such as launch or recovery mode.
  2. Map and Waypoint to show plant position and set waypoint coordinate.
  3. PID Gain Controller to tune PID gain in the real-time experiment.
  4. Servo Monitor to monitor control signal of servo motor.
  5. Camera Controller to show plant's point of view in launch condition.
  6. Holder Controller for switching to validate mode data sensor in the experiment.
  7. Sensor Monitor for monitoring plant condition in chart form.
  8. Compass to show rocket's yaw orientation.
  9. Artificial Horizon to show rocket's roll and pitch orientation.
  10. UART Controller for selecting UART port of the microcontroller.
- Creating sensory data conversion features because some of X-Plane 10's parameters are using imperial unit and sensory hardware are using metric unit, for example:
  Accelerometer:

  $\frac{\text{feet}}{\text{second}^2}$ to $\frac{\text{meter}}{\text{second}^2}$ to 16 bit digital

  This is intended to reduce microcontroller's workload to interpret data from X-Plane 10.
- Creating data buffering for some of parameter simulated sensory data (i.e. angular speed) with time dependence. Figure 12 shows how to handle that sensory data.

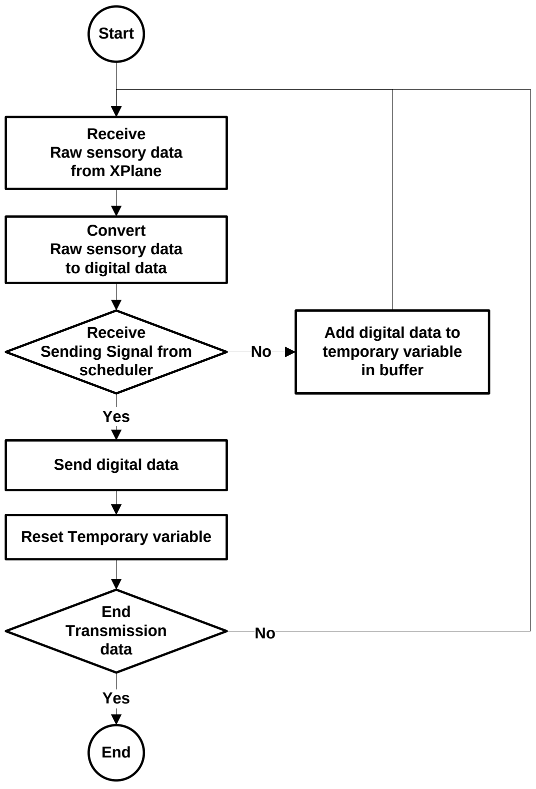

Start
Receive Raw sensory data from XPlane
Convert Raw sensory data to digital data
Receive Sending Signal from scheduler
No
Add digital data to temporary variable in buffer
Yes
Send digital data
Reset Temporary variable
End Transmission data
No
Yes
End

Figure 12. Buffering time-dependent data

V. Experiment

The experiment is conducted in few steps to ensure that middleware system can bridge the attitude data from simulator software for control signal computation in controller hardware. Each experiment is conducted by 30 times to ensure that each experiment result is consistent. These are experiment steps:

- Data transfer rate test.
- Validation simulated attitude data.
- Comparison of control response.

A. Data transfer rate test

The experiment conducted by transferring data from simulator software to controller hardware via middleware to imitate sampling rate data of sensory hardware. Each experiment uses 250 samples data and records time delay between data transmission using controller hardware's timestamp. Every time delay is recorded on controller hardware memory.

Figure 13 and Figure 14 are test sample on gyroscope and accelerometer. Accelerometer gives result of 49.94 samples/second average sampling rates with 0.42 samples/second of standard deviation. Gyroscope gives result of 49.91 samples/second average sampling rates with 0.46 samples/second of standard deviation.

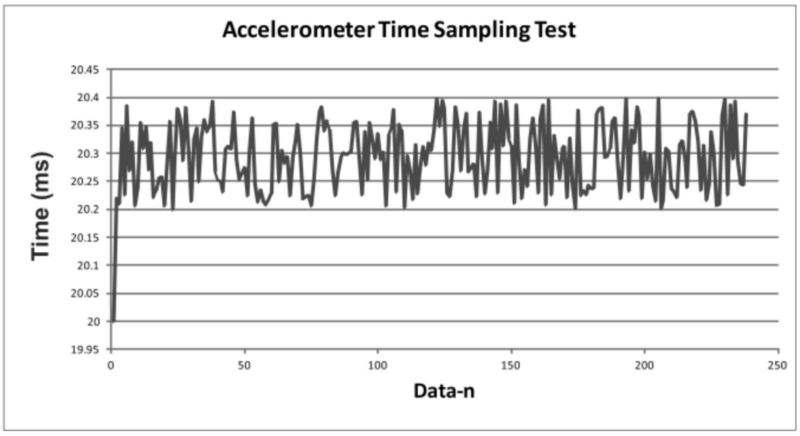


Figure 13. Accelerometer data transfer rate test

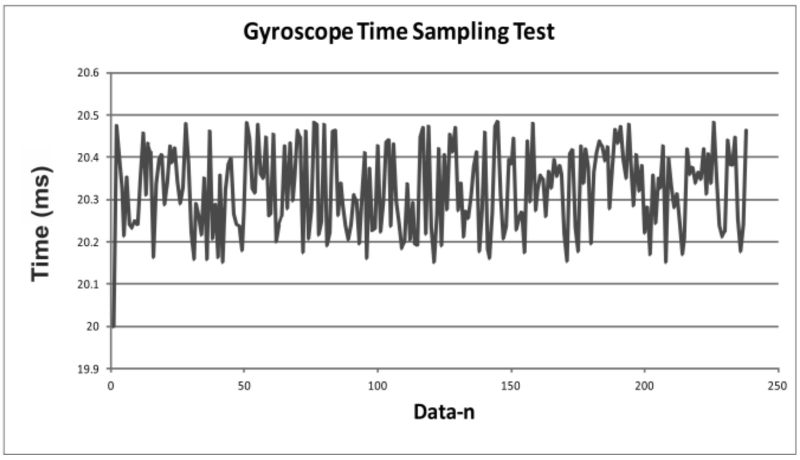


Figure 14. Gyroscope data transfer rate test

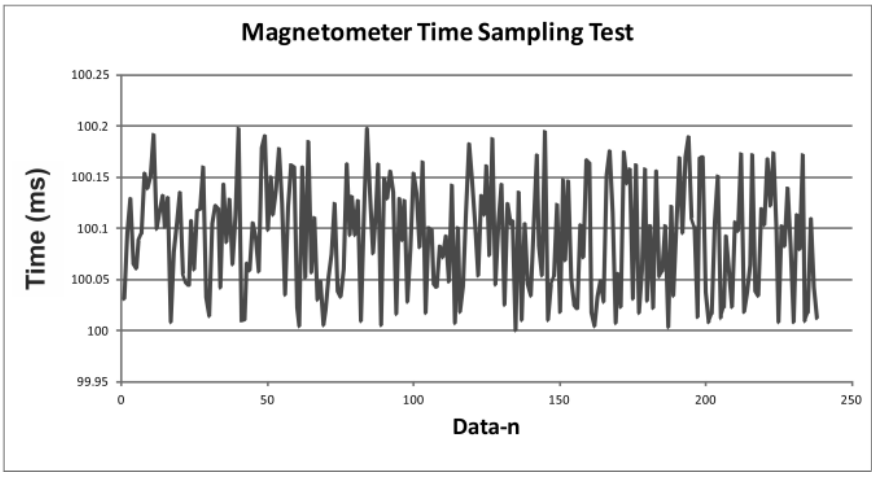


Figure 15. Magnetometer transfer rate test

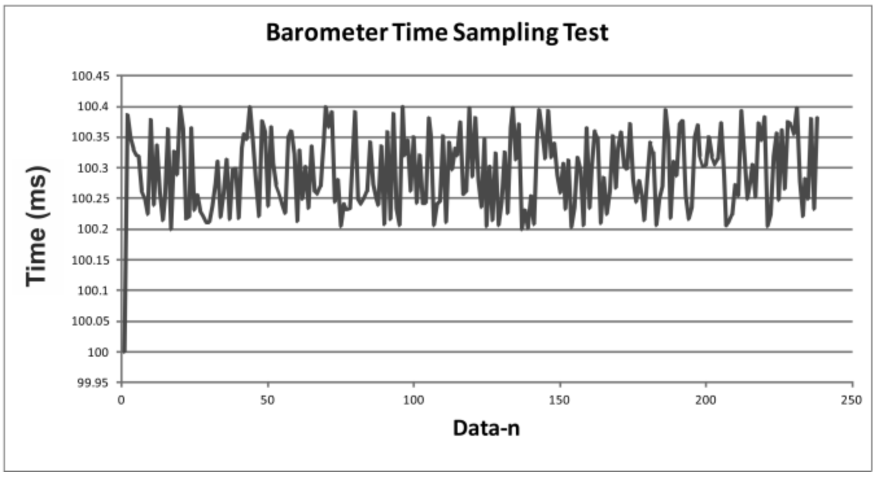


Figure 16. Barometer data transfer rate test

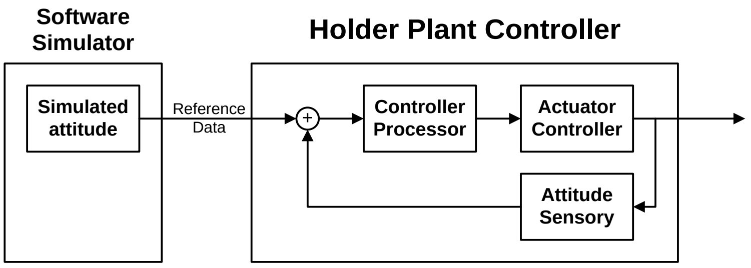


Figure 17. Controller holder plant

Figure 15 and Figure 16 are test sample on magnetometer and barometer. Magnetometer gives result of 9.97 samples/second average sampling rate with 0.29 samples/second standard deviation. Barometer gives result of 9.96 samples/second average sampling rate with 0.27 samples/second standard deviation.

B. Validation simulated attitude data

Validation process compares simulated sensor data and real sensory hardware to determine the deviation of sensory system which represents the real experiment in the laboratory. The good system produces responses that have a good fit to the actual system responses on field testing. Due to laboratory experiment limitation, plant system is placed on a regulated control system. This is intended to make sensory hardware attitude follows the plant's attitude on simulator software.

Following Figure 17, the regulated control system will have a role as plant holder to imitate plant's attitude on simulator software. The regulated control system imitates pitch angle because it is considered has an important role in autonomous take-off, landing and altitude control. The plant holder is depicted in Figure 18.

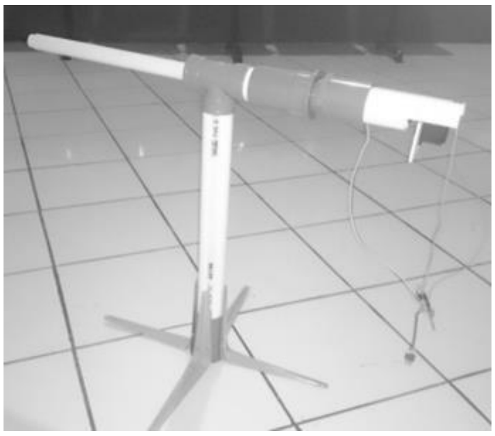
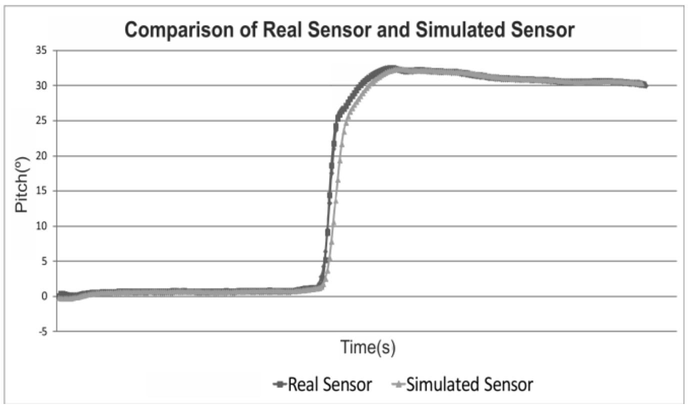
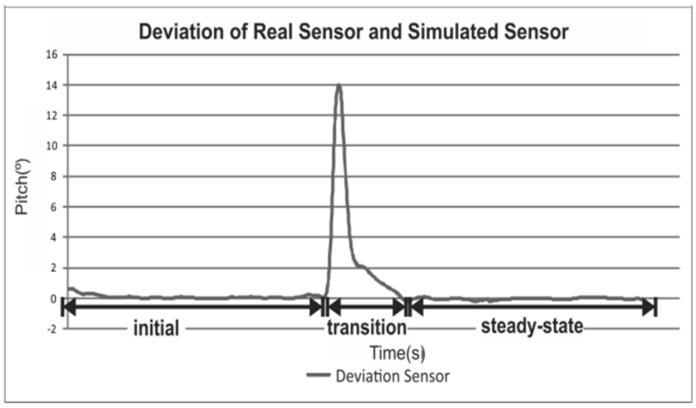


Figure 18. Implemented holder



(a)



(b)

Figure 19. Sensory data; (a) comparison; (b) deviation

The plant holder uses motorize control mechanism to generate pitching motion on the plant. The pitch motion makes microcontroller recalculate plant’s attitude using sensory hardware. Figure 19 shows result sample of this step. On steady-state condition, testing process for pitch angle gives comparison result differs by 0.36 degree average with 0.04 degree standard deviation.

C. Comparison of control response

The comparison of control response is done between a plant with sensory data from simulator software and sensory hardware. Rise time and steady-state error become comparison aspect. Figure 20 is an example of control response result and performance difference is presented in Table 2. According to Table 2, average of control responses differs by 0.46s for rise time and 0.53% for steady-state error.

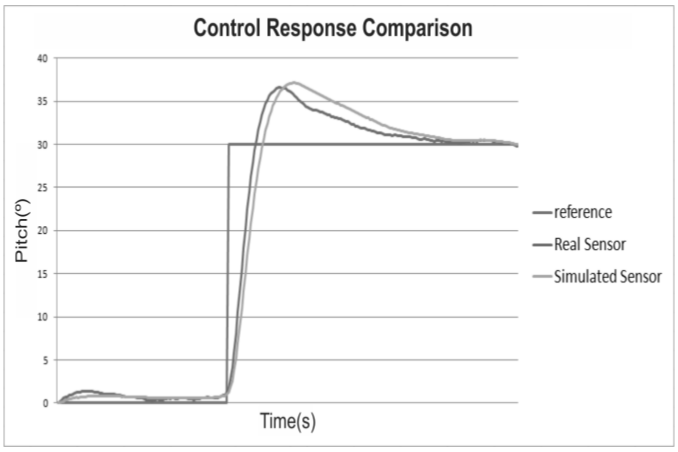


Figure 20. Control response comparison

Table 2.
Experiment result

	Average		Standard deviation	
	Rise time (s)	**Steady-state error (%)**	**Rise time (s)**	**Steady-state error (%)**
Simulated sensory	1.89	0.93	0.15	0.11
Sensory hardware	1.43	0.4	0.11	0.07

VI. Conclusions

According to the experimental results, it is concluded that the designed HILS system satisfies the desired specification and produces responses that have a good fit to the actual system responses. According to desired system specifications, sensory data validation can achieve occurrence error less than 2° and the rise time of control PID between simulated sensory data and sensory hardware has difference less than 1 second and steady-state error of each sensory data less than 2%. The attitude data sampling rate through middleware system suffices the demand of sensory data for controller hardware.

Acknowledgement

This research is funded by Ministry of Research, Technology, and Higher Education of Republic of Indonesia via research scholarship of Second Batch of Program Menuju Doktor Sarjana Unggul (PMDSU) (328/SP2H/LT/DRPM /IX/2016).

References

[1] H. Y. Irwanto, "HILS of auto take off system: For high Speed UAV using booster rocket," in *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2016, pp. 373–380.

[2] D. Joglekar *et al.*, "Autopilot design for aerospace vehicle using GUI &#x2014; A user friendly approach," in *2013 Annual IEEE India Conference (INDICON)*, 2013, pp. 1–6.

[3] N. P. Mahalik and Kiseon Kim, "A Prototype for Hardware-in-the-Loop Simulation of a Distributed Control Architecture," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.*, vol. 38, no. 2, pp. 189–200, Mar. 2008.

[4] H. Min *et al.*, "Thruster Control for Micro-satellite Attitude and Hardware-in-the-loop Demonstration," in *2012 International Conference on Industrial Control and Electronics Engineering*, 2012, pp. 588–591.

[5] A. Bittar and N. M. F. de Oliveira, "Hardware-in-the-loop simulation of an attitude control with switching actuators for SUAV," in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2013, pp. 134–142.

[6] A. Alsaraj and G. Stuffle, "Investigation of hardware-in-loop simulation (HILS) for guidance system," in *2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2015, pp. 704–708.

[7] J. Park *et al.*, "Hardware in-the-loop simulation for ABS using 32-bit embedded system," *Control. Autom. Syst. (ICCAS), 2011 11th Int. Conf.*, 2011, pp. 575–580.

[8] A. Nurhafid *et al.*, "Modelling and simulation of RKX200 rocket flight dynamics," in *2013 3rd International Conference on Instrumentation, Communications, Information Technology and Biomedical Engineering (ICICI-BME)*, 2013, pp. 230–236.

[9] In-Gyu Jang *et al.*, "Testing 32-bit embedded system using hardware-in-the-loop-simulation of automatic transmission," in *2007 International Conference on Control, Automation and Systems*, 2007, pp. 399–403.

[10] Y. W. Hadi and R. T. Bambang, "Development of hardware-in-the-loop simultion for rocket guidance system," in *2015 International Conference on Electrical Engineering and Informatics (ICEEI)*, 2015, pp. 229–234.

[11] B. J. Emran *et al.*, "A Cascaded Approach for Quadrotor's Attitude Estimation," *Procedia Technol.*, vol. 15, pp. 268–277, 2014.

[12] M. S. Islam *et al.*, "A low cost MEMS and complementary filter based attitude heading reference system (AHRS) for low speed aircraft," in *2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, 2016, pp. 1–5.