**Design of 8-bit Microcontroller Hardware in the Loops Simulation for Electronic Ducted Fan Rocket**

**Implemented by Real-Time Open-Source Middleware.**

Reza Aulia Yulnandi1, Carmadi Machbub2, Ary S. Prihatmanto3, Egi M.I. Hidayat4

Sekolah Teknik Elektro dan Informatika, Institusi Teknologi Bandung

Jl Ganesha 10, Bandung, Indonsia

[1reza97aulia@s.itb.ac.id](mailto:1reza97aulia@s.itb.ac.id), [2carmadi@stei.itb.ac.id](mailto:2carmadi@stei.itb.ac.id), [3asetijadi@lskk.ee.itb.ac.id](mailto:3asetijadi@lskk.ee.itb.ac.id), [4egi@lskk.ee.itb.ac.id](mailto:4egi@lskk.ee.itb.ac.id)

|  |
| --- |
| **Abstract**  Hardware in the Loops Simulation(HILS) is intended to reduce time and development cost of Rocket control system. Mostly, HILS systems are built by integrating hardware controller and software simulator yet not open-source. It will lead to hindrance of improvement simulation result accuracy. However, implementing HILS using manufactured system is costly, so designing HILS for a simple control system will be troubled by that problem. This paper deals with our research about design of HILS for Electronic Ducted Fan (EDF) rocket using open-source platform for development of a middleware to bridge hardware controller and software simulator. A low cost microcontroller 8-bit ATMEGA 2560 is used to calculate attitude with Direct Cosine Matrix (DCM) algorithm and PID controller to meet desired specifications. XPLANE-10 software simulator is used for generating simulated sensory data because it uses User Datagram Protocol (UDP) without handshaking to stream data simulation. C# language is used to design multithread for database management, sampling data, conversion data and linkage providing for middleware system. The test results validate that HILS design is met to the defined specifications, i.e. divergence angle of 0.3 degrees and rise time of 0.149 seconds on pitch angle.  Keywords: HILS, DCM, open source platform, XPLANE, middleware, EDF rocket |

1. **INTRODUCTION**

Electronic Ducted Fan (EDF) rockets are new type rocket with electric thruster[3] which is devised by Indonesian National Institute of Aeronautics and Space (LAPAN) for university’s student to participant in development electric thruster rocket. Designing process of autonomous EDF rocket has autopilot system which will have to go through testing process to ensure control running as expected design[7]. Testing process of EDF rocket’s autopilot system is considered difficult because is really easily damaged and demand large area testing.

Computer’s system evolution encourage development simulation method with manipulate virtual environment[2] to imitate real situation. However, simulation result is not always gives same result with real implementation process. Therefore, Hardware in the Loops Simulation(HILS) is simulation system which integrates hardware system to simulation process. HILS have been carried out[6][8][9] to improve accurate result. Most of development process are using costly hardware controller such as Motorola's MPC5554-MCU[4]. It is considered not suitable for simple autonomous system with Proportional Integral Derivative (PID) algorithm because the expense cost is not worth for performance result. Hardware controller doesn’t need high speed processing just relatively average calculating speed can covering control process. It also doesn’t need big capacity memory to implement program data. Apart from costly issue, most of them use integrated hardware and software system[1][5][10] which is not open-source or can’t be reprogram. Hence, further improvement testing and simulation result is nearly impossible.

Therefore, this paper deals with our research about design of HILS with middleware to ease bridge process between hardware controller and software simulator. Validation of middleware performance design will be determined by comparison simulated sensory form software simulator and real-sensory from sensory hardware with attitude control mechanism to imitate software simulator.

1. **COMPONENT OF HILS**

HILS is designed to consist of 3 main components or subsystem to classify outline task of each components into

* **Hardware Controller** for implemented algorithm to calculate sensory data from software simulator and sensory hardware. It will be used to generate feedback control.
* **Software Simulator** has task to generate sensory data as a replacement of hardware sensor and establish virtual environment to testing and give attitude result of response system control EDF Rocket.
* **Middleware system** is software component which will have task to bridge and control data flow between hardware controller and software simulator. Middleware system is needed because hardware and software have different communication protocol which is Universal Asynchronous Receive Transmit (UART) protocol for 8-bit microcontroller hardware and User Datagram Protocol (UDP) for software simulator.

All of them will create synergy data flow that will be need by each sub-system. Illustration task of each sub-system is shown by Figure 1.

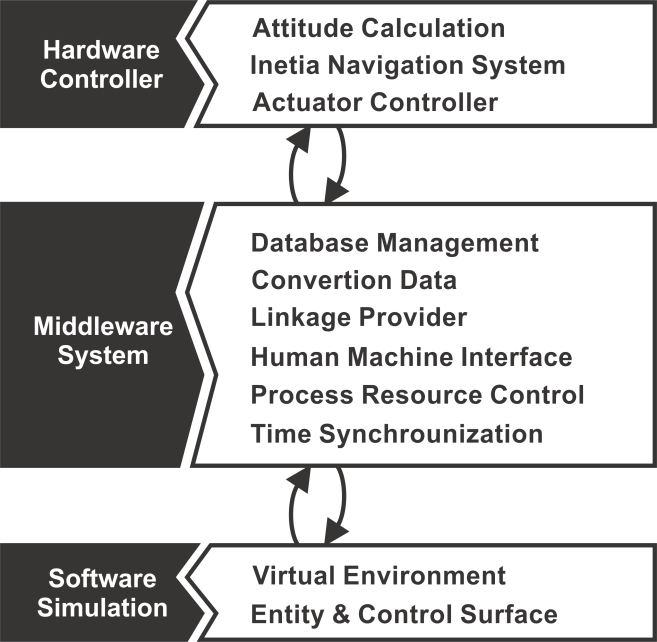


Figure 1. Subsystems of HILS

1. **DESIGN SPECIFICATION**

Designing system of HILS use microcontroller 8-bit as base of hardware controller which will devise to have specification as such as.

* Minimal transfer rate flow data of attitude and control between hardware controller and software simulator estimate by 10 Hz.
* Error simulated sensor from software simulation to sensory hardware not exceed in steady state condition
* Divergence rise time in control process with same parameter of PID between simulated sensor and sensory hardware from initial condition to reach the reference point not exceed 1 second and steady state error

Establishment of design specifications are intended to become main aim and direction on experiment to get result of this research.

1. **IMPLEMENTATION**

According to previous explanation about component of HILS, these are following designed component that used as well as their consideration for implementation process in real-time system.

* 1. **Hardware Controller**

Sensory data processing and give feedback controller will be handled by hardware controller. It will have relatively low price to solve the cost product problem yet can accomplish given task. Comparison between HILS with 64/32bit single board[1] computer and 8-bit microcontroller will give clearly answer that microcontroller have few times lower cost. Despite of that, 8-bit ATMEGA 2550 can accomplice devised design specification. These are reasons of uses 8-bit microcontroller ATMEGA 2560 is enough to designing HILS system.

* 16 MHz clock speed to calculate attitude and control algorithm which is use 100-120us to accomplice these task.
* It using Universal Asynchronous Receiver Transmitter (UART) protocol to transferring data from microcontroller to personal computer through serial port (RS232) with max speed 250Kbps that can achieve transfer rate data minimal 10Hz according to design specification.
* 400 KHz I2C protocol communication can be used for transferring sensory data frominertia navigation system (INS) sensory to hardware controller.
* 256 KB flash memory to save program data and have 1KB EEPROM which can use to save calibration data.
* It has interrupt features for generate pulse in manipulated way to control motor servo that will be used for control surface.

These are sensory components that will be used in HILS system are consist of

* MPU6050 for accelerometer and gyroscope with 20 Hz sampling rate data with 16 bit word length.
* HMC5883 for magnetometer with 10 Hz sampling rate data with 12 bit *word length*.
* MS561101BA for barometer 10 Hz sampling rate data with 24 bit *word length*.

Gyroscope sensor produces angular velocity to directly calculate orientation of plant. However, gyroscope can’t be use for long term because it has drift in process. Accelerometer and magnetometer can be used for long term but it generates data more slow so we can’t use them for short term.

* + 1. *Attitude Calculation Algorithm*

Direct cosine matrix is a sensor fusion method which can applied to calculate motion orientation of plant[12][13][14]. Advantages of this algorithm are simplistic implementation which can be linearly calculated so we will have less processing load in a brief moment.

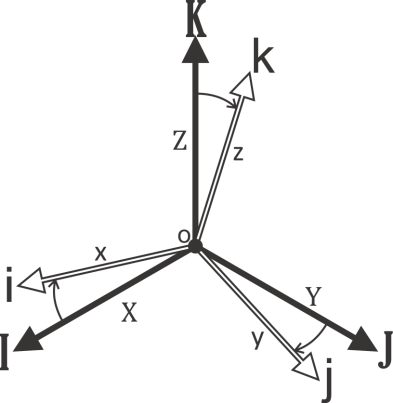


Figure 2. Vector body (white) and vector global (black)

Following Figure 2, vector and . We can describe as

We can use projection vector toward global axis X will become

(1)

where

is normalization of vector

is cosines angle that created by vector and vector .

With dan =1, equation 1 can be written as

since

As well as

Then

,,

Vector in global coordinate can be describe as

Vector in body coordinate can be described as

DCM defines rotation of plant body toward other coordinate. DCM also can be used to determine vector coordinates global of motion if we knew vector coordinates body of motion (vice versa).

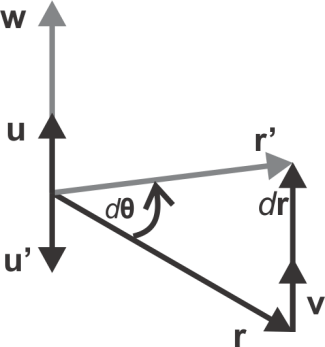


Figure 3. Rotating vector on axis

According to Figure 3, we can have basic equation of rotation vector on axis

Vector **r'** is rotation vector with angle vector d**θ** in interval time dt on axis vector **u** then vector **u** is cross product for **r** and **r'**

(2)

Where since rotation process does not alter length of vector.

The linear velocity of vector **r** can be defined as vector

(3)

Angular rate of displacement vector **r** can be described as

(4)

If we substitute equation 2 to equation 4, we will get equation 5.

(5)

As d**θ** approach 0 then sin(dθ) will approach 0.

We can also describe vector **v** by deriving the equation as

So we have

Then

Referring to previous formula explanation, we can use all of that to define sensory system to implement sensors fusion for calculating attitude of plant.

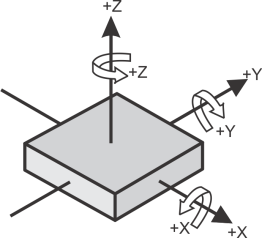


Figure 4. Illustration sensory hardware

According to Figure 4, let’s say

is accelerometer sensor data

is magnetometer sensor data

is gyroscope sensor data

with

because accelerometer can sense earth’s gravity which can become reference value.

because magnetometer can sense earth’s magnetic north as reference.

Knowing and, can be calculated by with rule of right handed coordinate.

asangular velocity for calculating that affected orientation by only accelerometer can be described as

where

Then

In order to get improvement accuracy of attitude we can fuse accelerometer and gyroscope sensors with complementary filter[13][14].

with *s* is weighted sensor value.

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

To know zenith (*K*) vector using gyroscope in small period time, we describe it as

(6)

with and is angular rate of gyroscope. Using same logic then

(7)

(8)

* + 1. *Control Algorithm*

Proportional Integral Derivative(PID) is relatively simple algorithm control that suitable for development simple UAV because it can be formed by linearization calculation which can be handled by microcontroller 8-bit. PID will give feedback control from error in order control plant in closed-loop system.

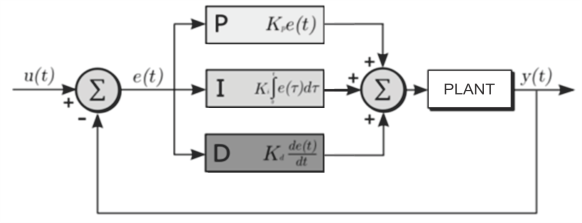


Figure 5. Closed-loop PID

Following Figure 5, PID consists in three main component calculations.

(9)

(10)

(11)

Equation 9 is describing proportional control, Equation 10 is describing integral control and Equation 11 is describing derivative control. Each of them has difference responses to handle error following Table 1 assuming step function input. PID tuning method will follow rule tuning Ziegler–Nichols method.

Table 1.

Proportional Integral Derivative Control Response

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *Rise time* | *Overshoot* | *Settling time* | *Steady-state error* |
| Propor-tional | Decrease | Increase | Increase  little | decrease |
| Integ-ral | Decrease | Increase | Increase | eliminate |
| Deri-vative | Alter little | decrease | decrease | Alter little |

* + 1. *Actuator Controller*

Control surface mechanism the plane will be handled by motor servo. It needs particular signal to make control motion which is pulse with 50 Hz frequency and mostly using 1000-2000 microsecond width pulse as depicted in Figure 6.

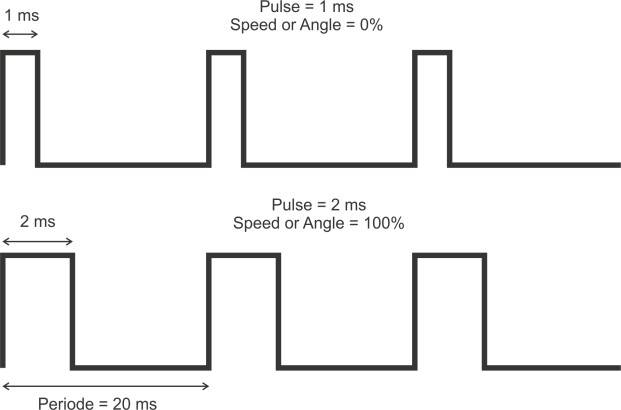


Figure 6. PPM signal

Therefore, ATMEGA’s interrupt feature will be great help to resolve amount pulse generator problem.

* 1. **Software Simulation**

Software simulation has task to generate manipulated virtual environment in order to generate streaming simulated sensory data to hardware controller via middleware.



Figure 7. XPLANE 10 Unit

XPLANE version 10 will be used for software simulation. According to Figure 7, most of unit on XPlane-10 are imperial unit so we need to convert it to metric unit which will become the task for middleware. XPlane uses user datagram protocol (UDP) to transfer attitude data. XPlane-10’s format data UDP described in Figure 8.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Byte | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| HEADER | | | | Type Data | | | |
| STRING 4 Byte | | | | 4-byte unsigned Integer | | | |
| Byte | | | | | | | |
| 9 | 10 | 11 | 12 | 13 | 14 | 14 | 15 |
| Data 1 | | | | Data 2 | | | |
| 4-Byte Sign Float | | | | 4-Byte Sign Float | | | |
| Byte | | | | | | | |
| ... | ... | ... | ... | 36 | 37 | 38 | 39 |
| Data ... | | | | Data 8 | | | |
| 4-Byte Sign Float | | | | 4-Byte Sign Float | | | |

Figure 8. XPLANE 10 UDP format data.

Xplane’s transfer rate data can be increased up to 100 Hz so we can assume that it can be produced simulated sensory data to replace sensory hardware with fastest sampling sensory data 20 Hz from accelerometer and gyroscope.

* 1. **Middleware system**

Shortly, middleware design has task to provide demands between hardware controller which is sensory data necessity and software simulator which is feedback control from generated sensory data. It will become the outline task to bridging process. According to previous explanation in hardware and software, the most important thing that middleware needs have are UDP and UART protocol method to solve dissimilarity communications protocol problem. Middleware’s tasks will consist in few parts such as

* Software simulator set to UDP send data every 10 second or 100 Hz sampling rate data in order to avoid bottleneck in sensory data.
* Make format data with error checking because UART communication method can’t detect occurrence of error data. That also prevents glitch data which can obstruct controlling process. Header data and checksum will be good solution to prevent error data. See Figure 9.

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

Figure 9. Format data protocol USART.

* Middleware system has UDP protocol and UART protocol to bridging attitude and control data. C# language with dot.NET framework 4.5 on visual studio 2012 can provide the communications protocol. This system will be implemented to real-time simulation system that will give us Human Machine Interface (HMI) about condition controlled plant that is shown in Figure 10.

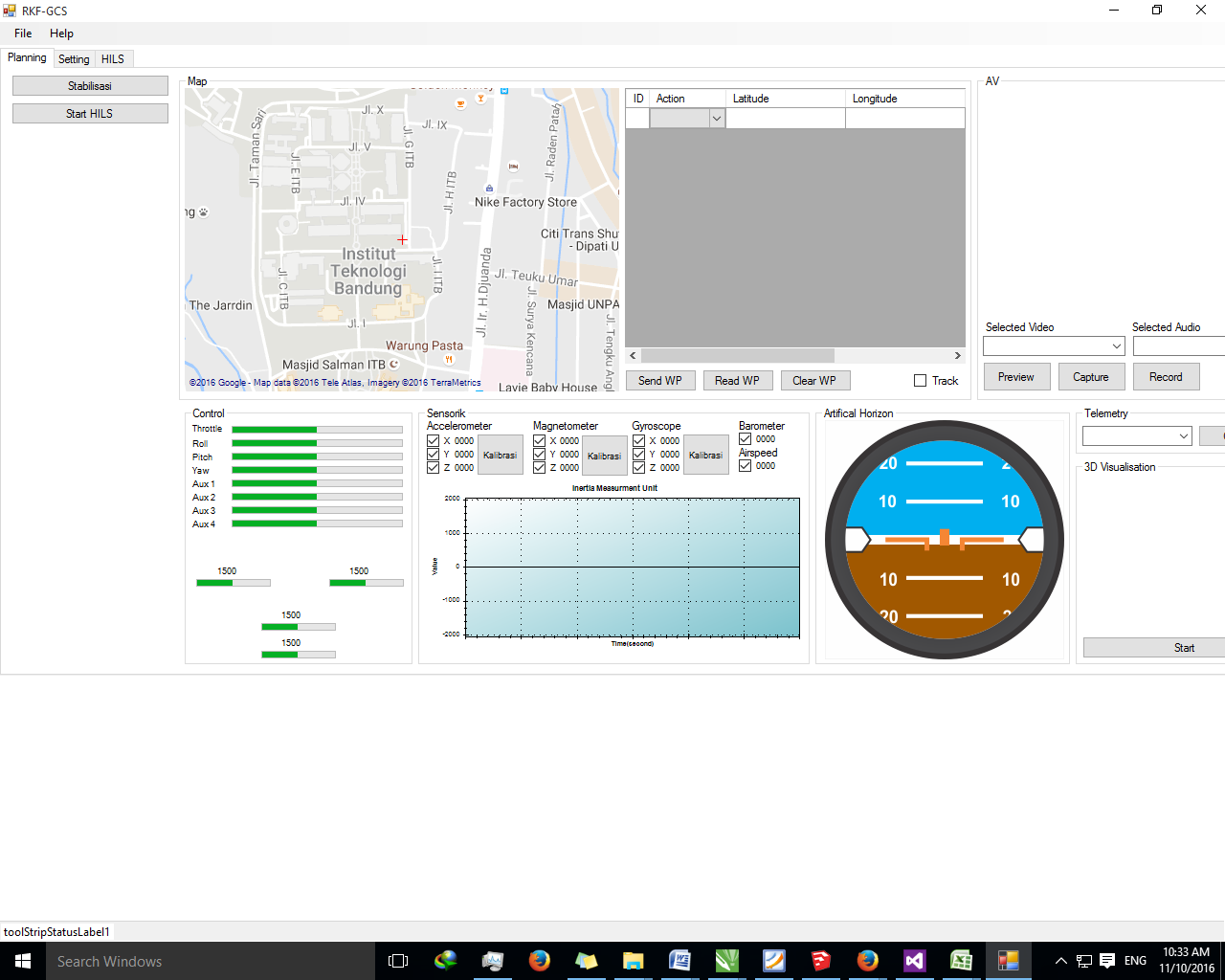


Figure 10. Designed Middleware HILS

* Middleware needs conversion sensory data features because some of XPlane-10’s parameters are using imperial unit and sensory hardware are using metric unit. Accelerometer:

to to 16 bit digital

Gyroscope:

to to 16 bit digital

Magnetometer:

Gauss to 12 bit digital

Barometer:

to to 24 bit digital

This feature can reduce workload hardware controller and control data sampling rate so it won’t diverse.

* Bank data system because some of parameter simulated sensory data (i.e. angular speed) is time dependant, so sum of them will depend on various time sampling. We use closed-loop scheduling iteration, as indicated in Figure 11, to solve that problem.

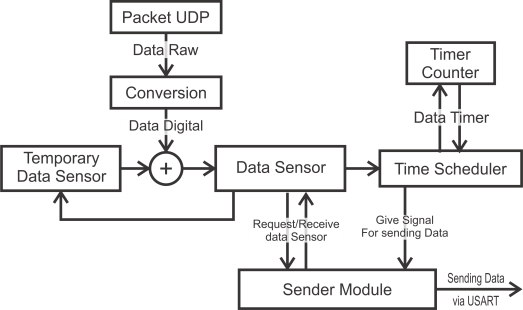


Figure 11. Closed-loop scheduling iteration

1. **EXPERIMENT**

Experiment will be conduct in few steps to ensure middleware system can provide attitude data for calculation control plant. These are step experiments:

* Transfer rate data testing.
* Validation simulated attitude data.
* Comparison for control response
  1. **Transfer rate data testing**

The experiment will be done by transferring data from software simulator to hardware controller via middleware to imitate sampling data rate of hardware sensory. We use 250 sample data to send and calculate the sampling data recorded by hardware controller using timestamp.

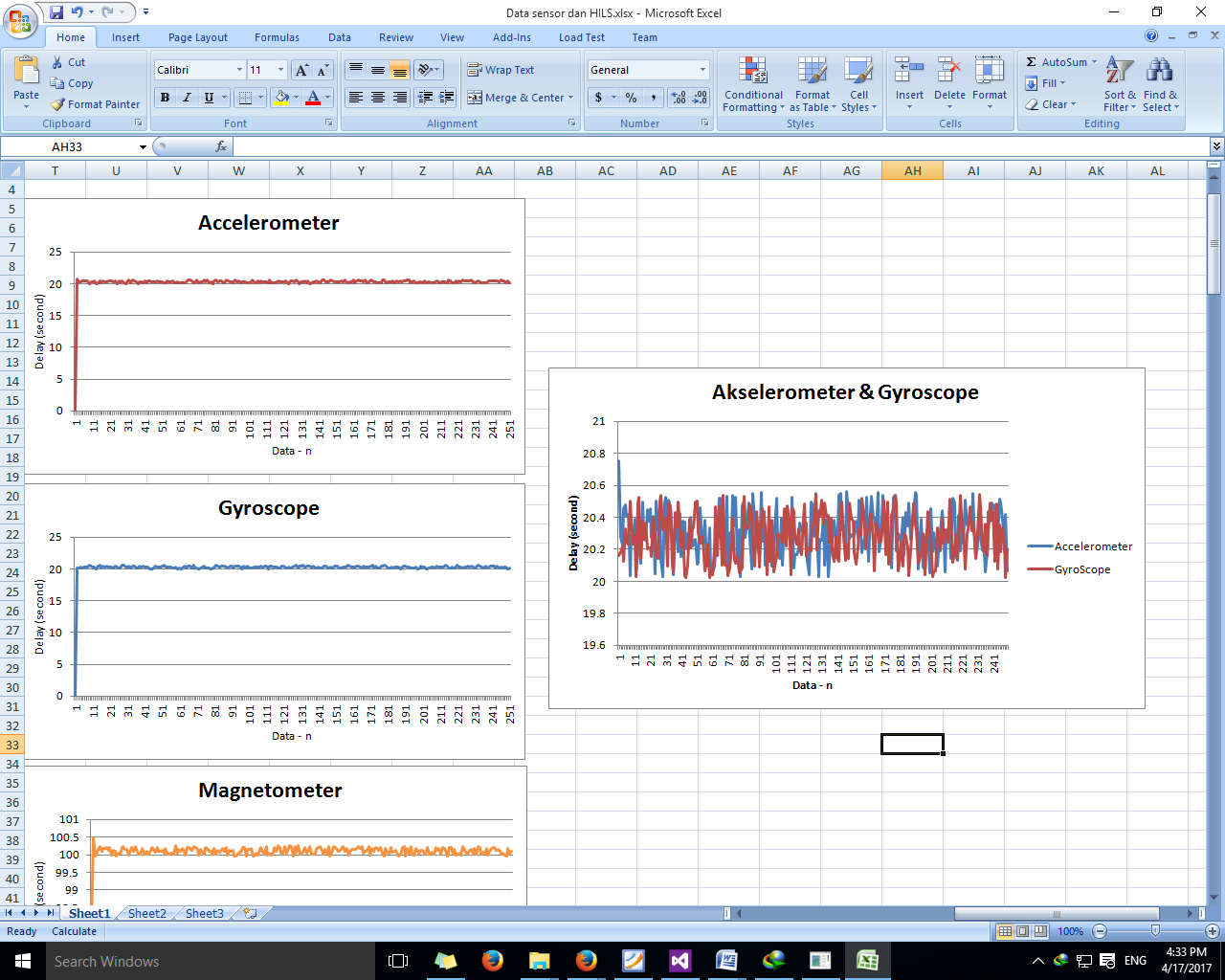


Figure 12. Transfer rate data accelerometer and gyroscope

According to Figure 12, the average transfer rate are

* Accelerometer : 20.306 ms
* Gyroscope : 20.279 ms

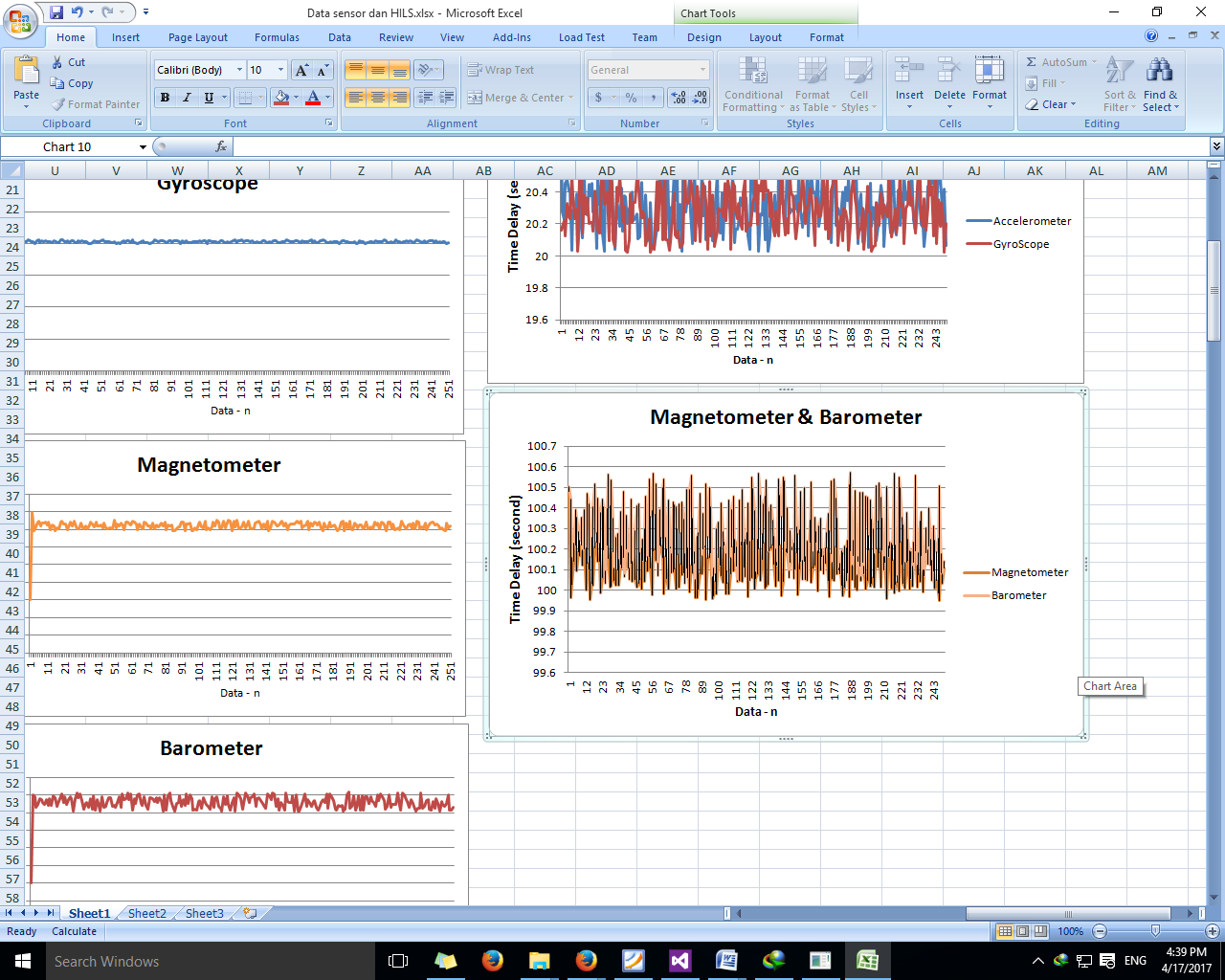


Figure 13. Transfer rate magnetometer and barometer

According to Figure 13, Average transfer rate

* Magnetometer : 100.108 ms
* Barometer : 100.297 ms
  1. **Validation simulated attitude data**

Validation data process will compare simulated sensory data and sensory hardware to determine diversity of sensory system. The good system will have little diversity result. Plant system will place on regulated control system to make sensory hardware will be moved to follow the behavior of plant on software simulator.



Figure 14. Controller holder plant

Following Figure 14, regulated control will have role as holder plant to imitate pitch (θ) condition of plant on software simulator. Pitch angle is considered important because it has important role in autonomous take-off, landing and altitude control. We realize the holder plant as indicated in Figure 15.

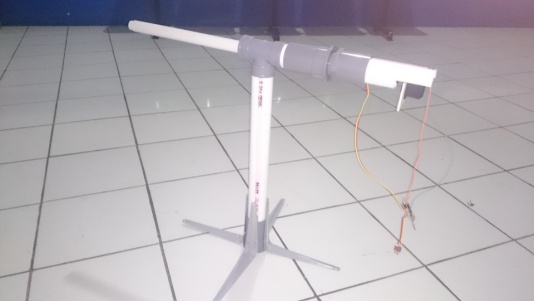


Figure 15. Implemented Holder

The holder plant will use control mechanism to create pitching motion by controlling pitch angle the sensory hardware. Figure 16 shows the experimental results.

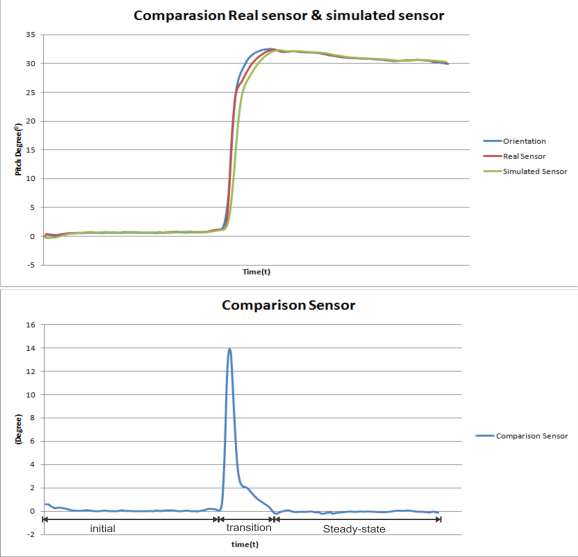


Figure 16. Comparison sensory data

with error data on steady state condition

* 1. **Comparison of control response.**

Comparison of control response between simulated sensory data from software simulator and hardware controller using holder plant will determined by divergence of rise time and steady state error.

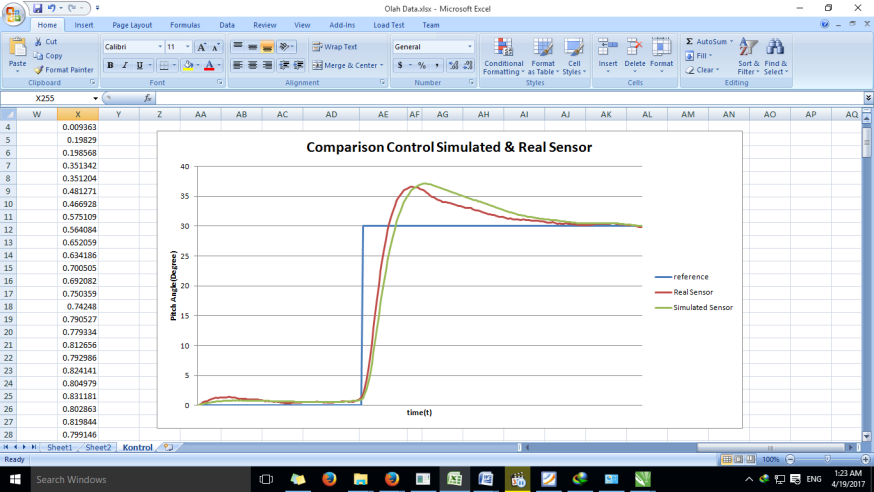


Figure 17. Comparison control response

According to Figure 17, we have experiment result that

Response control with simulated sensory

* Rise time : 1.998 s.
* Steady-state error : 0.397 %.

Response control with sensory hardware

* Rise time : 1.849 s.
* Steady-state error : 0.858 %.

Divergence response

* Rise time : 0.149 s.
* Steady-state error : 0.461 %.

1. **Conclusion**

According to the experimental results, we observe that

* Transfer rate data through middleware system can be fulfilled the demand sensory data for hardware controller.
* According to devised specification system, sensory data validation can achieve occurrence error less than 2O
* Rise time of control PID between simulated sensory data and sensory hardware has divergence less than 1 second and steady-state error of each sensory data less than 2%.

So, we can conclude that design process of HILS system give real-time simulation which is nearly convergent to real-time situation because it is meet the design specification.

1. **Acknowledgement**

The authors acknowledge research funding from 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).

**REFERENCE**

1. In-Gyu Jang, Inkeun Seo, Jaewook Jeon and Sung-Ho Hwang, "Testing 32-bit embedded system using hardware-in-the-loop-simulation of automatic transmission," 2007 International Conference on Control, Automation and Systems, Seoul, 2007, pp. 399-403.
2. N. P. Mahalik and K. Kim, "A Prototype for Hardware-in-the-Loop Simulation of a Distributed Control Architecture," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 38, no. 2, pp. 189-200, March 2008.
3. H. Y. Irwanto, "HILS of auto take off system: For high Speed UAV using booster rocket," 2016 International Seminar on Intelligent Technology and Its Applications (ISITIA), Lombok, 2016, pp. 373-380.
4. J. Park, B. Wang, J. Jeon and S. H. Hwang, "Hardware in-the-loop simulation for ABS using 32-bit embedded system," 2011 11th International Conference on Control, Automation and Systems, Gyeonggi-do, 2011, pp. 575-580.
5. A. Nurhafid, B. R. Trilaksono, R. A. Sasongko, A. Riyadl and H. Y. Irwanto, "Modelling and simulation of RKX200 rocket flight dynamics," 2013 3rd International Conference on Instrumentation, Communications, Information Technology and Biomedical Engineering (ICICI-BME), Bandung, 2013, pp. 230-236.
6. M. Hu, G. Zeng, and Y. Gao, “Thruster Control for Micro-satellite Attitude and Hardware-in-the-loop Demonstration”, 2012 *IEEE International Conference on Industrial Control and Electronics Engineering,* 2012, pp. 134-142.
7. Demonstration D. Joglekar, L. Chandini and P. Halder, "Autopilot design for aerospace vehicle using GUI — A user friendly approach," 2013 Annual IEEE India Conference (INDICON), Mumbai, 2013, pp. 1-6.
8. Hardware-In-the-Loop Simulation of an Attitude Control with Switching Actuators for SUAV.
9. A. Alsaraj and G. Stuffle, "Investigation of hardware-in-loop simulation (HILS) for guidance system," 2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, 2015, pp. 704-708.
10. Y. W. Hadi and R. T. Bambang, "Development of hardware-in-the-loop simultion for rocket guidance system," 2015 International Conference on Electrical Engineering and Informatics (ICEEI), Denpasar, 2015, pp. 229-234.
11. A. Ingalalli, H. Satheesh and M. Kande, "Platform for Hardware In Loop Simulation," 2016 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), Anacapri, 2016, pp. 41-46.
12. W. Premerlani and P. Bizard, ‘‘Direction cosine matrix IMU: Theory,’’ in Proc. *DIY DRONE USA*, 2009.
13. M. Euston, P. Coote, R. Mahony, J. Kim and T. Hamel, "A complementary filter for attitude estimation of a fixed-wing UAV," 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, 2008, pp. 340-345
14. G. Baldwin, R. Mahony, J. Trumpf, T. Hamel and T. Cheviron, "Complementary filter design on the Special Euclidean group SE(3)," 2007 European Control Conference (ECC), Kos, 2007, pp. 3763-3770.