Perspectives in Communication, Embedded-Systems and Signal-Processing (PiCES) – An International Journal
ISSN: 2566-932X, Vol. 2, Issue 8, November 2018
Proceedings of National Conference on Emerging Trends in VLSI, Embedded and Networking (NC-EVEN 18), April 2018

# Model Based Design Development and Testing

Harshita K, Ankitha B S, Shivputra
Department of ECE, Dr.Ambedkar Institute of Technology, Bengaluru, India

***Abstract: Hardware-in-the-loop (HIL) simulation is a technique that is used in the development and test of complex process systems. HIL simulation provides an effective platform by adding the complexity of the plant under control to the test platform. The complexity of the plant under control is included in test and development by adding a mathematical representation of all related dynamic systems. These mathematical representations are referred to as the plant simulation.***

***Keywords: HIL; EHSV; Plant Simulation; Mathematical Model; Controller***

## I. INTRODUCTION

The Hardware-in-the-Loop process has existed for no more than 15 to 20 years. Its roots are found in the Aviation industry. The reason the use of a HIL process is becoming more prevalent in all industries is driven by two major factors: time to market and complexity. Hardware-In-the- Loop is a form of real-time simulation. Hardware-In-the- Loop differs from real-time simulation by the addition of a real component in the loop. This component may be an Electronic Control Unit (ECU). The purpose of a Hardware- In-the-Loop system is to provide all of the electrical stimuli needed to fully exercise the ECU. In this way you fool the ECU into thinking that it is indeed connected to a real plant. The HIL simulation includes a mathematical model of the process and a hardware device/ECU you want to test, e.g. an industrial PID controller we will use in our example. The hardware device is normally an embedded system.

## II. OBJECTIVE AND OVERVIEW

The objective of HIL simulation is to provide an effective platform for developing and testing real-time embedded systems, often in close parallel with the development of the hardware. Software development no longer needs to wait for a physical plant in order to write and test code. In this project, we simulate the given Plant Model of Electro-Hydraulic Servo Valve (EHSV) and design a Controller for the same and verify it.

## III. PLANT MODEL

A plant in control theory is the combination of process and actuator. A plant is often referred to with a transfer function (not uncommonly in the s-domain) which indicates the relation between an input signal and the output signal of a system without feedback, commonly determined by physical properties of the system. An example would be an actuator with its transfer of the input of the actuator to its physical displacement. In a system with feedback, the plant still has the same transfer function, but a control unit and a feedback loop (with their respective transfer functions) are added to the system. In this paper, we have considered the Plant to be EHSV (Electro Hydraulic Servo Valve) and its plant model is shown below.

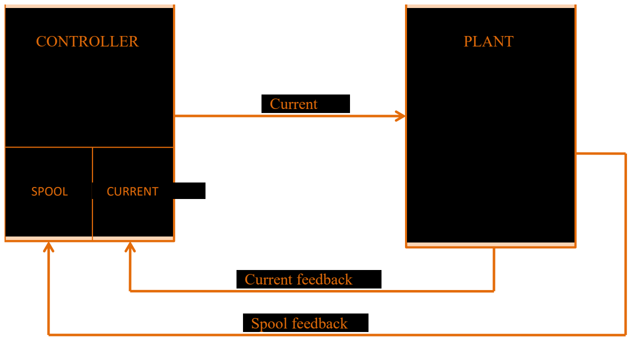



Fig 1. Functional Block Diagram

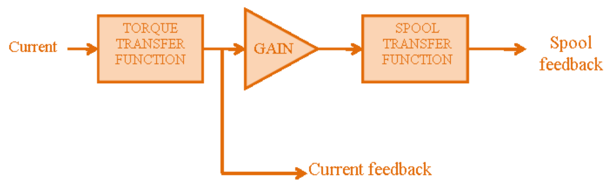



Fig 2. Plant Model

## IV. CONTROLLER

A proportional integral derivative controller (PID controller) is a control loop feedback mechanism control technique widely used in control systems. A PID controller continuously calculates an error value e(t) as the difference between a desired setpoint and a measured process variable and applies a correction based on proportional, integral, and derivative terms. PID is an initialism for Proportional- Integral-Derivative, referring to the three terms operating on the error signal to produce a control signal. The theoretical understanding and application dates from the 1920s, and they are implemented in nearly all analogue control systems; originally in mechanical controllers, and then using discrete electronics and latterly in industrial process computers. The PID controller is probably the most-used feedback control design. If u(t) is the control signal sent to the system, y(t) is the measured output and r(t) is the desired output,

$$e(t) = r(t) - y(t) \quad (1)$$

e(t) is the tracking error, a PID controller has the general form

$$u(t) = K_P e(t) + K_I \int e(\tau) d\tau + K_D \frac{de(t)}{dt}. \quad (2)$$

The desired closed loop dynamics is obtained by adjusting the three parameters Kp, Ki and Kd, often iteratively by ”tuning” and without specific knowledge of a plant model. Stability can often be ensured using only the proportional term. The integral term permits the rejection of a step disturbance (often a striking specification in process control). The derivative term is used to provide damping or shaping of the response.

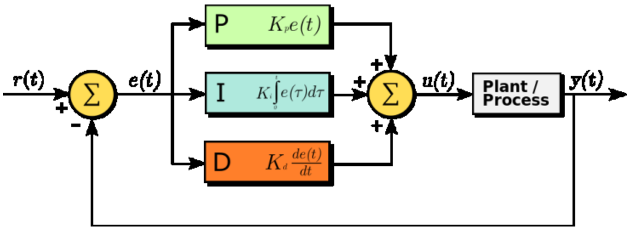
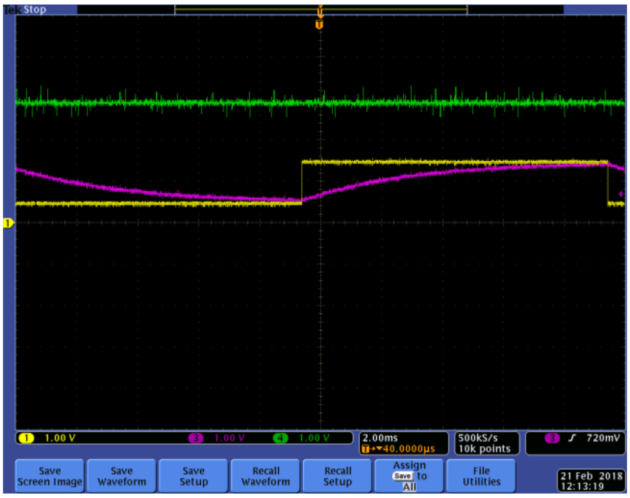



Fig 3. PID Controller

## V. VALIDATION



This is the output when no PI Controller was integrated with the Plant Model ( see Fig 4).

The signal in yellow is the Square input of 1V and 50Hz. The signal in purple is the current feedback from the Plant and the signal in green is the Spool feedback from the Plant.

This is the output of Current loop when the PI controller was integrated with the Plant.

The signal in yellow is the input of 1V and 50Hz and the signal in blue is the current feedback with Controller action.

This is the output of Spool loop when PI Controller was integrated with the Plant model.

The signal is yellow is the input of 1V and 50Hz. The signal in purple is the Spool feedback and the signal in blue is the current feedback.

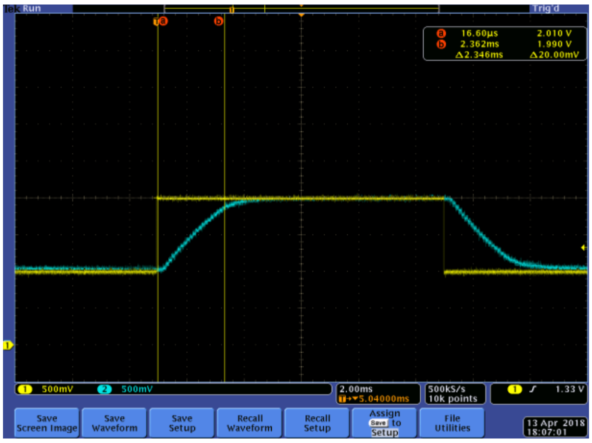


Fig 4. Plant output with Controller of current loop

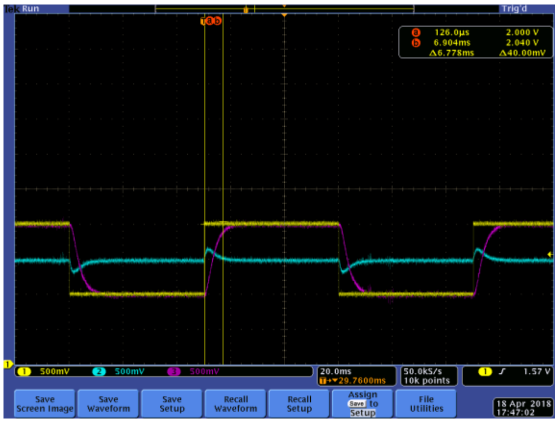


Fig 5. Plant output with Controller of Spool loop

## VI. CONCLUSION

This paper concludes that for the testing of Actuators, Physical Components are not necessary. The Mathematical model is derived for each physical components and dumped into the microcontroller. By designing the Controller to control the Plant, functionality of the Physical components can be determined. Once this is done, the Controller is then integrated with the Actuator.

## REFERENCES

[1] Chandler, David and Vallino, James, "Control system plant simulator: A Framework for hardware-in-the-loop simulation" (2008).

[2] Hardware-In-Loop Simulation, Wikipedia

[3] EHSV, Wikipedia