To Perform SIL And PIL Testing on Fast Dynamic System using Economical AVR Controller

Ashish V. Patel[1] Prof. Keerti Vashishtha[2]

[1]Student, Dept. of Instrumentation And Control, L. D. College of Engg., Ahmedabad, Gujarat, India
[2]Prof. Dept. of Instrumentation And Control, L. D. College of Engg., Ahmedabad, Gujarat, India

Abstract— Embedded Systems is indubitable the core of the current electronic era. From simple calculators to high-end control of the spacecraft it plays the key role. With development in embedded system, it's also used extensively in control domain. Applying Rapid Control Prototyping (RCP) methodology in controller development in model based design shorten the development period and provide higher accuracy compared to normal flow. Here we provide the economical way to perform the SIL (software in the loop) and PIL (processor in the loop) testing with low cost AVR controller compare to high-end expensive modules available in market. This paper focuses on performing SIL and PIL testing of DC motor with Arduino having Atmega328 microcontroller.

I. INTRODUCTION

With the development of semiconductor industries and with Moore's law, the hardware in embedded system becomes more complex with time. Now to extract every capability from controller or processor require the detailed understanding of architecture and with numbers of vendors this varies and learning for all controllers it an impossible task. But with hardware the Computer Aided Control System Design (CACSD) like MATLAB, LABVIEW develop into great extent.

Traditional controller development method has separated processes such as controller design, implementation, test and verification [1]. The whole process has to be started over again when an error or deviation occurs, as all processes are separated its sometime not possible to find the origin of the problem, making the controller development process costly and time consuming.

 With RCP methodology, which include MIL, SIL and PIL, all the process development integrated with each other allowing engineers to quickly test and iterate their control strategies with real hardware (controller). RCP methodology is highly used in Model based design as it eliminates potential bottlenecks, accelerate the development, and save cost and time [2]. It has been widely used in application like anti-lock braking, flight control, servo-control, vehicle stability and medical device development [2].

The Dynamic system chosen for the test is PMDC motor. DC Motor Speed Control is one of the benchmark problem and used here for implementing RCP methodology. DC motors are widely used in industry because of its low cost, less complex control structure and wide range of speed and torque. The application includes battery powered device like wheelchairs and power tools and door openers [6]. But tuning the controller directly on plant or on its proto-type may cause damage because it may lead to unsafe operating point. So with plant parameter we can design controller and simulate it to verify its behavior to all possible point without any risk.

The major advantage of RCP over the traditional method is that once the controller is designed and verified in simulation environment, we need not to develop software/code for controller in C/C++ or any other language with Auto code generation facility code for control strategy are generated automatically [3]. It saves a lot of time and useful to industries because it reduces the time-to-market which is one of the most important factors considering the high competition in this era.

We have used ARDUINO UNO, containing AVR At mega 328, which is an open source hardware developed by Massimo Banzi and David Cuartielles, Italy [5].
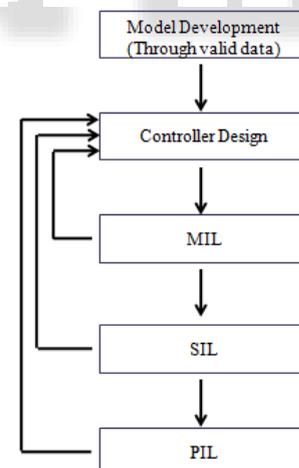
II. RCP METHODOLOGY

Fig.1: RCP Methodology

As RCP methodology applies to many fields of applications its structure gets modified accordingly but typical rapid control prototyping system is comprised of a math modeling program, a host computer, and a target hardware (controller) [4]. Here we have used MATLAB for modeling of the system. Below is the flow of RCP methodology,

Because of multiple feedbacks the error propagation from one stage to another is eliminated. From the valid data the

model of the system is developed in the model modeling environment i.e. software like MATLAB/SIMULINK. The controller is then designed depending on control strategy to be implemented. The controller is then tested in a simulation environment and this process is called MIL (Model in the Loop). After verified output and satisfactory behavior the C equivalent code for the controller is generated through Auto-code generation facility.

A. Software in the Loop (SIL)

 C code generated are need to be verified with a real controller design to verify the behavior of the generated code with real controller, thus eliminating error propagation to next stages. Below is a block diagram of SIL testing?
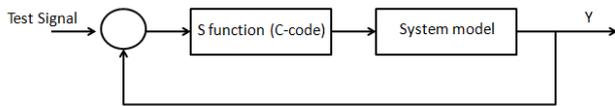

Fig.2: SIL Testing Block Diagram

The SIL testing is done by S-function generation from the controller in simulation environment. The S function represents the Generic C code generated; when we perform the SIL testing the S function block is executed by the local C compiler of the MATLAB [7]. Thus in SIL testing us basically works in two domains one C and other Simulink. The output of the simulation is compared with that of the MIL output and controller is redesign if both do not converge to predetermine tolerance. This allows the controller modification at an early stage of development and it also eliminates fix-point and floating point application error for algorithm implementation.

B. Processor in the Loop (PIL)

SIL verification is done with generic C code only, but ultimate application required running on the microcontroller. So the C codes are then compiled with the IDE compiler for this we need to integrate the IDE with MATLAB. Thus PIL provides a good verification of compiled object code on the target platform to ensure functional equivalence of code running on the target processor.
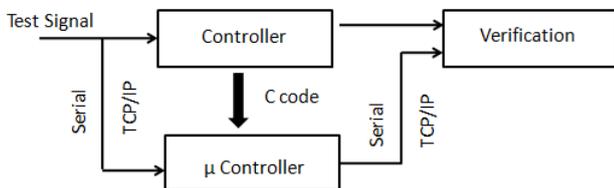

Fig.3: PIL Testing Block Diagram

As seen from block diagram, PIL testing is similar to SIL. The key difference is that, during PIL, code executes on the target processor. The communication between model and μ-Controller is achieved through serial or TCP/IP communication [7]. The advantages of the PIL testing that we can completely know the behavior of the code in an embedded environment. MIL, SIL testing is done by the processor of the computer which is either 32 or 64 bit but ultimate application is to be run in 8 bit μ-controller so

provide a better idea about the controller behavior and take necessary step before connecting it to real system.

III. MODEL DEVELOPMENT AND CONTROLLER DESIGN

DC motor is mechanically commutated electric motors powered from direct current [6]. System Model of a Permanent Magnet DC Motor with Speed Controlled by Armature Voltage can be obtained through Mathematical Model using Parameters given by Manufacturer Data-sheet. Here Typical Motor Data of standard Port-Escape motor is used.
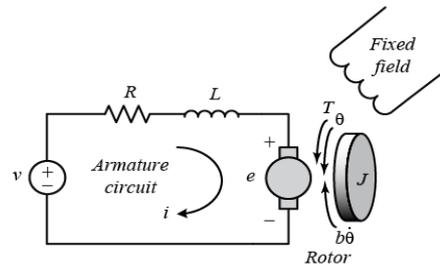

Fig.4: DC Motor Diagram [6]

From above figure based on Kirchhoff's voltage and Newton's law we get:

$$J\ddot{\theta} + b\dot{\theta} = Ki \qquad (1)$$

$$L\frac{di}{dt} + Ri = V - K\dot{\theta} \qquad (2)$$

Where, R= resistance, L=inductance, i= current,

$\dot{\theta}$ =angular velocity, b= back EMF. K=motor constant

These methods require a high understanding of mathematics and time consuming. We can also derive the model of DC motor with the help of MATLAB's SIMSCAPE where we need to place the component values of DC motor from data-sheet and thus provide high model fidelity and better understanding [7]. The model of motor is developed is shown below:
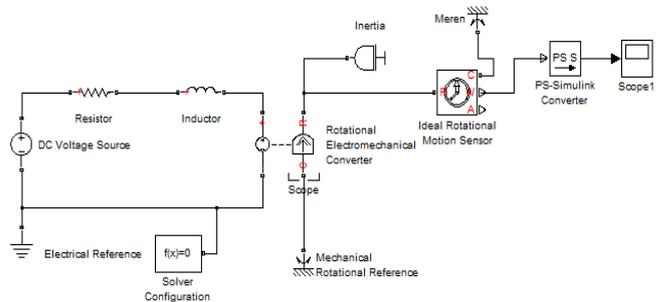

Fig.5: Sims cape Model of DC Motor

Output depends on component value we have selected. As we can see model is developed only by joining components together, No Mathematical equation is required for modeling.

A. Controller Development

After the Sims cape model has been developed, the PI control for speed control is developed through relay-feedback method. The Simulink model developed for relay-feedback method.
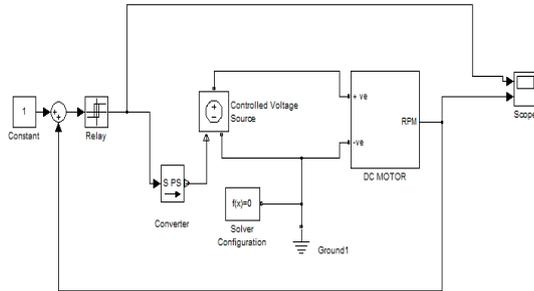

Fig.6: Sims cape Model for Relay Feedback Method

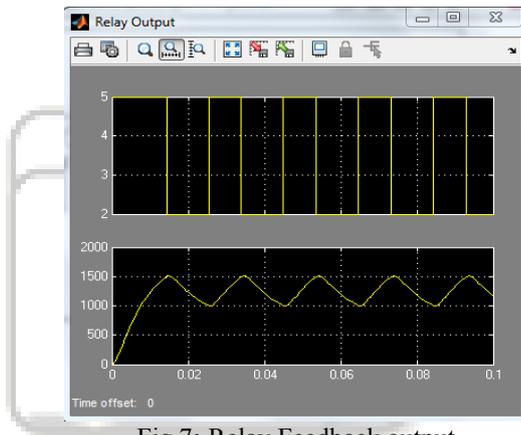From the output obtain we can find Critical period T_u Output of relay feedback is shown below.

Fig.7: Relay Feedback output

Upper window in scope is relay output while lower window shows system output. From equation:

$$K_u = \frac{4\,d}{\pi\,a} \qquad\qquad (3)$$

From K_u and The value of PI found from Ziegler-Nichols table [8].

IV. TESTING AND VERIFICATION

A. MIL Testing

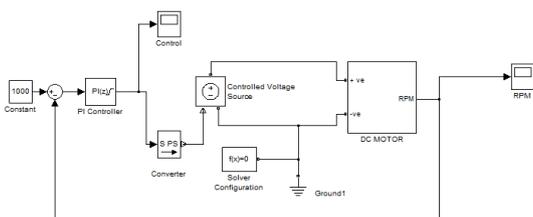The Simulink block diagram for MIL testing is shown in figure.

Fig.8: Simulink Model for MIL Testing

The controller output obtained after multiple iterations as PI obtain directly from Ziegler Nichols are not much accurate and require modification in gain values. For given set-point for 1000 RPM obtain shown.
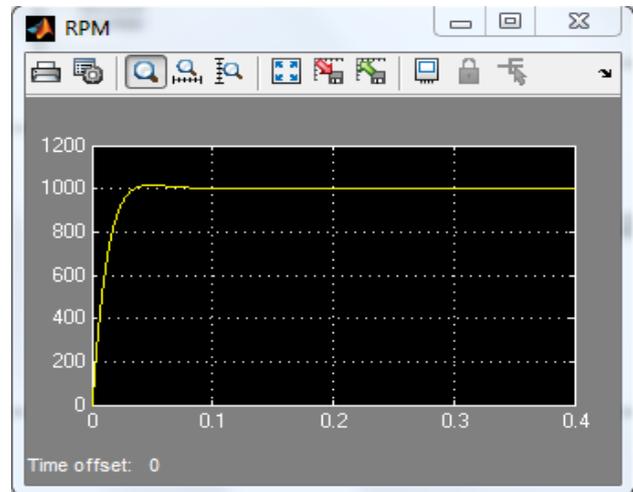

Fig. 9 Simulation of MIL testing

B. SIL Testing

As in RCP flow after MIL, SIL testing is done the Simulink block diagram for the SIL Testing shown.
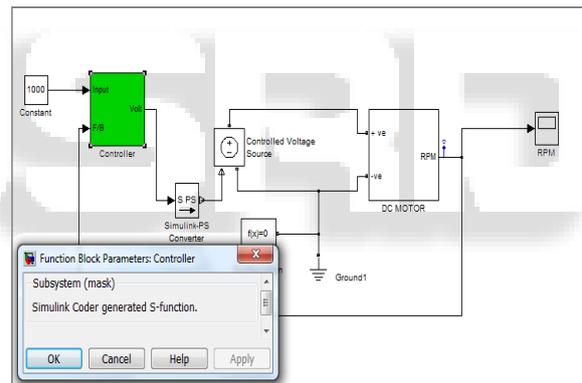

Fig.10 Simulink Block Diagram for SIL Testing

After the simulation is done the data is logged with help of data logger. With the help of Simulink data inspector the output of MIL and SIL is compared which is shown in Fig.11The Green Box indicated the S-function generated.
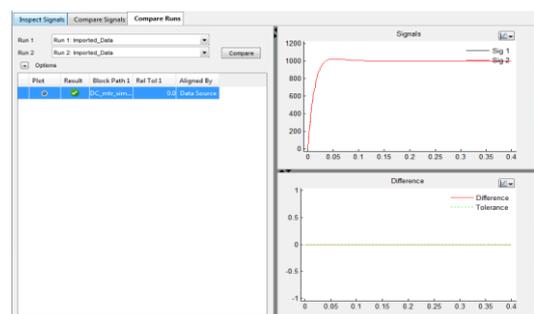

Fig. 11 Data Inspector Output

C. PIL Testing

As stated PIL test requires the embedded controller we have used Arduino Uno and integrate its IDE with

MATLAB. Arduino is an open source hardware board; it has Atmega328 AVR Controller which is an 8-bit µ controller with 16 MHz operational frequency [5]. Data from the Arduino is retrieved through serial communication. For serial communication the board has on board chip for serial communication capability.

PIL testing requires 2 Simulink programs:
1. At Tx end i.e. for Arduino, having control strategy and communication setup. Shown in fig. 12.
2. At host (Rx) computer having MIL tested controller and Arduino Tx-Rx block for data transmission
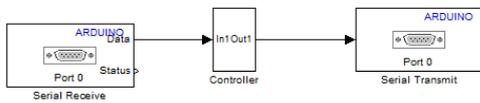

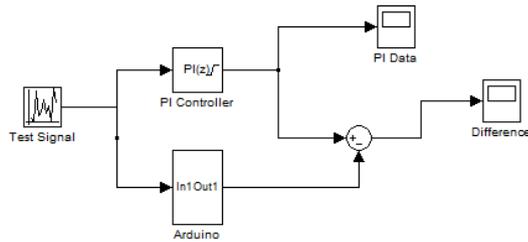Fig.12 Simulink Model for Arduino

Fig.13 Simulink Model for Host PC

For verification purpose the test signal is sent to both PI controller obtain after MIL and send to Arduino through the Tx / Rx block. Arduino process the data as programmed and send data back to PC.

These data are compared as shown in Fig.13. After executing above Simulink block diagram the output obtain is shown below.
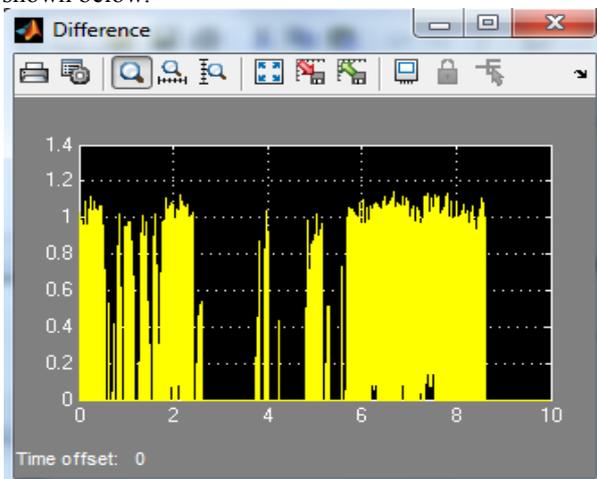

Fig.14 Difference between MIL and PIL Block

As seen above the data varies for maximum of 1.1 units, which used for PWM generation to control the motor, which is acceptable comparing high-end processor in PC v/s 8-bit µ controller. This error is also elevated by the fact that serial communications took place in uint8 only and not in float.

V. CONCLUSION AND FUTURE SCOPE

By employing RCP methodology, the code for complete embedded control is automatically generated. No hand-coding is required and generated code is verified with simulated output. The PIL testing ensures that the code generated is compatible and can be executed independently in AVR controller. The output obtained was within acceptable limits.

Because of low cost hardware it can be used by students at university to implement various control strategies for other benchmark problems like 2 tank system, inverted pendulum.

REFERENCES

[1] Junwon Jang; Choon Ki Ahn; Sekyung Han; Wook Hyun Kwon, "Rapid Control Prototyping for Robot Soccer System using SIMTool," *SICE-ICASE, 2006. International Joint Conference*, vol., no., pp.3035, 3039, 18-21 Oct. 2006
[2] PrecisionMBA, LLC. "Rapid Control Prototyping," Available: *http://www.precisionmba.com/rapid_control_prototyping.htm*
[3] Duma, R.; Petreus, D.; Sita, V. I.; Dobra, P.; Rusu, A., "Rapid Control Prototyping toolbox for Renesas M32C87 microcontroller," *Automation Quality and Testing Robotics (AQTR), 2010 IEEE International Conference on* , vol.1, no., pp.1,6, 28-30 May 2010
[4] Duma, R.; Dobra, P.; Abrudean, M.; Dobra, M., "Rapid prototyping of control systems using embedded target for TI C2000 DSP," *Control & Automation, 2007. MED '07. Mediterranean Conference on* , vol., no., pp.1,5, 27-29 June 2007
[5] Arduino datasheet. Available: *www.arduino.cc*
[6] DC-motor h*ttp://en.wikipedia.org/wiki/DC_motor*
[7] SIMSCAPE user guide from MATLAB
[8] *Kuo* BC, *Digital Control Systems*, Prentice-Hall, 1992
[9] Methoden,Anwendungen**,** *Rapid Control Prototyping,* Springer,2006.