AENSI Journals
Australian Journal of Basic and Applied Sciences
ISSN:1991-8178
Journal home page: www.ajbasweb.com

MATLAB - System Generator based Feedback Linearization and PID Control of Aero Thrust Pendulum using FPGA

[1]Sanjay Shreedharan, [2]Sibi Sankar, [3]Senthil Kumaran Mahadevan

[1]Undergraduate Student, SSN College of Engineering, Department of Electrical and Electronics Engineering, Chennai. India.
[2]Undergraduate Student, SSN College of Engineering, Department of Electrical and Electronics Engineering, Chennai. India.
[3]Associate Professor, SSN College of Engineering, Department of Electrical and Electronics Engineering, Chennai. India.

ARTICLE INFO
Article history:
Received 19 August 2014
Received in revised form
19 September 2014
Accepted 29 November 2014
Available online 15 December 2014

Keywords:
MATLAB – System Generator, Serial, Aero Thrust pendulum, PID, Anti Reset Windup, ADC, DRTSI

ABSTRACT
This paper presents a model based design and implementation of a closed loop PID controller for Aero Thrust Pendulum, using MATLAB - System Generator Toolbox, to demonstrate the Direct Real Time Simulation and Implementation (DRTSI) scheme. In general, simulation and implementation tasks are carried out in separate frameworks. The act of modelling and simulation are carried out in specialized numerical computing frameworks such as MATLAB/Simulink. Whereas, the system's real time control is implemented through an entirely different software framework, wherein the user is required to code in an entirely new paradigm, which is chip specific. The DRTSI scheme aims at modelling and implementation of a control system in an environment that provides a seamless transition between simulation and practical implementation. Software like d-Space, implements the same but at a reasonably high cost, wherein this scheme bridges the void between simulation and hardware implementation using an integrated system at a steeply lower cost. A method for live tuning of proportional, integral, derivative and system gain is validated and implemented to facilitate robust tuning. This serves to highlight the effect of each component of the PID controller on the response of the system for educational purposes. A conditional anti-reset windup circuit is implemented to counter the constant error integration by the integral controller. Also, Serial communication is established between FPGA and MATLAB/Simulink to facilitate extensive logging of the system's output and provides visual representation of the system's response for various gains.

To Cite This Article: Sanjay Shreedharan., Sibi Sankar., Senthil Kumaran Mahadevan. MATLAB – System Generator based Feedback Linearization and PID Control of Aero Thrust Pendulum using FPGA. ***Aust. J. Basic & Appl. Sci.,*** *8(18): 466-472, 2014*

INTRODUCTION

The major tasks in deploying robust real time control system are modelling, simulation and implementation. Seldom, do we see an implementation of an integrated environment that provides for all the three tasks in a cost effective manner. The MATLAB - System Generator Toolbox, provides a way of Direct Real Time Simulation and Implementation (DRTSI) using a Field Programmable Gate Array (FPGA), that provides low abstraction and modular design when compared with microcontrollers and is of lesser cost compared to d-Space. Apart from providing an integrated environment for development, the DRTSI scheme harnesses the parallel processing abilities of a FPGA providing the computational power required to solve complex tasks. The modular design environment of the MATLAB - System Generator Toolbox, enables incremental building of algorithms thereby facilitating reusability and rapid prototyping. The system chosen to demonstrate the DRTSI scheme is an Aero Thrust Pendulum. Section IV and V discusses the construction of the Aero Thrust Pendulum and the interfacing of the PMOD AD1 with the Spartan 3E FPGA Kit respectively. A modular one channel implementation of closed loop PID controller for the Aero Thrust Pendulum is shown in section VII, along with the implementation of an anti-reset windup circuit using system generator toolbox in section VIII. Section IX discusses the establishment of serial communication between FPGA Spartan 3E and the MATLAB/Simulink environment, for the purpose of data logging and visual representation of data. The results are discussed in section XI, followed by the conclusion in section XII.

Corresponding Author: Sanjay Shreedharan, SSN College of Engineering, Electrical and Electronics Engineering, Chennai, India.

II. System Generator– Matlab:

Xilinx System Generator is a MATLAB/Simulink-based design toolbox for Xilinx's series of FPGAs. The level of abstraction is very high with Microcontrollers and hence the difficulty of implementation increases either as the sophistication of the algorithm increases or when the hardware requirements become more demanding. GUI capabilities of System Generator, enables the overall design to be viewed in a modular manner with a low level of abstraction, which is seldom provided in any Microcontroller based development framework. The integration of System Generator with Simulink, makes the hardware design and verification to be performed within the same environment as that of the simulation model, reducing both the required design time and hardware resources (Monmasson and Cirstea, 2007).

III. Direct Real Time Simulation And Implementation Scheme:

Figure 1 shows the block diagram of the Direct Real Time Simulation and Implementation (DRTSI) scheme for the Aero Thrust Pendulum. The entire simulation and the direct implementation of the aero thrust pendulum is developed in MATLAB – System generator environment. The part of the simulation file is converted directly into a VHDL code using system generator toolbox which is used to program the FPGA Spartan 3E chip. The MATLAB file that is required for data logging is located in the personal computer. The FPGA controller communicates with the MATLAB/Simulink environment using the Serial communication.

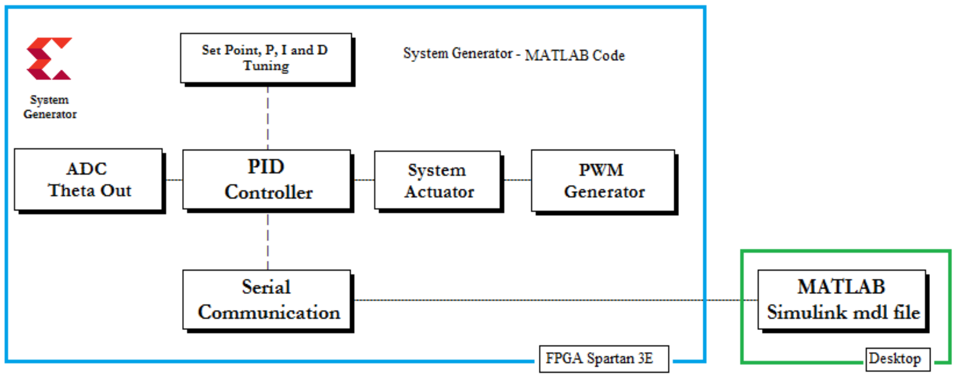

Fig. 1: Block diagram of the direct real time simulation and implementation scheme

IV. Hardware Description:

The system consists of a DC motor, the speed of which is controlled by a 20V Pulse Width Modulated (PWM) signal. It has got a two blade propeller, of 8-inch length attached to its shaft, as shown in Figure 2. The motor along with the propeller is attached to one end of a free hanging rod, the other end being connected to a servo pot, to ensure the rod moves perpendicularly to the axis of the servo pot. The output of the servo pot is connected to the PMOD AD1, which is a 12 bit ADC module (Eniko et al., 2011). The ADC is interfaced with the Spartan 3E FPGA Kit, which incorporates the PID control and generates the PWM signal for the speed control of the motor. The supply terminals of the motor are connected through the L298 H Bridge, which is fed by a PWM signal. The L298 consists of four MOS transistors, which allows bidirectional control. The peak current rating of L298 is 2A per channel, and peak voltage rating is 40V. Since, the maximum current that can be drawn by the motor is 3.5A, the two channels of H-Bridge are connected in parallel to satisfy current rating through division of current between the two channels.

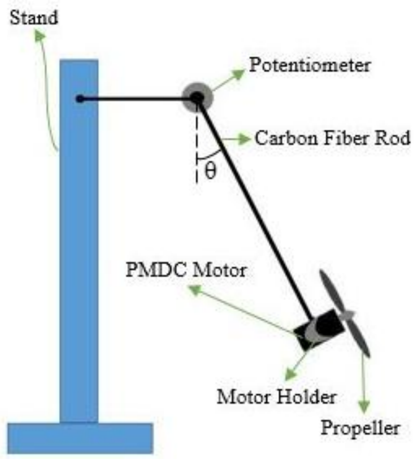

Fig. 2: Schematic of the Hardware Setup of the Aero Thrust Pendulum

V. Design Of Synchronisation Logic Circuit For Analog To Digital Converter:

The PMOD AD1 board comprises of two 12-bit ADCS7476MSPS Analog to Digital converter chips. It converts an analog signal to a 12-bit digital value in the range (0-4095). The digital conversion is based on the formula given in equation (1).

$$\text{Digital value} = \frac{\text{Input voltage}}{3.3\text{ V}} \bullet 4095 \qquad (1)$$

The essential parameters for achieving analog to digital conversion in AD7476 is listed below.

1. Timing waveforms ($\overline{\text{CS}}$, SCLK, SDATA), shown in Figure 3.
2. Maximum rate at which one data is sampled – 20 MHz. This is nothing but the frequency of ADC clock, SCLK which the user sets.
3. Dead time (t_{quiet}) – Minimum time required between bus relinquish and start of next conversion – 50 ns.
4. No. of clock cycles required for conversion – 16 SCLK cycles.

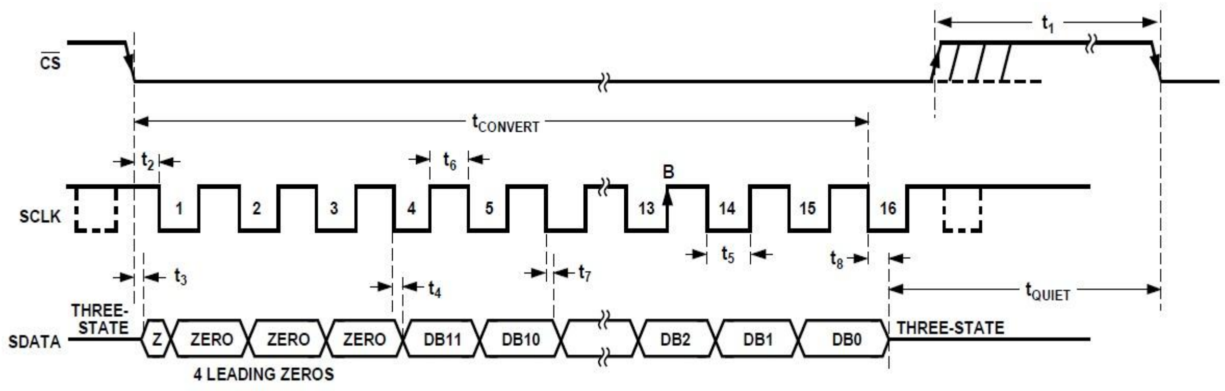

Fig. 4: AD7476 Serial interface timing diagram

The ADC pumps the 12-bit digital data, obtained using equation (1), in a serial fashion. Figure 4 shows the MATLAB - System Generator implementation of the chip select and the clock signal generation for the ADC (Sukhmeet and Parminder, 2012). The corresponding real time simulation output for the same is presented in the Figure 5. Figure 6 presents the serial to parallel data conversion results of the real-time simulation.

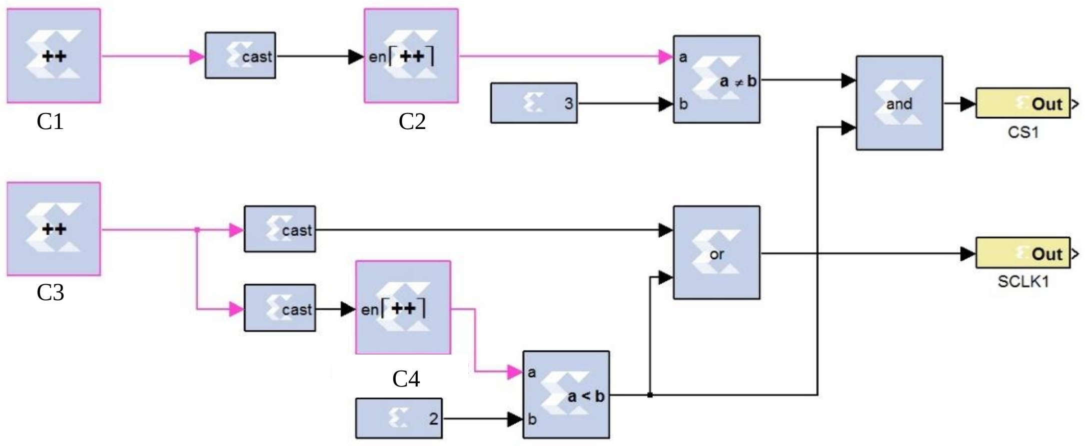

Fig. 5: Xilinx model to generate Chip Select & SCLK

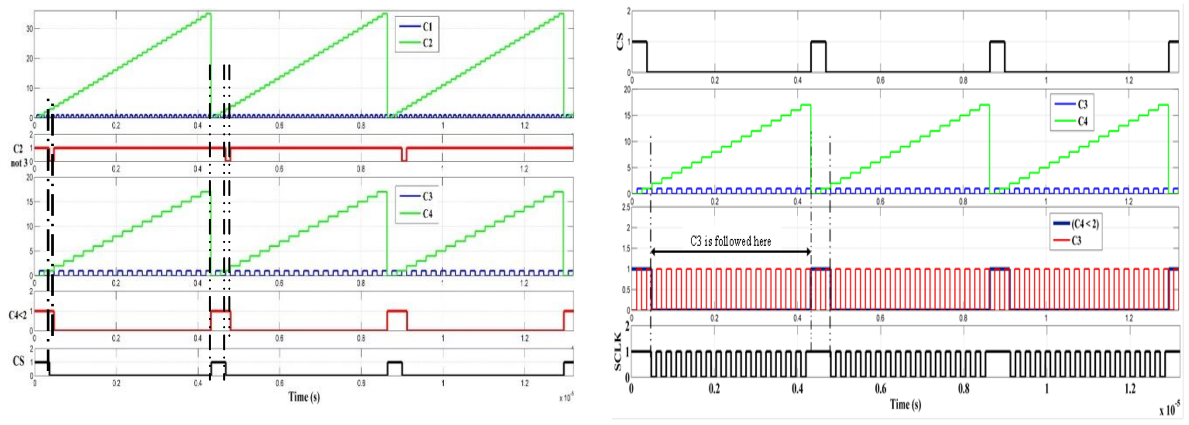

Fig. 5: Generation of Chip Select signal and SCLK signal

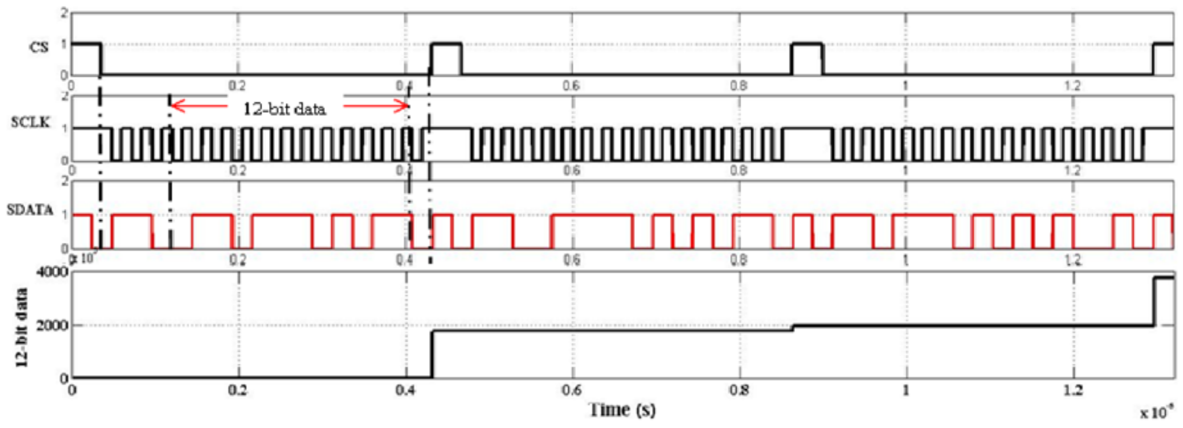

Fig. 6: Serial to Parallel data conversion of ADC

VI. System Modelling And Feedback Linearization:

Figure 7(a) discusses the open loop response of the system for different duty cycles. It was observed from the Figure 7 (a) that the system possesses a dead band. Dead band is the region where the system does not respond for any changes in the input of the system. The dead band was experimentally found to be 2.4V.

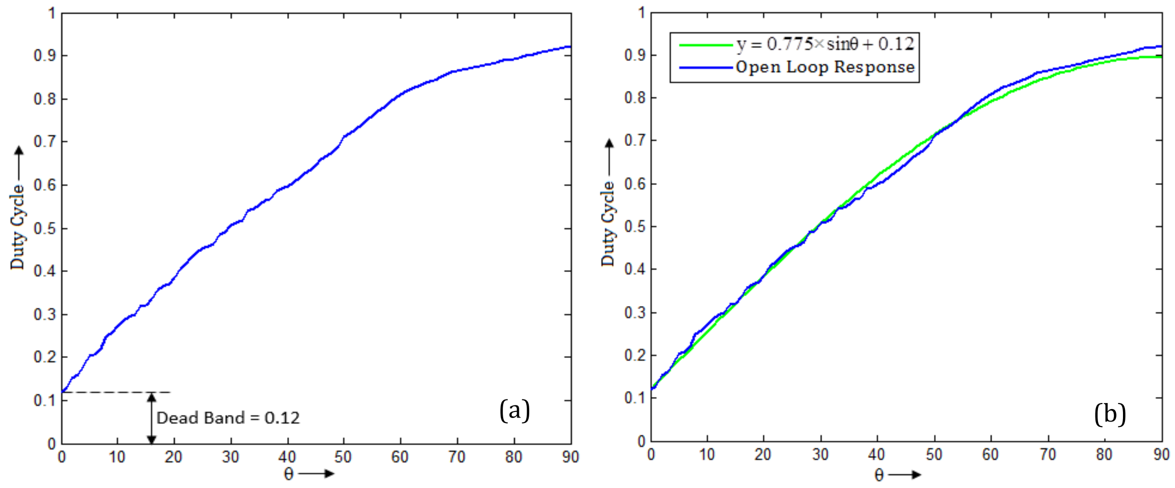

Fig. 1: (a) Open loop response, (b) Curve fitting of Open loop response

With the help of open loop response and curve fitting technique as shown in Figure 7(b), the system model was obtained and is given in equation (2).

$$\text{Duty Cycle} = K_s \sin\theta + DB \tag{2}$$

where, K_s is the System gain and DB is the dead band. The system gain, Ks was found to be 0.775 and the Dead band is 0.12. The control Law for the closed loop system which implements the feedback linearization scheme with the help of the system model described above is given in equation (3), where the input of the system is a function of feedback linearized output and the control signal (Eniko et al., 2010).

$$\text{Duty Cycle} = K_s \sin\theta + DB + K_p e(t) + K_i \int_0^t e(t)\,dt + K_d \frac{de(t)}{dt} \tag{3}$$

Figure 8 shows the implementation of feedback linearization using System Generator – MATLAB. It includes a saturation limiter which limits the duty cycle between the dead band and the stable operating region of the Aero Thrust Pendulum i.e. 70^0 which corresponds to a duty cycle of 0.8

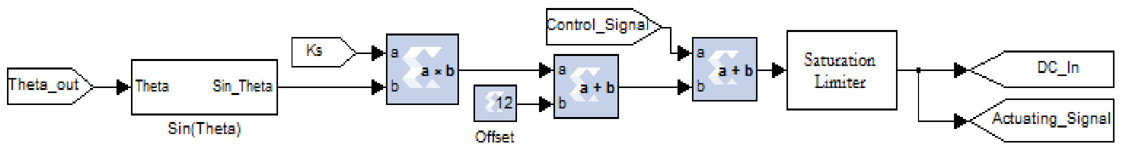

Fig. 8: System Generator - MATLAB implementation of Feedback Linearization Scheme

VII. Implementation Of Digital Pid Controller:

A Proportional Integral and Derivative (PID) controller is a closed loop feedback mechanism widely used to track the desired set point irrespective of the system dynamics, load disturbance and set point variations. The PID controller attempts to minimize the error by adjusting the actuating signal of the process through the use of a control variable as given in equation 4 (Kocur et al., 2014).

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\,dt + K_d \frac{de}{dt} \tag{4}$$

The proportional constant, K_p of the controller contributes to stability and is responsible for the middle range responsiveness of the system. The Integral gain, K_i of the controller is responsible for disturbance rejection. It contributes to tracking the reference, by reducing the steady state error to zero. The differential component, K_d contributes to the instantaneous responsiveness of the system. This, however can make the system susceptible to noise. The Figure 9 shows the discretized version of equation (4) implemented using MATLAB System Generator. The product of the K_p and error, e(t) accounts for the proportional component of the controller. The derivative component is discretized to a first order difference equation and then multiplied by K_d and a constant multiplier block of magnitude 5×10^7. Since, the FPGA operates at a frequency of 50 MHz, the time period of operation is $^1/_{50MHz}$= 20 ns. Hence, the rate of change of time is 20ns, and therefore, $^1/_{\Delta t}$ is $^1/_{20ns}$ = 5×10^7. The discrete approximation of the integral is obtained using the Riemann sum. The Riemann Sum was implemented by accumulation of the product of error and the integral constant, K_i at a rate of 20ns, which is represented using a constant multiplier of magnitude 20×10^{-9}. The problem of windup in the integral loop is addressed in the next section through the implementation of anti-reset windup.

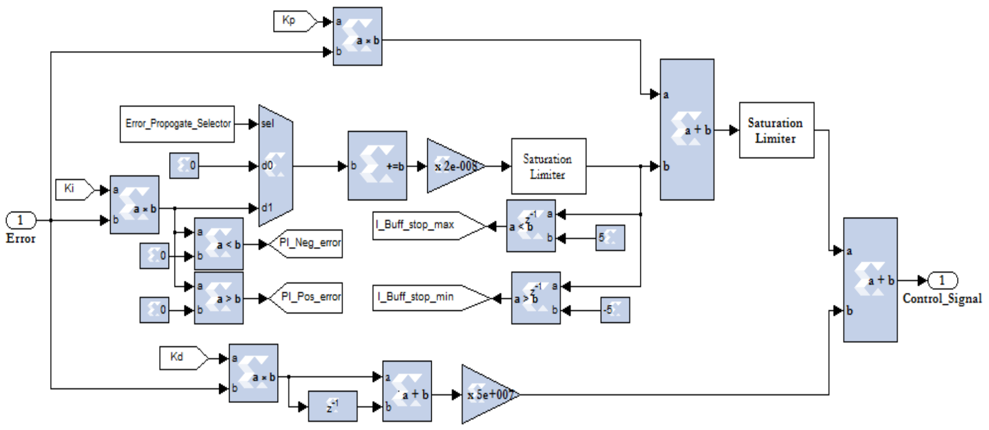

Fig. 2: Digital PID Controller

VIII. Anti-Reset Windup For Integral Control:

During practical implementation, a negligible error was found to exist even in the presence of PID controller. This led to a classic problem of perpetual integration or windup. Thus, increasing the error to values well beyond the saturation level. When a disturbance is suddenly introduced or if the set point is changed, a negative or positive error may be introduced. At this point, the integrator needs time to reduce from the very high value or increase from the very low value to which it has integrated, and then implement its action. During this period, a functional PD control and an unreliable integral control will be in action, which might induce oscillations in the system. This in turn might lead to instability. Traditional conditional anti-reset windup circuits are implemented using a saturation block to limit integration from perpetuating to very large values (Charaabi et al., 2002). A slightly modified version of the conditional anti-reset windup circuit was implemented in FPGA due to the clamping of the integral buffer on reaching saturation. To restore the integrating action of the controller, an Error Propagate selector was designed using a K-Map. The working of the Error Propagate Selector is shown in the tabular column.

Pos_Error	Neg_Error	I_buff_stop_max	I_buff_stop_min	Error Propagate Selector
Yes	No	Yes	No	0
Yes	No	No	No	1
No	Yes	No	Yes	0
No	Yes	No	No	1

When the error is positive, then two possible cases exist. The first, in which the integrator keeps on integrating, exceeding the Integral buffer maximum value and the other in which the integral buffer stays within Integral buffer's positive and negative limits. When the integrator keeps integrating and reaches the maximum value of the integral buffer, the Error Propagate Selector, produces an output of zero, which activates the 1st channel of the mux, and hence the present value is maintained without any increase as zero is being continually added each cycle. If the integrator stays within the limits of the positive and the negative error, then the Error Propagate Selector, gives an output of 1, which activates the 2nd channel of the mux and hence keeps integrating the value of the error. Similarly, when the error is negative, for negative saturation and intermediate case, 1st and 2nd channels are activated respectively. Figure 9 shows the System Generator implementation of the above logic. This prevents, the integrator integrating to an abnormally very high value or a very low value and hence makes the Integral Control to timely intervene in order to ensure the stability of the system.

IX. Serial Communication For Data Logging In Matlab:

Data logging is the act of retrieval and storage of essential data for visual representation and debugging purposes. Data logging was achieved through Serial RS-232 Communication protocol between the Host PC and FPGA Spartan 3E. Serial RS-232 Communication protocol was implemented in VHDL and imported as a black box in the MATLAB/System generator environment. The Host PC was equipped with MATLAB/Simulink mdl file. The Figure 10 show the basic MATLAB/Simulink blockset required for interfacing the serial output from the FPGA Spartan 3E at a baud rate of 115200. The port information and the baud rate are specified in the Packet Input block and the visual output is obtained through the Scope.

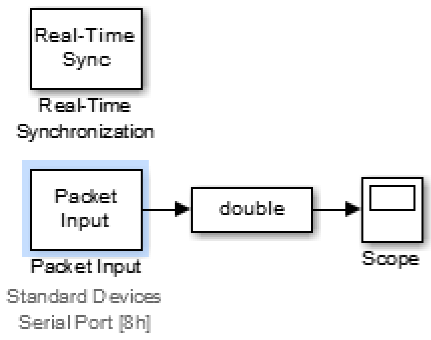

Fig. 3: MATLAB/Simulink mdl file

X. Experimental Setup:

The experimental setup consists of a free hanging carbon rod with a DC motor and a two blade propeller attached to it. The motor is supplied by a 30V, 4A variable power supply through a L298 H Bridge, which provides a voltage rating of 40V and a current rating of 2A per channel. The 500 KΩ servo pot is attached to the other end of the free hanging rod. The variable terminal of the servo pot is connected to the PMOD AD1, which is a 12-bit ADC. It possesses a built-in two 2-pole Sallen-Key anti-alias filter, which reduces the noise. The PMOD AD1 is interfaced with the Spartan 3E Nexys2, 500E FPGA Kit (Figure 11(a), (b), (c)).

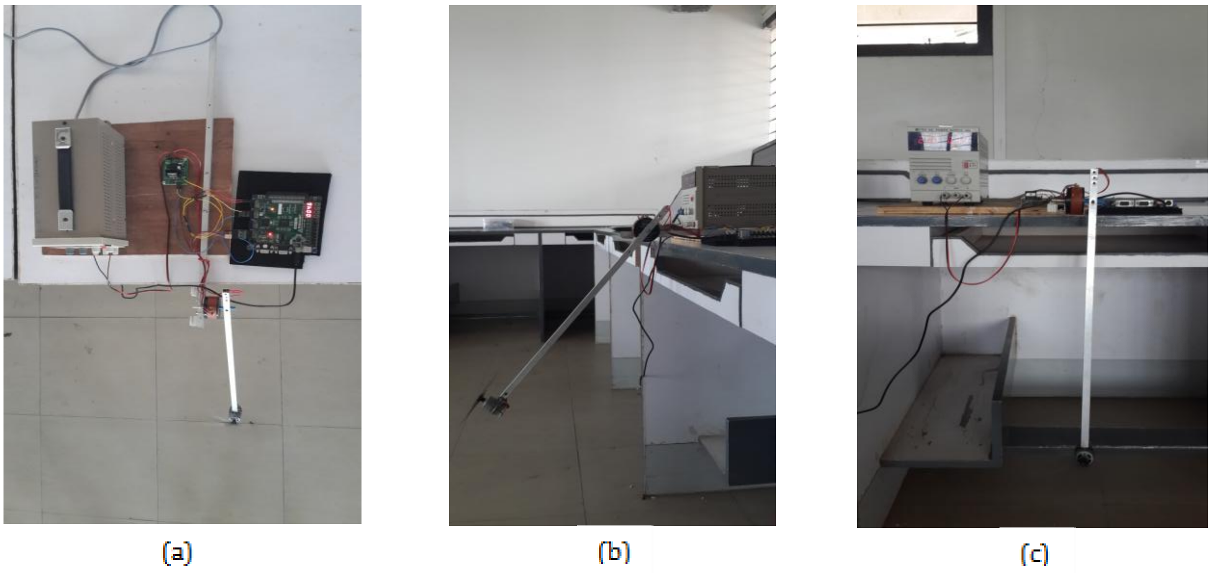

Fig. 4: (a) Top View, (b) Right Side View, (c) Front View of the Aero Thrust Pendulum Setup

XI. Results:

The data logging system implemented as described in section IX, enables the logging of the output response of the system. An online tuning method was implemented using the available push buttons on FPGA Spartan 3E for the purpose of increasing, decreasing and storing various gains. The 7-segment LED display available on the FPGA Spartan 3E provides the required visual aid for tuning purposes. Each gain was tuned separately to demonstrate its effect on the output of the system. The increase in K_p, decreases the rise time, increases the overshoot and decreases the steady state error. Hence, a trade-off must be made between the steady state error and overshoot while tuning K_p. K_i increases the overshoot but eliminates the steady state error. The appropriate value of K_i should be chosen to eliminate the steady state error and have minimum oscillations. K_d is tuned to dampen the oscillations due to K_p and K_i. The above stated theoretical inferences were demonstrated practically in the laboratory for learning purposes and the values of K_p, K_i and K_d were tuned.

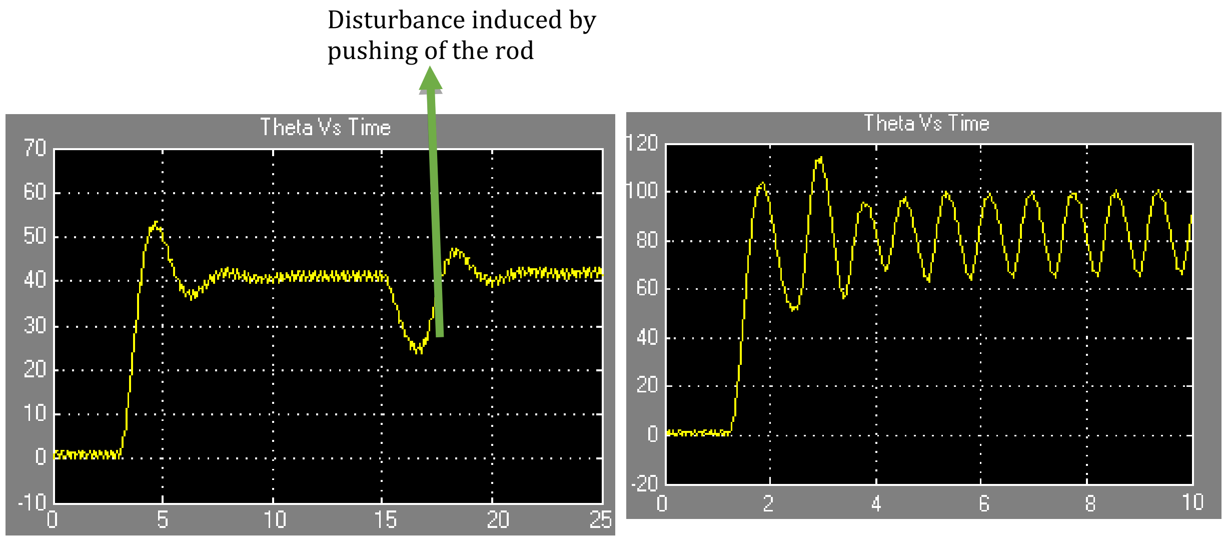

Fig. 12: (a) Closed loop response of the system for set point 400, (b) Unstable region of the system

As shown in Figure 12(a), for a set point of 40^0, the completely tuned PID controller enables the system to respond in a robust manner when subjected to disturbances. The system's response delay for switching between different set points is minimal due implementation of anti-reset windup in the integral loop of the PID controller. The response of the system is found to be unstable for set points greater than 80^0 as shown in Figure 12(b).

XII. Conclusion And Further Scope:

In this paper, the modelling of an Aero Thrust Pendulum and its PID controller is implemented in an environment that provides for the demonstration of the DRTSI scheme. The interfacing of servo pot using the PMOD AD1 with FPGA Spartan 3E and serial communication with the MATLAB/Simulink are discussed. The online tuning capabilities of the designed system are shown, enabling an efficient tuning environment. The results show the response of the system is robust to disturbances with instantaneous response due to the anti-reset windup circuit implementation. In order to make the system fully controllable over the complete range, compensators need to be designed.

REFERENCES

Charaabi, L., E. Monmasson and I. Slama-Belkhodja, 2002. Presentation of an efficient design methodology for FPGA implementation of control systems: Application to the design of an antiwindup PI controller. Proceedings of IEEE Industrial Electronics Society Annual Conference., 3: 1942-1947.

Eniko, T., Enikov., Giampiero Campa, 2012. Mechatronic Aeropendulum: Demonstration of Linear and Nonlinear Feedback Control Principles With MATLAB/Simulink Real-Time Windows Target. IEEE Transactions on Education., 55(4): 538-545.

Eniko, T., Enikov, Vasco Polyzoev and Joshua Gill, 2010. Low-Cost Take-Home Experiment on Classical Control Using Matlab/Simulink Real-Time Windows Target. Proceedings of the American Society for Engineering Education Zone IV Conference.

http://www.digilentinc.com/Pmods/Digilent-Pmod_%20Interface_Specification.pdf.

Jogalekar, K., A. Gunjal, D.N. Sonawane, 2013. Implementation of PID architecture in FPGA for DC motor speed control. International Conference on Circuits, Electronics and Communications (CCUBE). pp: 1-5.

Kocur, M., S. Kozak, B. Dvorscak, 2014. Design and Implementation of FPGA – digital based PID controller. 15th International Carpathian Control Conference (ICCC). pp: 223-236.

Monmasson E. and M.N. Cirstea, 2007. FPGA Design Methodology for Industrial control systems- A Review. IEEE Transaction on Industrial Electronics., 54(4): 1824-1841.

Sukhmeet, Kaur., Parminder, Singh, Jassal., 2012. Field Programmable Gate Array Implementation of 14 bit Sigma-Delta Analog to Digital Converter. International Journal of Emerging Trends and Technologies in Computer Science, 1(2): 229-232.

Swamy, T.N., K.M. Rashmi, 2013. Data Acquisition system based on FPGA. International Journal of Engineering Research and Applications., 3(2): 1504-1509.