International Journal of u- and e- Service, Science and Technology
Vol.8, No.3 (2015), pp.11-28
http://dx.doi.org/10.14257/ijunesst.2015.8.3.02

Research on FPGA-Based Controller for Nonlinear System

Farzin Piltan, Maryam Rahmani, Meysam Esmaeili, Mohammad Ali Tayebi, Mahsa Piltan Hamid Cheraghi, Mohammad R. Rashidian and Arzhang Khajeh

Intelligent System and Robotic Lab, Iranian Institute of Advance Science and Technology (IRAN SSP), Shiraz/Iran
Email: piltan_f@iranssp.com, www.iranssp.com/english

Abstract

Many of linear control applications require real-time operation; higher density programmable logic devices such as field programmable gate array (FPGA) can be used to integrate large amounts of logic in a single IC. This work, proposes a developed method to design PD controller (PDC) with optimal- gains using FPGA. The method used to design PD controller is to design it as digital design Proportional and Derivative controller in parallel through the summer. The proposed design is 32-bits FPGA-based controller (32PDC), which uses 32-bits for each input/output variable. The single joint of robot is used to test the controller in simulation environments, using VHDL code for the purpose of simulation in Xilinx. The same design is coded in MATLAB environment (MPDC) in order to make a comparison with the proposed FPGA-based design. PDC needs 16 clock cycles to complete one action with maximum frequency of 108.5 MHz. 32PDC is able to produce an output in 13.24 MHz with the robot system. Therefore, the proposed controller will be able to control a wide range of the systems with high sampling rate and 75.545 ns delays.

Keywords**:** *real-time operation, Field Programmable Gate Array (FPGA), Proportional and Derivative control, robot manipulator, VHDL, Xilinx, sampling time*

1. Introduction

The controller is a device which can sense information from linear or nonlinear system (*e.g.*, robot manipulator) to improve the system performance [1]. The main targets to design control systems are stability, good disturbance rejection, and small tracking error [2]. Several industrial robot manipulators are controlled by linear methodologies (*e.g.*, Proportional-Derivative (PD) controller, Proportional- Integral (PI) controller or Proportional- Integral-Derivative (PID) controller) and realize them as a computer program. However, most of linear control applications require real time operations with high speed constrains. Therefore, the common method cannot be considered as a suitable solution for these type applications. Higher density programmable logic devices such as FPGA can be used to integrate large amounts of logic in a single IC. FPGA provide additional flexibility than ASIC, and they can be used with tighter time-to-market schedules [3].

A Field Programmable Gate Array (FPGA) is similar to a PLD, but whereas PLDs are generally limited to hundreds of gates, FPGAs support thousands of gates. They are especially popular for prototyping integrated circuit designs. Once the design is set, hardwired chips are produced for faster performance. Field Programmable Gate Arrays (FPGAs) are divided into two major categories:

- SRAM-based FPGA
- Antifuse-based FPGA

Despite the much advancement in dedicated digital control processing (DCP) microcontrollers, the efficiency with which any FPGA hardware implementation is able

ISSN: 2005-4246 IJUNESST

to execute the long standing metric of multiply and accumulate still remains to be the limiting factor in overall performance. The specialized and constrained RISC-like instruction sets that give FPGA's such as Xilinx instruction deep SPARTAN 3E family their increased efficiency, oddly enough also lead to their performance ceilings for FPGA applications. First, it is incredibly difficult to produce efficient assembly and machine code for these specialized instruction sets, and thus FPGA's require a powerful, specially optimized compiler (typically C). Second, and most important, is their inability to do multiply and accumulate functions in parallel. Despite the 8 instruction deep pipeline, the fact still remains that the data flow must pass through a pipeline that is only capable of completing one multiply and accumulates function every few clock cycles at best. This translates to calculating one sample of a 512-point Fast Fourier Transform (FFT) every several thousands to 10's of thousands of clock cycles, depending on the architecture [1]. These two limitations led designers in the early 1990's to look for other alternatives that might alleviate the overhead of and dependency of the compiler, and offer a more parallel approach. At first, this search led designers down the path of Systems on a Chip (SOC) design. SOC's are very dense and fast Application Specific Integrated Circuits (ASIC's) that contain a central processor hard core (ex. IBM 405 PPC), surrounded by soft-core algorithm accelerators, memory, soft-core peripherals, and I/O interfaces all attached to a central bus. These chips are expensive, designed for large production quantities, and are one-time-programmable. An example of a current large SOC market is the second generation cellular phone.

Around the same time as the move to ASIC's, Field Programmable Gate Arrays (FPGA's) emerged in the engineering prototyping and emulation world. FPGA's are a reprogrammable, SRAM based, VLSI platform that allow a digital hardware designer to almost instantly (compared to ASIC's) upgrade a design. FPGA's lose their memory when power is removed from the circuit, and thus must be programmed upon each power up, either through JTAG or a PROM. FPGA's contain internal units such as built in memory, registers, and multiplier blocks. As the number logic resources in FPGA's grew, DSP designers began to take advantage of the vast amount of parallelism offered by such re-programmable chips, at the cost of reduced clock speed in comparison with current dedicated DSP microprocessors. By 2000, FPGA's began to approach the gate density of ASIC's, and make significant increases in maximum clocking frequencies. However, ASIC's are still superior for overall density and speed. Nevertheless, the vast parallelism allowing for the execution of hundreds of multiply and accumulate (MAC) operations in parallel, combined with nearly instant re-programmability, steered the implementation of computationally complex DSP algorithms down the path of FPGA's. Many companies specializing in FPGA algorithm design and implementation utilize FPGA's as their primary means of prototyping. Often times they are utilized as the final design implementation choice over an ASIC platform in situations where power consumption is not a major concern. This is most evident in companies such as Lockheed Martin who develop military applications that require cost effective rapid prototyping platforms that can be highly integrated with FPGA software development. One of the most world-renowned software tools for FPGA algorithm design is Matlab, produced by Mathworks. Many companies currently realize all of their high level FPGA modeling in Matlab, including all of the necessary test harnesses. They then convert these algorithm models to a hardware description language, such as Verilog or VHDL, either manually or by using some algorithm specific conversion software. From here, the normal design flow is followed to synthesize and place and route the algorithm into a target FPGA of choice. The original test vectors used to test the algorithm in Matlab are generally applied in some fashion to test the algorithm in the actual hardware. Therefore, printed circuit boards (PCB's) housing the FPGA's must be designed with interfaces that will allow data to be transmitted to and from the PCB via a PC running Matlab. This interface requires additional hardware and software overhead for testing. The bottom line is that today's

FPGA algorithm and hardware designers have a desire and need to begin designing with Matlab, target FPGA hardware, and then complete their verification by reading test vectors back into Matlab from the real hardware to compare against their original software algorithm results. In 2001, one of the leading FPGA companies, Xilinx, teamed up with IBM PowerPC (PPC) ASIC designers to develop a new VLSI design platform. The result was a new family of Xilinx FPGA's, called the Virtex II Pro (V-II Pro), built on Xilinx's Virtex family. Not only are these FPGA's faster and denser than previous programmable logic families from Xilinx or their competition, but they also integrate an IBM 405 PPC ASIC hardcore into the FPGA fabric. Along with the built in processor, Xilinx and IBM offered a set of soft cores to accompany the processor, which were optimized for FPGA implementation, as well as a new set of software tools to easily integrate them into FPGA designs [4-5].

The integration of an ASIC hardcore into the programmable fabric of an FPGA is not only a great milestone from a VLSI technology standpoint, but it also provides an essential bridge between the SOC market, traditionally only capable of being implemented in an ASIC, and re-programmable logic chips. With the availability of a central processor, FPGA optimized I/O cores (*ex.*, Ethernet), new software, and millions of FPGA gates allowing for user defined soft cores, Xilinx and IBM gave birth to a whole new market; the *re-programmable SOC*. Despite the advantages this offers to the general SOC market, there is one major drawback for FPGA implementations; the 405 PPC does not contain a floating-point unit (FPU). Floating-point operations are essential to FPGA algorithms. However, not all is lost. Since the user-defined SOC cores are programmed into the FPGA fabric, portions of a given FPGA algorithm can take advantage of vast hardware parallelism to implement floating-point MAC operations, while the rest of the algorithm is executed on the CPU. Future generations of the Xilinx re-programmable SOC will no doubt contain a FPU CPU, presenting the first fully integrated system to physically emulate (FPGA) algorithm tradeoffs and optimization. Therefore, the new reprogrammable SOC (*i.e.*, the Xilinx Virtex-II Pro) offers a unique and high tech platform to develop the ultimate FPGA engine [6-7].

In order to provide sufficient background information for the discussion of implementing control algorithms on Xilinx FPGA's, this part presents a thorough summary of the Xilinx architecture. It focuses primarily on the architectural information needed to understand multipliers and the implementation of FPGA algorithms.

The Xilinx Spartan FPGA architecture is basically the same as the Virtex II platform, with the exception that the V2P is shipped on 300 nm wafers with dies that are fabricated on 0.13um technology. These chips are SRAM based devices: that is, they do not retain their logical configuration once power is removed. Instead they contain an internal SRAM based configuration memory. Upon power up, application specific configuration data is loaded into the configuration memory, which is typically stored in an EEPROM (It can also be loaded via a PC through a JTAG boundary scan interface. The EEPROM communicates with the FPGA to facilitate a three phase loading sequence. First, the configuration memory is cleared, then the configuration data is loaded, finally followed by a start-up sequence that activates the logic (sequential release of clocks and control lines) [8]. Although FPGA stands for Field Programmable Gate Array, the Xilinx FPGA's at the highest level of abstraction are really more like a very dense array consisting of the 6 major building blocks shown in Figure 1. Configurable Logic Blocks (CLB.s), Block RAM's (BRAM), Multipliers, Digital Clock Managers (DCM's), and standard and high speed I/O (IOB.s), are all connected to each other through a fully buffered (SRAM controlled pass transistor) programmable Switching Matrix [8-10].

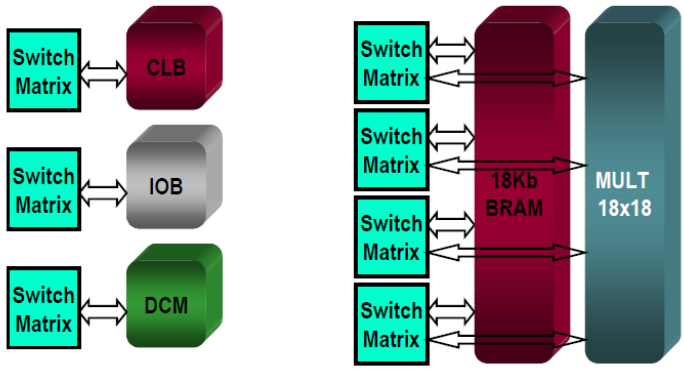


Figure 1. Building Blocks of Spartan Architecture

This switching matrix is programmed and controlled via the configuration data, known as a bit file, loaded into the configuration SRAM at power up. The CLB building blocks take up more than 75% of the area resources, and therefore each device within a the V2P family can be characterized by its CLB array size, with all of the other building blocks in relation to the CLB's, as depicted in Figure 2 [8]. The FPGA device that was supplied with the Xilinx development card that was utilized in the implementation portion of this research was the XC2VP7. This device has an array of 40x34, for a total of 1360 CLB's.

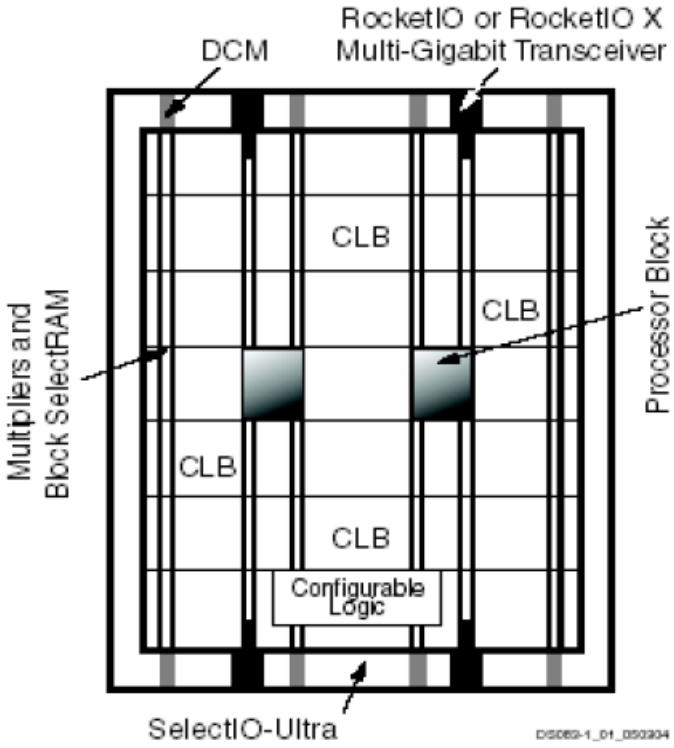


Figure 2. Array Architecture of Spartan FPGA

As can we seen from Figure 2, the Multipliers and Block RAM's are sandwiched in narrow columns between the CLB's. The Spartan 1600 E device has six such columns, containing a total of 44 multipliers and 44 Block RAM's. The maximum size V2P device goes up to 120x94 (11280) CLB's with 16 columns of 444 Block RAM's and 444 Multipliers [3-4].

Notice in Figure 2 that the Digital Clock Manager blocks are placed at the top and bottom of each Block RAM/ Multiplier column; thus there are a total of 12 DCM's in the XC2VP7 chip, with a maximum of 32 DCM's for the V2P family. The Rocket I/O Multi-Gigabit Transceivers are parallel to serial (and vice versa) embedded transceiver cores used for high-speed interfaces between multiple FPGA's, over a bus or back plane for example. Although these are very helpful to digital control designers who have need to parse a control algorithm across 2 chips and communicate quickly to keep processing real time data, neither the Rocket I/O nor the DCM's are the direct focus of discussion here, and thus the reader is referenced to [5] for further information on these topics. Although this is certainly a building block for the architecture, it really is an item that deserves separate attention, as it brings into the design a whole architecture of its own, with its own set of building blocks.

Clearly, the central building block of the V2P architecture is the CLB. Figure 3 illustrates the construction of a single CLB [6]. It consists of 4 slices, or sub-blocks, staggered into two columns, each with its own independent logic carry chain, as well a common shift chain connecting the staggered sets of slices. Each slice is connected to the programmable switch matrix such that each block may gain access to the IOB's, DCM's, BRAM's, Multipliers, and to other CLB's as Figure 1 illustrates. The fast connects allow for quick local feedback within the CLB [7].

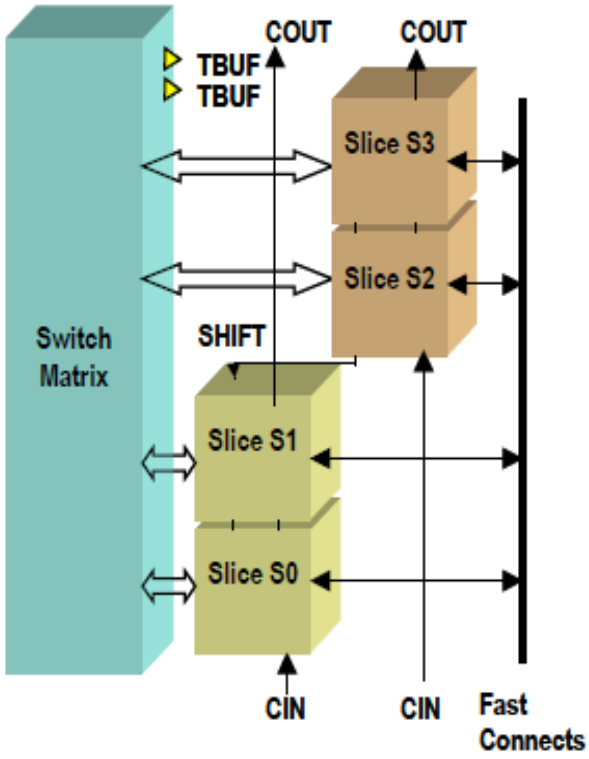


Figure 3. Spartan Configurable Logic Block (CLB)

This paper is organized as follows; Section 2, has served as an introduction to the system's dynamic data collection of robot manipulator. Part 3, introduces and describes the methodology based on FPGA. Section 4 presents the simulation results and discussion of this algorithm applied to a robot manipulator and the final section is describing the conclusion.

2. Theory

The equation of an *n-DOF* robot manipulator governed by the following equation:

$$\boldsymbol{M(q)\ddot{q} + N(q,\dot{q}) = \tau} \quad \textbf{(1)}$$

Where τ is actuation torque, M (q) is a symmetric and positive define inertia matrix, $N(q,\dot{q})$ is the vector of nonlinearity term. This robot manipulator dynamic equation can also be written in a following form:

$$\boldsymbol{\tau = M(q)\ddot{q} + B(q)[\dot{q}\,\dot{q}] + C(q)[\dot{q}]^2 + G(q)} \quad \textbf{(2)}$$

Where B(q) is the matrix of coriolios torques, C(q) is the matrix of centrifugal torques, and G(q) is the vector of gravity force. The dynamic terms in equation (2) are only manipulator position. This is a decoupled system with simple second order linear differential dynamics. In other words, the component $\ddot{q}$ influences, with a double integrator relationship, only the joint variableq_i, independently of the motion of the other joints. Therefore, the angular acceleration is found as to be:

$$\boldsymbol{\ddot{q} = M^{-1}(q).\{\tau - N(q,\dot{q})\}} \quad \textbf{(3)}$$

This technique is very attractive from a control point of view.

The data collection to implement the single joint robot arm is the relationship between the rate of torque and end-effector position. Regarding to this information Figure 4 shows the system dynamic behavior.

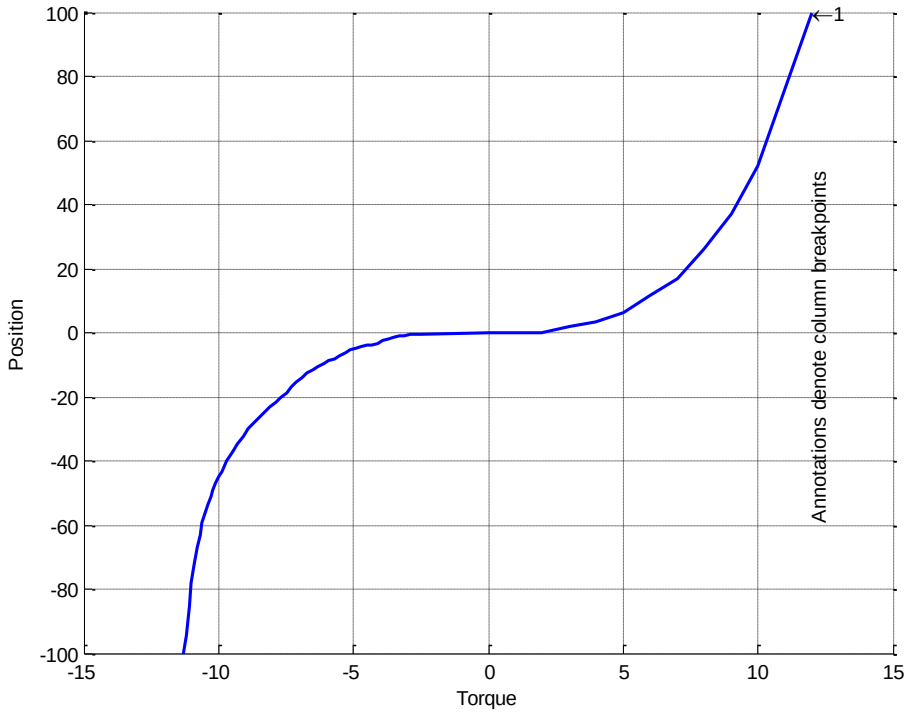


Figure 4. Single Joint Robot Arm System's Lookup Table

3. Methodology: Design FPGA-Based PD Controller

The current control algorithm to FPGA implementation flow must cross many boundaries, such as MATLAB to HDL, HDL to the FPGA, and FPGA back to MATLAB, while attempting to maintain the consistency of the original algorithm. What if design-flow, design time, and design resources for implementing control algorithms in hardware could be seriously improved? MATLAB is an interpretive language; it does not compile its code. Large nonlinear control algorithms can therefore take many minutes, even hours to execute. Those engineers who are strictly research oriented would most certainly be interested in a new MATLAB design flow that could speed up MATLAB and Simulink algorithms via an FPGA hardware accelerator. Those engineers who are interested in optimizing the entire nonlinear control algorithm to FPGA implementation flow would most certainly be interested in a new solution that packages the entire flow into one software tool, such as Simulink, which is part of MATLAB. Finally, for engineers who wish to explore re-programmable SOC's to develop more efficient nonlinear control engines, this design work would rely heavily on researching hardware and software tradeoffs between a PPC compiled algorithm and a soft-core algorithm accelerator. What if large control cores eventually targeted for an SOC embedded system could be tested at very high levels with assurance of timing and functional equivalency once they were implemented in the SOC as logic?

Xilinx and Math-works have the solution to these two questions. In addition to creating the concept of re-programmable SOC's, Xilinx has developed a new tool capable of a high level, single flow, implementation of control algorithms on FPGA.s. The tool, called System Generator, is integrated into Simulink, a tool within MATLAB used for modeling and simulating dynamic systems. Xilinx worked closely with Math-works to create an additional set of toolboxes for Simulink. The Xilinx FPGA tools include a bank of control and related logic cores, optimized for implementation on FPGA.s, which can be programmed to the users specifications into any HDL design. These cores are created with the Xilinx software tool called Core Generator. The new System Generator toolboxes extend the original capabilities of Simulink by integrating the cores from the Core Generator, allowing Simulink users to essentially implement logic designs at a high level. Interfacing these new hardware blocks with traditional Simulink blocks is possible, as well as interfacing to traditional MATLAB files. Once the design is complete and simulated from within MATLAB, the Xilinx specific blocks within the Simulink design can then be synthesized and placed and routed through the normal Xilinx FPGA flow. When this is complete, the Simulink algorithm can be simulated, with the Xilinx core functions literally being executed on an FPGA. This is accomplished through a JTAG cable connected to the JTAG interface on an FPGA development card and the parallel port on a PC. When the algorithm is executed, the tools automatically load the FPGA and run the simulation real-time on the hardware. The boundaries from MATLAB to HDL to FPGA and back to MATLAB are thus removed, thereby creating a high level, single flow methodology to implement a MATLAB control algorithm on an FPGA.

Since the emergence of the first models of robot manipulators, the control problem of such systems has drawn the attention of numerous researchers. The reasons behind this are the countless challenges imposed by these systems, for instance:

- The highly non-linear dynamics of the manipulator and the actuators, which include inertia, friction, backlash, and in some cases mechanical flexibility.
- Cross-coupling between neighboring inputs and outputs of the system makes the control problem more challenging for decoupling.
- Time-varying system dynamic parameters due to changes in specimen mass (payload), testing configuration, and speed of motion as well as component wear.

These techniques can be classified into model-based control (known as dynamic control strategies) and performance-based control (referred to as model-free or kinematic control strategies).

The model-free control strategy, also known as the kinematic control strategy, is based on the assumption that the joints of the manipulators are all independent and the system can be decoupled into a group of single-axis control systems. Therefore, the kinematic control method always results in a group of individual controllers, each for an active joint of the manipulator. Proportional-Derivative (PD) control, adaptive control, sliding mode control, fuzzy logic control, for tracking objectives as discussed in or for robust control using neural network control and hybrid neuro-fuzzy all belong to kinematic controllers. With the independent joint assumption, no a priori knowledge of parallel manipulator dynamics is needed in the kinematic controller design, so the complex computation of its dynamics can be avoided and the controller design can be greatly simplified. This is suitable for real-time control applications when powerful processors that can execute complex algorithms rapidly are not accessible. However, since joints coupling is neglected, control performance degrades as operating speed increases and a manipulator controlled in this way is only appropriate for relatively slow motion. The fast motion requirement results in even higher dynamic coupling between the various robot joints, which cannot be compensated for by a standard robot controller such as PD, and hence model-based control becomes the alternative.

Robot manipulators exhibit inherently non-linear coupled dynamics and model-based control techniques were proven to significantly improve their performance. For instance, based on approximated linear dynamic models, dynamic controllers were proposed for robot manipulators with improved tracking performance. Although this type of controller is easy to implement by adopting the approximated linear model of manipulators, the effect of the controller is limited to a small region of the configuration space for the inaccurate compensation of the non-linear dynamics. These works have placed solid foundation for further research on non-linear dynamic control strategies.Classical linear controllers (PD and PID) are still used today in commercial robots, because control actions are simplified by the linearization of the robot non-linear behavior. PD and PID controllers are more easily implemented, but their use for controlling coupled complex robotic manipulators is restricted to small and slow systems. It is shown in Figure 4 a PD joint controller, where q_d and q_r are the desired and real positions of the joint and $\dot{q_d}$ and $\dot{q_a}$ are the desired and real speeds. The parameters Kp and Kd are the PD controller parameters. As can be seen, the torque sign τdepends on the constants Kp and Kd. In Figure 5, a schematic view of a PD joint controller is illustrated. Again, the torque sign depends on the constant parameters Kp, and Kd.

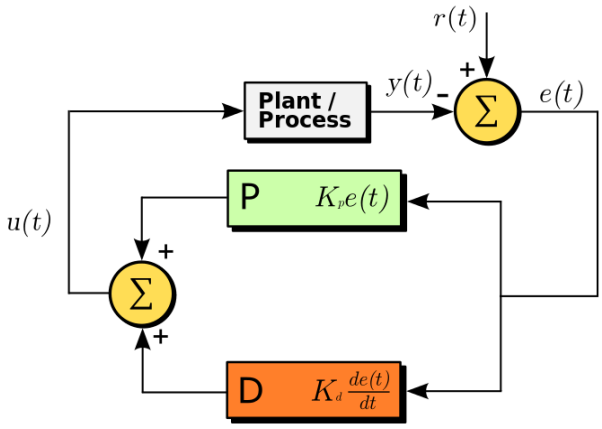


Figure 5. Schematic Diagram of PD Joint Controller

The dynamic formulation of PD controller is;

$$\mathbf{U = K_p e(t) + K_d \dot{e}} \tag{4}$$

$$\mathbf{U = K[e(t) + T_d \frac{d}{dt}]} \tag{5}$$

where the control variable u(t) is the controller output and K (the proportional gain), and Td (derivative time). Performing Laplace transform on (5), we get

$$\mathbf{G(s) = K[1 + S.}\boldsymbol{T_d}\mathbf{]} \tag{6}$$

We can easily convert the parameters from one form to another by noting that

$$\boldsymbol{K_p = K} \tag{7}$$
$$\boldsymbol{K_d = K.T_d}$$

For digital implementation, we are more interested in a Z-transform

$$\mathbf{U(z) = \boldsymbol{E(z)}[K_p + K_d(1 - Z^{-1})]} \tag{8}$$

Rearranging gives

$$\mathbf{U(z) = \boldsymbol{E(z)}[\frac{(K_p + K_d) + (-K_p - 2K_d)Z^{-1} + K_d Z^{-2}}{1 - Z^{-1}}]} \tag{9}$$

Before programming an FPGA, it is necessary to view and define a block's configurable parameters. These parameters include:
1. Arithmetic type − unsigned or twos complements
2. Latency − specifies the delay through the block
3. Overflow and quantization − the user can saturate or wrap overflow, and can select truncate quantization or rounding quantization.
4. Precision − full precision or the user can define the number of bits and where the binary point is for the block.
5. Sample period − an integer value
6. Equations − to simplify and speed up calculations
While most programs, MATLAB, Simulink, C++, and others use double-precision arithmetic, some FPGA manufacturers such as Xilinx use *n*-bit fixed point numbers specified as

$$Format = Sign\_Width\_Binary\ point\ from\ the\ LSB$$

Where

$$Sign = \begin{Bmatrix} Fix = Signed\ Value \\ UFix = Unsigned\ Value \end{Bmatrix}$$

A conversion is required whenever there is a communication between Simulink blocks and Xilinx blocks. Xilinx FPGA programming includes blocks referred to as *Gateway In and Out blocks*.

Their symbols are shown in Figure 6.

Figure 6. Symbols for Conversion from Double to N-bit Fixed Point and Vice Versa

A conversion example is shown in Figure7.

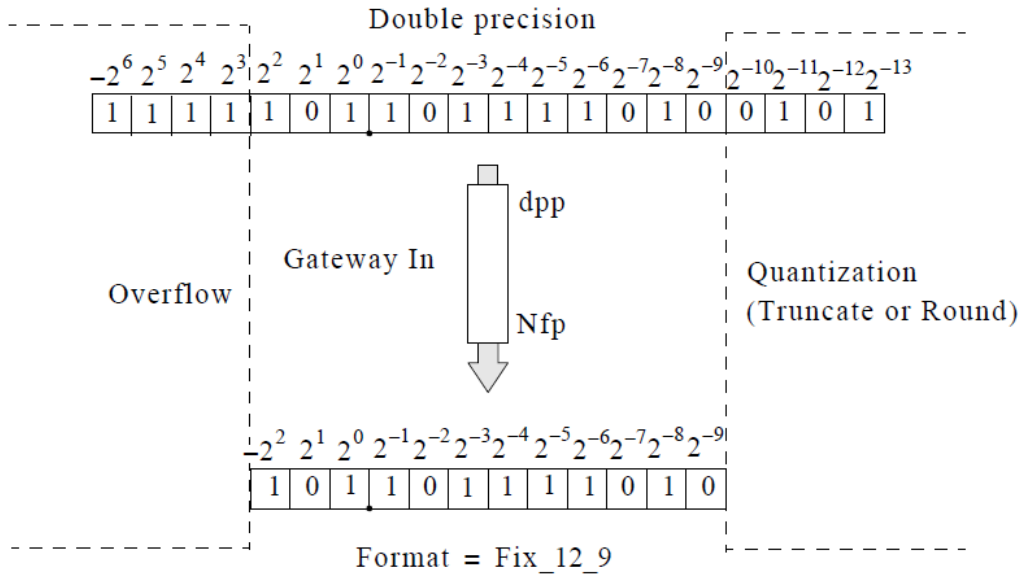


Figure 7. Conversion from Double Precision to N-bit Fixed Point Precision

In this project FPGA-based PD controller is design to control of single joint robot manipulator. This controller is implemented on XA Spartan-3E Automotive FPGA family. The Xilinx Automotive (XA) Spartan-3E family of FPGAs is specifically designed to meet the needs of high-volume, cost-sensitive automotive electronics applications. The five-member family offers densities ranging from 100,000 to 1.6 million system gates, as shown in Table 1.

Table 1. Summary of XA Spartan-3E FPGA Attributes

Device	System Gates	Equivalent Logic Cells	CLB Array (One CLB = Four Slices)				Distributed RAM bits(1)	Block RAM bits(1)	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs	Total Slices						
XA3S100E	100K	2,160	22	16	240	960	15K	72K	4	2	108	40
XA3S250E	250K	5,508	34	26	612	2,448	38K	216K	12	4	172	68
XA3S500E	500K	10,476	46	34	1,164	4,656	73K	360K	20	4	190	77
XA3S1200E	1200K	19,512	60	46	2,168	8,672	136K	504K	28	8	304	124
XA3S1600E	1600K	33,192	76	58	3,688	14,752	231K	648K	36	8	376	156

In this research the PD controller is implemented on SPARTAN-3E-XA3S1600E. Figure 8 shows the FPGA-based PD controller. Regarding to this Figure, this system has 5 inputs, namely; Actual inputs, desired inputs, CLK, reseat and sample CLK and has an output, namely; PD control.

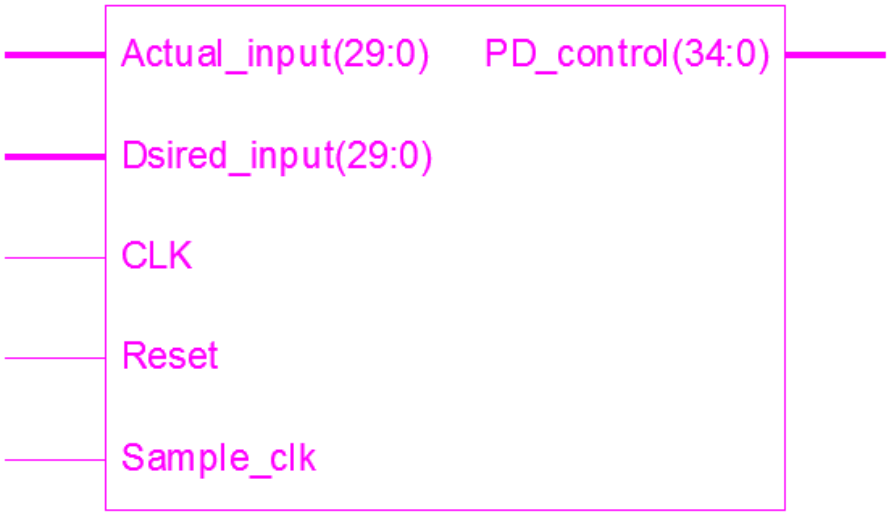


Figure 8. FPGA-based PD Control

This design has 2 steps:

1. Design P controller
2. Design D controller

Figure 9 shows the FPGA-based P controller.

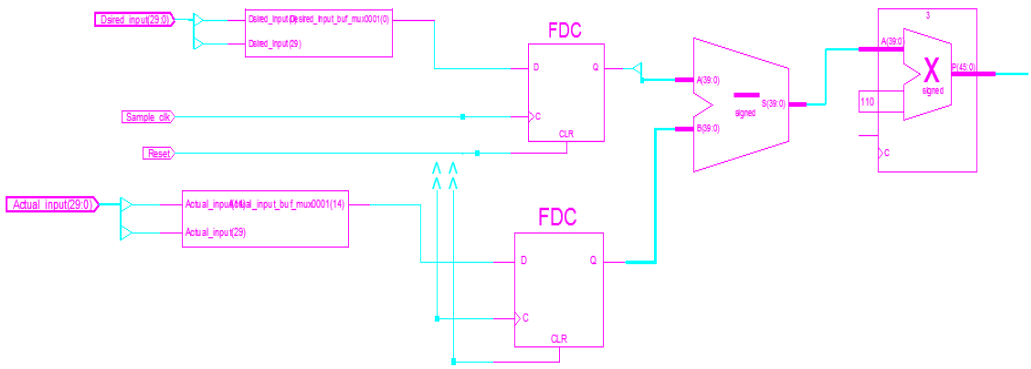


Figure 9. FPGA-based P Control

In derivative part, systems should derivative from error (the output of P controller). Figure 10 shows the derivative of error.

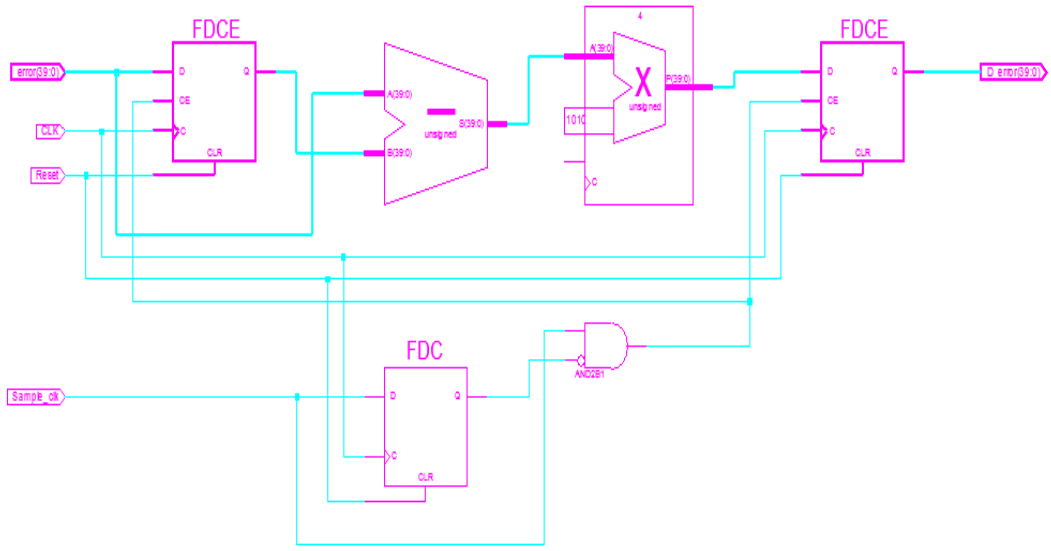


Figure 10. FPGA-based D Control

4. Result

Figure 11 shows the actual and desired input and also torque performance in transient state. Regarding to this Figure however actual and desired inputs equal to zero but torque performance has fluctuations in first 50 ns.

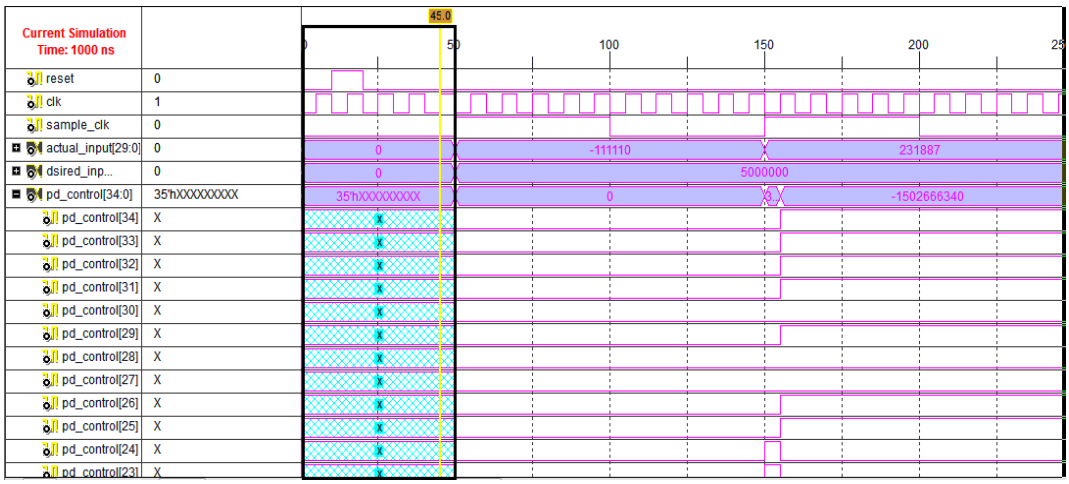


Figure 11. Transient State FPGA-Based Controller

Figure 12 indicates the actual and desired position, and also torque performance. In this state the desired position moved to 50 $degrees$ but in the next 100 ns the actual position moved to -11 $degrees$. Regarding to the following Figure it has about 61 degrees error.

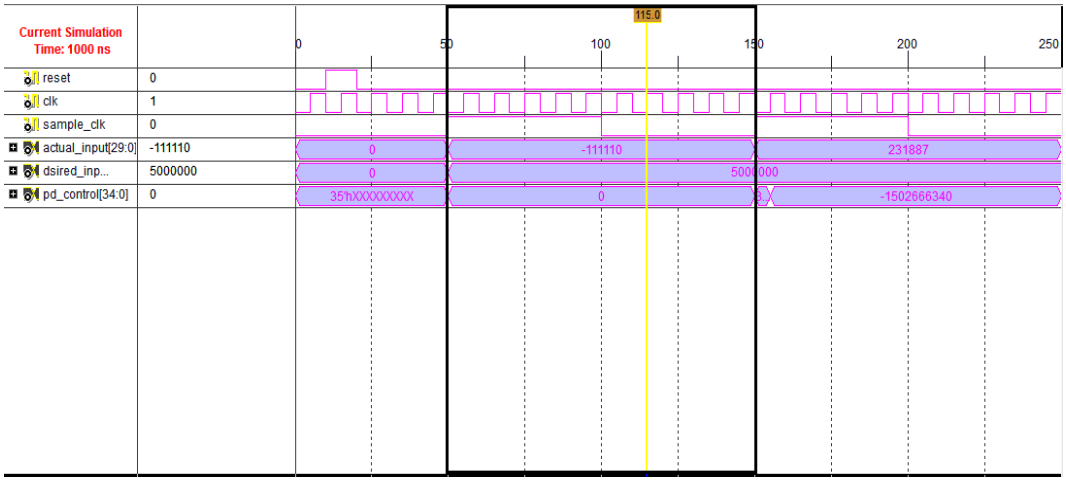


Figure 12. First 100 ns System Analysis

Regarding to Figure 12, the torque performance between 50 to 150 ns is equal to zero. In this time controller is inactive, this time is the controller's delay. The next $100\ ns\ (150-250\ ns)$ illustrate improvement the actual position from $-11\ degrees$ to $0.23\ degrees$. Regarding to the Figure 13 the error reduce from $61\ degrees$ to $49.77\ degrees$.

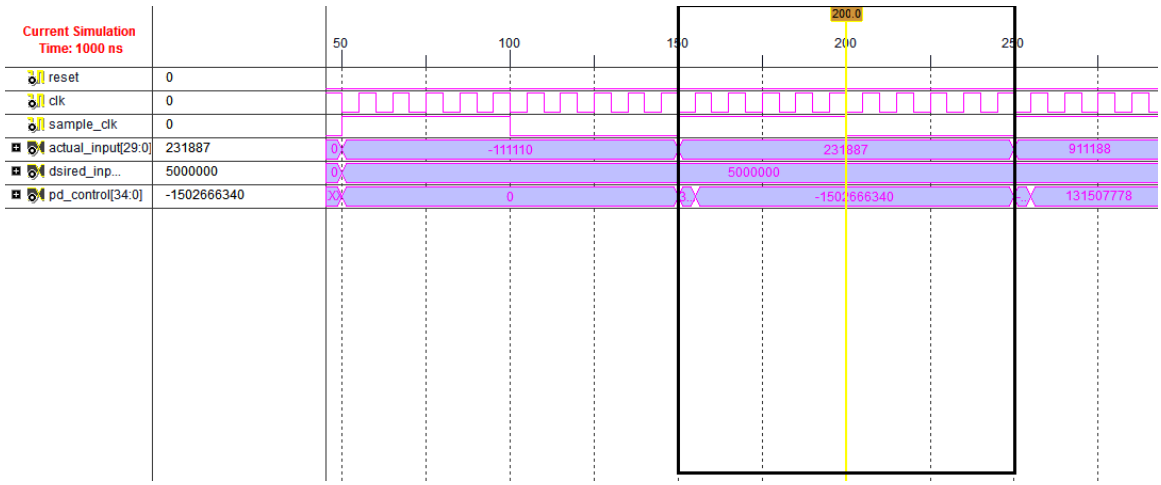


Figure 13. Second $100\ ns\ (150-250\ ns)$ System Analysis

The power of torque improve from zero (inactive) to $-1502.6\ \frac{N.M}{S}$. Figure 14 shows the final $100\ ns\ (850-950\ ns)$. Regarding to the following Figure the error reduce from $61\ degrees$ to $4.6\ degrees$ and the power of torque performance improve from zero (inactive) to $109.2\ \frac{N.M}{S}$.

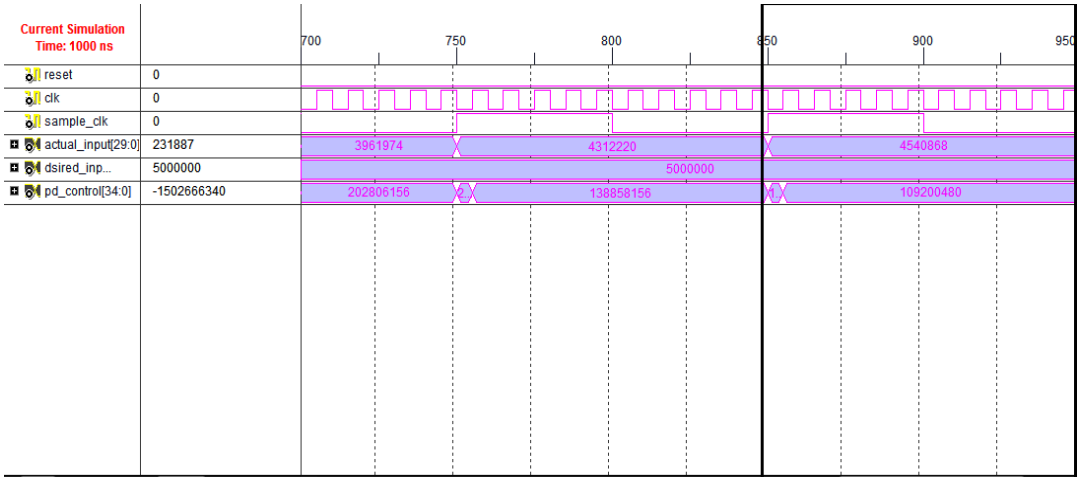


Figure 14. Final 100 ns (850 – 950 ns) System Analysis

5. Conclusion

From the design and simulation results of the proposed controller, it can be concluded that; higher execution speed versus small chip size is achieved by designing PD-FPGA based controller with simplified structure. This method improves the speed of system performance and reduces the delay of system's control. As a simulation result in XILINX, it is observed that; this controller is able to make as a fast response at 15.716 ns clock period with 63.7 MHZ of a maximum frequency and 4.362 ns for minimum input arrival time after clock. From investigation and synthesis summary, 19.727 ns for maximum input arrival time after clock with 50.69 MHZ frequencies, this design has 15.716 ns delays for each controller to 46 logic elements. Regarding to timing report 87.8% is logical delay and 12.2% is route delay. The offset before CLOCK is 1.946 ns for 1 logic gates. Therefore, the proposed controller will be able to control a wide range of the systems with high sampling rate.

Acknowledgement

The authors would like to thank the anonymous reviewers for their careful reading of this paper and for their helpful comments. This work was supported by the Iranian Institute of Advance Science and Technology Program of Iran under grant no. 2012-Persian Gulf-DIG.

Iranian center of Advance Science and Technology (IRAN SSP) is one of the independent research centers specializing in research and training across of Control and Automation, Electrical and Electronic Engineering, and Mechatronics & Robotics in Iran. At IRAN SSP research center, we are united and energized by one mission to discover and develop innovative engineering methodology that solve the most important challenges in field of advance science and technology. The IRAN SSP Center is instead to fill a long standing void in applied engineering by linking the training a development function one side and policy research on the other. This center divided into two main units:

- Education unit
- Research and Development unit

References

[1] J. L. Hennnesy and D. A. Pattterson, “Computer Architecture: A Quantitative Approach, 3rd ed.”, Morgan Kaufmann Publishers, **(2003)**.
[2] Xilinx Company, “XA Spartan 3E Field Programmable Gate Arrays”, Data sheet DS635, URL www.xilinx.com, **(2009)**.
[3] S. Lentijo, S. Pytel, A. Monti, J. Hudgins, E. Santi and G. Simin, "FPGA based sliding mode control for high frequency power converters", IEEE Conference on Power Electronics, **(2004)**, pp. 3588-3592.
[4] R. R. Ramos, D. Biel, E. Fossas and F. Guinjoan, "A fixed-frequency quasi-sliding control algorithm: application to power inverters design by means of FPGA implementation", IEEE Transactions on Power Electronics, vol. 18, no. 1, **(2003)**, pp. 344-355.
[5] F. J. Lin, D. H. Wang and P. K. Huang, "FPGA-based fuzzy sliding-mode control for a linear induction motor drive", IEEE Journal of Electrical Power Application, vol. 152, no. 5, **(2005)**, pp. 1137-1148.
[6] S. T. Karris, “Digital circuit analysis and design with Simulink modeling and introduction to CPLDs and FPGAs”, Orchard Pubns, **(2007)**.
[7] F. Piltan, A. Gavahian, N. Sulaiman, M.H. Marhaban and R. Ramli, “Novel Sliding Mode Controller for robot manipulator using FPGA”, Journal of Advanced Science & Engineering Research, vol. 1, no. 1, **(2011)**, pp. 1-22.
[8] F. Piltan, N. Sulaiman, M. H. Marhaban, A. Nowzary & M. Tohidian, “Design of FPGA-based Sliding Mode Controller for Robot Manipulator”, International Journal of Robotic and Automation, vol. 2, no. 3, **(2011)**, pp. 183-204.
[9] F. Piltan, N. Sulaiman, A. Jalali and K. Aslansefat, “Evolutionary Design of Mathematical tunable FPGA Based MIMO Fuzzy Estimator Sliding Mode Based Lyapunov Algorithm: Applied to Robot Manipulator”, International Journal of Robotics and Automation, vol. 2, no. 5, **(2011)**, pp. 317-343.
[10] F. Piltan, I. Nazari, S. Siamak and P. Ferdosali, “Methodology of FPGA-Based Mathematical error-Based Tuning Sliding Mode Controller”, International Journal of Control and Automation, vol. 5,no. 1, **(2012)**, pp. 89-118.

Authors

Farzin Piltan was born on 1975, Shiraz, Iran. In 2004 he is jointed Institute of Advance Science and Technology, Research and Development Center, IRAN SSP. Now he is a dean of Intelligent Control and Robotics Lab. He is led of team (47 researchers) to design and build of nonlinear control of industrial robot manipulator for experimental research and education and published about 54 Papers in this field since 2010 to 2012, team supervisor and leader (9 researchers) to design and implement intelligent tuning the rate of fuel ratio in internal combustion engine for experimental research and education and published about 17 Journal papers since 2011 to 2013, team leader and advisor (34 researchers) of filtering the hand tremors in flexible surgical robot for experimental research and education and published about 31 journal papers in this field since 2012 to date, led of team (21 researchers) to design high precision and fast dynamic controller for multi-degrees of freedom actuator for experimental research and education and published about 7 journal papers in this field since 2013 to date, led of team (22 researchers) to research of full digital control for nonlinear systems (e.g., Industrial Robot Manipulator, IC Engine, Continuum Robot, and Spherical Motor) for experimental research and education and published about 4 journal papers in this field since 2010 to date and finally led of team (more than 130 researchers) to implementation of Project Based-Learning project at IRAN SSP research center for experimental research and education, and published more than 110 journal papers since 2010 to date. In addition to 7 textbooks, Farzin

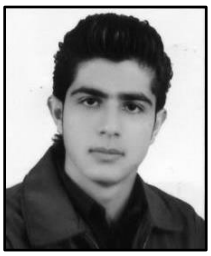
Piltan is the main author of more than 115 scientific papers in refereed journals. He is editorial review board member for 'international journal of control and automation (IJCA), Australia, ISSN: 2005-4297; 'International Journal of Intelligent System and Applications (IJISA)', Hong Kong, ISSN:2074-9058; 'IAES international journal of robotics and automation, Malaysia, ISSN:2089-4856; 'International Journal of Reconfigurable and Embedded Systems', Malaysia, ISSN:2089-4864. His current research interests are nonlinear control, artificial control system and applied to FPGA, robotics and artificial nonlinear control and IC engine modeling and control.

Maryam Rahmani is currently research assistant at Institute of Advance Science and Technology, Research and Development Center, IRAN SSP. She has been working at "Design High Precision and Fast Dynamic Controller for Multi-Degrees of Freedom Actuator for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "Pre-Master Student Researcher" of a research team composed of 21 researchers since Feb. 2013 to Feb. 2014. She has had the main roles in initiation and development of this project which has resulted thus far in three scientific publications up to now. She has been working at "Research of Full Digital Control for Nonlinear System for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "research assistant scholar" of a research team composed of 27 researchers since Feb. 2014 to date. Her current research interests are nonlinear control, artificial control system and design FPGA-based controller.

Meysam Esmaeili is currently research assistant at Institute of Advance Science and Technology, Research and Development Center, IRAN SSP. He has been working at "Design High Precision and Fast Dynamic Controller for Multi-Degrees of Freedom Actuator for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "Pre-PhD Student Researcher" of a research team composed of 21 researchers since Feb. 2013 to Feb. 2014. He has had the main roles in initiation and development of this project which has resulted thus far in three scientific publications up to now. He has been working at "Research of Full Digital Control for Nonlinear System for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "research assistant scholar" of a research team composed of 27 researchers since Feb. 2014 to date. His current research interests are nonlinear control, artificial control system and design FPGA-based controller.

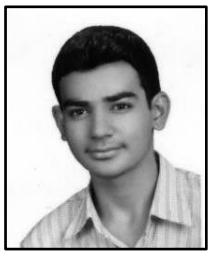
Mohammad Ali Tayebi is currently research assistant at Institute of Advance Science and Technology, Research and Development Center, IRAN SSP. He has been working at "Design High Precision and Fast Dynamic Controller for Multi-Degrees of Freedom Actuator for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "Pre-PhD Student Researcher"of a research team composed of 21 researchers since Feb. 2013 to Feb. 2014. He has had the main roles in initiation and development of this project which has resulted thus far in three scientific publications up to now. He has been working at "Research of Full Digital Control for Nonlinear System for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "research assistant scholar" of a research team composed of 27 researchers since Feb. 2014 to date. His current research interests are nonlinear control, artificial control system and design FPGA-based controller.

Mahsa Piltan is currently research assistant at Institute of Advance Science and Technology, Research and Development Center, IRAN SSP. She has been working at "DesignHigh Precision and Fast Dynamic Controller for Multi-Degrees of Freedom Actuator for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "Pre-PhD Student Researcher" of a research team composed of 21 researchers since Feb. 2013 to Feb. 2014. She has had the main roles in initiation and development of this project which has resulted thus far in three scientific publications up to now. She has been working at "Research of Full Digital Control for Nonlinear System for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "research assistant scholar" of a research team composed of 27 researchers since Feb. 2014 to date. Her current research interests are nonlinear control, artificial control system and design FPGA-based controller.

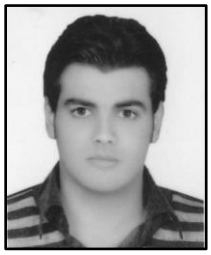


Hamid Cheraghi is currently research assistant at Institute of Advance Science and Technology, Research and Development Center, IRAN SSP. He has been working at "Design High Precision and Fast Dynamic Controller for Multi-Degrees of Freedom Actuator for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "Pre-Master Student Researcher" of a research team composed of 21 researchers since Feb. 2013 to Feb. 2014. He has had the main roles in initiation and development of this project which has resulted thus far in three scientific publications up to now. He has been working at "Research of Full Digital Control for Nonlinear System for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "research assistant scholar" of a research team composed of 27

researchers since Feb. 2014 to date. His current research interests are nonlinear control, artificial control system and design FPGA-based controller.



Mohammad Reza Rashidian is currently research assistant at Institute of Advance Science and Technology, Research and Development Center, IRAN SSP. He has been working at "Design and Build of Intelligent Controller for Industrial Robot Manipulator for Experimental Research and Education to Improve the Precision and Reduce the Position Error" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "Pre-Master Student Researcher" of a research team composed of 47 researchers since Aug. 2012 to Aug. 2013. He has had the main roles in initiation and development of this project which has resulted thus far in five scientific publications up to now. For the second project, he has been working at "Filtering the Hand Tremors in Flexible Surgical Robot for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "Pre-Master Student Researcher" of a research team composed of 44 researchers since Jun. 2013 to Aug. 2014. He has had the main roles in initiation and development of this project which has resulted thus far in four scientific publications up to now. Finally in 2014 he has been jointed to "Research of Full Digital Control for Nonlinear System for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "research assistant scholar" of a research team composed of 27 researchers. His current research interests are nonlinear control, artificial control system and design FPGA-based controller.



Arzhang Khajeh is currently research student at Institute of Advance Science and Technology, Research and Development Center, IRAN SSP. He has been working at "Filtering the Hand Tremors in Flexible Surgical Robot for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "Pre-Master Student Researcher" of a research team composed of 44 researchers since Aug. 2012 to Aug. 2013. He has had the main roles in initiation and development of this project which has resulted thus far in four scientific publications up to now. For the second project as "Pre-Master Student Researcher" in 2014 for three months he has been jointed to "Research of Full Digital Control for Nonlinear System for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) of a research team composed of 27 researchers. His current research interests are nonlinear control, artificial control system and design FPGA-based controller.

