

# CONTROL SYSTEM IMPLEMENTATION ON AN FPGA PLATFORM

Handan Gürsoy<sup>\*,\*\*</sup>, Mehmet Önder Efe<sup>\*</sup>

<sup>\*</sup> *Autonomous Systems Laboratory  
Department of Computer Engineering  
Hacettepe University, Beytepe, 06800, Ankara, Turkey,  
E-mails: {hgursoy,onderefe}@cs.hacettepe.edu.tr*

<sup>\*\*</sup> *Department of Computer Engineering, Iskenderun Technical University, Hatay, Turkey*

---

**Abstract:** This paper presents PID and SMC control schemes on a Field Programmable Gate Array device (FPGA) for robot manipulators. The implementation of the control systems on FPGA by using optimal hardware resources is one of the challenging research areas in control engineering. To accomplish this, MATLAB Xilinx System Generator toolbox plays an important role in control design on an FPGA device. In this paper, FPGA-based PD and SMC controllers are designed by using MATLAB Xilinx System Generator tool for the chosen robot system. The tracking performances of the presented control schemes, implemented in Matlab/Simulink and implemented on FPGA, are compared. Robustness and good trajectory performance of the system on FPGA are demonstrated.

© 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

**Keywords:** sliding mode controller, proportional-integral-derivative controller, Field Programmable Gate Array

---

## 1. INTRODUCTION

Control of robot manipulators has been extensively studied for many years (Young, 1978; Oded et al., 2004; Cong et al., 2009; Piltan et al., 2013). Robot manipulators constitute a prime application area of control systems due to the presence of nonlinearities and uncertainties in the governing dynamics. These aspects make the manipulator systems a good test bed to demonstrate the efficacy and performance of control algorithms.

PID (Proportional, Integral, Derivative) controller is a widely preferred control mechanism in many applications as it offers simplicity with good performance (Parra-Vega et al., 2003; Oded et al., 2004). Its operating principle can be explained by the use of measurable physics of the system. PID controller consists of three terms, which are proportional, integral and derivative terms. The coefficients of these terms are tuned to provide a robust performance. A considerable amount of literature has been reported on tuning methods (Ziegler et al., 1942,1943; Astrom et al., 1995; Papadopoulos, 2014; Leva, 2011). A well-known tuning scheme is Ziegler-Nichols' methods by J.G. Ziegler and N.B. Nichols proposed in 1942 and 1943. Another method is known as good gain method, which is an experimental method entailing simulation models as well. A number of researchers have reported how to tune the parameters in different situations, e.g. the cases elaborated in (Papadopoulos, 2014; Leva, 2011). However, the sole PID controller may not cope with the nonlinear characteristics or uncertain plants. Furthermore, several studies have revealed that PID controllers cannot provide the

desired performance for system models where there is very limited information about the plant.

Another robust control method is Sliding Mode Control (SMC) scheme. The idea of SMC emerged in 1950s, (Emelyanov, 1959), but the world became aware of the SMC with a paper in English by Utkin in 1970s. Since then, the approach has been attracting the interest of many researchers and engineers with a wide variety of application areas. A considerable amount of literature has been published on the SMC, (Slotine et al., 1991; Piltan et al., 2011a). These studies demonstrated that SMC is an efficient controller for nonlinear and uncertain systems. Other advantages of the SMC are that it overcomes the problem of large uncertainties and external disturbances. Despite its prominent features, SMC scheme is vulnerable to measurement noise. The main challenge faced by many researchers is the chattering phenomenon, which is caused by the measurement noise and the result of this is high frequency oscillations in the control signal. The need to the knowledge of uncertainty bounds is another critical issue to maintain robustness.

Chattering phenomenon can result in some harmful problems, such as erosion or fatigue of mechanical parts or drivers and the increase in electric power consumption. Numerous studies have attempted to reduce or eliminate the chattering (Bendaas et al., 2013; Roopaei et al., 2012). Two standard methods are emphasized so far, namely, boundary layer method and estimated uncertainties method. Recently, fuzzy logic and neural network methods are used with SMC scheme to obtain a chattering free response. It should be emphasized that the schemes removing the chattering and softening the

control signal inevitably give some concessions from the robustness of the control system.

Field Programmable Gate Arrays (FPGAs) have been attracting a growing increasing interest in the recent years. This device is particularly useful in applications of the realization of digital control systems. FPGA based controllers provide a number of advantages in real time applications, (Hwu et al., 2005; Sulaiman et al., 2009; Piltan et al, 2011b). The most important advantage of the use of FPGA is high speed and low power consumption. FPGA based controller has a short execution time due to its parallel and hardware based architecture. In addition to these advantages, this device has a small size and low cost compared to alternative digital platforms, (Roopaei et al., 2012). Aside from the speed features, the algorithms are coded on a reconfigurable platform that can be revised or refined very easily.

In (Lv et al., 2015), control of DC/DC Buck Converters is presented using a FPGA system. The conventional PID and modified adaptive PID control algorithms are chosen as the control methods. In (Cho et al., 2009), the design and implementation of a multiple-axis motion control chip is presented on FPGA and this chip is used in the control system of a robot manipulator. Erkal et al (2015) have developed a co-processor for six-legged robot. The co-processor contains a FPGA and a DSP block. While generating the positions of the robot by DSP block, FPGA, which reads the positions, generates the necessary joint angles. In (Piltan et al., 2011c), FPGA-based SMC is proposed for PUMA-560 robot manipulator and demonstrates a successful set of feedback control results.

This paper is organized as follows. In section II, dynamic formulation of the robot manipulator is presented. Section III briefly describes the studied control schemes, namely, PID and SMC controllers. In Section IV, FPGA-based control systems are presented. In the next section, the simulation results are demonstrated with a comparison of controllers realized in Matlab environment and those realized in FPGA platform. The concluding remarks of the paper are given in Section VI.

## 2. DYNAMIC FORMULATION OF ROBOT MANIPULATOR

The dynamic model of a robot manipulator, without the effect of friction, is given as

$$\tau = M(q)\ddot{q} + B(q)[\dot{q}\dot{q}] + C(q)[\dot{q}]^2 + G(q) \quad (1)$$

where  $\tau \in \mathbb{R}^n$  is vector of the applied joint torques;  $M(q) \in \mathbb{R}^{n \times n}$  is the symmetric and positive definite inertia matrix;  $B(q) \in \mathbb{R}^{n \times n}$  is the Coriolis forces;  $C(q) \in \mathbb{R}^{n \times n}$  is the Centrifugal forces;  $G(q) \in \mathbb{R}^{n \times 1}$  is the gravity force;  $q \in \mathbb{R}^n$  is vectors of the joint positions,  $\dot{q} \in \mathbb{R}^n$  is vectors of the joint velocities,  $\ddot{q} \in \mathbb{R}^n$  is vectors of the joint accelerations.

As can be seen from (1), the robot manipulator is a second-order differential model. According to (1), the angular acceleration can be written as in (2).

$$\ddot{q} = M^{-1}(q)\{\tau - [B(q)[\dot{q}\dot{q}] + C(q)[\dot{q}]^2 + G(q)]\} \quad (2)$$

In this paper, we are focused on cylindrical robot manipulator, which has three degrees of freedom. The first joint is a revolute joint and this is the base link of the manipulator. The new two joints are prismatic as shown in Fig. 1. The dynamic model of cylindrical robot manipulator illustrated in Fig. 1, can be described as follows. Let's rewrite (1) as below.

$$M(q) \begin{bmatrix} \ddot{q}_1 \\ \ddot{d}_2 \\ \ddot{d}_3 \end{bmatrix} + B(q) \begin{bmatrix} \dot{q}_1 & \dot{d}_2 \\ \dot{q}_1 & \dot{d}_3 \\ \dot{d}_2 & \dot{d}_3 \end{bmatrix} + C(q) \begin{bmatrix} \dot{q}_1^2 \\ \dot{d}_2^2 \\ \dot{d}_3^2 \end{bmatrix} + G(q) = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad (3)$$

Using the Lagrange approach for modeling, we can obtain the following analytical details embodying the dynamic model of the robot.

$$\begin{aligned} \tau_1 = & \frac{1}{4}l_1^2 + I_{zz1} + I_{zz2} + I_{yy3} \\ & + m_2 \left( l_1^2 + \frac{1}{4}l_2^2 - l_2 d_2 + d_2^2 \right) \dot{q}_1 \\ & - [l_1 m_2] \dot{d}_2 + m_2 (-l_2 + 2d_2) \dot{q}_1 \dot{d}_3 \end{aligned} \quad (4)$$

$$\tau_2 = [m_2 + m_3] \ddot{d}_2 + g(m_2 + m_3) \quad (5)$$

$$\tau_3 = -[m_2 l_1] \dot{q}_1 + m_2 \ddot{d}_3 + m_2 \left( \frac{1}{2}l_2 - d_3 \right) \dot{q}_1^2 \quad (6)$$

Using (2), it is possible to write three differential equations to obtain a simulation model. The variables of (3) are defined below.

$$M(q) = \begin{bmatrix} c & 0 & -m_2 l_1 \\ 0 & m_2 + m_3 & 0 \\ -m_2 l_1 & 0 & m_3 \end{bmatrix} \quad (7)$$

$$\begin{aligned} c = & \frac{1}{4}l_1^2 + I_{zz1} + I_{zz2} + I_{yy3} \\ & + m_2 \left( l_1^2 + \frac{1}{4}l_2^2 - l_2 d_2 + d_2^2 \right) \end{aligned} \quad (8)$$

$$B + C(q, \dot{q}) = \begin{bmatrix} m_2 (-l_2 + 2d_2) \dot{q}_1 \dot{d}_2 \\ 0 \\ m_2 \left( \frac{1}{2}l_2 - d_3 \right) \dot{q}_1^2 \end{bmatrix} \quad (9)$$

$$G(q) = \begin{bmatrix} 0 \\ g(m_2 + m_3) \\ 0 \end{bmatrix} \quad (10)$$

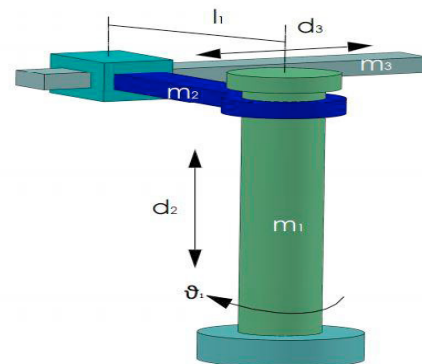


Fig. 1. Cylindrical robot manipulator

The dynamic model of the robot contains three differential equations, which are coupled and nonlinear. This fact is the driving force for considering it on a FPGA based closed loop control study. In the next section, we elaborate the control schemes.

### 3. CONTROL SCHEMES

#### 3.1 PID Control

A classical PID controller contains three terms, which are proportional, integral and derivative parts. Defining  $e(t)$  as the difference between the reference signal and the system output and  $u(t)$  as the output of the controller, time domain representation of the PID controller can be given as

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (11)$$

where  $K_P$ ,  $K_I$ ,  $K_D$  are proportional, integral and derivative coefficients, respectively. Transfer function of a PID controller can now be written as

$$G(s) = \frac{U(s)}{E(s)} = K_P + K_I \frac{1}{s} + K_D s \quad (12)$$

where  $U(s)$  and  $E(s)$  stand for the Laplace transform of the controller output and the error signal, respectively.

The proportional term directly affects the response of the system, but containing solely the proportional term is not enough to force the error signal toward zero. When a large proportional gain is used, undesired oscillations in the system response arise. The integral term, which is based on the sum of the past errors, is added to the control signal to compensate the possible steady state errors. The integral action slows down the system response yet improves the steady state performance. The last term is the derivative action, which improves the transient response significantly. The derivative action differentiates the error signal and any high frequency component in the error is differentiated, scaled and added to the control signal.

Depending on the needs of the control system, one might choose to implement a two term variant of a PID controller, e.g. PI or PD controllers. When the controller structure is chosen, the next step in the design is to set the gains that scale the error, integral of error and derivative of error. In the literature, a number of standard methods are available for PID tuning. Most widely-known one is the Ziegler-Nichols scheme. The designer then refines the values prescribed by Ziegler-Nichols approach manually. The reason for this is the fact that Ziegler-Nichols table is an average value for the performed measurements. Today's design software like Matlab provides very powerful graphical tools to obtain an acceptable response.

#### 3.2 Sliding Mode Control

SMC is a very popular nonlinear control law, utilized successfully in many control problems. The scheme is a discontinuous one with two different decisions taking place along a decision hyperplane. The response of a dynamical system under SMC displays two fundamentally different

behaviors. These are the reaching phase and the sliding phase. The error vector and its derivative span the phase space and the motion in this particular space is first guided toward a predefined subspace, called the switching subspace (or switching manifold). The rest of the SMC algorithm ensures that the motion after hitting the switching subspace takes place in that locus. The dynamical implication of this is the fact that error vector is forced toward the origin of the phase space and reference tracking with zero error is observed. In order to understand the SMC scheme, define  $S(x, t)$  as the switching variable, which is given as in (13).

$$S(x, t) = \left( \frac{d}{dt} + \lambda \right) \tilde{x} = (\dot{x} - \dot{x}_d) + \lambda(x - x_d) \quad (13)$$

where  $\lambda$  is a positive constant determining the slope of the switching line. In above,  $\tilde{x} = x - x_d$  is tracking error and  $x_d$  is the desired (reference) trajectory. The aim is to keep  $S(x, t)$  near to zero and keep it stable point in the method. The time derivative of  $S$  can be calculated

$$\dot{S} = (\ddot{x} - \ddot{x}_d) + \lambda(\dot{x} - \dot{x}_d) \quad (14)$$

The candidate Lyapunov function is chosen as below.

$$V = \frac{1}{2} S^T S \quad (15)$$

The function given above is positive definite ( $V > 0$ ), but derivative of this function is negative definite ( $\dot{V} < 0$ ). As the derivative of  $V$  is set to

$$\frac{dV(S)}{dt} = -S^T K \text{sgn}(S) \quad (16)$$

The function of  $\text{sgn}(S)$  represented as:

$$\text{sgn}(S) = \begin{cases} 1 & S > 0 \\ 0 & S = 0 \\ -1 & S < 0 \end{cases} \quad (17)$$

where  $K$  is a positive definite and diagonal matrix. According to Lyapunov stability criterion, the origin is a stable equilibrium point. Forcing (16) in the closed loop yields the control law given as

$$U = U_{eq} + U_{switch} \quad (18)$$

where  $U_{eq}$  is the equivalent control term given as below.

$$U_{eq} = [M^{-1}(B + C + G) - \dot{S}]M \quad (19)$$

The corrective control signal, which contains the switching part of the control signal, denoted by  $U_{switch}$  is given by

$$U_{switch} = K \text{sgn}(S) \quad (20)$$

As seen from the above term, the control signal depends on the sign of a variable that is very close to zero. This particularly makes the controller sensitive to noise available in the measurements. More explicitly, tiny noise components may change the sign of  $U_{switch}$  very fast and this introduces high frequency fluctuations into the control signal. The appearance of this in the phase space is the emergence of the phenomenon called *chattering*. For this problem, smoothing

the sign function around zero is accepted as a remedy. Among several alternatives, we adopt the expression in (20)

$$U_{switch} = K \frac{S}{\epsilon + |S|} \quad (21)$$

The price paid for this modification is the introduction of a boundary layer around the switching subspace. The robustness is very slightly degraded yet the control signal with such a modification becomes very smooth.

### 3. FPGA IMPLEMENTATION OF THE CONTROL SCHEMES

Development of FPGA based controller is carried out in two steps. Firstly, the proposed control systems are redesigned by using MATLAB Xilinx System Generator toolbox. This requires the conversion of all component models to models of digital hardware. The controller and the robot dynamics are realized separately and the generated code is executed on the FPGA board. Pipelining and parallel processing property of FPGA platform provides designing fast controllers and the study provides a list of resource usage for each test.

The proposed control systems in this paper are synthesized for implementation on XC7A100T FPGA from Xilinx Artix-7 family. The functional module for the synthesis of this system is designed using VHDL language, which is one of most commonly used hardware description language. The synthesis results are demonstrated in terms of FPGA resource consumption in Table 1 and Table 2. Resource consumption of the PD controller is listed in Table 1. These results are obtained a maximum clock frequency of 12 MHz. In Table 2 is demonstrated consumption of the SMC controller. The controller is synthesized at the same clock frequency.

**Table 1. Synthesis results and device utilization (for PD controller)**

	Used	Available	Utilization
FF	6829	126800	5,39%
LUT	40081	63400	63,22%
Memory LUT	5	19000	0,03%
I/O	16	210	7,62%
BRAM	9	135	6,67%
DSP48	198	240	82,50%
BUFG	1	32	3,12%

**Table 2. Synthesis results and device utilization (for SMC controller)**

	Used	Available	Utilization
FF	10846	126800	8,55%
LUT	53819	63400	84,89%
Memory LUT	113	19000	0,59%
I/O	16	210	7,62%
BRAM	10	135	7,41%
DSP48	225	240	93,75%
BUFG	1	32	3,12%

## 4. SIMULATIONS AND RESULTS

PD and SMC approaches are chosen to implement the closed loop feedback control mechanism. MATLAB/Simulink and Xilinx System Generator are used for modeling and simulation of the entire system. The proposed methodology is applied to the robot manipulator shown in Fig. 1. The values of the robot manipulator parameters are given in Table 3 and the gravitational acceleration ( $g$ ) is taken as  $9.81 \text{ m/s}^2$ . In the sequel, the results obtained in the Matlab/Simulink environment are shown and the whole system is synthesized and downloaded on the FPGA board and the necessary rearrangements are done. The loop is simulated once again on the FPGA board and the results are shown next.

**Table 3. Dynamics Parameters of the Robot Manipulator**

Parameter Name	Value
Link 1 mass (m1)	0.5 [kg]
Link 2 mass (m2)	0.5 [kg]
Link 3 mass (m3)	0.5 [kg]
Link 1 radius (r)	2 [m]
Link 1 length (l1)	0.5 [m]
Link 2 length (l2)	0.5 [m]
Link 3 length (l3)	0.5 [m]

### 4.1 Simulation in the MATLAB/Simulink Environment

In this part, we demonstrate the result obtained in the Matlab/Simulink environment. In Fig. 2, PD control results for robot manipulator are demonstrated. The PD parameters set  $k_p=256$ ,  $k_d=16$ . These parameters are determined by using a trial and error method.

In Fig. 3 illustrates SMC results. The best performing coefficient set contains  $\lambda=3$  and  $K=5$ . Besides, the state values are corrupted with a zero mean noise sequence that has a variance equal to  $10^{-7}$ . As mentioned in the previous section, chattering phenomenon is an undesirable result. Reducing or eliminating the chattering is critical and in this paper, an approximated sign function is used to reduce this phenomenon and  $\epsilon=0.25$  is selected in (20).

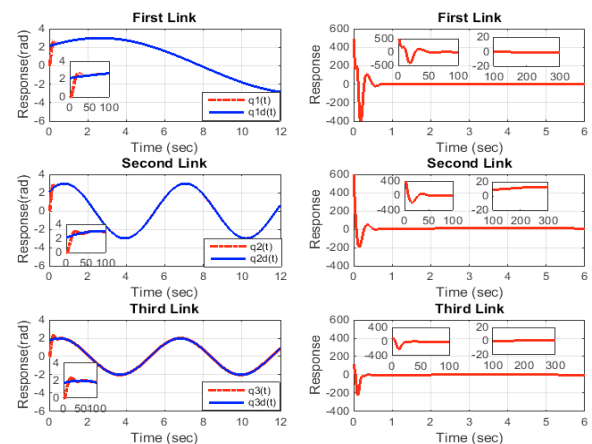


Fig. 2. MATLAB-based PD for first, second and third link trajectory without any disturbance and control signals.



The phase space behaviors have been shown in Fig. 4. This can be interpreted as the errors converge to the origin. Thus, the results indicate that the presented control scheme fulfills the proposed aim of SMC, that is, to compel the system states to the sliding manifold.

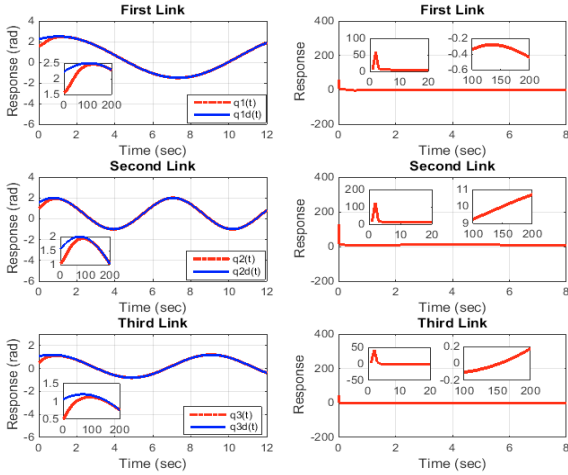


Fig. 3. MATLAB-based SMC for first, second and third link trajectory with noise and control signals.

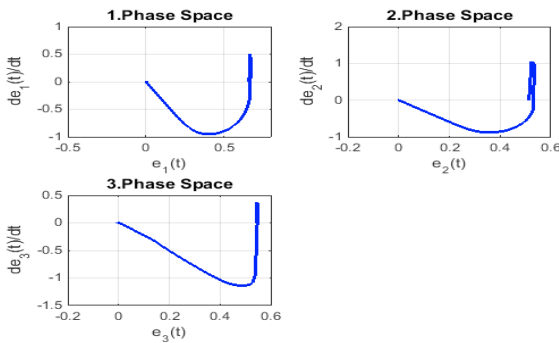


Fig. 4. Behavior in the phase space for the first, second and the third link.

#### 4.2 Implementation of the Control Schemes on the FPGA Platform

In this part, the results obtained on the FPGA platform are shown. Fig. 5 illustrates the results for FPGA-based PD control outputs. As seen from the figure, trajectory tracking performance of the FPGA based implementation, whose numerical representation precision range is poorer than a personal computer's, is the same with the control implemented in Matlab/Simulink environment.

In Fig. 6, results of the FPGA-based SMC controller are shown. The positional variables very quickly reach the reference signal as prescribed by the design and the error vector stays around the origin. The trajectory tracking performance is observed to be very good and the phase space behaviors are depicted in Fig. 7, where it is seen that a reaching mode response is observed in the initial transient and the errors converge the origin of the phase space by following the line defined by  $-\lambda$ .

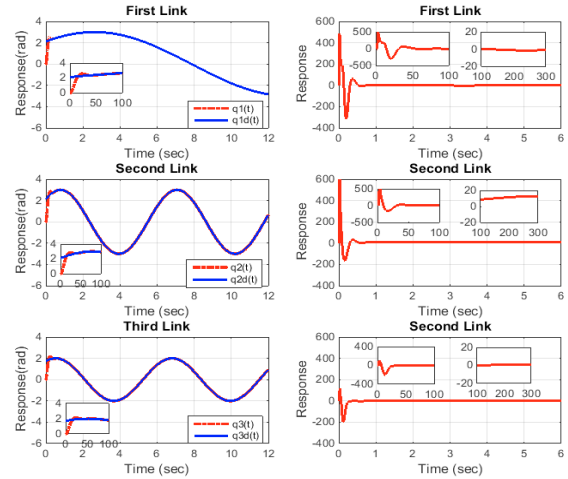


Fig. 5. Left: PD control results for the first, second and the third link. Right: Produced control signals.

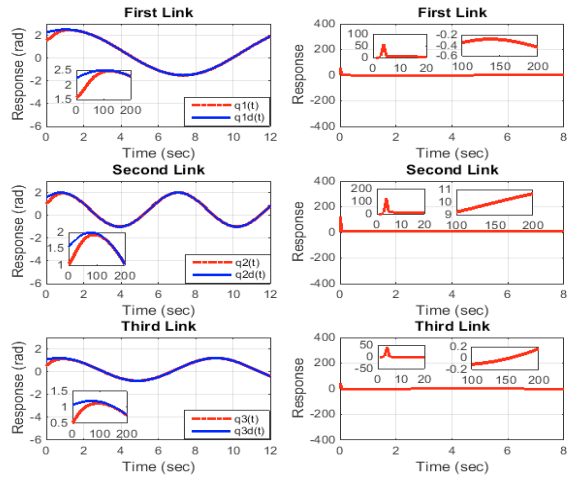


Fig.6. FPGA-based SMC for first, second and third link trajectory with noise and control signals

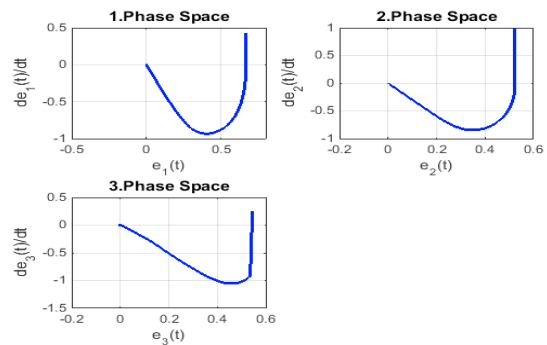


Fig.7. FPGA-based the behavior in the phase space for first, second and third link

## 6. CONCLUSIONS

This paper considers the application of two frequently studied control schemes in Matlab/Simulink environment and on an FPGA platform. The Matlab/Simulink based part is a

simulation study that is considered as the reference design and it is shown that a very similar performance can be obtained on an FPGA platform. A three degrees-of-freedom robot manipulator is chosen as the test bed and PD control scheme and a sliding mode control method are studied. The difficulty in implementing this approach was to convert all analytical details embodying the feedback control mechanism to a hardware description language model. Such models operate at certain a numerical precision, which is a limitation when good performance and precision in trajectory tracking are sought. The results have shown that FPGA based cloning of a control mechanism can produce as very good results as the Matlab/Simulink based counterpart and this makes FPGA platforms a good alternative for real time control system applications.

Future work of us aims to implement this result on a mobile manipulator with visual guidance functions governed on a FPGA platform.

#### ACKNOWLEDGMENTS

The authors gratefully acknowledge the TÜBİTAK-ARRS Project, Reference No 114E954.

#### REFERENCES

- Astrom, K.J., and Hagglund, T., (1995) "PID control-Theory, Design and Tuning." *Instrument Society of America, Research Triangle Park, NC*.
- Bendaas, I.I, and Farid N., (2013) "A new method to minimize the chattering phenomenon in sliding mode control based on intelligent control for induction motor drives." *Serbian Journal of Electrical Engineering* 10.2: 231-246.
- Cho, J.U., Quy N.L., and Jae W. J., (2009) "An FPGA-based multiple-axis motion control chip." *Industrial Electronics, IEEE Transactions on* 56.3: 856-870.
- Cong, S., and Liang, Y., (2009) "PID-like neural network nonlinear adaptive control for uncertain multivariable motion control systems." *Industrial Electronics, IEEE Transactions on* 56.10: 3872-3879.
- Emelyanov, S.V., (1959) "Control of first order delay systems by means of an astatic controller and nonlinear corrections," *Autom. Remote Control*, vol. 8, pp. 983–991.
- Erkol, H.O., and Demirel, H. (2015) "An FPGA Based Robot Controller Design and Implementation." *International Conference Intelligent Technologies in Logistics and Mechatronics System (ITELMS'2015)*.
- Hwu, K.I., and Tau, Y.T., (2005) "A Forward Converter Having an FPGA-based PID Controller with Parameters On-line Tuned." *Power Electronics and Drives Systems, 2005. PEDS 2005. International Conference on*. Vol. 2. IEEE.
- Leva, A., and Martina M., (2011) "A systematic way to extend ideal PID tuning rules to the real structure." *Journal of Process Control* 21.1: 130-136.
- Lv, L., Chang, C., Zhou, Z., and Yuan, Y. (2015) "An FPGA-Based Modified Adaptive PID Controller for DC/DC Buck Converters." *Journal of Power Electronics*, 15(2), 346-355.
- Oded, Y. and Nagurka. M., (2004) "Design of PID controllers satisfying gain margin and sensitivity constraints on a set of plants." *Automatica* 40.1: 111-116.
- Papadopoulos K., (2014) "PID Controller Tuning Using the Magnitude Optimum Criterion", *Springer*.
- Parra-Vega, V., Arimoto, S., Liu, Y. H., Hirzinger, G., and Akella, P., (2003) "Dynamic sliding PID control for tracking of robot manipulators: theory and experiments." *Robotics and Automation, IEEE Transactions on* 19.6: 967-976.
- Piltan F., Sulaiman N., Gavahian A., Soltani S. and Roosta S., (2011a) "Design mathematical tunable gain PID-like sliding mode fuzzy controller with minimum rule base", *International Journal of Robotic and Automation* 2 (3): 146-156.
- Piltan, F., Sulaiman, N., Marhaban, M. H., Nowzary, A., & Tohidian, M., (2011b) "Design of FPGA based sliding mode controller for robot manipulator." *International Journal of Robotic and Automation* 2.3: 183-204.
- Piltan, F., Sulaiman, N., Marhaban, M. H., Nowzary, A., and Tohidian, M., (2011c) "Design of FPGA based sliding mode controller for robot manipulator." *International Journal of Robotic and Automation* 2.3: 183-204.
- Piltan F., Nabaee A., Ebrahimi M.M. and Bazregar M., (2013) "Design robust fuzzy sliding mode control technique for robot manipulator systems with modeling uncertainties", *International Journal of Information Technology and Computer Science (IJITCS)* 5(8): 123-135.
- Roopaei M., Jahromi M. and Zolghadri (2012) "Chattering-free fuzzy sliding mode control in MIMO uncertain systems", *Nonlinear Analysis-Theory Methods & Applications* 71(10): 4430-4437.
- Slotine, J.J.E., and Weiping L., (1991) "Applied nonlinear control." Vol. 199. No. 1. *Englewood Cliffs, NJ: Prentice-Hall*.
- Sulaiman, N., Obaid, Z.A., Marhaban, M.H., and Hamidon, M. N., (2009) "Design and implementation of FPGA-based systems-a review." *Australian Journal of Basic and Applied Sciences* 3.4: 3575-3596.
- Utkin, V., (1977) "Variable structure systems with sliding modes," *IEEE Trans. Autom. Control*, vol. AC-22, no. 2, pp. 212–222.
- Young, K.K.D., (1978) "Controller Design for a Manipulator Using Theory of Variable Structure Systems", *IEEE Transactions On Systems, Man, And Cybernetics*, vol. 8(2).
- Ziegler, J.G. and Nichols, N.B., (1942) "Optimum settings for automatic controllers," *Trans. ASME*, vol. 64, pp. 759-768.
- Ziegler, J.G. and Nichols, N.B., (1943) "Process lags in automatic control circuits," *Trans. ASME*, vol. 65, pp. 433-444.