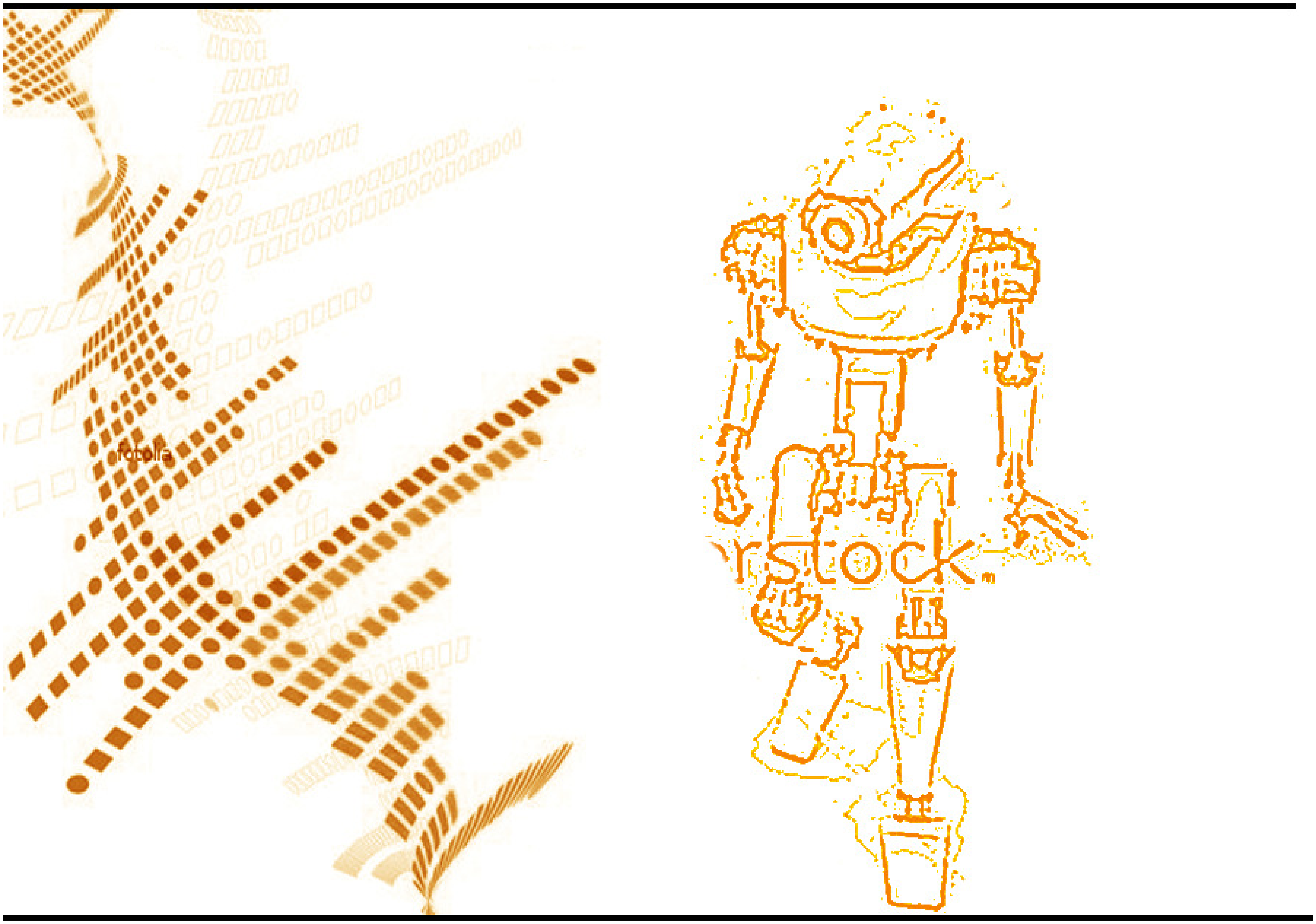
Volume 2 ▪ Issue 3 ▪ August 2011

INTERNATIONAL JOURNAL OF ROBOTICS AND AUTOMATION (IJRA)

ISSN : 2180-1312
Publication Frequency: 6 Issues / Year

CSC PUBLISHERS
http://www.cscjournals.org



INTERNATIONAL JOURNAL OF ROBOTICS AND AUTOMATION (IJRA)

VOLUME 2, ISSUE 3, 2011

EDITED BY
DR. NABEEL TAHIR

ISSN (Online): 2180-1312

I International Journal of Robotics and Automation (IJRA) is published both in traditional paper form and in Internet. This journal is published at the website http://www.cscjournals.org, maintained by Computer Science Journals (CSC Journals), Malaysia.

IJRA Journal is a part of CSC Publishers
Computer Science Journals
http://www.cscjournals.org

INTERNATIONAL JOURNAL OF ROBOTICS AND AUTOMATION (IJRA)

Book: Volume 2, Issue 3, August 2011
Publishing Date: July / August 2011
ISSN (Online): 2180-1312

IJRA Journal is a part of CSC Publishers
http://www.cscjournals.org

Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers, 2011

EDITORIAL PREFACE

Robots are becoming part of people's everyday social lives - and will increasingly become so. In future years, robots may become caretaking assistants for the elderly or academic tutors for our children, or medical assistants, day care assistants, or psychological counselors. Robots may become our co-workers in factories and offices, or maids in our homes. It is the fourth issue of volume first of International Journal of Robotics and Automation (IJRA). IJRA published six times in a year and it is being peer reviewed to very high International standards.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting with volume 2, 2011, IJRA appears in more focused issues. Besides normal publications, IJRA intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

IJRA looks to the different aspects like sensors in robot, control systems, manipulators, power supplies and software. IJRA is aiming to push the frontier of robotics into a new dimension, in which motion and intelligence play equally important roles. IJRA scope includes systems, dynamics, control, simulation, automation engineering, robotics programming, software and hardware designing for robots, artificial intelligence in robotics and automation, industrial robots, automation, manufacturing, and social implications etc. IJRA cover the all aspect relating to the robots and automation.

The IJRA is a refereed journal aims in providing a platform to researchers, scientists, engineers and practitioners throughout the world to publish the latest achievement, future challenges and exciting applications of intelligent and autonomous robots. IJRA open access publications has greatly speeded the pace of development in the robotics and automation field. IJRA objective is to publish articles that are not only technically proficient but also contains state of the art ideas and problems for international readership.

In order to position IJRA as one of the top International journal in signal processing, a group of highly valuable and senior International scholars are serving its Editorial Board who ensures that each issue must publish qualitative research articles from International research communities relevant to signal processing fields.

IJRA editors understand that how much it is important for authors and researchers to have their work published with a minimum delay after submission of their papers. They also strongly believe that the direct communication between the editors and authors are important for the welfare, quality and wellbeing of the Journal and its readers. Therefore, all activities from paper submission to paper publication are controlled through electronic systems that include electronic submission, editorial panel and review system that ensures rapid decision with least delays in the publication processes.

To build its international reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJRA. We would like to remind you that the success of our journal depends directly on the number of quality articles submitted for review. Accordingly, we would like to request your participation by submitting quality manuscripts for review and encouraging your colleagues to submit quality manuscripts for review. One of the great benefits we can provide to our prospective authors is the mentoring nature of our review process. IJRA provides authors with high quality, helpful reviews that are shaped to assist authors in improving their manuscripts.

Editorial Board Members
International Journal of Robotics and Automation (IJRA)

Editorial Board

TABLE OF CONTENTS

Volume 2, Issue 3, August 2011

Pages

Reactive Navigation of Autonomous Mobile Robot Using Neuro-Fuzzy System

Maulin M. Joshi maulin.joshi@scet.ac.in
Department of Electronics & Communications,
Sarvajanik College of Engineering and Technology
Surat,395001, India

Mukesh A Zaveri mazaveri@coed.svnit.ac.in
Department of Computer Engineering
Sardar vallabhbhai National Institute of Technology
Surat, 395007, India

Abstract

Neuro-fuzzy systems have been used for robot navigation applications because of their ability to exert human like expertise and to utilize acquired knowledge to develop autonomous navigation strategies. In this paper, neuro-fuzzy based system is proposed for reactive navigation of a mobile robot using behavior based control. The proposed algorithm uses discrete sampling based optimal training of neural network. With a view to ascertain the efficacy of proposed system; the proposed neuro-fuzzy system's performance is compared to that of neural and fuzzy based approaches. Simulation results along with detailed behavior analysis show effectiveness of our algorithm in all kind of obstacle environments.

Keywords: Reactive Navigation, Mobile Robot, Neural Network, Behavior Analysis, Discrete Sampling

1. INTRODUCTION

Autonomous robot navigation means the ability of a robot to move purposefully and without human intervention in environments that have not been specifically engineered for it [1]. Autonomous navigation requires a number of heterogeneous capabilities like ability to reach a given location in real time to unexpected events, to determine the robot's position; and to adapt to the changes in the environment. For a mobile robot to navigate automatically and rapidly, an important factor is to identify and classify mobile robots' currently perceptual environment [1]. The general theory for mobile robotics navigation is based on a following idea: robot must *Sense* the known world, be able to *Plan* its operations and then *Act* based on the model.

In spite of impressive advances in the field of autonomous robotics in recent years, it is still the area of an active research because of uncertainties involved due to unknown environments in real world scenarios. These uncertainties are due to following reasons [1]: no information or less information about a prior knowledge of an environment, lack of perceptually acquired information, limited range, adverse observation conditions, complex and unpredictable dynamics. It is also required that the behavior of the robot must be reactive to dynamic aspects of the unknown environments and must be able to generate robust behavior in the face of uncertain sensors, unpredictable environments and changing scenario.

Many approaches have been proposed to solve the above mentioned challenges for autonomous robot navigation. Some of the approaches focus on path planning methods [2], few approaches use potential field [3] in which the robot-motion reaction is determined by the resultant virtual force. Several other methods have been used like statistical methods, Partially Observable Markov Decision Process (POMDP) [4] and reinforcement learning schemes [5]. In last few years, research in the domain is more focused with neural and fuzzy based artificial intelligence based approaches because of their ability to mimic human expertise.

Humans have a remarkable capability to learn and perform a wide variety of physical and mental tasks via generalization of perceived knowledge. Neural network based approaches are used in robot navigation applications because neural network learns the humanoid expertise and then tries to mimic them by implementing in environment which may be similar or even different than used in its training (i.e. generalization). The attractive potential force attracts the robot toward the target configuration, while repulsive potential forces push it away from obstacles. The mobile robot is considered moving under the influence of resultant artificial potential field. The advantage of neural based approach lies in the learning capacity of the neural network. Performance of neural based system depends upon the effective training of its adjustable parameters (synaptic weights and bias parameters). Dahm et al. [6] have introduced a neural field based approach on robot ARNOLD. The approach was described by non linear competitive dynamical system. However, kinematics constraints were not considered for activation of set of artificial neurons. Zalama et. al. [7] have proposed reactive behavioral navigation of mobile robot using competitive neural network. The authors described various interconnected modules to generate wheel velocity using neural network. However, in such mechanisms many times learning convergence is very slow and generalization is not always satisfactory. A neural dynamics based architecture proposed by Yang and Meng [8]-[9] have discussed to reduce the computational complexity by avoiding learning procedures and also stability has been proven by lyapunov function and qualitative analysis. However, biologically inspired this neural method did not considered sensor information fusion and behavior combination. Some of earlier models are not found practical as they assumed that the whole workspace is definitely known considering only static environment.
Humans' capability to perform various tasks without any explicit measurements or computations is mimicked by fuzzy logic by providing formal methodology for representing and implementing the human expert's heuristic knowledge and perception based actions. Fuzzy logic based many approaches have been investigated in past years for controlling a mobile robot because of its capability to make inferences under uncertainty [10]. Artificial potential field approach has been proposed by Khatib[11] that discussed behavior based control. Saffoitti[12] has proposed fuzzy based methods for mobile robot navigation. Ismail and Nordin [13] have proposed reactive navigation by considering two separate fuzzy controllers for velocities and steering angle. In all these approaches, the purpose was restricted for fundamental and simple control actions. Fuzzy velocity control of mobile robot has been discussed by Mester [14]. However, only 10 heuristic fuzzy rules were used in their experiments. These approaches have inherent drawback that much efforts are needed to adjust tuning parameters and firing in advance. Intelligent navigation systems for omni directional mobile robots were described by Zavalang et.al.[15], which was influenced by potential field approach. Ishikawa [16] and Wei li et al.[17] have proposed behavior fusion for robot navigation in uncertainty using fuzzy logic. Both these approaches need improvements to handle complex environment. A system integrating techniques like dead–reckoning, self localization and environment are reported by Lee and Wu in [18]. In their approach membership functions and fuzzy rules were designed based on genetic algorithm. However, Genetic algorithm may not the best method for generation of rule base with 25 rules and priority based selection of heading directions does not take into account the behavior coordination and this algorithm focuses on direction control without considering velocity control. An obstacle avoidance approach using fuzzy logic has been proposed by Li and Yang [19]. A collision-avoidance approach using fuzzy logic is introduced by Lin and Wang [20] where, different modules e.g. Static-obstacle avoiding module, avoiding moving obstacle module and directing-toward-target module are created for the robot navigation. However, these modules are separately inferred and are not as coordinated as human reasoning. In mobile robots reactive navigation, key problem of local minima is addressed by Zhu and Yang [21] with state memory strategy; Wang and Liu [22] with minimum risk approach and by Xu and Tso [23] by considering π radian target switching. O.R.E. Motlagh et.al.[24] proposed virtual target switching strategy to resolve multiple dead end to improve the performance of earlier methods by considering three target states and six obstacle states resulting into 18 rules. However, with the limited number of rules such improvement not always guaranteed in dynamically changing environment with change in dead end shapes.

To improve the performance, some neuro fuzzy methods have been proposed. Song and sheen [25] have considered heuristic fuzzy- neuro network to reactive navigation of mobile robot. In their approach, resulted velocity command enabled robot to move in an unknown environment using Fuzzy Kohonen Clustering Network (FKCN). However, their heuristic approach considered nine typical obstacle classes to formulate total 16 rules. Wei li et. al.[26] have proposed two level neuro-fuzzy architecture for behavior based mobile. In that approach, neural training has been done by four layer standard back propagation network and used only few selected examples to train neural network. However, in both above approaches; generalization of neural network for complete input space with limited training examples can not be guaranteed. Marichal et al. [27] have suggested an-other neuro fuzzy strategy by considering a three-layer neural network with a competitive learning algorithm for a mobile robot. The approach has been able to extract the information for fuzzy rules and the membership functions from human guided set of trajectories. For complex situations, it is difficult to optimally set required trajectories and hence resulted rules may not work well for generalization. Zhu and yang [28] have proposed five layer neuro-fuzzy controller considering neural networks to improve the performance of fuzzy network. The approach includes an algorithm to surpass redundant rules by observing the response of fuzzy network and removing rules with hamming distance lesser than specified threshold. However, this obviously requires training to mobile robot in given environment. However, for some critical operations like mining and under water operations, such training in given environment is never possible. Approaches without proper generalization will fail to take best decision when mobile robot needs to take immediate actions without any prior scanning of the given environment. Heuristic based approaches do not guarantee satisfactory performance for in general, difficult unknown environment space.

In this paper, we propose two level neuro fuzzy based algorithm that overcomes the shortcoming of current approaches [25-28] in terms of learning mechanism used. In the proposed system, environment sense is done by neural network and behavioral control is executed by fuzzy system. Inputs to the neural network are outputs from multi sensors groups and heading angle. Output of neural network is reference heading angle that in connection with sensors data serves as input to fuzzy system. We propose discrete sampling based approach, in which optimal neural training is achieved by providing effective heterogeneity in training pairs while; retaining homogeneity in terms of providing different training pairs to the neural network. In our approach, we have generalized many parameters for robot navigation task like; number of sensors required for environmental sensing, arrangement of sensors, sensors grouping and quantization and heading angle inference. These make our approach unique and more generalized compared to approaches found in literature. Generalization of fuzzy based parameters enables us to select, to tune parameters as per requirements of given environmental conditions. Behavior based fuzzy systems used for mobile robot navigation demonstrate reasonably good performance; while navigating in cluttered and unknown environment

The rest of this paper is organized as follows: the proposed algorithm for neuro-fuzzy based mobile robot navigation is discussed in Section 2, including range computation from given obstacles, sensors arrangement, grouping quantization and inference of heading angle. Section 3 descries neuro-fuzzy system for reactive navigation. Section 4 illustrates simulation results and detailed behavior analysis of neuro-fuzzy based mobile robot navigation. Finally concluding remarks are given in Section 5.

2. PROPOSED ALGORITHM

In this section, we propose an algorithm for reactive navigation for a mobile robot using neuro-fuzzy based sys-tem. First, we describe the problem formulation for the motion planning problem. Let A be the single robot moving in a Euclidian space W, called workspace, represented as R^N, with N= 2 or 3. Let B_1, B_2…B_q be the rigid objects distributed in W. The Bi's are called obstacles. With assumptions that no kinematics constraints limit the motion of A in W, generate a path T specifying a sequence of positions and orientations of A avoiding contact with Bi's, i.e. starting at the initial position and orientation and terminating at the goal position and orientation.

2.1 Mobile Robot Configuration

We consider two dimensional workspace (N=2) for mobile robot as shown in Figure.1. Mobile robot is having initial and target position coordinates denoted as (x_o, y_o) and (x_t, y_t) respectively. Mobile robot's current position (calculated and updated at each step) can be denoted as (x_{curr} ,y_{curr}). Angle between target and positive y axis is θ_{tr}. Robot's pose (head) with respect to positive y axis is considered as θ_{hr} and θ_{head} is the heading angle between target and robot current position. Span (S) is the distance between left and right wheel. V_l and V_r are mobile robots left wheel and right wheel velocities, respectively.

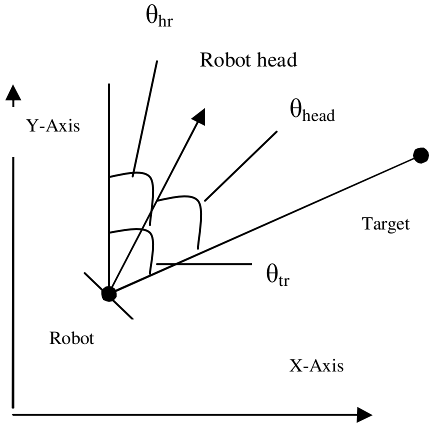


FIGURE 1: Mobile Robot Configuration

The mobile robot has two independently driven co-axial wheels. We consider a mobile robot with differential drive wheels. Final target positions are known to the robot at all the time. At each step, current location and orientation are computed. No history of past sensor readings are retained and thus robot is having pure reactive navigation. Obstacles may be stationary or may be mobile.

2.2 Range Calculation of Mobile Robot From Given Obstacles

Acquisition of precise range information of a mobile robot from each nearby obstacle is one of the most important tasks for robot navigation. Mobile robot needs to effectively sense surrounding environment. We have proposed an algorithm to find range information for robot navigation in presence of moving obstacles in our earlier work [29]. The important point is that because of presence of moving obstacles, prior geometry information may not help. But, our model acquires geometry information from sensed signals computed with the help of sensors mounted on robot. This makes our approach very general and can be used for any scenario. Following steps explains our algorithm to find out range of obstacle from robot A to Obstacle B:

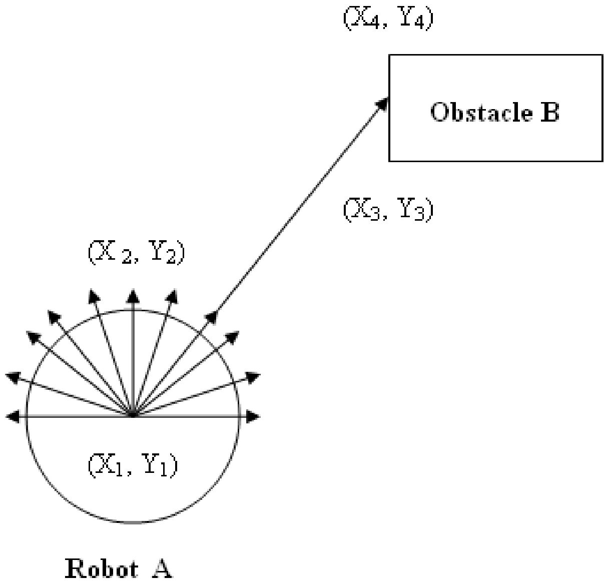


FIGURE 2: Range Calculation of a Mobile Robot from Obstacles

1) Let total N ultrasonic sensors be placed on robot to sense the surrounding environment as shown in Fig-ure.2. These sensors are represented as N_1,N_2... N_k, $N_{k+1..}$ N_N. Signal of kth sensor (N_k) is represented by an arrow towards the obstacle.

2) Let, (x_1, y_1) and (x_2, y_2) are two points on robot to represent kth sensor direction. The ray emerging from sensor mounted on mobile robot to obstacle can be considered in terms of parametric equation form of straight line as:

$$x= x_1+ (x_2- x_1) D_k$$
$$y= y_1+ (y_2- y_1) D_k \qquad (1)$$

Where, D_k is a real value that denotes the distance of a mobile robot from obstacle. In order to ensure that robot looks only in forward direction and the maximum range of ultrasonic sensor is set to D_{max},

$$0 < D_k< D_{max} \qquad (2)$$

3) Small line segment on an obstacle will be represented by points (x_3, y_3) and (x_4, y_4). This line segment will intersect with ray emitted by the sensors on robot. Particularly this line segment being very small can be considered as straight line segment. This assumption will allow us to calculate range for any shape obstacle in our algorithm. Consider (x_3, y_3) and (x_4, y_4) be two points representing one line segment on the ith obstacle and described by parametric equation form of straight line as:

$$x = x_3+ (x_4- x_3) S_{ij}$$
$$y = y_3+ (y_4- y_3) S_{ij} \qquad (3)$$

Where, S_{ij} - a real value presenting line segment of ith obstacle's jth side. To ensure that a particular ray emitted by sensor mounted on the robot hits the line segment (side of the obstacle); the value of S_{ij} should be between 0 and 1, i.e.

$$0 <= S_{ij} <= 1 \qquad (4)$$

4) Solution of equations (1) and (3) will give us the distance D_K, i.e. distance between robot's kth sensor to the ith obstacle's jth side:

$$D_K = \frac{((y_4\text{-}y_3)*(x_1\text{-}x_3)\text{-}(y_1\text{-}y_3)*(x_4\text{-}x_3))}{((y_2\text{-}\ y_1)*(x_4\text{-}x_3)\text{-}(y_4\text{-}y_3)*(x_2\text{-}x_1))} \qquad (5)$$

5) Computation of the value of D_K is to be carried out for each of total N sensors.

For example, rectangle shaped n obstacles will have 4*n edges. For total N sensors, there will N* (4* n) size matrix computed at each step.

Sensors Grouping

In our algorithm, we consider robot fitted with N ultrasonic sensors in the front. If the front (head) of the robot is at 0 degrees (w.r.t. +y axis), then the sensors are located between -90 to +90 degrees each being separated by θ_s =180/N degrees as shown in figure. 3(a).

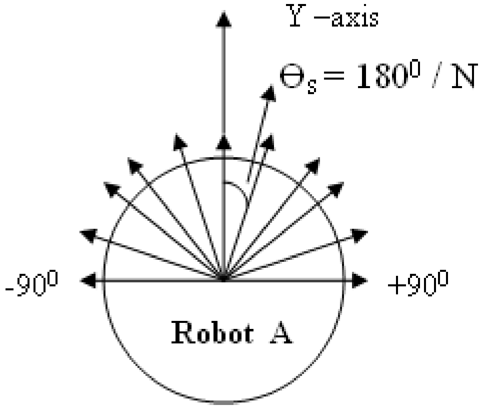


FIGURE 3(a): Arrangement of ultrasonic sensors

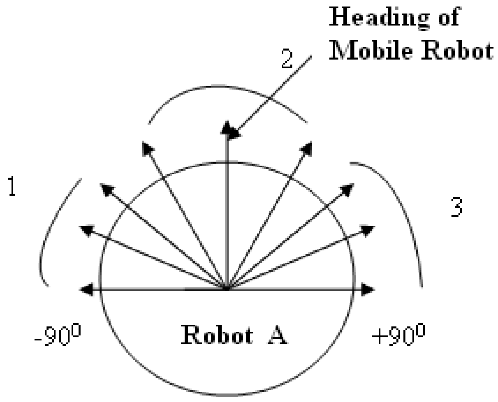


FIGURE 3(b): grouping of sensors

Sensors are grouped and final values are quantized before sending into the intelligent network. A sensor grouping will result into reduction of computational cost. In our algorithm, we sense unknown environment with N Sensors to extract more information about surroundings. At the same time, we resize sensors into M groups (M< N) before giving as input to intelligent system to reduce computational complexities still retaining the essence of more information. As the final value for each of M group, minimum value among the corresponding sensors readings are taken and then fed to intelligent system module. For figure.3 (b) Considering d(i)–ultrasonic data for ith sensor; distances to the obstacles may defined as below:

Left_obs = min{d (i)} where, i= 1,2…x.
Front_obs = min{d (i)} where, i= x+1,x+2 …y.
Right_obs = min{d (i)} where, i= y+1, y+2,.. N (6)

2.3 Quantization of Sensors Values

In our approach, we perform quantization to provide discrete samples for neural training. Quantization formula for groups (Xi) where, i=1, 2...M (M<=N) is as follows:

Xi = 1 for 0< d_i <=D_1,
2 for D1 < d_i <=D_2,
3 for D2 < d_i <=D_3,
……………. . .
Z for d_i >D_Z. (7)

Where, di is the minimum sensor value of the ith group and D_1, D_2 … D_Z are threshold values for quantization.

2.4 Defining Heading Angle

We define heading angle (θ_{head}) as follows:

- If $\theta_{head} < p$ then $\theta_{head} = \alpha$,
- If $p <= \theta_{head} <= q$ then $\theta_{head} = \beta$,
- If $\theta_{head} > q$ then $\theta_{head} = \gamma$ (8)

Once surrounding environment sensing is completed; set of information is available for planning. Next step is to train intelligent system with these set of information. As stated earlier, neuro-fuzzy systems have abilities to learn and then perform intelligent task based on learning. Next subsection describes neuro-fuzzy based system.

3. NEURO-FUZZY SYSTEM FOR REACTIVE NAVIGATION

Neural and Fuzzy based hybrid systems have been used in many applications in order to take advantage of individual systems. This motivates us to use combined neuro-fuzzy system for reactive navigation of a mobile robot in the presence of obstacles. We propose two stage, hybrid neuro-fuzzy system in which information from an environment (Sense) is obtained by neural networks while; more correct decisions (Act) are performed using a fuzzy system. As far as environment understanding (Sense) is concerned; neural network will be more effective candidate; as it gives computationally cost effective solutions than fuzzy system [30]. On the other had, we require tight control to exert final wheel velocity where fuzzy system would be better choice because of its functional mapping ability [31]. Our proposed framework provides an optimum learning of neural networks via discrete sampling that overcomes the problems faced by existing neuro fuzzy approaches based on experimental and heuristic bases training [25-28].

We consider two stages neuro-fuzzy based hybrid architecture as shown in Figure.4. In our proposed neuro-fussy system, the inputs from the sensors are fed to neural network which forms first stage of the proposed sys-tem and it is cascaded with a fuzzy system to generate final control action. First stage neural network has four inputs. Out of four inputs, three inputs are the distance information from the left, front and right obstacles pre-sent in robot's perceptual environment. The fourth input is the heading angle. As an output, neural network generates Reference Heading Angle (RHA); an inferred angle than original head angle. During this process , as heading angle inference is already been processed by neural network and bettered with the support of local sensory data, resulted reference heading angle imparts better information to the subsequent fuzzy system than an actual heading angle. Heading angle is one of critical parameters and should be inferred correctly for reactive robot navigation. The neural network does this task and provides a reference heading angle as an output. In the second stage, fuzzy logic processes this information and drives the output wheels of mobile robot. Outputs of the system are left and right wheel velocities. Input sensory information's cardinality for the neural and fuzzy networks can be shared or can be set to higher value for neural network to take maximum advantage of its learning capabilities.

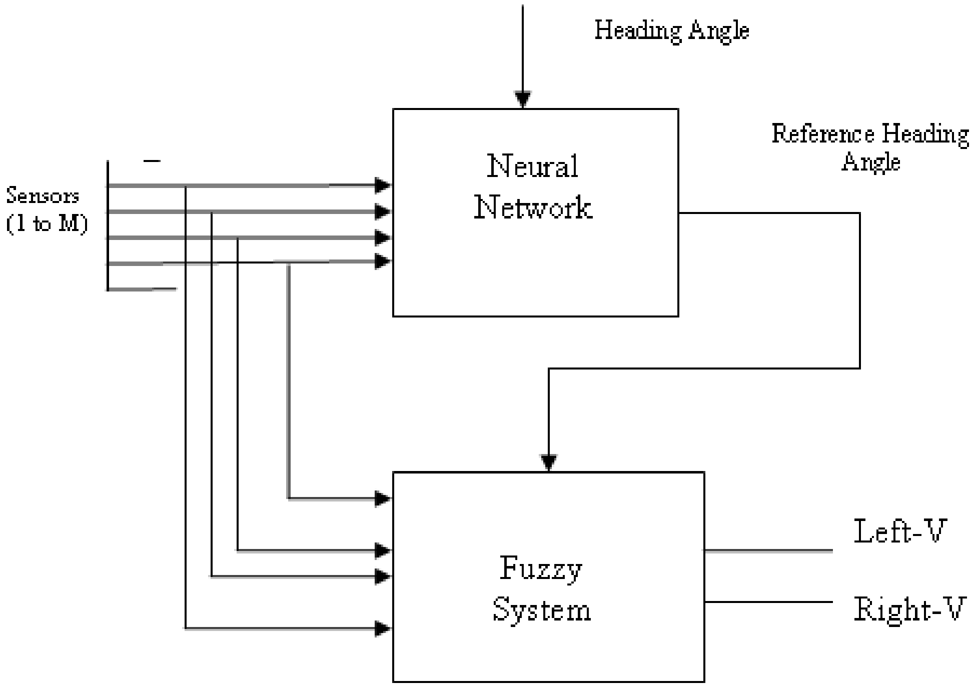


FIGURE 4: Two stage Neuro-Fuzzy System

3.5 Training Using Neural Network

In our framework, neural network has M inputs (one per each of sensor group) derived from grouping of N sensors and giving the distance information about robot's perceptual environment. One more input given to the neural network is the heading angle. Neural network processes these inputs and generates a reference heading angle. Neural networks have got remarkable generalization capabilities, once trained properly. Training of intelligent system is crucial for successful navigation of mobile vehicle. Generally, it is difficult to train such system as the input space may contain infinitely many possibilities and mobile robot needs to learn effectively for successful navigation. Many times mobile robot needs to execute operations in hazardous environments like fire or space missions where, online training is not feasible. Off line training is only possibility in such cases. Mobile robot needs to sense environment in real time and also to make precise decision based on learning.

Various training approaches reported in literature are of following categories: a) generating training sequences by experimental set up as in [17] and b) heuristic approach based on expert rules [25]. In the first approach, the system learns by setting the different environmental set ups. i.e. different start, end (target) positions, different obstacles positions etc. In this case, the number of training patterns resulted for different input conditions may not be evenly distributed. Some of the input patterns may appear more number of times, while some may appear lesser or even may not appear. Training may not be considered optimum as; for some inputs patterns are not learnt while some are over learnt. In case of second alternative (training by expert rules [8]), training is per-formed using fewer number of input patterns. This type of training may save training time, may give good performance in some cases but, they may not perform well in all kind of environmental conditions. This is because of the fact that selection of training pairs is for a particular task and they do not represent entire space uniformly.

In this paper, we propose mobile robot's training based on discrete uniform sampling that overcomes the problems with above mentioned methods. The proposed algorithm not only takes samples from the entire sample space (to provide heterogeneity), also takes equal number of sample data from all possible input space (to pro-vide homogeneity). In the proposed algorithm, actual sensor readings are considered to be quantized in to n linguistic values. Uniform sampling of these quantized values will enable us a) to consider entire space of input region and; b) will enable us to generate optimum number of training pairs required for training. In the proposed approach, we train the network as follows:

1. Let input cardinality (number of inputs i.e sensors plus heading angle) of the neural networks equal to M+1 and each input takes Z linguistic values (e.g. near, medium, far- as discussed in earlier section). Then we can generate total Z^{M+1} training patterns.
2. Output values of each of these input patterns are decided based on experimentation or by expert rules.
3. Neural network is trained accordingly to training pairs generated and the performance of the network can be verified using a proper evaluating function e.g. MSE (mean square error).
4. If any correction is required; make adjustment to step 2 and then repeat steps.

3.5 Fuzzy System (FS)

Fuzzy logic provides a formal methodology for representing and implementing the human expert's heuristic knowledge and perception based actions. We utilize the fuzzy system as shown in Figure.4. Out of total four in-puts, three inputs are the distance information from the left, front and right obstacles present in robot’s perceptual environment. The fourth input is the reference heading angle. Outputs of the network are Left and right wheel velocities. Fuzzy system needs to define the membership functions for input and output variables. These membership functions for input and output variables are defined in Table 1 and Table 2 respectively. Linguistic values near, med (medium) and far are chosen to fuzzify left_obs, front_obs and right_obs. We define linguistic values slow, med (medium) and fast to show output parameters left and right velocities.

Order	Linguistic Values	Membership Function	Corresponding fuzzy numbers
1	Near	Trapezoidal	[0.01, 0.01, 1.5, 2.0]
2	Medium	Triangular	[1.5, 2.0, 2.5]
3	Far	Trapezoidal	[2.0, 2.5, 4.0, 4.5]

TABLE 1: TRANSFORMATION RULES FOR FUZZY INPUTS

Order	Linguistic Values	Membership Function	Corresponding fuzzy numbers
1	Slow	Trapezoidal	[0.01, 0.01, 1.5, 2.0]
2	Medium	Triangular	[1.5, 2.0, 2.5]
3	Fast	Trapezoidal	[2.0, 2.5, 4.0, 4.5]

TABLE 2: TRANSFORMATION RULES FOR FUZZY OUTPUTS

3.5 Behaviors Fusion based on Fuzzy Reasoning

Mobile robot moves in a given environment from start position to the end position. In order to avoid obstacles in its path, reactive navigation is performed in response to the sensor data perception. In order to coordinate different type of behaviors, various methods are available: i) priorities based data fusion ii) inhibiting strategy and iii) behavior based fuzzy reasoning. In priority based fusion, certain rules are always given priorities compared to others which may not be true always. In second case, when multiple rules are fired simultaneously, few rules are dominating and hence other rules are inhibited at the particular stage. In both the cases, enough attention may not be given to some rules which in turn may become critical after some period.

In our work, we have used behavior based fuzzy reasoning in which all fired rules are given due weight age according to their firing level. For our proposed method, the following behaviors are realized: Target Steer, Obstacle Avoidance and Edge following. It is very difficult to acquire precise information about dynamic environments through ultrasonic sensors. A set of fuzzy logic rules to describe various behaviors are defined for the proposed system. Table 3 gives few samples of our defined fuzzy rules. These fuzzy rules show that the robot mainly adjusts its motion direction and quickly moves to the target if there are no obstacles around the robot. When the acquired information from the ultrasonic sensors shows that there are no obstacles to the left, front or right of robot, its main reactive behavior is target steer. When the acquired information from the ultrasonic sensors shows that there exist obstacles nearby robot; it must try to change its path in order to avoid those obstacles (i.e. Obstacle Avoidance behavior). When the robot is moving to a specified target inside a room or escaping from a U-shaped obstacle, it must reflect Edge Following behavior.

If input						then output	
Rule no..	*Fuzzy Behaviour*	*Left Obs.*	*Front Obs*	*Right Obs.*	*Head ang*	*Left Vel*	*Right Vel*
1	Target Steer	Far	Far	Far	Negative	Low	Fast
2	Target Steer	Far	Far	Far	Zero	Fast	Fast
3	Target Steer	Far	Far	Far	Positive	Fast	Low
4	Obstacle Avoidance	Near	Near	Far	Negative	Fast	Low
5	Edge Following	Far	Far	Near	Positive	Med	Med

TABLE 3: Fuzzy If-then rules

4. SIMULATION RESULTS

In this section, we demonstrate the effectiveness, robustness and comparison of various systems for robot navigation using single stage neural network, single stage fuzzy system and our proposed hybrid neuro-fuzzy system. We have considered mobile robot having differential drive mechanism with span of mobile robot 50 cm. Total 9 ultrasonic sensors (N) are used for the study after comparing the results for 5, 9 and 15 no. of sensors and their effectiveness. These sensors are equally separated by $\theta_s = \pi/8$ and detect the distance of obstacle along the radial direction up to 300 cm. The wheels can have a maximum velocity up to 30 cm/s. Input dimensions to the neural, fuzzy and neuro-fuzzy system are set to four. D_{min} is set to 100 cm and D_{med} is set to 200cm. In order to define heading angle (θ_{head}), we have set the values of p, q, α, β, γ to $-\pi/8$, $+\pi/8$, 1, 2, and 3 respectively. For our simulation we use two layers feed forward back propagation network (FF- BPN) for mapping the input quantized values to the output. Batch mode of training is used for neural network.

For neuro-fuzzy system, we have trained first stage neural network by considering 4 inputs as described earlier and each input takes 3 linguistic values (near (1), medium (2), far (3)). Hence, total 81 training pairs are generated and used for optimal training of neural network. For the second stage of our neuro-fuzzy architecture, i.e., fuzzy system, the fuzzy rules are generated using 3 linguistic values and 4 inputs. Total 3 groups are formed in order to give them as inputs to fuzzy system module. We use the minimum value among the corresponding sensors' readings as the final value for that group of sensors which is fed to the system module. Left, front and right obstacles are equally important inputs to the fuzzy systems. The fourth input to the fuzzy network is RHA which is output of neural network stage. The membership function values are fine tuned by simulating the navigation in many different setups and correcting the errors over number of experiments. For fuzzy reasoning Min – Max (Min- for the implication and Max- for aggregation) are used. De-fuzzification is done using centroid method. Using fuzzy reasoning, various behaviors are weighted to determine final control variables i.e. left and right velocities.

As stated earlier, we compare it with single stage neural network and single stage fuzzy based systems. Figure.5 shows the path comparison of a mobile robot between single stage neural [30] and fuzzy approaches [31] while; figure.6 depicts the mobile robot path comparison between neural and proposed neuro-fuzzy based systems. These results suggest that, in the case of second stage (driving stage), fuzzy systems are preferred. This is because of the fact that the neural network's output in the unexplored regions of inputs is not predictable and error at each stage gets accumulated and hence, do not give good and stable path.

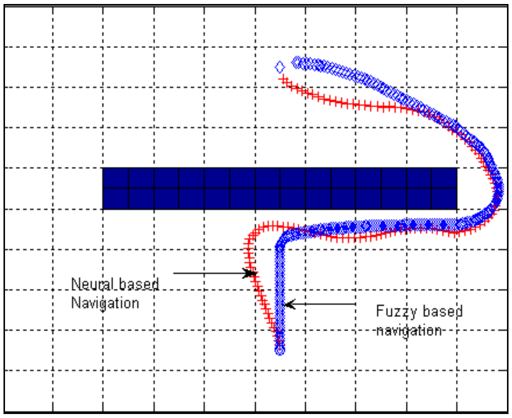


FIGURE 5: Comparison of Robot navigation: Neural & Fuzzy system

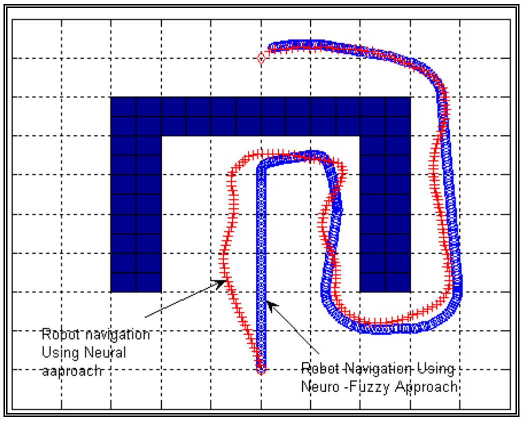


FIGURE 6: Comparison of Robot navigation: Neural and Neuro Fuzzy system

Figure.7 illustrates robot navigation with fuzzy system [31] while; figure. 8 shows robot navigation with proposed neuro-fuzzy system. Comparing the results, it is found that in figure.7 robot eventually strikes with the obstacle located to the left bottom corner while with the same scenario; the Neuro- fuzzy system avoids the same obstacle successfully. It is because of the fact that in the case of a single stage fuzzy system, one of the inputs (i.e. heading angle) contradicts to the perception by the other inputs while; in the case of neuro-fuzzy system (as shown in figure.8) the RHA, which is an inferred heading angle, has been proved very useful input to the fuzzy system. The use of neural network as first stage in neuro–fuzzy system architecture provides better inference of an environment using the sensed input values. The simulation results highlight the fact that adding the neural stage enhances environmental sensing capacity of the fuzzy system. The same fact is observed from the outputs of various experiments performed in different environmental conditions.

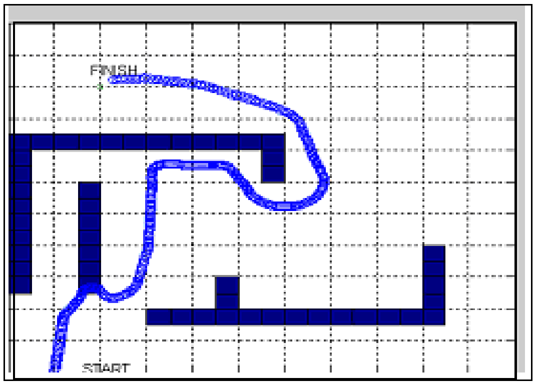


FIGURE 7: Robot navigation with single stage fuzzy system

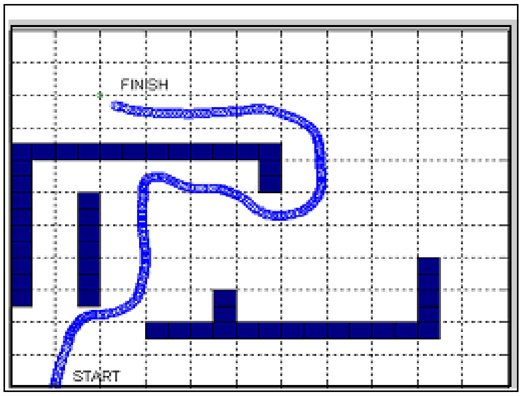


FIGURE 8: Robot navigation with two stage Neuro-Fuzzy system

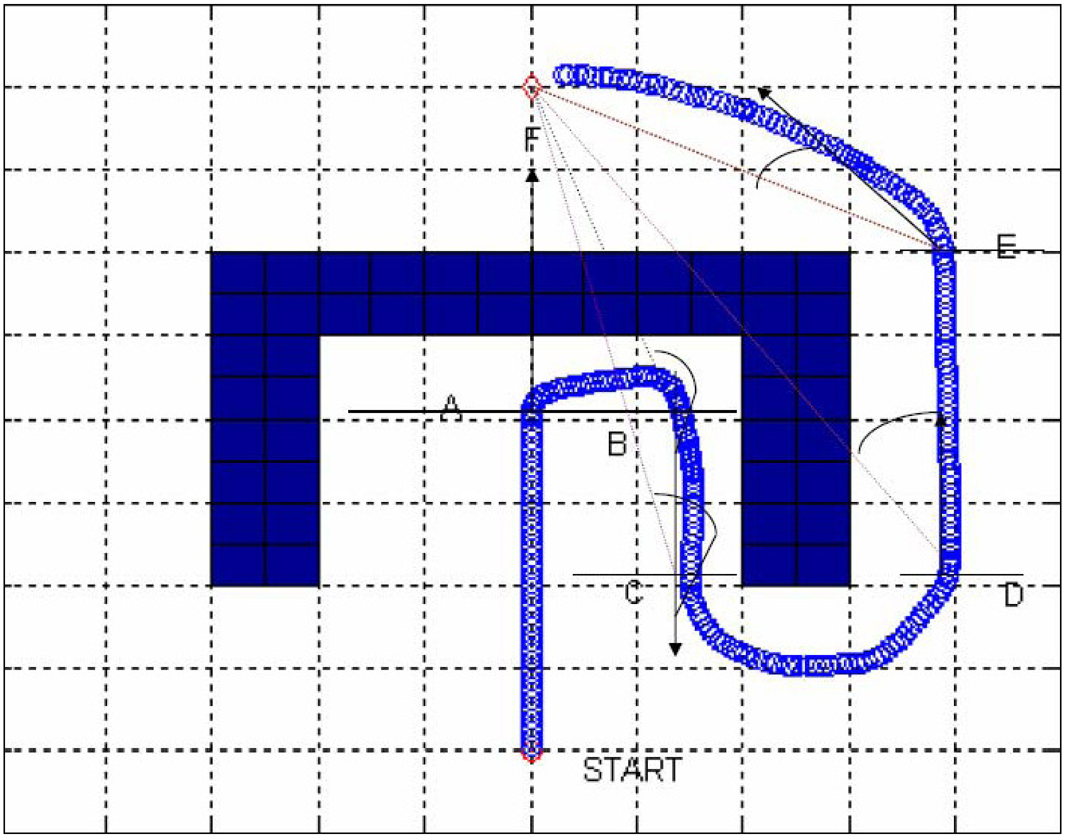


FIGURE 9: Neuro-fuzzy based mobile robot navigation

Next subsection presents the detailed behavior analysis of proposed neuro-fuzzy based systems that highlights the effectiveness of our proposed system in given environment.

A Various Fuzzy Based Behaviors and Heading Angle

Consider mobile robot navigation for the case shown in figure.9. Mobile robot starts its journey from position “START” to the final position “F”. For a given scenario, mobile robot follows path from START-> A -> B -> C -> E -> F as shown in figure.9. Figure.10 shows mobile robot’s heading angle during its journey. Head-ing angle is the difference between target and head of the mobile robot and provides information about current head orientation. Initially, robot performs "Target steer" behavior and reaches to position "A" with "ZERO" heading angle, where it takes right turn which is a result of "avoid obstacle" behavior; and heading angle changes quickly to -90. There after, the robot follows the wall, i.e., the wall following behavior, and it reaches to "B". At the same time, heading angle slightly varies from -90 to -100 degrees. At this point "B", it takes right turn again (head angle (equal to α) changed to -180) and following the wall and reaches to "C" by decreasing heading angle (equal to β) further to -200 degrees. Mobile robot finds the end of the wall and perceives potential attraction by the target and takes left turn by avoiding contact with obstacle and reaches to "D". During the same, the heading angle (equal to γ) starts increasing to -45 degrees. From position "D" to "E" it continues its quest to reach target following wall (heading angle (decreases slightly from -45 to -50 degree (δ)), finds opening at "E". Finally mobile robot reaches from "E" to "F" with "target steer" behavior (first reducing head angle to near zero and then with almost zero head reaches target "F").

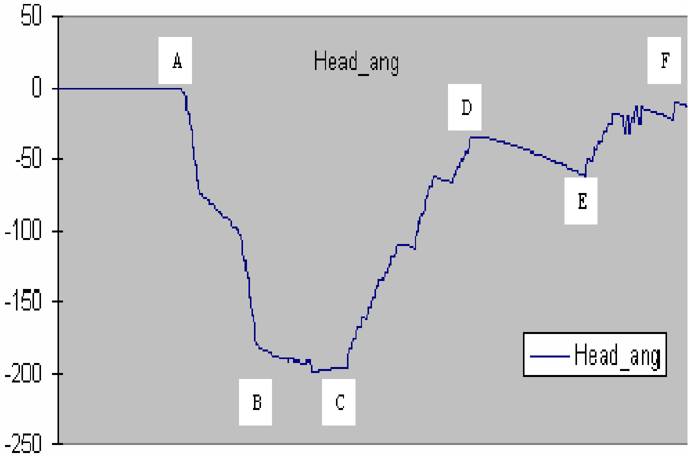


FIGURE 10: Mobile robot's heading angle

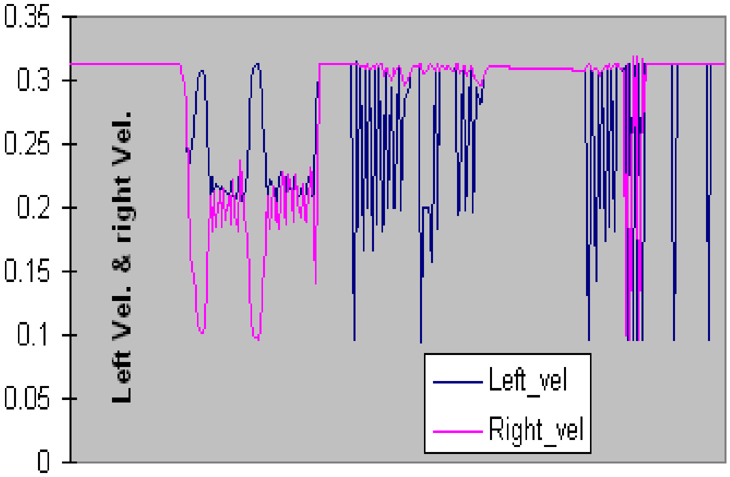


FIGURE 11: Left and right velocity control over time for mobile robot navigation

Speed Control of Mobile Robot

In mobile robot navigation, speed control analysis gives information about and robot's left and right velocities over the time. Figure.11 shows left and right wheel speed control. In differential drive mechanism, to take right turn; robot increases its left velocity and decreases right velocity and vice versa. As shown in figure11, from "START" position to point "A", left and right velocities are found same. From “A” to “B” position (first, mobile robot takes right turn and then follows straight line). Hence, initially left velocity is higher than right velocity and then both almost being same till “B”. At “B” robot has left velocity higher again (avoid obstacle- right turn) and then with almost same left and right velocities (wall following) reaches “C”. Journey from “C” to “D” is with right velocity values higher than left velocity values (avoid obstacle /attraction potential - left turn). Mo-bile robot moves straight with same left and right velocity values (wall following) from point “D” to “E”. At last, at point “E”, right velocity values are more than left velocity values (right turn) and finally, it settles to point “F”.

Target Distance and Distance Traveled

Figure.12 shows mobile robot’s target distances and distance traveled over the time period for a case discussed in figure. 9. Start and target (F) coordinates for mobile robots are (10, 2) and (10, 18) respectively. For mobile robot From “START” to “A”, target distance is linearly decreased (START->A). When mobile robot finds an obstacle in between and trying to move out of the same, target distance is gradually increased again till “D” (A->B->C->D). Then onwards, target distance is again linearly decreased on path from D -> E -> F. Second graph in figure.12 shows the total distance traveled a curve with a linear rise from start to finish point. For the given situation, in presence of given obstacle scenario; mobile robot travels total 39.65 meters compared to 16 meters if it had traveled without any obstacles from START to F position (i.e. distance between (10, 2) and (10, 18)) as shown in figure.9.

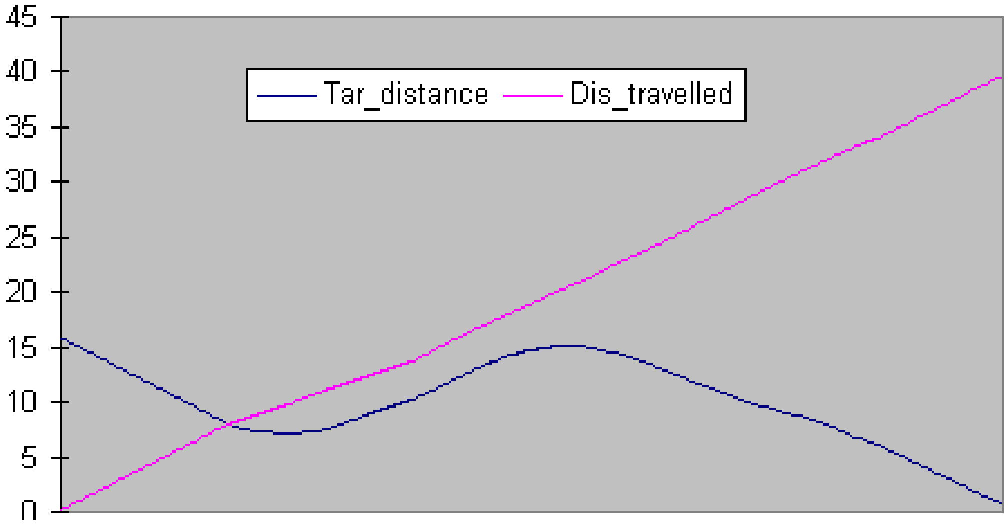


FIGURE 12: mobile robot’s target distance and distance traveled

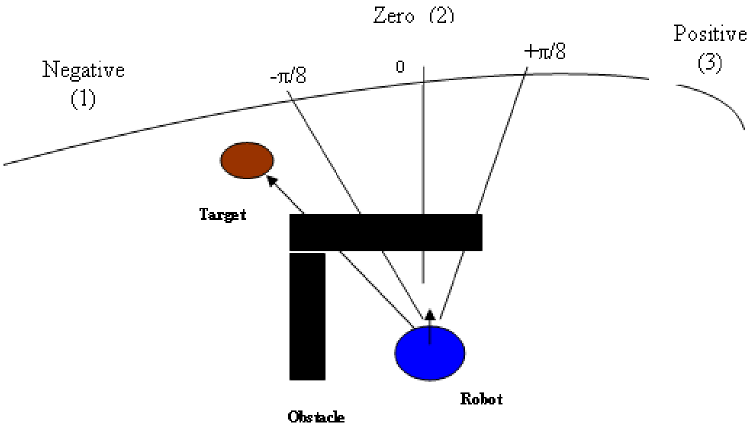


FIGURE 13(a): Typical scenario for obstacle avoidance: case01

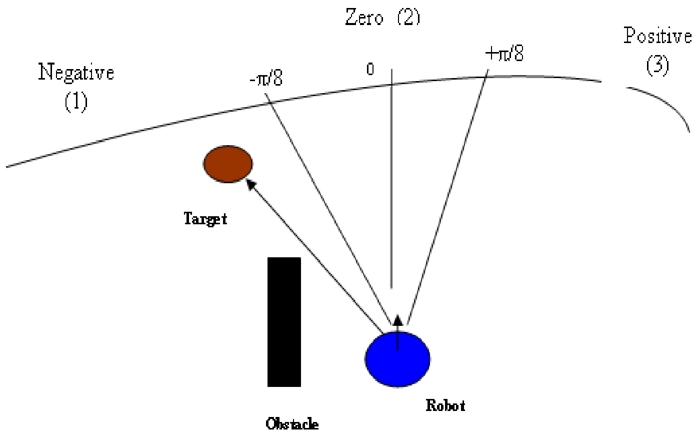


FIGURE 13(b): Typical scenario for obstacle avoidance: case02

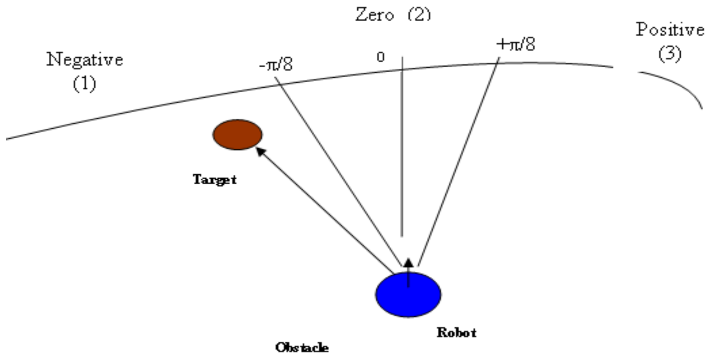


FIGURE 13(c): Typical scenario for obstacle avoidance : case03

Importance of Adding Input Neural Stage

Figure.13 (a-c) shows three different cases for mobile robot navigation which highlights the importance of neural network stage. In first case, current positions of robot, obstacle and target are as shown in Figure.13 (a). Perceptions from mobile robot sensors suggests that left and front obstacles are near, right obstacle is far and head angle is negative. These inputs suggest that mobile robot should move left (because head angle is negative) while; to the left and front there are obstacles at near distances. Here, two senor input groups distances left and front obstacles, contradict to another input, namely, head angle. Due to this fact, a single stage fuzzy system may not be able to take best decision. When neural network is used along with fuzzy system (i.e. neuro-fuzzy), it is observed that the output of neural stage, Reference heading angle (RHA), becomes positive; compared to earlier input, i.e., heading angle which was negative. When this inferred input (RHA) is given to fuzzy system for inference; now instead of contradicting with the rest of inputs it supports the inference by ultrasonic sensors. Hence, this enables fuzzy system to exert better control action.

In second case (as shown in figure.13 (b)), front obstacle is removed keeping other conditions same as in case one; first stage neural network corrects head angle to zero (indicating go straight i.e. wall following behavior) from negative (go left i.e target steer). As a third case (as shown in figure..13 (c)) both the obstacles are re-moved keeping robot, target positions and other conditions are kept unchanged. Here, first stage neural network output suggests to continue earlier perception (i.e. same as input - reference heading angle inference is negative). In summary, neural network suggests different behaviors like avoid obstacle (move to right), wall following (go straight) and target steer (continue left) in all three cases respectively. Inferences made by the neural stage, when in turn given as input to the subsequent fuzzy system, it strengthens fuzzy system's local perception provided by the local sensors. Resulted neuro-fuzzy system performs better than single stage neural or single stage fuzzy systems. These results highlight the importance of adding a neural stage before the fuzzy control stage in the proposed system.

5. CONCLUSIONS

In this paper, an approach for robot navigation using neuro-fuzzy based system is discussed. The mobile robot performs reactive navigation which is very useful for real time, dynamic environment rather than looking for an optimal path as performed by path planning techniques. Fuzzy system architecture for behavior based control of robot navigation gives better performance compared to neural based systems. Neural network's output in the case of unexplored regions of inputs is not predictable and error at each stage is accumulated. As a result it does not lead to good and stable navigation path. The performance of mobile robot navigation system is improved by cascading the neural network and fuzzy system. The simulation results show that RHA provides better inference compared to original heading angle. The behavior based analysis of mobile robot navigation using the proposed neuro-fuzzy system demonstrates the excellent performance in complex and unknown environments. Simulation results for dynamic, complex and cluttered environment of mobile robot navigation space with neuro-fuzzy based architecture demonstrate good performance compared to most recent comparable approaches. This is because of our generalization of most of the parameters likes number of sensors, threshold values to measure distances and heading angles, optimum training using discrete sampling based approach for neural system training.

6. REFERENCES

[1]G. Dudek and M. Jenkin. Computational Principles of Mobile Robotics, Cambridge university press, 2000, pp. 01-40.

[2]C.M. Clark & S. M. Rock. “Motion Planning for Multiple Mobile Robots using Dynamic Networks”, Proc. of IEEE international Conference on Robotics and Automation, 2003, pp.4222-4227.

[3]A. Haddad, M. Khatib, S. Lacroix and R. Chatila. “Reactive Navigation in Outdoor Environment using Potential Fields”, in proc. of IEEE international Conference on Robotics and Automation, 1998, pp. 1232-1237.

[4]G.Theocharous and S. Mahadevan. “Approximate planning with hierarchical partially observable Markov decision process models for robot navigation”, IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA'02. pp 1347– 1352.

[5]S.M.Chun, D.Y. Huang, C.H. Chou and C.C. Hsieh. “A reinforcement-learning approach to robot navigation”, IEEE International Conference on Networking, Sensing and Control, 2004, pp. 665 – 669.

[6]P. Dahm, C. Bruckhoff and F. Joublin. "A neural field approach for robot motion control," IEEE International Conference on Systems, Man, and Cybernetics, 1998, pp.3460-3465

[7]E. Zalama, J. Gomez, M.Paul and J.R. Peran. "Adaptive behavior navigation of a mobile robot," IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, vol.32, pp.160-169, Jan 2002.

[8]S.X. Yang and M. Meng. "Neural network approaches to dynamic collision-free trajectory generation," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 31(3), pp.302-318, Jun 2001.

[9]S.X. Yang and M. Meng. "Real-time collision-free motion planning of a mobile robot using a Neural Dynamics-based approach," IEEE Transactions on Neural Networks, vol.14 (6), pp. 1541- 1552, Nov. 2003.

[10]J.C. Latombe. Robot Motion Planning, kluwer academic publishers- 1991.

[11]M. Khatib, "Real-time obstacle avoidance for manipulators and automobile robots", Int. J. of Robotics Research, vol.05 (1), 1986.

[12]Saffiotti. "The uses of fuzzy logic in autonomous robot navigation", International journal on Soft Computing", vol. 1, pp. 180-197, 1997.

[13]I.I.Ismail and M.F. Nordin. "Reactive navigation of autonomous guided vehicle using fuzzy logic," Student Conference on Research and Development, 2002, pp. 153- 156.

[14]G. Mester, "Obstacle Avoidance and Velocity Control of Mobile Robots", proceedings of 6th international symposium on intelligent Systems and Interpretation, Sep. 2008, pp.1-5.

[15]P.G. Zavlanga, S.G.Tzafestas , K. Althoefer , "Fuzzy Obstacle Avoidance and Navigation for Omni directional Mobile Robots", ESIT, 2000.

[16]S. Ishikawa. "A method of indoor mobile robot navigation by using fuzzy control," Proceedings of Intelligent Robots and Systems IROS '91. vol.2, 3-5, 1991, pp.1013-1018.

[17]W. Li. "Fuzzy Logic based Perception-Action Behavior Control of a Mobile Robot in Uncertain Environments," IEEE International Conference on AI, 1994, pp. 231-235.

[18]T. Lee and C. Wu. "Fuzzy Motion Planning of Mobile Robots in Unknown Environments," presented at Journal of Intelligent and Robotic Systems, pp.177-191, 2003.

[19]H. Li and S. X. Yang. "A behavior-based mobile robot with a visual landmark recognition system," IEEE Trans. Mechatronics, vol. 8, no. 3, pp. 390–400, Sep. 2003.1.

[20]C.H. Lin and L.L. Wang. "Intelligent collision avoidance by fuzzy logic control," Robotics and Autonomous Systems, Volume 20, pp 61-83, 1997.

[21]A. Zhu and S.X. Yang. "A fuzzy logic approach to reactive navigation of behavior-based mobile robots," IEEE International Conference on Robotics and Automation, 2004. vol.5, no., pp. 5045- 5050.

[22]M.Wang and J.N.K.Liu. "Fuzzy logic based robot path planning in unknown environments", in Proceeding of International Conference on Machine Learning and Cybernetics, Vol.2,2005,pp.813–818. 1.

[23]W.L. Xu, S.K.Tso and Y.H. Fung. "Sensor-based reactive navigation of a mobile robot through local target switching," Proceedings of 8th International Conference on Advanced Robotics, ICAR '97, 1997, pp.361-366.

[24]E.O.Motlagh, T.S.Hong and N.Ismail. "Development of a new minimum avoidance system for a behavior-based mobile robot," Proceedings of international journal on Fuzzy Sets and Systems, Vol. 160,issue 13,pp.19129-1946,July 2009.

[25]K.T. Song. and L.H. Sheen. "Heuristic Fuzzy–neuro network and its application to reactive navigation of a mobile robot," International journal on Fuzzy Sets and systems Vol. 110, pp. 331-340, 2000.

[26]W. Li, M Chenya and F.M. Wahl. “A Neuro- Fuzzy system architecture for the behavior based control of a mobile in unknown environment,” International journal on Fuzzy Sets and systems, Vol. 87, pp.133-140, 1997.

[27] G. N. Marichal, L. Acosta, L. Moreno, J. A. M´endez, J. J. Rodrigo and M. Sigut. “Obstacle avoidance for a mobile robot: A neuro-fuzzy approach,” International journal on Fuzzy Sets and Systems, vol. 124, no. 2, pp. 171–179, Dec. 2001.

[28] Zhu and S.X. Yang. "Neuro fuzzy-Based Approach to Mobile Robot Navigation in Unknown Environments," IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol.37, no.4, pp.610-621, July 2007.

[29] M.M.Joshi and M.A. Zaveri. “Neuro-Fuzzy Based Autonomous Mobile Robot Navigation”, IEEE 11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010 , Singapore, Dec 2010.

[30] M.M.Joshi and M.A. Zaveri. “Optimally learnt, neural network based autonomous mobile robot navigation system”, International Conference on Advances in Electrical & Electronics , AEE 2010 , 20-23 December 2010.

[31] M.M.Joshi and M.A. Zaveri. “Fuzzy Based Autonomous Mobile Robot Navigation”, IEEE India Conference INDICON2009, Dec. 2009.

Design Mathematical Tunable Gain PID-Like Sliding Mode Fuzzy Controller with Minimum Rule base

Farzin Piltan SSP.ROBOTIC@yahoo.com
Department of Electrical and Electronic
Engineering, Faculty of Engineering,Universiti
Putra Malaysia 43400 Serdang, Selangor, Malaysia

N. Sulaiman nasri@eng.upm.edu.my
Department of Electrical and Electronic
Engineering, Faculty of Engineering,Universiti
Putra Malaysia 43400 Serdang, Selangor, Malaysia

Atefeh Gavahian SSP.ROBOTIC@yahoo.com
Industrial Electrical and Electronic Engineering
SanatkadeheSabze Pasargad. CO (S.S.P. Co),
NO:16 ,PO.Code 71347-66773, Fourth floor Dena
Apr , Seven Tir Ave , Shiraz , Iran

Samira Soltani SSP.ROBOTIC@yahoo.com
Industrial Electrical and Electronic Engineering
SanatkadeheSabze Pasargad. CO (S.S.P. Co),
NO:16 ,PO.Code 71347-66773, Fourth floor Dena
Apr , Seven Tir Ave , Shiraz , Iran

Samaneh Roosta SSP.ROBOTIC@yahoo.com
Industrial Electrical and Electronic Engineering
SanatkadeheSabze Pasargad. CO (S.S.P. Co),
NO:16 ,PO.Code 71347-66773, Fourth floor Dena
Apr , Seven Tir Ave , Shiraz , Iran

Abstract

In this study, a mathematical tunable gain model free PID-like sliding mode fuzzy controller (GTSMFC) is designed to rich the best performance. Sliding mode fuzzy controller is studied because of its model free, stable and high performance. Today, most of systems (e.g., robot manipulators) are used in unknown and unstructured environment and caused to provide sophisticated systems, therefore strong mathematical tools (e.g., nonlinear sliding mode controller) are used in artificial intelligent control methodologies to design model free nonlinear robust controller with high performance (e.g., minimum error, good trajectory, disturbance rejection). Non linear classical theories have been applied successfully in many applications, but they also have some limitation. One of the best nonlinear robust controller which can be used in uncertainty nonlinear systems, are sliding mode controller but pure sliding mode controller has some disadvantages therefore this research focuses on applied sliding mode controller in fuzzy logic theory to solve the limitation in fuzzy logic controller and sliding mode controller. One of the most important challenging in pure sliding mode controller and sliding mode fuzzy controller is sliding surface slope. This paper focuses on adjusting the gain updating factor and sliding surface slope in PID like sliding mode fuzzy controller to have the best performance and reduce the limitation.

Keywords: Sliding Mode Fuzzy Controller, Tunable Gain, Artificial Intelligence, Robust Controller, Sliding Mode Controller, Fuzzy Logic Theory, Sliding Surface Slope

1. INTRODUCTION

The aim of science and modern technology has making an easier life. Conversely, modern life includes complicated technical systems which these systems are nonlinear, time variant, and uncertain in measurement, they need to have controlled [2]. Controller (control system) is devices that can sense data from plant to improve the plants behavior through actuation and computation. Fuzzy logic theory was used in wide range applications that fuzzy logic controller (FLC) is one of the most important applications in fuzzy logic theory. Conversely pure FLC works in many areas, it cannot guarantee the basic requirement of stability and acceptable performance [4-5].

Sliding mode controller (SMC) is one of the influential nonlinear controllers in certain and uncertain systems which are used to present a methodical solution for two main important controllers' challenges, which named: stability and robustness. Conversely, this controller is used in different applications; sliding mode controller has subsequent drawbacks i.e. chattering phenomenon, and nonlinear equivalent dynamic formulation in uncertain systems[6-12].

Although both SMC and FLC have been applied successfully in many applications but they also have some limitations. The boundary layer method is used to reduce or eliminate the chattering [1, 3, 12] and proposed method focuses on applied sliding mode controller to proposed PID error-base fuzzy logic system with minimum rule base and adjust the sliding surface slope to implement easily and avoid mathematical model base controller.

This paper is organized as follows:

In section 2, Detail of classical sliding mode controller is presented. The main subject of fuzzy logic methodology is presented in section 3. In section 4, the proposed method is presented. Modelling PUMA-560 robot manipulator formulation is presented in section 5. In section 6, the simulation result is presented and finally in section 7, the conclusion is presented.

3. CLASSICAL SLIDING MODE CONTROL

Sliding mode controller (SMC) is a powerful nonlinear controller which has been analyzed by many researchers especially in recent years. The sliding mode control law divided into two main parts [1, 3];

$$\hat{\tau} = \hat{\tau}_{eq} + \hat{\tau}_{dis} \tag{1}$$

Where, the model-based component $\hat{\tau}_{eq}$ is compensated the nominal dynamics of systems and τ_{dis} is discontinuous part of sliding mode controller and it is computed as [16-18]

$$\hat{\tau}_{dis} = K.sgn(S) \tag{2}$$

A time-varying sliding surface S is given by the following equation [18]:

$$s(x,t) = (\frac{d}{dt} + \lambda)\ \mathrm{e} = 0 \tag{3}$$

Where λ is the constant and it is positive. To further penalize tracking error integral part can be used in sliding surface part as follows:

$$s(x,t) = (\frac{d}{dt} + \lambda)\left(\int_0^t \mathrm{e}\, dt\right) = 0 \tag{4}$$

The main target in this methodology is keep $s(x,t)$ near to the zero when tracking is outside of $s(x,t)$. The function of $sgn(S)$ defined as;

$$\boldsymbol{sgn(s)} = \begin{cases} \mathbf{1} & \boldsymbol{s} > 0 \\ \mathbf{-1} & \boldsymbol{s} < 0 \\ \mathbf{0} & \boldsymbol{s} = \mathbf{0} \end{cases} \tag{5}$$

The $\boldsymbol{K}$ is the positive constant. One of the most important challenges in sliding mode controller based on discontinuous part is chattering phenomenon which can caused to oscillation in output. To reduce or eliminate the chattering it is used the boundary layer method; in boundary layer method the basic idea is replace the discontinuous method by saturation (linear) method with small neighborhood of the switching surface. This replace is caused to increase the error performance.

$$\boldsymbol{B(t)} = \{\boldsymbol{x}, |\boldsymbol{S(t)}| \le \emptyset\}; \emptyset > 0 \tag{6}$$

Where $\emptyset$ is the boundary layer thickness. Therefore, to have a smote control law, the saturation function $\boldsymbol{Sat}(^{\boldsymbol{S}}/_{\emptyset})$ added to the control law: Suppose that $\boldsymbol{\tau_{sat}}$ is computed as [16-18]

$$\boldsymbol{\hat{\tau}_{sat}} = \boldsymbol{K.sat}\left(^{\boldsymbol{S}}/_{\emptyset}\right) \tag{7}$$

Where $\boldsymbol{Sat}\left(^{\boldsymbol{S}}/_{\emptyset}\right)$ can be defined as

$$\boldsymbol{sat}\left(^{\boldsymbol{S}}/_{\emptyset}\right) = \begin{cases} \mathbf{1} & (^{\boldsymbol{s}}/_{\emptyset} > 1) \\ \mathbf{-1} & \left(^{\boldsymbol{s}}/_{\emptyset} < 1\right) \\ ^{\boldsymbol{s}}/_{\emptyset} & (\mathbf{-1} < {^{\boldsymbol{s}}/_{\emptyset}} < 1) \end{cases} \tag{8}$$

Moreover by replace the formulation (7) in (1) the control output is written as;

$$\boldsymbol{\hat{\tau}} = \boldsymbol{\hat{\tau}_{eq}} + \boldsymbol{K.sat}\left(^{\boldsymbol{S}}/_{\emptyset}\right) = \begin{cases} \boldsymbol{\tau_{eq}} + \boldsymbol{K.sgn(S)} & , |\boldsymbol{S}| \ge \emptyset \\ \boldsymbol{\tau_{eq}} + \boldsymbol{K.}{^{\boldsymbol{S}}/_{\emptyset}} & , |\boldsymbol{S}| < \emptyset \end{cases} \tag{9}$$

4. PID FUZZY LOGIC CONTROLLER

A PID fuzzy controller is a controller which takes error, integral of error and derivative of error as inputs. Fuzzy controller with three inputs is difficult to implementation, because it needs large number of rules, in this state the number of rules increases with an increase the number of inputs or fuzzy membership functions [4-5, 24-31]. In the PID FLC, if each input has 7 linguistic variables, then $7 \times 7 \times 7 = 343$ rules will be needed. The proposed PID FLC is constructed as a parallel structure of a P+D sliding surface slope and P+I+D sliding surface slope, and the output of the PID FLC is formed by adding the output of two fuzzy control blocks. This work will reduce the number of rules needed to $7 \times 7 = 49$ rules only.

This controller has two inputs (S_1, S_2) and one output (τ_{fuzzy}). The inputs are first sliding surface (S_1) which measures by the equation (3), the second sliding surface (S_2) which measures by the equation (4). For simplicity in implementation and also to have an acceptable performance the triangular membership function is used. The linguistic variables for first sliding surface (S_1) are; Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB), and it is quantized in to thirteen levels represented by: -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6 the linguistic variables for second sliding surface (S_2) are; Fast Left (FL), Medium Left (ML), Slow Left (SL),Zero (Z), Slow Right (SR), Medium Right (MR), Fast Right (FR), and it is quantized in to thirteen levels represented by: -6, -5, -0.4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6 and the linguistic variables to find the output are; Large Left (LL), Medium Left (ML), Small Left (SL), Zero (Z), Small Right (SR), Medium Right (MR), Large Right (LR) and it is quantized in to thirteen levels represented by: -85, -70.8, -56.7, -42.5, -28.3, -

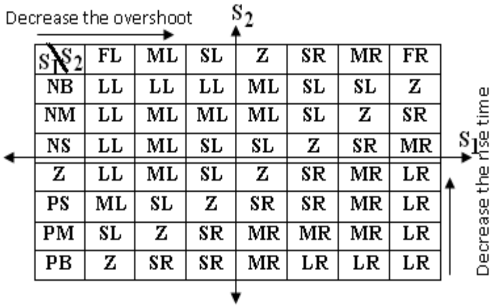
14.2, 0, 14.2, 28.3, 42.5, 56.7, 70.8, 85. Design the rule base of fuzzy inference system can play important role to design the best performance of sliding mode fuzzy controller, this paper focuses on heuristic method which, it is based on the behavior of the control systems.

The complete rule base for this controller is shown in Table 1. Rule evaluation focuses on operation in the antecedent of the fuzzy rules in sliding mode fuzzy controller. Max-Min aggregation is used in this work which the calculation is defined as follows;

$$\boldsymbol{\mu_U(x_k,y_k,U)=\mu_{\cup_{i=1}^{r}FR^i}(x_k,y_k,U)=max\left\{min_{i=1}^{r}\left[\mu_{R_{pq}}(x_k,y_k),\mu_{p_m}(U)\right]\right\}} \quad \textbf{(10)}$$

The last step to design fuzzy inference in sliding mode fuzzy controller is defuzzification. In this design the Center of gravity method (COG) is used and calculated by the following equation;

$$\boldsymbol{COG(x_k,y_k)=\frac{\sum_i U_i\sum_{j=1}^{r}\cdot\mu_u(x_k,y_k,U_i)}{\sum_i\sum_{j=1}^{r}\cdot\mu_u(x_k,y_k,U_i)}} \quad \textbf{(11)}$$


Decrease the overshoot
S_2
S_1
Decrease the rise time

$S_1 \backslash S_2$	FL	ML	SL	Z	SR	MR	FR
NB	LL	LL	LL	ML	SL	SL	Z
NM	LL	ML	ML	ML	SL	Z	SR
NS	LL	ML	SL	SL	Z	SR	MR
Z	LL	ML	SL	Z	SR	MR	LR
PS	ML	SL	Z	SR	SR	MR	LR
PM	SL	Z	SR	MR	MR	MR	LR
PB	Z	SR	SR	MR	LR	LR	LR

TABLE 1: Modified Fuzzy rule base table

This table used to describe the dynamics behavior of sliding mode fuzzy controller. Table 2 is shown the COG deffuzzification lookup table in fuzzy logic controller. These output values were obtained by mathematical on line tunable gain adjustment to reach the best performance in sliding mode fuzzy controller.

$\overrightarrow{S_2}$ $S_1\downarrow$	**Membership Function**												
	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	-85	-84.8	-84.8	-84	-82.1	-81	-79	-71	-68	-65	-62	-60	-54
-5	-84.8	-84	-82	-80	-78	-77	-74	-70	-64	-60	-56	-54	-47
-4	-78	-73	-70	-68	-64	-61	-60	-57	-55	-50	-47	-40	-38
-3	-70	-60	-58	-51	-42	-38	-34	-33	-31	-29	-28.4	-28.1	-28
-2	-50	-48	-45	-40	-38	-34	-32	-30	-28	-26	-25	-21	-20
-1	-30	-25	-21	-18	-16	-14	-10	-9	-8	-7	-6.8	-6	-5
0	-10	-8	-6	-1	2	3	6	7	8	10	12	15	17
1	15	18	21	22	23	25	27	28	29	30	30.5	30.8	31
2	29	29.8	31	33	34	34.6	35	35.2	36	37	38	39	42
3	40	41	42	43	45	45	46	46.3	46.8	47	48	51	52
4	48	49	50	52	53	55	56	57	58	59	60	61	63
5	60	61	62	63	64	66	67	68	68.5	69	70	70.8	71
6	66	68.7	68.9	70	72	74	75	77	78	79	81	83	84

TABLE 2 : COG lookup table in fuzzy sliding mode controller

5. THE PROPOSED METHOD

Sliding mode controller has two main parts: equivalent controller, based on dynamics formulation and sliding surface saturation part based on saturation continuous function to reduce or eliminate the chattering [19-23]. Reduce or eliminate the chattering regarding to reduce the error is play important role in this research. Applied sliding mode controller in fuzzy logic method have been proposed by several researchers [19-23] but in proposed method the new PID method with 49 rules is implemented and adjust by on line mathematical method. SMFC is fuzzy controller based on sliding mode method to easy implementation, stability, and robustness. A block diagram for sliding mode fuzzy controller is shown in Figure 1.

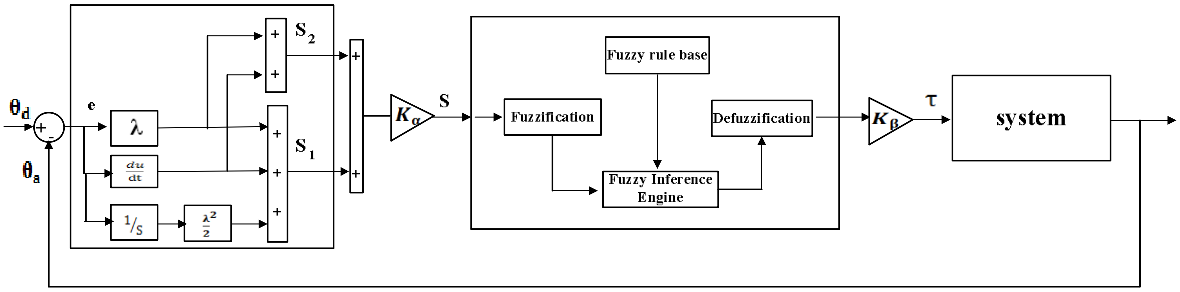


FIGURE 1: Sliding Mode Fuzzy Control (SMFC).

The system performance in this research is sensitive to the sliding surface slope λ input and output gain updating factor K_α & K_β for sliding mode fuzzy controller. Sliding surface slope can change the response of the output if large value of λ is chosen the response is very fast but the system is very unstable and conversely, if small value of λ considered the response of system is very slow but the system is very stable. Determine the optimum value of λ for a system is one of the most important challenging works in SMFC. For nonlinear, uncertain, and time-variant plants on-line tuning method can be used to self adjusting all coefficients. To keep the structure of the controller as simple as possible and to avoid heavy computation, a new supervisor tuner based on updated by a new coefficient factor kn is presented. In this method the supervisor part tunes the output scaling factors using gain online updating factors. The inputs of the supervisor term are error and change of error $(e, \dot{e})$ and the output of this controller is U, which it can be used to tune sliding surface slope, λ.

$$\boldsymbol{k_n = e^2 - \frac{(r_v - r_{vmin})^5}{1+|e|} + r_{vmin}} \tag{12}$$

$$r_v = \frac{\big(de(k) - de(k-1)\big)}{de(.)} \tag{13}$$

$$de(.) = \begin{Bmatrix} de(k); & if\ de(k) \geq de(k-1) \\ de(k-1) & if\ de(k) < de(k-1) \end{Bmatrix}$$

In this way, the performance of the system is improved with respect to the SMFC controller. So the new coefficient is calculated by;

$$\boldsymbol{\lambda_{new} = \lambda_{old} \times K_n} \tag{14}$$

$$\boldsymbol{K_{\alpha_{new}} = K_{\alpha_{old}}} \times K_n \tag{15}$$

5. APPLICATION

This method is applied to 3 revolute degrees of freedom (DOF) robot manipulator (e.g., first 3 DOF PUMA robot manipulator). The equation of an *n-DOF* robot manipulator governed by the following equation [1, 3]:

$$\boldsymbol{M(q)\ddot{q} + N(q,\dot{q}) = \tau} \qquad \textbf{(17)}$$

Where $\boldsymbol{\tau}$ is actuation torque, $\boldsymbol{M(q)}$ 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:

$$\boldsymbol{\tau = M(q)\ddot{q} + B(q)[\dot{q}\,\dot{q}] + C(q)[\dot{q}]^2 + G(q)} \qquad \textbf{(18)}$$

Where the matrix of coriolios torque is $\boldsymbol{B(q)}$, $\boldsymbol{C(q)}$ is the matrix of centrifugal torques, and $\boldsymbol{G(q)}$ is the vector of gravity force. The dynamic terms in equation (15) are only manipulator position. This is a decoupled system with simple second order linear differential dynamics. In other words, the component $\boldsymbol{\ddot{q}}$ 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[3]:

$$\boldsymbol{\ddot{q} = M^{-1}(q).\{\tau - N(q,\dot{q})\}} \qquad \textbf{(19)}$$

This technique is very attractive from a control point of view. This paper is focused on the design mathematical tunable gain model free PID-like sliding mode fuzzy controller for PUMA-560 robot manipulator based on [13-15].

6. RESULTS

PD sliding mode fuzzy controller (PD-SMFC) and mathematical tuneable gain model free PID-like sliding mode fuzzy controller (GTSMFC) were tested to compare response trajectory. In this simulation the first, second, and third joints are moved from home to final position without and with external disturbance. Trajectory performance, chattering phenomenon and disturbance rejection are compared in these two controllers. These systems are tested by band limited white noise with a predefined 40% of relative to the input signal amplitude which the sample time is equal to 0.1. This type of noise is used to external disturbance in continuous and hybrid systems.

Tracking performances: Figure 2 shows the tracking performance in GTSMFC and SMFC without disturbance for Step trajectory.

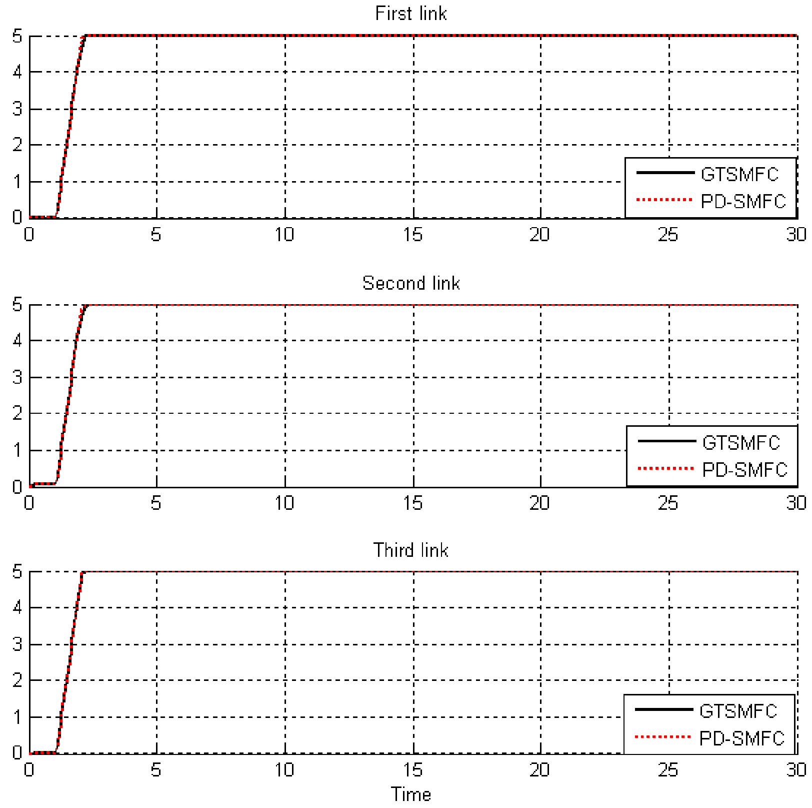


FIGURE 2 : Step GTSMFC and SMFC for First, second and third link trajectory without any disturbance.

By comparing, Figure 2, in GTSMFC and SMFC, both of them have the same overshoot **(1%)** the GTSMFC and SMFC's rise time are 0.483 Sec.

Disturbance Rejection

Figure 3 is indicated the power disturbance rejection in GTSMFC and SMFC. A band limited white noise with predefined of 40% the power of input signal is applied to these controllers; it found slight oscillations in SMFC's trajectory responses.

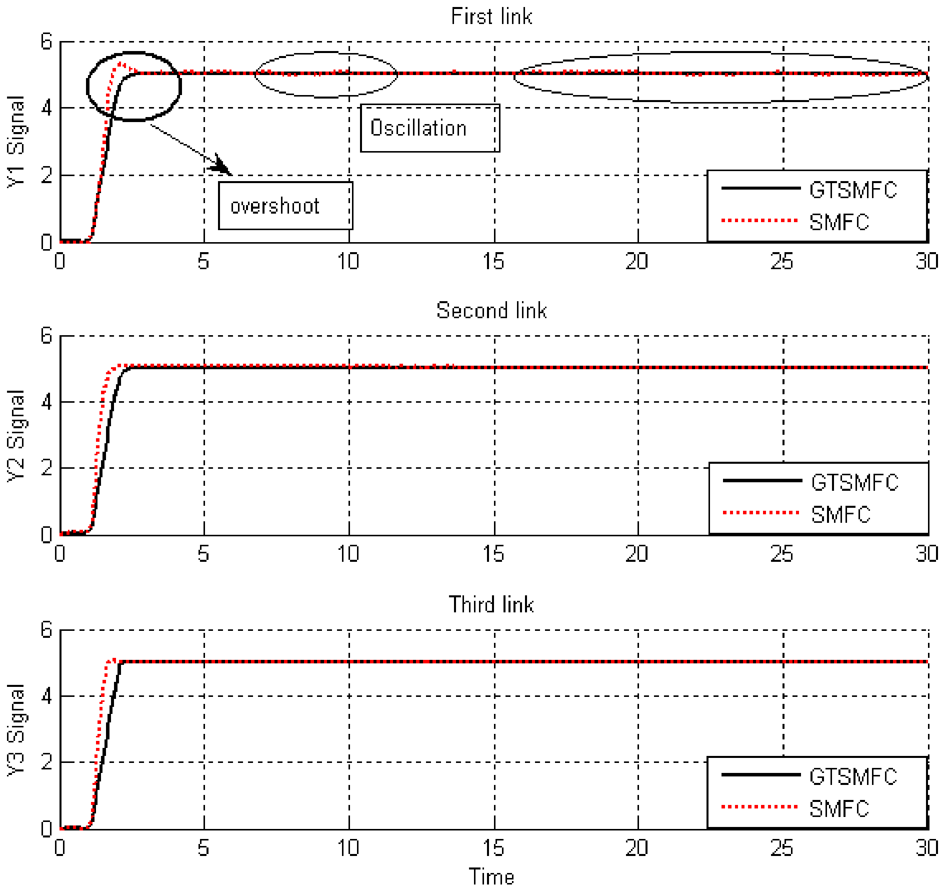


FIGURE 3 : Step GTSMFC and SMFC for First, second and third link trajectory with external disturbance**.**

Among above graph, relating to trajectory with external disturbance, SMFC has slightly fluctuations. By comparing overshoot; GTSMFC's overshoot **(1%)** is lower than SMFC's **(2.2%).**

Chattering Phenomenon

As mentioned in previous, chattering play important roles in sliding mode controller which one of the major objectives in this research is reduce or remove the chattering in system's output with uncertainty and external disturbance. Figure 4 has shown the power of boundary layer (saturation) method and online mathematical gain tuning methodto reduce the chattering in GTSMFC and also SMFC.

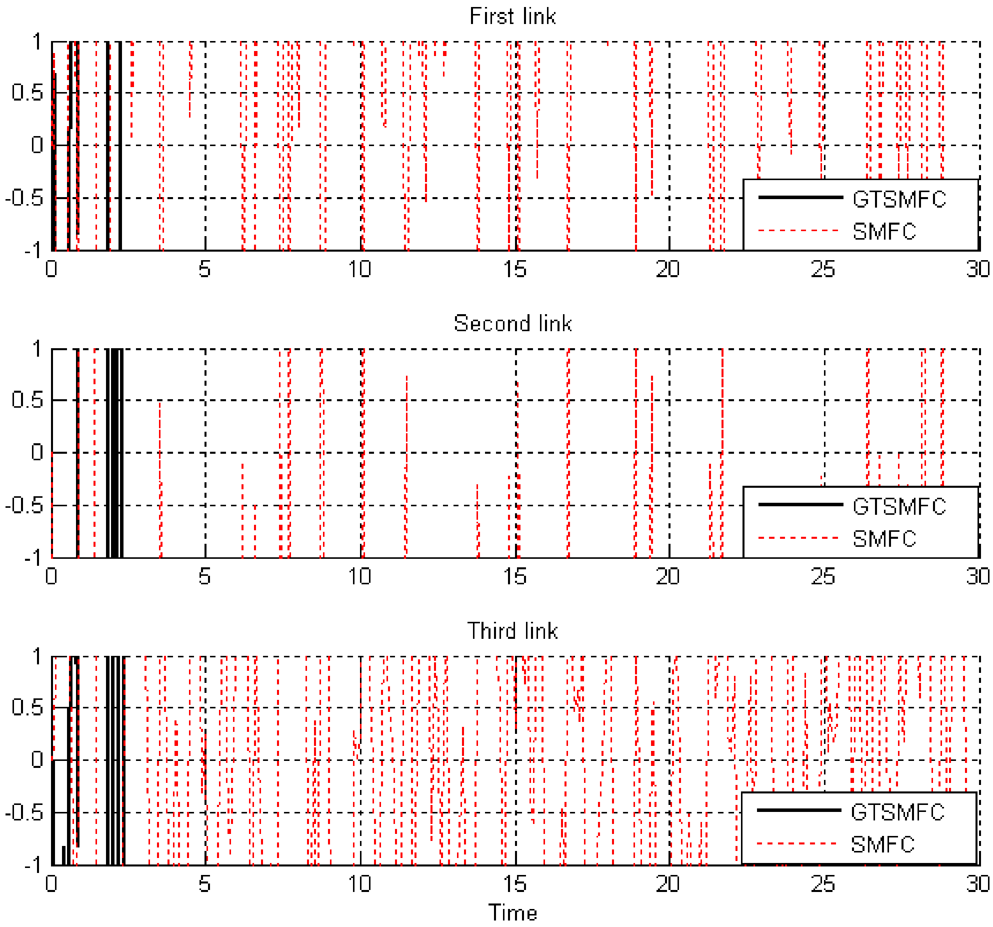


FIGURE 4: GTSMFC Vs. SMFC chattering with disturbance

7. CONCLUSION

Refer to the research, adjusting the gain updating factor and sliding surface slope in PID like sliding mode fuzzy controller design and application to robot manipulator has proposed in order to design high performance nonlinear controller in the presence of uncertainties. Regarding to the positive points in fuzzy logic controllers in uncertain systems, sliding mode controller which it has stability and robustness and on line tunable gain to tune the coefficient in structure and unstructured uncertain system the output responses have improved. Sliding mode controller by adding to the proposed PID fuzzy logic method with minimum rule base has covered negative points in pure fuzzy logic method and sliding mode methodology. Obviously the methodology of online tuning is the main goal in this research which most of researcher used fuzzy logic or neural network to adjust the parameters but in this method we used mathematical methodology that it is model free.

REFERENCES

[1] T. R. Kurfess, *Robotics and automation handbook*: CRC, 2005.

[2] K . Ogata, *Modern control engineering*: Prentice Hall, 2009.

[3] B. Siciliano and O. Khatib, *Springer handbook of robotics*: Springer-Verlag New York Inc, 2008.

[4] L. Reznik, *Fuzzy controllers*: Butterworth-Heinemann, 1997.

[5] S. Mohan and S. Bhanot, "Comparative study of some adaptive fuzzy algorithms for manipulator control," *International Journal of Computational Intelligence,* vol. 3, pp. 303–311, 2006.

[6] O. Kaynak, "Guest editorial special section on computationally intelligent methodologies and sliding-mode control," *IEEE Transactions on Industrial Electronics,* vol. 48, pp. 2-3, 2001.

[7] Shahnazi R., H. Shanechi, N. Pariz. "Position control of induction and servomotors: A novel adaptive fuzzy PI sliding mode control". IEEE Conferences on power engineering, 2006, P.P. 1-9.

[8] Medhaffar H., N. Derbel, and T. Damak. "A decoupled fuzzy indirect adaptive sliding mode controller with application to robot manipulator". Int. Journal on modeling, identification and control, 1(1), 23-29, 2006.

[9] Shahnazi R., H. Shanechi, N. Pariz. "Position control of induction and servomotors: A novel adaptive fuzzy PI sliding mode control". IEEE Journals on energy conversions, 23(1), 138-147, 2008.

[10] Weng C. C., W. S. Yu. "Adaptive fuzzy sliding mode control for linear time-varying uncertain systems". IEEE conference on fuzzy systems, 2008, P.P: 1483-1490.

[11] Yu Z. X. "Adaptive sliding mode-like fuzzy logic control for nonlinear systems". Journal of communication and computer, 6(1), 53-60, 2009.

[12] Piltan, F., et al. "Design sliding mode controller for robot manipulator with artificial tunable gain". Canaidian Journal of pure and applied science, 5 (2), 1573-1579, 2011.

[13] B. S. R. Armstrong, "Dynamics for robot control: friction modeling and ensuring excitation during parameter identification," 1988.

[14] B. Armstrong*, et al.*, "The explicit dynamic model and inertial parameters of the PUMA 560 arm," 2002, pp. 510-518.

[15] P. I. Corke and B. Armstrong-Helouvry, "A search for consensus among model parameters reported for the PUMA 560 robot," 2002, pp. 1608-1613.

[16] V. Utkin, "Variable structure systems with sliding modes," *Automatic Control, IEEE Transactions on,* vol. 22, pp. 212-222, 2002.

[17] R. A. DeCarlo*, et al.*, "Variable structure control of nonlinear multivariable systems: a tutorial," *Proceedings of the IEEE,* vol. 76, pp. 212-232, 2002.

[18] K. D. Young*, et al.*, "A control engineer's guide to sliding mode control," 2002, pp. 1-14.

[19] C. C. Weng and W. S. Yu, "Adaptive fuzzy sliding mode control for linear time-varying uncertain systems," 2008, pp. 1483-1490.

[20] M. Ertugrul and O. Kaynak, "Neuro sliding mode control of robotic manipulators," *Mechatronics,* vol. 10, pp. 239-263, 2000.

[21] P. Kachroo and M. Tomizuka, "Chattering reduction and error convergence in the sliding-mode control of a class of nonlinear systems," *Automatic Control, IEEE Transactions on,* vol. 41, pp. 1063-1068, 2002.

[22] Y. Li and Q. Xu, "Adaptive Sliding Mode Control With Perturbation Estimation and PID Sliding Surface for Motion Tracking of a Piezo-Driven Micromanipulator," *Control Systems Technology, IEEE Transactions on,* vol. 18, pp. 798-810, 2010.

[23] B. Wu*, et al.*, "An integral variable structure controller with fuzzy tuning design for electro-hydraulic driving Stewart platform," 2006, pp. 5-945.

[24] L. A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets and Systems,* vol. 90, pp. 111-127, 1997.

[25] J. Zhou and P. Coiffet, "Fuzzy control of robots," 2002, pp. 1357-1364.

[26] S. Banerjee and P. Y. Woo, "Fuzzy logic control of robot manipulator," 2002, pp. 87-88.

[27] K. Kumbla*, et al.*, "Soft computing for autonomous robotic systems," *Computers and Electrical Engineering,* vol. 26, pp. 5-32, 2000.

[28] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller. I," *IEEE Transactions on systems, man and cybernetics,* vol. 20, pp. 404-418, 1990.

[29] R. J. Wai*, et al.*, "Implementation of artificial intelligent control in single-link flexible robot arm," 2003, pp. 1270-1275.

[30] R. J. Wai and M. C. Lee, "Intelligent optimal control of single-link flexible robot arm," *Industrial Electronics, IEEE Transactions on,* vol. 51, pp. 201-220, 2004.

[31] M. B. Menhaj and M. Rouhani, "A novel neuro-based model reference adaptive control for a two link robot arm," 2002, pp. 47-52.

Active Control of Tool Position in the Presence of Nonlinear Cutting Forces in Orthogonal Cutting

A.H. El-Sinawi aelsinawi@aus.edu
Mechanical Engineering department
American University of Sharjah,
Sharjah, UAE PO Box 26666

Abstract

This work presents a practical approach to the control of tool's position, in orthogonal cutting, in the presence nonlinear dynamic cutting forces. The controller is Linear Quadratic Gaussian (LQG) type constructed from an augmented model of both, tool-actuator dynamics, and a nonlinear dynamic model relating tool displacement to cutting forces. The latter model is obtained using black-box system identification of experimental orthogonal cutting data in which tool displacement is the input and cutting force is the output. The controller is evaluated and its performance is demonstrated.

Key words: LQG, Nonlinear ARX Model, Orthogonal Cutting, Active Control, Nonlinear Cutting Force

1. INTRODUCTION

Performance of machining processes is assessed by dimensional and geometrical accuracy as well as the surface texture of the part. Factors such as cutting force, workpiece vibration, machine-tool vibration, process instability, tool wear and thermal deformation deteriorate this performance. Recent interest in high speed-machining and high manufacturing efficiency requires faster, higher bandwidth actuation, and controllers that are more robust. Besides determining part functional behavior, the surface texture also plays a key role in the area of manufacturing process control [1]. The basis for using surface texture as means of process control is derived from the fact that a slight change in the manufacturing process manifests itself as a corresponding change in the resulting surface geometry. The roughness, waviness, lay, and flaws constitute the texture of the workpiece surface [2]. Surface texture of the work-piece is highly dependent on tool position during cutting which in turn is affected by cutting dynamics. Therefore extensive research work has been done on modeling and analysis of tool work-piece interaction and the dynamics of the cutting process. Altintas [3] have presented a cutting force model as a function of regenerative chip thickness, cutting speed, and the velocity and acceleration terms of the vibrations. In their work, amplitude and frequency of inner and outer vibration waves are generated on the chip by an instrumented fast tool servo powered by a piezo actuator. They established a model in which the coefficients of dynamic cutting forces are identified and used in analyzing the effect of cutting speed, tool wear, vibration frequency and wavelength on the chatter stability of a turning process. Ikua [4] have studied the cutting force generated in ball end milling and presented a model for the radial and tangential forces as a function of depth of cut, and other factors related to the geometry of the tool and workpeices.

Researchers have tried to establish quantitative relations between the surface finish and the cutting parameters of the depth of cut, feed, and cutting speed using methods such as, among others, wavelength decomposition surface roughness, wavelet analysis, and tool vibration [4-9]. The cutter–workpiece interaction forces are assumed to cause relative displacement between the cutter and the work piece, which influence the surface generation mechanism. Analytical, experimental and mechanistic methods were used to predict the interaction forces. The analytical methods focused on establishing a relationship between the cutting force and the instantaneous uncut chip cross – section [10, 11] and the non – linear mechanisms [12, 13]. The analytical models were not capable of predicting the dynamic forces accurately due to the secondary non –

linear effects that stems from the tool/workpiece interaction. The experimental methods include static, dynamic and time–series methods. The static methods are assumed based on linear assumptions and linear cutting conditions [14]. The dynamic methods replaced the static methods and the cutting process is assumed to be a combination of two independent actions namely, the wave cutting action and the wave removing action [15]. The time-series method was formulated to identify the dynamic cutting force coefficients as well as the transfer functions of the three dimensional dynamic cutting process directly from operating data [16]. The mechanistic modeling methods view the machining process as a combination of the chip load/cutting force relationship, cutting – tool geometry, machining conditions and tool /work – piece displacement due to cutting forces. The workpiece dynamic behavior is constructed using either the distributed – parameter [17], or the lumped – parameter approaches [18]. Few researchers investigated the dynamic effect of the tool holder. Shawky and Elbestawi [19] concluded that the effect of tangential force is insignificant and no model of the tool holder is needed in that direction. Their experimental work showed that modeling the tool by a second order system in each axis provides significant approximation of the dynamic behavior of a single point tool during cutting [19].

Control of machining process includes three levels of control that one might encounter in a controller for machining process, namely they are the servo control, the process control [20] and the supervisory control [21]. In the servo control process, the motion of the cutting tool is taken relative to that of the workpiece. The process control level is used to control the cutting forces and tool wear to maintain high production rates and good part quality. The highest level of control is the supervisory control and it directly measures product related variables, such as part dimensions and surface finish. Different approaches are used to correct for machining errors by means of dynamic error compensation. More recently, Tian et al. [22] have presented a methodology for modeling and control of a high precision flexure-based mechanism for ultra-precision turning operation based on the position control of an auxiliary precision mechanism utilized on the turret of the conventional lathe to implement nanometer level infeed. El-Sinawi [23] has presented an optimal control method for controlling the tool position in orthogonal cutting in both feed are radial directions. Moradi et al. , [24] have presented a robust control method for Orthogonal turning process in the process was modeled as a single degree of freedom model that includes the effect of tool flank wear with a control input of the system being force applied to the tool provided by a piezo-actuator. Huang et al. [25] have presented a tool wear detection based on cutting force monitoring.

In this work, a control scheme for the purpose of improving surface texture of turned surfaces through control of tool position in the presence of nonlinear cutting force is developed.

Improvement of surface quality is achieved via active positioning of the tool or cutter through the attenuation of cutting forces effect on the tool in both radial and tangential directions. The process is assumed to be stochastic due to both process and measurement noise. Experimental force-displacement data is used to construct a nonlinear dynamic model of dynamic cutting forces. Various nonlinear models are constructed using system identification techniques including ARX, nonlinear ARX, and Hammerstein-Wiener techniques [26]. The best nonlinear model obtained that closely fits experimental data, is then linearized and constructed in state-space form to later utilize in the Linear Quadratic Gaussian Controller. Simulation results will be used to verify the effectiveness of the proposed modeling and control approach, and to enhance current understanding of the mechanisms responsible for generating both stochastic and deterministic components of the surface texture. Simulation work of the proposed modeling and control techniques will be based on actual parameters for actuators and the cutting process data, which is readily available in the literature [27].

2. MODEL DEVELOPMENT

Force-displacement data obtained from experimental cutting of a 6061 Aluminum workpiece using a carbide tool with 0° rake angle and 7° clearance angle, depth of cut of 0.4 mm, feed of 0.050 mm/rev, and spindle speed: 2200 rev/min, are shown in Figures 1 and 2, have been used to construct a force displacement model using black-box system identification.

Four dynamic models with tool displacement as input and cutting force as output are constructed are indentified and compared as shown in Figure 3. In reference to Figure 3, the first model identified as LinMod1, is a linear ARX model of the form

$$f_c(t)+a_1 f_c(t-1)+a_2 f_c(t-2)+......+a_\tau f_c(t-\tau)=b_1 x_c(t)+b_2 x_c(t-1)+.....+b_\mu x_c(t-\mu-1)+e(t) \quad (1)$$

Where f_c, x_c are the cutting force and tool displacement in the corresponding direction, respectively. $f_c(t-1)+f_c(t-2),..,f_c(t-\tau),\ x_c(t)+x_c(t-1),...,\ x_c(t-\mu-1)$ are delayed input and output variables called regressors. Linear ARX model predicts the output f_c as a weighted sum of its regressors. τ is the number of past output terms while μ is the number of past input terms used to predict the current output f_c and $e(t)$ is a white noise sequence.

The second and third models identified as LinMod3 and LinMod4 are constructed as follows;

LinMod3 is constructed using Predictive Error Minimization or PEM with various constraints on the order or number of states of the state-space model. This method yields a discrete time-domain state-space dynamic model of the form,

$$x(t+Ts)=Ax(t)+Bu(t)+Ke(t), \quad y(t)=Cx(t)+Du(t)+e(t) \quad (2)$$

In which x are the states, y is output, A, B, C, K, D represent dynamics, input output, disturbance and feed through matrices, respectively [28].

Narx1 model is constructed using Nonlinear ARX model which is an extension of ARX model given in Equation (1) except that the input –output mapping is nonlinear and of the form;

$$f_n(t)=F(f_c(t-1),f_c(t-2),....,x_c(t),x_c(t-1), ...) \quad (3)$$

Where f_n is the output and F is a nonlinear mapping function, such as Wavelet network, Tree partition, etc…, [29].

Nhw1 model shown in Figure 3, is constructed using Hammerstein-Wiener model; see [Yucai Zhu, 2002 and Lennart Ljung, 1999] for further details. It is clear from Figure 3 that the Narx1 has the best fit to experimental data with 99.9% fit.

2.1 Linearization of the Nonlinear ARX Model

The nonlinear model that best fits input-output data is clearly the nonlinear ARX model mapped with a single-layer sigmoid function. The nonlinear mapping of input-output data is shown in Figure 4, with

$$F(v)=L(v-r)+d+g(Q(v-r)) \quad (4)$$

v is a vector of the regressors. $L(v-r)+d$ is the output of the linear block shown in Figure 4, *d* is a scalar offset, $g(Q(v-r))$ is the output of the nonlinear function block, and Q is a projection matrix and r is the mean of regressors (v). Function $g(v)$ expressed as $g(v)=\sum_{k=1}^{n}\alpha_k\phi(\beta_k(v-\xi_k))$ with $\phi(s)=\left(e^s+1\right)^{-1}$ is the nonlinearity estimator of the sigmoid function type. β_k is a row vector such that $\beta_k(v-\xi_k)$ is a scalar, s is the Laplace variable and n is the number of sigmoid network units. The model obtained by this approach is of the form given in Equation 3 with $\tau=4$, and $\mu=4$.

Step response of the models obtained for both forces in the radial (f_x) and tangential (f_y) directions is shown in Figures 5 and 6, respectively. The step response shows the steady state values of both functions which are later on used in linearzing the model. See Appendix A for linearization details.

Linearized state-space model of the radial force is;

$$\begin{aligned}\dot{X}_r(t) &= A_r X_r(t) + B_r x_{cr} \\ f_x &= C_r X_r(t) + D_r x_{cr}\end{aligned} \tag{5}$$

Where $\dot{X}_r(t)$ are the states of the radial force model, A_r, B_r, C_r, D_r are dynamic, input, output and feed through matrices, respectively. f_x, x_{cr} are radial cutting force and radial tool displacement, respectively. The same procedure is carried out for the tangential cutting force model which yields;

$$\begin{aligned}\dot{X}_t(t) &= A_t X_t(t) + B_t x_{ct} \\ f_y &= C_t X_t(t) + D_t x_{ct}\end{aligned} \tag{6}$$

2.2 Modeling of the Machining Process

Figure7 shows the difference between tool post (passive) and proposed active tool fixture that will allow for implementation of the proposed controller. Figure (7.a) shows the conventional tool post rigidly attached to the machine tool structure, while Figure (7.b) shows the active tool holder platform equipped with two actuators placed between the tool and the machine tool structure. This will allow the control force to provide necessary manipulation of the tool to maintain a constant depth of cut.

During cutting, the tool is perturbed form nominal depth of cut by two inputs namely; the dynamic cutting forces in both radial and tangential directions, in addition to process noise $w(t)$.

State-space model of the tool-actuator assembly in the radial direction can be represented as;

$$\begin{aligned}\dot{x}_r &= A_{cr} x_r + B_{cr} F_{ar} \\ Y_{cr} &= C_{cr} x_r + D_{cr} F_{ar}\end{aligned} \tag{7}$$

Such that,

$$A_{cr} = \begin{bmatrix} 0 & 1 \\ -\frac{k_{ac}}{m} & -\frac{b_{ac}}{m} \end{bmatrix},\ B_{cr} = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix},\ C_{cr} = \begin{bmatrix} 1 & 0 \end{bmatrix},\ D_{cr} = 0,$$

where x_r is vector of the tool states, namely, displacement and velocity. m is the mass of the tool and its mount, k_{ac}, b_{ac} are the elastic stiffness and damping coefficients of the actuator, respectively. k_{ac}, b_{ac} can be obtained by an FR test of the tool-actuator assembly. Notice that, the same procedure can be carried out for the tangential direction since both actuators are orthogonal.

Refer to [23, 27] for more information on the model development.

3. CONTROL STRATEGY AND THE CONTROLLER DESIGN

The control strategy will be centered on reducing the amplitude of the tool' dynamic displacement to zero for the purpose of maintaining a constant depth of cut and subsequently, a smooth surface texture of the workpiece. This requires minimization of the error between the desired and the actual position of the tool. The desired or nominal tool position is the one yielding a constant depth of cut. However, the existence of the dynamic cutting forces perturbs the tool form its nominal position and thus, varies the depth of cut causing subsequent deterioration in surface texture. Attenuating the effect of the cutting force on the tool's position, the error in the tool position will be minimized. This can be achieved by reacting on the tool with an equal but opposite forces through proper actuation. This task is difficult due to (a) existence of process and measurement noise and (b) nonlinearity of the dynamic cutting forces. To overcome these difficulties, the controller must be able to estimate the force needed to minimize the error in the tool position from noisy process using measurement data contaminated with noise. In addition to that, the controller has to track and manipulate the position of the tool effectively in the presence of nonlinear dynamic cutting forces. Added to all of the above, the controller has to maintain high stability and performance under various disturbance characteristics. To do so, the controller has to be constructed based on a model that takes into account the dynamics of the tool-actuator system as well as the cutting force dynamics. Therefore, The plant used for this purpose is a combination (i.e., augmentation) of the dynamic cutting forces and tool-actuator dynamics in both radial and tangential directions. The first step in designing the proposed active controller, is the design of the LQG estimator and regulator gains L_e and K_R, respectively. The analysis presented here is in state-space such that the augmented system depicted by Equations (5), and (6) is;

$$A_a = \begin{bmatrix} A_{cr} & 0 \\ B_r\, C_{cr} & A_r \end{bmatrix} \tag{8}$$

$$B_a = \begin{bmatrix} B_{cr} \\ B_r\, D_{cr} \end{bmatrix} \tag{9}$$

$$C_a = \begin{bmatrix} D_r\, C_{cr} & C_r \end{bmatrix} \tag{10}$$

$$D_a = \begin{bmatrix} D_r\, D_{cr} \end{bmatrix} \tag{11}$$

where the matrices A_a, B_a, C_a, and D_a are respectively, the dynamic, input, output, and direct transmission matrices of the augmented system. The four matrices in the foregoing are used to design the LQG estimator and regulator gain matrices L_e and K_R, such that,

$$L_e = \begin{bmatrix} [L_{e\,cr}] \\ [L_{er}] \end{bmatrix}, K_R = \begin{bmatrix} [K_{R\,cr}] \\ [K_{R\,r}] \end{bmatrix} \tag{12}$$

This augmentation is necessary to incorporate the cutting force dynamics in the process of determining the optimal estimator and regulator gains. Equation 12 shows that L_e and K_R are both partitioned in two parts each where $[]_{cr}$ corrects x_{cr} (i.e., tool-actuator model states) and $[]_r$ corrects X_r (i.e., cutting force states). The LQG (virtual model of the system) should be subjected to all inputs that the actual plant is subjected to, including the control force. Figure 8 shows the control scheme implementation in the radial direction. Further details on construction of the controller are found in [30].

4. NUMERICAL EXAMPLE

Force-displacement results obtained from experimental cutting of a 6061 Aluminum workpiece using a carbide tool with 0° rake angle and 7° clearance angle, depth of cut of 0.4 mm, feed of 0.050 mm/rev, and spindle speed: 2200 rev/min, shown in Figures 1 and 2, are used to construct a force-displacement model using black-box system identification. Linearized State-space model obtained by Nonlinear ARX modeling, presented here in canonical form is;

$$\dot{X}_r(t)=\begin{bmatrix}1.145 & -0.7 & 0.5651 & -0.4435\\ 2 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0.5 & 0\end{bmatrix}X_r(t)+\begin{bmatrix}1024\\0\\0\\0\end{bmatrix}x_{cr}$$

$$f_x=1500\times[-266.2\ \ 153.9\ \ -213.8\ \ 460.7]X_r(t)+[0]x_{cr}$$

ETREMA actuator with peak-to-peak excursion of $50\times10^{-6}\,m$ is utilized as the active manipulator of the tool. The Actuator's elastic stiffness and damping used in this study were taken as follows (El-Sinawi et al. 2005): *m =0.53 kg* for the tool-actuator assembly. $k_{ac}=14.6\times10^{6}$ N/m, and $b_{ac}=10$ kg /s. The cutting process noise $w(t)$ is assumed to have a zero mean, and variation of ± 10% of the cutting force, while measurement noise $v(t)$ is also assumed to be random with zero mean, and variation of ± 10% of measurement values. Notice that the small values of $w(t)$ and $v(t)$ makes the linearization valid in the vicinity of nominal trajectory of the system.

5. RESULTS AND ANALYSIS

To implement the proposed control strategy, experimental force-displacement data shown in Figures 1 and 2 is utilized to construct a transfer function relating the tool's displacement as input to cutting force as output. The data is obtained from Altintas [3] and Matlab® System identification toolbox is used for the purpose of identifying the best transfer function that relates the input to the output. Figure 3 shows various algorithms implemented by the system identification process along with the percentage of match to the experimental input-output data. The figure shows that a transfer function obtained using Narx identification algorithm has a 99.99% match to experimental data. The model obtained from System identification data have the structure shown in Figure 4. Similar approach is used to obtain a transfer function for the input-output data in the feed direction. Step responses of radial and feed transfer function models obtained by System identification are shown in Figures 5 and 6. Latter figures show that both models are lightly damped with damping of approximately 0.5 % and 1 % for the radial and feed directions, respectively. Such light damping should not pose a problem since the damping can be increased using proper feedback control. With the tool's Displacement-Force model in both radial and feed directions are available, the tool is mounted on two orthogonal actuators intended to manipulate the tool's position in corresponding directions using proper control force as estimated by the controller. Transfer functions of the Identified models are augmented with the actuators models to obtain the suitable LQG controller gains Namely, L_e and K_R. During actual implementation of the controller, measurements of the tool position are compared to the estimates of the tool's position and the difference (i.e. error) is fed back through the estimator gain (L_e) to adjust the controller model and thus, improve estimates of the tool's position. The estimated position is then fed through the actuator's transfer function to produce the negative actuation force needed to maintain a constant depth of cut (i.e., zero tool displacement). The negative force provided by the actuator is only estimate of the external cutting force that perturbs the tool from its nominal position. Force estimates depend mainly on the values weighting matrices Q and R needed to minimize the performance index (J) of the LQG [31]. In general, choosing a larger value of Q compared to R implies the demand for high controller performance. On the Other hand, larger value of R implies minimal control energy. In this work, the process model obtained by the system identification is fairly accurate which allows for large Q. However, the limitation on the actuator force (approximately 500 N) and excursion (50 micrometer) poses a limit on the values of Q and R. In such application, the important factor that determines the performance vs. control energy is main the ratio of Q/R rather than the individual values of each.

In this application the ratio chosen for Q/R is 2. Assuming only 10% process and measurement noise, Figures 9 and 10 show that the proposed control technique has managed to reduce the tools dynamic displacement by approximately 40%. These results are reasonable only if the cutting process remains stable (i.e., no occurrence of chatter) and all factors that are not considered in the model remain unchanged; such factors include tool wear, temperature, micro-hardness variation of the workpiece and workpiece dynamics. Figures 11 and 12 show the control force exerted by the actuator on the tool is well within the capability of the actuator. It should be noted that the identified model will be affected by changes of depth of cut, cutting speed feed-rate and tool wear. Adaptive implementation of the proposed process is possible where the displacement-force transfer function identification is continuously updated as the cutting process progresses. However, the accuracy of the identified model can be a serious drawback of the control technique proposed by this work.

6. CONCLUSIONS

In this research work, active control of an orthogonal cutting process is presented. Nonlinear dynamic cutting force model is generated from actual machining data obtained in both radial and tangential directions. An LQG based controller is developed based on an augmented model of both tool-actuator dynamics and a linearized model of the dynamic cutting force. The control objective is to eliminate the tool’s dynamic displacement and maintain a constant depth of cut. Simulation results have shown that the proposed control strategy has managed to significantly reduce the dynamic displacement of the tool with minimal force and calculations efforts even in the presence of significant randomness in the process and measurement as well as the nonlinearity of the cutting force. Experimental study is undergoing to verify the integrity of the controller in real time applications.

ACKNOWLEDGEMENT

Author acknowledges the support of the American University of Sharjah

7. REFERENCES

[1]. M. Shiraishi and H. Sumiya. “Improvement of Geometrical Errors by Surface Roughness and Tool Position Control”. *The Annual Meeting of the ASME*, PED-vol. 45:9-22, 1990.

[2]. M. Sander. *A Practical Guide to the Assessment of Surface texture*. Mahr Perthen, Gottingen, Germany, 1991.

[3]. Y. Altintas, M. Eynian, H. Onozuka. “Identification of dynamic cutting force coefficients and chatter stability with process damping”. *CIRP Annals - Manufacturing Technology*, 2008, 57, 371–374.

[4]. Bernard W. Ikua, Hisataka Tanaka, Fumio Obata, Satoshi Sakamoto. “Prediction of cutting forces and machining error in ball end milling of curved surfaces -I theoretical analysis”. *Precision Engineering, 25(4): 266-273, 2001.*

[5]. S. M. Pandit and M. S. Shunmugan. “Signature of Machine Tool Errors on Surface Texture”. *Annals of the CIRP*, PED-Vol. 45:63-74, 1990.

[6]. W.S. Lin, B.Y. Lee, C.L. Wu, Modeling the surface roughness and cutting force for turning, *Journal of Materials Processing Technology*. 108, 286–293, 2001.

[7]. O.B. Abouelatta, J. Madl, Surface roughness prediction based on cutting parameters and tool vibrations in turning operations, J. Mater. Process. Technol. 118, 269–277, 2001.

[8]. S. Fu, B. Muralikrishnan, and J. Raja, Engineering Surface Analysis with Different Wavelet Bases. Transactions of the ASME, 125, 844-852, 2003.

[9]. K.A. Risbood, U.S. Dixit, A.D. Sahasrabudhe, Prediction of surface roughness and dimensional deviation by measuring cutting forces and vibrations in turning process. Journal of Materials Processing Technology 132, 203–214, 2003.

[10]. G.B. Boothroyd and W. A. Knight. *Fundamentals of Machining and Machine Tools*, Marcel Dekker, Inc., New York, 1989.

[11]. K.F. Ehmann, S. G. Kapoor, R. E. DeVor, and I. Lazoglu. “Machine Process Modeling Review”. *Journal of Manufacturing Science and Engineering*, 119:655-663, 1997

[12]. P. Albrecht. “Dynamics of the Metal Cutting Process”. *J. of Engineering for Industry*. 87:429-441, 1965.

[13]. N.H. Hanna and S. A. Tobias. “A Theory of Nonlinear Regenerative Chatter”. *J. of Engineering for Industry*, 96:247-253, 1974.

[14]. M.A. El Baradie. “Statistical Analysis of the Dynamic Cutting Coefficients and Machine Tool Stability”. *J. of Engineering for Industry*, 115:205-214, 1993.

[15]. J. Peters and P. Vanherck. “Machine Tool Stability Tests and the Incremental Stiffness”. *Annals of the CIRP*, 17:225-232, 1969.

[16]. S.M. Pandit, T.L. Subramanian, and S.M. Wu. “Modelling Machine Tool Chatter by Time Series”. *J. of Engineering for Industry*, 97:211-215, 1975.

[17]. M.U. Jen and E. B. Magrab. “The Dynamic Interactive of the Cutting Process, Workpiece, and Lathe’s Structure in Facing”. *J. of Manufacturing Science and Engineering*, 118:348-357, 1998.

[18]. L. Kops, M. Gould, and M. Mizrach. “ Improved Analysis of the Workpiece Accuracy in Turning Based on the Emerging Diameter”. *J. of Engineering for Industry*, 115:253-257, 1993.

[19]. A.M. Shawky and M. A. Elbestawi. “An Enhanced Dynamic Model in Turning Including the Effects of Ploughing Force”. *J. of Manufacturing Science and Engineering*, 119:10-20, 1997.

[20]. L.K. Daneshmend and H. A. Pak. “Model Reference Adaptive Control of Feed Force in Turning”. *J. of Dynamic Systems, Measurements and Control*, 108:215-222, 1986

[21]. T.E. Bailey, D.M. Jenkins, A.D. Spence, and M.A. Elbestawi, “Integrated Modelling for Metal Removal Operations”. *Proc. Of the ASME Dynamic Systems and Control Division*, 58:191-198, 1996.

[22]. Y. Tian, B. Shirinzadeh, D. Zhang. “A flexure-based mechanism and control methodology for ultra-precision turning operation”. *Precision Engineering. 2009, 33, 160–166*

[23]. A.H. El-Sinawi. “Two-dimensional vibration suppression in turning using optimal control of the cutting tool”. *Int. J. Machining and Machinability of Materials. 2008, Vol. 3, Nos. 1/2, 91-103.*

[24]. Hamed Moradi, M.R. Movahhedy, G. Reza Vossoughi. “Robust control strategy for suppression of regenerative chatter in turning”. *Journal of Manufacturing Processes.* (2009) , 11, 55_65

[25]. S.N. Huang, K.K. Tan, Y.S. Wong, C.W. de Silva, H.L. Goh, W.W. Tan. “Tool wear detection and fault diagnosis based on cutting force Monitoring”. *International Journal of Machine Tools & Manufacture*. 2007, 47, 444–451

[26]. Y. Zhu. “Estimation of an N-L-N Hammerstein-Wiener Model”. *Automatica*. 2002, 38, 1607-1614.

[27]. A.H. El-Sinawi, A. H., and A. R. Kashani. “Improving surface roughness in turning using a Kalman estimator-based feed forward control of tool's position”. *Journal of Materials Processing Technology*. 2005, 167, pp 54-61

[28]. M. Farina and L. Piroddi. "An iterative algorithm for simulation error based identification of polynomial input-output models using multi-step prediction". *International Journal of Control*, 83, Issue 7 July 2010 , 1442 – 1456.

[29]. L. Ljung. *System Identification: Theory for the User.* Prentice Hall; 2nd edition, 1999

[30]. A.H. El-Sinawi. "Vibration attenuation of a flexible beam mounted on a rotating compliant hub". Journal of Systems and Control Engineering, Part I, April, 2004, *218,* pp 121-135.

[31]. W. Gawronsky. *Advanced Structural Dynamics and Active Control of Structures*. Springer-Verlag, 2004.

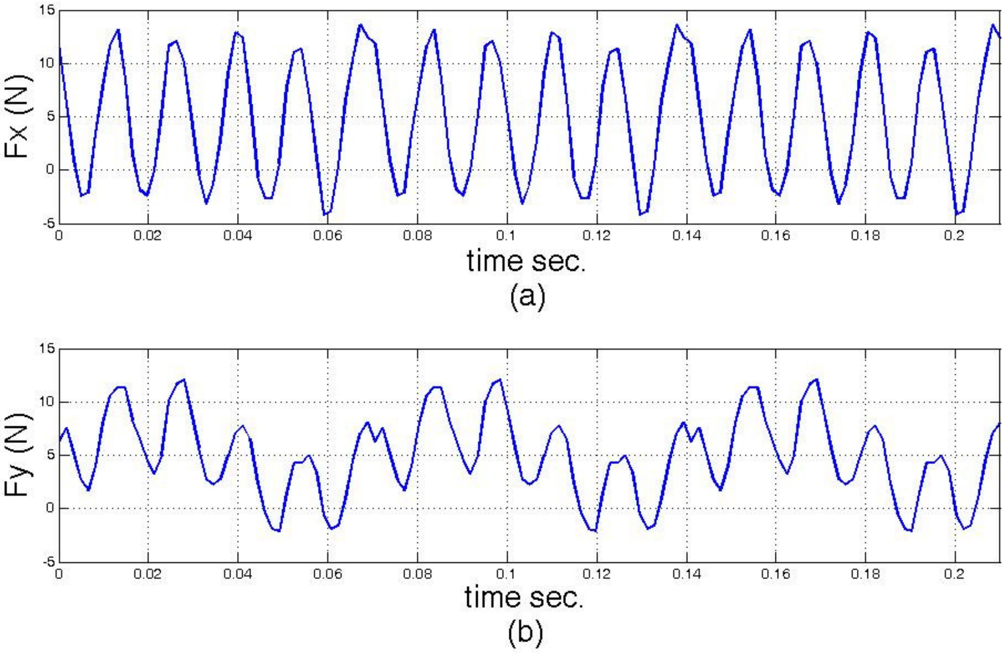


FIGURE 1: Sample of experimental dynamic cutting forces in (a) radial and (b) tangential directions.

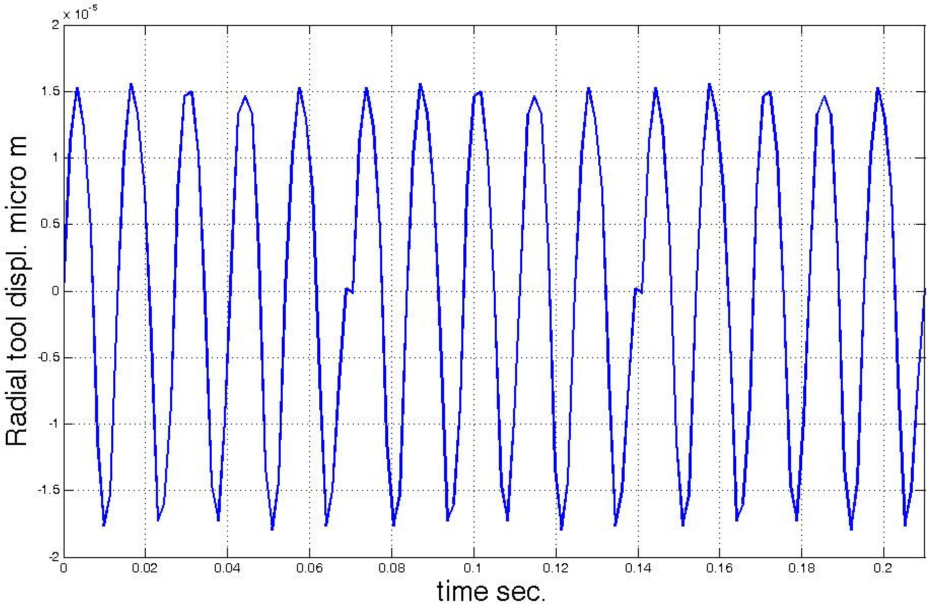


FIGURE 2: Tool displacement in the radial direction measurements during orthogonal cutting

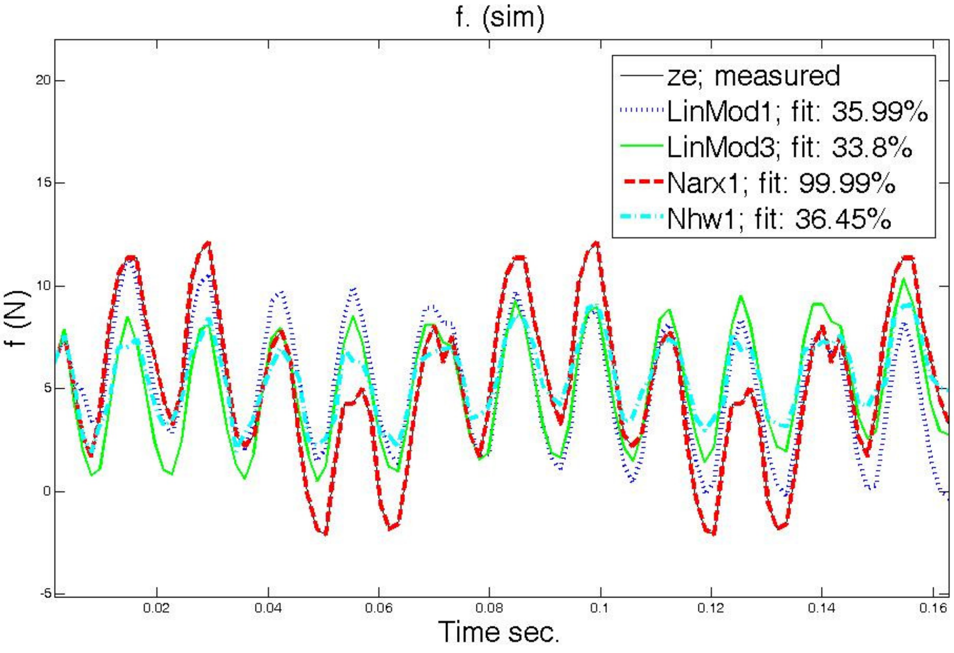


FIGURE 3: Nonlinear models constructed from experimental results

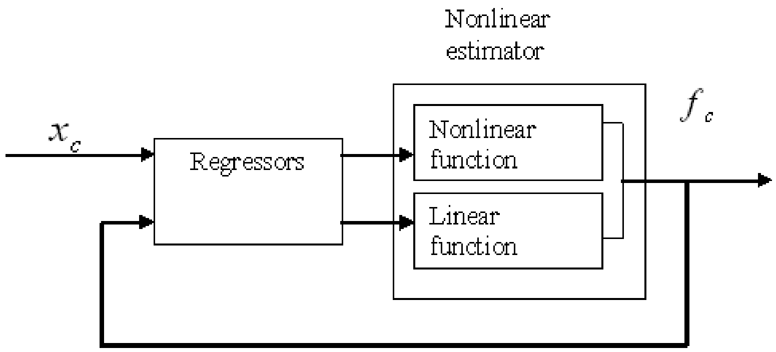


FIGURE 4: Nonlinear ARX model structure

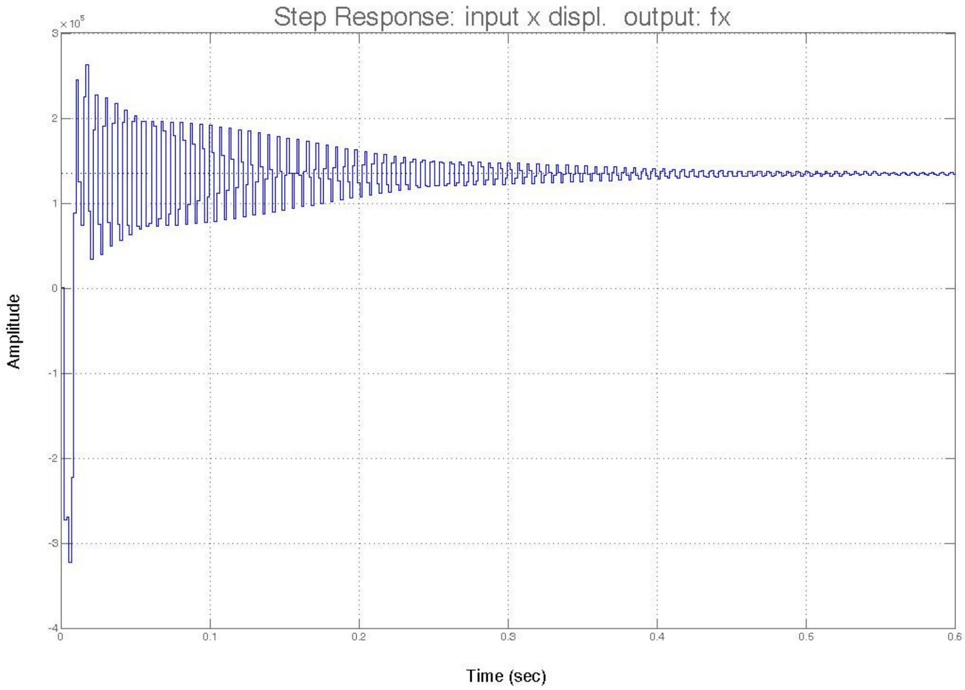


FIGURE 5: Step response of the radial force model f_x

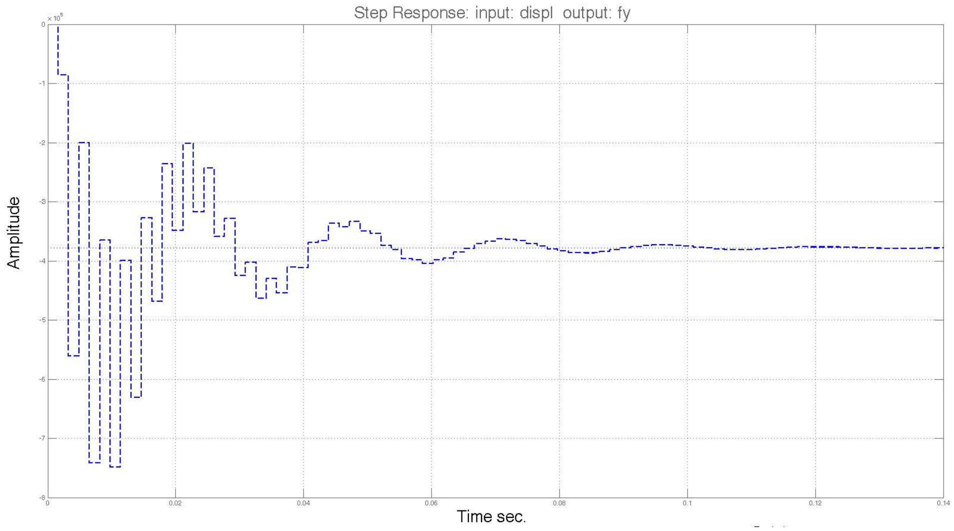


FIGURE 6: Step response of the tangential force model f_y

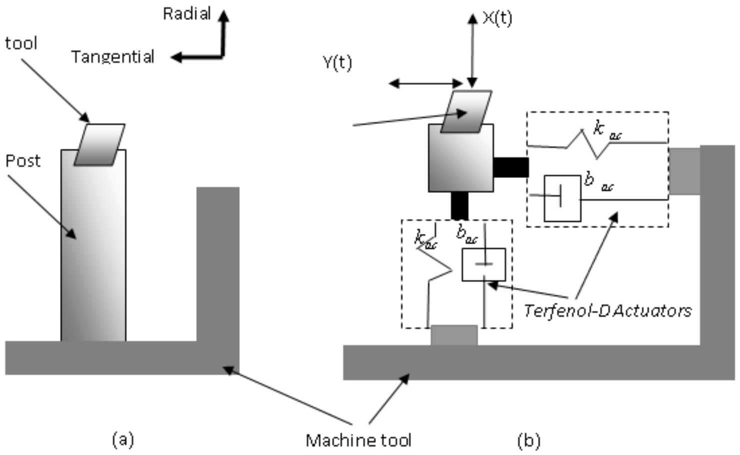


FIGURE 7: (a) Conventional machine tool, (b) machine tool retrofitted with two actuators transducer for cutter manipulation. *v(t)* measurements noise.

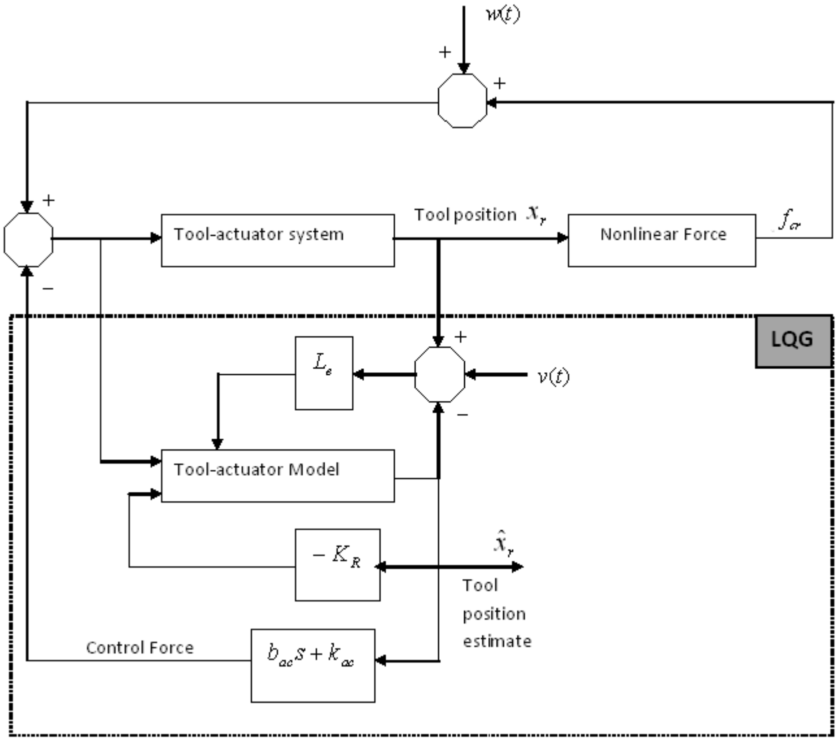


FIGURE 8: Controller implementation in the radial direction (i.e., x-direction)

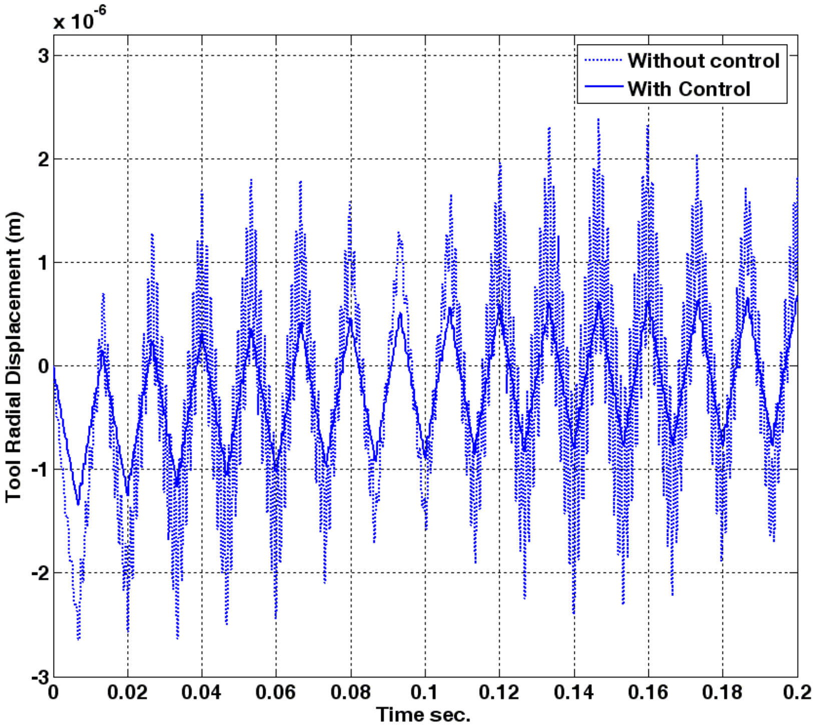


FIGURE 9: tool dynamic displacement in the radial direction

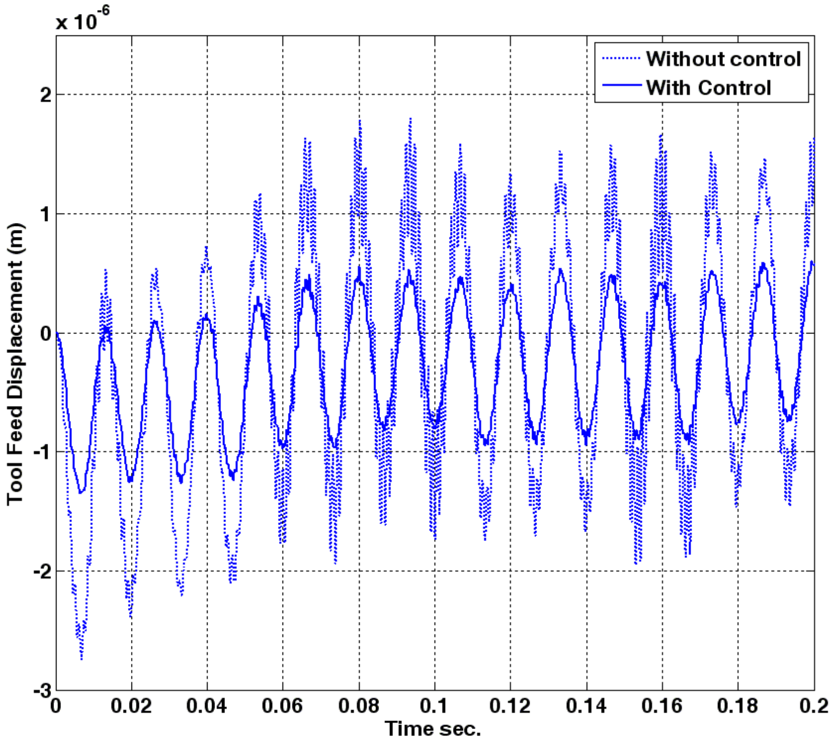


FIGURE 10: tool dynamic displacement in the tangential direction

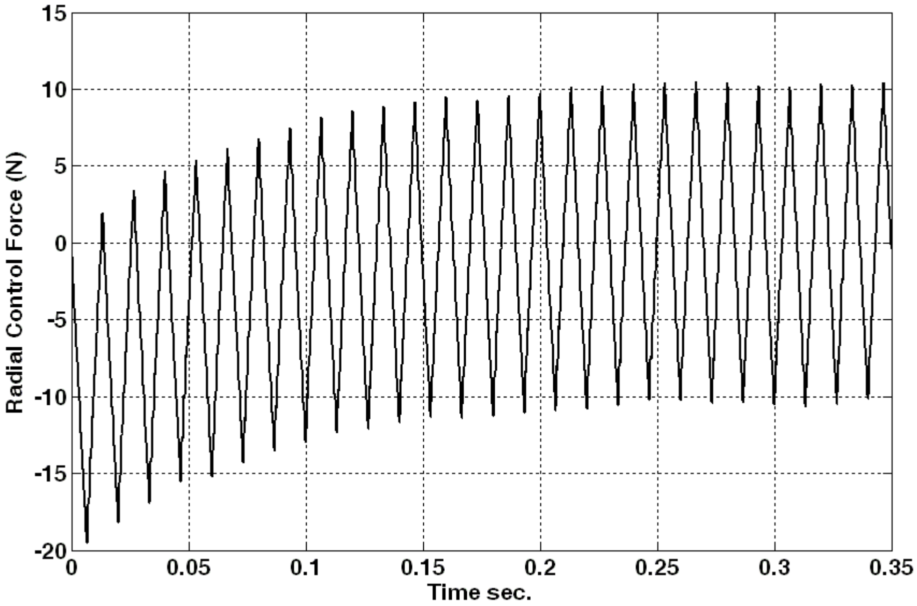


FIGURE 11: Control force in radial direction

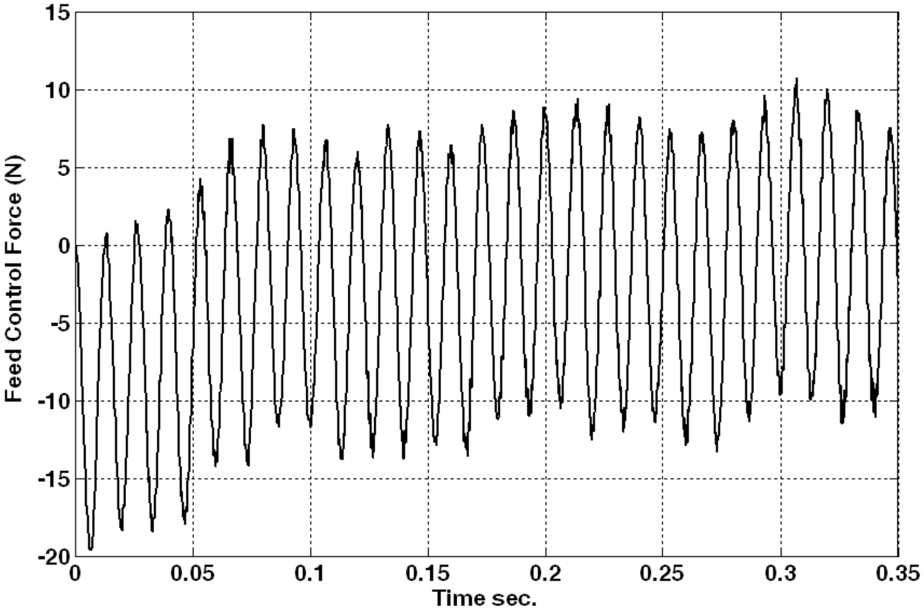


FIGURE 12: Control force in the feed direction

Appendix A: Linearization of a Nonlinear system [29]

Common use of time-varying linear systems is related to linearization of nonlinear systems around a certain “nominal” trajectory. Therefore, for a nonlinear system described by

$$\begin{aligned} x(t+1) &= f((x(t),u(t)) + r(x(t),u(t))w(t) \\ y(t) &= h(x(t)) + m(x(t),u(t))v(t) \end{aligned} \qquad \text{a.1}$$

Assuming that the disturbance terms $w(t)$ and $v(t)$ are white and small, with the nominal and disturbance-free $(w(t) \equiv 0,\ v(t) \equiv 0)$ behavior of the system corresponds to an input sequence $u^*(t)$ and corresponding trajectory $x^*(t)$. Neglecting nonlinear terms, the differences,

$$\begin{aligned} \Delta x(t) &= x(t) - x^*(t) \\ \Delta y(t) &= y(t) - h(x^*(t)) \\ \Delta u(t) &= u(t) - u^*(t) \end{aligned}$$

Are then subjected to

$$\begin{aligned} \Delta x(t+1) &= F(t)\Delta x(t) + G(t)\Delta u(t) + \overline{w}(t) \\ \Delta y(t) &= H(t)\Delta x(t) + \overline{v}(t) \end{aligned} \qquad \text{a.2}$$

Such that,

$$F(t) = \frac{\partial}{\partial x} f(x,u)\bigg|_{x^*(t),u^*(t)}, \quad G(t) = \frac{\partial}{\partial u} f(x,u)\bigg|_{x^*(t),u^*(t)}, \quad H(t) = \frac{\partial}{\partial x} h(x)\bigg|_{x^*(t)}$$

In view of the assumption of small disturbances, the cross terms with disturbance. The terms $\overline{w}(t)$ and $\overline{v}(t)$ in Eq. a.2 above are white disturbances with the following covariance properties:

$$\begin{aligned} R_1(t) &= E\,\overline{w}(t)\overline{w}^T(t) = r(x^*(t),u^*(t))E\,w(t)w^T(t)\,r^T(x^*(t),u^*(t)) \\ R_2(t) &= E\,\overline{v}(t)\overline{v}^T(t) = m(x^*(t),u^*(t))E\,v(t)v^T(t)m^T(x^*(t),u^*(t)) \\ R_{12}(t) &= r(x^*(t),u^*(t))E\,w(t)v^T(t)m^T(x^*(t),u^*(t)) \end{aligned} \qquad \text{a.3}$$

The model is now linear in the vicinity of nominal trajectory.

Design of FPGA-based Sliding Mode Controller for Robot Manipulator

Farzin Piltan SSP.ROBOTIC@yahoo.com
*Department of Electrical and Electronic Engineering,
Faculty of Engineering, Universiti Putra Malaysia 43400
Serdang, Selangor, Malaysia*

N. Sulaiman nasri@eng.upm.edu.my
*Department of Electrical and Electronic Engineering,
Faculty of Engineering, Universiti Putra Malaysia 43400
Serdang, Selangor, Malaysia*

M. H. Marhaban hamiruce@eng.upm.edu.my
*Department of Electrical and Electronic Engineering,
Faculty of Engineering, Universiti Putra Malaysia 43400
Serdang, Selangor, Malaysia*

Adel Nowzary adnowzary@yahoo.com
*Industrial Electrical and Electronic Engineering
SanatkadeheSabze Pasargad. CO (S.S.P. Co),
NO:16 , PO.Code 71347-66773, Fourth floor
Dena Apr , Seven Tir Ave , Shiraz , Iran*

Mostafa Tohidian mostafa.tohidian@gmail.com
*Industrial Electrical and Electronic Engineering
SanatkadeheSabze Pasargad. CO (S.S.P. Co),
NO:16 , PO.Code 71347-66773, Fourth floor
Dena Apr , Seven Tir Ave , Shiraz , Iran*

Abstract

One of the most active research areas in the field of robotics is robot manipulators control, because these systems are multi-input multi-output (MIMO), nonlinear, and uncertainty. At present, robot manipulators is used in unknown and unstructured situation and caused to provide complicated systems, consequently strong mathematical tools are used in new control methodologies to design nonlinear robust controller with satisfactory performance (e.g., minimum error, good trajectory, disturbance rejection). Robotic systems controlling is vital due to the wide range of application. Obviously stability and robustness are the most minimum requirements in control systems; even though the proof of stability and robustness is more important especially in the case of nonlinear systems. One of the best nonlinear robust controllers which can be used in uncertainty nonlinear systems is sliding mode controller (SMC). Chattering phenomenon is the most important challenge in this controller. Most of nonlinear controllers need real time mobility operation; one of the most important devices which can be used to solve this challenge is Field Programmable Gate Array (FPGA). FPGA can be used to design a controller in a single chip Integrated Circuit (IC). In this research the SMC is designed using VHDL language for implementation on FPGA device (XA3S1600E-Spartan-3E), with minimum chattering and high processing speed (63.29 MHz).

Keywords: Robot Manipulator, Sliding Mode Controller, Chattering Phenomenon, FPGA, VHDL language.

1. INTRODUCTION

A robot is a machine which can be programmed as a reality of tasks which it has divided into three main categories i.e. robot manipulators, mobile robots and hybrid robots. PUMA 560 robot manipulator is an articulated 6 DOF serial robot manipulator. This robot is widely used in industrial and academic area and also dynamic parameters have been identified and documented in the literature. From the control point of view, robot manipulator divides into two main sections i.e. kinematics and dynamic parts. Estimate dynamic parameters are considerably important to control, mechanical design and simulation[1].

Sliding mode controller (SMC) is one of the influential nonlinear controllers in certain and uncertain systems which are used to present a methodical solution for two main important controllers' challenges, which named: stability and robustness. Conversely, this controller is used in different applications; sliding mode controller has subsequent drawbacks i.e. chattering phenomenon, and nonlinear equivalent dynamic formulation in uncertain systems[1-2].

In order to solve the chattering in the systems output, boundary layer method should be applied so beginning able to recommended model in the main motivation. Conversely boundary layer method is constructive to reduce or eliminate the chattering; the error response quality may not always be so high. Besides using boundary layer method in the main controller of a control loop, it can be used to adjust the sliding surface slope to have the best performance (reduce the chattering and error performance)[3].

Commonly, most of nonlinear controllers in robotic applications need a mobility real time operation. FPGA-based controller has been used in this application because it is small device in size, high speed, low cost, and short time to market. Therefore FPGA-based controller can have a short execution time because it has parallel architecture [4-7].

This paper is organized as follows:
In section 2, main subject of modelling PUMA-560 robot manipulator formulation are presented. Detail of classical sliding mode controller is presented in section 3. In section 4, the main subject of FPGA-based sliding mode controller is presented. In section 5, the simulation result is presented and finally in section 6, the conclusion is presented.

2. DYNAMIC FORMULATION OF ROBOT

It is well known that the equation of an *n-DOF* robot manipulator governed by the following equation [1-2]:

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

Where $\boldsymbol{\tau}$ is actuation torque, $\boldsymbol{M}(\boldsymbol{q})$ 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:

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

Where the matrix of coriolios torque is $\boldsymbol{B}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q})$ is the matrix of centrifugal torques, and $\boldsymbol{G}(\boldsymbol{q})$ 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 $\ddot{\boldsymbol{q}}$ 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[2]:

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

)

This technique is very attractive from a control point of view. This paper is focused on the design FPGA-based controller for PUMA-560 robot manipulator.

2.1 PUMA 560 Dynamic Formulation

Position control of PUMA-560 robot manipulator is analyzed in this paper; as a result the last three joints are blocked. The dynamic equation of PUMA-560 robot manipulator is given as

$$\boldsymbol{M}(\ddot{\boldsymbol{\theta}})\begin{bmatrix}\ddot{\boldsymbol{\theta}}\mathbf{1}\\ \ddot{\boldsymbol{\theta}}_2\\ \ddot{\boldsymbol{\theta}}_3\end{bmatrix}+\mathbf{B}(\boldsymbol{\theta})\begin{bmatrix}\dot{\boldsymbol{\theta}}_1\dot{\boldsymbol{\theta}}_2\\ \dot{\boldsymbol{\theta}}_1\dot{\boldsymbol{\theta}}_3\\ \dot{\boldsymbol{\theta}}_2\dot{\boldsymbol{\theta}}_3\end{bmatrix}+\boldsymbol{C}(\boldsymbol{\theta})\begin{bmatrix}\dot{\boldsymbol{\theta}}_1^2\\ \dot{\boldsymbol{\theta}}_2^2\\ \dot{\boldsymbol{\theta}}_3^2\end{bmatrix}+\boldsymbol{G}(\boldsymbol{\theta})=\begin{bmatrix}\boldsymbol{\tau}_1\\ \boldsymbol{\tau}_2\\ \boldsymbol{\tau}_3\end{bmatrix} \qquad (4)$$

Where

$$\boldsymbol{M}(\boldsymbol{q})=\begin{bmatrix}M_{11} & M_{12} & M_{13} & 0 & 0 & 0\\ M_{21} & M_{22} & M_{23} & 0 & 0 & 0\\ M_{31} & M_{32} & M_{33} & 0 & M_{35} & 0\\ 0 & 0 & 0 & M_{44} & 0 & 0\\ 0 & 0 & 0 & 0 & M_{55} & 0\\ 0 & 0 & 0 & 0 & 0 & M_{66}\end{bmatrix} \qquad (5)$$

$$\boldsymbol{B}(\boldsymbol{q})=$$

$$\begin{bmatrix}b_{112} & b_{113} & 0 & b_{115} & 0 & b_{123} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & b_{214} & 0 & 0 & b_{223} & 0 & b_{225} & 0 & 0 & b_{235} & 0 & 0 & 0 & 0\\ 0 & 0 & b_{314} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ b_{412} & b_{412} & 0 & b_{415} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & b_{514} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} \qquad (6)$$

$$\boldsymbol{C}(\boldsymbol{q})=\begin{bmatrix}0 & C_{12} & C_{13} & 0 & 0 & 0\\ C_{21} & 0 & C_{23} & 0 & 0 & 0\\ C_{31} & C_{32} & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ C_{51} & C_{52} & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} \qquad (7)$$

$$\boldsymbol{G}(\boldsymbol{q})=\begin{bmatrix}0\\ g_2\\ g_3\\ 0\\ g_5\\ 0\end{bmatrix} \qquad (8)$$

Suppose $\ddot{q}$ is written as follows

$$\ddot{\boldsymbol{q}}=\boldsymbol{M}^{-1}(\boldsymbol{q}).\{\boldsymbol{\tau}-[\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}}\dot{\boldsymbol{q}}+\boldsymbol{C}(\boldsymbol{q})\dot{\boldsymbol{q}}^2+\boldsymbol{g}(\boldsymbol{q})]\} \qquad (9)$$

and I is introduced as

$$\boldsymbol{I}=\{\boldsymbol{\tau}-[\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}}\dot{\boldsymbol{q}}+\boldsymbol{C}(\boldsymbol{q})\dot{\boldsymbol{q}}^2+\boldsymbol{g}(\boldsymbol{q})]\} \qquad (1$$

0)

$\ddot{q}$ can be written as

$$\ddot{q} = M^{-1}(q).I \qquad (11)$$

Therefore I for PUMA-560 robot manipulator can be calculated by the following equation

$$I_1 = \tau_1 - [\ b_{112}\dot{q}_1\dot{q}_2 + b_{113}\dot{q}_1\dot{q}_3 + 0 + b_{123}\dot{q}_2\dot{q}_3] - [\ C_{12}{\dot{q}_2}^2 + C_{13}{\dot{q}_3}^2] - g_1 \qquad (12)$$

$$I_2 = \tau_2 - [\ b_{223}\dot{q}_2\dot{q}_3] - [\ C_{21}{\dot{q}_1}^2 + C_{23}{\dot{q}_3}^2] - g_2 \qquad (13)$$

$$I_3 = \tau_3 - \left[C_{31}{\dot{q}_1}^2 + C_{32}{\dot{q}_2}^2\right] - g_3 \qquad (14)$$

$$I_4 = \tau_4 - [\ b_{412}\dot{q}_1\dot{q}_2 + b_{413}\dot{q}_1\dot{q}_3] - g_4 \qquad (15)$$

$$I_5 = \tau_5 - [\ C_{51}{\dot{q}_1}^2 + C_{52}{\dot{q}_2}^2] - g_5 \qquad (16)$$

$$I_6 = \tau_6 \qquad (17)$$

3. CLASSICAL SLIDING MODE CONTROL

Sliding mode controller (SMC) is a powerful nonlinear controller which has been analyzed by many researchers especially in recent years. This theory was first proposed in the early 1950 by Emelyanov and several co-workers and has been extensively developed since then with the invention of high speed control devices[1-2].

A time-varying sliding surface $s(x,t)$ is given by the following equation:

$$s(x,t) = (\frac{d}{dt} + \lambda)^{n-1}\ \widetilde{x} = 0 \qquad (18)$$

where λ is the constant and it is positive. To further penalize tracking error integral part can be used in sliding surface part as follows:

$$\boldsymbol{s(x,t) = (\frac{d}{dt} + \lambda)^{n-1} \left(\int_0^t \tilde{x}\, dt\right) = 0} \tag{19}$$

The main target in this methodology is keep $\boldsymbol{s(x,t)}$ near to the zero when tracking is outside of $\boldsymbol{s(x,t)}$. Therefore, one of the common strategies is to find input $\boldsymbol{U}$ outside of $\boldsymbol{s(x,t)}$.

$$\boldsymbol{\frac{1}{2}\frac{d}{dt}s^2(x,t) \le -\zeta|s(x,t)|} \tag{20}$$

where $\boldsymbol{\zeta}$ is positive constant.

If **S(0)>0**→ $\frac{\mathrm{d}}{\mathrm{dt}}\mathrm{S(t)} \le -\zeta$ (21)

To eliminate the derivative term, we used an integral term from t=0 to t=$\boldsymbol{t_{reach}}$

$$\boldsymbol{\int_{t=0}^{t=t_{reach}} \frac{d}{dt}S(t) \le -\int_{t=0}^{t=t_{reach}} \eta \rightarrow S\,(t_{reach}) - S(0) \le -\zeta(t_{reach} - 0)} \tag{22}$$

Where t_{reach} is the time that trajectories reach to the sliding surface so, if we assume that S(t_{reach} = 0) then:

$$\boldsymbol{0 - S(0) \le -\eta(t_{reach}) \rightarrow t_{reach} \le \frac{S(0)}{\zeta}} \tag{23}$$

and

$$\boldsymbol{if}\; \boldsymbol{S(0)} < 0 \rightarrow 0 - S(\boldsymbol{0}) \le -\boldsymbol{\eta(t_{reach})} \rightarrow \boldsymbol{S(0)} \le -\boldsymbol{\zeta(t_{reach})} \rightarrow \boldsymbol{t_{reach}} \le \frac{|\boldsymbol{S(0)}|}{\boldsymbol{\eta}} \tag{24}$$

Equation (24) guarantees time to reach the sliding surface is smaller than $\frac{|S(0)|}{\zeta}$ if trajectories are outside of **S(t).**

$$\boldsymbol{if\; S_{t_{reach}} = S(0) \rightarrow error(x - x_d) = 0} \tag{25}$$

suppose **S** defined as

$$\boldsymbol{s(x,t) = (\frac{d}{dt} + \lambda)\quad \tilde{x} = (\dot{x} - \dot{x}_d) + \lambda(x - x_d)} \tag{26}$$

The derivation of S, namely, $\dot{S}$ can be calculated as the following formulation:

$$\boldsymbol{\dot{S} = (\ddot{x} - \ddot{x}_d) + \lambda(\dot{x} - \dot{x}_d)} \tag{27}$$

suppose define the second order system as,

$$\ddot{x} = f + u \ \rightarrow \dot{S} = f + U - \ddot{x}_d + \lambda(\dot{x} - \dot{x}_d) \qquad (28)$$

Where f is the dynamic uncertain, and also if $S = 0\ and\ \dot{S} = 0$, to have the best approximation ,$\widehat{U}$ defined by,

$$\widehat{U} = -\widehat{f} + \ddot{x}_d - \lambda(\dot{x} - \dot{x}_d) \qquad (29)$$

A simple solution to get the sliding condition when the dynamic parameters have uncertainty is the switching control law:

$$U_{dis} = \widehat{U} - K(\vec{x}, t).sgn(s) \qquad (30)$$

Where the function of $sgn(S)$ defined as;

$$sgn(s) = \begin{cases} 1 & s > 0 \\ -1 & s < 0 \\ 0 & s = 0 \end{cases} \qquad (31)$$

and the $K(\vec{x}, t)$ is the positive constant. Suppose to rewrite the equation (20) by the following equation,

$$\begin{aligned} \frac{1}{2}\frac{d}{dt}s^2(x,t) &= \dot{S}.S = \left[f - \widehat{f} - Ksgn(s)\right].S \\ &= \left(f - \widehat{f}\right).S - K|S| \end{aligned} \qquad (32)$$

Another method is using equation (23) instead of (24) to get sliding surface

$$\begin{aligned} s(x,t) &= (\frac{d}{dt} + \lambda)^2 \left(\int_0^t \widetilde{x}\, dt\right) \\ &= (\dot{x} - \dot{x}_d) + 2\lambda(\dot{x} - \dot{x}_d) - \lambda^2(x - x_d) \end{aligned} \qquad (33)$$

in this method the approximation of U can be calculated as

$$\widehat{U} = -\widehat{f} + \ddot{x}_d - 2\lambda(\dot{x} - \dot{x}_d) + \lambda^2(x - x_d) \qquad (34)$$

To reduce or eliminate the chattering it is used the boundary layer method; in boundary layer method the basic idea is replace the discontinuous method by saturation (linear) method with small neighborhood of the switching surface. This replace is caused to increase the error performance.

$$B(t) = \{x, |S(t)| \le \emptyset\}; \emptyset > 0 \qquad (35)$$

Where $\emptyset$ is the boundary layer thickness. Therefore, to have a smote control law, the saturation function $Sat(S/_{\emptyset})$ added to the control law:

$$\boldsymbol{U = K(\vec{x},t).Sat\left({}^{S}/_{\emptyset}\right)} \qquad (36)$$

Where $\boldsymbol{Sat}\left({}^{S}/_{\emptyset}\right)$ can be defined as

$$\boldsymbol{sat}\left({}^{S}/_{\emptyset}\right) = \begin{cases} \mathbf{1} & \left({}^{S}/_{\emptyset} > 1\right) \\ \mathbf{-1} & \left({}^{S}/_{\emptyset} < 1\right) \\ {}^{S}/_{\emptyset} & \left(\mathbf{-1} < {}^{S}/_{\emptyset} < 1\right) \end{cases} \qquad (37)$$

Based on above discussion, the control law for a multi degrees of freedom robot manipulator is written as:

$$\boldsymbol{\hat{\tau} = \hat{\tau}_{eq} + \hat{\tau}_{sat}} \qquad (38)$$

Where, the model-based component $\boldsymbol{\hat{\tau}_{eq}}$ is compensated the nominal dynamics of systems. Therefore $\boldsymbol{\hat{\tau}_{eq}}$ can calculate as follows:

$$\boldsymbol{\hat{\tau}_{eq} = \left[M^{-1}(B + C + G) + \dot{S}\right]M} \qquad (39)$$

Where

$$\boldsymbol{\hat{\tau}_{eq}} = \begin{bmatrix} \widehat{\tau_{eq1}} \\ \widehat{\tau_{eq2}} \\ \widehat{\tau_{eq3}} \\ \widehat{\tau_{eq4}} \\ \widehat{\tau_{eq5}} \\ \widehat{\tau_{eq6}} \end{bmatrix}, \boldsymbol{M^{-1}} = \begin{bmatrix} \boldsymbol{M_{11}} & \boldsymbol{M_{12}} & \boldsymbol{M_{13}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{M_{21}} & \boldsymbol{M_{22}} & \boldsymbol{M_{23}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{M_{31}} & \boldsymbol{M_{32}} & \boldsymbol{M_{33}} & \mathbf{0} & \boldsymbol{M_{35}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{M_{44}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{M_{55}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{M_{66}} \end{bmatrix}^{-1}$$

$$\boldsymbol{B + C + G} = \begin{bmatrix} \boldsymbol{b_{112}\dot{q}_1\dot{q}_2 + b_{113}\dot{q}_1\dot{q}_3 + 0 + b_{123}\dot{q}_2\dot{q}_3} \\ \boldsymbol{0 + b_{223}\dot{q}_2\dot{q}_3 + 0 + 0} \\ \mathbf{0} \\ \boldsymbol{b_{412}\dot{q}_1\dot{q}_2 + b_{413}\dot{q}_1\dot{q}_3 + 0 + 0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C_{12}\dot{q}_2}^{2} + \boldsymbol{C_{13}\dot{q}_3}^{2} \\ \boldsymbol{C_{21}\dot{q}_1}^{2} + \boldsymbol{C_{23}\dot{q}_3}^{2} \\ \boldsymbol{C_{31}\dot{q}_1}^{2} + \boldsymbol{C_{32}\dot{q}_2}^{2} \\ \mathbf{0} \\ \boldsymbol{C_{51}\dot{q}_1}^{2} + \boldsymbol{C_{52}\dot{q}_2}^{2} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{g_2} \\ \boldsymbol{g_3} \\ \mathbf{0} \\ \boldsymbol{g_5} \\ \mathbf{0} \end{bmatrix}$$

$$\dot{S} = \begin{bmatrix} \dot{S}_1 \\ \dot{S}_2 \\ \dot{S}_3 \\ \dot{S}_4 \\ \dot{S}_5 \\ \dot{S}_6 \end{bmatrix} \text{ and } M = \begin{bmatrix} M_{11} & M_{12} & M_{13} & 0 & 0 & 0 \\ M_{21} & M_{22} & M_{23} & 0 & 0 & 0 \\ M_{31} & M_{32} & M_{33} & 0 & M_{35} & 0 \\ 0 & 0 & 0 & M_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & M_{66} \end{bmatrix}$$

Suppose that τ_{sat} is computed as

$$\hat{\tau}_{sat} = K.sat\left({}^{S}/_{\emptyset}\right) \tag{40}$$

where

$$\hat{\tau}_{sat} = \begin{bmatrix} \widehat{\tau_{d\imath s1}} \\ \widehat{\tau_{d\imath s2}} \\ \widehat{\tau_{d\imath s3}} \\ \widehat{\tau_{d\imath s4}} \\ \widehat{\tau_{d\imath s5}} \\ \widehat{\tau_{d\imath s6}} \end{bmatrix}, K = \begin{bmatrix} K_1 \\ K_2 \\ K_3 \\ K_4 \\ K_5 \\ K_6 \end{bmatrix}, \left({}^{S}/_{\emptyset}\right) = \begin{bmatrix} \frac{S_1}{\emptyset_1} \\ \frac{S_2}{\emptyset_2} \\ \frac{S_3}{\emptyset_3} \\ \frac{S_4}{\emptyset_4} \\ \frac{S_5}{\emptyset_5} \\ \frac{S_6}{\emptyset_6} \end{bmatrix} \; and \; S = \lambda e + \dot{e}$$

Moreover by replace the formulation (40) in (38) the control output is written as ;

$$\hat{\tau} = \hat{\tau}_{eq} + K.sat\left({}^{S}/_{\emptyset}\right) = \begin{cases} \tau_{eq} + K.sgn(S) & , |S| \geq \emptyset \\ \tau_{eq} + K.{}^{S}/_{\emptyset} & , |S| < \emptyset \end{cases} \tag{41}$$

Figure 1 shows the position classical sliding mode control for PUMA-560 robot manipulator. By (41) and (39) the sliding mode control of PUMA 560 robot manipulator is calculated as;

$$\hat{\tau} = \left[M^{-1}(B + C + G) + \dot{S}\right]M + K.sat\left({}^{S}/_{\emptyset}\right) \tag{42}$$

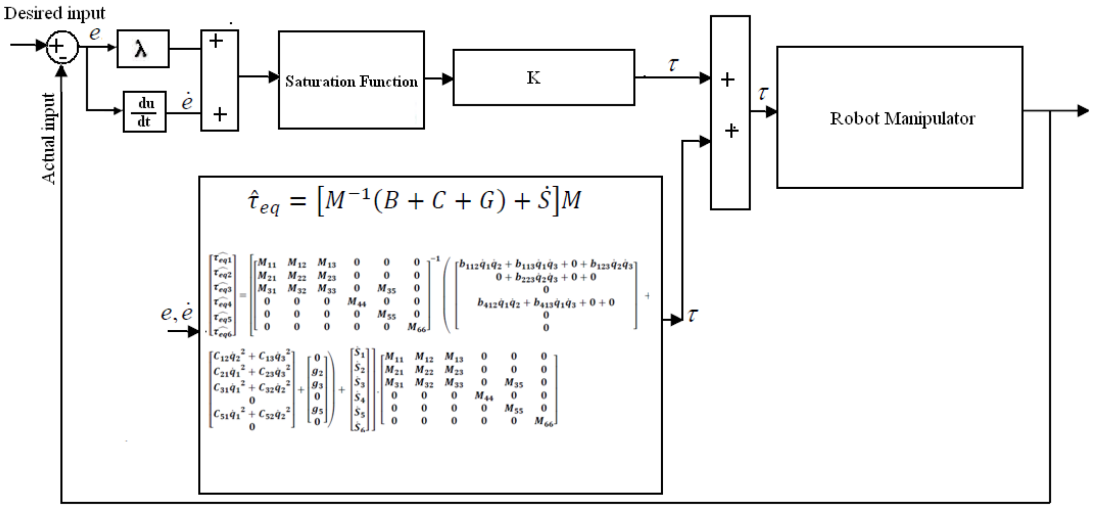


FIGURE 1: Block diagram of classical sliding mode controller

4. FPGA-BASED SLIDING MODE CONTROLLER

Research on FPGA-based control of systems is considerably growing as their applications such as industrial automation, robotic surgery, and space station's robot arm demand more accuracy, reliability, high performance. For instance, the FPGA-based controls of robot manipulator have been reported in [5-6, 8-13]. Shao and Sun [8, 10]have proposed an adaptive control algorithm based on FPGA for control of SCARA robot manipulator. They are designed this controller into two micro base controller, the linear part controller is implemented in the FPGA and the nonlinear estimation controller is implemented in DSP. Moreover this controller is implemented in a Xilinx-FPGA XC3S400 with the 20 KHz position loop frequency. The FPGA based servo control and inverse kinematics for Mitsubishi RV-M1 micro robot is presented in[9, 11-12] which to reduce the limitation of FPGA capacitance they are used 42 steps finite state machine (FSM) in 840 n second. Meshram and Harkare [5-6] have presented a multipurpose FPGA-based 5 DOF robot manipulator using VHDL coding in Xilinx ISE 11.1. This controller has two most important advantages: easy to implement and flexible. Zeyad Assi Obaid et al. [13] have proposed a digital PID fuzzy logic controller using FPGA for tracking tasks that yields semi-global stability of all closed-loop signals.

The basic information about FPGA has been reported in [4-5, 12-15]. A review of design and implementation of FPGA-based systems has been presented in [4]. The FPGA-based sliding mode control of systems has been reported in [7, 16-18]. Lin et al. [7] have presented low cost and high performance FPGA-based fuzzy sliding mode controller for linear induction motor with 80% of flip flops. The fuzzy inference system has 2 inputs (S & $\dot{S}$) and one output K_f with nine rules. Ramos et al. [16] have reported FPGA-based fixed frequency quasi sliding mode control algorithm to control of power inverter. Their proposed controller is implemented in XC4010E-3-PC84 FPGA from XILINX with acceptable experimental and theoretical performance. FPGA-based robust adaptive backstepping sliding mode control for verification of induction motor is reported in [17].

The introduction of language and architecture of Xilinx FPGA such as VHDL or Verilog in sliding mode control of robot manipulator will be investigated in this section. The Xilinx Spartan 3E FPGAs has 5 major blocks: Configurable Logic Blocks (CLBs), standard and high speed Input/output Blocks (IOBs), Block RAM's (BRAMs), Multipliers Blocks, and Digital Clock Managers (DCMs). CLBs is include flexible look up tables (LUTs) to implement memory (storage element) and logic gates. There are 4 slices per CLB each slice has two LUT's. IOB does control the rate of data between input/output pins and the internal logic gates or elements.

It supports bidirectional data with three state operation and multiplicity of signal standards. BRAMs require the data storage including 18-Kbit dual-port blocks. Product two 18-bit binary numbers is done by multiplier blocks. Self-calibrating, digital distributing solution, delaying, multiplying, dividing and phase-shift clock signal are done by DCM [15].

As shown in Figure 1, FPGA based sliding mode controller divided into two main parts: saturation part and equivalent part. To design FPGA based SMC controller using VHDL code, inputs and outputs is played important role. The block diagram of the FPGA-based sliding mode control systems for a robot manipulator is shown in Figure 2. Based on Figure 2 this block (controller) has 9 inputs and 3 outputs. Actual and desired displacements (inputs) are defined as 30 bits and the outputs (teta_dis) are defines as 35 bits in size. The desired inputs are generated from the operator and send to controllers for calculate the error and applied to sliding mode controller.

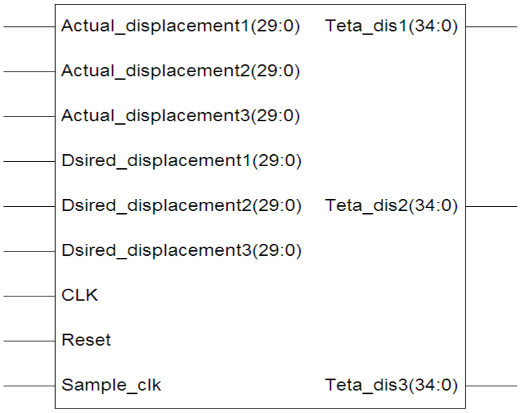

Actual_displacement1(29:0)
Actual_displacement2(29:0)
Actual_displacement3(29:0)
Dsired_displacement1(29:0)
Dsired_displacement2(29:0)
Dsired_displacement3(29:0)
CLK
Reset
Sample_clk
Teta_dis1(34:0)
Teta_dis2(34:0)
Teta_dis3(34:0)

FIGURE 2 : RTL FPGA-based controller schematic in XILINX-ISE

To convert float input data to the integer it should be multiply input value by 1000000 and then save these new values in the input files. After the completing simulation, output response should be divided over 1000000 integers to real convert values. But due to simulator (XILINX ISE 9.1) limitations and restrictions on integer data length (32 bits) and it results are 33 bit's words so at the first, controller results is divided over 2 and convert them to the integer part. Therefore the result should be divided over 500000 instead of 1000000. To robot manipulator's FPGA based position sliding mode control, controller is divided into three main sub blocks; Figure 3 shows the VHDL code and RTL schematic in Xilinx ISE software.

The table in Figure 4 indicates the Summary of XA Spartan-3E FPGA Attributes. As mentioned in above, the most significant resources are the LUT's (610 out of 29504), CLB (77 out of 3688), Slice (305 out of 14752), Multipliers (27 out of 36), registers (397), and Block RAM memory (648 K) which there are 4 slices per CLB, each slice has two LUT's. So, Number of 4 input LUTs=610, $\frac{610}{2} = 305$ slices, $\frac{305}{4} \cong 77$ CLB's, 610 registers and as a Map report Peak memory usage is 175 MB and registers in the XA3S1600E FPGA.
Moreover the table in Figure 5 illustrates the utilization summary of XA3S1600E-spartan.

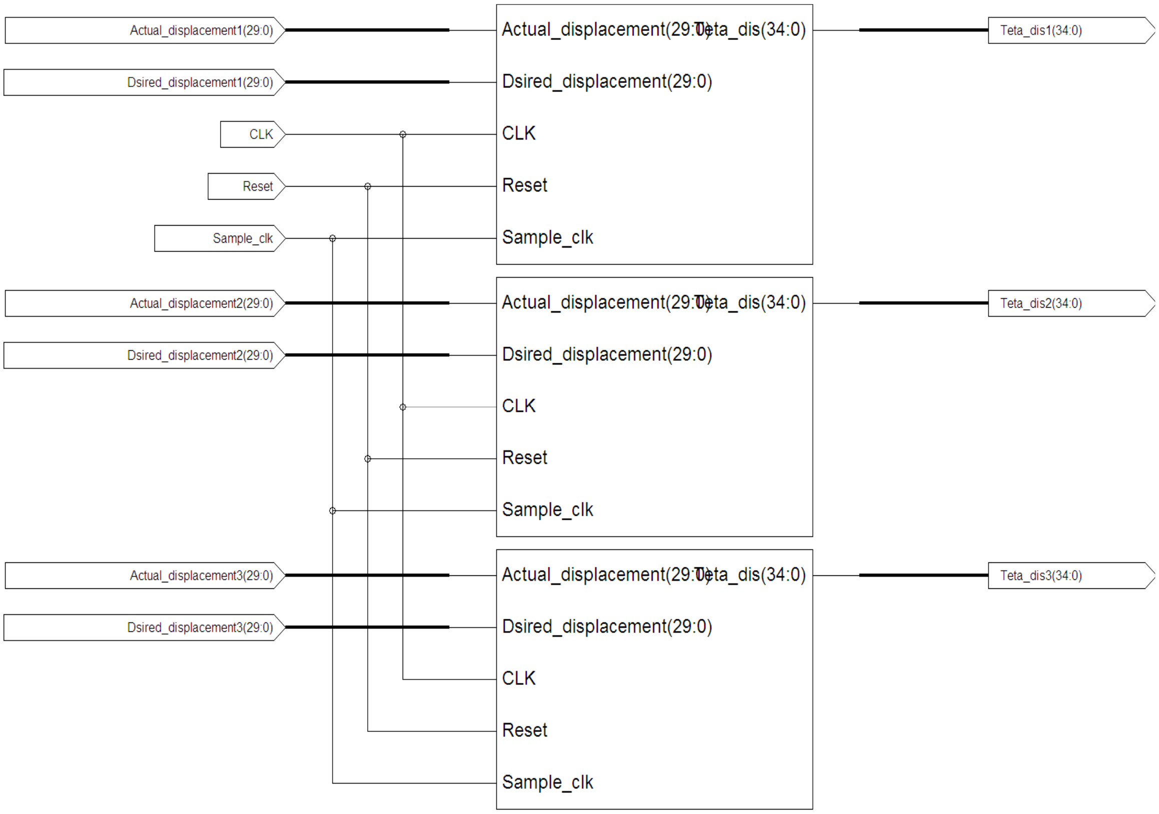
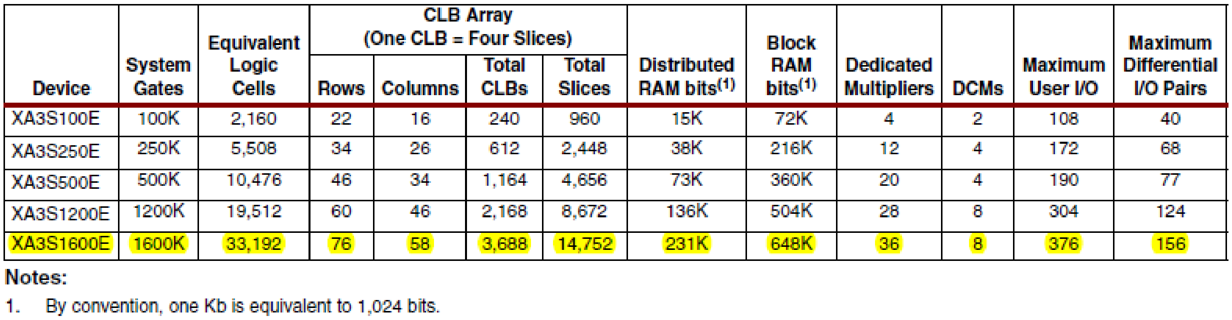

Actual_displacement1(29:0)
Dsired_displacement1(29:0)
CLK
Reset
Sample_clk
Actual_displacement2(29:0)
Dsired_displacement2(29:0)
Actual_displacement3(29:0)
Dsired_displacement3(29:0)
Actual_displacement(29:0)
Teta_dis(34:0)
Dsired_displacement(29:0)
CLK
Reset
Sample_clk
Teta_dis1(34:0)
Teta_dis2(34:0)
Teta_dis3(34:0)

FIGURE 3: Design RTL FPGA-based SMC using XILINX-ISE



Device	System Gates	Equivalent Logic Cells	CLB Array (One CLB = Four Slices) Rows	Columns	Total CLBs	Total Slices	Distributed RAM bits[1]	Block RAM bits[1]	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
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

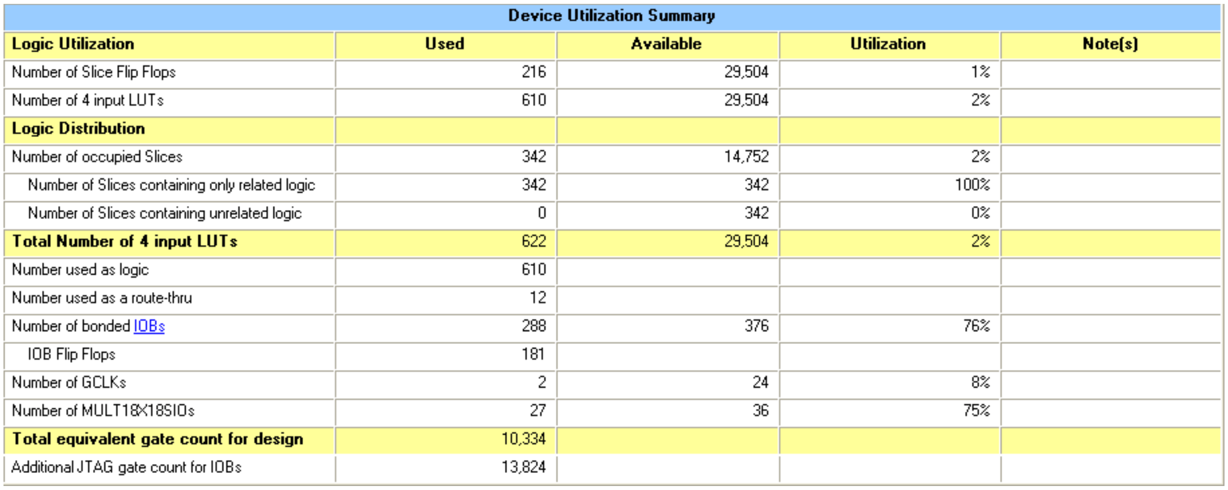
Notes:

1. By convention, one Kb is equivalent to 1,024 bits.

FIGURE 4: Summary of XA Spartan-3E FPGA attributes

5. RESULTS

PD Matlab-based sliding mode controller (PD-SMC) and PID Matlab-based sliding mode control ler (PID-SMC) and FPGA-based sliding mode controller were tested to Step response trajectory. In this simulation the first, second, and third joints are moved from home to final position without and with external disturbance. The simulation was implemented in Matlab/Simulink and Xilinx-ISE 9.1 environments. Trajectory performance, torque performance, disturbance rejection, steady state error and RMS error are compared in these controllers. It is noted that, these systems are tested by band limited white noise with a predefined 40% of relative to the input signal amplitude which the sample time is equal to 0.1. This type of noise is used to external disturbance in continuous and hybrid systems.



Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	216	29,504	1%	
Number of 4 input LUTs	610	29,504	2%	
Logic Distribution				
Number of occupied Slices	342	14,752	2%	
Number of Slices containing only related logic	342	342	100%	
Number of Slices containing unrelated logic	0	342	0%	
Total Number of 4 input LUTs	622	29,504	2%	
Number used as logic	610			
Number used as a route-thru	12			
Number of bonded IOBs	288	376	76%	
IOB Flip Flops	181			
Number of GCLKs	2	24	8%	
Number of MULT18X18SIOs	27	36	75%	
Total equivalent gate count for design	10,334			
Additional JTAG gate count for IOBs	13,824			

Figure 5 XA3S1600E device utilization summaries

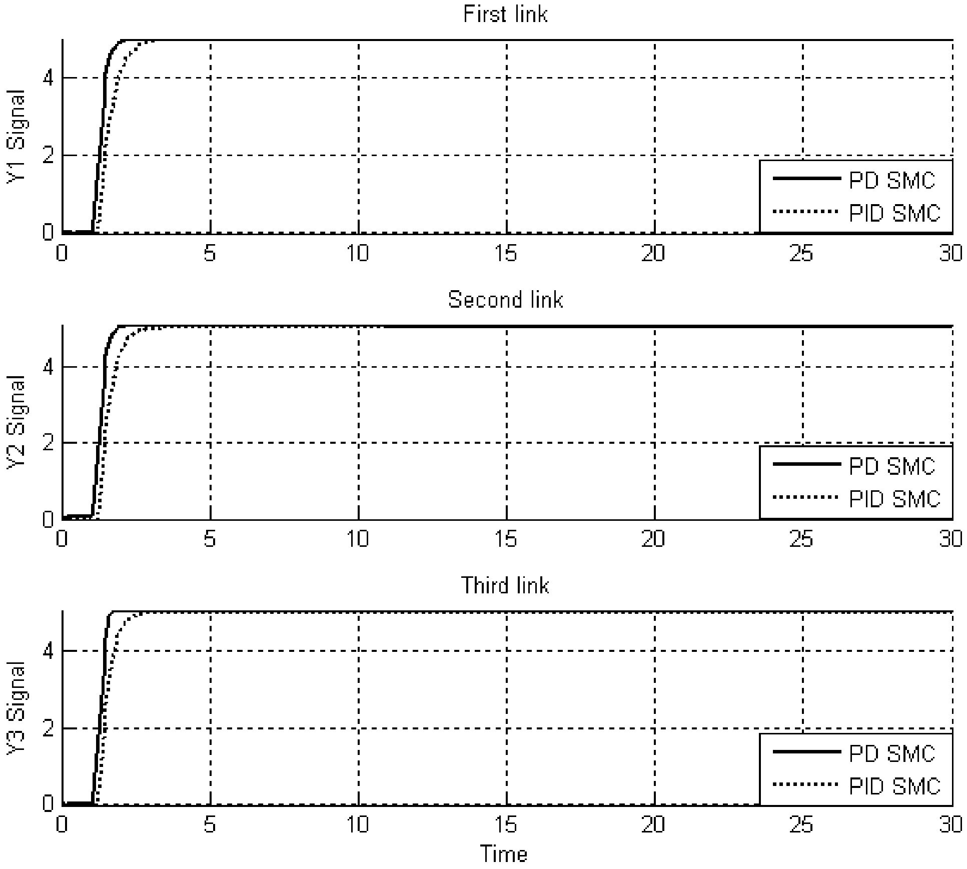


FIGURE 6 : Step PD-SMC and PID-SMC for first, second and third link trajectory without any disturbance.

By comparing step response, Figure 6, in PD and PID-SMC, conversely the PID's overshoot **(0%)** is lower than PD's **(1%)**, the PD's rise time **(0.483 Sec)** is dramatically lower than PID's **(0.9 Sec)**; in addition the Settling time in PD **(Settling time=0.65 Sec)** is fairly lower than PID **(Settling time=1.4 Sec)**.

Disturbance rejection: Figure 7 is indicated the power disturbance removal in PD and PID-SMC. As mentioned before, SMC is one of the most important robust nonlinear controllers. Besides a band limited white noise with predefined of 40% the power of input signal is applied to the step PD and PID-SMC; it found slight oscillations in trajectory responses.

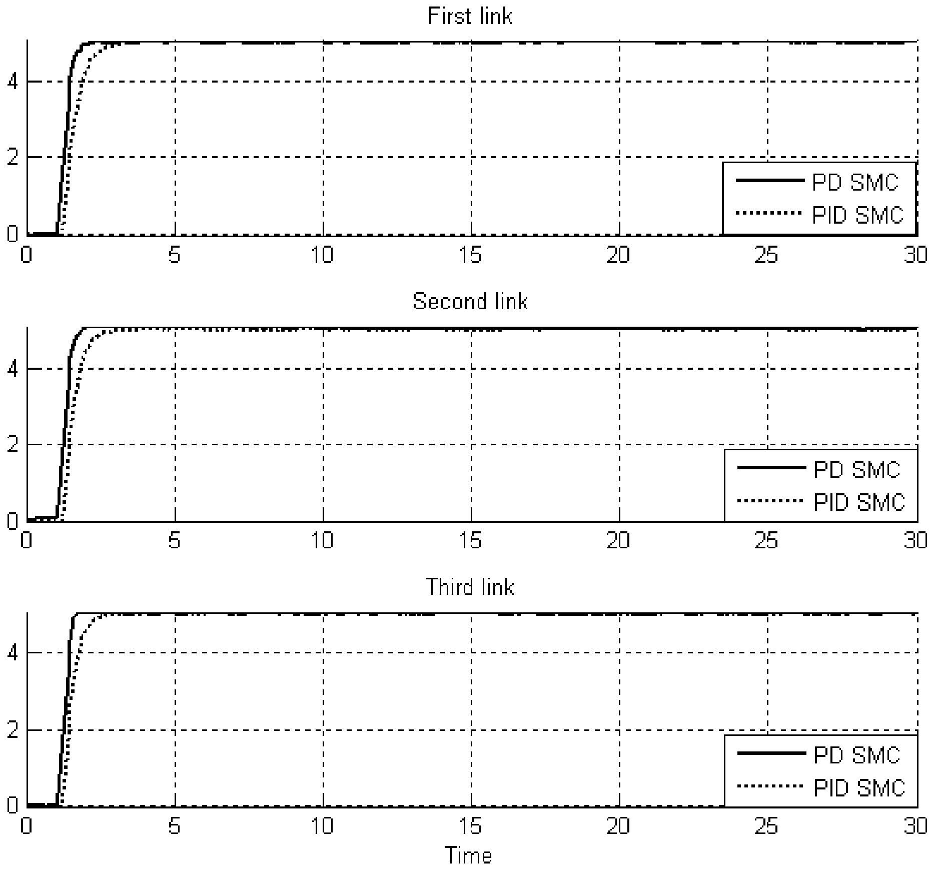


FIGURE 7: Step PD SMC and PID SMC for first, second and third link trajectory with external disturbance.

Among above graph, relating to step trajectory following with external disturbance, PID and PD SMC have slightly fluctuations. By comparing overshoot, rise time, and settling time; PID's overshoot **(0.9%)** is lower than PD's **(1.1%)**, PD’s rise time **(0.48 sec)** is considerably lower than PID’s **(0.9 sec)** and finally the Settling time in PD **(Settling time=0.65 Sec)** is quite lower than PID **(Settling time=1.5 Sec)**.

Chattering phenomenon: As mentioned in previous section, chattering is one of the most important challenges in sliding mode controller which one of the major objectives in this research is reduce or remove the chattering in system’s output. Figure 8 has shown the power of boundary layer (saturation) method to reduce the chattering in PD-SMC.

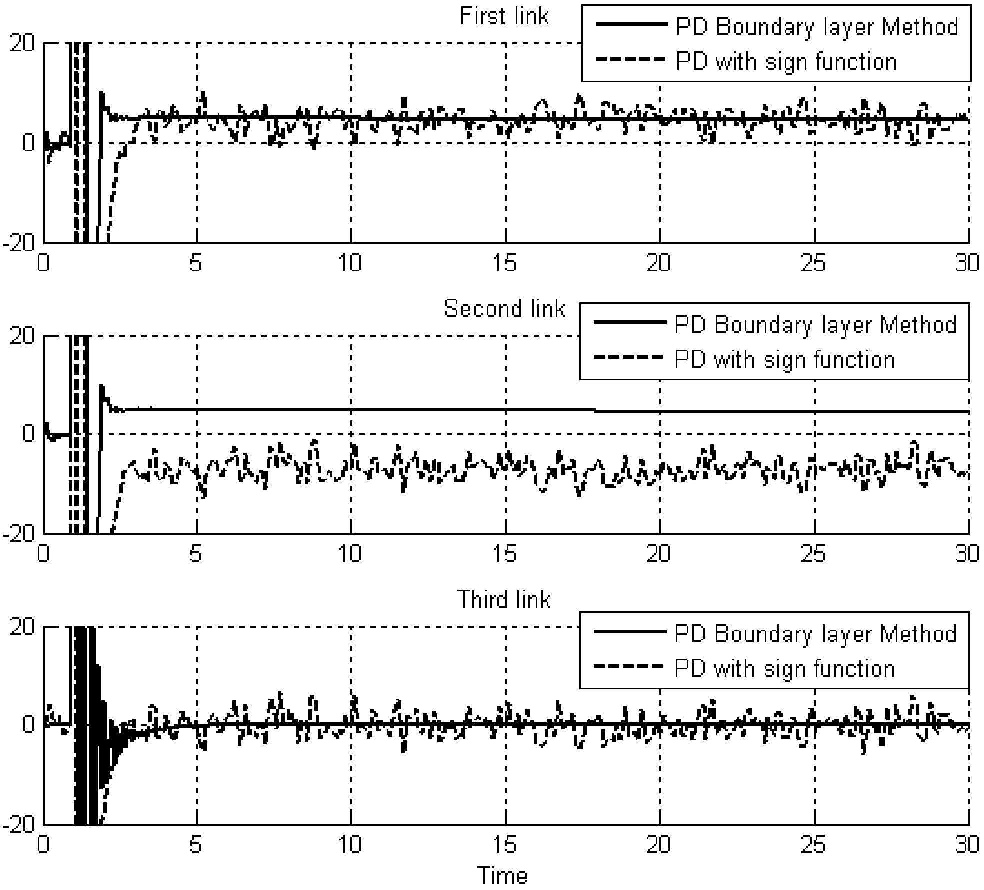


FIGURE 8 : PD-SMC boundary layer methods Vs. PD-SMC with discontinuous (Sign) function

Figure 9 has indicated the power of chattering rejection in PD and PID-SMC, with and without disturbance. As mentioned before, chattering can caused to the hitting in driver and mechanical parts so reduce the chattering is more important. Furthermore band limited white noise with predefined of 40% the power of input signal is applied the step PD and PID-SMC, it seen that the slight oscillations in third joint trajectory responses. Overall in this research with regard to the step response, PD-SMC has the steady chattering compared to the PID-SMC.

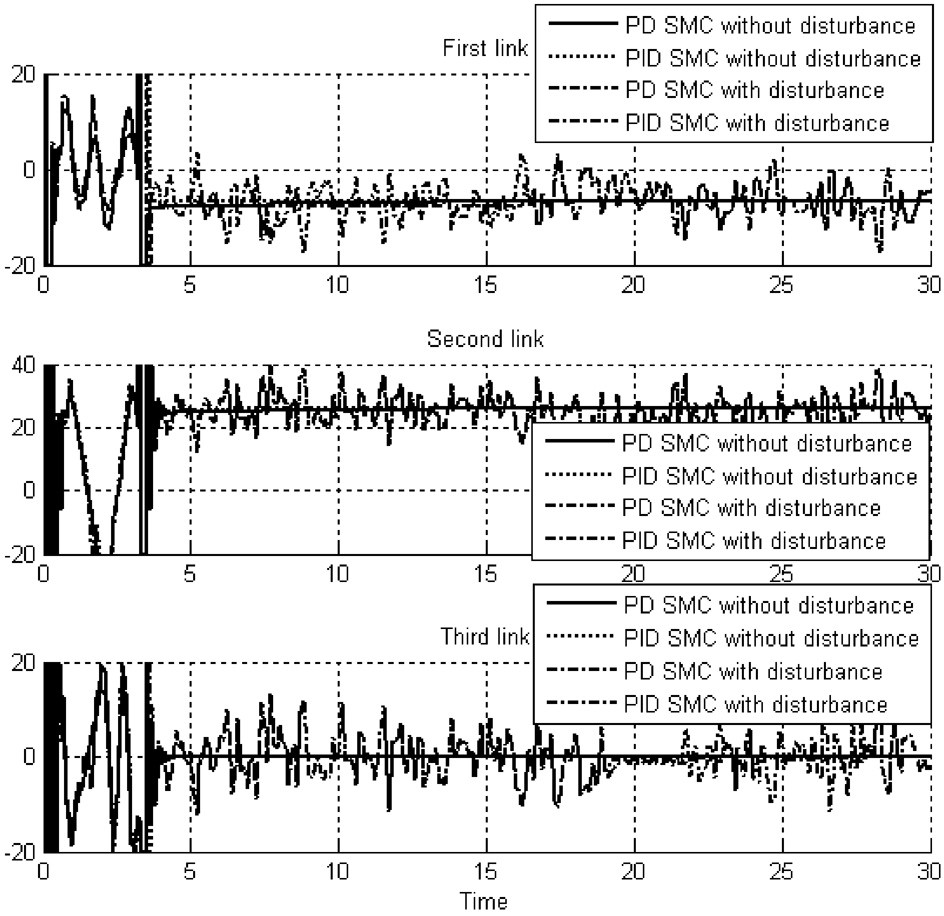


FIGURE 9 : Step PID SMC and PD SMC for first, second and third link chattering without and with disturbance.

Errors in the model: Figure 10 has shown the error disturbance in PD and PID SMC. The controllers with no external disturbances have the same error response, but PID SMC has the better steady state error. By comparing steady and RMS error in a system with no disturbance it found that the PID's errors **(Steady State error = 0 and RMS error=1e-8)** are approximately less than PD's **(Steady State error $\cong 1e-6$ and RMS error=$1.2e-6$)**.

Figure 10 shows that in first seconds; PID SMC and PD SMC are increasing very fast. By comparing the steady state error and RMS error it found that the PID's errors **(Steady State error = -0.0007 and RMS error=0.0008)** are fairly less than PD's **(Steady State error $\cong 0.0012$ and RMS error=0.0018)**, When disturbance is applied to PD and PID SMC the errors are about 13% growth.

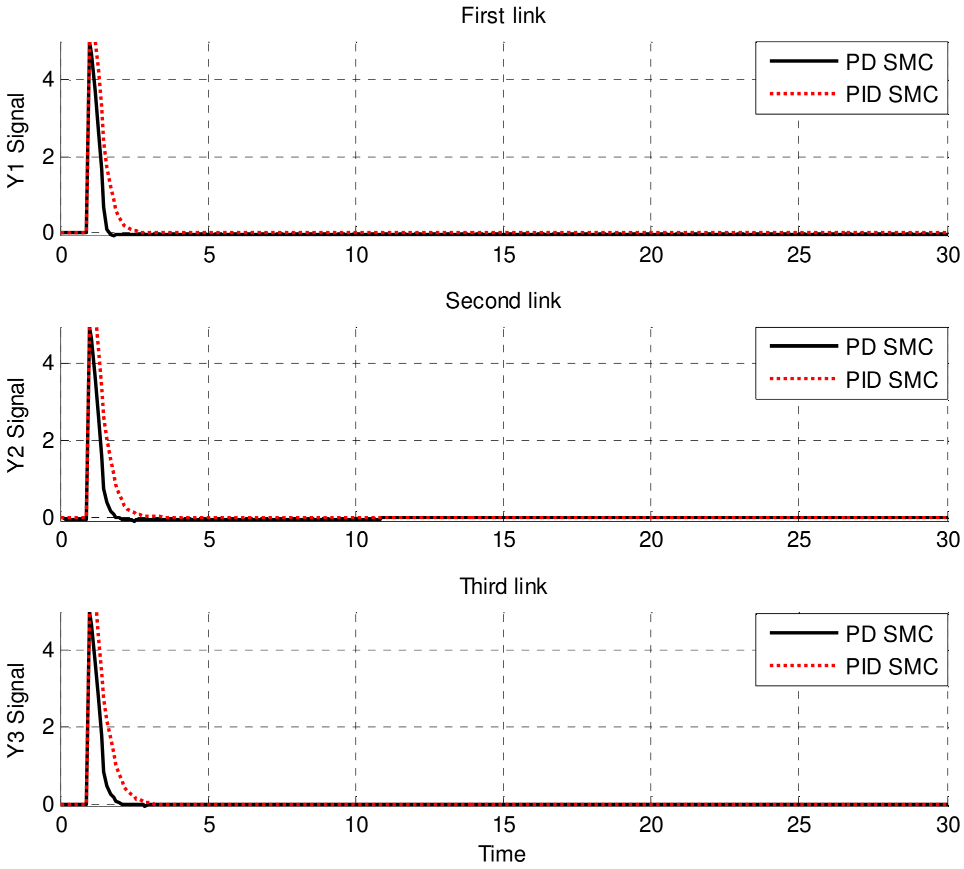
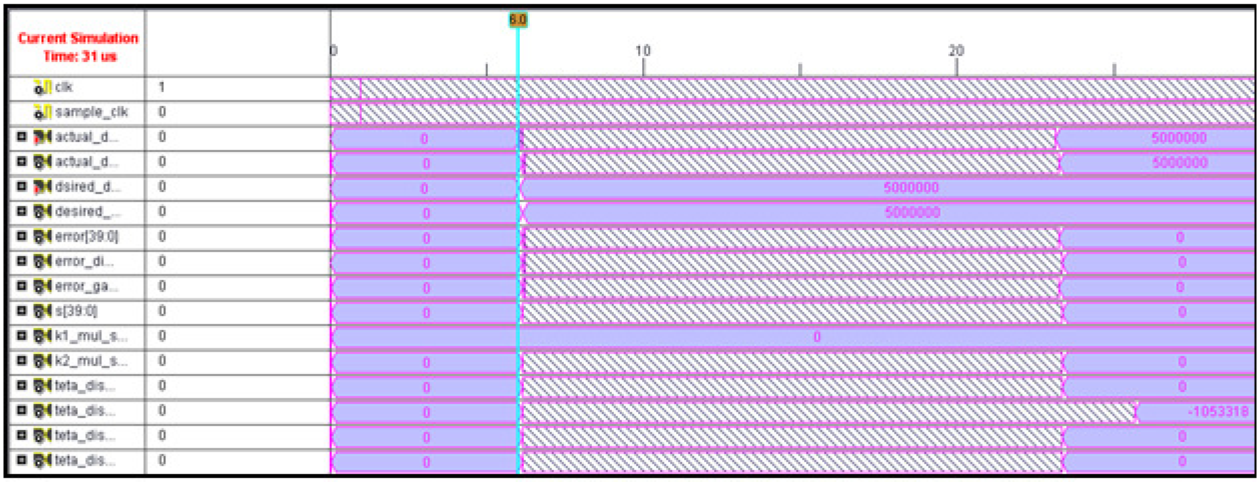


FIGURE 10 : Step PID SMC and PD SMC for first, second and third link steady state error performance.

5.2 FPGA-Based Sliding Mode Controller

Timing Detail: As a simulation result in XILINX-ISE 9.1, it found that this controller is able to make as a fast response at 15.716 ns with 63.29 MHz of a maximum frequency. From investigation and synthesis summary, this design has 15.716 ns delay to each controller for 46 logic elements and also the offset before CLOCK is 55.773 ns for 132 logic gates. Figures 11 to 13 have indicated the displacement, error performance, teta discontinuous (torque performance) at different time.

As shown in Figure 11 the controller gives action at 6 μs as a result before this time all signals and error equal to zeros.

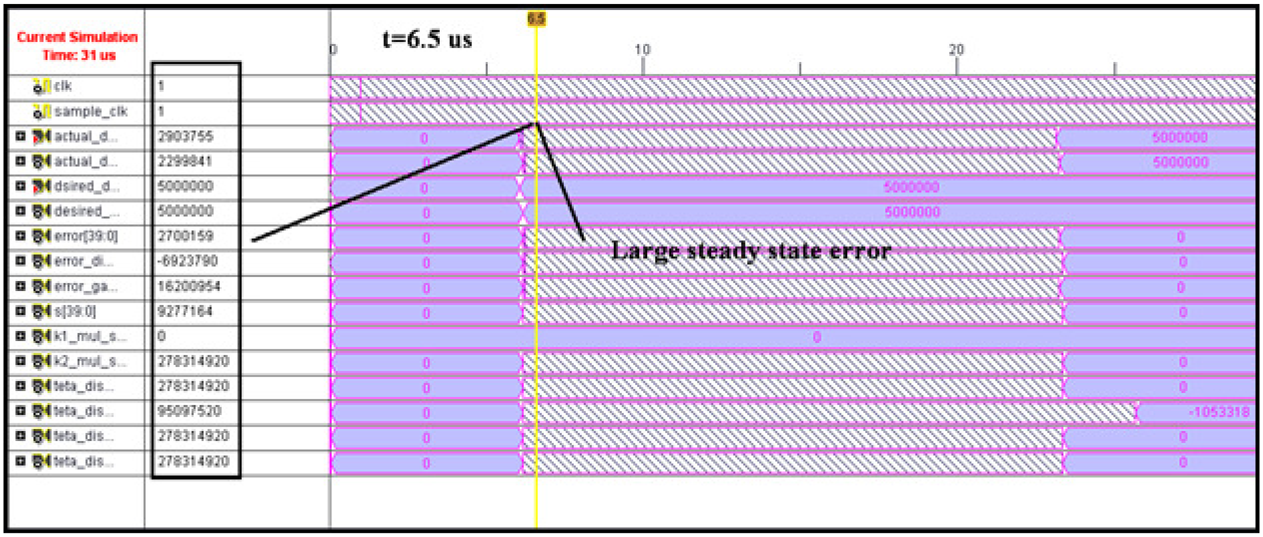


Current Simulation Time: 31 us

clk	1
sample_clk	0
actual_d...	0
actual_d...	0
dsired_d...	0
desired_...	0
error[39:0]	0
error_di...	0
error_ga...	0
s[39:0]	0
k1_mul_s...	0
k2_mul_s...	0
teta_dis...	0
teta_dis...	0
teta_dis...	0
teta_dis...	0

FIGURE 11 : Timing diagram using Xilinx ISE 9.1 of the FPGA-based SMC before running

In Figure 12 at 6.5 μs (transient response) the response has a large steady state error, 3.92, the desired displacement is 5, the actual displacement is 1.6 and the torque performance is 256.9 N.m.

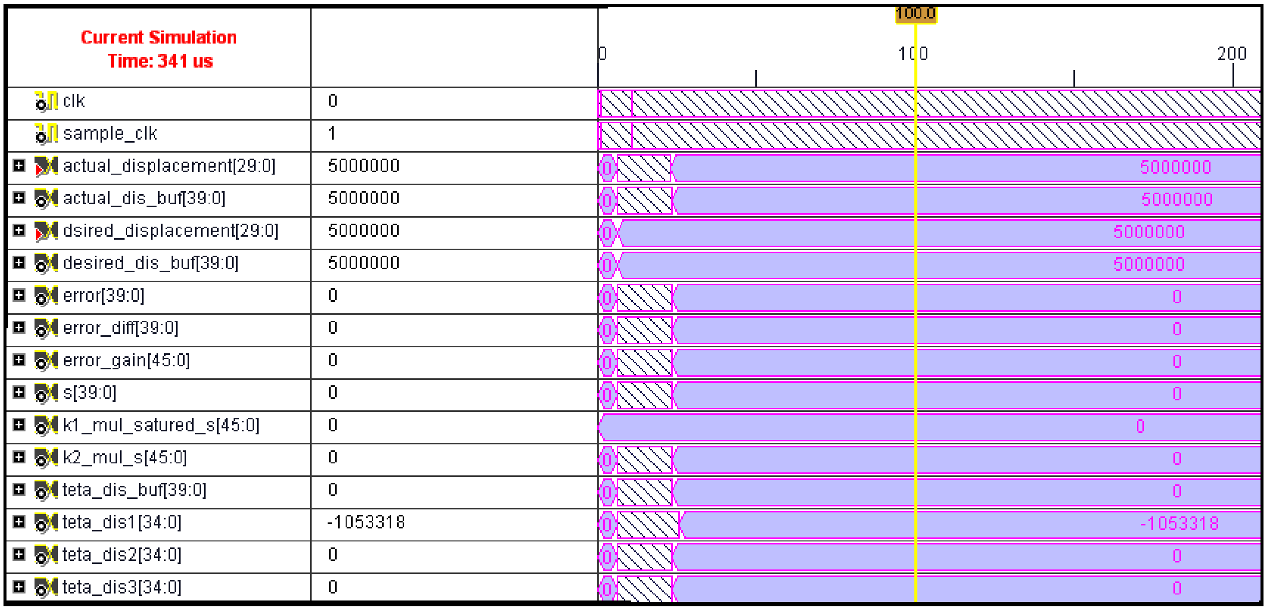


Current Simulation Time: 31 us

clk	1
sample_clk	1
actual_d...	2903755
actual_d...	2299841
dsired_d...	5000000
desired_...	5000000
error[39:0]	2700159
error_di...	-6923790
error_ga...	16200954
s[39:0]	9277164
k1_mul_s...	0
k2_mul_s...	278314920
teta_dis...	278314920
teta_dis...	95097520
teta_dis...	278314920
teta_dis...	278314920

FIGURE 12: Step PD SMC for first, second and third link for desired and actual inputs, error performance, and torque performance at 6.5 μs

Figure 13 has shown the PD-SMC at t=100μ s (steady state response), at this time the steady State error is equal to zero **,** the desired displacement is 5, the actual displacement is 5 and the torque performance is 1.005 Nm.

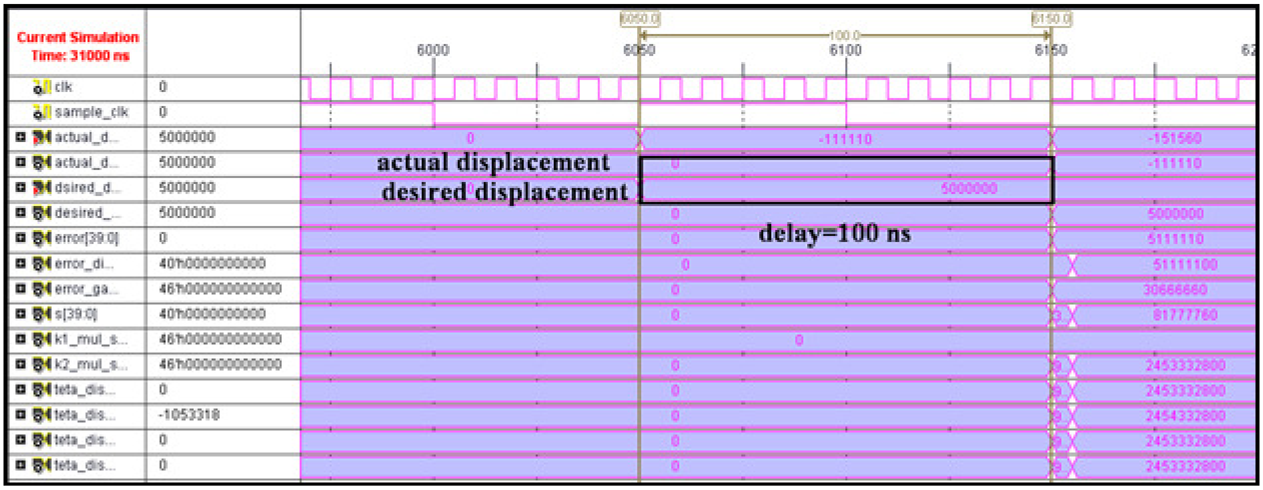


Current Simulation Time: 341 us

clk	0
sample_clk	1
actual_displacement[29:0]	5000000
actual_dis_buf[39:0]	5000000
dsired_displacement[29:0]	5000000
desired_dis_buf[39:0]	5000000
error[39:0]	0
error_diff[39:0]	0
error_gain[45:0]	0
s[39:0]	0
k1_mul_satured_s[45:0]	0
k2_mul_s[45:0]	0
teta_dis_buf[39:0]	0
teta_dis1[34:0]	-1053318
teta_dis2[34:0]	0
teta_dis3[34:0]	0

FIGURE 13 : Step PD SMC for first, second and third link for desired and actual inputs, error performance, and torque performance in $100\mu s$.

Figure 14 shows the delay with the robot manipulator affects the beginning of the response. Consequently the delay for this system is equal to $0.1\mu\ s$.



Current Simulation Time: 31000 ns

clk	0
sample_clk	0
actual_d...	5000000
actual_d...	5000000
dsired_d...	5000000
desired_...	5000000
error[39:0]	0
error_di...	40'h0000000000
error_ga...	46'h000000000000
s[39:0]	40'h0000000000
k1_mul_s...	46'h000000000000
k2_mul_s...	46'h000000000000
teta_dis...	0
teta_dis...	-1053318
teta_dis...	0
teta_dis...	0

FIGURE 14 : The delay time in PD-SMC between desired displacement and actual displacement

Figures 15 and 16 show the chattering in FPGA-based SMC. In Figure 15, the chattering analysis from 6.2 μs to $7\mu s$. It can be seen that the chattering is eliminated in this design.

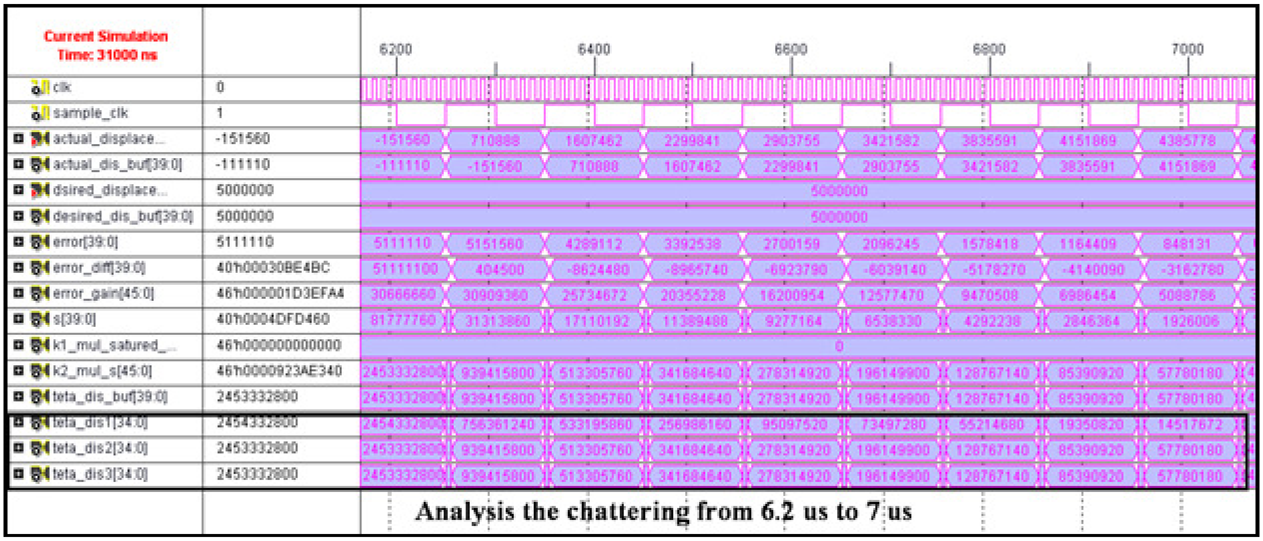

Current Simulation Time: 31000 ns
6200
6400
6600
6800
7000
clk 0
sample_clk 1
actual_displace... -151560
actual_dis_buf[39:0] -111110
dsired_displace... 5000000
desired_dis_buf[39:0] 5000000
error[39:0] 5111110
error_diff[39:0] 40'h00030BE4BC
error_gain[45:0] 46'h000001D3EFA4
s[39:0] 40'h0004DFD460
k1_mul_satured_... 46'h000000000000
k2_mul_s[45:0] 46'h0000923AE340
teta_dis_buf[39:0] 2453332800
teta_dis1[34:0] 2454332800
teta_dis2[34:0] 2453332800
teta_dis3[34:0] 2453332800
Analysis the chattering from 6.2 us to 7 us

FIGURE 15: Chattering rejections in FPGA-based SMC (from 6.2 μs to 7μs)

Figure 16 shows the power of chattering rejections in FPGA-based SMC, it found that this design is eliminated the chattering in certain situation as well as Matlab-based PD SMC.

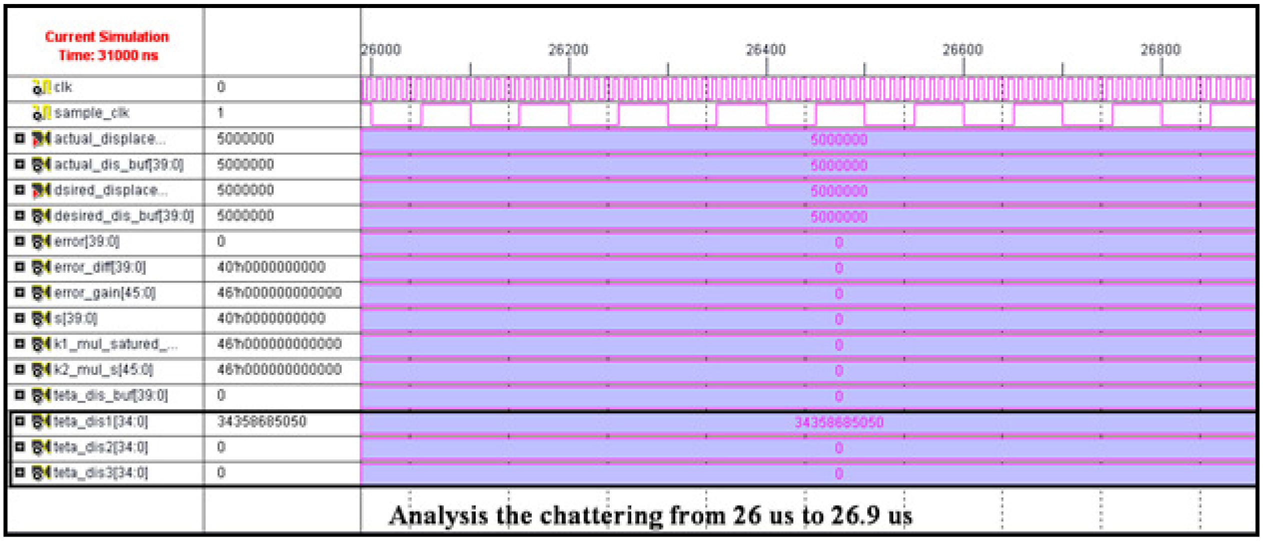

Current Simulation Time: 31000 ns
26000
26200
26400
26600
26800
clk 0
sample_clk 1
actual_displace... 5000000
actual_dis_buf[39:0] 5000000
dsired_displace... 5000000
desired_dis_buf[39:0] 5000000
error[39:0] 0
error_diff[39:0] 40'h0000000000
error_gain[45:0] 46'h000000000000
s[39:0] 40'h0000000000
k1_mul_satured_... 46'h000000000000
k2_mul_s[45:0] 46'h000000000000
teta_dis_buf[39:0] 0
teta_dis1[34:0] 34358685050
teta_dis2[34:0] 0
teta_dis3[34:0] 0
Analysis the chattering from 26 us to 26.9 us

FIGURE 16: Chattering rejections in FPGA-based SMC (from 26 μs to 26.9μs)

The best possible coefficients in Step FPGA-based PD-SMC are; $K_p = 000001 = 1, K_v = 011110 = 30,\ \emptyset_1 = \emptyset_2 = \emptyset_3 = 000001 = 1, and\ \lambda_1 = \lambda_2 = \lambda_3 = 000110 = 6.$ By comparing some control parameters such as overshoot, rise time, settling time and steady state error in Matlab–based PD-SMC, FPGA-based PD-SMC; overshoot (**PD-SMC=1% and FPGA-SMC=0.005%**), rise time (**PD-SMC=0.4 sec and FPGA-SMC8.2 μ s),** settling time (**PD-SMC=0.4 sec and FPGA-SMC=10 μ s)** and steady state error **(PD-SMC $\cong \mathbf{0.0003}$ and FPGA-SMC=0)** consequently it found that in fast response, the FPGA based-SMC's parameter has the high-quality performance.

6. CONCLUSION

Refer to the research, a position FPGA-based sliding mode control design and application to robot manipulator has proposed in order to design high performance nonlinear controller in the presence of certainties. Regarding to the positive points in sliding mode controller and FPGA the output has improved. Sliding mode controller by adding to the FPGA single chip IC has covered negative points. Obviously PUMA 560 robot manipulator is nonlinear so this paper focuses on comparison between Matlab-based sliding mode controller and FPGA-based sliding mode controller, to opt for mobility control method for the industrial manipulator.

Higher implementation speed and small chip size versus an acceptable performance is reached by designing FPGA-based sliding mode controller. This implementation considerably reduces the chattering phenomenon and error in the presence of certainties. The controller works with a maximum clock frequency of 63.29 MHz and the computation time (delay in activation) of this controller is $0.1\mu s$. As a result, this controller will be able to control a wide range of robot manipulators with a high sampling rates because it's small size versus high speed markets.

REFERENCES:

[1] T. R. Kurfess, *Robotics and automation handbook*: CRC, 2005.

[2] B. Siciliano and O. Khatib, *Springer handbook of robotics*: Springer-Verlag New York Inc, 2008.

[3] O. Kaynak, "Guest editorial special section on computationally intelligent methodologies and sliding-mode control," *IEEE Transactions on Industrial Electronics,* vol. 48, pp. 2-3, 2001.

[4] N. Sulaiman*, et al.*, "Design and Implementation of FPGA-Based Systems-A Review," *Australian Journal of Basic and Applied Sciences,* vol. 3, pp. 3575-3596, 2009.

[5] U. Meshram*, et al.*, "Robot arm controller using FPGA," 2009, pp. 8-11.

[6] U. D. Meshram and R. Harkare, "FPGA Based Five Axis Robot Arm Controller."

[7] F. J. Lin*, et al.*, "FPGA-based fuzzy sliding-mode control for a linear induction motor drive," 2005, pp. 1137-1148.

[8] X. Shao and D. Sun, "Development of an FPGA-based motion control ASIC for robotic manipulators," 2006, pp. 8221-8225.

[9] Y. S. Kung*, et al.*, "FPGA-implementation of inverse kinematics and servo controller for robot manipulator," *Proc. IEEE Int. on Robotics and Biomimetics,* pp. 1163–1168, 2006.

[10] X. Shao*, et al.*, "A new motion control hardware architecture with FPGA-based IC design for robotic manipulators," 2006, pp. 3520-3525.

[11] Y. S. Kung*, et al.*, "Design and Implementation of a Servo System for Robotic Manipulator," ed: CACS, 2005.

[12] Y. S. Kung and G. S. Shu, "Development of a FPGA-based motion control IC for robot arm," 2006, pp. 1397-1402.

[13] Z. A. Obaid*, et al.*, "Developed Method of FPGA-based Fuzzy Logic Controller Design with the Aid of Conventional PID Algorithm," *Australian Journal of Basic and Applied Sciences,* vol. 3, pp. 2724-2740, 2009.

[14] S. T. Karris, *Digital circuit analysis and design with Simulink modeling and introduction to CPLDs and FPGAs*: Orchard Pubns, 2007.

[15] K. D. Rogers, "ACCELERATION AND IMPLEMENTION OF A DSP PHASE-BASED FREQUENCY ESTIMATION ALGORITHM: MATLAB/SIMULINK TO FPGA VIA XILINX SYSTEM GENERATOR," Citeseer, 2004.

[16] R. R. Ramos*, et al.*, "A fixed-frequency quasi-sliding control algorithm: application to power inverters design by means of FPGA implementation," *Power Electronics, IEEE Transactions on,* vol. 18, pp. 344-355, 2003.

[17] F. J. Lin*, et al.*, "FPGA-based adaptive backstepping sliding-mode control for linear induction motor drive," *Power Electronics, IEEE Transactions on,* vol. 22, pp. 1222-1231, 2007.

[18] S. Lentijo*, et al.*, "FPGA based sliding mode control for high frequency power converters," 2004, pp. 3588-3592.

Design Artificial Nonlinear Robust Controller Based on CTLC and FSMC With Tunable Gain

Farzin Piltan SSP.ROBOTIC@yahoo.com
Department of Electrical and Electronic
Engineering, Faculty of Engineering,Universiti
Putra Malaysia 43400 Serdang, Selangor, Malaysia

N. Sulaiman nasri@eng.upm.edu.my
Department of Electrical and Electronic
Engineering, Faculty of Engineering,Universiti
Putra Malaysia 43400 Serdang, Selangor, Malaysia

Zahra Tajpaykar SSP.ROBOTIC@yahoo.com
Industrial Electrical and Electronic Engineering
SanatkadeheSabze Pasargad. CO (S.S.P.
Co),NO:16 , PO.Code 71347-66773, Fourth floor
Dena Apr , Seven Tir Ave , Shiraz , Iran

Payman Ferdosali SSP.ROBOTIC@yahoo.com
Industrial Electrical and Electronic Engineering
SanatkadeheSabze Pasargad. CO (S.S.P.
Co),NO:16 , PO.Code 71347-66773, Fourth floor
Dena Apr , Seven Tir Ave , Shiraz , Iran

Mehdi Rashidi SSP.ROBOTIC@yahoo.com
Industrial Electrical and Electronic Engineering
SanatkadeheSabze Pasargad. CO (S.S.P.
Co),NO:16 , PO.Code 71347-66773, Fourth floor
Dena Apr , Seven Tir Ave , Shiraz , Iran

Abstract

One of the most active research areas in the field of robotics is robot manipulators control, because these systems are multi-input multi-output (MIMO), nonlinear, time variant and uncertainty. An artificial non linear robust controller design is major subject in this work. At present, robot manipulators are used in unknown and unstructured situation and caused to provide complicated systems, consequently nonlinear classical controllers are used in artificial intelligence control methodologies to design nonlinear robust controller with satisfactory performance (e.g., minimum error, good trajectory, disturbance rejection). Sliding mode controller (SMC) and computed torque controller (CTC) are the best nonlinear robust controllers which can be used in uncertainty nonlinear. Sliding mode controller has two most important challenges in uncertain systems: chattering phenomenon and nonlinear dynamic equivalent part. Computed torque controller works very well when all nonlinear dynamic parameters are known. This research is focused on the applied non-classical method (e.g., Fuzzy Logic) in robust classical method (e.g., Sliding Mode Controller and computed torque controller) in the presence of uncertainties and external disturbance to reduce the limitations. Applying the Mamdani's error based fuzzy logic controller with minimum rules is the first goal that causes the elimination of the mathematical nonlinear dynamic in SMC and CTC. Second target focuses on the elimination of chattering phenomenon with regard to the variety of uncertainty and external disturbance in fuzzy sliding mode controller and computed torque like controller by optimization the tunable gain. Therefore fuzzy sliding mode controller with tunable gain (GTFSMC) and computed torque like controller with tunable gain (GTCTLC) will be presented in this paper.

Keywords: robot manipulator, nonlinear robust controller, classical controller, minimum error, good trajectory, disturbance rejection, sliding mode controller, computed torque controller, fuzzy sliding mode controller, computed torque like controller and tunable gain.

1. INTRODUCTION

A robot system without any controllers does not to have any benefits, because controller is the main part in this sophisticated system. The main objectives to control robot manipulators are stability and robustness. Many researchers work on designing the controller for robotic manipulators in order to have the best performance. Control of any systems divided in two main groups: linear and nonlinear controller [1].
Most of robot manipulators which work in industry are usually controlled by linear PID controllers. But the robot manipulator dynamic functions are, nonlinear with strong coupling between joints (low gear ratio), structure and unstructured uncertainty, and multi- inputs multi-outputs (MIMO) which, design linear controller is very difficult especially if the velocity and acceleration of robot manipulator be high and also when the ratio between joints gear be small [2]. To eliminate above problems in physical systems most of control researcher go toward to select nonlinear robust controller.

One of the most important powerful nonlinear robust controllers is sliding mode controller (SMC). Sliding mode control methodology was first proposed in the 1950 [3]. This controller has been analyzed by many researchers in recent years. The main reason to opt for this controller is its acceptable control performance wide range and solves some main challenging topics in control such as resistivity to the external disturbance and uncertainty. However pure sliding mode controller has some disadvantages. First, chattering problem can caused the high frequency oscillation of the controllers output [16-23]. Equivalent dynamic formulation is another disadvantage where calculation of equivalent control formulation is difficult since it is depending on the nonlinear dynamic equation [6-11].
Computed torque controller (CTC) is a powerful nonlinear controller which it widely used in control robot manipulator. It is based on Feed-back linearization and computes the required arm torques using the nonlinear Feed-back control law. This controller works very well when all dynamic and physical parameters are known but when the robot manipulator has variation in dynamic parameters, in this situation the controller has no acceptable performance[32-34]. In practice, most of physical systems (e.g., robot manipulators) parameters are unknown or time variant, therefore, computed torque like controller used to compensate dynamic equation of robot manipulator[1,3]. Research on computed torque controller is significantly growing on robot manipulator application which has been reported in [1,3, 32-34].

Some researchers had applied fuzzy logic methodology [4-5] in sliding mode controllers (FSMC) in order to reduce the chattering and to solve the nonlinear dynamic equivalent problems in pure sliding mode controller [6-11, 16-23] and the other researchers applied fuzzy logic methodology in computed torque controller (CTLC) in order to eliminate the nonlinear part in pure computed torque controller. [32-34]

This paper is organized as follows: In section 2, main subject of modelling robot manipulator formulation are presented. This section covers the following details, introducing the dynamic formulation of robot manipulator. In section 3, the main subject of sliding mode controller and formulation are presented. Detail of computed torque controller is presented in section 4. The main subject of designing fuzzy sliding mode controller with tuneable gain and computed torque like controller with tuneable gain are presented in section 5. This section covers the self tuning proposed fuzzy sliding mode controller and self tuning computed torque like controller. This method is used to reduce the chattering and estimate the equivalent (nonlinear) part in both controllers. In section 6, the simulation result is presented and finally in section 7, the conclusion is presented.

2. ROBOTIC MANIPULATOR FORMULATION

Dynamic modelling of robot manipulators is used to describe the behaviour of robot manipulator, design of model based controller, and simulation results. The dynamic modelling describe the relationship between joint motion, velocity, and accelerations to force/torque or current/voltage and also it can be used to describe the particular dynamic effects (e.g., inertia, coriolios, centrifugal, and the other parameters) to behaviour of system. It is well known that the equation of an *n-DOF* robot manipulator governed by the following equation [1,3, 13-15]:

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

Where τ is actuation torque, $\boldsymbol{M(q)}$ is a symmetric and positive define inertia matrix, $\boldsymbol{N(q,\dot{q})}$ is the vector of nonlinearity term. This robot manipulator dynamic equation can also be written in a following form [12, 35]:

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

Where $\boldsymbol{B(q)}$ is the matrix of coriolios torques, $\boldsymbol{C(q)}$ is the matrix of centrifugal torques, and $\boldsymbol{G(q)}$ 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 [1]:

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

3. CLASSICAL SLIDING MODE CONTROLLER

Basically formulation of a sliding mode controller is [3]:

$$\boldsymbol{U = U_{eq} + U_{dis}} \tag{4}$$

Where, the model-based component $\boldsymbol{U_{eq}}$ compensate for the nominal dynamics of the systems. So $\boldsymbol{U_{eq}}$ can be calculated as follows [1, 3]:

$$\boldsymbol{U_{eq} = \left[M^{-1}(B + C + G) + \dot{S}\right]M} \tag{5}$$

Suppose that $\boldsymbol{S = \lambda e + \dot{e}}$ therefore $\boldsymbol{\dot{S} = \lambda\dot{e} + \ddot{q}_d}$.

A simple solution to get the sliding condition when the dynamic parameters have uncertainty is the switching control law:

$$\boldsymbol{U_{dis} = K(\vec{x},t).sgn(s)} \qquad \boldsymbol{sgn(s)} = \begin{cases} \mathbf{1} & \boldsymbol{s} > 0 \\ \mathbf{-1} & \boldsymbol{s} < 0 \\ \mathbf{0} & \boldsymbol{s = 0} \end{cases} \tag{6}$$

Where the $\boldsymbol{K(\vec{x},t)}$ is the positive constant. To reduce chattering many researchers introduced the boundary layer methods, which in this method the basic idea is to replace the discontinuous method by saturation (linear) method with small neighbourhood of the switching surface. Therefore the saturation function $\boldsymbol{Sat}(\boldsymbol{S}/_{\emptyset})$ added to the control law:

$$\boldsymbol{U = K(\vec{x},t).Sat\left(S/_{\emptyset}\right)} \qquad \boldsymbol{sat\left(S/_{\emptyset}\right)} = \begin{cases} \mathbf{1} & (\boldsymbol{S}/_{\emptyset} > 1) \\ \mathbf{-1} & (\boldsymbol{S}/_{\emptyset} < 1) \\ \boldsymbol{S}/_{\emptyset} & (\mathbf{-1} < \boldsymbol{S}/_{\emptyset} < 1) \end{cases} \tag{7}$$

where $\emptyset$ is the width of the boundary layer, therefore the control output can be write as

$$\boldsymbol{U = U_{eq} + K.sat\left(S/_{\emptyset}\right)} = \begin{cases} \boldsymbol{U_{eq} + K.sgn(S)} & ,\boldsymbol{|S|} \geq \emptyset \\ \boldsymbol{U_{eq} + K.S/_{\emptyset}} & ,\boldsymbol{|S|} < \emptyset \end{cases} \tag{8}$$

Figure 1 shows the block diagram of classical sliding mode controller.

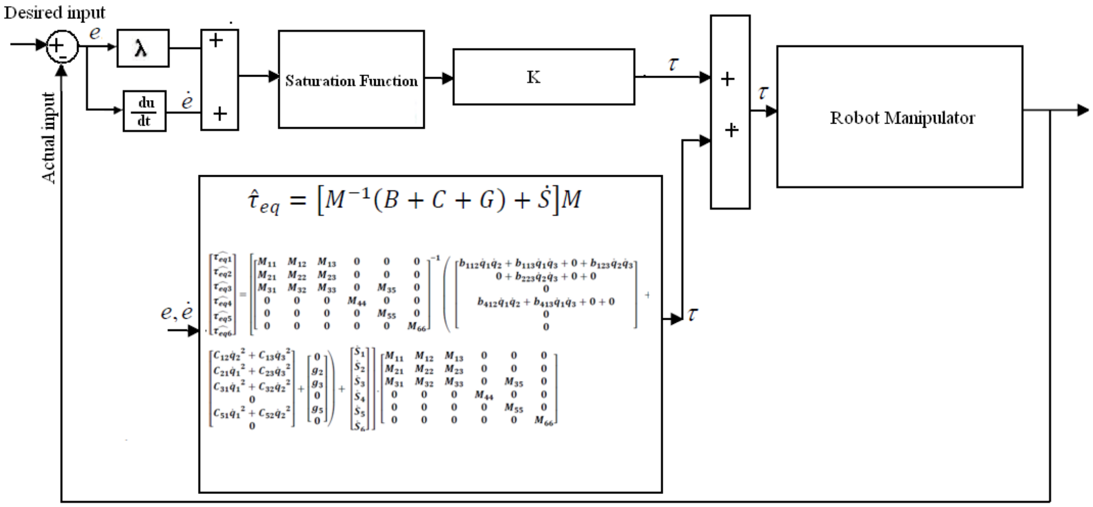


FIGURE 1 : Block diagram of classical sliding mode controller

4. COMPUTED TORQUE CONTROLLER

The central idea of Computed torque controller (CTC) is feedback linearization so, originally this algorithm is called feedback linearization controller. It is assumed that the desired motion trajectory for the manipulator $\boldsymbol{q_d(t)}$, as determined, by a path planner. Define the tracking error as:

$$\boldsymbol{e(t) = q_d(t) - q_a(t)} \qquad \textbf{(9)}$$

Where e(t) is error of the plant, $\boldsymbol{q_d(t)}$ is desired input variable, that in our system is desired displacement, $\boldsymbol{q_a(t)}$ is actual displacement. If an alternative linear state-space equation in the form $\dot{x} = \boldsymbol{Ax} + \boldsymbol{BU}$ can be defined as

$$\dot{x} = \begin{bmatrix} \mathbf{0} & I \\ \mathbf{0} & \mathbf{0} \end{bmatrix} x + \begin{bmatrix} \mathbf{0} \\ I \end{bmatrix} \boldsymbol{U} \qquad \textbf{(10)}$$

With $\boldsymbol{U} = -\boldsymbol{M}^{-1}(\boldsymbol{q}).\boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{M}^{-1}(\boldsymbol{q}).\boldsymbol{\tau}$ and this is known as the Brunousky canonical form. By equation (4) and (5) the Brunousky canonical form can be written in terms of the state $\boldsymbol{x} = [\boldsymbol{e}^T\ \dot{\boldsymbol{e}}^T]^T$ as:

$$\frac{\boldsymbol{d}}{\boldsymbol{dt}}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & I \\ \mathbf{0} & \mathbf{0} \end{bmatrix} . \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ I \end{bmatrix} \boldsymbol{U} \qquad \textbf{(11)}$$

$$\text{With } \mathbf{U} = \ddot{\mathbf{q}}_{\mathbf{d}} + \mathbf{M}^{-1}(\mathbf{q}).\{\mathbf{N}(\mathbf{q}.\dot{\mathbf{q}}) - \boldsymbol{\tau}\} \qquad \textbf{(12)}$$

Then compute the required arm torques using inverse of equation (12), namely, [1, 3, 6, 12]

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})(\ddot{\boldsymbol{q_d}} - \boldsymbol{U}) + \boldsymbol{N}(\dot{\boldsymbol{q}},\boldsymbol{q}) \qquad \textbf{(13)}$$

This is a nonlinear feedback control law that guarantees tracking of desired trajectory. Selecting proportional-plus-derivative (PD) feedback for U(t) results in the PD-computed torque controller[3];

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\big(\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{K}_{\boldsymbol{v}}\dot{\boldsymbol{e}} + \boldsymbol{K}_{\boldsymbol{p}}\boldsymbol{e}\big) + \boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}}) \qquad \textbf{(14)}$$

and the resulting linear error dynamics are

$$\big(\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{K}_{\boldsymbol{v}}\dot{\boldsymbol{e}} + \boldsymbol{K}_{\boldsymbol{p}}\boldsymbol{e}\big) = \mathbf{0} \qquad \textbf{(15)}$$

According to linear system theory, convergence of the tracking error to zero is guaranteed [2].

Where $\boldsymbol{K_p}$ and $\boldsymbol{K_v}$ are the controller gains.

The resulting schemes is shown in Figure 2, in which two feedback loops, namely, inner loop and outer loop, which an inner loop is a compensate loop and an outer loop is a tracking error loop. However, mostly parameter $\boldsymbol{N(q,\dot{q})}$ is all unknown. So the control cannot be implementation because non linear parameters cannot be determined. In the following section computed torque like controller will be introduced to overcome the problems.

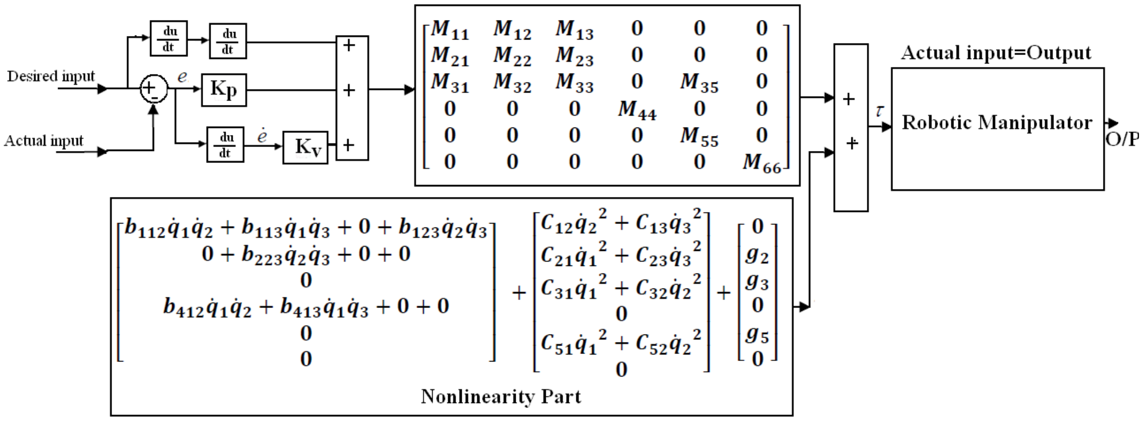

du/dt
du/dt
Desired input
Actual input
e
Kp
du/dt
$\dot{e}$
Kv
+
+
+
$\begin{bmatrix} M_{11} & M_{12} & M_{13} & 0 & 0 & 0 \\ M_{21} & M_{22} & M_{23} & 0 & 0 & 0 \\ M_{31} & M_{32} & M_{33} & 0 & M_{35} & 0 \\ 0 & 0 & 0 & M_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & M_{66} \end{bmatrix}$
+
+
τ
Actual input=Output
Robotic Manipulator
O/P
$\begin{bmatrix} b_{112}\dot{q}_1\dot{q}_2 + b_{113}\dot{q}_1\dot{q}_3 + 0 + b_{123}\dot{q}_2\dot{q}_3 \\ 0 + b_{223}\dot{q}_2\dot{q}_3 + 0 + 0 \\ 0 \\ b_{412}\dot{q}_1\dot{q}_2 + b_{413}\dot{q}_1\dot{q}_3 + 0 + 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} C_{12}\dot{q}_2^{\ 2} + C_{13}\dot{q}_3^{\ 2} \\ C_{21}\dot{q}_1^{\ 2} + C_{23}\dot{q}_3^{\ 2} \\ C_{31}\dot{q}_1^{\ 2} + C_{32}\dot{q}_2^{\ 2} \\ 0 \\ C_{51}\dot{q}_1^{\ 2} + C_{52}\dot{q}_2^{\ 2} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ g_2 \\ g_3 \\ 0 \\ g_5 \\ 0 \end{bmatrix}$
Nonlinearity Part

FIGURE 2 : Block diagram of PD-computed torque controller (PD-CTC)

5. FUZZY LOGIC AND ITS APPLICATION TO SLIDING MODE CONTROLLER (FSMC) AND COMPUTED TORQUE CONTROLLER (CTLC)

To compensate the nonlinearity for dynamic equivalent control several researchers used model base fuzzy controller instead of classical equivalent controller that was employed to obtain the desired control behaviour and a fuzzy switching control was applied to reinforce system performance. In the proposed fuzzy sliding mode control fuzzy rule base was designed to estimate the dynamic equivalent part [24-31]. A block diagram for proposed fuzzy sliding mode controller is shown in Figure 3. In this method fuzzy rule for sliding surface (S) to design fuzzy error base-like equivalent control was obtained the rules where used instead of nonlinear dynamic equation of equivalent control to reduce the chattering and also to eliminate the nonlinear formulation of dynamic equivalent control term.

$$\begin{aligned} &\mathbf{1} > if\ S\ is\ NB\ then\ \ \tau\ \ is\ NB \qquad \textbf{(16)} \\ &\mathbf{2} > if\ S\ is\ Z\ then\ \tau\ is\ Z \end{aligned}$$

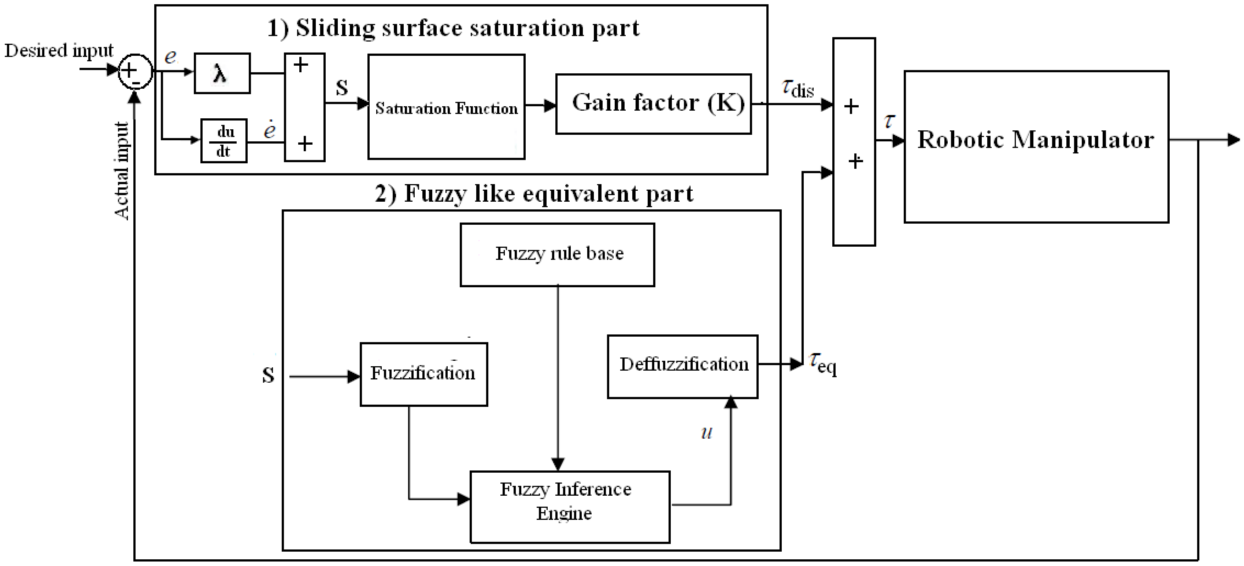


FIGURE 3 : Block diagram of proposed FSMC with minimum rule base

In FSMC the tracking error is defined as:

$$e = q_d - q_a \tag{17}$$

where $q_d = [q_{1d}, q_{2d}, q_{3d}]^T$ is desired output and $q_a = [q_{1a}, q_{2a}, q_{3a}]^T$ is an actual output. The sliding surface is defined as follows:

$$S = \dot{e} + \lambda e \tag{18}$$

where $\lambda = diag[\lambda_1, \lambda_2, \lambda_3]$ is chosen as the bandwidth of the robot manipulator controller. The time derivative of S can be calculated by the following equation

$$\dot{S} = \ddot{q}_d + \lambda_1 \dot{e} \tag{19}$$

Based on classical SMC the FSMC can be calculated as

$$\hat{\tau} = \tau_{fuzzy} + \tau_{sat} \tag{20}$$

Where, the model-based component $\hat{\tau}_{eq}$ compensate for the nominal dynamics of systems. So $\hat{\tau}_{eq}$ can be calculated as

$$\tau_{fuzzy} = \left[M^{-1}(B + C + G) + \dot{S}\right]M \tag{21}$$

and τ_{sat} is

$$\tau_{sat} = K.sat(S) \tag{22}$$

As a summary the design of fuzzy logic controller for FSMC has five steps:

1. **Determine inputs and outputs:** This controller has one input (S) and one output (τ_{fuzzy}). The input is sliding surface (S) and the output is torque(τ_{fuzzy}).
2. **Find membership function and linguistic variable:** The linguistic variables for sliding surface (S) are; Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB), and it is quantized into thirteen levels represented by: -1, -0.83, -0.66, -0.5, -0.33, -0.16, 0, 0.16, 0.33, 0.5, 0.66, 0.83, 1, and the

linguistic variables to find the torque (τ_{fuzzy}) are; Large Left (LL), Medium Left (ML), Small Left (SL), Zero (Z), Small Right (SR), Medium Right (MR), Large Right (LR) and it is quantized in to thirteen levels represented by: -85, -70.8, -56.7, -42.5, -28.3, -14.2, 0, 14.2, 28.3, 42.5, 56.7, 70.8, 85.

3. **Choice of shape of membership function:** In this work triangular membership function was selected as shown in Figure 4.
4. **Design fuzzy rule table:** design the rule base of fuzzy logic controller can play important role to design best performance SMFC, suppose that two fuzzy rules in this controller are

F.R^1: *IF S is Z, THEN τ is Z.* **(23)**
F.R^2: *IF e is (PB) THEN τ is (LR).*

The complete rule base for this controller is shown in Table 1.

S	NB	NM	NS	Z	PS	PM	PB
τ	LL	ML	SL	Z	SR	MR	LR

TABLE 1: Rule table for proposed FSMC

The control strategy that deduced by table.1 are

- If sliding surface (S) is N.B, the control applied is N.B for moving S to S=0.
- If sliding surface (S) is Z, the control applied is Z for moving S to S=0.

5. **Defuzzification**: The final step to design fuzzy logic controller is deffuzification , there are many deffuzzification methods in the literature, in this controller the COG method will be used, where this is given by

$$COG(x_k, y_k) = \frac{\sum_i U_i \sum_{j=1}^{r} \cdot \mu_u(x_k, y_k, U_i)}{\sum_i \sum_{j=1}^{r} \cdot \mu_u(x_k, y_k, U_i)} \qquad (24)$$

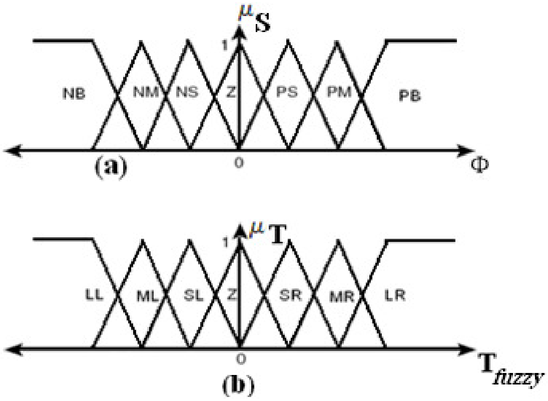


FIGURE 4 : Membership function: a) sliding surface b) torque

As mention previously, computed torque like controller (CTLC) is fuzzy controller based on computed torque method for easy implementation, stability, and robustness. The main drawback of CTLC is the value of gain updating factor K_p and K_v must be pri-defined very carefully and the most important advantage of

CTLC compare to pure CTC is a nonlinearity dynamic parameter. It is basic that the system performance is sensitive to the gain updating factors for both computed torque controller and computed torque like controller application. For instance, if large value of K_V is chosen the response is very fast but the system is very unstable and conversely, if small value of K_V considered the response of system is very slow but the system is very stable. Therefore, calculate the optimum value of gain updating factors for a system is one of the most important challenging works. However most of time the control performance for FLC and CTLC is similar to each other, but CTLC has two most important advantages:
The number of rule base is smaller
Increase the robustness and stability
In this method the control output can be calculated by

$$\boldsymbol{\tau} = \hat{\boldsymbol{\tau}} + \boldsymbol{\tau}_{\mathbf{fuzzy}(S)} \tag{25}$$

Where $\hat{\tau}$the nominal compensation is term and $\tau_{fuzzy(s)}$ is the output of computed torque fuzzy controller.

The most important target in computed torque like controller (CTLC) is design computed torque control combined to fuzzy logic systems to solve the problems in classical computed torque controller. To compensate the nonlinearity of nonlinear dynamic part several researchers used model base fuzzy controller instead of classical nonlinear dynamic part that was employed to obtain the desired control behaviour and a fuzzy switching control was applied to reinforce system performance. In proposed fuzzy computed torque controller the author design fuzzy rule base to estimate the dynamic nonlinear part. A block diagram for proposed fuzzy computed controller is shown in Figure 5.

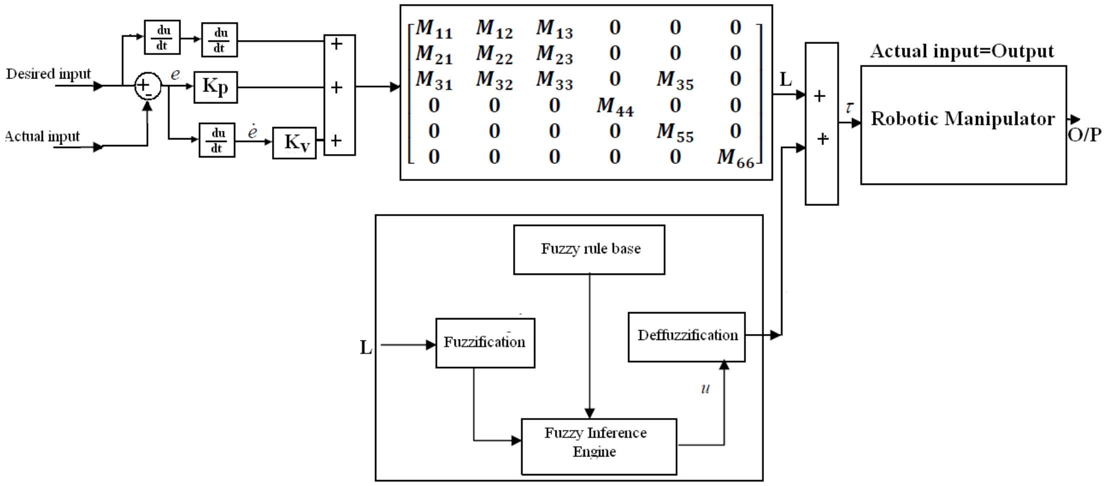


FIGURE 5 : Block diagram of proposed fuzzy computed torque controller with minimum rule base

The sliding surface is defined as follows:

$$\boldsymbol{L} = \boldsymbol{M}(\boldsymbol{q})\left(\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{K}_{\boldsymbol{v}}\dot{\boldsymbol{e}} + \boldsymbol{K}_{\boldsymbol{p}}\boldsymbol{e}\right) \tag{26}$$

Based on classical computed torque controller for a multi DOF robot manipulator:

$$\hat{\boldsymbol{\tau}} = \hat{\boldsymbol{\tau}}_{\boldsymbol{nonlinear}} + \boldsymbol{\tau}_{\boldsymbol{lin}} \tag{27}$$

where, the model-based component $\hat{\tau}_{nonlinear}$ compensate for the nominal dynamics of systems. So $\hat{\tau}_{nonlinear}$ can calculate as follows:

$$\hat{\boldsymbol{\tau}}_{\boldsymbol{nonlinear}} = \boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}}\dot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q})\dot{\boldsymbol{q}}^2 + \boldsymbol{g}(\boldsymbol{q}) \tag{28}$$

and $\boldsymbol{\tau}_{\boldsymbol{lin}}$ can calculate as follows:

$$\tau_{lin} = M(q)\left(\ddot{q}_d + K_v\dot{e} + K_p e\right) \tag{29}$$

In proposed FSMC nonlinear control part replaced by Mamdani's fuzzy inference term, therefore (27) can be rewrite as the following equation

$$\hat{\tau} = \tau_{fuzzy} + \tau_{lin} \tag{30}$$

6. GAIN TUNING FUZZY SLIDING MODE CONTROLLER (GTFSMC) AND GAIN TUNING COMPUTED TORQUE LIKE CONTROLLER (GTCTLC)

This section focuses on, self tuning gain updating factor for most important factor in FSMC, namely, sliding surface slope (λ) and in self tuning computed torque controller (GTCTLC) namely, nonlinear equivalent part (nonlinear term of controller). The block diagram for this method is shown in Figure 6. In this controller the actual sliding surface gain (λ) is obtained by multiplying the sliding surface with gain updating factor(α). The gain updating factor(α) is calculated on-line by fuzzy dynamic model independent which has sliding surface (S) as its inputs. The gain updating factor is independent of any dynamic model of robotic manipulator parameters.

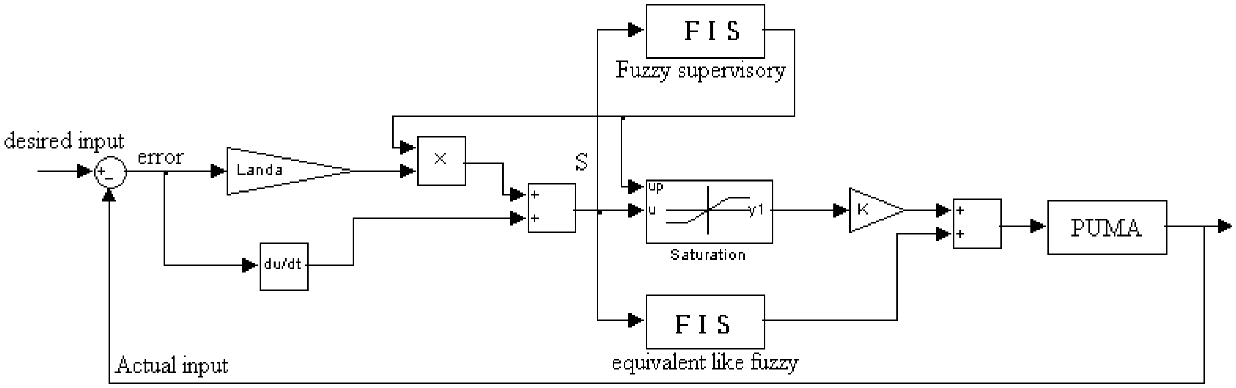


FIGURE 6 : Block diagram of proposed gain tuning fuzzy sliding mode controller with minimum rule base in fuzzy equivalent part and fuzzy supervisory.

proportional and derivative gain updating factor of the computed torque controller continuously in real-time. In this way, the performance of the overall system is improved with respect to the classical computed torque controller. Therefore this section focuses on, self tuning gain updating factor for two type most important factor in CTLC, namely, proportional gain updating factor (K_P) and derivative gain updating factor (K_v). Gain tuning-CTLC has strong resistance and solves the uncertainty problems. The block diagram for this method is shown in Figure 7.

In this controller the actual gain updating factor (K_{new}) is obtained by multiplying the old gain updating factor (K_{old}) with the output of supervisory fuzzy controller(α). The output of fuzzy supervisory controller (α) is calculated on-line by fuzzy dynamic model independent which has sliding surface (S) as inputs.. The value of α is not longer than 1 but it calculated on-line from its rule base. The scale factor, Kv and Kp are updated by equations (18) and (19),

$${K_v}^{new} = {K_v}^{old} \times \alpha \tag{31}$$

$${K_p}^{new} = {K_p}^{old} \times \alpha \tag{32}$$

6. Simulation Result

Computed torque controller (CTC), classical sliding mode control (SMC), fuzzy sliding mode control (FSMC), gain tuning computed torque like controller (GTCTLC) and gain tuning fuzzy sliding mode controller (GTFSMC) are implemented in Matlab/Simulink environment for 3 DOF robot manipulator. Tracking performance, error, and robustness are compared.

Tracking performances: Figure 8, 9 and 10 shows tracking performance for first, second and third link of robot manipulator with above controllers. By comparing step response trajectory without disturbance in above controllers, it is found that the GTCLC and GTFSMC overshoot **(1.32%)** are lower than CTC and SMC **(6.44%)**, all of them have about the same rise time. Besides the Steady State and RMS error in GTCTLC and GTFSMC **(Steady State error =0 and RMS error=0)** are fairly lower than CTC and SMC **(Steady State error** $\cong -3^{-5}$ **and RMS error=**-1.6×10^{-5}**)**.

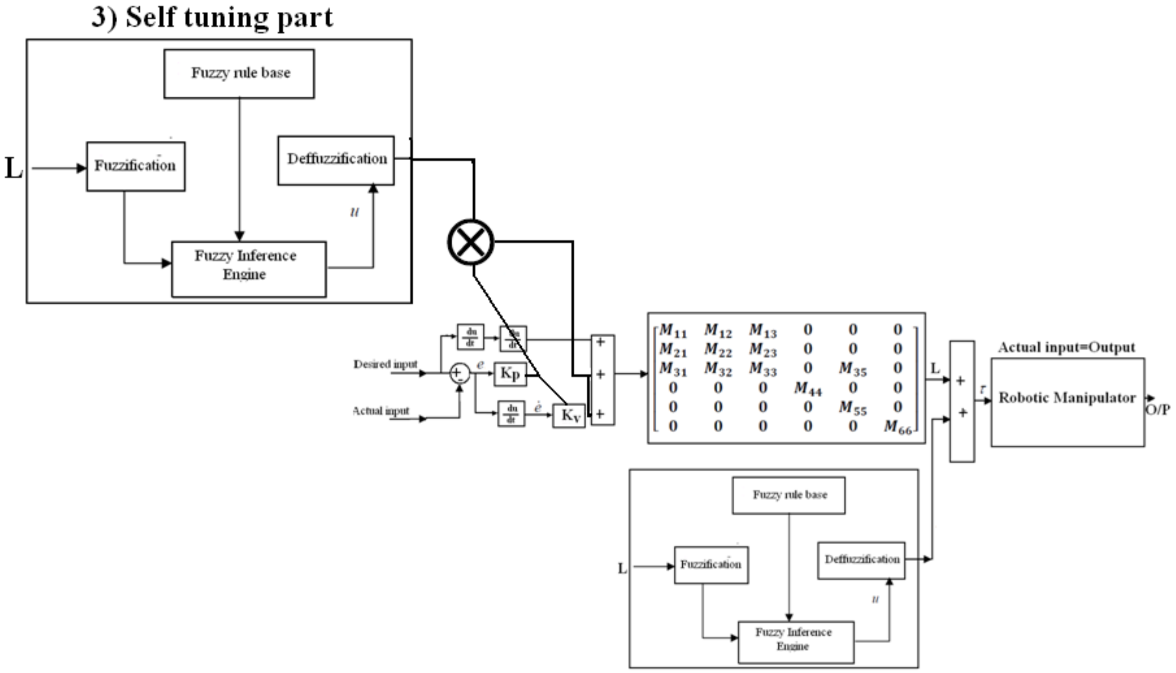


FIGURE 7: Block diagram of proposed gain tuning fuzzy computed torque like controller with minimum rule base in fuzzy nonlinear part and fuzzy supervisory.

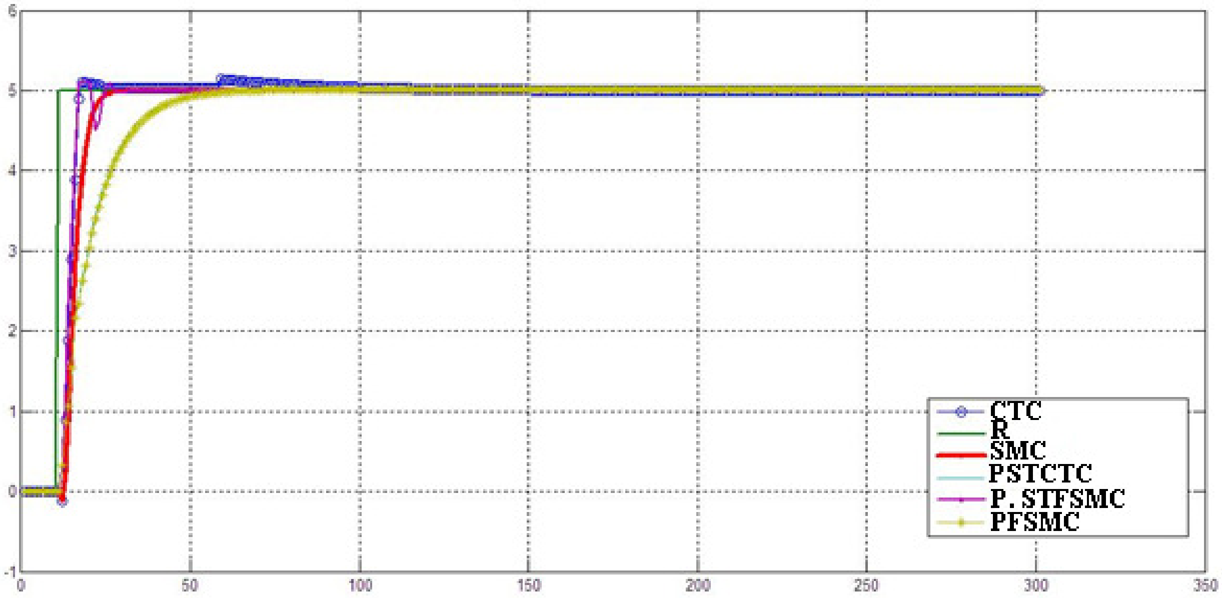


FIGURE 8 : first link step trajectory without disterbance

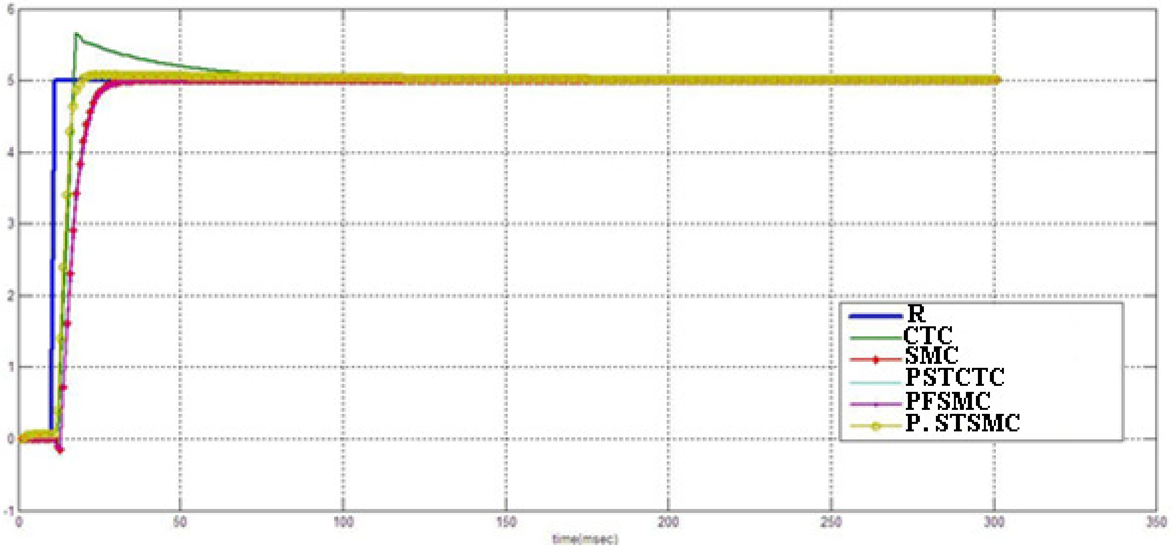


FIGURE 9: second link step trajectory without disterbance

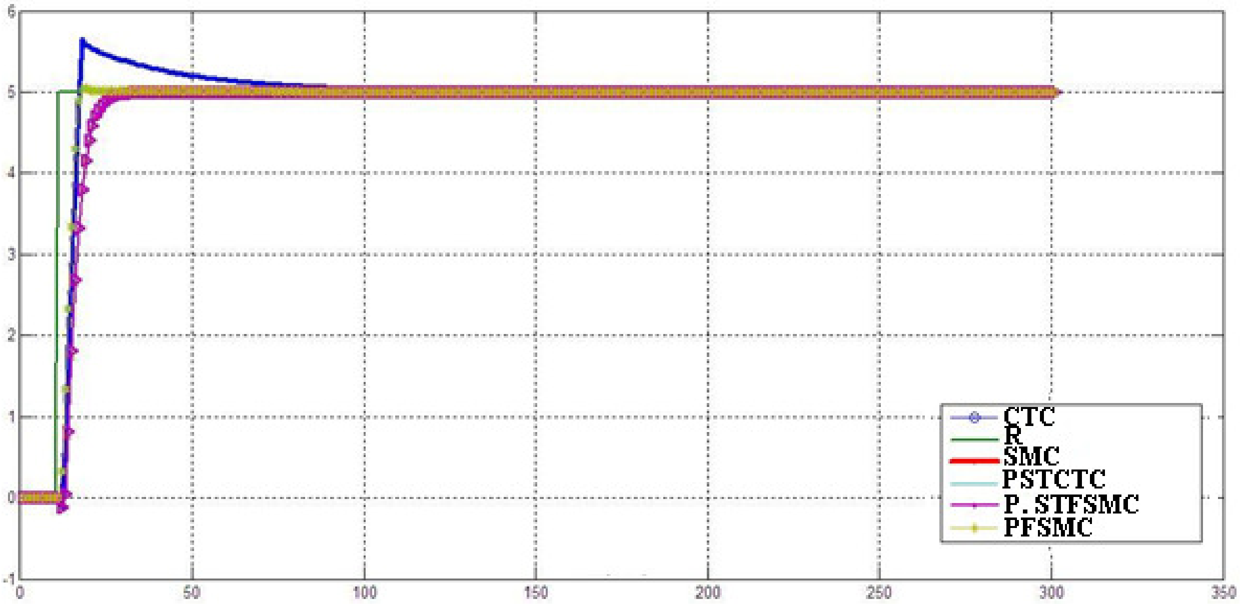


FIGURE 10 : Third link step trajectory without disterbance

Disturbance Rejection

Figure 11, 12 and 13 have shown the power disturbance elimination in above controllers. The main targets in these controllers are disturbance rejection as well as the other responses. A band limited white noise with predefined of 40% the power of input signal is applied to the step response. It found fairly fluctuations in trajectory responses. As mentioned earlier, CTC and SMC works very well when all parameters are known, this challenge plays important role to select the GTCTLC and GTFSMC as a based robust controller in this research.

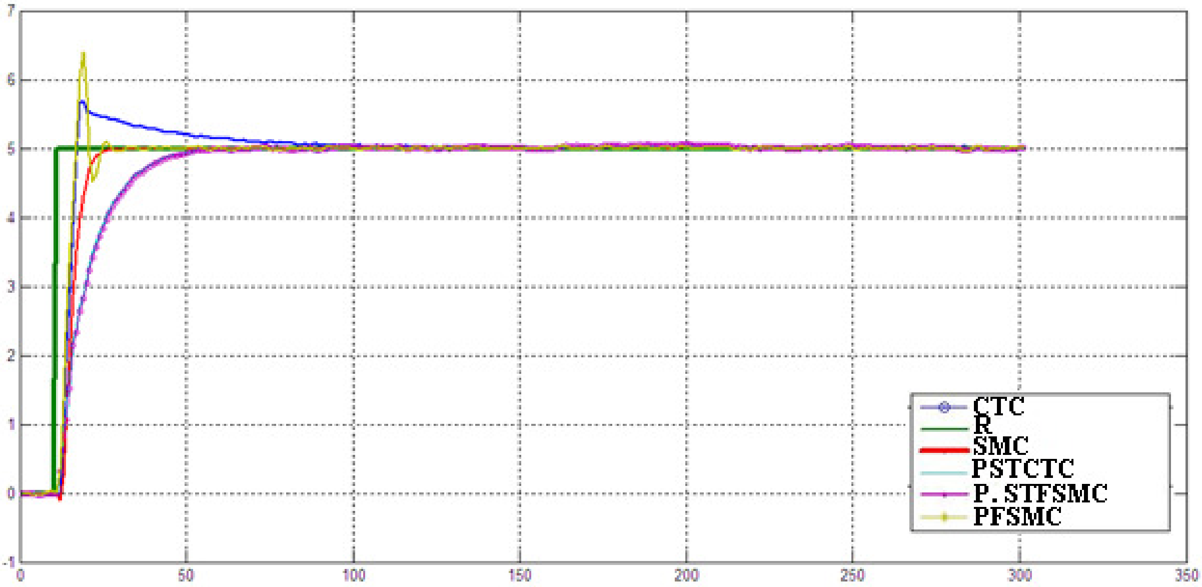


FIGURE 11 : First link step trajectory with disturbance

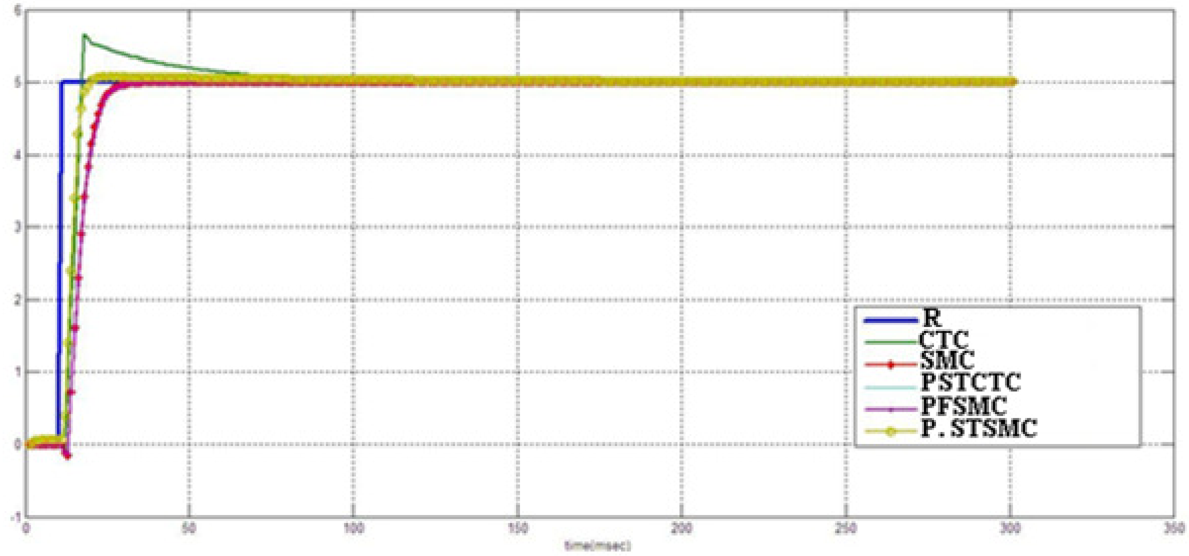


FIGURE 12: Second link step trajectory with disturbance

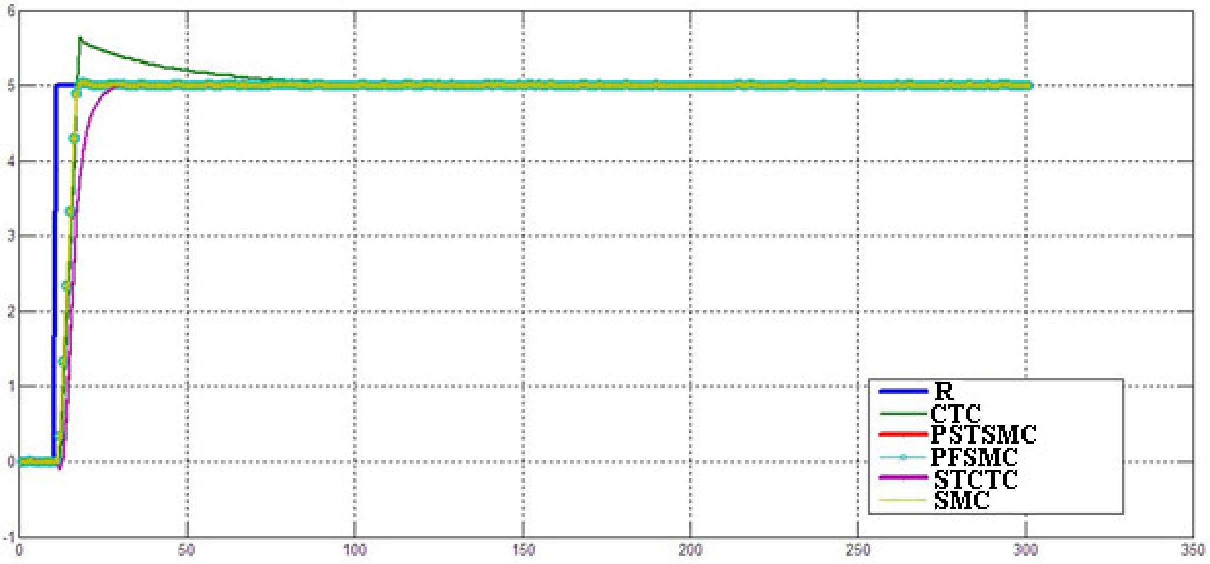


FIGURE 13: Third link step trajectory with disturbance

Errors in the Model: Figure 14 shows the tracking error for CTC, GTCTLC, SMC, FSMC, and GTFSMC. Equally, the proposed gain tuning computed torque like controller and proposed self tuning fuzzy sliding mode controller are more robust to changes of dynamic robot manipulator parameters value.

7. CONCLUSION

Refer to the research, a position artificial intelligence controller with tunable gain (GTCTLC and GTFSMC) design and application to robot manipulator has proposed in order to design high performance nonlinear controller in the presence of uncertainties. Regarding to the positive points in computed torque controller, sliding mode controller, fuzzy logic controller and tunable method, the performance has improved. Each method by adding to the previous controller has covered negative points. The system performance in computed torque controller, computed torque like controller, sliding mode controller and fuzzy sliding mode controller are sensitive to the gain updating factor. Therefore, compute the optimum value of gain updating factor for a system is the important challenge work. This problem has solved by adjusting gain updating factor of GTCTLC and GTFSMC. In this way, the overall system performance has improved with respect to the classical sliding mode controller and computed torque controller. This method solved chattering phenomenon as well as mathematical nonlinear equivalent part by applied fuzzy supervisory method.

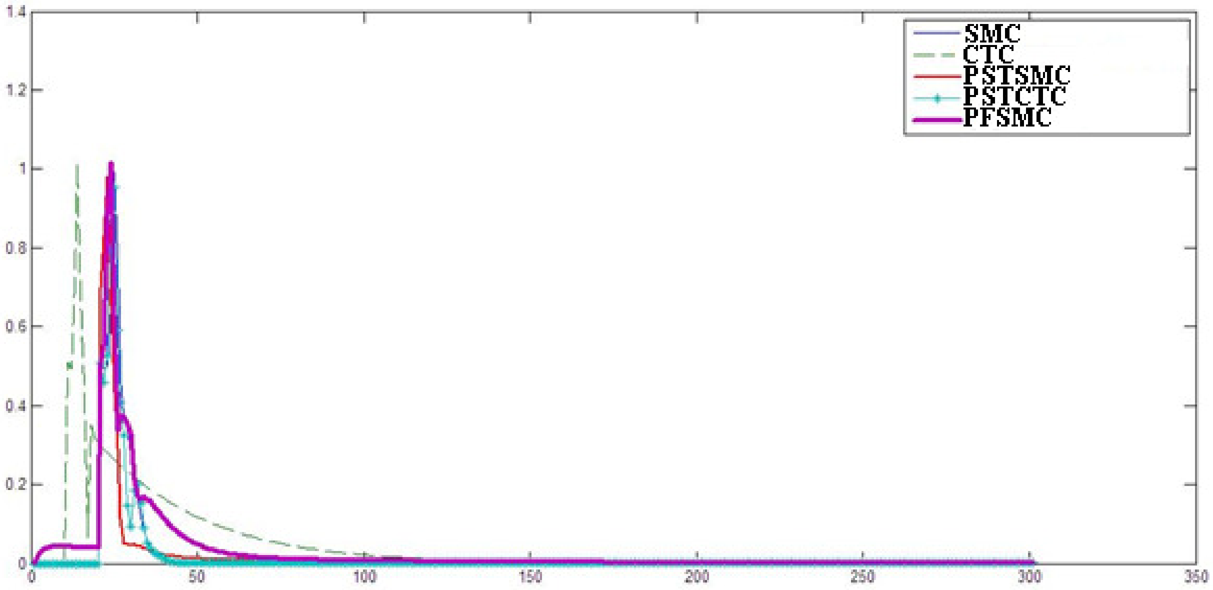


FIGURE 14: Errors in model

REFERENCES

[1] T. R. Kurfess, *Robotics and automation handbook*: CRC, 2005.

[2] K . Ogata, *Modern control engineering*: Prentice Hall, 2009.

[3] B. Siciliano and O. Khatib, *Springer handbook of robotics*: Springer-Verlag New York Inc, 2008.

[4] L. Reznik, *Fuzzy controllers*: Butterworth-Heinemann, 1997.

[5] S. Mohan and S. Bhanot, "Comparative study of some adaptive fuzzy algorithms for manipulator control," *International Journal of Computational Intelligence,* vol. 3, pp. 303–311, 2006.

[6] O. Kaynak, "Guest editorial special section on computationally intelligent methodologies and sliding-mode control," *IEEE Transactions on Industrial Electronics,* vol. 48, pp. 2-3, 2001.

[7] Shahnazi R., H. Shanechi, N. Pariz. "Position control of induction and servomotors: A novel adaptive fuzzy PI sliding mode control". IEEE Conferences on power engineering, 2006, P.P. 1-9.

[8] Medhaffar H., N. Derbel, and T. Damak. "A decoupled fuzzy indirect adaptive sliding mode controller with application to robot manipulator". Int. Journal on modeling, identification and control, 1(1), 23-29, 2006.

[9] Shahnazi R., H. Shanechi, N. Pariz. "Position control of induction and servomotors: A novel adaptive fuzzy PI sliding mode control". IEEE Journals on energy conversions, 23(1), 138-147, 2008.

[10] Weng C. C., W. S. Yu. "Adaptive fuzzy sliding mode control for linear time-varying uncertain systems". IEEE conference on fuzzy systems, 2008, P.P: 1483-1490.

[11] Yu Z. X. "Adaptive sliding mode-like fuzzy logic control for nonlinear systems". Journal of communication and computer, 6(1), 53-60, 2009.

[12] Piltan, F., et al. "Design sliding mode controller for robot manipulator with artificial tunable gain". Canaidian Journal of pure and applied science, 5 (2), 1573-1579, 2011.

[13] B. S. R. Armstrong, "Dynamics for robot control: friction modeling and ensuring excitation during parameter identification," 1988.

[14] B. Armstrong*, et al.*, "The explicit dynamic model and inertial parameters of the PUMA 560 arm," 2002, pp. 510-518.

[15] P. I. Corke and B. Armstrong-Helouvry, "A search for consensus among model parameters reported for the PUMA 560 robot," 2002, pp. 1608-1613.

[16] V. Utkin, "Variable structure systems with sliding modes," *Automatic Control, IEEE Transactions on,* vol. 22, pp. 212-222, 2002.

[17] R. A. DeCarlo*, et al.*, "Variable structure control of nonlinear multivariable systems: a tutorial," *Proceedings of the IEEE,* vol. 76, pp. 212-232, 2002.

[18] K. D. Young*, et al.*, "A control engineer's guide to sliding mode control," 2002, pp. 1-14.

[19] C. C. Weng and W. S. Yu, "Adaptive fuzzy sliding mode control for linear time-varying uncertain systems," 2008, pp. 1483-1490.

[20] M. Ertugrul and O. Kaynak, "Neuro sliding mode control of robotic manipulators," *Mechatronics,* vol. 10, pp. 239-263, 2000.

[21] P. Kachroo and M. Tomizuka, "Chattering reduction and error convergence in the sliding-mode control of a class of nonlinear systems," *Automatic Control, IEEE Transactions on,* vol. 41, pp. 1063-1068, 2002.

[22] Y. Li and Q. Xu, "Adaptive Sliding Mode Control With Perturbation Estimation and PID Sliding Surface for Motion Tracking of a Piezo-Driven Micromanipulator," *Control Systems Technology, IEEE Transactions on,* vol. 18, pp. 798-810, 2010.

[23] B. Wu*, et al.*, "An integral variable structure controller with fuzzy tuning design for electro-hydraulic driving Stewart platform," 2006, pp. 5-945.

[24] L. A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets and Systems,* vol. 90, pp. 111-127, 1997.

[25] J. Zhou and P. Coiffet, "Fuzzy control of robots," 2002, pp. 1357-1364.

[26] S. Banerjee and P. Y. Woo, "Fuzzy logic control of robot manipulator," 2002, pp. 87-88.

[27] K. Kumbla*, et al.*, "Soft computing for autonomous robotic systems," *Computers and Electrical Engineering,* vol. 26, pp. 5-32, 2000.

[28] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller. I," *IEEE Transactions on systems, man and cybernetics,* vol. 20, pp. 404-418, 1990.

[29] R. J. Wai*, et al.*, "Implementation of artificial intelligent control in single-link flexible robot arm," 2003, pp. 1270-1275.

[30] R. J. Wai and M. C. Lee, "Intelligent optimal control of single-link flexible robot arm," *Industrial Electronics, IEEE Transactions on,* vol. 51, pp. 201-220, 2004.

[31] M. B. Menhaj and M. Rouhani, "A novel neuro-based model reference adaptive control for a two link robot arm," 2002, pp. 47-52.

[32] D. Nguyen-Tuong*, et al.*, "Computed torque control with nonparametric regression models," 2008,

[33] A. Vivas and V. Mosquera, "Predictive functional control of a PUMA robot," 2005.

[34] Piltan. F. *et al.,* "Design Artificial of Robust Control of second Order System Based on Adaptive Fuzzy Gain Scheduling" World Applied Science Journal, 13(5), 1085-1092, 2011.

[35] Piltan. F. *et al.,* " Artificial Control of second Order System Based on AFGSMC" Austrailian Journal of Basic and Applied Science, 5(6), 509-522, 2011.

A Flexible Closed Loop PMDC Motor Speed Control System for Precise Positioning

Shakil Seeraji *shakil.ae@mist.edu.bd*
Department of Aeronautical Engineering
Military Institute of Science and Technology
Mirpur Cantonment, Dhaka-1216, Bangladesh

Enaiyat Ghani Ovy *enaiyat_ovy@yahoo.com*
Department of Mechanical and Chemical Engineering
Islamic University of Technology
Board Bazar, Gazipur-1704, BANGLADESH

Tasnim Alam *rim_mist@yahoo.com*
Department of Electrical electronic and communication Engineering
Military Institute of Science and Technology
Mirpur Cantonment, Dhaka-1216, Bangladesh

Ahsan Zamee *zamee_mist@yahoo.com*
Department of Electrical electronic and communication Engineering
Military Institute of Science and Technology
Mirpur Cantonment, Dhaka-1216, Bangladesh

Abdur Rahman Al Emon *emon1535@gmail.com*
Department of Electrical electronic and communication Engineering
Military Institute of Science and Technology
Mirpur Cantonment, Dhaka-1216, Bangladesh

Abstract

The speed control of DC motor is very significant especially in applications where precision is of great importance. Due to its ease of controllability the DC motor is used in many industrial applications requiring variable speed and load characteristics. So the precise speed control of a DC motor is very crucial in industry. In this case microcontrollers play a vital role for the flexible control of DC motors. This current research work investigates the implementation of an ATmega8L microcontroller for the speed control of DC motor fed by a DC chopper. The chopper is driven by a high frequency PWM signal. Controlling the PWM duty cycle is equivalent to controlling the motor terminal voltage, which in turn adjusts directly the motor speed. H-bridge circuit is implemented for the bi-directional control of the motor. A prototype of permanent magnet DC motor speed control system using the microcontroller is developed and tested.

Keywords: ATmega8L Microcontroller, DC Chopper, PWM, Tachogenerator, PMDC Motor, H-Bridge.

1. INTRODUCTION

DC motor transforms electrical energy into mechanical energy. They are used to drive devices such as fans, cars, door locks, seat adjust, mirror adjust, anti-lock braking system window lifts, robot arms, hoists etc. So the speed control of a DC motor is essential in this present industrial world. Many significant researches have been found regarding control techniques of the DC motors.

From earlier research work, some techniques to control the DC motors are notable to be mentioned. Refai [1] designed a PID controller which was based on MC68B00 microprocessor to control the speed of a DC motor. E^2PROM was used for program storage rather than RAM or ROM to improve the flexibility and avoid memory corruption or power interrupt. Abdelhay and Haque [2] applied a minimum-variance self-tuner to control the speed of a DC motor. By implementing this method satisfactory motor response was found and set point and load disturbances were sustained which was further compared with a PID controller. Microcontroller has been introduced later to regulate the speed of the DC motor. Ume et al. [3] used Motorola MC68HC11 microcontroller to control the speed of a permanent magnet DC motor.

From recent literature survey, the use of microcontroller for the control of DC motor has been found in many research works. Now-a-days, microcontrollers are used in the industrial world to control many types of equipments ranging from consumer to specialized devices. Chattopadhyay et al. [4] designed a microcontroller based position control system where the actuator was operated by the signal obtained from the PC (through key board). In this work, the position of the motor was controlled by the microcontroller based PI controller with interactive display control facilities. Adaptive fuzzy and neural speed controllers have been designed and implemented very recently to control the speed of a DC motor [5-8]. In this present work, the design and implementation of an ATmega8L microcontroller based controller to control a permanent magnet DC motor with speed feedback through a techogenerator is discussed elaborately. The actuator is regulated by a microcontroller based adjustable closed loop controller that controls the speed of a DC motor by using PWM and DC chopper. The system is interfaced to a LCD display so that the state of the system can be monitored by an operator. An H-bridge is implemented in the circuit for controlling the rotation of the motor in clockwise as well as anti-clockwise direction simultaneously. AVR studio4 software is used for programming and PonyProg is used for downloading the program to the microcontroller through parallel port.

2. SYSTEM DESCRIPTION REPEATABILITY

The block diagram of microcontroller based closed loop speed control of DC motor is shown in the figure 1. The motor to be controlled is fed by a DC source through a chopper. The techogenerator senses the speed, which gives voltage as output. And this voltage is fed to the microcontroller to drive the speed of the motor.

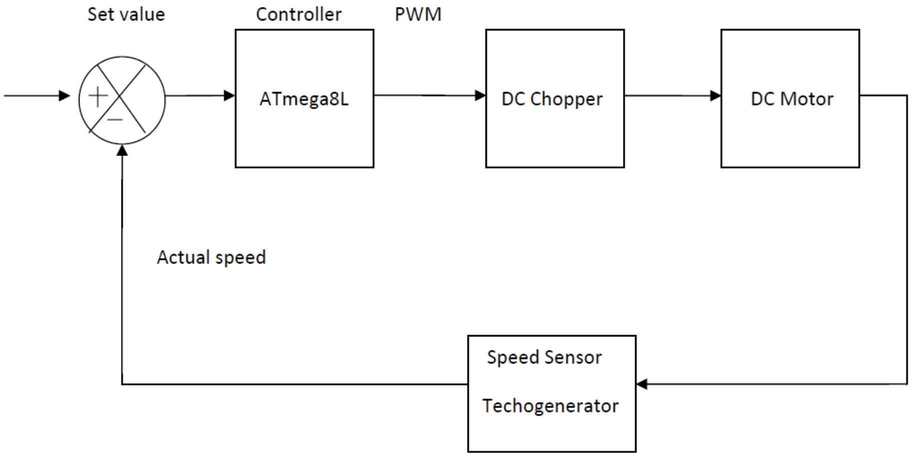


FIGURE 1: Block diagram of microcontroller based closed loop speed control of DC motor.

The output voltage of techogenerator is provided to the microcontroller and microcontroller determines the output voltage of the chopper fed to the DC motor for desired speed.

3. Circuit Hardware and Software Operation

Circuit hardware and software portions are discussed in the following sections.

3.1 Circuit Schematic Diagram

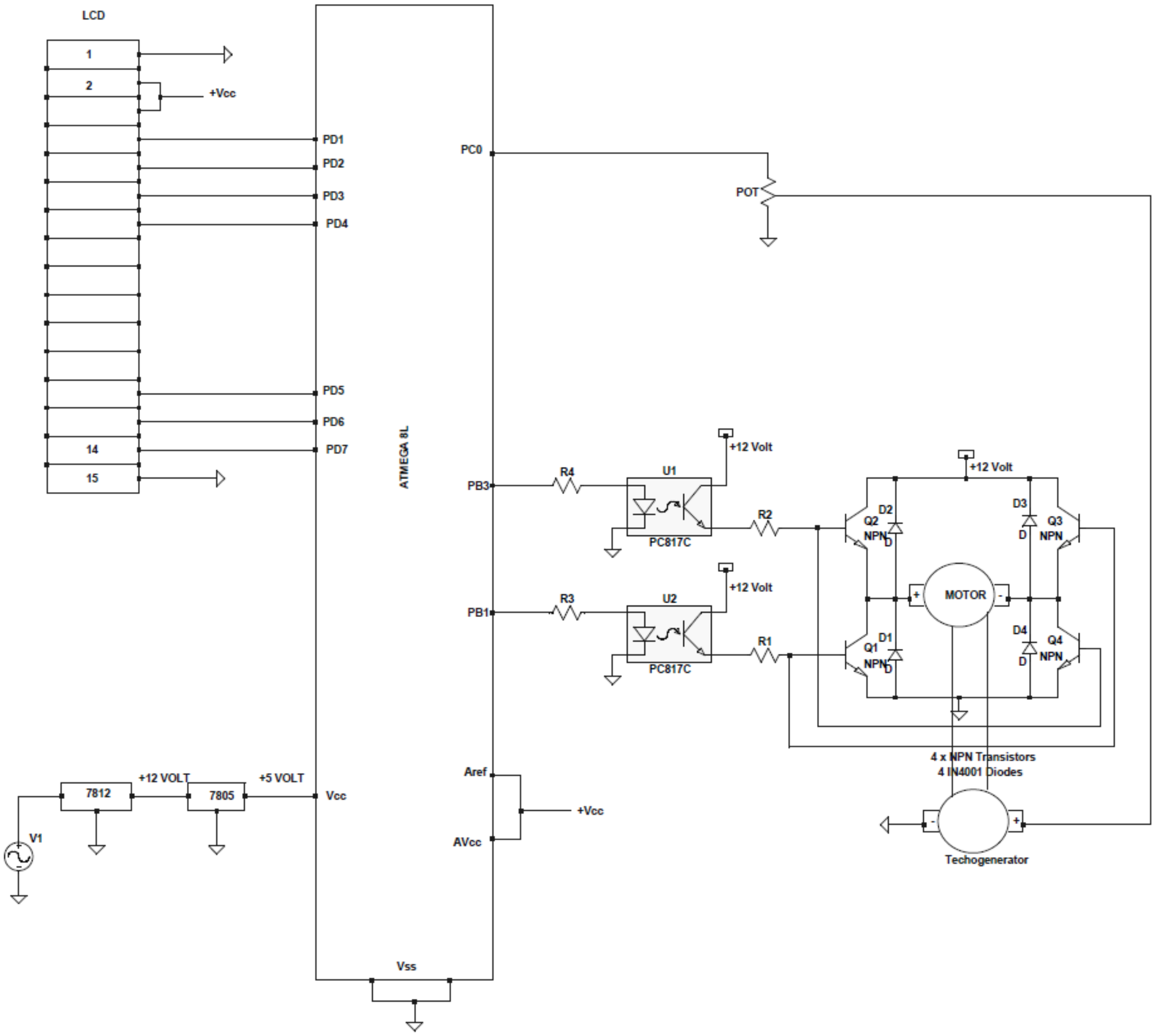


FIGURE 2: Circuit for the speed control of DC motor.

The operation of the control circuit is described below by its different portion.

3.1.1 Pulse Width Modulator Module (PWMM)

PWMM is a very efficient way of providing intermediate amounts of electrical power between fully on and fully off. A simple power switch with a typical power source provides full power only, when switched on. PWM is a comparatively-recent technique, made practical by modern electronic power switches. In this research work, timer/counter2 (8-bit) was used to generate PWM for varying the speed of DC motor (12V). Phase correct mode was used here. It has 2 different modes of operation. Non-inverted mode was selected for this work.

3.1.2 Chopper Circuit

A DC chopper is a dc-to-dc voltage converter. It is a static switching electrical appliance that in one electrical conversion, changes an input fixed dc voltage to an adjustable dc output voltage without inductive or capacitive intermediate energy storage. The name chopper is connected with the fact that the output voltage is a ‘chopped up’ quasi-rectangular version of the input dc voltage. For this research work, a Buck converter (step down chopper) was implemented.

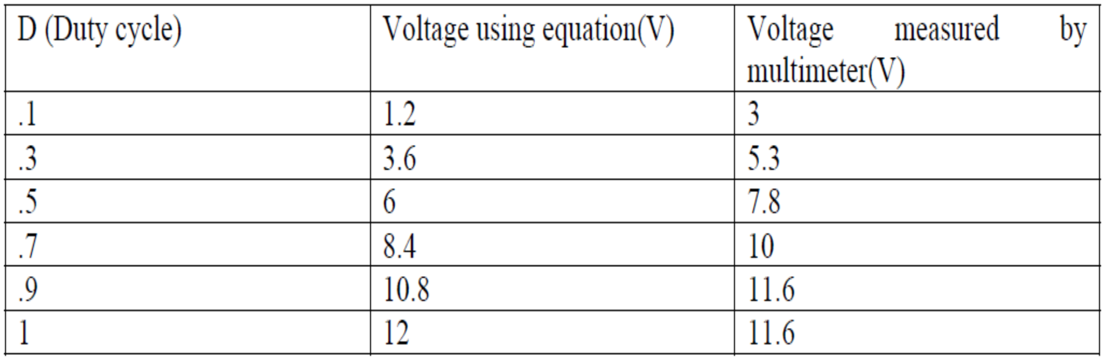
A pulse with fixed frequency is generated by the microcontroller, which is fed to the base of transistor (D400). Transistor acts here as a switch. The output voltage of the motor is dependent on the amount of the on time of the transistor. The more time transistor remains on more the voltage will produce. A Freewheeling diode (1n4001) is used for back e.m.f. protection given to other portion of the circuit. Output voltage of the motor terminal can be shown by the equation given below.

$$V_{out} = V_{in} \times D$$

Where $D = \dfrac{T_{on}}{T_{on} + T_{off}}$

And $V_{in} = 12V$

Using the equation and by measuring the voltage by multimeter the following values are obtained.



D (Duty cycle)	Voltage using equation(V)	Voltage measured by multimeter(V)
.1	1.2	3
.3	3.6	5.3
.5	6	7.8
.7	8.4	10
.9	10.8	11.6
1	12	11.6

TABLE 1: Motor terminal voltages at various duty cycles.

3.1.3 Sensor Design

For speed sensing purpose, another motor is used. This motor is used here as a tachogenerator. As it is known that for a DC motor voltage is directly proportional to the speed. The tachogenerator is coupled with the motor. And a potentiometer is connected to the terminal of the tachogenerator. Tachogenerator gives voltage drop across the potentiometer according to the speed of the motor. If the motor runs at a low speed it gives a lower value. And when it runs at its maximum speed it gives a larger amount of voltage. As it is known that speed can be regulated by regulating the pulse width, so varying the duty cycle the speed can be regulated. Output voltages at different duty cycles have been found by varying the duty cycle controller register OCR (output compare register).

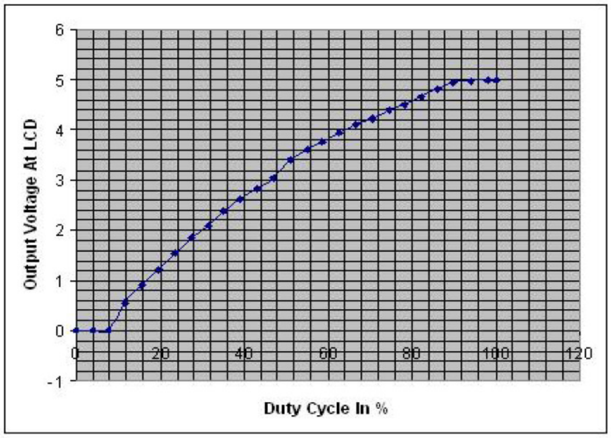


FIGURE 3: Duty cycle vs Output voltage.

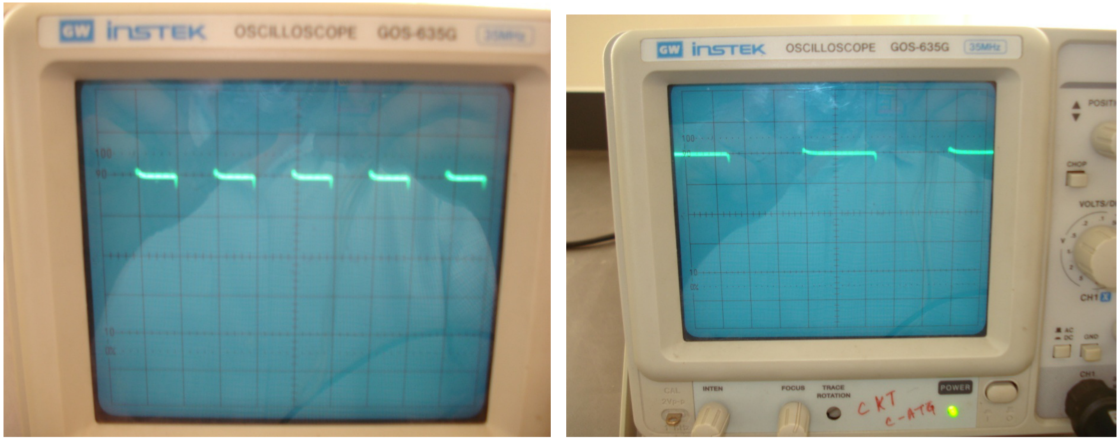

GW instek OSCILLOSCOPE GOS-635G 35MHz
GW instek OSCILLOSCOPE GOS-635G 35MHz
INTEN
FOCUS
TRACE ROTATION
POWER
CKT
c-ATG

FIGURE 4: Pulses at 20% and 50% duty cycle.

Now the voltage drop across the potentiometer is fed to ADC of the microcontroller. According to the ADC value, microcontroller will take decision whether pulse width needs increment or decrement.

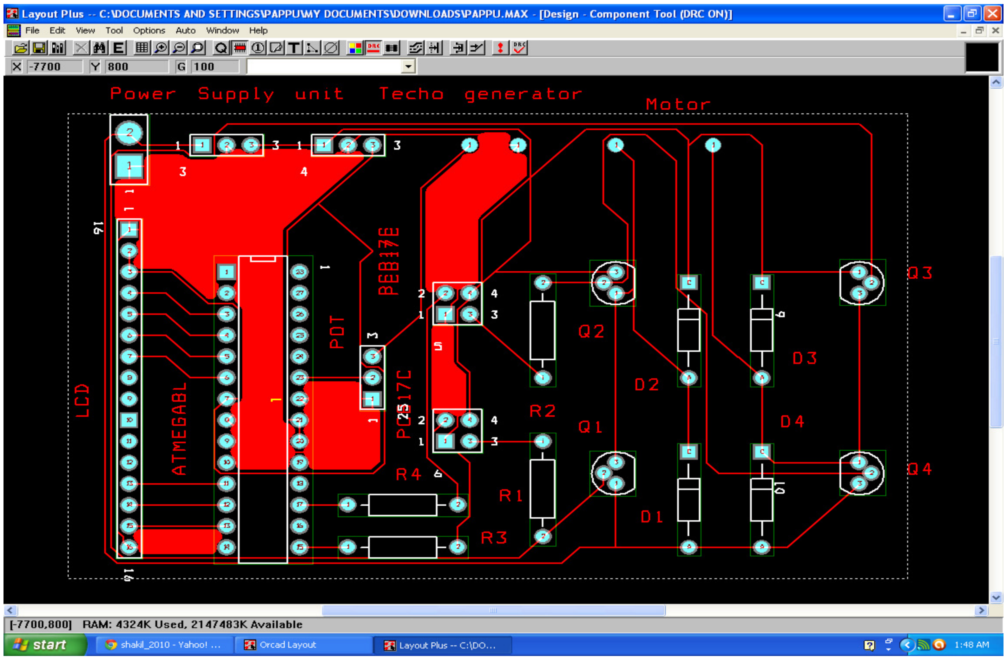
3.1.4 Circuit Logic Development

As this present work is based on the speed controlling of a DC motor, so in this work the desired goal is to achieve a system with constant speed at any load condition. That means the motor will run at fixed speed at any load condition. As it is known that the speed of a DC motor can be varied by PWM technique so according to the value of duty cycle the motor speed can be varied. Now a question arises how it can be measured the variation of speed of the motor? To do this another DC motor is used, which is coupled with the main motor. When the motor will run, it will also make the second motor starts rotating. And the motor will act as a speed to voltage transducer or tachogenerator. That means it will give an output voltage according to speed. By measuring the output voltage drop of the tachogenerator the speed can be measured easily. It is mentioned earlier in the paper that the desire is to maintain constant speed at any load condition so initially PWM variation registers are set at a fixed value. As a result a fixed output voltage at the tachogenerator end will be found. Speed of the motor does not remain same all the time because of the various external forces like air or defect in the motor coupler. Though it does not vary by huge value so the output voltage at the microcontroller is set not by a single value rather it is set by giving a range. For bi-directional controlling of the motor, an H-bridge circuit is used. In the H-bridge circuit four NPN transistors are used as switch to change or choose the direction of current flows to the motor. Opto-coupler is used between the motor and microcontroller for isolation.

Now if a load is applied on the motor, the speed of the motor will suddenly decrease. And with the decrement of the speed output voltage will also decrease. This output voltage is fed to the ADC input of the microcontroller by using a potentiometer. The output voltage of the tachogenerator is scaled to 5V as the microcontroller ADC inputs are 5V supported. As earlier a range of voltages are set for the fixed duty cycle, so when the new value of voltage will be sensed by the microcontroller it will also sense the decrement of the speed by comparing two values. Now the controller unit will tend to improve the speed of the system, so that the output voltage remains same of the tachogenerator. To improve the system speed microcontroller now will start increasing the value of PWM controller register i.e. output compare register (OCR) until the input voltage of the ADC reaches the desired level of voltage. Now after reaching the desired level of voltage, OCR will stop further increment. Now two more condition arises here, that what would happen if the sudden load drops down to very low amount and what would happen if the load is very huge that motor cannot run at its desired speed? When the load will drop down to a low value, the speed of the motor will be very high. As a result output voltage will be also very high. So again controller unit will sense output voltage and will compare with the desired level of voltage. The PWM controller register i.e. OCR will do the reverse operation. Now the value OCR will decrease until the output voltage reaches its desired level. And for the second condition if the load amount is so high that motor cannot run at its desired speed, then OCR will start increasing until reaches its maximum value. For 8 bit timer/counter the maximum value of OCR is 255. Even if after reaching the maximum value, there remains no improvement of the speed, i.e. output voltage does not matches the desired level then microcontroller will send a message “OVERLOAD” using the LCD (16×2 line), so that the user can understand the condition and hence reduce the load of the motor.

3.1.5 PCB (Printed Circuit Board) Design

A printed circuit board, or PCB, is used to mechanically support and electrically connect electronic components using conductive pathways, tracks, or traces, etched from copper sheets laminated onto a non-conductive substrate. ORCAD family release 9.2(layout plus) was used for control circuit design. Designed PCB figure is given below.


Layout Plus -- C:\DOCUMENTS AND SETTINGS\PAPPU\MY DOCUMENTS\DOWNLOADS\PAPPU.MAX - [Design - Component Tool (DRC ON)]

X -7700 Y 800 G 100

[-7700,800] RAM: 4324K Used, 2147483K Available

FIGURE 5: PCB layout of the control circuit.

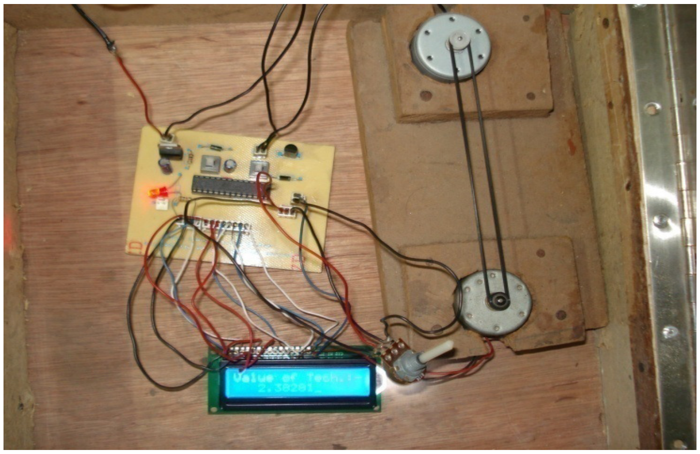


FIGURE 6: The physical setup of the research work.

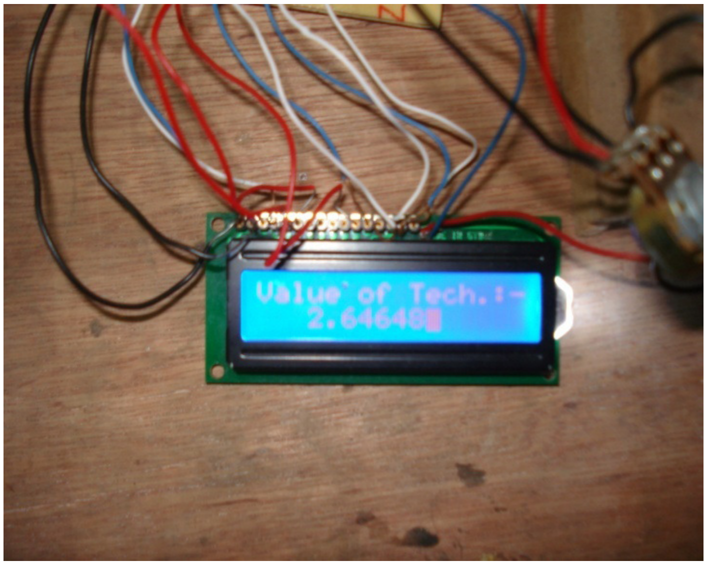


FIGURE 7: Showing value of output voltage of techogenerator.

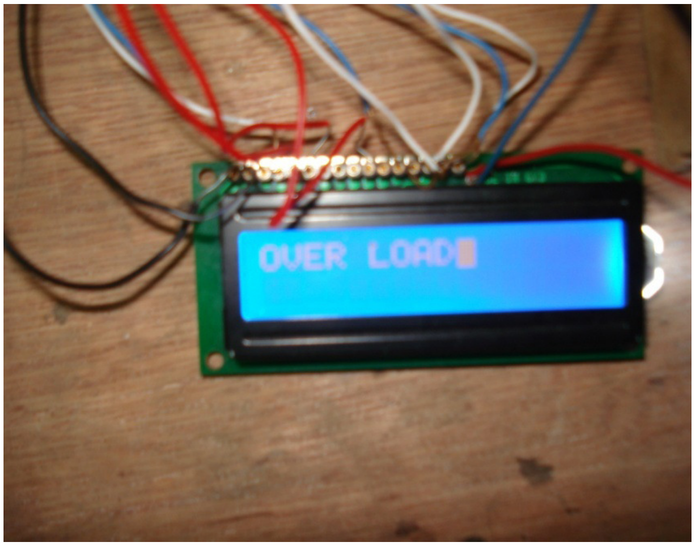


Figure 8: Showing “OVERLOAD” message.

4. OPERATIONAL FLOW CHART

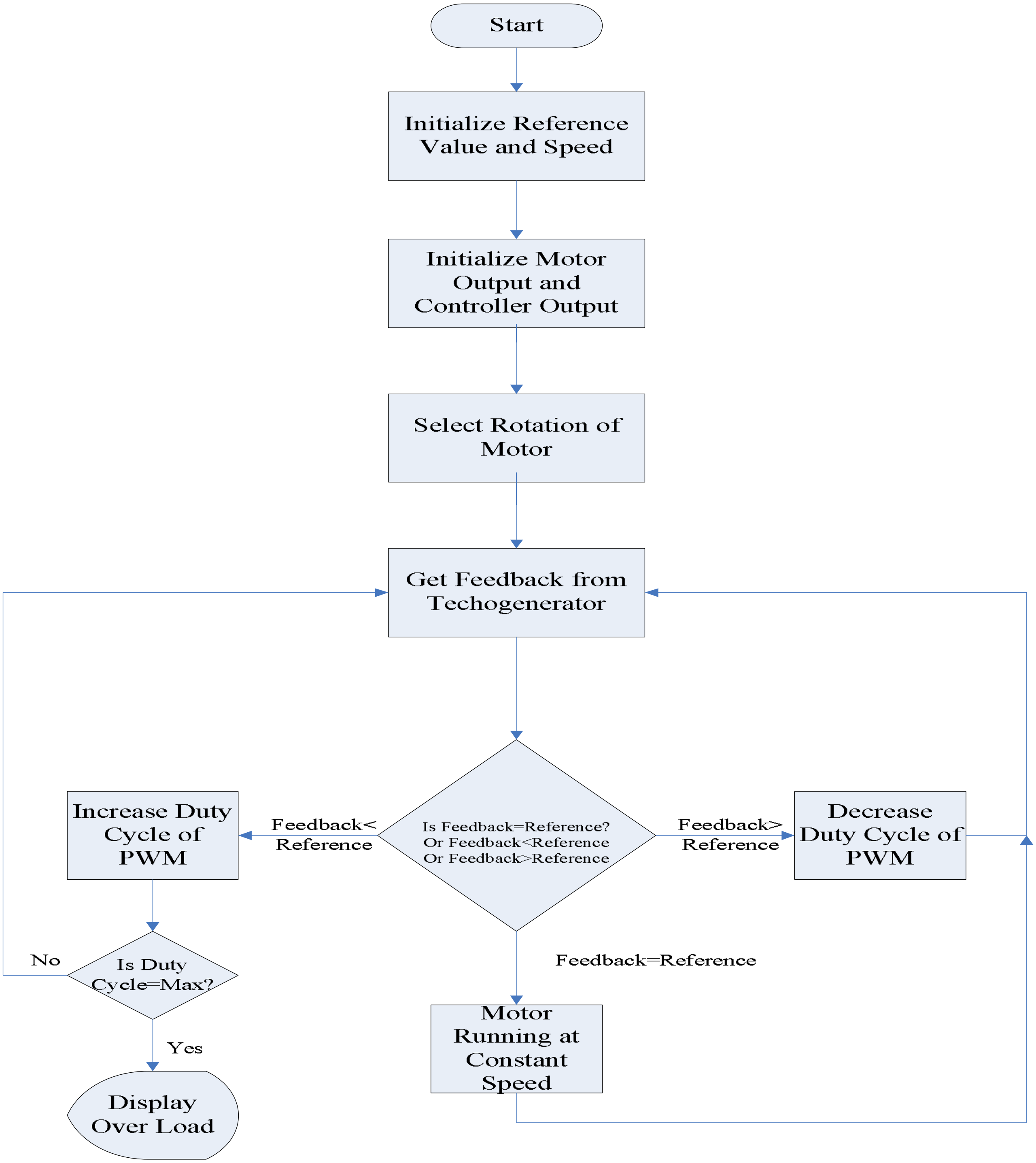

Start
Initialize Reference Value and Speed
Initialize Motor Output and Controller Output
Select Rotation of Motor
Get Feedback from Techogenerator
Is Feedback=Reference? Or Feedback<Reference Or Feedback>Reference
Feedback< Reference
Feedback> Reference
Increase Duty Cycle of PWM
Decrease Duty Cycle of PWM
Feedback=Reference
No
Is Duty Cycle=Max?
Yes
Motor Running at Constant Speed
Display Over Load

FIGURE 9: Operational flow chart for DC motor speed controller.

5. CONCLUSION

An embedded system is designed and implemented in this work. Controlling a permanent magnet DC motor with speed feedback through a techogenerator is successfully implemented using an ATmega8L microcontroller. Microcontroller provides very less requirement of hardware. The system is made user friendly so that anybody can operate the system without any trouble. LCD display is used to show the condition of the system. After knowing the condition the amount of load can be changed if necessary. Finally it can be concluded here that the system reliability is higher where the PMDC motor can be regulated easily as well as the maintenance of the motor can be also improved.

6. REFERENCES

[1] MK Refai, Microprocessor-based digital controller for DC motor speed control, Microprocessors and Microsystems, Volume 10, Issue 10, December 1986, Pages 543-552.

[2] Sofyan A Abdelhay, M Azharul-haque, DC-motor control using a minimum-variance self-tuner, Microprocessing and Microprogramming, Volume 19, Issue 3, June 1987, Pages 227-231.

[3] Charles I. Ume, John Ward, Jay Amos, Application of MC68HC11 microcontroller for speed control of a DC motor, Journal of Microcomputer Applications, Volume 15, Issue 4, October 1992, Pages 373-385.

[4] Subrata CHATTOPADHYAY, Utpal CHAKRABORTY, Arindam BHAKTA and Sagarika PAL, Microcontroller Based Closed Loop PMDC Motor Position Control System, Sensors & Transducers Journal, Vol. 102, Issue 3, March 2009, pp. 62-70.

[5] M. D. Minkova, D. Minkov, J. L. Rodgerson, R. G. Harley, Adaptive neural speed controller of a dc motor, Electric Power Systems Research, Volume 47, Issue 2, 15 October 1998, Pages 123-132.

[6] Gerasimos G. Rigatos, Adaptive fuzzy control of DC motors using state and output feedback, Electric Power Systems Research, Volume 79, Issue 11, November 2009, Pages 1579-1592.

[7] Karim H. Youssef, Hasan A. Yousef, Omar A. Sebakhy, Manal A. Wahba, Adaptive fuzzy APSO based inverse tracking-controller with an application to DC motors, Expert Systems with Applications, Volume 36, Issue 2, Part 2, March 2009, Pages 3454-3458.

[8] Jui-Hong Horng, Neural adaptive tracking control of a DC motor, Information Sciences, Volume 118, Issues 1-4, September 1999, Pages 1-13.

HRI for Interactive Humanoid Head Amir-II for Visual Tracking and Servoing of Human Face

Aseef Iqbal aseef.iqbal@gmail.com
Department of Mechatronics Engineering
Faculty of Engineering
International Islamic University Malaysia
Off Jalan Gombak, Kuala Lumpur, 53100
Malaysia

Amir A Shafie aashafie@iium.edu.my
Department of Mechatronics Engineering
Faculty of Engineering
International Islamic University Malaysia
Off Jalan Gombak, Kuala Lumpur, 53100
Malaysia

Md Raisuddin Khan raisuddin@iium.edu.my
Department of Mechatronics Engineering
Faculty of Engineering
International Islamic University Malaysia
Off Jalan Gombak, Kuala Lumpur, 53100
Malaysia

M Farid Alias mfarid.alias@gmail.com
Department of Mechatronics Engineering
Faculty of Engineering
International Islamic University Malaysia
Off Jalan Gombak, Kuala Lumpur, 53100
Malaysia

Jamil Radhi en_radhi@hotmail.com
Department of Mechatronics Engineering
Faculty of Engineering
International Islamic University Malaysia
Off Jalan Gombak, Kuala Lumpur, 53100
Malaysia

Abstract

In this paper, we describe the HRI (Human-Robot Interaction) system developed to operate a humanoid robot head capable of visual tracking and servoing of human face through image processing. The robotic humanoid head named Amir-II, equipped with a camera and servoing mechanism is used as the platform. The Amir-II tracks the human face within the field-of-vision (FOV) while the servoing mechanism ensures the detected human face remains at the center of its FOV. The algorithm developed in this research utilizes the capability offered by scientific computing program MATLAB along with its Image Processing Toolbox. The algorithm basically compares the locations of the face in the image plane that is detected from the static face image captured from real-time video stream. The calculated difference is then used to produce appropriate motion command for the servo mechanism to keep track of the human face moving within the range of its FOV.

Keywords: Humanoid Head, Human-Robot Interaction (HRI), Emotional Expression, Face Detection, Visual Servoing, SMQT, Split-up SNoW Classifier, Matlab, Image Processing.

1. INTRODUCTION

As robots have been predicted to become part of our everyday life, there has been a significant number of active research in the area of Human-Robot Interaction (HRI) for socially interactive humanoid robots. Among the innovative methods proposed includes Michalowski [1] that shows a rhythmic movement technique to engage a robot with human for an effective interaction. Cynthia [2] and Rosalind [3] suggests that HRI for applications like socially interactive machines can be very effective if it can exchange emotional expressions with the human counterpart. To date, robots have been studied in a variety of therapeutic application domains, ranging from using robots as exercise partners, using robots in pediatrics, robots as pets for children and elderly people, and robots in autism therapy. Researchers have developed robots engaged in social interaction with human using various modes of communication. Robots such as Paro [4], Robota [5], Keepon [6], Infanoid [7], Kismet [8] have been used successfully to emotionally engage with human very effectively via speech, vision, touch etc. as the channel for interaction.

The most common way of expressing emotional state of a human is via facial expression augmented with verbal cues and physical gestures. Some of the significant works in analyzing facial expressions are presented in [9-11]. Robots like Buddy [12], Kobian [13] and Kismet are developed as research platforms capable of displaying emotions through facial expressions towards their human operators.

An important step towards developing an emotionally responsive and intelligent robot is the capability of using visual cue as an input and analyzing it when interacting with human operator. In this type of application, capability of detecting and tracking a human face from video sequences is necessary. But locating and tracking of human face from visual input is particularly challenging because human faces have a very high degree of variability in terms of pose, scale and significant facial features.

This paper discusses the techniques used in the humanoid head AMIR-II [14] for detecting and tracking human face as a part of its HRI design. The following section discusses on the evolution of the robotic head AMIR-II. In section 3, the development of the graphical user interface (GUI) to operate AMIR-II is discussed with brief details on its different functional modules. Section 4 elaborates on the techniques used for face detection that also includes the results of using the techniques adopted and improvements achieved. Servoing and tracking of the detected face is explained in section 5. In section 6, some experimental results are presented to display the capacity of AMIR-II with the present implementation. The concluding remarks in section 7 summarize the achievements and scopes of further improvements.

2. AMIR: THE HUMANOID HEAD

The first prototype of the robotic head, named AMIR-I [15, 16], had Basic Stamp 2 microcontroller at its heart. The controller was linked to 17 parallax servo motors connected with different parts of the mechanical structure. The aim of AMIR-I was to head-start into this emerging field of research and create a test bed for development and iterative improvement towards developing an interactive and facially expressive humanoid head. AMIR – I was capable of displaying only 3 basic emotions and valid head movements (pan-tilt) with its limited Degree-of-Freedom. AMIR-I had a PING))) ultrasonic sensor attached for identifying the presence of any operator in front and was only a platform to initiate the research on developing an effective human-robot interaction system.

AMIR-II in figure 1 is an improvement over previous prototype replacing all the electronics and some minor changes in mechanism inside AMIR-I. Amir-II is capable of producing 5 different facial expressions, i.e. neutral, happy, angry, sad and disgust (figure 2). The facial expressions conveyed through facial features as the mouth shape together with the positioning of the eyebrows and the eyelids. A major upgrade is inclusion of a vision system for visual feedback to the system. The mechanical structure of Amir-II consists of 9 degrees of freedom (DOFs), in which 2 DOFs are for its neck (pan-tilt), 3 DOFs for its mouth, 1 DOF for each eyelids and 1 DOF for each eyebrow. In Amir-II, the parallax servo motors are replaced with more capable Dynamixel AX-12+ smart servo motors from Robotis. These motors have very unique characteristics. The AX-12+ robot servo has the ability to track its speed, temperature, shaft position, voltage, and load. Features also include 300 degree of movements in 1024 increments, 1,000 kbps communication speed in half-duplex mode and a huge 16.5 kg-cm holding torque at 12V operating voltage etc. – all in roughly the same size of a standard servo. These servo motor are controlled by a PC with USB2DYNAMIXEL – an interface between the PC and AX-12+ via high-speed USB2.0 port. Matlab and Image Processing Toolbox 2.0 was used to develop the controller for the system. Use of Matlab made the system integration very easy and allowed us to concentrate more on implementing efficient algorithm and rapid development of Graphical User Interface (GUI) [9] for an effective HRI for AMIR-II.

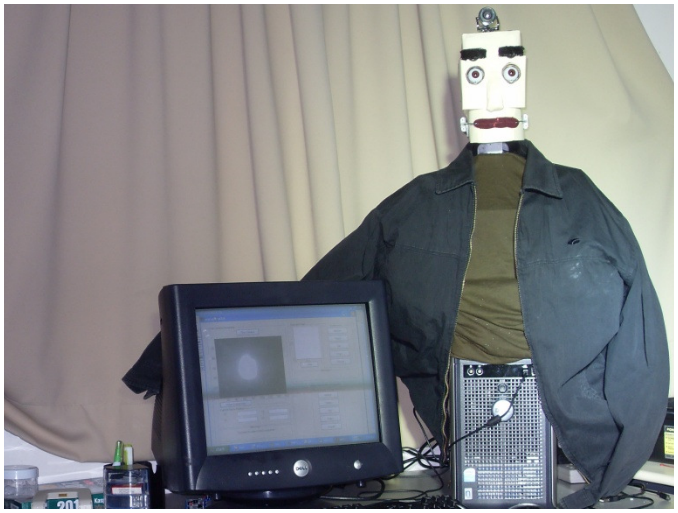


FIGURE 1: The robotic head AMIR-II

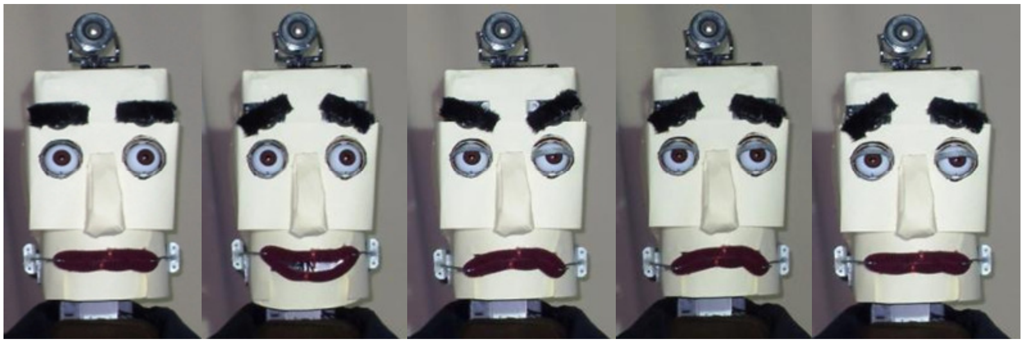


FIGURE 2: Five different facial expressions by Amir-II. From left to right: neutral, happy, angry, sad, disgusted.

3. INTERFACE DESIGN

Panel Design

The GUI program serves as the control and monitoring tool for the operation of Amir-II. In figure 3, the GUI contains 3 panels in which each panel carries its own functionality as follows:

1) *Live Camera Streaming Panel:* The Live Camera Streaming panel allows the user to monitor the live operation of the robot Amir-II operator can view the live image streaming from the camera by clicking the Start/Stop button within this GUI panel. The face detection module will automatically display a red bounding box as a human face is detected within the field of view (FOV) of the

camera. Meanwhile, the message box at the bottom of the panel will display the status of the USB2Dynamixel connection to the PC.

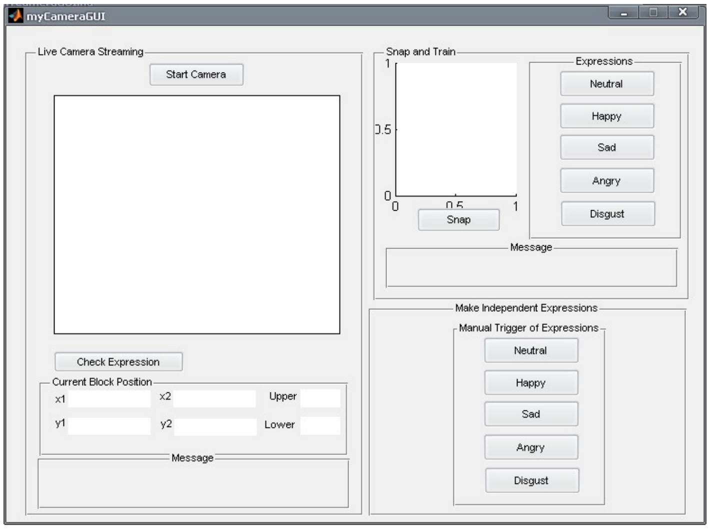


FIGURE 3: The graphical user interface for Amir-II

2) Snap Panel: The Snap panel provides the capturing, labeling and saving of the facial expression images into a database. The snap segment contains a snap button, a screenshot window of captured image and 5 buttons of the respective facial expressions. As the Snap button is clicked, the detected face area within the bounding box will be captured and displayed inside the screenshot window. After that the operator can click the related expression button to save the image and its expression label into the database.

3) Manual Trigger of Expressions: The Manual Trigger of Expression panel lets the operator to trigger the respective facial expressions for Amir-II. To demonstrate the capability of Amir-II to produce facial expression, its operator can select which expression is to be produced at a time by clicking the related expression button within this panel. Such manual trigger of expression is also useful in inspecting the condition of Amir-II mechanical structure.

For its back-end, the GUI program consists of several modules as shown in figure. 4. The functionality of each program module is described as follows:

1) Image Acquisition Module: In order to reduce the complexity of processing the image, the original RGB streaming image is converted to its grayscale format. The live streaming image is being displayed based on the default resolution of the camera used. With camera resolution set to its lowest acceptable value (at 120 pixels x 160 pixels), the image frame rate becomes near 30 frame per second.

2) Face Detection Module: The face detection module uses SNoW classifier algorithm [14] to detect a human face within the camera FOV. The live detected face area is marked with a bounding box on the image streaming window. High frame rate is crucial for the face to be kept tracked since it can consistently appear on the camera FOV in real time.

3) Face Tracking Module: Once the face is detected, the face tracking module requests the servo callback functions to trigger the servo movements. There is an acceptable boundary (AB) defined for a tracked face to move around within the visual input. As the tracked face moves beyond the AB, the servo callback function is called to send the related commands to the servos through USB2Dynamixel device [14]. This process involves the repositioning the neck servos through their pan-tilt movements to refocus the tracked face back within the AB. The location of the face within the image will determine the required direction and magnitude of rotation for the neck servos.

4) *Face Image Capture Module:* This module is the one functioning for the Snap Panel. The database directory and the method of labeling the expression image data are determined here.

5) *Facial Expression Recognition Module:* To generate the proper facial expression by Amir-II, the human facial expression within the camera FOV is first recognized. This information is needed in the decision-making process for human-robot interaction, to produce appropriate expression in response, based on the chosen behavior model.

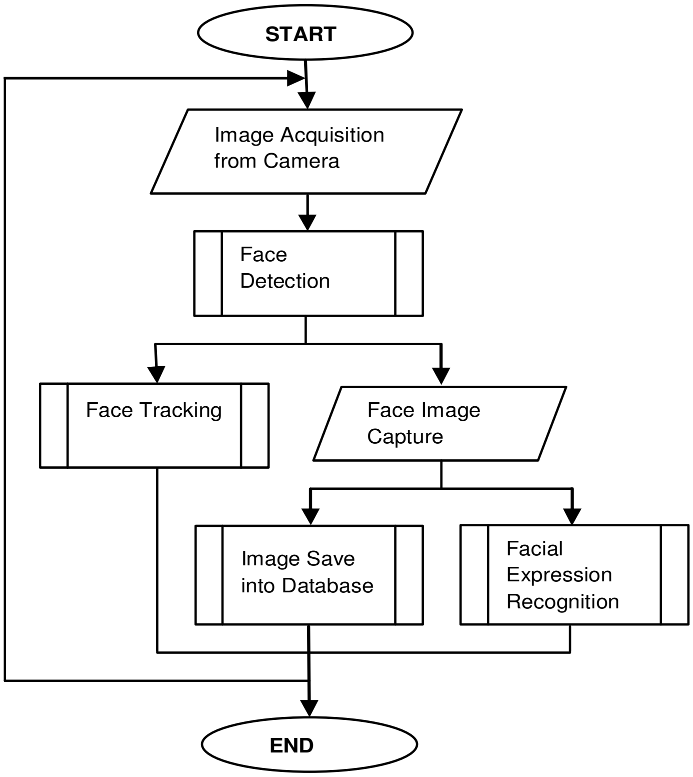


FIGURE 4: Flowchart of the interfacing program for AMIR-II

4. FACE DETECTION METHOD FOR AMIR-II

The foremost task of the system in terms of functionality is to identify a human face from the scene that is *face detection* or *face segmentation*. Functionality of face tracking and, as a whole, the success of entire system depends on the accuracy of the human face detection.

Initial face segmentation techniques were only capable of detecting single frontal-face from image with simple uncluttered background using neural network, template-matching or skin color properties [17]. Recent advancements in technology allowed researchers to attempt more computing-intensive techniques such as appearance-based or optical methods to increase the detection rate. Some of the established face-detection systems are eignefaces [18], which implements Principal Component Analysis (PCA), Fisherfaces using Linear Discriminant Analysis [19], Bayesian method using probabilistic distance metric [20], etc. These techniques proved to be efficient in segmenting multiple human faces even with partial occlusion and complex, cluttered background images.

Segmentation of moving faces from a video sequence requires a different approach. One method is detecting face in single frame of the video with any of the techniques applied on static image. Subsequent video frames are then compared using pixel-based change detection procedures based on difference images. More recent methods use optical-flow techniques for detecting human face from the video. They extract the color information from the video frames to identify possible location of human face in the image-plane. These methods can also be applied for segmenting multiple faces.

Face Detection in AMIR-II using skin color

In AMIR-II, we initially used skin color to find the face from an image [21]. Although there are people from different ethnicities, the color distribution of skin is relatively clustered together in a specific area of an image. Our algorithm captured an image frame from live video feed via camera in RGB color-space, converted it into HSV color-space and used a combination of Hue and saturation value to correctly identify the face skin from the captured frame. The assumption is that the facial skin is the only part of the body skin exposed within the field-of-view of the camera. The results are shown in figure 5 (a).

Even though this method is very easy to implement to cluster out skin in finding the face, the obvious problem comes when there is any other object in the scene which contains similar color tone. Variation of illumination also poses a great challenge for this method to work properly. Problems due to these limitations can be observed from figure 5(b).

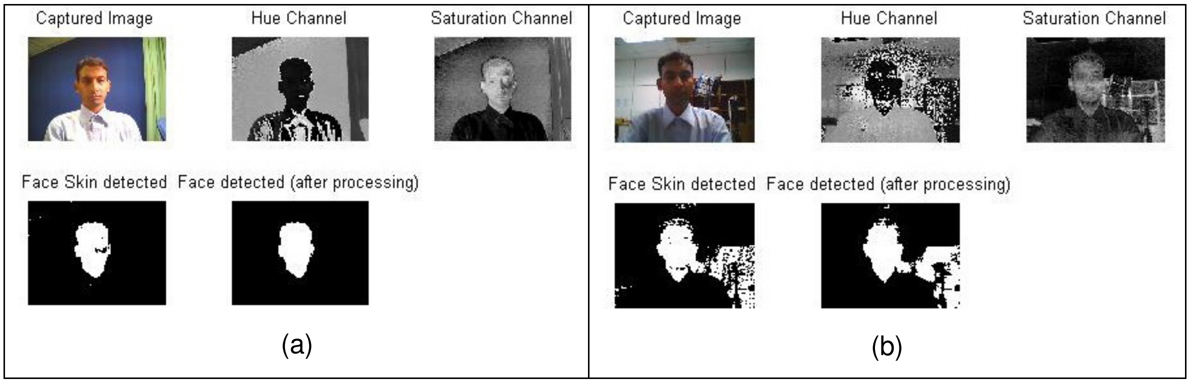


FIGURE 5: (a) Different stages of processing the input video frame to identify a human face using skin color information. (b) Error in face detection using skin color picked up some part of a bookshelf as part of detected face.

Improved Face Detection Using Appearance-based Method

To overcome the limitations of previous method, we adopted a new face detection technique using local Successive Mean Quantization Transform (SMQT) and the split-up Sparse Network of Winnows (SNoW) classifier [22]. Local SMQT features are used to extract the illumination-insensitive properties from image data. Split up SNoW classifier is an appearance-based pattern recognition method that can be utilized to find face object. Results of applying this method can be observed in figure 6.

5. VISUAL SERVOING AND TRACKING OF HUMAN FACE

Visual Servoing is the way of using vision data to control the motion of a robot while target tracking refers to constantly following a moving target body and adapt its own motion to maintain the target within its observation range. Visual servoing and tracking of an object, as the name implies, exploits computer vision for its input, manipulates the input data using different image processing techniques to convert it into an acceptable form to the system, and finally utilizes this information to control its own motion so that the target object remains within its field-of-vision. The target object in this case is the human face.

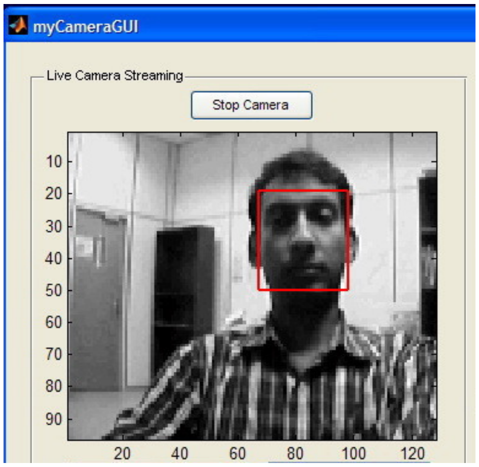


FIGURE 6: Face Detection of AMIR-II using local SMQT and split-up SNoW classifier.

As discussed in [23], visual servoing techniques can be categorized in two broad classes – Image-Based Visual Servo control (IBVS) and Position-Based Visual Servo (PBVS) control. In IBVS control, the parameters that are required for the system to control its motion are immediately available from the image data. In PBVS control, a set of intermediate 3D parameters are computed from the image measurement which would be used for control system.

Image-based visual tracking of human face can take two approaches – a) head tracking, tracking the motion of head as a rigid object [24], b) facial-features tracking, tracking the deformations of shape of the facial features, i.e. eyes, nose, lips, confined within the anatomical head region [25].

For AMIR-II, we implemented Image-Based Visual Servoing and Tracking of human head as a whole which also contains face. Mathematically, the process can be described as minimization of an error **e(t)**, where

$$e(t) = s[m(t),\ a] - s^{*} \tag{1}$$

Here, ***s*** is the set of image-plane coordinates of the points within the FOV, ***m*** are the pixel coordinates constructing the box encircling the face detected (figure 4), and ***a*** is a set of camera dependant parameters which have to be input manually.

6. EXPERIMENTAL RESULTS AND ANALYSIS

Our system is executed on a Personal Computer with Intel Core2Duo processor running at 2.00 GHz speed, 1 GB DDR2 RAM under Microsoft Windows XP (SP3) as its operating system. The program is developed in Matlab with Image Processing Toolbox 2.0. To actuate the AX-12+ Dynamixel servo motors from the program, the supplied library file dynamixel.h from Dynamixel SDK was utilized.

Our experiments are constrained with following assumptions:

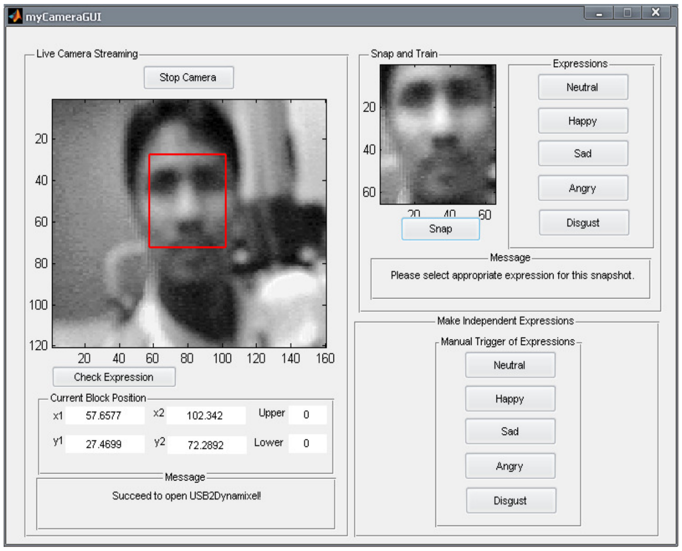
- The face should take up a significant area in the image.
- The system should be operated in indoor illumination condition.
- The face movement should be considerably slow for tracking to be effective.

The GUI program is executed with the USB camera and AX-12+ servos connected to the PC as seen in figure 7. Firstly the message box displays the status of the connection, i.e. "Succeeded to open the USBDynamixel", which means that the PC-servos connection is working properly. As the Start button is clicked the live image streaming starts and automatically the program detects the available human face within the camera FOV.

At the same time, the detected face is being tracked automatically by Amir-II through the movements of its neck joints. Whenever the tracked face moves beyond the AB, Amir-II will adjust its orientation accordingly so that the tracked face can be refocused back into the AB. However, if the tracked face moves too fast, Amir-II would lose its focus towards the face and revert back to the natural position of its neck. To deal with this condition, its face tracking module needs to be further developed in terms of its algorithm to become more adaptive.

To capture the snapshot of the face image, the Snap button is clicked by the robot operator. Satisfied by the human facial expression and the image quality, the relevant facial expression label for the snapshot is then clicked. The image file will be saved into the database with the related expression as part of its filename.

For face detection, we executed some experiments to identify the range of deviation of face view (frontal / partial), pose and the maximum distance from the robot vision system.



x1 57.6577 x2 102.342 Upper 0

y1 27.4699 y2 72.2892 Lower 0

Message

Succeed to open USB2Dynamixel!

Message

Please select appropriate expression for this snapshot.

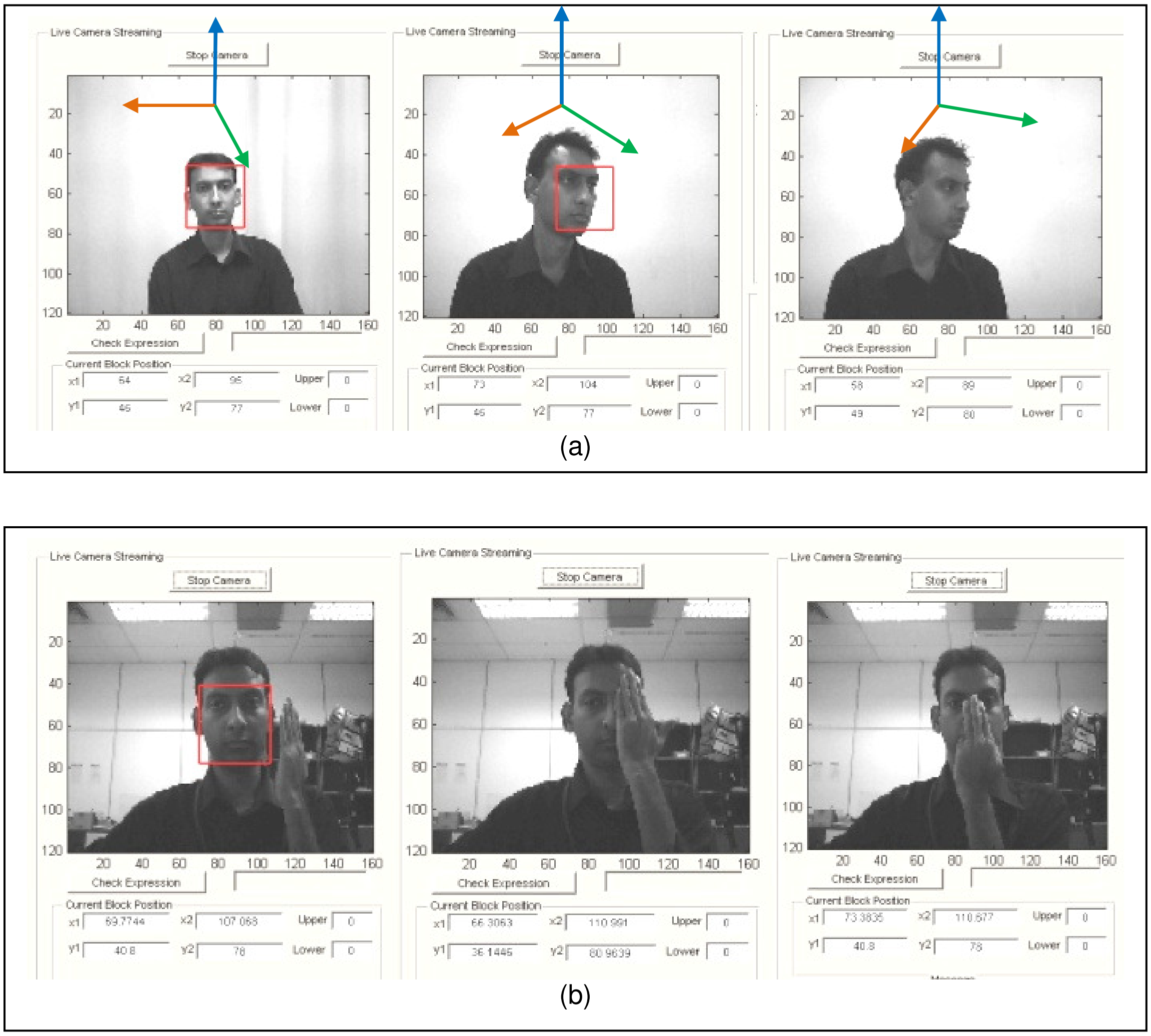
FIGURE 7: The operational GUI

Our face detector could properly identify a face in figure 8(a) as long as significant facial features (nose, two eyes etc) remain visible within the FOV of the camera. In our experiment, the system could detect face correctly with the face rotated by 60º- 65º about vertical axis, depending on how much the face occupies in the entire image. The system cannot detect faces rotated more than 65º about the vertical axis passing through the head.

Even in frontal view of the face, the detector could not identify the face when part of the face was obstructed in figure 8(b). This is because important facial features were missing from their expected symmetric position and hence in the principal component of the image, which was input to the classifier to find face in input image.

Similar limitation was also observed in the pose of frontal face view. The detector can find the face when the head is much deviated from it vertical orientation. In our experiment, the system could not identify a face when it was rotated more than 20º about the horizontal axis as seen in figure 9(a).

The face detector also needs the person whose face is to be detected and tracked be occupying at least 5% of the entire image frame. Beyond this, the size of face image compared to the entire image becomes small enough making the system fail to detect the face as some important facial features lose the details required. This is evident from figure 9(b).



x1 64 x2 96 Upper 0
y1 46 y2 77 Lower 0

x1 73 x2 104 Upper 0
y1 46 y2 77 Lower 0

x1 58 x2 89 Upper 0
y1 49 y2 80 Lower 0

(a)

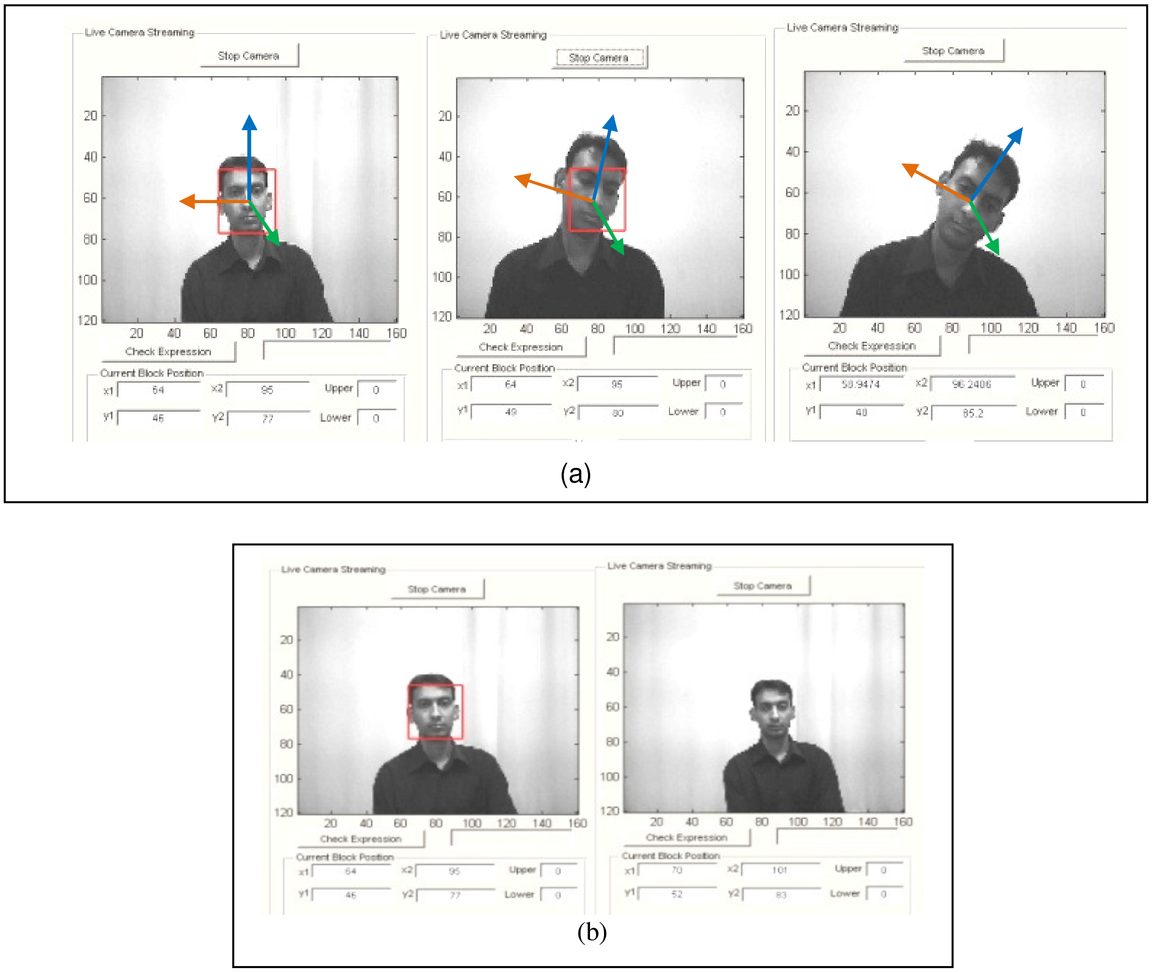
x1 69.7744 x2 107.068 Upper 0
y1 40.8 y2 78 Lower 0

x1 66.3063 x2 110.991 Upper 0
y1 36.1445 y2 80.9639 Lower 0

x1 73.3835 x2 110.677 Upper 0
y1 40.8 y2 78 Lower 0

(b)

FIGURE 8: (a) Detection of face partially occluded due to rotation in an image, (b) Face detection failed with partial obstruction.



x1 64 x2 95 Upper 0
y1 46 y2 77 Lower 0

x1 64 x2 95 Upper 0
y1 49 y2 80 Lower 0

x1 58.9474 x2 96.2406 Upper 0
y1 48 y2 85.2 Lower 0

(a)

x1 64 x2 95 Upper 0
y1 46 y2 77 Lower 0

x1 70 x2 101 Upper 0
y1 52 y2 83 Lower 0

(b)

FIGURE 9: (a) System detecting the face with limited change in pose, (b) the face was not detected even with its frontal view with no occlusion and vertical pose, because the ratio of the face to the entire image is high.

As for servoing and tracking, AMIR-II can track a face nicely as long as the face is detected. One of the significant observations is when more than one face is detected and they have different motions in different directions, AMIR-II vision system can track the faces successfully as long as they remain within the FOV of camera but cannot generate its motion for servoing as it does not have any information regarding which face to follow.

7. CONCLUSION AND FUTURE DIRECTION

The servoing and tracking of human face by AMIR-II enabled us to experiment new techniques for further development of the system. Current detection rate of the face detector of AMIR-II is acceptable according to recent findings where the system was able to successfully track human face as a single object. However, enabling the system to track the facial-features would make it more interactive as the system could be upgraded to recognize and track the facial expressions. The upgrade is already in progress for this project.

Range of servoing of tracked human face could also be improved. Also, the tracking actuation at the moment only involve servoing the motors from point to point so the tracked object remains within its field-of-view. The process could be further improved implementing PID within the loop to reduce the jitter in motion.

The next step forward to progress with this research is to use the face matrix for extracting facial features to identify the emotional states of the human.

8. ACKNOWLEDGEMENT

This work is funded by the Fundamental Research Grant Scheme (FRGS) of the Ministry of Higher Education Malaysia.

9. REFERENCES

[1] H. K. Marek P. Michalowski, "Rhythm in human-robot social interaction," *IEEE Intelligent Systems,* vol. 23, pp. 78-80, 2008.

[2] C. Breazeal, "Socially intelligent robots," *interactions,* vol. 12, pp. 19-22, 2005.

[3] W. P. Rosalind, "Affective computing: challenges," *Int. J. Hum.-Comput. Stud.,* vol. 59, pp. 55-64, 2003.

[4] K. W. Takanori Shibata, Tomoko Saito, Kazuo Tanie "Human Interactive Robot for Psychological Enrichment and Therapy," in *Social Intelligence and Interaction in Animals, Robots and Agents (AISB) 2005*, University of Hertfordshire, Hatfield, England, 2005, pp. 98-107.

[5] A. Billard, B. Robins, K. Dautenhahn, and J. Nadel, "Building Robota, a Mini-Humanoid Robot for the Rehabilitation of Children with Autism," *the RESNA Assistive Technology Journal,* vol. 19, 2006 2006.

[6] H. Kozima, M. Michalowski, and C. Nakagawa, "Keepon - A Playful Robot for Research, Therapy, and Entertainment," *International Journal of Social Robotics,* vol. 1, pp. 3-18, 2009.

[7] H. Kozima, "Infanoid: An experimental tool for developmental psycho-robotics," *International Workshop on Developmental Study,* 2000.

[8] C. L. Breazeal, *Designing sociable robots*. Cambridge, Mass.: MIT Press, 2002.

[9] P. Maja and B. Marian Stewart, "Machine Analysis of Facial Expressions," in *Face Recognition*, Kresimir Delac and M. Grgic, Eds., First Edition ed Viena, Austria: I-Tech Education and Publishing, 2007, pp. 377-416.

[10] Y. I. Tian, T. Kanade, and J. F. Cohn, "Facial Expression Analysis," in *Handbook of Face Recognition*, K. J. Anil and Z. L. Stan, Eds., ed: Springer, 2005, pp. 247-276.

[11] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan, "Recognizing facial expression: machine learning and application to spontaneous behavior," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, pp. 568-573 vol. 2.

[12] M. S. J. K. G. Oh, S.-J. Kim and S. S. Park, "Function and driving mechanism for face robot, buddy," *The Journal of Korea Robotics Society,* vol. 3, pp. 270–277, 2008.

[13] M. Zecca, Y. Mizoguchi, K. Endo, F. Iida, Y. Kawabata, N. Endo, K. Itoh, and A. Takanishi, "Whole body emotion expressions for KOBIAN humanoid robot - preliminary experiments with different Emotional patterns - " in *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, 2009, pp. 381-386.

[14] A. A. I. Shafie, A. Khan, M.R., "Visual tracking and servoing of human face for robotic head Amir-II," in *International Conference on Computer and Communication Engineering (ICCCE), 2010*, Kuala Lumpur, Malaysia, 2010, pp. 1-4.

[15] K. M. Shafie A., "Design and Development of Humanoid Head," in *Proceeding of International Conference of Man Machine System (ICOMMS)*, Langkawi, Malaysia, 2006.

[16] M. N. K. a. s. A. A. Shafie. , N. I. Taufik Y., "Humanoid Robot Head," in *3rd International Conference on Mechatronics (ICOM)*, Kuala Lumpur, Malaysia, 2008.

[17] C. R. Zhao Wenyi, "A Guided Tour of Face Processing," in *Face Processing: Advanced Modeling and Methods*, ed: Academic Press, 2006, pp. 3-53.

[18] T. Matthew and P. Alex, "Eigenfaces for recognition," *J. Cognitive Neuroscience,* vol. 3, pp. 71-86, 1991.

[19] W. Zhao, R. Chellappa, and A. Krishnaswamy, "Discriminant analysis of principal components for face recognition," in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, 1998, pp. 336-341.

[20] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 19, pp. 696-710, 1997.

[21] Y. Yang, S. Ge, T. Lee, and C. Wang, "Facial expression recognition and tracking for intelligent human-robot interaction," *Intelligent Service Robotics,* vol. 1, pp. 143-157, 2008.

[22] M. Nilsson, J. Nordberg, and I. Claesson, "Face Detection using Local SMQT Features and Split up Snow Classifier," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, 2007, pp. II-589-II-592.

[23] F. Chaumette and S. Hutchinson, "Visual Servoing and Visual Tracking," in *Springer Handbook of Robotics*, ed, 2008, pp. 563-583.

[24] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland, "Visually controlled graphics," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 15, pp. 602-605, 1993.

[25] D. Terzopoulos and K. Waters, "Analysis and synthesis of facial image sequences using physical and anatomical models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 15, pp. 569-579, 1993.

INSTRUCTIONS TO CONTRIBUTORS

Robots are becoming part of people's everyday social lives - and will increasingly become so. In future years, robots may become caretaking assistants for the elderly or academic tutors for our children, or medical assistants, day care assistants, or psychological counselors. Robots may become our co-workers in factories and offices, or maids in our homes.

The International Journal of Robotics and Automation (IJRA), a refereed journal aims in providing a platform to researchers, scientists, engineers and practitioners throughout the world to publish the latest achievement, future challenges and exciting applications of intelligent and autonomous robots. IJRA is aiming to push the frontier of robotics into a new dimension, in which motion and intelligence play equally important roles. IJRA scope includes systems, dynamics, control, simulation, automation engineering, robotics programming, software and hardware designing for robots, artificial intelligence in robotics and automation, industrial robots, automation, manufacturing, and social implications.

To build its International reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJRA.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting with volume 2, 2011, IJRA appears in more focused issues. Besides normal publications, IJRA intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

We are open to contributions, proposals for any topic as well as for editors and reviewers. We understand that it is through the effort of volunteers that CSC Journals continues to grow and flourish.

IJRA LIST OF TOPICS
The realm of International Journal of Robotics and Automation (IJRA) extends, but not limited, to the following:

- Automation Control
- Automation Engineering
- Autonomous Robots
- Biotechnology & Robotics
- Emergence of The Thinking Machine
- Forward Kinematics
- Household Robots and Automation
- Inverse Kinematics
- Jacobian and Singularities
- Methods for Teaching Robots
- Nanotechnology & Robotics (Nanobots)
- Orientation Matrices
- Robot Controller
- Robot Structure and Workspace
- Robotic & Automation Software Development
- Robotic Exploration
- Robotic Surgery
- Robotic Surgical Procedures
- Robotic Welding
- Robotics Applications
- Robotics Programming
- Robotics Technologies
- Robots Society and Ethics
- Software and Hardware Designing for Robots
- Spatial Transformations
- Trajectory Generation
- Unmanned (Robotic) Vehicles

CALL FOR PAPERS

Volume: 3 - **Issue:** 1 - February 2012

i. Paper Submission: November 30, 2011 **ii. Author Notification:** January 01, 2012

iii. Issue Publication: January / February 2012

CONTACT INFORMATION

Computer Science Journals Sdn BhD
B-5-8 Plaza Mont Kiara, Mont Kiara
50480, Kuala Lumpur, MALAYSIA

Phone: 006 03 6207 1607
006 03 2782 6991

Fax: 006 03 6207 1697

Email: cscpress@cscjournals.org

CSC PUBLISHERS © 2011
COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA

PHONE: 006 03 6207 1607
006 03 2782 6991

FAX: 006 03 6207 1697
EMAIL: cscpress@cscjournals.org