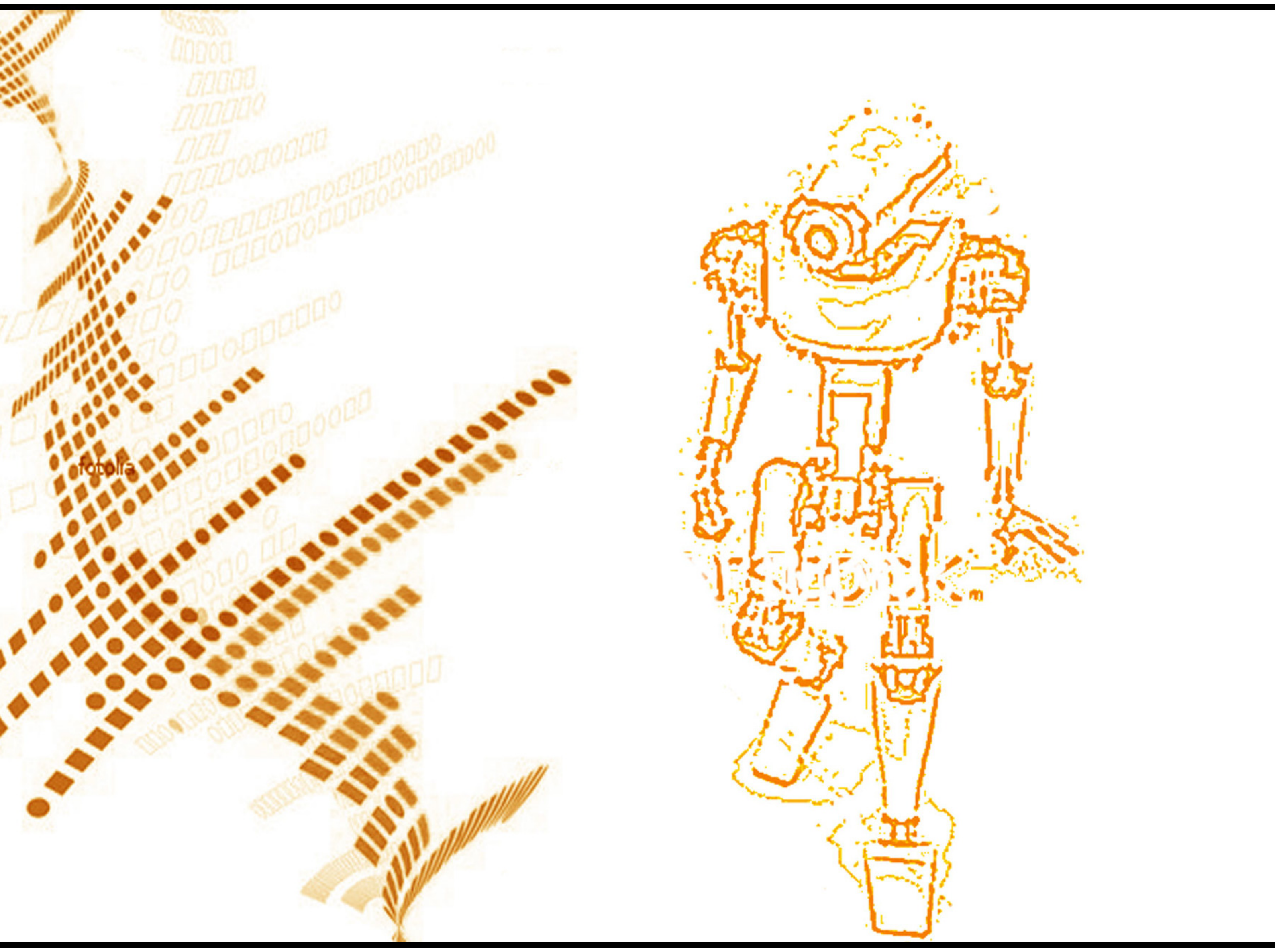
Volume 2 ▪ Issue 4 ▪ September / October 2011

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 4, 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 4, September / October 2011
Publishing Date: 05-10- 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 *Two* 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. Started with Volume 2, 2011, IJRA appear with more focused issues. Besides normal publications, IJRA intends to organize 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 have 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 robotics, 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 4, September / October 2011

Pages

Design of Model Free Adaptive Fuzzy Computed Torque Controller: Applied to Nonlinear Second Order System

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

Amin Jalali Max.Jalali@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

Fereshteh Danesh Narouei f_danesh_na@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 model free adaptive fuzzy computed torque controller (AFCTC) is designed for a two-degree-of freedom robot manipulator to rich the best performance. Computed torque controller is studied because of its high performance. AFCTC has been also included in this study because of its robust character and high performance. Besides, this control method can be applied to non-linear systems easily. Today, robot manipulators are used in unknown and unstructured environment and caused to provide sophisticated systems, therefore strong mathematical tools are used in new control methodologies to design adaptive nonlinear robust controller with acceptable performance (e.g., minimum error, good trajectory, disturbance rejection). The strategies of control robot manipulator are classified into two main groups: classical and non-classical methods, however both classical and non-classical theories have been applied successfully in many applications, but they also have some limitation. One of the most important nonlinear robust controller that can used in uncertainty nonlinear systems, are computed torque controller. This paper is focuses on applied non-classical method in robust classical method to reduce the limitations. Therefore adaptive fuzzy computed torque controller will be presented in this paper.

Keywords: Adaptive Fuzzy Computed Torque Controller, Robot Manipulator, Classical Control, Non-Classical Control, Computed Torque Controller.

1. INTRODUCTION

Controller design is the main part in robotic manipulator as well as the major objectives stability and robustness. Consequently to improve the system's performance lots of researchers are about control systems [3]. One of the most important challenges in control algorithms is a linear behavior controller design for nonlinear systems. When system works with various parameters and hard nonlinearities this technique is very useful in order to implement easily but it has some limitations such as working near the system operating point [3]. Some of robot manipulators which work in industrial processes are controlled by linear PID controllers, but design linear controller for robotic manipulators is extremely difficult because they are nonlinear, uncertainty, and multi input multi output (MIMO) [1-2]. To eliminate the above problems control researchers have used in nonlinear robust controller.

One of the most important powerful nonlinear robust controllers is computed torque controller (CTC), however this controller has analyzed by many researchers recently but the first proposed was in the 1948 [4]. This controller is used in wide range areas such as in robotics, in control process and in aerospace applications because it has an acceptable control performance and solve some main challenging topics in control such as resistivity to the external disturbance. However, this controller is used in wide range areas but, pure CTC has the following disadvantage: uncertainty problem and depending on the dynamic equation. This controller works very well when all dynamic and physical parameters are known but when the robotic manipulator has variation in dynamic parameters, in this situation the controller has no acceptable performance .Calculate the dynamic parameters control formulation is difficult because it depends on the system's dynamic equation [6].

On the other hand, after the invention of fuzzy logic theory in 1965, this theory was used in wide range applications that fuzzy logic controller (FLC) is one of the most important applications in fuzzy logic theory because the controller has been used for nonlinear and uncertain (e.g., robot manipulator) systems controlling. However pure FLC works in many areas but it cannot guarantee the basic requirement of stability and acceptable performance [5].

However both CTC and FLC have been applied successfully in many applications but they also have some limitations. Some researchers are applied fuzzy logic methodology in nonlinear controller to solve the nonlinear dynamic problems in classical controller so called fuzzy classical controller and the other researchers are applied nonlinear classical controller in fuzzy logic controller to improve the stability of systems. In this paper the researcher is applied fuzzy logic method in pure CTC to eliminate the nonlinear dynamic parameter to easy implementation.

Adaptive control is used in systems with various dynamic parameters and need to be training on line. In general states, adaptive control is classified into two main groups: traditional adaptive method and fuzzy adaptive method. Traditional adaptive method need to have some information about dynamic plant and some dynamic parameters must be known. Fuzzy adaptive method can train the parameters variation by expert knowledge. Combined adaptive method for artificial sliding mode controllers can solve the uncertainty challenge in nonlinear systems.

This paper is organized as follows: In section 2, main subject of modelling two degrees of freedom robot manipulator formulation are presented. Detail of fuzzy logic controllers and fuzzy rule base is presented in section 3. In section 4, the main subject of computed torque controller and formulation are presented. The main subject of design fuzzy CTC is presented in section 5. In section 6, design adaptive fuzzy CTC is presented. This section covered the self tuning proposed fuzzy CTC. In section 7, the simulation result is presented and finally in section 8, the conclusion is presented.

2. DYNAMIC FORMULATION OF ROBOT MANIPULATOR

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

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

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

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

Where B(q) is the matrix of coriolios torques, C(q) is the matrix of centrifugal torques, and G(q) is the vector of gravity force. The dynamic terms in equation (2) are only manipulator position. This is a decoupled system with simple second order linear differential dynamics. In other words, the component $\ddot{q}$ influences, with a double integrator relationship, only the joint variableq_i, independently of the motion of the other joints. Therefore, the angular acceleration is found as to be [2, 15-22]:

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

This technique is very attractive from a control point of view. Figure 1 shows the 2 DOF's robot manipulator.

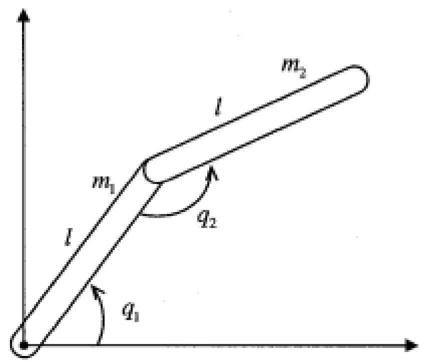


FIGURE 1: 2 DOF robotic manipulators

3. FUZZY INFERENCE SYSTEM

In recent years, artificial intelligence theory has been used in CTC systems. Neural network, fuzzy logic, and neuro-fuzzy are synergically combined with nonlinear classical controller and used in nonlinear, time variant, and uncertainty plant (e.g., robot manipulator). After the invention of fuzzy logic theory in 1965 by Zadeh [10], this theory was used in wide range area. Fuzzy logic controller (FLC) is one of the most important applications of fuzzy logic theory. This controller can be used to control of nonlinear, uncertain, and noisy systems. This method is free of some model-based techniques that used in classical controllers. It should be mentioned that fuzzy logic application is not only limited to the modelling of nonlinear systems [11-15] but also this method can help engineers to design easier controller.

The main reasons to use fuzzy logic technology are able to give approximate recommended solution for unclear and complicated systems to easy understanding and flexible. Fuzzy logic provides a method which is able to model a controller for nonlinear plant with a set of IF-THEN rules, or it can identify the control actions and describe them by using fuzzy rules. Besides using fuzzy logic in the main controller of a control loop, it can be used to design adaptive control, tuning parameters, working in a parallel with the classical and non classical control method [11]. However the application area for fuzzy control is really wide, the basic form for all command types of controllers consists of;

- Input fuzzification (binary-to-fuzzy[B/F]conversion)
- Fuzzy rule base (knowledge base)
- Inference engine
- Output defuzzification (fuzzy-to-binary[F/B]conversion).

The basic structure of a fuzzy controller is shown in Figure 2.

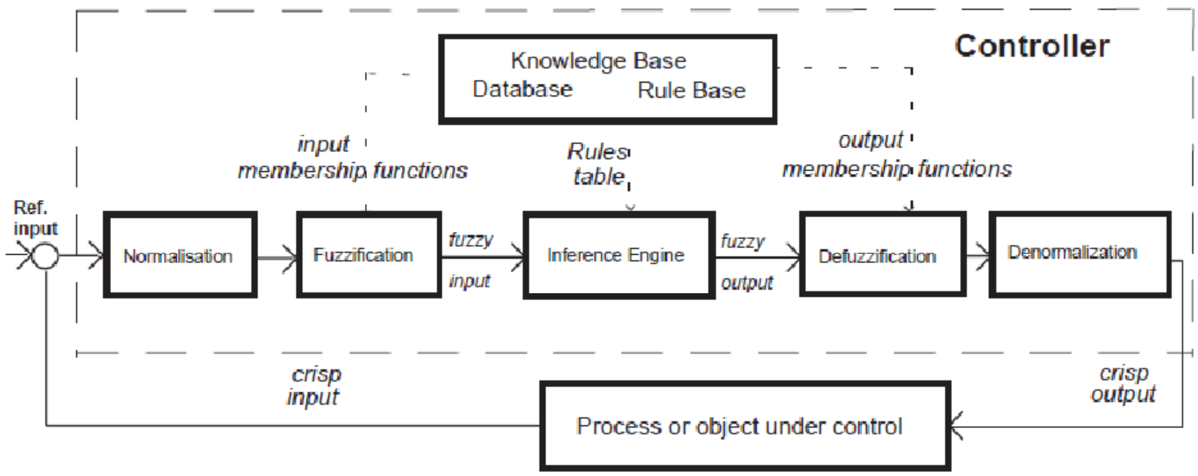


FIGURE 2: Block diagram of a fuzzy controller with details.

4. DESIGN OF A COMPUTED TORQUE CONTROLLER FOR ROBOT ARM

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:

$$\boldsymbol{e(t) = q_d(t) - q_a(t)} \tag{4}$$

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 $\dot{\boldsymbol{x}} = \boldsymbol{Ax + BU}$ can be defined as

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{U} \tag{5}$$

With $\boldsymbol{U} = -\boldsymbol{M^{-1}(q)}.\boldsymbol{N(q,\dot{q})} + \boldsymbol{M^{-1}(q)}.\boldsymbol{\tau}$ 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 $\boldsymbol{x} = [\boldsymbol{e^T}\ \dot{\boldsymbol{e}}^{\boldsymbol{T}}]^{\boldsymbol{T}}$ as:

$$\frac{\boldsymbol{d}}{\boldsymbol{dt}} \begin{bmatrix} \boldsymbol{e} \\ \dot{\boldsymbol{e}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} . \begin{bmatrix} \boldsymbol{e} \\ \dot{\boldsymbol{e}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{U} \tag{6}$$

With $\mathrm{U} = \ddot{\mathrm{q}}_{\mathrm{d}} + \mathrm{M}^{-1}(\mathrm{q}).\{\mathrm{N}(\mathrm{q}.\dot{\mathrm{q}}) - \tau\}$ **(7)**

Then compute the required arm torques using inverse of equation (7), namely, [2]

$$\boldsymbol{\tau = M(q)(\dot{q_d} - U) + N(\dot{q}, q)} \tag{8}$$

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; [1-2]; [6]; [7-8]

$$\boldsymbol{\tau = M(q)(\ddot{q}_d + K_v\dot{e} + K_p e) + N(q, \dot{q})} \tag{9}$$

and the resulting linear error dynamics are

$$\boldsymbol{(\ddot{q}_d + K_v\dot{e} + K_p e) = 0} \tag{10}$$

According to linear system theory, convergence of the tracking error to zero is guaranteed [2, 9].
Where $\boldsymbol{K_p}$ and $\boldsymbol{K_v}$ are the controller gains.
The resulting schemes is shown in Figure 3, 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.

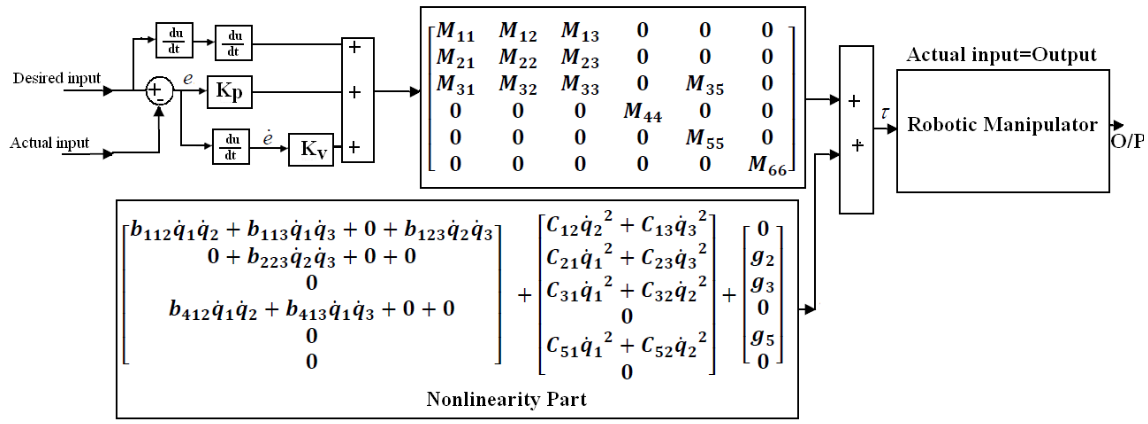


FIGURE 3: Block diagram of PD-computed torque controller (PD-CTC)

5. FUZZY LOGIC AND ITS APPLICATION TO COMPUTED TORQUE CONTROLLER

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

$$\boldsymbol{\tau} = \hat{\boldsymbol{\tau}} + \boldsymbol{\tau}_{\mathbf{fuzzy(S)}} \tag{11}$$

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 fuzzy computed torque controller (FCTC) is design computed torque control combined to fuzzy logic systems to solve the problems in classical computed torque controller [9, 15-19]. 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 4. In this method author obtained the following fuzzy rules for linear part to design fuzzy error base-like nonlinear dynamic parameter control. This rules used instead of nonlinear dynamic equation control to eliminate the nonlinear formulation of dynamic equivalent control term [21-22].

$1 > if\ L\ is\ N.B\ then\ \tau\ is\ N.B$
$2 > if\ L\ is\ Z\ then\ \tau\ is\ Z$

The sliding surface is defined as follows:

$$\boldsymbol{L} = \boldsymbol{M}(\boldsymbol{q})\left(\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{K}_{\boldsymbol{v}}\dot{\boldsymbol{e}} + \boldsymbol{K}_{\boldsymbol{p}}\boldsymbol{e}\right) \tag{12}$$

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

$$\hat{\boldsymbol{\tau}} = \hat{\boldsymbol{\tau}}_{\boldsymbol{nonlinear}} + \boldsymbol{\tau}_{\boldsymbol{lin}} \tag{13}$$

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

$$\hat{\boldsymbol{\tau}}_{\boldsymbol{nonlinear}} = \boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}}\dot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q})\dot{\boldsymbol{q}}^{2} + \boldsymbol{g}(\boldsymbol{q}) \tag{14}$$

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

$$\boldsymbol{\tau}_{\boldsymbol{lin}} = \boldsymbol{M}(\boldsymbol{q})\left(\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{K}_{\boldsymbol{v}}\dot{\boldsymbol{e}} + \boldsymbol{K}_{\boldsymbol{p}}\boldsymbol{e}\right) \tag{15}$$

In proposed FSMC nonlinear control part replaced by Mamdani's fuzzy inference term, therefore (13) can be rewrite as the following equation

$$\hat{\boldsymbol{\tau}} = \boldsymbol{\tau}_{\boldsymbol{fuzzy}} + \boldsymbol{\tau}_{\boldsymbol{lin}} \tag{16}$$

Design fuzzy logic controller for FCTC has five steps:

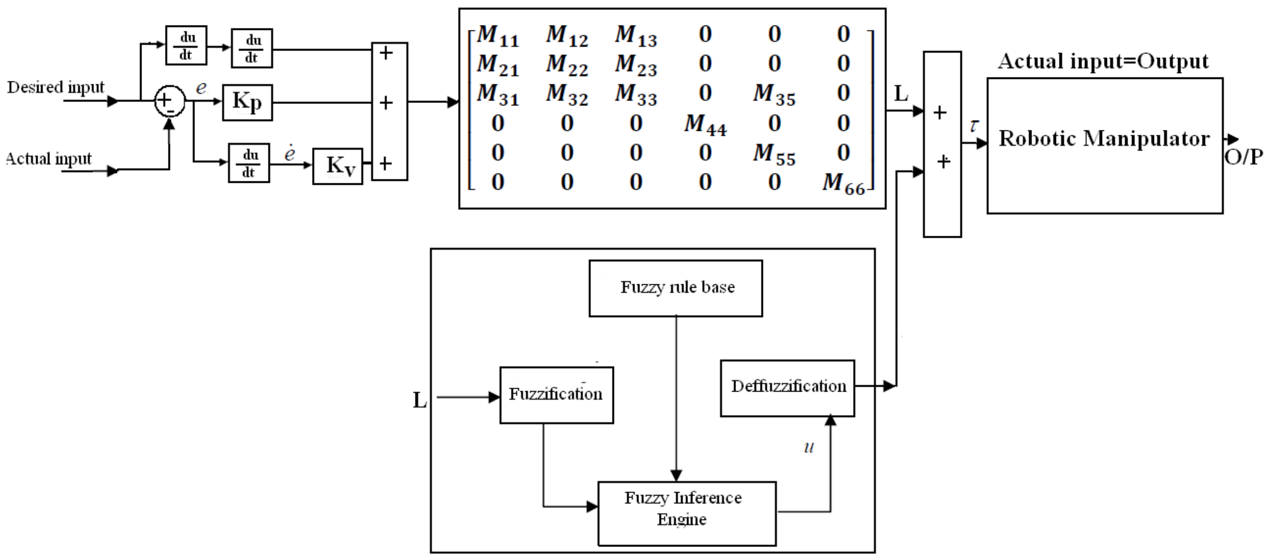


FIGURE 3: Block diagram of proposed fuzzy computed torque controller with minimum rule base

Determine inputs and outputs; find membership function and linguistic variable, type of membership function, fuzzy rule table, and defuzzification. This controller has one input (L) and one output (τ_{fuzzy}). The input is linear part (L) and the output is torque(τ_{fuzzy}). The linguistic variables for linear part (L) 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: -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. The triangular membership function selected in this paper that can be shown in Figure 5. Design the rule base of fuzzy logic controller can play important role to design best performance FCTC. The complete rule base for this controller is shown in Table 1.

TABLE 1: Rule table for proposed FCTC

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

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, which COG method used the following equation to calculate the defuzzification

$$COG(x_k, y_k) = \frac{\sum_i U_i \sum_{j=1}^{r} \cdot \mu_u(x_k, y_k, U_i)}{\sum_i \sum_{j=1}^{r} \cdot \mu_u(x_k, y_k, U_i)} \qquad \textbf{(17)}$$

However, pure FLC, CTC, FCTC, and CTFC used in wide range but they have common limitation to adjust several factors, that the next section is discussed about self tuning the scale factor without using fuzzy rule base.

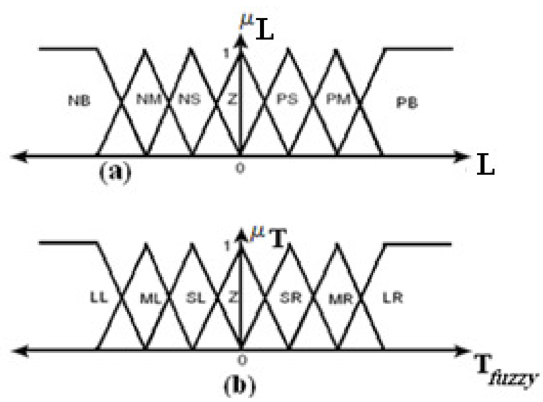


FIGURE 5: Membership function: a) sliding surface b) torque

6. DESIGN OF SELF TUNING FUZZY COMPUTED TORQUE CONTROL (AFCTC) FOR ROBOT ARM

It is a basic fact that the system performance in FCTC is sensitive to gain updating factor, K. Thus, determination of an optimum *K* value for a system is an important problem. If the system parameters are unknown or uncertain, the problem becomes more highlighted. This problem is solved by adjusting the proportional and derivative gain updating factor of the computed torque controller continuously in real-time. Several researchers are worked on adaptive computed torque control and their applications in robotic manipulator in the following references [1-2]; [7-8]. 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 FCTC, namely, proportional gain updating factor (K_P) and derivative gain updating factor (K_v). Self tuning-FCTC has strong resistance and solves the uncertainty problems. The block diagram for this method is shown in Figure 6.

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),

$$K_v^{\,new} = K_v^{\,old} \times KU \qquad \textbf{(18)}$$

$$K_p^{\,new} = K_p^{\,old} \times KU \qquad \textbf{(19)}$$

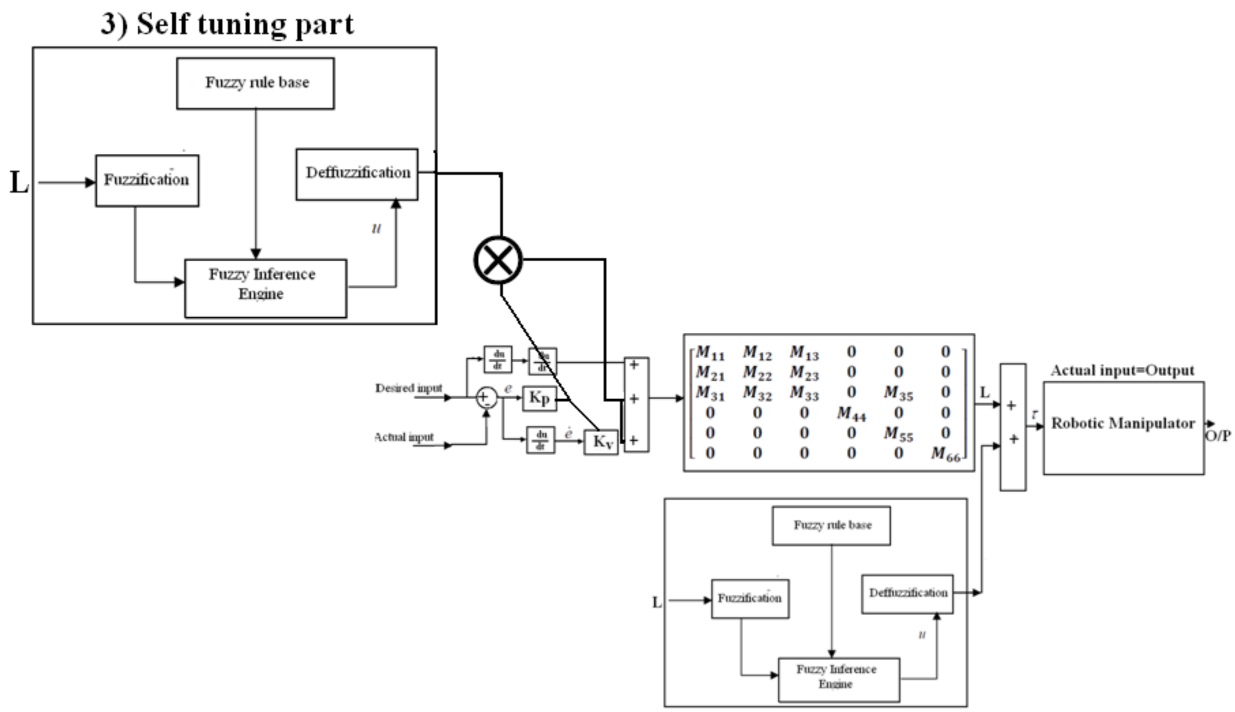


FIGURE 6: Block diagram of proposed self tuning fuzzy computed torque controller with minimum rule base in fuzzy nonlinear part and fuzzy supervisory.

7. SIMULATION RESULT

PD-Classical computed torque controller (PD-CTC), PD-fuzzy computed torque controller (PD-FCTC), and PD-self tuning fuzzy computed torque controller (PD-STFCTC) are implemented in Matlab/Simulink environment. Tracking performance, disturbance rejection and error are compared.

7.1 Tracking Performances:

From the simulation for first and second trajectory without any disturbance, it was seen that PD-STFCTC, PD-CTC and PD-FCTC have the same performance. This is primarily because this system is worked on certain environment. The STFCTC gives significant trajectory good following when compared to FCTC. Figure 7 shows tracking performance without any disturbance for STFCTC, PD-CTC and PD-FCTC.

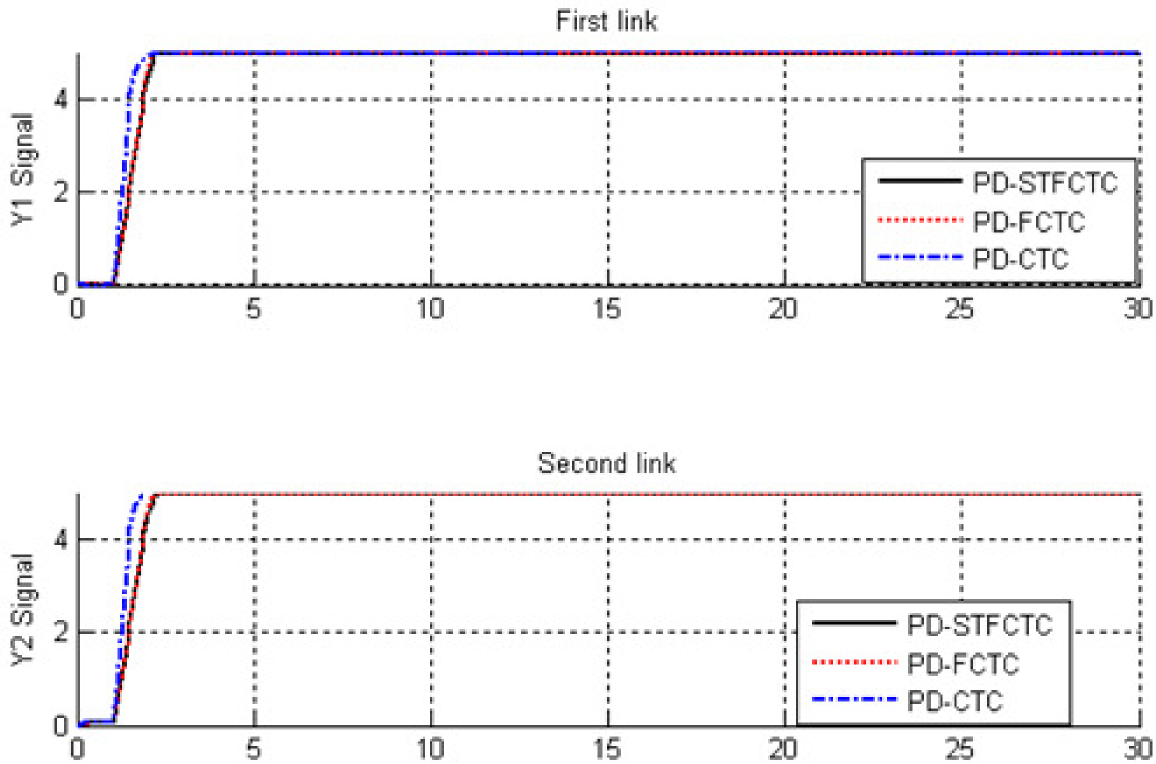


FIGURE 7: Step PD-CTC, PD-FCTC and PD-STFCTC for First and second link trajectory.

By comparing step response trajectory without disturbances in PD-CTC, PD-FCTC and PD-STFCTC it is found that the STFCTC's overshoot **(0%)** is lower than FCTC's **(1%)** and CTC's **(6.4%)**, although all of them have about the same rise time; CTC **(0.403 sec)**, FCTC and STFCTC **(0.5 sec)**.

7.2 Disturbance Rejection

Figure 8 has shown the power disturbance elimination in PD-CTC, PD-FCTC and PD-STFCTC. The main target in these controllers is 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 CTC, FCTC and STFCTC. It found fairly fluctuations in CTC and FCTC trajectory responses. As mentioned earlier, CTC works very well when all parameters are known, this challenge plays important role to select the AFCTC as a based robust controller.

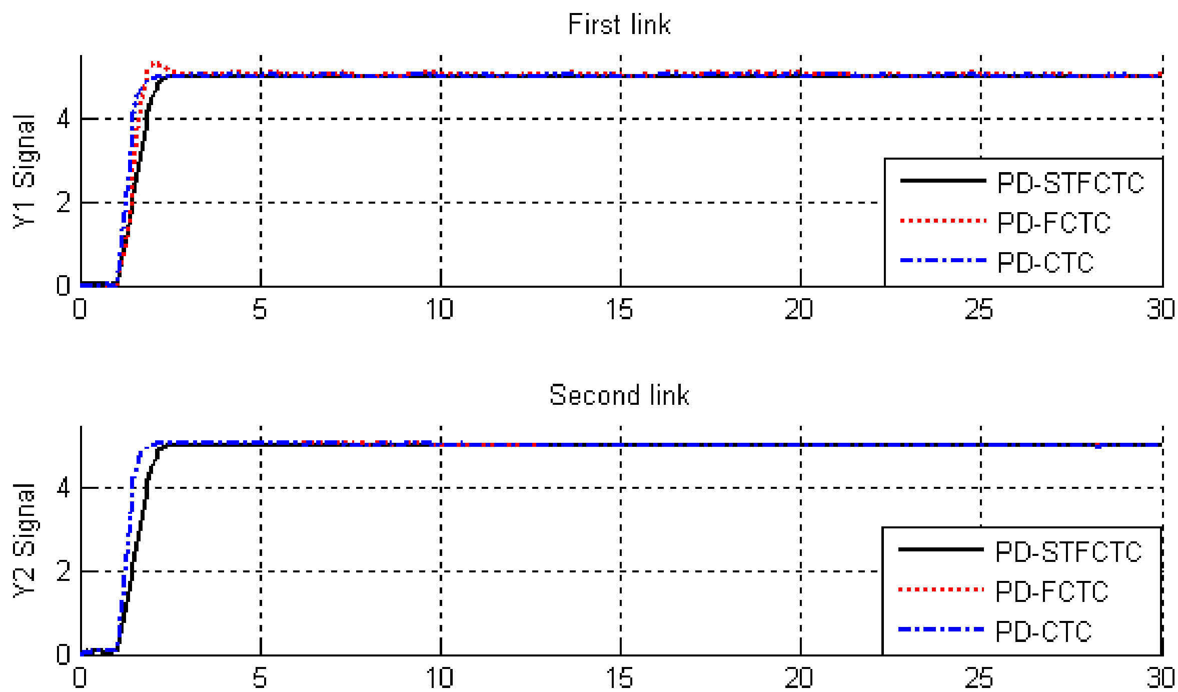


FIGURE 8: Step PD-CTC, PD-FCTC and PD-STFCTC for First and second link trajectory with external disturbances.

Among above graph relating to Step trajectory following with external disturbance, PD-CTC and PD-FCTC have fairly fluctuations. By comparing some control parameters such as overshoot and rise time it found that the STFCTC's overshoot **(0%)** is lower than FCTC's **(8%)** and CTC's **(4%)**, although all of them have about the same rise time; CTC **(0.403 sec)** and FCTC and STFCTC **(0.5 sec).**

7.3 Calculate Errors

The values of RMS and steady state errors for various controllers are tabulated in tables 2 and 3.

8. CONCLUSION

This paper presents a new methodology for designing s self tuning fuzzy computed torque controller with minimum rule bases and high performance for 2 DOF robotic manipulator. From the simulation, it found that self tuning-FCTC has 7 rule base for supervisory and 7 rule base for main controller but in previous design by the other researcher has about 49 rules for supervisory and 49 rules for main controller because it has 2 inputs for main controller (PD) and also it has 2 inputs for supervisory controller $(e, and\ \dot{e})$therefore proposed method is easy to implement. The classical CTC works very well when all parameters are known. In self tuning FCTC, the fuzzy supervisory controller can changed K_p & K_v to achieve the best performance and in this method the supervisory controller is changed the gain updating factor of main FCTC to get the best performance. In one word the implementation of self tuning FCTC is easier than the previous method with more rule base and this reason plays important role in system controller.

Setup input without disturbance	*Overshoot%*		
	PD-CTC	**PD-FCTC**	**PD-STSCTC**
First link	6.44%	1 %	0
Second link	6.44%	1%	0
	$Rise\ time_{(sec)}\ (Tr)$		
	PD-CTC	**PD-FCTC**	**PD-STSCTC**
First link	0.403	0.5	0.5
Second link	0.403	0.5	0.5
	Steady state error		
	PD-CTC	**PD-FCTC**	**PD-STSCTC**
First link	-3.6e-5	4e-5	0
Second link	-2.54e-5	4e-5	0
	$settling\ time_{(sec)}\ (Ts)$		
	PD-CTC	**PD-FCTC**	**PD-STSCTC**
First link	0.9	0.76	0.65
Second link	0.9	0.76	0.60
	RMS error		
	PD-CTC	**PD-FCTC**	**PD-STSCTC**
RMS error	-1.34e-5	1.5e-5	0

TABLE 2: Three types of controllers summaries without disturbance.

Setup input with 40% disturbance	*Overshoot%*		
	PD-CTC	**PD-FCTC**	**PD-STSCTC**
First link	4%	8 %	0
Second link	4%	8%	0
	$Rise\ time_{(sec)}\ (Tr)$		
	PD-CTC	**PD-FCTC**	**PD-STSCTC**
First link	0.403	0.5	0.5
Second link	0.403	0.5	0.5
	Steady state error		
	PD-CTC	**PD-FCTC**	**PD-STSCTC**
First link	0.005	-0.0019	0
Second link	0.005	-0.0019	0
	$settling\ time_{(sec)}\ (Ts)$		
	PD-CTC	**PD-FCTC**	**PD-STSCTC**
First link	1	0.8	0.65
Second link	1	0.8	0.60
	RMS error		
	PD-CTC	**PD-FCTC**	**PD-STSCTC**
RMS error	0.0042	0.0025	0.0001632

TABLE 3: Three types of controllers summaries with external disturbance.

REFERENCES

[1] Thomas R. Kurfess, Robotics and Automation Handbook: CRC press, 2005.

[2] Bruno Siciliano and Oussama Khatib, Handbook of Robotics: Springer, 2007.

[3] Slotine J. J. E., and W. Li., Applied nonlinear control: Prentice-Hall Inc, 1991.

[4] Boiko I, L. Fridman, A. Pisano, and E. Usai, "Analysis of chattering in systems with second-order sliding modes", IEEE transactions on Automatic control, 52(11): 2085-2102, 2007.

[5] L.X.Wang, "stable adaptive fuzzy control of nonlinear systems", IEEE transactions on fuzzy systems, 1(2): 146-154, 1993.

[6] Frank L.Lewis, Robot dynamics and control, in robot Handbook: CRC press, 1999.

[7] A.Vivas, V.Mosquera, "predictive functional control of puma robot", ACSE05 conference, 2005.

[8] D.Tuong, M.Seeger, J.peters," Computed torque control with nonparametric regressions models", American control conference, pp: 212-217, 2008.

[9] Farzin Piltan, A. R. Salehi and Nasri B Sulaiman.," Design artificial robust control of second order system based on adaptive fuzzy gain scheduling," world applied science journal (WASJ), 13 (5): 1085-1092, 2011.

[10] Lotfi A. Zadeh" Toward a theory of fuzzy information granulation and its centrality in human easoning and fuzzy logic" Fuzzy Sets and Systems 90 (1997) 111-127

[11] Reznik L., Fuzzy Controllers, First edition: BH NewNes, 1997.

[12] Zhou, J., Coiffet, P," Fuzzy Control of Robots," Proceedings IEEE International Conference on Fuzzy Systems, pp: 1357 – 1364, 1992.

[13] Banerjee, S., Peng Yung Woo, "Fuzzy logic control of robot manipulator," Proceedings Second IEEE Conference on Control Applications, pp: 87 – 88, 1993.

[14] Akbarzadeh-T A. R., K.Kumbla, E. Tunstel, M. Jamshidi. ,"Soft Computing for autonomous Robotic Systems," IEEE International Conference on Systems, Man and Cybernatics, pp: 5252-5258, 2000.

[15] Lee C.C.," Fuzzy logic in control systems: Fuzzy logic controller-Part 1," IEEE International Conference on Systems, Man and Cybernetics, 20(2), P.P: 404-418, 1990.

[16] F. Piltan*, et al.*, "Artificial Control of Nonlinear Second Order Systems Based on AFGSMC," *Australian Journal of Basic and Applied Sciences, 5(6)*, pp. 509-522, 2011.

[17] 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.

[18] Piltan, F., et al., "Design Artificial Nonlinear Robust Controller Based on CTLC and FSMC with Tunable Gain," International Journal of Robotic and Automation, 2 (3): 205-220, 2011.

[19] Piltan, F., et al., "Design Mathematical Tunable Gain PID-Like Sliding Mode Fuzzy Controller with Minimum Rule Base," International Journal of Robotic and Automation, 2 (3): 146-156, 2011.

[20] Piltan, F., et al., "Design of FPGA based sliding mode controller for robot manipulator," International Journal of Robotic and Automation, 2 (3): 183-204, 2011.

[21] Piltan, F., et al., "A Model Free Robust Sliding Surface Slope Adjustment in Sliding Mode Control for Robot Manipulator," World Applied Science Journal, 12 (12): 2330-2336, 2011.

[22] Piltan, F., et al., "Design Adaptive Fuzzy Robust Controllers for Robot Manipulator," World Applied Science Journal, 12 (12): 2317-2329, 2011.

Human Arm Inverse Kinematic Solution Based Geometric Relations and Optimization Algorithm

Mohammed Z. Al-Faiz, , MIEEE *mzalfaiz@ieee.org*
omputer Engineering Dept. .
Nahrain University
Baghdad, Iraq

Abduladhem A.Ali *Abduladem1@yahoo.com*
Computer Engineering Dept. .
Basrah University
Basrah, Iraq

Abbas H.Miry *abbasmiry83@yahoo.com*
Electrical Engineering Dept. .
AL-Mustansiriyah University
Baghdad, Iraq

Abstract

Kinematics for robotic systems with many degrees of freedom (DOF) and high redundancy are still an open issue. Namely, computation time in robotic applications is often too high to reach good solution, for parts of the kinematic chain; the problem of inverse kinematics is not linear, as rotations are involved. This means that analytical solutions are only available in limited situations. In all other cases, alternative methods will have to be an employed.The most-used alternative is numerical solutions optimization. This paper presents a strategy based on combine's analytical solutions with nonlinear optimization algorithm solutions to solution the IKP. A analytical solutions is used to reduce the size of problem from seven variable of joint angle to single variable and nonlinear optimization algorithm was used to find approximate solution which make the computation time is very small

Keywords: Inverse Kinematic, Human Arm, Levenbrge Marquite.

1. INTRODUCTION

Kinematics is the study of motion without regard to the forces that create it. The representation of the robot's end-effecter position and orientation through the geometries of robots (joint and link parameters) are called forward Kinematics.[1].The forward kinematics is a set of equations that calculates the position and orientation of the end-effector in terms of given joint angles. This set of equations is generated by using the D-H parameters obtained from the frame assignation. [2].The inverse kinematics problem (IKP) for a robotic manipulator involves obtaining the required manipulator joint values for a given desired end-point position and orientation. It is usually complex due to lack of a unique solution and closed-form direct expression for the inverse kinematics mapping. [3].

Forward kinematics can be formulated for all serial manipulators. However, with increasing degrees-of-freedom (DOF), these solutions become extremely complex and possibly computationally inefficient. Inverse position solutions are however only possible for non-redundant robots that have a finite number of solutions. For redundant systems, inverse kinematics leads to an infinite number of solutions and numerical approaches are the best to be used. Additionally, decision-making and optimization becomes important for redundant system inverse kinematics [4].

Kinematics of the human body is concerned with formulating and solving for the translational and rotational position, analysis problems for each human body segment of interest, for various real world motions. Forward kinematics calculates the pose (position and orientation) of each human body segment of interest given the joint angles. The forward kinematics is the problem of finding an end-effector or tool pose from a set of given joint angles. Inverse kinematics calculates the required joint angles given the current human body (or portion thereof) pose. Statics requires the positions and angles of each segment for static free-body diagrams[1].

Several inverse kinematics algorithms have been proposed. The latest approach dealing with inverse kinematics using nonlinear optimization solution for IKP by Sugihara [5]he use Levenberg-Marquardt method with robust damping with n variable depend on problem which may take long time .

In order to describe a kinematic chain, we are going to view the rotations performed by the joints of a chain separately from the setup of the chain, i.e. length of links,position and orientation of joints, etc , In the space, translation and rotation can be expressed by homogeneous transformation matrices. In the 3D Euclidean

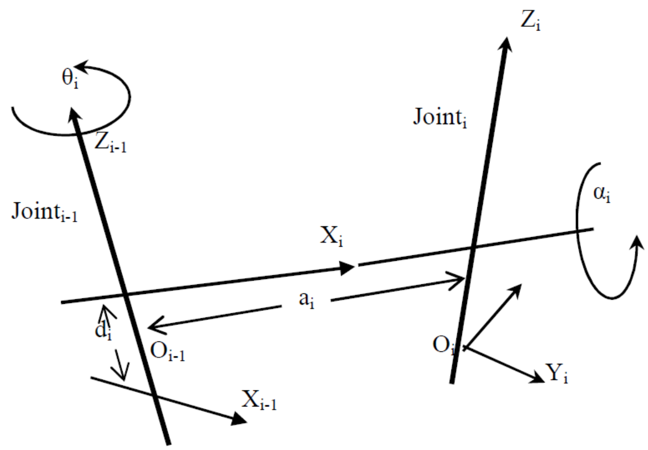


FIGURE 1. The relation between two consecutive coordinates

space, these take the form of 4 × 4 matrices which can be concatenated simply by matrix multiplication. Since it is essential to combine the frame transitions with other translations and rotations, foremost the rotations performed by the joints, the rotation matrices and translation vectors have to be transformed to homogeneous transformation matrices To transform a 3 × 3 rotation matrix or a 3 × 1 translation vector into homogeneous transformation matrices, they are positioned in a 4 × 4 matrix where the remaining entries are filled with the identity matrix. A rotation matrix R replaces the upper left part of the 4 × 4 matrix, a translation vector T replaces the three upper entries of the last column[6]:

$$\left.\begin{aligned} rot_{hom} &= \begin{bmatrix} & R_{3x3} & & \begin{matrix}0\\0\\0\end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ trans_{hom} &= \begin{bmatrix} 1 & 0 & 0 & \\ 0 & 1 & 0 & t_{3x1} \\ 0 & 0 & 1 & \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ rot_{hom}\, trans_{hom} &= \begin{bmatrix} & R_{3x3} & & t_{3x1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}\right\} \qquad (1)$$

As demonstrated, a single homogeneous transformation matrix can be equipped with both a translational and a rotational part. This enables a composition of the rotations performed by the joints and the frame transitions caused by the nature of the kinematic chain.

$$A_i = Rot_{x,\alpha_i}\, Trans_{x,a_i}\, Rot_{z,\theta_i}\, Trans_{z,d_i}$$

$$= \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & b_i \\ \cos\alpha_i \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i & -d_i \sin\alpha_i \\ \sin\alpha_i \sin\theta_i & \sin\alpha_i \cos\theta_i & \cos\alpha_i & d_i \cos\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The kinematic analysis of an n-link manipulator can be extremely complex and the conventions introduced below simplify the analysis considerably. Moreover, they give rise to a universal language with which robot engineers can communicate. A commonly used convention for selecting frames of reference in robotic applications is the Denavit-Hartenberg, (DH) convention. In this convention, each homogeneous transformation A_i is represented as a product of four basic transformations[7].where the four quantities θ_i α_i, a_i, d_i, are parameters associated with link (*i*) and joint (*i*). The four parameters a_i, α_i, d_i, and θ_i in equation(2) are generally given the names link length, link twist, link offset, and joint angle, These names are derive from specific aspects of the geometric relationship between two coordinate frames, as will become apparent below. Since the matrix A_i is a function of a single variable, it turns out that three of the above four quantities are constant for a given link, while the fourth parameter, θ_i for a revolute joint and d_i for a prismatic joint, is the joint variable.

In order to obtain a systematic representation of the workspace produced by the motion of a point of interest (typically called a point on the end-effector), we will use the (D-H) method adopted from the field of robotics [8].

Let Z_{i-1} and Z_i represent fixed axes at either end of link *i*-1, about which or along which links *i*-1 and *i* move, respectively. Let axes X_{i-1} be defined from Z_{i-1} to Z_i and perpendicular to both. Let Y_{i-1} represent the unique axis that together with X_{i-1} and Z_{i-1} completes a right-hand Cartesian coordinate system. Let Z_i represent a vector from O_{i-1} parallel to Z_i . Let X_{i-1} a vector from O_i parallel to X_{i-1} as illustrated in Fig.1.

Note that the four parameters θ_i α_i, a_i, d_i, completely define the relation between any two consecutive frames. These values are entered in a table, which is typically known as the DH Table. The overall Denavit-Hartenberg respectively coordinate transformation matrix from frame *i* coordinate system relative to the frame *i-1* coordinate system is then given by matrix A_i same as equation(2).

From above, it is clear that the position and orientation of the manipulator end effecter are obtained based on the joint displacements. The joint displacement corresponding to a given end effecter location is obtained by solving the inverse kinematics equations. Hence here, we are concerned not only with the final position of the end effecter, but also with the velocity with which the end effecter moves. describes the kinematics of the manipulator and it maps the joint vector into the end effecter position vector.

2. STRUCURE AND KINEMATIC OF HUMAN ARM

The development of a high-DOF, kinematic is discusses human model that can be used to predict realistic human arm postures. one may deal with anthropomorphic arm by 7-DOF and assume the origin at shoulder joint. The first joint is the shoulder joint s with 3 DOFs. The elbow joint e has only one DOF. The wrist joint w is of the same type as the shoulder joint s and also has 3 DOFs.

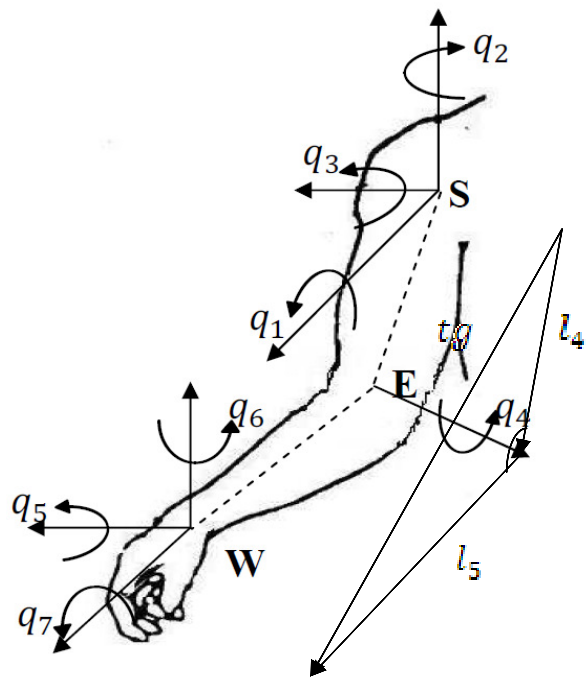


FIGURE 2. Kinematic chain of human arm

Note that the arrow at the end of the chain indicates the end effectors orientation and is not another link. It can be focused on a kinematic chain that is formed after a human arm. This means the kinematic chain has 3 joints with spherical joints as shoulder and wrist joint and a hinge joint as the elbow joint. The spherical joints have 3 DOFS while the hinge joint has only one DOF, giving a total of 7 DOFs for this kinematic chain, see Fig.2. The homogeneous transformation matrices for the frame transitions are set up with D-H parameters [9].

TABLE.1 Numeric Value for D-H Parameters

Frame (joint)	q_i(rad)	d_i(cm)	a_i(cm)	α_i(rad)
1	q_1	0	0	$\pi/2$
2	$q_2+\pi/2$	0	0	$\pi/2$
3	q_3	0	0	$-\pi/2$
4	q_4	0	L4	$\pi/2$
5	q_5	0	L5	$-\pi/2$
6	$q_6-\pi/2$	0	0	$-\pi/2$
7	q_7	0	0	$-\pi/2$

$$A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_3 = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_3 & -c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} c_4 & -s_4 & 0 & l_4 \\ 0 & 0 & -1 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_5 = \begin{bmatrix} c_5 & -s_5 & 0 & l_5 \\ 0 & 0 & 1 & 0 \\ -s_5 & -c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_7 = \begin{bmatrix} c_7 & -s_7 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_7 & -c_7 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

It has following transformation matrices where 'c' is cosine of theta and 's' is sin of theta, The forward kinematic represents by T

$$T = A_1 * A_2 * A_3 * A_4 * A_5 * A_6 * A_7$$

$$T = \begin{bmatrix} nx & ox & ax & dx \\ ny & oy & ay & dy \\ nz & oz & az & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

3. PROPSED METHOD OF INVERSE KINEMATICS

3.1 Analytical Section

The IK will be considered now. The proposed algorithm can be explain in the following:

Step one: In first we find θ_4 because it depend on posture of arm , tg represent the distance between start point and target point as shown in fig .2 , according to kinematics equation

$$tg = \sqrt{(dx^2 + dy^2 + dz^2)} \quad (4)$$

and from the fig .2

$$tg = l_4{}^2 + l_5{}^2 - 2l_4 l_5 cos\theta_4 \quad (5)$$

Then $\theta_4 = cos^{-1}\frac{tg^2 - l_4{}^2 - l_5{}^2}{2l_4 l_5}$ (6)

Step 2: In this step find θ_1: dx, dz have similar form as shown

$$dx = -l_5(c_4(s_1s_3 + c_1c_3s_2) + c_1c_2s_4) - l_4(s_1s_3 + c_1c_3s_2) \quad (7)$$

$$dz = l_5(c_4(c_1s_3 - c_3s_1s_2) - c_2s_1s_4) + l_4(c_1s_3 - c_3s_1s_2) \quad (8)$$

Rearrangement dx and dz

$$-dx = s_1(l_5c_4s_3 + l_4s_3) + c_1(l_5c_4c_3s_2 + l_5c_2s_4 + l_4c_3s_2)$$

$$dz = c_1(l_5c_4s_3 + l_4s_3) - s_1(l_5c_4c_3s_2 + l_5c_2s_4 + l_4c_3s_2)$$

$$let\ k_1 = (l_5c_4s_3 + l_4s_3) \quad (9)$$

$$k_2 = (l_5c_4c_3s_2 + l_5c_2s_4 + l_4c_3s_2) \quad (10)$$

Sub k_1, k_2 in eq 7 and 8 to get

$$s_1\ k_1 + c_1\ k_2 = -dx \quad (11)$$

$$c_1\ k_1 - s_1\ k_2 = dz \quad (12)$$

$$let\ rcos\varphi = k_1 \quad (13)$$

$$rsin\varphi = k_2 \quad (14)$$

Sub eq 13 and 14 in eq 11 and 12

$$r\ sin\theta_1\ cos\varphi + r\ cos\theta_1\ sin\varphi = -dx \quad (15)$$

$$r\ cos\theta_1\ cos\varphi - r\ sin\theta_1\ sin\varphi = dz \quad (16)$$

Rearrangement eq 15 and 16

$$r\ (sin\theta_1\ cos\varphi + cos\theta_1\ sin\varphi) = -dx$$

$$r\ (cos\theta_1\ cos\varphi - sin\theta_1\ sin\varphi) = dz$$

Using triangular formula of sinusoidal

$$r\ sin(\theta_1 + \varphi) = -dx \quad (17)$$

$$r\cos(\theta_1+\varphi) = dz \quad (18)$$

Divide eq 17 by eq 18 to get

$$\tan(\theta_1+\varphi) = -\frac{dx}{dz} \quad (19)$$

from this derivative we find the value of θ_1 which depended on φ

$$\theta_1 = \begin{cases} \tan^{-1}\left(\frac{-dx}{dz}\right) - \varphi & if\ dz \neq 0 \\ -\frac{\pi}{2} - \varphi & if\ dz = 0 \end{cases} \quad (20)$$

Step 3: In this step find θ_3: from eq 17 the value of r is found

$$r = \frac{-dx}{\sin(\theta_1+\varphi)} \quad (21)$$

Eq 9 is equal to eq 13

$$l_5\cos\theta_4\sin\theta_3 + l_4\sin\theta_3 = r\cos\varphi$$

Therefore θ_3 can be found

$$\theta_3 = \sin^{-1}\frac{r\cos\varphi}{l_5\cos\theta_4 + l_4} \quad (22)$$

Step 4: In this step find θ_2: By compare dy with k_2

$$dy = l_5(s_2s_4 - c_2c_3c_4) - l_4c_2c_3 \quad (23)$$

Rearrangement k_2 and dy

$$s_2(l_5c_4c_3 + l_4c_3) + c_2(l_5s_4) = k_2 \quad (24)$$

$$-c_2(l_5c_4c_3 + l_4c_3) + s_2(l_5s_4) = dy \quad (25)$$

Let

$$k_3 = l_5c_4c_3 + l_4c_3 \quad (26)$$

$$k_4 = l_5s_4 \quad (27)$$

Which all coefficient are specified, sub eq 26 and in 27 in eq 25 and 26

$$s_2k_3 + c_2k_4 = k_2 \quad (28)$$

$$-c_2k_3 + s_2k_4 = dy \quad (29)$$

Solving eq 28 and 29 using matrix inversion

$$\begin{bmatrix} k_3 & k_4 \\ k_4 & -k_3 \end{bmatrix}\begin{bmatrix} s_2 \\ c_2 \end{bmatrix} = \begin{bmatrix} k_2 \\ dy \end{bmatrix} \quad (30)$$

$$\begin{bmatrix} s_2 \\ c_2 \end{bmatrix} = \begin{bmatrix} k_3 & k_4 \\ k_4 & -k_3 \end{bmatrix}^{-1}\begin{bmatrix} k_2 \\ dy \end{bmatrix} \quad (31)$$

Let

$$\begin{bmatrix} k_5 \\ k_6 \end{bmatrix} = \begin{bmatrix} k_3 & k_4 \\ k_4 & -k_3 \end{bmatrix}^{-1}\begin{bmatrix} k_2 \\ dy \end{bmatrix} \quad (32)$$

Then

$$\sin\theta_2 = k_5 \quad (33)$$

$$\cos\theta_2 = k_6 \quad (34)$$

From latest equations we get

$$\theta_2 = \begin{cases} \tan^{-1}\frac{k_5}{k_6}\ if\ k_6 \neq 0 \\ \frac{\pi}{2}\ if\ k_6 = 0 \end{cases} \quad (35)$$

After these procedure $\theta_1, \theta_2, \theta_3\ and\ \theta_4$ are found

In next steps we find $\theta_5, \theta_6\ and\ \theta_7$, In eq 3 , $A_1 * A_2 * A_3 * A_4$ is specified ,to find $A_5 * A_6 * A_7$ apply

$$A_5 * A_6 * A_7 = [A_1 * A_2 * A_3 * A_4]^{-1} * T \quad (36)$$

Step 5: In this step find θ_6: from eq 37

$$A_5 * A_6 * A_7 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (37)$$

And from from given DH parameter

$$A_5 * A_6 * A_7 = \begin{bmatrix} c_5 * c_7 * s_6 - s_5 * s_7 & -c_7 * s_5 - c_5 * s_6 * s_7 & c_5 * c_6 \\ -c_6 * c_7 & c_6 * s_7 & s_6 \\ -c_5 * s_7 - c_7 * s_5 * s_6 & s_5 * s_6 * s_7 - c_5 * c_7 & -c_6 * s_5 \end{bmatrix} \quad (38)$$

By comparing eq 38 and eq 39 we get

$sin\theta_6 = r_{23}$ (39)
then
$\theta_6 = \sin^{-1} r_{23}$ (40)
Step 6: In this step find θ_5: by comparing eq 37 and eq 38 we get
$r_{13} = c_5 * c_6$ (41)
$r_{33} = -c_6 * s_5$ (42)
then dived eq 42 by eq 41 to obtain
$\tan \theta_5 = -\frac{r_{33}}{r_{13}}$ (43)

$$\theta_5 = \begin{cases} \tan^{-1} -\frac{r_{33}}{r_{13}} & if \quad r_{13} \neq 0 \\ \frac{\pi}{2} & \text{if} \quad r_{13} = 0 \end{cases} \quad (44)$$

Step 6: In this step find θ_7: by comparing eq 37 and eq 38 we get
$-c_6 * c_7 = r_{21}$ (45)
$c_6 * s_7 = r_{22}$ (46)

$$\theta_7 = \begin{cases} \tan^{-1} -\frac{r_{22}}{r_{21}} & if \quad r_{21} \neq 0 \\ \frac{\pi}{2} & if \; r_{21} = 0 \end{cases} \quad (47)$$

From previous derivative we make all angle $(\theta_1, \theta_2, \theta_3, \theta_5, \theta_6 \; and \; \theta_7)$ in term single angle φ therefore the our problem convert from multivariable to one variable which reduce the time required to find the solution .until last step analytical solution is performed.

3.2 . Optimization Section

Now the non linear optimization solution is performed depend on seven angle which depend on single angle φ therfor the next problem has single varaible .The robot kinematics is mathematically represented by a set of constraints on the joint displacement vector $\theta = [\theta_1 \; \theta_2 \cdots \theta_7]^T$,where *7* is the degree of freedom. A positional constraint is represented as
$p(\theta) = p^d$ (48)
Where
The vector θ depended on the value of φ from provieus procdure
p^d Target position in the space.
$p(\theta)$ Calculated position as function of joint space

For an orientation constraint,
$R(\theta) = R^d$ (49)
R^d Target orientation in the space.
$R(\theta)$ Calculated orientation as function of joint space
In both cases, the residual vector *e(q)* can be defined as

$$e(\theta) = \begin{cases} P^d - P(\theta) \; (\text{for a positional constraint}) \\ \alpha(R^d * R(\theta)^T)(\text{for an orientational constraint}) \end{cases} \quad (50)$$

Where $\alpha(RT) = \left[\frac{1}{||l||} tan^{-1} \frac{||l||}{r_{11}+r_{22}+r_{33-1}}\right] l$ (51)

$$RT = R^d * R^T(q) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (52)$$

$$l = \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{12} - r_{12} \end{bmatrix} \quad (53)$$

Our interest starts from solving the following nonlinear
Equation:

e(θ)=0 (54)

The conventional IK based on NR tries to find $\theta = \theta^*$ which satisfies Eq.(54) by the following update rule

$$q^{k+1} = q^k - J^{-1}e(\theta^k) \tag{55}$$

Where

$$J = \nabla e(\theta^k) \tag{56}$$

After introduce both analytic and non linear optimization the IK solution will be considered now. The explanation of solution for the kinematic chain introduced in the previous section is present. It required minimizing the error between the target transformation matrix and calculated transformation matrix ,the problem can be formularizing as optimization problem as following:
Minimize

$$e(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6,\theta_7) = \begin{bmatrix} |P_x^d - dx| \\ |P_y^d - dy| \\ |P_z^d - dz| \\ a \end{bmatrix} \tag{57}$$

4. OPTIMIZATION PROBLEM

For optimization of a reference joint angle configuration regarding the similarity measure, one can use the Levenberg-Marquardt algorithm. The algorithm, which was first introduced in [10], provides a standard technique for solving nonlinear least squares problems by iteratively converging to a minimum of function expressed as sum of squares. Combinating the Gauss-Newton and the steepest descent method, the algorithm unites the advantages of both methods. Hence,using the LM method, a more robust convergence behavior is achieved at points far from a local minimum, while a faster convergence is gained close at a minimum. Due to its numerical stability, the LM method has also become a popular tool for solving inverse kinematics problems as demonstrated in [11]

5. LEVENBERG-MARQUARDT ALGORITHM

The Levenberg–Marquardt method uses the second-order derivatives of the mean squared error, so that better convergence behavior is observed [12]. It is assumed that function f (θ) and its Jacobean J are known at point [θ]. The aim of the Levenberg–Marquardt algorithm is to compute the variable vector[θ] such that

$$e(\theta) = d_t - F(\theta) \tag{58}$$

is minimum[13]. It offers an efficient technique that combines regularization with second-order training. It capitalizes on the squared error function

$$E(\theta) = (d_t - F(\theta))^2 \tag{59}$$

Where $F(\theta)$ function of variable and d_t target and uses an efficient approximation to the Hessian [14] the Hessian matrix can be approximated as

$$H = J^T J \tag{60}$$

and the gradient can be computed as

$$g = J^T e \tag{61}$$

The Levenberg–Marquardt algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$\theta_{k+1} = \theta_k - [J^T J + \mu I]^{-1} J^T e \tag{62}$$

where J contains first derivatives of the function errors with respect to the variables, I is the identity matrix.When the scalar μ is zero, this is just Newton's method, using the approximate Hessian matrix. When μ is large, this becomes the gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the goal is to shift toward Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step increases the performance function. In this way, the performance function will always be reduced at each iteration of the algorithm[15].In this paper Eq (62) replaced with

$$\theta_{k+1} = \theta_k - [V[S^T S + (\lambda + c)I]V^T]^{-1} J^T e \tag{63}$$

Where
V,S singular value decomposition of J .
c constant is chosen to avoidance the singularity problem,.

6 . EXPERIMENTAL RESULTS

Several experiments were conducted to validate the derived inverse kinematics algorithm. An experiment was conducted to check the correctness of the proposed method derived in Section 4, with reachable positions and orientations.
In t experiment y=13 cm and z=23 cm while x is vary from 25 cm to 30 cm with rotation matrix

$$R = \begin{bmatrix} 0 & 0.7071 & -0.7071 \\ 0 & -0.7071 & -0.7071 \\ -1 & 0 & 0 \end{bmatrix}$$

Table 2 shows the norm error and the computation time in proposed method , LM Algorithm and Sugihara method , experiment results show that the derived inverse kinematics provide lower error with minimum time with respect to other method because the optimization problem with one variable as shown solutions in this experiment.

TABLE. 2 Computation Time and the Error of simulation result of the targets for experiment result with x vary from 25 to 30

x	Error performance index e(Q)			Computation Time (sec)		
	LM Algorithm	Sugihara Method	proposed Method	LM Algorithm	Sugihara Method	proposed Method
25	0.6694	0.6692	0.0005	0.0585	0.0992	0.0139
26	0.6801	0.6801	0.0004	0.0505	0.0995	0.0093
27	0.6937	0.6933	0.0004	0.0515	0.0986	0.0111
28	0.7091	0.7091	0.0003	0.0512	0.0983	0.0115
29	0.7274	0.7272	0.0003	0.0504	0.1000	0.0141
30	0.7477	0.7474	0.0003	0.0516	0.0985	0.0118

Through the simulation it notice that, the error of LM Algorithm and Sugihara method increased with the time because the error at each step will effect on the error of next step ,this will make the error will be accumulative. In the proposed method the difference between actual joints angles and their desired approximately zero, this lead to make the error will not increase with the time. The reason behind the good result of proposed method that, it was near to the perfect solution due to the derivation of equation of human arm and take the advantage of the geometric relations between the equation of human arm. Therefore to find the desired angle only a small displacement in this geometric relations to get the required angles, this make the proposed method provide minimum error and lower time with respect to Sugihara method. The computation time of proposed method is lower than other methods due to using single variable in the search instead the seven variables,

CONCULSION

In this paper, a strategy based on combines an analytical solution with nonlinear optimization algorithm solutions was proposed to solution the IKP. A analytical solutions was used to reduce the size of problem from seven variable of joint angle to one variable nonlinear and optimization algorithm was used to find approximate solution .Combining these method can remedy the weakness of each other, by take the advantage of analytical solutions which provides the correct joint angles for manipulation of the arm end-effectors to any given reachable position and orientation with nonlinear optimization method which find the approximate solution when no exact solution is provide. Sufficient and necessary criteria were provided to determine whether correct solutions existed using forward kinematics. The minimization of the distance between the end effecter and a prescribed Cartesian point is the natural constraint which has been used to reach a solution the proposed method for solving the IKP is that it can be extended to any robotic manipulator, given a set of operation space and joint space parameter values. From the simulation it concluded that, the new method is introduced with minimum error and lower computational time are achieved

REFERENCE

[1] V. Grecu, N. Dumitru, and L. Grecu," Analysis of Human Arm Joints and Extension of the Study to Robot Manipulator", Proceedings of the nternational MultiConference of Engineers and Computer Scientists, Vol 2, March 18 - 20, 2009 pp:1348-1351. .

[2] De Xu, and A. Acosta ," An Analysis of the Inverse Kinematics for a 5-DOF Manipulator" ,International Journal of Automation and Computing vol. 2 ,2005 pp :114-124

[3] C. Hua, C. and Wei-shan, "Wavelet network solution for the inverse kinematics problem in robotic manipulator",Chen Zhejiang Univ Science , Vol. 7 ,No.4 ,2006 PP: 525-529.

[4] Chetan K. and Delbert T. "Kinematics Abstraction for General Manipulator Control", Robotics Research Group, The University of Texas at Austin, JJPRC/MERB 1.206, mail Code R9925, Austin, Texas 78712-1100, 1990 .

[5] T. Sugihara," Solvability-unconcerned Inverse Kinematics based on Levenberg-Marquardt Method", 2009 IEEE-RAS International Conference on Humanoid Robots, , Paris, Dec, 2009 ,pp:555-560

[6] H. Yali and W. Xingsong, "Kinematics analysis of lower extremity exoskeleton", Control and Decision Conference, 2008. CCDC 2008. Chinese, 2-4 July 2008 ,pp: 2837 – 2842 .

[7] M. W. Spong, S. Hutchinson, M. Vidyasagar, "Robot Modeling and Control", JOHN WILEY & SONS, INC, First Edition, 2001, PP:66-67.

[8] K. Abdel-Malek, W. Yu, "Human Placement for Maximum Dexterity", Department of Mechanical Engineering The University of Iowa, Technical report , 2004 pp :1-27.

[9] M. Z. Al-Faiz, Y. I. Al-Mashhadany, "Analytical Solution for Anthropomorphic Limbs Model (IK of Human Arm)", IEEE Symposium on Industrial Electronics and Applications (ISIEA 2009), Malaysia ,October 2009 pp: 684-689.

[10] K. Levenberg, "A Method for the Solution of Certain Non-Linear Problems in Least Squares," The Quarterly of Applied Mathematics, Vol. 2, 1944 ,pp: 164–168

[11] Do, M.; Azad, P.; Asfour, T.," Imitation of human motion on a humanoid robot using non-linear optimization", Humanoid Robots, 2008. 8th IEEE-RAS International Conference pp : 545 – 552 .

[12] Saraoglu, H.M.; Kocan, M. ,"Determination of Blood Glucose Level-Based Breath Analysis by a Quartz Crystal Microbalance Sensor Array", IEEE Sensor Journal, Vol. 10, No. 1, January 2010 pp: 104-109

[13] E.Derya," Detecting variabilities of ECG signals by Lyapunov exponents", Neural Computing & Applications, Springer Vol.18 ,No.7, 2009 pp: 653-662 .

[14] Derrick T. Mirikitani, and Nikolay Nikolaev ," Recursive Bayesian Recurrent Neural Networks for Time-Series Modeling "IEEE Trans on Neural Network, Vol .21, No. 2, Feb 2010 pp: 262-274.

[15] E. Kolay , K. Kayabali and Y. Tasdemir ,"Modeling the slake durability index using regression analysis,artificial neural networks and adaptive neuro-fuzzy methods",Bull Eng Geol Environ Springer Vol.69 ,No.7, 2010 pp:275–286.

Inverse Kinematics Analysis for Manipulator Robot With Wrist Offset Based On the Closed-Form Algorithm

Mohammed Z. Al-Faiz ,MIEEE *mzalfaiz@ieee.org*
Computer Engineering Dept.
Nahrain University
Baghdad, Iraq

Mohammed S.Saleh *mohselman@gmail.com*
Electronics Engineering Dept
Diyala University
Diyala, Iraq

Abstract

This paper presents an algorithm to solve the inverse kinematics for a six degree of freedom (6 DOF) manipulator robot with wrist offset. This type of robot has a complex inverse kinematics, which needs a long time for such calculation. The proposed algorithm starts from find the wrist point by vectors computation then compute the first three joint angles and after that compute the wrist angles by analytic solution. This algorithm is tested for the TQ MA2000 manipulator robot as case study. The obtained results was compared with results of rotational vector algorithm where both algorithms have the same accuracy but the proposed algorithm saving round about 99.6% of the computation time required by the rotational vector algorithm, which leads to used this algorithm in real time robot control.

Keyword: Manipulator Robot, Inverse Kinematics, DOF.

1. INTRODUCTION

A robotic manipulator (arm) consists of a chain of links interconnected by joints. There are typically two types of joints; revolute joint (rotation joint) and prismatic joint (sliding). It would be desirable to control both the position and orientation of a end-effecter or work piece, located at the tip of the manipulator, in its three-dimensional workspace. The end-effecter can be programmed to follow a planned trajectory, provided relationships between joint variables and position and the orientation of the end-effecter are formulated. This task is called the direct kinematics problem [1].

There are some difficulties to solve the inverse kinematics (IK) problem when the kinematics equations are coupled, multiple solutions and singularities exist. There are mainly two types of IK solution: analytical solution and numerical solution. In the first type, the joint variables are solved analytically according to given configuration data. In the second type of solution, the joint variables are obtained based on the numerical techniques. However, they are slow in practical applications [2]. An Artificial Neural Network (ANN) using backpropagation algorithm is applied to solve inverse kinematics problems of industrial robot manipulator. 6R robot manipulator with offset wrist was chosen as industrial robot manipulator because geometric feature of this robot does not allow solving inverse kinematics problems analytically [3].

2. DENAVIT-HARTENBERG (DH) PARAMETERS

A general arm equation that represents the kinematic motion of the manipulator can be obtained by systematically assigning coordinate frames for each link. The parameters associated with joint k are defined with respect to z_{k-1}, which is aligned with the axis of joint k. The first joint parameter, θ_k is called the joint angle. It is the rotation about z_{k-1} needed to make x_{k-1} parallel to x_k. The second joint parameter, d_k is called the joint distance. It is the translation along z_{k-1} needed to make x_{k-1} intersect with x_k see Fig.1. Note that for each joint, it will always be the case that one of these parameters is fixed and the other is variable. The variable joint parameter depends on the type of joint.

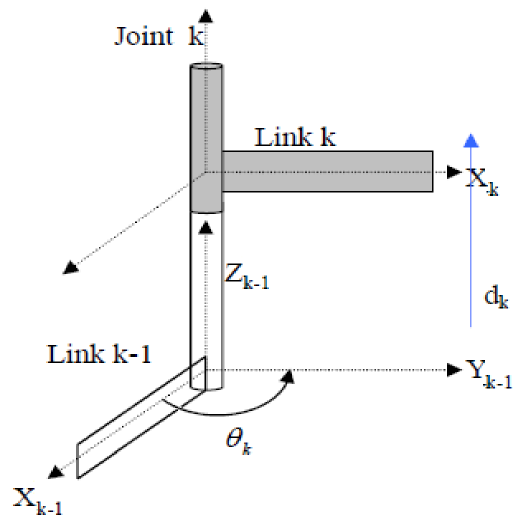


FIGURE 1: Joint angle θ and joint distance d

For instance, for a revolute joint, the joint angle θ_k is variable while the joint distance d_k is fixed while for a prismatic joint, the joint distance d_k is variable and the joint angle θ_k is fixed [1]. See Fig.2.

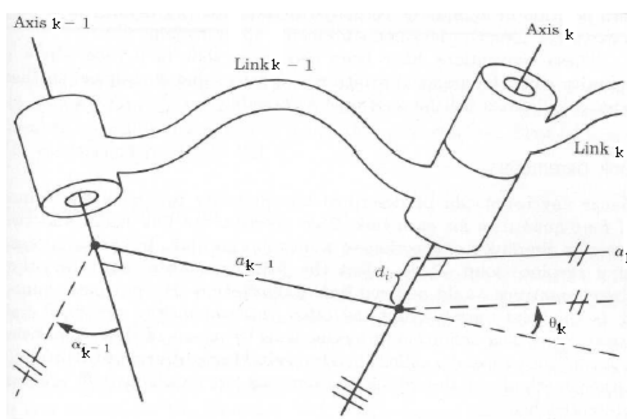


FIGURE 2: DH parameters defined for two links of a serial manipulator

3. FORWARD KINEMATIC ANALYSIS

These industrial robots are basically composed by rigid links, connected in series by joints (normally six joints), having one end fixed (base) and another free to move and perform useful work when properly end-effecter. As with the human arm, robot manipulators use the first three joints (arm) to position the structure and the remaining joints (wrist, composed of three joints in the case of the industrial manipulators) are used to orient the end-effecter. There are five types of arms commonly used by actual industrial robot manipulators: Cartesian, cylindrical, polar, SCARA and revolution [4]. The TQ MA2000 is considered in this work, it has 6 degree of freedom (DOF), one per each joint which means that each joint can be represented by a independent variable θ_n given that each joint is actuated, i.e. in this case each link has its own servo motor acting on it. Robot manipulator is an open kinematic chain whose joint positions can be defined by a vector of six single variables θ_n and the number of joints equals the number of degrees of freedom. Recall that the number of DOF is the number of independent position variables that should be specified in order to completely define a pose for the manipulator. It defines in how many ways the robot is able to move. In practice, to cover the entire 3D workspace with a 3-component orientation vector, six degrees of freedom are sufficient. Table I shows the DH parameters whole set of Kinematic parameters for the TQ MA2000 Robot. See Fig.2 The graphic representation of the TQ MA2000 robot is shown in Fig.3

Link i	Length a_i(cm)	Twist Angle α_i	Offset d_i (cm)	Joint angles θ_i
1	0	π/2	26	θ 1
2	23	0	0	θ2
3	24	0	0	θ3
4	0	π/2	5	θ4
5	0	π/2	4.4	θ5
6	0	0	8	θ6

TABLE 1: Complete DH parameter table for the TQ MA2000 6-DOF manipulator shown in Fig. 2

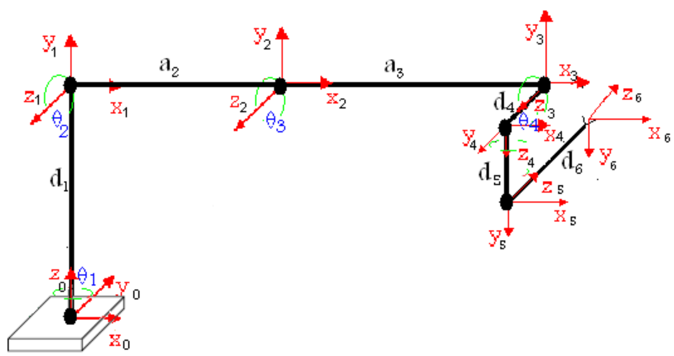


FIGURE 3: Graphic Represent of TQ MA2000 robot

The arm motion can be represented by transformation matrix as follow

Arm motion = $T_{base}^{wrist}(\theta_1,\theta_2,\theta_3)$ (1)

Where, T_i^j represent the transformation matrix from i[th] location to j[th] location. And T_{base}^{wrist} (T_0^3) is the orientation and position of the arm with respect to the base coordinate frame, depend on the first three joint angles $(\theta_1,\theta_2,\theta_3)$. And the wrist motion can be represented by the transformation matrix from wrist to end-effecter as follow.

Wrist motion =$T_{wrist}^{end-effecter}(\theta_4,\theta_5,\theta_6)$ (2)

And $T_{wrist}^{end-effecter}$ (T_3^6) is the orientation and position of the grip of the wrist with respect to the third link coordinate frame, depend on the last three joint angles $(\theta_4,\theta_5,\theta_6)$. Note that $T_5^6(\theta_6)$ maps tool-tip coordinates into roll coordinates, $T_4^5(\theta_5)$ maps roll coordinates into yaw coordinates, and $T_3^4(\theta_4)$ maps yaw coordinates into wrist pitch coordinates. Thus the composite transformation $T_3^6(\theta_4,\theta_5,\theta_6)$ maps end-effecter coordinates into wrist coordinates. Similarly $T_2^3(\theta_3)$ maps wrist coordinates into elbow coordinates, $T_1^2(\theta_2)$ maps elbow coordinates into shoulder coordinates, and $T_0^1(\theta_1)$ maps shoulder coordinates into base coordinates. Thus the composite transformation $T_0^3(\theta_1,\theta_2,\theta_3)$ maps wrist coordinates into base coordinates. The general solution can be expressed as:

[Total motion] = [arm motion] [wrist motion] (3)

Using the D-H method, the end-effector position can be written as: [1].

$$T_{base}^{end-effecter}(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) = T_0^1(\theta_1) T_1^2(\theta_2) T_2^3(\theta_3)\, T_3^4(\theta_4) T_4^5(\theta_5) T_5^6(\theta_6) \quad (4)$$

The homogenous transformation matrix of the T_0^6 which include the rotation matrix (3x3) and position vector (3x1) which given by equation (5) may be given as

$$T_{base}^{end-effecter} = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \quad (5)$$

The upper left 3x3 matrix, R, specifies the orientation of the end-effecter, while the 3 x 1 upper right sub matrix P specifies the position of the end-effecter. Thus, in the direct kinematic solution, for any given value of the joint angles θ, the arm matrix $T_{base}^{end-effecter}$ can be evaluated.

4. INVERSE KINEMATIC ANALYSIS

One of the most fundamental and ever present problems in robotics and computer animation is the inverse kinematics (IK). This problem may be stated as follows: given a desired hand position and orientation (posture), and the forward kinematics map, find the set of all configurations (joint angle vectors) of the robot or animation character that satisfy the forward kinematics map. The IK mapping is in general one to many involves complex inverse trigonometric functions, and for most manipulators and animation figures has no closed form solutions. Extremely fast IK computation is required in computer animation, for real-time applications in fast moving manipulator. Inverse kinematics deals with the problem of finding the required joint angles to produce a certain desired position and orientation of the end-effecter. Finding the inverse kinematics solution for a general manipulator can be a very tricky task. Generally they are non-linear equations. Close-form solutions may not be possible and multiple, infinity, or impossible solutions can arise. Nevertheless, special cases have a closed-form solution and can be solved [5].

5. PROPOSED ALGORITHM FOR IK

The proposed algorithm can be applied directly to robots with shoulder and elbow offset. However, wrist offset makes the hand position dependent on all joint angles rather than just the first four. This requires dealing with position and orientation simultaneously and entails the creation of 6-dimensional workspace which although possible, would substantially increase the off-line processing and storage requirements. Manipulators can often be considered to be composed of two parts. The first 3 joints form a regional structure whose primary purpose is to position the wrist in space. The last 3 joints form the orienting structure whose purpose is to orient the hand or grasped object.

5.1 Decompose the Manipulator at the Wrist

This is the single most important step of the procedure. The spherical wrist makes this possible. Find the position of the wrist point, which subsequently is used to solve for the first 3 joint angles. The wrist joint angles are then solved to give the correct orientation.

Fig.3 show the robot at reset position (all the joint angles equal zero), the proposed algorithm start with specify the required position in (x, y, z) coordinate and orientation angles of pitch, yaw, and roll.(φ_p, φ_y, and φ_r) angles respectively. Then change the wrist joint angles (θ_4, θ_5, and θ_6) from the zero (reset state) to new values that equal the required orientation angles of pitch, yaw, and roll.(φ_p, φ_y, and φ_r) angles respectively, the arm joint angles remain unchanged (at reset state, i.e. equal zero), then compute the foreword kinematics to compute the resulted position of the arm end point (O_4) and position of end-effecter (O_6) for the robot, find the vector between these two points as in equation (6)

$$V = O_6 - O_4 \qquad (6)$$

To compute the suitable position of the arm end point to reach the required position use equation (6) with that the vector from the origin point O_0 to the required position O_r is equal to the vector from origin base frame O_0 to the arm end point position O_4 sum with the vector V which is from O_4 to the required position as in equation (7). as shown in Fig. 4

$$V_{04} = V_{or} - V \qquad (7)$$

Where V is the vector compute by equation (6), and V_{or} is the vector from the origin O_0 to the required position. The position of the arm end point O_4 is the end of the vector V_{04}. A planar two-link manipulator often makes up the last two links of the regional structure.

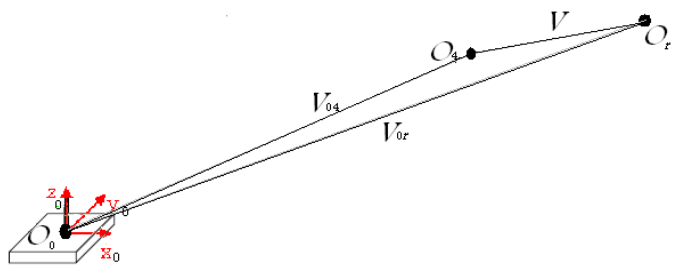


FIGURE 4: Wrist with setting its angles

The link 4 parameters have been included in the regional structure because they locate the wrist point O_4, which will be known relative to O_0. Equation (8) as shown in fig.(5).

$$V_{04} = O_4 - O_0 \qquad (8)$$

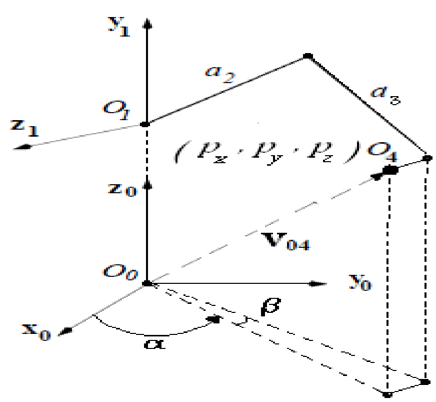


FIGURE 5: Vector from O_0 to O_4

Project the position of the wrist point into the x_0, y_0 plane, i.e., just select the p_x, p_y coordinates of O_4.

5.2 Solution for Θ_1

The wrist offset makes the calculation of θ_1 more complex, because the x_1 axis is no longer in the vertical plane of the upper arm and forearm Figure. When x_1 and the wrist point are projected onto the x_0, y_0 plane, then the angle θ_1 from x_0 to x_1 is no longer the same as the angle from x_0 to the projected wrist point. The projection on the x_0, y_0 plane is redrawn in Figure for clarity. The position of the arm end point at (p_x, p_y, p_z), see Fig.5. Then compute α and β as in equation (9) and (10).

$$\alpha = \text{atan2}(p_y, p_x) \tag{9}$$

$$\beta = \text{asin}\,(d_4/L) \tag{10}$$

Where $L = \sqrt{p_x{}^2 + p_y{}^2}$, and hence for right move then θ_1 will be as follow

$$\theta_1 = \alpha + \beta \tag{11}$$

And for lefty move

$$\theta_1 = (\alpha - \beta) - 180 \tag{12}$$

5.3 Solution For Θ_2 And Θ_3

For each solution of θ_1 find two set of solution for θ_2 and θ_3, elbow up and elbow down, therefore will compute four set of solution of the first three angles that is (left, right, elbow up, elbow down):

$$\theta = \text{atan}\,\left(\frac{pz_0}{r_0}\right) \tag{13}$$

Where $pz_0 = p_z - d_1$, and $r_0 = \sqrt{p_x{}^2 + p_y{}^2}$

The cosine law for link 2 and link 3:

$$\alpha 2 = \pm\text{acos}\,\left(\frac{r^2 + a_2^2 - a_3^2}{2ra_2}\right) \tag{14}$$

$$\alpha 3 = \pm\text{acos}\,\left(\frac{r^2 + a_3^2 - a_2^2}{2ra_3}\right) \tag{15}$$

Where $r = \sqrt{p_x{}^2 + p_y{}^2 + pz_0^2}$ (16)

Then compute θ_2 and θ_3

$$\theta_2 = \theta - \alpha 2 \tag{17}$$

$$\theta_3 = \alpha 2 + \alpha 3 \tag{18}$$

5.4 The Solution of the Last Three Joints $(\theta_4, \theta_5$ and $\theta_6)$

It is possible to obtain the following equations from the forward kinematics problem:

$$\mathbf{T}_0^6 = \mathbf{T}_0^3\,\mathbf{T}_3^6 \tag{19}$$

$$\mathbf{T}_0^6 = \begin{bmatrix} R_0^3 & P_3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_3^6 & P_w \\ 0 & 1 \end{bmatrix} \tag{20}$$

$$T_0^6 = \begin{bmatrix} R_0^3 R_3^6 & R_0^3 P_w + P_a \\ 0 & 1 \end{bmatrix} \quad (21)$$

Where Pw is the wrist position and Pa position of arm end point then the rotation part of equation (5).

$$R = R_0^3 R_3^6 \quad (22)$$

where R is the required orientation of the robot. Translation part of equation (5)

$$P = R_0^3 P_w + P_a \quad (23)$$

Then $R_3^6 = (R_0^3)^{-1} R$ (24)

Where $R = \begin{bmatrix} N_x & O_x & A_x \\ N_y & O_y & A_y \\ N_z & O_z & A_z \end{bmatrix}$, and $R_3^6 = \begin{bmatrix} n_{wx} & o_{wx} & a_{wx} \\ n_{wy} & o_{wy} & a_{wy} \\ n_{wz} & o_{wz} & a_{wz} \end{bmatrix}$ (25)

$$R_3^6 = \begin{bmatrix} C_4 C_5 C_6 + S_4 S_6 & C_4 S_5 & C_4 C_5 S_6 - S_4 C_6 \\ S_4 C_5 C_6 - C_4 S_6 & S_4 S_5 & S_4 C_5 S_6 + C_4 C_6 \\ S_4 C_6 & -C_5 & S_5 S_6 \end{bmatrix} \quad (26)$$

Solving for orientation angles:

$\theta_4 = \operatorname{atan}\left(\frac{o_{wy}}{o_{wx}}\right)$, $\theta_5 = \operatorname{atan}\left(\frac{c_4 o_{wx} + s_4 o_{wy}}{-o_{wz}}\right)$, and $\theta_6 = \operatorname{atan}\left(\frac{s_4 n_{wx} - c_4 n_{wy}}{s_4 a_{wy} - c_4 a_{wx}}\right)$ (27)

5.5 Case Study

In [6] they use the relative orientation representation by the rotation vector are based on the Euler theorem which states that (a displacement of a rigid body with one fixed point can be described as a rotation about some axis). And applied this algorithm to solve the inverse kinematics for the TQ MA2000 manipulator robot to compute the sets of angles. Consider the required position is (-25, -10, 50) cm as (x, y, z) with respect to the base coordinate frame. The orientation of the gripper rotates (180o, 225o, 135o) degree as $(\varphi_p, \varphi_y, \varphi_r)$ the rotating angle of the gripper about pitch , yaw, and roll rotating axis respectively. There are four set of the joint variables (solution) of the inverse kinematics of the robot.

Set = [θ1 , θ2 , θ3 , θ4 , θ5 , θ6]
Set 1 = [-162.6436, -7.8977, 79.7788, 108.1189, 27.6436, 135]
Set 2 = [-162.6436, 75.8702, -79.7788, -176.0914, 27.6436, 135]
Set 3 = [-1.2296,187.8977, -79.7788, 71.8811, -133.7704, 135]
Set 4 = [-1.2296, 104.1298, 79.7788, 356.0914, -133.7704, 135]

The structure of the robot for all solution sets(1, 2, 3, 4) are shown in Fig.6 (a- right elbow down, b- right elbow up, c- lefty elbow down, and d- lefty elbow up) respectively.
The proposed algorithm is solving the same case steady and finds the same results with the same accuracy. By using DELL laptop computer with Central Processing Unit (CPU) Intel® Core™ 2 Duo with 2.2GHz; the computation time for the proposed algorithm is equal 0.016 second and the computation time for rotation vectors is equal 3.984 second these less computation time is suitable for used the proposed algorithm with real time manipulator robot control system.

6. CONCLUSION

By camper this algorithm with the vector rotation algorithm that is used for solve the inverse kinematics of the same manipulator robot [6]. They have same accuracy but the computation time by proposed method is less than that for the first method to compute all the angles. The computation time for the solving of inverse kinematics of the manipulator robot by the proposed algorithm is about 0.4% of the computation time that required by use the rotational vector algorithm, therefore this algorithm is best to used for real time manipulator robot control system.

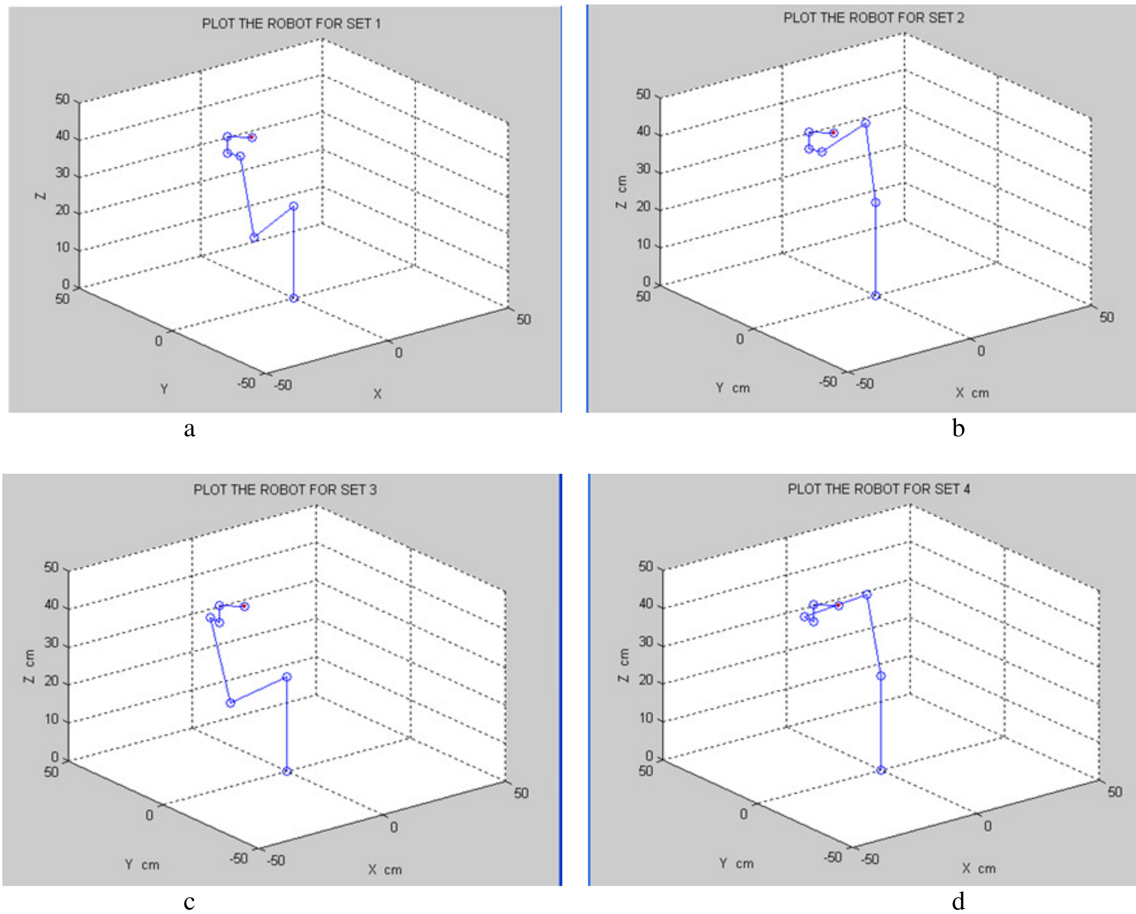


FIGURE 6: Drawing of the robot according to set of solution a- set 1 b-set 2 c- set 3 d- set 4

7. REFERENCES

[1] A. N. Aljaw, A. S. Balamesh, T. D. Almatrafi and M. Akyurt "Symbolic Modeling Of Robotic Manipulators" the 6th Saudi engineering conference, KFUPM, vol. 4, December 2002

[2] S. Kucuk and Z. Bingul "Inverse Kinematics Solutions of Fundamental Robot Manipulators with Offset Wrist" in Proc. International Conference on Mechatronics, 2005, pp 197-202.

[3] Z. Bingul, H. M. Ertunc and C. Oysu "Comparison of Inverse Kinematics Solutions Using Neural Network for 6R Robot Maipulator with Offset" In ICSC Congress on Computational intelligence, Turkey, 15-17 December 2005, pp. 5–10

[4] J.N. Pires. Industrial Robots Programming: Building Applications for The Factories of The Future. Springer Science Business Media, 2007,pp 36

[5] C.F. Rose, P-P. J. Sloan and M.F. Cohen, “Artist-Directed Inverse Kinematics Using Radial Basis Function Interpolation” EUROGRAPHICS, vol.20, no. 3, pp. 1-12, 2001

[6] M. Z. Al-Faiz, M. Z. Othman, and B. B. Al-Bahri ” An Algorithm to Solve the Inverse Kinematics Problem of a Robotic Manipulator Based on Rotation Vectors” 3rd IEEE-FCC 2006, Bahrain

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. Started with Volume 2, 2011, IJRA appear with more focused issues. Besides normal publications, IJRA intends to organize 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: 2 - **Issue: 5**

i. Paper Submission: October 31, 2011 **ii. Author Notification:** November 30, 2011

iii. Issue Publication: December 2011

CONTACT INFORMATION

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

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