International Journal of Hybrid Information Technology
Vol.8, No.1 (2015), pp.121-132
http://dx.doi.org/10.14257/ijhit.2015.8.1.10

Novel Intelligent-Based Gravity Control for Industrial Robot Arm

Abdol Reza Babakhani, Esmail Moradi, Mohammad Salooki and Rouholla Fakhraie

Department of Mechanical Engineering, Delvar Branch, Islamic Azad University, Delvar, Iran
rezababakhani2014@gmail.com

Abstract

According to this research, a new intelligent-based parallel gravity controller is proposed for industrial robot arm. All model-base conventional controllers are related to dynamic model of systems especially in uncertain systems. To solve this challenge intelligent theory and partly linear methodology are play important role. Fuzzy logic theory is used to estimate the nonlinear dynamic part of robot manipulator and to improve the fuzzy performance PD gravity is used. To estimate the 2 DOF robot manipulator system's dynamic, proportional plus modified derivative with 7 rules Mamdani inference system is design and applied to modified PD gravity methodology. The proportional coefficient of controller is tuned by new methodology in limitation uncertainties. The results demonstrate that the proposed controller is a partly model-free controllers which works well in certain and partly uncertain system.

Keywords: *fuzzy logic theory, modified PD control, PD gravity technique, online tuning, robot manipulator*

1. Introduction

Automatic control has played an important role in advance science and engineering and its extreme importance in many industrial applications, *i.e.*, aerospace, mechanical engineering and robotic systems. The first significant work in automatic control was James Watt's centrifugal governor for the speed control in motor engine in eighteenth century[1-2]. There are several methods for controlling a robot manipulator, which all of them follow two common goals, namely, hardware/software implementation and acceptable performance. However, the mechanical design of robot manipulator is very important to select the best controller but in general two types schemes can be presented, namely, a joint space control schemes and an operation space control schemes[3]. Joint space and operational space control are closed loop controllers which they have been used to provide robustness and rejection of disturbance effect. The main target in joint space controller is design a feedback controller that allows the actual motion ($q_a(t)$) tracking of the desired motion ($q_d(t)$). This control problem is classified into two main groups. Firstly, transformation the desired motion $X_d(t)$ to joint variable $q_d(t)$ by inverse kinematics of robot manipulators [4]. The main target in operational space controller is to design a feedback controller to allow the actual end-effector motion $X_a(t)$ to track the desired endeffector motion $X_d(t)$ [5].

Controller (control system) is a device which can sense information from linear or nonlinear system (*e.g.*, robot arm) to improve the systems performance and the immune system behavior. In feedback control system considering that there are many disturbances and also variable dynamic parameters something that is really necessary is keeping plant variables close to the desired value. Feedback control system development is the most important thing in many different fields of safety engineering. The main targets in design control systems are safety stability, good disturbance rejection to reach the best safety, and small tracking error [6-7]. At present, in some applications robot arms are used in

ISSN: 1738-9968 IJHIT

unknown and unstructured environment, therefore strong mathematical tools used in new control methodologies to design nonlinear robust controller with an acceptable safety performance (*e.g.*, minimum error, good trajectory, disturbance rejection). Advanced control techniques such as sliding mode controller, feedback linearization methodology, adaptive and robust have been applied to the control of numerous single- axis machines and robotic manipulators. One of the most important partly nonlinear safety controllers is linear plus nonlinear methodology which is used in nonlinear certain and partly uncertain systems. This methodology is used in wide range areas such as in safety control access process; in aerospace applications and in IC engines because this methodology can solve some main challenging topics in safety control access such as resistivity to the external disturbance and stability. Even though, this methodology is used in wide range areas but, PD gravity method has an important drawbacks beside uncertain system in presence of external disturbance. This problem is solved by applied soft computing theory [8]. Although the fuzzy-logic control is not a new technique, its application in this current research is considered to be novel since it aimed for an automated dynamic-less response rather than for the traditional objective of uncertainties compensation[9-10]. The intelligent tracking control by the fuzzy-logic technique provides a cost-and-time efficient control implementation due to the automated dynamic-less input. This in turn would further inspire multi-uncertainties testing for continuum robot [11]. Fuzzy logic theory is used to estimate the system's dynamics. To estimate the continuum robot manipulator's dynamic of system, Mamdani fuzzy inference system is design and applied to inverse dynamic methodology [12-15].

This method is based on design modified fuzzy PD gravity controller and resolves the uncertainty term by fuzzy logic methodology. To have the best performance modified PD method based on boundary derivative methodology is design and to tune the fuzzy logic gain updating factor as well as improve the output performance the iteration algorithm based on Gradient Descent Optimal Algorithm (GDOA) is introduced. The gain updating factor of this controller is adjusted off line depending on the iterations.

This paper is organized as follows; Section 2, is served as an introduction to PD gravity method and its application to control of robot manipulator, dynamic of robot manipulator and fuzzy inference system. Part 3, introduces and describes the methodology algorithm. Section 4 presents the simulation results and discussion of this algorithm applied to the robot manipulator and the final section describe the conclusion.

2. Theory

Dynamic Formulation of 2-DOF Robot Manipulator: Dynamic modeling of robot manipulators is used to describe the behavior of robot manipulator such as linear or nonlinear dynamic behavior, design of model based controller such as pure sliding mode controller which design this controller is based on nonlinear dynamic equations, and for simulation. The dynamic modeling describes 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 behavior of system[1]. The equation of an *n-DOF* robot manipulator governed by the following equation [1, 4]:

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

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:

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

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:

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

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

Forward Kinematics of robot: Calculate the relationship between rigid bodies and end-effector without any forces is called Robot manipulator Kinematics. Study of this part is pivotal to calculate accurate dynamic part, to design with an acceptable performance controller, and in real situations and practical applications. As expected the study of manipulator kinematics is divided into two main parts: forward and inverse kinematics. Forward kinematics has been used to find the position and orientation of task (end-effector) frame when angles and/or displacement of joints are known. Inverse kinematics has been used to find possible joints variable (displacements and angles) when all position and orientation of end-effector be active [1].
The main target in forward kinematics is calculating the following function:

$$\boldsymbol{\Psi}(\boldsymbol{X}, \boldsymbol{q}) = \boldsymbol{0} \tag{4}$$

Where $\Psi(.) \in R^n$ is a nonlinear vector function, $X = [X_1, X_2, \dots \dots, X_l]^T$ is the vector of task space variables which generally endeffector has six task space variables, three position and three orientation, $q = [q_1, q_2, \dots., q_n]^T$ is a vector of angles or displacement, and finally n is the number of actuated joints. The Denavit-Hartenberg (D-H) convention is a method of drawing robot manipulators free body diagrams. Denvit-Hartenberg (D-H) convention study is necessary to calculate forward kinematics in serial robot manipulator. The first step to calculate the serial link robot manipulator forward kinematics is link description; the second step is finding the D-H convention after the frame attachment and finally finds the forward kinematics. Forward kinematics is a 4×4 matrix which 3×3 of them shows the rotation matrix, 3×1 of them is shown the position vector and last four cells are scaling factor [1, 6]. Singularity is a location in the robot manipulator's workspace which the robot manipulator loses one or more degrees of freedom in Cartesian space. Singularities are one of the most important challenges in inverse kinematics which Cheng *et al.*, have proposed a method to solve this problem [12]. A systematic Forward Kinematics of robot manipulator solution is the main target of this part. The first step to compute Forward Kinematics (F.K) of robot manipulator is finding the standard D-H parameters. The following steps show the systematic derivation of the standard D-H parameters.

1. Locate the robot arm
2. Label joints
3. Determine joint rotation or translation (θ or d)
4. Setup base coordinate frames.
5. Setup joints coordinate frames.
6. Determineα_i, thatα_i, link twist, is the angle between Z_i and Z_{i+1} about an X_i.
7. Determine d_i and a_i , that a_i, link length, is the distance between Z_i and Z_{i+1} along X_i. d_i, offset, is the distance between X_{i-1} and X_i along Z_i axis.
8. Fill up the D-H parameters table.

The second step to compute Forward kinematics for robot manipulator is finding the rotation matrix (R_n^0). The rotation matrix from$\{F_i\}$ to $\{F_{i-1}\}$ is given by the following equation;

$$\boldsymbol{R_i^{i-1} = U_{i(\theta_i)} V_{i(\alpha_i)}} \tag{5}$$

Where $U_{i(\theta_i)}$ is given by the following equation [1];

$$\boldsymbol{U_{i(\theta_i)} = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i) & 0 \\ sin(\theta_i) & cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}} \tag{6}$$

and $V_{i(\alpha_i)}$ is given by the following equation [1];

$$\boldsymbol{V_{i(\theta_i)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha_i) & -sin(\alpha_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) \end{bmatrix}} \tag{7}$$

So (R_n^0) is given by [1]

$$\boldsymbol{R_n^0 = (U_1 V_1)(U_2 V_2) \dots \dots \dots (U_n V_n)} \tag{8}$$

The third step to compute the forward kinematics for robot manipulator is finding the displacement vector d_n^0, that it can be calculated by the following equation [1]

$$\boldsymbol{d_n^0 = (U_1 S_1) + (U_1 V_1)(U_2 S_2) + \cdots + (U_1 V_1)(U_2 V_2) \dots . (U_{n-1} V_{n-1})(U_n S_n)} \tag{9}$$

The forth step to compute the forward kinematics for robot manipulator is calculate the transformation ${}_n^0T$ by the following formulation [1]

$$\boldsymbol{{}_n^0T = {}_1^0T . {}_2^1T . {}_3^2T \dots \dots . {}_n^{n-1}T = \begin{bmatrix} R_n^0 & d_n^0 \\ 0 & 1 \end{bmatrix}} \tag{10}$$

Design PID Controller: Design of a linear methodology to control of continuum robot manipulator was very straight forward. Since there was an output from the torque model, this means that there would be two inputs into the PID controller. Similarly, the outputs of the controller result from the two control inputs of the torque signal. In a typical PID method, the controller corrects the error between the desired input value and the measured value. Since the actual position is the measured signal.

$$\boldsymbol{e(t) = \theta_a(t) - \theta_d(t)} \tag{11}$$

$$\boldsymbol{U_{PID} = K_{p_a} e + K_{V_a} \dot{e} + K_I \sum e} \tag{12}$$

The model-free control strategy is based on the assumption that the joints of the manipulators are all independent and the system can be decoupled into a group of single-axis control systems [2-4]. Therefore, the kinematic control method always results in a group of individual controllers, each for an active joint of the manipulator. With the independent joint assumption, no a priori knowledge of robot manipulator dynamics is needed in the kinematic controller design, so the complex computation of its dynamics can be avoided and the controller design can be greatly simplified. This is suitable for real-time control applications when powerful processors, which can execute complex algorithms rapidly, are not accessible. However, since joints coupling is neglected, control performance degrades as operating speed increases and a manipulator controlled in this way is only appropriate for relatively slow motion. The fast motion requirement results in even higher dynamic coupling between the various robot joints, which cannot be compensated for by a standard robot controller such as PID, and hence model-based control becomes the alternative.

Design Computed Torque Controller: Computed torque controller (CTC) is a powerful nonlinear method, which it is widely used in control of robot manipulator. It is based on feedback linearization and computes the required results using the nonlinear feedback control law. This controller works very well when all dynamic and physical parameters are known. In practice, most of physical systems parameters are unknown or time variant, therefore, CTC must to mixed to the other methodology to compensate dynamic equation of robot manipulator. VIVAS and MOSQUERA have proposed a computed torque controller for tracking response in uncertain environment. They compared this method and predictive methodology, however both controllers have been used in feedback linearization, but predictive strategy gives better result as a performance in above research. Selecting proportional-plus-derivative (PD) feedback for N(t) results in the PD-CTC ;

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\left(\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{K}_{\boldsymbol{v}}\dot{\boldsymbol{e}} + \boldsymbol{K}_{\boldsymbol{p}}\boldsymbol{e}\right) + \boldsymbol{N}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{13}$$

According to the linear system theory, convergence of the tracking error to zero is guaranteed. Where $\boldsymbol{K}_{\boldsymbol{p}}$ and $\boldsymbol{K}_{\boldsymbol{v}}$ are the controller gains.

Fuzzy Logic Theory: Fuzzy set theory introduced by Zadeh proposed that set membership was the key to a decision making process when that process faced uncertainty. Zadeh introduced fuzzy set theory as a modified set theory in which an individual element could possess a degree of membership which could range over a continuum rather than being defined as either 0 or 1. Fuzzy sets may imply to the uninitiated a lack of preciseness or clarity but, in reality, the use of linguistic terms provides a powerful tool for describing many real world properties.

A classical crisp set is a collection of distinct object. The concept of a set has become one of the most fundamental notions of mathematics. So-called set theory was founded by German mathematician George Cantor (1845-1918). It is defined in such a way as to dichotomize the elements of a given universe of discourse into two groups: members and non-members. Finally a crisp set can be defined by the characteristic equation. Let U be the universe of discourse. The characteristic function of a crisp set A in U takes its values in {0, 1}. Supposed that U is the universe of discourse and x is the element of U, therefore, a crisp set can be defined as a set which consists of different elements (x) will all or no membership in a set.

The membership function which is often used in practical applications includes

- triangular form
- trapezoidal form
- bell-shaped form
- Gaussian form.

Linguistic variable can open a wide area to use of fuzzy logic theory in many applications. In a natural artificial language all numbers replaced by words or sentences. $If - then$ Rule statements are used to formulate the condition statements in fuzzy logic. $If - then$ rules have three parts

- fuzzify inputs

- fuzzy operator
- implication

where fuzzification is used to transformation crisp to fuzzy. In this part, inputs the fuzzy statements in the antecedent replaced by the degree of membership. Fuzzy operator is divided into two groups:

- And operator
- Or operator

Fuzzy implication method used in consequent of fuzzy rule to replaced by the degree of membership.

The fuzzy inference engine offers a mechanism for transferring the rule base in fuzzy set. Fuzzy inference engine has 2 main methods:

- Mamdani method
- Sugeno method

Mamdani method is one of the common fuzzy inference systems and he designed one of the first fuzzy controllers to control of system engine. Any types of fuzzy inference system are divided into four major steps

- Fuzzification
- rule evaluation
- aggregation of the rule outputs
- defuzzification

The following definition shows the Mamdani and Sugeno fuzzy rule base

$$\begin{aligned}&\boldsymbol{Mamdani\quad F.R^{1}: if\quad x\ is\ A\ and\ y\ is\ B\quad then\quad z\ is\ C}\\&\boldsymbol{Sugeno\quad F.R^{1}: if\quad x\ is\ A\ and\ y\ is\ B\quad then\quad f(x,y)\ is\ C}\end{aligned} \qquad \textbf{(14)}$$

The aggregation is used to calculate the output fuzzy set and several methodologies can be used in fuzzy logic controller aggregation are;

- Max-Min aggregation
- Sum-Min aggregation
- Max-bounded product
- Max-drastic product
- Max-bounded sum
- Max-algebraic sum
- Min-max

Max-min aggregation defined as below

$$\boldsymbol{\mu_{U}(x_k,y_k,U)=\mu_{\cup_{i=1}^{r}FR^{i}}(x_k,y_k,U)=\max\left\{\min_{i=1}^{r}\left[\mu_{R_{pq}}(x_k,y_k),\mu_{p_m}(U)\right]\right\}} \qquad \textbf{(15)}$$

The Sum-min aggregation defined as below

$$\boldsymbol{\mu_{U}(x_k,y_k,U)=\mu_{\cup_{i=1}^{r}FR^{i}}(x_k,y_k,U)=\sum\min_{i=1}^{r}\left[\mu_{R_{pq}}(x_k,y_k),\mu_{p_m}(U)\right]} \qquad \textbf{(16)}$$

where r is the number of fuzzy rules activated by x_k and y_k and also $\mu_{\cup_{i=1}^{r}FR^{i}}(x_k,y_k,U)$ is a fuzzy interpretation of $i-th$ rule.

Defuzzification is the final part to design fuzzy logic controller which it is used to transform fuzzy set to crisp set. Consequently defuzzification's input is the aggregate output and the defuzzification's output is a crisp number. Two most popular methods in defuzzification are:

- Centre of gravity method (COG)
- Centre of area method (COA)

The COG method used the following equation to calculate the defuzzification

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

The COA method used the following equation to calculate the defuzzification

$$COA(x_k, y_k) = \frac{\sum_i U_i \cdot \mu_u(x_k, y_k, U_i)}{\sum_i \mu_U \cdot (x_k, y_k, U_i)} \quad (18)$$

Where $COG(x_k, y_k)$ and $COA(x_k, y_k)$ illustrates the crisp value of defuzzification output, $U_i \in U$ is discrete element of an output of the fuzzy set, $\mu_U \cdot (x_k, y_k, U_i)$ is the fuzzy set membership function, and r is the number of fuzzy rules. Figure 1 shows the block diagram of fuzzy logic control methodology based on two inputs and one output.

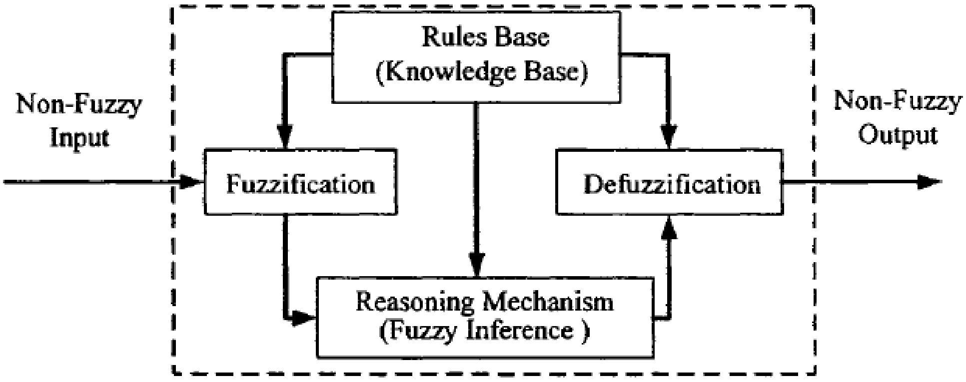

Figure 1. Block Diagram of Fuzzy Logic Control Methodology

Gradient Descent Algorithm: GDA is initialized with a random population of solutions in N-dimensional problem space, the i th particle changes and updates its position and velocity according to the following formula:

$$V_{id} = w \times (V_{id} + C_1 \times rand_1 * (P_{id} - X_{id}) + C_2 \times rand_2 \times (P_{gd} - X_{id})) \quad (19)$$

Where X_{id} is calculated by

$$X_{id} = X_{id} + V_{id} \quad (20)$$

Where V_{id} is the inertia weight implies the speed of the particle moving along the dimensions in a problem space. C_1 and C_2 are acceleration parameters, called the cognitive and social parameters; $rand_1$ and $rand_2$ are functions that create random values in the range of (0, 1). X_{id} is the particle's current location; P_{id} (personal best) is the location of the particle experienced its personal best fitness value; $\boldsymbol{P_{gd}}$ (global best) is the location of the particle experienced the highest best fitness value in entire population; $\boldsymbol{d}$ is the number of dimensions of the problem space; $\boldsymbol{W}$ is the momentum part of the particle or constriction coefficient and it is calculated based on the following equation;

$$\boldsymbol{W = 2/\,(2 - \varphi - \sqrt{\varphi^2 - 4\varphi})} \tag{21}$$

$$\boldsymbol{\varphi = C_1 + C_2} \ , \ \boldsymbol{\varphi} > 4 \tag{22}$$

$$\boldsymbol{X_i(t+1)} = \begin{cases} \boldsymbol{X_i(t)} & \boldsymbol{f(P_d(t+1)) \le X_i(t)} \\ \boldsymbol{f(P_d(t+1))} & \boldsymbol{f(P_d(t+1)) > X_i(t)} \end{cases} \tag{23}$$

In GDO, the knowledge of each particle will not be substituted until the particle meets a new position vector with a higher competence value than the currently recorded value in its memory. So the methodology which is applied in this paper in order to select the best values for these deterministic coefficients to accomplish high performance control is the Gradient Descent Optimization algorithm. This algorithm tunes the gains and determines the appropriate values for these parameters in harmony with the system which was introduced in rear part.

3. Methodology

According to model-base computed torque controller, this controller is high quality nonlinear controller in certain system. Most of robot manipulator’s applications work in uncertain and unknown area, therefore conventional CTC cannot support these systems. Because pure CTC has equivalent problem in uncertain area. To solve this challenge computed like controller can be introduced but in this design we also need to know minimum information about our systems. In this research the nonlinear equivalent dynamic (equivalent part) formulation problem in uncertain system is eliminated based on PD Gravity and estimated by fuzzy logic theorem. Fuzzy logic theory is used to estimate of the equivalent part in fuzzy partly linear PD Gravity controller. In this method fuzzy logic theorem is parallel applied to PD Gravity controller to remove the nonlinear equivalent part which it is based on nonlinear dynamic formulation. To achieve this goal, the dynamic equivalent part of pure CTC is modeled by Mamdani’s performance/ error-based fuzzy logic methodology. This technique was employed to obtain the desired control behavior with a number of information about dynamic model of system and a parallel fuzzy control was applied to reinforce system performance. Equivalent part is based on robot manipulator’s dynamic formulation which these formulations are nonlinear; MIMO and some of them are unknown. To solve the challenge of CTC based on nonlinear dynamic formulation this research is focused on eliminate the nonlinear equivalent formulation. In this method; dynamic nonlinear equivalent part is replaced by performance/error-based fuzzy logic controller plus gravity term. In this method error based Mamdani’s fuzzy inference system has considered with two inputs, one output and totally 49 rules instead of the dynamic equivalent part. In a typical PD method, the controller corrects the error between the desired input value and the measured value. Since the actual position is the measured signal. The derivative part of PD methodology is worked based on change of error and the derivative coefficient. In this research the modified PD is used based on boundary derivative part.

$$\boldsymbol{X_i(t+1)} = \begin{cases} \boldsymbol{X_i(t)} & \boldsymbol{f\big(P_d(t+1)\big) \le X_i(t)} \\ \boldsymbol{f\big(P_d(t+1)\big)} & \boldsymbol{f\big(P_d(t+1)\big) > X_i(t)} \end{cases} \tag{23}$$

$$\boldsymbol{\dot{e}(t)} \triangleq \boldsymbol{(\frac{KS}{100\,S+A}) \times e(t)} \tag{24}$$

$$\boldsymbol{U_{PD} = K_{p_a} e + K_{V_a} \dot{e}} \tag{25}$$

This is suitable for real-time control applications when powerful processors, which can execute complex algorithms rapidly, are not accessible. The result of modified PD method shows the power of disturbance rejection in this methodology.

For CTC and parallel fuzzy inverse dynamic controller plus gravity applications the system performance is sensitive to the linear PD coefficient(K_p & K_v). For instance, if large value of K_p & K_v are chosen the response is very fast the system is unstable and conversely, if small value of K_p & K_v are considered the response of system is very slow but system is stable. Therefore to have a good response, compute the best value linear coefficients are very important. The on-line tuning parallel fuzzy error-based PD Gravity output is written;

$$\boldsymbol{\hat{\tau} = \tau_{eq_{fuzzy}} + \tau_{partly\,linear} + G(q) = \sum_{l=1}^{M} \theta^T \zeta(x) + M(q).\Big[\ddot{q}_d + K_{p_a} e + \Big(\frac{KS}{100\,S+A}\Big) \times e(t) K_{V_a}\Big] + G(q)} \tag{26}$$

Based on fuzzy logic methodology

$$\boldsymbol{f(x) = U_{fuzzy} = \sum_{l=1}^{M} \theta^T \zeta(x)} \tag{27}$$

where θ^T is adjustable parameter (gain updating factor) and $\zeta(x)$ is defined by;

$$\boldsymbol{\zeta(x) = \frac{\sum_i \mu(x_i) x_i}{\sum_i \mu(x_i)}} \tag{28}$$

Where $\mu(x_i)$ is membership function. τ_{fuzzy} is defined as follows;

$$\boldsymbol{\tau_{fuzzy} = \sum_{l=1}^{M} \theta^T \zeta(x) = [(B+C)]} \tag{29}$$

Adaption law in this methodology is used to adjust the linear PD coefficient and gain updating factors. Linear error-based tuning part is a supervisory controller based on the following formulation methodology. This controller has three inputs namely; error (e), change of error ($\dot{e}$) and the integral of error ($\sum e$) and an output namely; gain updating factor(α). As a summary design a linear error-based tuning is based on the following formulation:

$$\alpha = K.e + \dot{e} + \frac{(K)^2}{2}\sum e \tag{30}$$

$$\boldsymbol{K_{p_a}} = \alpha.K_p \Rightarrow \boldsymbol{K_{p_a}} = (K.e + \dot{e} + \frac{(K)^2}{2}\sum e)K_p$$

$$\boldsymbol{K_{v_a}} = \alpha.K_v \Rightarrow \boldsymbol{K_{v_a}} = (K.e + \dot{e} + \frac{(K)^2}{2}\sum e)K_v$$

Where (α) is gain updating factor, $(\sum e)$ is the integral of error, $(\dot{e})$ is change of error, (e) is error and K is a coefficient.

4. Results and Discussion

Modified PD fuzzy PD Gravity technique was tested to Step response trajectory. In this simulation, to control position of continuum robot the first, second, and third joints are moved from home to the final position without and with external disturbance. The simulation was implemented in MATLAB/SIMULINK environment. These systems are tested by band limited white noise with a predefined 40% of relative to the input signal amplitude. This type of noise is used to external disturbance in continuous and hybrid systems and applied to nonlinear dynamic of these controllers.

Tracking performances: In proposed controller; the performance is depended on the gain updating factor (K) and PD coefficient. These coefficients are computed by gradient descent optimization in the first time and on-line tune based on adaptive methodology. Figure 2 shows tracking performance in PD gravity and proposed method.

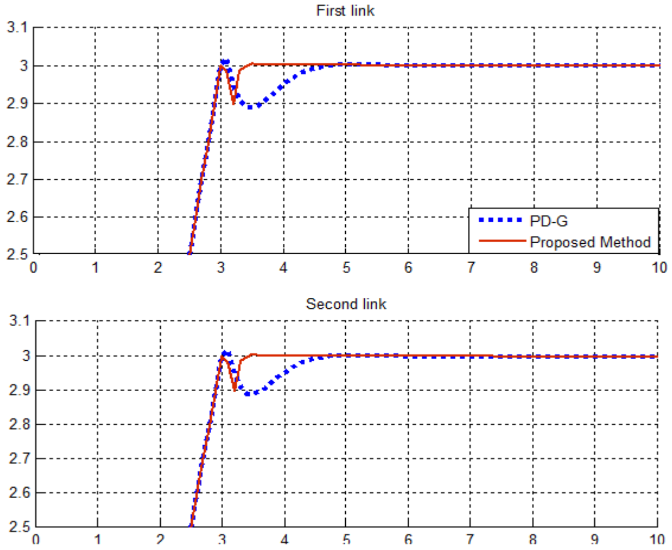

Figure 2. Proposed Method vs. PD Gravity

Disturbance rejection: Figure 3 shows the power disturbance elimination in proposed method and PD gravity in presence of disturbance for step trajectory. The disturbance rejection is used to test the robustness comparisons of these controllers for step trajectory. A band limited white noise with predefined of 40% the power of input signal value is applied to the step trajectory. It found fairly fluctuations in PD gravity trajectory responses but proposed method is more robust.

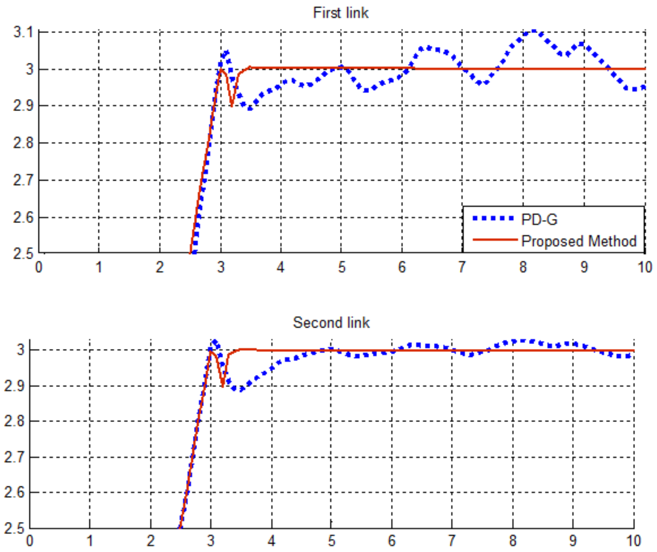

Figure 3. Proposed Method vs. PD Gravity: in Presence of 40% Disturbance

5. Conclusion

This research is used to solve the nonlinear dynamic formulation problem in conventional computed torque controller based on two methods:

- Artificial intelligence base, based on fuzzy logic
- Eliminate the equivalent problem in CTC based on design PD plus gravity method

Proposed fuzzy modified PD Gravity online tuning has shown growing popularity in both industry and academia. PD gravity provides us an effective tool to control nonlinear systems. Fuzzy inference controller is one of the industrial nonlinear controllers in certain systems. Mixed performance criteria have been used to design the controller and the relative weighting matrices of these criteria can be achieved by choosing different coefficient matrices. According to the simulation; proposed method can highly reject the disturbance in presence of 40% disturbance.

References

[1] G. I. Vachtsevanos, K. Davey and K. M. Lee, “Development of a Novel Intelligent Robotic Manipulator”, IEEE Control System Magazine, (**1987**), pp. 9-15.

[2] K. Davey, G. I. Vachtsevanos and R. Powers, “An analysis of Fields and Torques in Spherical Induction Motors”, IEE Transactions on Magnetics, vol. MAG-23, (**1987**), pp. 273-282.

[3] A. Foggia, E. Oliver and F. Chappuis, “New Three Degrees of Freedom Electromagnetic Actuator”, Conference Record -IAS Annual Meeting, vol. 35, New York, (**1988**).

[4] K. M. Lee, G. Vachtsevanos and C-K. Kwan, “Development of a Spherical Wrist Stepper Motor”, Proceedings of the 1988 IEEE International Conference on Robotics and Automation, Philadelphia, PA, April 26-29.

[5] K. M. Lee and I. Pei, “Kinematic Analysis of a Three Degree-of-Freedom Spherical Wrist Actuator”, The Fifth International Conference on Advanced Robotics, Italy, (**1991**).

[6] I. Wang, G. Jewel and D. Howe, “Modelling of a Novel Spherical Pennanent Magnet Actuator”< Proceedings of IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, (**1997**), pp. 1190-1195.

[7] I. Wang, G. Jewel and D. Howe, “Analysis, Design and Control of a Novel Spherical Pennanent Magnet Actuator”, IEE Proceedings on Electrical Power Applications, vol. 154, no. 1, (**1998**).

[8] G. S. Chirikjian and D. Stein, “Kinematic Design and Commutation of a Spherical Stepper Motor”, IEEEIASME Transactions on Mechatronics, vol. 4, no. 4, Piscataway, New Jersey, (**1999**) December, pp. 342-353.

[9] K. Kahlen and R. W. De Doncker, “CW'l'ent Regulators for Multi-phase Pennanent Magnet Spherical Machines”, Industry Applications Conference Record of the 2000 IEEE, vol. 3, (**2000**), pp. 2011-2016.

[10] K. M. Lee, I. Pei and U. Gilboa, "On the Development of a Spherical Wrist Actuator", Proceedings of the 16th NSF Conference on Manufacturing Systems Research, Tempe AZ, (**1990**) January 8-12.

[11] C. Yang and Y. S. Back, "Design and Control of the 3-dcgn:es of freedom actuator by Controlling the Electromagnetic Force", IEEE Transactions on Magnetics, (**1999**) May, pp. 3607-3609.

[12] F. Piltan, S. Emamzadeh, Z. Hivand, F. Shahriyari and M. Mirazaei, "PUMA-560 Robot Manipulator Position Sliding Mode Control Methods Using MATLAB/SIMULINK and Their Integration into Graduate/Undergraduate Nonlinear Control, Robotics and MATLAB Courses", International Journal of Robotics and Automation, vol. 3, no. 3, (**2012**), pp. 106-150.

[13] F. Piltan, M. H. Yarmahmoudi, M. Shamsodini, E. Mazlomian and A. Hosainpour, "PUMA-560 Robot Manipulator Position Computed Torque Control Methods Using MATLAB/SIMULINK and Their Integration into Graduate Nonlinear Control and MATLAB Courses", International Journal of Robotics and Automation, vol. 3, no. 3, (**2012**), pp. 167-191.

[14] F. Piltan, M. Eram, M. Taghavi, O. R. Sadrnia and M. Jafari, "Nonlinear Fuzzy Model-base Technique to Compensate Highly Nonlinear Continuum Robot Manipulator", IJISA, vol. 5, no. 12, (**2013**), pp. 135-148, DOI: 10.5815/ijisa.2013.12.12.

[15] F. Piltan, N. Sulaiman, A. Zargari, M. Keshavarz and A. Badri, "Design PID-Like Fuzzy Controller with Minimum Rule Base and Mathematical Proposed On-line Tunable Gain: Applied to Robot Manipulator", International Journal of Artificial Intelligence and Expert System, vol. 2, no. 4, (**2011**), pp. 184-195.