OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

Trajectory Control With MPC For A Robot Manipülatör Using ANN Model

Bekir Cirak
Siirt University, Engineering Faculty, Mechanical Engineering Department, Kezer Campus, Siirt-TURKEY

Abstract: *In this study, in a computer the dynamic motion modelling of manipulator and control of trajectory with an algorithm this has been tested. First after dynamic motion simulation of manipulator has been made MPC Control. The result in this study we can observe that computed torque method gives better results than MPC methods. So in trajectory control it is approved of using computed torque method. In last part of this study the results are estimated forward development are exemined and suggested. The model predictive control (MPC) technique for an articulated robot with n joints is introduced in this paper. The proposed MPC control action is conceptually different with the trajectory robot control methods in that the control action is determined by optimising a performance index over the time horizon. A neural network (NN) is used in this paper as the predictive model.*

Keywords: *Robot manipülators, Model predictive control (MPC), ANN based model, NMPC,* DMPC

I. Introduction

When designing a new process, it is important to exercise some decision making restraint. One critical area to hold back is selecting an advanced process control technology. The choice of PID, model predictive control, or something else should not be decided until the project has been qualified and reasonable objectives have been established. Overlooking this basic principle has caused companies to waste a lot of time and money. When selecting an advanced process control technology, three perspectives are important; process characteristics, process operating objectives and system and application security. Comparing these perspectives to the major process control technologies covered in this series provides a fitting conclusion. An articulated robot with two or more joints is a complex nonlinear time varying MIMO (Multi İnput Multi Output) system with dynamic interaction between its inputs and outputs. Up till now, the majority of practical industrial approaches to the robot ann control design treat each joint of the manipulator as a simple linear servomechanism with, for example, a PD or a PID controller. In designing this kind of controllers, the nonlinear, coupled and time varying dynamics of the mechanical part of the robot manipulator have usually been completely ignored, or treated as disturbances [1].

This method generally gives satisfactory performance when properly tuned and drive one joint at a time. However, when the links are moving simultaneously and at high speeds, the nonlinear coupling effects and the interaction forces between the manipulator links may degrade the performance of the overall system badly and hence increase the tracking error.

Theoretically speaking, centralised control strategies, such as the Computed Torque Control Method and adaptive control, can solve above problems. But in practice uncertainties existing in the robot dynamic model may seriously degrade the performance of the both methods. There are two types of uncertainties, structured and unstructured. Structured uncertainty is defined as the case of a correct dynamical model but with parameter uncertainty due to tolerance variations in the manipulator link: properties, unknown loads, inaccuracies in the torque constants of the actuators, and so on. Unstructured uncertainty describes the case of unmodeled dynamics which result from the presence of high frequency modes in the manipulator, neglected time delays, nonlinear friction, and so on. Although adaptive control has the ability to cope with structured uncertainties, it does not solve the problem of unstructured uncertainties[2].

Therefore, trajectory control approaches are not suitable for the occasion where the robot arm moves at higher velocity. In this paper the method of the model predictive control (MPC) for robot trajectory tracking will be investigated.The concept of MPC comes from the area of industrial process control. Its using in robot control has less been reported. The proposed MPC approach is conceptually different with the trajectory robot control methods in that the control action is determined by optimising a perfonnance index, typically the error between the output prediction derived from the model and the desired output, over the time horizon. Then apply the optimal control actions to the system, measure the system outputs over the time horizon and repeat the above steps until the tracking errors are within the pennitted range.The predictive model for a conventional MPC controller is usually either impulse or step response model which is preferred as being more intuitive and requiring less a prior infonnation for its identification. However, these models are not suitable. for such a nonlinear system as a robot. To solve this problem neural networks are proposed to be the predictive model of the robot for the MPC controller, because. the neural networks have the ability to map any nonlinear relationships between an input and output set. There have also been many reports on the application of neural network to robot modelling and identification [3].

II. Model Predictive Control

MPC is not a specific control strategy but a wide class of optimal control based algorithms that use an explicit process model to predict the behavior of a plant. There is a wide variety of MPC algorithms that have been developed over past 30 years. For example, the Model Predictive Heuristic Control reported by Richalet et al. in 1976 which is used an impulse response model as its linear model. Model predictive control is a generic term for a group of related algorithms that make an explicit use of a process model to calculate control moves minimizing the objective function. The main ideas of MPC for basic structure of MPC in Figure 1.

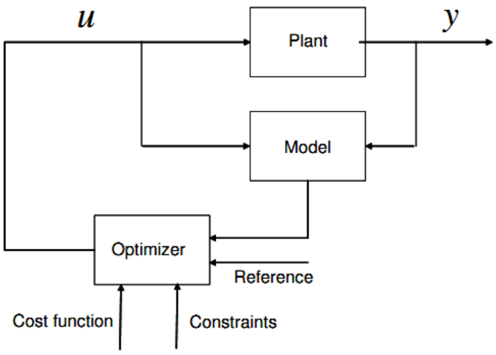


Figure 1. Basic structure of MPC

1. Using an explicit dynamic model of plant to predict the effect of future moves on manipulated variables.
2. Calculating these moves such that they minimize a specific performance criterion while satisfying given operational constraints.
3. Solving this (quadratic) optimization problem in receding horizon manner, using the most recent measurements from the plant to update the prediction. The fundamental framework of MPC algorithms is in common for any kinds of MPC schemes. The basic elements of MPC for MPC strategy are illustrated in Figure 2.

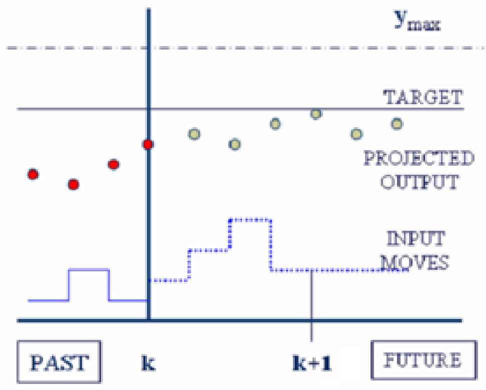


Figure 2. MPC strategy

Model predictive control is especially useful for applications involving constraints on manipulated and/or controlled variables. MPC has been successfully applied in petroleum refineries and extended to numerous other application areas including those found in chemicals, food processing, automotive and aerospace industries. The reason for its popularity is that it addresses the key practical issues often encountered in process control problems including multivariable interactions, constraints, and potentially process nonlinearity all in a single systematic framework [4].

The model predictive control is a strategy that is based on the explicit use of some kind of system model to predict the controlled. Variables over a certain time horizon, the prediction horizon. The control strategy can be described as follows,

1. At each sampling time, the value of the controlled variable $y(t+k)$ is predicted over the prediction horizon $k=l, ..., N$. This prediction depends on the future values of the control variable $u(t+k)$ within a control horizon $k=l,..., NC$,

where $NC \leq N$. If $NC < N$, then $u(t+k)=u(t+NC), k=NC+l, ... ,N$.

2. A reference trajectory $r(t+k), k=1,..,N$ is defined which describes the desired system trajectory over the prediction horizon.

3. The vector of future controls $u(t+k)$ is computed such that a cost function, usually. A function of the errors between the reference trajectory and the predicted output of the model is minimised.

4. Once the minimisation is achieved, the first optimised control action is applied to the plant and measurement of the plant states as the initial states of the model to perform the next iteration. Steps 1 to 4 are repeated at each sampling instant; this is called a receding horizon strategy. The above steps can be expressed by the following equations:

$$\text{Min } \left(\sum_{i=1}^{p} [x_d(k+1) - x(k+1)]^2 \right) \quad (1)$$

subject to

$$U_{min} < U < U_{max}, \quad \forall k \quad (2)$$

Where k is the time step, $u(k)$ is the control vector at time k, $x_d(k)$ and $x(k)$ ate the desired output (reference) and predicted output vector of the model at time k respectively, p is the prediction time horizon. The block diagram of a model predictive controller is shown in Figure 1.

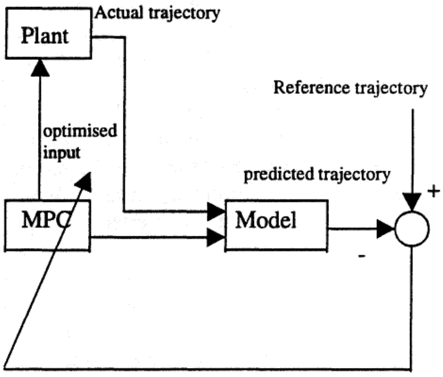


Figure 3. Block diagram of MPC controller

As the control variables in a MPC controller are calculated based on the predicted output, the model thus needs to be able to reflect the dynamic behaviour of the system as accurately as possible, and at least a prior information for the systems identification is required. In the conventional MPC controller, a linear predictive model is used because the theory of the identification of a linear system has well been established. The nonlinear part of the system response is treated as disturbance. But a linear model, no matter how well has it been structured and tuned, may be acceptable only in the case where the system is working around the operating point. If the systelll is highly nonlinear, such as a robot manipulator, control based on the prediction from a linear mouel may result in acceptable response. In some cases, remarkable static errors exist, and in other cases, oscillation or even instability may occur. Therefore some kinds of nonlinear models should be used to describe the behaviour of a highly nonlinear system[5].

To overcome the problems produced by using linear models some researchers have tried to extend the MPC to include nonlinear models. The technique Joseph et al used is to obtain a nonlinear model through system analysis to help the control calculation arrive at appropriate action. The predictive methods using such non linear models have also been made adaptive by estimating parameters of the model that are most likely to change. This requires the model to be of the correct structure, otherwise steady state offsets from the setpoints may result despite parameter adaptation. Selecting such an accurate structure requires significant analysis. However, due to the complexity of the underlying systems, or lack of knowledge of critical parameters of the models in many cases it is impossible to obtain a suitable physically founded system model through an analytical way.

Since the late 1980's, Artificial Neural Networks (ANN) have found wide applications in the engineering field, because of the development of the error backpropagation algorithm. Most engineering researchers are interested in the following two properties of ANN. The fIrst is ANN universal approximation ability that is, ANN could be used to approximate any nonlinear mapping relationship between the inputs and outputs. The second is ANN learning and parallel processing abilities. Based on above two properties the engineering researchers have successfully applied ANN to pattern recognition, nonlinear system identification, controls and many other engineering areas. All above features naturally allow one to think that ANN may be used as an effective tool for the model predictive control of a nonlinear system[6].

III. Neural Network Model

Neural Networks basically comprised of interconnected simulated neurons. A neuron is the smallest unit in network and is used to receive and send signals. Normally each neuron receives signals from other neurons, sums these signals and transforms this sum by means of an activation function, which is monotonic continuously differentiable, bounded function. Frequent used activation function including logistic sigmoid and hyperbolic tangent functions. In addition, there are weights associated with each connection that scale the input to target and training process is to determine optimal weight. The neuron can be arranged into multi-layers which are normally known as multi layer perceptron. The Figure 4 illustrated the basic structure of a neuron[7].

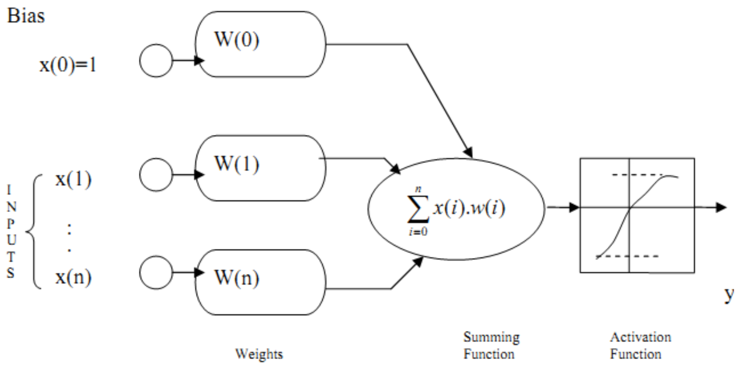


Figure 4. A basic architecture of a neuron

In robotics application to date the most widely used neural network is the Feed forward Neural Network. This is large due its simplicity compared to other networks and its ability to learn the implicit governing relation between the inputs and outputs if sufficient training data is supplied. Feedforward networks is network structure in which the information or signals ill propagates only in one direction on contrary to the recurrent networks in which the delayed time neural net outputs will feed back in to the neural networks as inputs. The Feed forward Neural Network typically consist of three or four layers including input layer, hidden layer and output layer in Figure 5.

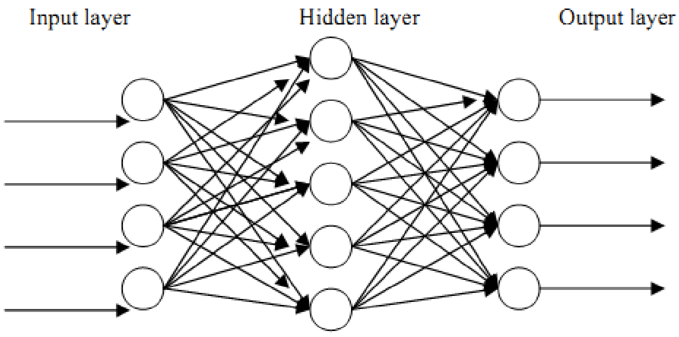


Figure 5. A feedforward artificial neural network structure

For the prediction of the behaviour of the robot forward dynamics, choosen a feedforward neural network with sigmoid activation functions. This kind of neural network is well known and relatively well understood. To set up the neural network predictive model, we can rewrite the robot dynamic equation as following form since the inertial matrix in the equation is symmetric positive definite, so this inertia matrix is always invertible,

$\ddot{q} = H^{-1}(q)(t-h\,(q,\dot{q}) - c(q)-f\,)$ (3)

where the vectors $\ddot{q}$, $\dot{q}$; q are joint acceleration, joint velocity and joint angle and $H(q)$ is the $n \times n$ symmetric positive definite inertia matrix; h (q, $\dot{q}$) is the $n \times \mathbf{1}$ vector of Coriolis and centrifugal torques; $c(q)$ is $n \times \mathbf{1}$ gravitational torques, f is the $n \times \mathbf{1}$ vector representing Coulomb friction and viscous friction forces. t is the $n \times \mathbf{1}$ vector of joint actuator torques.

The NN model trained using these data will not be accurate. To avoid this we can do numerical differentiation on equation (3). Through Taylor fonnula we have the first order difference quotient,

$$\dot{q}(t) = \frac{q(t+h) - q(t-h)}{2h} - [\ddot{q}(t_1) + \ddot{q}(t_2)]$$

$$\dot{q}(t) = \frac{q(t+h) - q(t-h)}{2h} - \frac{h^2}{6} + \ddot{q}(t_m) \qquad (4)$$

Where $t\text{-}h > t_M > t\text{+}h$, and second order difference quotient,

$$\ddot{q}(t) = \frac{q(t+h) - 2q(t) + q(t-h)}{h^2} - \frac{h^2}{4!}[q^{(4)}(t_3) + q^{(4)}(t_4)]$$

$$\ddot{q}(t) = \frac{q(t+h) - 2q(t) + q(t-h)}{h^2} - \frac{h^2}{12} q^{(4)}(t_M) \qquad (5)$$

where $t\text{-}h > t_M > t\text{+}h$, Substitute (4) and (5) into (3), after being straightened up we have:

$$\dot{q}(t+h) = F\,(q(t),\ \dot{q}(t), \dot{q}(t-h), t(t) \qquad (6)$$

or

$$q(t+h) = G\ (q(t), q(t-h), t(t) \qquad (7)$$

where t is the time under consideration, h is the numerical differentiation step, F and G are certain function relationships The accuracy for both (6) and (7) is $O\ (h^2)$. Theiabove procedure shows that with one step time delay the dynamic system (3) could be expressed by (6) or (7) accurately. Thus the robot dynamic characteristic could be emulated by a FNN model approximating the relationships (6), (7) or both. This FNN model could be easily connected to the MPC controller because both (6) and (7) are in predictive form[8].

3.1. Training

Training is basically a systematic adjustment of weights to get a chosen neural network to predict a desired output data training set and it can be done in either supervised or unsupervised way. The training for FNN is supervised. In the supervised training, the connection weights for each processing element are randomly initialized. As the training begins, the training algorithm will start to compare network predicted outputs to the desired outputs from training data set and any error will be used to correct the network. The correction is done by adjusting the set of connection weights of each processing element neuron and this will continues until the algorithm meet the prespecified convergence criteria. The frequent used criteria including the limit of error and the numbers of iteration.

However, care must taken to ensure that the network does not overfit or overfamiliarize with the training data set and hence lose its generalization ability. Various approaches can be used to avoid this problem including regularization theory which attempt to smooth the network mapping and cross validation which using as independent test data.To control the simulated robot system the MPC controller is designed by using the neural network predictive model developed in here[8].

$v = v_{rand}.\ \sin\,(2\pi.\ f_{rand}.\ t\,)$ (8)

Where, v is sinusoidal voltage signals, v_{rand} *is* random magnitude and f_{rand} is random frequency. The frequencies and amplitudes of the signals used to excite the simulated robot system are limited within certain scopes to avoid processing too many data. In this paper the varying ranges of the amplitudes and frequencies of the exciting sinusoidal voltage are within ± 50 volts and 0 to 10 Hz separately. Input voltages v_1, v_2 and v_3 and corresponding time responses at 10 second interval are then collected. A total of 300 sets of voltage signals are used to excite the robot system and 35.000 data sets are collected.

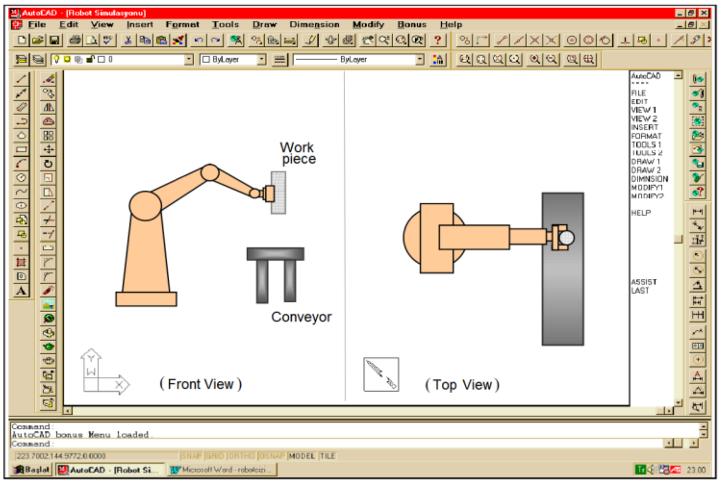
21 inputs composed of nine displacements. q_1, q_2 and q_3 at time *t* and *t-1* respectively, nine velocities $\dot{q}_1$, $\dot{q}_2$ *and* $\dot{q}_3$ at time *t* and *t-l* respectively, and three voltages v_1 *(t)*, v_2 *(t)* and v_3 *(t)*. Hidden layers with 45 neurons respectively. Six outputs representing q_l *(t+1)*, q_2 *(t+1)*, q_3 *(t+1)* and $\dot{q}_1$*(t+1)*, $\dot{q}_2$ *(t+1)*,$\dot{q}_3$*(t+1)* respectively.Thus the feedforward neural network has a structure of 21-45-6.

The momentum coefficient (β) and training rate coefficient (α) are set to be 0.90 and 0.004 respectively. The training was terminated after 2 million iterations without further significant reduction being observed. The algorithm used to train the neural network models is the standard backpropagation.Then this neural network model is connected to the MPC controller.

IV. Simulations

4.1 Simulation of robot system

In this study, the inverse kinematics calculations and trajectory planning of a robot arm with three axes has been done. its motion has been obtained and the motion of robot arm has been simulated by a computer. AutoCAD programming was used for the calculations and the 3D simulation of the robot arm. After the image processing detection were done using Matlab R2009 b, the results were transferred to the program written AutoCAD. The robot system to be controlled is simulated by a computer program. The prototype of the simulated robot system is a modified PUMA robot in which the joint motors are voltage controlled. In this study, movement simulation of a three armed robot has been realized by using AutoLISP programing language which is supplied with AutoCAD. Analytical and matrix solution methods have been used in simulation equations. Shows top and front views of PUMA robot maked by simulation programme in Figure 2.



```
Command:
AutoCAD bonus Menu loaded.
Command:
```

Figure 4. The views of Puma Robots in Simulation Programmes

It is assumed that there are three links with the simulated robot system. The torque produced by a DC motor is used to drive each link through a set of transmission mechanism connecting the motor and load shafts. The DC motor in this paper is simplified as a resistance inductance circuit with voltage source. The voltage source in this circuit is the voltage input to the motor. Back electromotive force produced across the armature is proportional to the angular velocity of the motor shaft. And torque produced by the motor is proportional to the armature current. There is a backlash between the mating gears in the transmission mechanism [9].

The dynamic equation used in the simulated robot system is,

$$\tau_i = H(q_i)\ddot{q}_i + h(q_i, \dot{q}_i) + c\,(q_i) + f \tag{8}$$

The meaning of all the symbols. used above are similar with those for (3). The friction force. is represented using the following equation,

$$f_i(q_i) = c_i\ sig\ (q_i) + v_i\dot{q}_i \tag{9}$$

where *c* and *v* represent the coulomb and viscous friction coefficients, *i* is the joint number under consideration, *sgn(x)* is the sig function[10].

4.2. Simulation of MPC controller

The optimisation problem expressed in equation (1) and (2) is a simple bounded variable nonlinear optimisation problem without constraints. The bounded variables are the control inputs (control torques) of the robot. To solve this optimisation problem, the quadratic nonlinear programming routine NLPQ provided in IMSL is used. This method is chosen because the authors are familiar with this routine. Many other methods may also be used to solve this problem, perhaps in a more efficient way.To test the performance of the proposed model predictive control strategy, the desired trajectories for the simulated robot system to follow are generated by inputting the system a group of sinusoidal excitations, in which the amplitudes and frequencies are within the frequency and amplitude limits used.The desired trajectories to be followed by the robot joints are shown in Figures 5, 6 and 7.

A comparison of the effectiveness of the model predictive control based on the neural network model (NMPC), the model predictive control based on robot model in which the dynamic equations are with nominal parameters (DMPC) is used. Because of the inevitable measurement errors, it is difficult to obtain the accurate values of dynamical parameters for the robot model. In this paper, it is assumed that there are 14 % of measurement errors in the moment of inertia for each link, and the friction terms are neglected. The tracking errors obtained by applying DMPC and NMPC are presented in Figures 8, 9 and 10. The results show that NMPC provides a beter performance than DMPC, as predicted. It is clear the performance of the DMPC will improve if the parameters of the nominal model can be obtained more accurately.

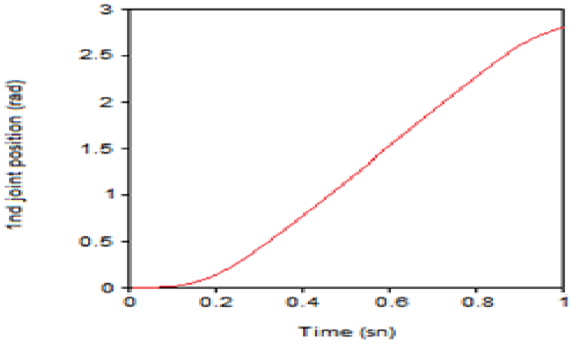


Figure 5. Desired trajectories

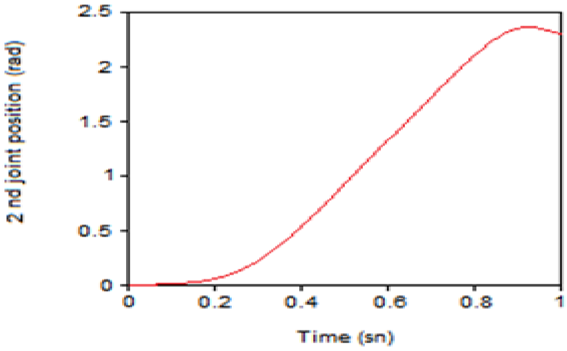


Figure 6. Desired trajectories

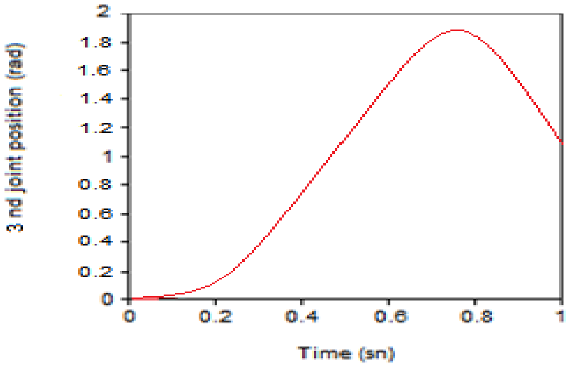


Figure 7. Desired trajectories

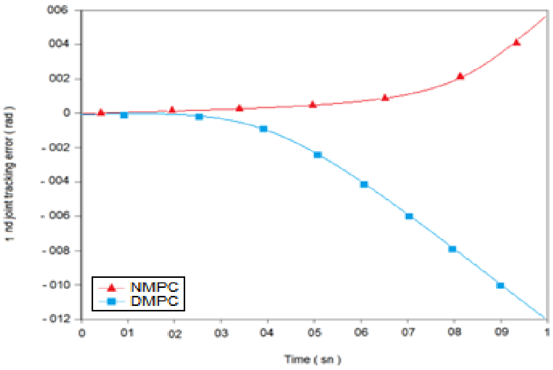


Figure 8. Tracking errors of 1 nd joint

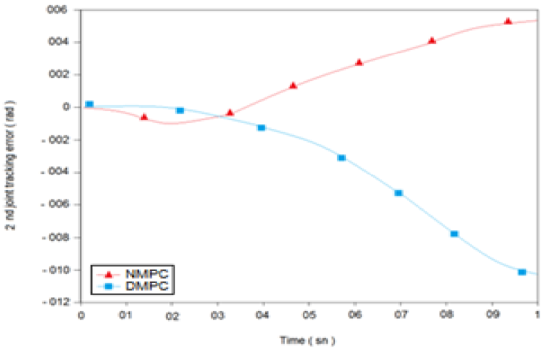


Figure 9. Tracking errors of 2 nd joint

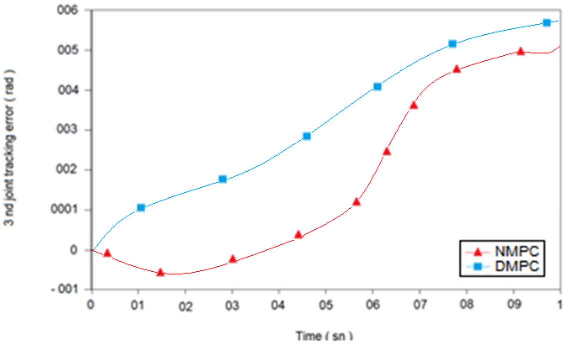


Figure 10. Tracking errors of 3 nd joint

V. Conclusion

The use of DMPC and NMPC to control of trajectory for three joint manipülator, was investigated. An unconstrained, Multi İnput Multi Output (MIMO) DMPC and NMPC algorithms were developed using a step response model and two Feedforward Neural Networks respectively. Additionally, the comparison between DMPC and NMPC controller based was conducted. The trajectory tracking results with higher accuracy in this paper are obtained without considering the regulation of the control voltages. the NMPC controller is a potential effective way for robot trajectory tracking. This is not impractical because to obtain a higher tracking accuracy, the frequency of the control voltage may be so high that it is far beyond the frequency response scope of an actual driving motor. Further research on the real time execution of NMPC and the regulation of control voltage is required and is currently being carried out. Comparison of the NMPC and the DMPC controller, both are the industrially popular and successful control strategies in this research had clarify the significant features of NMPC.

REFERENCES

[1]. Kosmatopoulos E.B., Chassiakos A. And Christodoulo M.A., 'Robot Identification using dynamical neural networks." Engineering Systems with Intelligence. Kluwer Academic Publishers. 1991. pp. 187-195.

[2]. Gupta M. “ Robust repetıtıve model predıctıve control for systems wıth Uncertaın perıod-tıme ” Master Thesis of Science in Chemical Engineering Georgia Institute of Technology May, 2004

[3]. Fu K.S., Gonzalez R. C., Lee C.S.G., Robotics: control, sensing, vision and intelligence. McGraw Hill, New York, 1987

[4]. Nebot E.M. and Fang G., 'Robot Dynamic Identification using ·Neural Networks." Proceedings of Mechatronics & Machine Vision in Practice. Toowoomba, Qld., Australia. Sept. 1994. pp. 200-205

[5]. Boo Chin E., Hong Mei K. ve Amy Tan S. “ Formulatıon of model predıctıve control algorıthm for nonlınear processes “Jabatan Kejuruteraan Kimia Universiti Teknologi Malaysia 2006

[6]. Seidl D.A., Lam S. Putman J.A. and Lorenz R.D., "Neural Network Compensation of Gear Backlash Hysteresis in Position-Controlled Mechanisms." IEEE Transactions on Industry Applications. Vol. 31, No.6, 1995. pp 1475-1483.

[7]. Eskandarian A, Bedewi N.E., Kramer B.M. and Barbera A. J., “ Dynamics modelling of robotic manipulators using an artificial neural network. ”Journal of Robotic Systems. Vol. 11, No.1, 1994. pp.41-56

[8]. Zhouping W. and Gu F. ” Model Predictive Control for Robot Manipulators Using A Neural Network Model ” School of Mechatronic, Computer and Electrical Engineering, University of Western Sydney, Nepean, PO Box 10, IGngswood, NSW 2747

[9]. Garcia C.E., and MorarIM., 'Internal model control. 1. A unifying review and some new results." Ind. Eng. Chern. .Process Des. Dev. Vol. 21, No.2, 1982.pp. 308-323.

[10]. Joseph, B., Jang, S. and Mukai, H., 'Integrated Model Based Control of Multivariable Non linear Systems." Proceedings IFAC Workshop on Process Model Based Control. Atlanta, 1988.