Model Predictive Control for Robot Manipulators Using A Neural Network Model

Zhouping Wei and Gu Fang

School of Mechatronic, Computer and Electrical Engineering, University of Western Sydney, Nepean, PO Box 10, Kingswood, NSW 2747

Abstract: 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 traditional 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. The training data to set up the NN model come from measuring the robot joint variables and torques during the running of the real robot. The computer simulations are given in the paper to demonstrate the advantages of the method.

KEYWORDS: Robot control; Model predictive control; ANN-based control methods

1. Introduction

An articulated robot with two or more joints is a complex nonlinear time-varying MIMO system with dynamic interaction between its inputs and outputs. Up till now, the majority of practical industrial approaches to the robot arm 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 non-linear, coupled and time-varying dynamics of the mechanical part of the robot manipulator have usually been completely ignored, or treated as disturbances. 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 non-linear 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 (CTM) 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. Therefore, traditional 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 traditional robot control methods in that the control action is determined by optimising a performance 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 permitted 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 information

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 [1,5,7,8,9].

In this paper a method for modelling the robot dynamic behaviour by using the feedforward neural networks is introduced. This method uses a feedforward neural network, trained by the data measured from the running of the real robot, to model the forward dynamics of a robot system in a predictive way so that it could be easily used in the proposed MPC controller.

We begin our discussion by proposing a MPC control structure in section 2. The method to build up the neural network predictive model is introduced in section 3. The MPC controller is then connected with the neural network model in section 4. Simulation results are also given in section 4. The conclusions are presented in section 5.

2. The Model Predictive Control Algorithm

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 [4]:

1) At each sampling time, the value of the controlled variable $y(t+k)$ is predicted over the prediction horizon k=1,..., N. This prediction depends on the future values of the control variable $u(t+k)$ within a control horizon k=1,..., NC, where NC≤N. If NC<N, then $u(t+k)=u(t+NC)$, k=NC+1,... ,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 the plant outputs are measured. Use this 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:

$$\min \left(\sum_{l=1}^{p} [x_d(k+l)-x(k+l)]^2 \right) \qquad (1)$$

subject to

$$u_{min} < u < u_{max} \quad \forall k \qquad (2)$$

where k is the time step, u(k) is the control vector at time k, $x_d(k)$ and x(k) are 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.

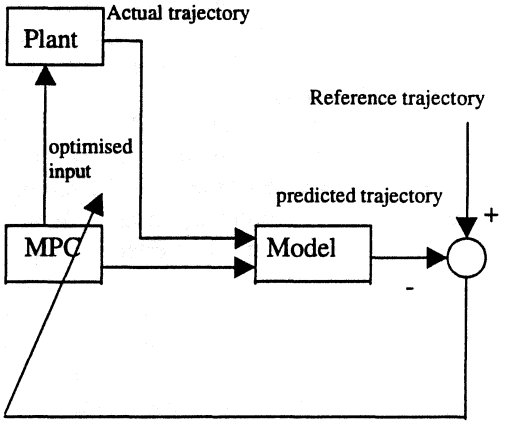


Figure 1. Block diagram of MPC controller.

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 system's 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 system is highly non-linear, such as a robot manipulator, control based on the prediction from a linear model may result in unacceptable response. In some cases, remarkable static errors exist, and in other cases,

oscillation or even instability may occur. Therefore, some kinds of non-linear models should be used to describe the behaviour of a highly non-linear system.

To overcome the problems produced by using linear models some researchers have tried to extend the MPC to include non-linear models. The technique Joseph et al [6]. used is to obtain a non-linear 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 (ANNs) 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 ANNs. The first is ANNs' universal approximation ability; that is, ANNs could be used to approximate any non-linear mapping relationship between the inputs and outputs. The second is ANN's learning and parallel processing abilities. Based on above two properties the engineering researchers have successfully applied ANNs to pattern recognition, non-linear 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.

3. The Neural Network Predictive Model for MPC Controller

For the prediction of the behaviour of the robot forward dynamics, we choose a feedforward neural network (FNN) with hyperbolic tangent 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:

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

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

Based on (3), Fang and Dissanayake [2] proposed a method to emulate robot joint accelerations through FNN together with a solver for ODEs, by using FNN's property of being able to approximate any nonlinear relationship.

However the measurement of acceleration is always required if the FNN model is used to emulate (3) directly. Yet in practice the measurement of acceleration in high accuracy is very difficult. Therefore, the NN model trained using these data will not be accurate. To avoid this we can do numerical differentiation on equation (3). Through Taylor formula we have the first order difference quotient:

$$q'(t) = \frac{q(t+h) - q(t-h)}{2h} - \frac{h^2}{12}[q'''(t_1) + q'''(t_2)]$$

$$= \frac{q(t+h) - q(t-h)}{2h} - \frac{h^2}{6}q'''(t_m) \qquad (4)$$

where $t\text{-}h > t_m > t+h$, and second order difference quotient:

$$q''(t) =$$

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

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

where $t\text{-}h > t_M > t+h$.

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 $O(h^2)$. The above 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.

4. Simulations

4.1 The simulated robot system

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 paper Fortran programming language is used. It is assumed that there are three links with the simulated robot system. The torque produced by a permanent magnet 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 [3]. 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. The backlash phenomenon is described through following equations [10]:

$$t_l = \begin{cases} 0 & (8a) \\ -b < q_m - q_l/n < b & \\ J_m J_l (J_m + J_l/n^2)^{-1} (\dot{q}_m(t^-)/n - \dot{q}_l(t^-)) d & (8b) \\ \begin{cases} q_m - q_l/n = b \quad and \quad \dot{q}_m > n\dot{q}_l \quad or: \\ q_m - q_l/n = -b \quad and \quad \dot{q}_m < n\dot{q}_l \end{cases} & \\ \max(0, t_m) & (8c) \\ q_m - q_l/n = b \quad and \quad \dot{q}_m = n\dot{q}_l = 0 & \\ \min(0, t_m) & (8d) \\ q_m - q_l/n = -b \quad and \quad \dot{q}_m = n\dot{q}_l = 0 & \\ n[t_m - (J_m n\ddot{q}_l + f_m n\dot{q}_l)] & (8e) \\ \begin{cases} q_m - q_l/n = b \quad and \quad \dot{q}_m = n\dot{q}_l \neq 0 \quad or: \\ q_m - q_l/n = -b \quad and \quad \dot{q}_m = n\dot{q}_l \neq 0 \end{cases} & \end{cases} \quad (8)$$

In (8), b is the half of the angular deadzone distance between gears on the motor shaft side. J is the moment of inertia. q and $\dot{q}$ are the angular displacement and velocity respectively. τ is torque. Subscript m and l denote motor and load shaft respectively. δ is impulse signal. (8a) indicates the gear torque is zero when the gear is in the backlash region. (8b) introduces torque impulses at the instant when the gears reach the boundaries of the backlash region with sufficient relative velocity to collide. The collision is inelastic. (8c) and (8d) give the torque transmitted when the gears are engaged but not moving. (8e) gives the transmitted torque when the gears are engaged.

The dynamic equation used in the simulated robot system is:

$$\tau_l = H(q_l)\ddot{q}_l + h(q_l, \dot{q}_l) + c(q_l) + f \qquad (9)$$

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

$$f_i(q_i) = c_i \, \mathrm{sgn}(q_i) + v_i \dot{q}_i \qquad (10)$$

where c and v represent the Coulomb and viscous friction coefficients, i is the joint number under consideration, sgn(x) is the sign function.

4.2 Training of the neural network model

To excite the robot system, sinusoidal voltage signals ($v = V_{rand} \sin(2\pi f_{rand} t)$) are used with random magnitude V_{rand} and random frequency f_{rand}. 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 ± 40 volts and 0 to 5 Hz separately. Input voltages v_1, v_2 and v_3, and corresponding time responses at 5 *msec* interval are then collected. A total of 250 sets of voltage signals are used to excite the robot system and 50,000 data sets are collected.

The feedforward neural network has a structure of 15-45-20-6. 15 inputs composed of six displacements q_1, q_2 and q_3 at time t and t-1 respectively, six velocities $\dot{q}_1$, $\dot{q}_2$ and $\dot{q}_3$ at time t and t-1 respectively, and three

voltages $v_1(t)$, $v_2(t)$ and $v_3(t)$. Two hidden layers with 45 and 20 neurons respectively. Six outputs representing $q_1(t+1)$, $q_2(t+1)$, $q_3(t+1)$, $\dot{q}_1(t+1)$ $\dot{q}_2(t+1)$, and $\dot{q}_3(t+1)$ respectively. The algorithm used to train the neural network models is the standard backpropagation. The momentum and training rate are set to be 0.95 and 0.005 respectively. The training was terminated after 1.5 million iterations without further significant reduction being observed. Then this neural network model is connected to the MPC controller.

4.3 Simulation result of the MPC controller

To control the simulated robot system the MPC controller is designed by using the neural network predictive model developed in section 4.2.

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 in section 4.2. In this section, the desired trajectories to be followed by the robot joints are shown in Figure 2. As a comparison of the effectiveness of the model predictive control based on the neural network model (annotated as NNMPC), the model predictive control based on robot model in which the dynamic equations are with nominal parameters (NMMPC) 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 20% of measurement errors in the moment of inertia for each link, and the friction terms are neglected. The tracking errors obtained by applying NMMPC and NNMPC are presented in Figure 3. The results show that NNMPC provides a better performance than NMMPC, as predicted. It is clear, however, the performance of the NMMPC will improve, if the parameters of the nominal model can be obtained more accurately.

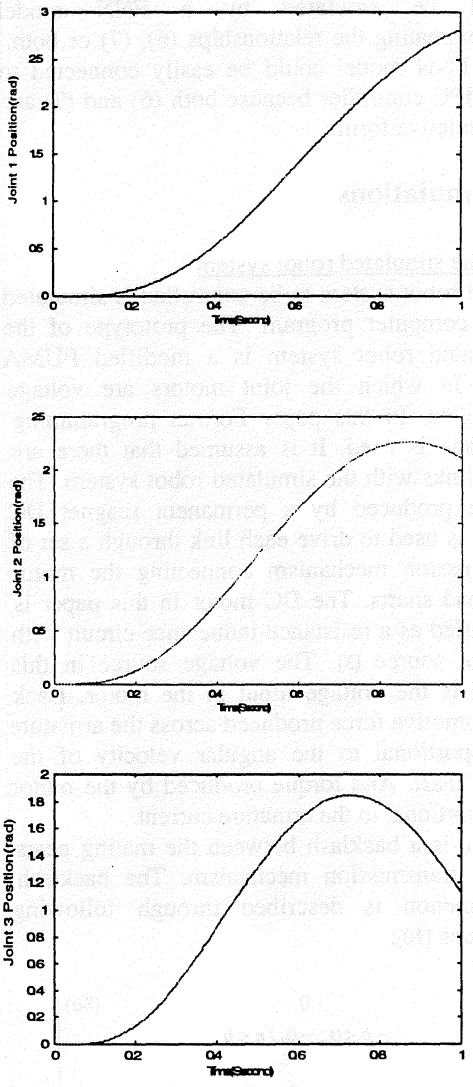
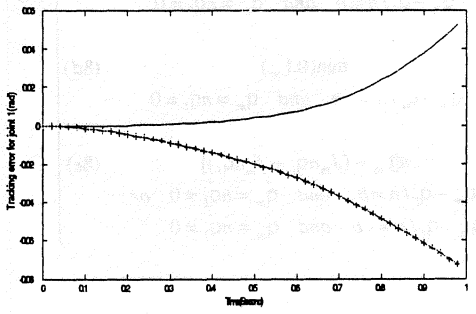
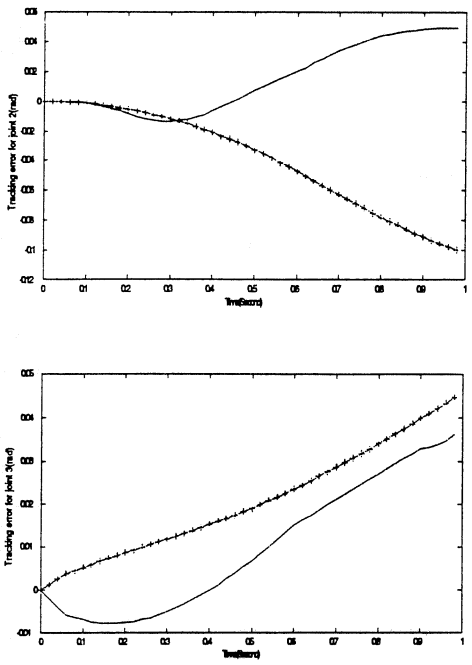


Figure 2. Desired trajectories





+: NMMPC -: NNMPC

Figure 3. Tracking errors in following the trajectories in Figure 2

5. Conclusion

From the results presented above, it can be seen that the NNMPC controller is a potential effective way for robot trajectory tracking.

However, one of the major drawbacks in applying MPC in robot applications is that the computational time required to perform the MPC is considerably long. Besides, the trajectory tracking results with higher accuracy in this paper are obtained without considering the regulation of the control voltages. 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 NNMPC and the regulation of control voltage is required and is currently being carried out.

References

1. 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
2. Fang, G and Dissanayake, M.W.M.G., "A model predictive controller for robot trajectory tracking based on a neural network model." *Proceedings on the 2nd International Power Electronics and Motion Control Conference.* 3-6 Nov. 1997, Hangzhou, China. pp. 689-694.
3. Fu K.S., Gonzalez R. C., Lee C.S.G., *Robotics: control, sensing, vision and intelligence.* McGraw Hill, New York, 1987
4. Garcia C.E., and Morarl M., "Internal model control. 1. A unifying review and some new results." *Ind. Eng. Chem. Process Des. Dev.* Vol. 21, No. 2, 1982. pp. 308-323.
5. Hunt K.J., Sbarbaro D., Zbikowski R. and Gawthrop P.J., "Neural Networks for Control Systems — A Survey." *Automatica.* Vol. 28, No.6, 1992. Pp. 1083-1112.
6. 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
7. 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.
8. Khemaissia and Morris, A.S., "Fast-Robot System Identification Based on Neural Network Models." *Proceedings of the 1992 IEEE International Symposium on Intelligent Control.* Glasgow, UK , 1992. pp.13-15
9. 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
10. 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.