17th ICEE 2009
Iranian Conference on Electrical Engineering
www.icee.ir/2009

Obstacle Avoidance and Grasping Moving Objects in Non-Stationary Environments for Redundant Robot Manipulators Using NMPC

Ashkan M. Jasour*, Mohammad Farrokhi**
* IRAN University of Science and Technology, TEHRAN, IRAN, amjasour@ee.iust.ac.ir
**Center of Excellence for Power System Automation and Operation
** IRAN University of Science and Technology, TEHRAN, IRAN, farrokhi@iust.ac.ir

Abstract: *In this paper, a Nonlinear Model Predictive Control (NMPC) is designed for redundant robot manipulators in non-stationary environments. Using NMPC, the end-effector of the robot follows a moving target in the Cartesian space and at the same time avoids collision with obstacles and singular configurations in the workspace. To avoid collisions with moving obstacles and capturing moving target, the future position of obstacles and the moving object in 3D space is predicted using artificial neural networks. Using online training of the neural network, no knowledge about obstacles and motion of the moving object is required. The nonlinear dynamic of the robot including actuators dynamic is also considered. Numerical simulations performed on a 4DOF redundant spatial manipulator actuated by DC servomotors shows effectiveness of the proposed method.*

Keywords: robot manipulator, nonlinear model predictive control, moving target, obstacle avoidance, artificial neural networks.

1. Introduction

Today, robot manipulators are increasingly used in many tasks such as industry, medicine and space. One of the main reasons for the development of manipulator robots is to replace human in doing long, repetitive, and unhealthy tasks. In some applications, these robots are used to capture an object in such a way that no collision with obstacles in the environment occurs. However, high degrees of freedom for redundant manipulators lead to infinite number of possible joint positions for the same pose of the end-effector. Hence, for a given end-effector coordinate in the Cartesian space, the robot can reach it in many different configurations, among which, the collision free and singular free situations must be selected. Finding feasible paths for joints of redundant manipulators for a given end-effector coordinate is called redundancy resolution [1]. Redundancy resolution and obstacle avoidance are already considered in papers. Using the gradient projection technique, redundancy is solved considering the obstacle avoidance [2]. In task-priority redundancy resolution technique, the tasks are performed based on the order of priority; path tracking is given the first priority and obstacle avoidance or singularity avoidance is given the second priority [3, 4]. This technique yields locally optimal solution that is suitable for real-time redundancy control but not for large number of tasks. The generalized inverse Jacobin technique and extended Jacobin technique, which are used for redundancy solution, are time consuming [5]--[7]. Optimization techniques, which minimize a cost function subject to constraints, like the end-effector path tracking and obstacle avoidance, are not suitable for on-line applications [4].

This paper presents a solution for obstacle avoidance in non-stationary environments. The space-time and velocity adjusting methods are proposed in [8, 9] on this problem.

In this paper, Nonlinear Model Predictive Control (NMPC) method is presented for redundancy resolution considering moving obstacles and singularity avoidance. Although Model Predictive Control (MPC) is not a new control method, works related to manipulator robots using MPC is limited. Most of the related works are about joint space control and end-effector coordinating. The linear MPC is used in [10, 11, 12] and NMPC is used in [13, 14, 15, 16] for joint space control of manipulators.

This paper is organized as follows: Section 2 presents nonlinear dynamic of a 4DOF spatial redundant manipulator including the actuators dynamic. Section 3 describes the nonlinear model predictive control. Section 4 explains prediction of moving obstacles and the target position. Section 5 implements the NMPC for path tracking and obstacle avoidance of a 4DOF manipulator. Simulation results are presented in Section 6. Conclusions are drawn in Section 7.

2. Manipulator Robot Dynamic

Schematic diagram of a 4DOF spatial redundant manipulator robot is shown in Fig. 1. Table I gives Denavit-Hartenberg parameters of this robot [17]. The position of the end-effector in Cartesian space can be calculated in terms of joint angles as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} c_1(l_4 c_{234} + l_3 c_{23} + l_2 c_2) \\ s_1(l_4 c_{234} + l_3 c_{23} + l_2 c_2) \\ l_4 s_{234} + l_3 s_{23} + l_2 s_2 \end{bmatrix} \tag{1}$$

The dynamic model of the robot manipulator can be obtained using the Lagrangian method as [17, 18]

$$\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{D}(\dot{\boldsymbol{\theta}}) + \mathbf{G}(\boldsymbol{\theta}) = \boldsymbol{\tau} \tag{2}$$

where $\boldsymbol{\theta} \in R^n$ is the angular position of joints, $\mathbf{M}(\boldsymbol{\theta}) \in R^{n \times n}$ is the symmetric and positive definite inertia matrix, $\mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in R^n$ is the centrifugal and coriolis force vector, $\mathbf{G}(\boldsymbol{\theta}) \in R^n$ is the gravity vector, $\mathbf{D}(\dot{\boldsymbol{\theta}}) \in R^n$ is the vector for joints friction of the links, $\boldsymbol{\tau} \in R^n$ is the torque vector of joints, and n is the degree of freedom, which is

equivalent to four for the robot considered in this paper. The above matrix and vectors are given in Appendix.

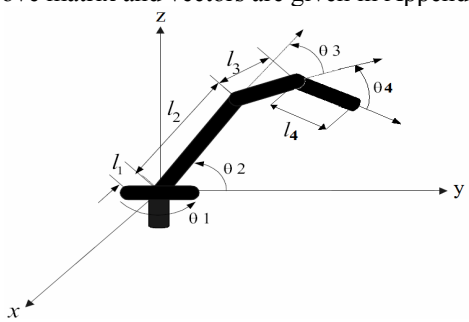

Fig. 1 Schematic diagram of a 4DOF spatial manipulator

TABLE I
DENAVIT-HARTENBERG PARAMETERS OF THE ROBOT IN FIG.1

Link	α^o	A	d	θ^o
1	90	0	0	θ_1
2	0	l_2	0	θ_2
3	0	l_3	0	θ_3
4	0	l_4	0	θ_4

Friction for the ith joint is defined as [18]

$$D(i) = D_v \dot{\theta}_i + D_d \operatorname{sgn}(\dot{\theta}_i) \tag{3}$$

where D_v and D_d are coefficients of the viscous and dynamic frictions, respectively.

The dynamics of the armature-controlled DC servomotors that drive links can be expressed as [18]

$$\begin{aligned} \boldsymbol{\tau}_e &= \mathbf{K}_T \mathbf{i}_a \\ \boldsymbol{\tau}_e &= \mathbf{J}_m \ddot{\boldsymbol{\theta}}_m + \mathbf{B}_m \dot{\boldsymbol{\theta}}_m + \boldsymbol{\tau}_m \\ \mathbf{V}_t &= \mathbf{R}_a \mathbf{i}_a + \mathbf{L}_a \dot{\mathbf{i}}_a + \mathbf{K}_E \dot{\boldsymbol{\theta}}_m \end{aligned} \tag{4}$$

where $\boldsymbol{\tau}_e \in R^n$ is the vector of electromagnetic torque, $\mathbf{K}_T \in R^{n\times n}$ is the diagonal matrix of the motor torque constant, $\mathbf{i}_a \in R^n$ is the vector of armature currents, $\mathbf{J}_m \in R^{n\times n}$ is the diagonal matrix of the moment inertia, $\mathbf{B}_m \in R^{n\times n}$ is the diagonal matrix of torsional damping coefficients, $\boldsymbol{\theta}_m, \dot{\boldsymbol{\theta}}_m, \ddot{\boldsymbol{\theta}}_m \in R^n$ denote the vectors of motor shaft positions, velocities and accelerations, respectively, $\boldsymbol{\tau}_m \in R^n$ is the vector of load torque, $\mathbf{V}_t \in R^n$ is the vector of armature input voltages, $\mathbf{R}_a \in R^{n\times n}$ is the diagonal matrix of armature resistances, $\mathbf{L}_a \in R^{n\times n}$ is the diagonal matrix of armature inductances, and $\mathbf{K}_E \in R^{n\times n}$ is the diagonal matrix of the back electromotive force (EMF) coefficients.

The relationship between robot joints and motor shafts can be represented as

$$\Gamma = \operatorname{diag}\left[\theta_i / \theta_{m_i}\right] = \operatorname{diag}\left[\tau_{m_i} / \tau_i\right] \tag{5}$$

where $\Gamma \in R^{n\times n}$ is a diagonal positive definite matrix representing the gear ratios for n joints. Since armature inductances are small and negligible, Eq. (4) can be expressed as [15]

$$\mathbf{J}_m \ddot{\boldsymbol{\theta}}_m + (\mathbf{B}_m + \mathbf{K}_E \mathbf{K}_T \mathbf{R}_a^{-1}) \dot{\boldsymbol{\theta}}_m + \boldsymbol{\tau}_m = \mathbf{K}_T \mathbf{R}_a^{-1} \mathbf{V}_t \tag{6}$$

Using Eq. (5) to eliminate θ_m and τ_m in Eq. (6) and then substituting for $\boldsymbol{\tau}$ using Eq. (2), the governed equation of n-link robot manipulator including actuator dynamics can be written as

$$\begin{aligned} (\mathbf{J}_m + \mathbf{R}^2\mathbf{M})\ddot{\boldsymbol{\theta}} + (\mathbf{B}_m + \mathbf{K}_E \mathbf{K}_T \mathbf{R}_a^{-1})\dot{\boldsymbol{\theta}} + \mathbf{R}^2(\mathbf{C} + \mathbf{G} + \mathbf{D}) \\ = \Gamma \mathbf{K}_T \mathbf{R}_a^{-1} \mathbf{V}_t \end{aligned} \tag{7}$$

According to Eq. (7), the armature input voltages are considering as control efforts, respectively.

3. Model Predictive Control

Unlike classical control methods, where control actions are calculated based on the past output(s) of the system, the MPC is a model-based optimal controller, which uses predictions of system outputs to calculate the control law [19, 20].

Based on measurements obtained at the sampling time k, the controller predicts the output of the system over the prediction horizon N_P in future instances using the system model and determines the input over the control horizon $N_C \le N_P$ such that a predefined cost function is minimized.

To incorporate the feedback, only the first member of the obtained input is applied to the system until the next sampling time [19]. Using new measurements at the next sampling time, the whole procedure of prediction and optimization is repeated.

From the theoretical point of view, the MPC algorithm can be expressed as follow:

$$u = \arg_u \min(J(k)) \tag{8}$$

such that

$$\begin{aligned} &\mathbf{x}(k \mid k) = \mathbf{x}_0 \\ &u(k+j \mid k)) = u(k+N_u \mid k)) \ , \ j \ge N_C \\ &\mathbf{x}(k+j+1 \mid k) = f_d(\mathbf{x}(k+j \mid k), u(k+j \mid k)) \\ &y(k+j+1 \mid k) = h_d(\mathbf{x}(k+j+1 \mid k)) \\ &\mathbf{x}_{\min} \le \mathbf{x}(k+j+1 \mid k) \le \mathbf{x}_{\max} \\ &u_{\min} \le u(k+j \mid k) \le u_{\max} \end{aligned} \tag{9}$$

where $j \in [0 \ , \ N_P\text{-}1]$, x and u are states and input of the system, x_0 is the initial condition, and f_d and h_d are the model of the system used for prediction. The notation $a(m|n)$ indicates the value of a at instant m predicted at instant n. Moreover, intervals $[x_{\min} \ , \ x_{\max}]$ and $[u_{\min} \ , \ u_{\max}]$ stand for the lower and the upper bound of states and input, respectively. The cost function J is defined in terms of the predicted and the desired output of the system over the prediction horizon. MPC schemes that are based on the nonlinear model of the system or consider a non-quadratic cost function and nonlinear constrains on inputs and states are called Nonlinear MPC [20].

The optimization problem (9) must be solved at each sampling time k, yielding a sequence of optimal control law as $\{\mathbf{u}^*(k \mid K), \ldots, \mathbf{u}^*(k+N_u-1)\}$. For optimization, the SQP method is used in this paper [21].

4. Prediction of Moving Target and Obstacles Position Using Artificial Neural Networks

In order to capture moving target and avoid robot collision with obstacles in the workspace, the future position of the moving object and obstacles in 3D space must be predicted over the prediction horizon. In this paper, a Multilayer Perceptron (MLP) is used for this purpose. The prediction is

based on the coordinate of the moving target and obstacles at past sampling times; the predictor output is the coordinate of the moving target and obstacles at the next sampling time. Using this structure, a one-step-ahead prediction of the moving target and obstacles position is obtained. However, in the predictive control, multi-step predictions over the prediction horizon are needed. By applying one-step prediction recursively, the multi-step prediction can be achieved. In this case, outputs of the Neural Network (NN) are considered as inputs for the next step. To acquire high accuracy for the one-step-ahead prediction the NN structure must be selected carefully. The structure of the employed MLP is shows in Fig. 2. Transfer functions for the hidden and output layer neurons are of hyperbolic tangent and linear types, respectively. To predict the obstacle position, the obstacle coordinates at the current and at the past 2 samples are fed as inputs to the NN; the output of the NN is the obstacle coordinates at the next sampling time (Fig. 2). The same method is employed for predicting the coordinates of the moving target in 3D space.

Training of NNs is performed online using the recursive least squares algorithm [23]. Using this algorithm at each sampling time k, summation of the matching errors for all the input-output pairs up to the kth sample is minimized. But in this paper, summation of errors for the last N input-output pairs is considered for minimization. In this way, the training time can remain constant and N can be selected in such a way that training time over every sampling time does not exceed the sampling rate. One of the main advantages of using NNs for prediction is that by employing the online training scheme, no prior knowledge about the motion of obstacles is needed.

5. Path Tracking and Obstacle Avoidance Using NMPC

The purpose of the object catching and the obstacle avoidance in robot manipulators is to obtain a control law such that the end-effector reaches a moving target in the Cartesian space and at the same time avoids collisions with obstacles. To this end, the NMPC method is implemented in this paper. The block diagram of NMPC is shown in Fig. 3.

As explained in Section 3, in order to obtain the control law, an appropriate cost function should be defined for the NMPC algorithm.

For the path tracking, the cost function must have direct relation with the tracking error between the end-effector and the moving object coordinates. On the other hand, for obstacles avoidance the cost function must have an inverse relationship with the distance between the obstacle and the manipulator links. In this paper the cost function is considered as

$$J = \sum_{j=1}^{N_P} Q\left(\frac{e^{D_P(k+j|k)} - e^{D_{P\min}}}{e^{D_{P\max}} - e^{D_{P\min}}}\right) + R\left(\frac{e^{-D_O(k+j|k)} - e^{-D_{O\max}}}{e^{-D_{O\min}} - e^{-D_{O\max}}}\right) \quad (10)$$

where D_P is the Euclidean distance between the end-effector and the moving object, D_O is the minimum Euclidean distance between the manipulator and obstacles, and $Q \geq 0$ and $R \geq 0$ are the weighting parameters; notation $a(m|n)$ indicates the value of a at the instant m predicted at instant n. Moreover, intervals $[D_{P\min}, D_{P\max}]$ and $[D_{O\min}, D_{O\max}]$ are the range of variations for D_P and D_O, respectively. According to the length of manipulator links, the value for $D_{P\min}$ and $D_{P\max}$ is 0 and 2.4 meter and the value of $D_{O\min}$ and $D_{O\max}$ is 0 and 1.2 meter, respectively. Using the cost function (10), the two terms for the path tracking and the obstacle avoidance are normalized to [0 1].

The NMPC in this paper uses the nonlinear dynamic model of the manipulator in the optimization of the cost function.

Substituting $(\theta(k+1) - \theta(k))/T$ for $\dot{\theta}$ in dynamic Eq. (7), a one-step-ahead prediction for joints angles can be expressed as

$$\theta(k+1) = f_d(\theta(k), V_t(k)) \quad (11)$$

where k is the sampling time and T is the sampling rate. Using the forward kinematics (1), a one-step-ahead prediction of the end-effector position can be obtained. However, in the predictive control, this equation is used for multi-step predictions over the prediction horizon by recursively applying the one-step-ahead prediction.

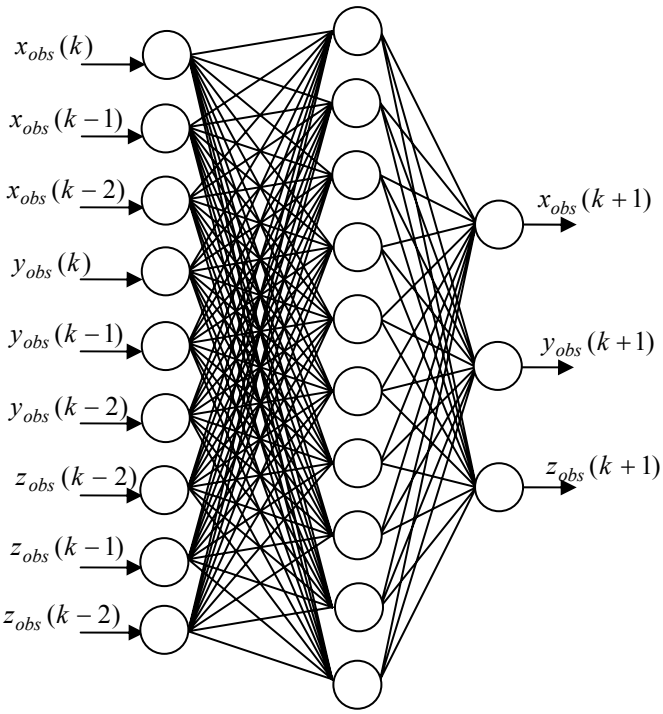

Fig. 2 Neural Network structure

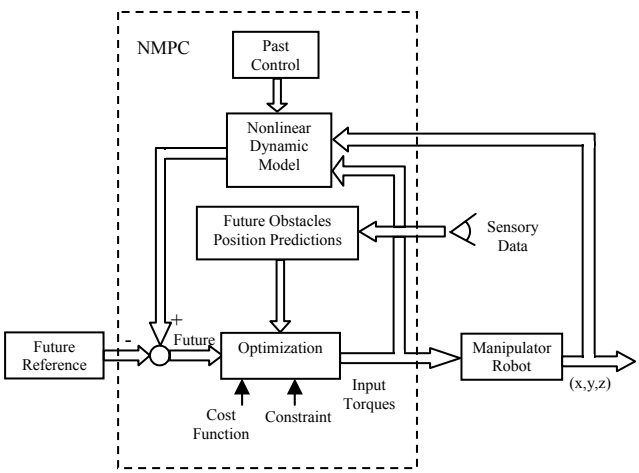

Fig. 3 Block diagram of NMPC

Next, constrains in the optimization problem is considered. Considering the fact that the amplitude of input voltages is limited, one of the constrains is

$$V_{t\min} \le V_t \le V_{t\max} \tag{12}$$

where $V_{t\min}$ and $V_{t\max}$ stand for the lower and the upper bound of input voltages of servo DC motors, respectively.

Next, in a singular configuration, theoretically the joint velocities become infinite. However, in practice, these velocities are limited to

$$\dot{\theta}_{\min} \le \dot{\theta} \le \dot{\theta}_{\max} \tag{13}$$

where $\dot{\theta}_{\min}$ and $\dot{\theta}_{\max}$ are the lower and the upper bound of the joints velocity, respectively.

By incorporating constrains (12) and (13) into the cost function, the optimization problem will be solved.

6. Simulation Result

The simulated results of the proposed control method are presented in this section. Parameters of the robot manipulator and DC servomotors are given in Tables II and III, respectively. Moreover, the sampling rate T is equal to 0.5 sec. Furthermore, based on the robot parameters, it is assumed that $\dot{\theta}_{\min}$ and $\dot{\theta}_{\max}$ are equal to –400 and 400 degree/s, respectively.

It is assumed that the velocity and acceleration of the moving object and obstacles are within an acceptable range as compared to the manipulator joints limitations. To show the prediction accuracy of the NN, a highly nonlinear motion for the target and obstacles is considered here. The motion equation of the target and the obstacle is defined as

$$\begin{aligned} R_1(t) &= 0.1\times\cos(0.06\times\pi\times t)+0.5 \\ R_2(t) &= 0.2\times\cos(0.01\times\pi\times t) \\ x_{obs}(t) &= R_1(t)\times\cos(0.01\times\pi\times t) \\ y_{obs}(t) &= R_1(t)\times\sin(0.01\times\pi\times t) \\ z_{obs}(t) &= -0.1+R_2(t)\times\sin(0.05\times\pi\times t) \end{aligned} \tag{14}$$

$$\begin{aligned} R_3(t) &= 0.2\times\cos(0.06\times\pi\times t)+0.5 \\ x_t(t) &= R_3(t)\times\cos(0.01\times\pi\times t) \\ y_t(t) &= R_3(t)\times\sin(0.01\times\pi\times t) \\ z_t(t) &= 0.5+0.2\times\cos(0.01\times t) \end{aligned} \tag{15}$$

where t is the time and $[x_{obs}(t), y_{obs}(t), z_{obs}(t)]$ and $[x_t(t), y_t(t), z_t(t)]$ are the obstacle and the target coordinates in 3D space, respectively. NNs consists of three layers: 9 neurons in the input layer, 10 neurons in the hidden layer, and 3 neuron in the output layer.

In Fig. 4, coordinates of the obstacle and in Fig. 5 path of the dynamic obstacle and the moving target in 3D space is shown. The input data applied to the NN is normalized to [-1, 1].

Simulation results for the case $Np = 5$, $Nc = 1$, and $N = 100$ are shown in Figs. 6 to 11. Fig. 6 shows that the end-effector reaches the target in a reasonable time. Figs. 7, 8, and 9 show positions, velocities, and input voltages of manipulator joints, respectively. Figs. 10 and 11 show the ability of NNs to predict coordinates of obstacle at $k = 4$ and $k = 40$, respectively. The maximum prediction error of NNs over the prediction horizon at each sampling time is shown in Fig. 12. As this figure shows, the NNs are able to learn the moving behaviour of the obstacle as the time passes. Fig. 13 shows that the training time for NNs at every sampling time is less than the sampling rate T, yielding a method that can be easily implemented in practice.

7. Conclusion

In this paper, for obstacle avoidance and catching moving targets by robot manipulators in non-stationary environments, the NMPC method was proposed. For this reason, two terms were introduced in the cost function, one for the tracking problem and the other one for the obstacle avoidance. Moreover, by introducing constrains to the joints velocities, singularities were avoided. To avoid robot collision with moving obstacles and to grasp moving targets, the future position of obstacles and moving targets in 3D space were predicted using artificial neural networks. Moreover, by using the online training scheme, no prior knowledge about obstacles and target motion is needed. The proposed control method can be implemented in humanoid robots in order to control arms of the robot in a similar way a human does.

TABLE II
PARAMETERS OF ROBOT MANIPULATOR

Link	1	2	3	4
l **(m)**	1	0.5	0.4	0.3
m **(kg)**	1	0.5	0.4	0.3

TABLE III
DC SERVO MOTORS PARAMETERS

Motor	1	2	3	4
$\mathbf{R}_a$	6.51	6.51	6.51	6.51
$\mathbf{K}_E$	0.7	0.7	0.7	0.7
$\mathbf{K}_T$	0.5	0.5	0.5	0.5
$\mathbf{B}_m$	64×10^{-4}	64×10^{-4}	64×10^{-4}	64×10^{-4}
$\mathbf{J}_m$	0.2	0.2	0.2	0.2
R	1:100	1:100	1:10	1:10
$\mathbf{V}_t$	24	24	24	24

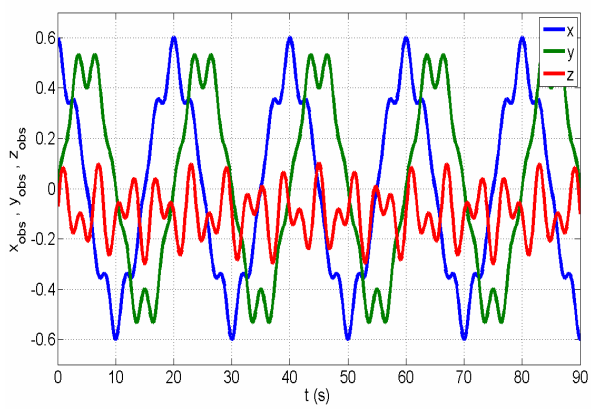

Fig. 4 Coordinates of moving obstacle

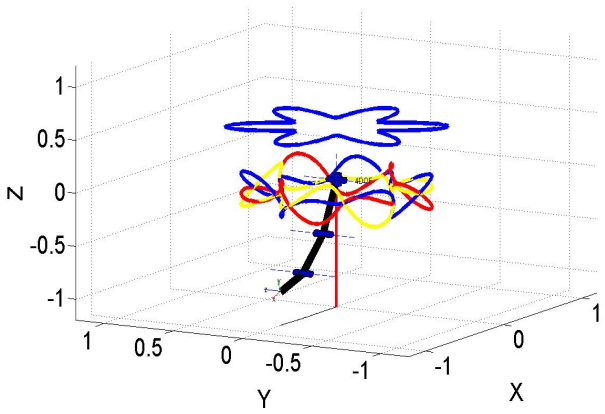

Fig. 5 path of dynamic obstacle and target

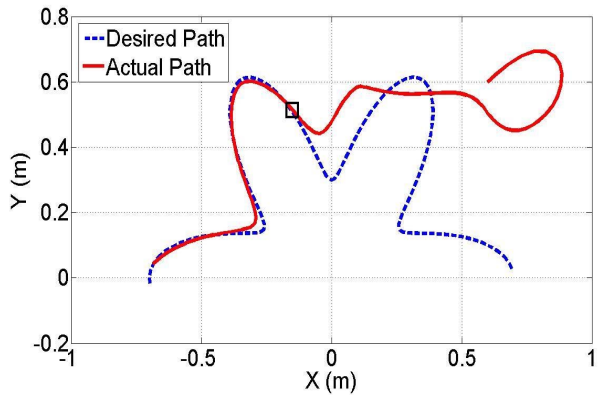

Fig. 6 Desired and actual end-effector path

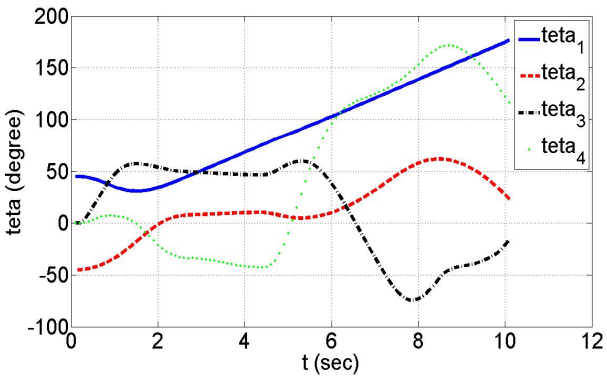

Fig. 7 Positions of manipulator joints

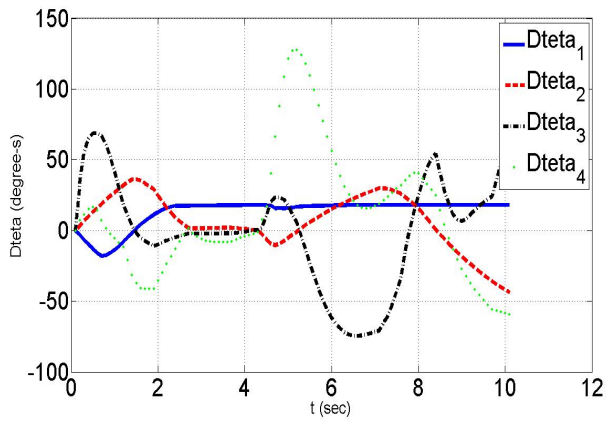

Fig. 8 Velocities of manipulator joints

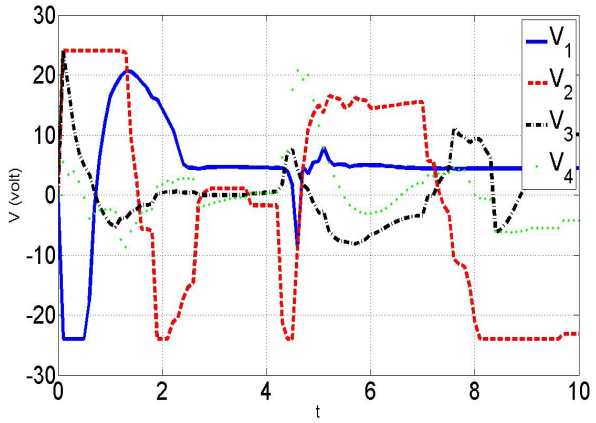

Fig. 9 Input voltages of servo DC motors

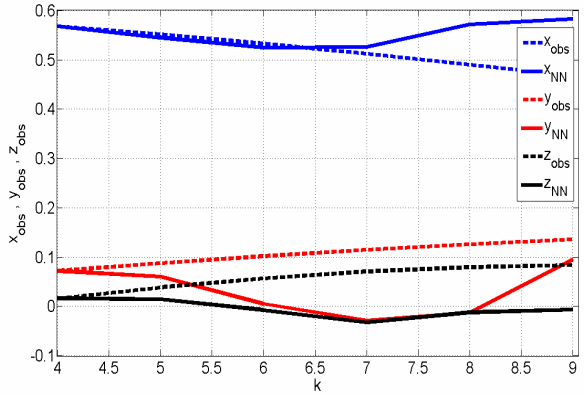

Fig. 10 neural network prediction over prediction horizon at sampling time $k = 4$

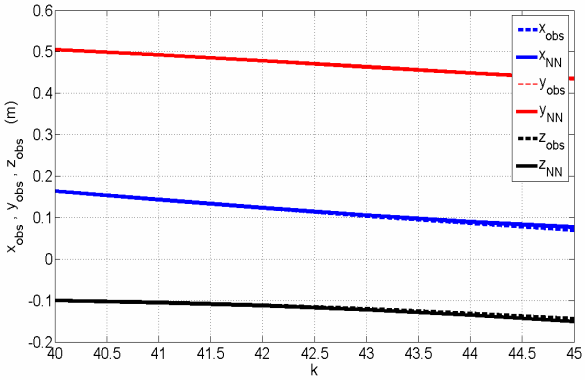

Fig. 11 neural network prediction over prediction horizon at sampling time $k = 40$

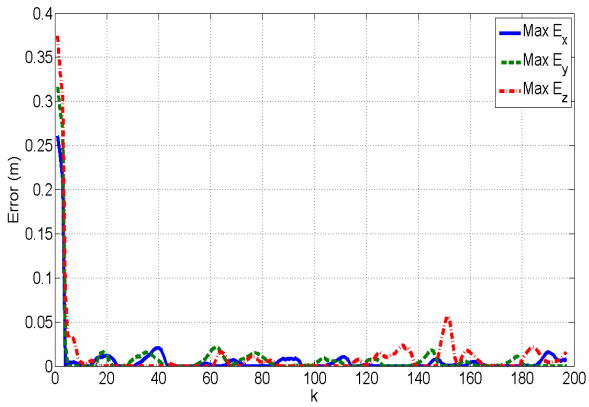

Fig. 12 maximum prediction error of neural network over prediction horizon at each sampling time

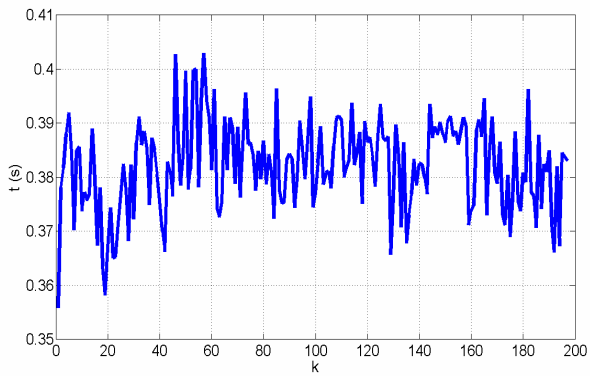

Fig. 13 training time for neural network at every sampling time

References

[1] E. S. Conkur, "Path planning using potential fields for highly redundant manipulators", *Robotics and Autonomous Systems*, Vol. 52, pp. 209–228, 2005.

[2] J. L. Chen, J. S. Liu, W. C. Lee, T. C. Liang, "On-line multi-criteria based collision-free posture generation of redundant manipulator in constrained workspace", *Robotica*, Vol. 20, pp. 625–636, 2002.

[3] P. Chiacchio, S. Chiaverini, L. Sciavicco, B. Siciliano, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy", *International Journal of Robotics Research*, Vol. 10, pp. 410-426, 1991.

[4] Y. Nakamuro, *Advanced Robotics Redundancy and Optimization*, Addison-Wesley Publishing Co., 1991.

[5] C. L. Boddy and J. D. Taylor,"Whole arm reactive collision avoidance control of kinematically redundant manipulators", *in Proc. of IEEE Int. Conf. on Robotics and Automation,* Vol. 3, pp. 382–387, Atlanta, Georgia, USA, May, 1993.

[6] T. Yoshikawa, "Analysis and control of robot manipulators with redundancy", *in Proc. of the First International Symposium in Robotic Research* , MIT Press, Cambridge, MA, pp. 439-446, 1993.

[7] C. Chevallereau, W. Khalil, "A new method for the solution of the inverse kinematics of redundant robots", *in Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 37–42, Washington, DC, USA, April, 1988.

[8] C.L.Shin, T.T.Lee and W.A.Gruver, " A unifiedapproach for robot motion planning with moving polyhedral obstacles", *IEEE Trans. Syst. Man Cyber.,* Vol. 20, pp. 903-915, 1990.

[9] B.H.Lee, " Constraint identification in time-varying obstacle avoidance for mechanical manipulators", *IEEE Trans. Syst. Man Cyber.,* Vol. 19, pp. 140-143, 1989.

[10] F. Valle, F. Tadeo, T. Alvarez, "Predictive control of robotic manipulators", *in Proc. of IEEE Int. Conf. on Control Applications*, pp. 203-208, Glasgow, Scotland, UK, September, 2002.

[11] J. K. Kim, M. C. Han,"Adaptive robust optimal predictive control of robot manipulators", *30th Annual Conf. of the IEEE Industrial Electronics Society*, Busan, Korea, November, 2004.

[12] A. Vivas, V. Mosquera, "Predictive functional control of a PUMA robot", *ACSE Conf.*, CICC, Cairo, Egypt, December, 2005.

[13] W. Wroblewski, "Implementation of a model predictive control algorithm for a 6dof Manipulator-simulation results", *Fourth Int. Workshop on Robot Motion and Control*, Puszczykowo, Poland, June, 2004.

[14] R. Hedjar, R. Toumi, P. Boucher, D. Dumur, S. Tebbani, "Finite horizon non linear predictive control with integral action of rigid link manipulators", *IEEE Conference on Control Applications, Int. J. Appl*. Math. Comput. Sci., Vol. 15, No. 4, pp. 101-113, août, Canada, 2005.

[15] Ph. Poignet, M. Gautier, "Nonlinear model predictive control of a robot manipulator", *in Proc. of 6th Int. Workshop on Advanced Motion Control*, pp. 401-406, Nagoya,Japan, March, 2000.

[16] R. Hedjar, R. Toumi, P. Boucher, D. Dumur, "Feedback nonlinear predictive control of rigid link robot manipulators", *American Control Conference*, Anchorage, Alaska, May, 2002.

[17] F. L. Lewis, C. T. Abdallah, and D. N. Dawson, *Control of Robot Manipulators Theory and Practice*, Marcel Dekker Inc, 2004.

[18] T. Yoshikawa, *Foundation of Robotics, Analysis and Control*, the MIT Press, Boston, USA, 1990.

[19] F. Allgower, R. Findeisen, Z. K. Nagy, "Nonlinear model predictive control: from theory to application", *J. Chin. Inst. Chem. Engrs*, Vol. 35, No. 3, pp. 299-315, 2004.

[20] R. Findeisen, L. Imsland, F. Allgower, and B. A. Foss, "State and output feedback nonlinear model predictive control: an overview", *European Journal of Control*, Vol. 9, pp. 190-206, 2003.

[21] R. Fletcher, *Practical methods of optimization*, John Wiley & Sons, 1987.

APPENDIX

$$G(1,1)=0$$

$$G(1,2)=\left(\frac{1}{2}\mathrm{m}_2 gl_2+\mathrm{m}_3 gl_2+\mathrm{m}_4 gl_2\right)\mathrm{c}_2+\left(\frac{1}{2}\mathrm{m}_3 gl_3+\mathrm{m}_4 gl_3\right)\mathrm{c}_{23}+\frac{1}{2}\mathrm{m}_4 gl_4\mathrm{c}_{234}$$

$$G(3,1)=\left(\frac{1}{2}\mathrm{m}_3 gl_3+\mathrm{m}_4 gl_3\right)\mathrm{c}_{23}+\frac{1}{2}\mathrm{m}_4 gl_4\mathrm{c}_{234}$$

$$G(4,1)=\frac{1}{2}\mathrm{m}_4 gl_4\,\mathrm{c}_{234} \tag{A.1}$$

$$M(1,1)=\frac{2}{3}m_1 l_1^2+\left(\frac{1}{3}m_2 l_2^2+m_3 l_2^2+m_4 l_2^2\right)\mathrm{c}_2^2+\left(\frac{1}{3}m_3 l_3^2+m_4 l_3^2\right)\mathrm{c}_{23}^2+\frac{1}{3}m_4 l_4^2 c_{234}^2$$
$$+\left(m_3 l_3 l_2+2m_4 l_3 l_2\right)\mathrm{c}_{23}\mathrm{c}_2+m_4 l_4 l_3 c_{234}c_{23}+m_4 l_4 l_2 c_{234}\mathrm{c}_2$$

$$M(2,2)=\frac{1}{3}m_2 l_2^2+m_3 l_2^2+\frac{1}{3}m_3 l_3^2+m_4 l_3^2+m_4 l_2^2+\frac{1}{3}m_4 l_4^2+m_4 l_4 l_2+m_4 l_3 l_4\mathrm{c}_4$$
$$+\left(m_3 l_3 l_2+2m_4 l_2 l_3\right)\mathrm{c}_3$$

$$M(3,3)=\frac{1}{3}m_3 l_3^2+m_4 l_3^2+\frac{1}{3}m_4 l_4^2+m_4 l_3 l_4\mathrm{c}_4$$

$$M(4,4)=\frac{1}{3}m_4 l_4^2$$

$$M(2,3)=M(3,2)=\frac{1}{3}m_3 l_3^2+m_4 l_3^2+\frac{1}{2}m_4 l_4 l_2+\frac{1}{3}m_4 l_4^2+\left(\frac{1}{2}m_3 l_3 l_2+m_4 l_2 l_3\right)\mathrm{c}_3$$
$$+m_4 l_4 l_3\,\mathrm{c}_4$$

$$M(2,4)=M(4,2)=\frac{1}{2}m_4 l_4 l_2+\frac{1}{3}m_4 l_4^2+\frac{1}{2}m_4 l_4 l_3\mathrm{c}_4$$

$$M(3,4)=M(4,3)=\frac{1}{3}m_4 l_4^2+\frac{1}{2}m_4 l_3 l_4\mathrm{c}_4$$

$$M(1,2)=M(2,1)=0$$
$$M(1,3)=M(3,1)=0 \tag{A.2}$$
$$M(1,4)=M(4,1)=0$$

$$C(1,1)=\begin{pmatrix}-\frac{1}{3}m_2 l_2^2\mathrm{s}_{22}-m_3 l_2^2\mathrm{s}_{22}-m_3 l_2 l_3\mathrm{s}_{223}-\frac{1}{3}m_3 l_3^2\mathrm{s}_{2233}-m_4 l_3^2\mathrm{s}_{2233}-m_4 l_2^2\mathrm{s}_{22}\\ -2m_4 l_2 l_3\mathrm{s}_{223}-m_4 l_4 l_3\mathrm{s}_{22334}-m_4 l_4 l_2\mathrm{s}_{2234}-\frac{1}{3}m_4 l_4^2\mathrm{s}_{223344}\end{pmatrix}\dot{\theta}_1\dot{\theta}_2$$
$$+\begin{pmatrix}-m_3 l_2 l_3\mathrm{s}_{23}\mathrm{c}_2-\frac{1}{3}m_3 l_3^2\mathrm{s}_{2233}-m_4 l_3^2\mathrm{s}_{2233}-2m_4 l_3 l_2\mathrm{s}_{23}\mathrm{c}_2-m_4 l_3 l_4\mathrm{s}_{22334}\\ -m_4 l_2 l_4\mathrm{s}_{234}\mathrm{c}_2-\frac{1}{3}m_4 l_4^2\mathrm{s}_{223344}\end{pmatrix}\dot{\theta}_1\dot{\theta}_3$$
$$+\left(-m_4 l_4 l_3\mathrm{s}_{234}c_{23}-m_4 l_4 l_2\mathrm{s}_{234}c_2-\frac{1}{3}m_4 l_4^2\mathrm{s}_{223344}\right)\dot{\theta}_1\dot{\theta}_4$$

$$C(2,1)=\begin{pmatrix}\frac{1}{6}m_2 l_2^2\mathrm{s}_{22}+\frac{1}{2}m_3 l_2^2\mathrm{s}_{22}+\frac{1}{2}m_3 l_2 l_3\mathrm{s}_{223}+\frac{1}{6}m_3 l_3^2\mathrm{s}_{2233}+\frac{1}{2}m_4 l_3^2\mathrm{s}_{2233}\\ +\frac{1}{2}m_4 l_2^2\mathrm{s}_{22}+m_4 l_2 l_3\mathrm{s}_{223}+\frac{1}{2}m_4 l_4 l_3\mathrm{s}_{22334}+\frac{1}{2}m_4 l_2 l_4\mathrm{s}_{2234}\\ +\frac{1}{6}m_4 l_4^2\mathrm{S}_{223344}\end{pmatrix}\dot{\theta}_1^2$$
$$+\left(-\frac{1}{2}m_3 l_2 l_3\mathrm{s}_3-m_4 l_3 l_2\mathrm{s}_3\right)\dot{\theta}_3^2+\left(-\frac{1}{2}m_4 l_4 l_3\mathrm{s}_4\right)\dot{\theta}_4^2+\left(-m_4 l_3 l_4\mathrm{s}_4\right)\dot{\theta}_2\dot{\theta}_4$$
$$+\left(-m_3 l_2 l_3\mathrm{s}_3-2m_4 l_3 l_2\mathrm{s}_3\right)\dot{\theta}_2\dot{\theta}_3+\left(-m_4 l_3 l_4\mathrm{s}_4\right)\dot{\theta}_3\dot{\theta}_4$$

$$C(3,1)=\begin{pmatrix}\frac{1}{2}m_3 l_3 l_2\mathrm{s}_{23}c_2+\frac{1}{6}m_3 l_3^2\mathrm{s}_{2233}+\frac{1}{2}m_4 l_3^2\mathrm{s}_{2233}+m_4 l_2 l_3\mathrm{s}_{23}c_2+\frac{1}{2}m_4 l_4 l_3\mathrm{s}_{22334}\\ +\frac{1}{2}m_4 l_4 l_2\mathrm{s}_{234}c_2+\frac{1}{6}m_4 l_4^2\mathrm{s}_{223344}\end{pmatrix}\dot{\theta}_1^2$$
$$+\left(\frac{1}{2}m_3 l_3 l_3\mathrm{s}_3+m_4 l_3 l_2\mathrm{s}_3\right)\dot{\theta}_2^2+\left(-m_4 l_3 l_4\mathrm{s}_4\right)\dot{\theta}_2\dot{\theta}_4+\left(-\frac{1}{2}m_4 l_3 l_4\mathrm{s}_4\right)\dot{\theta}_4^2$$
$$+\left(-m_4 l_3 l_4 s_4\right)\dot{\theta}_3\dot{\theta}_4$$

$$C(4,1)=\left(\frac{1}{2}m_4 l_4 l_3\mathrm{s}_{234}c_{23}+\frac{1}{2}m_4 l_4 l_2\mathrm{s}_{234}\mathrm{c}_2+\frac{1}{6}m_4 l_4^2\mathrm{s}_{223344}\right)\dot{\theta}_1^2+\left(\frac{1}{2}m_4 l_4 l_3\mathrm{s}_4\right)\dot{\theta}_2^2$$
$$+\left(\frac{1}{2}m_4 l_4 l_3\mathrm{s}_4\right)\dot{\theta}_3^2+\left(m_4 l_4 l_3\mathrm{s}_4\right)\dot{\theta}_3\dot{\theta}_4 \tag{A.3}$$

where, l_i and m_i ($i=1,\ldots,4$) are the length and mass of the ith link, respectively, θ_i and $\dot{\theta}_i$ are the angular position and the angular velocity of the ith joint, respectively, and $c_i=\cos(\theta_i)$, $s_i=\sin(\theta_i)$, $c_{ij}=\cos(\theta_i+\theta_j)$, $s_{ij}=\sin(\theta_i+\theta_j)$, and so forth.