Adaptive Neuro-NMPC Control of Redundant Robotic Manipulators for Path Tracking and Obstacle Avoidance

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

***Abstract*—This paper presents a nonlinear model predictive control (NMPC) method with adaptive neuro-modelling for redundant robotic manipulators. Using the NMPC, the end-effector of the robot tracks a predefined geometry path in the Cartesian space without colliding with obstacles in the workspace and at the same time avoiding singular configurations of the robot. Furthermore, using the neural network for the model prediction, no knowledge about system parameters is necessary; hence, yielding robustness against changes in parameters of the system. Numerical results for a 4DOF redundant spatial manipulator actuated by DC servomotors shows effectiveness of the proposed method.**

***Keywords*—Robotic Manipulator, Path Tracking, Obstacle Avoidance, NMPC, Neural Networks.**

1. Introduction

One of the major challenges in robotic manipulators is the path tracking and the obstacle avoidance. The main reason for development of manipulator robots is to replace human in doing long and repetitive operations and unhealthy tasks. In this case, robots must be capable to operate effectively in complex environments. In particular, these robots are needed to track a predefined path in such a way that no collision with obstacles in the environment occurs. 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 path in the Cartesian space, the robot can track it in many different configurations; among these, the collision free and singular free tracking must be selected. Finding feasible path for joints of redundant manipulators for a given end-effector path is called the redundancy resolution [1]. Redundancy resolution and obstacle avoidance are considered in papers. Using the gradient projection technique, redundancy and obstacle avoidance can be solved [2]. In task-priority redundancy resolution technique, the tasks are performed with 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 and extended Jacobin techniques, which are used for redundancy solution, are time consuming [5]-[7]. Optimization techniques, which minimize the cost function subject to constraints, like the end-effector path tracking and obstacle avoidance, are not suitable for on-line applications [4].

In this paper, a Nonlinear Model Predictive Control (NMPC) method is presented for redundancy resolution considering obstacles and singularities avoidance. Although the 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 on the joint space control and on the end-effector coordinating. The linear MPC is used in [8]-[10] and NMPC is used in [11]-[14] for joint space control of manipulators.

Using NMPC, the input voltages of DC servomotors of joints are obtained in such a way that the end-effector of the redundant manipulator tracks a given path in the Cartesian space considering obstacles and singularities avoidance.

When the robot dynamic are uncertain, adaptive control laws must be used in order to tackle this problem. Adaptive control methods using neural network are implemented in literatures for controlling highly uncertain and nonlinear systems [15, 16]. In this paper, neural networks are used as a prediction system in NMPC. Using neural network no knowledge about system parameters is necessary. Moreover, system robustness against changes in parameters is obtained.

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 method. Section 4 explains how to obtain prediction model of the robot using neural networks. Section 5 implements the NMPC for the path tracking and obstacles avoidance for 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]. Then, the position of the end-effector in the Cartesian space can be calculated in terms of joint angles as fallows:

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

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

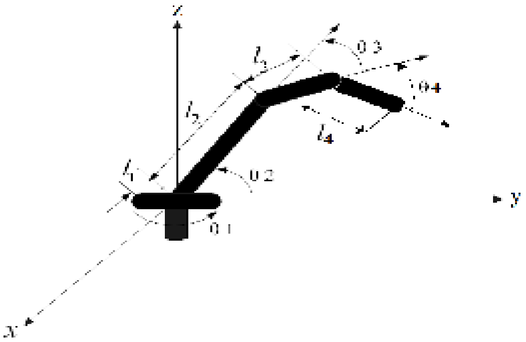

Fig. 1 Schematic of a 4DOF spatial manipulator

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

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.

Friction for joint *i* is defined as [18]

$$D(i) = D_v \dot{q}_i + D_d \,\mathrm{sgn}(\dot{q}_i) \tag{3}$$

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 the links can be expressed in the form [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.

Since dynamic equations of DC servomotors are considered here, the relationship between the robot joint and the motor-shaft can be represented as

$$\mathbf{R} = \frac{\boldsymbol{\theta}}{\boldsymbol{\theta}_m} = \frac{\boldsymbol{\tau}_m}{\boldsymbol{\tau}} \tag{5}$$

where $\mathbf{r} \in R^{n\times n}$ 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]

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

Using Eq. (5) to eliminate q_m and t_m in Eq. (6) and then substituting for t using Eq. (2), the governed equation of *n*-link robot manipulator including actuator dynamics can be obtained as

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

According to Eq. (7), the armature input voltages are considering as control efforts, respectively. The detailed parameters of the robot manipulator and DC servomotors are given as Table II and Table III, respectively.

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

Link	α^o	a	d	θ^o
1	90	0	0	q_1
2	0	l_2	0	q_2
3	0	l_3	0	q_3
4	0	l_4	0	q_4

TABLE II
MANIPULATOR ROBOT PARAMETERS

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

3. Model Predictive Control

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

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

To incorporate feedbacks, only the first member of the obtained input is applied to the system until the next sampling time [19]. Using the new measurement 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

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

such that the following conditions are satisfied:

$$\begin{aligned}
&x(k\,|\,k) = x_0 \\
&u(k+j\,|\,k)) = u(k+N_u\,|\,k))\,,\ j \ge N_C \\
&x(k+j+1\,|\,k) = f_d(x(k+j\,|\,k), u(k+j\,|\,k)) \\
&y(k+j+1\,|\,k) = h_d(x(k+j+1\,|\,k)) \\
&x_{\min} \le x(k+j+1\,|\,k) \le x_{\max} \\
&u_{\min} \le u(k+j\,|\,k) \le u_{\max}
\end{aligned} \tag{9}$$

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. The 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 nonlinear model or consider non-quadratic cost function and nonlinear constrains on inputs and states are called Nonlinear MPC [19].

The optimization problem (8) must be solved at each sampling time k, yielding a sequence of optimal control laws as $\{\mathbf{u}^*(k\,|\,K), \mathbf{K}, \mathbf{u}^*(k+N_u-1)\}$. For optimization, the sequential quadratic programming method is used in this paper [21].

4. Prediction Model Using Artificial Neural Network

As it was explained, a model of the system is needed for prediction in MPC. In this paper, a Multilayer Perceptron (MLP) is used for modeling nonlinear dynamics of the robot in MPC. Using the neural network for model prediction, no prior knowledge about system parameters is needed. The prediction is based on the input voltage, angular position, and angular velocity of joints at the current sampling time. The output of the predictor is the angular position of the joint at the next sampling time. Using this structure, a one-step-ahead prediction of joints position can be obtained. However, in the predictive control, multi-step predictions over the prediction horizon are needed. By applying one-step prediction recursively, multi-step prediction can be achieved. In this case, the outputs of the neural network are considered as inputs for the next step. Using the neural network recursively, high accuracy of one-step-ahead prediction is needed. For this reason, one neural network is implemented for every link of the robot to predict the position of every joint angle. The structure of the employed MLP is shows in Fig. 2. Transfer functions for the hidden and the output layer neurons are of tangent hyperbolic and linear types, respectively. The inputs are the angular position and the angular velocity of joints, and the input voltage of the ith joint at the current sampling time [$v_{ti}(k)$, $q_1(k)$, $q_2(k)$, $q_3(k)$, $q_4(k)$, $Dq_1(k)$, $Dq_2(k)$, $Dq_3(k)$, $Dq_4(k)$] and the output is $q_i(k+1)$, which refers to the ith joint position at the next sampling time.

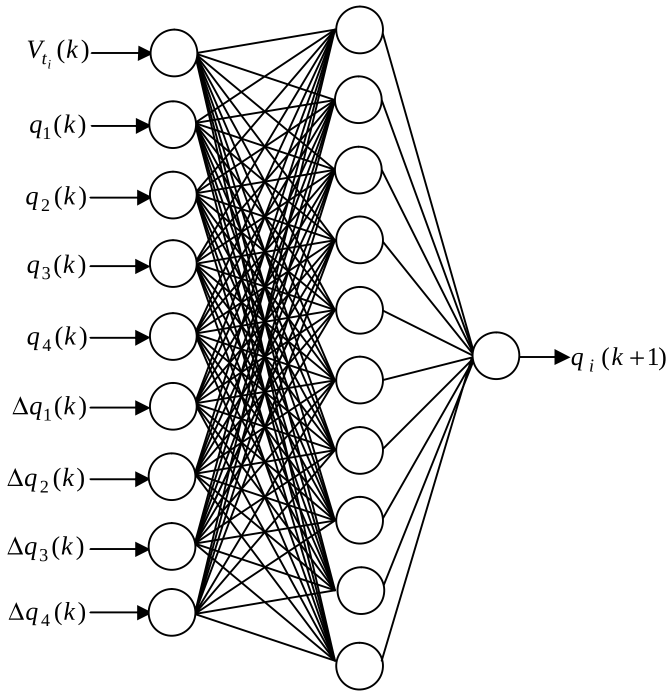

Fig. 2 Neural Network structure

The neural networks are trained offline first using the data gathered form the system. The main reason for this is that the weights of NNs are initialized with random numbers; hence, without any offline training the robot might collide with obstacles. The offline training does not need to be very accurate; just a rough familiarity of NNs with the robot behavior to avoid large initial tracking error and/or collision with obstacles suffices. It should be noted that the online training of NNs can cope with all changes in robot parameters, including the masse of links, the friction of joints and parameters of servomotors.

5. Path Tracking and Obstacle Avoidance Using NMPC

The purpose of the path tracking and obstacle avoidance of robot manipulators is to obtain a control law such that the end-effector tracks a given geometry path in the Cartesian space and at the same time to avoid collision of robot links with obstacles. To achieve this purpose, the NMPC method is implemented in this paper. Block diagram of NMPC is shown in Fig. 3.

According to the NMPC algorithm, an appropriate cost function must be determined in order to obtain the control law.

For path tracking, the cost function must be directly related to the tracking error. On the other hand, the cost function for the obstacles avoidance must be inversely related to the distance between the obstacle and the manipulator. Hence, in this paper, the cost function is defined as

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

where D_P is the Euclidean distance between the end-effector and the geometry path in the Cartesian space, D_O is the minimum Euclidean distance between the manipulator and obstacles; moreover, $Q \geq 0$ and $R \geq 0$ are the weighting parameters.

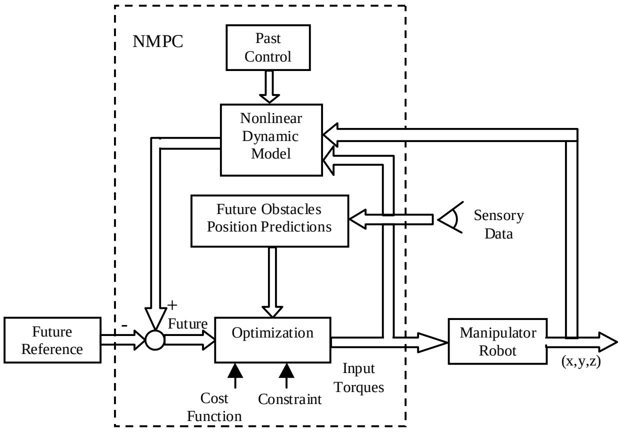

Fig. 3 Block diagram of NMPC

Notation $a(m|n)$ in (10) indicates the value of a at the instant m predicted at instant n. The 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 in (10), the two terms for the path tracking and the obstacle avoidance are normalized to [0 1].

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

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

where $V_{t\min}$ and $V_{t\max}$ stand for the lower and the upper bound of input voltages for servo DC motors, respectively (-24 V and 24 V as Table III shows).

Next, considering the fact that in singular configurations the joint velocities are infinite, the following constrain must be taken into account:

$$\dot{q}_{\min} \leq \dot{q} \leq \dot{q}_{\max} \tag{13}$$

where $\dot{q}_{\min}$ and $\dot{q}_{\max}$ are the lower and the upper bound of the joints velocities (which are -400 and 400 deg./s, considering the robot and motors parameters), respectively.

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

6. Simulation Results

The simulated results of the proposed control method are presented in this section. A rectangular path in the Cartesian space along with obstacles inside the work space is considered. The developed neural network for every link consists of three layers: 9 neurons in the input layer, 10 neurons in the hidden layer, and 1 neuron in the output layer.

Simulation results for the case of sampling time of 0.5 sec., $N_p = 5$, $N_c = 1$, without any changes in robot parameters are shown in Figs. 4 to 9.

Figs. 10 and 11 show the results when an additional 20 kg mass at $t = 25\,\text{sec.}$ is added to the fourth link of the robot.

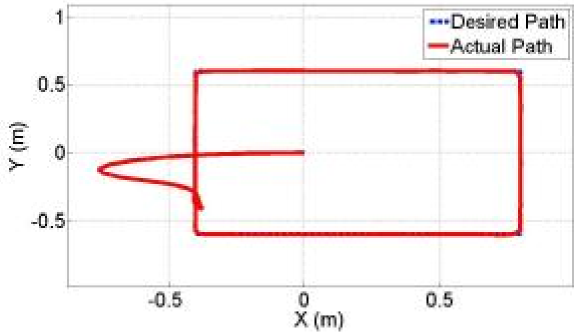

Fig. 4 Desired and actual end-effector path

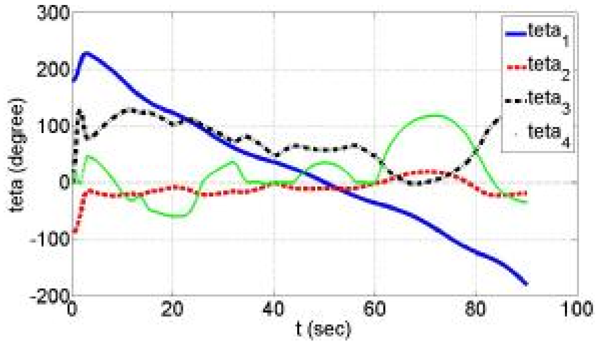

Fig. 5 Positions of manipulator joints

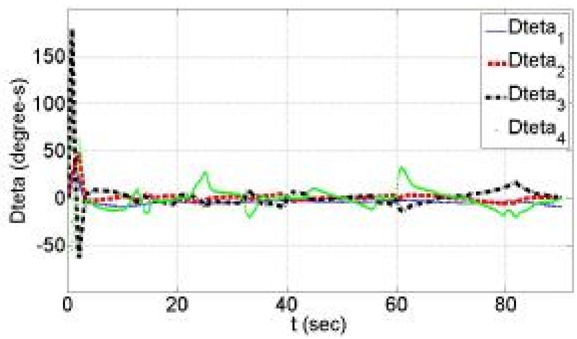

Fig. 6 Velocities of manipulator joints

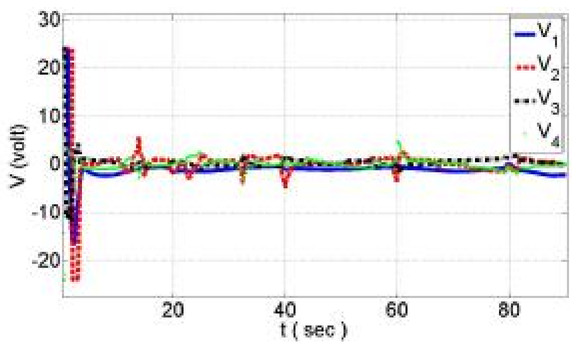

Fig. 7 Input voltages of servo DC motors

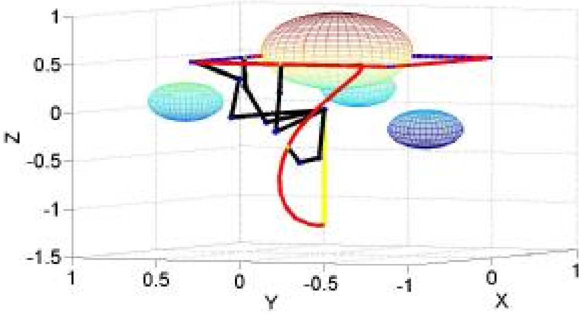

Fig. 8 Path following of a 4DOF manipulator with four obstacles on the workspace

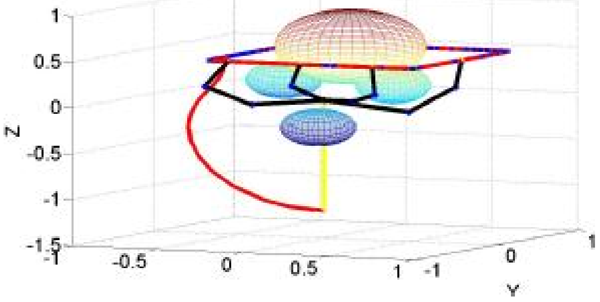

Fig. 9 The path tracking of Fig.8 from a different view angle

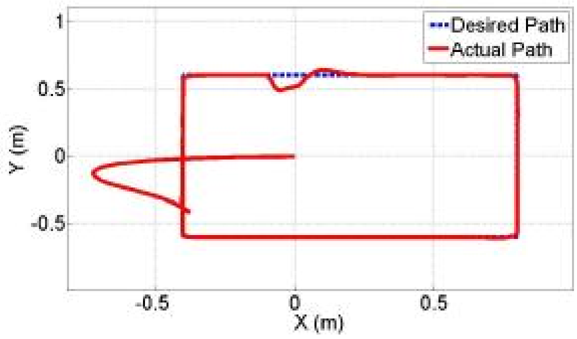

Fig. 10 End-effector path tracking with sudden changes in the mass of link 4

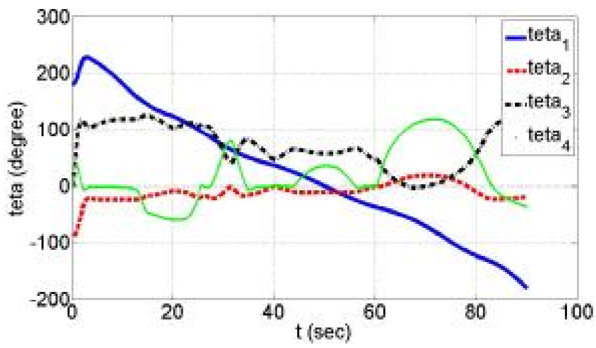

Fig. 11 Positions of manipulator joints

The optimization problem is solved using the SQP method. For more information refer to the MATLAB routine "fmincon".
Fig. 12 shows the simulation runtime for the proposed method. As this figure shows, the runtime is mostly less than the sampling time. However, it should be noted that: 1) the simulations are performed in MATLAB software, which requires more time than a lower level computer program such as C++, 2) in simulations, solving the robot equations also incurs some

computational time, which is not required in practice, 3) some aspects of computations, such training of NNs, can be processed in parallel. Hence, the simulation run time can be much less than Fig. 12. Therefore, in practice, the proposed method can be successfully applied to robot manipulators using conventional digital computers.

7. Conclusion

In this paper, to achieve path tracking and obstacle avoidance for robotic manipulators, the NMPC method was employed. For this reason, two terms were introduced in the cost function, one for the tracking problem and one for the obstacle avoidance and by introducing constrains to joints velocities, singularities were avoided. Moreover, neural networks are employed for adaptive model prediction that can cope with changes in system parameters. The main advantage of NNs is that no prior knowledge about nonlinear equations of the robot or its parameters is needed.

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] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

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

[15] S.Huang, K.KiongTan, T.Heng Lee, A.S.Putra, “Adaptive Control of Mechanical Systems Using Neural Networks”, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol 37, Issue 5 ,pp. 897 - 903,Sept, 2007.

[16] Hayakawa, T. Haddad, W.M. Hovakimyan, N. , “Neural Network Adaptive Control for a Class of Nonlinear Uncertain Dynamical Systems With Asymptotic Stability Guarantees”, IEEE Transactions on Neural Networks, Vol 19 , Issue: 1 , pp. 80 - 89 ,Jan. 2008

[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.

[22] Li.Xin Wang, *A Course in Fuzzy System and Control*, Prentice-Hall International, Inc, 1997.

APPENDIX

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

$$G(1,2) = \left(\frac{1}{2}\mathrm{m}_2 g l_2 + \mathrm{m}_3 g l_2 + \mathrm{m}_4 g l_2\right)\mathrm{c}_2 + \left(\frac{1}{2}\mathrm{m}_3 g l_3 + \mathrm{m}_4 g l_3\right)\mathrm{c}_{23} + \frac{1}{2}\mathrm{m}_4 g l_4 \mathrm{c}_{234}$$

$$G(3,1) = \left(\frac{1}{2}\mathrm{m}_3 g l_3 + \mathrm{m}_4 g l_3\right)\mathrm{c}_{23} + \frac{1}{2}\mathrm{m}_4 g l_4 \mathrm{c}_{234}$$

$$G(4,1) = \frac{1}{2}\mathrm{m}_4 g l_4 \mathrm{c}_{234} \qquad \text{(A.1)}$$

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

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

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

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

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

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

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

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

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

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

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

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

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