2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Path Tracking and Obstacle Avoidance for Redundant Robotic Arms Using Fuzzy NMPC

Ashkan M. Jasour, Mohammad Farrokhi

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) for redundant robotic arms. Using NMPC, the end-effector of robotic arm tracks a predefined geometry path in the Cartesian space in such a way that no collision with obstacles in the workspace and no singular configurations for robot occurs. Nonlinear dynamic of the robot including actuators dynamic is also considered. Moreover, the on-line tuning of the weights in NMPC is performed using the fuzzy logic. The proposed method automatically adjusts the weights in cost function in order to obtain good performance. Numerical simulations of a 4DOF redundant spatial manipulator actuated by DC servomotors shows effectiveness of the proposed method.**

I. Introduction

Today, robotic 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 and repetitive operations and unhealthy tasks. 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 an infinity 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 redundancy resolution [1]. Redundancy resolution and obstacle avoidance are already considered in papers. With gradient projection technique, redundancy can be solved considering obstacle avoidance [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 is 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, 6, 7]. Optimization techniques, which minimize a cost function subject to constraints, like end-effector path tracking and obstacle avoidance, are not suitable for on-line applications [4].

In this paper, Nonlinear Model Predictive Control (NMPC) method is presented for redundancy resolution, considering 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 [8, 9, 10] and NMPC is used in [11, 12, 13, 14] for joint space control of manipulators.

In this paper, using NMPC, the input voltages of DC servomotors of joints are obtained in such a way that the end-effector of a redundant manipulator tracks a given path in the Cartesian space, considering obstacles and singularity avoidance. Moreover, using fuzzy logic, an automatic mechanism for the on-line tuning of the weights for the path tracking and obstacle avoidance terms in the cost function is proposed.

This paper is organized as follows: Section II presents nonlinear dynamic of 4DOF spatial redundant manipulator including the actuators dynamic. Section III describes the nonlinear predictive control. In Section IV, NMPC is implemented for path tracking and obstacle avoidance of a 4DOF manipulator. Section V presents the proposed modified NMPC using fuzzy logic. Conclusions are drawn in Section VI.

II. Manipulator Robot Dynamic

Schematic diagram of a 4DOF spatial redundant manipulator robot is shown in Fig. 1.

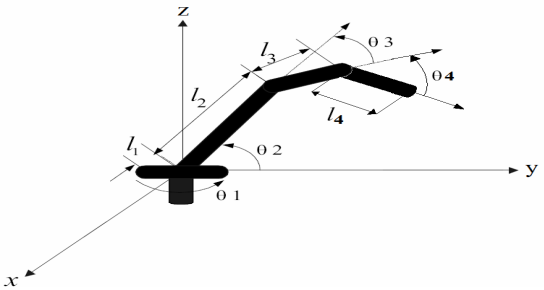

Fig. 1 Schematic of a 4DOF spatial manipulator

According to Denavit-Hartenberg parameters [15] of the shown robot in Table I, the position of the end-effector in

Cartesian space can be calculated in terms of joint angles as fallows:

TABLE I
DENAVIT-HARTENBERG PARAMETERS OF ROBOT 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

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} c_1(l_4c_{234} + l_3c_{23} + l_2c_2) \\ s_1(l_4c_{234} + l_3c_{23} + l_2c_2) \\ l_4s_{234} + l_3s_{23} + l_2s_2 \end{bmatrix} \quad (1)$$

The dynamic model of the robot manipulator can be obtained using the Lagrangian method as follows [15, 16]:

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

where θ_i is the angle of the i^{th} joint, $M(\theta) \in R^{n\times n}$ is the symmetric and positive definite inertia matrix, $C(\theta,\dot{\theta}) \in R^n$ is the centrifugal and coriolis force vector, $G(\theta) \in R^n$ is the gravity vector, $D(\dot{\theta}) \in R^n$ is the vector for joints friction of the links, $\tau \in R^n$ 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 as follow [15]:

$$D(i) = D_V\dot{\theta}_i + D_d\,\text{sgn}(\dot{\theta}_i) \quad (3)$$

where D_v is the coefficient of the viscous friction and D_d is the coefficient of the dynamic friction.

The dynamics of the armature-controlled DC servomotors that drive the links are expressed in the following form [15]:

$$\begin{aligned} \tau_e &= K_T i_a \\ \tau_e &= J_m\ddot{\theta}_m + B_m\dot{\theta}_m + \tau_m \\ V_t &= R_a i_a + L_a\dot{i}_a + K_E\dot{\theta}_m \end{aligned} \quad (4)$$

Where $\tau_e \in R^n$ is the vector of electromagnetic torque, $K_T \in R^{n\times n}$ is the diagonal matrix of the motor torque constant, $i_a \in R^n$ is the vector of armature currents, $J_m \in R^{n\times n}$ is the diagonal matrix of the moment inertia, $B_m \in R^{n\times n}$ is the diagonal matrix of torsional damping coefficients, $\theta_m, \dot{\theta}_m, \ddot{\theta}_m \in R^n$ denote the vectors of motor shaft positions, velocities and acceleration, respectively, $\tau_m \in R^n$ is the vector of load torque, $V_t \in R^n$ is the vector of armature input voltages, $R_a \in R^{n\times n}$ is the diagonal matrix of armature resistance, $L_a \in R^{n\times n}$ is the diagonal matrix of armature inductance and $K_E \in R^{n\times n}$ is the diagonal matrix of the back electromotive force (EMF) coefficients.

In order to apply the DC servomotors for actuating an n-link robot manipulator, a relationship between the robot joint and the motor-shaft can be represented as:

$$r = \frac{\theta}{\theta_m} = \frac{\tau_m}{\tau} \quad (5)$$

where $r \in R^{n\times n}$ is a diagonal positive definite matrix of the gear ratios for n joints. According to the fact that the armature inductance is small and negligible, the Eq. (4) can be expressed as follow [15]:

$$J_m\ddot{\theta}_m + (B_m + \frac{K_E K_T}{R_a})\dot{\theta}_m + \tau_m = \frac{K_T}{R_a}V_t \quad (6)$$

Using Eq. (5) to eliminate θ_m and τ_m in Eq. (6) and then substituting for τ from Eq. (2), the governed equation of n-link robot manipulator including actuator dynamics can be obtained as:

$$(J_m + r^2M)\ddot{\theta} + (B_m + \frac{K_E K_T}{R_a})\dot{\theta} + r^2(C + G + D) = \frac{rK_T}{R_a}V_t \quad (7)$$

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

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
R_a	6.51	6.51	6.51	6.51
K_E	0.7	0.7	0.7	0.7
K_T	0.5	0.5	0.5	0.5
B_m	64×10^{-4}	64×10^{-4}	64×10^{-4}	64×10^{-4}
J_m	0.2	0.2	0.2	0.2
R	1:100	1:100	1:10	1:10
V_t	24	24	24	24

III. MODEL PREDICTIVE CONTROL

Unlike classical control schemes, in which the control actions are taken based on the past output of the system, the MPC is a model-based optimal controller, which uses predictions of the systems output to calculate the control law [17, 18].

At every sampling time k, based on measurements obtained at time k, the controller predicts the output of the system over prediction horizon N_P in future using 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 feedback, only the first member of the obtained input is applied to system until the next sampling time [17]. Using the new measurement at 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

$$\begin{aligned} & x(k \mid k) = x_0 \\ & u(k+j \mid k)) = u(k+N_C \mid k)) \text{ , } j \geq N_C \\ & x(k+j+1 \mid k) = f_d(x(k+j \mid k), u(k+j \mid k)) \\ & y(k+j+1 \mid k) = h_d(x(k+j+1 \mid k)) \\ & x_{\min} \leq x(k+j+1 \mid k) \leq x_{\max} \\ & u_{\min} \leq u(k+j \mid k) \leq u_{\max} \end{aligned} \tag{9}$$

Where $j \in [0 \text{ , } N_P\text{-}1]$, x and u are states and input of the system and the notation $a(m|n)$ indicates the value of a at the instant m predicted at instant n, x_0 is the initial condition and f_d and h_d are the model of the system used for prediction. $[x_{\min} \text{ , } x_{\max}]$ and $[u_{\min} \text{ , } u_{\max}]$ 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 the inputs and states are called Nonlinear MPC [17]. The optimization problem (8) must be solved at each sampling time k, yielding a sequence of optimal control law as $\{u^*(k \mid k),...,u^*(k+N_C \mid k)\}$. For optimization, the SQP method is used in this paper [19].

IV. 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 collision between the end-effector and links is avoided. To achieve this purpose, the NMPC is implemented in this section. Block diagram of NMPC is shown in Fig. 2. 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 have direct relation with the tracking error between the end-effector coordination and the given path in the Cartesian space; on the other hand, for obstacles avoidance the cost function must have inverse relation with the distance between the obstacle and the manipulator. One of the proper candidates for the cost function can be introduced as:

$$J = \sum_{j=1}^{N_P} D_P(k+j \mid k) Q D_P(k+j \mid k) + \frac{R}{D_O(k+j \mid k) D_O(k+j \mid k)} \tag{10}$$

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, notation $a(m|n)$ indicates the value of a at the instant m predicted at instant n and $Q \geq 0$, $R \geq 0$ are the weighting parameters. According to Eq. (10), the path tracking term of the cost function is described as distance but the obstacle avoidance term of the cost function is described as the inverse of distance. Hence, it is important to notice that the distance is bounded in the workspace, but the inverse of the distance is unbounded.

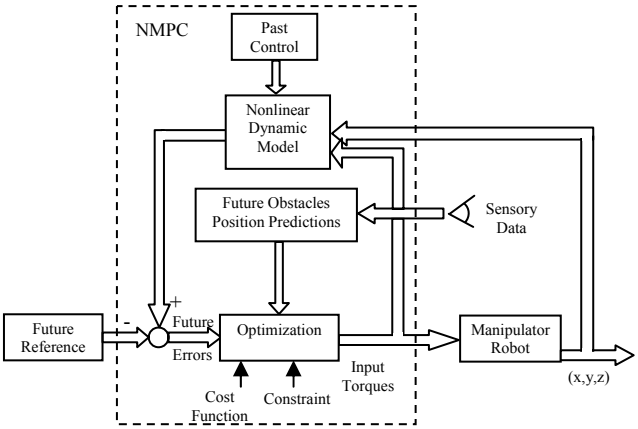

Fig. 2 Block diagram of NMPC

Therefore, combination of these two inconsistent terms as a cost function is not appropriate for an optimization problem. To tackle this problem, these two terms are normalized to [0 1] using a nonlinear map. Hence, the modified cost function takes the following form:

$$J = \sum_{j=1}^{N_P} Q\left(\frac{e^{D_P(k+j|k)} - e^{D_{P\min}}}{e^{D_{P\max}} - e^{D_{P\min}}}\right) + R\left(\frac{e^{-D_O(k+j|k)} - e^{-D_{O\max}}}{e^{-D_{O\min}} - e^{-D_{O\max}}}\right) \tag{11}$$

Where $[D_{P\min} \text{ , } D_{P\max}]$ and $[D_{O\min} \text{ , } D_{O\max}]$ 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. Predictive controller discussed in this paper uses a nonlinear dynamic model of the manipulator in the optimization of the cost function. Substituting $(\theta(k+1) - \theta(k))/T$ for $\dot{\theta}$ in the dynamic Eq. (7), a one-step ahead prediction for joints angle can be expressed as:

$$\theta(k+1) = f_d(\theta(k), V_t(k)) \tag{12}$$

where k is the sampling time and T is the sampling rate, which is equivalent to 0.5 s in this paper. Using forward kinematics as Eq. (1), a one-step ahead prediction of the end-effector position can be obtained. However, in the predictive control, multi-step predictions are used over prediction horizon by applying one-step prediction recursively. Next, constraints in the optimization problem are considered. Considering the fact that the amplitude of input voltages is limited, one of the constraints is:

$$V_{t\min} \leq V_t \leq V_{t\max} \tag{13}$$

where $V_{t\min}$ and $V_{t\max}$ stand for the lower and the upper bound of input voltages of servo DC motors, respectively (-24 and 24 as Table III shows). Besides, considering the fact that in a singular configuration, for the case of limited velocity for the end-effector, the joint velocities are infinite. Therefore, the following constraint must be taken into account:

$$\dot{\theta}_{\min} \leq \dot{\theta} \leq \dot{\theta}_{\max} \tag{14}$$

where $\dot{\theta}_{\min}$ and $\dot{\theta}_{\max}$ are the lower and the upper bound of the joints velocity, respectively, which are -400 and 400 degree/s, considering the robot and motors parameters. By

incorporating constrains (13) and (14) into the cost function, the optimization problem can be solved. Solving this optimization problem at each sampling time, the input voltages of DC servomotors of joints are obtained in such a way that the end-effector of a redundant manipulator tracks a given path in the Cartesian space considering obstacles and singularity avoidance. Simulation results for a rectangular path in the Cartesian space with obstacles inside the work space are shown in figures 3 to 7. In this case, $N_P = 5$, $N_C = 1$, $Q = 10$ and R = 0.8. Figures 8 to 9 show the case, where the obstacle is located on the path. In this case, $N_P = 5$, $N_C = 1$. However, the best results are obtained when $Q = 10$ and $R = 1.3$. That is, when the coordinates of obstacles are changed, the weights in the cost function must be customized accordingly.

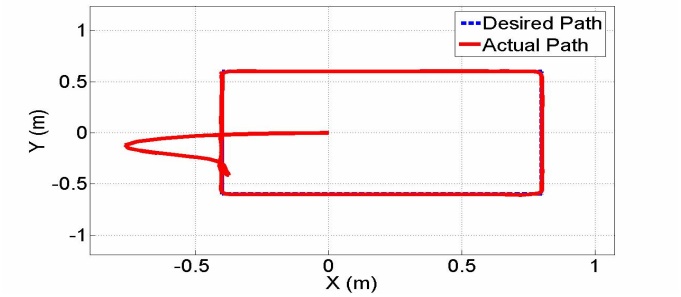

Fig. 3 Desired and actual end-effector path

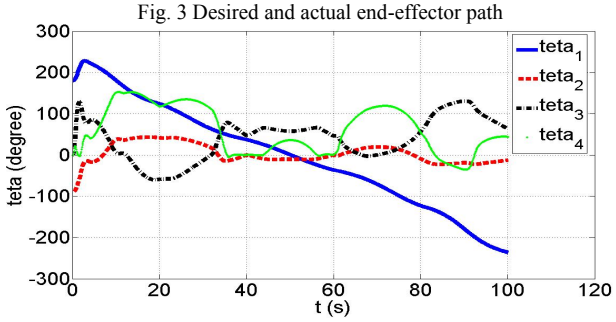

Fig. 4 Positions of manipulator joints

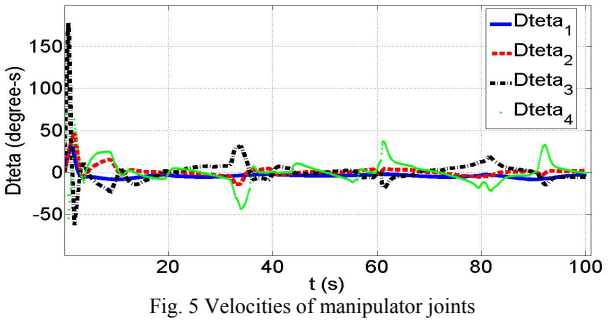

Fig. 5 Velocities of manipulator joints

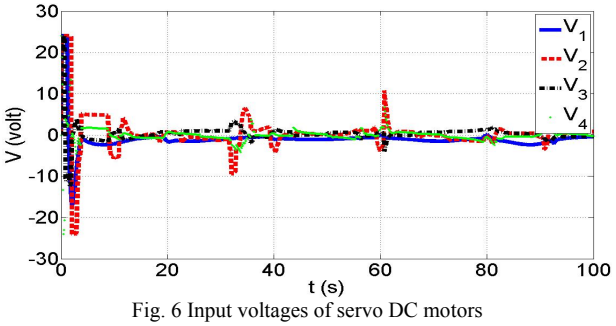

Fig. 6 Input voltages of servo DC motors

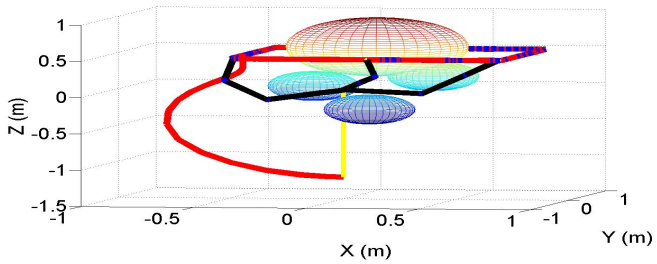

Fig. 7 Path following of a 4DOF manipulator with obstacles in the workspace

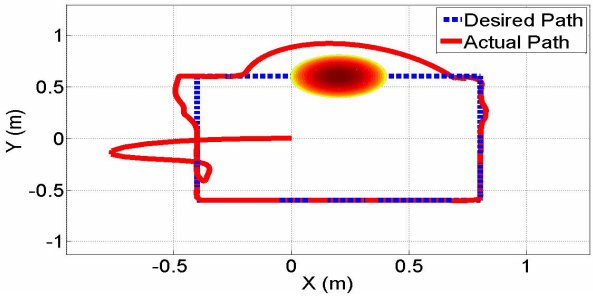

Fig. 8 Desired and actual end-effector path

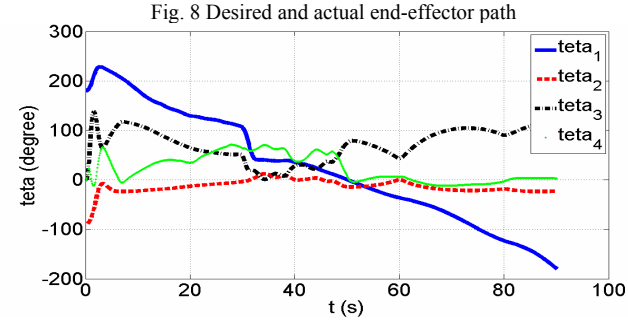

Fig. 9 Positions of manipulator joints

V. Path Tracking and Obstacle Avoidance Using Fuzzy NMPC

In the previous section, it was observed that for different paths and different positions of obstacles, the weights Q and R must be changed and finetuned in order to produce satisfactory results, thereby following the desired path as closely as possible and avoiding the obstacles at the same time. To provide a proper solution to this problem, fuzzy logic is employed in this paper for the on-line tuning of these weights. The proposed fuzzy system uses minimum distance between the manipulator and the obstacle and the rate of change of this distance as the inputs. The outputs of the fuzzy system are the weights Q and R. To design the fuzzy system, a boundary around each obstacle is considered in such a way that the control algorithm does not care about obstacles unless the end-effector or any links of the manipulator enter this boundary region. Parameters of fuzzy systems are tuned in such a way that when the manipulator is outside the obstacle regions, R is equal to zero and when the manipulator is inside this region, R is increased and Q is decreased adaptively. Fuzzy rules, membership functions, and fuzzy operations are shown in figures 10 to 13 and Tables IV and V.

Using the proposed fuzzy system, when the distance between the manipulator and the obstacle is more than 0.2 m, $R = 0$ and $Q = 10$ and for distances less than 0.2 m, $5 \leq Q < 10$ and $0 < R \leq 5$. In each step in prediction over prediction

horizon, Q and R are changed by fuzzy system and consequently each terms of cost function are calculated with different Q and R.

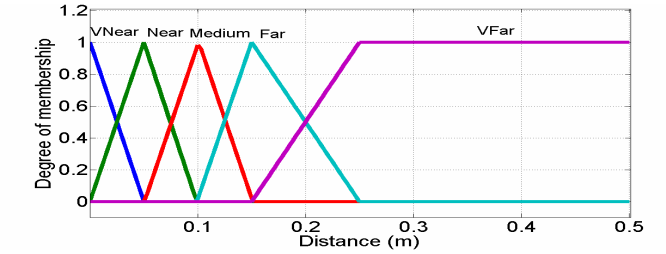

Fig. 10 Membership functions of distance

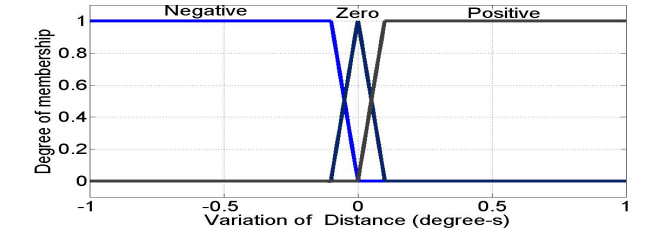

Fig. 11 Membership functions of distance variation

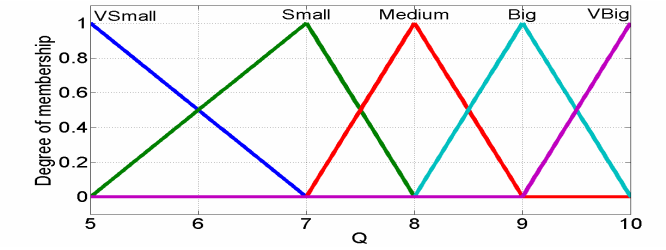

Fig. 12 Membership functions of weight Q

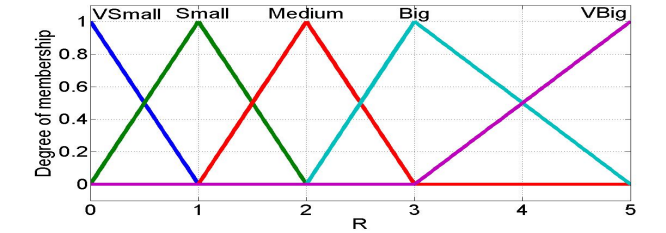

Fig. 13 Membership functions of weight R

TABLE IV
FUZZY OPERATIONS

And	Implication	Aggregation	Defuzzification
min	prod	max	lom

TABLE V
FUZZY RULES

D_O \ $\dot{D}_O$	Positive	Zero	Negative
Very Far	Q=VBig R=VSmall	Q=VBig R=VSmall	Q=VBig R=VSmall
Far	Q=VBig R=VSmall	Q=VBig R=VSmall	Q=Big R=Small
Medium	Q=Big R=VSmall	Q=Big R=Small	Q=Medium R=Medium
Near	Q=Big R=Small	Q=Medium R=Big	Q=Small R=Big
Very Near	Q=Medium R=Medium	Q=VSmall R=VBig	Q=VSmall R=VBig

Simulation results of the proposed fuzzy NMPC are shown in figures 14 to 17. As these figures show, the manipulator can follow the desired path with better accuracy as compared to the previous case. Moreover, Fig. 17 shows that the fuzzy system effectively changes the weighting parameters in the optimization process for better path following and obstacle avoidance.

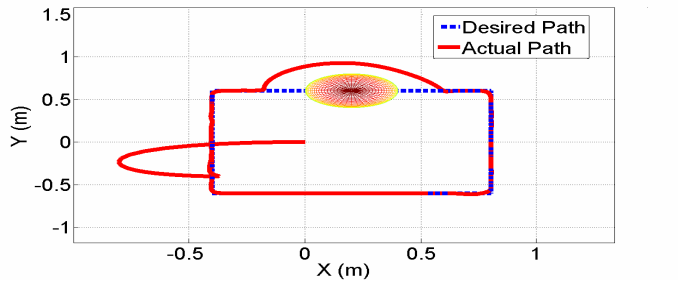

Fig. 14 Desired and actual end-effector path

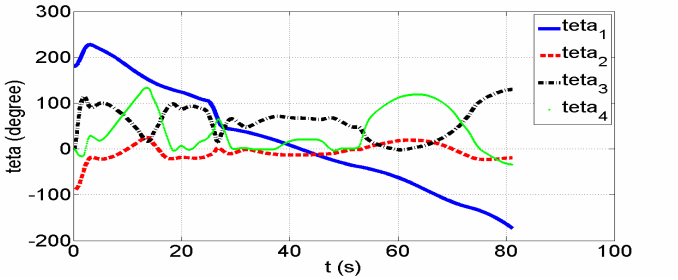

Fig. 15 Positions of manipulator joints

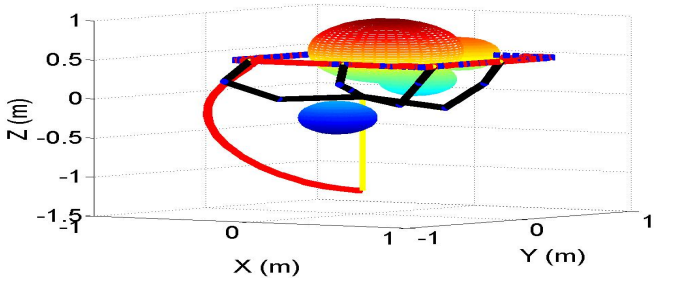

Fig. 16 Path following of a 4DOF manipulator with obstacles in the workspace

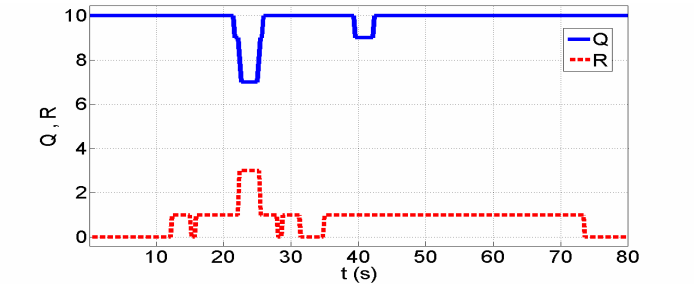

Fig. 17 Tuning of weights Q and R in cost function

VI. CONCLUSION

To achieve better path tracking and obstacle avoidance for robotic arms, the NMPC method was proposed in this paper. 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 constraints to the joints velocities, singularities were avoided. Furthermore, on-line

tuning of the weighting factors in NMPC was achieved using fuzzy logic. The proposed fuzzy system automatically adjusts the path tracking and obstacle avoidance weights in the cost function for obtaining better performance. Using the tuning mechanism, obstacles do not affect performance of the manipulator unless they enter the predefined boundary regions around obstacles. Future works in this area include considering moving obstacles, robustness of the method against changes in the system parameters and stability analysis of closed loop system.

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

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

[17] 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.

[18] 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.

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

APPENDIX

$$
\begin{aligned}
&G(1,1) = 0\\
&G(1,2) = \left(\frac{1}{2}\mathrm{m}_2 g l_2 + \mathrm{m}_3 g l_2 + \mathrm{m}_4 g l_2\right)\mathrm{c}_2 + \left(\frac{1}{2}\mathrm{m}_3 g l_3 + \mathrm{m}_4 g l_3\right)\mathrm{c}_{23} + \frac{1}{2}\mathrm{m}_4 g l_4 \mathrm{c}_{234}\\
&G(3,1) = \left(\frac{1}{2}\mathrm{m}_3 g l_3 + \mathrm{m}_4 g l_3\right)\mathrm{c}_{23} + \frac{1}{2}\mathrm{m}_4 g l_4 \mathrm{c}_{234}\\
&G(4,1) = \frac{1}{2}\mathrm{m}_4 g l_4\ \mathrm{c}_{234}
\end{aligned}
\tag{A.1}
$$

$$
\begin{aligned}
M(1,1) &= \frac{2}{3}m_1 l_1^2 + \left(\frac{1}{3}m_2 l_2^2 + m_3 l_2^2 + m_4 l_2^2\right)\mathrm{c}_2^2 + \left(\frac{1}{3}m_3 l_3^2 + m_4 l_3^2\right)\mathrm{c}_{23}^2 + \frac{1}{3}m_4 l_4^2 c_{234}^2\\
&\quad + \left(m_3 l_3 l_2 + 2m_4 l_3 l_2\right)\mathrm{c}_{23}\mathrm{c}_2 + m_4 l_4 l_3 c_{234} c_{23} + m_4 l_4 l_2 c_{234}\mathrm{c}_2\\
M(2,2) &= \frac{1}{3}m_2 l_2^2 + m_3 l_2^2 + \frac{1}{3}m_3 l_3^2 + m_4 l_3^2 + m_4 l_2^2 + \frac{1}{3}m_4 l_4^2 + m_4 l_4 l_2 + m_4 l_3 l_4 \mathrm{c}_4\\
&\quad + \left(m_3 l_3 l_2 + 2m_4 l_2 l_3\right)\mathrm{c}_3\\
M(3,3) &= \frac{1}{3}m_3 l_3^2 + m_4 l_3^2 + \frac{1}{3}m_4 l_4^2 + m_4 l_3 l_4 \mathrm{c}_4\\
M(4,4) &= \frac{1}{3}m_4 l_4^2\\
M(2,3) &= M(3,2) = \frac{1}{3}m_3 l_3^2 + m_4 l_3^2 + \frac{1}{2}m_4 l_4 l_2 + \frac{1}{3}m_4 l_4^2 + \left(\frac{1}{2}m_3 l_3 l_2 + m_4 l_2 l_3\right)\mathrm{c}_3\\
&\quad + m_4 l_4 l_3\ \mathrm{c}_4\\
M(2,4) &= M(4,2) = \frac{1}{2}m_4 l_4 l_2 + \frac{1}{3}m_4 l_4^2 + \frac{1}{2}m_4 l_4 l_3 \mathrm{c}_4\\
M(3,4) &= M(4,3) = \frac{1}{3}m_4 l_4^2 + \frac{1}{2}m_4 l_3 l_4 \mathrm{c}_4\\
M(1,2) &= M(2,1) = 0\\
M(1,3) &= M(3,1) = 0\\
M(1,4) &= M(4,1) = 0
\end{aligned}
\tag{A.2}
$$

$$
\begin{aligned}
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\\
&\quad + \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\\
&\quad + \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\\
&\quad + \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\\
&\quad + \left(-m_3 l_3 l_2 \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}\mathrm{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}\mathrm{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}\mathrm{c}_2 + \frac{1}{6}m_4 l_4^2 \mathrm{s}_{223344} \end{pmatrix}\dot{\theta}_1^2\\
&\quad + \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}_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_4 l_4 \mathrm{s}_4\right)\dot{\theta}_4^2\\
&\quad + \left(-m_4 l_3 l_4 \mathrm{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\\
&\quad + \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
\end{aligned}
\tag{A.3}
$$

where, l_i and m_i ($i=1,\ldots,4$) are the length and mass of the i^{th} link, respectively, θ_i and $\dot{\theta}_i$ are the angular position and the angular velocity of the i^{th} 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.