www.icgst.com

Control of redundant robot and singularities avoidance based anfis network

*M. BENZAOUI, ** H. CHEKIREB and ** M. TADJINE
*Dpt de l'électrotechnique et de l'automatique
Institue de génie électrique et électronique Université M'hamed Bougara,
Boumerdès, avenue de l'Indépendance35000 Boumerdès
Algeria
**Laboratoire de Commande des Processus. Dpt de Génie Electrique, ENP
BP 182, 10 avenue Hassan Badi, 16200 El-harrach Alger,
Algeria
Email: benzaouimess@mail.com, chekireb@yahoo.fr

Abstract

In this work we exploit an anfis network to achieve the singularities avoidance of a redundant robot. This latter must carry out a trajectory tracking in the Cartesian space near a singularity point. The singularity avoidance without affecting trajectory tracking is involved via self-motion method. The analytical determination of this self motion is obtained on the optimization of scalar function depending on the robot manipulability measure. In view to reduce the on line cumbersome computations due to the analytical method, a learning network based anfis is used to generate this self-motion. The learning process uses the input-output data coming from the analytical self-motion. The two methods of avoiding singularity (based on analytical method and on anfis one) are tested in the case of 3 dof planar robot performing, in Cartesian space, a trajectory near a singular point. The obtained results show that the proposed criteria ensure a good control when the robot operates near a singularity point.

Keywords: *redundant robot, singularity avoidance, self-motion. adaptive neuro fuzzy inference system (anfis).*

1. Introduction

Singularities are considered as geometric constraints of the robot. Beside, on these points, the robot control is lost and the driven torques become huge, which can conduct to a technical failure. So, avoidance is often done by modifying the trajectory so that it moves away from these points; this inevitably led to a workspace reduction.

Kinematic redundancy can be used to overcome these difficulties by maximizing the available workspace. This must be done in parallel with the desired task in order to keep the performance required by the execution of the main task [1]-[2].

Robots are cinematically redundant if the *n* dof is greater than the *m* degrees necessary to fix the position and orientation of the effector in the operational space. The *r* additional DOF are exploited at the step of the inverse kinematics resolution to create an internal motion of joints (i.e. self-motion).

Singularities avoidance problem is tackled as an optimization of a criteria (a scalar function h(.)). Singularities avoidance constitutes the secondary task, which must be carried out in addition to the main task resulting from the tracking trajectory in Cartesian space.

Our goal is to reduce the computation time inherent to the analytical calculations of singularity avoidance by using adaptive neuro-fuzzy networks.

Thus, first, a redundant robot is defined from the PUMA 560 robot as a planar 3 dof robot [3] [4]. This robot must perform a trajectory tracking in Cartesian space while avoiding a singularity point located on its trajectory. To this end, self-motion method is used. The analytical method is obtained on the optimization of scalar function depending on the robot manipulability measure [4] [6]. In view to reduce the on line cumbersome computations due to the analytical method, a learning network based anfis is used to generate the self-motion [9]-[11]. Finally, the control of redundant robot when it tracks a trajectory near singular point is tested where the self-motion is performed using analytical and anfis methods.

2. Position of the problem

A three dof (n=3) planar robot is used in this work which derives from PUMA 560 where the θ_1, θ_4 and θ_6 joints are locked and the θ_2, θ_3 and θ_5 joints are free to move. As the desired task is the the end-effector positioning, on the vertical plan, in arbitrary orientation so, this requires only two dof (*m*=2). Therefore, the degree of redundancy *r* for this robot is one: *r=n-m=1*. The state vector is defined as following:

$$q = [q_1 \quad q_2 \quad q_3]^T = [\theta_2 \quad \theta_3 \quad \theta_5]^T$$

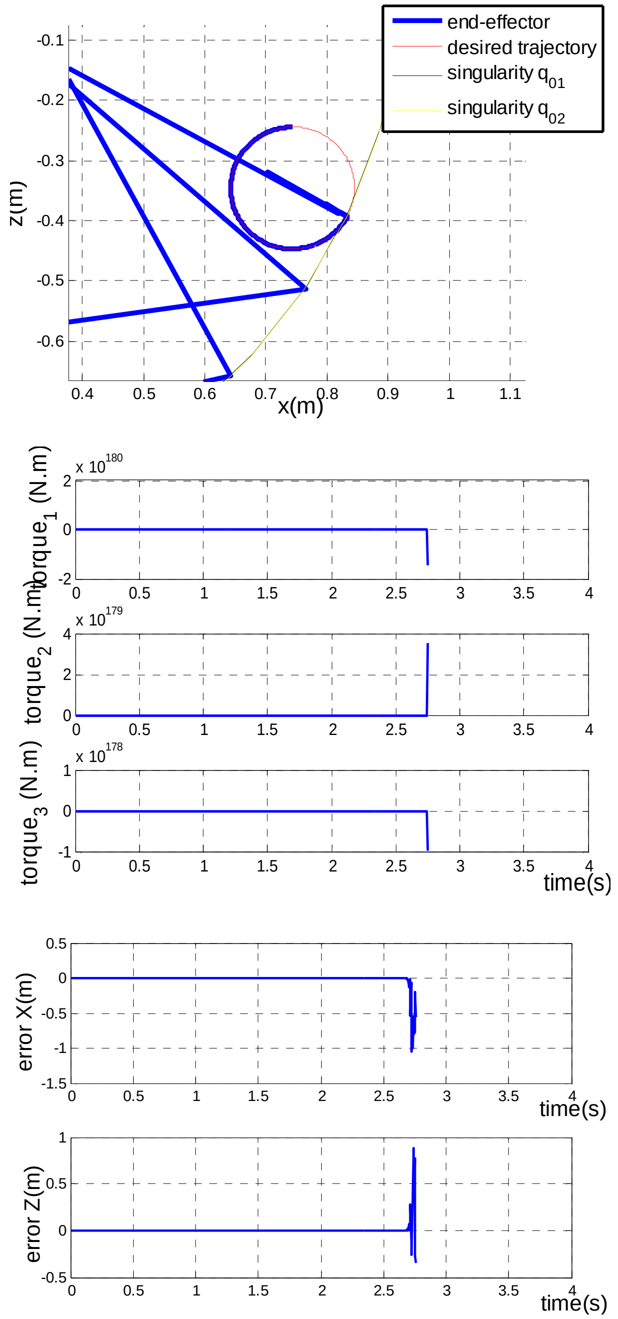

Figure 1. Robot Control without singularity avoidance

The direct geometrical model (DGM), gives the position of the end-effector in this vertical plane according to the joint position (q).

$$x_1 = a_2.\cos(q_1) + a_3.\cos(q_1+q_2) + d_4.\sin(q_1+q_2) + d_6.\sin(q_1+q_2+q_3) \quad (2)$$

$$x_2 = -a_2.\sin(q_1) - a_3.\sin(q_1+q_2) + d_4.\cos(q_1+q_2) + d_6.\cos(q_1+q_2+q_3) \quad (3)$$

Where $x=[x_1,\ x_2]^T$ is the vector of co-ordinates in Cartesian space and (a_2, a_3, d_4, L) are geometric parameters of PUMA 560.

Direct kinematic model (DKM) gives joint velocity according to Cartesian velocity, such as:

$$\dot{x} = J.\dot{q} \quad (4)$$

Where J is the Jocobian of the robot

Since in our case the Jacobian J is not square matrix so, its inverse is calculated by the pseudo-inverse method and it is noted by J^+. Using the pseudo inverse, the joint velocity is [1][4]:

$$\dot{q}_p = J^+\dot{x} \quad (5)$$

and

$$J^+ = J^T.(J.J^T)^{-1} \quad (6)$$

Sometimes, the robot cannot reach to some desired points in its workspace. This is due to of unusual kinematics. The joint solutions q_0 to these points make Jacobian singular and therefore not reversible.

The actuator torque vector τ is determinate using computed torque method and it is related to the robot dynamics by [5]:

$$\tau = A.J^+(w - \dot{J}.\dot{q}) + Q(q,\dot{q}) \quad (6a)$$

Where:

$$w(t) = \ddot{x}_d + K_v.(\dot{x}_d - \dot{x}) + K_p.(x_d - x) \quad (6b)$$

$$Q(q,\dot{q}) = B(q,\dot{q}) + C(q,\dot{q}) + G(q) \quad (6c)$$

With:

A : inertia matrix of the robot.
B: Vector of Coriolis forces.
C: Vectors of centrifugal forces.
G: Vector of gravitational forces.
x_d: desired trajectory in operational space

The desired trajectory in operational space (a circle in our case) is imposed such as it passes close to the image in Cartesian space of a singular point q_0. The application of control law (6) conducts to the robot behavior shown in Figure 1. This last is obtained when the end-effector tracks the circular path near an image in Cartesian space of singular point q_0. It appears that the trajectory tracking is achieved well far of the singularity meanwhile near it, the torques and tracking errors values explode which may conduct to the failure of actuators. To solve this problem, it is possible to exploit the redundancy which is the subject of the next section

3. Singularity avoidance based on self motion

In the case of redundant robots, the self-motion can be exploited in order to achieve singularities avoidance. The problem can be tackled by having recourse to a scalar function $h(q)$ [1-4]. Beside, the desired trajectory $x_d(t)$ for a redundant robot is represented by a vector of dimension m related to the main task constraint. The r DOF, in more can be exploited to perform a secondary task related to the optimization of an objective function $h(.)$ [1][3].

In order to achieve the secondary task, we add, to $\dot{q}_p$ given by (7), the self-motion term $\dot{q}_n$. This last is taken as being projection into the kernel space of J.

$$\dot{q}_n = \alpha(I - J^+J).z \quad (7)$$

The self-motion term $\dot{q}_n$ confers on the robot arms an ability to be reconfigured without affecting the main task. This leads to modify the solution of equation (4) as follows [1][4] :

$$\dot{q} = \dot{q}_p + \dot{q}_n \tag{8}$$

z in (7) is the matrix (vector for the case $r=1$) of an arbitrary projection on the kernel of J. It can correspond to the projection related to the gradient of $h(q)$:

$$z = \nabla h \tag{9}$$

To satisfy the singularity avoidance, the function $h(q)$ can be selected as being the manipulability measure proposed firstly Yoshikawa[6]:

$$h(q) = \sqrt{\det(J(q).J^T(q))} \tag{10}$$

So, for the robot used in this work, if we set:

$$\det(J^T J) = Me_{11}Me_{22} - Me_{12}^2 \tag{11}$$

With:

$$\left.\begin{aligned}
Me_{11} &= 3d_6^2.\cos^2(q_1+q_2+q_3) + 2a_3^2\sin^2(q_1+q_2) + a_2^2.\sin^2(q_1) + \\
&\quad 2.d_4^2\cos^2(q_1+q_2) + 4.d_6d_4.\cos(q_1+q_2).\cos(q_1+q_2+q_3) - \\
&\quad 4d_6a_3\sin(q_1+q_2)\cos(q_1+q_2+q_3) - 2.a_2.d_6\sin(q_1)\cos(q_1+q_2+q_3) - \\
&\quad 4.d_4.a_3.\cos(q_1+q_2).\sin(q_1+q_2) - 2d_4a_2.\cos(q_1+q_2)\sin(q_1) + \\
&\quad 2.a_2a_3.\sin(q_1).\sin(q_1+q_2) \\
Me_{22} &= 3.d_6^2.\sin^2(q_1+q_2+q_3) + 2.a_3^2\cos^2(q_1+q_2) + 2.d_4^2\sin^2(q_1+q_2) + \\
&\quad 4d_6d_4.\sin(q_1+q_2).\sin(q_1+q_2+q_3) + 4d_6.a_3\cos(q_1+q_2)\sin(q_1+q_2+q_3) + \\
&\quad 2.a_2d_6\cos(q_1)\sin(q_1+q_2+q_3) + 4.d_4.a_3.\sin(q_1+q_2).\cos(q_1+q_2) + \\
&\quad 2d_4a_2.\sin(q_1+q_2)\cos(q_1) + 2a_2a_3.\cos(q_1)\cos(q_1+q_2) + a_2^2\cos^2(q_1) \\
Me_{12} &= -3.d_6^2\cos(q_1+q_2+q_3)\sin(q_1+q_2+q_3) - 2.d_6d_4.\sin(2(q_1+q_2)+q_3) - \\
&\quad 2.d_6a_3\cos(2(q_1+q_2)+q_3) - a_2.d_6\cos(2q_1+q_2+q_3) - .d_4^2.\sin(2(q_1+q_2)) - \\
&\quad 2.a_3.d_4\cos(2(q_1+q_2)) - 2.a_2d_4.\cos(2q_1+q_2) + a_3^2.\sin(2(q_1+q_2)) + \\
&\quad a_2a_3\sin(2q_1+q_2) + a_2^2\cos(q_1).\sin(q_1)
\end{aligned}\right\rangle \tag{12}$$

In order to apply this method for 3 dof planar robot, we firstly isolate a singularity point q_0. This can be carried out by using 2×2 minors Mn_i ($i=1,2,3$) in the condition that the searched q_0 makes all $Mn_i = 0$ [6]. In our case, we have isolated a singularity point which makes $Mn_1=0$, $Mn_2=0$, and $Mn_3 \approx 0$.

The calculation of ∇h involved in generalised inverse term is such as:

$$\nabla h = [\delta_1 h \quad \delta_2 h \quad \delta_3 h]^T \tag{13}$$

where

$$\delta_i h = \frac{1}{2}\frac{\partial_i(Me_{11}.Me_{22}) + \partial_i(Me_{22}.Me_{11}) - 2.\partial_i(Me_{12}.Me_{12})}{(\det(Me))^{\frac{1}{2}}} \tag{14}$$

and $\partial_i(.)$ stands for $\partial(.)/\partial q_i$ with i=(1,2,3)

Thereafter, the relations (10) to (14) are computed in order to determine (9) and to obtain the robot control incorporating self-motion related to the used criteria.

4. Control robot and singularity avoidance based anfis

4.1 Anfis network of the self-motion

In this section, our goal is to design a black box generating the self-motion which ensures singularity avoidance. Firstly, we simulate the robot accomplishing the path trajectory near the singular point q_0 where we use the control law (6) where the self-motion is performed using previous analytical method with relations (7) to (14). The Cartesian positions, velocities and Jacobian of robot arm are well defined. Moreover, data related to scalar function $h(q)$ and the self-motion arms $\dot{q}_n$ obtained from this analytical method are stored. Thereafter, these data will be used as an off-line training of an anfis network which must generate at its output the velocity vector $\dot{q}_n$ when the manipulability measure $h(q)$ is applied to its input.

Learning can be made, on the basis of measurements given by N_j input-output pairs of data with the k^{th} pair is denoted by:

$$z(k) \text{ and } y(k)=[y_1(k)\ldots\ y_n(k)]^T \text{ with } k=(1,N_j) \tag{16}$$

For the k^{th} measurement, the elements $z(k)$ and $y_j(k)$ with (j = 1, n) are respectively the manipulability measure $h(q(k)$ and the self-motion velocity $\dot{q}_{nj}(k)$ related to the j^{th} arm.

For the k^{th} measurement, the output $\hat{y}_j(k)$ is estimated by a anfis network based on TSK model (net_j) where its i^{th} rule is given by [10-11]:

$$\text{If } \ z \text{ is } \overline{A}_j^i \text{ then } \ \hat{y}_j^i = a_{j0}^i + a_{j1}^i z\,; \tag{17}$$

Where z is input fuzzy variable and $\hat{y}_j^i$ is the j^{th} output value for the i^{th} rule. Input variable z for net_j is described by mj fuzzy sets ($A_j^1 \ldots A_j^{mj}$) which are respectively associated to membership functions ($\mu_{A_j^1} \ldots \mu_{A_j^{mj}}$) and $\overline{A}_j^i$ stands for one of fuzzy set among the mj fuzzy sets. Moreover, the coefficients (a_{j0}^i, a_{j1}^i) are the j^{th} consequence parameters. The overall output $\hat{y}_j$ is obtained by combining results over all N_t rules as:

$$\hat{y}_j = \frac{\sum_{i=1}^{Nt}\mu^i\hat{y}_j^i}{\sum_{i=1}^{Nr}\mu^i} \tag{18}$$

Where, the firing degree μ^i of the i^{th} rule is determined using the t-norm operator $t_{j=1}^n(.)$:

$$\mu^i = t_{j=1}^n(\overline{A}_j^i(z)) \tag{19}$$

Since the analytical computation of the self-motion is of MIMO type: the three variables $\delta_1 h$, $\delta_2 h$ and $\delta_3 h$ are simultaneously used to determine the joint velocities $\dot{q}_{n1}$, $\dot{q}_{n2}$ and $\dot{q}_{n2}$. However, in the context of this work, we intend to build networks generating the fuzzy self-motion in the form SISO also, there will be a network Net_j for each self-motion $\dot{q}_{nj}$ with j=(1,2,3).

The Table1 gives the conditions and the performances of the learning process. So, for net_j, we have the number N_j of used input-output pairs of data, the membership function shape, the number of fuzzy set m_j, the epochs number N_{ep} necessary to perform the quadratic errors e_Q on the exemplar.

Net	N_j	sheap	m_j	N_{ep}	e_Q
Net_1	400	Π-sig	7	400	1.29
Net_2	400	bell	3	700	2.21
Net_3	400	bell	9	400	3.78

Table1: Conditions and performances of learning process

4.2 Simulation results

The robot must follow the trajectory (a circle) near a singularity point q_0. In this aim, the control law is yet involved by equation (6) but, for this case, the generation of the self-motion, achieving the singularity avoidance, is carried out by the previously trained networks (Net_1, Net_2 and Net_3). Results of the robot's behaviour in the plan, shown in Figure 2 and Figure 3, reveal that the trajectory tracking is achieved with around the same tracking errors while maintaining singularity avoidance. Moreover, the driving torques remains in allowable intervals. The Table 2 gives results for average values of robot control when the self-motion is carried out with analytical method or with learning network. The results examination reveals that the performances of the control robot with self-motion based anfis are sensibility similar to the control with self-motion based on analytical method.

Method	Torques			Tracking errors	
	τ_1	τ_2	τ_3	e_x	e_z
analy	43.9138	5.8775	0.4900	0.5661	0.5172
anfis	43.0217	6.5886	0.4596	0.0866	0.1174

Tab.2 Results of robot control with avoidance singularity based on analytical or anfis method

5. Conclusion

Redundancy can be applied to singularity avoidance problem. The r DOF in more are exploited to create a self-motion that acts towards the optimization of a scalar function can be selected by exploiting the characteristics of jacobian. A control incorporating this procedure is developed for trajectory of a planar 3 DOF robot resulting from a PUMA560. That operates in an area containing a singularity point. Results showed that beside this singular point, measuring the manipulability criteria leads to keep performances.

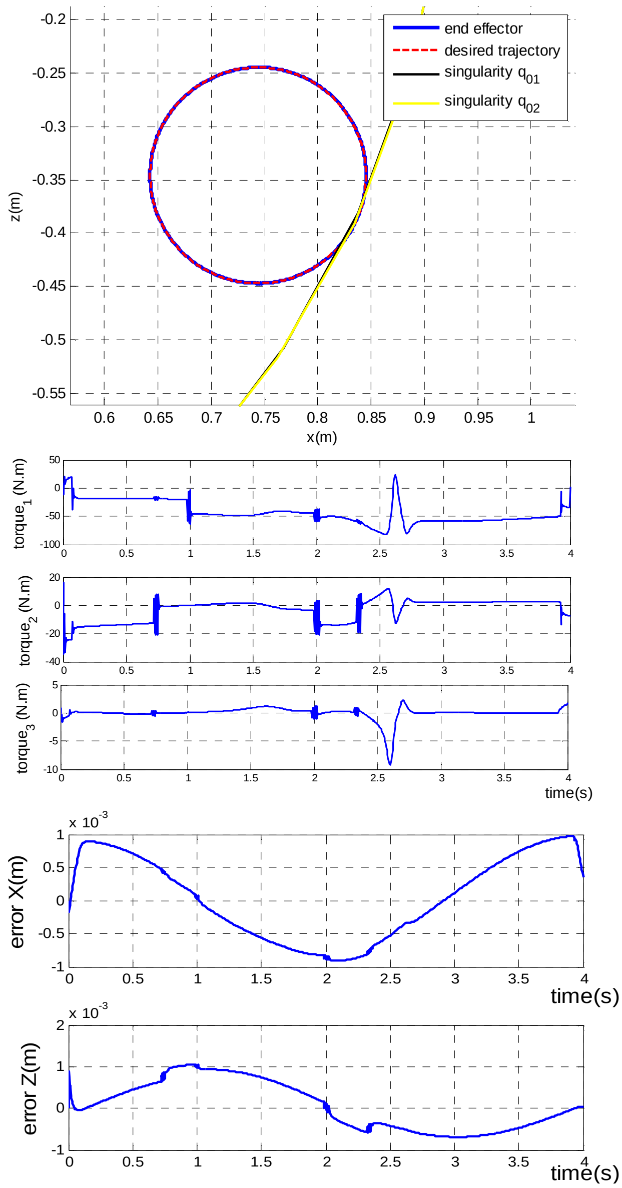

Figure 2. Robot Control with singularity avoidance Analytical case

Learning Network FIS is achieved using the criterion of manipulability measure. The network proposed can also copy the shape. One can say it allows having better performances by simplifying calculations.

6. References

[1] [1] Kevin A.O'Neil,Yu-chechen, and Jiaping Seng. On the Existence and characteristic of solution path of Algorithmic Singularities. International Journal of Robotics Research. pages 171 – 184 Vol.16. April 1997.

[2] [2] John Baillieul, A Constraint Oriented Approach to Inverse Problems For Kinematically Redundant Manipulators. pages 1827– 1833. In proceedings of the International Conference Robotics and Automation.. IEEE , Vol. 4. Mar 1987.

[3] [3] B. Armstrong, O. Khatib and J. Burdick. The explicit dynamic model and inertial parameters of the PUMA 560 arm, pages 510 – 518. IEEE International Conference on Robotics and Automation. 1986.

[4] [4] M. Benzaoui, H. Chekireb. Tracking of Cartesian Trajectory near Singularities for Redundant Robot, JASE Volume 3, Issue 4. December 2009.

[5] [5] L. Sciavicco and B. Siciliano. A computational technique for solving robot end-effector trajectories into joint trajectories. pages 535 – 536. American Control Conf., Atlanta, GA. 1988.

[6] [6] Krzysztof Tchon and Aleksander Matuszok. On avoiding Singularities in Redundant Robot Kinematics. pages 599 – 606. Robotica, Vol. 13. 1995

[7] [7] Denny Oetomo, MarceloH.Ang. Singularity robust algorithm in serial manipulators. pages 122– 134 Robotics and Computer-Integrated Manufacturing 25 2009.

[8] [8] Henzeh Leeghima, HyochoongBang, Jong-OhPark. Singularity avoidance of control moment gyros by one-step ahead singularity index. pages 935 – 945, Acta Astronautica . 2009.

[9] [9] W. Utama. An artificial neural network for redundant robot kinematics computation, Master of Science in Electrical Engineering. New jersey institute of Technology. 1990.

[10] [10] H. Chekireb, M. tadjine, D. Bouchaffra. Direct Adaptative Fuzzy Control of Nonlinear System Class With Application. pages 113 – 121. Control and Intelligent System, Vol. 31, N° 2. 2003.

[11] [11] Mahdi Aliyari Shoorehdeli, Mohammad Teshnehlab, AliKhakiSedigh. Training ANFIS as an identifier with intelligent hybrid stable learning algorithm based on particle swarm optimization and extended Kalman filter . pages 922 – 948. Fuzzy Sets and Systems 160. 2009.

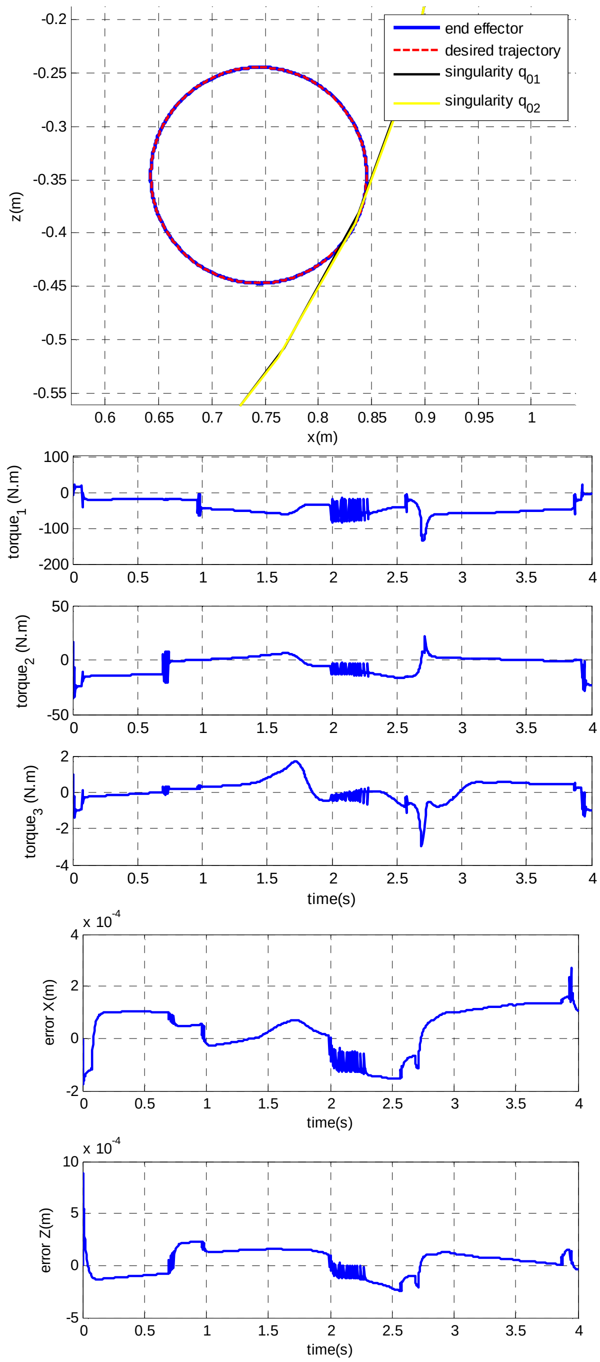

Figure 3. Robot Control with singularity avoidance ANFIS case