TRAJECTORY TRACKING CONTROL OF SCORBOT-ER V PLUS ROBOT MANIPULATOR BASED ON KINEMATICAL APPROACH

HIMANSHU CHAUDHARY

Department of Electrical Engineering, IIT Roorkee, India
himan.74@gmail.com

DR. RAJENDRA PRASAD

Professor, Department of Electrical Engineering, IIT Roorkee, India
rpdeefee@iitr.ernet.in

DR. N. SUKAVANUM

Associate Professor, Department of Mathematics, IIT Roorkee, India
nsukvfma@iitr.ernet.in

Abstract:
In this paper, two approaches to generating such trajectories: straight lines in joint space and straight lines in Cartesian space have been discussed. This is one of the most common requirements in robotics for moving the end-effector smoothly from initial location to goal location. These are known respectively as joint space and Cartesian space tracking. Two user defined algorithms are developed for Joint space as well as Cartesian space trajectory tracking. The algorithm has been tested in simulation yielding fair results, which have also been compared with those provided by another important trajectory planning technique methods.

Keywords: DOF; Trajectory Planning; Smoothness; Joint Path Generation; Cartesian Path Generation.

1. Introduction

A path-tracking algorithm to compensate for path deviation due to torque limits was proposed by (Eoma, Suha, & Chungb, 2000) which in turns used a disturbance observer to obtain a simple equivalent robot dynamic (SERD) model to modify the desired acceleration of the nominal trajectory in Cartesian space. A technique based on continuous genetic algorithms (CGAs) to solve the path generation problem for robot manipulators was presented in (Abo-Hammoura, Mirza, Mirza, & Arif, 2002). The inverse kinematics problem was formulated as an optimization problem based on the concept of the minimization of the accumulative path deviation and is then solved by the presented technique. A unified approach to optimal pose trajectory planning for robot manipulators in Cartesian space through a genetic algorithm (GA) enhanced optimization of the pose ruled surface was presented in (Zha, 2002). The optimization model is established based on functional analysis and dynamics planning, and instantiated by using high-order parametric space curves as position and orientation trajectories. (Chettibi, Lehtihet, Haddada, & b, 2004) discussed the problem of minimum cost trajectory planning for robotic manipulators, in which the generic optimal control problem was transformed into a non-linear constrained optimization problem which is treated then by the Sequential Quadratic Programming (SQP) method. Simulated Annealing (SA) technique was applied to the problem of robot path planning in [22]. Three situations were considered: the path is represented as a polyline; as a BAzier curve; and, as a spline interpolated curve. In order to ensure that the resulting trajectory is smooth enough, an objective function containing a term proportional to the integral of the squared jerk (defined as the derivative of the acceleration) along the trajectory was considered in (Gasparetto & Zanotto, A new method for smooth trajectory planning of robot manipulators, 2007). Then another term proportional to the total execution time was added to the expression of the objective function. Fifth-order B-splines were then used to compose the overall trajectory. The path planning problem with general end-effector constraints for robot manipulators was addressed in (Yao & Gupta, 2007). Two approaches were proposed. The first approach was the Adapted Randomized Gradient Descent (ARGD) method, and the second approach was radically different as it worked in both task space and Cartesian space in comparison with first which searches purely in Cartesian space. A planning mode of trajectory motion for serial-link manipulators with higher-degree polynomials application was discussed in (Boryga & Grabos, 2009). The linear acceleration profiles of end-effector, for each coordinate, were planned as the polynomials of degrees 9, 7 and 5. This approach to polynomial form structure necessitates the determination of only one polynomial coefficient, irrespective of its order. Method based on trajectory planning to avoid the detachment of joint elements of a manipulator with clearances was offered in (Bu, Liu, Tan, & Gao, 2010). An improved detachment criterion for the revolute and spherical joints was proposed. An analysis based on cubic splines or fifth-order B-splines trajectory of an algorithm for optimal trajectory planning of robot manipulators was presented in (Gasparetto & Zanotto, Optimal trajectory planning for industrial robots, 2010). The objective function to be minimized is a weighted sum of the integral squared jerk and the execution time. The proposed technique allowed setting constraints on the robot motion, expressed as upper bounds on the absolute values of velocity, acceleration and jerk. A robust and fast procedure that could be used to identify the joint stiffness values of any six-revolute serial robot was presented in (Dumas, Caro, Garnier, & Furet, 2011). The proposed method aimed to evaluate joint stiffness values by considering both translational and rotational displacements of the robot end-effector for a given applied wrench (force and torque).

Kinematical analysis based solutions are very vital when one wants to perform modelling of robotic arm. It turns out to be a difficult task to find the solution through inverse kinematics with increase in DOF (Degree of Freedom) of robot. The conventional methods used for calculating inverse kinematics of any robot manipulator are: geometric (R, 1983) (S., 1982), algebraic (J., 1980) (Manocha & Canny, 1994) (Paul, Shimano, & Mayer, 1982) and iterative (Korein & Balder, 1982) approaches. While algebraic methods cannot promise closed form solutions. Geometric methods must be able to produce closed form solutions for the first three joints of the manipulator geometrically. The iterative methods on the other hand approach only to a single solution and that solution also depends on the starting point. To solve the inverse kinematics problem for three different cases of a 3-degrees-of freedom (DOF) manipulator in three dimensional spaces, a solution was proposed in (Bu, Liu, Tan, & Gao, 2010) using feed-forward neural networks. This introduces the fault-tolerant and high-speed advantages of neural networks to the inverse kinematics problem. A three-layer partially recurrent neural network was proposed by (Arahjo & Jr., 1998) for trajectory planning and to solve the inverse kinematics as well as the inverse dynamics problems in a single processing stage for the PUMA 560 manipulator**.** Hierarchical control technique based on the establishment of a non-linear mapping between Cartesian and joint coordinates using fuzzy logic in order to direct each individual joint was proposed in (Howard & Zilouchian, 1998)**,** for controlling a robotic manipulator. Commercial Microbot with 3DOF was utilized to evaluate the proposed method. A novel modular neural network system to overcome the discontinuity of the inverse kinematics function was proposed in (Oyama, Chong, Agah, Maeda, & Tachi, 2001) that consist of a number of expert neural networks. Neural network based three-joint robotic manipulator simulation software was developed in (Koker, Oz, Cakar, & Ekiz, 2004) for inverse kinematics solution of a robotic manipulator. Then a designed neural network was used to solve the inverse kinematics problem**.** An Artificial Neural Network (ANN) based on Bees Algorithm using backpropagation algorithm was applied in (Bingul, Ertunc, & Oysu, 2005) to solve inverse kinematics problems of industrial robot manipulator. That in turns used to train multi-layer perceptron neural networks in (Pham, Castellani, & Fahmy, 2008) to model the inverse kinematics of an articulated robot manipulator arm. An Artificial Neural Network (ANN) based approach for fast inverse kinematics computation and effective geometrically bounded singularities prevention of redundant manipulators was presented in (Mayorga & Sanongboon, 2005). First some bounded geometrical concepts were used to establish some characterizing matrices, to get a simple performance index, and a null space vector for singularities avoidance/prevention and secure way generation. Then inverse kinematics based on above assumptions was computed using a properly trained ANN. A reliability neural network based inverse kinematics solution approach was presented in (Koker, Reliability-based approach to the inverse kinematics solution of robots using Elman’s networks, 2005), which in turns applied to a six-degree of freedom robot manipulator. The structure of the proposed method was based on Elman network using three networks designed parallel to minimize the error of the whole system. An adaptive learning strategy using an artificial neural network (ANN) to control the motion of a 6-DOF manipulator robot by overcoming the inverse kinematics problem was proposed in (Hasan, Hamouda, Ismail, & Al-Assadi, 2006) which mainly included singularities and uncertainties in arm configurations. The proposed control technique didn’t require any prior knowledge of the kinematics model of the system being controlled. An Adaptive Neuro-Fuzzy Inference System (ANFIS) method based on the Artificial Neural Network (ANN) was applied in (Chaudhary & Prasad, 2011) to design an Inverse Kinematic based controller for the inverse kinematical control of SCORBOT-ER V Plus. The proposed ANFIS controller combined the advantages of a fuzzy controller as well as the quick response and adaptability nature of an Artificial Neural Network (ANN). Computer Simulation was carried out to demonstrate the accuracy of the proposed controller to generate an appropriate joint angle for reaching desired Cartesian state, without any error.

In this paper, a simulation based solution for trajectory tracking of industrial robot manipulator is presented. It can efficiently solve practical and real size problems. Robot kinematical singularities and workspace constraints were considered in developing the model. The method was tested on a SCORBOT-ER V PLUS robot in the Control System and Robotics Laboratory at IIT Roorkee.

This paper is organized into four sections. In the next section, the Positional analysis (Forward as well as inverse kinematics) of SCORBOT-ER V Plus has been briefed with the help of DH algorithm as well as conventional techniques methods. The trajectory planning based on the positional analysis is introduced in section 3. It also explains about the two approaches used for trajectory tracking with the help of flow charts. Simulation results are discussed in section 4. Section 5 gives concluding remarks.

2. KINEMATICS OF SCORBOT-ER V PLUS (Chaudhary & Prasad, 2011)

SCORBOT-ER V Plus is a vertical articulated robot, with five revolute joints. It has a Stationary base, shoulder, elbow, tool pitch and tool roll. Figure 1.1 shows the schematic diagram of SCORBOT-ER V Plus Robot manipulator.

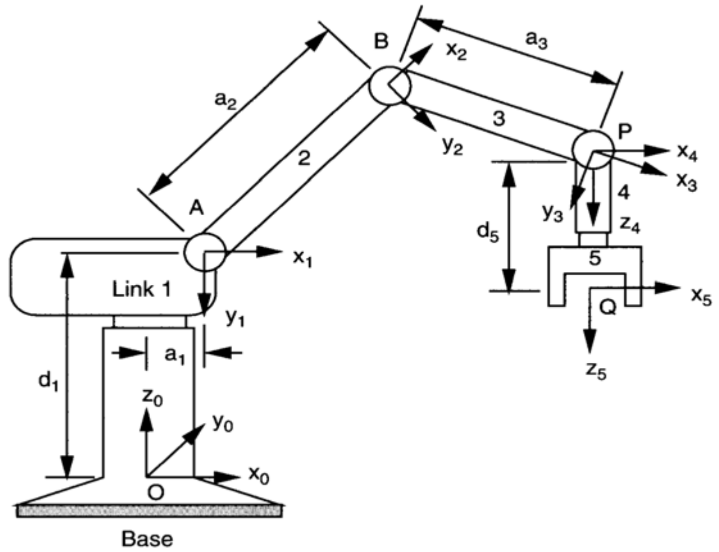

Figure 1.1: Schematic Diagram of SCORBOT-ER V Plus Robot Manipulator

For the kinematic model of SCORBOT first we have to assign frame to each link starting from base (frame {0}) to end-effector (frame {5}). The frame assignment is shown in Figure 1.1. The corresponding link parameters are listed in Table 1.1.

Table1.1: D-H Parameter for SCORBOT-ER V Plus

Joint i	⍰⍰	⍰⍰(⍰ ⍰)⍰	⍰⍰(⍰ ⍰)⍰	⍰⍰	Operating range
1	- ⍰/2	16	349	⍰1	−155° ⍰⍰+ 155°
2	0	221	0	⍰2	−35° ⍰⍰+ 130°
3	0	221	0	⍰3	−130° ⍰⍰+ 130°
4	-⍰/2	0	0	⍰/2 + ⍰4	−130° ⍰⍰+ 130°
5	0	0	145	⍰5	−570° ⍰⍰570°

2.1. *FORWARD KINEMATIC OF SCORBOT–ER V PLUS*

Once the DH coordinate system has been established for each link, a homogeneous transformation matrix can easily be developed considering frame {i-1} and frame {i}. This transformation consists of four basic transformations.

$$^{0}T_5 = {}^{0}T_1 * {}^{1}T_2 * {}^{2}T_3 * {}^{3}T_4 * {}^{4}T_5 \qquad (1)$$

Finally, the transformation matrix is as follow: -

$$T = {}^{0}T_{5} = \begin{bmatrix} -S_1S_5 - C_1C_5S_{234} & -C_5S_1 + C_1S_5S_{234} & C_1C_{234} & C_1(a_1 + a_2C_2 + a_3C_{23} + d_5C_{234}) \\ C_1S_5 - S_1C_5S_{234} & C_1C_5 + S_1S_5S_{234} & S_1C_{234} & S_1(a_1 + a_2C_2 + a_3C_{23} + d_5C_{234}) \\ -C_5C_{234} & S_5C_{234} & -S_{234} & (d_1 - a_2S_2 - a_3S_{23} - d_5S_{234}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Where, $S_i = (\theta_i)$, $C_i = (\theta_i)$ *and* $S_{ijk} = (\theta_i + \theta_j + \theta_k)$, $C_{ijk} = (\theta_i + \theta_j + \theta_k)$.
The T is all over transformation matrix of kinematic model of SCORBOT-ER V Plus, from this we have to extract position and orientation of end –effector with respect to base is done in the following section. As SCORBOT-ER-V is having 5 DOF, only five of the six of end effector parameters can be specified.

2.2. *Obtaining Position in Cartesian Space*

The value of x, y, z is found from last column of transformation matrix as: -

$$X = C_1(a_1 + a_2C_2 + a_3C_{23} + d_5C_{234}) \quad (3)$$

$$Y = S_1(a_1 + a_2C_2 + a_3C_{23} - d_5C_{234}) \quad (4)$$

$$Z = (d_1 - a_2S_2 - a_3S_{23} - d_5S_{234}) \quad (5)$$

Pitch: Pitch is the angle of rotation about y_5 axis of end-effector

$$pitch\beta = \theta_2 + \theta_3 + \theta_4 = \theta_{234} \quad (6)$$

$$\theta_{234} = \mathrm{a}\tan 2(r13, \pm\sqrt{r23^2 + r33^2}) \quad (7)$$

Here we use atan2 because its range is $[-\pi, \pi]$, where the range of atan is $[-\pi/2, \pi/2]$.
Roll: The $roll\ \gamma = \theta_5$ is derived as follow: -

$$\theta_5 = a\tan 2(r12 / C_{234}, r11 / C_{234}) \quad (8)$$

Yaw: Here for SCORBOT *yaw* is not free and bounded by θ_1.

2.3. Inverse Kinematics of SCORBOT-ER V Plus

For SCORBOT we have five parameter in Cartesian space is x, y, z, roll (β), *pitch* (γ).For joint parameter evaluation we have to construct transformation matrix from five parameters in Cartesian coordinate space. For that rotation matrix is generated which is depends on only *roll, pitch and yaw* of robotic arm. For SCORBOT there is no *yaw* but it is the rotation of first joint θ_1.
So the calculation of *yaw* is as follow: -

$$\alpha = \theta_1 = a\tan 2(x, y) \quad (9)$$

So, the total transformation matrix is as follows: -

$$T = \begin{bmatrix} -S_\beta S_\alpha - C_\alpha C_\beta S_\gamma & -S_\beta C_\alpha + S_\alpha C_\beta S_\gamma & C_\beta C_\gamma & X \\ C_\beta S_\alpha - C_\alpha S_\beta S_\gamma & C_\beta C_\alpha + S_\beta S_\gamma S_\alpha & C_\gamma S_\beta & Y \\ -C_\alpha C_\gamma & C_\gamma S_\alpha & -S_\gamma & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

After comparing the transformation matrix in equation (2) with matrix in equation (14), one can deduce: -

$$\theta_1 = \alpha,$$

$$\theta_{234} = \beta,$$

$$\theta_5 = \gamma,$$

$$\theta_3 = a\tan 2(\pm\sqrt{1 - \cos^2\theta_3}, \cos\theta_3) \quad (11)$$

$$\theta_2 = -a\tan 2(Y_3, X_3) - a\tan 2(a_3\sin\theta_3, a_2 + \cos\theta_3) \quad (12)$$

Finally we will get: -

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \quad (13)$$

3. Trajectory Control

A trajectory is the track followed by the manipulator, plus the time profile along the path. Trajectories can be calculated either in joint space (directly specifying the time evolution of the joint angles) or in Cartesian space (specifying the position and orientation of the end frame). Matters in trajectory planning include attaining a specific target from an initial starting point, avoiding obstacles, and staying within manipulator capabilities. Planning in joint space is the simplest and fastest, because inverse kinematics is avoided. The shortcoming is that the end effector posture is not directly controlled, and hence collision avoidance is challenging. Planning in Cartesian space allows being met the geometric constraints of the external world, but only after solving inverse kinematics. Two algorithms were developed to test the trajectory tracking based on positional analysis.

3.1. *Joint Space TRAJECTORY and Cartesian Space Trajectory*

The process followed for developing joint space as well as Cartesian space trajectory tracking algorithm is given below: -

- Define initial as well as final point.
- Whether joint space or Cartesian motion/?
- If joint space then Calculate inverse kinematics solution from initial point to the final point.
- Assign total time using maximal velocities in joints.
- Discretize the individual joint trajectories in time.
- Blend a continuous function between these points.
- If Cartesian space then Calculate path from the initial point to the final point.
- Assign a total time to traverse the path.
- Discretize the points in time and space.
- Blend a continuous time function between these points.
- Solve inverse kinematics at each step.

A flow chart based on the established algorithm is given in Figure 1.2 as follows: -

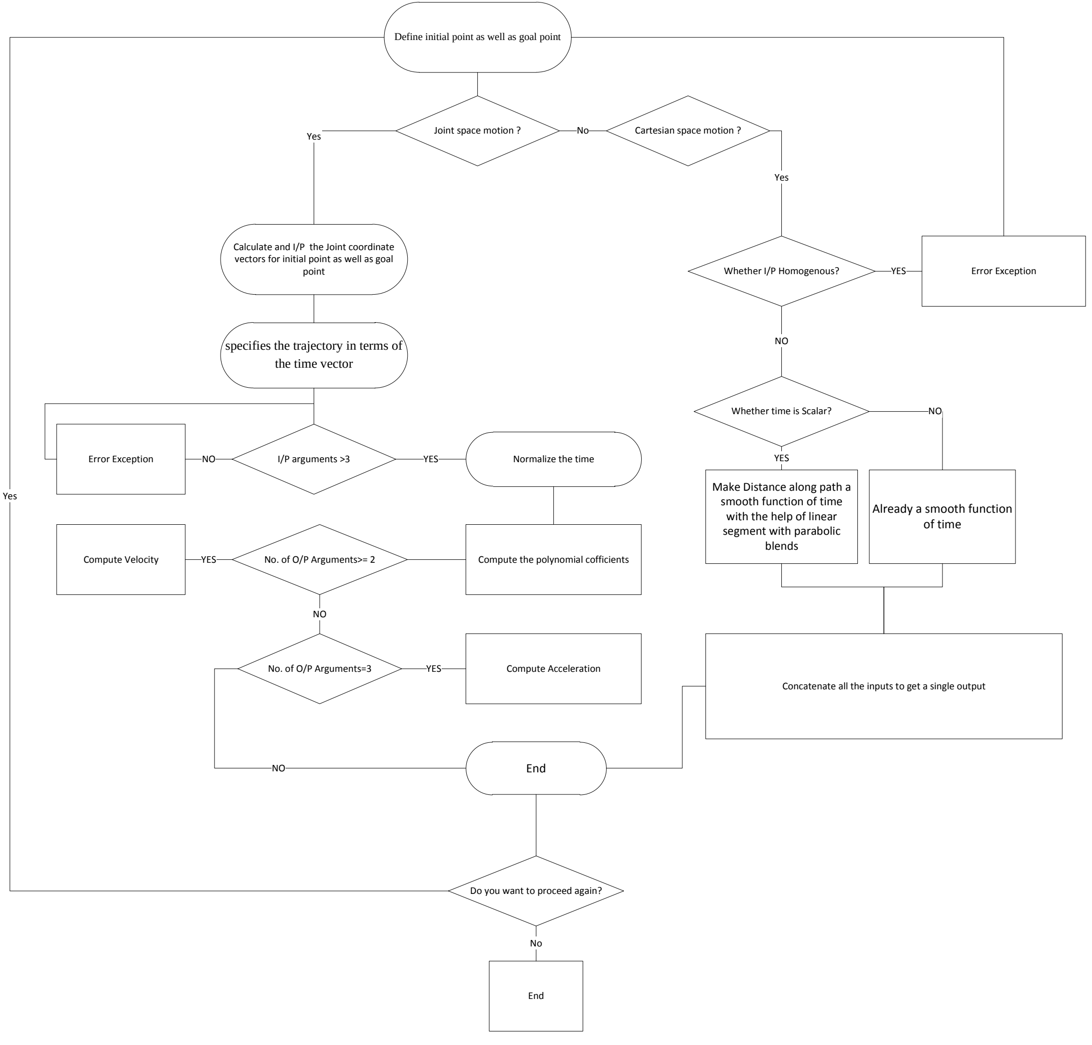
Define initial point as well as goal point
Yes
Joint space motion ?
No
Cartesian space motion ?
Yes
Calculate and I/P the Joint coordinate vectors for initial point as well as goal point
Whether I/P Homogenous?
YES
Error Exception
NO
specifies the trajectory in terms of the time vector
Whether time is Scalar?
NO
Error Exception
NO
I/P arguments >3
YES
Normalize the time
YES
Yes
Make Distance along path a smooth function of time with the help of linear segment with parabolic blends
Already a smooth function of time
Compute Velocity
YES
No. of O/P Arguments>= 2
Compute the polynomial cofficients
NO
No. of O/P Arguments=3
YES
Compute Acceleration
Concatenate all the inputs to get a single output
NO
End
Do you want to proceed again?
No
End

Figure 1.2: Joint and Cartesian space motion flow chart

4. Simulation Results and Discussion

4.1. *Joint Space Trajectory*

The end-effector movement in between two Cartesian poses which falls in the XY plane was considered. The oriented of the end-effector was taken downwardly. Time period for the whole exercise was of 2 seconds in 50ms of time steps. The joint, velocity and acceleration were attained as a function of time by designing a joint space trajectory for smoothly incorporating between two formations. The joint angles contrasted with time are revealed in Figure 1.3. Though it illustrates a smooth joint coordinate trajectory but unable to provide perfect evidence regarding Cartesian space end- effector trajectory. But by applying forward kinematics to the joint coordinate trajectory, it can be calculated easily.

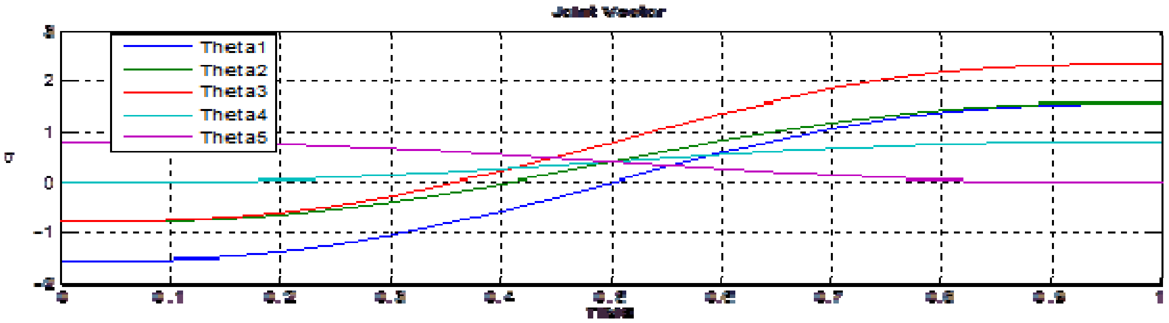

Figure 1.3: The Joint Coordinate Vector

The end-effector position in Cartesian space is plotted against time in Figure 1.4.

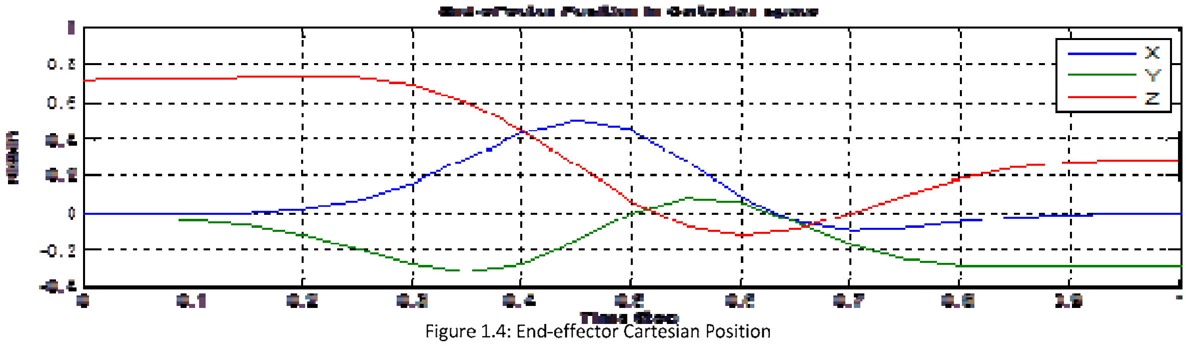

Figure 1.4: End-effector Cartesian Position

End-effector route in the XY plane is shown in Fig. 1.5 and one can easily predict that the track is not a straight line. This is to be anticipated due to the Cartesian nature of the end-points. Due to rotation around the waist, the end-effector will logically track a circular arc. It can cause accidents in reality among the robot and neighboring entities though they do not fall on the track between poses A and B.

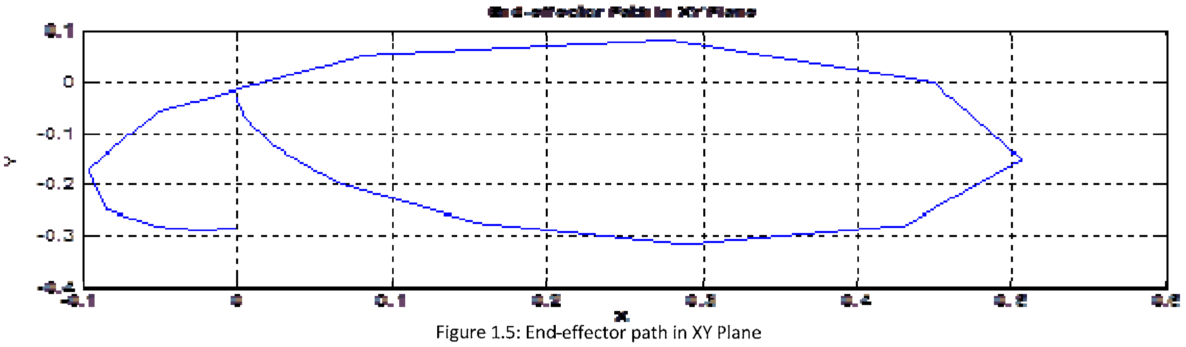

Figure 1.5: End-effector path in XY Plane

The orientation of the end-effector in Eular angle form is plotted in Figure 1.6.

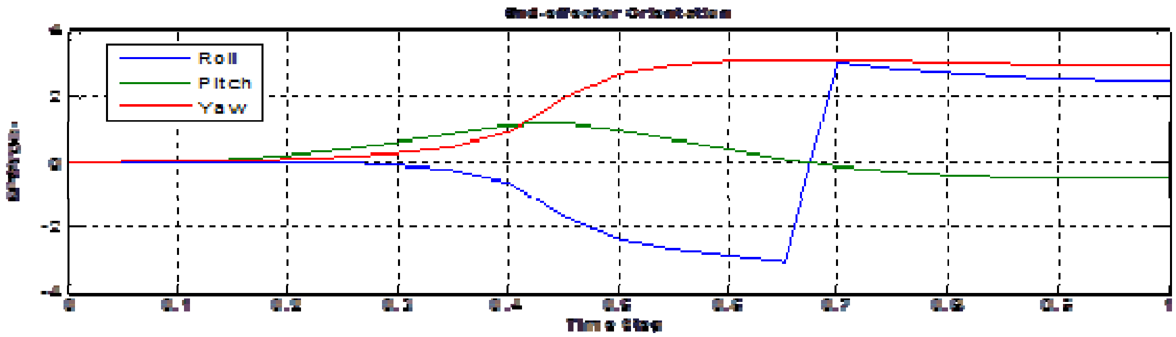

Figure 1.6: End-effector Orientation

4.2. *Cartesian Space Trajectory*

Straight-line motion for many applications is required in Cartesian space which is known as Cartesian motion. The equivalent joint-space trajectory by applying the inverse kinematics is acquired and is drawn in Figure 1.7.

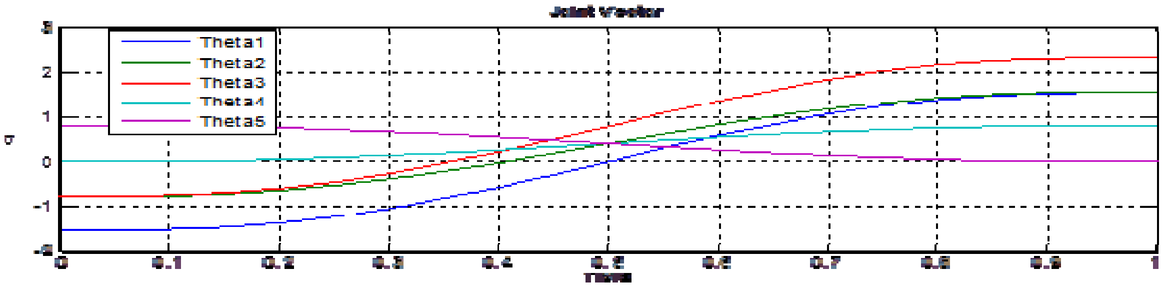

Figure 1.7: The Joint Coordinate Vector

The end-effector position in Cartesian space is plotted against time in Figure 1.8.

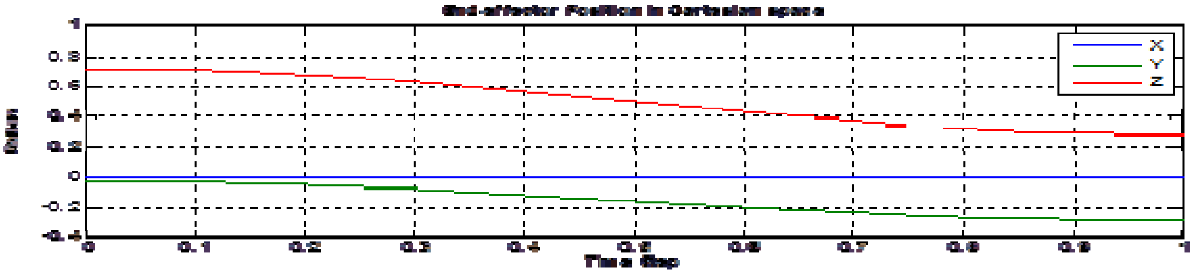

Figure 1.8: End-effector Cartesian Position

If Figure 1.4 and Figure 1.5 are compared, one can find certain significant dissimilarities. The position in Figure 1.9 as well as orientation presented in Figure 1.10 varies linearly with time. For orientation the pitch angle is constant at zero at start as well as end and varies negatively along the path. The end-effector tracks a straight line in the XY plane as presented in Figure. 1.9.

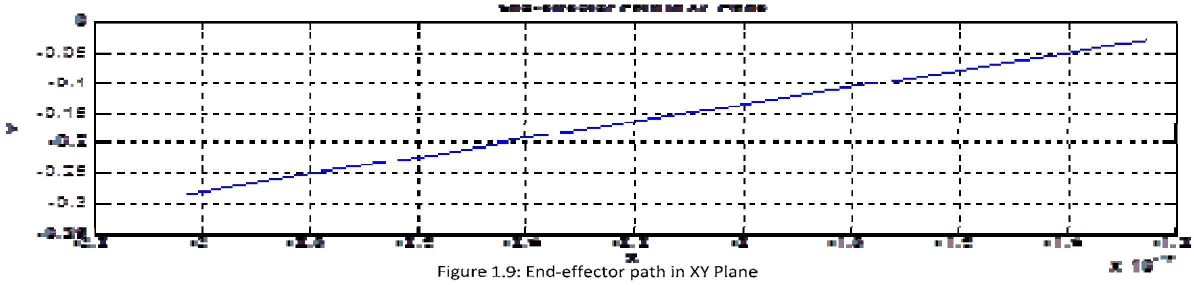

Figure 1.9: End-effector path in XY Plane

The orientation of the end-effector in Eular angle form is plotted in Figure 1.10.

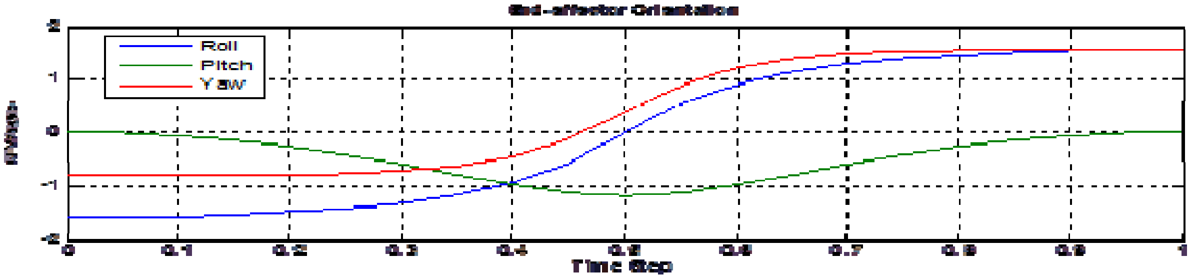

Figure 1.10: End-effector Orientation

4.3. *Motion Through Singularity*

The Cartesian trajectory endpoints taken previously were intentionally altered for a robot to travel in the y-direction with the end-effector z-axis pointing in the x-direction through singularity. Due to this, the manipulator has lost one degree of freedom and acts as 5-DOF manipulator. The joint space tracking between the two postures marked in Figure 1.3 is invulnerable to this difficulty since it is does not need the inverse kinematics, but it will sustain the posture of the tool in the *x*-direction only at the two end points, not for the entire pathway. The manipulability measure for the track was virtually zero about the time of the swift wrist joint motion, is displayed in Figure 1.11.

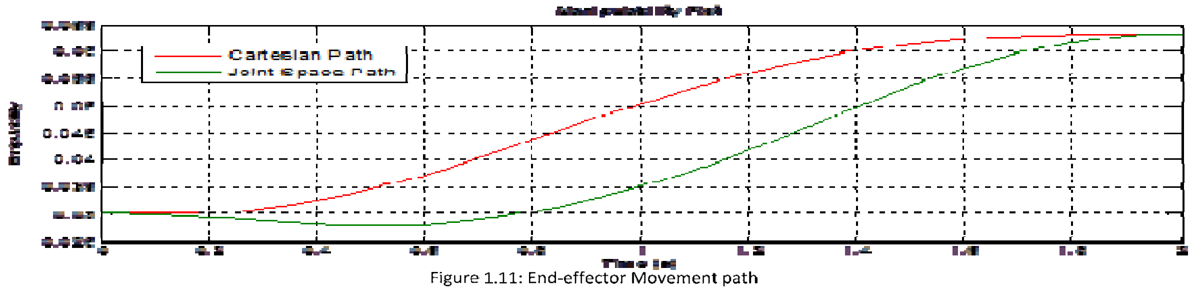

Figure 1.11: End-effector Movement path

5. Conclusion

Collision avoidance is difficult due to non-operability of end-effector in case of joint space trajectory tracking. On the other to realize the geometric constraints of the external world in Cartesian space one has to solve the inverse kinematics. The methodology presented here can be extended to be used for trajectory planning and quite a few tracking applications based on positional analysis with real world disturbances. The present work did not make use of dynamics of robot manipulator, so it could be extended for the same also. Two methodologies principally centered on joint space as well as Cartesian space tracking for tracking the end-effector effortlessly between dissimilar postures were presented. The precision of the output of the projected algorithms for effective kinematical tracking control of industrial robot can be experienced from the simulation results. Joint space trajectory may be challenging for some applications due to not result in Cartesian space straight line trajectory. Due to singularities in the workspace, though straight line Cartesian space trajectory can be established but may lead to very high joint rates.

Acknowledgments

As it is the case in almost all parts of human endeavor so also the development in the field of robotics has been carried on by engineers and scientists all over the world. It can be regarded as a duty to express the appreciation for such relevant, interesting and outstanding work to which ample reference is made in this paper.

References

[1] Featherstone R, "Position and velocity transformation between robot end-effector coordinate and joint angle," *International Journal of Robotics*, vol. 2, no. 2, pp. 35-45, 1983.

[2] Lee G. C. S., "Robot arm kinematics, dynamics and control," *IEEE*, vol. 15, no. 12, pp. 62-79, 1982.

[3] Duffy J., *Analysis of Mechanism and Robot Manipulators*. New York, USA: Wiley, 1980.

[4] D. Manocha and JF Canny, "Efficient inverse kinematics for general 6r manipulators," *IEEE Transaction on Robotics Automation, IEEE*, vol. 10, no. 5, pp. 648-657, 1994.

[5] RP Paul, B. Shimano, and GE Mayer, "Kinematic control equations for simple manipulators," *IEEE Transaction on System, Man and Cybernetics, IEEE*, vol. SMC-11, no. 6, pp. 66-72, 1982.

[6] JU Korein and NI Balder, "Techniques for generating the goal-directed motion of articulated structures," *IEEE Computer Graphics and Application, IEEE*, vol. 2, no. 9, pp. 71-81, 1982.

[7] D. M. A. Lee, R. O. Buchal, W. H. Elmaraghy H. Jack, "Neural Networks and The Inverse Kinematics Problem," *Journal of Intelligent Manufacturing*, vol. 4, pp. 43-66, 1993.

[8] Aluizio F. R. Arahjo and Hclio D'arbo Jr., "A Partially Recurrent Neural Network To Perform Trajectory Planning, Inverse Kinematics and Inverse Dynamics," *Systems, Man and Cybernetics, IEEE*, vol. 2, pp. 1784 - 1789, 1998.

[9] David W. Howard and Ali Zilouchian, "Application of Fuzzy Logic for The Solution Of Inverse Kinematics And Hierarchical Controls Of Robotic Manipulators," *Journal of Intelligent and Robotic Systems, Kluwer Academic Publishers*, vol. 23, pp. 217–247, 1998.

[10] Eimei Oyama, Nak Young Chong, Arvin Agah, Taro Maeda, and Susumu Tachi, "Inverse Kinematics Learning By Modular Architecture Neural Networks With Performance Prediction Networks," *International Conference On Robotics & Automation, IEEE*, pp. 1006-1012, 2001.

[11] Rasit Koker, Cemil Oz, Tarik Cakar, and Huseyin Ekiz, "A Study Of Neural Network Based Inverse Kinematics Solution For A Three-Joint Robot," *Robotics and Autonomous Systems, Elsvier*, vol. 49, pp. 227–234, 2004.

[12] Z. Bingul, H.M. Ertunc, and C. Oysu, "Comparison Of Inverse Kinematics Solutions Using Neural Network For 6r Robot Manipulator With Offset," *Computational Intelligence Methods And Applications, IEEE*, pp. 1-5, 2005.

[13] D.T. Pham, M. Castellani, and A. A. Fahmy, "Learning The Inverse Kinematics Of A Robot Manipulator Using The Bees Algorithm," *International Conference On Industrial Informatics, IEEE*, pp. 493-498, 2008.

[14] Ren e V. Mayorga and Pronnapa Sanongboon, "Inverse kinematics and geometrically bounded singularities prevention of redundant manipulators: An Artificial Neural Network approach," *Elsevier, Robotics and Autonomous Systems*, vol. 53, pp. 164–176, 2005.

[15] Rasit Koker, "Reliability-based approach to the inverse kinematics solution of robots using Elman's networks," *Elsevier, Engineering Applications of Artificial Intelligence*, vol. 18, pp. 685–693, 2005.

[16] Ali T. Hasan, A.M.S. Hamouda, N. Ismail, and H.M.A.A. Al-Assadi, "An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 D.O.F serial robot manipulator," *Elsevier, Advances in Engineering Software*, vol. 37, pp. 432–438, 2006.

[17] Himanshu Chaudhary and Rajendra Prasad, "Intelligent Inverse Kinematic Control of SCORBOT-ER V Plus Robot Manipulator," *International Journal of Advances in Engineering & Technology*, vol. 1, no. 5, pp. 158-169, November 2011.

[18] Kwang Sik Eoma, Il Hong Suha, and Wan Kyun Chungb, "Disturbance observer based path tracking control of robot manipulator considering," *PERGAMON, MECHATRONICS*, vol. 11 , pp. 325-343, 2000.

[19] Zaer S. Abo-Hammoura, Nasir M. Mirza, Sikander M. Mirza, and Muhammad Arif, "Cartesian path generation of robot manipulators using continuous genetic algorithms," *Elsevier, Robotics and Autonomous Systems*, vol. 41, pp. 179–223, 2002.

[20] Xuan F. Zha, "Optimal pose trajectory planning for robot manipulators," *PERGAMON, Mechanism and Machine Theory*, vol. 37, pp. 1063–1086, 2002.

[21] T. Chettibi, H.E. Lehtihet, M. Haddada, and S. Hanchi b, "Minimum cost trajectory planning for industrial robots," *Elsevier, European Journal of Mechanics A/Solids*, vol. 23, pp. 703–715, 2004.

[22] Marcos de Sales Guerra Tsuzuki, Thiago de Castro Martins, and Fdbio Kawaoka Takase, "ROBOT PATH PLANNING USING SIMULATED ANNEALING," *Elsevier, Information Control Problems in Manufacturing - MCOM*, pp. 173-178, 2006.

[23] Gasparetto and V. Zanotto, "A new method for smooth trajectory planning of robot manipulators," *Elsevier, Mechanism and Machine Theory*, vol. 42, pp. 455–471, 2007.

[24] Zhenwang Yao and Kamal Gupta, "Path planning with general end-effector constraints," *Elsevier, Robotics and Autonomous Systems*, vol. 55, pp. 316–327, 2007.

[25] M. Boryga and A. Grabos, "Planning of manipulator motion trajectory with higher-degree polynomials use," *Elsevier, Mechanism and Machine Theory*, vol. 44, pp. 1400–1419, 2009.

[26] Wanghui Bu, Zhenyu Liu, Jianrong Tan, and Shuming Gao, "Detachment avoidance of joint elements of a robotic manipulator with clearances based on trajectory planning," *Elsevier, Mechanism and Machine Theory*, vol. 45, pp. 925–940, 2010.

[27] Gasparetto and V. Zanotto, "Optimal trajectory planning for industrial robots," *Elsevier, Advances in Engineering Software*, vol. 41, pp. 548–556, 2010.

[28] Claire Dumas, Stephane Caro, Sebastien Garnier, and Benoıt Furet, "Joint stiffness identification of six-revolute industrial serial robots," *Elsevier, Robotics and Computer-Integrated Manufacturing*, vol. 27, pp. 881–888, 2011.