A Comparison Study of the Numerical Integration Methods in the Trajectory Tracking Application of Redundant Robot Manipulators

Emre Sarıyıldız and Hakan Temeltaş

Istanbul Technical University, 34469 Istanbul, Turkey
esariyildiz@itu.edu.tr, hakan.temeltas@itu.edu.tr

Abstract

Differential kinematic has a wide range application area in robot kinematics. The main advantage of the differential kinematic is that it can be easily implemented any kind of mechanisms. In differential kinematic method, Jacobian is used as a mapping operator in the velocity space. The joint velocities are required to be integrated to obtain the pose of the robot manipulator. This integration can be evaluated by using numerical integration methods, since the inverse kinematic equations are highly complex and nonlinear. Thus, the performances of the numerical integration methods affect the trajectory tracking application. This paper compares the performances of numerical integration methods in the trajectory tracking application of redundant robot manipulators. Four different and widely used numerical integration methods are implemented to the trajectory tracking application of the 7-DOF redundant robot manipulator named PA-10 and simulation results are given.

1. Introduction

Redundant robot manipulators have wide range application areas in many robotic applications such as obstacle avoidance, singularity avoidance, complex manipulation, service robots and humanoids [1, 2, 3 and 4]. The main advantage of redundant robot manipulators is that their configurations offer the potential to overcome many difficulties by increased manipulation ability and versatility [5 and 6]. However the redundant robot manipulators have many advantages, they have quite complex control structures and suffer from singularity problem.

A fundamental research task of redundant robot manipulation is to find out the appropriate way to control the system of redundant robot manipulator in the work space at any stage of the trajectory tracking. This control can be achieved by using dynamic or kinematic models based solutions. However a dynamic model based solutions give more realistic results than kinematic based solutions, they have quite complex structures. Therefore, kinematic model based solutions are generally used in many robotic applications which do not require force and torque controls.

Differential kinematic is one of the most important solution methods to cope with the redundancy problem [7, 8]. The main advantage of the differential kinematic is that it can be easily implemented any kind of mechanism. Also, accurate and efficient kinematic based trajectory tracking applications can be easily implemented by using this method. Jacobian is used as a velocity mapping operator which transforms the joint velocities into the Cartesian linear and angular velocities. A highly complex and nonlinear inverse kinematic problem can be numerically solved by just inversing the Jacobian matrix operator. However, differential kinematic based solutions can be easily implemented any kind of mechanisms, it has some disadvantages. The first one is that differential kinematic based solutions are locally linearized approximation of the inverse kinematic problem [9]. The second one is that it has heavy computational calculation and big computational time because of numerical iterative approach [10]. And the last disadvantage of this method is that, it requires numerical integration which suffers from numerical errors, to obtain the joint positions from the joint velocities [11]. The numerical integration of joint velocities to compute joint positions causes a numerical drift which in turn corresponds to a task space error [12-13]. An effective inversion of differential kinematics mappings can be realized by adopting the so-called closed-loop inverse kinematics algorithms which are based on the use of a feedback correction term on the task space error [14]. However the drift-phenomena can be overcome by using the closed-loop inverse kinematic algorithm, the performance of the algorithm is still extremely affected by the chosen numerical integration method.

In this paper, a performance analysis of the numerical integration methods in the trajectory tracking application of the redundant robot manipulators is presented in detail. Two single-step numerical integration methods which are called Euler Integration and Runge-Kutta 4 and also two multi-step numerical integration methods which are called Predictor & Corrector, and Adams-Moulton methods are implemented into the differential kinematic based solution of the trajectory tracking application of the redundant robot manipulators. These methods are compared with respect to computational efficiency and accuracy. Simulation results are given in section V. This paper is also included the differential robot kinematics in section II, numerical integration methods in section III, trajectory tracking algorithms in section IV. Conclusions and future works are drawn in the final section.

2. Differential Kinematic Model

It is very hard even impossible to find the analytical solutions of the inverse kinematic problem of the redundant robot manipulators except the limited special structures or very easy mechanisms. Therefore, differential kinematic based solution of the inverse kinematic problem of the redundant robot manipulators is widely used [15]. In the differential kinematic based solution, a velocity mapping which transforms the Cartesian linear and angular velocities of the robot manipulator's end effector into the joint velocities, is used as follows,

$$\dot{\mathbf{q}} = J^g \ \mathbf{q} \ \mathbf{V_{tip}} \tag{1}$$

where $\mathbf{q} = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix}$ indicates the joint angles and $\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \cdots & \dot{q}_n \end{bmatrix}$ indicates the joint velocities. $J^g \ \mathbf{q}$ indicates generalized inverse of the Jacobian matrix and $\mathbf{V_{tip}}$ indicates the linear and angular velocities of the robot manipulator's end effector.

Jacobian can be obtained by using analytical or geometric approaches which can be found in many basic robotic books [16-17]. The joint angles can be found by integrating the joint velocities given by

$$\mathbf{q} = \int_0^t \dot{\mathbf{q}} dt = \int_0^t J^g \ \mathbf{q} \ \mathbf{V_{tip}} dt \tag{2}$$

3. Numerical Integration of the Joint Velocities

The joint angles are obtained by numerically integrating the joint velocities. Therefore, the chosen numerical integration method extremely affects the computational efficiency and accuracy of the differential kinematic based trajectory tracking algorithm.

Here, several numerical integration methods are introduced. These integration methods can be divided into two main different approaches. These are single-step numerical integration methods which called Explicit Euler Integration and Runge-Kutta 4 and multi-step numerical integration methods which called Euler Trapezoidal Predictor & Corrector, and Adams Moulton methods. The formulations of these integration methods are as follows, [18-19]

3.1 Explicit Euler Integration Method

Explicit Euler integration is the simplest numerical integration method. It can be formulated as follows

$$\mathbf{q} \ t_{k+1} \ = \mathbf{q} \ t_k \ + \dot{\mathbf{q}} \ t_k \ \Delta t \ . \tag{3}$$

where $\dot{\mathbf{q}} \ t_k \ = J^g \ \mathbf{q} \ t_k \ \mathbf{V_{tip}} \ t_k$ (4)

The strengths of this method are that it can be easily implemented and also it has a very computationally light equation. However, the accuracy of this method is quite poor.

3.2 Runge-Kutta 4 Method

The formulation of the fourth order Runge-Kutta numerical integration method is as follows,

$$\mathbf{q} \ t_{k+1} \ = \mathbf{q} \ t_k \ + \frac{1}{6} \ \dot{\mathbf{q}}_1 \ t_k \ + 2\dot{\mathbf{q}}_2 \ t_k \ + 2\dot{\mathbf{q}}_3 \ t_k \ + \dot{\mathbf{q}}_4 \ t_k \tag{5}$$

where

$$\dot{\mathbf{q}}_1 \ t_k \ = J^g \ \mathbf{q} \ t_k \ \mathbf{V_{tip}} \ t_k$$

$$\dot{\mathbf{q}}_2 \ t_k \ = J^g \ \mathbf{q}_1 \ t_k \ \mathbf{V_{tip}} \left(t_{k+\frac{1}{2}} \right)$$

$$\dot{\mathbf{q}}_3 \ t_k \ = J^g \ \mathbf{q}_2 \ t_k \ \mathbf{V_{tip}} \left(t_{k+\frac{1}{2}} \right)$$

$$\dot{\mathbf{q}}_4 \ t_k \ = J^g \ \mathbf{q}_3 \ t_k \ \mathbf{V_{tip}} \ t_{k+1} \tag{6}$$

in which

$$\mathbf{q}_1 \ t_k \ = \mathbf{q} \ t_k \ + \frac{\Delta t}{2} \ J^g \ \mathbf{q} \ t_k \ \mathbf{V_{tip}} \ t_k$$

$$\mathbf{q}_2 \ t_k \ = \mathbf{q} \ t_k \ + \frac{\Delta t}{2} \left(J^g \ \mathbf{q}_1 \ t_k \ \mathbf{V_{tip}} \left(t_{k+\frac{1}{2}} \right) \right)$$

$$\mathbf{q}_3 \ t_k \ = \mathbf{q} \ t_k \ + \Delta t \left(J^g \ \mathbf{q}_2 \ t_k \ \mathbf{V_{tip}} \left(t_{k+\frac{1}{2}} \right) \right) \tag{7}$$

This method requires four calculations of the generalized inverse Jacobian for each step, so that the computational load of this method is higher than Explicit Euler Integration method. This extra computation improves the numerical integration results and the solutions which are more accurate and stable than Explicit Euler Integration method based solutions, can be derived by using this method.

3.3 Euler Trapezoidal Predictor & Corrector Method

Euler Trapezoidal Predictor & Corrector method is an algorithm that proceeds in two steps. In the first step, a rough approximation of the desired quantity is calculated. And the second step, the initial approximation is refined using another means. The formulation of this method is as follows,

$$\mathbf{q} \ t_{k+1} \ = \mathbf{q} \ t_k \ + \frac{\Delta t}{2} \ \dot{\mathbf{q}} \ t_k \ + \dot{\hat{\mathbf{q}}} \ t_{k+1} \tag{8}$$

where $\dot{\hat{\mathbf{q}}} \ t_k$ are the predicted joint velocities in which

$$\hat{\mathbf{q}} \ t_{k+1} \ = \hat{\mathbf{q}} \ t_k \ + \dot{\hat{\mathbf{q}}} \ t_k \ \Delta t$$

$$\hat{\mathbf{q}} \ t_{k+1} \ = \hat{\mathbf{q}} \ t_k \ + J^g \ \hat{\mathbf{q}} \ t_k \ \mathbf{V_{tip}} \ t_k \ \Delta t \tag{9}$$

Euler Trapezoidal Predictor & Corrector method is also requires two computation of the generalized inverse of Jacobian operator so that the computational load increases. It gives more accurate and stable results than Euler Integration method.

3.4 Adams-Moulton Method (Fourth Order)

Adams-Moulten is a widely used multi-step implicit numerical integration method. Here, Adams-Bashforth algorithm is used in the numerical integration of the predicted joint velocities and Adams-Moulton algorithm is used in the numerical integration of the corrected joint velocities. It can be formulated as follows,
Predictor Algorithm (Adams-Bashforth Algorithm)

If $t_k = t_1$, then $\hat{\mathbf{q}} \ t_{k+1} \ = \hat{\mathbf{q}} \ t_k \ + \dot{\hat{\mathbf{q}}} \ t_k \ \Delta t$ (10)

If $t_k = t_2$, then $\hat{\mathbf{q}} \ t_{k+1} \ = \hat{\mathbf{q}} \ t_k \ + \frac{\Delta t}{2} \ 3\dot{\hat{\mathbf{q}}} \ t_k \ - \dot{\hat{\mathbf{q}}} \ t_{k-1}$ (11)

If $t_k = t_3$, then

$$\hat{\mathbf{q}} \ t_{k+1} \ = \hat{\mathbf{q}} \ t_k \ + \frac{\Delta t}{12} \ 23\dot{\hat{\mathbf{q}}} \ t_k \ - 16\dot{\hat{\mathbf{q}}} \ t_{k-1} \ + 5\dot{\hat{\mathbf{q}}} \ t_{k-2} \tag{12}$$

If $t_k \geq t_4$, then

$$\hat{\mathbf{q}}(t_{k+1}) = \hat{\mathbf{q}}(t_k) + \frac{\Delta t}{24}\left(55\dot{\hat{\mathbf{q}}}(t_k) - 59\dot{\hat{\mathbf{q}}}(t_{k-1}) + 37\dot{\hat{\mathbf{q}}}(t_{k-2}) - 9\dot{\hat{\mathbf{q}}}(t_{k-3})\right) \tag{13}$$

where $\dot{\hat{\mathbf{q}}}(t_k) = J^g(\hat{\mathbf{q}}(t_k))\mathbf{V_{tip}}(t_k)$ (14)

Corrector Algorithm (Adams Moulton Method)

If $t_k = t_1$ then, $\mathbf{q}(t_{k+1}) = \mathbf{q}(t_k) + \frac{\Delta t}{2}\left(\dot{\hat{\mathbf{q}}}(t_{k+1}) + \dot{\mathbf{q}}(t_k)\right)$ (15)

If $t_k = t_2$ then,

$$\mathbf{q}(t_{k+1}) = \mathbf{q}(t_k) + \frac{\Delta t}{12}\left(5\dot{\hat{\mathbf{q}}}(t_{k+1}) + 8\dot{\mathbf{q}}(t_k) - \dot{\mathbf{q}}(t_{k-1})\right) \tag{16}$$

If $t_k \geq t_3$ then,

$$\mathbf{q}(t_{k+1}) = \mathbf{q}(t_k) + \frac{\Delta t}{24}\left(9\dot{\hat{\mathbf{q}}}(t_{k+1}) + 19\dot{\mathbf{q}}(t_k) - 5\dot{\mathbf{q}}(t_{k-1}) + \dot{\mathbf{q}}(t_{k-2})\right) \tag{17}$$

where $\dot{\hat{\mathbf{q}}}(t_k)$ are the predicted joint velocities

Here, we use the fourth order Adams-Moulten algorithm that is the most widely used one. This method requires two step backward values of joint velocities and one step forward predicted joint velocities. It also requires two computations of generalized inverse of Jacobian operator for each step so that computational load increases. This extra computation improves the numerical integration results and the solutions which are more accurate than Adams-Bashforth based solutions, can be derived by using this method.

4. Trajectory Tracking Application

The trajectory tracking application of the redundant robot manipulator is implemented by using the following two simulink block diagrams which are shown in figures 1 and 2. The first one shows us the trajectory tracking application by using the explicit numerical integration methods which are Euler Integration, and Runge-Kutta 4. In this application, a desired trajectory is generated for the end effector of the robot arm in the Desired Trajectory block and it is transferred to the Jacobian block. In the Jacobian block, the joint velocities are obtained by using the velocity mapping. Then the joint velocities are transferred to the Numerical Integration block. In the Numerical Integration block, explicit numerical integration methods are used to obtain the joint angles and these angles are transferred to the Forward Kinematics block. In the Forward Kinematics block, we obtain the pose of the robot manipulator's end effector and each robot manipulator's joints. The pose of the each robot manipulator's joints are required to obtain the Jacobian operator iteratively and the pose of the end effector is required to obtain the Jacobian operator iteratively and also the closed-loop kinematic structure. The closed loop inverse kinematic solution which can be shown in figure 1 is used to cope with the drift phenomena [14]. The second simulink block diagram shows us the trajectory tracking application by using the implicit numerical integration methods which are Euler Trapezoidal Predictor & Corrector and Adams-Moulton numerical integration methods. In the second simulink diagram, both trajectory tracking algorithms are obtained by using the first simulink diagram.

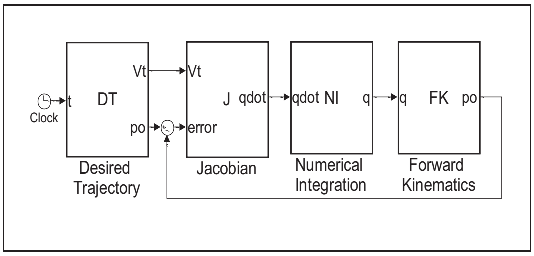

Fig.1. Simulink Block Diagram of Trajectory Tracking Simulation Application

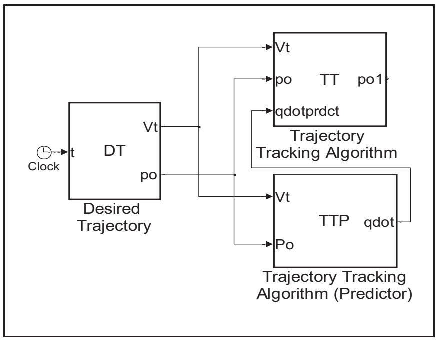

Fig.2. Simulink Block Diagram of Predictor Based Trajectory Tracking Simulation Application

5. Simulation Results

PA-10 redundant robot manipulator is used for the simulation studies. PA-10 robot arm features an articulated arm with 7 degrees of freedom for high flexibility. It spreads a wide range area in many robot applications. The simulation study of the trajectory tracking application is performed by using Matlab and the animation application is performed by using virtual reality toolbox (VRML) of Matlab which can be seen in figure 3.

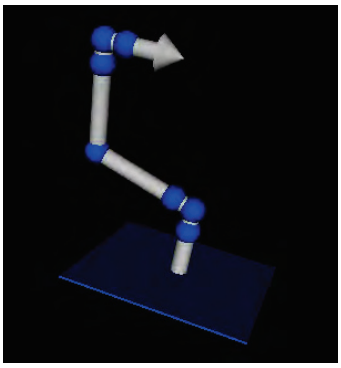

Fig.3. PA-10 Robot arm animation in virtual reality toolbox

A circular trajectory tracking application is implemented by using the proposed numerical integration methods and the algorithms are compared with respect to their computational efficiency and accuracy. The computational efficiency is very important requirement in the real time numerical integration applications. The computational efficiency results can be seen in figure 4.

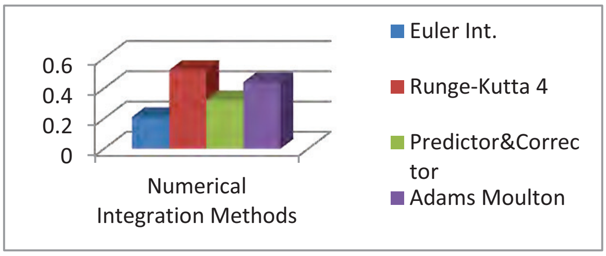

Fig.4. Simulation times of the numerical integration methods algorithms (second)

As it can be seen from the figure 4 the most computationally efficient method is Euler integration and the least computationally efficient method is Runge-Kutta 4. Accuracy is the other important requirement in the numerical integration applications. The accuracy results of the proposed numerical integration methods are given in the figures 5, 6, 7 and 8.

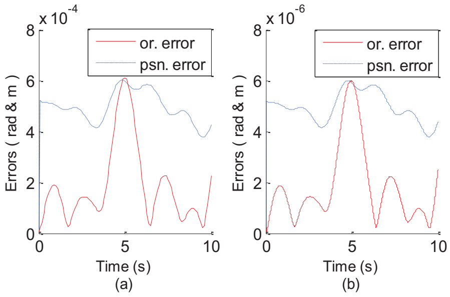

Fig.5. Total orientation and position errors (radian and meter) of the end effector for the Explicit Euler Integration method and the sampling rates are (a) $\Delta t = 100\ ms$ (b) $\Delta t = 10\ ms$

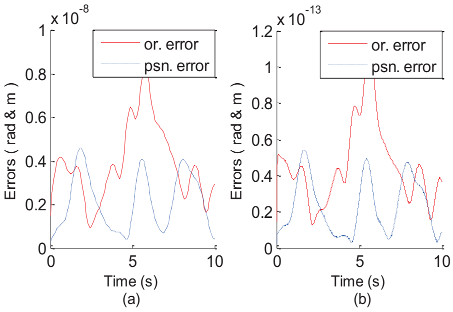

Fig.6. Total orientation and position errors (radian and meter) of the end effector for the Runge-Kutta 4 and the sampling rates are (a) $\Delta t = 100\ ms$ (b) $\Delta t = 10\ ms$

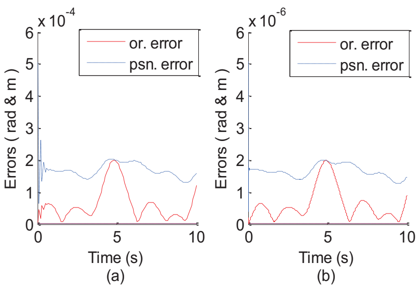

Fig.7. Total orientation and position errors (radian and meter) of the end effector for the Euler Trapezoidal Predictor & Corrector method and the sampling rates are (a) $\Delta t = 100\ ms$ (b) $\Delta t = 10\ ms$

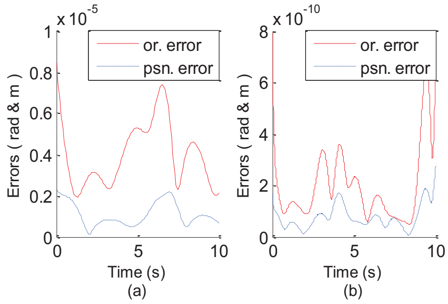

Fig.8. Total orientation and position errors (radian and meter) of the end effector for the fourth order Adams-Moulten method and the sampling rates are (a) $\Delta t = 100\ ms$ (b) $\Delta t = 10\ ms$

As it can be seen from the figures 5, 6, 7 and 8, the most accurate method is Runge-Kutta 4 and the least accurate method is Euler integration. Euler Integration based solution gives poor accuracy results in the trajectory tracking application. The results of this method can be seen in the figure 5. Small sampling rates which increase the computational loads of the trajectory tracking algorithm should be used to improve the accuracy of the numerical integration method.

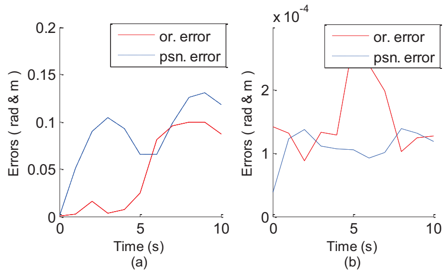

Fig.9. Total orientation and position errors (radian and meter) of the end effector at sampling rates $\Delta t = 1\ s$ for (a) Explicit Euler Integration (b) Runge-Kutta 4

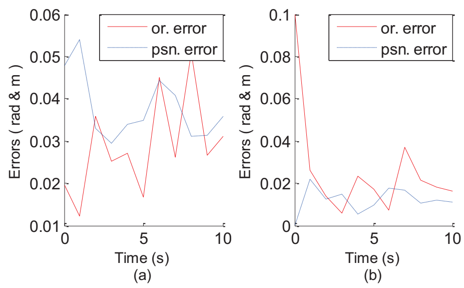

Fig.10. Total orientation and position errors (radian and meter) of the end effector at sampling rates $\Delta t = 1\ s$ for (a) Euler Trapezoidal Predictor & Corrector (b) Adams-Moulten

Beside the accuracy, sampling rate of the numerical integration method affects the stability of the system. If the sampling rate of the numerical integration method is too big then, the system may be unstable. The instability depends on both of the sampling rate and chosen numerical integration method. As it can be seen from the figure 10 Explicit Euler Integration based solution makes the system unstable and the errors get bigger when the sampling rate is $\Delta t = 1$ second. However, the result of Runge-Kutta 4 based solution is stable and the error is about 10^{-4} . In figure 11, the performance of the Euler Trapezoidal Predictor & Corrector and Adams-Moulton numerical integration methods can be seen when the sampling rate is $\Delta t = 1$ second. As it can be seen from the figure 11, both of the numerical integration methods give poor accuracy results at $\Delta t = 1$ second, however they still satisfy the stability.

6. Conclusion

In this paper, we analyzed the performance of numerical integration methods in the trajectory tracking application of the redundant robot manipulators. The performance of the trajectory tracking algorithm is drastically affected by the chosen numerical integration method. For instance, more accurate and more computationally efficient trajectory tracking algorithm can be obtained by changing the numerical integration methods. Even, the trajectory tracking algorithm may become unstable because of the chosen numerical integration method. Here, we compared four different numerical integration methods with respect to computational efficiency and accuracy. Among these methods, Runge-Kutta and Adams-Moulton numerical integration methods give satisfactory results. When we compare the Runge-Kutta and Adams-Moulton methods, Runge-Kutta based algorithm gives more accurate and stable results however; they require extra computation. Thus, the Adams-Moulton method is more computationally efficient than Runge-Kutta method. In the trajectory tracking application, Runge-Kutta based algorithm gives quite satisfactory results when the sampling rates are high. As the sampling rates increase, computational load of the trajectory tracking algorithm decreases. However Runge-Kutta based algorithms require extra computations and they have high computational load, the satisfactory results at high sampling rates may reduce even eliminate this disadvantage.

7. References

[1] M. Uchiyama, "Study in Dynamic Control of Artificial Arm-Part I", *Trans. JSME*, vol. 45, no. 391, pp 314-322, 1979

[2] G. Ping, B. Wei, X. Li and X. Luo, "Real-Time Obstacle Avoidance for Redundant Robot", *IEEE International Conference on Mechatronics and Automation*, pp. 223-228, Aug., 2009

[3] C. Cheni C. Lin, "Motion Planning of Redundant Robots", *Journal of Robotic Systems,* 14(12), pp. 839-850, 1997

[4] D. Bertram, J. Kuffner, R. Dillmann, T. Asfour, "An Integrated Approach to Inverse Kinematics and Path Planning for Redundant Manipulators", *IEEE, International Conference on Robotics and Automation (ICRA)*, pp. 1874-1879, May., 2006

[5] P. H. Chang, "A Closed-Form Solution for Inverse Kinematics of Robot Manipulators with Redundancy", *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 5, Oct. ,1987

[6] T. Yoshikawa, "Analysis and Control of Robot Manipulators with Redundancy," In *Robotics Research--The First International Symposium*, MIT Press, Cambridge, pp. 735-747, May, 1984

[7] W. A. Wolovich, H. Elliott, "A Computational Technique for Inverse Kinematics", *Proceedings of 23rd Conference on Decision and Control*, Dec., 1984

[8] F. Caccavale, P. Chiacchio, S. Chiaverini, B. Siciliano, "Experiments of Kinematic Control on a Redundant Robot Manipulator", *Laboratory Robotics and Automation*, vol. 8, pp. 25–36, 1996

[9] B. Siciliano, O. Khatib, "Handbook of Robotics", *Springer,*2008

[10] A.T. Hasan, N. Ismail, A.M.S Hamouda, I. Aris, M.H. Marhaban, H.M.A.A. Al-Assadi, "Artificial Neural Network-Based Kinematics Jacobian Solution for Serial Manipulator Passing through Singular Configurations", *Advances in Engineering Software,* vol.41, pp. 359–367, 2010

[11] S. Lin, J. Huang, "Numerical Integration of Multibody Mechanical Systems Using Baumgarte's Constraint Stabilization Method", *Journal of the Chinese Institute of Engineers*, vol. 25, no. 2, pp. 243-252 , 2002

[12] L. Sciavicco, B. Siciliano "Coordinate Transformation: A Solution Algorithm for One Class of Robots," *IEEE Transaction on Systems, Man and Cybenetics,* vol. 16, pp. 550-559, 1986.

[13] C.W. Wampler, L.J. Leifer, "Applications of Damped Least-Squares Methods to Resolved-Rate and Resolved-Acceleration Control of Manipulators," *Journal of Dynamic Systems, Measurement and Control*, vol. 110, pp. 31-38, 1988

[14] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Closed-Loop Inverse Kinematics Schemes for Constrained Redundant Manipulators with Task Space Augmentation and Task Priority Strategy", *Int. Journal of Robotics Research*, vol. 10, pp. 410-425, 1991.

[15] C.C. Cheah, M. Hirano, S. Kawamura, S. Arimoto, "Approximate Jacobian Control for Robots with Uncertain Kinematics and Dynamics", *IEEE Transactions on Robotics and Automation*, vol.19, no. 4, Aug., 2003

[16] M.W. Spong, S. Hutchinson, M. Vidyasagar, "Robot Modeling and Control", *Wiley*, 2005

[17] J.J. Craig, "Introduction to robotics: Mechanics and Control", 3rd edn*., Prentice Hall*, 2006

[18] R. J. Schilling, "Applied Numerical Methods for Engineers", *Brooks/Cole,* 2002

[19] E. Kreyszig, "Advanced Engineering Mathematics", *Wiley*, 8th Edition, 2000.