Military Technical College
Kobry El-Kobbah,
Cairo, Egypt.

18th International Conference
on Applied Mechanics and
Mechanical Engineering.

AN EFFICIENT FULL-NUMERICAL SOLUTION FOR THE INVERSE KINEMATICS OF THE 6-DOF FULL-ARTICULATED MANIPULATOR

A. A. Saad[1]

ABSTRACT

The inverse kinematics (IK) problem for the six degrees of freedom (DoF) full-articulated manipulator is solved either analytically through closed form solutions, or numerically through iterative algorithms. Analytical solutions are in general, if exist, yield redundant solutions at an affordable and reliable computational cost. Numerical solutions that employ iterative algorithms could yield an accurate single convergent solution that depends upon the picked initial guess, but at relatively higher computational cost. In this paper, an efficient numerical solution that employs an innovative implicit iterative algorithm is applied to the decoupled wrist and shoulder DoF, with the relevant jacobians approximated by first-order finite difference schemes. The presented numerical solution has been applied to the HUBO humanoid robot arm to solve for the IK of a single point in the workspace, and also for trajectory tracking problems. The results for both cases have demonstrated high accuracy at an appropriate computational cost suitable for real-time applications such as IK-based walking of humanoid robots.

KEYWORDS

Robot Inverse Kinematics, Numerical Solutions for Inverse Kinematics, 6-DoF Full-Articulated Manipulators.

NOMENCLATURE

The nomenclature is spottily declared the first time mentioned within the paper.

Abbreviations

CPU	Central Processing Unit.
D-H	The Denavit-Hartenberg Frame Convention [1]
DoF	Degrees-of-Freedom.
IK	Inverse Kinematics.
GB	Giga Byte.
GHz	Giga Hertz.
L.H.S	Left Hand Side.
RAM	Random Access Memory.
R.H.S	Right Hand Side.

[1] Assistant Professor, Department of Mechatronics Engineering, Pyramids High Institute for Eng. and Technology, 6th October City, Cairo, Egypt. Email: ahmsaad64@gmail.com

INTRODUCTION

Six DoF fully articulated manipulators are widely utilized in both industry, and extensively, in humanoid robots as well. While the end-effector trajectory is commonly planned and constructed in the workspace frame of reference, control of robot is done in the joints frame of reference. This requires the development of efficient methods to solve the problem of finding the value of joint variables that finally locate the end-effector at the desired position and orientation, which is commonly known in robotics as the problem of inverse kinematics (IK).

Unfortunately, the problem of IK for a 6-DoF fully articulated manipulator is very complex due to the inherent redundancy of the trigonometric functions encountered in the solution. The solution is developed either analytically or numerically, and the inherent redundancy of the solution is treated by imposing constraints on the joints movements or the planned trajectory. Numerical solutions were particularly avoided in the control of manipulators that requires the control processor to be embedded (such as humanoids) due to its relatively associated computational cost.

The existence of the closed-form solution depends on the kinematic structure of the arm. Pieper [2] and Thanhtam [3] showed that the 6-DOF manipulator with a spherical wrist has a closed-form solution. Many researchers have obtained closed-form solutions for inverse kinematics of 6-DOF manipulators including Lee et. al. [4], Kang [5] and others [6-8] for 6-DOF PUMA robots, and Schilling [9] for a 6-DOF Intelledex 660T robot. However, these solutions are ones for industrial manipulators that are different in configuration from the human-like arm shown in Fig. 1 that needs a computationally efficient solver appropriate for running on embedded platforms.

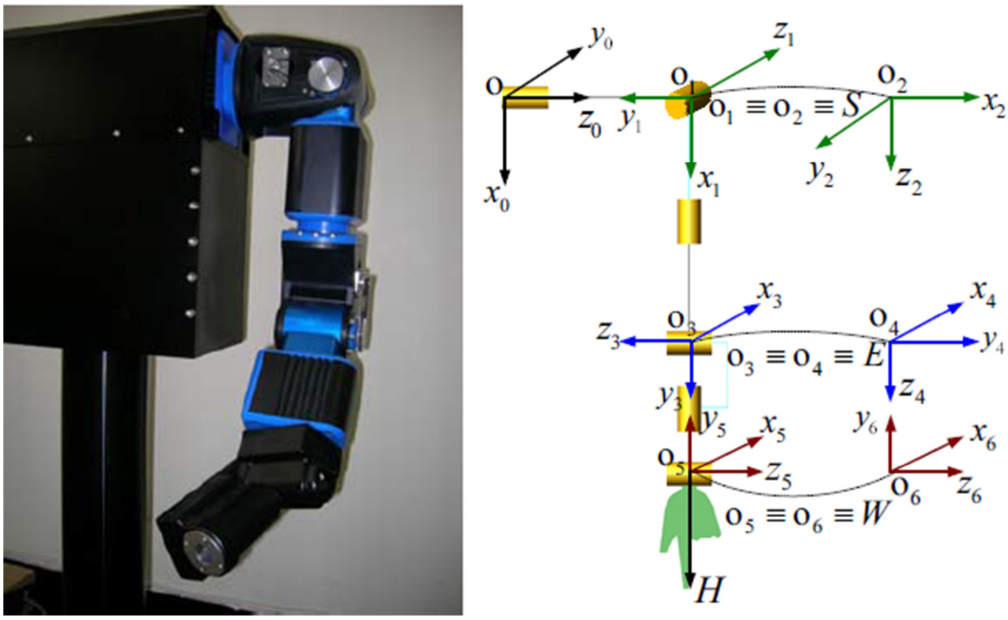


Fig. 1. Picture of the robot arm and its coordinate frames [3].

Ali et. al. [11] presented a closed-form solution for the inverse kinematics of the limbs of the HUBO2+ robot platform. They used a reverse decoupling mechanism method by viewing the kinematic chain of a limb in reverse order and decoupling the position and orientation. O'Flaherty et. al. [10] then used the inverse transform method to compute eight possible solutions for each limb. The correct solution was selected

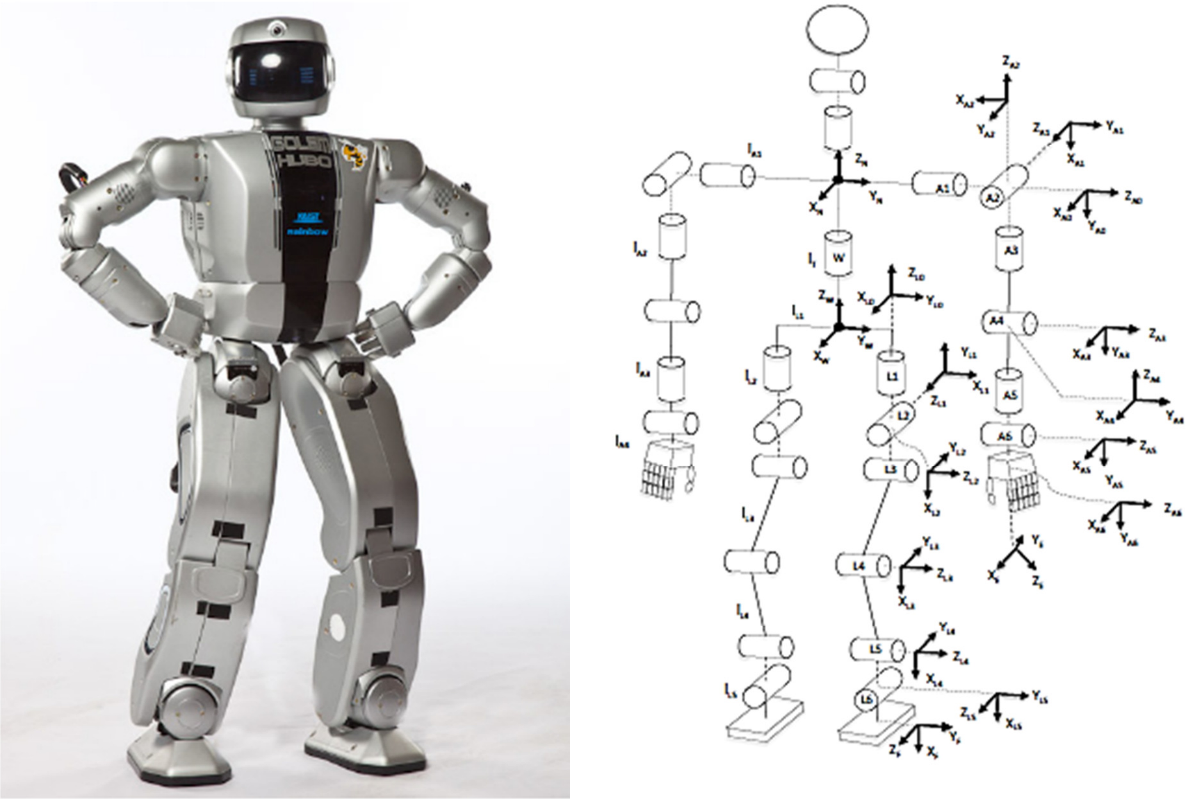


Fig. 2. The HUBO-II humanoid robot with its coordinate frames and joints [10].

based on joint limits and constraints. In working through the solution developed by [11], discrepancies were found in the calculations [10]. A corrected version of the solution was formulated for the IK of the entire four limbs of the HUBO2+ humanoid [10].

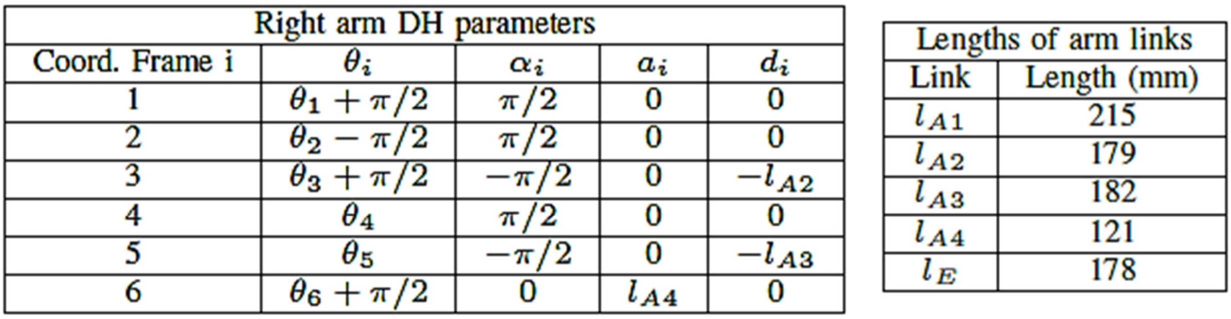
In this paper, the problem of inverse kinematics of the humanoid robot is revisited to present through a full numerical solution at a reasonable computational cost that made it appropriate for real time applications on modern fast embedded controllers. The numerical algorithm is applicable for any 6 DoF full-articulated arm, and was specifically applied to the HUBO-II arm [11]. The presented algorithm can be successfully applied to the entire humanoid robot, where the two arms and legs are each of 6 DoF and fully-articulated, leaving the head and torso, which are almost trivial. The numerical solution of IK was found very efficient in trajectory tracking problems, where the solution at any time step would be a nice initial guess for the next time step, leading to unique solution with few iterations or lower computational cost. This feature is very useful in developing IK-based walking algorithms, especially for this class of humanoids, where a bi-bed mechanism is adopted for the foot [12].

KINEMATICAL MODELING OF THE HUBO-II HUMANOID ROBOT

The HUBO-II humanoid robot, shown in Fig. 2, consists of 27 joints that could be broken up into six parts. These parts consist of the two arms (six joints each), the two legs (six joints each), the torso (1 joint), and the head (2 joints).

The D-H table and link parameters for the arm are given in Table-1. Note that a positive l_{A1} refers to the left arm, while negative l_{A1} refers to the right arm.

Table 1. D-H table for the left arm and the value for link parameters [10].



Right arm DH parameters				
Coord. Frame i	θ_i	α_i	a_i	d_i
1	$\theta_1 + \pi/2$	$\pi/2$	0	0
2	$\theta_2 - \pi/2$	$\pi/2$	0	0
3	$\theta_3 + \pi/2$	$-\pi/2$	0	$-l_{A2}$
4	θ_4	$\pi/2$	0	0
5	θ_5	$-\pi/2$	0	$-l_{A3}$
6	$\theta_6 + \pi/2$	0	l_{A4}	0

Lengths of arm links	
Link	Length (mm)
l_{A1}	215
l_{A2}	179
l_{A3}	182
l_{A4}	121
l_E	178

Solution of the IK Using Full-Numerical Iterative Algorithm

The algorithm will be applied to the inverse kinematics (IK) of the left Arm, which has similar configuration as for the right arm, and also the left & right legs. The arm consists of 6 revolute joints, three for the shoulder that controlling the position of the end-effector (the hand), and the other three joints are for the wrist controlling the orientation of the end-effector.

Solving the IK problem is to find the values of the 6 joint variables, namely: θ_1, θ_2, ……, and θ_6 that corresponding to an arbitrary position & orientation of the end-effector that is given as follows:

$$ {}^0T_6 = \begin{bmatrix} x_6 & y_6 & z_6 & p_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n & s & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

where_${}^0T_6$ is the homogeneous transformation matrix from the hand frame (frame 6) to the first shoulder joint (frame 0), x_6, y_6, *and* z_6 describe the orientation of the hand coordinate frame relative to the shoulder coordinate frame, p_6 is the position vector describing the location of the hand relative to the shoulder, *n, s, a, and p* represent the normal, sliding, approach, and position vectors of the hand, respectively.

The forward kinematics transformation matrix from the *end-effector* to the *Neck* is given by:

$$ {}^NT_E = {}^NT_0\, {}^0T_6\, {}^6T_E $$

where ${}^NT_E$ is the transformation matrix from the end-effector to the neck, ${}^NT_0$ is the transformation matrix from the first shoulder joint to the neck and ${}^6T_E$ is the transformation matrix from the end-effector to the hand frame (frame#6).

$$ {}^N\boldsymbol{T_0} = \begin{bmatrix} 0 & 0 & 1 & l_{A1} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Fortunately, reversing the kinematical chain to start from the hand to the shoulder decouples the wrist degrees of freedom (DoF), namely θ_4, θ_5, and θ_6, from the shoulder DoF, namely θ1, θ2, and θ3. By doing this, the IK problem could be split into two sub-problems, the shoulder & wrist IK problems.

The transformation matrix from the shoulder to the hand is obtained by taking the inverse of both sides of equation (1), which yields:

$$ {}^{6}T_0 = inv({}^{0}T_6) = \begin{bmatrix} x_6' & y_6' & z_6' & p_6' \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Solution of the IK for the Wrist

Multiply both sides of (2) by ${}^{5}T_6$:

$$\Rightarrow \qquad {}^{5}T_6\,{}^{6}T_0 = {}^{5}T_6(\theta_6)\begin{bmatrix} x_6' & y_6' & z_6' & p_6' \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Assume the joint variables vector, θ = [θ1 θ2 θ3 θ4 θ5 θ6]^T

Rewrite equation (3) in the following form:

$$ {}^{5}T_0(\theta) = {}^{5}G_0(\theta_6) \tag{4}$$

where:

$$ {}^{5}G_0(\theta_6) = {}^{5}T_6(\theta_6)\begin{bmatrix} x_6' & y_6' & z_6' & p_6' \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equate the position vector of the frame origin for both sides (the 4th column) of (4) yields to the following:

$$ {}^{5}d_0(\theta_4, \theta_5) = {}^{5}g_0(\theta_6) \tag{5}$$

Approximate both sides of (4) to a first-order approximate backward difference, which yields the following finite-difference equation:

$$ {}^{5}{d_0}^{i-1}(\theta_4, \theta_5) + \begin{bmatrix} \frac{\partial^5 d_0}{\partial\theta_4} & \frac{\partial^5 d_0}{\partial\theta_5} \end{bmatrix}^{i-1} [\Delta\theta_4\ \Delta\theta_5]^T = {}^{5}{g_0}^{i-1}(\theta_6) + \frac{\partial^5 g_0}{\partial\theta_6}^{i-1} \Delta\theta_6 \tag{6}$$

Rearrange (6) to put in the following form:

$$\begin{bmatrix} \frac{\partial^5 d_0}{\partial\theta_4} & \frac{\partial^5 d_0}{\partial\theta_5} & \frac{-\partial^5 g_0}{\partial\theta_6} \end{bmatrix}^{i-1} [\Delta\theta_4\ \Delta\theta_5\ \Delta\theta_6]^T = \frac{\partial^5 g_0}{\partial\theta_6}^{i-1} \Delta\theta_6 - {}^{5}{d_0}^{i-1}(\theta_4, \theta_5) \tag{7}$$

or

$$J_w^{i-1}\, \Delta\theta_w = {\Delta E_w}^{i-1}$$

$$\Rightarrow {\Delta\theta_w}^{i} = \left(J_w^{i-1}\right)^{-1} {\Delta E_w}^{i-1} \tag{8}$$

where:

θw = [θ4 θ5 θ6]^T …….The *wrist* variables.

J_w^{i-1} is the wrist Jacobean, and is defined as follows:

$$J_w^{i-1} = \begin{bmatrix} \frac{\partial^5 d_{0x}}{\partial\theta_4} & \frac{\partial^5 d_{0x}}{\partial\theta_5} & -\frac{\partial^5 g_{0x}}{\partial\theta_6} \\ \frac{\partial^5 d_{0y}}{\partial\theta_4} & \frac{\partial^5 d_{0y}}{\partial\theta_5} & -\frac{\partial^5 g_{0y}}{\partial\theta_6} \\ \frac{\partial^5 d_{0z}}{\partial\theta_4} & \frac{\partial^5 d_{0z}}{\partial\theta_5} & -\frac{\partial^5 g_{0z}}{\partial\theta_6} \end{bmatrix}^{i-1}$$

$$\Delta\theta_w{}^i = [\Delta\theta_4{}^i\ \Delta\theta_5{}^i\ \Delta\theta_6{}^i]^T$$

$$\Delta E_w{}^{i-1} = {}^5g_0{}^{i-1} - {}^5d_0{}^{i-1}$$

And hence,

$$\theta_w{}^i = \theta_w{}^{i-1} + \Delta\theta_w{}^i \qquad (9)$$

All partial derivatives at (*i-1*) are approximated by first-order forward differencing:

$$\frac{\partial(\quad)^{i-1}}{\partial\theta_j} = \frac{[(\quad)^i - (\quad)^{i-1}]}{\Delta\theta_j} + O(\Delta\theta_j)$$

where:
The index (*i*) denotes the iteration step.
The index (*j*) denotes the joint variable number (*j* = 1,2…………,6).

Repeat the above iterative solution until the difference between two successive solutions for wrist variables, $\Delta\theta_w{}^i$, reaches an appropriate value according to the specified accuracy.

Analysis for the selection of (Δθ$_j$)

In a conservative way, the most effective value of $\Delta\theta_j$ in the position accuracy of the end-effector is for j=2 or $\Delta\theta_2$. The accuracy in the end-effector position, P_6, ($\Delta S = [\Delta X_6\ \ \Delta Y_6\ \ \Delta Z_6]^T$) is given by the maximum absolute value, $|\Delta S_{max}|$, which is correlated with the accuracy in the solution of joint variables, $\Delta\theta_2$, as follows:
$(l_2+l_3+l_4)\ \Delta\theta_2 \le |\Delta S_{max}|$, given that θ_3, θ_4, θ_5, and θ_6 are zeros (the extended arm configuration as shown in Fig. 3).

$$\Rightarrow \Delta\theta_2 \le \frac{\Delta S_{max}}{l_2+l_3+l_4}$$

Where l_2, l_3, l_4 are the arm link lengths.

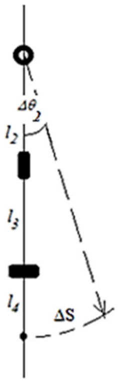


Fig. 3. Extended Arm configuration.

Solution for the IK of the Shoulder

Multiply both sides of (2) by ${}^3T_6 = {}^3T_4{}^4T_5{}^5T_6$:

$$\Rightarrow \qquad {}^3T_6\,{}^6T_0 = {}^3T_6(\theta_w)\begin{bmatrix} x_6{}' & y_6{}' & z_6{}' & p_6{}' \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow \qquad {}^3T_0(\theta_S) = G_3(\theta_w) \tag{10}$$

where

$\theta_S = [\theta_1 \;\; \theta_2 \;\; \theta_3]^T$ …….The *shoulder* variables.

Equate the matrix elements of column #3 in both sides of equation (10), yields:

$$F\left(\theta_S{}^i\right) = H\left(\theta_w{}^i\right) \tag{11}$$

where

$$F = \begin{bmatrix} {}^3T_0(1,3) \\ {}^3T_0(2,3) \\ {}^3T_0(3,3) \end{bmatrix}, \text{ and } H = \begin{bmatrix} G_3(1,3) \\ G_3(2,3) \\ G_3(3,3) \end{bmatrix}$$

<u>Note that:</u>
By solving the wrist equation, the wrist variables ($\theta_w{}^i$) at the current iteration step, *i*, are known and hence the R.H.S of (11), $H(\theta_w{}^i)$, is known too.

$F\left(\theta_S{}^i\right)$ is approximated by a first-order approximate backward difference, which yields the following equation:

$$\frac{\partial F}{\partial \theta_S}^{i-1} \Delta\theta_S{}^i = H\left(\theta_w{}^i\right) - F\left(\theta_S{}^{i-1}\right) \tag{12}$$

Similarly, all partial derivatives at (*i-1*) are approximated by first-order forward differencing.

or

$$J_S^{i-1}\,\Delta\theta_S{}^i = \Delta P_S{}^{i-1}$$

$$\Rightarrow \Delta\theta_S{}^i = \left(J_S^{i-1}\right)^{-1} \Delta P_S{}^{i-1} \tag{13}$$

where

$$J_S^{i-1} = \begin{bmatrix} \dfrac{\partial\, {}^3T_0(1,3)}{\partial \theta_1} & \dfrac{\partial\, {}^3T_0(1,3)}{\partial \theta_2} & \dfrac{\partial\, {}^3T_0(1,3)}{\partial \theta_3} \\ \dfrac{\partial\, {}^3T_0(2,3)}{\partial \theta_1} & \dfrac{\partial\, {}^3T_0(2,3)}{\partial \theta_2} & \dfrac{\partial\, {}^3T_0(3,3)}{\partial \theta_3} \\ \dfrac{\partial\, {}^3T_0(3,3)}{\partial \theta_1} & \dfrac{\partial\, {}^3T_0(3,3)}{\partial \theta_2} & \dfrac{\partial\, {}^3T_0(3,3)}{\partial \theta_3} \end{bmatrix}^{i-1}$$

$$\Delta\theta_S{}^i = [\Delta\theta_1{}^i\;\Delta\theta_2{}^i\;\Delta\theta_3{}^i]^T$$

$$\Delta P_S{}^{i-1} = H\left(\theta_w{}^i\right) - F\left(\theta_S{}^{i-1}\right)$$

And hence,

$$\theta_S{}^i = \theta_S{}^{i-1} + \Delta\theta_S{}^i \tag{14}$$

Repeat the above numerical iteration until the difference between two successive solutions for the shoulder variables, $\Delta\theta_S{}^i$, reaches an appropriate value according to the specified accuracy.

The same algorithm can be applied to the right arm, and for both legs.

For the case of three-dimensional trajectory tracking problems, the presented algorithm was slightly modified by considering the solution for the current time step as an initial guess for the solution at the next time step, which has significantly reduced the number of iterations to convergence, and hence reduced the computational time.

$$q(t^{i+1}) = q(t^i) + \Delta q$$

The use of the presented algorithm in trajectory tracking applications has also the advantage of treating singular points by increasing the time step to pass over the singular point and then interpolate to find out the solution at the singular point.

NUMERICAL RESULTS AND ANALYSIS

The results and analysis presented hereafter in this section are for the left arm of the HUBO-II robot, and of course is applicable for any full-articulated 6-DoF arm. The arm is known to have two cases of singularities [10] as follows:

1- Shoulder singularity: When $\theta_2 = \pi/2$ (for the left arm) or $\theta_2 = -\pi/2$ (for the right arm), joints θ_1 and θ_3 become collinear leading to an infinite number of solutions.

2- Elbow singularity: When $\theta_4 = 0$, joints θ_3 and θ_5 become collinear, leading to an infinite number of solutions.

Five configuration test cases have been developed for the purpose of validation of the numerical algorithm; all of the cases were selected singular-free. A known values for the joint variables were used to find out the end-effector position and orientation by solving the forward kinematics. These end-effector positions and orientations were used to inversely find out the corresponding joint variables using the presented numerical algorithm.

The accuracy of the presented numerical algorithm has been determined by the value of maximum absolute error in joint variables, $\mathrm{Max}(|\theta - \theta_r|)$, where θ_r is the reference value and θ represents the solution. Also the maximum absolute error in end-effector position is used as another accuracy measure. Table 2 depicts the results for the five cases that shows maximum error in the joint variables in the order of (10^{-4} deg) and maximum error in the end-effector position in the order of (10^{-6} m).

The numerical algorithm has been applied to solve for the IK for both an end-effector point in the workspace frame and for a three-dimensional trajectory as well.

The maximum error in joints variables and end-effector position for the case of the three-dimensional space trajectory shown in Fig. 4 of arc length 0.5 [m] is in the order of (10^{-4} deg) and (10^{-6} m) respectively, which lays well within the physical resolution of most commercial joint motors. The variation of the maximum absolute error in joint variables & end-effector position with number of time steps is depicted in Fig. 5, which shows a finite jump in the value of maximum absolute error, especially for the joint variables, at the16th time step. This is due to the trajectory approaches a point of inflection as shown for the end-effector Z-coordinate (Fig. 6) and the first–order accurate finite approximation used in the presented numerical algorithm.

Table 2. Error in Joint Variables & End-effector Position for Different Arm Configurations.

	θ_1 [deg]	θ_2 [deg]	θ_3 [deg]	θ_4 [deg]	θ_5 [deg]	θ_6 [deg]	**Max. Error Max($\lvert\theta - \theta_r\rvert$) [deg]**	**Max. Error in *end-effector* Position [m]**
Case#1	12	18	22	32	43	50	7.4525e-006	5.6147e-008
Case#2	7	11	21	29	37	43	1.1850e-005	8.8958e-008
Case#3	-30	-25	45	-35	20	25	8.0691e-006	1.2025e-008
Case#4	10	45	15	90	5	15	5.5712e-005	1.9712e-007
Case#5	20	30	-20	-30	-10	-20	1.8403e-004	1.1083e-006

Analysis of Computational Cost

The following results have been developed on the following computational platform:
Pentium Dual-Core CPU T4200 @ 2.00 GHz – RAM 3.00 GB
Windows 7.0, 64-bit Operating system.

The computational time is in general depends on the initial guess value, and how much is far from the solution. The average computational time per single point solution is (CPU time = 0.0624 sec).

The presented algorithm was also applied to a three-dimensional space trajectory (Fig. 4), which has yield an average CPU time per time step of 0.0074 sec.

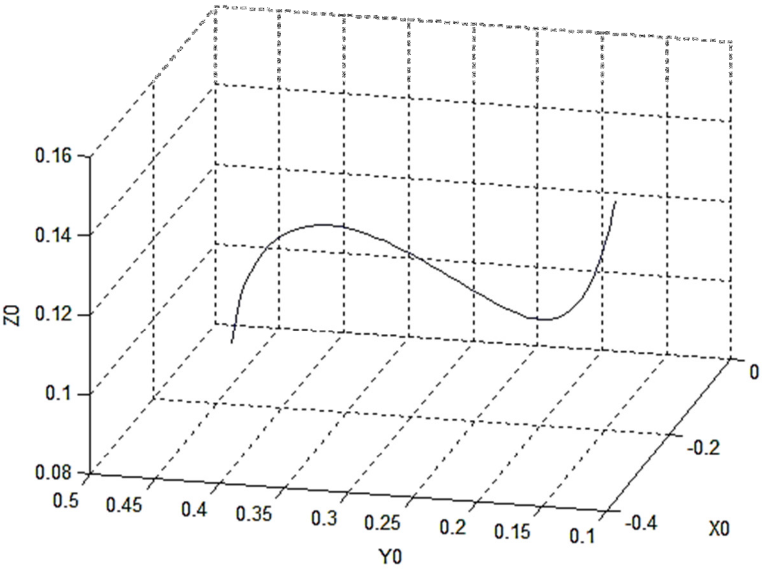


Fig. 4. Three-Dimensional Space Trajectory.

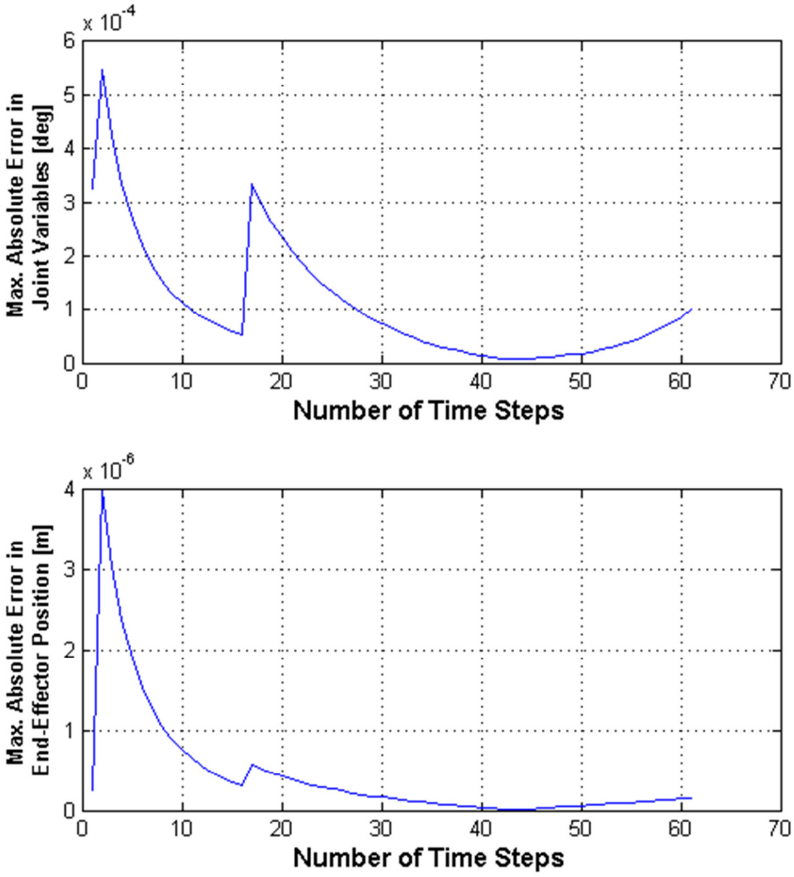


Fig. 5. Variation of the Maximum Absolute Error in Joint Variables & End-Effector Position with Time.

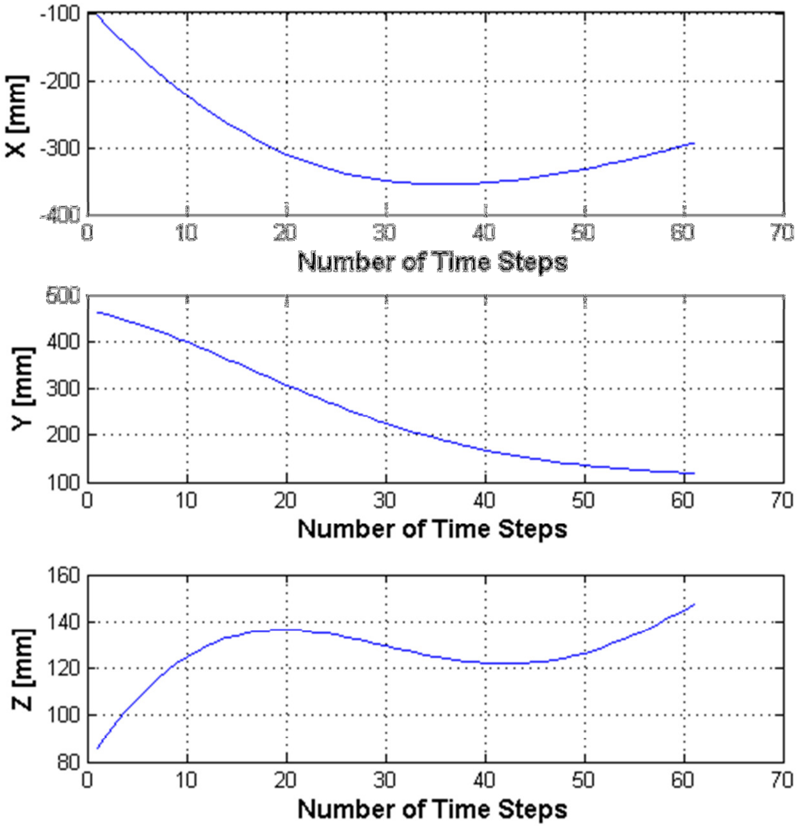


Fig. 6. Variation of the End-Effector Position with Time.

CONCLUSIONS AND RECOMMENDATIONS

- In this paper, a full-numerical algorithm for solving the IK of a fully articulated 6 DoF manipulator is developed and validated. The algorithm utilizes the idea of decoupling the wrist and shoulder DoF and based upon an iterative finite difference scheme.

- The presented algorithm has been applied to solve the IK for two cases: the case of single end-effector position and orientation in the workspace frame, and the case of three-dimensional space trajectory.

- For the case of single end-effector position and orientation in the workspace frame, the algorithm has been applied to five different cases. The results revealed minimum accuracy in the order of (10^{-4} deg) in joint variables, and (10^{-6} m) in the end-effector position. The computational cost for this case has an average CPU-time of 0.0624 sec when run on the computational platform mentioned above.

- The results for a 0.5m-length three-dimensional space trajectory case revealed a minimum accuracy in joints variables and end-effector position of order (10^{-4} deg) and (10^{-6} m) respectively with an average CPU-time per time step of 0.0074 sec.

- The results for both cases have demonstrated an efficient performance for the presented numerical algorithm in terms of joint variables and end-effector position accuracy that were provided at an affordable computational cost.

- The author wishes to employ the presented algorithm to simulate the IK-based walking in the joints space for the HUBO-II humanoid robot in future work.

REFERENCES

[1] Denavit, J. and Hartenberg, R.S., A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices, ASME J. Appl. Mech., vol. 77, (1955).

[2] D. Pieper and B. Roth, “The kinematics of manipulators under computer control”, Proceedings of the 2nd Int. Congress on Theory of Machines and Mechanisms, pp. 159-169, 1969.

[3] Thanhtam H., Chul-Goo K., and Lee S., “Efficient Closed-Form Solution of Inverse Kinematics for a Specific Six-DOF Arm”, International Journal of Control, Automation, and Systems, Vol. 10, No. 3, pp. 567-573, (2012).

[4] Lee C. G. S. and Ziegler M., “Geometric approach in solving the inverse kinematics of PUMA robots”, IEEE Trans. on Aerospace and Electronic Systems, vol. 20, pp. 695-706, (1984).

[5] Kang C. G., “Online trajectory planning for a PUMA robot”, Int. J. of Precision Engineering and Manufacturing, vol. 8, pp. 16-21, (2007).

[6] Elgazzar S., “Efficient kinematic transformations for the PUMA 560 robot”, IEEE J. Robotics and Automations, vol. 1, pp. 142-151, (1985).

[7] Cote J., Osselin C. M. G, and Laurendeau D., “Generalized inverse kinematic functions for the Puma manipulators”, IEEE Trans. on Robotics and Automation, vol. 11, pp. 404-408, (1985).

[8] R. P. Paul and H. Zhang, “Computationally efficient kinematics for manipulators with spherical wrists based on the homogeneous transformation representation”, The International Journal of Robotics Research, vol. 5, pp. 32-44, (1986).

[9] Schilling R. J., Fundamentals of Robotics: Analysis and Control, Prentice Hall, (1990).

[10] O’Flahertyy R., Vieiray P., Greyy M., Ohz P., Bobicky A., Egerstedty M., and Stilman M., “Kinematics & Inverse Kinematics of the Humanoid Robot HUBO2+”, Georgia Institute of technology, Technical Report, GT-GOLEM-2013-001, (2013).

[11] Ali M. A., Park H. A., and Lee C. S. G., “Closed-form Inverse Kinematic Joint Solution for Humanoid Robots”, pp. 704–709, (2010).

[12] Ill-Woo P., Jung-Yup K. and Jun-Ho O., “Online Biped Walking Pattern Generation for Humanoid Robot KHR-3(KAIST Humanoid Robot – 3: HUBO)”, IEEE, (2006).