International Journal of Control, Automation, and Systems (2012) 10(3):567-573 ISSN:1598-6446 eISSN:2005-4092
DOI 10.1007/s12555-012-0313-9 http://www.springer.com/12555

Efficient Closed-Form Solution of Inverse Kinematics for a Specific Six-DOF Arm

Thanhtam Ho, Chul-Goo Kang*, and Sangyoon Lee

Abstract: Inverse kinematics solutions for multi-DOF arms can be classified as analytical or numerical. In general, analytical solutions are preferable to numerical solutions because analytical ones yield complete solutions and are computationally fast and reliable. However, analytical closed-form solutions for inverse kinematics of 6-DOF arms rarely exist for real-time control purposes of fast moving arms. In this paper, we propose a fast inverse kinematics algorithm with a closed-form solution for a specific 6-DOF arm. The proposed algorithm is verified using simulation modules developed by us for demonstrations.

Keywords: Closed-form solution, inverse kinematics, robotic arm, simulation.

1. INTRODUCTION

Inverse kinematics plays a key role in robotics and computer animations. Given the pose (position and orientation) of the end-effector, inverse kinematics problems correspond to computing joint variables for that pose. Inverse kinematics solutions for 6-dergee-of-freedom (DOF) arms may be characterized as analytical or numerical [1]. Analytical solutions can be further subdivided into geometry-based closed-form solutions [2] and algebraic-elimination-based solutions [3]. In general, closed-form solutions can only be obtained for 6 or less than 6-DOF systems with a specific structure. Solutions based on algebraic elimination express joint variables as solutions to a system of multivariable polynomial equations, or express a single joint variable as the solution to a very high degree polynomial and determine the other joint variables using closed-form computations.

In contrast to the analytical solutions, numerical approaches iteratively converge to a solution based on an initial guess [4]. In numerical approaches, computation time to converge may vary, and thus numerical solutions are not appropriate to fast real-time control applications even if they are applicable to ill-posed systems. In general, analytical solutions are preferable to numerical ones for real-time control applications because analytical ones yield all solutions (completeness) and are computationally fast and reliable. In some cases, partly closed-form and partly numerical solutions have been tried [5].

The existence of the closed-form solution depends on the kinematic structure of the arm. Pieper [2] 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*. [6], Kang [7] and others [8-10] for 6-DOF PUMA robots, and Schilling [11] 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. Fast and efficient closed-form solutions for the 6-DOF arm such as Fig. 1 rarely exist for the real-time control purposes of fast moving arms.

Among earlier works on anthropomorphic arms, Tolani *et al*. [1] considered the inverse kinematics of 7-DOF anthropomorphic limbs, and they simplified the arm structure to two segments including the upper arm and the forearm while the shoulder blade is neglected. The simplification makes the inverse kinematics solvable but the solution is not able to apply to many robotic arms where shoulder blades are included. Also, Asfour *et al*. [12] derived a closed-form solution of the inverse kinematics for a 7-DOF arm of a humanoid robot ARMAR. After selecting a specific value of the *z* axis of the elbow, they determined joint angles by matrix equations using the decomposition approach.

This paper presents an efficient (i.e., fast and reliable) closed-form solution of inverse kinematics for a 6-DOF arm similar to the human arm structure. Previously for the commercial arm by Robot and Design Co., Ltd. shown in Fig. 1, a pseudo-inverse numerical solution has been used for inverse kinematics of the arm, and the average converging time was reported to be about 1 ms. This is a big computational burden for real-time control of the fast moving arm, and, consequently, the sampling frequency of the feedback control system becomes quite low to control such a robot arm where high precision and fast response are the most primary requirements. Diffe-

Manuscript received September 4, 2010; revised August 9, 2011; accepted January 4, 2012. Recommended by Editorial Board member Shinsuk Park under the direction of Editor-in-Chief Jae-Bok Song.

This work was supported by Konkuk University in 2010.

Thanhtam Ho and Sangyoon Lee are with the Department of Mechanical Design and Production Engineering, Konkuk University, Gwangjin-gu, Seoul 143-701, Korea (e-mails: thanhtam.h@gmail.com, slee@konkuk.ac.kr).

Chul-Goo Kang is with the Department of Mechanical Engineering, Konkuk University, Gwangjin-gu, Seoul 143-701, Korea (e-mail: cgkang@konkuk.ac.kr).

* Corresponding author.

rently from the work by Asfour *et al*. [12], we obtained joint angles by geometric relationships, and thus our solution is computationally faster than the Asfour's solution.

2. KINEMATIC STRUCTURE

The robotic arm considered in this paper has 6-DOF motion as shown in Fig. 1. Even if the human arm has redundancy in motion, the basic 6-DOF motion of it is similar to the motion of this robot arm. The structure of the arm is simplified to four segments which represent the shoulder blade, the upper arm, the forearm and the hand.

The actual motor positions of the arm are shown as cylinders in Fig. 2, but the coordinate frames and home configuration of the robot arm are chosen as the ones in Fig. 2 to fit the Denavit-Hartenberg (DH) convention, in which the origin o_2 and o_4 are placed at o_1 and o_3 respectively. The chosen home configuration is natural since the shoulder lies in a horizontal line while the upper arm and the forearm lie in a vertical direction as in human

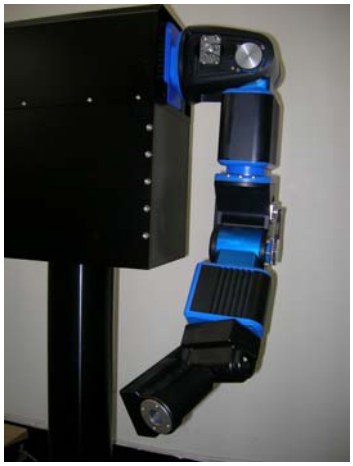


Fig. 1. Picture of the robot arm (Robot and Design Co, Ltd.).

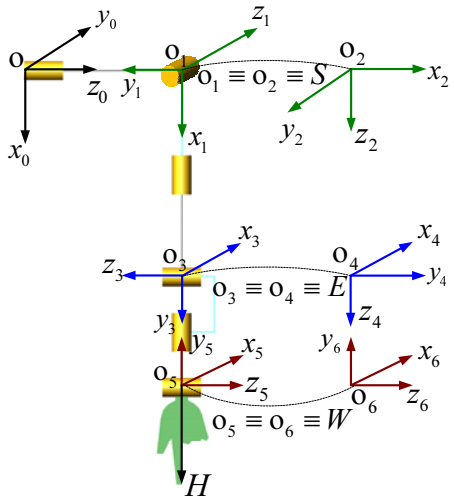


Fig. 2. The coordinate frames and the home configuration.

arms. In the paper, l_1 implies shoulder length, l_2 does upper arm length, and l_3 does forearm length.

The base frame is attached at the origin of the shoulder blade while the last frame, the tool frame $x_6y_6z_6$ of the hand, is attached at the joint between the forearm and the hand, the wrist. The chosen coordinate frames are peculiar in the sense that the origin o_2 of the frame $x_2y_2z_2$ lies at the same point with the origin o_1 (this point is denoted by S), the origin o_4 lies at the same point with the origin o_3 (this point is denoted by E), and the origin o_6 lies at the same point with the origin o_5 (this point is denoted by W).

The forward kinematics problem for this robot arm is solved as usual using the DH parameters and homogeneous transformation matrices A_i for each joint i. Since all six joints are revolute, the forward kinematics solution is given by

$$ {}^0T_6 = A_1A_2A_3A_4A_5A_6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{1} $$

The position of the end-effector (i.e., wrist) is observed from the transformation matrix ${}^0T_6$ to be

$$ {}^0\mathbf{o}_6 = \begin{bmatrix} -l_3\left((c_1c_2c_3 + s_1s_3)s_4 + c_1s_2c_4\right) - l_2c_1s_2 \\ -l_3\left((s_1c_2c_3 - c_1s_3)s_4 + s_1s_2c_4\right) - l_2s_1s_2 \\ l_3(s_2c_3s_4 - c_2c_4) - l_2c_2 + l_1 \end{bmatrix} \tag{2} $$

and the orientation of the hand is observed as a rotation matrix which is the first 3×3 submatrix of ${}^0T_6$. Notice that the notation s_i means $\sin(\theta_i)$ and c_i means $\cos(\theta_i)$. In the paper, lower-case bold face characters represent vectors expressed in component forms, and left superscript notation represents the coordinate frame with which the vector is expressed.

3. CLOSED-FORM SOLUTION

The inverse kinematics problem is equivalent to the problem of solving the system of nonlinear equations obtained from the forward kinematics problem. Unfortunately, the general analytical solution for this problem does not exist.

We present a closed-form solution that is computationally fast and reliable. Compared to the Asfour's work [12], we use different constraint equations to obtain the elbow pose, and also use geometric relationships to obtain θ_5 and θ_6 instead of matrix inversions. The calculation process is composed of two steps. First, the position of the elbow position E is determined by solving the geometric constraints, and then each joint angle is determined using the calculated elbow position E.

Before the elbow position is computed, the position and orientation of the origin o_5 at the wrist are extracted from the given data. Let H be the position of the end point on the hand and ${}^0R_6$ be the orientation of the coordinate frame $x_6y_6z_6$.

From the geometric relationship, the position vector of the wrist expressed in the frame $x_0y_0z_0$ is extracted from the given data to be as in (3).

$$W = \begin{bmatrix} x_W \\ y_W \\ z_W \end{bmatrix} = \begin{bmatrix} x_H \\ y_H \\ z_H \end{bmatrix} + l_4 \, {}^0\mathbf{j}_6, \tag{3}$$

where l_4 is the length of the hand. In (3), ${}^0\mathbf{j}_6$ is the unit vector of the y_6 axis expressed in the base frame. The unit vector ${}^0\mathbf{j}_6$ can be extracted as the second column of the orientation matrix ${}^0R_6$.

$${}^0R_6 = [\,{}^0\mathbf{i}_6 \quad {}^0\mathbf{j}_6 \quad {}^0\mathbf{k}_6\,] \tag{4}$$

The elbow point E is obtained by solving the system of constraint equations. Since the point S is fixed and the lengths of the upper arm and forearm are constants, the point E can be placed on a circle as shown in Fig. 3. Due to the geometric relationships, one easily sees that point E must satisfy the set of constraints in (5).

$$\begin{cases} {}^0\mathbf{k}_6 \perp WE \\ WE = l_3 \\ SE = l_2 \end{cases} \tag{5}$$

Now let's define the position vector E expressed in the base frame, and the unit vector ${}^0\mathbf{k}_6$ as

$$\begin{cases} E = [x \quad y \quad z]^T \\ {}^0\mathbf{k}_6 = [k_x \quad k_y \quad k_z]^T. \end{cases} \tag{6}$$

Then the constraints (5) can be written in terms of coordinates of elbow E.

$$\begin{cases} (x - x_W)k_x + (y - y_W)k_y + (z - z_W)k_z = 0 \\ (x - x_W)^2 + (y - y_W)^2 + (z - z_W)^2 = l_3^2 \\ (x - x_S)^2 + (y - y_S)^2 + (z - z_S)^2 = l_2^2 \end{cases} \tag{7}$$

To solve the simultaneous equations (7) conveniently, we define temporary variables

$$\begin{cases} t_x = x - x_W \\ t_y = y - y_W \\ t_z = z - z_W \end{cases} \tag{8}$$

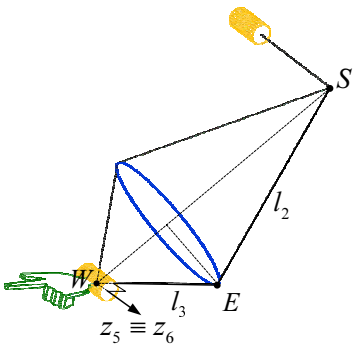


Fig. 3. Geometric constraints of the elbow point E.

and a vector from W to S as

$$\mathbf{r} = [a \quad b \quad c]^T \text{ with } \|\mathbf{r}\|^2 = a^2 + b^2 + c^2 \tag{9}$$

and another temporary variable k_1 as

$$k_1 = \frac{l_2^2 - l_3^2 - \|\mathbf{r}\|^2}{2}. \tag{10}$$

Then one can express the variable t_x and t_y in terms of t_z. from the first and third equation of (7). Here we demonstrate the method to obtain a solution in one case where parameters t_x and a are not equal to zero. The solutions in other cases will be achieved in the same manner.

With the assumption that $a k_y - b k_x \neq 0$, t_y can be written in terms of t_z (in case $a k_y - b k_x = 0$, the algorithm is done in the similar way).

$$t_y = k_2 + k_3 t_z, \tag{11}$$

where

$$k_2 = -\frac{k_x k_1}{(a k_y - b k_x)}, \quad k_3 = -\frac{(a k_z - c k_x)}{(a k_y - b k_x)}. \tag{12}$$

Also t_x can be expressed in terms of t_z.

$$t_x = k_4 + k_5 t_z, \tag{13}$$

where

$$k_4 = -\frac{k_y k_2}{k_x}, \quad k_5 = -\frac{k_y k_3 + k_z}{k_x}. \tag{14}$$

In the next step, if one substitutes t_y in (11) and t_x in (13) into the second equation of (7), a quadratic equation in terms of t_z is obtained.

Let's define

$$\Delta = (k_4 k_5 + k_2 k_3)^2 - (k_5^2 + k_3^2 + 1)(k_4^2 + k_4^2 - l_3^2). \tag{15}$$

Then the solutions of the quadratic equation depend on the value of Δ. In case of $\Delta > 0$, the value of t_z is calculated as in (16).

$$t_{z1}, t_{z2} = \frac{-(k_4 k_5 + k_2 k_3) \pm \sqrt{\Delta}}{(k_5^2 + k_3^2 + 1)} \tag{16}$$

With respect to each value of t_z, the value of t_y and t_x are determined by (11) and (13). Eventually, by using these variables and (8), the position of the elbow $E = [x\ y\ z]^T$ is obtained.

After the position of the elbow is obtained, the rotation angle of each joint in the inverse kinematics problem is calculated gradually by geometric approach. First of all, the rotation angle of the fourth joint, θ_4, is computed independently by the law of cosine.

$$\alpha = \cos^{-1}\left(\frac{l_2^2 + l_3^2 - r^2}{2 l_2 l_3}\right), \tag{17}$$

$$\theta_4 = \pi - \alpha,$$

where

$$r^2 = (x_W - x_S)^2 + (y_W - y_S)^2 + (z_W - z_S)^2.$$

The rotation angles of the first and second joints θ_1 and θ_2 are computed directly from the elbow position.

It is observed that the joint angle θ_2 decides the z coordinate of the elbow while the angle θ_1 relates to the x and y coordinate of this point. Therefore

$$\theta_1 = \text{Atan2}(y_E, x_E) \tag{18}$$

and

$$\theta_2 = \cos^{-1}\left(\frac{l_1 - z_E}{l_2}\right). \tag{19}$$

Since the position of the elbow is not affected by the joint angle θ_3, the position of the wrist should be considered in order to obtain θ_3. It is obvious that the wrist position is decided by the first four joint angles. Since θ_1, θ_2 and θ_4 are known, one can utilize the wrist position to obtain the angle θ_3. Being taken from the result of the forward kinematics, the position of the wrist is the fourth column in the homogenous transformation matrix ${}^0T_5$. It is seen that

$$\theta_3 = \cos^{-1}\left(\frac{z_W - l_1 + l_2 c_2 + l_3 c_2 c_4}{l_3 s_2 s_4}\right) \tag{20}$$

under the assumption that the upper arm and forearm are not in line or $\theta_4 \neq 0$ (i.e., not a singular configuration).

At the home configuration of the robot, the z axes of the frame $x_3y_3z_3$ and the frame $x_5y_5z_5$ are set parallel (see Fig. 2). The misalignment of these axes during the operation of the robot is caused by the joint angle θ_5. This fact is applied for the calculation of the angle θ_5.

$$\theta_5 = \cos^{-1}({}^0\mathbf{k}_3 \bullet {}^0\mathbf{k}_5), \tag{21}$$

where $\bullet$ implies the inner product of two vectors. The unit vector ${}^0\mathbf{k}_5$ is identical to ${}^0\mathbf{k}_6$, and the unit vector ${}^0\mathbf{k}_3$ is extracted from the homogenous transformation matrix ${}^0T_3$ of the frame $x_3y_3z_3$.

Finally, the joint angle θ_6 is calculated by considering the angle between the y-axis of the last frame $x_6y_6z_6$ and the vector $\mathbf{r}$ formed from the wrist to the elbow.

$$\theta_6 = \cos^{-1}\left(\frac{\mathbf{r} \bullet {}^0\mathbf{j}_6}{l_3}\right) \tag{24}$$

The advantage of the above algorithm calculating all six joint angles is that each angle is obtained in only one formulation. The computation time is therefore very short while the result is reliable due to analytical solutions. The computation time of the algorithm in [12] is about 15 microseconds for each loop with C code in Intel Core Quad 2.4 GHz computer, while the computation time of the proposed solution is about 1.5 microseconds for each loop in the same condition. The high computational speed comes mainly from the usage of geometric property instead of inversions of the homogenous transformation matrices in [12].

Among several sets of inverse kinematics solutions, we select a solution at the current instant to be a neighboring solution of the previous one at the previous instant without any jumps.

It is noted that the proposed algorithm for the inverse kinematics computation can be applied for any 6-DOF arm which has a similar kinematics configuration. That is after applying the Denavit-Hartenberg notation to six joints of the arm, two coordinates are set at the shoulder, two at the elbow, and one at the wrist.

4. VALIDATION

Several additional works are performed to verify and demonstrate the proposed algorithm. First, the algorithm is implemented by C++ programming language to verify the computation time and solution accuracy. The test is performed with the Microsoft Windows XP operating system on a personal computer with 2.4Ghz Intel CPU. Only 2 microseconds are needed to implement whole algorithm calculation including the inverse kinematics for the 6-DOF arm, which is much faster than the pseudo-inverse algorithm used previously. This is also faster than Asfour *et al.*'s algorithm [12], and Badler and Tolani's algorithm [13] which was reported to take 50 microseconds on a 200-MHz SGI workstation that corresponds to 4.2 microseconds in the present computer condition.

In order to validate the algorithm, the demonstration software as shown in Fig. 4 is developed in 3D OpenGL environment. The software is planned not only for kinematics validation but also for control purposes of the robot arm. However, it is used in this paper only for kinematics validation. As for the validation of the computation accuracy, a closed-loop process is embedded in the software, as shown in Fig. 5. At the initial state, the position and the orientation of the hand are given. Using the input data, the inverse kinematics computation is done, and the outputs from the state are the rotation angles of arm joints. The forward kinematics takes the joint angles and produces the actual position and orientation of the hand. The actual position and orientation of the hand are compared to the given ones in

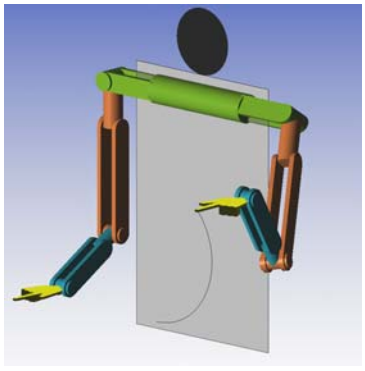


Fig. 4. The robot model in kinematics simulation software.

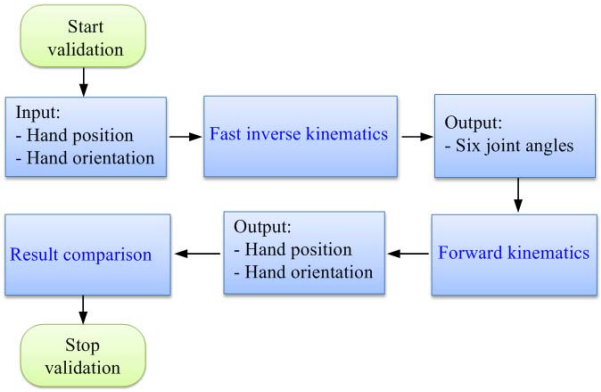


Fig. 5. Validation process for the inverse kinematics algorithm.

Table 1. Coordinates of thee points and position errors.

		Point 1	Point 2	Point 3
Coordinates	x	0	100.89	50.76
	y	-400	-450.45	-330.13
	z	0	300.42	100.74
Position errors	x	2.84e-14	4.26e-14	-7.11e-15
	y	0	-5.68e-14	5.68e-14
	z	-1.42e-14	5.68e-14	-1.42e-14

the first step. The errors can be acquired as a result (refer to Table 1).

In the program of the proposed inverse kinematics, all kinematic parameters including arm lengths, joint angles and hand positions are expressed in the double type with at least 15 digits of precision. In Table 1 below, the position errors of the hand are calculated at three points and are summarized. Note that all results in the table are obtained when the orientation of the hand is set parallel to the vector $r_H = [0 \quad \sqrt{3}/2 \quad 1/2]$. It is found from the validation process and the table that the position error is less than 10^{-13} mm which is due to round-off errors. For the human scale robot arm, it can be actually considered as zero errors. These results are obvious since our solution of the inverse kinematics is an analytic one instead of pseudoinverse or numerical solutions.

The second demonstration module is developed with MSC VisualNastran Desktop and Matlab Simulink software to demonstrate the proposed inverse kinematics solution for motion tracking control of the 6-DOF arm. In this co-simulation work, the dynamic model of the robot is simulated by the VisualNastran Desktop while the controller is implemented in Simulink as shown in Fig. 6. The parameters such as mass and moment of inertia of each link is selected based on the given data from the humanoid arm manufacturer (Robot and Design Co, Ltd).

In the simulation, the inverse kinematics computation is set prior to the system controller. As shown in Fig. 7, the joint angles that are the results of the inverse kinematics are the inputs to the system controller. For applications where the speed and the precision are required, such as path tracking, fast computation of the inverse kinematics takes an important role. Here the path may be divided into many tiny segments and the inverse

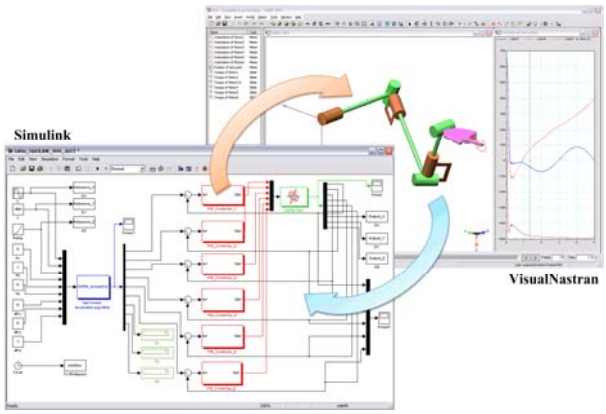


Fig. 6. Simulink and VisualNastran screens.

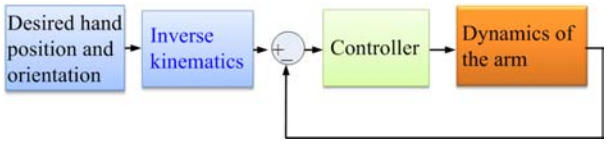
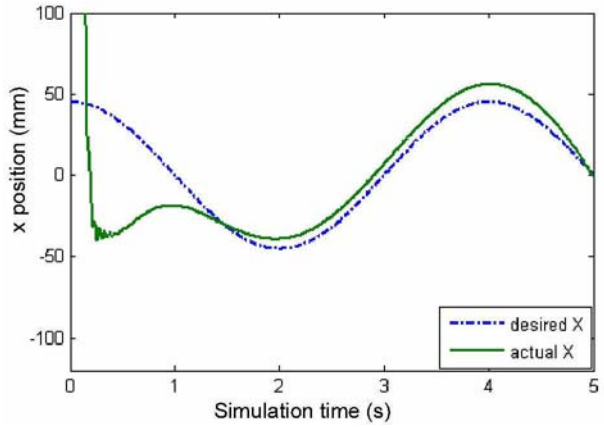


Fig. 7. Schematic diagram of a control system for robot arms.

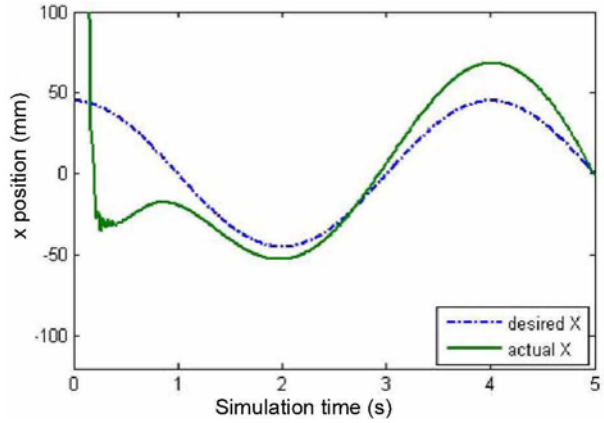
kinematics computation should be done for the distal end of each segment. As the length of the segment gets smaller, the inverse kinematics calculation is required to be faster and more accurate.

We executed a computer simulation of path tracking to demonstrate the inverse kinematics algorithm. In this example, the arm is controlled so that the finger tip can follow a sine curve. The path is divided into many small segments and the inverse kinematics is calculated for the distal end of each segment. Six independent PID controllers are designed for controlling the six motors in the joints. Visual Nastran Desktop (VND) and Matlab Simulink are used for the co-simulation work. The inverse kinematics and the control algorithm are implemented in Matlab Simulink while the robot dynamics is calculated by the VND software.

Fig. 8 displays the simulation results for the two inverse kinematics algorithms that are tested in the same condition. The sampling time for this simulation is set to 0.1 ms. The trajectory's amplitude is 75mm and the frequency is 0.25Hz. Simulation is performed for 5 seconds. Initially, the robot arm is set at the home configuration. Since the initial position and the destination are far apart, at the initial time the actual position of the end point extends far over the desired one. However, after about 1.25 seconds, the robot tracks to the reference stably. In the actual tracking control of the arm, the motion command of each motor can be precalculated from the path planning result of the end-effector during the remaining time of the previous sampling period as a background job, but to see the effect of computational time in this paper, we calculate motion commands (using inverse kinematics) and controller outputs consecutively during the present sampling period, and simulate the dynamics of the robot arm. Fig. 8(a) shows the results using the proposed analytic inverse kinematics algorithm, and Fig. 8(b)



(a) The proposed analytic inverse kinematics algorithm.



(b) The existing slower inverse kinematics algorithm.

Fig. 8. Motion tracking simulation results using.

shows the results using the existing slower pseudoinverse kinematics algorithm. Although the tracking performance mostly depends on the control algorithm, it can also be affected by the inverse kinematics computation as shown in Fig. 8. These demonstrations clarify that the proposed inverse kinematics algorithm is fast and accurate enough to implement in the actual feedback control system of the robot arm at a high sampling frequency.

Recently, another approach to avoid computational load of inverse kinematics has been tried for motion control of a robotic manipulator, which has doubtful practicability for fast and accurate motion control [14].

5. CONCLUSIONS

We presented a fast closed-form solution for the inverse kinematics of a 6-DOF arm with a specific structure similar to the human arm. A closed-form solution is obtained by geometric approach with a novel assignment of coordinate systems and geometric relationships, and is verified by C++ programs. The calculation time of this inverse kinematics algorithm is about 1.5 microseconds under a PC environment with Microsoft Windows XP and a 2.4Ghz Intel CPU, which is faster than the previous algorithms. The proposed inverse kinematics algorithm is computationally fast enough to be applied for real-time control with a high sampling frequency. The limitation of the proposed algorithm is only to be applicable to 6-DOF robots with the specific configuration shown in Fig. 1.

In this paper, furthermore, visualization modules for verifying inverse kinematics of the arm, and for demonstrating the motion tracking control of the arm including arm dynamics are developed. Using these visualization modules, the validity of the proposed inverse kinematic solution (analytic) is demonstrated.

REFERENCES

[1] D. Tolani, A. Goswami, and N. I. Badler, "Real-time inverse kinematics techniques for anthropomorphic limbs," *Graphical Models*, vol. 62, pp. 353-388, 2000.

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

[3] D. Manocha and J. F. Canny, "Efficient inverse kinematics for general 6R manipulators," *IEEE Trans. on Robotics and Automation*, vol. 10 pp. 648-657, 1994.

[4] C. Klein and C. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 7, 868-871, 1997.

[5] M. Benati, P. Morasso, and V. Tagliasco, "The inverse kinematic problem for anthropomorphic manipulator arms," *J. of Dynamics Systems, Measurement, and Control*, vol. 104, pp. 110-113, 1982.

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

[7] C. G. Kang, "Online trajectory planning for a PUMA robot," *Int. J. of Precision Engineering and Manufacturing*, vol. 8, pp. 16-21, 2007.

[8] S. Elgazzar, "Efficient kinematic transformations for the PUMA 560 robot," *IEEE J. Robotics and Automations*, vol. 1, pp. 142-151, 1985.

[9] 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.

[10] J. Cote, C. M. Gosselin, and D. Laurendeau, "Generalized inverse kinematic functions for the Puma manipulators," *IEEE Trans. on Robotics and Automation*, vol. 11, pp. 404-408, 1985.

[11] R. J. Schilling, *Fundamentals of Robotics: Analysis and Control*, Prentice Hall, 1990.

[12] T. Asfour and R. Dillmann, "Human-like motion of a humanoid robot arm based on a closed-form solution of the inverse kinematics problem," *Proc. of Int. Conf. on Intelligent Robots and Systems*, pp. 1407-1412, 2003.

[13] N. I. Badler and D. Tolani, "Real-time inverse kinematics of the human arm," *Presence* (Cambridge,

Mass.), vol. 5, pp. 393-401, 1996.
[14] R. Garcia-Rodriguez and V. Parra-Vega, "Task-space neuro-sliding mode control of robot manipulators under Jacobian uncertainties," *International Journal of Control, Automation, and Systems*, vol. 9, no. 5, pp. 895-904, 2011.

Thanhtam Ho received his B.S. degree from the Mechatronics Department at Hochiminh City University of Technology, Vietnam, in 2005 and his M.S. degree in Mechanical Design and Production Engineering in 2008 from Konkuk University, Seoul, Korea, where he is currently pursuing his Ph.D. degree in Mechanical Engineering. His research interests consist of the humanoid robot arm, biomimetics robots, lateral position control for the roll-to-roll printing system and computational simulation.

Chul-Goo Kang received his B.S. and M.S. degrees in Mechanical Design and Production Engineering from Seoul National University in 1981 and 1985, respectively. He received his Ph.D. degree in Mechanical Engineering from the University of California, Berkeley, in 1989. Currently, he is a professor at Konkuk University in Seoul, Korea. He served as the General Chair of 2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2011). His research interests include intelligent motion and force control, force sensors, train brake systems, and intelligent robots.

Sangyoon Lee received his B.S. degree from Seoul National University in 1993, his M.S. degree from KAIST in 1996, and his Ph.D. degree in Mechanical Engineering from Johns Hopkins University in 2002. Since then, he has been a professor at Konkuk University, Seoul, Korea. His research interests include robotics, automation, and robotics applications to bioengineering.