Robotica (2005) volume 23, pp. 123–129. © 2005 Cambridge University Press
DOI: 10.1017/S0263574704000529 Printed in the United Kingdom

A complete analytical solution to the inverse kinematics of the Pioneer 2 robotic arm

John Q. Gan*, Eimei Oyama†, Eric M. Rosales* and Huosheng Hu*

(Received in Final Form: May 7, 2004)

SUMMARY

For robotic manipulators that are redundant or with high degrees of freedom (*dof*), an analytical solution to the inverse kinematics is very difficult or impossible. Pioneer 2 robotic arm (*P2Arm*) is a recently developed and widely used 5-*dof* manipulator. There is no effective solution to its inverse kinematics to date. This paper presents a first complete analytical solution to the inverse kinematics of the *P2Arm*, which makes it possible to control the arm to any reachable position in an unstructured environment. The strategies developed in this paper could also be useful for solving the inverse kinematics problem of other types of robotic arms.

KEYWORDS: Inverse kinematics; Manipulator control; Modelling and control; Robotic arm.

1. INTRODUCTION

Inverse kinematics modelling has been one of the main problems in robotics research. The most popular method for controlling robotic arms is still based on look-up tables that are usually designed in a manual manner[1–3]. Alternative methods include neural networks[4–8] and optimal search[9], which often encounter problems caused by the fact that the inverse kinematics systems of most robotic arms are multi-valued and discontinuous functions[7] and hardly provide satisfactory solutions to the modelling and control of high-*dof* robotic arms in practice. For robotic manipulators that are redundant or with high *dof*, there are hardly effective solutions to the inverse kinematics problem except for the manually designed look-up table method that is limited to applications with *a priori* known trajectory movements. The *P2Arm* developed by ActivMedia Robotics has been widely used for robotics research, teaching, and development (http://robots.activmedia.com/). However, to date there is no complete analytical inverse kinematics solution or other effective solutions for controlling the *P2Arm*.

In our previous work[10], a hybrid approach was proposed, which combines a partial analytical inverse kinematics model with an optimal search method and provides an almost complete solution to the inverse kinematics of the *P2Arm*. Using a completely different switching mechanism for separating various possible situations, this paper derives a complete analytical inverse kinematics model which is able to control the *P2Arm* to any given position and orientation in its reachable space so that the *P2Arm* gripper mounted on a mobile robot can be controlled to move to any reachable position in an unknown environment. Except for providing a complete solution, the analytical inverse kinematics model is more robust than the one proposed in our previous work[10]. In Section 2, the *P2Arm* inverse kinematics model is derived in an analytical way. Section 3 presents experimental results and discusses the quality of the proposed analytical solution. Conclusions are included in Section 4.

2. DERIVATION OF THE P2ARM KINEMATICS

2.1. Forward kinematics

P2Arm is a 5-*dof* robotic arm with a gripper, as shown in Figure 1. All its joints are revolute. Driven by 6 servomotors, the arm can reach up to 50 cm from the centre of its rotating base to the tip of its closed fingers. The Denavit-Hartenberg (DH) convention and methodology are used in this section to derive its kinematics. The coordinate frame assignment and the DH parameters are depicted in Figure 2 and listed in Table I, respectively, where (x_0, y_0, z_0) to (x_4, y_4, z_4) represent the local coordinate frames at the five joints respectively, (x_5, y_5, z_5) represents the local coordinate frame at the end-effector, α, γ, and θ are the rotation angles about x, y, and z axis respectively. (x_0, y_0, z_0) overlaps with the global coordinate system (X, Y, Z). More details about the definitions of the coordinates and DH parameters can be found in our technical report[11].

Based on the DH convention, the transformation matrix from joint n to joint $n + 1$ is given by:

* Department of Computer Science, University of Essex, Colchester CO4 3SQ (UK) jqgan@essex.ac.uk, emrosa@essex.ac.uk, hhu@essex.ac.uk
(Corresponding author: John Q. Gan).
† Intelligent Systems Institute, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba Science City, Ibaraki, 305-8564 (JAPAN) eimei.oyama@aist.go.jp

$$
{}^{n}A_{n+1} = \begin{bmatrix} \cos\theta_{n+1}\sin\gamma_{n+1} - \sin\theta_{n+1}\sin\alpha_{n+1}\sin\gamma_{n+1} & -\sin\theta_{n+1}\cos\alpha_{n+1} & \cos\theta_{n+1}\sin\gamma_{n+1} + \sin\theta_{n+1}\sin\alpha_{n+1}\cos\gamma_{n+1} & a_{n+1}\cos\theta_{n+1} \\ \sin\theta_{n+1}\cos\gamma_{n+1} + \cos\theta_{n+1}\sin\alpha_{n+1}\sin\gamma_{n+1} & \cos\theta_{n+1}\cos\alpha_{n+1} & \sin\theta_{n+1}\sin\gamma_{n+1} - \cos\theta_{n+1}\sin\alpha_{n+1}\cos\gamma_{n+1} & a_{n+1}\sin\theta_{n+1} \\ -\cos\alpha_{n+1}\cos\gamma_{n+1} & \sin\alpha_{n+1} & \cos\alpha_{n+1}\cos\gamma_{n+1} & d_{n+1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}
$$

The general transformation matrix from the first joint to the last joint of the *P2Arm* can be derived by multiplying all the individual transformation matrices, which is as follows:

$$
\begin{aligned}
{}^{0}T_5 &= {}^{0}A_1 \cdot {}^{1}A_2 \cdot {}^{2}A_3 \cdot {}^{3}A_4 \cdot {}^{4}A_5 \\
&= \begin{bmatrix} -s_1c_4 - c_1s_{23}s_4 & -c_1c_{23}s_5 + s_1s_4c_5 - c_1s_{23}c_4c_5 & c_1c_{23}c_5 + s_1s_4s_5 - c_1s_{23}c_4s_5 & a_5(c_1c_{23}c_5 + s_1s_4s_5 - c_1s_{23}c_4s_5) \\ & & & + c_1(d_4c_{23} + a_2c_2 + a_1) \\ c_1c_4 - s_1s_{23}s_4 & -s_1c_{23}s_5 - c_1s_4c_5 - s_1s_{23}c_4c_5 & s_1c_{23}c_5 - c_1s_4s_5 - s_1s_{23}c_4s_5 & a_5(s_1c_{23}c_5 - c_1s_4s_5 - s_1s_{23}c_4s_5) \\ & & & + s_1(d_4c_{23} + a_2c_2 + a_1) \\ c_{23}s_4 & -s_{23}s_5 + c_{23}c_4c_5 & s_{23}c_5 + c_{23}c_4s_5 & a_5(s_{23}c_5 + c_{23}c_4s_5) + d_4s_{23} + a_2s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{2}
$$

where $s_i = \sin(\theta_i)$, $c_i = \cos(\theta_i)$, $s_{23} = \sin(\theta_2 + \theta_3)$, and $c_{23} = \cos(\theta_2 + \theta_3)$. On the other hand, if the global position and orientation of the end-effector are given, then the general transformation matrix can be represented as follows:

$$
\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\phi\cos\gamma & \cos\phi\sin\gamma\sin\alpha - \sin\phi\cos\alpha & \cos\phi\sin\gamma\cos\alpha + \sin\phi\sin\alpha & p_x \\ \sin\phi\cos\gamma & \sin\phi\sin\gamma\sin\alpha + \cos\phi\cos\alpha & \sin\phi\sin\gamma\cos\alpha - \cos\phi\sin\alpha & p_y \\ -\sin\gamma & \cos\gamma\sin\alpha & \cos\gamma\cos\alpha & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}
$$

where p_x, p_y, and p_z are the global coordinates indicating the spatial position of the end-effector, $[n_x, n_y, n_z]$, $[o_x, o_y, o_z]$, and $[a_x, a_y, a_z]$ represent the global orientation in terms of the DH convention, and α, γ, and ϕ represent the global orientation in terms of the Euler angles ZYX convention. By equalizing the matrices in (2) and (3), the following equations are derived:

$$p_x = a_5(c_1c_{23}c_5 + s_1s_4s_5 - c_1s_{23}c_4s_5) + c_1(d_4c_{23} + a_2c_2 + a_1) \tag{4}$$

$$p_y = a_5(s_1c_{23}c_5 + c_1s_4s_5 - s_1s_{23}c_4s_5) + s_1(d_4c_{23} + a_2c_2 + a_1) \tag{5}$$

$$p_z = a_5(s_{23}c_5 + c_{23}c_4s_5) + d_4s_{23} + a_2s_2 \tag{6}$$

$$n_x = -s_1c_4 - c_1s_{23}s_4 \tag{7}$$

$$n_y = c_1c_4 - s_1s_{23}s_4 \tag{8}$$

$$n_z = c_{23}s_4 \tag{9}$$

$$o_x = -c_1c_{23}s_5 + s_1s_4c_5 - c_1s_{23}c_4c_5 \tag{10}$$

$$o_y = -s_1c_{23}s_5 - c_1s_4c_5 - s_1s_{23}c_4c_5 \tag{11}$$

$$o_z = -s_{23}s_5 + c_{23}c_4c_5 \tag{12}$$

$$a_x = c_1c_{23}c_5 + s_1s_4s_5 - c_1s_{23}c_4s_5 \tag{13}$$

$$a_y = s_1c_{23}c_5 + c_1s_4s_5 - s_1s_{23}c_4s_5 \tag{14}$$

$$a_z = s_{23}c_5 + c_{23}c_4s_5 \tag{15}$$

The Euler angles can be calculated as follows:

$$
if \quad (n_x = 0 \;\&\; n_y = 0) \Rightarrow \begin{cases} \alpha = \text{atan2}(o_x, o_y) \\ \gamma = \pi/2 \\ \phi = 0 \end{cases} \tag{16}
$$

$$
if \quad (n_x \neq 0 \; or \; n_y \neq 0) \Rightarrow \begin{cases} \alpha = \text{atan2}(o_z, a_z) \\ \gamma = \text{atan2}\big(-n_z, \sqrt{n_x^2 + n_y^2}\big) \\ \phi = \text{atan2}(n_y, n_x) \end{cases} \tag{17}
$$

https://doi.org/10.1017/S0263574704000529 Published online by Cambridge University Press

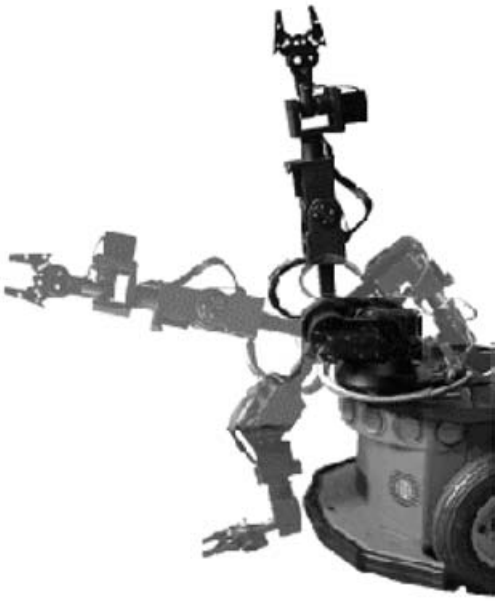

Fig. 1. *P2Arm* and the robot configuration.[12]

From (4)–(17), the position and orientation of the *P2Arm* end-effector can be calculated if all the joint angles are given. This is the solution to the forward kinematics.

2.2. Inverse Kinematics

The forward kinematics equations (4)–(17) are highly nonlinear. It is obvious that the inverse kinematics solution is very difficult to derive. This paper uses various tricky strategies to tackle the challenging inverse problem.

Table I. Denavit-Hartenberg parameters for the *P2Arm*.

Link /Joints	θ	d (cm)	a (cm)	α	γ
1/0-1	θ_1	0	$a_1 = 6.875$	90º	0º
2/1-2	θ_2	0	$a_2 = 16$	0	0º
3/2-3	θ_3	0	0	0	90º
4/2-4	θ_4	$d_4 = 13.775$	0	0	−90º
5/4-endpoint	θ_5	0	$a_5 = 11.321$	0	90º

From (4) and (13) we derive the following equation:

$$p_x - a_5 a_x = c_1(d_4 c_{23} + a_2 c_2 + a_1) \tag{18}$$

Manipulating (5) and (14) in a similar way results in the following equation:

$$p_y - a_5 a_y = s_1(d_4 c_{23} + a_2 c_2 + a_1) \tag{19}$$

It should be noted that θ_2 and θ_3 in the *P2Arm* only take integral values in a limited range. It has been proved by checking all the possible joint angles θ_2 and θ_3 that $(d_4 c_{23} + a_2 c_2 + a_1) \neq 0$ holds, which means that $p_x - a_5 a_x$ and $p_y - a_5 a_y$ will not equal zero at the same time. Now consider two possible situations. If $(d_4 c_{23} + a_2 c_2 + a_1) > 0$, the solution for θ_1 is as follows:

$$\theta_1 = \text{atan2}(p_y - a_5 a_y, p_x - a_5 a_x) \tag{20}$$

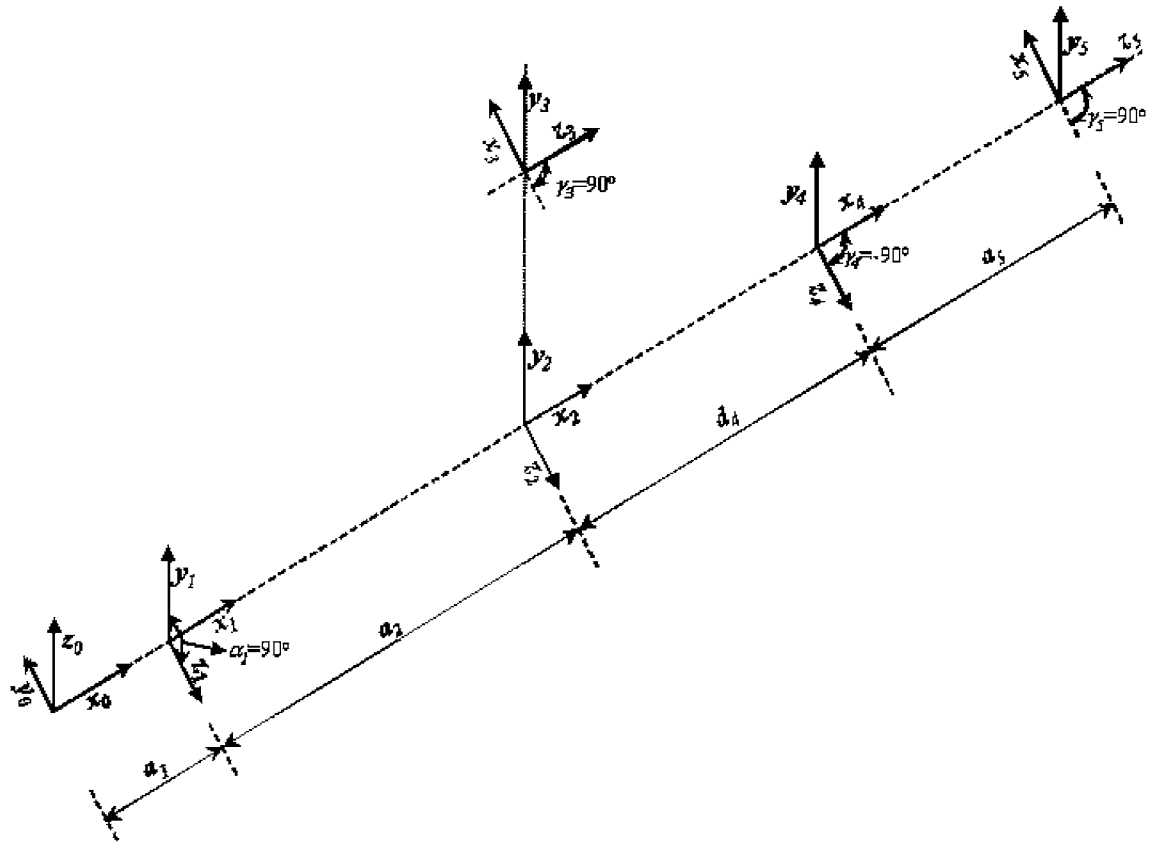

Fig. 2. Coordinate frame assignment.

https://doi.org/10.1017/S0263574704000529 Published online by Cambridge University Press

Otherwise, we have

$$\theta_1 = \text{atan2}(a_5 a_y - p_y, a_5 a_x - p_x) \tag{21}$$

For deriving solutions for θ_2 and θ_3, (18) and (19) can be represented as follows:

$$d_4 c_{23} + a_2 c_2 = (p_x - a_5 a_x)/c_1 - a_1 \tag{22}$$
$$d_4 c_{23} + a_2 c_2 = (p_y - a_5 a_y)/s_1 - a_1 \tag{23}$$

From (6) and (15) we can derive the following equation:

$$d_4 s_{23} + a_2 s_2 = p_z - a_5 a_z \tag{24}$$

It should be noted that (22) and (24) or (23) and (24) describe the forward kinematics of a 2-*dof* robotic arm, which has analytical inverse solutions. To avoid division by zero, let $r = (p_x - a_5 a_x)/c_1 - a_1$ if s_1 is very small or $r = (p_y - a_5 a_y)/s_1 - a_1$ if c_1 is very small, and let $r_z = p_z - a_5 a_z$. One set of possible solutions for θ_2 and θ_3 are as follows:

$$\theta_2 = \text{atan2}(r_z, r) - \text{acos}\left(\frac{r^2 + r_z^2 + a_2^2 - d_4^2}{2a_2\sqrt{r^2 + r_z^2}}\right) + 2m_1\pi \tag{25}$$

$$\theta_3 = \pi - \text{acos}\left(\frac{a_2^2 + d_4^2 - r^2 - r_z^2}{2a_2 d_4}\right) \tag{26}$$

where $m_1 = -1, 0$, or 1, making $-\pi \le \theta_2 \le \pi$. Another set of possible solutions for θ_2 and θ_3 are as follows:

$$\theta_2 = \text{atan2}(r_z, r) + \text{acos}\left(\frac{r^2 + r_z^2 + a_2^2 - d_4^2}{2a_2\sqrt{r^2 + r_z^2}}\right) + 2m_1\pi \tag{27}$$

$$\theta_3 = -\pi + \text{acos}\left(\frac{a_2^2 + d_4^2 - r^2 - r_z^2}{2a_2 d_4}\right) \tag{28}$$

The derivation of the above solutions has been given in Appendix A. It should be noted that $r \neq 0$, which means the above equations always provide certain solutions for θ_2 and θ_3.

Now that θ_1, θ_2 and θ_3 are known, the solutions for θ_4 and θ_5 can be found by using the remaining forward kinematics equations. It is important to avoid division by zero or the two variables in atan2 function being zero at the same time.

Consider the case when $c_{23} \neq 0$ first. From (9) we have

$$s_4 = n_z/c_{23} \tag{29}$$

From (7) and (8) we obtain

$$c_4 = -(n_x + n_z c_1 s_{23}/c_{23})/s_1 \tag{30}$$

or if s_1 is too small,

$$c_4 = (n_y + n_z s_1 s_{23}/c_{23})/c_1 \tag{31}$$

The solution for θ_4 is

$$\theta_4 = \text{atan2}(n_z/c_{23}, -(n_x + n_z c_1 s_{23}/c_{23})/s_1) \tag{32}$$

or if s_1 is too small,

$$\theta_4 = \text{atan2}(n_z/c_{23}, (n_y + n_z s_1 s_{23}/c_{23})/c_1) \tag{33}$$

Because n_x, n_y, and n_z will not equal zero at the same time, which means s_4 and c_4 will not equal zero at the same time, thus the solution here for θ_4 is certain. Knowing $\theta_1, \theta_2, \theta_3$ and θ_4, it is easy to get a solution for θ_5. If $c_4 \neq 0$ or $s_{23} \neq 0$, from (12) and (15) we can derive the following equations:

$$s_5 = \frac{a_z c_{23} c_4 - o_z s_{23}}{c_{23}^2 c_4^2 + s_{23}^2} \tag{34}$$

$$c_5 = \frac{o_z c_{23} c_4 + a_z s_{23}}{c_{23}^2 c_4^2 + s_{23}^2} \tag{35}$$

$$\theta_5 = \text{atan2}(a_z c_{23} c_4 - o_z s_{23}, o_z c_{23} c_4 + a_z s_{23}) \tag{36}$$

If $c_4 = 0$ and $s_{23} = 0$, from (10) and (11) we obtain another solution for θ_5, which is as follows:

$$s_5 = -(o_x c_1 + o_y s_1)/c_{23} \tag{37}$$

$$c_5 = (o_x s_1 - o_y c_1)/s_4 \tag{38}$$

$$\theta_5 = \text{atan2}(-(o_x c_1 + o_y s_1)/c_{23}, (o_x s_1 - o_y c_1)/s_4) \tag{39}$$

Now consider the solutions for θ_4 and θ_5 in the case when $c_{23} = 0$. From (12) and (15) we obtain the solution for θ_5 as follows:

$$\theta_5 = \text{atan2}(-o_z/s_{23}, a_z/s_{23}) \tag{40}$$

From (7) and (8) we have

$$s_4 = -(n_x c_1 - n_y s_1)/s_{23} \tag{41}$$

$$c_4 = -n_x s_1 + n_y c_1 \tag{42}$$

$$\theta_4 = \text{atan2}(-(n_x c_1 - n_y s_1)/s_{23}, -n_x s_1 + n_y c_1) \tag{43}$$

It is difficult to prove that the two variables in the above atan2 functions for calculating θ_4 and θ_5 will not equal zero at the same time. Some alternative solutions can be derived to replace those that have the above problem. For instance, after θ_5 is calculated, (41) can be replaced by:

$$s_4 = \begin{cases} (a_x s_1 - a_y c_1)/s_5 & \textit{if } c_5 \textit{ is too small} \\ (o_x s_1 - o_y c_1)/c_5 & \textit{if } s_5 \textit{ is too small} \end{cases} \tag{44}$$

The above derivation, with various conditions being taken into account, provides a complete analytical solution to the inverse kinematics of the *P2Arm*. It should be noted that we do not know which solution for θ_1 is true, (20) or (21), until we obtain all the joint angles and check if they provide a correct solution using the forward kinematics. Similarly, there are two sets of possible solutions for θ_2 and θ_3, (25)–(26) or (27)–(28), and we do not know which set is true until we check using the forward kinematics equations. All the other conditions used for solving θ_4 and θ_5 can be checked at once when necessary. Therefore, four sets of possible solutions to the inverse kinematics of the *P2Arm* have been derived. Our strategy for choosing the correct solution is to calculate all the four sets of possible solutions (joint angles), which generate four possible corresponding positions and orientations using the forward kinematics. By

https://doi.org/10.1017/S0263574704000529 Published online by Cambridge University Press

comparing the errors between these four generated positions and orientations and the given position and orientation, one set of joint angles, which produces the minimum error, is chosen as the correct solution.

3. EXPERIMENTAL RESULTS AND DISCUSSIONS

Theoretically, the equations for calculating joint angles θ_1 to θ_5 are correct. However, in practice there could be problems in atan2 and acos calculation. For instance, the absolute value of the variable in acos could be slightly greater than 1 due to computing inaccuracy, although it should not happen according to the derivation process in Appendix A. It will not happen that the two variables in the atan2 functions for calculating θ_1 to θ_3 equal zero at the same time. However, it is difficult to theoretically prove the same for equations of calculating θ_4 and θ_5. In this section, experiments were carried out to demonstrate the quality of the derived analytical inverse kinematics model.

In the first batch of experiments, the testing position and orientation data were generated using the forward kinematics model with random joint angles that are within physically limited ranges so that they are guaranteed to be reachable. The experiments were conducted using both the simulated arm model and the real *P2Arm*. We set the allowed error for position as $e_{position} = 0.1$ cm and for orientation as $e_{orientation} = 1^\circ$. No error larger than the error threshold was produced at all with the program kept running on the simulated arm for one week using over 500 million random testing positions and orientations. Because we do not have equipment to measure directly and precisely the position and orientation of the end-effector of real *P2Arm*, what was done was to read out the joint angle values from the *P2Arm* and then calculate the position and orientation using the forward kinematics model. This treatment will rule out mechanical error and control error, which is reasonable because the main purpose of the experiments in this paper is to examine whether the inverse kinematics model is correct, rather than to evaluate the whole robotic arm system. Again, no error larger than the error threshold was produced with the program running on the real *P2Arm*.

The second batch of experiments was similar to the first batch except that the testing positions and orientations were disturbed with small random values. That is, the target positions and orientations could not be exactly reached (The working space of the *P2Arm* is discontinuous). This is for testing the robustness of the derived inverse kinematics model. In this batch of experiments, there were less than 1% of the inverse solutions that did not satisfy the error criterion: $e_{position} = 1$ cm or $e_{orientation} = 3^\circ$ (The orientation is more difficult to satisfy than the position). This is still a very satisfactory result, and in case an error exists (this happens less than once out of 100 trials), the optimal search method developed in reference [10] can always be used to find a better solution.

In order to illustrate how the inverse kinematics model works in controlling the real *P2Arm* more intuitively, a trajectory following task was performed, in which positions on a circle trajectory were used as the input to the inverse kinematics model. It should be noted that these positions could not be exactly reachable. Therefore this experiment also tests the robustness of the inverse kinematics model. Figure 3 shows the trajectory which is a circle in the XY plane with Z = 15 cm. The target positions sent to the inverse

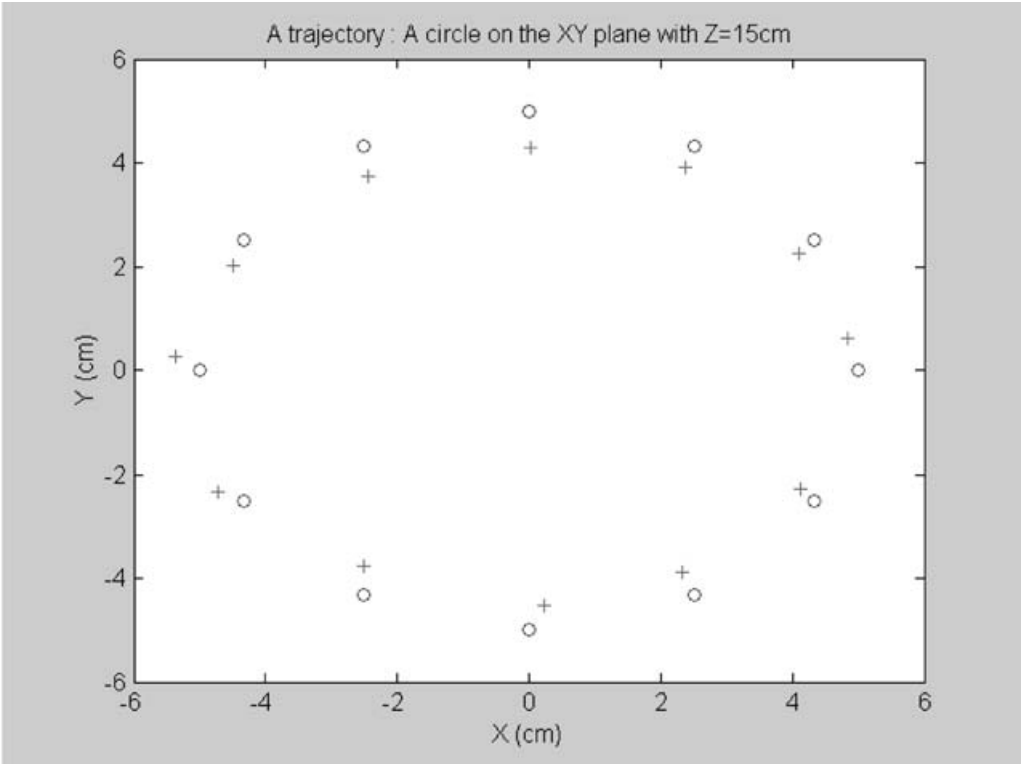

Fig. 3. Trajectory following: '○' represents planned positions on a circle with Z = 15 cm, '+' represents reached positions by the robotic arm end-effector.

https://doi.org/10.1017/S0263574704000529 Published online by Cambridge University Press

Table II. Coordinates of positions on the trajectory.

X	Y	Z	X′	Y′	Z′
5.00	0.00	15.00	4.84	0.60	15.19
4.33	2.50	15.00	4.11	2.26	14.89
2.50	4.33	15.00	2.38	3.91	15.08
0.00	5.00	15.00	0.02	4.28	15.10
−2.50	4.33	15.00	−2.43	3.75	14.86
−4.33	2.50	15.00	−4.50	2.02	14.97
−5.00	0.00	15.00	−5.36	0.27	15.30
−4.33	−2.50	15.00	−4.72	−2.33	14.98
−2.50	−4.33	15.00	−2.50	−3.78	15.10
0.00	−5.00	15.00	0.24	−4.54	14.98
2.50	−4.33	15.00	2.33	−3.88	14.97
4.33	−2.50	15.00	4.12	−2.29	15.09

kinematics model are represented by 'o' and the reached positions using the inverse kinematics model control are represented by '+'. The coordinates of the target positions (X, Y, Z) and reached positions (X′, Y′, Z′) are given in Table II.

The above experimental results show that if the target position is exactly reachable the inverse kinematics model is able to produce very accurate result, as shown in the first batch of experiments, and that if the target position is not exactly reachable the inverse kinematics model is able to provide good approximate solutions.

4. CONCLUSIONS

A complete analytical solution to the inverse kinematics of a widely used robotic arm, *P2Arm*, is derived for the first time in this paper. The derived analytical inverse kinematics model always provides correct joint angles for moving the arm end-effector to any given reachable positions and orientations. If the given positions/orientations cannot be exactly reached, the model is able to give very good approximate solutions for over 99% situations. We believe that the solution developed in this paper will make the *P2Arm* more useful in applications with unpredictable trajectory movements in unknown environments. Without this solution, the trajectory movements of the *P2Arm* would have to be done by manually making the arm follow the trajectory and recording a sequence of joint angles for the later use in the trajectory following task. The analytical solution is able to provide joint angles automatically for a given trajectory. The methods used for deriving the inverse kinematics model for the *P2Arm* could be applied to other types of robotic arms, such as the EduBots developed by the Robotica Ltd[13]. Our software for the *P2Arm* control based on the derived inverse kinematics model will be made available to the public after this paper is published.

References

1. R. D. Klafter, T. A. Chmielewski and M. Negin, *Robotic Engineering: An Integrated Approach* (Prentice Hall, 1989).
2. P. J. McKerrow, *Introduction to Robotics* (Addison-Wesley, 1991).
3. S. B. Niku, *Introduction to Robotics: Analysis, Systems, Applications* (Prentice Hall, 2001).
4. D. Demers and K. Kreutz-Delgado, "Learning global properties of nonredundant kinematic mappings," *Int. J. of Robotics Research* **17**, 547–560 (1998).
5. S. Tejomurtula and S. Kak, "Inverse kinematics in robotics using neural networks," *Information Sciences* **116**, 147–164 (1999).
6. E. Oyama, A. Agah, K. F. MacDprman, T. Maeda and S. Tachi, "A modular neural network achitecture for inverse kinematics model learning," *Neurocomputing* **38–40**, 797–805 (2001).
7. E. Oyama, N. Y. Chong, A. Agah, T. Maeda and S. Tachi, "Inverse kinematics learning by modular architecture neural networks with performances prediction networks," *Proc. IEEE Int. Conf. on Robotics and Automation* (2001) pp. 1006–1012.
8. Y. Li and S. H. Leong, "Kinematics control of redundant manipulators using CMAC neural network," *Proc. 5th World Multiconference on Syst. Cybern. & Inform. (SCI2001)*, (2001) pp. 274–279.
9. B. Schäfer, R. Krenn and B. Rebele, "On inverse kinematics and kinetics of redundant space manipulator simulation," *J. of Computational and Applied Mechanics*, **4**, 53–70 (2003).
10. E. M. Rosales, J. Q. Gan, H. Hu and E. Oyama, "A hybrid approach to inverse kinematics modeling and control of Pioneer 2 robotic arms," *Technical Report* (University of Essex, October 2003).
11. E. M. Rosales and J. Q. Gan, "Forward and Inverse Kinematics Models for a 5-dof Pionner 2 Robotic Arm," *Technical Report* (University of Essex, September 2003).
12. http://www.activrobots.com/ACCESSORIES/pioneerarm.html.
13. http://www.robotica.co.uk/robotica/index.htm.

APPENDIX A. DERIVATION OF SOLUTIONS FOR θ_2 AND θ_3

Equations (22) and (24) or (23) and (24) can be represented as

$$d_4 c_{23} + a_2 c_2 = r \tag{A1}$$

$$d_4 s_{23} + a_2 s_2 = r_z \tag{A2}$$

From (A1)–(A2), the following equation can be derived:

$$d_4^2 + 2a_2 d_4 (c_2 c_{23} + s_2 s_{23}) + a_2^2 = r^2 + r_z^2 \tag{A3}$$

From trigonometry we have

$$\begin{aligned} c_2 c_{23} + s_2 s_{23} &= \cos(\theta_3) = \cos(-\theta_3) \\ &= -\cos(\pi - \theta_3) = -\cos(\theta_3 - \pi) \end{aligned} \tag{A4}$$

Therefore there are several possible solutions for θ_3, which are as follows:

$$\theta_3 = \pm \text{acos}\left(\frac{r^2 + r_z^2 - a_2^2 - d_4^2}{2a_2 d_4}\right) \tag{A5}$$

or

$$\theta_3 = \pm\left[\pi - \text{acos}\left(\frac{a_2^2 + d_4^2 - r^2 - r_z^2}{2a_2 d_4}\right)\right] \tag{A6}$$

Now consider possible solutions for θ_2. Expanding c_{23} and s_{23} in (A1) and (A2) generates the following equations:

$$(d_4 c_3 + a_2) c_2 - (d_4 s_3) s_2 = r \tag{A7}$$

$$(d_4 s_3) c_2 + (d_4 c_3 + a_2) s_2 = r_z \tag{A8}$$

https://doi.org/10.1017/S0263574704000529 Published online by Cambridge University Press

Dividing both sides of (A7) and (A8) by $\sqrt{r^2+r_z^2}$, we have

$$\frac{(d_4c_3+a_2)}{\sqrt{r^2+r_z^2}}c_2-\frac{(d_4s_3)}{\sqrt{r^2+r_z^2}}s_2=\frac{r}{\sqrt{r^2+r_z^2}} \tag{A9}$$

$$\frac{(d_4s_3)}{\sqrt{r^2+r_z^2}}c_2+\frac{(d_4c_3+a_2)}{\sqrt{r^2+r_z^2}}s_2=\frac{r_z}{\sqrt{r^2+r_z^2}} \tag{A10}$$

Assuming $\sin(\theta)=(d_4s_3)/\sqrt{r^2+r_z^2}$ and $\cos(\theta)=(d_4c_3+a_2)/\sqrt{r^2+r_z^2}$, which is acceptable as this assumption leads to $\sin^2(\theta)+\cos^2(\theta)=1$, we transfer (A9) and (A10) into the following:

$$\cos(\theta+\theta_2)=\frac{r}{\sqrt{r^2+r_z^2}} \tag{A11}$$

$$\sin(\theta+\theta_2)=\frac{r_z}{\sqrt{r^2+r_z^2}} \tag{A12}$$

Therefore,

$$\theta+\theta_2=\text{atan2}(r_z,r)+2m_1\pi \tag{A13}$$

$$\begin{aligned}\theta&=\pm\text{acos}\left(\frac{d_4c_3+a_2}{\sqrt{r^2+r_z^2}}\right)\\&=\pm\text{acos}\left(\frac{r^2+r_z^2+a_2^2-d_4^2}{2a_2\sqrt{r^2+r_z^2}}\right)\end{aligned} \tag{A14}$$

where $m_1=-1$, 0, or 1. It is clear that θ could be in $[0,\pi]$ or $(-\pi,0)$. The range of θ will depend on the range of θ_3. If $0\le\theta_3\le\pi$, then $s_3>0$ and $\sin(\theta)>0$, thus $0\le\theta\le\pi$. If $-\pi<\theta_3<0$, then $s_3<0$ and $\sin(\theta)<0$, thus $-\pi<\theta<0$. Therefore, if

$$\theta_3=\pi-\text{acos}\left(\frac{a_2^2+d_4^2-r^2-r_z^2}{2a_2d_4}\right) \tag{A15}$$

i.e., $0\le\theta_3\le\pi$ and $0\le\theta\le\pi$, then

$$\theta_2=\text{atan2}(r_z,r)-\text{acos}\left(\frac{r^2+r_z^2+a_2^2-d_4^2}{2a_2\sqrt{r^2+r_z^2}}\right)+2m_1\pi \tag{A16}$$

Otherwise, if

$$\theta_3=-\pi+\text{acos}\left(\frac{a_2^2+d_4^2-r^2-r_z^2}{2a_2d_4}\right) \tag{A17}$$

i.e., $-\pi<\theta_3<0$ and $-\pi<\theta<0$, then

$$\theta_2=\text{atan2}(r_z,r)+\text{acos}\left(\frac{r^2+r_z^2+a_2^2-d_4^2}{2a_2\sqrt{r^2+r_z^2}}\right)+2m_1\pi \tag{A18}$$

https://doi.org/10.1017/S0263574704000529 Published online by Cambridge University Press