UNIVERSITY OF ESSEX

Department of Computer Science

TECHNICAL REPORT
CSM-412

A Hybrid Appproach to Inverse Kinematics Modeling and Control of Pioneer 2 Robotic Arms

Eric M. Rosales, John Q. Gan, Huosheng Hu
Department of Computer Science
University of Essex
Colchester CO4 3SQ, UK
emrosa@essex.ac.uk, jqgan@essex.ac.uk, hhu@essex.ac.uk

Eimei Oyama
Intelligent Systems Institute
National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba Science City, Ibaraki, 305-8564 JAPAN
eimei.oyama@aist.go.jp

Technical Report No CSM-414

Abstract — **For robotic manipulators that are redundant or with high degrees of freedom, an analytical solution to the inverse kinematics is very difficult or impossible. As alternative approaches, neural networks and optimal search methods have been widely used for inverse kinematics modeling and control in robotics. This paper presents a first analytical solution to the inverse kinematics of a widely used robotic arm (Pioneer 2 robotic arm), which, combined with an optimal search method, provides an effective solution to the modeling and control of the Pioneer 2 robotic arm.**

Keywords: Inverse kinematics, manipulator control, modeling and control, optimization, robotic arm.

I. INTRODUCTION

Inverse kinematics modeling 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]-[11] and optimal search [12], which often encounter problems caused by the fact that the inverse kinematics systems of most robotic arms are multi-valued and discontinuous functions [9]. For robotic manipulators that are redundant or with high degrees of freedom (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 Pioneer 2 robotic arm (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 analytical inverse kinematics solution for the P2Arm.

This paper derives an almost complete analytical inverse kinematics model which, combined with an optimal search method, is able to control a 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. In Section II, the P2Arm inverse kinematics model is derived in an analytical way. Section III presents an optimal search method as a complementary approach for the P2Arm inverse kinematics control. Section IV proposes a hybrid approach that combines the analytical inverse kinematics model with an optimal search method for inverse kinematics modeling and control. Experimental results with discussions are given in Section V and conclusions are included in Section VI.

II. DERIVATION OF THE P2ARM KINEMATICS

A. *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 center of its rotating base to the tip of its closed fingers. The Denavit-Hartenberg (DH) convention and methodology [1]-[3] 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 1, respectively. Details about the definitions of the coordinates and DH parameters can be found in our technical report [13].

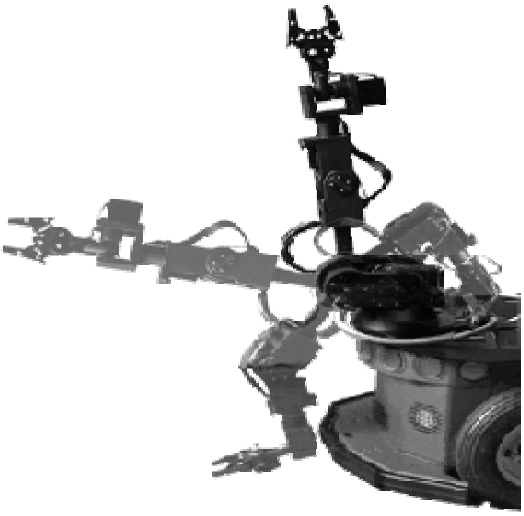

Figure 1 P2Arm and the robot configuration [14]

TABLE I. DENAVIT-HARTENBERG PARAMETERS FOR THE P2ARM

Link / Joints	θ	d (cm)	a (cm)	α	γ
1 / 0-1	$\theta 1$	0	a1=6.875	90^o	0^o

2 / 1-2	θ2	0	a2=16	0	0 °
3 / 2-3	θ3	0	0	0	90 °
4 / 2-4	θ4	d4=13.775	0	0	-90 °
5 / 4-endpoint	θ5	0	a5=11.321	0	90 °

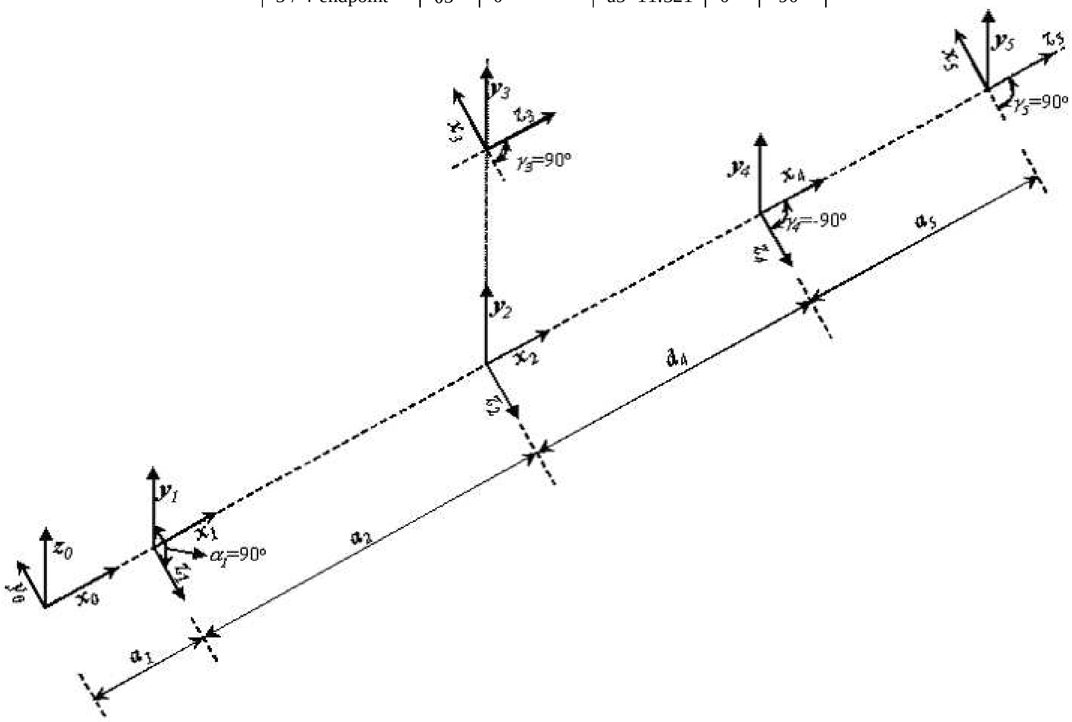

Figure 2 Coordinate frame assignment

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

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

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:

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

where s_i=sin(θ_i), c_i=cos(θ_i), s_{23}=sin($\theta_2+\theta_3$), and c_{23}=cos($\theta_2+\theta_3$). On the other hand, if the 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 coordinates indicating the spatial position of the end-effector, and α, γ, and ϕ represent the orientation in terms of the Euler angles ZYX convention. By equalizing the matrices in (2) and (3), the following equations are derived:

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

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

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

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

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

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

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

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

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

$$a_z = s_{23}c_5 + c_{23}c_4s_5 \tag{13}$$

$$if\,(n_x = 0 \;\&\; n_y = 0) \Rightarrow \begin{cases} \alpha = \operatorname{atan2}(o_x, o_y) \\ \gamma = \pi/2 \\ \phi = 0 \end{cases} \tag{14}$$

$$if\,(n_x \neq 0 \; or \; n_y \neq 0) \Rightarrow \begin{cases} \alpha = \text{atan2}(o_z, a_z) \\ \gamma = \text{atan2}(-n_z, \sqrt{n_x^2 + n_y^2}) \\ \phi = \text{atan2}(n_y, n_x) \end{cases} \qquad (15)$$

From (4)-(15), 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.

B. Inverse Kinematics

Although it is very difficult, the inverse kinematics solution can be found by manipulating the following equation:

$$\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -s_1c_4 - c_1s_{23}s_4 & -c_1c_{23}s_5 + s_1s_4c_5 - c_1s_{23}c_4c_5 & c_1c_{23}c_5 + s_1s_4s_5 - c_1s_{23}c_4s_5 & a_5(c_1c_{23}c_5 + s_1s_4s_5 - c_1s_{23}c_4s_5) + c_1(d_4c_{23} + a_2c_2 + a_1) \\ c_1c_4 - s_1s_{23}s_4 & -s_1c_{23}s_5 - c_1s_4c_5 - s_1s_{23}c_4c_5 & s_1c_{23}c_5 - c_1s_4s_5 - s_1s_{23}c_4s_5 & a_5(s_1c_{23}c_5 + c_1s_4s_5 - s_1s_{23}c_4s_5) + s_1(d_4c_{23} + a_2c_2 + a_1) \\ c_{23}s_4 & -s_{23}s_5 + c_{23}c_4c_5 & s_{23}c_5 + c_{23}c_4s_5 & a_5(s_{23}c_5 + c_{23}c_4s_5) + d_4s_{23} + a_2s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (16)$$

For instance, after multiplying both sides by the inverse matrix of ${}^0A_1$, some elements in the matrices will contain one joint variable only. Paring those elements in both sides will produce possible solutions to some joint variables. This process can be repeated until solutions for all the joint angles are obtained. More details about this process can be found in [13].

If the position and orientation of the P2Arm end-effector are given, potential inverse kinematics solutions can be obtained in terms of the following assumptions:

a) Assuming sin(θ_5)>=0 and sin(θ_4)>0:

$$\theta_1 = \text{atan}\,2((a_5a_y - p_y)\cdot sign(a_yp_x - a_xp_y), (a_5a_x - p_x)\cdot sign(a_yp_x - a_xp_y)) \qquad (17)$$

$$\theta_5 = \text{atan}\,2(a_xs_1 - a_yc_1, o_xs_1 - o_yc_1) \qquad (18)$$

$$\theta_4 = \text{atan}\,2((o_xs_1 - o_yc_1)/c_5, n_yc_1 - n_xs_1) \qquad (19)$$

$$\theta_2 = \text{atan}\,2((a_5c_5 + d_4)(n_xc_1 + n_ys_1) + p_zs_4 - a_5n_zc_4s_5, s_4(p_xc_1 + p_ys_1 - a_1) - nz(a_5c_5 + d_4) - a_5c_4s_5(n_xc_1 + n_ys_1)) \qquad (20)$$

$$\theta_3 = \text{atan}\,2((-n_xc_1c_2 - n_ys_1c_2 - n_zs_2)\cdot sign(s_4), (-n_xc_1s_2 - n_ys_1s_2 + n_zc_2)\cdot sign(s_4)) \qquad (21)$$

b) Assuming sin(θ_5)<=0 and sin(θ_4)<0:

$$\theta_1 = \text{atan}\,2((a_5a_y - p_y)\cdot sign(a_yp_x - a_xp_y), (a_5a_x - p_x)\cdot sign(a_yp_x - a_xp_y)) \qquad (22)$$

$$\theta_5 = \operatorname{atan2}(a_y c_1 - a_x s_1, o_y c_1 - o_x s_1) \tag{23}$$

$$\theta_4 = \operatorname{atan2}((o_x s_1 - o_y c_1)/c_5, n_y c_1 - n_x s_1) \tag{24}$$

$$\begin{aligned}\theta_2 = \operatorname{atan2}(&a_5 n_z c_4 s_5 - (a_5 c_5 + d_4)(n_x c_1 + n_y s_1) - p_z s_4,\\ &nz(a_5 c_5 + d_4) + a_5 c_4 s_5 (n_x c_1 + n_y s_1) - s_4 (p_x c_1 + p_y s_1 - a_1))\end{aligned} \tag{25}$$

$$\begin{aligned}\theta_3 = \operatorname{atan2}(&(-n_x c_1 c_2 - n_y s_1 c_2 - n_z s_2) \cdot sign(s_4),\\ &(-n_x c_1 s_2 - n_y s_1 s_2 + n_z c_2) \cdot sign(s_4))\end{aligned} \tag{26}$$

c) Assuming sin(θ5)<=0 and sin(θ4)>0:

$$\begin{aligned}\theta_1 = \operatorname{atan2}(&(p_y - a_5 a_y) \cdot sign(a_y p_x - a_x p_y),\\ &(p_x - a_5 a_x) \cdot sign(a_y p_x - a_x p_y))\end{aligned} \tag{27}$$

$$\theta_5 = \operatorname{atan2}(a_x s_1 - a_y c_1, o_x s_1 - o_y c_1) \tag{28}$$

$$\theta_4 = \operatorname{atan2}((o_x s_1 - o_y c_1)/c_5, n_y c_1 - n_x s_1) \tag{29}$$

$$\begin{aligned}\theta_2 = \operatorname{atan2}(&(a_5 c_5 + d_4)(n_x c_1 + n_y s_1) + p_z s_4 - a_5 n_z c_4 s_5,\\ &s_4 (p_x c_1 + p_y s_1 - a_1) - nz(a_5 c_5 + d_4) - a_5 c_4 s_5 (n_x c_1 + n_y s_1))\end{aligned} \tag{30}$$

$$\begin{aligned}\theta_3 = \operatorname{atan2}(&(-n_x c_1 c_2 - n_y s_1 c_2 - n_z s_2) \cdot sign(s_4),\\ &(-n_x c_1 s_2 - n_y s_1 s_2 + n_z c_2) \cdot sign(s_4))\end{aligned} \tag{31}$$

d) Assuming sin(θ5)>=0 and sin(θ4)<0:

$$\begin{aligned}\theta_1 = \operatorname{atan2}(&(p_y - a_5 a_y) \cdot sign(a_y p_x - a_x p_y),\\ &(p_x - a_5 a_x) \cdot sign(a_y p_x - a_x p_y))\end{aligned} \tag{32}$$

$$\theta_5 = \operatorname{atan2}(a_y c_1 - a_x s_1, o_y c_1 - o_x s_1) \tag{33}$$

$$\theta_4 = \operatorname{atan2}((o_x s_1 - o_y c_1)/c_5, n_y c_1 - n_x s_1) \tag{34}$$

$$\begin{aligned}\theta_2 = \operatorname{atan2}(&a_5 n_z c_4 s_5 - (a_5 c_5 + d_4)(n_x c_1 + n_y s_1) - p_z s_4,\\ &nz(a_5 c_5 + d_4) + a_5 c_4 s_5 (n_x c_1 + n_y s_1) - s_4 (p_x c_1 + p_y s_1 - a_1))\end{aligned} \tag{35}$$

$$\begin{aligned}\theta_3 = \operatorname{atan2}(&(-n_x c_1 c_2 - n_y s_1 c_2 - n_z s_2) \cdot sign(s_4),\\ &(-n_x c_1 s_2 - n_y s_1 s_2 + n_z c_2) \cdot sign(s_4))\end{aligned} \tag{36}$$

e) Assuming sin(θ4)=0:

$$\theta_1 = \operatorname{atan2}(-n_x, n_y) \tag{37}$$

$$\theta_2 = \text{atan}\,2(r_z, r_x) - \text{acos}(\frac{r_x^2 + r_z^2 + a_2^2 - d_4^2}{2a_2\sqrt{r_x^2 + r_z^2}}) + 2m_1\pi \tag{38}$$

$$\theta_3 = \pi - \text{acos}(\frac{a_2^2 + d_4^2 - r_x^2 - r_z^2}{2a_2 d_4}) \tag{39}$$

$$\theta_4 = 0 \tag{40}$$

$$\theta_5 = \text{atan}\,2(a_z, o_z) - \theta_2 - \theta_3 + 2m_2\pi \tag{41}$$

where $r_x = p_x / c_1 - a_5 o_z - a_1$ or $r_x = p_y / s_1 - a_5 o_z - a_1$ if c_1 is too small, and $r_z = p_z - a_5 a_z$, m_1, m_2 = -1, 0, or 1. For a similar reason, if c_5 is too small, equations (19), (24), (29), and (34) should be replaced by

$$\theta_4 = \text{atan}\,2((a_x s_1 - a_y c_1)/s_5, n_y c_1 - n_x s_1) \tag{42}$$

In case e), if the solution is checked as incorrect, (38) and (39) should be replaced by

$$\theta_2 = \text{atan}\,2(r_z, r_x) + \text{acos}(\frac{r_x^2 + r_z^2 + a_2^2 - d_4^2}{2a_2\sqrt{r_x^2 + r_z^2}}) + 2m_1\pi \tag{43}$$

$$\theta_3 = -\pi + \text{acos}(\frac{a_2^2 + d_4^2 - r_x^2 - r_z^2}{2a_2 d_4}) \tag{44}$$

The solution under the assumption of $sin(\theta_5)=0$ is also available. However, this solution provides the same joint angles as those provided by a), b), c) or d). It should be noted that before the joint angles are solved we do not know which assumption is correct. Our strategy for choosing the correct solution is to try all the potential solutions and check using the forward kinematics which solution produces the given position and orientation correctly. Due to the inaccuracy problem caused by atan2(y, x), when $x \approx y \approx 0$, there could be no correct solution among all the potential solutions for some given positions and orientations. How serious this problem is will be investigated in Section V, and an alternative approach in case this problem exists is developed in the next section.

III. Inverse Kinematics Solution by Optimal Search

When the analytical inverse model gives an incorrect solution, a common alternative approach is optimal search that finds a solution by minimizing the error between the desired and current positions and orientations:

$$E = \frac{1}{2}\left\|\boldsymbol{X}_{desired} - \boldsymbol{X}_{current}\right\|^2 \tag{45}$$

where the current end-effector position and orientation $\boldsymbol{X}_{current}$ is calculated by the forward kinematics equations (4)-(15). The updating of joint angles can be carried out as follows based on a gradient-descent algorithm:

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta \partial E / \partial \boldsymbol{\theta} = \boldsymbol{\theta}(t) + \eta \boldsymbol{J}^T(\boldsymbol{X}_{desired} - \boldsymbol{X}_{current}) \tag{46}$$

where η is a small positive number controlling the search step, $\boldsymbol{J}$ is the Jacobian matrix of the manipulator:

$$\boldsymbol{J} = \begin{bmatrix} \partial p_x/\partial\theta_1 & \partial p_x/\partial\theta_2 & \partial p_x/\partial\theta_3 & \partial p_x/\partial\theta_4 & \partial p_x/\partial\theta_5 \\ \partial p_y/\partial\theta_1 & \partial p_y/\partial\theta_2 & \partial p_y/\partial\theta_3 & \partial p_y/\partial\theta_4 & \partial p_y/\partial\theta_5 \\ \partial p_z/\partial\theta_1 & \partial p_z/\partial\theta_2 & \partial p_z/\partial\theta_3 & \partial p_z/\partial\theta_4 & \partial p_z/\partial\theta_5 \\ \partial\alpha/\partial\theta_1 & \partial\alpha/\partial\theta_2 & \partial\alpha/\partial\theta_3 & \partial\alpha/\partial\theta_4 & \partial\alpha/\partial\theta_5 \\ \partial\gamma/\partial\theta_1 & \partial\gamma/\partial\theta_2 & \partial\gamma/\partial\theta_3 & \partial\gamma/\partial\theta_4 & \partial\gamma/\partial\theta_5 \\ \partial\phi/\partial\theta_1 & \partial\phi/\partial\theta_2 & \partial\phi/\partial\theta_3 & \partial\phi/\partial\theta_4 & \partial\phi/\partial\theta_5 \end{bmatrix} \quad (47)$$

The derivatives in the Jacobian matrix can be easily derived using the forward kinematics equations. The gradient-descent-based search will stop either when a preset maximum number of epochs have been reached or when the correctness criterion has been met. Another optimal search strategy is to design a trajectory between the desired and current positions and orientations. The pseudo-inverse of the Jacobian matrix is applied to update the joint angles in each step following the trajectory [9]. However, this strategy does not work properly due to the singularity of the Jacobian matrix when the dof of the robotic arm is less than 6. More effort for improving the optimal search performance, e.g., using adaptive search steps, will be made and will be reported in the final version of this paper.

IV. A HYBRID APPROACH FOR P2ARM INVERSE KINEMATICS CONTROL

The results of testing the derived analytical inverse kinematics model show that it can provide inverse solutions for almost any given positions and orientations within the reachable space, with the advantages of high speed and high accuracy over neural networks and optimal search methods. However, on some rare occasions, the analytical inverse model provides completely wrong solutions due to the inaccuracy problem in atan2 function, which is a disadvantage of the analytical inverse model over neural networks and optimal search methods. In order to avoid this disadvantage of the analytical model, we use a hybrid approach, as shown in Figure 3.

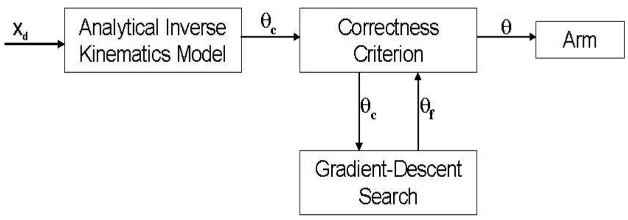

Figure 3 A hybrid approach to inverse kinematics control

Given a position and orientation $\mathbf{x}_d$, the analytical inverse model will provide a joint angle vector $\boldsymbol{\theta}_c$. Its corresponding position and orientation will be calculated using the forward kinematics model. If this solution meets the correctness criterion, the joint angles will be sent to the robotic arm as control commands, otherwise, an optimal search will be conducted to get a satisfactory solution, which will be checked and sent to the robotic arm if correct.

The correctness criterion used here is defined as follows:

$$\begin{aligned} &|x-\hat{x}|^2 + |y-\hat{y}|^2 + |z-\hat{z}|^2 \le e_{position} \\ &|\alpha-\hat{\alpha}| \le e_{orientation}, |\gamma-\hat{\gamma}| \le e_{orientation}, |\phi-\hat{\phi}| \le e_{orientation} \end{aligned} \quad (48)$$

where (x, y, z, α, γ, ϕ) represents the given position and orientation, the variables with ^ represent the reached position and orientation by the inverse kinematics control, and $e_{position}$ and $e_{orientation}$ are error thresholds for position and orientation respectively.

Usually the optimal search will take a relatively longer time in comparison with the analytical solution. This is a shortcoming for real-time control. However, the optimal search is hardly activated, as shown in the experimental results in Section V.

V. Experimental Results

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. As the analytical inverse method is able to provide accurate solutions, we set $e_{position}=1cm$ and $e_{orientation}=1^o$ in the correctness criterion for the analytical solution. For the optimal search solution, we set $e_{position}=1cm$ and $e_{orientation}=10^o$, because it is difficult for the optimal search to achieve very high orientation accuracy. We have conducted experiments with multiple runs, each run was based on 1 million or 10 million positions and orientations. Table 2 shows some average performance of the analytical inverse model and the hybrid approach. It can be seen from the table that the inverse kinematics problem of the P2Arm has been almost perfectly solved. We noticed that all the errors correspond to positions and orientations generated by $\theta_4 = \pm 90^o$ and $\theta_2 = -\theta_3$, which result in $n_x=n_y=o_z=a_z=0$ and thus make the orientation angles calculated by (14) and (15) uncertain. Therefore the errors are not actually caused by the inverse process. These errors could be removed by using (n_x, n_y, n_z, o_x, o_y, o_z, a_x, a_y, a_z) to represent the orientation rather than (α, γ, ϕ). This will be investigated further and the results will be included in the final version of this paper.

TABLE II. Errors in the P2Arm Inverse Kinematics Solutions

No. of testing positions & orientations	No. of errors from analytical inverse alone (percentage)	No. of errors from hybrid approach (percentage)
1,000,000	5 (0.0005%)	2 (0.0002%)
10,000,000	50 (0.0005%)	23 (0.00023%)

VI. Conclusions

An analytical inverse kinematics model for a widely used robotic arm, P2Arm, is firstly derived in this paper. A hybrid approach combining the derived analytical inverse kinematics model with an optimal search method is adopted, which provides an almost perfect solution to the P2Arm inverse kinematics problem. We believe that the solution developed in this paper will make the P2Arm more useful in applications with unpredictable trajectory movements in unknown environments. The methods used for deriving the inverse kinematics model for the P2Arm could be applied to other types of robotic arms. Future work would include looking for better analytical inverse kinematics models as the model derived in this paper is not a unique solution, improving the optimal search algorithm, integrating with neural networks, and the robustness analysis of the proposed approach.

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] J. Guo and V. Cherkassky, "A solution to the inverse kinematics problem in robotics using neural network processing," *Proc. IEEE Conf. on Neural Networks*, 1989, vol. II, pp. 299-304.

[5] A.P. Lüdtke F. and P.M. Engel, "Positioning a robot arm: an adaptive neural network," *Proc. Int. Workshop on Neural Networks for Indentification, Control, Robotics and Signal Processing*, 1996, pp. 440-448.

[6] D. Demers and K. Kreutz-Delgado, "Learning global properties of nonredundant kinematic mappings," *Int. J. of Robotics Research*, vol. 17, no 5, pp. 547-560, 1998.

[7] S. Tejomurtula and S. Kak, "Inverse kinematics in robotics using neural networks," *Information Sciences*, vol. 116, pp 147-164, 1999.

[8] E. Oyama and S. Tachi, "Modular neural net system for inverse kinematics learning," *Proc. IEEE Int. Conf. on Robotics and Automation*, 2000, pp 3239-3246.

[9] 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.

[10] 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.

[11] M.B. Menhaj and M. Rouhani, "A novel neuro-based model reference adaptive control for a two link robot arm," *Proc. IEEE Int. Joint Conf. on Neural Networks*, 2002, pp. 45-52.

[12] B. Schäfer, R.Krenn and B. Rebele, "On inverse kinematics and kinetics of redundant space manipulator simulation," J. of Computational and Applied Mechanics, vol. 4, No 1, pp 53-70, 2003.

[13] 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.

[14] http://www.activrobots.com/ACCESSORIES/pioneerarm.html.