Analytical Inverse Kinematics for a 5-DoF Robotic Arm with a Prismatic Joint

Vighnesh Vatsal¹*, Guy Hoffman¹

Abstract
We present an analytical solution for the inverse kinematics (IK) of a robotic arm with one prismatic joint and four revolute joints. This 5-DoF design is a result of minimizing weight while preserving functionality of the device in a wearable usage context. Generally, the IK problem for a 5-DoF robot does not guarantee solutions due to the system being over-constrained. We obtain an analytical solution by applying geometric projections and limiting the ranges of motion for each DoF. We validate this solution by reconstructing randomly sampled end-effector poses, and find position errors below 2 cm and orientation errors below 4°.

Keywords
Inverse Kinematics — Over-constrained Manipulators — Wearable Robotic Forearm

1. Introduction
We present an analytical solution for the inverse kinematics (IK) of a robotic arm with five degrees of freedom (DoFs). The kinematic structure of the robot is RRPRR, where “R” denotes a revolute joint and “P” denotes a prismatic joint. Our use context is the development of a wearable robotic “third arm” intended for close range human-robot collaboration. It aims to extend a user’s reach, and provide the ability for manipulation below and behind the user in tasks similar to those shown in Fig. 1. We find that a 5-DoF structure (excluding gripper) is sufficient for performing these tasks while keeping the weight to a minimum (Fig. 2), as indicated by results from an iterative user-centered design procedure [1].

Closed-form analytical IK solutions, and numerical solutions exist for general robotic arms with 6-R joints [2, 3, 4]. When possible, analytical methods are preferred over numerical ones due to faster computational speeds and the ability to predict the existence of solutions [5].

However, the IK problem for a 5-DoF robot is over-constrained, and no general solution exists [6]. Gan et al. [7] have developed an analytical IK solution for poses reachable by a 5-R Pioneer 2 robotic arm. We extend their approach to the case where the robotic arm includes a prismatic joint (Fig. 3), and present an analytical solution along with the constraints on it.¹ We evaluate the numerical quality of this solution by reconstructing a randomly sampled set of end-effector poses within the robot’s workspace bounding box. We obtain average position errors below 2 cm, and orientation errors below 4° in this

¹The code for this solution can be found here: http://github.com/vighv/5DoF_IK

Figure 1. The wearable robotic arm in two illustrative usage scenarios: self-handover (left), and handover to another person (right)
2. Forward Kinematics Equations

Forward Kinematics (FK) provides the mapping from a robot’s joint positions to the pose of the end-effector. We assign coordinate frames to each link of the robotic arm, and derive the FK equations using the Denavit-Hartenberg (D-H) convention [8].

The base frame $O_0$ in our case lies at the top of horizontal panning DoF (Fig. 4), and the final frame $O_6$ is attached to the end-effector, lying at the mid point of the gripper’s fingers. The D-H method uses four parameters: $\alpha, a, d$ and $\theta$ to specify the relationship between frames (Table 1). Using these parameters, a homogeneous transformation matrix $T^{i+1}_i$ is defined for the pose of frame $i+1$ as seen in frame $i$:

$$
T^{i+1}_i = \begin{bmatrix}
    \cos \theta_i & -\sin \theta_i \cos \alpha & \sin \theta_i \sin \alpha & a_i \cos \theta_i \\
    \sin \theta_i & \cos \theta_i \cos \alpha & -\cos \theta_i \sin \alpha & a_i \sin \theta_i \\
    0 & \sin \alpha_i & \cos \alpha_i & d_i \\
    0 & 0 & 0 & 1
\end{bmatrix} \quad (1)
$$

The joint variables are $\theta_i$ for the four revolute joints, and $d_3$ for the prismatic joint. The end-effector is fixed with respect to the wrist pitching DoF. However, it is listed separately in order to decouple it from wrist pitching for ease of solution of the inverse kinematics, as in [9].

The transformation $T^6_0$, between the base frame $O_0$ and the end-effector frame $O_6$ can be found by successive multiplication:

$$
T^6_0 = T^1_0 T^2_1 T^3_2 T^4_3 T^5_4 T^6_5 \quad (2)
$$

Expanded out in terms of individual elements:

$$
T^6_0 = \begin{bmatrix}
    R_{1x} & R_{2x} & R_{3x} & P_x \\
    R_{1y} & R_{2y} & R_{3y} & P_y \\
    R_{1z} & R_{2z} & R_{3z} & P_z \\
    0 & 0 & 0 & 1
\end{bmatrix} \quad (3)
$$

The rotation between these frames is described by $R = [\hat{R}_1, \hat{R}_2, \hat{R}_3] \in SO(3)$, where $SO(3)$ is the 3D rotation group consisting of $3 \times 3$ orthogonal matrices with unit determinant. $\vec{P} = [P_x, P_y, P_z]^T \in \mathbb{R}^3$ describes the translation between these frames. We will denote $\cos \theta_i$ as $c_i$ and $\sin \theta_i$ as $s_i$ in the rest of this paper. The length parameters for the robot are $l_1$, $l_2$ and $l_3$, which correspond to $d_1$, $d_4$ and $a_6$ respectively from Table 1. There are twelve equations that constitute the forward kinematics:

$$
R_{1x} = c_1 s_2 s_3 - c_3 s_1 s_4 + c_1 c_2 c_4 \quad (4)
$$
The inverse kinematics problem involves finding the values of the joint variables for a desired position and orientation (pose) of the end-effector.

We start with the solution to the horizontal panning angle \( \theta_1 \) using the geometric projection method described in [9].

\[ R_{1y} = c_5(c_1s_4 - c_2c_4s_1) + s_1s_2s_5 \]  
\[ R_{1z} = -c_2s_5 - c_4c_5s_2 \]  
\[ R_{2x} = -c_4s_1 - c_1c_2s_4 \]  
\[ R_{2y} = -c_1c_4 - c_2s_1s_4 \]  
\[ R_{2z} = -s_2s_4 \]  
\[ R_{3x} = -c_1s_2c_5 - s_5(s_1s_4 + c_1c_2c_4) \]  
\[ R_{3y} = s_5(c_1s_4 - c_2c_4s_1) - s_1s_2c_5 \]  
\[ R_{3z} = c_2c_5 - s_2c_4s_5 \]  
\[ P_x = l_2c_1s_2 - l_3(c_5s_1s_4 + c_5c_1c_2c_4) + l_3c_1s_2s_5 + d_3s_1s_2 \]  
\[ P_y = l_2s_1s_2 + l_3(c_5c_1s_4 + c_5s_1c_2c_4) + l_3s_1s_2s_5 + d_3s_1s_2 \]  
\[ P_z = l_1 - l_2c_2 - l_3c_2s_5 - l_3c_4c_5s_2 - d_3c_2 \]  

### 3. Inverse Kinematics

The inverse kinematics problem involves finding the values of the joint variables for a desired position and orientation (pose) of the end-effector.

We start with the solution to the horizontal panning angle \( \theta_1 \) using the geometric projection method described in [9].

![Figure 5. Projecting the position of O5 onto the XY-plane of the base frame O0 to solve for \( \theta_1 \).](image)

As shown in Fig. 5, \( \theta_1 \) is the angle between the projection of \( \vec{P} \) onto the XY-plane, and the X-axis of the base frame \( O_0 \). \( \vec{P} \) can be obtained by using the transformation between \( O_5 \) and \( O_0 \):

\[ T_0^5 = T_0^5 [T_3^6]^{-1} T_1^3 \]

The components \( P_x' \) and \( P_y' \) of \( \vec{P} \) can be found from the corresponding components in the last column of \( T_0^5 \). The relation between \( \theta_1 \) and the projections of \( \vec{P} \) is as follows:

\[ \tan \theta_1 = \frac{P_x'}{P_y'} = \frac{-l_3R_{1y} + P_x}{-l_3R_{1x} + P_x} \]

\( \theta_1 \) can be found using the four-quadrant inverse tangent function \( \text{atan2}(x, y) \), as it lies in the desired range of \((-180^\circ, 180^\circ)\). Also note that \( \theta_2 \) is never fully vertical, since we have restricted the range of DoF-2 in the open interval \((0^\circ, 90^\circ)\). This ensures that a solution for \( \theta_1 \) always exists, since \( O_0, O_1, O_2 \) and \( O_3 \) are never collinear.

The solution strategies for other joint variables can be grouped into two cases, depending on whether \( R_{2z} \neq 0 \) or \( R_{2z} = 0 \), which correspond to whether the links of the robot become coplanar.

### 3.1 Case 1: \( R_{2z} \neq 0 \)

From the forward kinematics, we know that:

\[ R_{2z} = -s_2s_4 \]
\[ R_{2z} \neq 0 \] implies that both \( s_2 \neq 0 \) and \( s_4 \neq 0 \). Then we can perform the following substitutions:

\[ s_4 = -\frac{R_{2z}}{s_2} \quad (19) \]

This results in a pair of linear equations in \( \cos \theta_4 \) and \( \cot \theta_2 \):

\[
\begin{bmatrix}
    s_4 \\
    -c_1
\end{bmatrix}
\begin{bmatrix}
    c_1 R_{2z} \\
    -s_1 R_{2z}
\end{bmatrix}
\begin{bmatrix}
    \cos \theta_4 \\
    \cot \theta_2
\end{bmatrix}
= \begin{bmatrix}
    R_{2x} \\
    R_{2y}
\end{bmatrix} \quad (20)
\]

We can find \( \tan \theta_2 \) by solving this pair of linear equations, and constrain the solution to lie in the range \((0^\circ, 90^\circ)\). These equations also give us \( \cos \theta_4 \), with which we can find \( \tan \theta_4 \):

\[ \tan \theta_4 = -\frac{R_{2z}}{s_2 \cos \theta_4} \quad (21) \]

\( \theta_4 \) is computed using the four-quadrant inverse tangent function \( \text{atan}2(x, y) \) as it lies in the range \((-180^\circ, 180^\circ)\). An inverse tangent function is preferred over an inverse cosine function due to greater numerical stability. Once \( \theta_1, \theta_2, \) and \( \theta_4 \) are known, equations (6) and (12) reduce to linear expressions in terms of \( \sin \theta_1 \) and \( \cos \theta_4 \):

\[
\begin{bmatrix}
    -c_2 \\
    -c_4 s_2
\end{bmatrix}
\begin{bmatrix}
    \sin \theta_5 \\
    \cos \theta_5
\end{bmatrix}
= \begin{bmatrix}
    R_{1z} \\
    R_{3z}
\end{bmatrix} \quad (22)
\]

\( \theta_5 \) can be obtained using the inverse tangent function \( \text{atan}2(x, y) \), and constrained to lie in the range \((0^\circ, 180^\circ)\). Once all joint angles are known, equations (13-15) yield three candidate solutions for the length of the prismatic joint \( d_3 \):

\[
d_3^1 = (P_1 - l_2 c_1 s_2 + l_3 (c_5 s_1 s_4 + c_5 c_1 c_2 c_4) - l_3 c_1 s_2 s_5) / c_1 s_2 \quad (23)
\]

\[
d_3^2 = (P_1 - l_2 s_1 s_2 - l_3 (c_5 c_1 s_4 + c_5 s_1 c_2 c_4) - l_3 s_1 s_2 s_5) / s_1 s_2 \quad (24)
\]

\[
d_3 = (P_z - l_1 + l_2 c_2 + l_3 c_2 s_5 + l_3 c_4 s_5 s_2) / c_2 \quad (25)
\]

The candidate for \( d_3 \) that lies within the range \([0.33 \text{ m}, 0.45 \text{ m}]\) is chosen as the solution.

### 3.2 Case 2: \( R_{2z} = 0 \)

A different strategy must be employed in situations when \( R_{2z} \) is zero. Since \( R_{2z} = s_2 s_4 \), and we have restricted \( \theta_2 \) to lie in the open interval \((0^\circ, 90^\circ)\), \( R_{2z} = 0 \Rightarrow s_4 = 0 \). Additionally, since \( \theta_4 \) is also restricted to be in the open interval \((-180^\circ, 180^\circ)\), we obtain \( \theta_4 = 0 \).

This places the robot in a configuration similar to the one shown in Fig. 6, where all the links of the arm become co-planar. With \( \theta_4 = 0 \), equations (4, 5) and (10 - 12) simplify to:

\[
R_{1x} = -c_1 \cos(\theta_2 + \theta_5) \quad (26)
\]

\[
R_{1y} = -s_1 \cos(\theta_2 + \theta_5) \quad (27)
\]

\[
R_{3x} = -c_1 \sin(\theta_2 + \theta_5) \quad (28)
\]

\[
R_{3y} = -s_1 \sin(\theta_2 + \theta_5) \quad (29)
\]

\[
R_{3z} = \cos(\theta_2 + \theta_5) \quad (30)
\]

This gives us the sum (\( \theta_2 + \theta_5 \)):

\[
\tan(\theta_2 + \theta_5) = \frac{R_{3x}}{R_{1y}} = \frac{R_{3y}}{R_{1x}} \quad (31)
\]

To find the the joint angle \( \theta_2 \) separately, we look at the triangle formed by the points \( O_0, O_2 \) and \( O_5 \), as shown in Fig. 7.

The edge lengths \( t_1, t_2 \) and \( t_3 \) of this triangle are known quantities, since the vector \( O_0 \vec{O}_2 \) is fixed, and the vector \( O_0 \vec{O}_5 \) can be found using the last column of the matrix \( T_0^5 \) in equation (16). We can apply the cosine formula to this triangle:

\[
t_1 = \|O_0 \vec{O}_2\| \quad (32)
\]
In Sections 3.1 and 3.2, we chose between multiple candidate solutions by applying the condition that the variables must lie within the intervals specified in Table 1. Apart from these, there are also conditions on the revolute joint angles $\theta_1$ - $\theta_5$ that arise from the forward kinematic equations. These conditions make use of the individual transformation matrices $T_{i}^{i+1}$ for each row of Table 1.

Let us consider the product $T_0^1T_1^2T_2^3$:

$$
T_0^1T_1^2T_2^3 = \begin{bmatrix}
-c_1c_2 & -s_1 & c_1s_2 & d_3c_1s_2 \\
-c_2s_1 & c_1 & s_1s_2 & d_3s_1s_2 \\
-s_2 & 0 & -c_2 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix} 
$$

The entry in row 3, column 2 of this matrix is zero. Using equation (2), we can compute this matrix in terms of the final transformation $T_0^6$:

$$
T_0^1T_1^2T_2^3 = T_0^6[T_3^4T_4^5T_5^6]^{-1}
$$

Equating corresponding matrix entries at row 3, column 2 on both sides, we obtain the following condition on $\theta_4$ and $\theta_5$:

$$
R_1c_5s_4 - R_2c_4 + R_3s_4s_5 = 0
$$

Similarly, in the product $T_3^4T_4^5T_5^6$, the element at row 3, column 2 is also zero:

$$
T_3^4T_4^5T_5^6 = \begin{bmatrix}
c_4c_5 & s_4 & c_4s_5 & l_3c_4c_5 \\
s_4c_5 & -c_4 & s_4s_5 & l_3s_4c_5 \\
s_5 & 0 & -c_5 & l_2 + l_3s_5 \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

This can also be written in terms of the final transformation matrix:

$$
T_3^4T_4^5T_5^6 = [T_0^1T_1^2T_2^3]^{-1}T_0^6
$$

Similar to equation (40), we obtain the following condition on $\theta_1$ and $\theta_2$:

$$
R_2c_1s_2 - R_2c_2 + R_3s_1s_2 = 0
$$

Candidate IK solutions that do not satisfy conditions (40) and (43) are rejected.

## 5. Evaluation

The analytical IK solution presented here is exact. It should yield no errors if the computations are performed with infinite numerical precision. However, the inverse tangent and inverse cosine functions used in our formulation are numerically sensitive, and are susceptible to propagation of errors [7].

To evaluate the quality of our solution, we follow the procedure by which a robotic arm is generally actuated, where the
joint angles for a desired pose are computed using IK and sent to the motors, resulting in an end-effector pose through FK. Our evaluation measures the numerical errors that will manifest in a physical implementation of this pose reconstruction, even with the assumption of arbitrarily precise actuators.

The bounding volume of the robot’s workspace lies in a hemispherical shell of inner radius $r_i = d_i^{\text{min}} + l_2$ and outer radius $r_o = d_i^{\text{max}} + l_2 + l_3$. We sample end-effector positions from uniform distributions in $X$, $Y$ and $Z$ coordinate ranges within this volume. Orientations are sampled as Yaw-Pitch-Roll Euler angles from uniform distributions in $X$, $Y$, and $Z$ coordinate ranges, and [0, 2π] for yaw and roll, and [0, π] for pitch (Fig. 8).

![Figure 8.](image)

Each sampled pose is represented by a position vector $\vec{P}$, and a rotation matrix $R$ constructed from these Euler angles. From the IK computation on this pose, we obtain a vector $\vec{\Theta}$ of joint variables. This pose is reconstructed into a position $\vec{P}'$ and rotation matrix $R'$ on applying FK equations to $\vec{\Theta}$. We consider the absolute error as the accuracy metric for position:

$$
\begin{bmatrix}
\Delta X_{abs} \\
\Delta Y_{abs} \\
\Delta Z_{abs}
\end{bmatrix} = 
\begin{bmatrix}
|P_x - P'_x| \\
|P_y - P'_y| \\
|P_z - P'_z|
\end{bmatrix}
$$

(44)

For the accuracy of rotation reconstruction, we consider the geodesic metric on the rotation group SO(3), as described in [10]:

$$
\Delta R_{abs} = ||\log(R'R^T)||
$$

(45)

Where the norm of a matrix $A$ is defined as:

$$
||A|| = \sqrt{\frac{1}{2} \text{trace}(A^TA)}
$$

(46)

This metric returns the magnitude of rotation angle in the range [0, π] measured along the geodesic between $R$ and $R'$ in SO(3). It captures rotational deviation more accurately compared to the Euclidean distance between Euler angles, since a smaller Euclidean distance can map to a larger distance in SO(3) and vice versa [10].

| Table 2. Absolute errors in pose reconstruction |
|-----------------------------------------------|
| Quantity | Mean | Std. Dev. |
|-----------------|-------------|----------|
| $\Delta X_{abs}$ (m) | 6.099 $\times 10^{-3}$ | 6.543 $\times 10^{-3}$ |
| $\Delta Y_{abs}$ (m) | 6.092 $\times 10^{-3}$ | 6.554 $\times 10^{-3}$ |
| $\Delta Z_{abs}$ (m) | 1.681 $\times 10^{-2}$ | 1.448 $\times 10^{-2}$ |
| $\Delta R_{abs}$ (rad) | 5.343 $\times 10^{-2}$ | 1.972 $\times 10^{-1}$ |

We sampled $10^7$ poses in the robot’s bounding workspace volume, out of which $N = 7.4 \times 10^6$ poses satisfied the constraint equations (40) and (43), and were within the range of prismatic joint lengths. Table 2 lists the means and standard deviations of the absolute position and orientation errors for the reconstruction of these poses. The mean position errors are in the range of ~ 6 mm in the $X$ and $Y$ coordinates, and ~ 2 cm in the $Z$ coordinate. These errors respectively correspond to 1.18% and 3.92% of the arm’s total maximum length (0.51 m). The mean error in the $Z$ coordinate is higher than in $X$ and $Y$, possibly due to the greater sensitivity of $P_z$ to errors in the estimate of the prismatic joint length $d_3$ (equation 15). The mean orientation error is ~ 0.054 radians, or just above 3°.

6. Conclusion

We presented an analytical solution for the inverse kinematics of a 5-DoF wearable robotic arm with a prismatic actuator present in the serial chain. We validated our solution by testing it on $10^7$ randomly generated poses, and found that we were able to achieve orientation targets to within 4°, and position targets to within 2 cm in terms of numerical errors. We also derived constraints on the joint variables to verify the correctness of solutions for a particular end-effector pose.

Although this method returns solutions for poses that lie within the subspace reachable by our 5-DoF arm, it is numerically unstable for poses that lie near the ends of the joint variable ranges. On the physical robot, we would require approximate solutions for planned end-effector trajectories that may contain poses near these singular points. In such cases, we can augment our analytical solution with a numerical approach, such as gradient descent-based search [9] to obtain a more stable, hybrid solution.

In the context of developing control methods for the wearable robotic arm, the analytical inverse kinematics solution presented here will reduce the computational costs compared to a numerical strategy, leading to better responsiveness of the robot to the user.

Acknowledgments

This work was supported by the National Science Foundation under NRI Award no. 1734399.
References

[1] V. Vatsal and G. Hoffman. Wearing your arm on your sleeve: Studying usage contexts for a wearable robotic forearm. In 2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pages 974–980, Aug 2017.

[2] CS George Lee. Robot arm kinematics, dynamics, and control. Computer, 15(12):62–80, 1982.

[3] Dinesh Manocha and John F Canny. Efficient inverse kinematics for general 6r manipulators. IEEE transactions on robotics and automation, 10(5):648–657, 1994.

[4] Richard P Paul and Bruce Shimano. Kinematic control equations for simple manipulators. In Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on, pages 1398–1406. IEEE, 1979.

[5] Kenneth J Waldron and James Schmiedeler. Kinematics. In Springer Handbook of Robotics, pages 11–36. Springer, 2016.

[6] Deepak Tolani, Ambarish Goswami, and Norman I Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. Graphical models, 62(5):353–388, 2000.

[7] John Q Gan, Eimei Oyama, Eric M Rosales, and Huosheng Hu. A complete analytical solution to the inverse kinematics of the pioneer 2 robotic arm. Robotica, 23(1):123–129, 2005.

[8] Richard S Hartenberg and Jacques Denavit. A kinematic notation for lower pair mechanisms based on matrices. Journal of applied mechanics, 77(2):215–221, 1955.

[9] De Xu, Carlos A Acosta Calderon, John Q Gan, Huosheng Hu, and Min Tan. An analysis of the inverse kinematics for a 5-dof manipulator. International Journal of Automation and Computing, 2(2):114–124, 2005.

[10] Du Q Huynh. Metrics for 3d rotations: Comparison and analysis. Journal of Mathematical Imaging and Vision, 35(2):155–164, 2009.