[2024-1 Robotics]
Chapter 6. Inverse Kinematics

Gyunghoon Park

School of ECE, University of Seoul

Kinematics?: Revisited

Kinematics defines the relation between
the joint coordinate θ and the pos./ori. of the end-effector frame, x_{EE}.

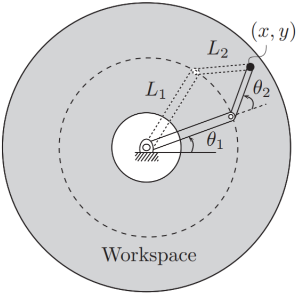
- Forward kinematics (in Chapter 4): $\theta \to x_{\mathrm{EE}}$
- Inverse kinematics (in Chapter 6): $x_{\mathrm{EE}} \to \theta$
- Velocity kinematics (in Chapter 5): $(\theta, \dot{\theta}) \to \dot{x}_{\mathrm{EE}}$

Inverse kinematics problem: For given desired $x^{\star}_{\mathrm{EE}}$, find $\theta^{\star} = \mathrm{IK}(x^{\star}_{\mathrm{EE}})$.

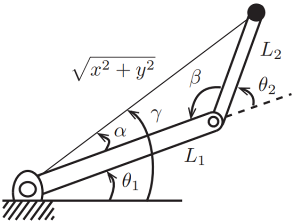
- Help to construct a reference signal for control of each joint.
- Example: `https://spotmicroai.readthedocs.io/en/latest/`

Inverse kinematics: A 2R planar robot case

6.0. Introduction to Chapter



(a) A workspace, and lefty and righty configurations.



(b) Geometric solution.

Figure 6.1: Inverse kinematics of a 2R planar open chain.

Note: The forward kinematics is given by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) \\ L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) \end{bmatrix}$$

(Review) atan2 function

6.0. Introduction to Chapter

The function $\operatorname{atan2}(y, x)$ is a 2-arguments version of $\arctan(y/x)$.

Why required?

- (y, x) and $(-y, -x)$ gives the same value of $\arctan$.
- Yet, the actual $\theta \in (-\pi, \pi]$ are different.

$$\operatorname{atan2}(y, x) = \begin{cases} \arctan(y/x), & \text{if } x > 0 \\ \arctan(y/x) + \pi, & \text{if } x < 0 \text{ and } y \geq 0 \\ \arctan(y/x) - \pi, & \text{if } x < 0 \text{ and } y < 0 \\ +\pi/2, & \text{if } x = 0 \text{ and } y > 0 \\ -\pi/2, & \text{if } x = 0 \text{ and } y < 0 \\ \text{undefined}, & \text{if } x = 0 \text{ and } y = 0 \end{cases}$$

– atan2 in Wikipedia: `https://en.wikipedia.org/wiki/Atan2`

Inverse kinematics: A 2R planar robot case (Cont'd)

By the law of cosines,

$$
\begin{aligned}
L_2^2 &= x^2 + y^2 + L_1^2 - 2\sqrt{x^2+y^2}\,L_1 \cos\alpha, \\
x^2 + y^2 &= L_1^2 + L_2^2 - 2L_1 L_2 \cos\beta,
\end{aligned}
$$

so that

$$
\begin{aligned}
\alpha &= \cos^{-1}\left(\frac{x^2 + y^2 + L_1^2 - L_2^2}{2L_1\sqrt{x^2+y^2}}\right), \\
\beta &= \cos^{-1}\left(\frac{L_1^2 + L_2^2 - x^2 - y^2}{2L_1 L_2}\right)
\end{aligned}
$$

and

$$
\frac{y}{x} = \tan(\gamma) \quad \Rightarrow \quad \gamma = \operatorname{atan2}(y, x).
$$

Thus, a possible solution for the IK problem is

$$\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} = \begin{bmatrix}\gamma - \alpha\\ \pi - \beta\end{bmatrix}$$ where α, β, γ are given above (as functions of (x, y)).

Note:

- IK may not have an unique solution, $\because$

$$\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} = \begin{bmatrix}\gamma + \alpha\\ \beta - \pi\end{bmatrix}.$$

  is also a solution of the IK problem.
- No solution exists for the case when $x^2 + y^2$ lies outside the range $[L_1 - L_2, L_1 + L_2]$.

Two ways of solving IK

6.1. Analytic Inverse Kinematics

One can express the forward kinematics as

$$T(\theta) = e^{[\mathcal{S}_1]\theta_1} \cdots e^{[\mathcal{S}_6]\theta_6} M.$$

Then the IK problem can be understood as:

for given $X^\star \in SE(3)$, find θ such that $T(\theta) = X^\star$.

- Analytic method: Compute the inverse function T^{-1} by hand!
  - Not always possible
  - Only limited structure of the robot is available
- Numerical method: Solve it using computers
  - More general

Analytic IK solution for 6R PUMA-type arm

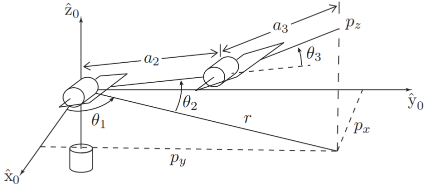


Figure 6.2: Inverse position kinematics of a 6R PUMA-type arm.

Features:

- Two shoulder joints axes intersect orthogonally at a common point
- Joint axis 3 lies in $\hat{x}_0$-$\hat{y}_0$ plane and is aligned parallel with joint axis 2.
- Joint axes 4–6 intersect orthogonally at a common point.

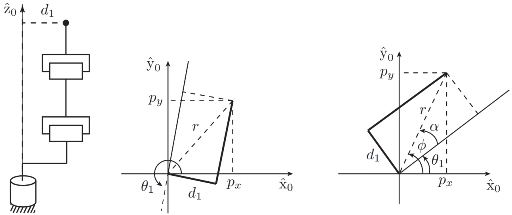
This allows to decompose the IK problem as

- Inverse position kinematics (for joints 1–3)
- Inverse orientation kinematics (for joints 4–6)

(Cont'd)

6.1. Analytic Inverse Kinematics

We consider two cases: without and with the offset d_1.



(a) Elbow arm with offset.

(b) Kinematic diagram.

Figure 6.3: A 6R PUMA-type arm with a shoulder offset.

- Case 1 ($d_1 = 0$): Projecting p onto the $\hat{x}_0 - -\hat{y}_0$-plane, it can be seen that

$$\theta_1 = \operatorname{atan2}(p_x, p_y) \text{ or } \theta_1 = \operatorname{atan2}(p_x, p_y) + \pi.$$

Since $\operatorname{atan2}(0, 0)$ is ill-defined, we have singularity at $p_x = p_y = 0$.

(Cont'd)

Singular configuration with zero-offset:

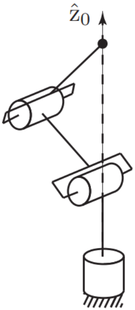
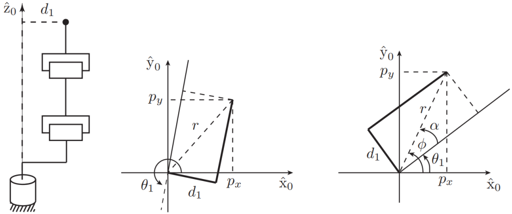


Figure 6.4: Singular configuration of the zero-offset 6R PUMA-type arm.

(Cont'd)

6.1. Analytic Inverse Kinematics



(a) Elbow arm with offset.

(b) Kinematic diagram.

Figure 6.3: A 6R PUMA-type arm with a shoulder offset.

- Case 2 ($d_1 \neq 0$): We have two general solutions:

$$\theta_1 = \phi - \alpha = \text{atan2}(p_y, p_x) - \text{atan2}(d_1, \sqrt{r^2 - d_1^2}), \quad \text{or}$$

$$\theta_1 = \pi + \text{atan2}(p_y, p_x) + \text{atan2}(-\sqrt{p_x^2 + p_y^2 - d_1^2}, d_1).$$

Note: Once θ_1 is determined, the remaining part can be viewed as a 2R planar robot (why?)

(Cont'd)

6.1. Analytic Inverse Kinematics

Four possible inverse kinematics solutions with shoulder offset:

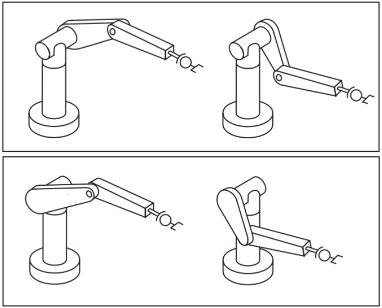


Figure 6.5: Four possible inverse kinematics solutions for the 6R PUMA-type arm with shoulder offset.

- $\exists\ 2$ cases in selecting θ_1,
- $\exists\ 2$ cases in selecting θ_2 and θ_3.

Newton-Raphson method

We want to solve the equation, for a differentiable function g,

$$g(\theta) = 0.$$

Note that the Taylor expansion of g at $\theta = \theta^0$ is given by

$$g(\theta) = g(\theta^0) + \frac{\partial g}{\partial \theta}(\theta^0)(\theta - \theta^0) + \underbrace{\text{H.O.T.}}_{=\text{ Higher-order terms}}$$

By truncating the H.O.T. and computing θ in an iterative fashion,

we have an iterative algorithm (under the assumption that $\partial g / \partial \theta$ invertible)

$$\theta^{k+1} = \theta^k - \left(\frac{\partial g}{\partial \theta}(\theta^k)\right)^{-1} g(\theta^k)$$

whose solution θ^k is expected to converge to the real solution of $g(\theta) = 0$.

Numerical IK via Newton-Raphson method

In the IK problem, we have

$$g(\theta_d) = x_d - f(\theta_d) = 0$$

where x_d is given and θ_d needs to be computed.

Similar computations as above lead to

$$\begin{aligned} x_d = f(\theta_d) &= f(\theta^0) + \frac{\partial f}{\partial \theta}(\theta^0)(\theta_d - \theta^0) + \text{H.O.T.} \\ &= f(\theta^0) + J(\theta^0)\Delta\theta + \text{H.O.T.} \end{aligned}$$

We thus have, with the zero H.O.T.,

$$J(\theta^0)\Delta\theta = x_d - f(\theta^0) \quad \Rightarrow \quad \Delta\theta = J(\theta^0)^{-1}\left(x_d - f(\theta^0)\right)$$

which can be solved by the Newton-Raphson method (when J is invertible!)

(Cont'd)

6.2. Numerical Inverse Kiematics

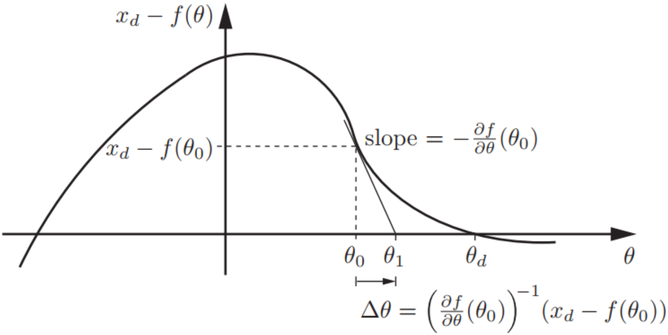


Figure 6.7: The first step of the Newton–Raphson method for nonlinear root-finding for a scalar x and θ. In the first step, the slope $-\partial f/\partial\theta$ is evaluated at the point $(\theta^0, x_d - f(\theta^0))$. In the second step, the slope is evaluated at the point $(\theta^1, x_d - f(\theta^1))$ and eventually the process converges to θ_d. Note that an initial guess to the left of the plateau of $x_d - f(\theta)$ would be likely to result in convergence to the other root of $x_d - f(\theta)$, and an initial guess at or near the plateau would result in a large initial $|\Delta\theta|$ and the iterative process might not converge at all.

When J is not invertible

6.2. Numerical Inverse Kiematics

We may have to use the (Moore-Penrose) pseudo-inverse $J^\dagger$ instead of J^{-1}

- If J is fat and has full rank, then
  - $J^\dagger = J^\top (JJ^\top)^{-1}$ (right-inverse)
  - there may exist several solutions $y^\star$ for $Jy^\star = z$
  - among the candidates, $y^\star = J^\dagger z$ has the minimum two-norm.
- If J is tall and has full rank, then
  - $J^\dagger = (J^\top J)^{-1} J^\top$ (left-inverse)
  - no solution exists for $Jy^\star = z$
  - among the candidates, $y^\star = J^\dagger z$ minimizes the two-norm error $\|Jy^\star - z\|$.

For both cases, we will employ

$$\Delta\theta = J(\theta^0)^\dagger \left(x_d - f(\theta^0)\right)$$

in solving the IK numerically.

Newton-Raphson iterative algorithm

6.2. Numerical Inverse Kiematics

Algorithm:

(a) (Initialization) Given x_d and initial guess θ^0, set $i = 0$.

(b) Set $e = x_d - f(\theta^i)$. While $\|e\| > \epsilon$ for some small ϵ:

- Set $\theta^{i+1} = \theta^i + J^\dagger(\theta^i)e$, and
- Increment i.

(A similar method can be derived for geometric Jacobian.)

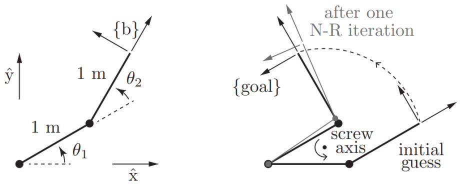


Figure 6.8: (Left) A 2R robot. (Right) The goal is to find the joint angles yielding the end-effector frame {goal} corresponding to $\theta_1 = 30°$ and $\theta_2 = 90°$. The initial guess is $(0°, 30°)$. After one Newton–Raphson iteration, the calculated joint angles are $(34.23°, 79.18°)$. The screw axis that takes the initial frame to the goal frame (by means of the curved dashed line) is also indicated.

Closed-loop IK

Supplementary material: (B. Siciliano *et al.*)

We now study another method for solving the IK with the differential kinematics.

In view of the differential kinematics, the IK problem can be understood as

For given x_d, find $\dot{\hat{\theta}}(\mathsf{t})$ such that

- $\hat{x}(\mathsf{t}) := f(\hat{\theta}(\mathsf{t})) \to x_d$ as $\mathsf{t} \to \infty$.
- $\hat{x}(\mathsf{t})$ satisfies $\dot{\hat{x}} = J_a(\hat{\theta})\dot{\hat{\theta}}$.

We now define the operational space error

$$e(\mathsf{t}) := \hat{x}(\mathsf{t}) - x_d$$

Then its time derivative is computed by

$$\dot{e} = J_a(\hat{\theta})\dot{\hat{\theta}} - \dot{x}_d$$

where $\dot{\hat{\theta}}$ is viewed as a virtual input to stabilize e.

Method 1: Jacobian (psuedo-)inverse

Supplementary material: (B. Siciliano *et al.*)

Suppose that $J_a(\hat{\theta})$ has full rank.

Then the Jacobian (psuedo-)inverse-based IK algorithm is given by

$$\dot{\hat{\theta}} = \begin{cases} J_a^{-1}(\dot{x}_d + He), & \text{if } J_a \text{ is square,} \\ J_a^{\dagger}(\dot{x}_d + He) + (I - J_a^{\dagger} J_a)\dot{\hat{\theta}}_{\text{red}}, & \text{if } J_a \text{ is fat} \end{cases}$$

where $\dot{\hat{\theta}}_{\text{red}}$ can be any vector with a proper dimension.

In both cases, we have

$$\dot{e} = He.$$

$\therefore$ If H is Hurwitz, then $e(\mathrm{t}) \to 0$ as $\mathrm{t} \to \infty$.

(For more details, we should study the Lyapunov stability analysis.)

Method 2: Jacobian transpose

Supplementary material: (B. Siciliano *et al.*)

Now suppose that $\dot{x}_d = 0$.

The Jacobian transpose-based IK algorithm is given by

$$\dot{\hat{\theta}} = J_a^\top(\hat{\theta}) P e$$

where P is a positive definite matrix.

It then follows that

$$V(e) := \frac{1}{2} e^\top P e \quad \Rightarrow \quad \dot{V} = -e^\top P J_a(\hat{\theta}) J_a(\hat{\theta})^\top P e \leq 0.$$

$\therefore$ Then the error $e(\mathrm{t})$ converges to 0 IF $\mathcal{N}(J_a^\top) = \emptyset$.

Note:

- Jacobian transpose-based algorithm does not require the inverse of J.
- If $Pe(\mathrm{t})$ eventually enters $\mathcal{N}(J_a^\top)$ (with $e(\mathrm{t}_0) \neq 0$), then $\hat{\theta}$ gets stuck (so that $e(\mathrm{t}) \neq 0$) for all $\mathrm{t} \geq \mathrm{t}_0$.

Orientation error?

Supplementary material: (B. Siciliano *et al.*)

Note: When we consider the error e, it must have the form

$$e := \begin{bmatrix} e_O \\ e_P \end{bmatrix} = \begin{bmatrix} \text{(Error in orientation)} \\ \text{(Error in position)} \end{bmatrix}.$$

- e_P can be easily expressed as $p_d - \hat{p}$.
- How about e_O???

From now on, we consider several types of the orientation error e_O, represented in

- Euler angles
- Angle and axis
- Unit quaternion (will be studied later...)

Orientation error 1: Euler angle

Supplementary material: (B. Siciliano *et al.*)

Consider

- $\phi_d \in \mathbb{R}^3$: Desired orientation of end-effector expressed in Euler angle;
- $\hat{\phi} \in \mathbb{R}^3$: Computed orientation of end-effector expressed in Euler angle.

Then we simply have

$$\text{(Orientation error)} \quad e_O = \phi_d - \hat{\phi},$$
$$\text{(Its time differentiation)} \quad \dot{e}_O = \dot{\phi}_d - \dot{\hat{\phi}}.$$

by which the Jacobian inverse method becomes

$$\dot{\hat{\theta}} = J_a^{-1}(\dot{x}_d + He) = J_a^{-1} \begin{bmatrix} \dot{\phi}_d + K_O(\phi_d - \hat{\phi}) \\ \dot{p}_d + K_P(p_d - \hat{p}) \end{bmatrix}.$$

Drawbacks? Possible singularity at some configurations

Orientation error 2: Angle and axis

Supplementary material: (B. Siciliano *et al.*)

Consider

- $R_d = \begin{bmatrix} n_d & s_d & a_d \end{bmatrix} \in \mathrm{SO}(3)$: Desired orientation of end-effector expressed in rotation matrix;
- $\hat{R} = \begin{bmatrix} n & s & a \end{bmatrix} \in \mathrm{SO}(3)$: Desired orientation of end-effector expressed in rotation matrix.

Then one can define another form of the orientation error as

$$e_O := \hat{r} \sin\vartheta \in \mathbb{R}^3 \quad \text{where } R_d \hat{R}^\top = \mathrm{Rot}(\hat{r}, \vartheta)$$

in which ϑ is constrained as $-\pi/2 \leq \vartheta \leq \pi/2$ for uniqueness.

Comparing the off-diagonal terms of $R_d \hat{R}^\top = \mathrm{Rot}(\hat{r}, \vartheta)$ and $R_d \hat{R}^\top$,

$$\text{(Orientation error)} \quad e_O = \frac{1}{2}\big(n \times n_d + s \times s_d + a \times a_d\big) = \frac{1}{2}\big([n]n_d + [s]s_d + [a]a_d\big)$$

(Cont'd)

Supplementary material: (B. Siciliano *et al.*)

Now, differentiating e_O (with $[a]b = -[b]a$ and $\dot{R} = [\omega]R$) gives to

$$\text{(Its time differentiation)} \quad \dot{e}_O = L^\top \omega_d - L\omega$$

where

$$L := -\frac{1}{2}\big([n_d][n] + [s_d][s] + [a_d][a]\big) \in \mathbb{R}^{3\times 3}.$$

Thus the time derivative of $e = (e_O, e_P)$ can be represented using the spatial manipulator Jacobian J_m as

$$\dot{e} = \begin{bmatrix} \dot{e}_O \\ \dot{e}_P \end{bmatrix} = \begin{bmatrix} L^\top \omega_d \\ \dot{p}_d \end{bmatrix} - \begin{bmatrix} L & 0 \\ 0 & I \end{bmatrix} J_m \dot{\hat{\theta}}$$

which introduces a Jacobian inverse method with a geometric Jacobian:

$$\dot{\hat{\theta}} = J_m^{-1} \begin{bmatrix} L^{-1}\big(L^\top \omega_d + K_O e_O\big) \\ \dot{p}_d + K_P e_P \end{bmatrix}.$$

Inverse differential (or velocity) kinematics

Supplementary material: (B. Siciliano *et al.*)

The problem of inverse differential kinematics is that,

for given θ_d and $\dot{x}_d$, find a solution $\dot{\theta}_d$ of the matrix equation $\dot{x}_d = J(\theta_d)\dot{\theta}_d$.

- If J is square and nonsingular, the solution is uniquely determined as

$$\dot{\theta}_d = J^{-1}\dot{x}_d.$$

- If J is fat or is singular, then the matrix equation may have infinitely many solutions.

  In this case, one can choose the optimal solution among candidates, which is the solution of the following minimization problem:

$$\begin{aligned} \min_{\dot{\theta}_d \in \mathbb{R}^n} \quad & \frac{1}{2}\dot{\theta}_d^\top W \dot{\theta}_d \\ \text{subject to} \quad & \dot{x}_d = J(\theta_d)\dot{\theta}_d \end{aligned}$$

  where W is positive definite.

The method of Lagrange multiplier

Supplementary material: (B. Siciliano *et al.*)

Consider a general equality-constrained minimization problem

$$\begin{aligned} \min_{\chi \in \mathbb{R}^n} \quad & f(\chi) \\ \text{subject to} \quad & g(\chi) = 0. \end{aligned}$$

The problem is in fact equivalent to an unconstrained minimization problem

$$\min_{(\chi,\lambda) \in \mathbb{R}^{n+m}} \quad \mathcal{L}(\chi, \lambda) := f(\chi) + \lambda^\top g(\chi)$$

where $\mathcal{L}$ is called the Lagrangian, and the variable λ is called the Lagrange multiplier.

Note that the solution of the second problem must satisfy

$$\frac{\partial \mathcal{L}}{\partial \chi}(\chi^\star, \lambda^\star) = 0, \quad \frac{\partial \mathcal{L}}{\partial \lambda}(\chi^\star, \lambda^\star) = g(\chi^\star) = 0.$$

By the latter, one can conclude that minimizing $\mathcal{L}$ is the same as that of f.

(Cont'd)

Supplementary material: (B. Siciliano *et al.*)

In our problem, the Lagrangian has the form

$$\mathcal{L}(\dot{\theta}_d, \lambda) = \frac{1}{2}\dot{\theta}_d^\top W \dot{\theta}_d + \lambda^\top \big(J(\theta_d)\dot{\theta}_d - x_d\big).$$

The necessary condition for optimality is then given by

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}_d}^\top = W\dot{\theta}_d + J(\theta_d)^\top \lambda = 0 \quad \Rightarrow \quad \dot{\theta}_d^\star = -W^{-1}J(\theta_d)^\top \lambda^\star.,$$

$$\frac{\partial \mathcal{L}}{\partial \lambda}^\top = J(\theta_d)\dot{\theta}_d - x_d = 0 \quad \Rightarrow \quad x_d = J\dot{\theta}_d^\star = -JW^{-1}J^\top \lambda^\star.$$

If J (is fat but) has full rank, then

$$\lambda^\star = -(JW^{-1}J^\top)^{-1}x_d \quad \Rightarrow \quad \dot{\theta}_d^\star = W^{-1}J^\top(JW^{-1}J^\top)^{-1}x_d.$$

Note: The solution for the special case with $W = I$ is

$\dot{\theta}_d^\star = J^\top(JJ^\top)^{-1}x_d = J^\dagger x_d$ where $J^\dagger$ is Penrose-Moore psuedo-inverse.