Robotics

PS05 Solution

1 Problem

Given the planar (2D) robot arm from figure 1 with 3 DoF:

- a rotational joint in the origin of the world frame with DoF α_1,
- followed by a fixed link of length $l_1 = 10$ with rotational joint at its end with DoF α_2,
- and a prismatic joint linked to it with the DoF l_2 with $l_2 \in [5, 10]$, which is co-aligned with l_1 for $\alpha_2 = 0^o$ (see figure 1, right).

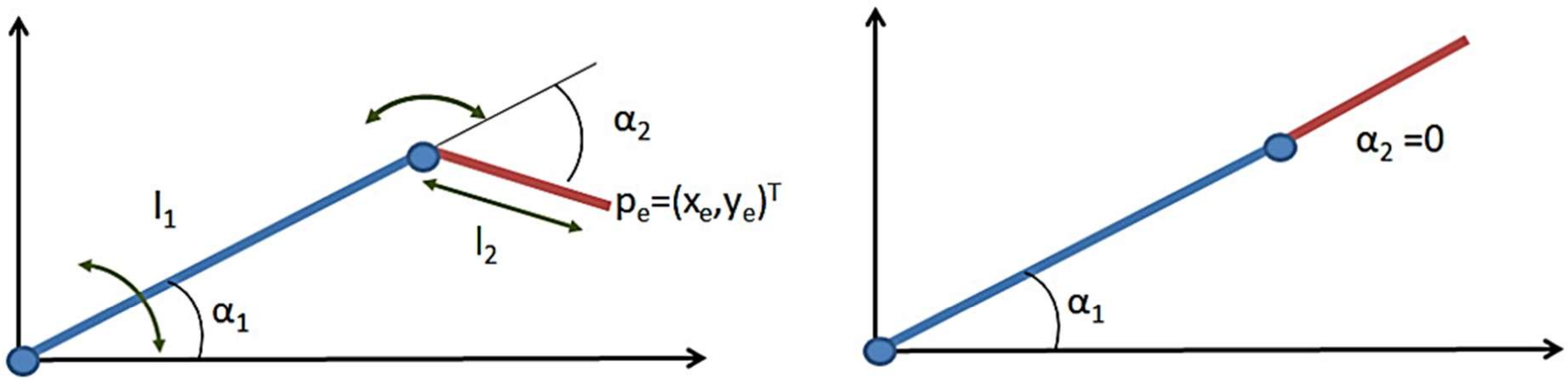
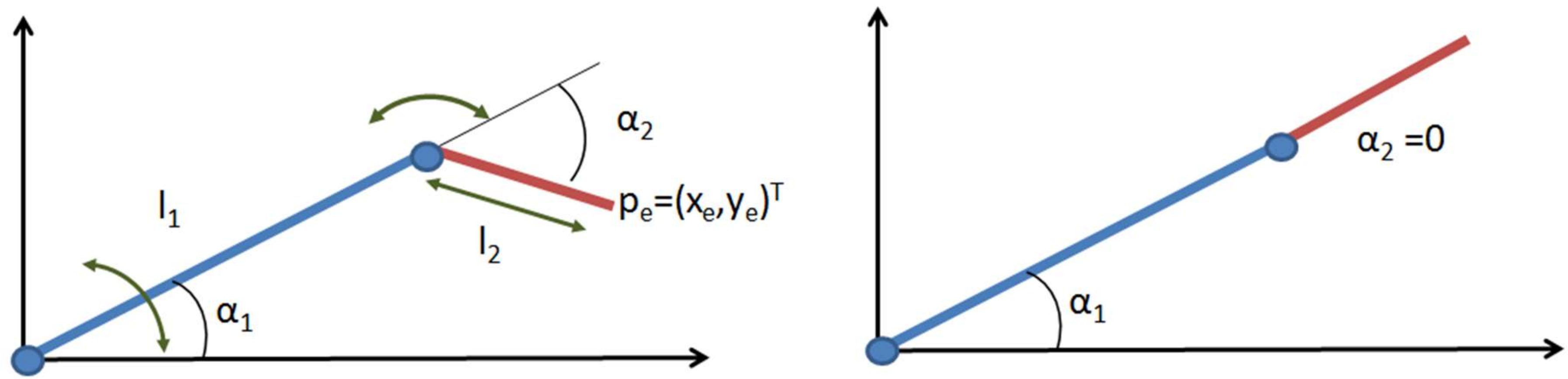


Figure 1: A planar robot arm with 3 DoF. The alignment of the prismatic joint l_2 for $\alpha_2 = 0^o$ is shown on the right.

Provide the forward kinematics for the position $p_e = (x_e, y_e)$ of the end-effector of this robot.

Problem 1



$$p_e = \begin{pmatrix} x_e \\ y_e \\ 1 \end{pmatrix} = {}^{F_0}R(\alpha_1)\,{}^{F_0}T(l_1)\,{}^{F_0}R(\alpha_2)\,{}^{F_0}T(l_2)\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

note: in general, swap the order of the transformations to be in the world frame

$${}^{F_n}p_e = \ \ldots {}^{F_1}_{F_2}B\,{}^{F_0}_{F_1}A\,{}^{F_0}o \ \Rightarrow \ {}^{F_0}p_e = {}^{F_0}A\,{}^{F_0}B\,{}^{F_0}\ldots o$$

Problem 1

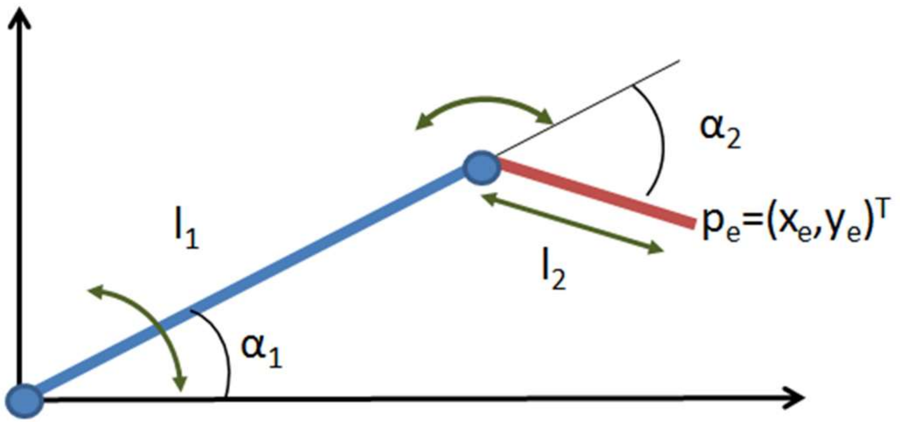
$$\begin{pmatrix} x_e \\ y_e \\ 1 \end{pmatrix} = {}^{F_0}R(\alpha_1)\,{}^{F_0}T(l_1)\,{}^{F_0}R(\alpha_2)\,{}^{F_0}T(l_2)\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} c\alpha_1 & -s\alpha_1 & 0 \\ s\alpha_1 & c\alpha_1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 10 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} c\alpha_2 & -s\alpha_2 & 0 \\ s\alpha_2 & c\alpha_2 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} c\alpha_1 & -s\alpha_1 & c\alpha_1 \cdot 10 \\ s\alpha_1 & c\alpha_1 & s\alpha_1 \cdot 10 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} c\alpha_2 & -s\alpha_2 & c\alpha_2 \cdot l_2 \\ s\alpha_2 & c\alpha_2 & s\alpha_2 \cdot l_2 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} c\alpha_1 & -s\alpha_1 & c\alpha_1 \cdot 10 \\ s\alpha_1 & c\alpha_1 & s\alpha_1 \cdot 10 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} c\alpha_2 \cdot l_2 \\ s\alpha_2 \cdot l_2 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} c\alpha_1 c\alpha_2 \cdot l_2 - s\alpha_1 s\alpha_2 \cdot l_2 + c\alpha_1 \cdot 10 \\ s\alpha_1 c\alpha_2 \cdot l_2 + c\alpha_1 s\alpha_2 \cdot l_2 + s\alpha_1 \cdot 10 \\ 1 \end{pmatrix}$$



2 Problem

Take the robot's forward kinematics from the previous problem.

- Find the related Jacobian matrix J.
- Which options do you know to compute the pseudo-inverse J^+ of J, and when are they applicable?
- Given the goal position $p_e(n_g) = (5, 10)$ and the starting DoF values $\alpha_1(0) = 90^o, \alpha_2(0) = 0^o, l_2(0) = 8$, formulate the numerical IK with a) Newton's method, respectively b) Gradient descent.
- How can we formulate the IK problem if the full pose $p'_e = (x_e, y_e, \theta_e)$ is to be found?

2 Problem

Take the robot's forward kinematics from the previous problem.

Find the related Jacobian matrix J.

- Which options do you know to compute the pseudo-inverse J^+ of J, and when are they applicable?
- Given the goal position $p_e(n_g) = (5, 10)$ and the starting DoF values $\alpha_1(0) = 90^o, \alpha_2(0) = 0^o, l_2(0) = 8$, formulate the numerical IK with a) Newton's method, respectively b) Gradient descent.
- How can we formulate the IK problem if the full pose $p'_e = (x_e, y_e, \theta_e)$ is to be found?

Problem 2: Jacobian

$$K(\alpha_1,\alpha_2,l_2)=\begin{pmatrix} x_e \\ y_e \end{pmatrix}=\begin{pmatrix} c\alpha_1 c\alpha_2\cdot l_2 - s\alpha_1 s\alpha_2\cdot l_2 + c\alpha_1\cdot 10 \\ s\alpha_1 c\alpha_2\cdot l_2 + c\alpha_1 s\alpha_2\cdot l_2 + s\alpha_1\cdot 10 \end{pmatrix}$$

$$J = DK(\alpha_1,\alpha_2,l_2) = \begin{pmatrix} \frac{\partial K_x}{\partial \alpha_1} & \frac{\partial K_x}{\partial \alpha_2} & \frac{\partial K_x}{\partial l_2} \\ \frac{\partial K_y}{\partial \alpha_1} & \frac{\partial K_y}{\partial \alpha_2} & \frac{\partial K_y}{\partial l_2} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{c\alpha_1 c\alpha_2\cdot l_2 - s\alpha_1 s\alpha_2\cdot l_2 + c\alpha_1\cdot 10}{\partial \alpha_1} & \frac{c\alpha_1 c\alpha_2\cdot l_2 - s\alpha_1 s\alpha_2\cdot l_2 + c\alpha_1\cdot 10}{\partial \alpha_2} & \frac{c\alpha_1 c\alpha_2\cdot l_2 - s\alpha_1 s\alpha_2\cdot l_2 + c\alpha_1\cdot 10}{\partial l_2} \\ \frac{s\alpha_1 c\alpha_2\cdot l_2 + c\alpha_1 s\alpha_2\cdot l_2 + s\alpha_1\cdot 10}{\partial \alpha_1} & \frac{s\alpha_1 c\alpha_2\cdot l_2 + c\alpha_1 s\alpha_2\cdot l_2 + s\alpha_1\cdot 10}{\partial \alpha_2} & \frac{s\alpha_1 c\alpha_2\cdot l_2 + c\alpha_1 s\alpha_2\cdot l_2 + s\alpha_1\cdot 10}{\partial l_2} \end{pmatrix}$$

$$= \begin{pmatrix} -s\alpha_1 c\alpha_2\cdot l_2 - c\alpha_1 s\alpha_2\cdot l_2 - s\alpha_1\cdot 10 & -c\alpha_1 s\alpha_2\cdot l_2 - s\alpha_1 c\alpha_2\cdot l_2 & c\alpha_1 c\alpha_2 - s\alpha_1 s\alpha_2 \\ c\alpha_1 c\alpha_2\cdot l_2 - s\alpha_1 s\alpha_2\cdot l_2 + c\alpha_1\cdot 10 & -s\alpha_1 s\alpha_2\cdot l_2 + c\alpha_1 c\alpha_2\cdot l_2 & s\alpha_1 c\alpha_2 + c\alpha_1 s\alpha_2 \end{pmatrix}$$

Problem 2 : Jacobian

$$K(\alpha_1,\alpha_2,l_2)=\begin{pmatrix}x_e\\ y_e\end{pmatrix}=\begin{pmatrix}c\alpha_1 c\alpha_2\cdot l_2 - s\alpha_1 s\alpha_2\cdot l_2 + c\alpha_1\cdot 10\\ s\alpha_1 c\alpha_2\cdot l_2 + c\alpha_1 s\alpha_2\cdot l_2 + s\alpha_1\cdot 10\end{pmatrix}$$

$$J = DK(\alpha_1,\alpha_2,l_2) =$$

$$\begin{pmatrix}-s\alpha_1 c\alpha_2 l_2 - c\alpha_1 s\alpha_2 l_2 - 10 s\alpha_1 & -c\alpha_1 s\alpha_2 l_2 - s\alpha_1 c\alpha_2 l_2 & c\alpha_1 c\alpha_2 - s\alpha_1 s\alpha_2\\ c\alpha_1 c\alpha_2 l_2 - s\alpha_1 s\alpha_2 l_2 + 10 c\alpha_1 & -s\alpha_1 s\alpha_2 l_2 + c\alpha_1 c\alpha_2 l_2 & s\alpha_1 c\alpha_2 + c\alpha_1 s\alpha_2\end{pmatrix}$$

2 Problem

Take the robot's forward kinematics from the previous problem.

- Find the related Jacobian matrix J.
- Which options do you know to compute the pseudo-inverse J^+ of J, and when are they applicable?
- Given the goal position $p_e(n_g) = (5, 10)$ and the starting DoF values $\alpha_1(0) = 90^o, \alpha_2(0) = 0^o, l_2(0) = 8$, formulate the numerical IK with a) Newton's method, respectively b) Gradient descent.
- How can we formulate the IK problem if the full pose $p'_e = (x_e, y_e, \theta_e)$ is to be found?

Problem 2 : Pseudo-Inverse (Option 1)

f(): $\mathbb{R}^n \rightarrow \mathbb{R}^m$, Jacobian Df() *mxn* matrix: either

- m>n
  =>linearly independent columns
  => A^TA is invertible (left Ps.Inv. $A^+A=I$)
- m<n
  => linearly independent rows
  => AA^T is invertible (right Ps.Inv. $AA^+=I$)

left: $A^+=(A^TA)^{-1}A^T$

right: $A^+=A^T(AA^T)^{-1}$

Problem 2 : Pseudo-Inverse (Option 1)

here: K(): $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ $\quad K(\alpha_1,\alpha_2,l_2)=\begin{pmatrix} x_e \\ y_e \end{pmatrix}$

Jacobian **DK()** = **A** (*2x3* matrix)

=> linearly independent rows

=> AA^T is invertible (**right Ps.Inv.** $AA^+=I$) $\quad A^+=A^T(AA^T)^{-1}$

$$DK(\alpha_1,\alpha_2,l_2)^+ = DK(\alpha_1,\alpha_2,l_2)^T(DK(\alpha_1,\alpha_2,l_2)DK(\alpha_1,\alpha_2,l_2)^T)^{-1} =$$

$$\begin{pmatrix} -s\alpha_1 c\alpha_2 l_2 - c\alpha_1 s\alpha_2 l_2 - 10 s\alpha_1 & c\alpha_1 c\alpha_2 l_2 - s\alpha_1 s\alpha_2 \cdot l_2 + 10 c\alpha_1 \\ -c\alpha_1 s\alpha_2 l_2 - s\alpha_1 c\alpha_2 l_2 & -s\alpha_1 s\alpha_2 l_2 + c\alpha_1 c\alpha_2 l_2 \\ c\alpha_1 c\alpha_2 - s\alpha_1 s\alpha_2 & s\alpha_1 c\alpha_2 + c\alpha_1 s\alpha_2 \end{pmatrix} \cdot$$

$$\left(\begin{pmatrix} -s\alpha_1 c\alpha_2 l_2 - c\alpha_1 s\alpha_2 l_2 - 10 s\alpha_1 & -c\alpha_1 s\alpha_2 l_2 - s\alpha_1 c\alpha_2 l_2 & c\alpha_1 c\alpha_2 - s\alpha_1 s\alpha_2 \\ c\alpha_1 c\alpha_2 l_2 - s\alpha_1 s\alpha_2 l_2 + 10 c\alpha_1 & -s\alpha_1 s\alpha_2 l_2 + c\alpha_1 c\alpha_2 l_2 & s\alpha_1 c\alpha_2 + c\alpha_1 s\alpha_2 \end{pmatrix} \cdot \begin{pmatrix} -s\alpha_1 c\alpha_2 l_2 - c\alpha_1 s\alpha_2 l_2 - 10 s\alpha_1 & c\alpha_1 c\alpha_2 l_2 - s\alpha_1 s\alpha_2 l_2 + 10 c\alpha_1 \\ -c\alpha_1 s\alpha_2 l_2 - s\alpha_1 c\alpha_2 l_2 & -s\alpha_1 s\alpha_2 l_2 + c\alpha_1 c\alpha_2 l_2 \\ c\alpha_1 c\alpha_2 - s\alpha_1 s\alpha_2 & s\alpha_1 c\alpha_2 + c\alpha_1 s\alpha_2 \end{pmatrix}\right)^{-1}$$

Problem 2 : Pseudo-Inverse (Option 1)

$$DK(\alpha_1,\alpha_2,l_2)^+ = DK(\alpha_1,\alpha_2,l_2)^T\left(DK(\alpha_1,\alpha_2,l_2)DK(\alpha_1,\alpha_2,l_2)^T\right)^{-1}$$

works "always", i.e., pseudo-inverse is a fct $DK^+(\alpha_1, \alpha_2, l_2)$

- but computationally quite complex
- unless some effort spend to derive simpler form (multiply matrices out, use trigonometric laws, etc.)

Problem 2 : Pseudo-Inverse (Option 1)

option 1 with concrete values: step 1, the Jacobian

$$DK(\alpha_1 = 90^o, \alpha_2 = 0^o, l_2 = 8)$$

$$= \begin{pmatrix} -s(90^o)c(0^o)\cdot 8 - c(90^o)s(0^o)\cdot 8 - s(90^o)\cdot 10 & -c(90^o)s(0^o)\cdot 8 - s(90^o)c(0^o)\cdot 8 & c(90^o)c(0^o) - s(90^o)s(0^o) \\ c(90^o)c(0^o)\cdot 8 - s(90^o)s(0^o)\cdot 8 + c(90^o)\cdot 10 & -s(90^o)s(0^o)\cdot 8 + c(90^o)c(0^o)\cdot 8 & s(90^o)c(0^o) + c(90^o)s(0^o) \end{pmatrix}$$

$$= \begin{pmatrix} -8 - 0 - 10 & 0 - 8 & 0 - 0 \\ 0 - 0 + 0 & 0 + 0 & 1 + 0 \end{pmatrix}$$

$$= \begin{pmatrix} -18 & -8 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

rows of this “wide” Jacobian A are linearly independent

=> $A^+ = A^T(AA^T)^{-1}$

Problem 2 : Pseudo-Inverse (Option 1)

$$DK(\alpha_1 = 90^o, \alpha_2 = 0^o, l_2 = 8) = \begin{pmatrix} -18 & -8 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

rows linearly independent
=> $A^+ = A^T(AA^T)^{-1}$

$$DK^+(\alpha_1 = 90^o, \alpha_2 = 0^o, l_2 = 8)$$

$$= \begin{pmatrix} -18 & 0 \\ -8 & 0 \\ 0 & 1 \end{pmatrix} \left(\begin{pmatrix} -18 & -8 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -18 & 0 \\ -8 & 0 \\ 0 & 1 \end{pmatrix} \right)^{-1} = \begin{pmatrix} -18 & 0 \\ -8 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 388 & 0 \\ 0 & 1 \end{pmatrix}^{-1}$$

$$= \begin{pmatrix} -18 & 0 \\ -8 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{388} & 0 \\ 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} -0.04644 & 0 \\ -0.02064 & 0 \\ 0 & 1 \end{pmatrix}$$

example with
fixed values for
input DoF

Problem 2 : Pseudo-Inverse (Option 2)

$$DK(\alpha_1 = 90^o, \alpha_2 = 0^o, l_2 = 8) = \begin{pmatrix} -18 & -8 & 0 \\ 0 & 0 & 1 \end{pmatrix} = UWV^T$$

can be used when fixed DoF values are given

$$U = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$W = \begin{pmatrix} 19.698 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$$V^T = \begin{pmatrix} -0.914 & -0.406 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

singular value decomposition (SVD)

Problem 2 : Pseudo-Inverse (Option 2)

$$DK^{+}(\alpha_1 = 90^o, \alpha_2 = 0^o, l_2 = 8) = VW^{+}U^{T}$$

$$= \begin{pmatrix} -0.914 & -0.406 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}^{T} \begin{pmatrix} 19.698 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^{+} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}^{T}$$

$$= \begin{pmatrix} -0.914 & 0 & 0 \\ -0.406 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0.05077 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} -0.04640 & 0 \\ -0.02061 & 0 \\ 0 & 1 \end{pmatrix}$$

Problem 2 : Pseudo-Inverse (Option 2)

given: $$DK() = \begin{pmatrix} -18 & -8 & 0 \\ 0 & 0 & 1 \end{pmatrix} = UWV^T$$

$$U = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$W = \begin{pmatrix} 19.698 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$$V^T = \begin{pmatrix} -0.914 & -0.406 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

provide pseudo-inverse as product of 3 simple matrices (no transpose, etc.):

$$DK^+() = VW^+U^T = \begin{pmatrix} -0.914 & 0 & 0 \\ -0.406 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0.05077 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Problem 2 : Pseudo-Inverse

option 1: $A^{+} = A^{T}(AA^{T})^{-1}$

$$DK^{+}(q) = \begin{pmatrix} -0.04644 & 0 \\ -0.02064 & 0 \\ 0 & 1 \end{pmatrix}$$

works "always", i.e., pseudo-inverse can be derived as fct $DK^{+}(q)$

option 2: SVD $A^{+} = VS^{+}U^{T}$

$$DK^{+}(q) = \begin{pmatrix} -0.04640 & 0 \\ -0.02061 & 0 \\ 0 & 1 \end{pmatrix}$$

can be used when fixed DoF values are given

(note: did some rounding in calculations for both options)

2 Problem

Take the robot's forward kinematics from the previous problem.

- Find the related Jacobian matrix J.
- Which options do you know to compute the pseudo-inverse J^+ of J, and when are they applicable?
- Given the goal position $p_e(n_g) = (5, 10)$ and the starting DoF values $\alpha_1(0) = 90^o, \alpha_2(0) = 0^o, l_2(0) = 8$, formulate the numerical IK with a) Newton's method, respectively b) Gradient descent.
- How can we formulate the IK problem if the full pose $p'_e = (x_e, y_e, \theta_e)$ is to be found?

Problem 2 : Newton IK

$$q(k+1) = q(k) + \alpha \cdot \Delta q$$

$$\Delta q = J(q(k))^{-1/T/+} \left[t - K(q(k)) \right]$$

start:

$$q_0 = \begin{pmatrix} \alpha_1(0) \\ \alpha_2(0) \\ l_2(0) \end{pmatrix} = \begin{pmatrix} 90^o \\ 0^o \\ 8 \end{pmatrix}$$

forward: from P1

$$K(\alpha_1, \alpha_2, l_2) = \begin{pmatrix} x_e \\ y_e \end{pmatrix}$$

$$= \begin{pmatrix} c\alpha_1 c\alpha_2 \cdot l_2 - s\alpha_1 s\alpha_2 \cdot l_2 + c\alpha_1 \cdot 10 \\ s\alpha_1 c\alpha_2 \cdot l_2 + c\alpha_1 s\alpha_2 \cdot l_2 + s\alpha_1 \cdot 10 \end{pmatrix}$$

Problem 2 : Newton IK

$$q(1) = \begin{pmatrix} 90^o \\ 0^o \\ 8 \end{pmatrix} + \alpha \cdot \Delta q, \ \Delta q = J(q(0))^+ \left[\begin{pmatrix} 5 \\ 10 \end{pmatrix} - \begin{pmatrix} 0 \\ 18 \end{pmatrix} \right] = J(q(0))^+ \begin{pmatrix} 5 \\ -8 \end{pmatrix}$$

goal target:

$$t = p_e(n_g) = \begin{pmatrix} 5 \\ 10 \end{pmatrix}$$

forward at start:

$$K(q(0)) = \begin{pmatrix} c(90^o)c(0^o) \cdot 8 - s(90^o)s(0^o) \cdot 8 + c(90^o) \cdot 10 \\ s(90^o)c(0^o) \cdot 8 + c(90^o)s(0^o) \cdot 8 + s(90^o) \cdot 10 \end{pmatrix}$$

$$= \begin{pmatrix} 0 - 0 + 0 \\ 8 + 0 + 10 \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ 18 \end{pmatrix}$$

Problem 2 : Newton IK

$$q(1) = \begin{pmatrix} 90^o \\ 0^o \\ 8 \end{pmatrix} + \alpha \cdot \Delta q$$

pseudo-inverse of Jacobian at start:

$$J^+(q(0)) = \begin{pmatrix} -0.04644 & 0 \\ -0.02064 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\Delta q = \begin{pmatrix} -0.04644 & 0 \\ -0.02064 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 5 \\ -8 \end{pmatrix} = \begin{pmatrix} -0.23220 \\ -0.10320 \\ 8 \end{pmatrix}$$

Problem 2 : Newton IK

$$q(1) = \begin{pmatrix} 90^o \\ 0^o \\ 8 \end{pmatrix} + \alpha \cdot \Delta q$$

pseudo-inverse of Jacobian at start:

$$J^+(q(0)) = \begin{pmatrix} -0.04644 & 0 \\ -0.02064 & 0 \\ 0 & 1 \end{pmatrix}$$

note: the angular changes are in radians!!!

$$\Delta q = \begin{pmatrix} -0.23220 \\ -0.10320 \\ 8 \end{pmatrix} = \begin{pmatrix} -0.23220 / \pi \cdot 180^o \\ -0.10320 / \pi \cdot 180^o \\ 8 \end{pmatrix} = \begin{pmatrix} -13.30^o \\ -5.91^o \\ 8 \end{pmatrix}$$

Problem 2 : Newton IK

e.g.,$\alpha = 0.1$:

$$q(1) = \begin{pmatrix} 90^o \\ 0^o \\ 8 \end{pmatrix} + 0.1 \cdot \begin{pmatrix} -13.30^o \\ -5.91^o \\ 8 \end{pmatrix} = \begin{pmatrix} 90^o \\ 0^o \\ 8 \end{pmatrix} + \begin{pmatrix} -1.330^o \\ -0.591^o \\ 0.8 \end{pmatrix}$$

$$= \begin{pmatrix} 88.670^o \\ 359.319^o \\ 8.8 \end{pmatrix}$$

$$q(k+1) = q(k) + \alpha \cdot \Delta q$$

keep on iterating...

$$\Delta q = J(q(k))^{-1/T/+} \left[t - K(q(k)) \right]$$

Problem 2 : Newton IK

$$q(1) = \begin{pmatrix} 88.670^o \\ 359.319^o \\ 8.8 \end{pmatrix}$$

$$q(k+1) = q(k) + \alpha \cdot \Delta q$$

$$\Delta q = J(q(k))^{-1/T/+} \left[t - K(q(k)) \right]$$

keep on iterating:

- forward kinematics of q(1)
- new Jacobian at q(1) (i.e., new SVD, etc.)

$\Rightarrow$q(2)

and so on... (until small error to target) $t - K(q(n)) < \varepsilon$

Problem 2 : Gradient Descent

for IK, minimize error function E(q)

$$E(q) = \frac{1}{2}\,|\,t - K(q)\,|^2 = \frac{1}{2}\left[t - K(q)\right]\left[t - K(q)\right]^T$$

gradient of E(q)

$$\nabla E(q) = -J(q)^T\left[t - K(q)\right]$$

just like "dirty" Newton

Problem 2 : Gradient Descent

excursus

some notes
on the error function in numerical IK
using gradient descent

Problem 2 : Gradient Descent

notes for error fct

$\frac{1}{2} | t - K(q) |^2$

- $| x |^2 = xx^T$
- ½ for “easy” derivative

$$| x |^2 = \left(\sqrt{x_1{}^2 + x_2{}^2 + \ldots + x_n{}^2} \right)^2$$

$$= \left(x_1{}^2 + x_2{}^2 + \ldots + x_n{}^2 \right)$$

$$= \left(x_1, x_2, \ldots, x_n \right) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

$$= xx^T$$

P2: derivation gradient of E(q)

f,g: $\mathbb{R} \to \mathbb{R}$ chain rule $$(f \circ g)' = (f' \circ g) \cdot g'$$

f,g: $\mathbb{R}^n \to \mathbb{R}^m$ Jacobian chain rule

$$J_{f \circ g}(x) = J_f(g(x)) \cdot J_g(x)$$

P2: derivation gradient of E(q)

for IK: $E(q) = \frac{1}{2} | t - K(q) |^2$

$$J_{f \circ g}(x) = J_f(g(x)) \cdot J_g(x)$$

$$f(x) = \frac{1}{2} | x |^2$$

$$g(x) = t - K(x)$$

P2: derivation gradient of E(q)

Jacobian of (Euclidean aka L_2) vector norm squared

$$J_{\|^2} = D(|x|^2) = D(x^T x) = 2x^T$$

hence $$J_f = D(\tfrac{1}{2}|x|^2) = x^T$$

furthermore $$J_g = D(t - K(x)) = -J(x)$$

note: “default” J

always the IK Jacobian

P2: derivation gradient of E(q)

$$E(q) = \tfrac{1}{2} \, | t - K(q) |^2$$

$$J_{E(x)} = J_{f \circ g}(x) = J_f(g(x)) \cdot J_g(x)$$

with $J_f = D(\tfrac{1}{2} \, | x |^2) = x^T$ $\qquad g(x) = t - K(x)$ $\qquad J_g = -J(x)$

$$J_{E(q)} = \left(t - K(q) \right)^T \cdot -J(q)$$

P2: derivation gradient of E(q)

Jacobian of E(q)

$$J_{E(q)} = \left(t - K(q)\right)^T \cdot -J(q)$$

gradient of E(q)

$$\nabla E(q) = J_{E(q)}^T = \left(\left(t - K(q)\right)^T \cdot -J(q)\right)^T$$

$$= -J(q)^T \cdot \left(t - K(q)\right)^T$$

Problem 2 : Gradient Descent

end of the excursus

Problem 2 : Gradient Descent

iteration

$$q_{k+1} = q_k - \alpha \nabla E(q_k) = q_k + \alpha J(q_k)^T [t - K(q_k)]$$

$$q(0) = \begin{pmatrix} 90^o \\ 0^o \\ 8 \end{pmatrix}, \text{ target: } t = \begin{pmatrix} 5 \\ 10 \end{pmatrix}, \text{ forward: } K(q(0)) = \begin{pmatrix} 0 \\ 18 \end{pmatrix}$$

$$q(1) = \begin{pmatrix} 90^o \\ 0^o \\ 8 \end{pmatrix} + \alpha \cdot J(q(0))^T \left[\begin{pmatrix} 5 \\ 10 \end{pmatrix} - \begin{pmatrix} 0 \\ 18 \end{pmatrix} \right]$$

Problem 2 : Gradient Descent

$$J(q(0)) = DK(\alpha_1 = 90^o, \alpha_2 = 0^o, l_2 = 8) = \begin{pmatrix} -18 & -8 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; J(q(0))^T = \begin{pmatrix} -18 & 0 \\ -8 & 0 \\ 0 & 1 \end{pmatrix}$$

$$q(1) = \begin{pmatrix} 90^o \\ 0^o \\ 8 \end{pmatrix} + 0.01 \cdot \begin{pmatrix} -18 & 0 \\ -8 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 5 \\ -8 \end{pmatrix} = \begin{pmatrix} 90^o \\ 0^o \\ 8 \end{pmatrix} + \begin{pmatrix} -0.9 \cdot \frac{180^o}{\pi} \\ -0.4 \cdot \frac{180^o}{\pi} \\ -0.08 \end{pmatrix}$$

$$= \begin{pmatrix} 90^o \\ 0^o \\ 8 \end{pmatrix} + \begin{pmatrix} -51.57^o \\ -22.92^o \\ -0.08 \end{pmatrix} = \begin{pmatrix} 38.43^o \\ 337.08^o \\ 7.92 \end{pmatrix}$$

and so on...

2 Problem

Take the robot's forward kinematics from the previous problem.

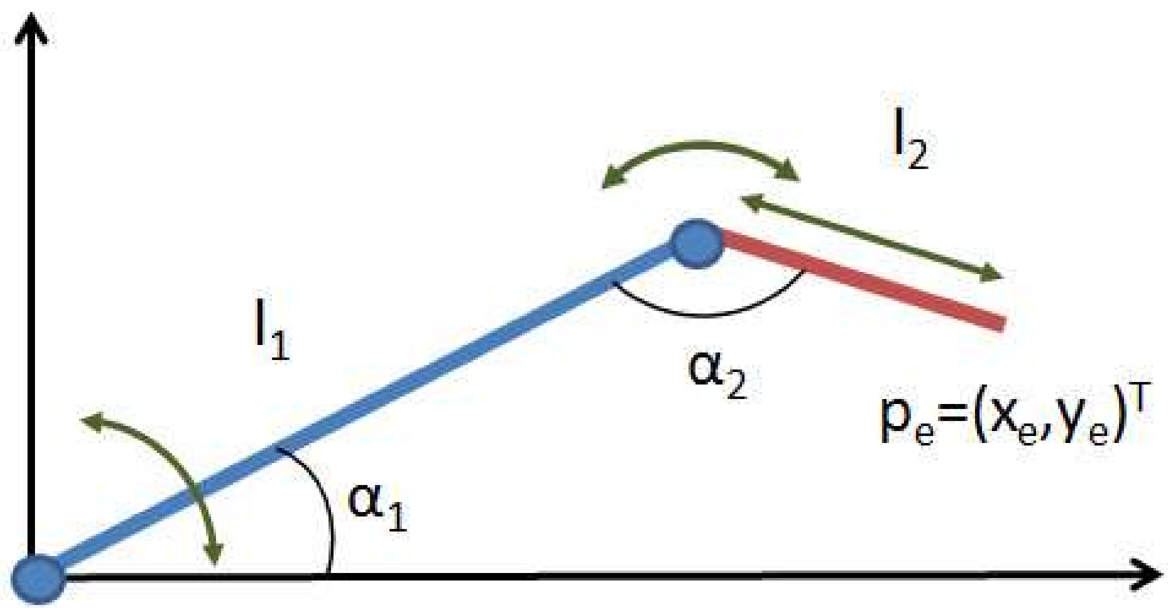
- Find the related Jacobian matrix J.
- Which options do you know to compute the pseudo-inverse J^+ of J, and when are they applicable?
- Given the goal position $p_e(n_g) = (5, 10)$ and the starting DoF values $\alpha_1(0) = 90^o, \alpha_2(0) = 0^o, l_2(0) = 8$, formulate the numerical IK with a) Newton's method, respectively b) Gradient descent.
- How can we formulate the IK problem if the full pose $p_e' = (x_e, y_e, \theta_e)$ is to be found?

Problem 2 : Orientation

simple in 2D

- translational joints do not affect the orientation
- end-effector orientation is sum of joint angles

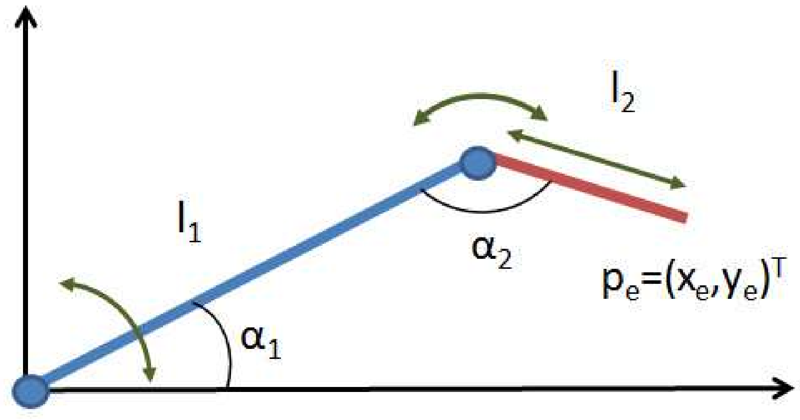
$$\alpha_e = f(\alpha_1, \ldots, \alpha_n)$$
$$= \alpha_1 + \ldots + \alpha_n$$



Problem 2 : Orientation

2D pose (not only location) forward kinematics K(q)

- has just 1 more component in result vector (number of input DoF stays the same)
- computation of location part as before



$$K(\alpha_1,\alpha_2,l_2)=\begin{pmatrix} x_e \\ y_e \\ \alpha_e \end{pmatrix}$$

$$=\begin{pmatrix} c\alpha_1 c\alpha_2 \cdot l_2 - s\alpha_1 s\alpha_2 \cdot l_2 + c\alpha_1 \cdot 10 \\ s\alpha_1 c\alpha_2 \cdot l_2 + c\alpha_1 s\alpha_2 \cdot l_2 + s\alpha_1 \cdot 10 \\ \alpha_1 + \alpha_2 \end{pmatrix}$$

Problem 2 : Orientation

in 3D a bit more complicated: several options, e.g.,

1. Euler angles as FK result components
   - use homogeneous matrices to derive K()
   - and conversion of rotation matrix to Euler
2. add rotation matrix components to FK result
3. use quaternions
   - to calculate forward kinematics orientation
   - and as part of the result of K()

Problem 2 : Orientation

in 3D a bit more complicated: several options, e.g.,

1. Euler angles as FK result components
2. add rotation matrix components to FK result
3. use quaternions

$$K_1(q) = \begin{pmatrix} x \\ y \\ z \\ \alpha_r \\ \alpha_p \\ \alpha_y \end{pmatrix} \qquad K_2(q) = \begin{pmatrix} x \\ y \\ z \\ r_{1.1} \\ r_{1.2} \\ r_{1.3} \\ r_{2.1} \\ \vdots \\ r_{3.3} \end{pmatrix} \qquad K_3(q) = \begin{pmatrix} x \\ y \\ z \\ q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix}$$