CONSTRUCTOR UNIVERSITY

Robotics

PS05 – Solutions

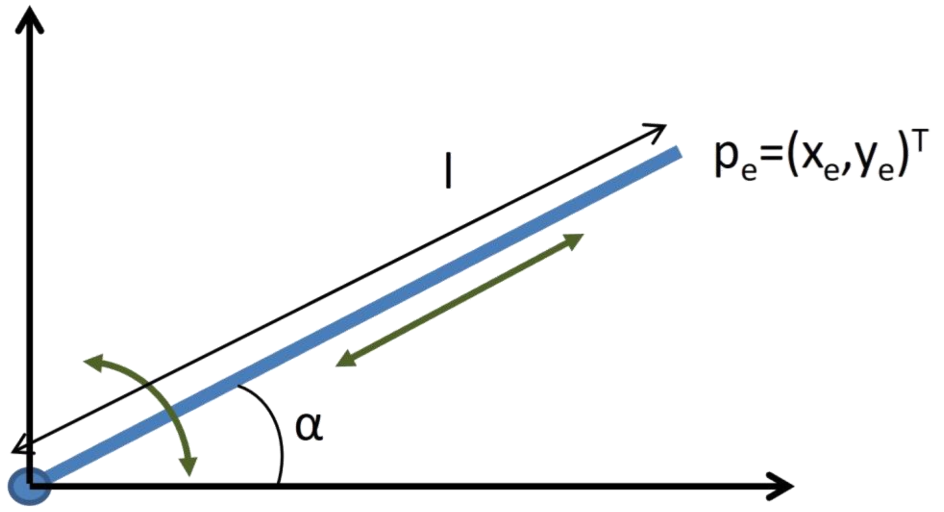
Andreas Birk

Constructor University

Part 5: Inverse Kinematics

Problem 1

Take the planar arm from **PS03, Problem 1** with a rotational joint and a prismatic joint linked to it with the DoF α and l.



Use its forward kinematics to find

- the proper Jacobian matrix J, respectively
- the numerical approximation of J at point $(1,2)$ with $\delta = 0.1$

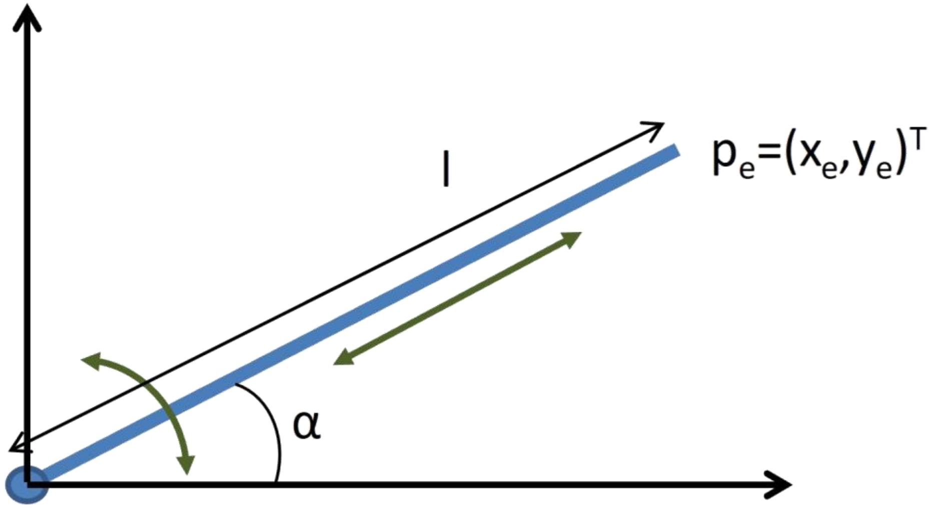
as basis for inverse kinematics.

Problem 1

$$\begin{pmatrix} x_e \\ y_e \\ 1 \end{pmatrix} = \begin{pmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} c\alpha & -s\alpha & c\alpha \cdot l \\ s\alpha & c\alpha & s\alpha \cdot l \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} c\alpha \cdot l \\ s\alpha \cdot l \\ 1 \end{pmatrix}$$



FK from PS03, problem 1

Problem 1

$$f(\alpha, l) = \binom{x_e}{y_e} = \binom{c\alpha \cdot l}{s\alpha \cdot l}$$

$$J = Df(\alpha, l) = \begin{pmatrix} \frac{\partial c\alpha \cdot l}{\partial \alpha} & \frac{\partial c\alpha \cdot l}{\partial l} \\ \frac{\partial s\alpha \cdot l}{\partial \alpha} & \frac{\partial s\alpha \cdot l}{\partial l} \end{pmatrix} = ???$$

Problem 1

note:

$$\mathbf{sin}'(\boldsymbol{ax}+\boldsymbol{b}) = \boldsymbol{a}\,\mathbf{cos}(\boldsymbol{ax}+\boldsymbol{b})$$

$$\mathbf{cos}'(\boldsymbol{ax}+\boldsymbol{b}) = -\boldsymbol{a}\,\mathbf{sin}(\boldsymbol{ax}+\boldsymbol{b})$$

example:

$$f(\alpha_1,\alpha_2) = \sin(\alpha_1+\alpha_2) \Rightarrow \frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_1} = \cos(\alpha_1+\alpha_2)$$

using 1st rule with $a = 1, x = \alpha_1, b = \alpha_2$

Problem 1

$$f(\alpha, l) = \binom{x_e}{y_e} = \binom{c\alpha \cdot l}{s\alpha \cdot l}$$

$$\boldsymbol{J} = Df(\alpha, l) = \begin{pmatrix} \dfrac{\partial c\alpha \cdot l}{\partial \alpha} & \dfrac{\partial c\alpha \cdot l}{\partial l} \\ \dfrac{\partial s\alpha \cdot l}{\partial \alpha} & \dfrac{\partial s\alpha \cdot l}{\partial l} \end{pmatrix} = \begin{pmatrix} \boldsymbol{-s\alpha \cdot l} & \boldsymbol{c\alpha} \\ \boldsymbol{c\alpha \cdot l} & \boldsymbol{s\alpha} \end{pmatrix}$$

Problem 1

$$Df(\alpha, l) = \begin{pmatrix} -s\alpha \cdot l & c\alpha \\ c\alpha \cdot l & s\alpha \end{pmatrix} \Longrightarrow Df(1,2) = \begin{pmatrix} \mathbf{-1.683} & \mathbf{0.540} \\ \mathbf{1.081} & \mathbf{0.841} \end{pmatrix}$$

note (1,2):
$\alpha = 1$ in radians
$l = 2$ in m
[SI as default]

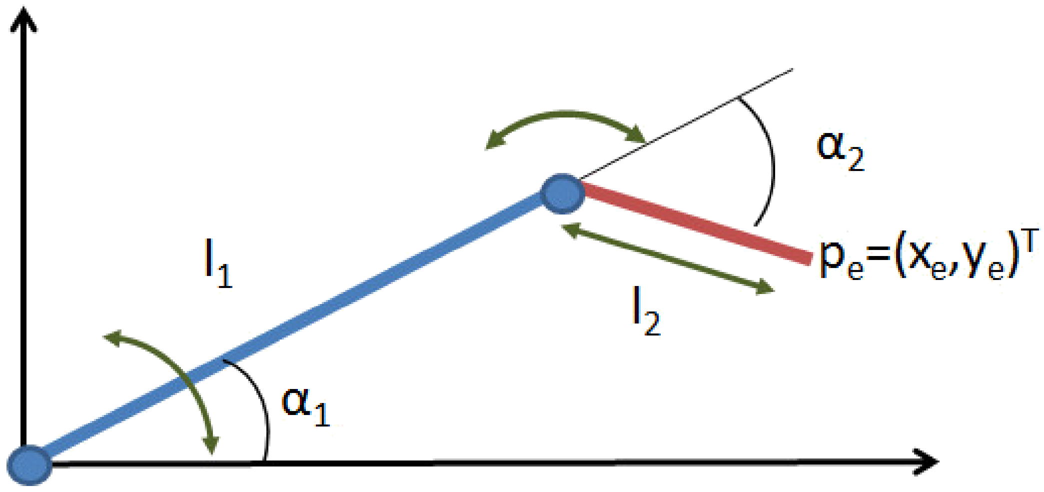
approximation with $\delta = 0.1$

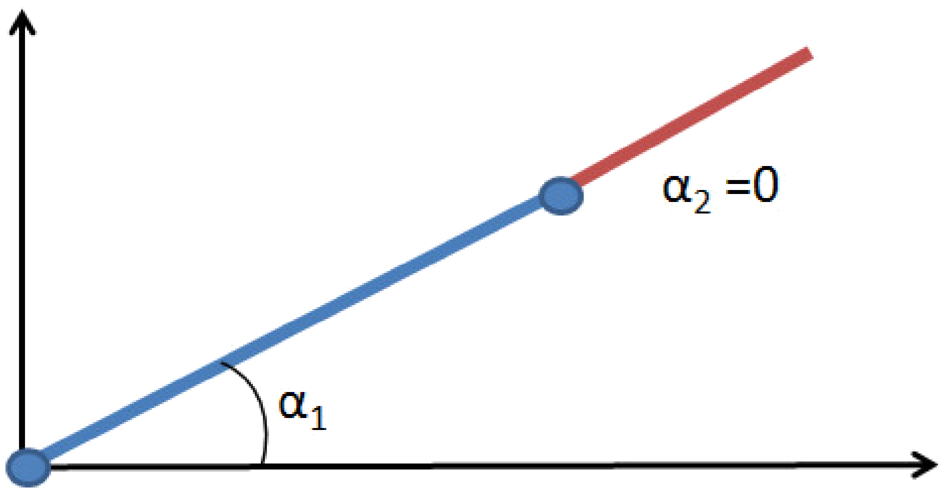
$$D_{\delta=0.1}f(1,2) = \begin{pmatrix} \frac{f_1(1+\delta, 2) - f_1(1,2)}{\delta} & \frac{f_1(1,2+\delta) - f_1(1,2)}{\delta} \\ \frac{f_2(1+\delta, 2) - f_2(1,2)}{\delta} & \frac{f_2(1,2+\delta) - f_2(1,2)}{\delta} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{c(1.1)\cdot 2 - c(1)\cdot 2}{0.1} & \frac{c(1)\cdot 2.1 - c(1)\cdot 2}{0.1} \\ \frac{s(1.1)\cdot 2 - s(1)\cdot 2}{0.1} & \frac{s(1)\cdot 2.1 - s(1)\cdot 2}{0.1} \end{pmatrix} = \begin{pmatrix} \mathbf{-1.734} & \mathbf{0.540} \\ \mathbf{0.995} & \mathbf{0.841} \end{pmatrix}$$

Problem 2

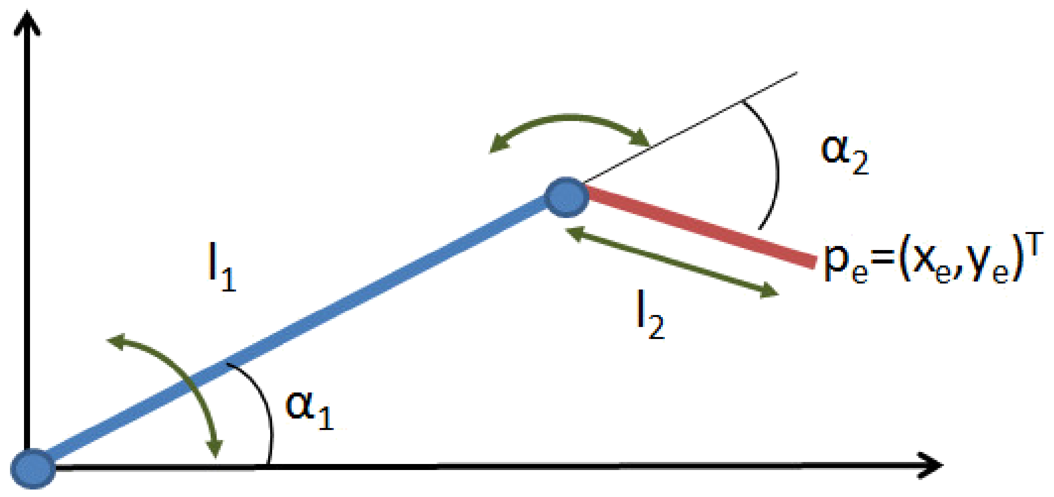
Take the planar arm from **PS03, Problem 2** with the DoF α_1, α_2 and l_2.
Use its FK to derive the related Jacobian matrix J.

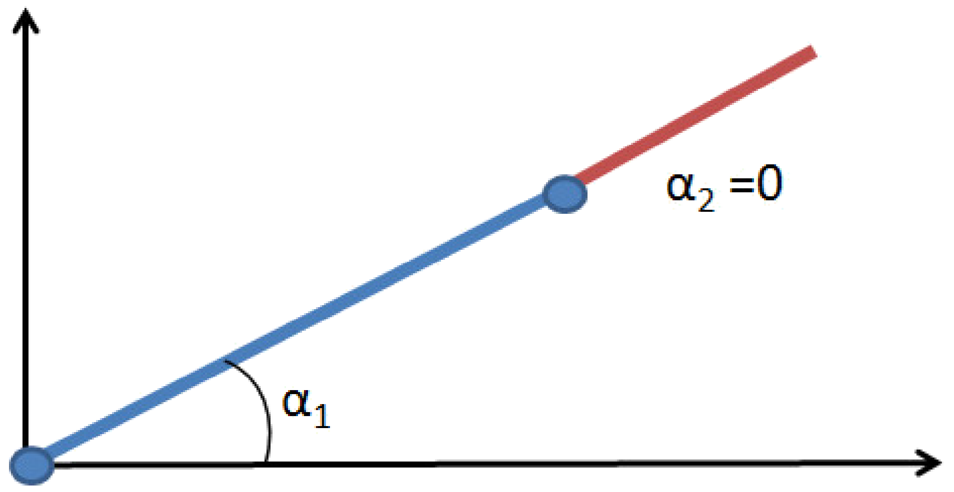




Problem 2

Forward Kinematics (PS03, problem 2)





$$K(\alpha_1, \alpha_2, l_2) = \begin{pmatrix} x_e \\ y_e \\ 1 \end{pmatrix} = \begin{pmatrix} c\alpha_1 c\alpha_2 \cdot l_2 - s\alpha_1 s\alpha_2 \cdot l_2 + c\alpha_1 \cdot 10 \\ s\alpha_1 c\alpha_2 \cdot l_2 + c\alpha_1 s\alpha_2 \cdot l_2 + s\alpha_1 \cdot 10 \\ 1 \end{pmatrix}$$

Problem 2

Jacobian of $\boldsymbol{K(\alpha_1, \alpha_2, l_2)}$

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

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

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

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

Problem 3

Take the Jacobian J from the previous problem 2. Which options do you know to compute the pseudo-inverse J^+ of J, and when are they applicable?

Problem 3

- option 1: closed form solution (formula for left or right A^+)
- option 2: numerical solution using SVD

Problem 3 (Option 1)

$m \times n$ matrix A , $n \times m$ matrix A^+

- $m > n$, i.e., $\boldsymbol{A}$ **is “tall”** :
  $A^+ = (A^T A)^{-1} A^T$ aka **left pseudo-inverse**

- $n > m$, i.e., $\boldsymbol{A}$ **is “wide”** :
  $A^+ = A^T (A A^T)^{-1}$ aka **right pseudo-inverse**

note: $n = m => A^+ = A^{-1}$

A “tall” :

$$\left(\begin{array}{c} \\ \quad A \quad \\ \\ \end{array}\right)$$

A “wide” :

$$\left(\begin{array}{c} \qquad\quad A \qquad\quad \end{array}\right)$$

Problem 3 (Option 1)

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

Jacobian $\boldsymbol{DK}() = \boldsymbol{A}$
$\Rightarrow$ A is a 2×3 matrix, i.e., “wide”
$\Rightarrow$ **right pseudo-inverse** $AA^+ = I$

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

Problem 3 (Option 1)

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

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

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

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

may be multiplied out, simplified, inverted, simplified…

Problem 3 (Option 1)

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

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

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

Problem 3 (Option 1)

option 1 with concrete values: step 1, the concrete Jacobian

$DK(90^o, 0^o, 8)$

$$= \begin{pmatrix} -s90^o c0^o \cdot 8 - c90^o s0^o \cdot 8 - s90^o \cdot 10 & -c90^o s0^o \cdot 8 - s90^o c0^o \cdot 8 & c90^o c0^o - s90^o s0^o \\ c90^o c0^o \cdot 8 - s90^o s0^o \cdot 8 + c90^o \cdot 10 & -s90^o s0^o \cdot 8 + c90^o c0^o \cdot 8 & s90^o c0^o + c90^o s0^o \end{pmatrix}$$

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

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

as mentioned, $DK() = A$ is “wide” => $A^+ = A^T (AA^T)^{-1}$

Problem 3 (Option 1)

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

wide $A \Longrightarrow A^+ = A^T(AA^T)^{-1}$

$$DK^+(90^o, 0^o, 8) = \begin{pmatrix} -18 & 0 \\ -8 & 0 \\ 0 & 1 \end{pmatrix} \left(\begin{pmatrix} -18 & -8 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -18 & 0 \\ -8 & 0 \\ 0 & 1 \end{pmatrix} \right)^{-1}$$

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

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

closed form pseudo-inverse of the Jacobian with concrete values for input DoF

Problem 3 (Option 2)

singular value decomposition (SVD)
can only be used when fixed DoF values are given

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

$$U = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, W = \begin{pmatrix} 19.698 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, V^T = \begin{pmatrix} -0.914 & -0.406 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Problem 3 (Option 2)

$$U = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, W = \begin{pmatrix} 19.698 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, V^T = \begin{pmatrix} -0.914 & -0.406 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

$$DK^+(90^o, 0^o, 8) = VW^+U^T$$

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

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

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

Problem 3 (Option 2)

note: $DK^{+}(90^{o}, 0^{o}, 8) = VW^{+}U^{T}$ with W^{+}

- transpose of W, i.e., $W_{ij}^{+} = W_{ji}$, and
- reciprocals of non-Zero diagonal entries, i.e., $W_{ii}^{+} = 1/W_{ii}$ if $W_{ii} \neq 0$

$$W^{+} = \begin{pmatrix} 19.698 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^{+} = \begin{pmatrix} \frac{1}{19.698} & 0 & 0 \\ 0 & \frac{1}{1} & 0 \end{pmatrix}^{T} = \begin{pmatrix} 0.05077 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

Problem 3: Pseudo-Inverse

option 1: closed form

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

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: both examples done with some rounding in the calculations

Problem 4

Take the arm, its FK, and the related pseudo-inverse of the Jacobian J^+ from the previous problems 2 and 3.

Given the goal position $p_t = (5, 10)^T$ 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
b) Gradient descent

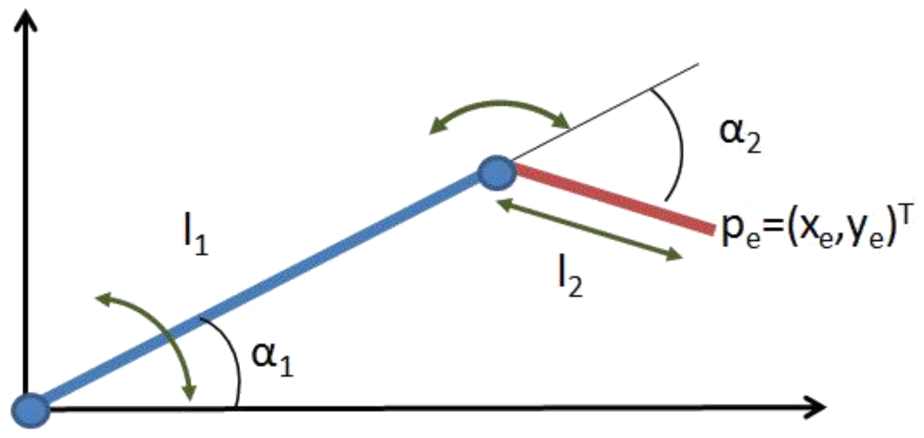
Problem 4

a) Newton's method: $q_{k+1} = q_k + \alpha J(q_k)^+(p_t - K(q_k))$

b) Gradient descent: $q_{k+1} = q_k + \alpha J(q_k)^T(p_t - K(q_k))$

note: yet an other “clear from context”

- dof, i.e., q, in kinematics as a column vector
- q as “input” to fct $K()$ as tupel, i.e., like row vector



$$q = (\alpha_1, \alpha_2, l_2)^T$$

$$K(q) = K(\alpha_1, \alpha_2, l_2) = \begin{pmatrix} x_e \\ y_e \\ 1 \end{pmatrix} = \begin{pmatrix} c\alpha_1 c\alpha_2 \cdot l_2 - s\alpha_1 s\alpha_2 \cdot l_2 + c\alpha_1 \cdot 10 \\ s\alpha_1 c\alpha_2 \cdot l_2 + c\alpha_1 s\alpha_2 \cdot l_2 + s\alpha_1 \cdot 10 \\ 1 \end{pmatrix}$$

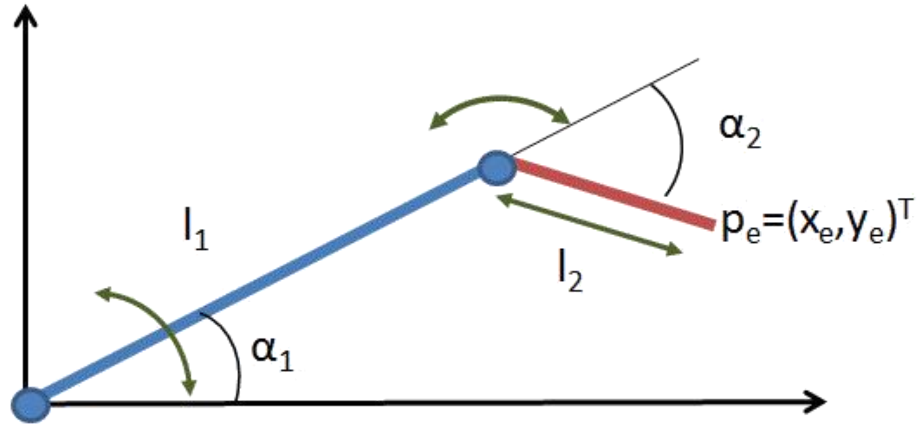
Problem 4

a) Newton's method: $q_{k+1} = q_k + \alpha J(q_k)^+ (p_t - K(q_k))$

b) Gradient descent: $q_{k+1} = q_k + \alpha J(q_k)^T (p_t - K(q_k))$

see problem 2

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



$$J = DK(q)$$

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

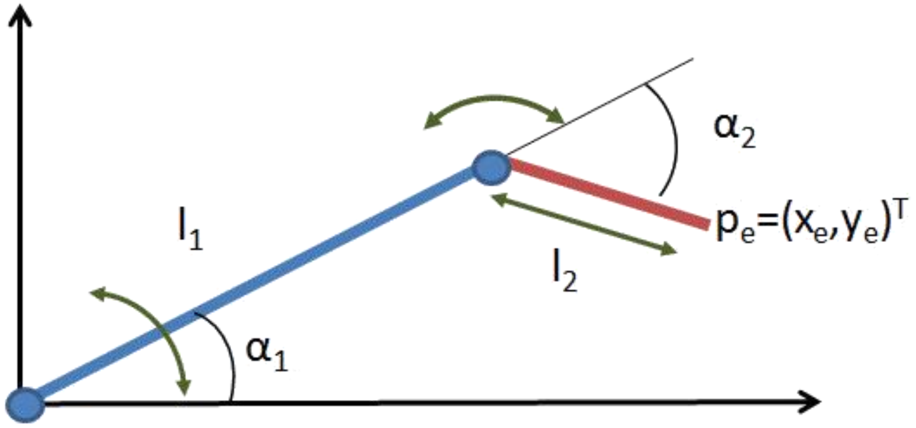
Problem 4: Newton-Raphson

note: here () as notation for the time-steps instead of subscripts

$$q(k+1) = q(k) + \alpha J(q(k))^{+}\left(p_t - K\big(q(k)\big)\right)$$
$$= q(k) + \alpha\Delta\mathrm{q} \;\; \text{with} \;\; \Delta\mathrm{q} = J(q(k))^{+}(p_t - K(q(k)))$$

start: $k = 0$

$$q(0) = \big(\alpha_1(0), \alpha_2(0), l_2(0)\big)^T$$
$$= (90^o, 0^o, 8)^T$$



Problem 4: Newton-Raphson

$q(0) = (90^o, 0^o, 8)^T$

$q(1) = q(0) + \alpha\Delta q$ with $\Delta q = J(q(0))^+ (\boxed{p_t} - \boxed{K(q(0))})$

hand target: $\boxed{p_t = \begin{pmatrix} 5 \\ 10 \end{pmatrix}}$

hand at $k = 0$:

$$\boxed{K\big(q(0)\big)} = K(90^o, 0^o, 8)$$

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

$$= \begin{pmatrix} 0 - 0 + 0 \\ 8 + 0 + 10 \end{pmatrix} = \boxed{\begin{pmatrix} 0 \\ 18 \end{pmatrix}}$$

Problem 4: Newton-Raphson

$q(0) = (90^o, 0^o, 8)^T$

$q(1) = q(0) + \alpha \Delta q$ with

$$\Delta q = J(q(0))^+ \left(\begin{pmatrix} 5 \\ 10 \end{pmatrix} - \begin{pmatrix} 0 \\ 18 \end{pmatrix} \right) = J(q(0))^+ \begin{pmatrix} 5 \\ -8 \end{pmatrix}$$

pseudo-inverse of the Jacobian at $k = 0$:

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

problem 3

Problem 4: Newton-Raphson

$q(0) = (90^o, 0^o, 8)^T$

$q(1) = q(0) + \alpha\Delta q$ with $J(q(0))^+ \left(p_t - K\left(q(0)\right)\right)$

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

note: the angular changes are in radians

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

Problem 4: Newton-Raphson

$q(0) = (90^o, 0^o, 8)$

$q(1) = q(0) + \alpha\Delta q$ with $J(q(0))^+ \left(p_t - K\big(q(0)\big)\right)$

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

$$q(1) = \begin{pmatrix} 90^o \\ 0^o \\ 8 \end{pmatrix} + 0.1 \cdot \begin{pmatrix} -13.30^o \\ -5.91^o \\ 8 \end{pmatrix} = \begin{pmatrix} 90^o \\ 0^o \\ 8 \end{pmatrix} + \begin{pmatrix} -1.330^o \\ -0.591^o \\ 0.8 \end{pmatrix}$$
$$= (88.670^o \quad 359.319^o \quad 8.8)^T$$

Problem 4: Newton-Raphson

keep on iterating:

- compute forward kinematics $K(q(1))$ of $q(1)$
- next Jacobian $J(q(1))$ at $q(1)$
- next pseudo-inverse $J^{+}(q(1))$
- get $q(2) = q(1) + \alpha J(q(1))^{+}\left(p_t - K\big(q(1)\big)\right)$
- and so on...

until small error to target, i.e., $|p_t - K(q(n))| < \varepsilon$

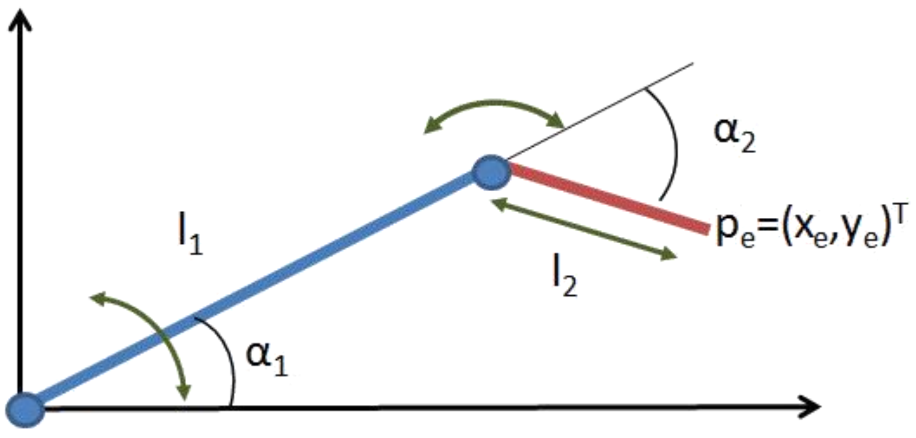
Problem 4: Gradient Descent

note: also here () for the time-steps

$$q(k+1) = q(k) + \alpha J(q(k))^T \left(p_t - K\big(q(k)\big)\right)$$
$$= q(k) + \alpha \Delta \mathrm{q} \;\text{ with }\; \Delta \mathrm{q} = J(q(k))^T (p_t - K(q(k)))$$

start: $k = 0$

$$q(0) = \big(\alpha_1(0), \alpha_2(0), l_2(0)\big)^T$$
$$= (90^o, 0^o, 8)^T$$



Problem 4: Gradient Descent

$q(0) = (90^o, 0^o, 8)^T$

$q(1) = q(0) + \alpha\Delta q$ with $\Delta q = J(q(0))^+ (\boxed{p_t} - \boxed{K(q(0))})$

this part is exactly like Newton

hand target: $\boxed{p_t = \begin{pmatrix} 5 \\ 10 \end{pmatrix}}$

hand at $k = 0$:

$$\boxed{K\big(q(0)\big)} = K(90^o, 0^o, 8)$$

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

$$= \begin{pmatrix} 0 - 0 + 0 \\ 8 + 0 + 10 \end{pmatrix} = \boxed{\begin{pmatrix} 0 \\ 18 \end{pmatrix}}$$

Problem 4: Gradient Descent

$q(0) = (90^o, 0^o, 8)^T$

$q(1) = q(0) + \alpha\Delta q$ with

$$\Delta q = J(q(0))^T \left(\begin{pmatrix} 5 \\ 10 \end{pmatrix} - \begin{pmatrix} 0 \\ 18 \end{pmatrix} \right) = J(q(0))^T \begin{pmatrix} 5 \\ -8 \end{pmatrix}$$

transpose of the Jacobian at $k = 0$:

Newton needs pseudo-inverse

$$J(q(0))^T = DK^T(q) = \begin{pmatrix} -18 & 0 \\ -8 & 0 \\ 0 & 1 \end{pmatrix}$$

Problem 4: Gradient Descent

$q(0) = (90^o, 0^o, 8)$

$q(1) = q(0) + \alpha \Delta q$ with $J(q(0))^T \left(p_t - K\big(q(0)\big)\right)$

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

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

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

Problem 4: Gradient Descent

keep on iterating:

- compute forward kinematics $K(q(1))$ of $q(1)$
- next Jacobian $J(q(1))$ at $q(1)$
- next transpose $J^T(q(1))$
- get $q(2) = q(1) + \alpha J(q(1))^T \left(p_t - K\big(q(1)\big)\right)$
- and so on...

until small error to target, i.e., $|p_t - K(q(n))| < \varepsilon$

Problem 4

IK with

a) Newton's method: $q_{k+1} = q_k \boxed{+} \alpha J(q_k)^{+}(p_t - K(q_k))$

b) Gradient descent: $q_{k+1} = q_k \boxed{+} \alpha J(q_k)^{T}(p_t - K(q_k))$

in general,

a) Newton's method: $x_{k+1} = x_k \boxed{-} \alpha J_F(x_k)^{+} F(x_k)$

b) Gradient descent: $x_{k+1} = x_k \boxed{-} \alpha \nabla F(x_k)$ with $\nabla F(x_k) = J_F\,(x_k)^T$

why once "+" and once "–"?

Problem 4

general: $x_{k+1} = x_k - \alpha J_F(x_k)^{+,T} F(x_k)$

IK: $q_{k+1} = q_k + \alpha J(q_k)^{+,T} (p_t - K(q_k))$

$$F(x_k) \leftrightarrow F(q_k) = (p_t - K(q_k))$$

$$\begin{aligned} J_F(x_k) \leftrightarrow J_F(q_k) &= DF(q_k) \\ &= \frac{\delta(p_t - K(q_k))}{\delta q_k} \quad (p_t \text{ is constant}) \\ &= -J_K(q_k) \end{aligned}$$

Problem 4

IK with

a) Newton's method: $q_{k+1} = q_k + \alpha J(q_k)^+ (p_t - K(q_k))$

b) Gradient descent: $q_{k+1} = q_k + \alpha J(q_k)^T (p_t - K(q_k))$

in general,

a) Newton's method: $x_{k+1} = x_k - \alpha J_F(x_k)^+ F(x_k)$

b) Gradient descent: $x_{k+1} = x_k - \alpha \nabla F(x_k)$ with $\nabla F(x_k) = J_F\,(x_k)^T$

which is better?

Problem 4

just as a very rough guideline

- **Newton Raphson** $\boldsymbol{x_{k+1} = x_k - \alpha J_F(x_k)^+ F(x_k)}$
  - finds the **root** of a multivariate, **vector-valued function**
  - i.e., $\boldsymbol{F(\hat{x}) = 0}$ for $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$
  - it is more complex per step (needs pseudo-inverse) but it can converge faster
- **gradient descent** $\boldsymbol{x_{k+1} = x_k - \alpha \nabla F(x_k)}$
  - finds the **minimum** of a multivariate, **real-valued function**
  - i.e., $\boldsymbol{\hat{x} = \min_{x} F(x)}$ for $F : \mathbb{R}^n \rightarrow \mathbb{R}$
  - it is simpler (just needs J transpose) but smaller steps towards the minimum

note: and there are also other methods for numerical optimization… (and for IK)