



Robotics: Inverse Kinematics

Vladimír Petřík

vladimir.petrik@cvut.cz

16.10.2023

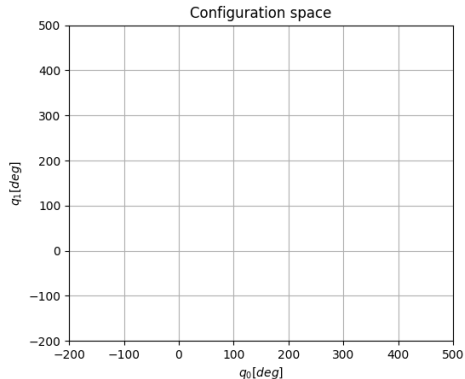
Kinematics tasks

- ▶ Forward kinematics (FK)
 - ▶ how to compute end-effector pose from the configuration
 - ▶ $\mathbf{x} = f_{\text{fk}}(\mathbf{q})$
 - ▶ \mathbf{x} is expressed in task-space, *i.e.* $SE(2)$, $SE(3)$, or \mathbb{R}^2 , \mathbb{R}^3 for position only
 - ▶ $\mathbf{q} \in \mathbb{R}^N$ is configuration (joint space)
- ▶ Differential kinematics
 - ▶ relates end-effector velocity to joint velocities
 - ▶ $\dot{\mathbf{x}} = J(\mathbf{q})\dot{\mathbf{q}}$
- ▶ Inverse kinematics (IK)
 - ▶ how to compute robot configuration(s) for given end-effector configuration
 - ▶ $\mathbf{q} \in f_{\text{ik}}(\mathbf{x})$



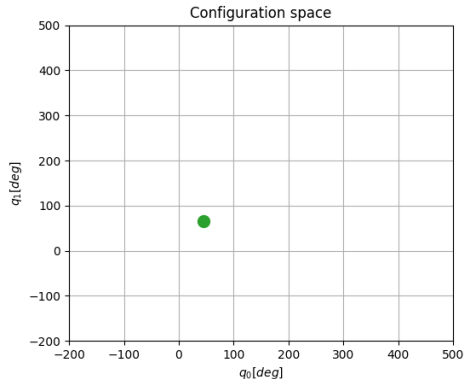
Example: Analytical IK for RR structure

- ▶ Task space: translation of end-effector, $\boldsymbol{x} \in \mathbb{R}^2$
- ▶ Configuration (joint) space: $\boldsymbol{q} \in \mathbb{R}^2$



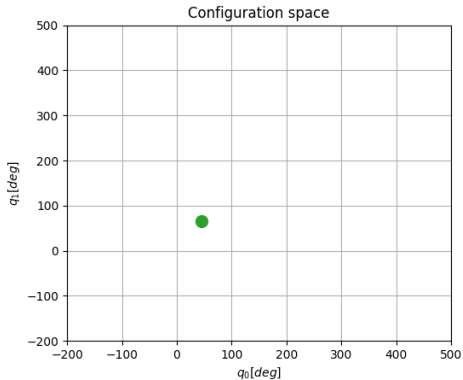
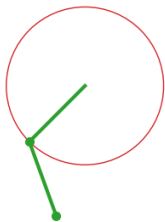
Example: Analytical IK for RR structure

- ▶ Task space: translation of end-effector, $x \in \mathbb{R}^2$
- ▶ Configuration (joint) space: $q \in \mathbb{R}^2$



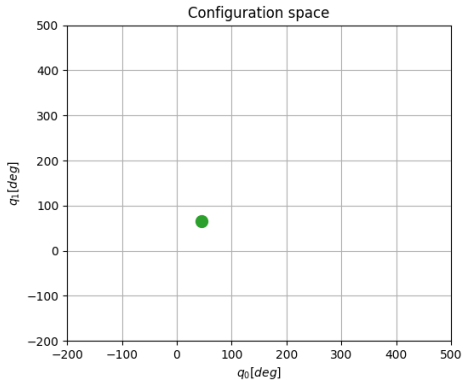
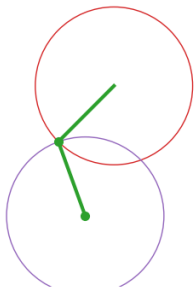
Example: Analytical IK for RR structure

- ▶ Task space: translation of end-effector, $x \in \mathbb{R}^2$
- ▶ Configuration (joint) space: $q \in \mathbb{R}^2$



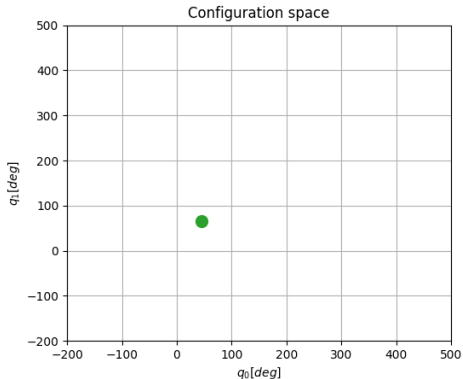
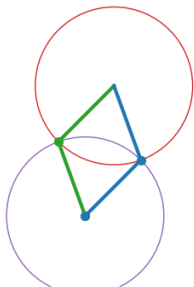
Example: Analytical IK for RR structure

- ▶ Task space: translation of end-effector, $x \in \mathbb{R}^2$
- ▶ Configuration (joint) space: $q \in \mathbb{R}^2$



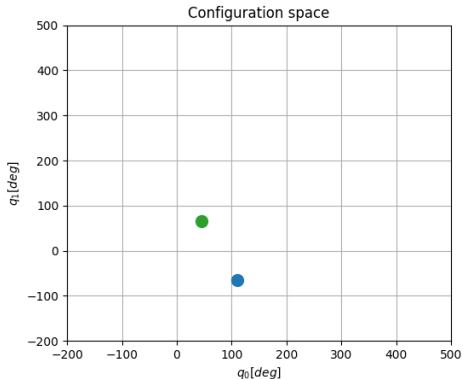
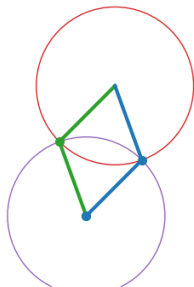
Example: Analytical IK for RR structure

- ▶ Task space: translation of end-effector, $x \in \mathbb{R}^2$
- ▶ Configuration (joint) space: $q \in \mathbb{R}^2$



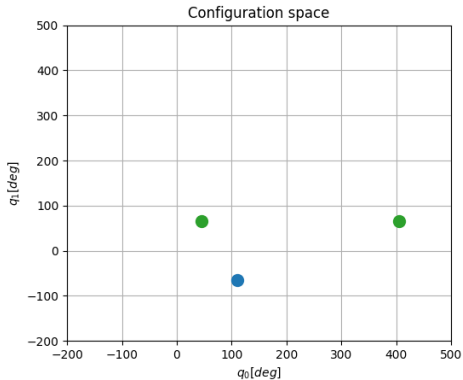
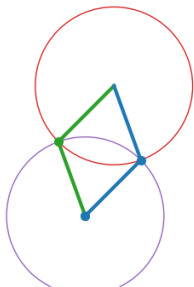
Example: Analytical IK for RR structure

- ▶ Task space: translation of end-effector, $x \in \mathbb{R}^2$
- ▶ Configuration (joint) space: $q \in \mathbb{R}^2$



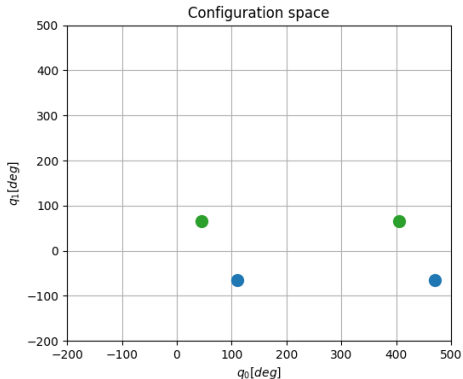
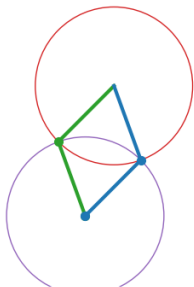
Example: Analytical IK for RR structure

- ▶ Task space: translation of end-effector, $x \in \mathbb{R}^2$
- ▶ Configuration (joint) space: $q \in \mathbb{R}^2$



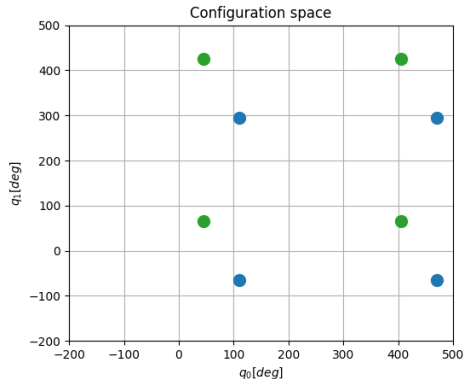
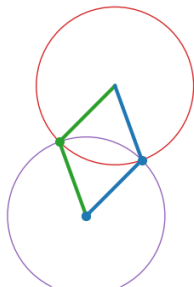
Example: Analytical IK for RR structure

- ▶ Task space: translation of end-effector, $x \in \mathbb{R}^2$
- ▶ Configuration (joint) space: $q \in \mathbb{R}^2$



Example: Analytical IK for RR structure

- ▶ Task space: translation of end-effector, $x \in \mathbb{R}^2$
- ▶ Configuration (joint) space: $q \in \mathbb{R}^2$



Example: Analytical IK for RR structure

- ▶ Task space: translation of end-effector, $\mathbf{x} \in \mathbb{R}^2$
- ▶ Configuration (joint) space: $\mathbf{q} \in \mathbb{R}^2$
- ▶ Algorithm:
 - ▶ Compute position of all joints and end-effector
 - ▶ No solution, 1 solution, 2 solutions, or ∞ solutions
 - ▶ For each solution, compute joint configurations $\theta_i = \text{atan2}(y, x) + 2k\pi, k \in \mathbb{Z}$
 $\begin{pmatrix} x & y \end{pmatrix}^\top = \mathbf{t}_{i,i+1}$, i.e. translation part of $T_{i,i+1}$



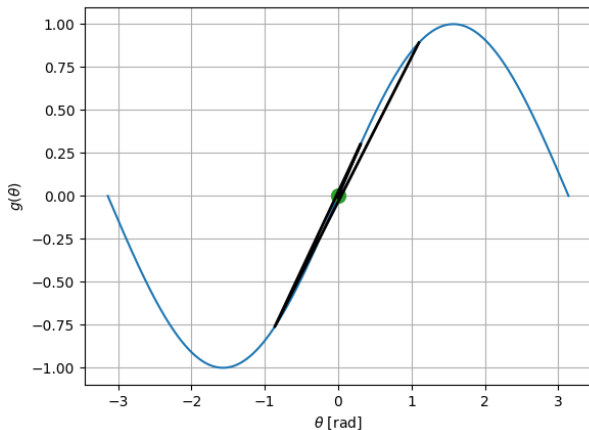
Numerical optimization

- ▶ Analytical solution is often unavailable
 - ▶ solution does not exist and we seek for the closest approximate
 - ▶ infinite solutions exist and we seek for configuration w.r.t. given criteria
- ▶ We can use generic numerical algorithm, that iteratively reduce error
- ▶ **Newton–Raphson** method
 - ▶ solve $g(\theta) = 0$, $g : \mathbb{R} \rightarrow \mathbb{R}$
 - ▶ Taylor expansion of $g(\theta)$ at θ^0 :
$$g(\theta) = g(\theta^0) + \frac{\partial g}{\partial \theta}(\theta^0)(\theta - \theta^0) + \text{higher-order terms}$$
 - ▶ set $g(\theta) = 0$, ignore higher-order terms, and solve for θ :
$$\theta \approx \theta^0 - \left(\frac{\partial g}{\partial \theta}(\theta^0) \right)^{-1} g(\theta^0)$$
 - ▶ as we ignore higher-order terms, we need to iterate:
$$\theta^{k+1} = \theta^k - \left(\frac{\partial g}{\partial \theta}(\theta^k) \right)^{-1} g(\theta^k)$$



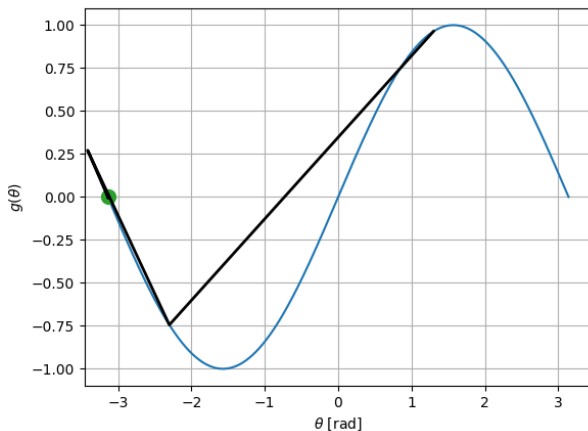
1D Newton–Raphson method example

- ▶ $g(\theta) = \sin(\theta)$, find θ^* s.t. $g(\theta^*) = 0$, $\theta^0 = 1.1$



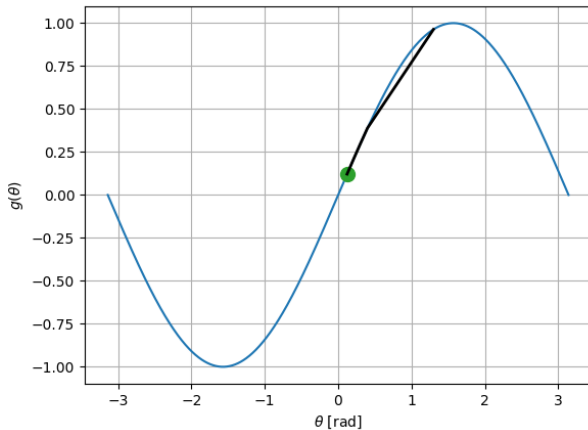
1D Newton–Raphson method example

- ▶ $g(\theta) = \sin(\theta)$, find θ^* s.t. $g(\theta^*) = 0$, $\theta^0 = 1.3$
- ▶ Quality of the solution depends on the initial guess



1D Newton–Raphson method example

- ▶ $g(\theta) = \sin(\theta)$, find θ^* s.t. $g(\theta^*) = 0$, $\theta^0 = 1.3$, $\alpha = 0.5$
- ▶ $\theta^{k+1} = \theta^k - \alpha \left(\frac{\partial g}{\partial \theta}(\theta^k) \right)^{-1} g(\theta^k)$



How to find α ?

- ▶ Line-search algorithm
- ▶ Find α s.t. $g(\theta^{k+1}) < g(\theta^k)$
- ▶ Algorithm:
 - ▶ $\alpha^0 = 1$
 - ▶ if $g(\theta^{k+1}) < g(\theta^k)$: break
 - ▶ $\alpha^{i+1} = \tau\alpha^i$, $0 < \tau < 1$, e.g. $\tau = 0.5$
 - ▶ repeat
- ▶ More sophisticated line-search algorithms exist



Numerical solution for RR IK

- ▶ Newton–Raphson method for n -dimensional case

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \alpha \left(\frac{\partial g}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^k) \right)^{-1} g(\boldsymbol{\theta}^k) \text{ solves } g(\boldsymbol{\theta}) = \mathbf{0}$$

- ▶ For manipulator kinematics:

$$g(\mathbf{q}) = \mathbf{x}_d - f_{\text{fk}}(\mathbf{q}), \mathbf{x}_d \in \mathbb{R}^2$$

- ▶ Following NR method (for $g(\mathbf{q}) = 0$):

$$\begin{aligned} \mathbf{x}_d = f_{\text{fk}}(\mathbf{q}_d) &\approx f_{\text{fk}}(\mathbf{q}^0) + \frac{\partial f_{\text{fk}}}{\partial \mathbf{q}}(\mathbf{q}^0)(\mathbf{q}_d - \mathbf{q}^0) = f_{\text{fk}}(\mathbf{q}^0) + J(\mathbf{q}^0)(\mathbf{q}_d - \mathbf{q}^0) \\ \mathbf{q}_d &\approx \mathbf{q}^0 + J(\mathbf{q}^0)^{-1}(\mathbf{x}_d - f_{\text{fk}}(\mathbf{q}^0)) \end{aligned}$$

- ▶ Iteratively with line-search:

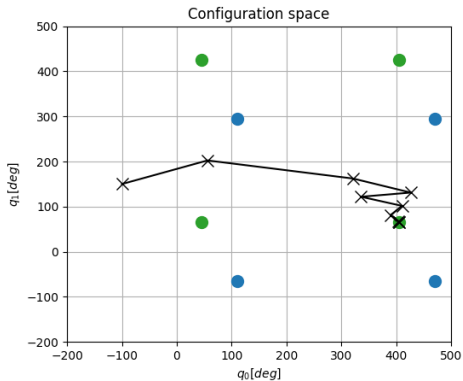
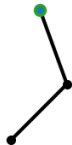
$$\mathbf{q}^{k+1} = \mathbf{q}^k + \alpha J(\mathbf{q}^k)^{-1}(\mathbf{x}_d - f_{\text{fk}}(\mathbf{q}^k))$$

- ▶ Intuition via differential kinematics:

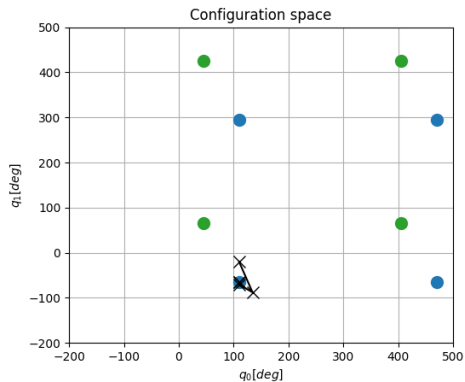
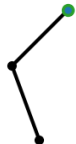
- ▶ what should be velocity in joint space s.t. we achieve given velocity in task-space
- ▶ $\dot{\mathbf{q}} = J^{-1}\dot{\mathbf{x}}$



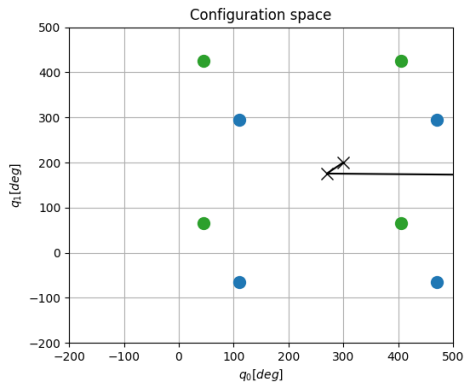
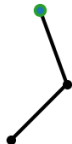
Numerical solution for RR IK #1



Numerical solution for RR IK #2



Numerical solution for RR IK #3



Numerical solution - takout message

- ▶ Numerical solution is easy to implement for general manipulators
- ▶ Initial guess is important
 - ▶ if we are *close* to the solution, FK is almost linear we will converge to the *closest* solution
 - ▶ if we are *too far away* we have no control about which solution is selected
 - ▶ tuning step-size might help
- ▶ We need to define stopping criteria
 - ▶ e.g. $\|\mathbf{x}_d - f_{fk}(\mathbf{q}^k)\| < \varepsilon$

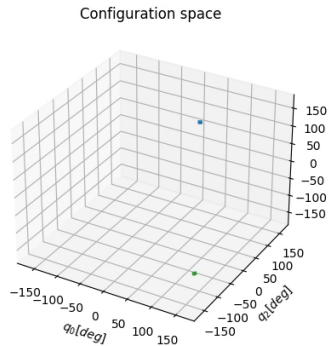


What if J is not invertible?

- ▶ Redundant robots, Underactuated robots, Singularity
- ▶ Moore–Penrose pseudoinverse J^\dagger
- ▶ $\mathbf{q}^{k+1} = \mathbf{q}^k + \alpha J^\dagger(\mathbf{q}^k)(\mathbf{x}_d - f_{fk}(\mathbf{q}^k))$
- ▶ Redundant robots
 - ▶ infinite solutions to achieve same task space velocity
 - ▶ pseudoinverse will additionally minimize $\|\mathbf{q}\|$
- ▶ Underactuated robots or singularity
 - ▶ no exact solution exist for task space velocity
 - ▶ pseudoinverse will minimize the error in task-space



IK solution for redundant robot

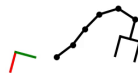
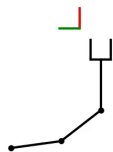


IK in $SE(2)$ for RRR

- ▶ Given desired pose $T_{RG}^D \in SE(2)$
 - ▶ R - reference frame
 - ▶ G - gripper frame
- ▶ Analytical solution
 - ▶ decouple problem into rotation (last joint) and position (other joints)
 - ▶ $\mathbf{t}_{RC} = T_{RG}^D \begin{pmatrix} -l_3 & 0 & 1 \end{pmatrix}^\top$
 - ▶ \mathbf{t}_{RB} compute as for RR for translation task-space
 - ▶ use `atan2` to compute joint configurations
- ▶ Numerical solution
 - ▶ error in reference frame:
$$\mathbf{e}(\mathbf{q}) = \left(x_{RG}^D - x_{RG}(\mathbf{q}) \quad y_{RG}^D - y_{RG}(\mathbf{q}) \quad \phi_{RG}^D - \phi_{RG}(\mathbf{q}) \right)^\top$$
 - ▶ NR step:
$$\mathbf{q}^{k+1} = \mathbf{q}^k + \alpha J^\dagger(\mathbf{q}^k) \mathbf{e}(\mathbf{q}^k)$$



Numerical solution in $SE(2)$



IK in $SE(3)$

- ▶ Numerical IK algorithm is almost the same
 - ▶ error needs to be computed via transformations
 - ▶ as in planar case, error needs to be represented in reference frame
- ▶ Analytical solution might not exist for general 6 DoF manipulator
- ▶ For 6 DoF spatial robot with revolute joints
 - ▶ solution can be decoupled if last three joint axes intersect each other
 - ▶ use last three joints to orient gripper
 - ▶ use the first three joints to position the flange



Example of importance of multiple solutions



Summary

- ▶ Inverse kinematics
 - ▶ analytical solution via geometrical analysis
 - ▶ leads to computation of intersections of geometrical primitives
 - ▶ numerical solution, Newton–Raphson method
 - ▶ Jacobian
 - ▶ pseudoinverse
- ▶ Number of solutions of inverse kinematics
 - ▶ no solution
 - ▶ multiple solutions
 - ▶ periodical solutions
 - ▶ infinite number of solutions



Laboratory

- ▶ Numerical IK in $SE(2)$
- ▶ Analytical IK in $SE(2)$ for RRR manipulator
- ▶ Analytical IK in $SE(2)$ for PRR manipulator [optional]

