Inverse Kinematics

Forwards kinematics

Homogeneous transformations:

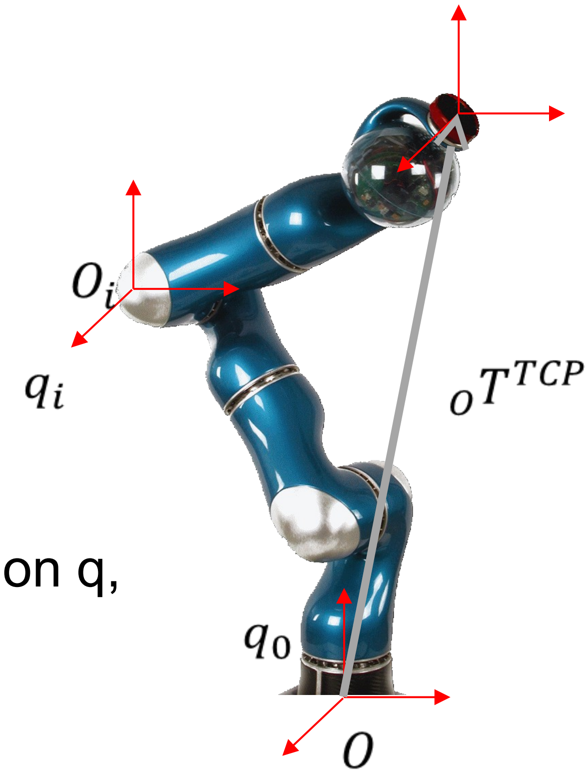
$$_0T^{TCP} = {}_0T^1 \, {}_1T^2 \ldots {}_iT^{i+1} \ldots {}_{n-1}T^{TCP} = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$$

Cartesian representation:

$$x = \begin{bmatrix} p \\ \phi \end{bmatrix} \in \Re^6$$

$$\Downarrow$$

$$x = f(q)$$



Forward kinematics: “Given the joints position q, calculates the Cartesian position x”

Inverse kinematics

$$x = f(q)$$

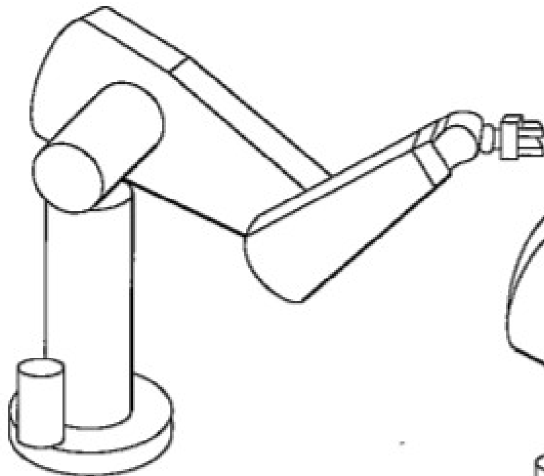
$$\Downarrow$$

$$q = f^{-1}(x)$$

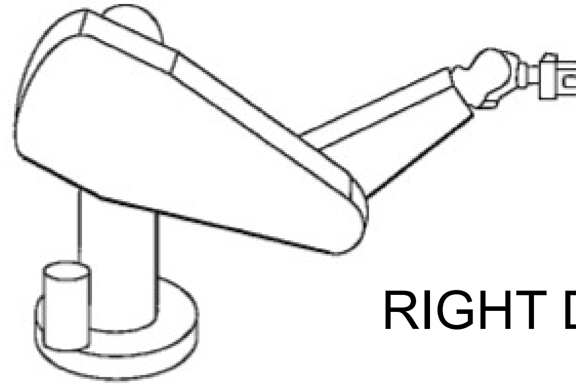
Inverse kinematics: "Given the Cartesian position x, calculates the joints position q"

- Nonlinear problem
- Existence of the solution: $x \in WS$
- Multiple solutions: $q_i = f^{-1}(x) \quad i = 1, n, \infty$

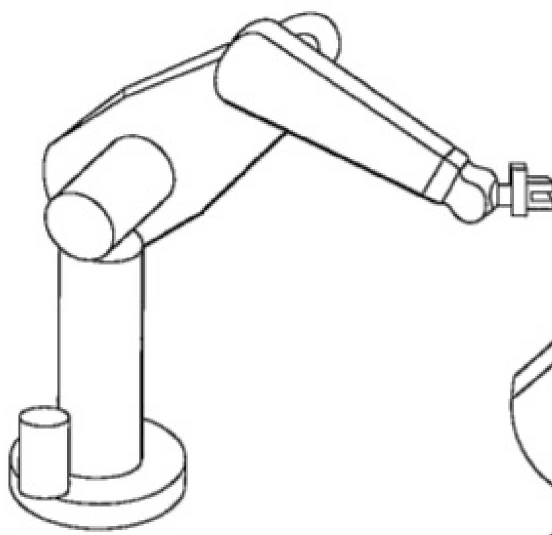
Multiple solutions



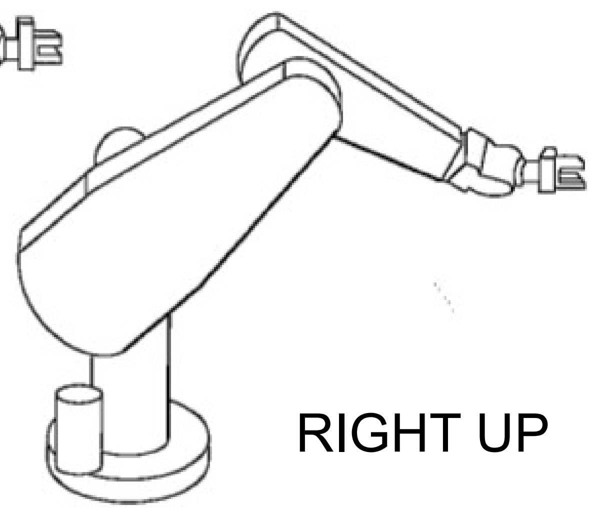
LEFT DOWN



RIGHT DOWN



LEFT UP



RIGHT UP

Analyitical solution approach

Solving the nonlinear kinematics equations analytically is in general a very difficult problem, but is feasible for particular kinematics

Advantages:
- Fast implementations
- Possibility to find all solutions

Drawbacks:
- Difficult or even impossible
- Only non redundant case

Methods: algebraic, geometric inspection, etc ..

Solvability conditions (sufficient conditions):
- Three consecutive joints with parallel axes *or*
- Three consecutive joints with incident axes

Complexity

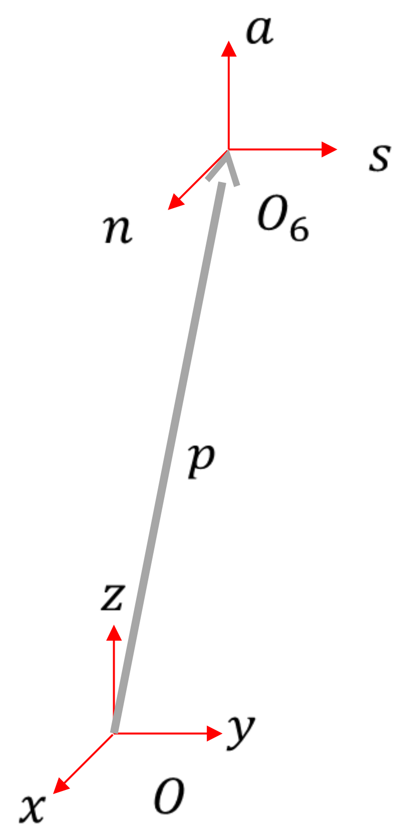
Example 6DOF: Unimation PUMA 600

$$\left.\begin{aligned}
n_x &= C_1[C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6] \\
&\quad - S_1[S_4C_5C_6 + C_4S_6] \\
n_y &= S_1[C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6] \\
&\quad + C_1[S_4C_5C_6 + C_4S_6] \\
n_z &= -S_{23}(C_4C_5C_6 - S_4S_6) - C_{23}S_5C_6
\end{aligned}\right\} \; n = {}^0x_6(q)$$

$$\left.\begin{aligned}
o_x &= C_1[-C_{23}(C_4C_5S_6 + S_4C_6) + S_{23}S_5S_6] \\
&\quad - S_1[-S_4C_5S_6 + C_4C_6] \\
o_y &= S_1[-C_{23}(C_4C_5S_6 + S_4C_6) + S_{23}S_5S_6] \\
&\quad + C_1[-S_4C_5S_6 + C_4C_6] \\
o_z &= S_{23}(C_4C_5S_6 + S_4C_6) + C_{23}S_5S_6
\end{aligned}\right\} \; s = {}^0y_6(q)$$

$$\left.\begin{aligned}
a_x &= C_1(C_{23}C_4S_5 + S_{23}C_5) - S_1S_4S_5 \\
a_y &= S_1(C_{23}C_4S_5 + S_{23}C_5) + C_1S_4S_5 \\
a_z &= -S_{23}C_4S_5 + C_{23}C_5
\end{aligned}\right\} \; a = {}^0z_6(q)$$

$$\left.\begin{aligned}
p_x &= C_1(d_4S_{23} + a_3C_{23} + a_2C_2) - S_1d_3 \\
p_y &= S_1(d_4S_{23} + a_3C_{23} + a_2C_2) + C_1d_3 \\
p_z &= -(-d_4C_{23} + a_3S_{23} + a_2S_2).
\end{aligned}\right\} \; p = 0_6(q)$$



Numerical approach

Advantages:
- Simple implementation
- Applicable to redundant robots

Drawbacks:
- Computationally heavy
- Numerical problems near singularity
- Convergence issues of the nonlinear solution

Concept:

Bring to zero the error: $e = x_d - f(q)$

Thus reduce solving the nonlinear algebraic equation to solving iteratively its local linear approximation using the Jacobian

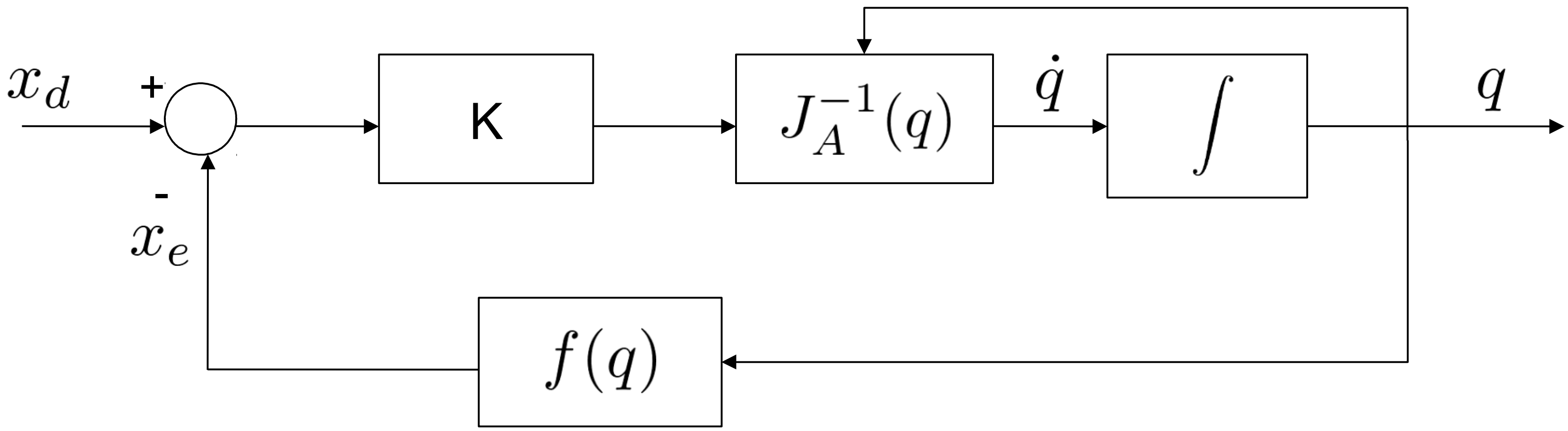
$\dot{e} = -J_A(q)\dot{q}$ or discrete $\delta e = J_A(q)\delta q$

analytical Jacobian $J_A = \dfrac{\partial f(q)}{\partial q}$

Inverse Jacobian approach

For a non-redundant non-singular robot it can be chosen:

$$\dot{q} = J_A^{-1} K e$$



Closed loop (linear dynamics):

$$\dot{e} + Ke = 0 \qquad K = K^T > 0 \qquad e \to 0$$

Inverse Jacobian approach

Monodimensional case

$$J_A(q) = \frac{\partial f(q)}{\partial q} = f'(q)$$

For $x_d = 0$

$$e = x_d - f(q) = -f(q)$$

$$\dot{q} = J_A^{-1} K e \;\Rightarrow\; \Delta q \cong J_A^{-1} K e \Delta t \qquad [K = \alpha / \Delta t]$$

$$\Delta q = J_A^{-1} e$$

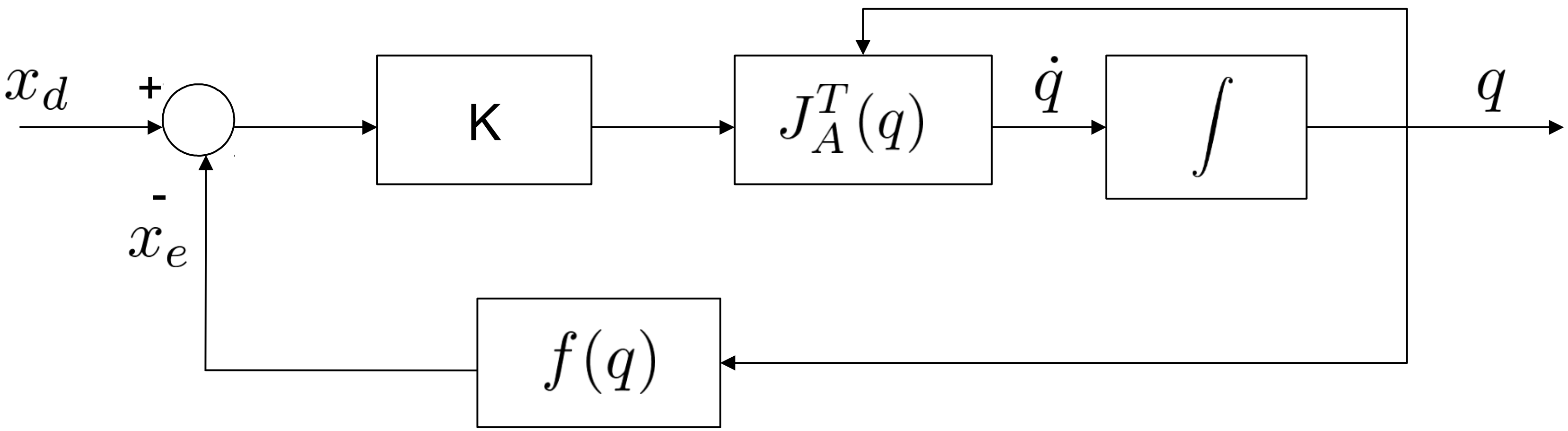
$$\Delta q = -\alpha \frac{f(q)}{f'(q)} \quad \underset{\text{for } \alpha = 1}{\Rightarrow} \quad q_{k+1} = q_k - \frac{f(q_k)}{f'(q_k)}$$

Newton's method for finding the roots of $f(q) = 0$

The convergence is guaranteed only for small initial errors.
In general it can also diverge. A practical solution: $K = \frac{\alpha}{\Delta t} < \frac{1}{\Delta t}$

Transposed Jacobian approach

It's chosen: $\dot{q} = J_A^T K e$ $\qquad K = K^T > 0$



Closed loop (nonlinear dynamics):

$$\dot{e} + J_A J_A^T K e = 0$$

$$V = \frac{1}{2} e^T K e > 0 \quad \Rightarrow \quad \dot{V} = e^T K \dot{e} = -e^T K J_A J_A^T K e \leq 0$$

$$\dot{V} \to 0 \quad \Rightarrow \dot{e} \to 0 \quad \Rightarrow e \to 0$$

Transposed Jacobian approach

Gradient method for minimizing an error function:

$$H(q) = \frac{1}{2}\|e\|^2 = \frac{1}{2}e^T e \quad \Rightarrow \quad \nabla H = \left[\frac{\partial e}{\partial q}\right]^T = -J_A^T e$$

$$q_{k+1} = q_k - \alpha \nabla H(q_k) = \boxed{q_k + \alpha J_A^T e}$$

Transposed Jacobian algorithm:

$$\dot{q} = J_A^T K e \;\Rightarrow\; \Delta q \cong J_A^T K e \Delta t \qquad [K = \alpha/\Delta t]$$

$$\Delta q = \alpha J_A^T e$$

$$\boxed{q_{k+1} = q_k + \alpha J_A^T e}$$

If $\alpha < \frac{1}{\lambda_{\max}}$ with $\lambda_{\max}$ being the maximal eigenvalue of $J_A^T J_A$, then the algorithm is a **contraction mapping** and it will therefore **converge** to the next local minimum.

Comparison

Convergence rate and computational complexity

J^{-1} much faster than J^T (quadratic vs linear convergence)

J^T lighter implementation

Convergence guaranteed only for $\Delta t \rightarrow 0$ without additional restrictions on K

Behavior in the vicinity of singularity:

$J_A^{-1} K e$ grows unbounded and the algorithm explodes.

Safety checks needed

$J_A^T K e$ goes to zero and the algorithm stops.

Pratical implementation

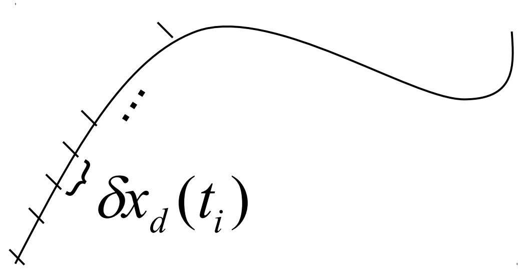
Interpolation of desired Cartesian positions:

Instead of solving inverse kinematics for each pose $x_d(t_i)$ indipendently, divide the Cartesian trajectory in increments $\delta x_d(t_i)$ and solve the inverse kinematics incrementally based on the presented methods

$$\delta x_d = J_A \delta q_d$$

Interpolation of Cartesian trajectory

$x_d(t_i), \qquad i = 0,1, \ldots n$



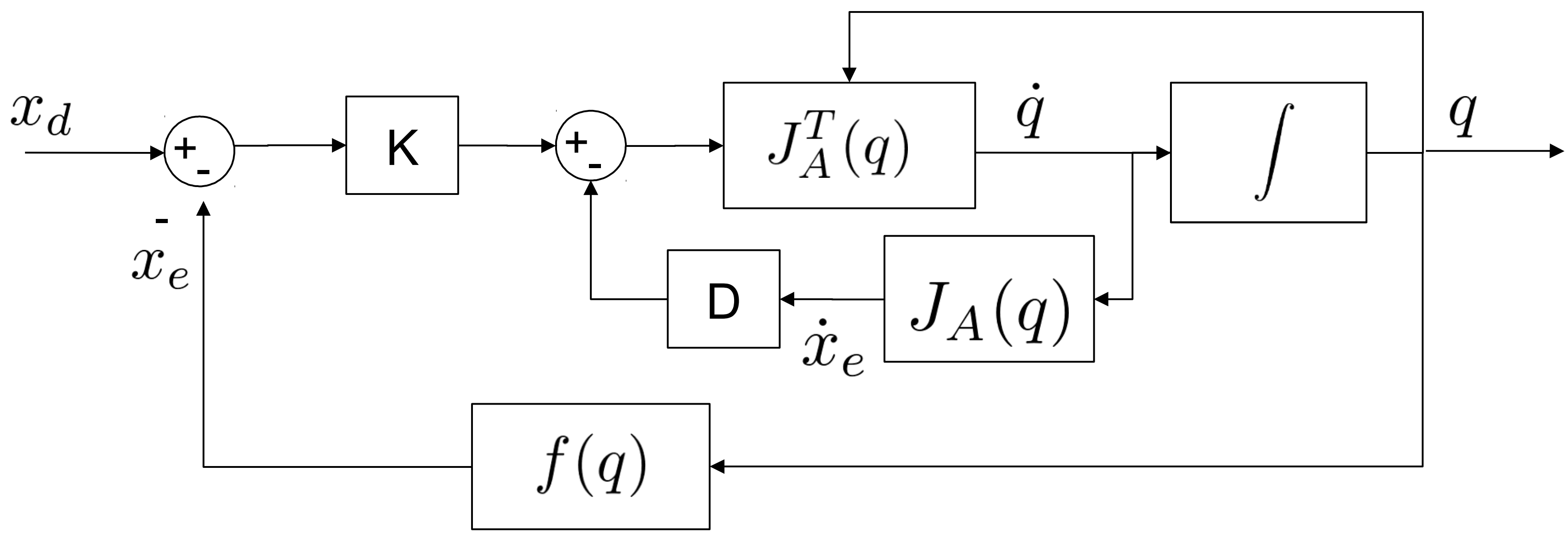
Implementation issues

Convergence criterion $\|x_d - f(q_k)\| \leq \epsilon$

Initial guess q0 ⟶ Different solutions reached

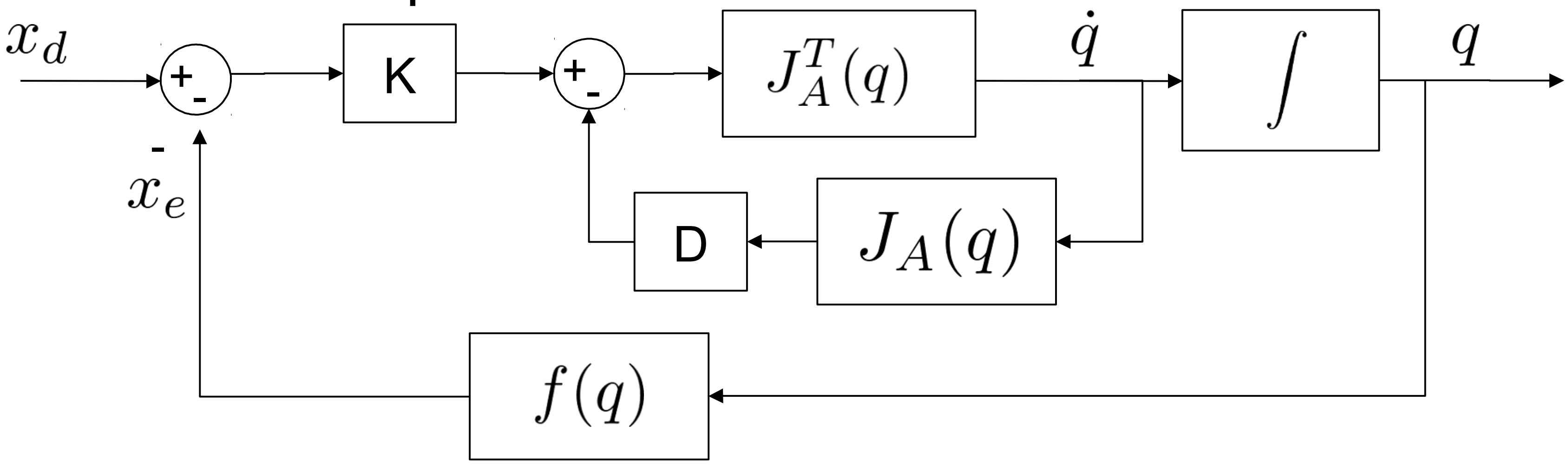
Fixed sampling time ⟶ Instability for high gains K due to the afore mentioned discretization effect and numerical noise

The stability could be improved using a damping term:

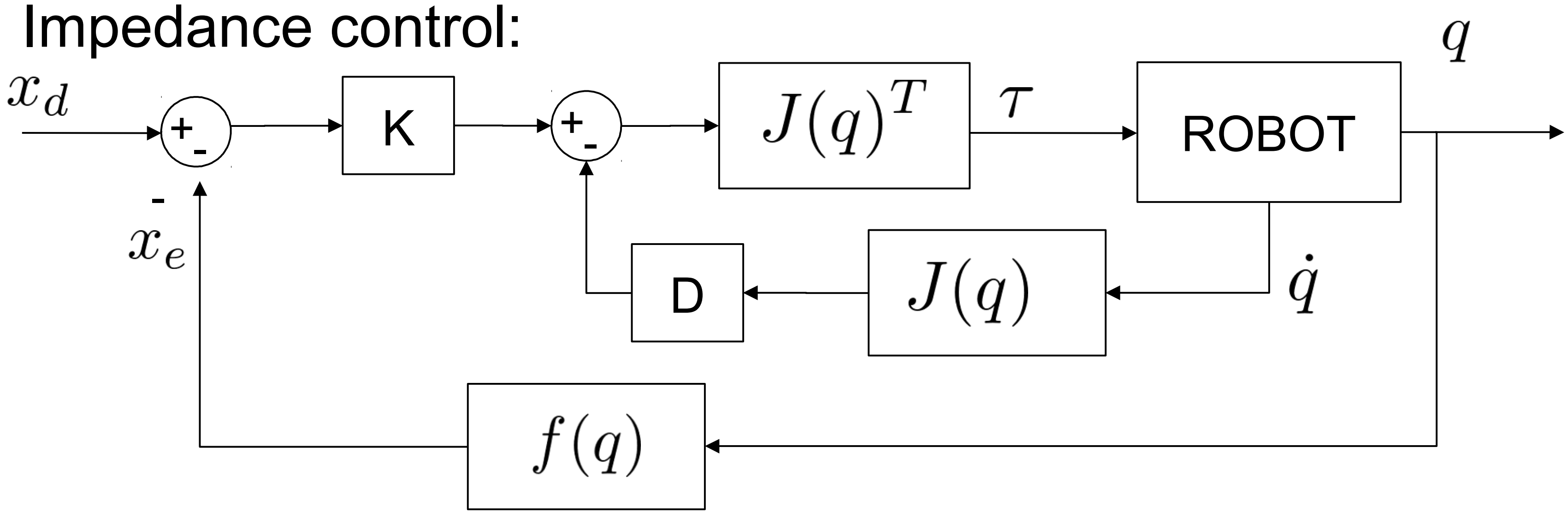


Parallel with impedance control

Jacobian transposed based inverse kinematics:



Impedance control:



References

Analytical methods:

- J. Craig - Introduction to Robotics
- M. Spong – Robot Modeling and Control

Numerical methods:

- B. Siciliano – Robotics. Modelling, Planning and Control
- A. De Luca lecture notes