Inverse of the Jacobian

IK for target pose **t:**
find **q** with F(**q**) = **t** - K(**q**) = **t** – **p** = 0

note:

- target t is constant
- hence Jacobian J for numerical IK
- directly via Jacobian of the forward kinematics K()

$$J = J_{F(q)} = -J_{K(q)}$$

Inverse of the Jacobian

- IK for target pose **t:**
  find **q** with F(**q**) = **t** - K(**q**) = **t** - **p** = 0
- forward kinematics (e.g., 7 DoF robot arm)
  - **p** = K(**q**) with
  - joint parameters **q** (7-dim)
  - **p** and also **t:** pose (6-dim)
- i.e., F() not $\mathbb{R}^N \rightarrow \mathbb{R}^N$

Inverse of the Jacobian

IK: for $\mathbf{t} \in \mathbb{R}^n$ find $\mathbf{q} \in \mathbb{R}^m$ with $F(\mathbf{q}) = |\mathbf{t} - K(\mathbf{q})| = 0$

in general: F() not necessarily $\mathbb{R}^N \rightarrow \mathbb{R}^N$

$\Rightarrow$hence: Jacobian not necessarily square?!?!

$\Rightarrow$use **pseudo-inverse**
(aka Moore-Penrose pseudoinverse)

Moore-Penrose Pseudoinverse

- generalization of the inverse matrix
- for non-square matrices
- twice discovered
  - Moore, 1920
  - Penrose, 1955 (independently)
- e.g., via Singular Value Decomposition (SVD)
- very useful (will see some uses)

Pseudoinverse Properties

n x m matrix A,

pseudoinverse A^+: m x n matrix with

1. $A A^+ A = A$
2. $A^+ A A^+ = A^+$
3. $(AA^+)^T = A A^+$
4. $(A^+ A)^T = A^+ A$

$\forall A \, \exists! A^+$:

1.- 4. are fulfilled

Pseudoinverse Properties

nxm matrix A is rectangular, hence either

- n>m, i.e., "tall"
  - if A has **linearly independent columns**
  - i.e., **A^TA is invertible**

$$\begin{pmatrix} \\ A \\ \\ \end{pmatrix}$$

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

- n<m, i.e., "wide"
  - if A has **linearly independent rows**
  - i.e., **AA^T is invertible**

$$\begin{pmatrix} \quad A \quad \end{pmatrix}$$

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

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

Pseudoinverse Properties

n>m (“tall”) & linearly independent columns

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

aka **left pseudo-inverse** as $A^+A = I$

n<m (“wide”) & linearly independent rows

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

aka **right pseudo-inverse** as $AA^+ = I$

Pseudoinverse & IK

- if IK m>n ("tall" Jacobian)
  - i.e., less system DoF than end-effector DoF
  - then left pseudo-inverse $A^+=(A^TA)^{-1}A^T$
- if IK n>m ("wide" Jacobian)
  - i.e., more system DoF than end-effector DoF
  - then right pseudo-inverse $A^+=A^T(AA^T)^{-1}$

but computation often via SVD

Singular Value Decomposition (SVD)

- given $m \times n$ matrix $\mathbf{A}$
- **SVD** computes matrices **U**, **V**, **W** with
- $\mathbf{A} = \mathbf{U}\,\mathbf{W}\,\mathbf{V}^{\mathrm{T}}$
  - **U**, $m \times m$: orthonormal
  - **W**, $m \times n$: diagonal (aka singular values)
  - **V**, $n \times n$: orthonormal

very useful in general (will see it a few times)

Singular Value Decomposition (SVD)

$$\begin{pmatrix} & & \\ & \mathbf{A} & \\ & & \end{pmatrix} = \begin{pmatrix} & \mathbf{U} & \end{pmatrix} \begin{pmatrix} w_1 & \cdots & 0 & & \vdots \\ \vdots & \ddots & \vdots & \cdots & 0 \\ 0 & \cdots & w_n & & \vdots \\ & \vdots & & & \vdots \\ \cdots & 0 & \cdots & \cdots & 0 \end{pmatrix} \begin{pmatrix} & & \\ & \mathbf{V} & \\ & & \end{pmatrix}^{\mathrm{T}}$$

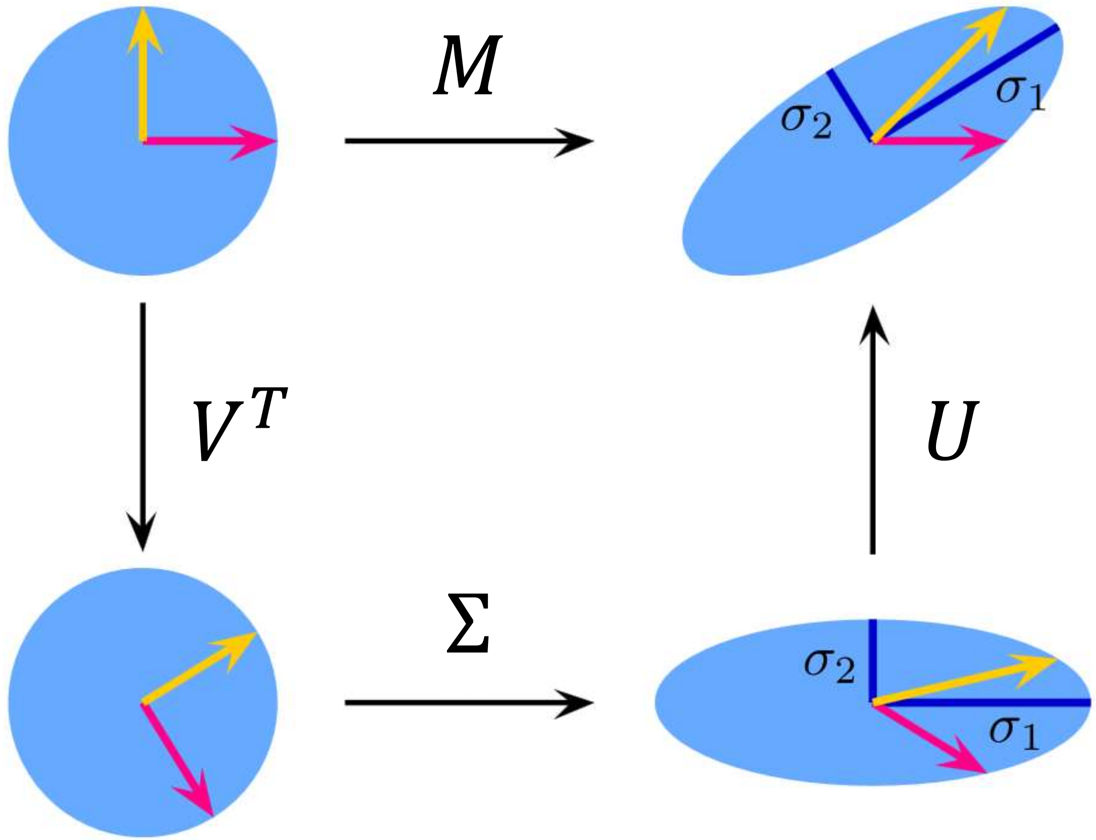
- *A: m*×*n*
- *U: m*×*m*, orthonormal
- *W: m*×*n*, diagonal (aka singular values)
- *V: n*×*n*, orthonormal

Singular Value Decomposition (SVD)

intuition:

M decomposed into

1. rotation V^T
2. scaling Σ
3. rotation U



$$M = U \cdot \Sigma \cdot V^T$$

SVD and Inverse

- orthogonal matrices: inverse = transpose
- **W** is diagonal **=>** $\mathbf{W}^{-1}$ also diagonal
  - with reciprocals of the diagonal entries
  - ie., $W_{ij}^{-1} = 1/\, W_{ij}$

$$\mathbf{A}^{-1} = (\mathbf{UWV}^{T})^{-1} = (\mathbf{V}^{T})^{-1}\, \mathbf{W}^{-1}\, \mathbf{U}^{-1}$$

$$= \mathbf{V}\, \mathbf{W}^{-1}\, \mathbf{U}^{T}$$

=> *very easy to compute (given the SVD)*

SVD and Pseudo-Inverse

SVD of $\mathbf{A} = \mathbf{U}\,\mathbf{W}\,\mathbf{V}^T$

pseudo-inverse $\mathbf{A}^+ = \mathbf{V}\,\mathbf{W}^+\,\mathbf{U}^T$

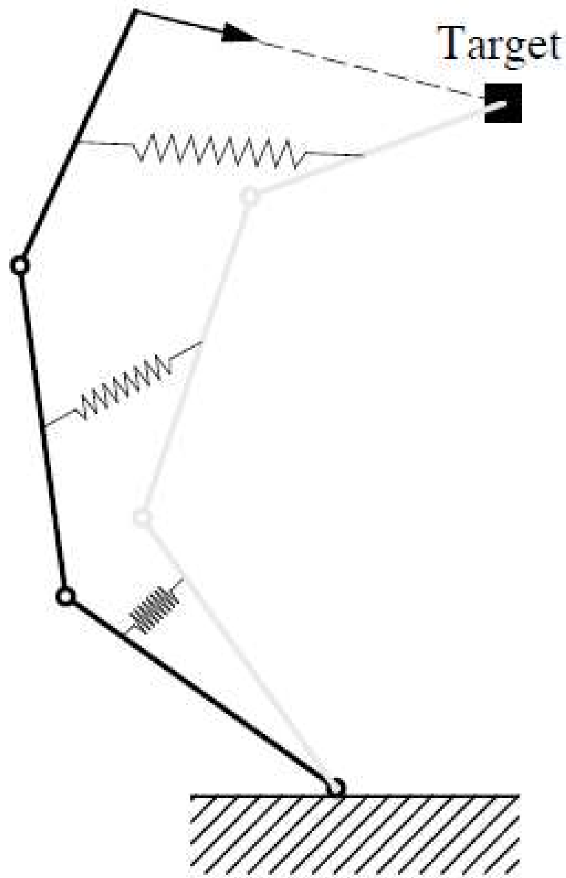
with $\mathbf{W}^+$

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

(note: diagonal elements can be 0, too)

Newton & IK

$$q_{k+1} = q_k + J(q_k)^{+}[t - K(q_k)]$$



- iterations needed
- often: speed critical
- trick
  - transpose J^T
  - instead of (pseudo)inverse

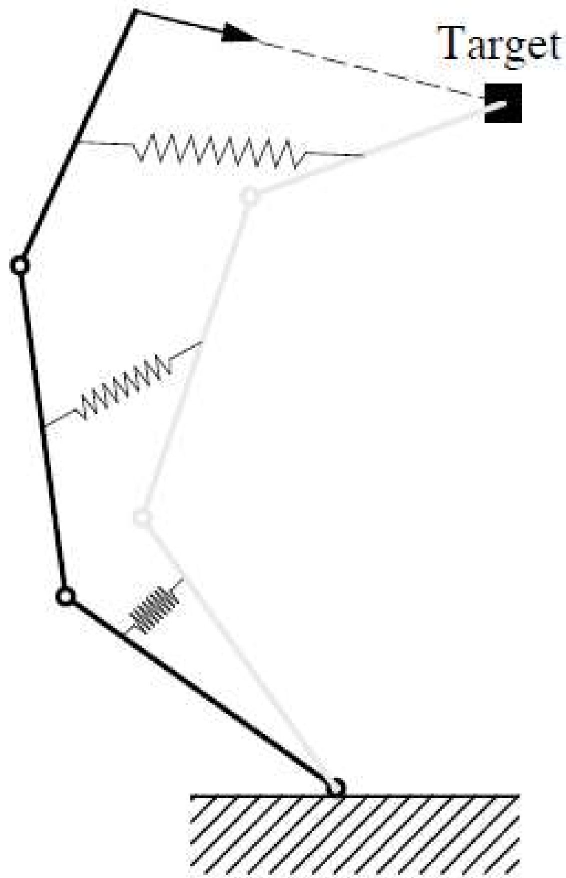
$$q_{k+1} = q_k + J(q_k)^{T}[t - K(q_k)]$$

Newton & IK

- unlikely to start close to root
- step-wise motion towards target needed anyway

⇒only take a small partial step

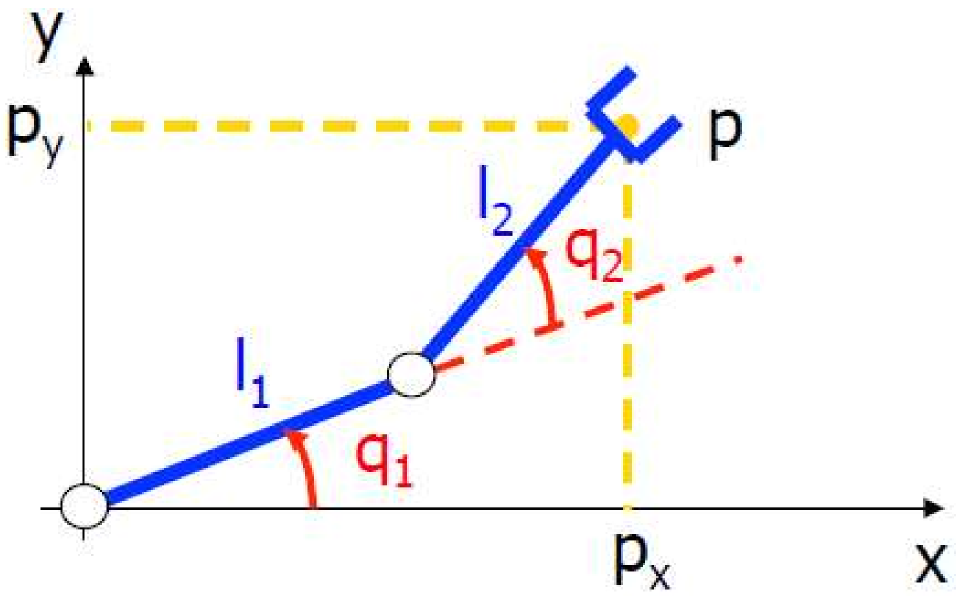
⇒small factor α



$$q_{k+1} = q_k + \alpha \cdot \Delta q$$
$$\Delta q = J(q_k)^{+}[t - K(q_k)]$$

Example: Numerical IK

planar (2D) 2 link arm with 2 rotational joints

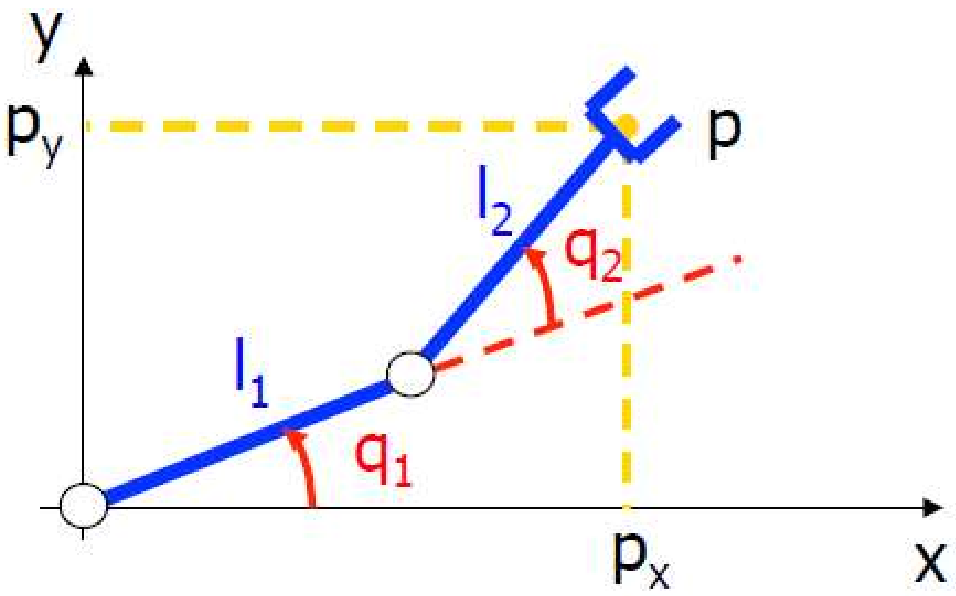


forward kinematics: $\mathbf{p} = f(\mathbf{q})$

- $p_x = l_1 \cos(q_1) + l_2 \cos(q_1+q_2)$
- $p_y = l_1 \sin(q_1) + l_2 \sin(q_1+q_2)$

(note: closed form inverse kinematics easy; just as example)

Example: Numerical IK



numerical IK =>

need Jacobian

$$J_{ij} = \frac{\partial p_i}{\partial q_j}$$

notational abbreviations:

sin / cos(q_1) = s_1 / c_1

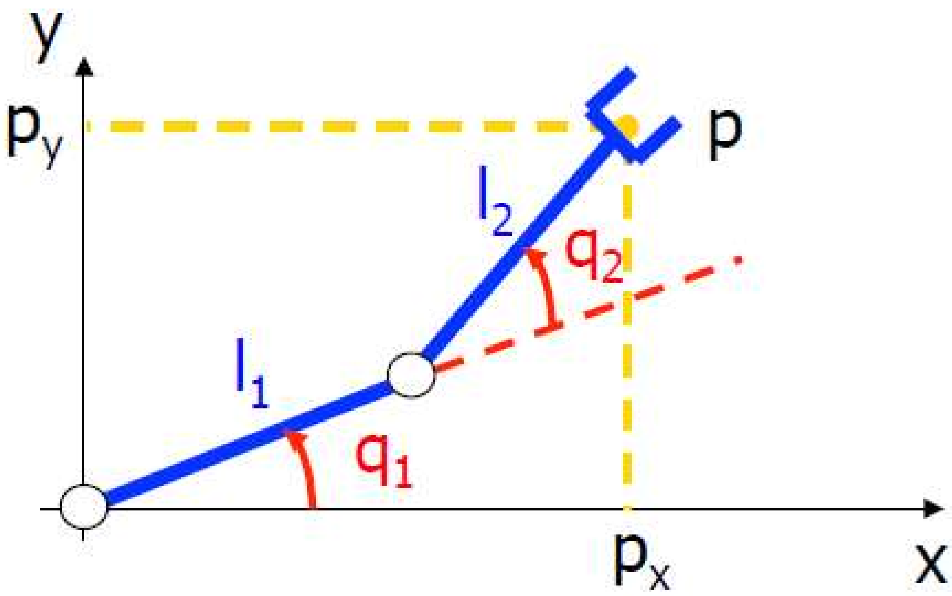
sin / cos(q_1 + q_2) = s_{12}/ c_{12}

i.e., forward kinematics

$p_x = l_1 c_1 + l_2 c_{12}$

$p_y = l_1 s_1 + l_2 s_{12}$

Example: Numerical IK

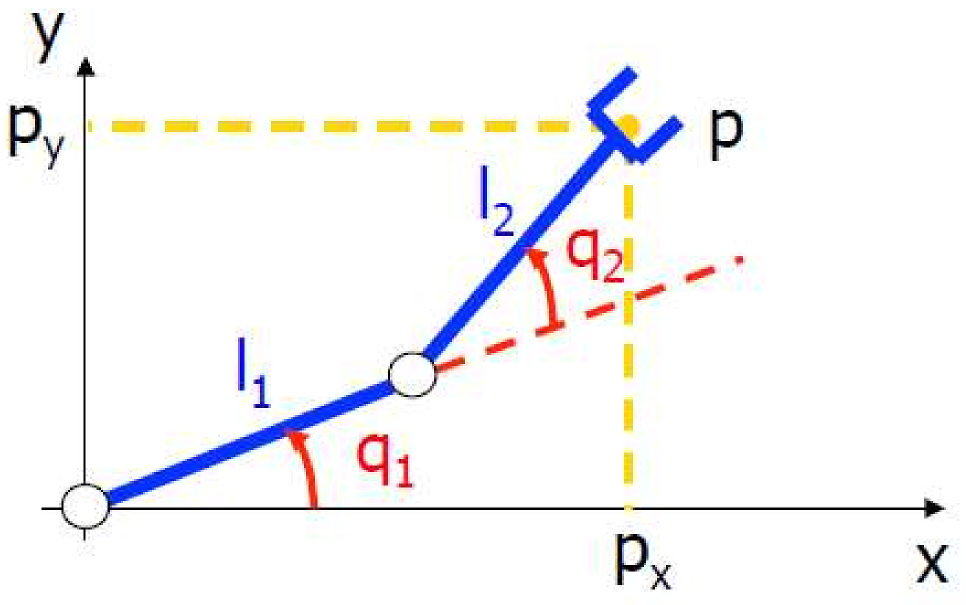


Jacobian

$$J_{\mathrm{ij}} = \frac{\partial p_{\mathrm{i}}}{\partial q_{\mathrm{j}}}$$

$$J = \begin{pmatrix} \dfrac{\partial p_{\mathrm{y}}}{\partial q_1} & \dfrac{\partial p_{\mathrm{y}}}{\partial q_2} \\ \dfrac{\partial p_{\mathrm{x}}}{\partial q_1} & \dfrac{\partial p_{\mathrm{x}}}{\partial q_2} \end{pmatrix} = \begin{pmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{pmatrix}$$

Example: IK with Newton

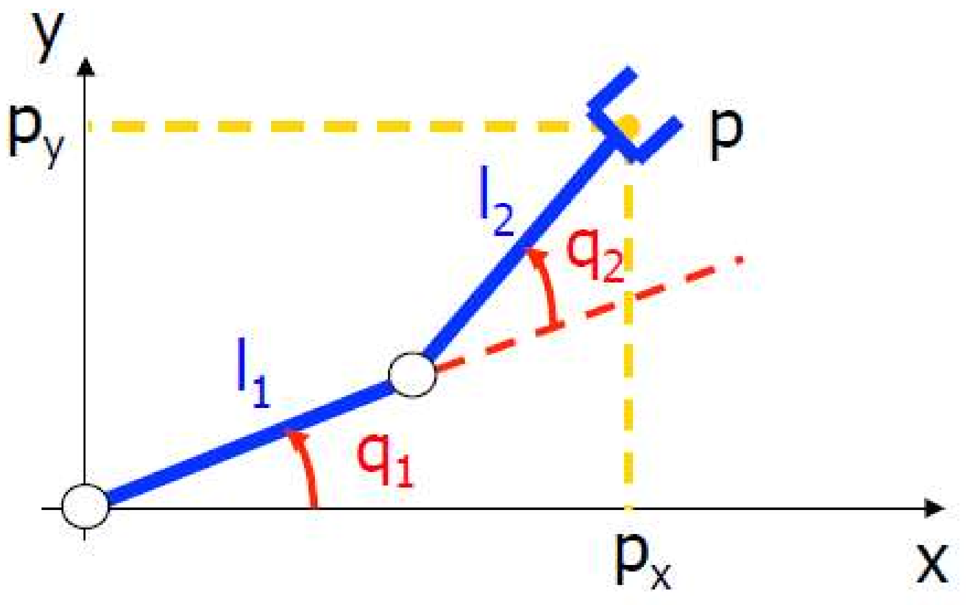


$$J^{-1} = \begin{pmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{pmatrix}^{-1}$$

Newton: $$q_{k+1} = q_k + \alpha J(q_k)^{-1}\left[p_t - f(q_k)\right]$$

(just take an initial guess and iterate)

Example: IK with Newton(2)



$$J^T = \begin{pmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{pmatrix}^T$$

$$= \begin{pmatrix} -l_1 s_1 - l_2 s_{12} & l_1 c_1 \\ -l_2 s_{12} + l_2 c_{12} & l_2 c_{12} \end{pmatrix}$$

use Transpose: $q_{k+1} = q_k + \alpha J(q_k)^T \left[p_t - f(q_k) \right]$

(just take an initial guess and iterate)

Alternative Option for IK: Gradient Descent

- in general useful optimization method
- to find minimum of F() : $\mathbb{R}^N \rightarrow \mathbb{R}$
- i.e., $\mathbf{x}^* \in \mathbb{R}^N$: min F($\mathbf{x}^*$)
- by taking steps "down" the gradient

Gradient Descent

F() : $\mathbb{R}^N \rightarrow \mathbb{R}$

gradient of F()

$$\nabla F(x) = \begin{bmatrix} \frac{\partial F(x)}{\partial x_1} \\ \vdots \\ \frac{\partial F(x)}{\partial x_N} \end{bmatrix}$$

$$\nabla F(x) = J^T$$

note: Jacobian is a generalization of the gradient, respectively gradient a special case of the Jacobian (for F(): $\mathbb{R}^N \rightarrow \mathbb{R}$)

Gradient Descent

iteration to find **x***$\in \mathbb{R}^N$: min F(**x***)

$$x_{k+1} = x_k - \alpha_k \nabla \mathrm{F}(x_k)$$

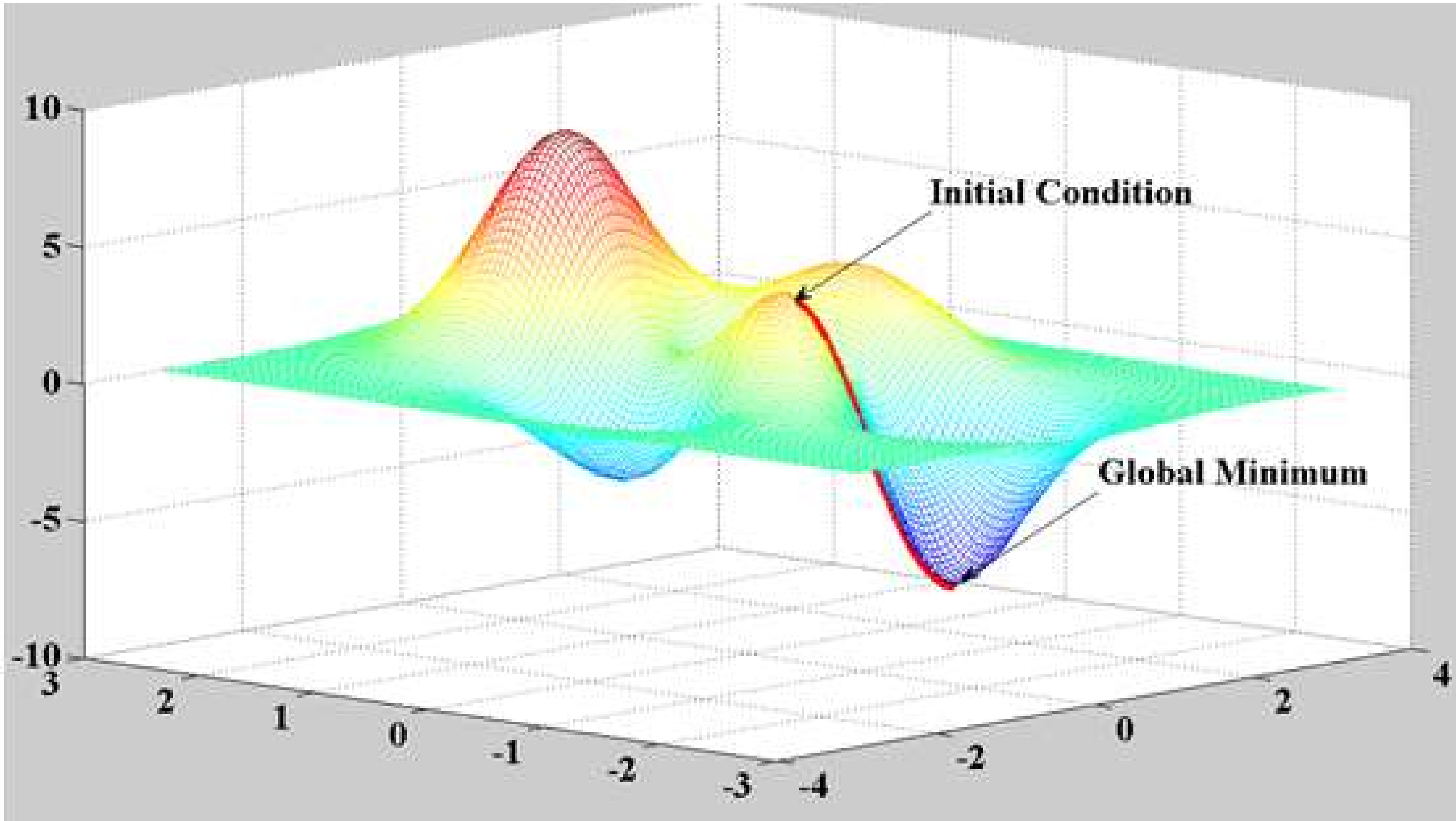
with a small factor α that can be

- constant ($\alpha_k = \alpha$)
- or adaptive using one of many possible strategies, e.g.,
- Barzilai-Borwein method:

$$\alpha_k = \frac{(x_k - x_{k-1})^T \left[\nabla \mathrm{F}(x_k) - \nabla \mathrm{F}(x_{k-1})\right]}{|\nabla \mathrm{F}(x_k) - \nabla \mathrm{F}(x_{k-1})|^2}$$

Gradient Descent

note: initial guess is important



option: randomization with multiple guesses

Gradient Descent

- can e.g. in general be used to solve
- a system of (noisy) linear equations ***Ax-b = 0***
- formulated as least squares problem

$$F(x) = |Ax - b|^2$$

gradient: $\nabla F(x) = 2A^T\left(Ax - b\right)$

Gradient Descent & IK

for IK: minimize error function E(q)

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

gradient of E(q)

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

Gradient Descent & IK

i.e., iteration function:

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

with step-width parameter α