Humanoid robots - Kinematics II
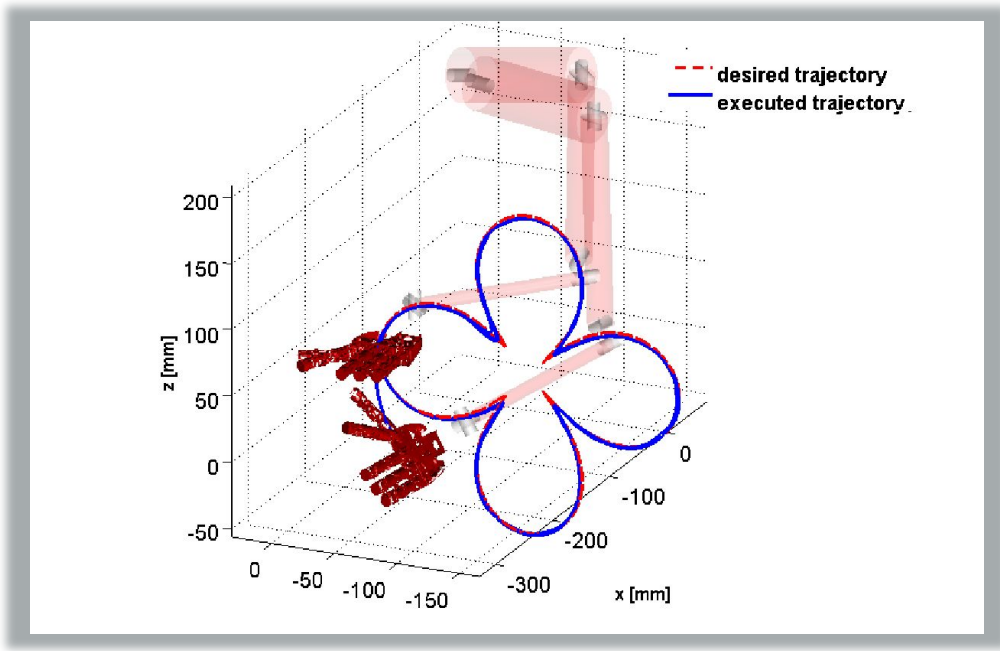
Doc. Mgr. Matěj Hoffmann, Ph.D.

Kinematics - forward and inverse

Study of properties of motion (position, velocity, acceleration) without considering body inertias and internal/external forces.

The Problem

$$\begin{cases} \mathbf{x} = \mathbf{f}\left(\mathbf{q}\right) \\ \mathbf{q} \in \mathbb{R}^n \\ \mathbf{x} \in \mathbb{R}^6 \end{cases}$$
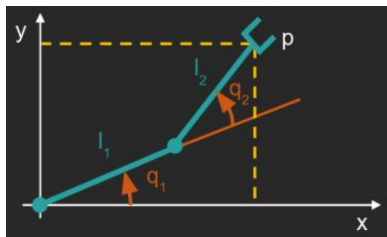
$$\mathbf{q} \overset{?}{=} \mathbf{f}^{-1}\left(\mathbf{x}\right)$$

Slide source: https://github.com/vvv-school/vvv18/blob/master/material/kinematics/kinematics.pdf. Courtesy Ugo Pattacini.

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

Forward and inverse kinematics – recap

n=7, m=6 (x,y,z,ϕ,θ,ψ)

Consider a general case: $\mathbf{q} \in \mathbb{R}^n$ $\quad \mathbf{x} \in \mathbb{R}^m$

n=2, m=3 (x,y,ϕ)   n=6, m=6 (x,y,z,ϕ,θ,ψ)   n=5, m=6 (x,y,z,ϕ,θ,ψ)

What is n and m?

	Mapping type
$\mathbf{x} = \mathbf{f}(\mathbf{q})$	
$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x})$	

Forward and inverse kinematics – recap

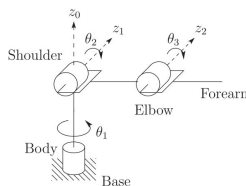Consider a general case: $\mathbf{q} \in \mathbb{R}^n$ $\mathbf{x} \in \mathbb{R}^m$

n=2, m=3 (x,y,φ)   n=6, m=6 (x,y,z,φ,θ,ψ)   n=5, m=6 (x,y,z,φ,θ,ψ)
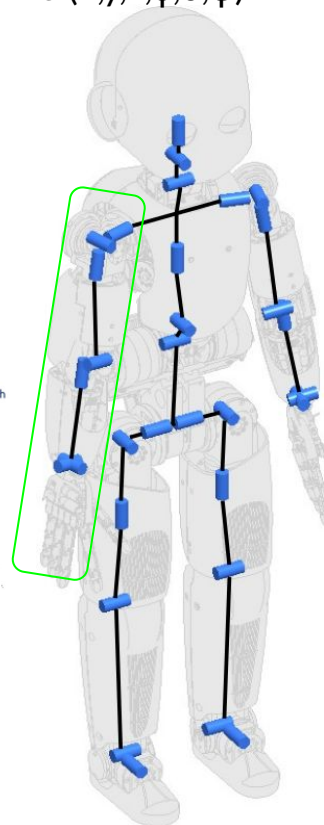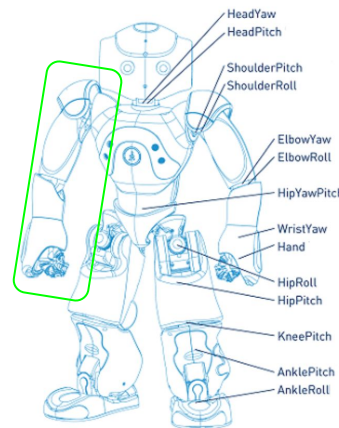
What is n
and m?

	Mapping type
$\mathbf{x} = \mathbf{f(q)}$	Many-to-one (for n>= m)
$\mathbf{q} = \mathbf{f^{-1}(x)}$	One-to-many (for n>1)

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

Inverse kinematics – analytic (closed-form) solutions
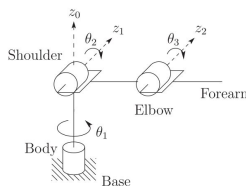
$$\begin{cases} \mathbf{x} = \mathbf{f}\left(\mathbf{q}\right) \\ \mathbf{q} \in \mathbb{R}^n \\ \mathbf{x} \in \mathbb{R}^6 \end{cases}$$

$$\mathbf{q} = \mathbf{f}^{-1}\left(\mathbf{x}\right)$$

TABLE 1
Number of Analytic Solutions for
6-Degree-of-Freedom Systems

n	Upper bound on solutions
<6	
>6	
6R, 5RP	
4R2P, 6R with S joint	
3R3P	

Tolani, D., Goswami, A., & Badler, N. I. (2000). Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical models*, *62*(5), 353-388.

Inverse kinematics – analytic (closed-form) solutions

$$\begin{cases} \mathbf{x} = \mathbf{f}\left(\mathbf{q}\right) \\ \mathbf{q} \in \mathbb{R}^n \\ \mathbf{x} \in \mathbb{R}^6 \end{cases}$$

$$\mathbf{q} = \mathbf{f}^{-1}\left(\mathbf{x}\right)$$

TABLE 1

Number of Analytic Solutions for 6-Degree-of-Freedom Systems

n	Upper bound on solutions
<6	0
>6	∞
6R, 5RP	16
4R2P, 6R with S joint	8
3R3P	2

Tolani, D., Goswami, A., & Badler, N. I. (2000). Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical models*, *62*(5), 353-388.

Kinematic redundancy

- Whenever the limb mobility, determined by the number of limb joints n, exceeds the DoFs of the end link (six), the limb is characterized as kinematically redundant.
- Some humanoid robots are equipped with kinematically redundant, 7-DoF arms. Such robots can control the position of their elbows without affecting thereby the instantaneous motion of the hands. Thus, they attain the capability to perform tasks in cluttered environments avoiding collisions with their elbows, similar to humans.
- Also, there are humanoids that comprise 7-DoF legs. With proper control, their gait appears more human-like than that of robots with 6-DoF legs.
- The difference r = n − 6 is referred to as the *degree of redundancy* (DoR).

Section 2.7 in Nenchev, D. N., Konno, A., & Tsujita, T. (2018). Humanoid robots: Modeling and control. Butterworth-Heinemann.
Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

<https://youtu.be/sZYBC8Lrmdo>
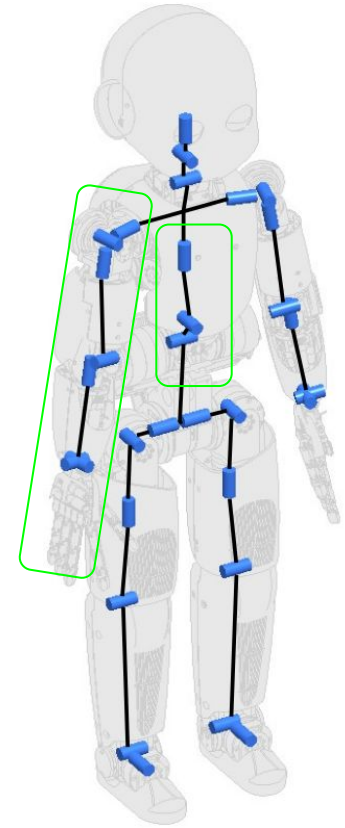
Inverse kinematics – closed form

- fast at runtime
- suited for platforms designed with IK solutions in mind (joint alignment - e.g. spherical wrist)
- difficult for "general" or redundant robots
  - laborious to develop a solution
  - complicated to incorporate constraints, especially dynamic constraints
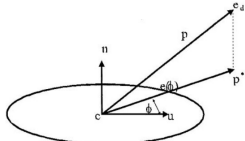- does not provide trajectories (motion control)

Wrist position and E-E pose
Inverse solutions for an articulated 6R robot [PUMA 560]

LEFT UP — RIGHT DOWN

4 inverse solutions (for the position of the wrist center only)

LEFT UP — RIGHT UP

8 inverse solutions considering the complete E-E pose (spherical wrist: 2 alternative solutions for the last 3 joints)

TABLE 1
Number of Analytic Solutions for
6-Degree-of-Freedom Systems

n	Upper bound on solutions
<6	0
>6	∞
6R, 5RP	16
4R2P, 6R with S joint	8
3R3P	2

FIG. 3. Finding the elbow position that is the closest possible to a desired position.

TABLE 2
Summary of Methods Used When the Goal Is Reachable

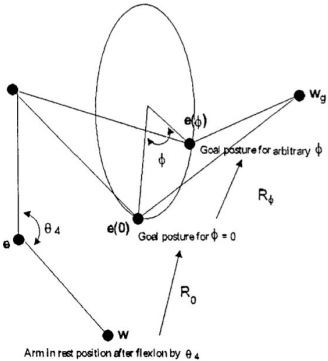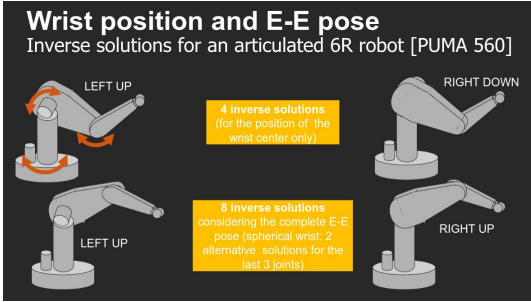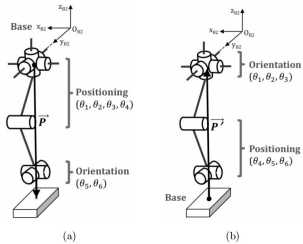|	Goal reachable (joint limits off)	Goal reachable (joint limits on)
Position	Analytic	Analytic
Position and orientation	Analytic	Analytic
Position and partial orientation	Analytic	Analytic + 2DOF unconstrained optimization
Aiming	Analytic	Analytic if θ_4 given 2DOF unconstrained optimization otherwise

FIG. 5. Decomposing R_1 into R_0 and R_ϕ.

Fig. 5. (a) Forward Decoupling of the Right Leg of a Hubo KHR-4 Robot, (b) Reverse Decoupling of the Right Leg of a Hubo KHR-4 Robot.
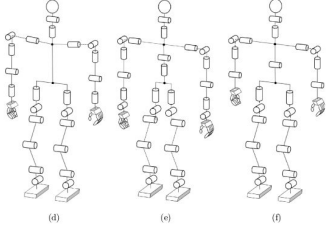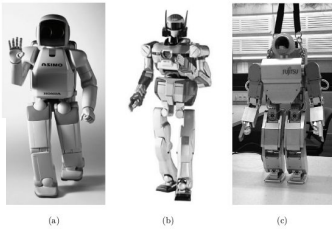
Fig. 4. (a) HONDA ASIMO Robot and its associated kinematic diagram in (d), (b) AIST HRP-2 Robot and its associated kinematic diagram in (e), and (c) Fujitsu HOAP-2 Robot and its associated kinematic diagram in (f).

Tolani, D., Goswami, A., & Badler, N. I. (2000). Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical models*, *62*(5), 353-388.

Park, H. A., Ali, M. A., & Lee, C. G. (2012). Closed-form inverse kinematic position solution for humanoid robots. *International Journal of Humanoid Robotics*, *9*(03), 1250022.

Inverse kinematics – iterative method

- iCub Cartesian solver

Can you decode
what is going on?

$$\tilde{q}_d = \arg \min_{q \in \mathbb{R}^n} \left(\left\| \alpha_d - K_\alpha (q) \right\|^2 + \lambda \cdot \left(q_{\text{rest}} - q \right)^T W \left(q_{\text{rest}} - q \right) \right)$$

$$\text{s.t.} \begin{cases} \left\| x_d - K_x (q) \right\|^2 < \varepsilon \\ q_L < q < q_U \\ \text{other obstacles ...} \end{cases}$$

where
$K_\alpha(q)$ .... forward kinematics in orientation
$K_x(q)$ .... forward kinematics in position
q_{rest} ... preferred joint configuration
W ... diagonal matrix of weighting factors

➤ **Quick convergence**: real-time compliant, < **20 ms**
➤ **Scalability**: n can be high and set on the fly
➤ **Singularities handling**: no Jacobian inversion
➤ **Joints bound handling**: no explicit boundary functions
➤ **Tasks hierarchy**: no use of null space
➤ **Complex constraints**: intrinsically nonlinear

Pattacini, U., Nori, F., Natale, L., Metta, G., & Sandini, G. (2010). An experimental evaluation of a novel minimum-jerk Cartesian controller for humanoid robots. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

https://github.com/vvv-school/vvv18/blob/master/material/kinematics/kinematics.pdf, courtesy Ugo Pattacini

How do we get to the target position?

Figure 2. Experimental apparatus for measuring arm trajectories in a horizontal plane. T_1 to T_6 are the LED targets. θ_1 and θ_2 are, respectively, the subject's shoulder and elbow joint angles. Movement of the handle was measured by way of the potentiometers, P_1 and P_2.
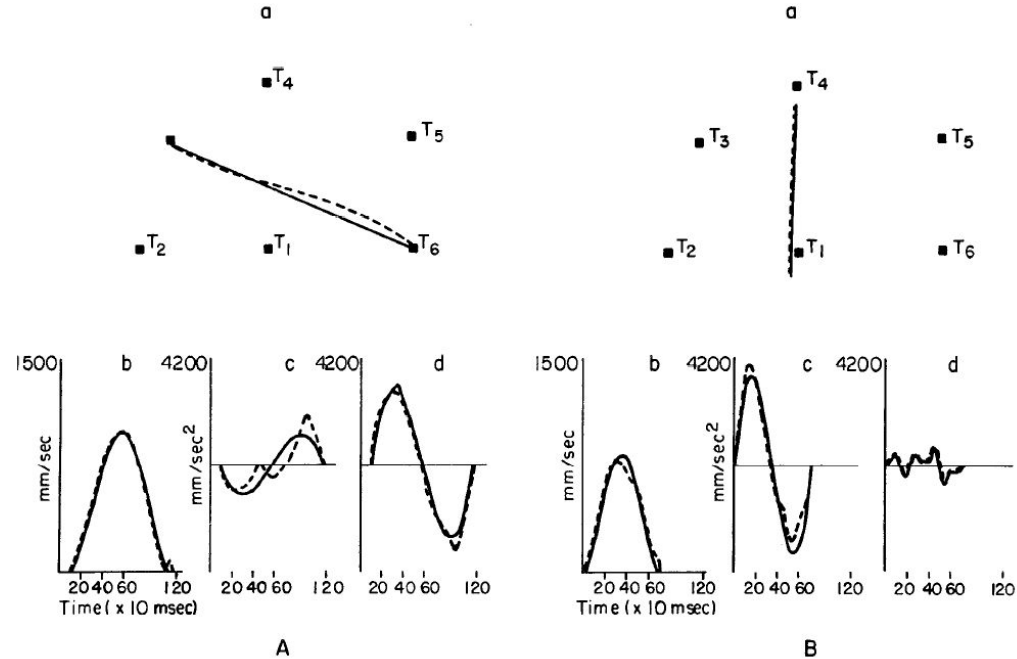
Figure 3. Overlapped predicted (*solid lines*) and measured (*dashed lines*) hand paths (*a*), speeds (*b*), and acceleration components along the *y*-axis (*c*), and along the *x*-axis (*d*) for two unconstrained point-to-point movements. *A*, A movement between targets 3 and 6. *B*, A movement between targets 1 and 4.

Flash, T., & Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience*, *5*(7).
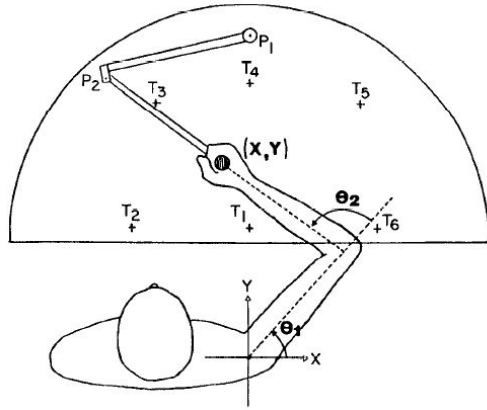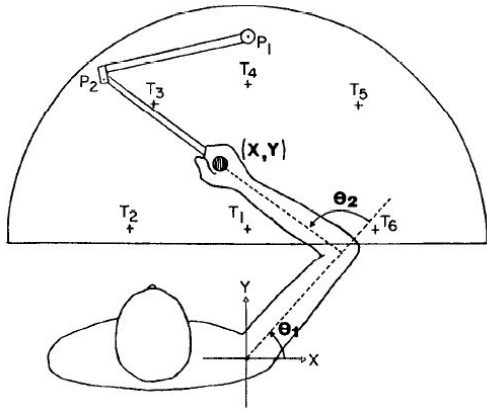
Figure 2. Experimental apparatus for measuring arm trajectories in a horizontal plane. T_1 to T_6 are the LED targets. θ_1 and θ_2 are, respectively, the subject's shoulder and elbow joint angles. Movement of the handle was measured by way of the potentiometers, $P1$ and $P2$.

Figure 4. Measured and predicted shoulder, S, and elbow, E, joint angles and angular velocity profiles. A, A movement between targets 3 and 6. B, A movement between targets 2 and 5.

Flash, T., & Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience*, *5*(7).
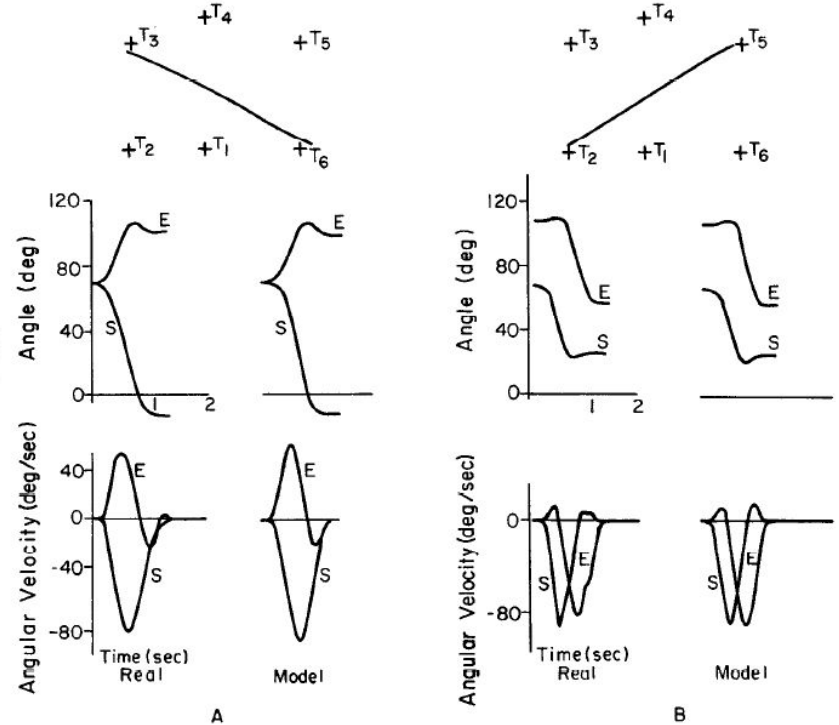
Minimum jerk trajectory
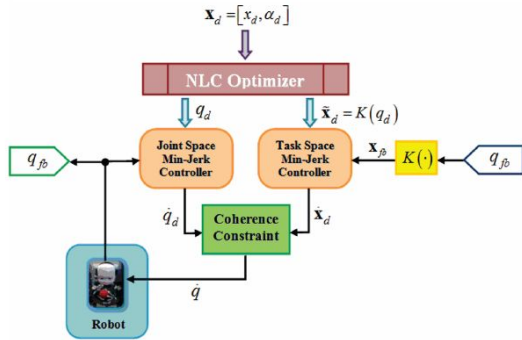
Fig. 3. Multi-Referential scheme. $K(\cdot)$ is the forward kinematic map.
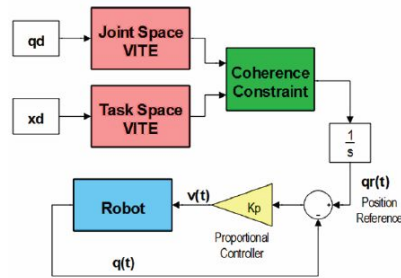
Fig. 4 Schematic of implemented multi-referential VITE controllers in the work of Hersch and Billard [2].
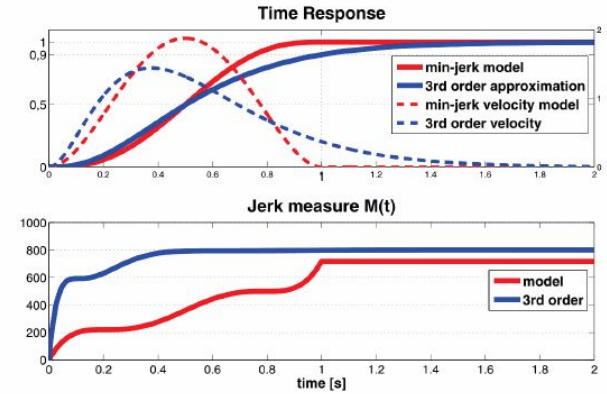
Fig. 5. Comparison between the responses of the 3rd order dynamical time-invariant system found through the minimization (blue) and the responses of the minimum-jerk model (red).

Pattacini, U., Nori, F., Natale, L., Metta, G., & Sandini, G. (2010). An experimental evaluation of a novel minimum-jerk Cartesian controller for humanoid robots. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Other stakeholders

- Who else is interested in inverse kinematics of anthropomorphic creatures?
- Computer graphics / animation / game industry!

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

Other stakeholders

- Who else is interested in inverse kinematics of anthropomorphic creatures?
- Computer graphics / animation / game industry!

https://youtu.be/0a9qIj7kwiA?t=321

https://youtu.be/SHpImEc6iv0?t=156

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024
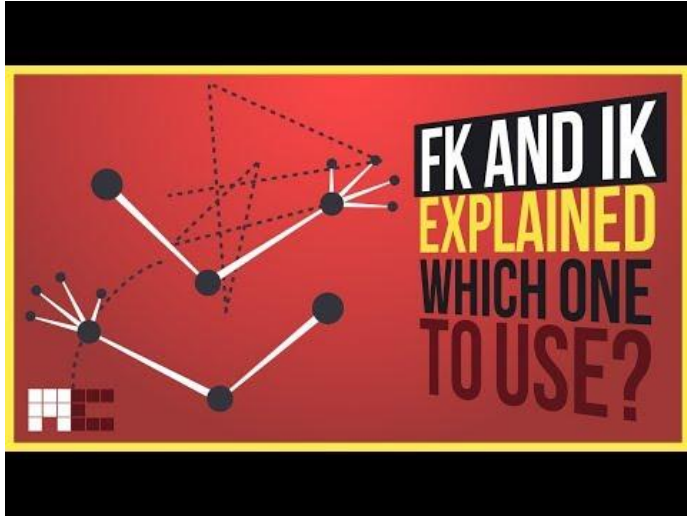
Inverse kinematics – robotics vs. animations

Humanoid robots ~ human-like characters in games etc.

Which problem is harder?

Constraints seem less strict for the animation industry:

- joint limits (position, velocity…)
- self-collisions
- [balance]

Differential kinematics

$$\nu = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \cdots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \cdots & \frac{\partial y}{\partial q_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \omega_z}{\partial q_1} & \frac{\partial \omega_z}{\partial q_2} & \cdots & \frac{\partial \omega_z}{\partial q_n} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} = J(q) \cdot \dot{q}$$

kinematics

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

differential kinematics

$$\dot{\mathbf{x}} = \mathbf{J(q)}\dot{\mathbf{q}} \qquad \dot{\mathbf{q}} = \mathbf{J}^{-1}\dot{\mathbf{x}}$$

- Velocity relationships relating linear and angular velocities of the end effector to the joint velocities.
- "Mathematically, the forward kinematic equations define a function from the configuration space of joint positions to the space of Cartesian positions and orientations." (Spong et al., pg. 101)
- The velocity relationships are then determined by the **Jacobian** of this function. The Jacobian is a matrix that generalizes the notion of the ordinary derivative of a scalar function. The Jacobian is one of the most important quantities in the analysis and control of robot motion. It arises in virtually every aspect of robotic manipulation:
  - planning and execution of smooth trajectories
  - determination of singular configurations
  - execution of coordinated anthropomorphic motion
  - derivation of the dynamic equations of motion
  - transformation of forces and torques from the end effector to the manipulator joints (Spong et al., pg. 101)
- Today: iterative methods for IK, singularity and manipulability.
- Basics of Jacobian, manipulability etc. covered in Robotics (B3B33ROB1).

Chapter 4 in Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2020). *Robot modeling and control*. John Wiley & Sons.

Tutorial

Manipulator Differential Kinematics

Part I: Kinematics, Velocity, and Applications
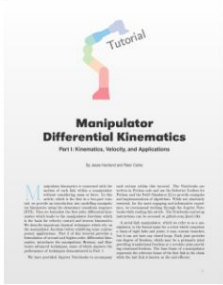
By Jesse Haviland and Peter Corke

Haviland, J., & Corke, P. (2023). Manipulator Differential Kinematics: Part I: Kinematics, Velocity, and Applications. *IEEE Robotics & Automation Magazine*. https://doi.org/10.1109/MRA.2023.3270228

A Tutorial on Manipulator Differential Kinematics

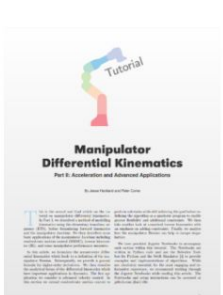| powered by | robotics toolbox | python | 3.7 | 3.8 | 3.9 | 3.10 | 3.11 | collection | QUT Robotics | License | MIT |

By Jesse Haviland and Peter Corke

This repository contains a collection of Jupyter Notebooks designed to accompany the two-part Tutorial on Manipulator Differential Kinematics published in the IEEE Robotics and Automation Magazine. Each Notebook corresponds to a section within the tutorial articles. The notebooks are easily extensible to encourage experimentation. The intention is that you read a section of the tutorial and then complete the corresponding Notebook. The articles can be accessed via PDF via the links below.

Part 1: Kinematics, Velocity, and Applications
- Access Published Article via IEEE Xplore
- Access Article Preprint from Arxiv

Jupyter Notebooks	Open in Colab Link
1 Manipulator Kinematics	Open in Colab
2 The Manipulator Jacobian	Open in Colab
3 Resolved-Rate Motion Control	Will not run on Colab
4 Numerical Inverse Kinematics	Open in Colab
5 Manipulator Performance Measures	Open in Colab

Part 2: Acceleration and Advanced Applications
- Access Published Article via IEEE Xplore
- Access Article Preprint from Arxiv

Jupyter Notebooks	Open in Colab Link
1 The Manipulator Hessian	Open in Colab
2 Higher Order Derivatives	Open in Colab
3 Analytic Forms	Open in Colab
4 Null-Space Projection for Motion Control	Will not run on Colab
5 Quadratic Programming for Motion Control	Will not run on Colab
6 Advanced Numerical Inverse Kinematics	Open in Colab

Tutorial

Manipulator Differential Kinematics

Part II: Acceleration and Advanced Applications

By Jesse Haviland and Peter Corke

Haviland, J., & Corke, P. (2023). Manipulator Differential Kinematics: Part 2: Acceleration and Advanced Applications. *IEEE Robotics & Automation Magazine*. https://doi.org/10.1109/MRA.2023.3270221

https://github.com/jhavl/dkt

Geometrical Jacobian

$$\nu = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \cdots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \cdots & \frac{\partial y}{\partial q_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \omega_z}{\partial q_1} & \frac{\partial \omega_z}{\partial q_2} & \cdots & \frac{\partial \omega_z}{\partial q_n} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} = J(q) \cdot \dot{q}$$

twist

- it is a function of the joint configuration q
- contains all of the partial derivatives of f , relating every joint angle to every velocity
- tells us how small changes in joint space will affect the end-effector's position in Cartesian space
- columns: how each component of velocity changes when the configuration (i.e., angle) of a particular joint changes
- rows: how movement in each joint affects a particular component of velocity

3.3.1 in Nikolaus Correll, Bradley Hayes, Christoffer Heckman and Alessandro Roncone. Introduction to Autonomous Robots: Mechanisms, Sensors, Actuators, and Algorithms, MIT Press, 2022 (forthcoming).
https://github.com/Introduction-to-Autonomous-Robots/Introduction-to-Autonomous-Robots

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

Analytical Jacobian

$$\mathbf{x} = \mathbf{f}(\mathbf{q}) \overset{d/dt}{\Longrightarrow} \dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$$

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \dfrac{\partial f_1}{\partial q_1} & \cdots & \dfrac{\partial f_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_m}{\partial q_1} & \cdots & \dfrac{\partial f_m}{\partial q_n} \end{bmatrix}$$

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_P \\ \mathbf{J}_\phi \end{bmatrix} \dot{\mathbf{q}}$$

Matej Hofmann, Humanoid robots, FEE CTU in Prague, 2024

Slide source: https://github.com/FEE-CTU/school/v-2024/blob/master/material/kinematics/kinematics.pdf. Courtesy Ugo Pattacini.

Planar arm with 2 rotational joints – forward kinematics (position only)
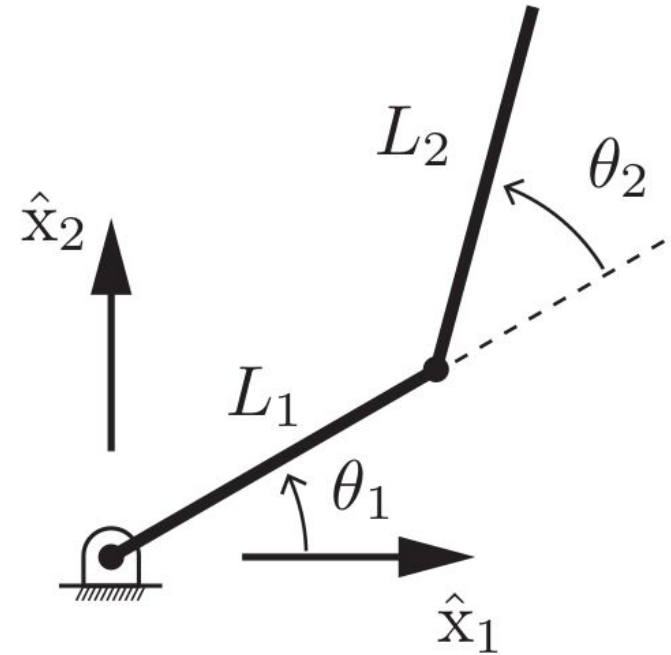
$$x_1 = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2)$$
$$x_2 = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2).$$

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

Planar arm with 2 rotational joints – forward differential kinematics (position only)

$$x_1 = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2)$$
$$x_2 = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2).$$

Differentiating both sides with respect to time yields

$$\dot{x}_1 = -L_1 \dot{\theta}_1 \sin \theta_1 - L_2 (\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_1 + \theta_2)$$
$$\dot{x}_2 = L_1 \dot{\theta}_1 \cos \theta_1 + L_2 (\dot{\theta}_1 + \dot{\theta}_2) \cos(\theta_1 + \theta_2),$$

which can be rearranged into an equation of the form $\dot{x} = J(\theta)\dot{\theta}$:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -L_1 \sin \theta_1 - L_2 \sin(\theta_1 + \theta_2) & -L_2 \sin(\theta_1 + \theta_2) \\ L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) & L_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}. \quad (5.1)$$

Writing the two columns of $J(\theta)$ as $J_1(\theta)$ and $J_2(\theta)$, and the tip velocity $\dot{x}$ as v_{tip}, Equation (5.1) becomes

$$v_{\text{tip}} = J_1(\theta)\dot{\theta}_1 + J_2(\theta)\dot{\theta}_2. \quad (5.2)$$

$J_1(\theta)$ + $J_2(\theta)$: basis for the linear velocities of the end effector (with coefficients equal to joint velocities)
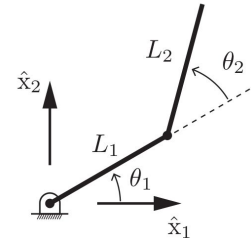
end effector velocity when joint 1 rotates at unit speed (joint 2 is still)

Ch.5 Velocity kinematics and statics in Lynch, K. M., & Park, F. C. (2017). Modern robotics. Cambridge University Press.
(see also https://youtu.be/6tj8QLF69Ok)

Straightforward generalization to 3D
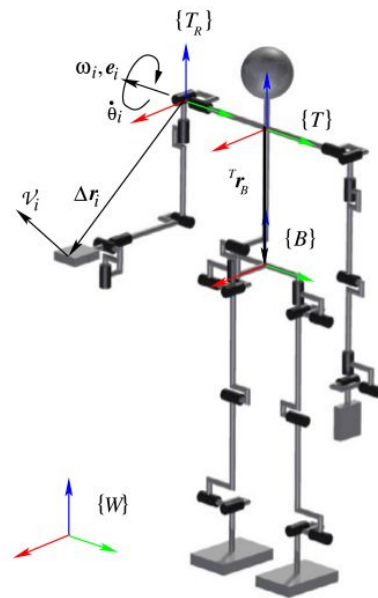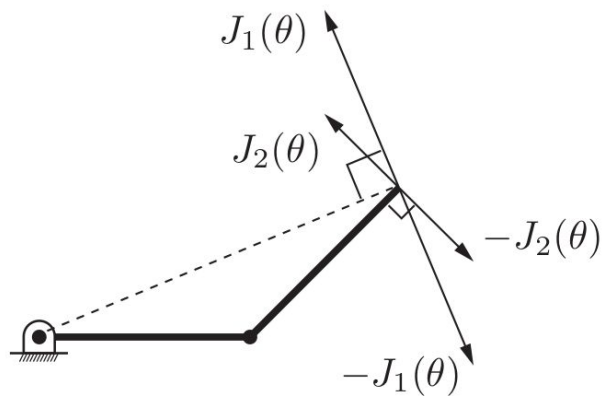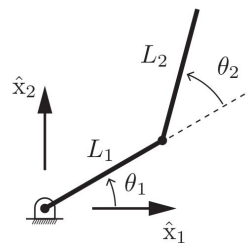
FIGURE 2.3 End-link spatial velocity $\mathcal{J}_i = [(e_i \times r_i)^T \quad e_i^T]^T$ is obtained with joint rate $\dot{\theta}_i = 1$ rad/s. Vector e_i signifies the joint axis of rotation. The position r_i of the characteristic point on the end link is determined w.r.t. reference frame $\{T_R\}$, obtained by translating the common root frame for the arms, $\{T\}$, to a suitably chosen point on the joint axis, e.g. according to the Denavit and Hartenberg notation [26].

Section 2.4 in Nenchev, D. N., Konno, A., & Tsujita, T. (2018). Humanoid robots: Modeling and control. Butterworth-Heinemann.

iCub matlab demo – illustrate Jacobian

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024
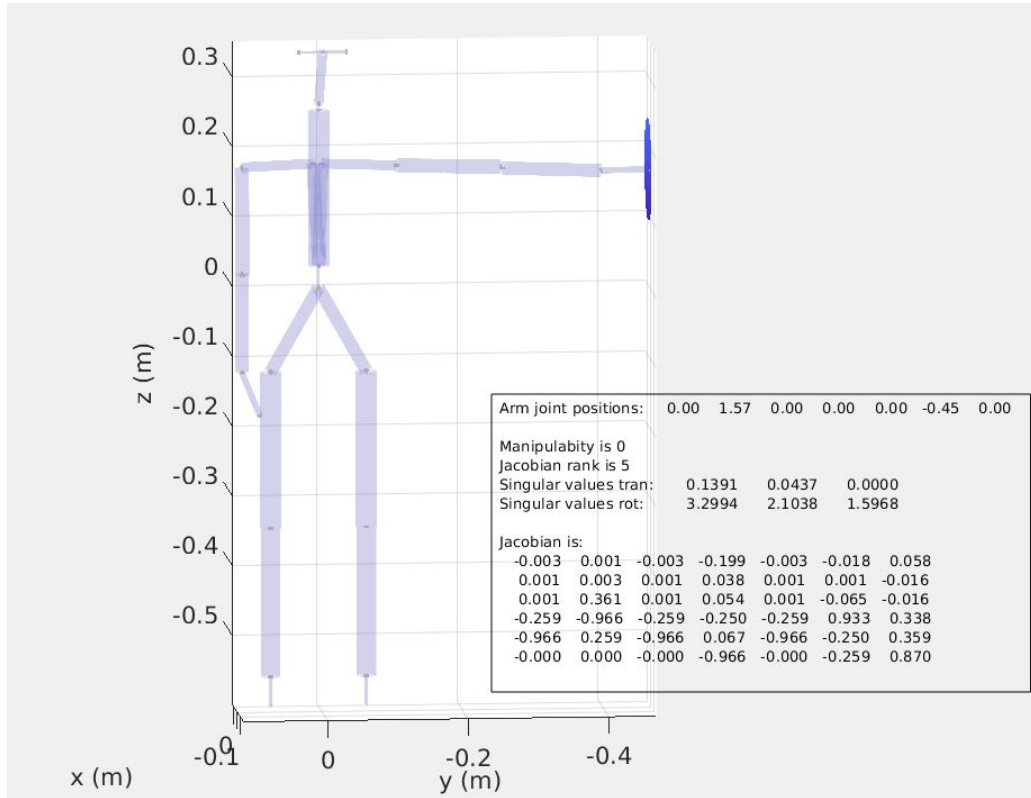
Planar arm with 2 rotational joints – singularities (position only)

$$x_1 = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2)$$
$$x_2 = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2).$$

Differentiating both sides with respect to time yields

$$\dot{x}_1 = -L_1\dot{\theta}_1 \sin\theta_1 - L_2(\dot{\theta}_1 + \dot{\theta}_2)\sin(\theta_1 + \theta_2)$$
$$\dot{x}_2 = L_1\dot{\theta}_1 \cos\theta_1 + L_2(\dot{\theta}_1 + \dot{\theta}_2)\cos(\theta_1 + \theta_2),$$

which can be rearranged into an equation of the form $\dot{x} = J(\theta)\dot{\theta}$:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -L_1 \sin\theta_1 - L_2\sin(\theta_1+\theta_2) & -L_2\sin(\theta_1+\theta_2) \\ L_1 \cos\theta_1 + L_2\cos(\theta_1+\theta_2) & L_2\cos(\theta_1+\theta_2) \end{bmatrix}\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}. \quad (5.1)$$

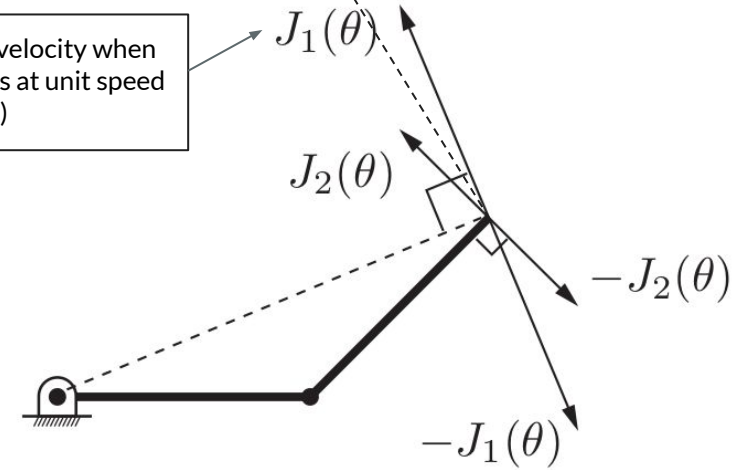Writing the two columns of $J(\theta)$ as $J_1(\theta)$ and $J_2(\theta)$, and the tip velocity $\dot{x}$ as v_{tip}, Equation (5.1) becomes

$$v_{\text{tip}} = J_1(\theta)\dot{\theta}_1 + J_2(\theta)\dot{\theta}_2. \quad (5.2)$$

- $\theta_2 = 0°$ (or $180°$)
- $J_1(\theta)$ and $J_2(\theta)$ aligned
- impossible to generate any end effector velocity except along this line
- dimension of the column space of the Jacobian drops from its maximum value
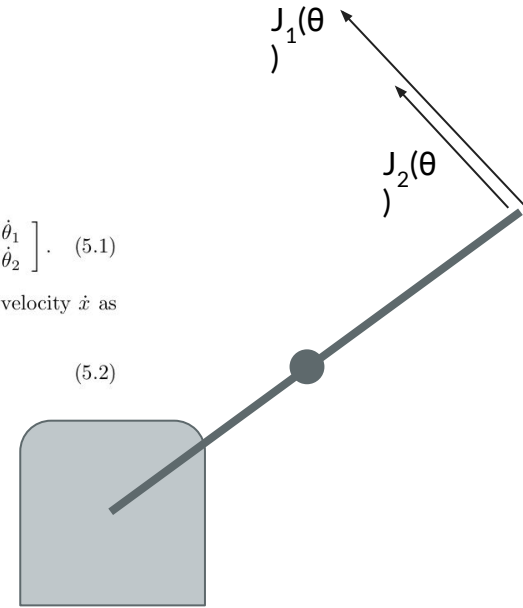
Ch.5 Velocity kinematics and statics in Lynch, K. M., & Park, F. C. (2017). Modern robotics. Cambridge University Press. (see also https://youtu.be/6tj8QLF69Ok)

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

Differential kinematics at singular configurations

- There are certain limb configurations where the end link loses mobility, i.e. the ability of instantaneous motion in one or more directions.
- At singularities, bounded end-effector velocities may correspond to unbounded joint velocities.
- Since the arms are fully extended, the hands cannot move in the downward direction w.r.t. the {T} frame.
- Since the legs are stretched, the {B} frame cannot be moved in the upward direction.
- Elbow / knee singularities - unavoidable.
  - There are no alternative nonsingular configurations that would place the end links at the same locations, at the workspace boundaries of each limb.
  - Inherent to both redundant and nonredundant limbs.
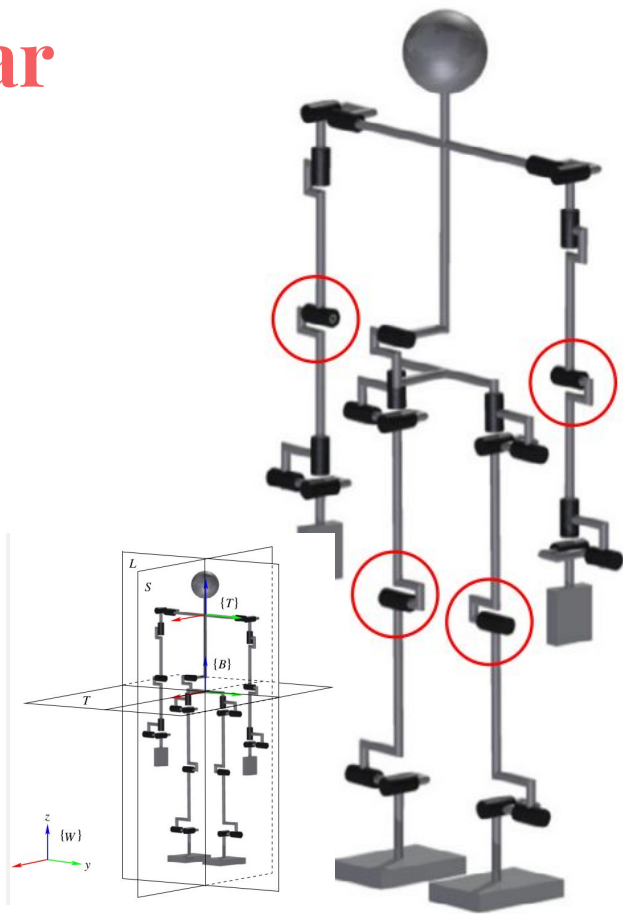
Section 2.5 in Nenchev, D. N., Konno, A., & Tsujita, T. (2018). Humanoid robots: Modeling and control. Butterworth-Heinemann.

Differential kinematics at singular configurations

- Singularities can be also useful though! When?
- Resisting external forces with minimal load in the joints.

Section 2.5 in Nenchev, D. N., Konno, A., & Tsujita, T. (2018). Humanoid robots: Modeling and control. Butterworth-Heinemann.
Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024
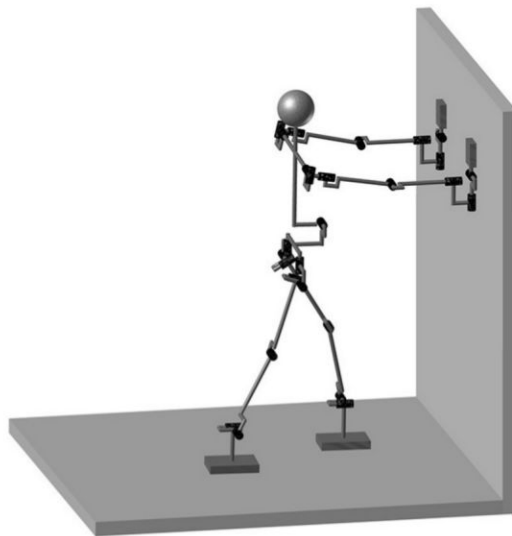
Velocity and force manipulability are orthogonal!

Force ellipsoid

Velocity ellipsoid
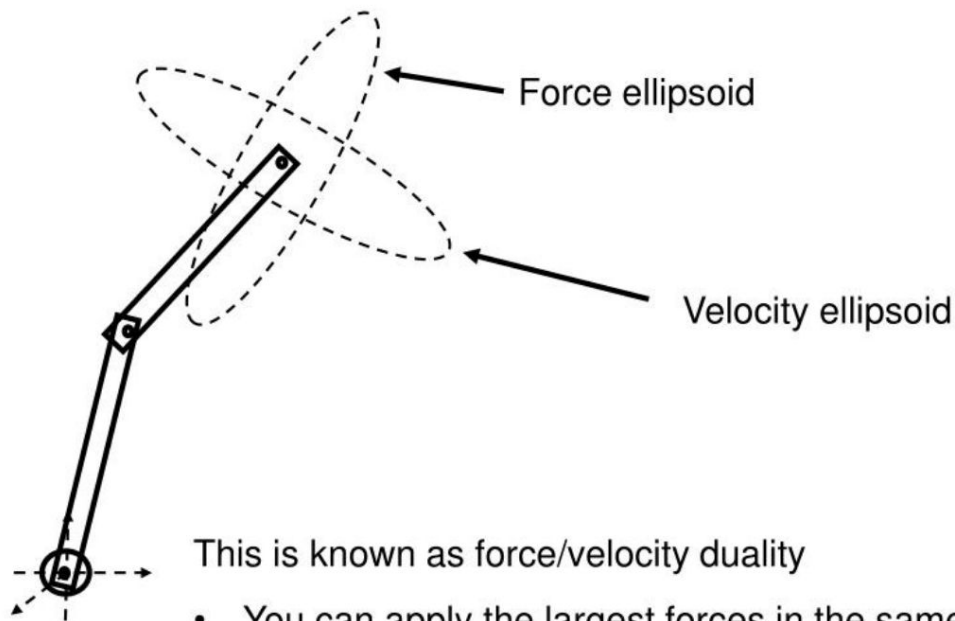
This is known as force/velocity duality

- You can apply the largest forces in the same directions that your max velocity is smallest

- Your max velocity is greatest in the directions where you can only apply the smallest forces

Slide 40 https://www.slideserve.com/antonia/inverting-the-jacobian-and-manipulability.
See slides 38-39 for details how this is derived.

Manipulability

2.6 MANIPULABILITY ELLIPSOID

From the differential kinematics relation (2.11), it is apparent that the ability of the end link to move instantaneously along a given spatial (rigid-body motion) direction will depend on the current limb configuration. In particular, as already clarified, at a singular configuration the ability to move along the singular directions becomes zero, and hence, mobility is lost in these directions. To facilitate instantaneous motion analysis and control, it is quite desirable to quantify the mobility in a given direction, at any given configuration. This can be done via *Singular-Value Decomposition* (SVD) [42,147,90] of the Jacobian matrix. For the general case of an n-DoF kinematically redundant limb, we have

$$J(\theta) = U(\theta)\Sigma(\theta)V(\theta)^T, \qquad (2.26)$$

where $U(\theta) \in \Re^{6\times6}$ and $V(\theta) \in \Re^{n\times n}$ are orthonormal matrices and

$$\Sigma(\theta) = \begin{bmatrix} \text{diag}\{\sigma_1(\theta), \ \sigma_2(\theta), ..., \sigma_6(\theta)\} & | & 0 \end{bmatrix} \in \Re^{6\times n}. \qquad (2.27)$$

Here $\sigma_1 \geq \sigma_2 \geq, ..., \geq \sigma_6 \geq 0$ are the singular values of the Jacobian. The columns of matrix $U(\theta)$, u_i, $i = 1, ..., 6$, provide a basis for the instantaneous motion space of the end link at the given limb configuration. At a nonsingular limb configuration, all singular values are positive. At a singular configuration of corank $6 - \rho$ ($\rho = \text{rank} J$), $6 - \rho$ of the singular values become zeros, i.e. $\sigma_1 \geq \sigma_2 \geq, ..., \geq \sigma_\rho > 0$, $\sigma_{\rho+1} = ... = \sigma_6 = 0$. The singular value σ_i quantifies the instantaneous mobility of the end link along the instantaneous motion direction u_i. Assuming that the magnitude of the joint rate vector is limited at each limb configuration as $\|\dot{\theta}\| \leq 1$, the highest mobility is along the direction corresponding to the maximum singular value. At a singular configuration of corank 1, $\sigma_{min} = 0$ and the respective direction u_{min} becomes a singular direction. Vectors $\sigma_i u_i$ constitute the principal axis of an ellipsoid—a useful graphic tool for visualizing the instantaneous mobility along each possible motion direction.

Section 2.6 in Nenchev, D. N., Konno, A., & Tsujita, T. (2018). Humanoid robots: Modeling and control. Butterworth-Heinemann.

Manipulability ellipsoid

not isotropic ✖

- Unit circle of joint velocities maps through the Jacobian to an ellipse in the space of tip velocities - the *manipulability ellipsoid.*
- As the manipulator configuration approaches a singularity, the ellipse collapses to a line segment, since the ability of the tip to move in one direction is lost.

$J(\theta)$
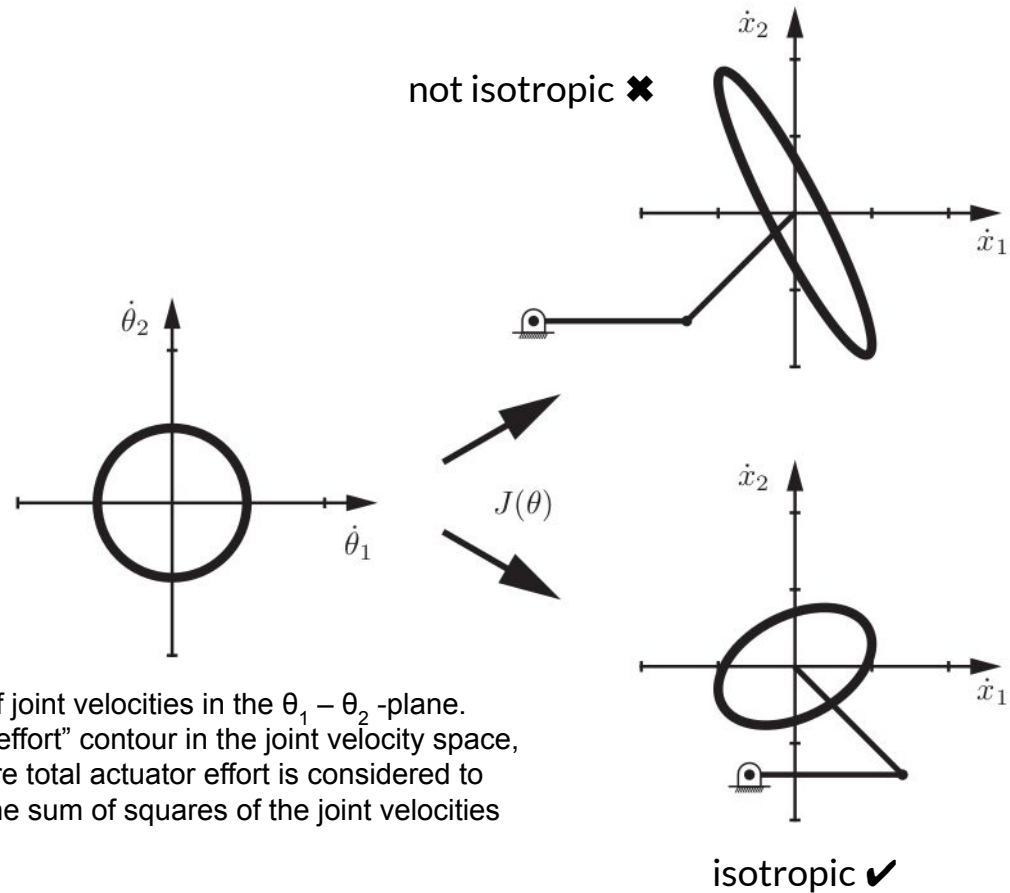
Unit circle of joint velocities in the $\theta_1 - \theta_2$ -plane.
- "iso-effort" contour in the joint velocity space, where total actuator effort is considered to be the sum of squares of the joint velocities

isotropic ✔

Ch.5 Velocity kinematics and statics in Lynch, K. M., & Park, F. C. (2017). Modern robotics. Cambridge University Press.
(see also https://youtu.be/6tj8QLF69Ok)

Manipulability ellipsoid
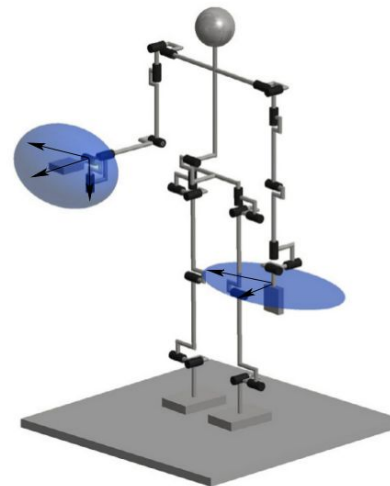
FIGURE 2.8 Manipulability ellipsoid for translational motion. The right arm is in a nonsingular configuration and the respective ellipsoid is 3D, with principal axes $\sigma_1 u_1$, $\sigma_2 u_2$, and $\sigma_3 u_3$. The left arm is at a singular configuration: the downward translational mobility has been lost, and therefore, the manipulability ellipsoid is only 2D. The principal axes are $\sigma_1 u_1$ and $\sigma_2 u_2$.

The dimension of the ellipsoid is determined by the rank of the Jacobian. Fig. 2.8 shows a robot configuration wherein the right arm is at a nonsingular configuration, whereas the left one is at the elbow singularity. The two ellipsoids at the end links visualize the instantaneous translational motion abilities. The ellipsoid for the right arm is 3D (full translational mobility), while that for the left arm is flat (an ellipse). The ellipse lies in a plane parallel to the floor since translational mobility in the vertical direction is nil at the singularity. The ellipsoid-based instantaneous mobility analysis has been introduced in [166]; the ellipsoid is referred to as the *manipulability ellipsoid*.

Section 2.6 in Nenchev, D. N., Konno, A., & Tsujita, T. (2018). Humanoid robots: Modeling and control. Butterworth-Heinemann.

iCub matlab demo – illustrate manipulability
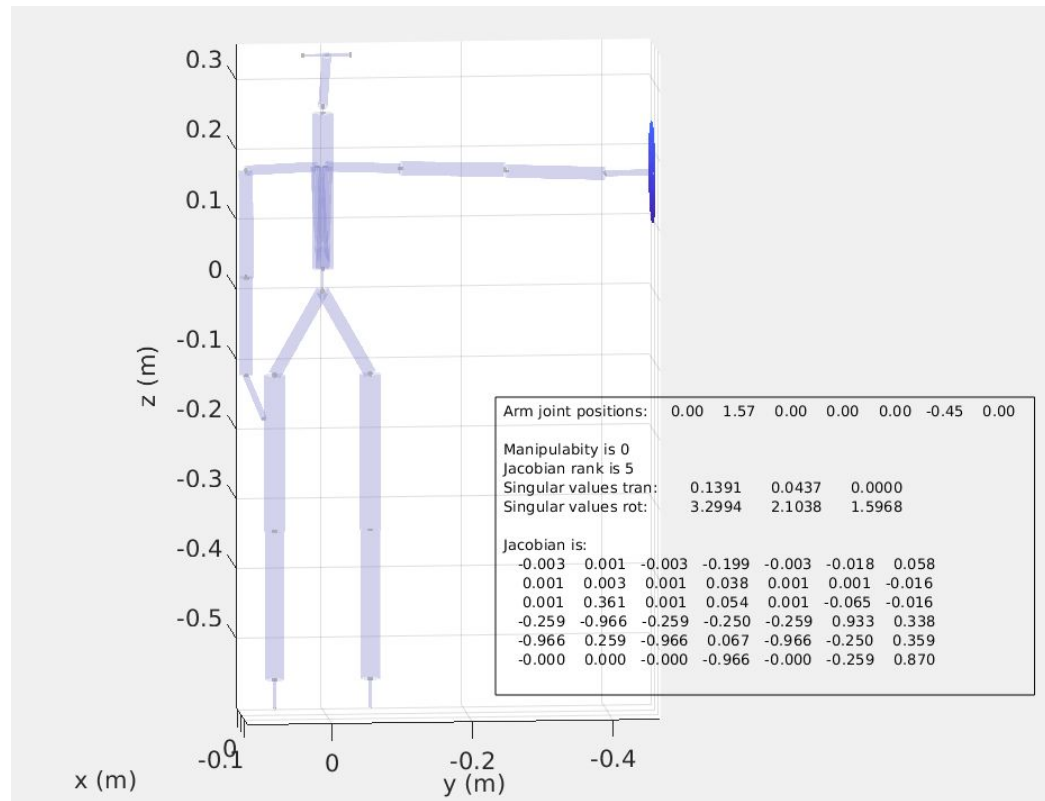
See also:

- Vahrenkamp, N., Asfour, T., Metta, G., Sandini, G., & Dillmann, R. (2012, November). Manipulability analysis. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)* (pp. 568-573). IEEE.
- https://github.com/robotology/community/discussions/559#:~:text=Ipopt%20doesn%27t%20deal,Jacobian%20per%20se

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

Resolved Rate Motion Control (RRMC)

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$$

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\dot{\mathbf{x}}$$

- generate joint velocities that create a desired Cartesian end effector **motion** (cf. inverse kinematics in position)
- *reactive control*

Resolved Motion Rate Control of Manipulators and Human Prostheses

DANIEL E. WHITNEY, MEMBER, IEEE

Abstract—The kinematics of remote manipulators and human prostheses is analyzed for the purpose of deriving resolved motion rate control. That is, the operator is enabled to call for the desired hand motion directly along axes relevant to the task environment. The approach suggests solutions to problems of coordination, motion under task constraints, and appreciation of forces encountered by the controlled hand.

INTRODUCTION

RATE control is currently one of the two most common ways of controlling a remote manipulator. The operator seeks to specify the direction and speed with which the manipulator is to move using a joystick or a set of switches. It is used in most industrial applications where large force amplification is needed and where environmental constraints or distance between controller and manipulator dictate a nonmechanical control link. The other common control method is master–slave control in which the operator guides a (usually full-scale) model of the manipulator so that the remote slave will follow a specified path and come to rest at a specified point. It is usually used with purely mechanical linkages to operate "hot lab" manipulators for precise tasks and has the favorable attributes of spatial correspondence and force feedback [1].

Rate control seems the predominant mode in most prototype artificial arms to date. An exception is the "Boston arm."[1] This is a powered elbow prosthesis, which amplifies EMG signals to generate force output (with force and velocity feedback) in one degree of freedom.

When the manipulator is powered by electric motors, hydraulic actuators, or the like, rather than by the operator's own muscles, the problem of coordinating the actuators arises. The problem is solved automatically in master–slave control if master and slave are geometrically similar, since coordinated motor drive signals may

Whitney, D. E. (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on man-machine systems*, *10*(2), 47-53.

Using J⁻¹ for Cartesian control

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$$

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\dot{\mathbf{x}}$$
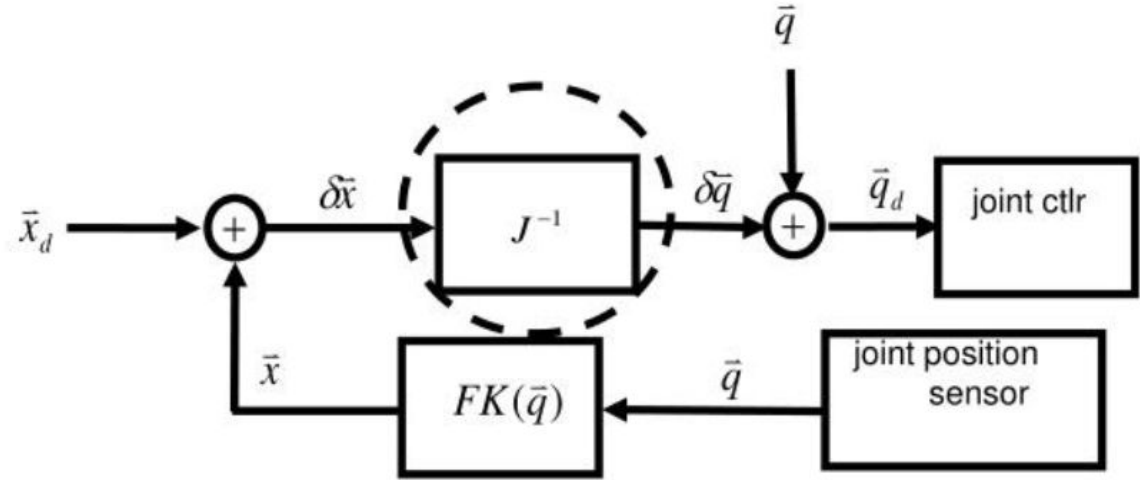
Is this always possible?

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = J^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$$

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

Underactuation, full actuation, and over-actuation
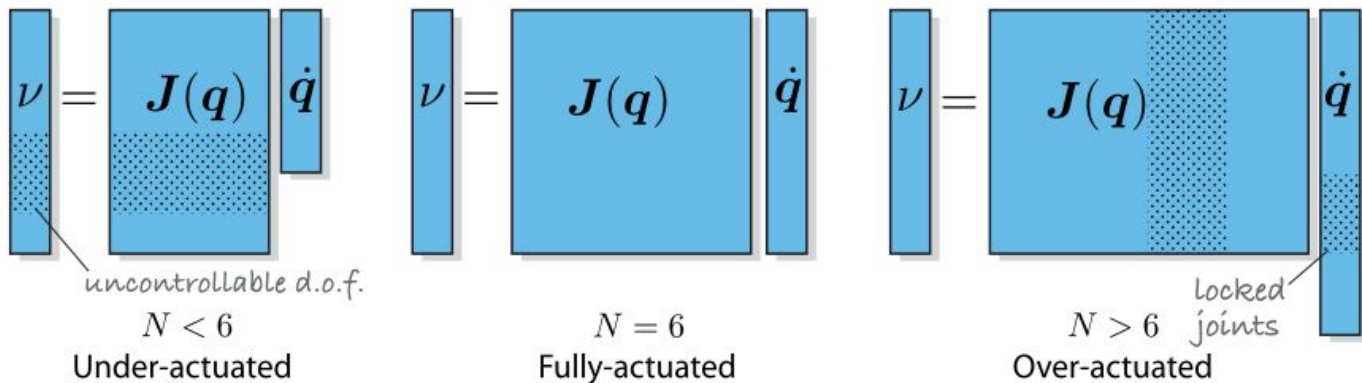
Fig. 8.6 in Corke, P. I. (2013). Robotics, vision and control: fundamental algorithms in MATLAB Berlin: Springer.

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}(\boldsymbol{\theta})^{-1}\mathcal{V}$$

What if **J(q)** is not square (and full rank) hence the inverse does not exist?

Section 8.2 in Corke, P. I. (2013). Robotics, vision and control: fundamental algorithms in MATLAB Berlin: Springer.

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

Underactuation

$$\nu = J(q) \dot{q}$$

uncontrollable d.o.f.

$N < 6$

Under-actuated

$N = 6$

Fully-actuated

$N > 6$ locked joints

Over-actuated

n=5, m = 6 (x,y,z, ϕ, θ, ψ)

$$\dot{\theta} = J(\theta)^{-1} \mathcal{V}$$
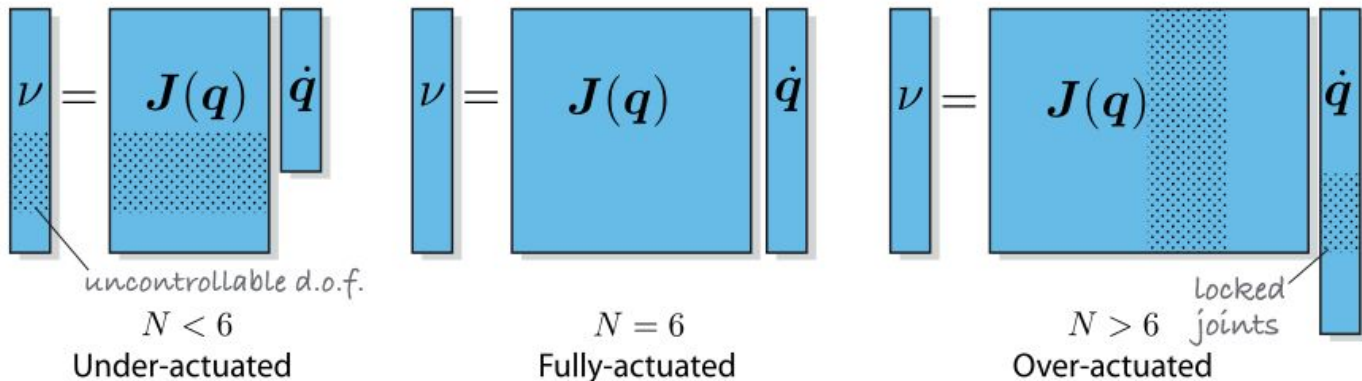
- What if **J(q)** is not square (and full rank) hence the inverse does not exist?
- Underactuated case:
    - Cannot be solved (unless we are lucky) because the system of equations is overdetermined.
    - "Quick hack": The system can be squared up by deleting some rows of v and J – accepting that some Cartesian degrees of freedom are not controllable given the low number of joints.

Overactuation

n=7 (or 10), m = 6 (x,y,z, ϕ, θ, ψ)

$$\dot{\theta} = J(\theta)^{-1}\mathcal{V}$$

- What if **J(q)** is not square (and full rank) hence the inverse does not exist?
- Overactuated case:
    - The system of equations is underdetermined. Multiple solutions exist and we can find a least squares solution (later).
    - "Quick hack": Alternatively we can square up the Jacobian to make it invertible by deleting some columns – effectively locking the corresponding axes.
    - What do we do with these extra axes?

Corke, P. I. (2013). Robotics, vision and control: fundamental algorithms in MATLAB Berlin: Springer.   38
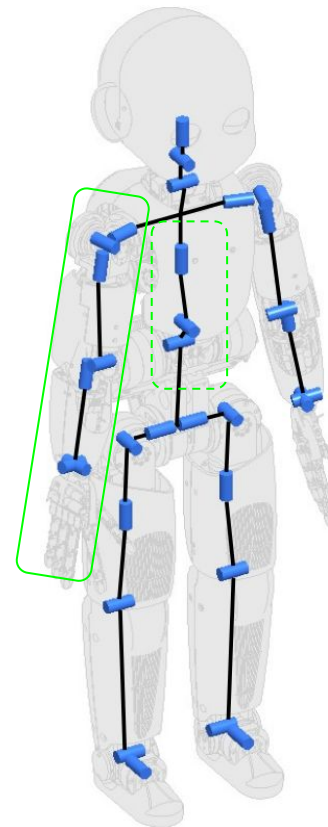
Self–motion

2.7.1 Self-Motion

In contrast to a nonredundant limb, a kinematically redundant limb can move even when its end link is immobilized ($\mathcal{V} = \mathbf{0}$). Such motion is shown in Fig. 2.9 for the arm; the hand remains fixed w.r.t. the arm root frame while the elbow rotates around the line connecting the shoulder and wrist joints. Such type of motion is known as *self-motion*, *internal motion*, or *null motion*.

Self-motion is generated by the joint velocity obtained from the following homogeneous differential relation:

$$J(\theta)\dot{\theta} = \mathbf{0}, \quad \dot{\theta} \neq \mathbf{0}. \tag{2.28}$$

Since $n > 6$, the Jacobian is nonsquare ($6 \times n$) and the above equation is characterized as an underdetermined linear system. Hence, there is an infinite set of solutions, each nontrivial
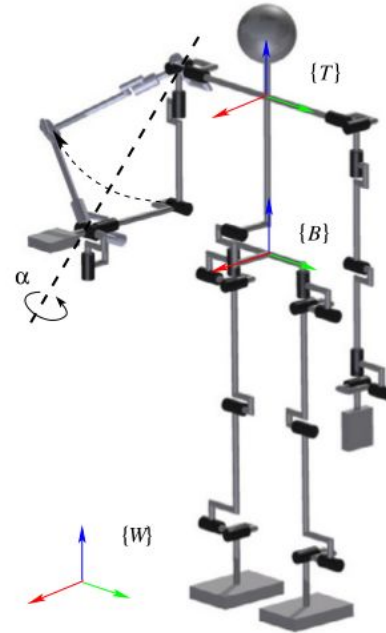
FIGURE 2.9 The self-motion of the arm is shown as a rotation of the arm plane, determined by the upper/lower arm links, around the line connecting the shoulder and wrist joints. The rotation angle α can be associated with parameter b_v in (2.35).

Section 2.7.1 in Nenchev, D. N., Konno, A., & Tsujita, T. (2018). Humanoid robots: Modeling and control. Butterworth-Heinemann.

Inverse differential kinematic relations

How about iCub?

- Given the joint angles and the end-link spatial velocity, find the motion rates in the joints.
- In order to find a solution in a straightforward manner, the following two conditions have to be satisfied:
    a. the Jacobian matrix at branch configuration θ should be of full rank;
    b. the number of joints of the branch should be equal to the DoF of the end link.
- These conditions imply that the inverse of the Jacobian matrix exists.
- When the conditions are satisfied, solving $\mathcal{V}_n = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$ the joint rates yields the following solution to the inverse kinematics problem:

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}(\boldsymbol{\theta})^{-1}\mathcal{V}$$

- A branch configuration yielding a full-rank Jacobian is called a *nonsingular configuration.*
- A branch with a number of joints that conforms to the second condition is called a kinematically nonredundant branch.
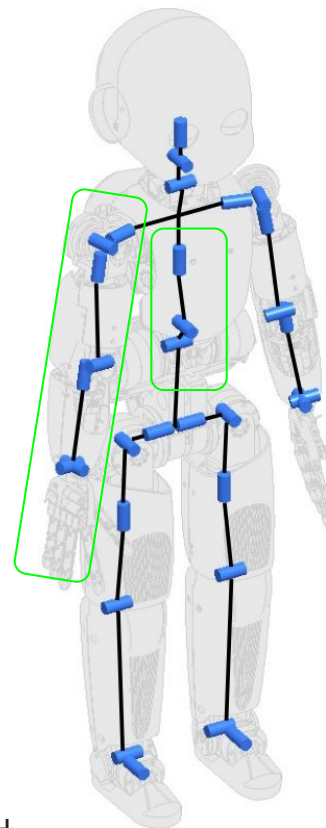
Section 2.4.3 in Nenchev, D. N., Konno, A., & Tsujita, T. (2018). Humanoid robots: Modeling and control. Butterworth-Heinemann.

Inverse differential kinematic relations – challenges

- Whenever any of the above two conditions cannot be met, the inverse problem needs to be handled with care.
- Special branch configurations where the Jacobian loses rank. Such configurations are called *singular*.
  - The branch can attain a singular configuration irrespective of the number of its joints.
- Further on, when the branch comprises more joints than the DoF of its end link (n > 6), then $\mathcal{V}_n = J(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$ is underdetermined. This implies the existence of an infinite set of inverse kinematics solutions for the joint rates. In this case, the branch is referred to as a *kinematically redundant branch*.

Generalized inverse

- We are looking for a matrix **J#** such that: $\dot{\theta} = J^{\#}\nu$
- Two cases:
  - Underactuated manipulator (~ overdetermined system of equations): Find $\dot{\theta}$

    such that $J\dot{\theta} - \nu$ is minimized.

  - Redundant manipulator (~ underdetermined system of equations): Find $\dot{\theta}$

    solving $\nu = J\dot{\theta}$ optimizing some additional property.

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

Forward, inverse, and differential kinematics – recap
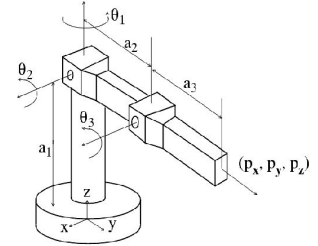
	Mapping type
$\mathbf{x} = \mathbf{f}(\mathbf{q})$	Many-to-one (for n>= m)
$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x})$	One-to-many (for n>1)
$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$	
$\dot{\mathbf{q}} = \mathbf{J}^{-1}\dot{\mathbf{x}}$	

$$p_x = \cos\theta_1 \, (a_3 \cos(\theta_2 + \theta_3) + a_2 \cos\theta_2)$$
$$p_y = \sin\theta_1 \, (a_3 \cos(\theta_2 + \theta_3) + a_2 \cos\theta_2)$$
$$p_z = a_3 \sin(\theta_2 + \theta_3) + a_2 \sin\theta_2 + a_1$$

https://www.slideserve.com/antonia/inverting-the-jacobian-and-manipulability

Forward, inverse, and differential kinematics – recap
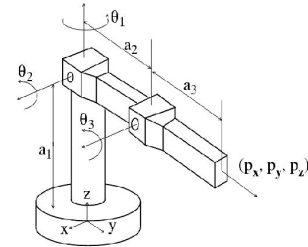
	Mapping type
$\mathbf{x} = \mathbf{f}(\mathbf{q})$	Many-to-one (for n>= m)
$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x})$	One-to-many (for n>1)
$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$	Many-to-one (for n>= m)
$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\dot{\mathbf{x}}$	One-to-many (for n>1)

$p_x = \cos\theta_1 \, (a_3 \cos(\theta_2 + \theta_3) + a_2 \cos\theta_2)$
$p_y = \sin\theta_1 \, (a_3 \cos(\theta_2 + \theta_3) + a_2 \cos\theta_2)$
$p_z = a_3 \sin(\theta_2 + \theta_3) + a_2 \sin\theta_2 + a_1$

https://www.slideserve.com/antonia/inverting-the-jacobian-and-manipulability
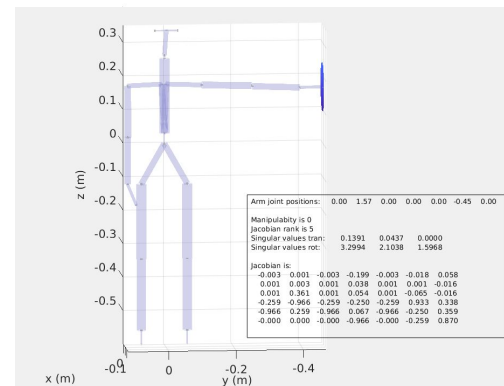
Using differential kinematics for IK

- They are numerical, iterative methods.
- Methods relying on some form of Jacobian inverse.
  - Jacobian transpose
  - Jacobian pseudoinverse
    - Used by Orocos Kinematics and Dynamics Library (KDL) - used in ROS.
  - Damped least squares
- Methods using optimization to find solutions to forward differential kinematics.
- More details in
  - Haviland, J., & Corke, P. (2023). Manipulator Differential Kinematics: Part I: Kinematics, Velocity, and Applications. IEEE Robotics & Automation Magazine. https://doi.org/10.1109/MRA.2023.3270228
  - Haviland, J., & Corke, P. (2023). Manipulator Differential Kinematics: Part 2: Acceleration and Advanced Applications. IEEE Robotics & Automation Magazine. https://doi.org/10.1109/MRA.2023.3270221
  - Buss, S. R. (2004). Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods. IEEE Journal of Robotics and Automation, 17(1-19), 16.
  - 2.7.2 - 2.7.5 in Nenchev et al. (2018)
  - Jacobian transpose - duality of kinematics and statics - 8.4 in Corke, P. I. (2013). Robotics, vision and control: fundamental algorithms in MATLAB Berlin: Springer.
  - https://github.com/vvv-school/vvv18/blob/master/material/kinematics/kinematics.pdf
  - https://www.slideserve.com/antonia/inverting-the-jacobian-and-manipulability

Differential kinematics for IK – Jacobian "inversion"

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

Jacobian transpose vs. pseudoinverse

- Which is more direct? Jacobian pseudoinverse or transpose?

$$\dot{q} = J^T \xi \qquad \text{or} \qquad \dot{q} = J^\# \xi$$

They do different things:

- Transpose: move toward a reference pose as quickly as possible
  - One dimensional goal (squared distance meteric)
- Pseudoinverse: move along a least squares reference twist trajectory
  - Six dimensional goal (or whatever the dimension of the relevant twist is)

Howie Choset: https://www.slideserve.com/antonia/inverting-the-jacobian-and-manipulability

Jacobian transpose vs. pseudoinverse

The pseudoinverse moves the end effector in a straight line path toward the goal pose using the least squared joint velocities.

- The goal is specified in terms of the reference twist

- Manipulator follows a straight line path in Cartesian space

x_d
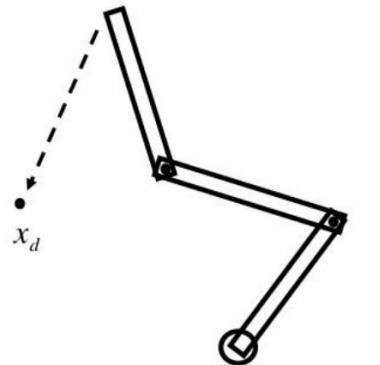
The transpose moves the end effector toward the goal position

- In general, not a straight line path *in Cartesian space*

- Instead, the transpose follows the gradient in *joint space*

x_d

Howie Choset: https://www.slideserve.com/antonia/inverting-the-jacobian-and-manipulability

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024
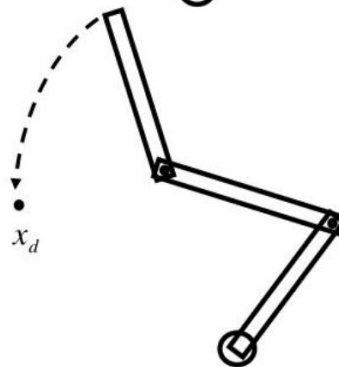
Damped Least Squares

To deal with kinematic singularities

$$\dot{\mathbf{q}}^* = \arg\min_{\dot{\mathbf{q}}} \left(\frac{1}{2} \left(\dot{\mathbf{x}} - \mathbf{J}\dot{\mathbf{q}} \right)^T \left(\dot{\mathbf{x}} - \mathbf{J}\dot{\mathbf{q}} \right) + \frac{1}{2} k^2 \dot{\mathbf{q}}^T \dot{\mathbf{q}} \right)$$

Levenberg-Marquardt method
for system of nonlinear equations

$$\mathbf{J}^* = \mathbf{J}^T \left(\mathbf{J}\mathbf{J}^T + k^2 \mathbf{I}_n \right)^{-1}$$

$$\dot{\mathbf{q}} = \mathbf{J}^* \dot{\mathbf{x}}$$

k establishes a synergy between:
- Jacobian (Pseudo-)inverse: $k = 0$
- Jacobian Transpose: $k \gg \max\{|\sigma_i|\}$

https://github.com/vvv-school/vvv18/blob/master/material/kinematics/kinematics.pdf, courtesy Ugo Pattacini

Differential kinematics for IK – Jacobian "inversion" overview

Iterative Methods

Jacobian Transpose

$$\dot{\mathbf{q}} = \mathbf{J}^T \mathbf{K} \mathbf{e}, \quad \mathbf{e} = \mathbf{x}_d - \mathbf{x}_e$$

Jacobian Pseudoinverse

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \mathbf{K} \mathbf{e} + \left(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J} \right) \dot{\mathbf{q}}_0, \quad \mathbf{J}^\dagger = \mathbf{J}^T \left(\mathbf{J} \mathbf{J}^T \right)^{-1}$$

Damped Least Squares

$$\dot{\mathbf{q}} = \mathbf{J}^* \mathbf{K} \mathbf{e}, \quad \mathbf{J}^* = \mathbf{J}^T \left(\mathbf{J} \mathbf{J}^T + k^2 \mathbf{I} \right)^{-1}$$

- More details in
  - Buss, S. R. (2004). Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods. IEEE Journal of Robotics and Automation, 17(1-19), 16.
  - 2.7.2 - 2.7.5 in Nenchev et al. (2018); Redundancy resolution 2.7.4-2.7.5
  - Jacobian transpose - duality of kinematics and statics - 8.4 in Corke, P. I. (2013). Robotics, vision and control: fundamental algorithms in MATLAB Berlin: Springer.
  - https://github.com/vvv-school/vvv18/blob/master/material/kinematics/kinematics.pdf
  - Howie Choset: https://www.slideserve.com/antonia/inverting-the-jacobian-and-manipulability

IK is an active research topic

- Orocos Kinematics and Dynamics Library (KDL) - used in ROS / MoveIt!
  - Uses joint-limit constrained pseudoinverse Jacobian solver

Despite its popularity, KDL's IK implementation[3] exhibits numerous false-negative failures on a variety of humanoid and mobile manipulation platforms. In particular, KDL's IK implementation has the following issues:

1) frequent convergence failures for robots with joint limits,
2) no actions taken when the search becomes "stuck" in local minima,
3) inadequate support for Cartesian pose tolerances,
4) no utilization of tolerances in the IK solver itself.

Beeson, P., & Ames, B. (2015, November). TRAC-IK: An open-source library for improved solving of generic inverse kinematics. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)* (pp. 928-935). IEEE.

The current state-of-the-art and defacto-usage of numerical inverse kinematics fall into variations of these aforementioned algorithms. The open-source *Orocos Kinematics and Dynamics Library* (KDL) [17] is arguably the most popular generic inverse kinematics solver [7], and features various solvers based on a Levenberg–Marquard optimization approach, as well as a Newton-Raphson pseudoinverse solver. The KDL library is used in the ROS libraries, along with another solver called Trac-IK that improves reliability by simultaneously running both an Newton-Raphson solver and a BFGS solver, and stopping when either of the two reaches a successful solution [7]. Mathworks' popular *MATLAB Robotics Toolbox* features two solvers—one based on the Levenberg–Marquard algorithm (implemented as in [12]), and another based on the BFGS algorithm (implemented as in [1]).

Lloyd, S., Irani, R. A., & Ahmadi, M. (2022). Fast and Robust Inverse Kinematics of Serial Robots Using Halley's Method. IEEE Transactions on Robotics, 38(5), 2768-2780.

TABLE 1. Numerical IK methods compared on 10,000 problems with a 6-DoF UR5 manipulator.

METHOD	SEARCHES ALLOWED	ITER. ALLOWED	MEAN ITER.	MEDIAN ITER.	INFEASIBLE COUNT	INFEASIBLE %	MEAN SEARCHES	MAX SEARCHES
NR	1	500	21.34	16	1,093	10.93%	1	1
GN	1	500	21.6	16	1,078	10.78%	1	1
NR pseudoinverse	1	500	21.24	16	1,100	11%	1	1
GN pseudoinverse	1	500	21.72	16	1,090	10.9%	1	1
LM (Wampler $\lambda = 1e-4$)	1	500	20.1	14	934	9.34%	1	1
LM (Wampler $\lambda = 1e-6$)	1	500	29.84	17	529	5.29%	1	1
LM (Chan $\lambda = 1.0$)	1	500	16.58	14	1,011	10.11%	1	1
LM (Chan $\lambda = 0.1$)	1	500	9.43	9	963	9.63%	1	1
LM (Sugihara $w_n = 1e-3$)	1	500	20.54	15	1,024	10.24%	1	1
LM (Sugihara $w_n = 1e-4$)	1	500	17.01	14	1,011	10.11%	1	1
NR	100	30	30.16	18	0	0%	1.47	25
GN	100	30	30.33	18	0	0%	1.48	23
NR pseudoinverse	100	30	30.27	18	0	0%	1.47	20
GN pseudoinverse	100	30	30.65	18	0	0%	1.49	20
LM (Wampler $\lambda = 1e-4$)	100	30	25.23	15	0	0%	1.35	17
LM (Wampler $\lambda = 1e-6$)	100	30	29.3	18	0	0%	1.45	24
LM (Chan $\lambda = 1.0$)	100	30	22.6	15	0	0%	1.25	18
LM (Chan $\lambda = 0.1$)	100	30	15.33	9	0	0%	1.2	18
LM (Sugihara $w_n = 1e-3$)	100	30	26.49	16	0	0%	1.35	18
LM (Sugihara $w_n = 1e-4$)	100	30	23.04	15	0	0%	1.26	18

ITER.: iterations.

Haviland, J., & Corke, P. (2023). Manipulator Differential Kinematics: Part I: Kinematics, Velocity, and Applications. *IEEE Robotics & Automation Magazine*. https://doi.org/10.1109/MRA.2023.3270228

Differential kinematics for IK – "forward formulation" – optimization

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

- Jacobian (pseudo-)inverse approaches are problematic near singular configurations.
- Instead of computing the inverse, we can use optimization to solve the forward differential kinematics formulation.
  - Quadratic programming

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\dot{\mathbf{x}}$$

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$$

NEO – quadratic programming
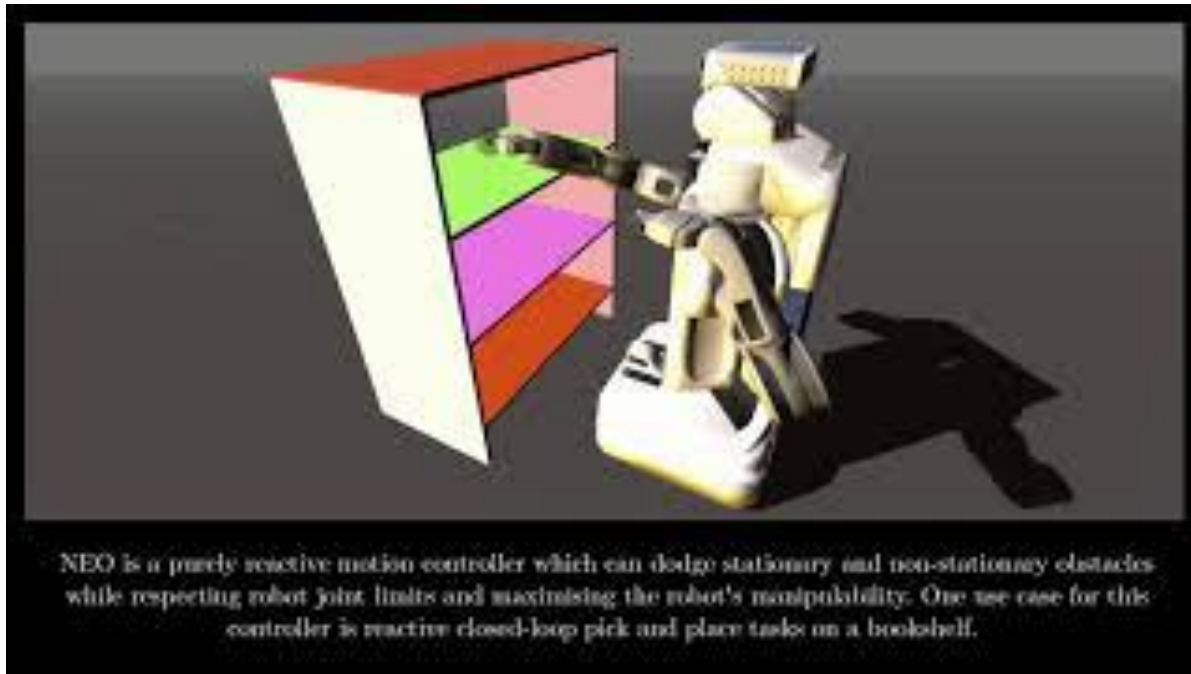
Haviland, J., & Corke, P. (2021). NEO: A novel expeditious optimisation algorithm for reactive motion control of manipulators. *IEEE Robotics and Automation Letters*, *6*(2), 1043-1050.

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

NEO – quadratic programming

Problem formulation [edit]

The quadratic programming problem with n variables and m constraints can be formulated as follows.[2] Given:

- a real-valued, n-dimensional vector $\mathbf{c}$,
- an $n{\times}n$-dimensional real symmetric matrix Q,
- an $m{\times}n$-dimensional real matrix A, and
- an m-dimensional real vector $\mathbf{b}$,

the objective of quadratic programming is to find an n-dimensional vector $\mathbf{x}$, that will

minimize $\quad \frac{1}{2}\mathbf{x}^{\mathrm{T}}Q\mathbf{x} + \mathbf{c}^{\mathrm{T}}\mathbf{x}$

subject to $\quad A\mathbf{x} \preceq \mathbf{b}$,

where $\mathbf{x}^{\mathrm{T}}$ denotes the vector transpose of $\mathbf{x}$, and the notation $A\mathbf{x} \preceq \mathbf{b}$ means that every entry of the vector $A\mathbf{x}$ is less than or equal to the corresponding entry of the vector $\mathbf{b}$ (component-wise inequality).

https://en.wikipedia.org/wiki/Quadratic_programming

A. Incorporating the Differential Kinematics Into a QP

The first-order differential kinematics for a serial-link manipulator are described by

$$\boldsymbol{\nu}(t) = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}(t) \qquad (2)$$

where $\boldsymbol{\nu}(t) = \begin{pmatrix} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{pmatrix}^{\top} \in \mathbb{R}^6$ is the end-effector spatial velocity and $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{6 \times n}$ is the manipulator Jacobian which relates $\boldsymbol{\nu}$ to the joint velocities $\dot{\boldsymbol{q}}(t)$, and n is the number of joints in the robot.

We can incorporate (2) into a quadratic program as

$$\min_{\dot{\boldsymbol{q}}} \quad f_o(\dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^{\top} \boldsymbol{I} \dot{\boldsymbol{q}},$$

$$\text{subject to} \quad \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{\nu},$$

$$\dot{\boldsymbol{q}}^- \leq \dot{\boldsymbol{q}} \leq \dot{\boldsymbol{q}}^+ \qquad (3)$$

where $\boldsymbol{I}$ is an $n \times n$ identity matrix, $\dot{\boldsymbol{q}}^-$ and $\dot{\boldsymbol{q}}^+$ represent the upper and lower joint-velocity limits, and no inequality constraints need to be defined. If the robot has more degrees-of-freedom than necessary to reach its entire task space, the QP in (3) will achieve the desired end-effector velocity with the minimum joint-velocity norm. However, there are other things we can optimise for, such as manipulability.

Haviland, J., & Corke, P. (2021). NEO: A novel expeditious optimisation algorithm for reactive motion control of manipulators. *IEEE Robotics and Automation Letters*, *6*(2), 1043-1050.

HARMONIOUS
(iCub playing "Bubbles")

Rozlivek, J., Roncone, A., Pattacini, U., & Hoffmann, M. (2023). HARMONIOUS - Human-like reactive motion control and multimodal perception for humanoid robots. https://arxiv.org/abs/2312.02711
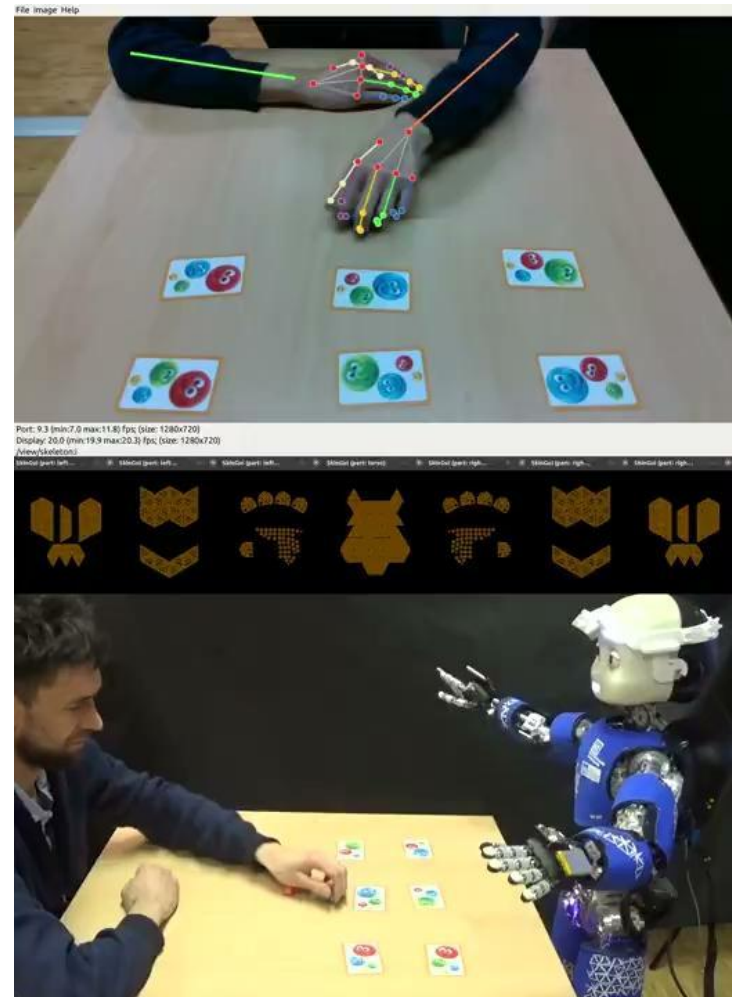
HARMONIOUS

QP solver

$$\min_{\dot{\mathbf{q}},\lambda} \frac{\mu}{2}\dot{\mathbf{q}}^T \mathbf{W_q}\dot{\mathbf{q}} + \frac{1}{2}\lambda^T \mathbf{W}_\lambda \lambda + \frac{c_n}{2}\left(\dot{\mathbf{q}} - \dot{\mathbf{q}}_n\right)^T \mathbf{W_q}\left(\dot{\mathbf{q}} - \dot{\mathbf{q}}_n\right)$$

$$\text{s.t.} \begin{cases} \nu - \lambda = \mathbf{J(q)}\dot{\mathbf{q}} \\ \dot{\mathbf{q}}_L < \dot{\mathbf{q}} < \dot{\mathbf{q}}_U \\ \mathbf{q}_L < \mathbf{q} < \mathbf{q}_U \\ -\varepsilon < \lambda < \varepsilon \\ \mathbf{A_o} \leq \mathbf{b_o} \\ \mathbf{b_{L,t}} \leq \mathbf{A_t} \leq \mathbf{b_{U,t}} \end{cases},$$

For our purpose, we can relax the trajectory constraint by adding a slack variable λ to help the controller with finding a solution when other (e.g., safety) constraints appear:

$$\nu - \lambda = \mathbf{J(q)}\dot{\mathbf{q}} \tag{3}$$

Then, we formulate the problem as a strictly convex QP problem with joint velocities $\dot{\mathbf{q}}$ and slack variables λ as optimization variables. In our cost function, we minimize the sum of the squared Euclidean norm of the joint velocities, the distance from the natural joint positions, and the squared Euclidean norm of the vector of slack variables. It is formulated as:

$$\min_{\dot{\mathbf{q}},\lambda} \frac{\mu}{2}\dot{\mathbf{q}}^T \mathbf{W_q}\dot{\mathbf{q}} + \frac{1}{2}\lambda^T \mathbf{W}_\lambda \lambda + \frac{c_h}{2}\left(\dot{\mathbf{q}} - \dot{\mathbf{q}}_n\right)^T \mathbf{W_q}\left(\dot{\mathbf{q}} - \dot{\mathbf{q}}_h\right) \tag{4}$$

where diagonal matrices $\mathbf{W_q}$ and $\mathbf{W}_\lambda$ contain the weights of the individual joints and the relaxations for the trajectory constrain, $\dot{\mathbf{q}}_h$ are the velocities needed to reach natural (home) robot configuration, c_h is a weight of the motivation to keep robot close to his natural configuration, and μ is a damping

Rozlivek, J., Roncone, A., Pattacini, U., & Hoffmann, M. (2023). HARMONIOUS - Human-like reactive motion control and multimodal perception for humanoid robots. https://arxiv.org/abs/2312.02711

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

HARMONIOUS

QP solver

$$\min_{\dot{\mathbf{q}},\lambda} \ \frac{\mu}{2}\dot{\mathbf{q}}^T\mathbf{W_q}\dot{\mathbf{q}} + \frac{1}{2}\lambda^T\mathbf{W}_\lambda\lambda + \frac{c_n}{2}\left(\dot{\mathbf{q}} - \dot{\mathbf{q}}_n\right)^T\mathbf{W_q}\left(\dot{\mathbf{q}} - \dot{\mathbf{q}}_n\right)$$

$$\text{s.t.} \quad \begin{cases} \boldsymbol{\nu} - \lambda = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \\ \dot{\mathbf{q}}_L < \dot{\mathbf{q}} < \dot{\mathbf{q}}_U \\ \mathbf{q}_L < \mathbf{q} < \mathbf{q}_U \\ -\varepsilon < \lambda < \varepsilon \\ \mathbf{A_o} \leq \mathbf{b_o} \\ \mathbf{b_{L,t}} \leq \mathbf{A_t} \leq \mathbf{b_{U,t}} \end{cases},$$

robot close to his natural configuration, and μ is a damping factor calculated from the manipulability index $\omega = \sqrt{\mathbf{J}\mathbf{J}^T}$ [49] as:

$$\mu = \begin{cases} (1 - \frac{\omega}{\omega_0})^2 + 0.01 & \omega < \omega_0, \quad \omega_0 \text{ is a threshold}, \\ 0.01 & \text{otherwise}. \end{cases} \quad (5)$$

This damping factor increases when the manipulability index decreases to zero, implying that the robot is close to a singularity position (the manipulability index is zero in that position). The higher damping factor makes the minimization of joint velocities more important, leading to lower joint velocities and singularity avoidance [27].

Rozlivek, J., Roncone, A., Pattacini, U., & Hoffmann, M. (2023). HARMONIOUS - Human-like reactive motion control and multimodal perception for humanoid robots. https://arxiv.org/abs/2312.02711

HARMONIOUS

QP solver

$$\min_{\dot{\mathbf{q}},\lambda} \frac{\mu}{2}\dot{\mathbf{q}}^T\mathbf{W_q}\dot{\mathbf{q}} + \frac{1}{2}\lambda^T\mathbf{W_\lambda}\lambda + \frac{c_n}{2}\left(\dot{\mathbf{q}} - \dot{\mathbf{q}}_n\right)^T\mathbf{W_q}\left(\dot{\mathbf{q}} - \dot{\mathbf{q}}_n\right)$$

$$\text{s.t.} \begin{cases} \boldsymbol{\nu} - \lambda = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \\ \dot{\mathbf{q}}_L < \dot{\mathbf{q}} < \dot{\mathbf{q}}_U \\ \mathbf{q}_L < \mathbf{q} < \mathbf{q}_U \\ -\varepsilon < \lambda < \varepsilon \\ \mathbf{A_o} \le \mathbf{b_o} \\ \mathbf{b_{L,t}} \le \mathbf{A_t} \le \mathbf{b_{U,t}} \end{cases},$$

Apart from the mentioned relaxed trajectory constrained, the minimization is done with respect to several other constraints, such as joint velocity and position bounds (see Sec. II-K), slack variable bounds, collision avoidance (see Sec. II-F), and other task or kinematic constraints (see Sec. II-J). The constraints can be written as follows:

where $\dot{\mathbf{q}}_{L,U}, \mathbf{q}_{L,U}, \varepsilon$ are the lower and upper bounds for the joint velocities, the joint positions, and the slack variables, respectively. $\mathbf{A_o}$ and $\mathbf{b_o}$ define obstacles avoidance constraints; $\mathbf{A_t}, \mathbf{b}_{L,t}$, and $\mathbf{b}_{U,t}$ define other task constraints.

Rozlivek, J., Roncone, A., Pattacini, U., & Hoffmann, M. (2023). HARMONIOUS - Human-like reactive motion control and multimodal perception for humanoid robots. https://arxiv.org/abs/2312.02711
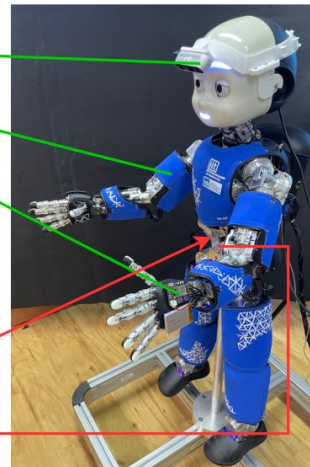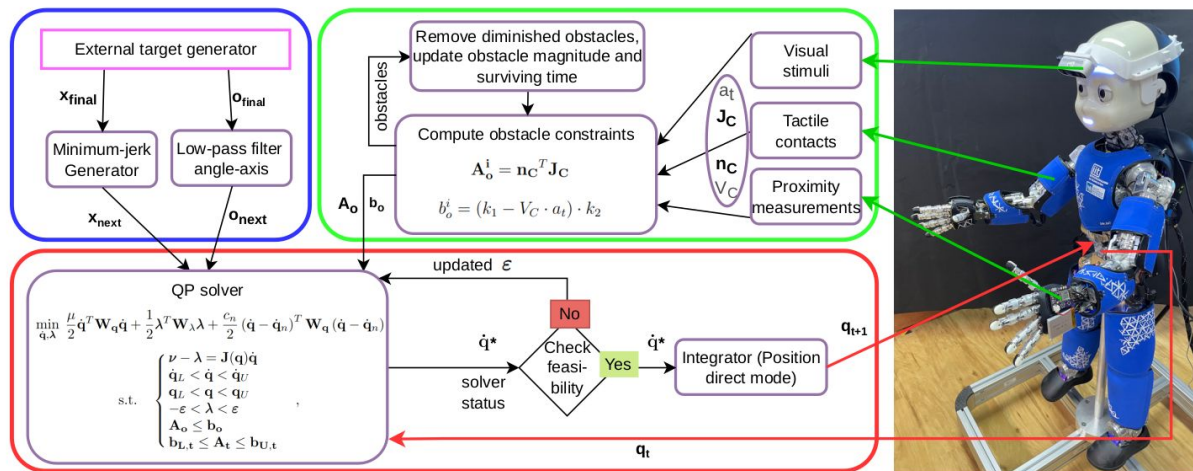
HARMONIOUS – COMPLETE CONTROLLER

Fig. 2: *HARMONIOUS* – overview. The blue box represents the local trajectory sampling. The trajectory towards the target pose is sampled using a minimum-jerk generator (position) and a low-pass filter (orientation). The target pose is received from an external target generator. The green box represents the obstacle processing. It takes inputs from proximity sensors, skin parts (tactile stimuli), and RGB-D camera (visual stimuli) and computes linear inequality constraints to dodge obstacles or react to the collision that occurred. The obstacle exists virtually for a specific survival time with decreasing threat level, giving humans time to react. The red box represents the controller with a QP solver that solves the differential kinematics problem. It takes the next desired pose and current joint positions as input. If the problem is feasible, the computed joint velocities are integrated into new joint positions and sent to the robot. Otherwise, the position constraint is relaxed, and the QP problem is solved again.

Rozlivek, J., Roncone, A., Pattacini, U., & Hoffmann, M. (2023). HARMONIOUS - Human-like reactive motion control and multimodal perception for humanoid robots. https://arxiv.org/abs/2312.02711

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

HARMONIOUS – Obstacle processing
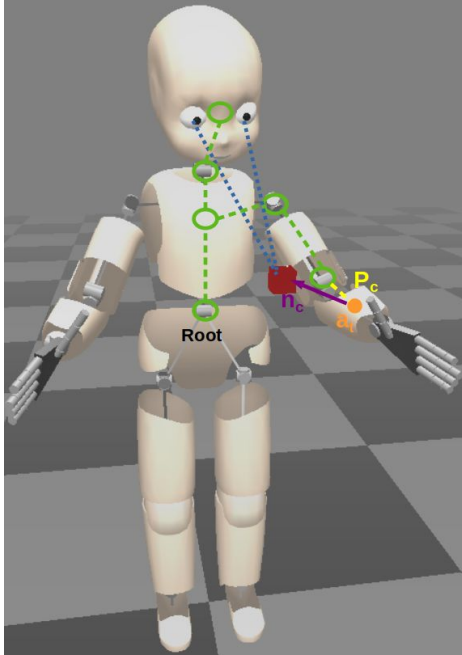
Visual obstacles
human keypoints mapped to skin using peripersonal space projection
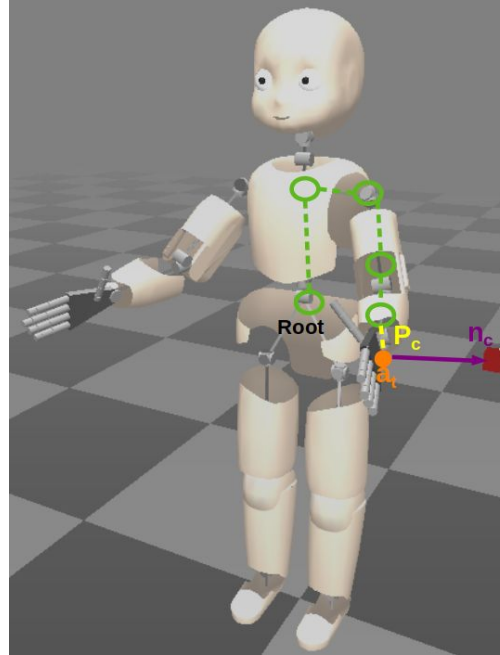
Proximity obstacles
proximity measurements projected to the sensor position

Tactile obstacles
contacts detected by skin sensors and clustered to super contacts

Inverse kinematics for humanoids

(a bit speculative)

There's no free lunch…

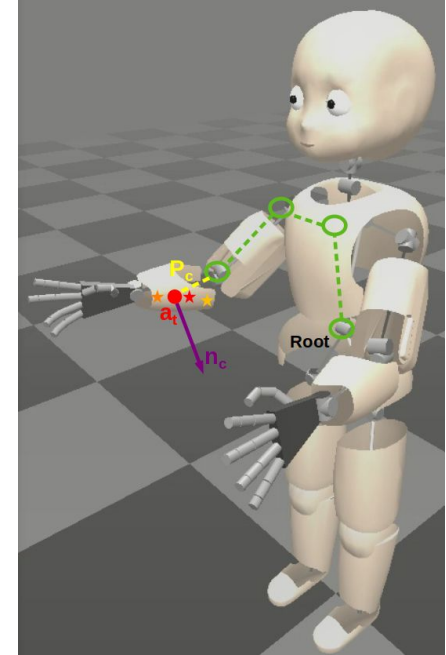| Method	General solution (not robot specific)	Takes care of motion control (trajectories)	Possibility of including additional constraints	Problems with singularities	Problems with local minima	Runtime	Examples
Analytical (closed-form)	😞	😞	😑	😑	😌	😄	
Iterative using some form of Jacobian inverse	😌	😌	😞	😞 (DLS compensates)	😑	😌	KDL / Orocos (ROS)
Optimization using forward kinematics	😌	😌	😑	😑	😑	😌	Velocity space - Neo (Haviland & Corke); position space - iCub IK solver

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

Work	Control problem	ROA	Sensory modalities to perceive obstacles	DoF shown	Singularities handling	NPR	OEM	ETS
Ours	DiffK - QP problem	LIC	3 - vision, proximity, touch	17	Velocity damping	Yes	Yes	Yes
[2]	IK - Sequential QP	-	-	15	No	No	Yes	No
[3]	IK - Jacobian based	-	-	16	Damped variant	No	Yes	No
[4]	IK - Nonlinear opt.	-	-	7	No	Yes	Yes	Yes
[5]	DiffK - Damped LS	LIC	0 - known obstacles	8	Damped LS	No	Yes	No
[6]	DiffK - Pseudoinverse	PF	0 - known obstacles	7	No	No	Yes	Yes
[7]	DiffK - Pseudoinverse	RV	1 - vision	7	No	No	No	No
[8]	DiffK - QP problem	LIC	0 - known obstacles	7	Maximize manipulability	No	Yes	No
[9]	DiffK - QP problem	LIC	2 - proximity, touch	7	Velocity damping	Yes	No	No
[10]	DiffK - QP problem	LIC	1 - proximity	7	Velocity damping	No	No	No
[11]	DiffK - QP problem	LIC	0 - known obstacles	7	No	No	Yes	No
[12]	DiffK - QP problem	-	-	12	Maximize manipulability	No	Yes	No
[13]	DiffK - QP problem	LIC	0 - known obstacles	6	No	No	Yes	No
[14]	IK - Nonlinear opt.	CF	0 - known obstacles	8	Maximize condition number	No	Yes	No
[15]	Operational space control	PF	0 - known obstacles	6	No	No	Yes	No
[16]	ERG + PD control	PF	1 - vision	7	No	No	Yes	No
[17]	IK - Nonlinear opt.	RV	2 - vision, touch	10	No	No	Yes	Yes

TABLE I: Comparison of different approaches to solving the IK problem. Control problem acronyms are: differential kinematics (DiffK), inverse kinematics (IK), quadratic programming (QP), least squares (LS), and explicit reference governor (ERG). Reactive obstacle avoidance (ROA) types are linear inequality constraint (LIC), self-motion (SM), potential field (PF), danger index (DI), repulsive vector (RV), and part of the cost function (CF). The last columns are natural position reward (NPR), orientation error minimization (OEM), and explicit trajectory sampling (ETS).

Rozlivek, J., Roncone, A., Pattacini, U., & Hoffmann, M. (2023). HARMONIOUS - Human-like reactive motion control and multimodal perception for humanoid robots. https://arxiv.org/abs/2312.02711
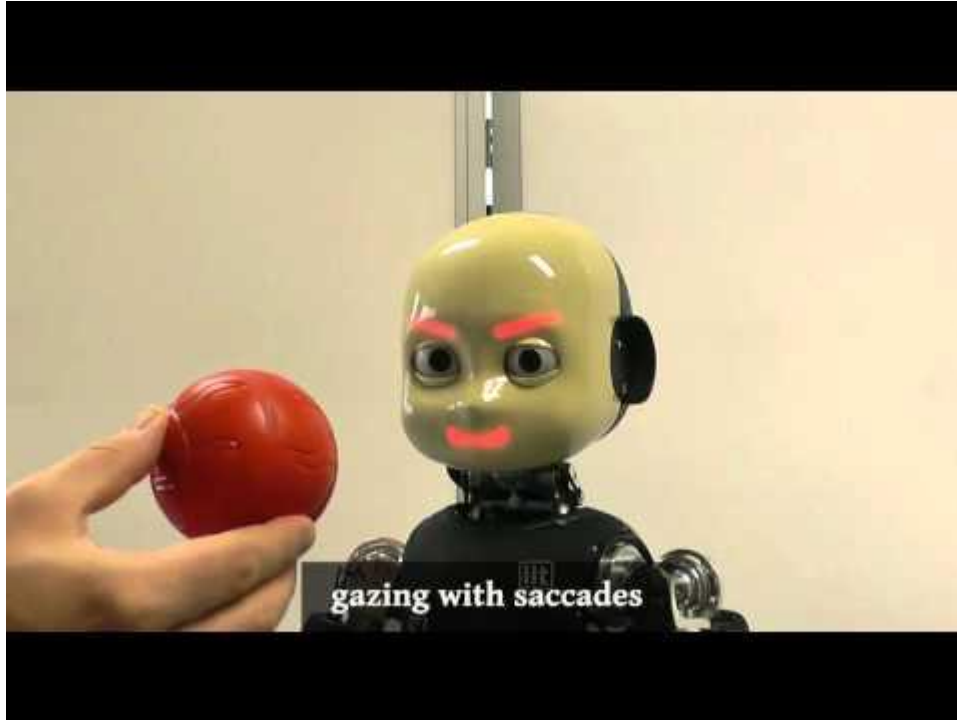
Gaze control

What did you see in the video?

Roncone, A., Pattacini, U., Metta, G., & Natale, L. (2016, June). A Cartesian 6-DoF Gaze Controller for Humanoid Robots. In *Robotics: science and systems* (Vol. 2016).
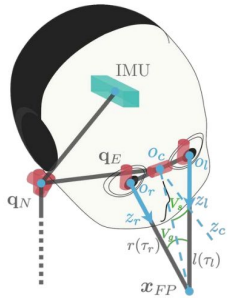
Gaze control – functions

- fixation point in image plane or 3D space
- switching attention
- tracking / smooth pursuit
- image stabilization
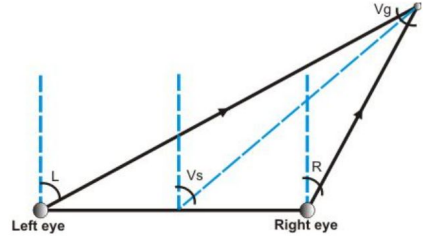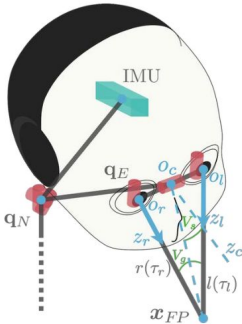- human-robot interaction - communication - e.g. joint attention…
- …

Gaze control as inverse kinematics

Joint #	Part	Joint Name	Range	Unit
0	Neck	**Pitch**	$+/-$	[deg]
1	Neck	**Roll**	$+/-$	[deg]
2	Neck	**Yaw**	$+/-$	[deg]
3	Eyes	**Tilt**	$+/-$	[deg]
4	Eyes	**Version**	$+/-$	[deg]
5	Eyes	**Vergence**	≥ 0	[deg]

- ● What kind of inverse kinematics problem is it?
  - ○ How many DoF in joint space?
  - ○ How many DoF for the task?

- ● What can the redundancy be used for?
  - ○ vestibulo-ocular reflex (VOR)
  - ○ saccadic behavior
  - ○ gaze/image stabilization
  - ○ [All three used in: Roncone, A., Pattacini, U., Metta, G., & Natale, L. (2016, June). A Cartesian 6-DoF Gaze Controller for Humanoid Robots. In Robotics: science and systems.]

$$\begin{cases} V_g = L - R \\ V_s \approx (L + R)/2 \end{cases}$$
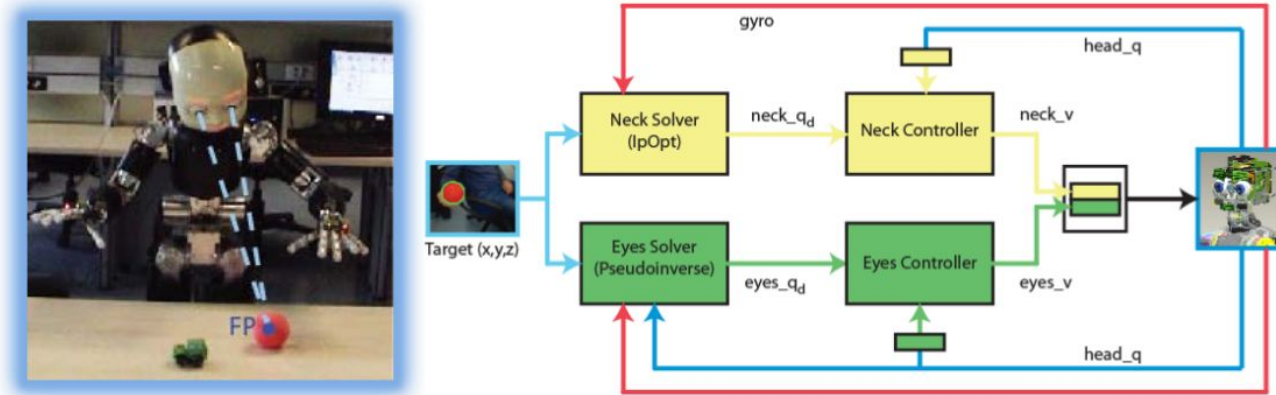
https://github.com/vvv-school/vvv18/blob/master/material/kinematics/kinematics.pdf, courtesy Ugo Pattacini

Gaze/image stabilization
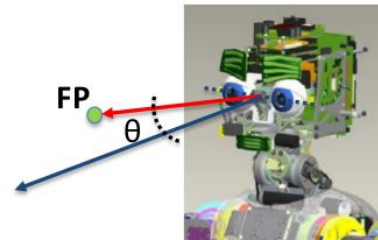

https://youtu.be/_dPlkFPowCc?t=35

iCub gaze controller

Yet another Cartesian Controller: **reuse ideas** ...

Then, apply easy transformations from Cartesian to ...

1. Egocentric angular space
2. Image planes (mono and stereo)

Inverse kinematics solution under multiple task constraints

- For arm reach of a humanoid, whole-body motion motion may be employed.
- Total number of joints, e.g.: $n_{total} = n_{leg} + n_{torso} + n_{arm} = 6 + 1 + 7 = 17$.
- Degree of kinematic redundancy = $17 - 6 = $ **11**.
- With such a high degree of redundancy, it is possible to realize multiple additional tasks.
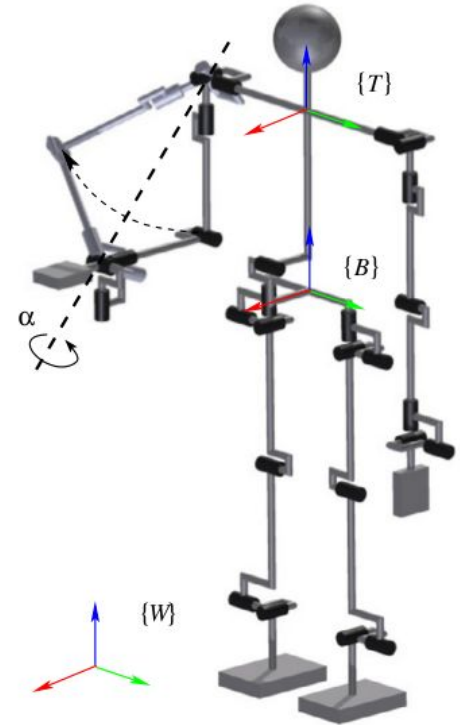
Section 2.8 in Nenchev, D. N., Konno, A., & Tsujita, T. (2018). Humanoid robots: Modeling and control. Butterworth-Heinemann.

Motion task constraints

- Main task: hand motion ~ reaching.
- Other motion tasks:
  - keep balance
  - avoid obstacles
  - avoid self-collisions
  - avoid singularities
  - avoid joint limits
  - gaze task
  - ...
- Constraints helpful in resolving kinematic redundancy.
  - Caveat: Overconstrained state (task conflict).

Section 2.8.1 in Nenchev, D. N., Konno, A., & Tsujita, T. (2018). Humanoid robots: Modeling and control. Butterworth-Heinemann.

Types of motion task constraints

- link-motion constraints
  - movement of reaching hand
  - movement of Center of Mass
- joint-motion constraints
  - joint ranges
  - joint velocity limits
  - singularity avoidance
- equality- and inequality (unilateral)-type constraints
- permanently active and temporal constraints
- high-priority and low-priority constraints

Section 2.8.1 in Nenchev, D. N., Konno, A., & Tsujita, T. (2018). Humanoid robots: Modeling and control. Butterworth-Heinemann.

Examples – Classify

- joint limits
  - inequality, permanent, high-priority
- reaching task
  - inequality / minimization term, middle priority
- obstacles
  - inequality, temporal, high priority
- singularities
  - inequality, temporal, middle priority
- postural constraints
  - min. term / inequality, permanent, low-priority

- equality- and inequality (unilateral)-type constraints
- permanently active and temporal constraints
- high-priority and low-priority constraints

Section 2.8.1 in Nenchev, D. N., Konno, A., & Tsujita, T. (2018). Humanoid robots: Modeling and control. Butterworth-Heinemann.

iCub red ball demo – IK solver (+ Cartesian controller) + gaze controller
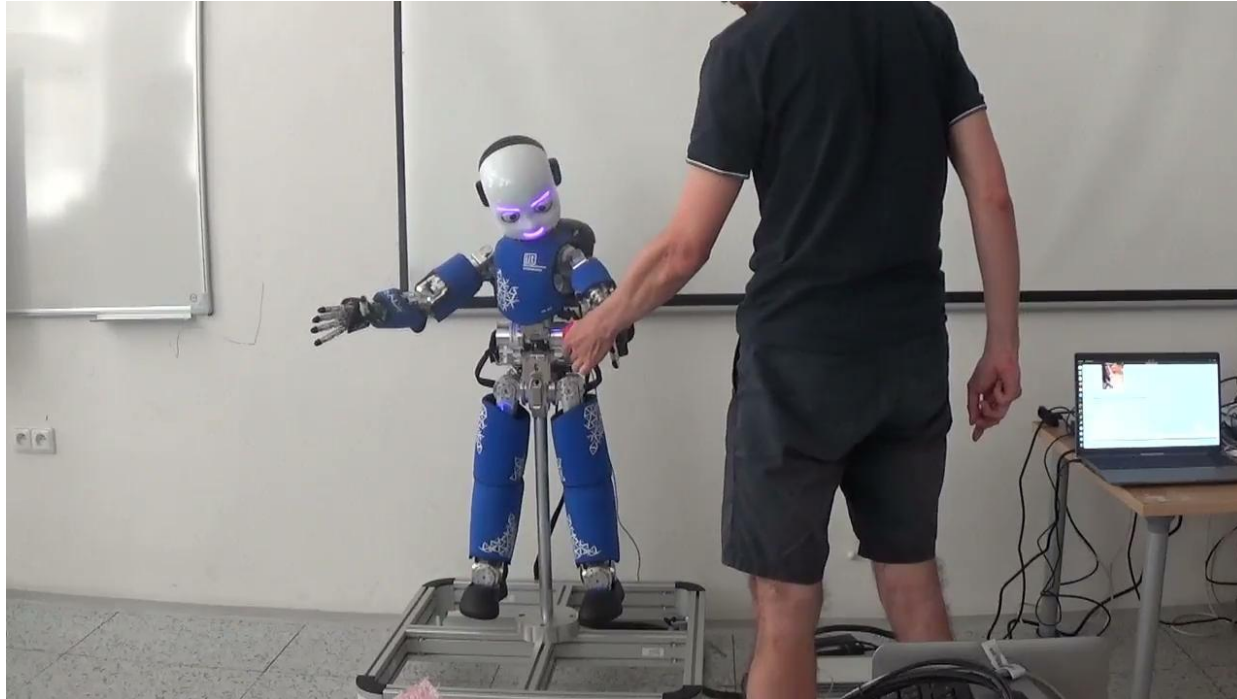
Pattacini, U., Nori, F., Natale, L., Metta, G., & Sandini, G. (2010, October). An experimental evaluation of a novel minimum-jerk Cartesian controller for humanoid robots. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1668-1674). IEEE.

Roncone, A., Pattacini, U., Metta, G., & Natale, L. (2016, June). A Cartesian 6-DoF Gaze Controller for Humanoid Robots. In *Robotics: Science and Systems* (Vol. 2016).

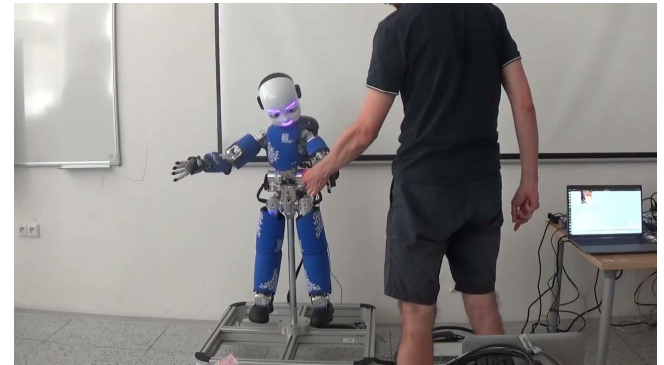# iCub IK solver (+ Cartesian controller) + gaze controller

- Task 1 - Reach
  - Task DoFs: m = 6
    - https://github.com/robotology/icub-basic-demos/blob/master/demoRedBall/src/main.cpp#L147
  - Joint DoFs: ?
    - m = 7 per arm + 2 of the torso = 9
    - https://github.com/robotology/icub-basic-demos/blob/master/demoRedBall/src/main.cpp#L136
- Task 2 - Gaze
  - Task DoFs: m = 3
  - Joint DoFs: n = 6
  - https://github.com/robotology/icub-basic-demos/blob/master/demoRedBall/src/main.cpp#L898

https://github.com/robotology/icub-basic-demos

Pattacini, U., Nori, F., Natale, L., Metta, G., & Sandini, G. (2010, October). An experimental evaluation of a novel minimum-jerk Cartesian controller for humanoid robots. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1668-1674). IEEE.

Roncone, A., Pattacini, U., Metta, G., & Natale, L. (2016, June). A Cartesian 6-DoF Gaze Controller for Humanoid Robots. In *Robotics: Science and Systems* (Vol. 2016).

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

Conflict resolution for multiple motion task constraints

- Inverse kinematics with constraints in a **hierarchical structure**, using null space projections
- How to set priorities? (Sentis & Khatib 2005)
  - joint motion constraints
  - link-motion constraints
    - balance - CoM control
    - hand position
  - postural-variation constraints
- Pros: Stability may be guaranteed.
- Cons: Difficulty of incorporating inequality constraints.

- Inverse kinematics as multiobjective optimization problem
- Pros: All kinds of constraints, incl. inequality constraints, can be incorporated, even on the run.
- Cons: Difficult to guarantee control stability.

Section 2.8.1 in Nenchev, D. N., Konno, A., & Tsujita, T. (2018). Humanoid robots: Modeling and control. Butterworth-Heinemann.
L. Sentis, O. Khatib, Synthesis of whole-body behaviors through hierarchical control of behavioral primitives, International Journal of Humanoid Robotics 02 (2005) 505–518.

iCub balancing and reaching
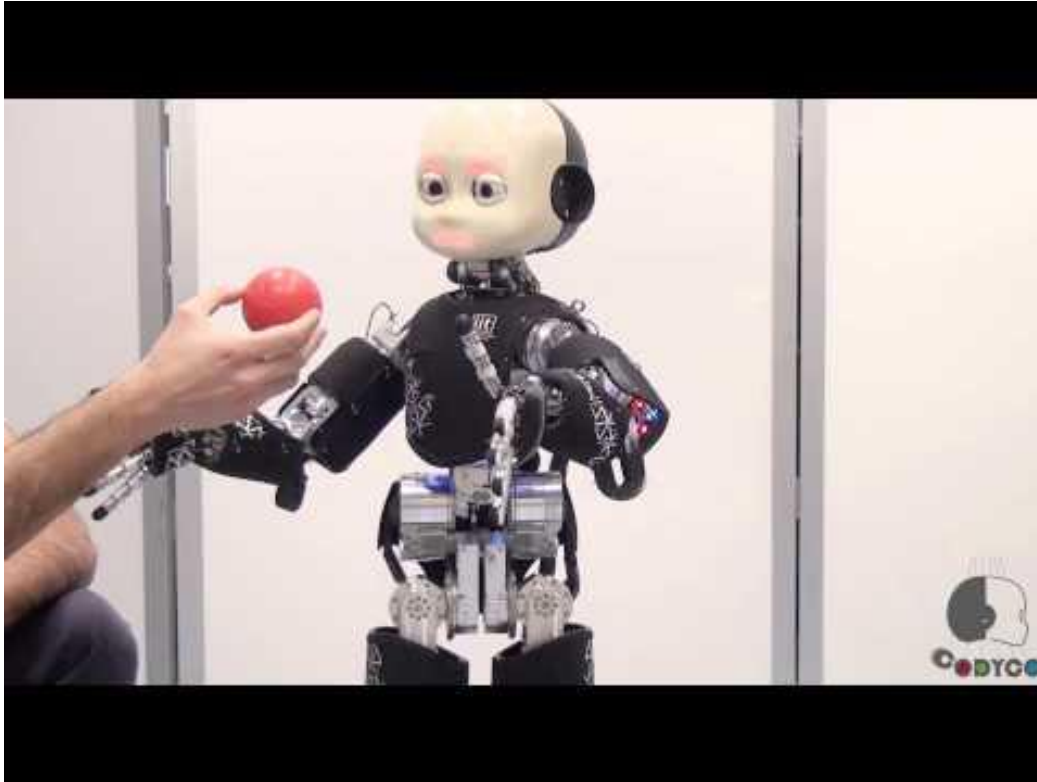
Set of admissible tasks:

- right foot wrench task - regulate the right foot interaction wrench to a predefined value
- left foot wrench
- right arm wrench
- left arm wrench
- postural task
- reaching task
- gaze task

(N.B. wrench is the analog of twist for forces - 6D "force")

https://youtu.be/7CxaynVnsCI

Nori, F., Traversaro, S., Eljaik, J., Romano, F., Del Prete, A., & Pucci, D. (2015). iCub whole-body control through force regulation on rigid non-coplanar contacts. *Frontiers in Robotics and AI*, *2*, 6. https://www.frontiersin.org/articles/10.3389/frobt.2015.00006/full

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

Wrap–up

- Kinematic redundancy
- Singular configurations
- Manipulability
- Self motion / Null space
- Inverse kinematics
  - Analytical / closed-form
  - Iterative - Jacobian (pseudo)inverse
  - Optimization without Jacobian inverse

Matej Hoffmann, Humanoid robots, FEE CTU in Prague, 2024

Resources

- Book sections
    - Sections 2.4.2 - 2.7.3 in Nenchev, D. N., Konno, A., & Tsujita, T. (2018). Humanoid robots: Modeling and control.
    - Ch.5 Velocity kinematics and statics in Lynch, K. M., & Park, F. C. (2017). Modern robotics. (see also https://youtu.be/6tj8QLF69Ok).
    - Part III in Corke, P. I. (2013). Robotics, vision and control: fundamental algorithms in MATLAB Berlin: Springer.

- Online resources
    - Howie Choset: https://www.slideserve.com/antonia/inverting-the-jacobian-and-manipulability
    - Modern robotics (Lynch and Park): https://modernrobotics.northwestern.edu/nu-gm-book-resource/velocity-kinematics-and-statics/
    - http://handbookofrobotics.org/view-chapter/videodetails/10

- Tutorials
    - https://github.com/jhavl/dkt - Jupyter notebooks
    - Haviland, J., & Corke, P. (2023). Manipulator Differential Kinematics: Part I: Kinematics, Velocity, and Applications. *IEEE Robotics & Automation Magazine*. https://doi.org/10.1109/MRA.2023.3270228
    - Haviland, J., & Corke, P. (2023). Manipulator Differential Kinematics: Part 2: Acceleration and Advanced Applications. *IEEE Robotics & Automation Magazine*. https://doi.org/10.1109/MRA.2023.3270221

- Articles
    - Chiaverini, S., Oriolo, G., & Maciejewski, A. A. (2016). Redundant robots. In *Springer Handbook of Robotics* (pp. 221-242). Springer, Cham.
    - Lloyd, S., Irani, R. A., & Ahmadi, M. (2022). Fast and Robust Inverse Kinematics of Serial Robots Using Halley's Method. IEEE Transactions on Robotics, 38(5), 2768-2780.
    - Pattacini, U., Nori, F., Natale, L., Metta, G., & Sandini, G. (2010, October). An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots. In *2010 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1668-1674). IEEE.
    - Rozlivek, J., Roncone, A., Pattacini, U., & Hoffmann, M. (2023). HARMONIOUS--Human-like reactive motion control and multimodal perception for humanoid robots. https://arxiv.org/abs/2312.02711