Humanoid robots - Kinematics II

Mgr. Matěj Hoffmann, Ph.D.

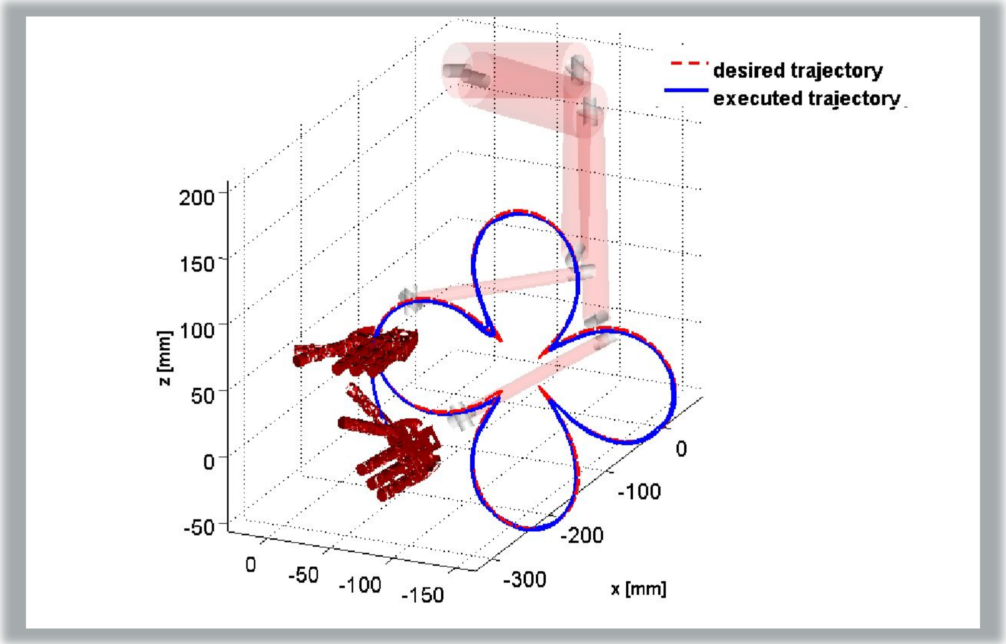
Kinematics - forward and inverse

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

The Problem

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

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

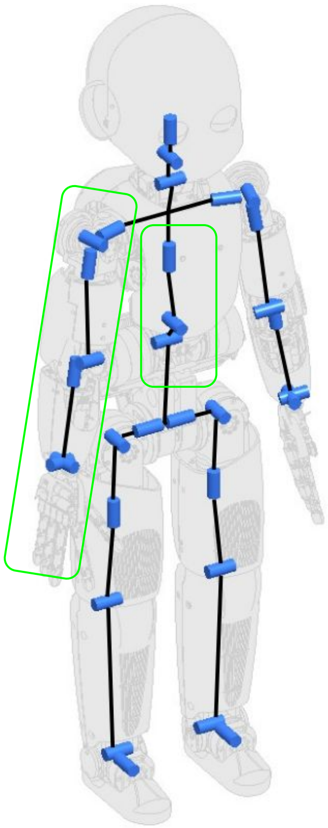


Slide source: https://github.com/vvv-school/vvv18/blob/master/material/kinematics/kinematics.pdf. Courtesy Ugo Pattacini.

How about iCub?

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.

Forward and inverse kinematics - recap

n=7, m = 6 (x,y,z, ϕ, θ, ψ)

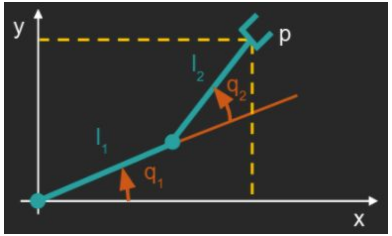
Consider a general case: $\mathbf{q} \in \mathbb{R}^n \quad \mathbf{x} \in \mathbb{R}^m$

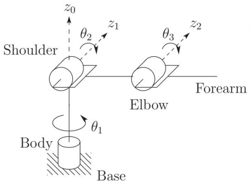
n=2, m = 3 (x,y,ϕ)

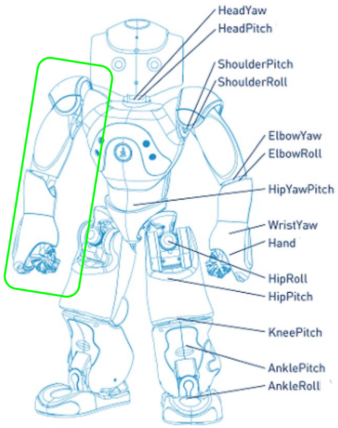
n=6, m = 6 (x,y,z, ϕ, θ, ψ)

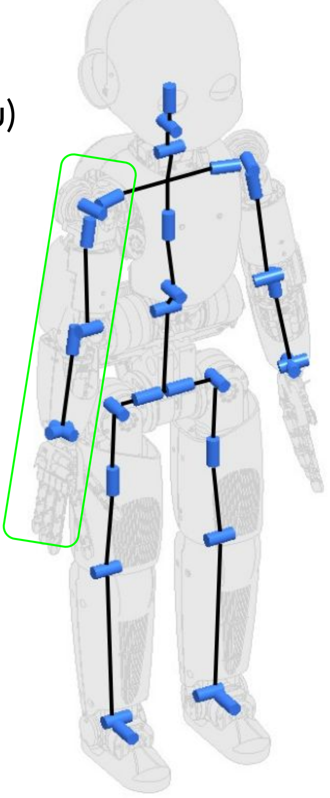
n=5, m = 6 (x,y,z, ϕ, θ, ψ)

What is n and m?









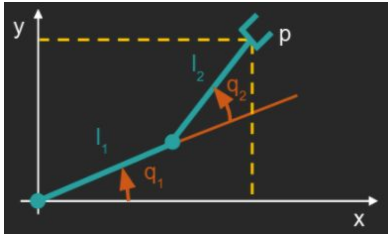
	Mapping type	linear / nonlinear
$\mathbf{x} = \mathbf{f}(\mathbf{q})$		
$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x})$		

Forward and inverse kinematics - recap

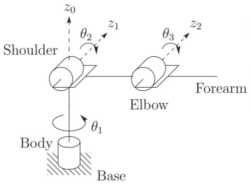
Consider a general case: $\mathbf{q} \in \mathbb{R}^n \quad \mathbf{x} \in \mathbb{R}^m$

What is n and m?

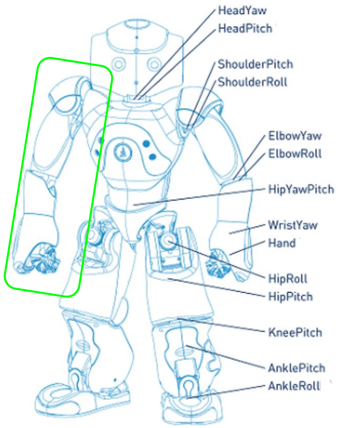
n=2, m = 3 (x,y,ϕ)



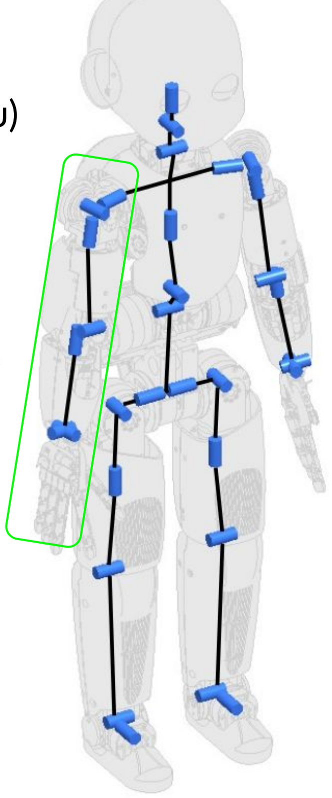
n=6, m = 6 (x,y,z, ϕ, θ, ψ)



n=4, m = 6 (x,y,z, ϕ, θ, ψ)



n=7, m = 6 (x,y,z, ϕ, θ, ψ)



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

Inverse kinematics - analytic (closed-form) solutions

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

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

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

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

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

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.

Inverse kinematics - closed form

- appealing
- fast
- laborious
- not very flexible w.r.t. new constraints...

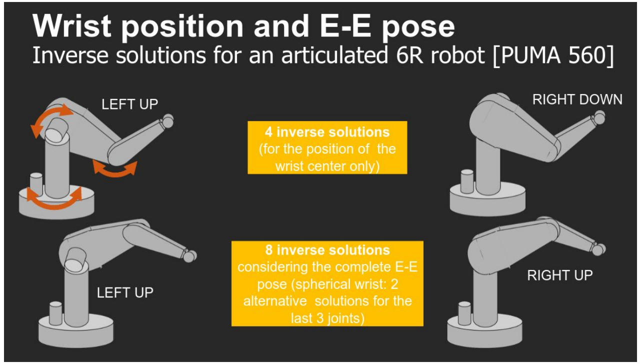


TABLE 1
Number of Analytic Solutions for 6-Degree-of-Freedom Systems

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

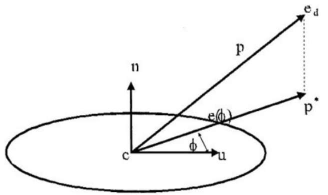


FIG. 3. Finding the elbow position that is the closest possible to a desired position.

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

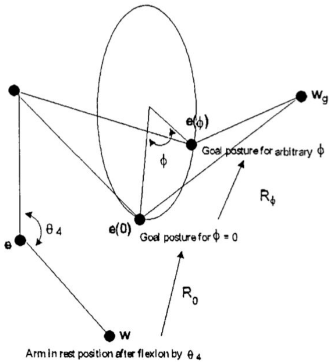


FIG. 5. Decomposing $\mathbf{R}_1$ into $\mathbf{R}_0$ and $\mathbf{R}_\phi$.

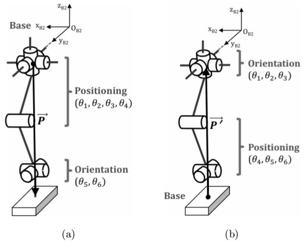


Fig. 5. (a) Forward Decoupling of the Right Leg of a Hubo KHR-4 Robot, (b) Reverse Decoupling of the Right Leg of a Hubo KHR-4 Robot.

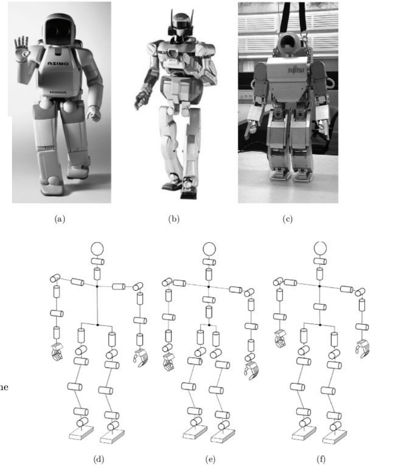


Fig. 4. (a) HONDA ASIMO Robot and its associated kinematic diagram in (d), (b) AIST HRP-2 Robot and its associated kinematic diagram in (e), and (c) Fujitsu HOAP-2 Robot and its associated kinematic diagram in (f).

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.

Other stakeholders

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

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/SHplmEc6iv0?t=156

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

- 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: in the planning and execution of smooth trajectories, in the determination of singular configurations, in the execution of coordinated anthropomorphic motion, in the derivation of the dynamic equations of motion, and in the 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 by V. Smutný (B3B33ROB1).

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

Geometrical Jacobian

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

- 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

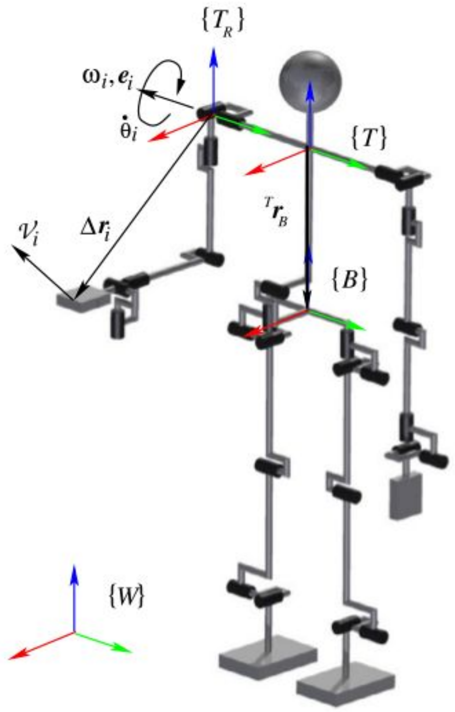


FIGURE 2.3 End-link spatial velocity $\mathcal{J}_i = [(e_i \times r_i)^T \;\; e_i^T]^T$ is obtained with joint rate $\dot{\theta}_i = 1$ rad/s. Vector e_i signifies the joint axis of rotation. The position r_i of the characteristic point on the end link is determined w.r.t. reference frame $\{T_R\}$, 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.

Analytical Jacobian

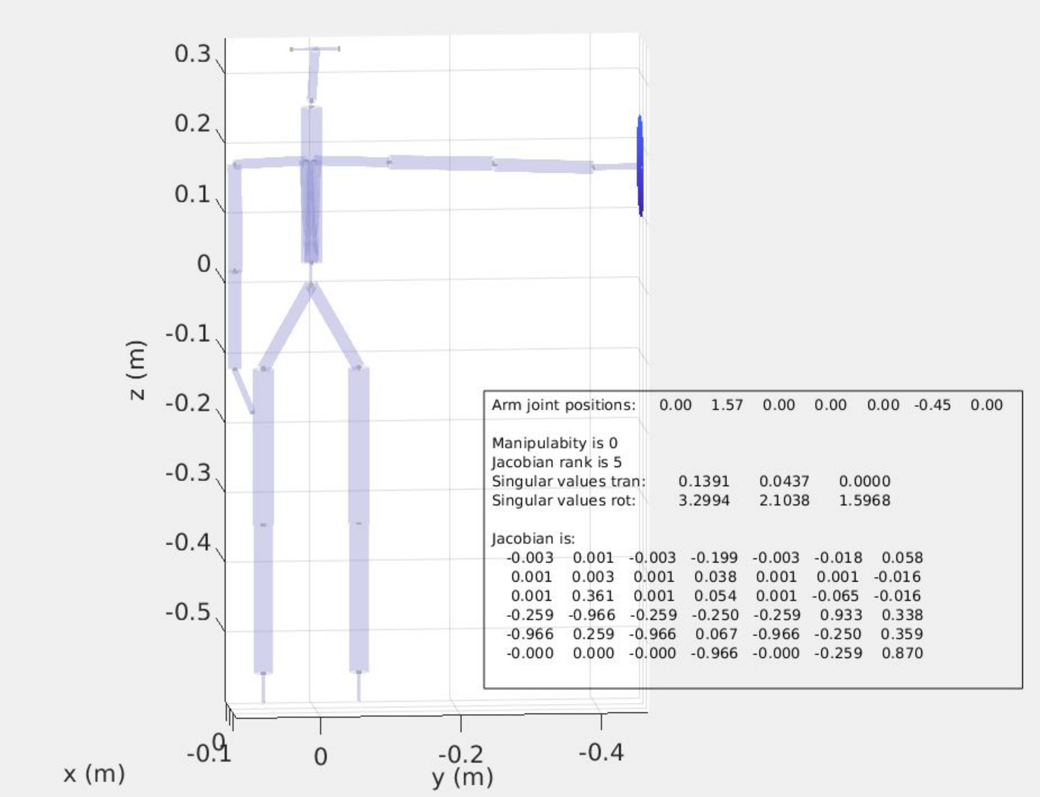
$$\mathbf{x} = \mathbf{f}(\mathbf{q}) \overset{d/dt}{\Longrightarrow} \dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$$

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

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

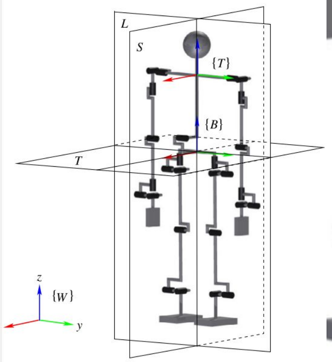
Slide source: https://github.com/vvv-school/vvv18/blob/master/material/kinematics/kinematics.pdf. Courtesy Ugo Pattacini.

iCub matlab demo



Differential kinematics at singular configurations

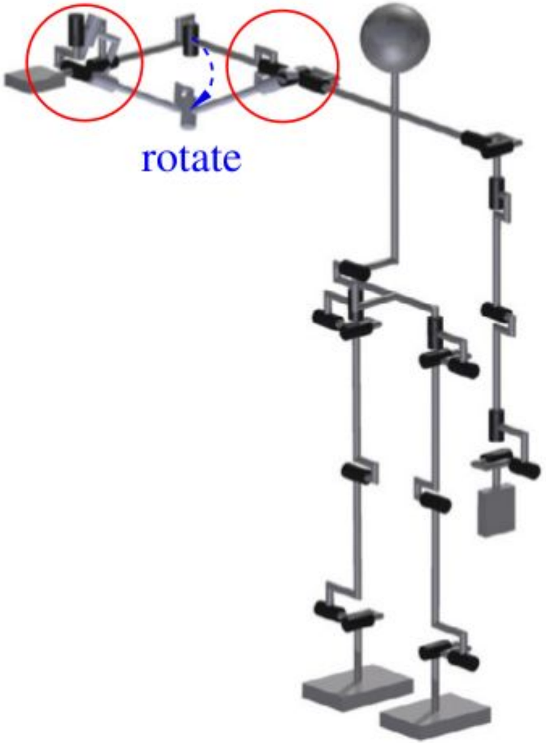
- There are certain limb configurations where the end link loses mobility, i.e. the ability of instantaneous motion in one or more directions.
- At singularities, bounded end-effector velocities may correspond to unbounded joint velocities.
- Since the arms are fully extended, the hands cannot move in 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

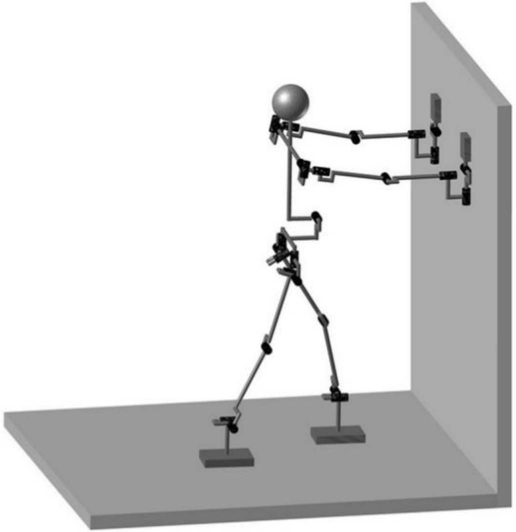
- “Fig. 2.4C shows another type of singular configuration for the right arm. The singularity is due to the alignment of the two axes in the shoulder joint, the elbow joint being at 90 degrees. The end link loses mobility in the translational direction of the lower-arm link. This configuration is called *shoulder singularity*. Note that the end link is placed within the workspace; it is not on the boundary. In this case, the self-motion of the arm, i.e. a motion whereby the end link is fixed, yields a transition to a nonsingular configuration. Such types of singular configurations are characterized as *avoidable*.”



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.

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

$$\boldsymbol{J}(\boldsymbol{\theta}) = \boldsymbol{U}(\boldsymbol{\theta})\boldsymbol{\Sigma}(\boldsymbol{\theta})\boldsymbol{V}(\boldsymbol{\theta})^T, \tag{2.26}$$

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

$$\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \left[\mathrm{diag}\{\sigma_1(\boldsymbol{\theta}),\ \sigma_2(\boldsymbol{\theta}), ..., \sigma_6(\boldsymbol{\theta})\} \ \middle|\ \boldsymbol{0}\right] \in \Re^{6\times n}. \tag{2.27}$$

Here $\sigma_1 \geq \sigma_2 \geq, ..., \geq \sigma_6 \geq 0$ are the singular values of the Jacobian. The columns of matrix $\boldsymbol{U}(\boldsymbol{\theta})$, $\boldsymbol{u}_i$, $i = 1, ..., 6$, 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 $6 - \rho$ $(\rho = \mathrm{rank}\,\boldsymbol{J})$, $6 - \rho$ 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 $\boldsymbol{u}_i$. Assuming that the magnitude of the joint rate vector is limited at each limb configuration as $\|\dot{\boldsymbol{\theta}}\| \leq 1$, 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 $\boldsymbol{u}_{min}$ becomes a singular direction. Vectors $\sigma_i \boldsymbol{u}_i$ 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

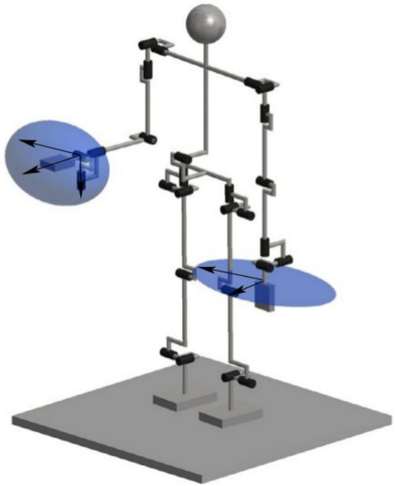
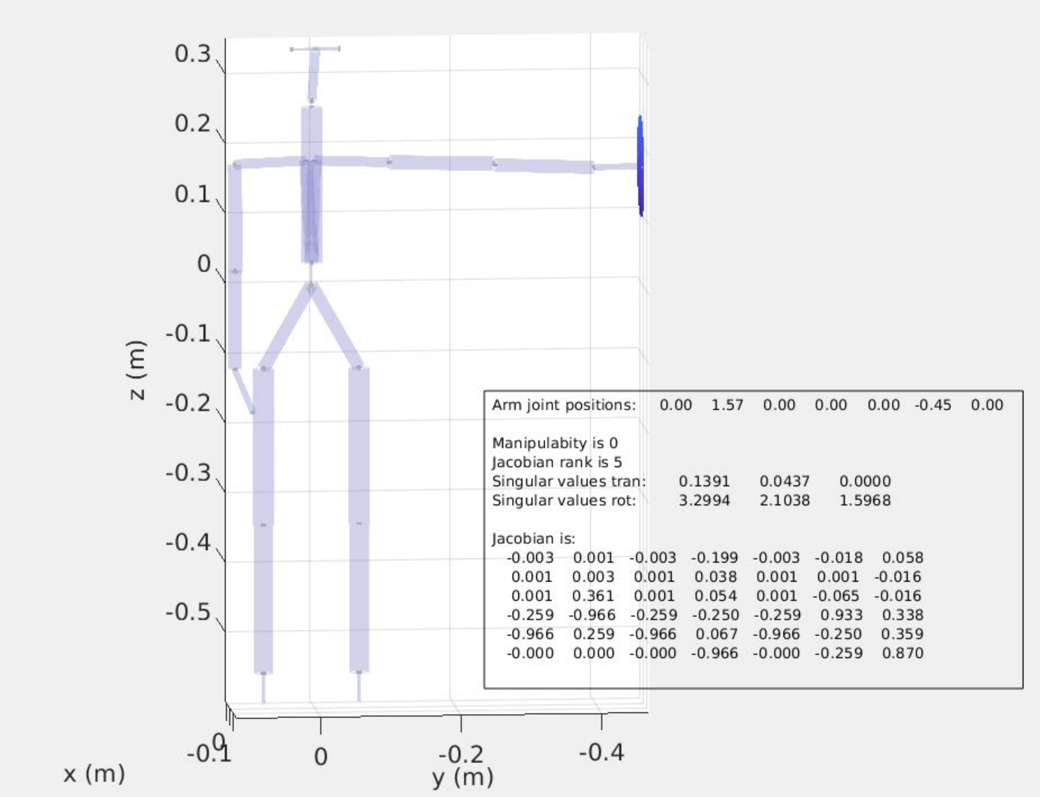


FIGURE 2.8 Manipulability ellipsoid for translational motion. The right arm is in a nonsingular configuration and the respective ellipsoid is 3D, with principal axes $\sigma_1 \boldsymbol{u}_1, \sigma_2 \boldsymbol{u}_2$, and $\sigma_3 \boldsymbol{u}_3$. The left arm is at a singular configuration: the downward translational mobility has been lost, and therefore, the manipulability ellipsoid is only 2D. The principal axes are $\sigma_1 \boldsymbol{u}_1$ and $\sigma_2 \boldsymbol{u}_2$.

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



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

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:

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

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

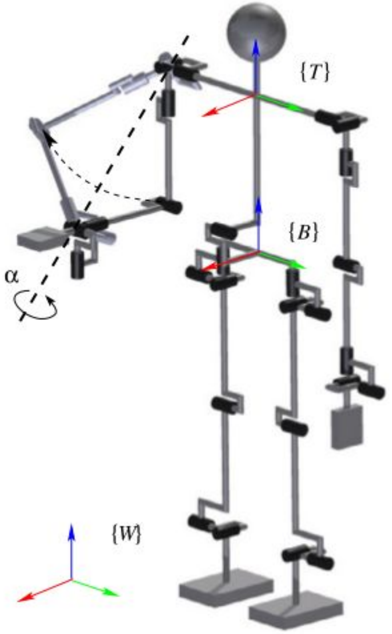


FIGURE 2.9 The self-motion of the arm is shown as a rotation of the arm plane, determined by the upper/lower arm links, around the line connecting the shoulder and wrist joints. The rotation angle α can be associated with parameter b_v in (2.35).

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

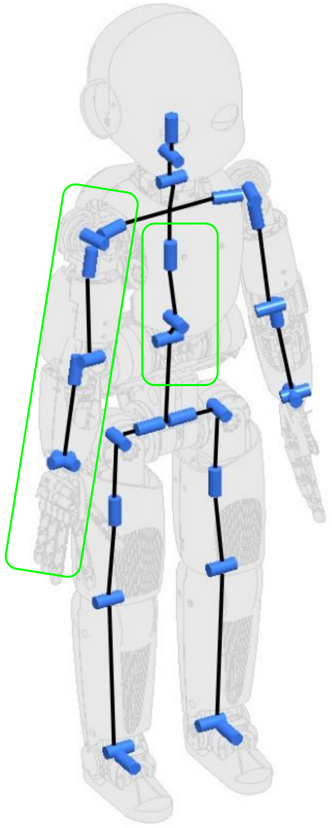
Inverse differential kinematic relations

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.

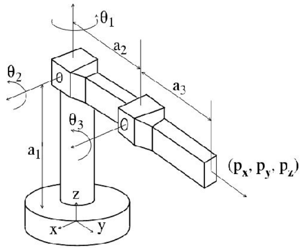
Inverse differential kinematic relations - challenges

- 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 $\mathcal{V}_n = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$ 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*.

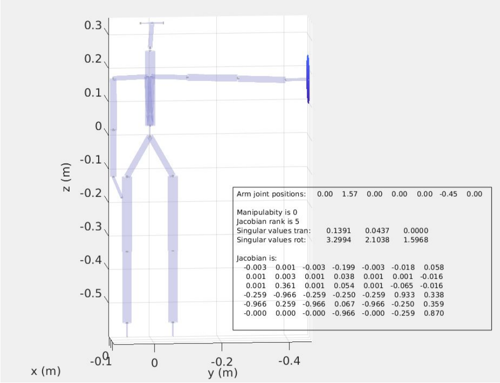
Section 2.4.3 in Nenchev, D. N., Konno, A., & Tsujita, T. (2018). Humanoid robots: Modeling and control. Butterworth-Heinemann.

Forward, inverse, and differential kinematics - recap

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



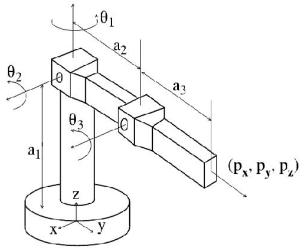
$$p_x = \cos\theta_1\,(a_3\cos(\theta_2+\theta_3) + a_2\cos\theta_2)$$
$$p_y = \sin\theta_1\,(a_3\cos(\theta_2+\theta_3) + a_2\cos\theta_2)$$
$$p_z = a_3\sin(\theta_2+\theta_3) + a_2\sin\theta_2 + a_1$$



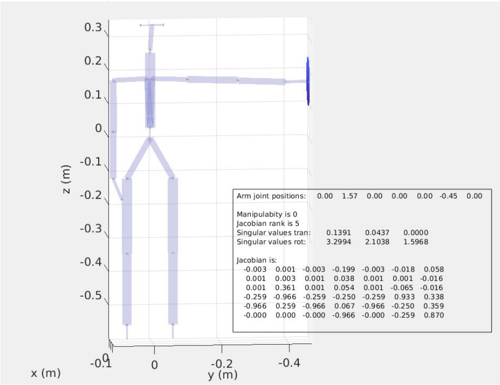
https://www.slideserve.com/antonia/inverting-the-jacobian-and-manipulability

Forward, inverse, and differential kinematics - recap

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



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



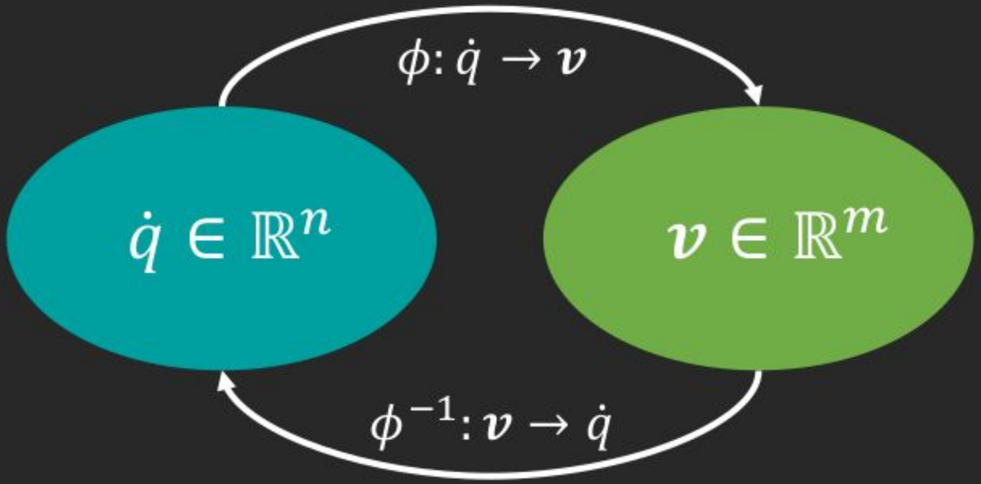
https://www.slideserve.com/antonia/inverting-the-jacobian-and-manipulability

Using differential kinematics for IK

- When closed-form solutions do not exist.
- They are numerical, iterative methods.
    - [Jacobian transpose]
    - Jacobian pseudoinverse
        - Used by Orocos Kinematics and Dynamics Library (KDL) - used in ROS.
    - Damped least squares

- 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)
    - 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

Inversion of differential kinematics



Find the joint velocity vector that realizes a **desired** end-effector [linear and angular] **velocity**

slide courtesy Alessandro Roncone

Incremental solution to IK problems

- Joint velocity inversion can be used to solve on-line and incrementally a sequence of IK problems
- Each problem differs by a small amount dr from the previous one

Direct kinematics: $r = f_r(q)$

Differential kinematics : $dr = \frac{\partial f_r(q)}{\partial q} dq = J_r(q) dq$

$$r \rightarrow r + dr$$

First, increment the desired task variables

$$r + dr = f_{r(q)}$$

➡ Then, solve the inverse kinematics problem

$$q = f_r^{-1}(r + dr)$$

First, solve the differential inverse kinematics problem

$$dq = J_r^{-1}(q) dr$$

➡ Then, increment the original joint variables

$$q \rightarrow d + dq$$

slide courtesy Alessandro Roncone

Inversion of differential kinematics

$$v = J(q)\dot{q} \qquad \Leftrightarrow \qquad \dot{q} = J^{-1}(q)v$$

[if J is square and non-singular]

Problems:

- Near a **singularity** of the Jacobian matrix (high $\dot{q}$)
- For **redundant** robots (J is $[6xn], n > 6$ → there is no standard “inverse” of a rectangular matrix)
- **More robust inversion methods are needed!**

slide courtesy Alessandro Roncone

Pseudo-inverse method

$J^{\#}$ is also known as Moore-Penrose inverse of J

$$\min_{\dot{q}} H = \frac{\lambda}{2}\|\dot{q}\|^2 \quad \text{such that } J\dot{q} - v = 0$$

$$\min_{\dot{q}\epsilon S} H = \frac{\lambda}{2}\|\dot{q}\|^2$$

$$S = \{\, \dot{q}\epsilon\mathbb{R}^n : \|J\dot{q} - v\| \text{ is minimum}\}$$

Solution → $\dot{q} = J^{\#}v\,,\quad J^{\#} = J^T\left(JJ^T\right)^{-1}$

- Inversion of differential kinematics as a **CONSTRAINED optimization problem**
- If $v \in \Re(J)$ → the constraint is satisfied
- If $v \notin \Re(J)$ → the constraint is impossible:
  - $J\dot{q} = v^{\perp}$ → the **orthogonal** projection of v on $\Re(J)$
  - $v^{\perp}$ minimizes the error $\|J\,\dot{q}\; - v\|$

slide courtesy Alessandro Roncone

Pseudo-inverse method

$J^\#$ is also known as Moore-Penrose inverse of J

- $J^\#$ is the only matrix that satisfies the four relationships:

$$JJ^\#J = J \qquad J^\#JJ^\# = J^\# \qquad (J^\#J)^T = J^\#J \qquad (JJ^\#)^T = JJ^\#$$

- If rank $\rho = m = n \rightarrow J^\# = J^{-1}$
- If rank $\rho = m < n \rightarrow J^\# = J^T(JJ^T)^{-1}$
- $J^\#$ **always** exists and can be computed numerically using Singular Value Decomposition [SVD, e.g. in Matlab with pinv]

slide courtesy Alessandro Roncone

Pseudo-inverse method

$J^{\#}$ is also known as Moore-Penrose inverse of J

$$\min_{\dot q} H = \frac{\lambda}{2}\|\dot q\|^2 \quad \text{such that } J\dot q - v = 0$$

⟷

$$\min_{\dot q \epsilon S} H = \frac{\lambda}{2}\|\dot q\|^2$$

$$S = \{\, \dot q \epsilon \mathbb{R}^n : \|J\dot q - v\| \text{ is minimum}\}$$

Solution → $\dot q = J^{\#} v\,,\quad J^{\#} = J^T\left(JJ^T\right)^{-1}$

- At singularity:
  - $\det\left(JJ^T\right) = 0$
  - Pseudo-inverse is ill-defined
  - Inverse kinematics $dq = J^T\left(JJ^T\right)^{-1}$dv computes “infinite” steps

slide courtesy Alessandro Roncone

Damped Least Square

Also known as Levenberg–Marquardt method

$$\min_{\dot{q}} H = \frac{\lambda}{2}\|\dot{q}\|^2 + \frac{1}{2}\|J\dot{q} - v\|^2, \qquad \lambda \geq 0$$

$$\dot{q} = (\lambda I_n + J^T J)^{-1} J^T v = J^T(\lambda I_m + JJ^T)^{-1} v$$

J_{DLS} [this expression is better for redundant robots]

J_{DLS} can be used for both $m = n$ and $m < n$ cases!!

- Inversion of differential kinematics as an **optimization problem**
- $H \rightarrow$ weighted sum of: i) minimum error norm on end-effector velocity; ii) minimum norm of joint velocity
- J_{DLS} is singularity robust
- λI_m is called regularization term
- $\lambda = 0$ when "far enough" from a singularity
- If $\lambda > 0 \rightarrow$ there is a vector error $\varepsilon = v - J\dot{q}$ in executing the desired end-effector velocity v, but the joint velocities are **always reduced ("damped")**

slide courtesy Alessandro Roncone

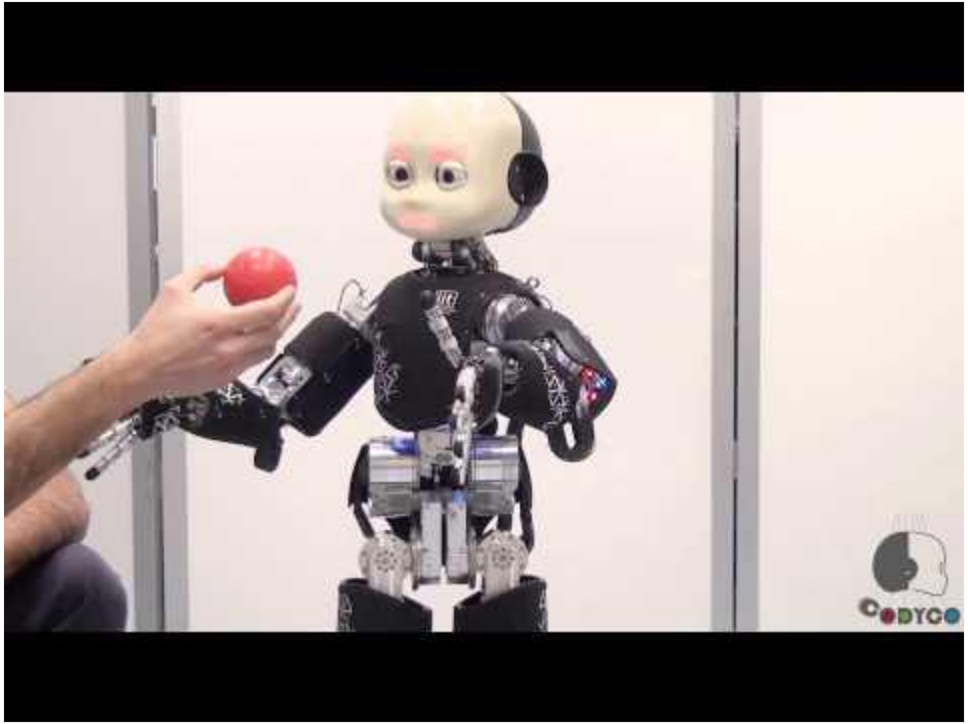
IK for humanoid-like robots is an active research topic

- 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.

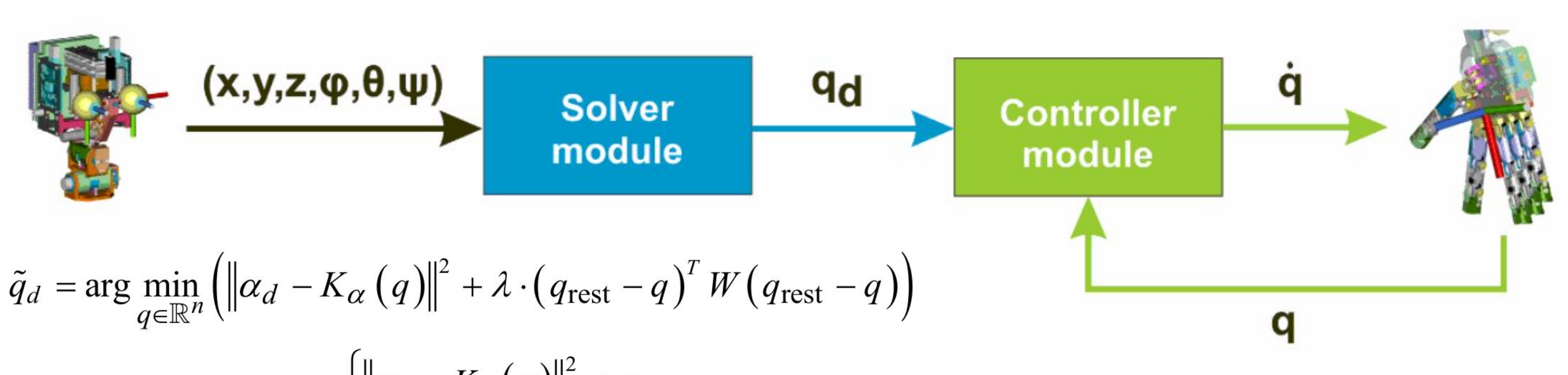
iCub IK solver (+ Cartesian controller)



https://youtu.be/7CxaynVnsCI

(video shows more recent work
- combined with balancing)

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.

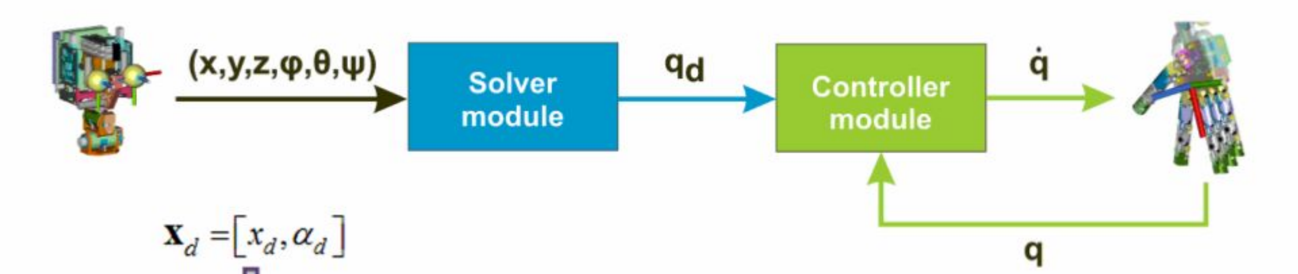


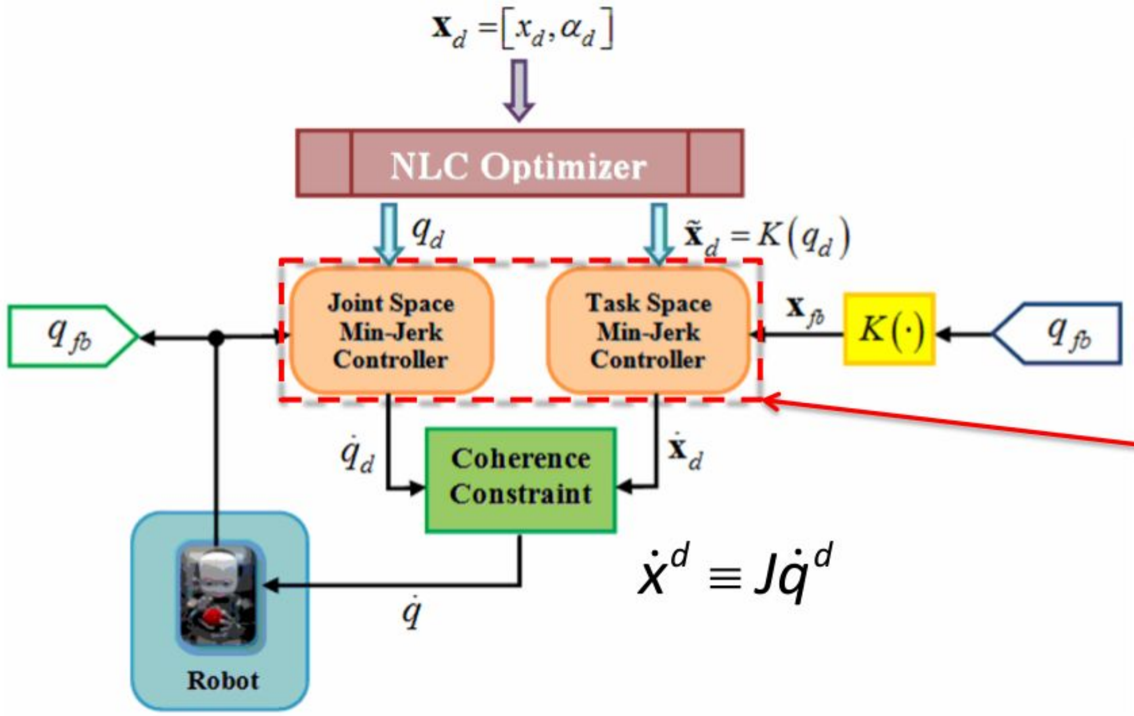
$$\tilde{q}_d = \arg\min_{q \in \mathbb{R}^n} \left(\left\| \alpha_d - K_\alpha(q) \right\|^2 + \lambda \cdot (q_{\text{rest}} - q)^T W (q_{\text{rest}} - q) \right)$$

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

- **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

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





Multi Referential Dynamical Systems

M. Hersch, A.G. Billard, “*Reaching with multi-referential dynamical systems*”, Springer-Verlag.

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

Inverse kinematics for humanoids

(a bit speculative)

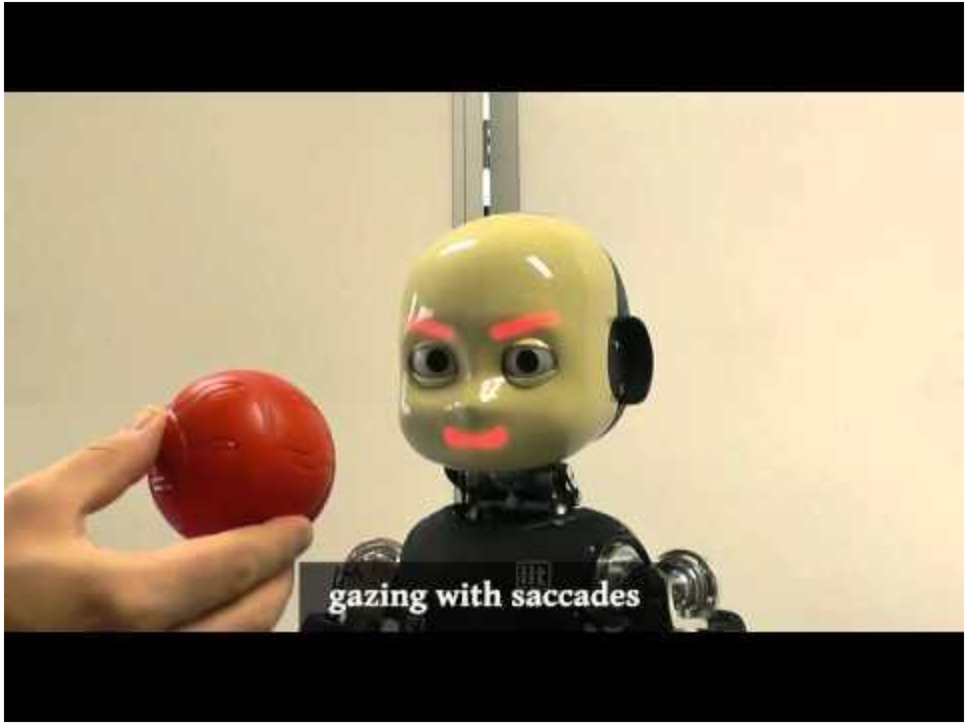
There’s no free lunch...

Method	General solution (not robot specific)	Possibility of including additional constraints	Problems with singularities	Problems with local minima	Runtime	Examples
Analytical (closed-form)	😞	😑	😌	😌	😄	
Iterative using some form of Jacobian inverse	😌	😞	😞	😑	😌	KDL / Orocos (ROS)
Optimization using forward kinematics	😌	😑	😌	😑	😌	iCub IK solver

Gaze control

https://youtu.be/I4ZKfAvs1y0

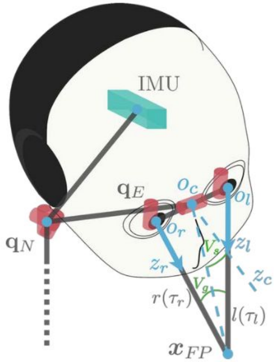
https://youtu.be/_dPlkFPowCc?t=35



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).

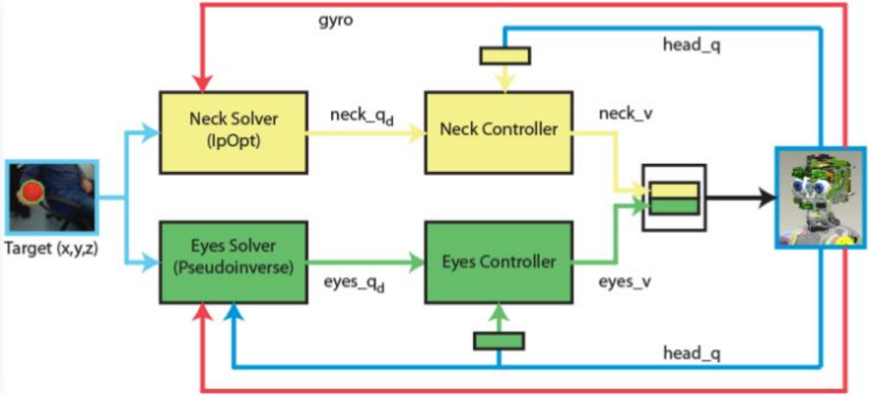
Gaze control

- 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?



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]

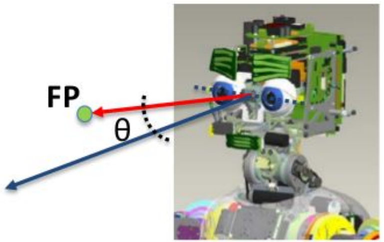
iCub gaze controller



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

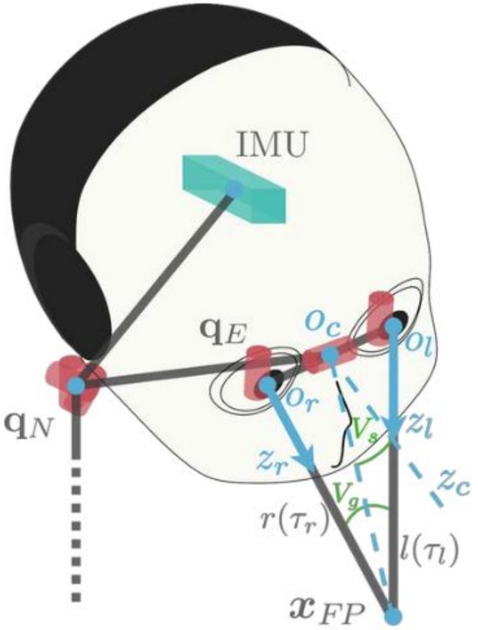
Then, apply easy transformations from Cartesian to ...

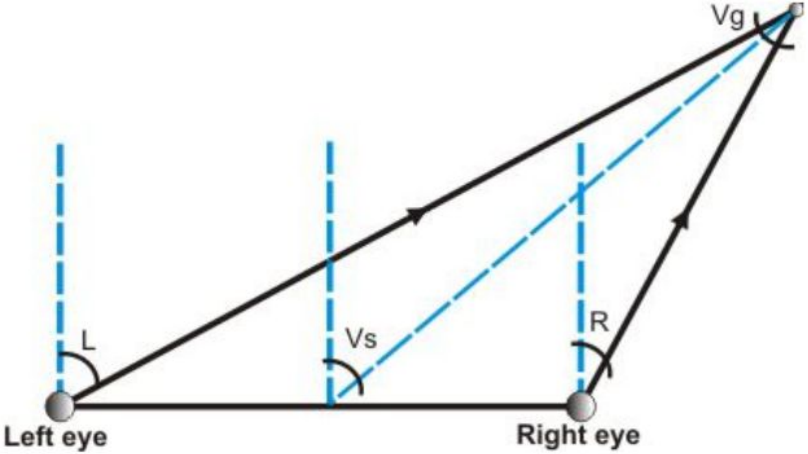
1. Egocentric angular space
2. Image planes (mono and stereo)



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

iCub gaze controller

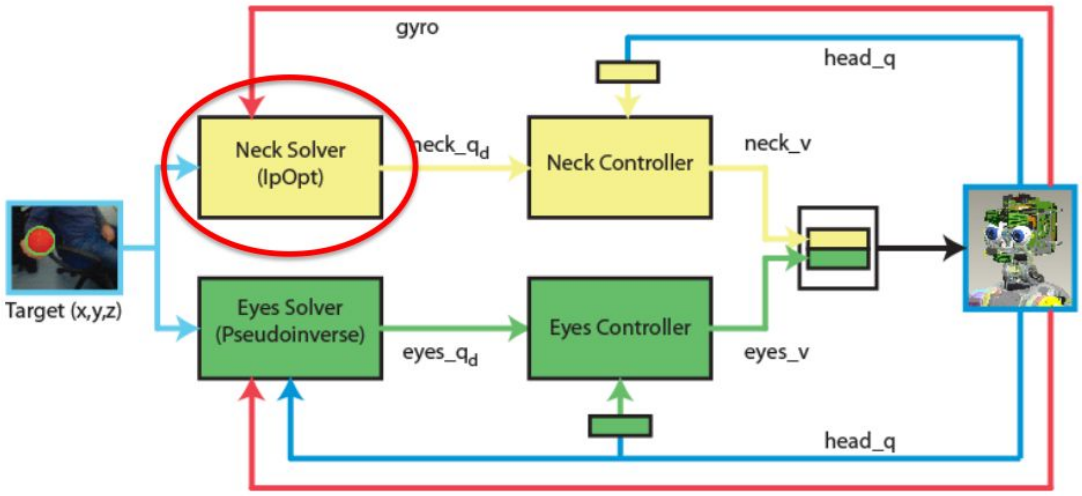


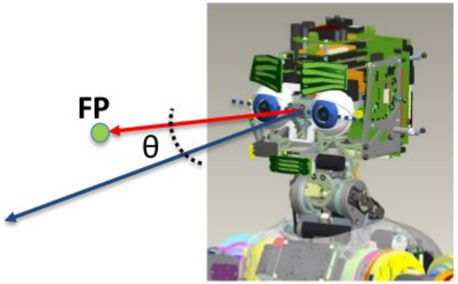


$$\begin{cases} V_g = L - R \\ V_s \approx (L + R)/2 \end{cases}$$

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

iCub gaze controller



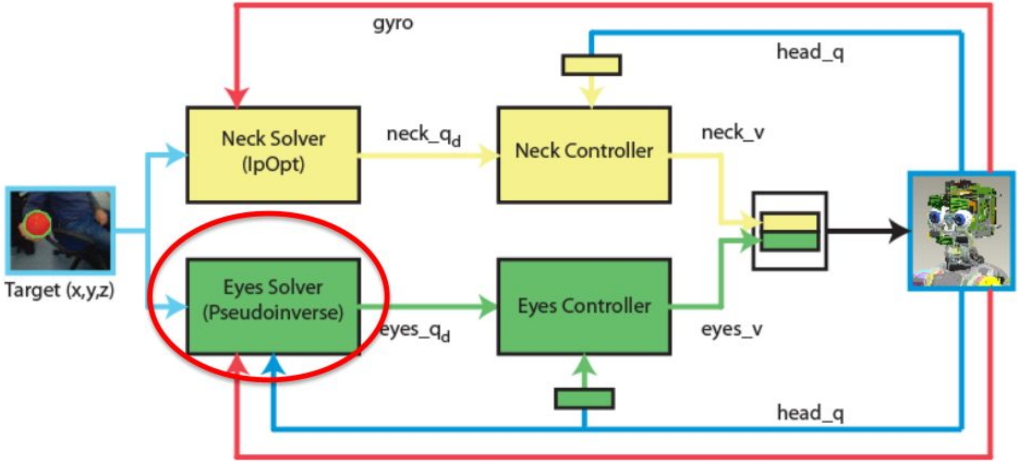


$$q^*_{neck} = \arg \min_{q_{neck} \in \mathbb{R}^3} \left\| q_{rest} - q_{neck} \right\|^2$$

$$\text{s.t.} \begin{cases} \cos\left(\theta\left(q_{neck}\right)\right) > 1 - \varepsilon \\ q_{neck_L} < q_{neck} < q_{neck_U} \end{cases}$$

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

iCub gaze controller



$$q^{*}_{\text{eyes}} = \arg \min_{q_{\text{eyes}} \in \mathbb{R}^3} \left\| FP_d - K_{FP}\left(q_{\text{eyes}}\right) \right\|^2$$

Gyro

$$q_{\text{eyes}_{t+1}} = q_{\text{eyes}_t} + \Delta T\left(G \cdot J^{\#} \cdot \left(FP_d - K_{FP}\left(q_{\text{eyes}_t}\right)\right) - \dot{q}_c\right)$$

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

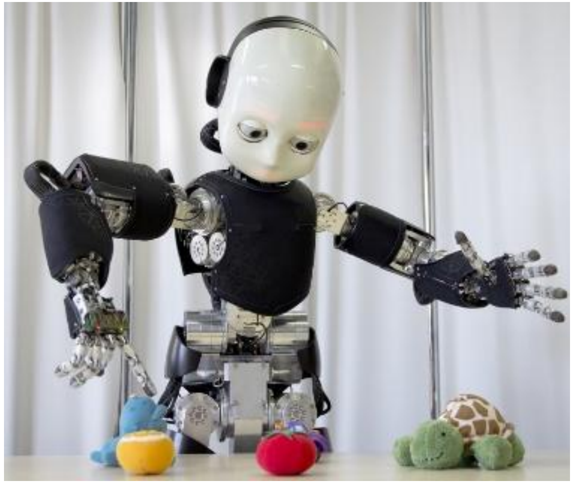
Wrap-up

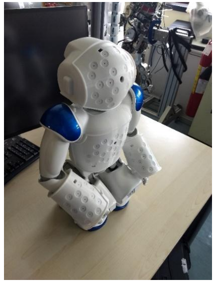
- Kinematic redundancy
- Singular configurations
- Manipulability
- Self motion / Null space
- Inverse kinematics
    - Analytical / closed-form
    - Iterative - Jacobian (pseudo)inverse
    - Optimization without Jacobian inverse

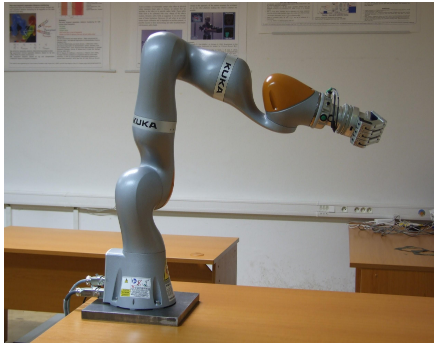
Next week

- Demo time at E210. Please come to the lecture E301 as usual (12:45) and the TAs will pick you up here.
- Labs will take place as usual at E132.

Robots - humanoids and cobots

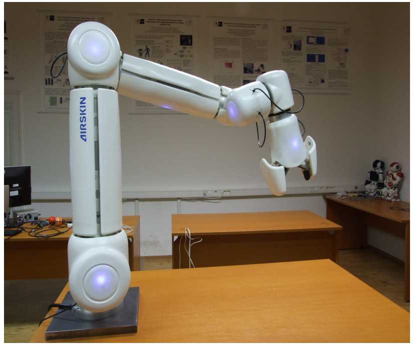






KUKA LBR iiwa

Kinova Gen3



UR10 + Airskin

Robot hands and grippers

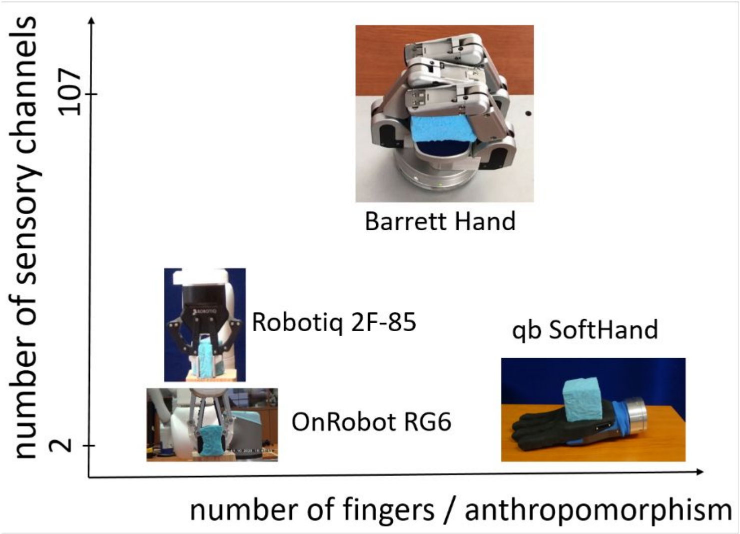
Anthropomorphic hands

Barrett Hand
(96 tactile + 3 fingertip joint torque + 8 joint pos. sensors)

Qb SoftHand (1 motor with position and current sensor)

iCub hand

Industrial parallel jaw 2-finger grippers
Robotiq 2F-85, OnRobot RG6



YCB object and model set

deformable objects set

Resources

- Books
  - Sections 2.4.2 - 2.7.3 in Nenchev, D. N., Konno, A., & Tsujita, T. (2018). Humanoid robots: Modeling and control. Butterworth-Heinemann.
- Online resources
  - https://www.slideserve.com/antonia/inverting-the-jacobian-and-manipulability
  - https://modernrobotics.northwestern.edu/nu-gm-book-resource/5-3-singularities/
- Articles
  - 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.