UNIVERSITÉ Concordia UNIVERSITY

Robotic Manipulator II: Control

ENGR 7401 – Direct and Inverse Kinematics

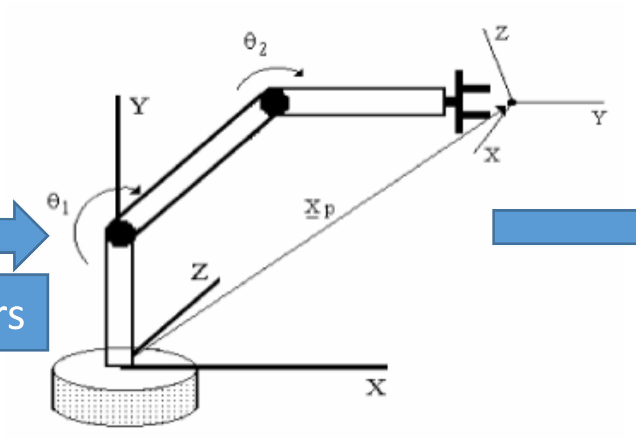
(Lecture 2)

Prof. Walter Lucia

Previous Lecture Main Concepts

Direct Chain (Recap)

Forces and Torques applied to the Joints
u(t)



Actuators

Dynamic Model

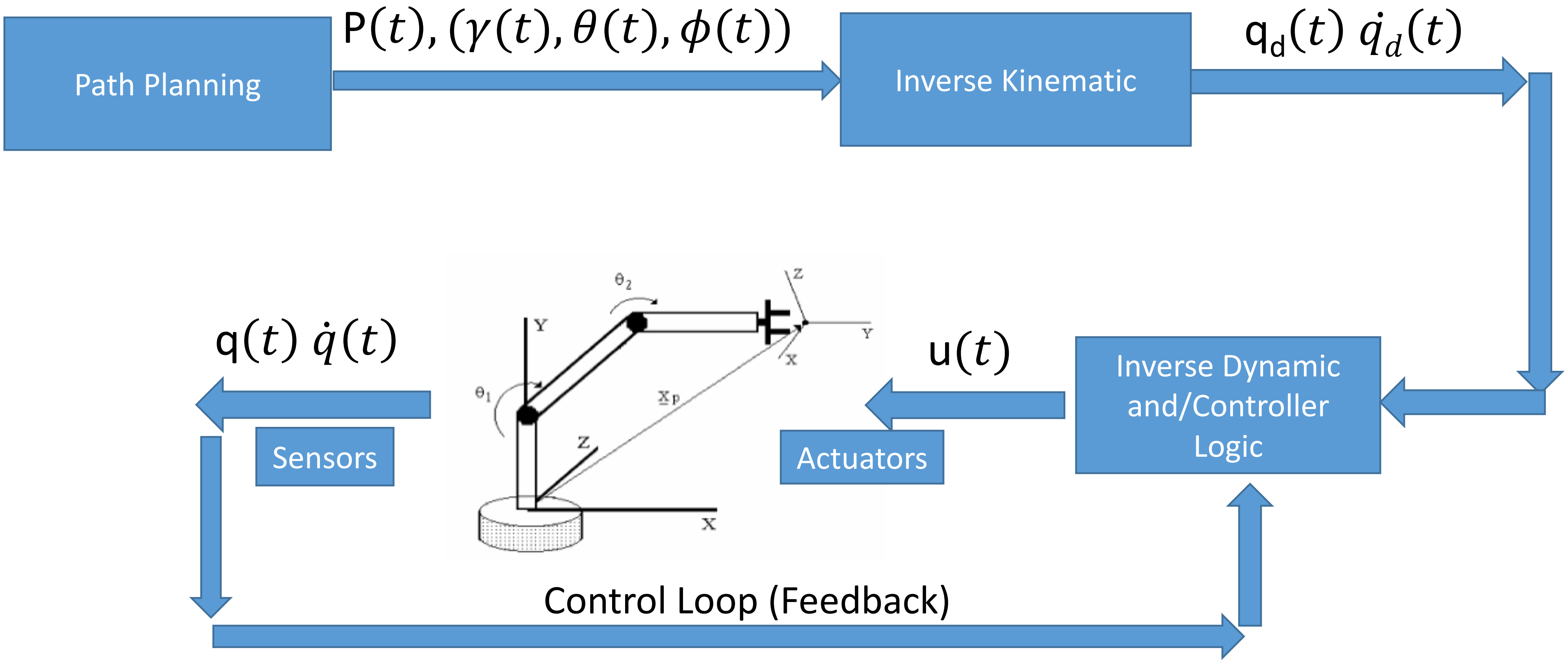
Joints variables
q(t) $\dot{q}(t)$ $\ddot{q}(t)$

Sensors

Kinematic Model

End Effector Position and Orientation
P, (γ, θ, ϕ)

Inverse Chain and Control (Recap)



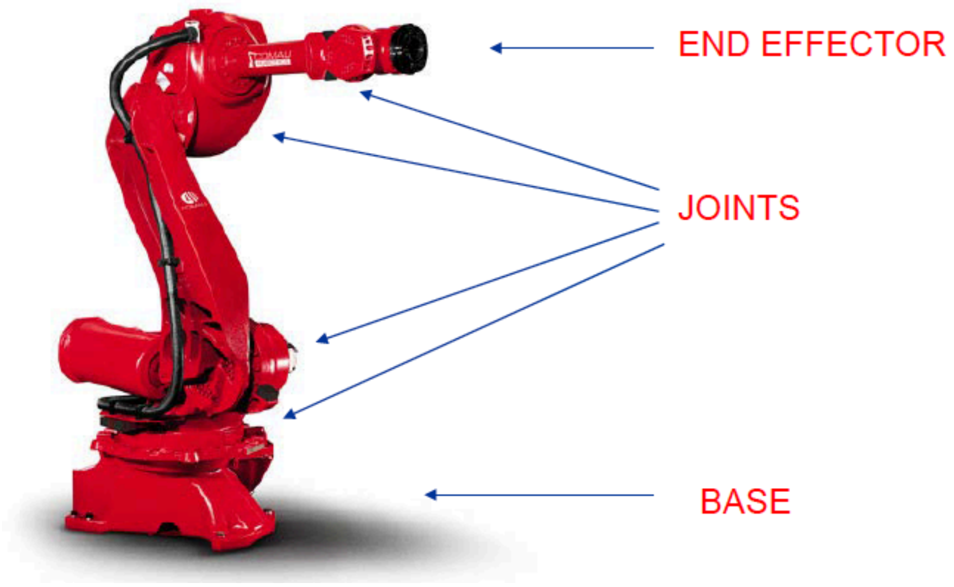
Lecture Outline

- Direct Kinematics
- Denavit –Hartenberg Convention
- Inverse Kinematics Problem
- Robotic Toolbox: Introduction

Direct Kinematics Problem

Direct Kinematics

- The all robot structure is called **kinematic chain**
  - One hand is constrained to a base
  - An **end-effector** is connected to the other end
- The kinematic chains is said ***open*** when there is only 1 sequence of link connecting the two ends
- <u>The aim of direct kinematics is to compute the pose of the end-effector as a function of the joint variables</u>



Recap From the Previous Lecture

Pose of a Rigid Body

- A rigid body is completely described in space by its position and orientation (in brief pose) with respect to a reference frame.

Unit Vectors

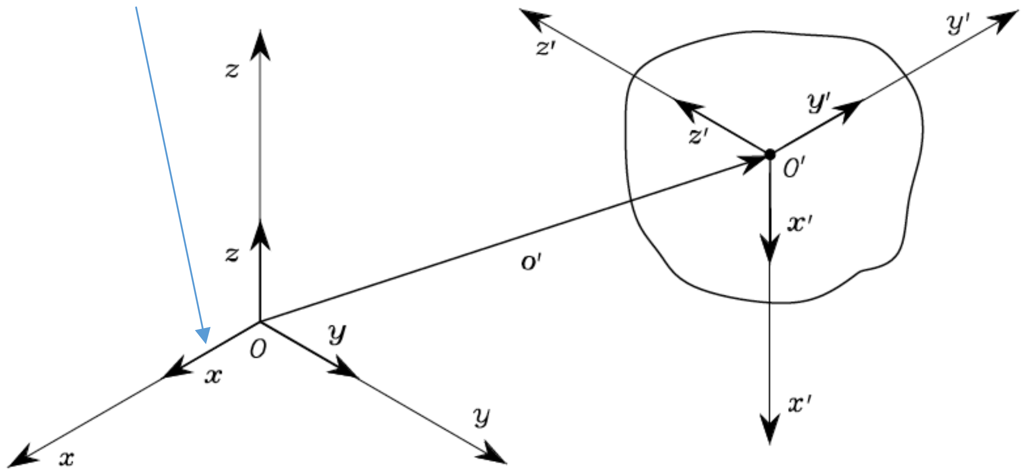


Fig. 2.1. Position and orientation of a rigid body

Position
$$\boldsymbol{o}' = \begin{bmatrix} o'_x \\ o'_y \\ o'_z \end{bmatrix}$$

Orientation
$$\begin{aligned} \boldsymbol{x}' &= x'_x \boldsymbol{x} + x'_y \boldsymbol{y} + x'_z \boldsymbol{z} \\ \boldsymbol{y}' &= y'_x \boldsymbol{x} + y'_y \boldsymbol{y} + y'_z \boldsymbol{z} \\ \boldsymbol{z}' &= z'_x \boldsymbol{x} + z'_y \boldsymbol{y} + z'_z \boldsymbol{z} \end{aligned}$$

- The components of each unit vector are the direction cosines of the axes of frame O – x’y’z’ w.r.t the reference frame O –xyz.

Rotation Matrix

$$\boldsymbol{x}' = x'_x \boldsymbol{x} + x'_y \boldsymbol{y} + x'_z \boldsymbol{z}$$
$$\boldsymbol{y}' = y'_x \boldsymbol{x} + y'_y \boldsymbol{y} + y'_z \boldsymbol{z}$$
$$\boldsymbol{z}' = z'_x \boldsymbol{x} + z'_y \boldsymbol{y} + z'_z \boldsymbol{z}$$

- By adopting a compact matrix notation, the three unit vectors can be combined in a (3x3) matrix called **Rotation Matrix**

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{x}' & \boldsymbol{y}' & \boldsymbol{z}' \end{bmatrix} = \begin{bmatrix} x'_x & y'_x & z'_x \\ x'_y & y'_y & z'_y \\ x'_z & y'_z & z'_z \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}'^T \boldsymbol{x} & \boldsymbol{y}'^T \boldsymbol{x} & \boldsymbol{z}'^T \boldsymbol{x} \\ \boldsymbol{x}'^T \boldsymbol{y} & \boldsymbol{y}'^T \boldsymbol{y} & \boldsymbol{z}'^T \boldsymbol{y} \\ \boldsymbol{x}'^T \boldsymbol{z} & \boldsymbol{y}'^T \boldsymbol{z} & \boldsymbol{z}'^T \boldsymbol{z} \end{bmatrix},$$

Rotation Matrices Interpretations

1

It describes the mutual orientation between two coordinate frames;

2

coordinate transformation between the coordinates two different frames (with common origin).

3

rotation of a vector in the same coordinate frame

Rigid Body Pose in 3D

- If we have two frames with different origins

$$\boldsymbol{p}^0 = \boldsymbol{o}_1^0 + \boldsymbol{R}_1^0 \boldsymbol{p}^1$$

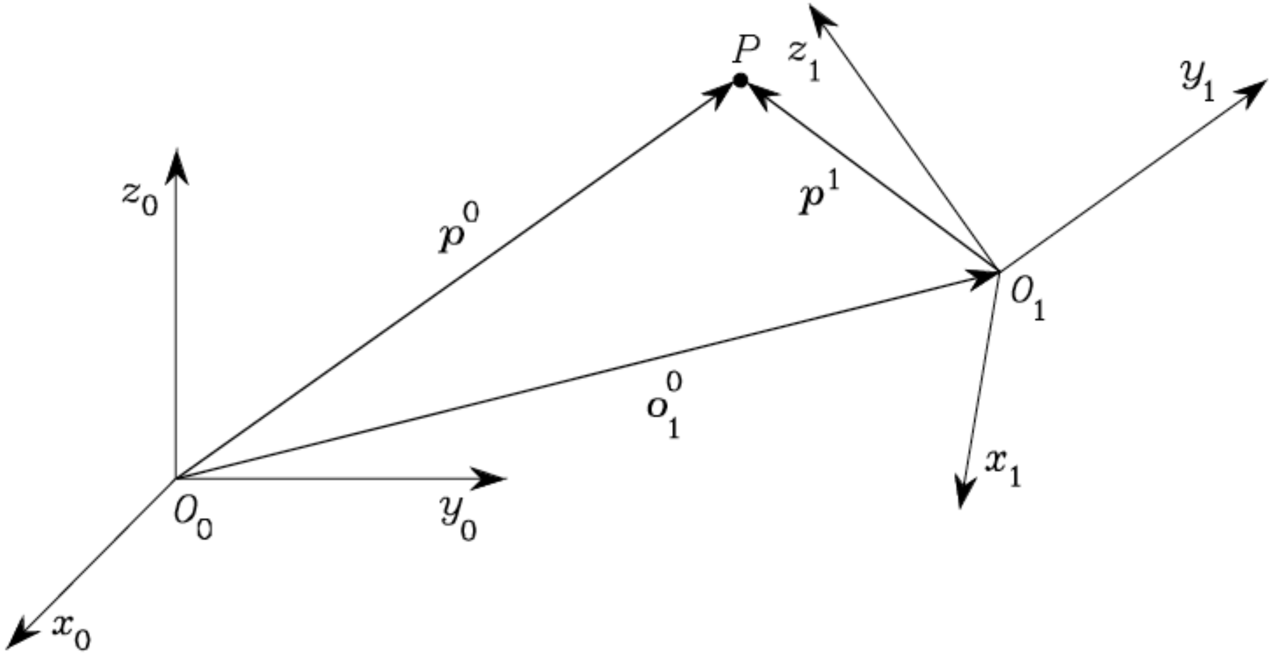


Fig. 2.11. Representation of a point P in different coordinate frames

Homogeneous Transformations

- A compact representation of the relationship between the coordinates of the same point in two different frames, is given by the homogeneous transformation matrix (adding a 4th unit component)

$$\widetilde{p} = \begin{bmatrix} p \\ 1 \end{bmatrix} \qquad A_1^0 = \begin{bmatrix} R_1^0 & o_1^0 \\ 0^T & 1 \end{bmatrix}$$

- Therefore,

$$\widetilde{p}^0 = A_1^0 \widetilde{p}^1$$

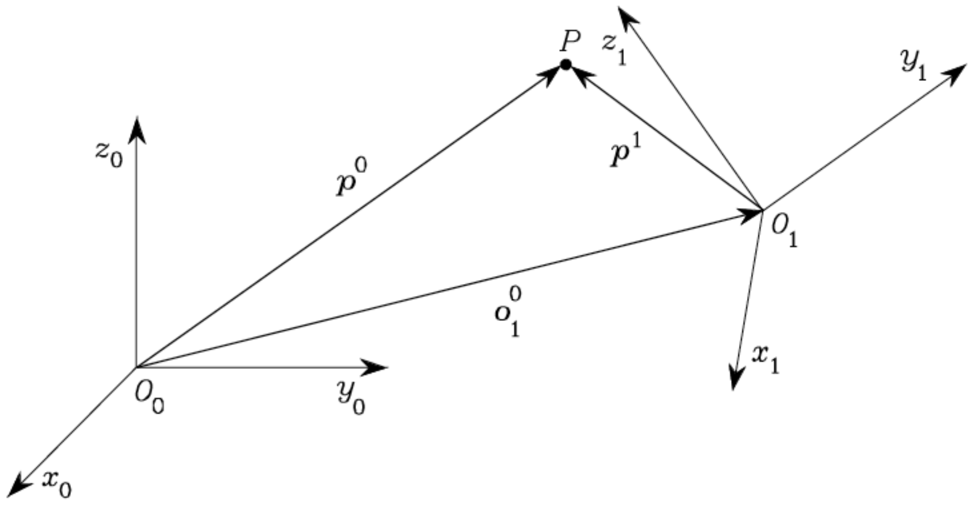


Fig. 2.11. Representation of a point P in different coordinate frames

End-effector (tool): frame choice

- The tool frame is usually chosen such that

- $\mathbf{a_e}$ (approach): unit vector in the direction of the work-piece
- $\mathbf{s_e}$ (sliding): orthogonal to $\mathbf{a_e}$ in the sliding plane of the gripper
- $\mathbf{n_e}$ (normal): orthogonal to both other vectors and such that the frame is right-handed
- $\mathbf{p_e}$ ($\mathbf{o_e}$): central point of the gripper

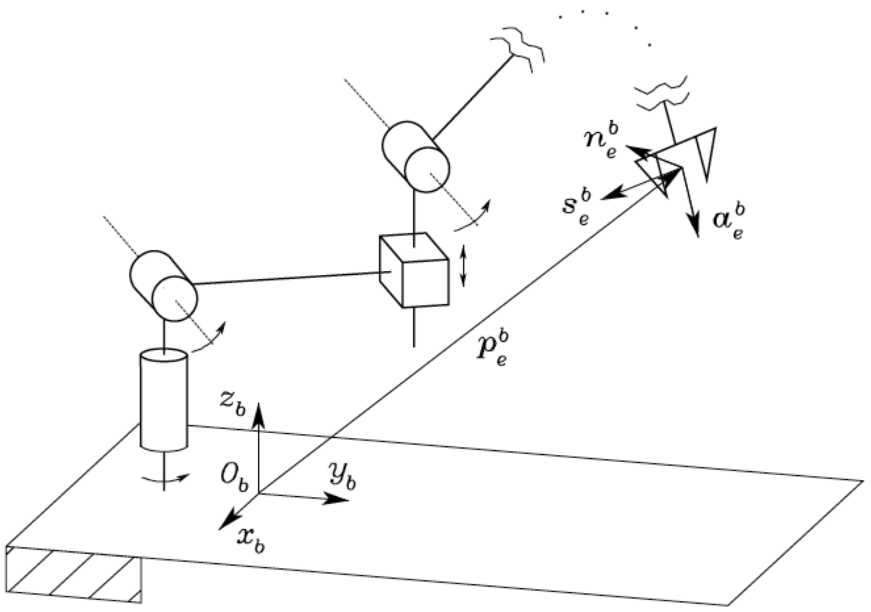


Fig. 2.13. Description of the position and orientation of the end-effector frame

Pose of the End-Effector of a Robot

- The pose of a body is described by the position of the origin and by the unit vectors of a frame attached to the body. Such information describe an homogenous transformation matrix

Rotation Matrix

Central Point in the base frame

$$\boldsymbol{T}_e^b(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{n}_e^b(\boldsymbol{q}) & \boldsymbol{s}_e^b(\boldsymbol{q}) & \boldsymbol{a}_e^b(\boldsymbol{q}) & \boldsymbol{p}_e^b(\boldsymbol{q}) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **q** is the vector of joint variables (each component represents either a rotation or a translation)
- **b**=base frame, **e**=end-effector frame

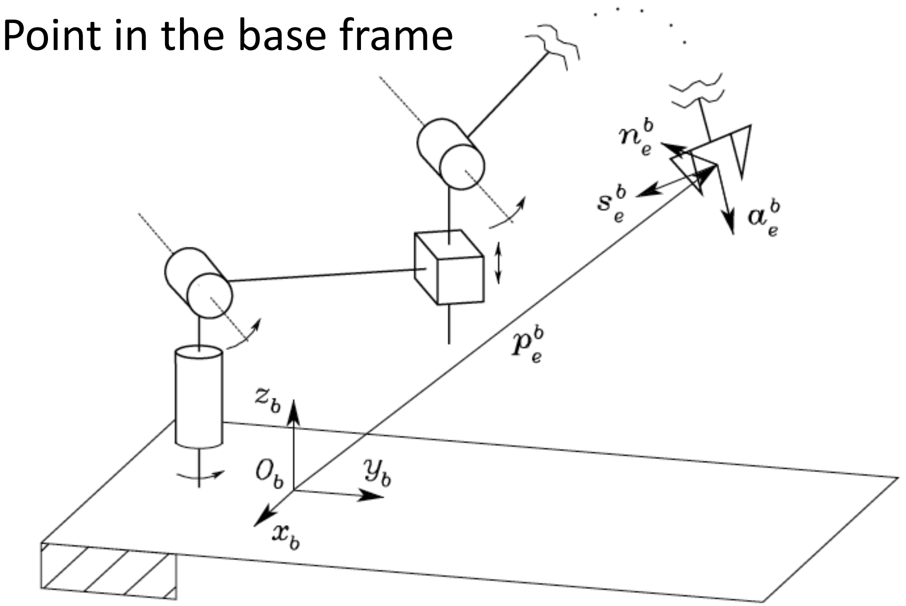


Fig. 2.13. Description of the position and orientation of the end-effector frame

How to compute direct kinematics

- Geometric analysis of the structure of the manipulator

$$T_e^b(q) = \begin{bmatrix} n_e^b & s_e^b & a_e^b & p_e^b \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & s_{12} & c_{12} & a_1c_1 + a_2c_{12} \\ 0 & -c_{12} & s_{12} & a_1s_1 + a_2s_{12} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

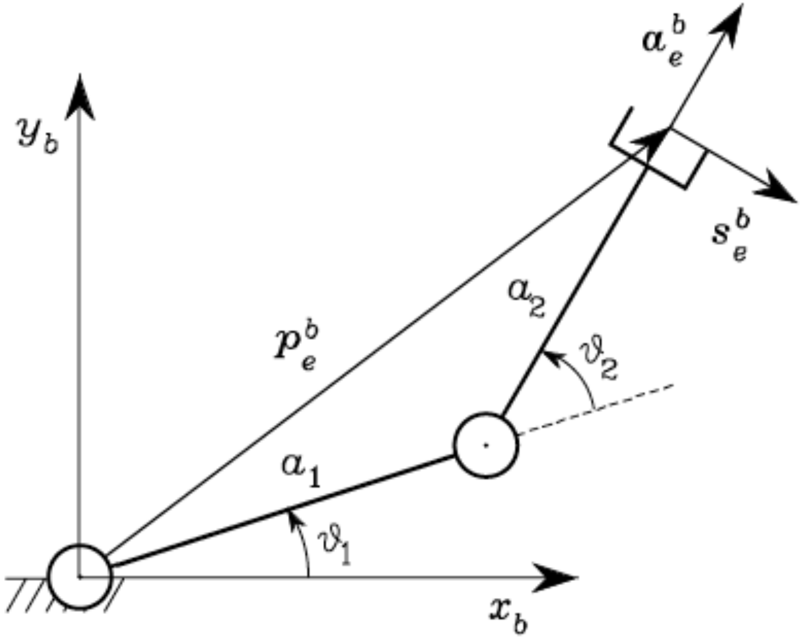


Fig. 2.14. Two-link planar arm

- Whenever the manipulator structure is complex and the number of joints increases, it is preferable to adopt a less direct solution, which, though, is based on a systematic, general procedure.

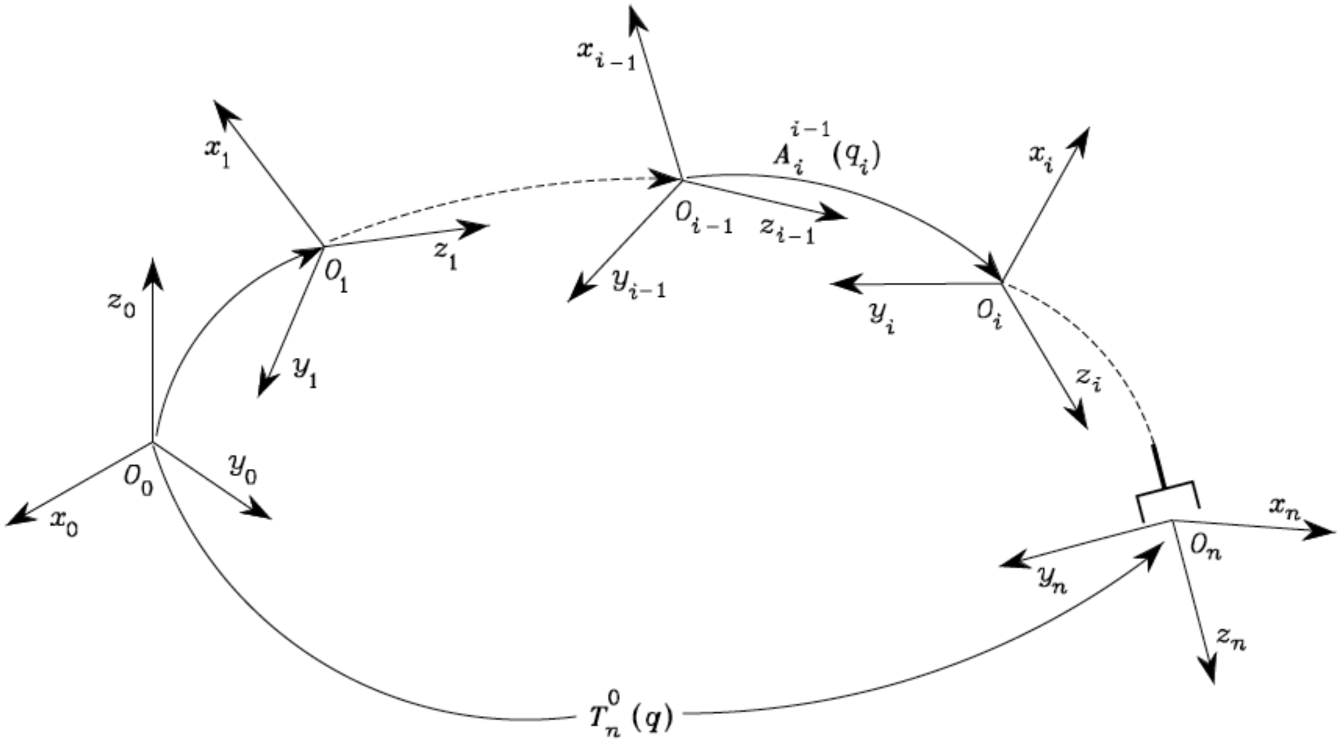
Computation of Complex direct Kinematics

- First consider the description of kinematic relationship between consecutive links
- Then obtain the overall description in a recursive fashion

$$T_n^0(q) = A_1^0(q_1)A_2^1(q_2)\dots A_n^{n-1}(q_n)$$

- Then add the end-effector and base frames (usually constant transformations)

$$T_e^b(q) = T_0^b T_n^0(q) T_e^n$$



Denavit –Hartenberg Convention

- It defines a systematic, general algorithm to compute the direct kinematics from *i* to *i-1* (A_i^{i-1})
  - It defines rules to determine frames attached to the robot links. They can be arbitrarily chosen but it is good to set some rules

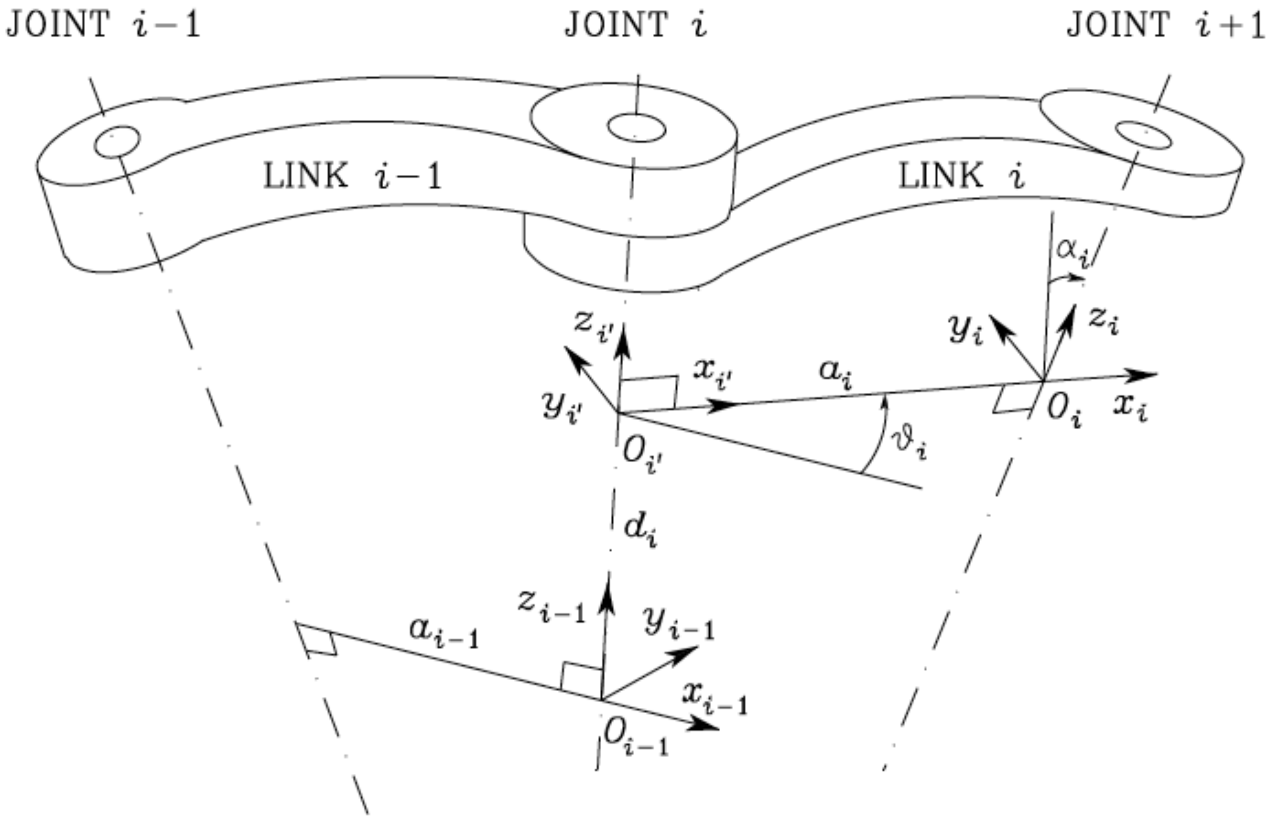


Fig. 2.16. Denavit–Hartenberg kinematic parameters

Denavit –Hartenberg Convention

- The coordinate transformation from *i* to *i-1* is the following

$$A_i^{i-1}(q_i) = A_{i'}^{i-1} A_i^{i'} = \begin{bmatrix} c_{\vartheta_i} & -s_{\vartheta_i} c_{\alpha_i} & s_{\vartheta_i} s_{\alpha_i} & a_i c_{\vartheta_i} \\ s_{\vartheta_i} & c_{\vartheta_i} c_{\alpha_i} & -c_{\vartheta_i} s_{\alpha_i} & a_i s_{\vartheta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- The transformation matrix is function of the joint variable q_i that is ϑ_i (if revolute joint) or d_i (if prismatic joint)
- 4 parameters are needed (look next video to understand how are they defined)

$$a_i, d_i, \alpha_i, \vartheta_i$$

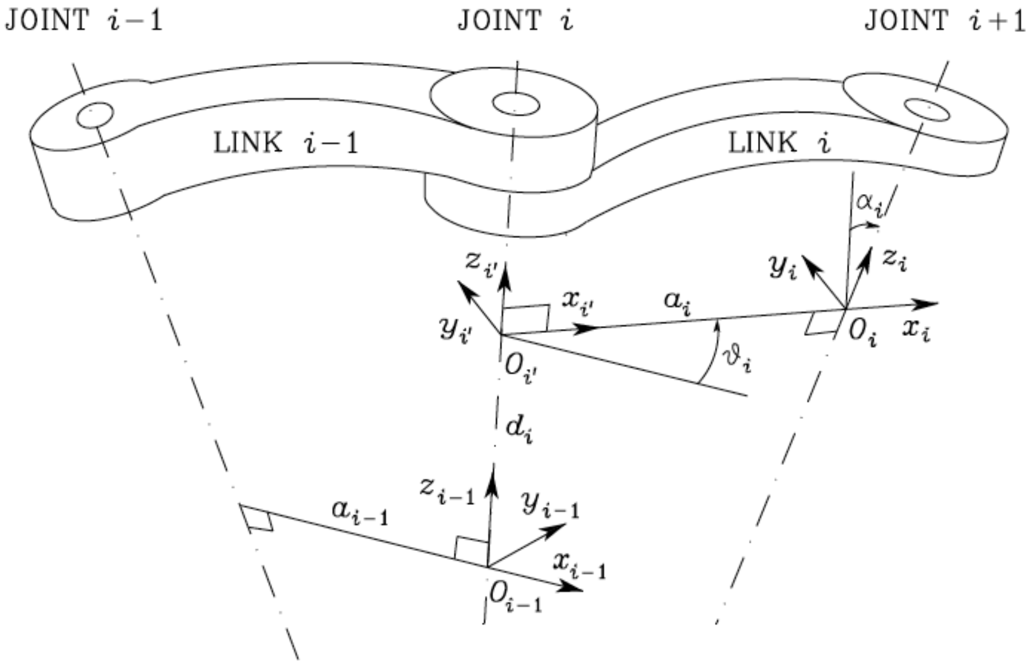


Fig. 2.16. Denavit–Hartenberg kinematic parameters

Denavit –Hartenberg: parameters computation

https://www.youtube.com/watch?v=rA9tm0gTln8

Kinematics of Typical Manipulator Structures

- Three-link Planar Arm

Table 2.1. DH parameters for the three-link planar arm

Link	a_i	α_i	d_i	ϑ_i
1	a_1	0	0	ϑ_1
2	a_2	0	0	ϑ_2
3	a_3	0	0	ϑ_3

$$\boldsymbol{T}_3^0(\boldsymbol{q}) = \boldsymbol{A}_1^0\boldsymbol{A}_2^1\boldsymbol{A}_3^2 = \begin{bmatrix} c_{123} & -s_{123} & 0 & a_1c_1 + a_2c_{12} + a_3c_{123} \\ s_{123} & c_{123} & 0 & a_1s_1 + a_2s_{12} + a_3s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

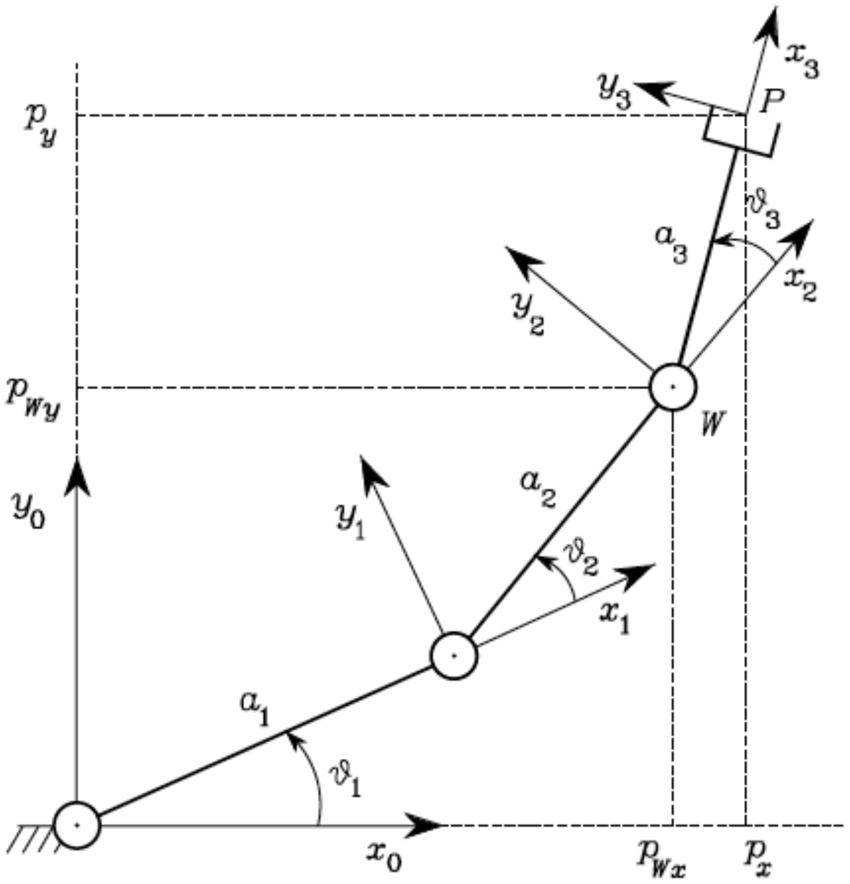


Fig. 2.20. Three-link planar arm

Kinematics of Typical Manipulator Structures

- Anthropomorphic Arm

Table 2.4. DH parameters for the anthropomorphic arm

Link	a_i	α_i	d_i	ϑ_i
1	0	$\pi/2$	0	ϑ_1
2	a_2	0	0	ϑ_2
3	a_3	0	0	ϑ_3

$$\boldsymbol{T}_3^0(\boldsymbol{q}) = \boldsymbol{A}_1^0 \boldsymbol{A}_2^1 \boldsymbol{A}_3^2 = \begin{bmatrix} c_1 c_{23} & -c_1 s_{23} & s_1 & c_1(a_2 c_2 + a_3 c_{23}) \\ s_1 c_{23} & -s_1 s_{23} & -c_1 & s_1(a_2 c_2 + a_3 c_{23}) \\ s_{23} & c_{23} & 0 & a_2 s_2 + a_3 s_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

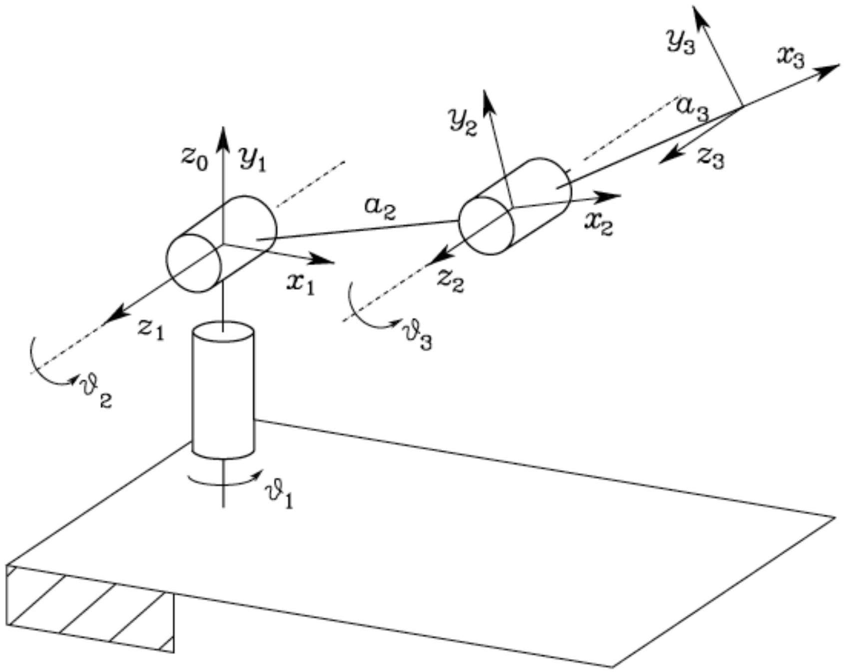


Fig. 2.23. Anthropomorphic arm

Joint Space and Operational Space

- The joint space is defined by the vector of joint variables

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix} \qquad \begin{array}{l} q_i = \vartheta_i \text{ (rotating joint)} \\ \\ q_i = d_i \text{ (prismatic joint)} \end{array}$$

- The operational space is the space where the manipulator task has to be accomplished. It is defined by the posture **x**

$$\underbrace{\boldsymbol{x}}_{m \text{ components}} = \begin{bmatrix} \boldsymbol{p} \\ \phi \end{bmatrix} \qquad \begin{array}{l} \boldsymbol{p} \text{ (position)} \\ \phi \text{ (minimal representation of the orientation)} \end{array}$$

Direct Kinematic relation end-effector

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p} \\ \phi \end{bmatrix}$$

$\boldsymbol{p}$ (position)
ϕ (minimal representation of the orientation)

m components

$$\boldsymbol{x}_e = \boldsymbol{k}(\boldsymbol{q}).$$

- The vector function **k** is usually a nonlinear function of the joint variables.
- The relation between the joint variables and the end-effector position is usually simple (the **p** compoents of $\mathbf{x_e}$). The relation with the minimal orientation cannot be expressed in closed form
  1. First, we need to compute the element of the rotation matrix $\boldsymbol{n}_e(\boldsymbol{q}),\ \boldsymbol{s}_e(\boldsymbol{q}),\ \boldsymbol{a}_e(\boldsymbol{q})$
  2. Then, we determine the Euler angles (the orientation) from the rotation matrix

Example: Three Link Planar Arm

- The geometry suggests that the end-effector position is determined by the two coordinates $p_x$and p_x , while the orientation by the angle θ

$$\boldsymbol{x}_e = \begin{bmatrix} p_x \\ p_y \\ \phi \end{bmatrix} = \boldsymbol{k}(\boldsymbol{q}) = \begin{bmatrix} a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ \vartheta_1 + \vartheta_2 + \vartheta_3 \end{bmatrix}$$

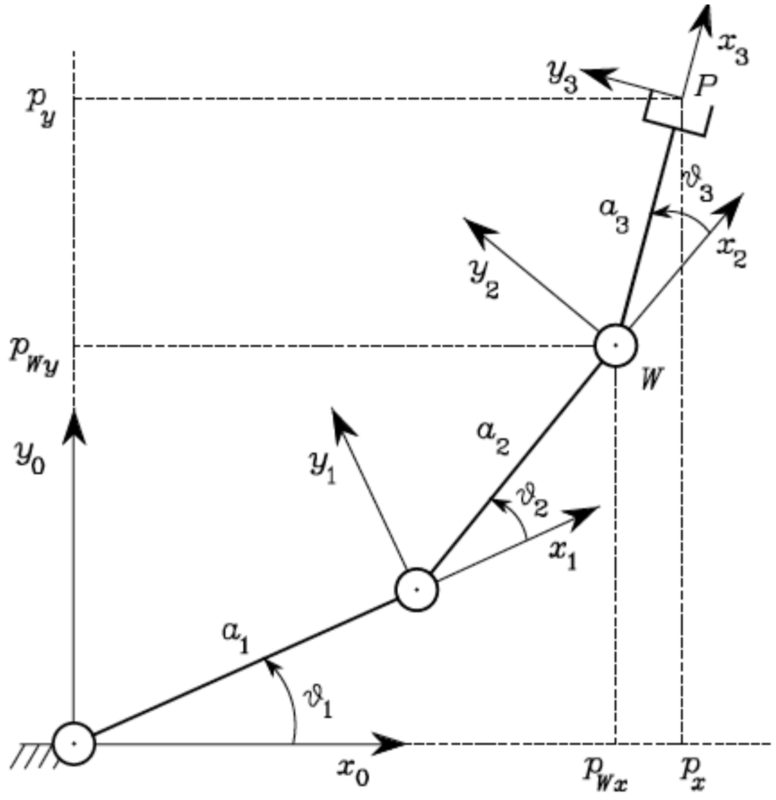


Fig. 2.20. Three-link planar arm

Kinematic Redundancy

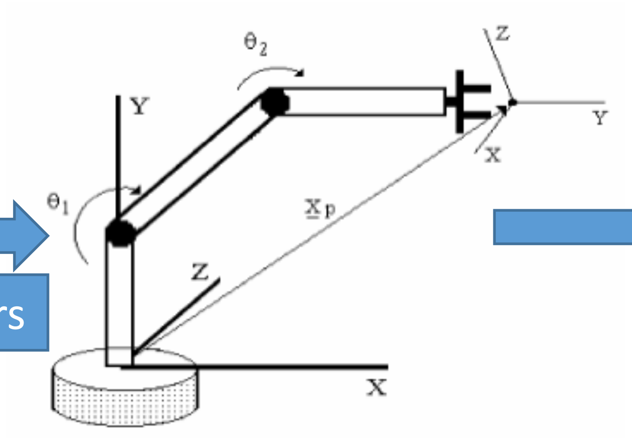
- A manipulator is termed kinematically redundant when it has a number of DOFs which is greater than the number of variables that are necessary to describe a given task
  - A manipulator is intrinsically redundant when the dimension of the operational space is smaller than the dimension of the joint space ($m < n$).
- However, redundancy is, anyhow, a concept relative to the task assigned to the manipulator (r (components of interest) $<m$)
  - E.g. The three-DOF planar arm:
    - If only the end-effector position is specified, then the structure is redundant ($n = m = 3$, $r = 2$);
    - If also the end-effector orientation is specified ($n = m = r = 3$), then the structure is not anymore redundant.
    - A four-DOF planar arm is intrinsically redundant ($n = 4$, $m = 3$)

Inverse Kinematics Problem

Direct Chain (Recap)

Forces and Torques applied to the Joints
$u(t)$

Actuators



Dynamic Model

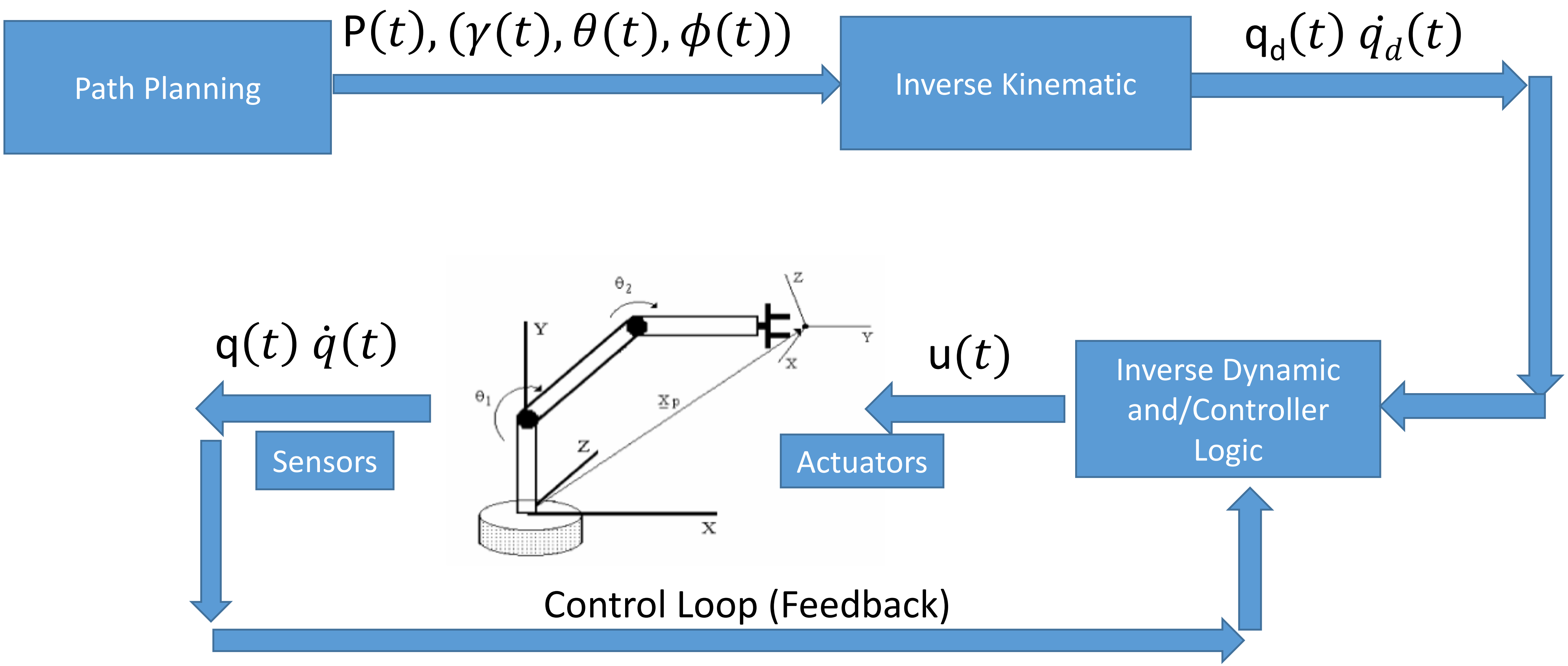
Joints variables
$q(t)\ \dot{q}(t)\ \ddot{q}(t)$

Sensors

Kinematic Model

End Effector Position and Orientation
$P, (\gamma, \theta, \phi)$

Inverse Chain and Control (Recap)



Inverse Kinematics Problem

- Given position and orientation of the end-effector, find the corresponding joint variables

$$\boldsymbol{x} \quad \Rightarrow \quad \boldsymbol{q}$$

- The solution to this problem is of fundamental importance to transform the motion specifications (in the operational space), into the corresponding joint space motions that allow execution of the desired motion
- The direct kinematics equation are computed in a unique manner, once the joint variables are known . On the other hand, the inverse kinematics problem is way more complex

Inverse Kinematics Problem: Problematics

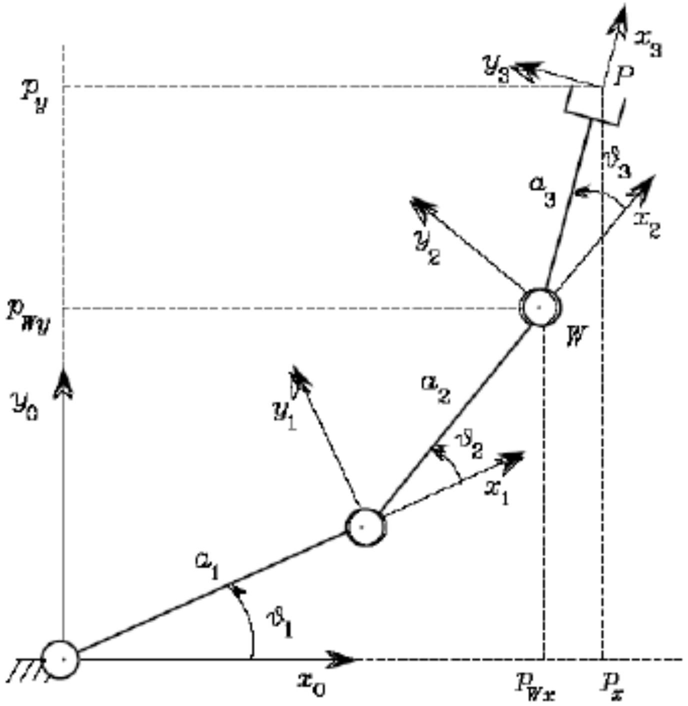
- The equations to solve are in general nonlinear, and thus it is not always possible to find a closed-form solution .
- Multiple solutions may exist.
- Infinite solutions may exist, e.g., in the case of a kinematically redundant manipulator.
- There might be no admissible solutions, in view of the manipulator kinematic structure.
- The existence of solutions is guaranteed only if the given end-effector position and orientation belong to the manipulator dexterous workspace (region that the end-effector can reach with different orientations)

Solution of the Inverse Kinematics Problem

- Closed Form Solution
  - Requires algebraic intuition to find those significant equations containing the unknowns
  - We can find all the admissible solutions (the entire set of solutions)
  - Exists for Standard and Simple robot Configurations

- Numerical Solution
  - In all those cases when it is difficult to find closed-form solutions.
  - Have the advantage of being applicable to any kinematic structure, but in general they do not allow computation of all admissible solutions (it will find one solution belonging to the set admissible solutions (which is not known))

3 D.O.F planar manipulator (Closed Form Solution)



Data: p_x, p_y, ϕ

Unknowns: $\vartheta_1, \vartheta_2, \vartheta_3$

Let:

$$p_{Wx} = p_x - a_3 c_\phi = a_1 c_1 + a_2 c_{12}$$

$$p_{Wy} = p_y - a_3 s_\phi = a_1 s_1 + a_2 s_{12}$$

Squaring and summing:

$$c_2 = \frac{p_{Wx}^2 + p_{Wy}^2 - a_1^2 - a_2^2}{2a_1 a_2} \quad \Rightarrow \quad \vartheta_2 = \mathrm{Atan}\,2(s_2, c_2)$$

$$s_2 = \pm\sqrt{1 - c_2^2}$$

2 solutions

Finally:

$$c_1 = \frac{(a_1 + a_2 c_2) p_{Wx} + a_2 s_2 p_{Wy}}{p_{Wx}^2 + p_{Wy}^2}$$

$$s_1 = \frac{(a_1 + a_2 c_2) p_{Wy} - a_2 s_2 p_{Wx}}{p_{Wx}^2 + p_{Wy}^2}$$

$$\Rightarrow \quad \vartheta_1 = \mathrm{Atan}\,2(s_1, c_1) \qquad \vartheta_3 = \phi - \vartheta_1 - \vartheta_2$$

Inverse Kinematic: Anthropomorphic Arm

- Shoulder–right/elbow–up, shoulder–left/elbow–up,
- shoulder–right/elbow–down, shoulder–left/elbow–down;
- Obviously, the orientation is different for the two pairs of solutions

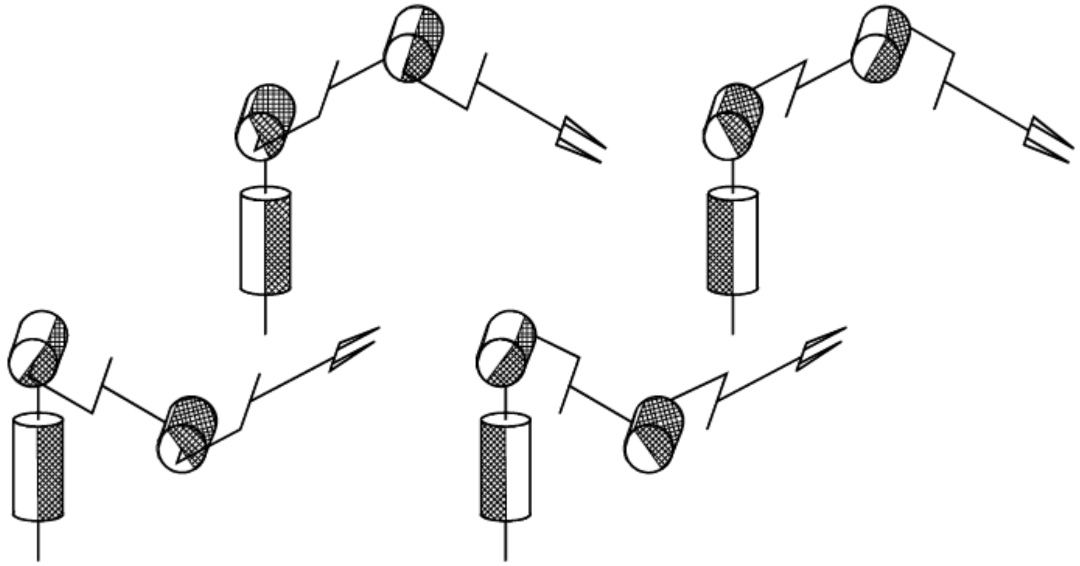


Fig. 2.33. The four configurations of an anthropomorphic arm compatible with a given wrist position

Robotic Toolbox

Robotic Toolbox: Overview

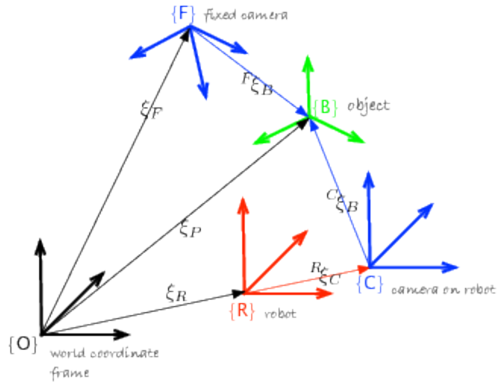
- Developed by Dr. Peter Corke (Professor at Queensland University of Technology (Australia) and co-director of the CyPhy lab)
- http://petercorke.com/Robotics_Toolbox.html
- The toolbox is very useful to understand/simulate robotic manipulators principles
  - Direct Kinematics
  - Inverse Kinematics
  - Trajectory Planning
  - Dynamics
  - Control

How to install the toolbox

- Download RTB-10.2 mltbx format (32.7 MB) in MATLAB toolbox format (.mltbx)
  - From within the MATLAB file browser double click on this file, it will install and configure the paths correctly
- Download RTB-10.2 zip format (31.9 MB) as a zip file (.zip)
  - Unpack the archive which will create the directory (folder) rvctools, and within that the directories robot, simulink, and common.
  - Adjust your MATLABPATH to include rvctools
  - Execute the startup file rvctools/startup_rvc.m and this will place the correct directories in your MATLAB path.
- The Toolbox is tested with MATLAB R2016b (It might have backward compability problem if the matlab version is very old)
- Run the demo *rtbdemo* to see what it can do
- Dr. Corke keeps updating the software often to fix bugs (please give a look at the website once in a while)

Position and Orientation

- The toolbox can represent position and orientation in 2-3 D dimensions.
- All the standard representations are full supported
  - Homogeneous Transformations
  - Euler Angles
  - Roll-Pitch-Yaw Angles
  - Etc…



- Some functions:
  - ***trplot2*** *(draws a 2D coordinate frame represented by the homogeneous transformation matrix T (3x3).)*
  - ***trplot*** *(draws a 3D coordinate frame represented by the homogeneous transformation T)*
  - ***rotx***, ***roty***, ***rotz***, ***trplot*** *(elementary rotations)*
  - ***tranimate(P1, P2, OPTIONS)*** *animates a 3D coordinate frame moving from pose P1 to pose P2. P1 and P2 are homogeneus transformations or ration matrices*

Position and Orientation: functions

- Euler Angles, some functions
  - **R = eul2r(PHI, THETA, PSI, OPTIONS)** is an SO(3) orthonormal rotation matrix (3x3) equivalent to the specified Euler angles
  - **EUL = tr2eul(T, OPTIONS)** are the ZYZ Euler angles (1x3) corresponding to the rotational part of a homogeneous transform T (4x4). The 3 angles EUL=[PHI,THETA,PSI] correspond to sequential rotations about the Z, Y and Z axes respectively.

Homogeneous Transformations

- Some functions:
  - ***T = transl(X, Y, Z)*** *is an SE(3) homogeneous transform (4x4) representing a pure translation of X, Y and Z.*
  - ***T = trotx(THETA)*** *is a homogeneous transformation (4x4) representing a rotation of THETA radians about the x-axis. trplot*
  - ***T = eul2tr(PHI, THETA, PSI, OPTIONS)*** *is an SE(3) homogeneous transformation matrix (4x4) with zero translation and rotation equivalent to the specified Euler angles.*
  - ***T = angvec2tr(THETA, V)*** *is a homogeneous transform matrix (4x4) equivalent to a rotation of THETA about the vector V.*

THANK YOU!