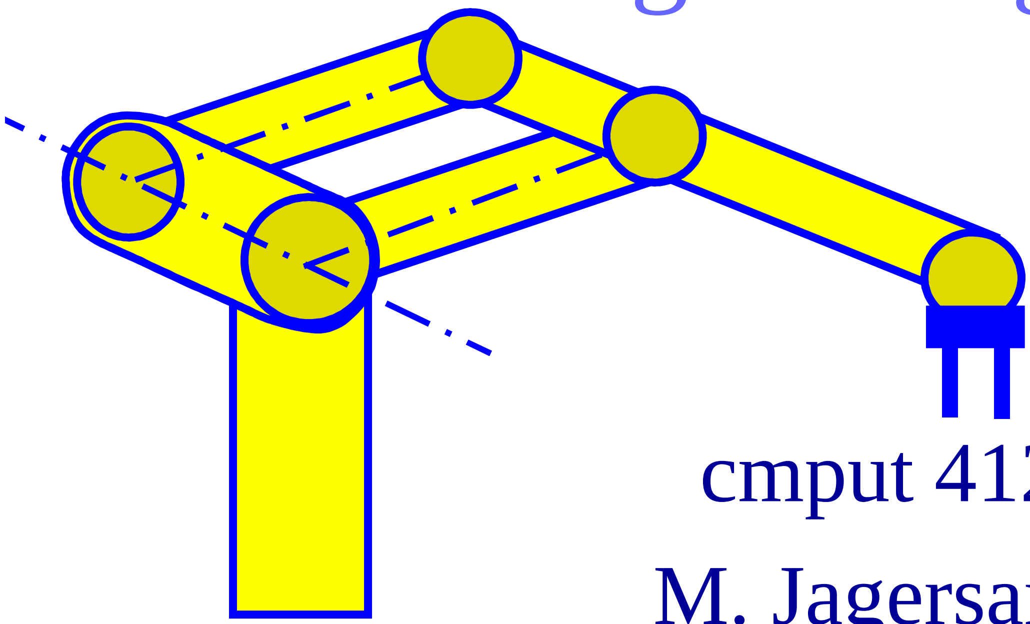
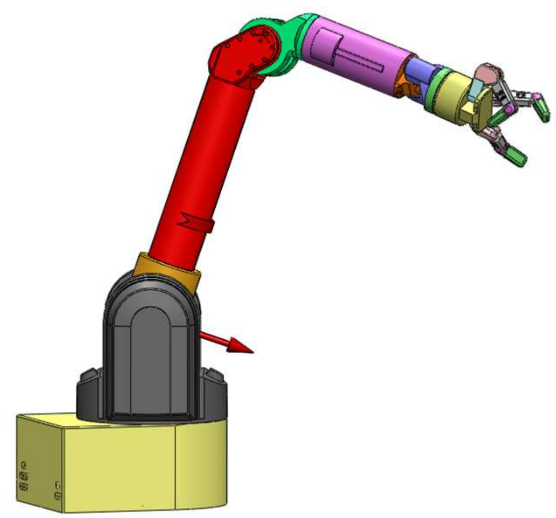
Robot Kinematics and linkage configurations 2



cmput 412

M. Jagersand

With slides from A. Shademan
and A Casals



Review Kinematics

Forward Kinematics (angles to position)

What you are given:	The length of each link The angle of each joint
What you can find:	The position of any point (i.e. it's (x, y, z) coordinates

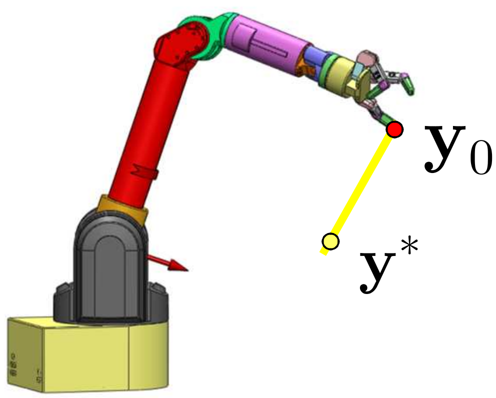
Inverse Kinematics (position to angles)

What you are given:	The length of each link The position of some point on the robot
What you can find:	The angles of each joint needed to obtain that position

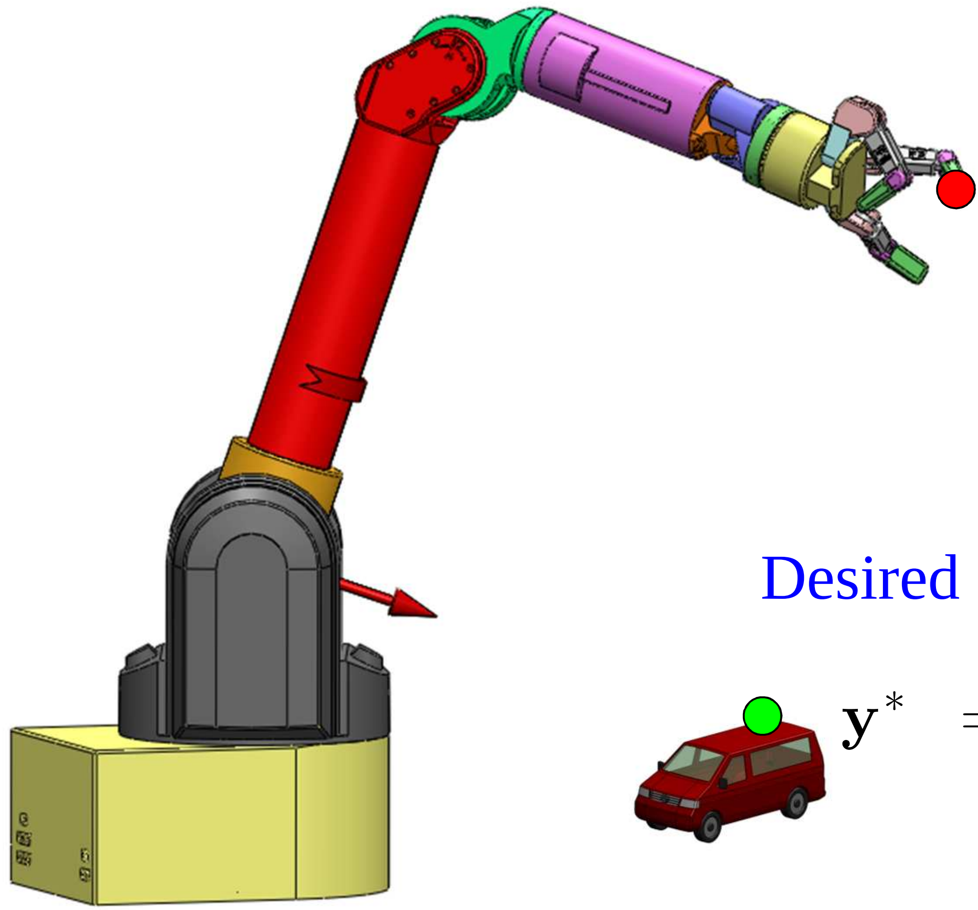
Numerical Inverse Kinematics

- Cartesian Location $\mathbf{y} = [x, y, z]^T \;=\; f(\mathrm{x})$
- Motor joint angles: $\mathbf{x} = [\, x_1, \quad x_2, \ldots \quad x_n \,]^T$
- Local linear model: $\Delta\mathbf{y} = \mathbf{J}(\mathbf{x})\Delta\mathbf{x}$
- Numerical steps: 1 Solve: $\mathbf{y}^* - \mathbf{y}_k = \mathbf{J}\Delta\mathbf{x}$

  2 Update: $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}$



Cartesian Trajectory Motion



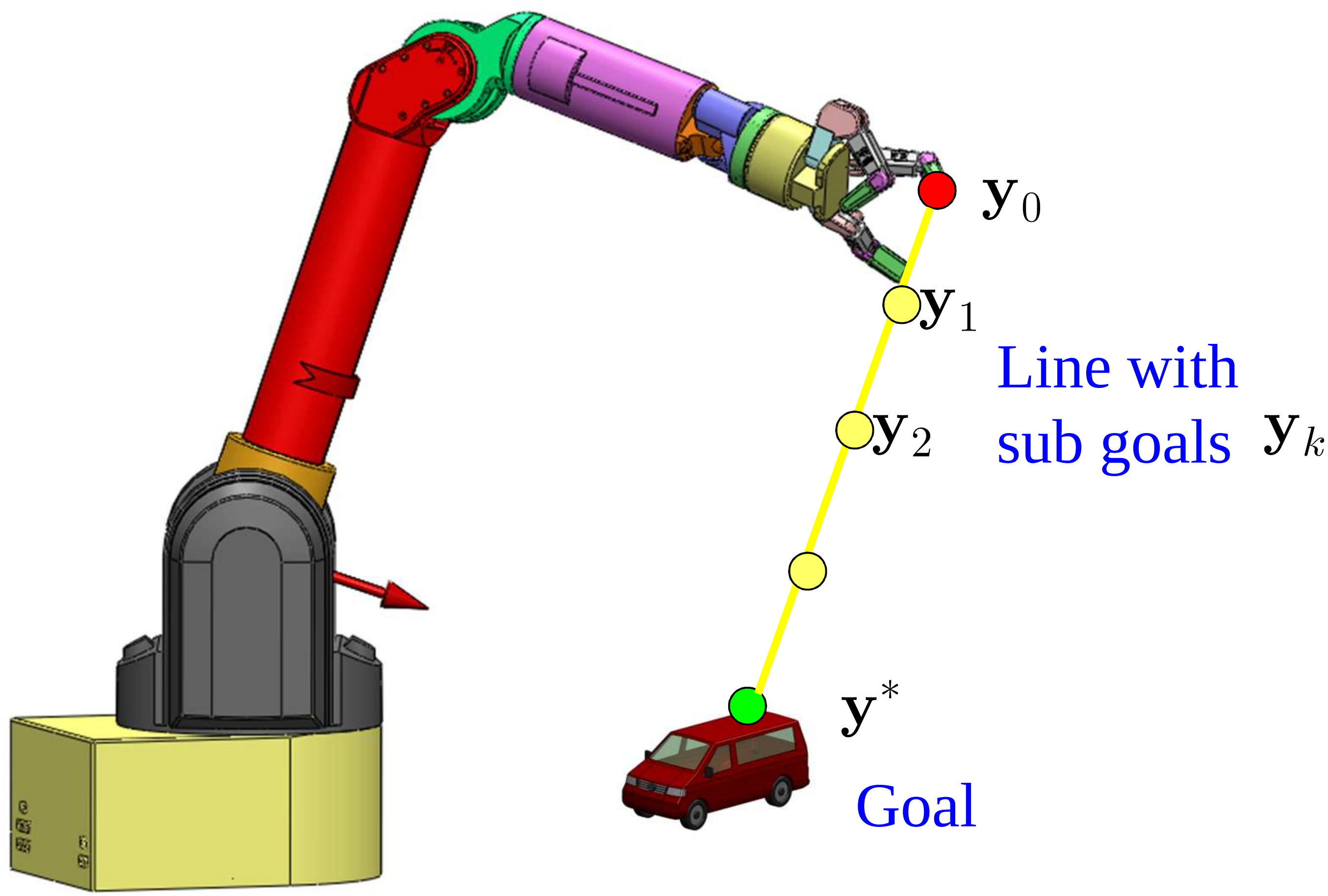
Current
Cartesian
position

$\mathbf{y}_0$

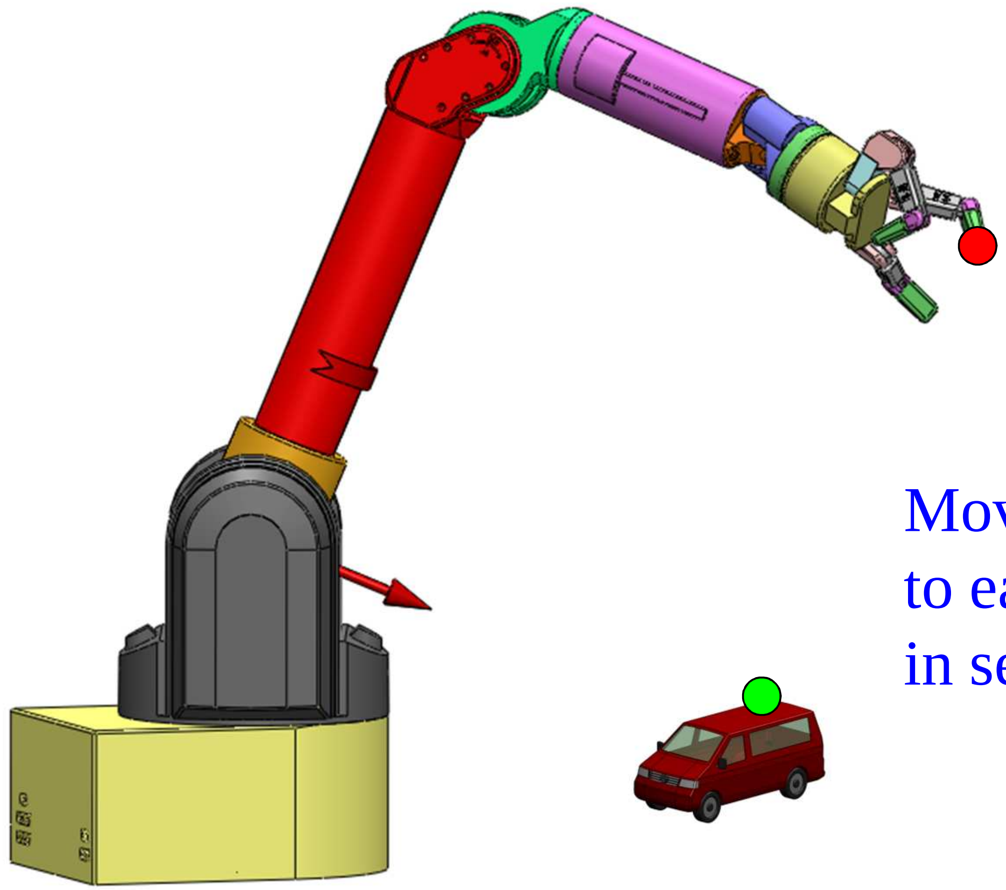
Desired goal:

$$\mathbf{y}^* = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Cartesian Trajectory Motion

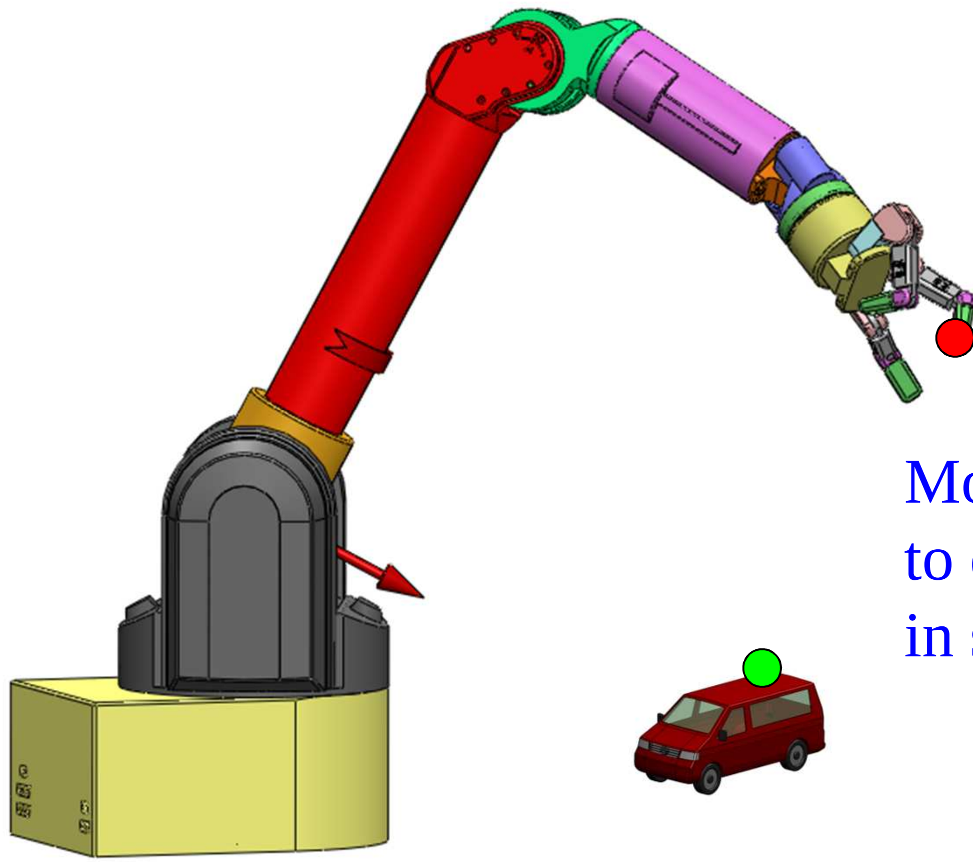


Cartesian Trajectory Motion



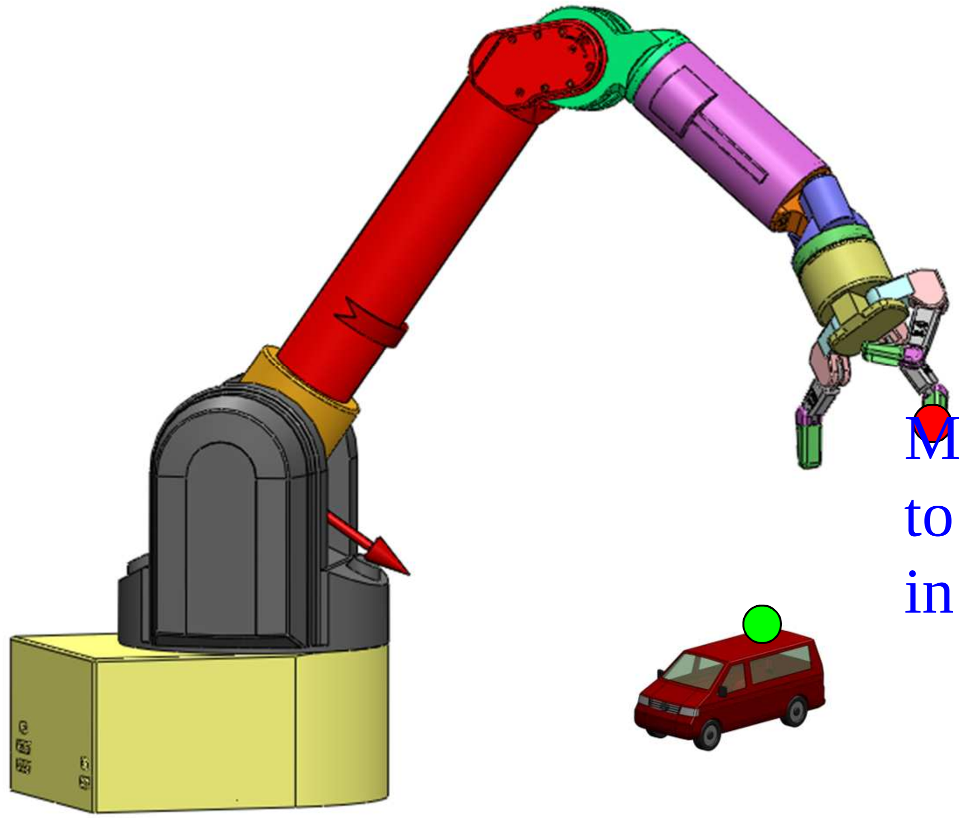
Move the robot
to each subgoal
in sequence

Cartesian Trajectory Motion



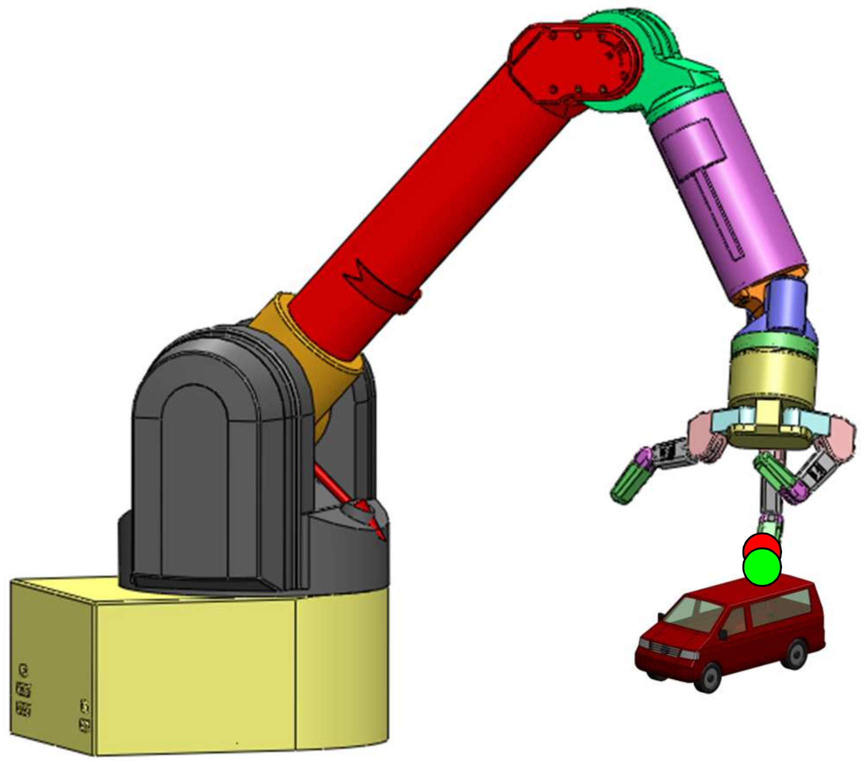
Move the robot
to each subgoal
in sequence

Cartesian Trajectory Motion



Move the robot
to each subgoal
in sequence

Cartesian Trajectory Motion



Iterate until convergence at final goal

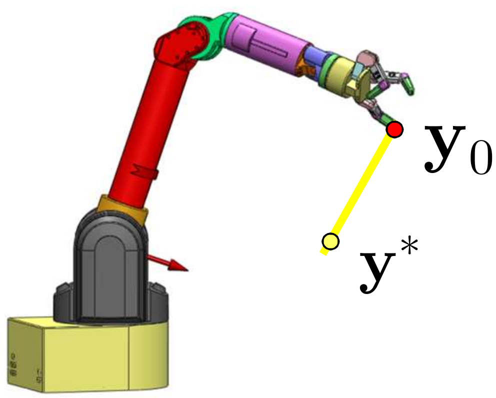
Numerical Inverse Kinematics

How do we find the Jacobian J(x)?

- Cartesian Location $\mathbf{y} = [x, y, z]^T \;=\; f(\mathrm{x})$
- Motor joint angles: $\mathbf{x} = [\, x_1, \quad x_2, \ldots \quad x_n\,]^T$
- Local linear model: $\Delta \mathbf{y} = \mathbf{J}(\mathbf{x})\Delta \mathbf{x}$
- Visual servoing steps:

1 Solve: $\mathbf{y}^* - \mathbf{y}_k = \mathbf{J}\Delta \mathbf{x}$

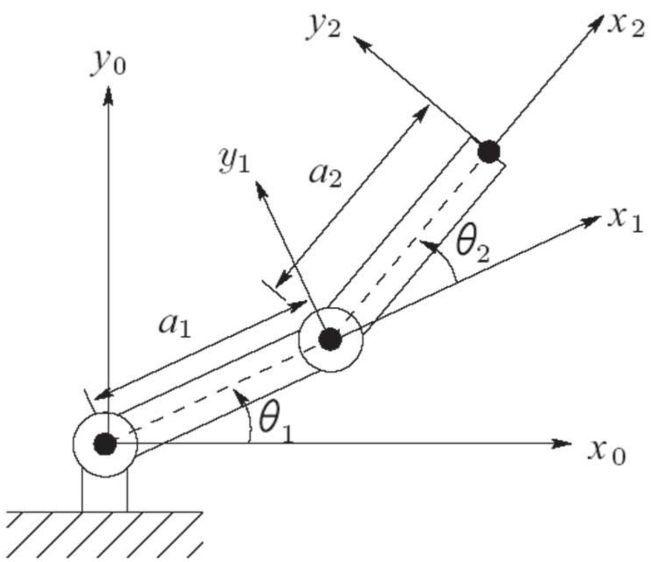
2 Update: $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{x}$



Jacobian

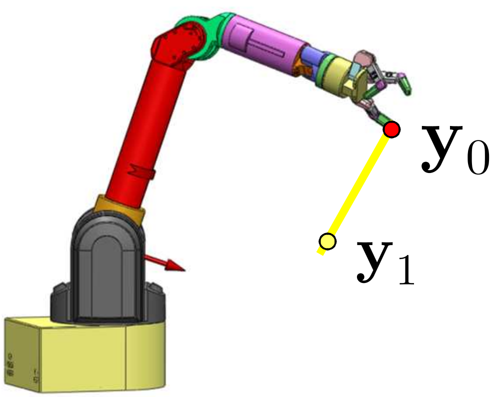
- We can analytically derive the Jacobian from the forward kinematics
- EX: two link manipulator
  - Analytic Jacobian J is:

$$J(q)=\begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$$



Jacobian

- We can also numerically compute the Jacobian in various ways based on the Secant constraint
  1. Move to joint angles $\mathbf{x} = [\, x_1, \quad x_2, \ldots \quad x_n \,]^T$
  2. Fwd kin gives Eucl pos $\mathbf{y} = [x, y, z]^T \quad = \quad f(\mathrm{x})$
  3. Relative movements $\Delta\mathbf{y}, \Delta\mathbf{x}$
  4. Secant constraint for J: $\Delta\mathbf{y} = \mathbf{J}(\mathbf{x})\Delta\mathbf{x}$



Find J Method 1:
Test movements along basis

- Remember: J is unknown m by n matrix
  - For position only: 3x3, position and orientation 6x6 or mx6

$$\mathbf{J} = \begin{pmatrix} \frac{\partial \mathrm{f}_1}{\partial x_1} & \cdots & \frac{\partial \mathrm{f}_1}{\partial x_n} \\ \vdots & \ddots & \\ \frac{\partial \mathrm{f}_m}{\partial x_1} & & \frac{\partial \mathrm{f}_\mathrm{m}}{\partial x_n} \end{pmatrix}$$

$$\Delta\mathbf{x}_1 = [1, 0, \ldots, 0]^T$$
$$\Delta\mathbf{x}_2 = [0, 1, \ldots, 0]^T$$
$$\vdots$$
$$\Delta\mathbf{x}_n = [0, 0, \ldots, 1]^T$$

- Do test movements
- Finite difference:

$$\mathbf{Jt} \quad \left(\begin{bmatrix} \vdots \\ \Delta\mathbf{y}_1 \\ \vdots \end{bmatrix} \begin{bmatrix} \vdots \\ \Delta\mathbf{y}_2 \\ \vdots \end{bmatrix} \cdots \begin{bmatrix} \vdots \\ \Delta\mathbf{y}_n \\ \vdots \end{bmatrix} \right)$$

Find J Method 2: Secant Constraints

- Constraint along a line:
- Defines m equations

$$\Delta\mathbf{y} = \mathbf{J}\Delta\mathbf{x}$$

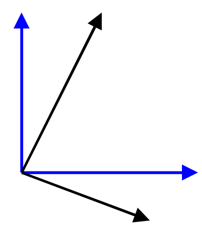
- Collect n arbitrary, but different measures y
- Solve for J

$$\begin{pmatrix} \begin{bmatrix} \cdots & \Delta\mathbf{y}_1^T & \cdots \end{bmatrix} \\ \begin{bmatrix} \cdots & \Delta\mathbf{y}_2^T & \cdots \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \cdots & \Delta\mathbf{y}_n^T & \cdots \end{bmatrix} \end{pmatrix} = \begin{pmatrix} \begin{bmatrix} \cdots & \Delta\mathbf{x}_1^T & \cdots \end{bmatrix} \\ \begin{bmatrix} \cdots & \Delta\mathbf{x}_2^T & \cdots \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \cdots & \Delta\mathbf{x}_n^T & \cdots \end{bmatrix} \end{pmatrix} \mathbf{J}^T$$

Find J Method 3: Recursive Secant Constraints Broydens method

- Based on initial J and one measure pair $\Delta \mathbf{y}, \Delta \mathbf{x}$
- Adjust J s.t. $\Delta \mathbf{y} = \mathbf{J}_{k+1} \Delta \mathbf{x}$
- Rank 1 update:

$$\hat{J}_{k+1} = \hat{J}_k + \frac{(\Delta \mathbf{y} - \hat{J}_k \Delta \mathbf{x}) \Delta \mathbf{x}^T}{\Delta \mathbf{x}^T \Delta \mathbf{x}}$$

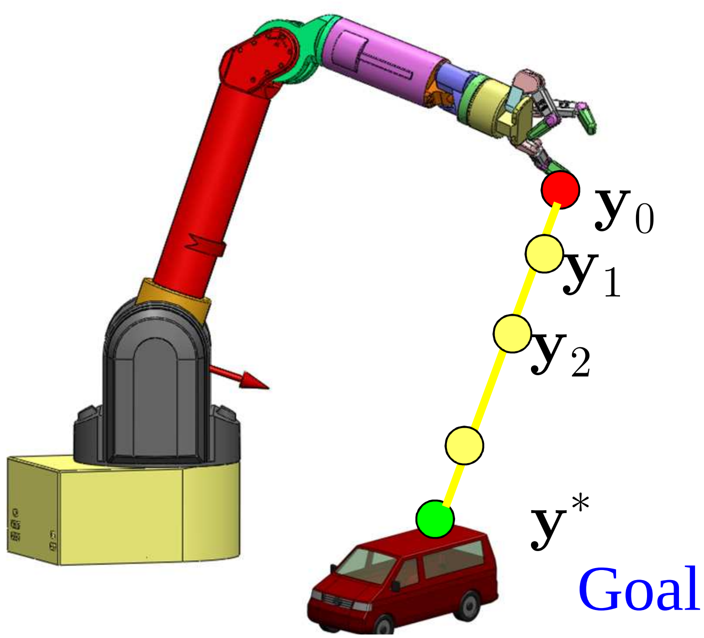


- Consider rotated coordinates:
  – Update same as finite difference for n orthogonal moves $\Delta \mathbf{x}$

Numerical Inverse Kinematics

repeat

1. Solve for motion: $[\mathbf{y}^* - \mathbf{y}_k] = \mathbf{J_k}\Delta\mathbf{x}$
2. Move robot joints: $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}$
3. Read actual Cartesian move $\Delta\mathbf{y}$
4. Update Jacobian: $\hat{J}_{k+1} = \hat{J}_k + \frac{(\Delta\mathbf{y} - \hat{J}_k\Delta\mathbf{x})\Delta\mathbf{x}^T}{\Delta\mathbf{x}^T\Delta\mathbf{x}}$



Move the robot to each subgoal in sequence $\mathbf{y}_k$

Iterate until convergence at final goal

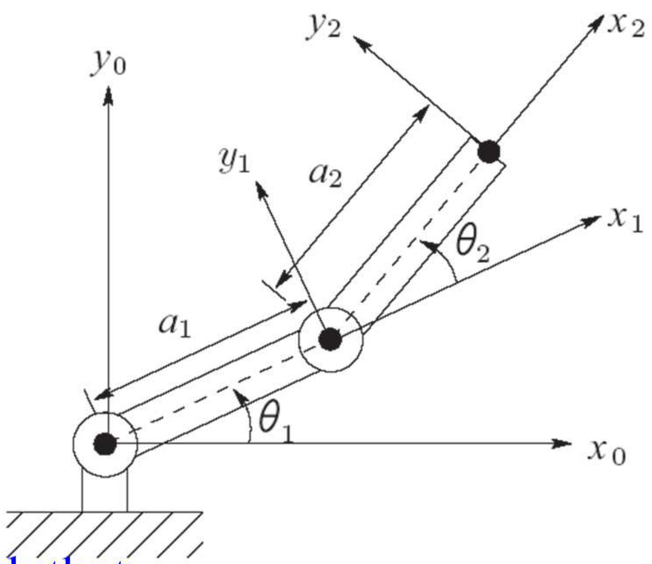
Singularities

- J singular ⇔ cannot solve eqs $[\mathbf{y}^* - \mathbf{y}_k] = \mathbf{J}\Delta\mathbf{x}$
- Definition: we say that any configuration in which the rank of *J* is less than its maximum is a singular configuration
  - i.e. any configuration that causes *J* to lose rank is a singular configuration
- Characteristics of singularities:
  - At a singularity, motion in some directions will not be possible
  - At and near singularities, bounded end effector velocities would require unbounded joint velocities
  - At and near singularities, bounded joint torques may produce unbounded end effector forces and torques
  - Singularities often occur along the workspace boundary (i.e. when the arm is fully extended)

Singularities

- How do we determine singularities?
  - Simple: construct the Jacobian and observe when it will lose rank
- EX: two link manipulator
  - Analytic Jacobian J is:

$$J(q)=\begin{bmatrix} -a_1s_1-a_2s_{12} & -a_2s_{12} \\ a_1c_1+a_2c_{12} & a_2c_{12} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$$



  - This loses rank if we can find some α such that columns are linearly dependent $J_1 = \alpha J_2$ for $\alpha \in \mathbf{R}$

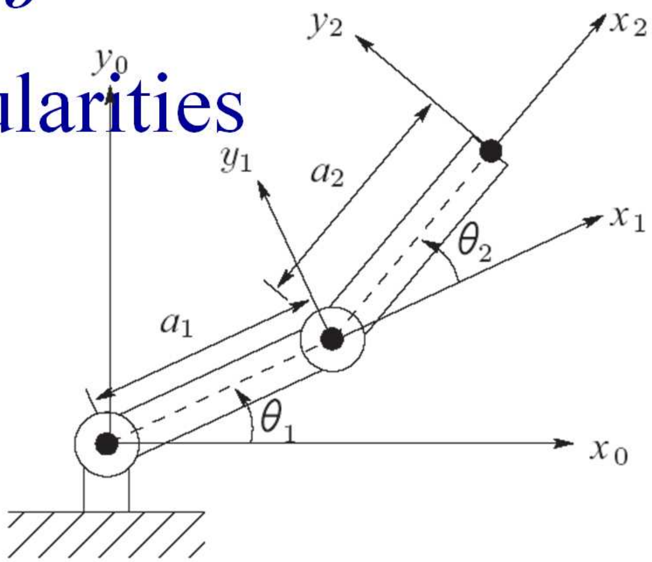
Singularities

- This is equivalent to the following:

$$a_1 s_1 + a_2 s_{12} = \alpha(a_2 s_{12})$$
$$a_1 c_1 + a_2 c_{12} = \alpha(a_2 c_{12})$$

- Thus if $s_{12} = s_1$, we can always find an α that will reduce the rank of J
- Thus $\theta_2 = 0, \pi$ are two singularities

$$\alpha = \frac{a_1 + a_2}{a_2} \qquad \alpha = \frac{a_2 - a_1}{a_2}$$

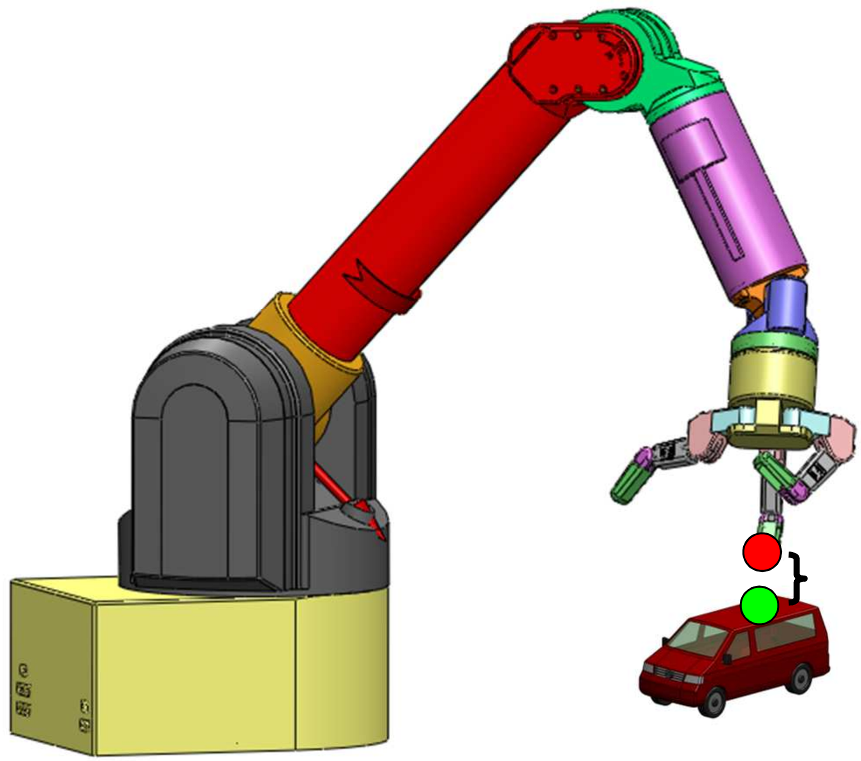


Determining Singular Configurations

- In general, all we need to do is observe how the rank of the Jacobian changes as the configuration changes
- Can study analytically
- Or numerically: Singular if eigenvalues of square matrix 0, or singular values of rectangular matrix zero. (Compute with SVD), or condition number tends to infinity.

How accurate is the movement?

- Does the robot always reach the goal?

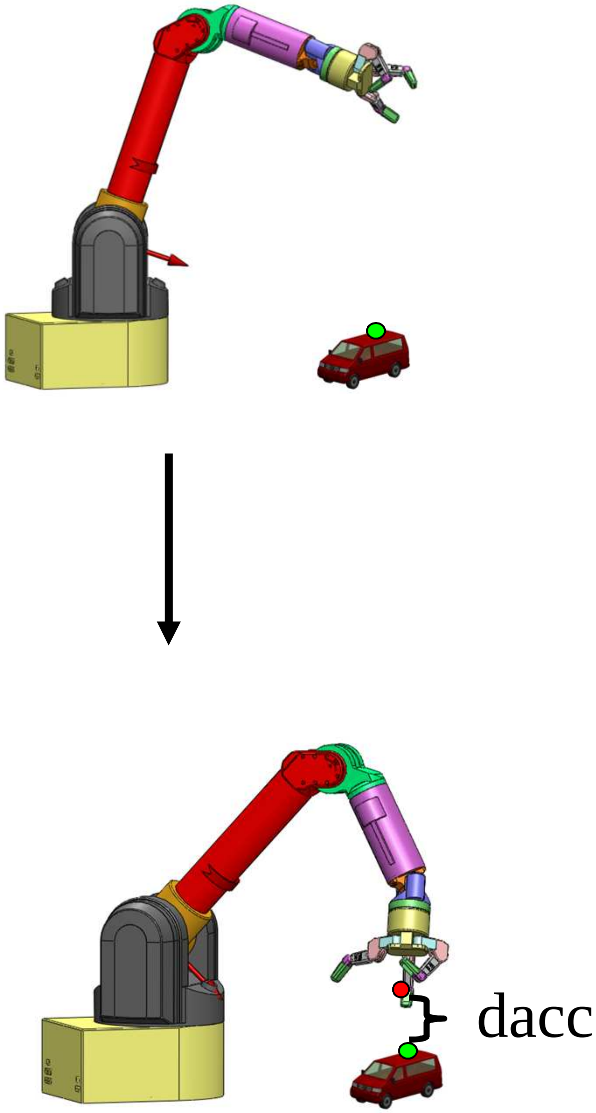


Final position error dy

Accuracy, Repeatability and Resolution

Accuracy:

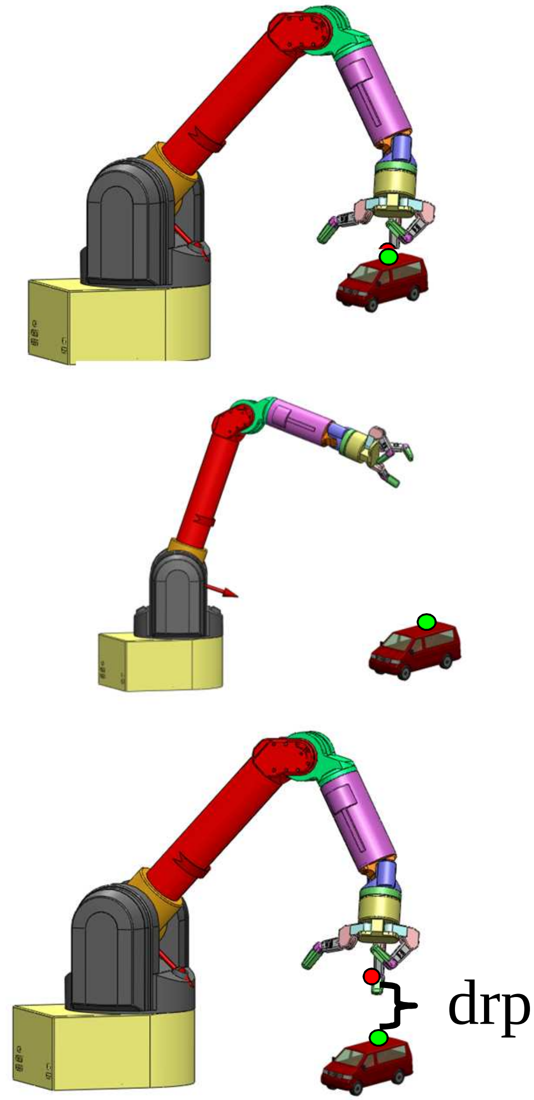
- Start far away
- Move a long distance
- Measure error dacc



Accuracy, Repeatability and Resolution

Repeatability:

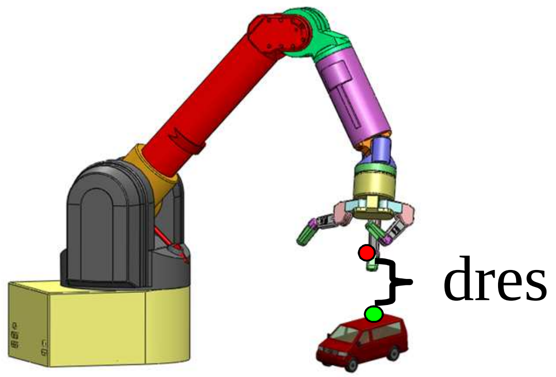
- Start at goal
- Move away a long distance
- Move back to goal
- Measure error drp

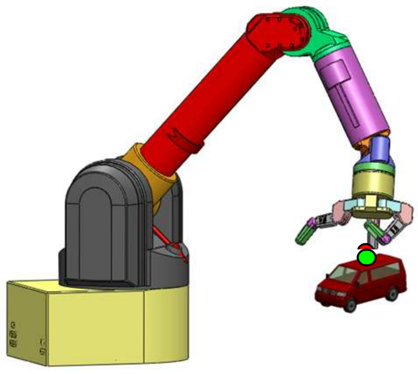


Accuracy, Repeatability and Resolution

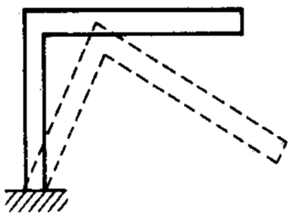
Resolution:

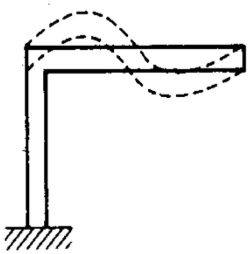
- The smallest incremental distance a robot can move.
- Typically limited by joint encoder resolution.





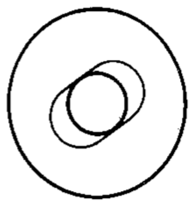
Points potentially weak in mechanical design

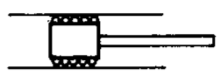


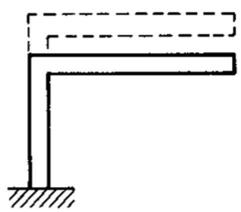


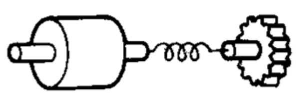
Weak points	Mechanical correction
Permanent deformation of the whole structure and the components	• Increase rigidness • Weight reduction • Counterweight
Dynamic deformation	• Increase rigidness • Reduction of the mass to move • Weight distribution
Backlash	• Reduce gear clearances • Use more rigid transmission elements

Points potentially weak in the mechanical design





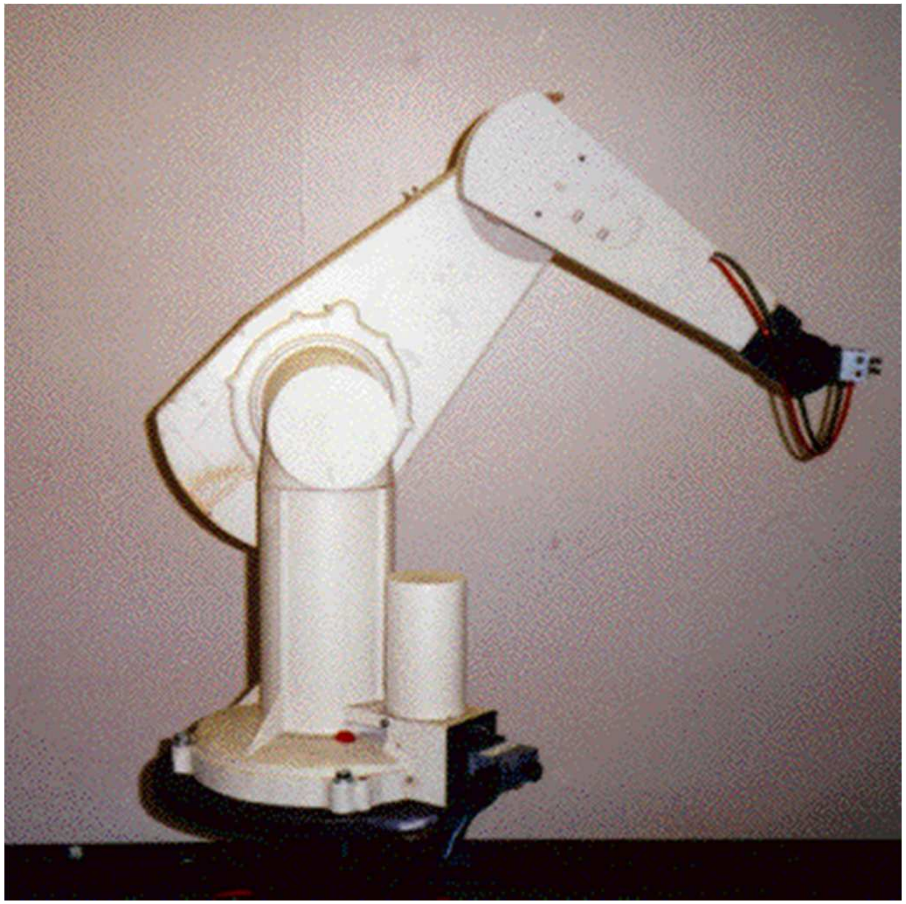




Weak points	Mechanical correction
Axes clearance	• Use pre stressed axes
Friction	• Improve clearance in axes • Increase lubrication
Thermal effects	• Isolate heat source
Bad transducers connection	• Improve mechanical connection • Search for a better location • Protect the environment

Robot Morphology:

An classic arm - The PUMA 560

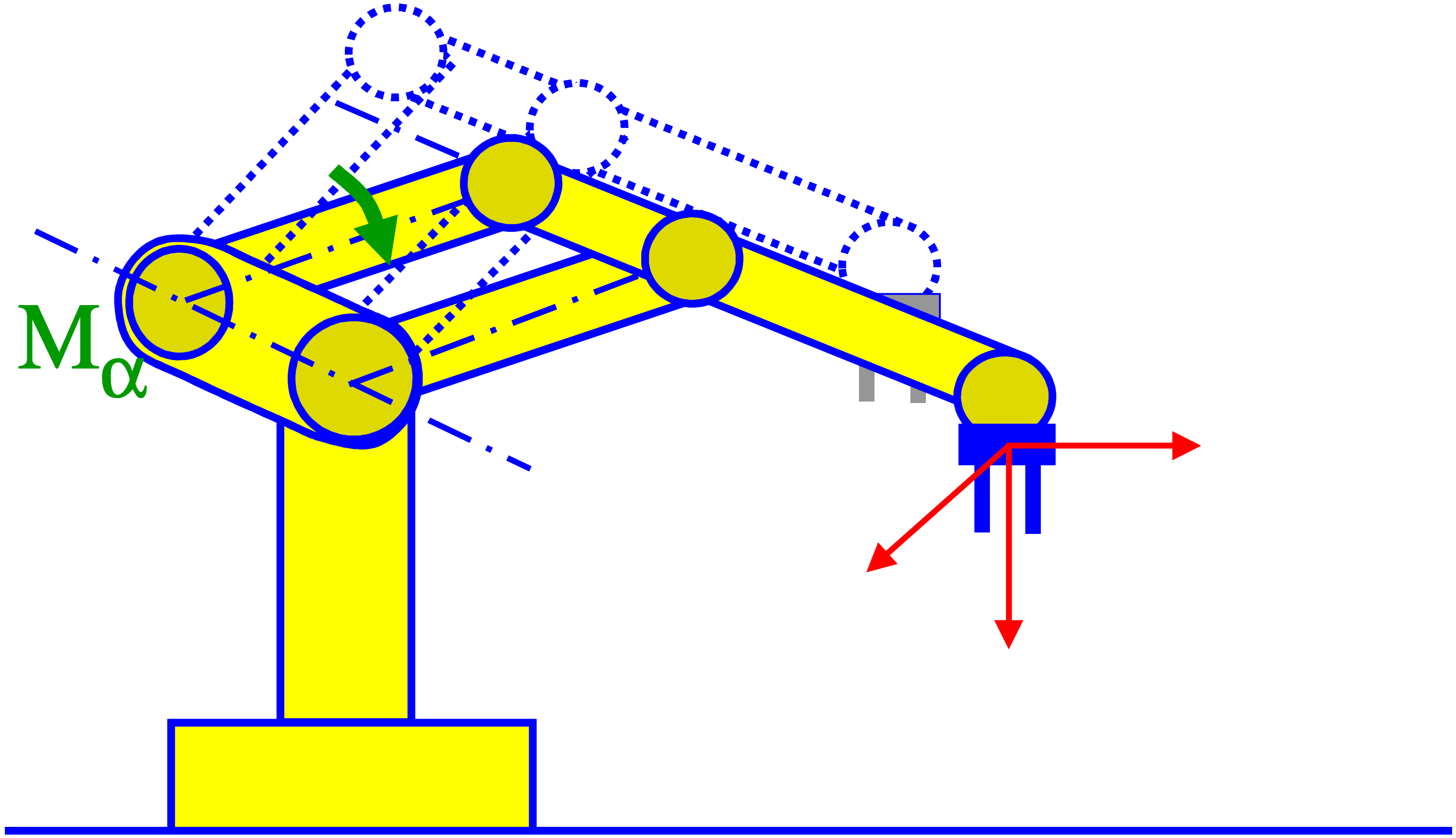


- Accuracy
  - Factory about 10mm
  - Calibrated 1-5mm
- Repeatability 1mm
- Resolution 0.1mm

An modern arm - The Barrett WAM

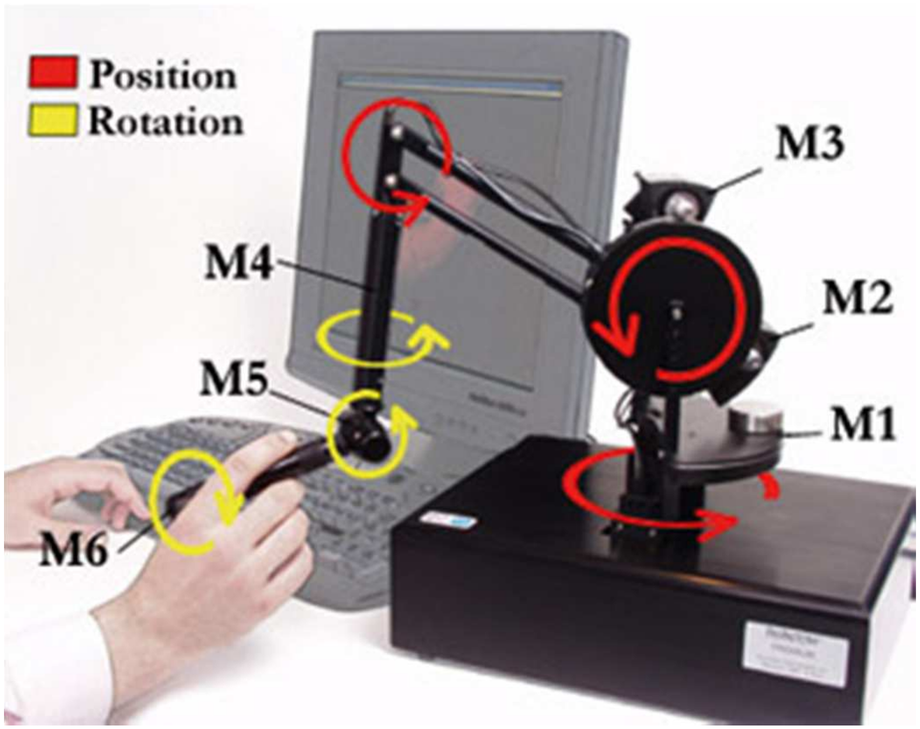
- Resolution and repeatability similar or better than PUMA
- Accuracy worse due to lighter, more flexible linkage
- But can move faster
- Needs external feedback (e.g. camera vision) for accurate motions

5 – Bar linkage

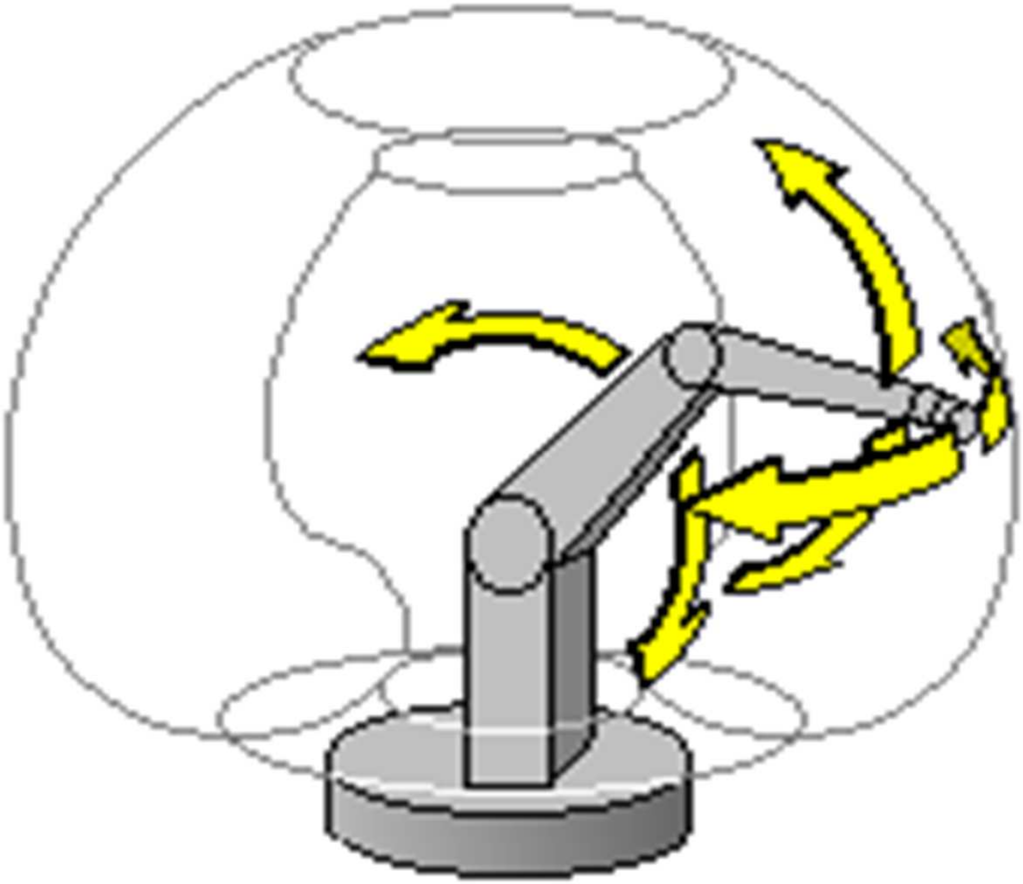


Can build this with LEGO!

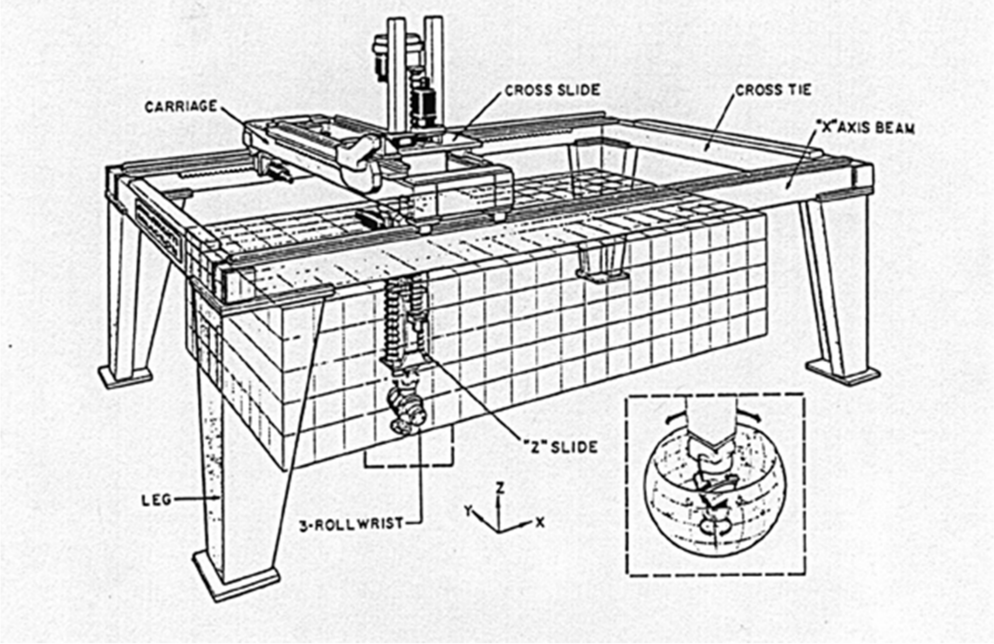
Parallel linkage
Sensable Phantom



Workspace of a typical serial arm



Cartesian robot



Typical performance

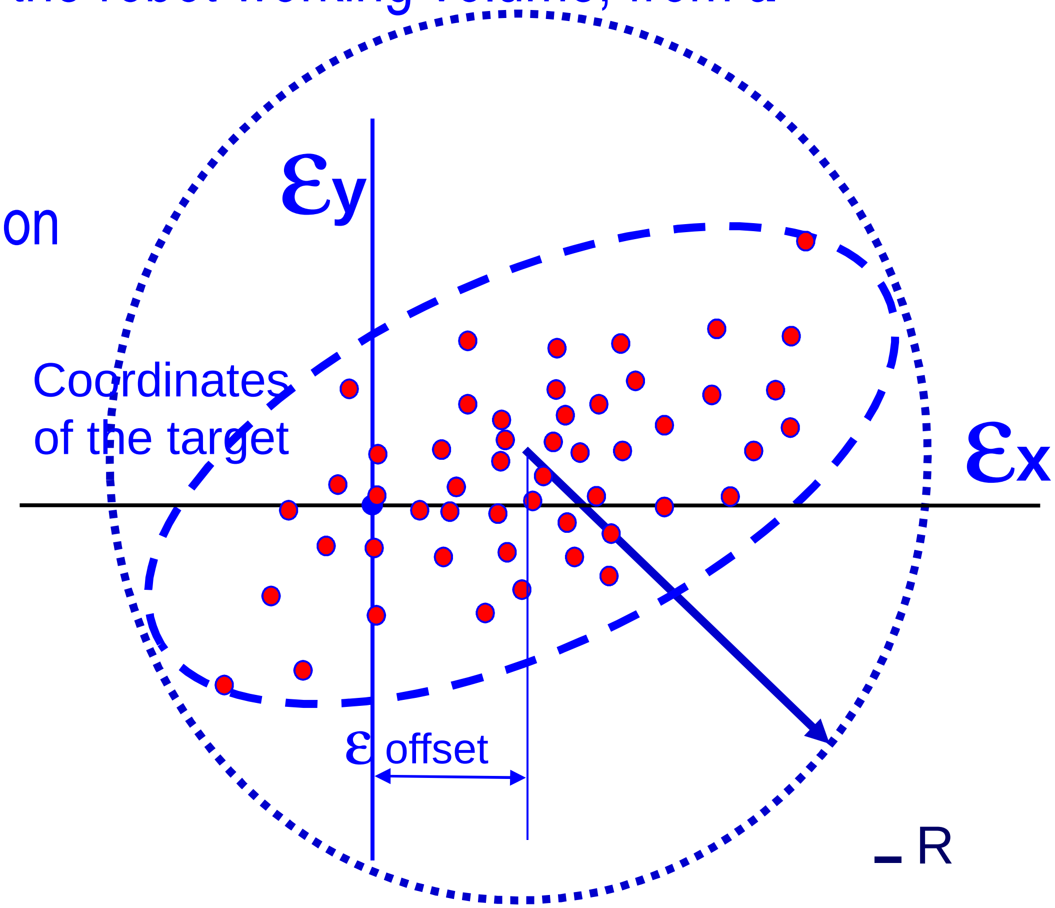
- Accuracy 0.1-1mm
- Repeatability: 0.01mm
- Resolution: 0.01mm
- Drawbacks
  - Large, heavy
  - Workspace “inside” the robot

Measuring Accuracy

Accuracy

- Capacity to place the end effector into a given position and orientation (pose) within the robot working volume, from **a random** initial position.

ε increases with the motion distance



Measure:

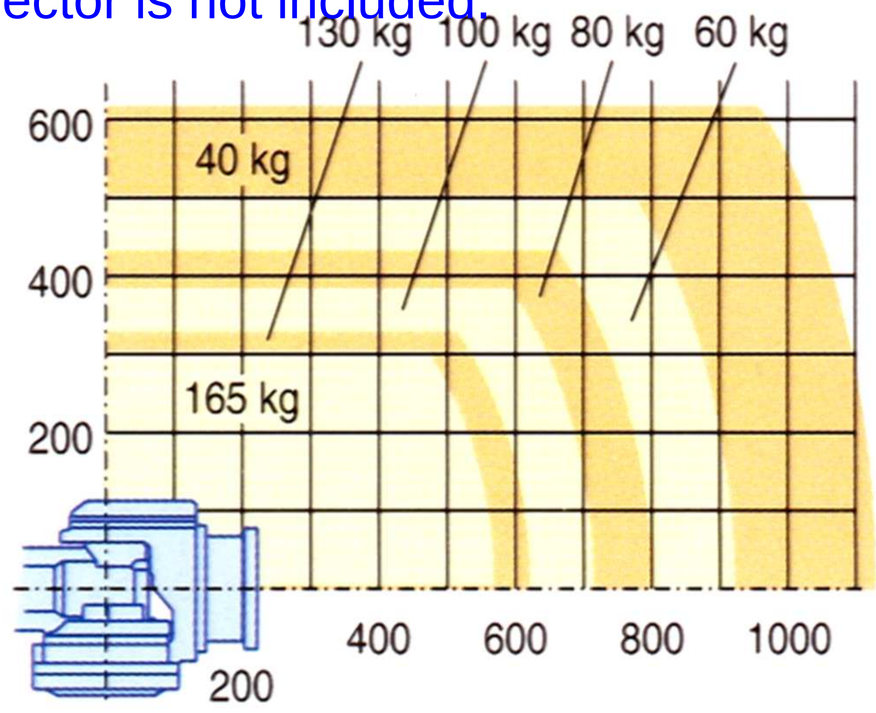
- Sample many start and end positions
- Characterize errors:
  - systematic ε (offset)
  - random

Dynamic Characteristics

Payload:

- The load (in Kg) the robot is able to transport in a continuous and precise way (stable) to the most distance point
- The values usually used are the maximum load and nominal at acceleration = 0
- The load of the End-Effector is not included.

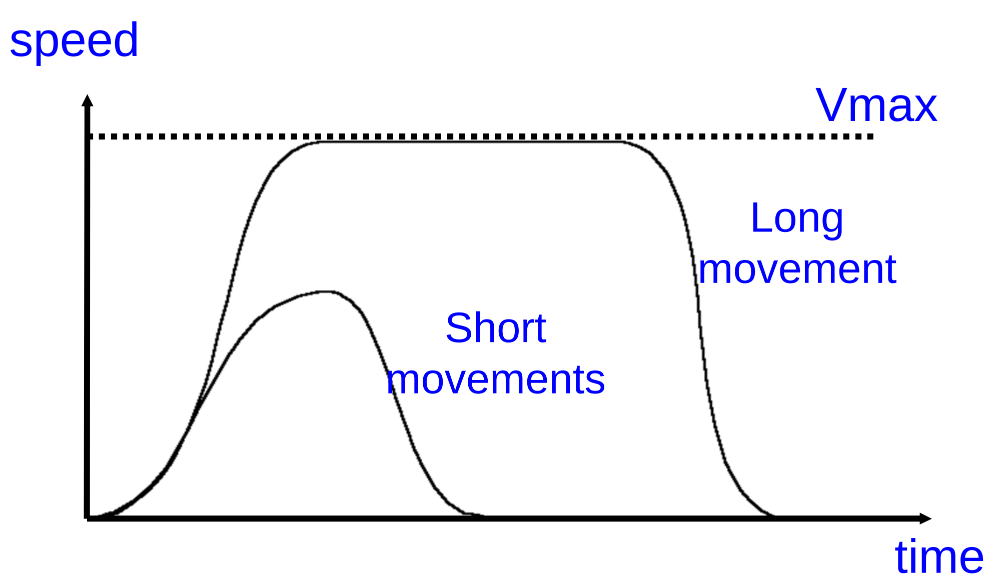
Example of Map of admitted loads, in function of the distance to the main axis



Dynamic Characteristics

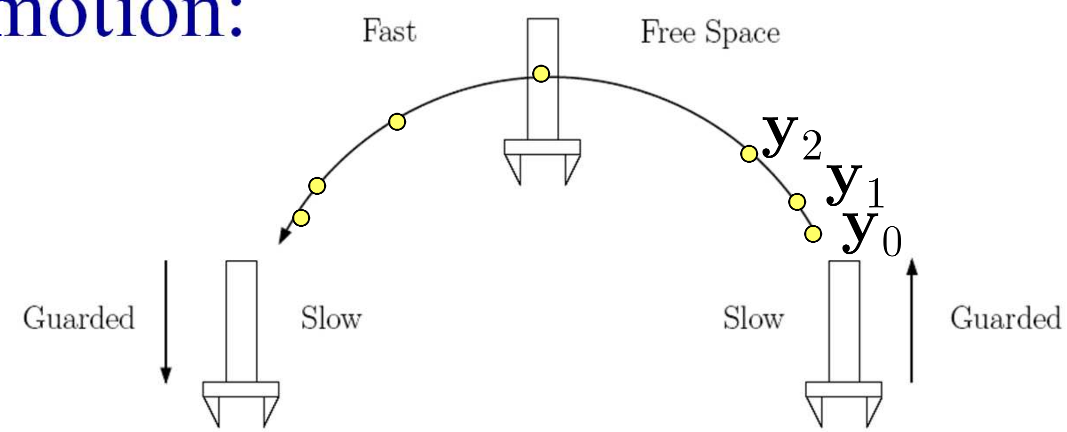
Velocity

- Maximum speed (mm/sec.) to which the robot can move the End-Effector.
- It has to be considered that more than a joint is involved.
- If a joint is slow, all the movements in which it takes part will be slowed down.
- For shorts movements acceleration matters more.



Path/trajectory planning

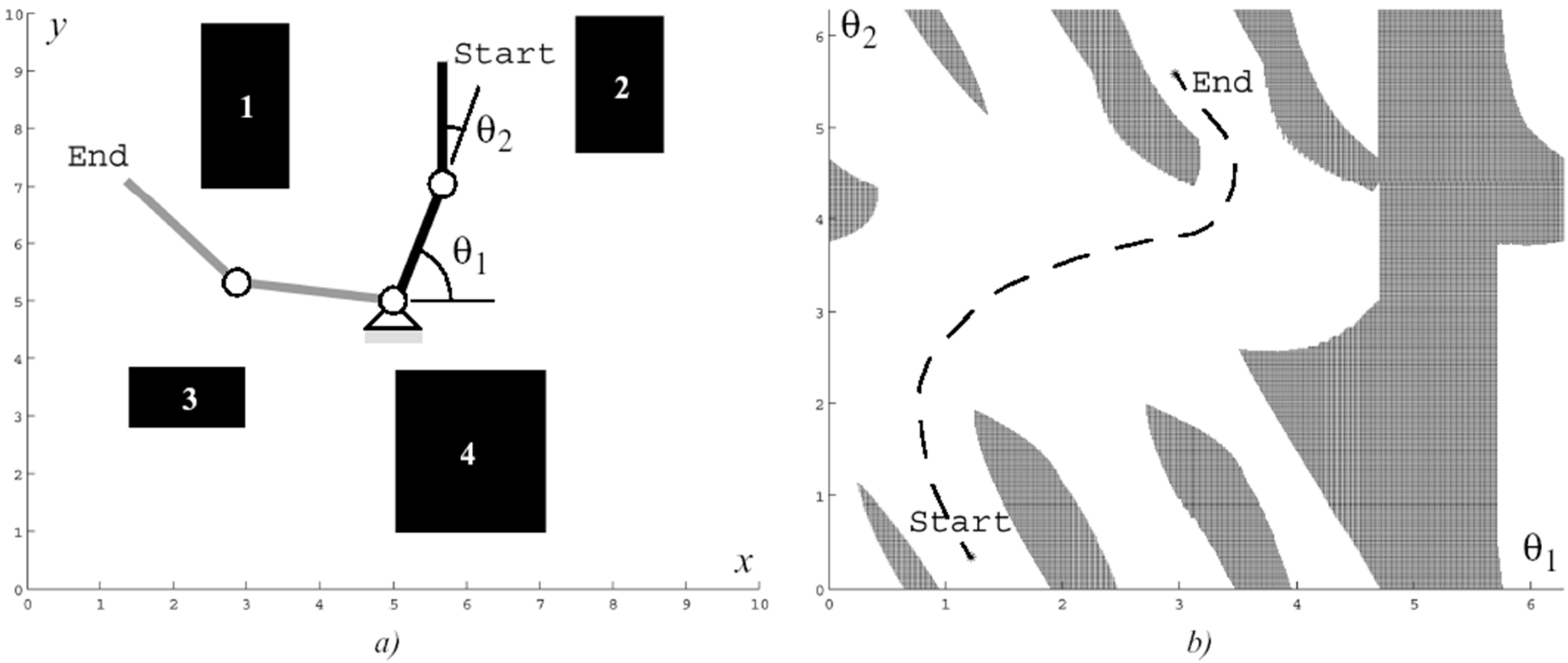
- Split the workspace into areas of fast and slow guarded motion:



- In practice robot has to smoothly accelerate.
- Can create position/velocity profiles with linear or higher order polynomials/splines.

Path/trajectory planning

- With many obstacles may need motion planning algorithms (Potential fields, RRT etc -- later)



Conclusions:

- Different architectures have different kinematics and different accuracy.
  - Serial arms: Slender, agile but somewhat inaccurate
  - Cartesian (x,y,z- table): Very accurate, but bulky
- Accuracy can be improved by:
  - Cartesian calibration
  - Visual or other sensory feedback (next in course)