Robot Equations of Motion

Topics

- Forward and Inverse Kinematics
    - Single-body Robots (Mobile Robots)
    - Multi-bodied Robots (Robot Manipulators)

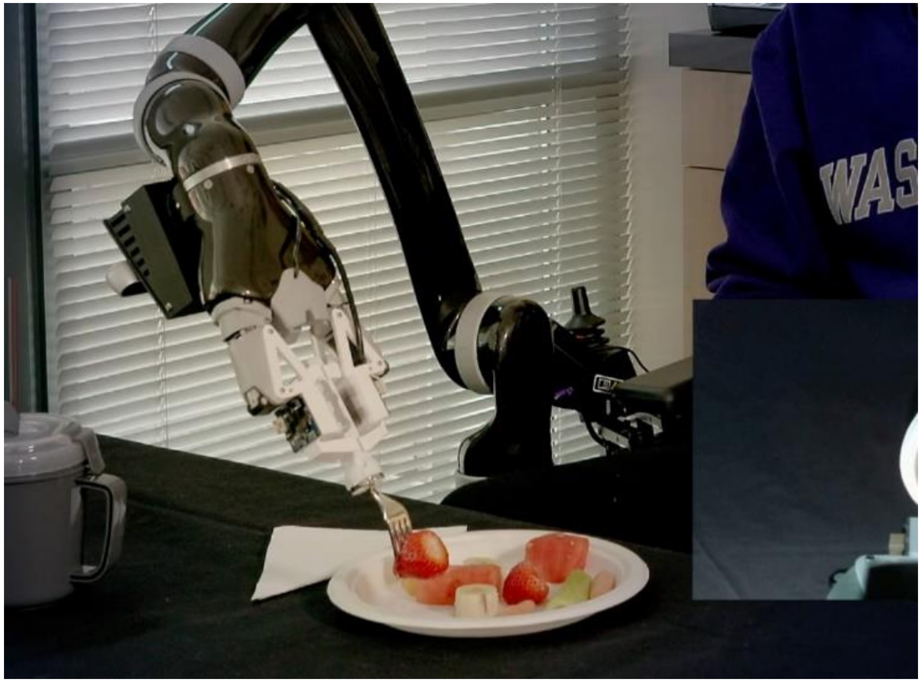
<u>Complementary Reading:</u> J.J. Craig, Introduction to Robotics, Chapter 2,3,4.

Why are models useful for robot learning?

- May be used for simulation and prediction
  - Used for virtual exploration of a robot’s controllability
  - Used for predicting the value of a controller over a time horizon
  - Useful for reducing the amount of data required to learn a controller
  - *Learning* the model allows the robot system to be transferrable to environments where data was not collected
  - Having approximate models, even with residual errors, make the learning task into learning residuals functions, which is often easier to model with less data.

Motion Modeling between Mobile Robots and Manipulators

- Geometry and mass distribution remains constant
- Sensing remains fixed relative to base



- Articulated Robot
- Volume changes
- Mass distribution changes
- Sensing (may) change relative to base
- Robot often occludes external sensing

Kinematics

Kinematics: A set of functions describing the geometric position of a robot given joint angles.

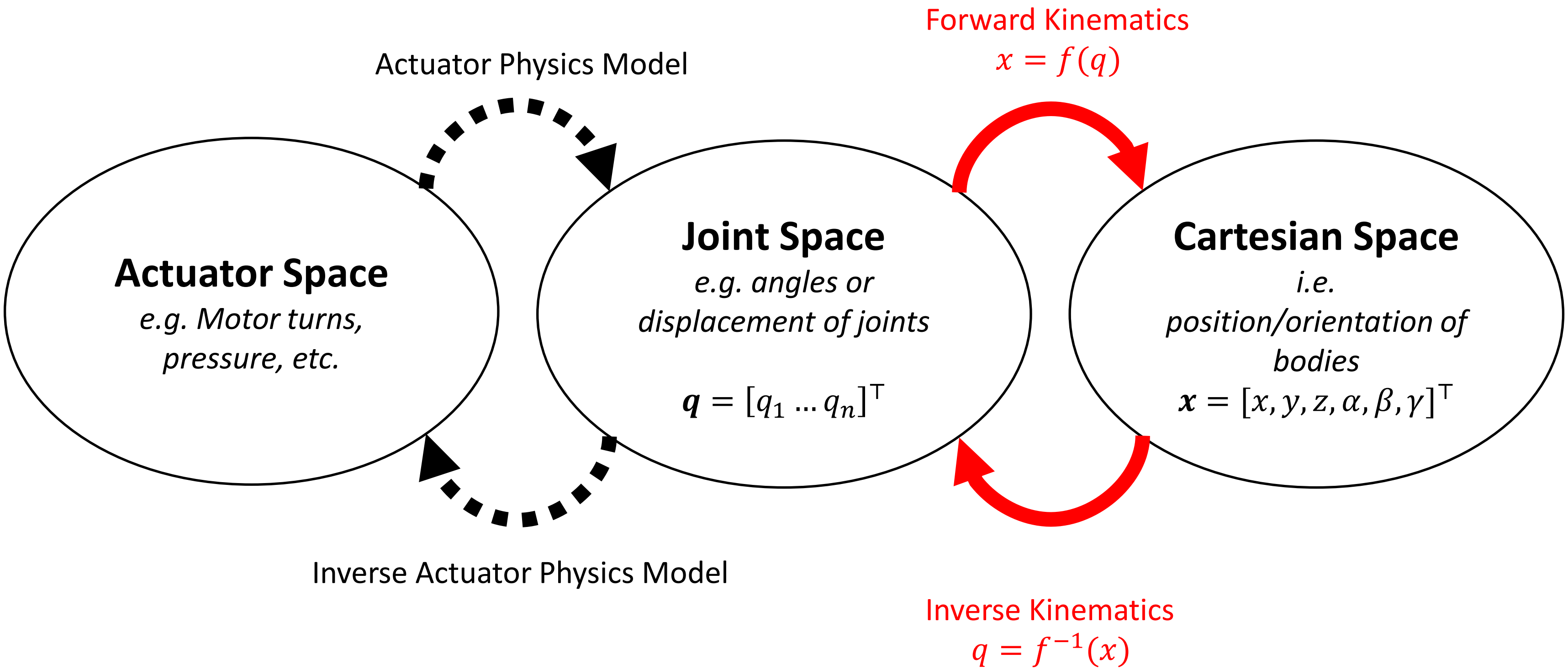
- Maps "joint" angles θ to position of robot manipulator in 3D Cartesian space, x.
  - Kinematics can describe positions of *all* the bodies; i.e. $\mathrm{N} - 6DOF$ parameters (x,y,z,r,p,y).
- θ measures either signed angles or displacement between two bodies
- By analytically taking derivatives, it maps speeds and acceleration of angles $\dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}$ to speed and acceleration of robot manipulator $\dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}$.
  - The notion of multiple coordinate frames becomes especially useful.
- Measures purely geometric functions and its derivatives, there are no *phenomenological parameters* such as mass, stiffness, force, torque, inertia, damping, friction, etc.
  - Dynamics modeling requires kinematics models first.

We are interested in two kinematics topics

- Forward Kinematics (angles to Cartesian position)
  - *Given:*
    - The shape of each link and joint locations
    - The angle/displacement of each joint
  - *Find:*
    - The position/orientation of any location on the robot
      - but typically, it's the end effector
- Inverse Kinematics (Cartesian position to angles)
  - *Given:*
    - The shape of each link and joint locations
    - The position/orientation of any location in Cartesian space

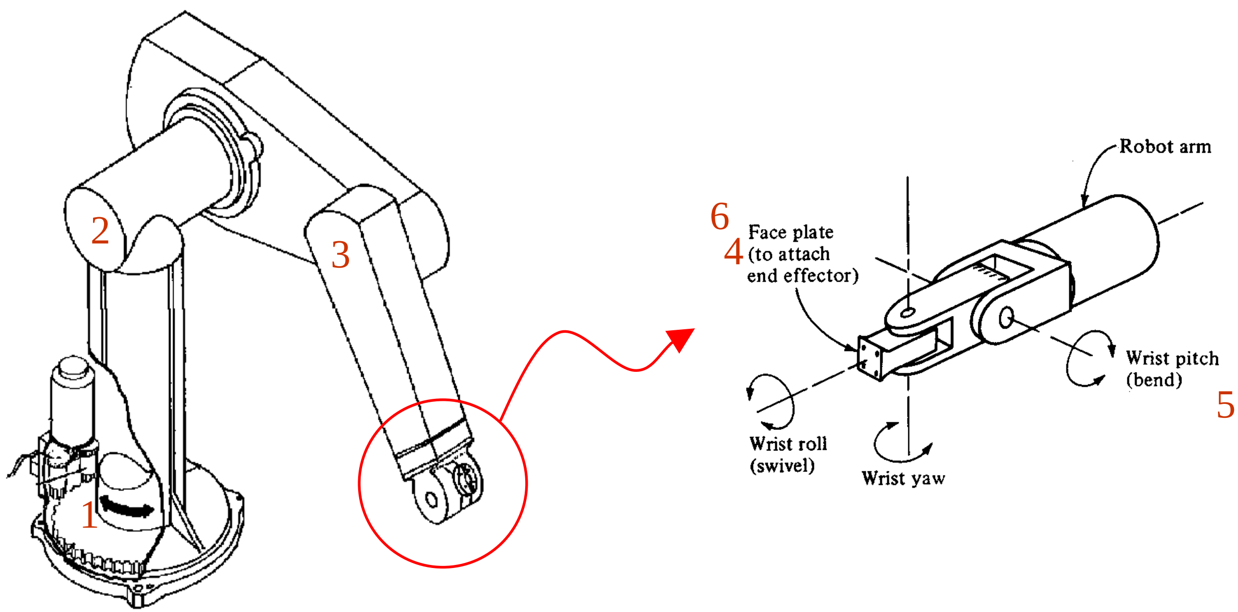
  Find:
    - The angles of each joint needed to obtain that position from a given location on the robot (typically the end-effector)

High-level Kinematics Overview



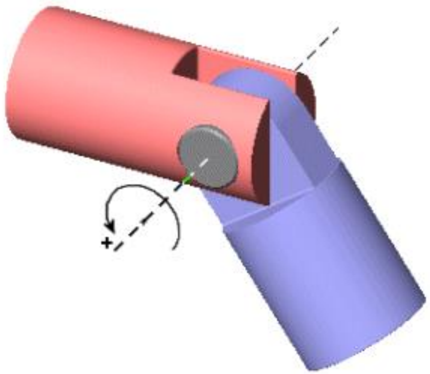
Forward Kinematics for Robot Manipulators

- A joint connects two rigid bodies together
  - i.e, a revolute joint has ONE degree of freedom (1 DOF) that changes the angle between two bodies around a single axis.
  - A prismatic joint has 1 DOF that changes the translation between bodies along a single axis.
- Example: The PUMA 560 has SIX revolute joints

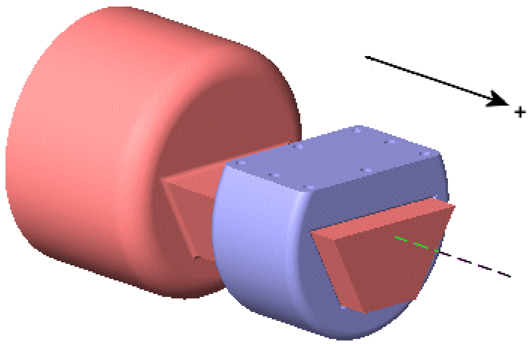


Basic Joints (R,P,S)

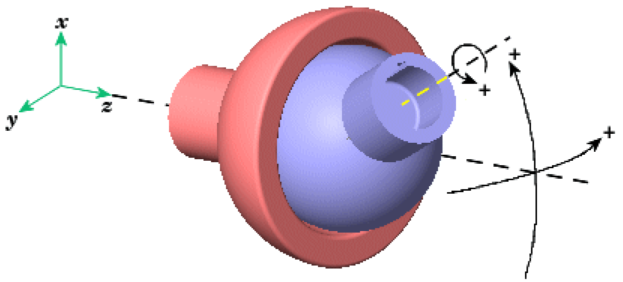
- **R**evolute Joint: 1 DOF (Variable - θ)



- **P**rismatic Joint: 1 DOF (linear) (Variables - d)

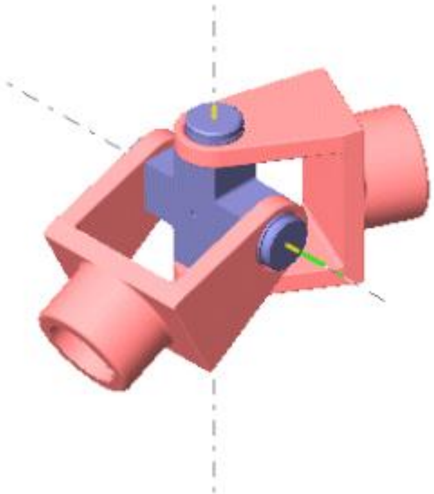


- **S**pherical Joint: 3 DOF (Variables - $\theta_1, \theta_2, \theta_3$)



Most other joints are compositions of the previous joints

- Two Revolute Joints



- Example (6-DOF arm)

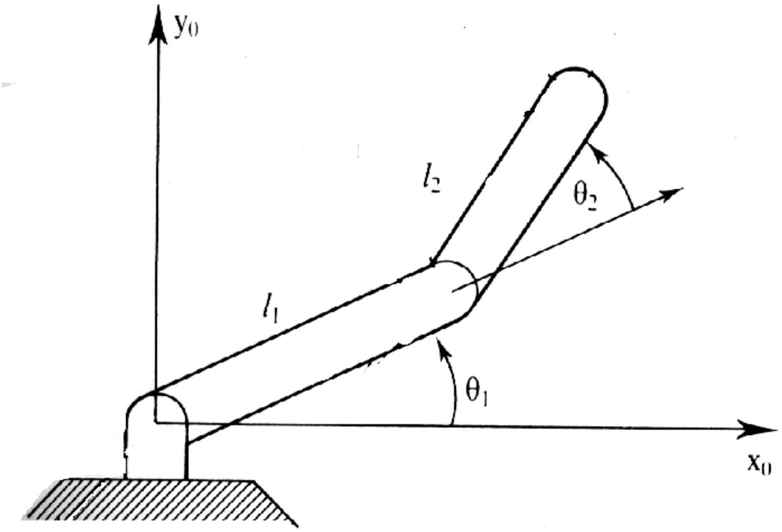
2D robot manipulator example

Forward kinematics problem:

You have a robotic arm that starts out aligned with the x_o-axis.

- Set link 1 to move by θ_1
- Set link 2 to move by θ_2

What is the position of the end of the robotic arm?



- **Geometric Solution**
  - no coordinates, no vectors, pure geometry
  - Easy for very low dimensional spaces
  - Non-trivial for higher-dimensional spaces
  - Non-generalizable
  - (skip for now)

- **Algebraic Solution**
  - vector algebra approach
  - Involves coordinate systems and vector representations
  - Generalizable to any multi-body dynamical system
  - Can follow a certain convention for labeling coordinate systems that is standardized (called Denavit-Hartenberg, or D-H, parameters)

Nomenclature for Vector Algebra

Vectors: ${}^{i}a_j \quad \in \mathbb{R}^3$

- vector a describing j in coordinate frame $\{C_i\}$

Orientations: ${}^{i}R_j \coloneqq [{}^{i}u_{x,j} \quad {}^{i}u_{y,j} \quad {}^{i}u_{z,j}] \in \mathbb{R}^{3\times3}$

- The basis vectors describing the principle axes of a body or a coordinate frame $\{C_j\}$
- These vectors are described in a coordinate frame $\{C_i\}$

Coordinate Frames: $\{C_i\} \coloneqq \{R_i, p_i\}$

- Coordinate frame attached to body i

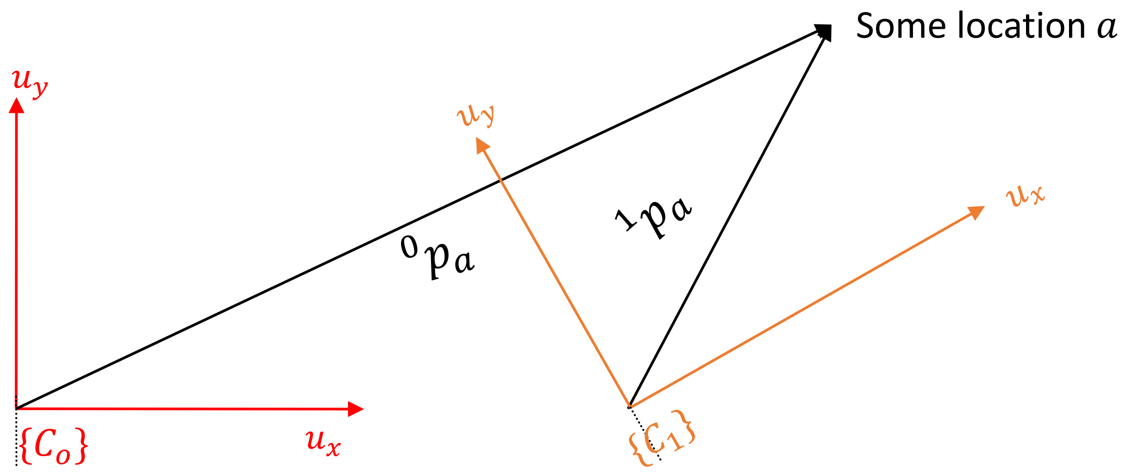
Rotation Matrix: ${}^{i}_{j}R \in \mathbb{R}^{3\times3}$

- Describe vectors, currently defined in $\{C_j\}$, in a new frame $\{C_i\}$

Moving Vectors Between Coordinate Frames

Given position of a location a in some coordinate frame $\{C_1\}$, ${}^1p_a$,

What is the position of the same point in another coordinate frame $\{C_0\}$, ${}^0p_a$?



Equation of a position vector described in $\{C_j\}$ relative to $\{C_i\}$:

$$ {}^ip_a = {}^ip_j + {}^i_jR\, {}^jp_a $$

${}^ip_j$: position of origin $\{C_j\}$ in coordinate frame $\{C_i\}$

${}^j_iR$: Orientation of frame $\{C_j\}$ described in coordinate frame $\{C_i\}$

Homogenous Matrices in SE(3)

The operation ${}^{i}p_a = {}^{i}p_j + {}^{i}_{j}R\,{}^{j}p_a$ relies on a two-step process.

- Cumbersome to work with
- Equations get messy

The Homogenous Matrix $\boldsymbol{H}$ is a 4×4 matrix is a single **operation** that describes a vector in frame A in a new frame B.

- First, augment position vectors with a trailing 1: $p \Rightarrow \begin{bmatrix} p \\ 1 \end{bmatrix}$
- Then describe the homogenous matrix as

$$ {}^{i}_{j}H = \begin{bmatrix} {}^{i}_{j}R & {}^{i}p_j \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} $$

The operation for describing vectors between coordinate frames becomes:

$$ \begin{bmatrix} {}^{i}p_a \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{i}p_j + {}^{i}_{j}R\,{}^{j}p_a \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{i}_{j}R & {}^{i}p_j \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{j}p_a \\ 1 \end{bmatrix} = {}^{i}_{j}H \begin{bmatrix} {}^{j}p_a \\ 1 \end{bmatrix} $$

Solving Forward Kinematic Chain

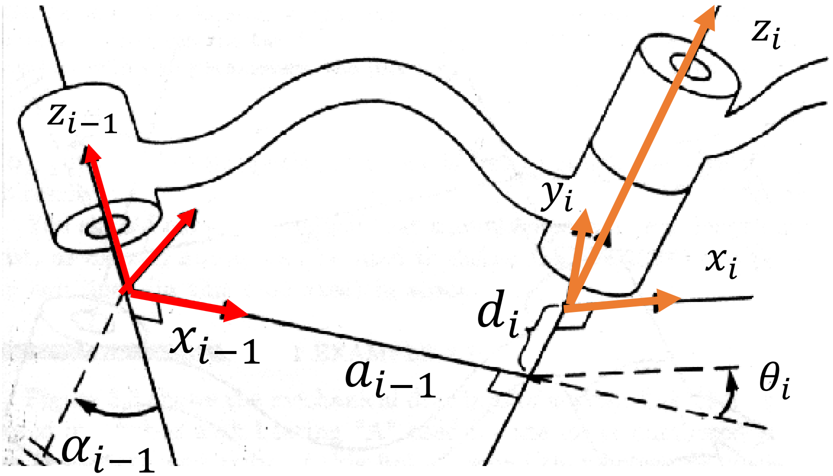
1. Choose a "World" Frame $\{C_0\}$
2. Place a coordinate frame on every Link, $\{C_1\} \ldots \{C_N\}$ where $\{R_N, p_N\}$ is the orientation and position of the end-effector,

   Describe all frames in a 4x4 matrix form, ${}^0\{C_1\} \coloneqq \begin{bmatrix} {}^0R_1 & {}^0p_1 \\ 0 & 1 \end{bmatrix}$

3. Starting at the base, ${}^0\{C_1\}$, work your way forward to ${}^0\{C_N\}$ with successive homogenous transforms:

$$
\begin{aligned}
{}^0\{C_1\} &\coloneqq \begin{bmatrix} {}^0R_1 & {}^0p_1 \\ 0 & 1 \end{bmatrix} \\
{}^0\{C_2\} &= {}^0_1H\,{}^1\{C_2\} \\
&= \begin{bmatrix} {}^0_1R & {}^0p_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^1R_2 & {}^1p_\mathbf{2} \\ 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} {}^0_1R\,{}^1R_2 & {}^0_1R\,{}^1p_\mathbf{2} + {}^0p_1 \\ 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} {}^0_2R & {}^0p_\mathbf{2} \\ 0 & 1 \end{bmatrix}
\end{aligned}
$$

Establishing Coordinate Frames on Multi-body Robots with Denavit-Hartenberg Method



- Denavit-Hartenberg (DH) rules help standardize where to put coordinate frames. Follow below:

  i. Draw axis of motion on all joints. Let this axis be the z axis.

  ii. Decide on an origin on this axis (start from the end-effector and work your way back)

  iii. Draw x axes such that it points towards the next link's joint, along its common normal.

  iv. Y is now defined based on right-hand rule.

D-H parameters explained

$a_{(i-1)}$:

length of the perpendicular between the joint axes. The joint axes is the axes around which revolution takes place which are the Z(i-1) and Z(i) axes.

α:

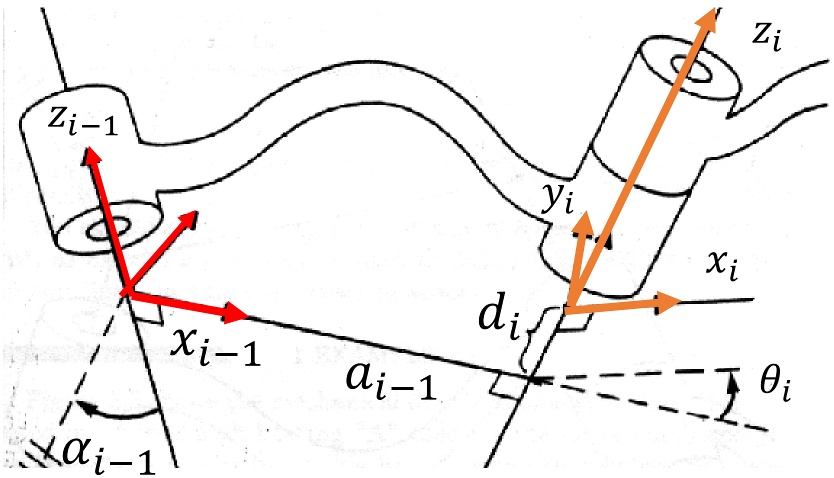
Amount of rotation around the common perpendicular so that the joint axes are parallel.

θ:

Amount of rotation around the z_i axis needed to align the x_{i-1} and x_i.

d:

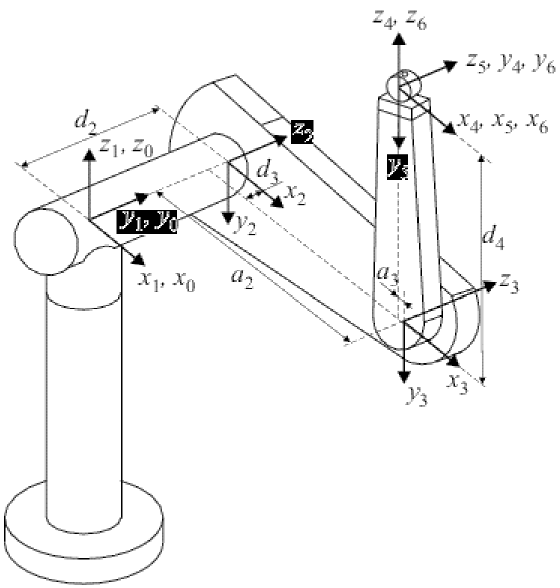
displacement along the z_i axis to align the x_{i-1} and x_i axes after rotating θ.



The D-H Matrix:

$$
{}^{i-1}_{\;\;i}H = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin\alpha_{i-1}d_i \\ \sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Note the presented D-H method is called modified DH. It is generally thought to be more intuitive than the original formulation by Denavit and Hartenberg.



PUMA 560

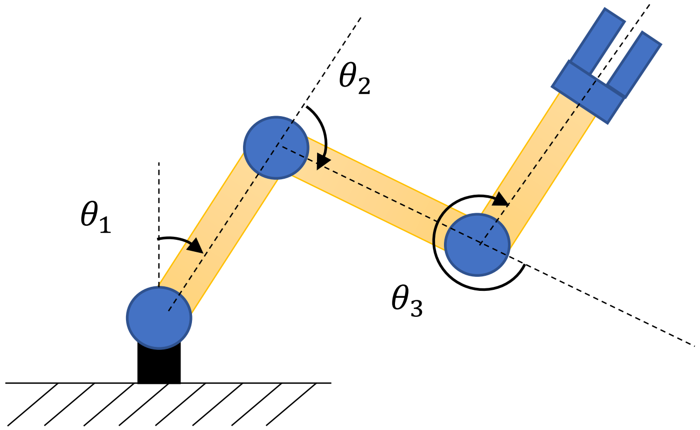
α_{i-1}	a_{i-1}	θ_i	d_i
0	0	θ_1	0
$-\pi/2$	0	θ_2	d_2
0	a_2	θ_3	d_3
$\pi/2$	a_3	θ_4	d_4
$-\pi/2$	0	θ_5	0
$\pi/2$	0	θ_6	0

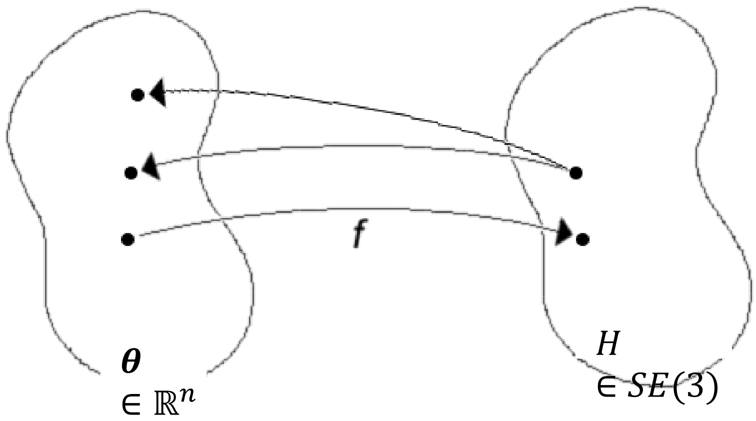
Inverse Kinematics

- Consider the forward kinematics of a robot, that maps $\theta \rightarrow {}^0_xR, {}^0p_x$
- Inverse kinematics maps:

$$ {}^0_xR, {}^0p_x \rightarrow \theta $$

- Inverse kinematics can results in in infinite, several, a single, or no solution.
- No standard analytical approach to solving inverse kinematics
    - Iterative approach used, and no guarantee of convergence





Levenberg Marquardt (LM)

- A more useful gradient descent method in than G-N.
- Has dynamic adjustment of step size, using parameter λ to converge faster, reduce overshoot.
- Let $f(\theta) = \frac{1}{2}(residual^{\top} residual)$. What is the residual for a 6-DOF pose?

Algorithm 5: Levenberg-Marquardt algorithm

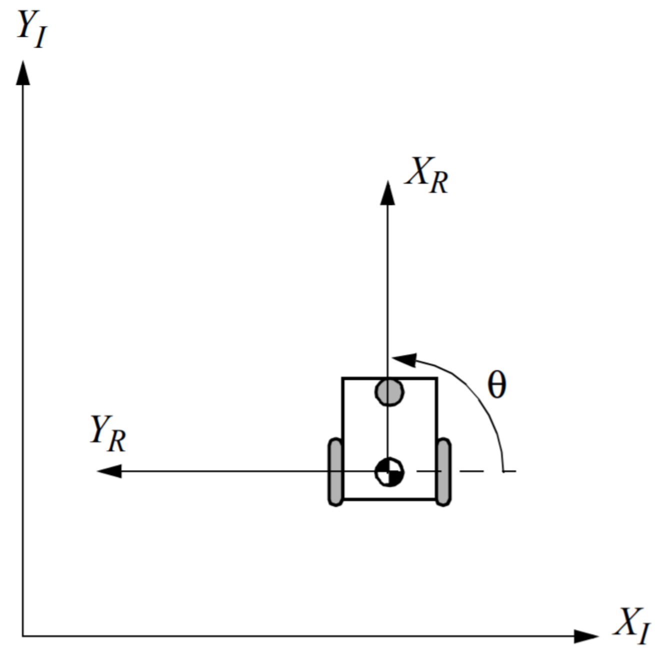
input : $f : \mathbb{R}^n \to \mathbb{R}$ a function such that $f(\mathbf{x}) = \sum_{i=1}^{m} (f_i(\mathbf{x}))^2$
where all the f_i are differentiable functions from $\mathbb{R}^n$ to $\mathbb{R}$
$\mathbf{x}^{(0)}$ an initial solution

output: $\mathbf{x}^\star$, a local minimum of the cost function f.

begin
- $k \leftarrow 0$;
- $\lambda \leftarrow \max \operatorname{diag}(\mathbf{J}^\top \mathbf{J})$;
- $\mathbf{x} \leftarrow \mathbf{x}^{(0)}$;
- **while** STOP-CRIT **and** $(k < k_{max})$ **do**
  - Find $\boldsymbol{\delta}$ such that $(\mathbf{J}^\top \mathbf{J} + \lambda \operatorname{diag}(\mathbf{J}^\top \mathbf{J}))\boldsymbol{\delta} = \mathbf{J}^\top \mathbf{f}$;
  - $\mathbf{x}' \leftarrow \mathbf{x} + \boldsymbol{\delta}$;
  - **if** $f(\mathbf{x}') < f(\mathbf{x})$ **then**
    - $\mathbf{x} \leftarrow \mathbf{x}'$;
    - $\lambda \leftarrow \frac{\lambda}{\nu}$;
  - **else**
    - $\lambda \leftarrow \nu\lambda$;
  - $k \leftarrow k + 1$;
- **return** $\mathbf{x}$

end

Mobile Robots



Differential Drive Kinematics

- Two co-axial wheels with independent speeds
- Their position is described with

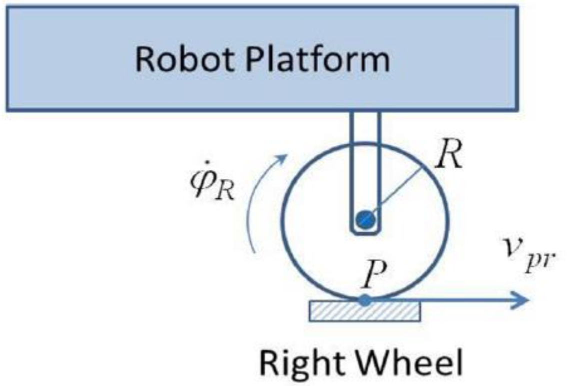
$$^0x = {}^0[p_{c,x}, p_{c,y}, \theta]$$

- Their velocity kinematics may be described easily *in the robot's body-fixed* frame R as

$$^0\dot{x} = Rot_z^{-1}(\theta)\,{}^R\dot{x}_R$$

- Given a wheel of radius r, spinning of the wheels of $\dot{\phi}_1$ and $\dot{\phi}_2$, and wheels offset from the centerline of the robot body by l, we can substitute velocities to ${}^R\dot{x}_R = [{}^R\dot{p}_{c,x}, {}^R\dot{p}_{c,y}, {}^R\dot{\theta}]$ and return

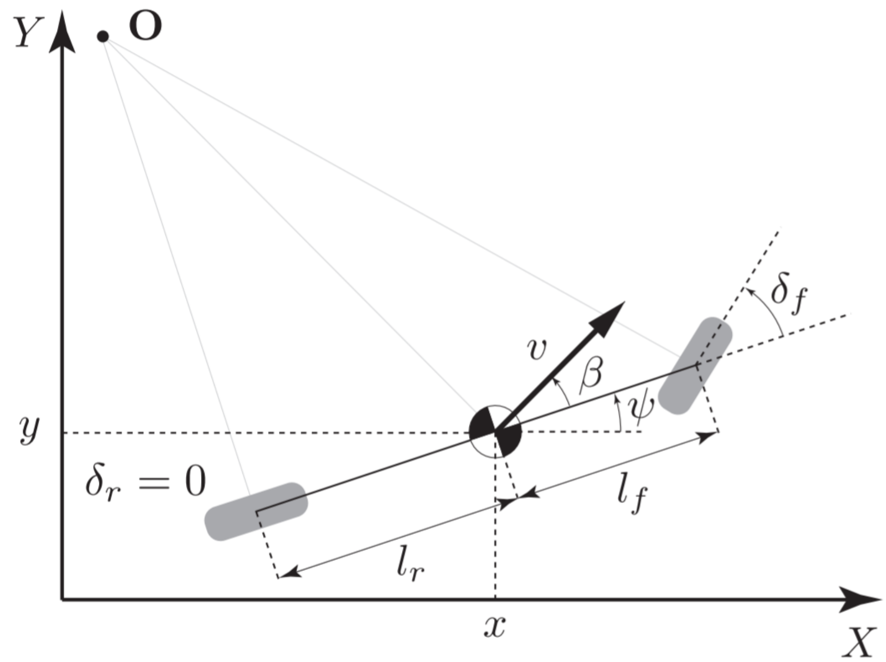
$$^0\dot{x} = Rot_z^{-1}(0)\begin{bmatrix} \frac{r(\dot{\phi}_1+\dot{\phi}_2)}{2} \\ 0 \\ \frac{r(\dot{\phi}_1-\dot{\phi}_2)}{2l} \end{bmatrix}$$



Mobile Robots

Bicycle (or Tricycle) Car Model

- Two-axle vehicle
- Back axle passive, controlled actions come from front axle steer (δ_f) and wheel acceleration (a).



- The kinematic equations are:

$$\dot{p}_{c,x} = v cos(\psi + \beta)$$

$$\dot{p}_{c,y} = v sin(\psi + \beta)$$

$$\dot{\psi}_{c,x} = \frac{v}{l_r} sin(\beta)$$

$$\dot{v} = a$$

$$\beta = tan^{-1}\left(\frac{l_r}{l_r + lf} tan(\delta_f)\right)$$

Complicating circumstances: Most robotic applications of interest are under constrained environments and involve interaction, complicating the scenario.

- With the robot has modeling errors (ϵ) or model deficiencies
- When there is more than one feasible solution to the inverse problem
- When there are constraints on the input space

$$c_{neq}(q) \leq 0$$

$$c_{eq}(q) = 0$$

- When you cannot observe x directly and require a measurement model $y = g(x)$
- When your desired workspace exceeds your reachable workspace

$$x \in X_{reachable}$$

- When the environment influences feasibility of movement
  - Collision vs. non-collision