Autonomous Robots

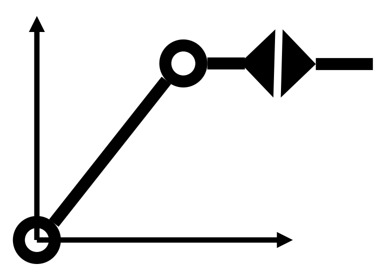
Manipulator Kinematics
Forward Kinematics

Kinematics

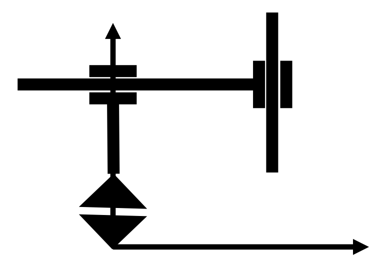
- Kinematics allows to describe the geometry and configuration of a robot
- An important aspect of this is to be able to determine where point at which the robot interacts with the environment is given the configuration of the robot
  - This can generally not be done uniquely for a mobile robot
    - There is no deterministic mapping from internal robot configuration (angles of the motors) and external position
  - On a static manipulator there is a function that determines the position of the end-effector (or any part of the robot) given its configuration
    - The Forward Kinematic function of the manipulator

Forward Kinematics of Manipulators

- Manipulators are described using a sequence of joints and links
  - Given the links and relations between joints, the joint parameters (joint angles in revolute joints and joint extensions in prismatic joints) completely describe the configuration of the system



3DOF Revolute arm



3DOF Revolute and prismatic arm

Forward Kinematics of Manipulators

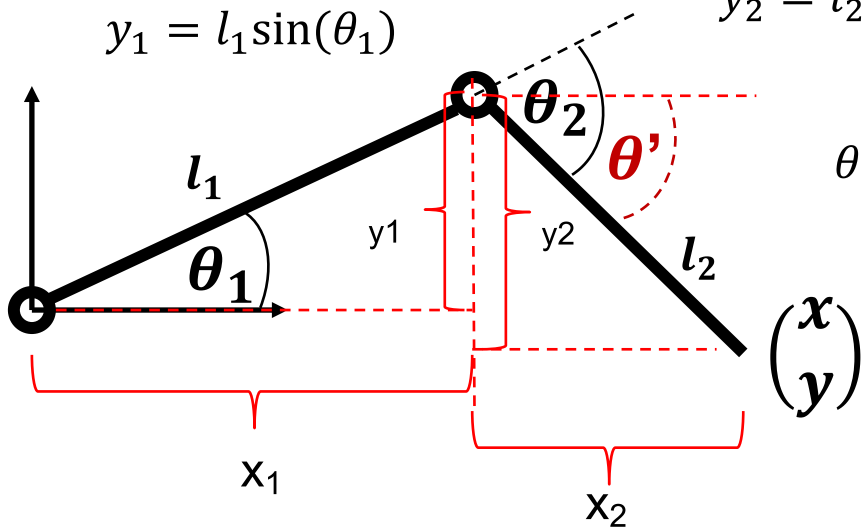
- Forward kinematics is the function that determines the endpoint location given the joint parameters
  - The forward kinematics is usually computed one of two ways
    - Using Trigonometry
    - Using Linear Algebra
- Trigonometry deals with the properties of triangles
  - Trigonometric methods thus use triangles occurring in the structure to solve the kinematics
    - Note that triangles are 2 dimensional objects and due to their definition in trigonometry all angles inside a triangle are positive and add up to π
    - In robot manipulators, joint angles are signed (i.e. can have positive and negative values) to indicate which direction a joint has been moved

Trigonometric Solution

- To solve the end-effector kinematics trigonometrically we need to find triangles in the robot structure

$y_1 = l_1 \sin(\theta_1)$

$y_2 = l_2 \sin(\theta_1 + \theta_2)$



$\theta' = \theta_1 + \theta_2$

$x_1 = l_1 \cos(\theta_1)$

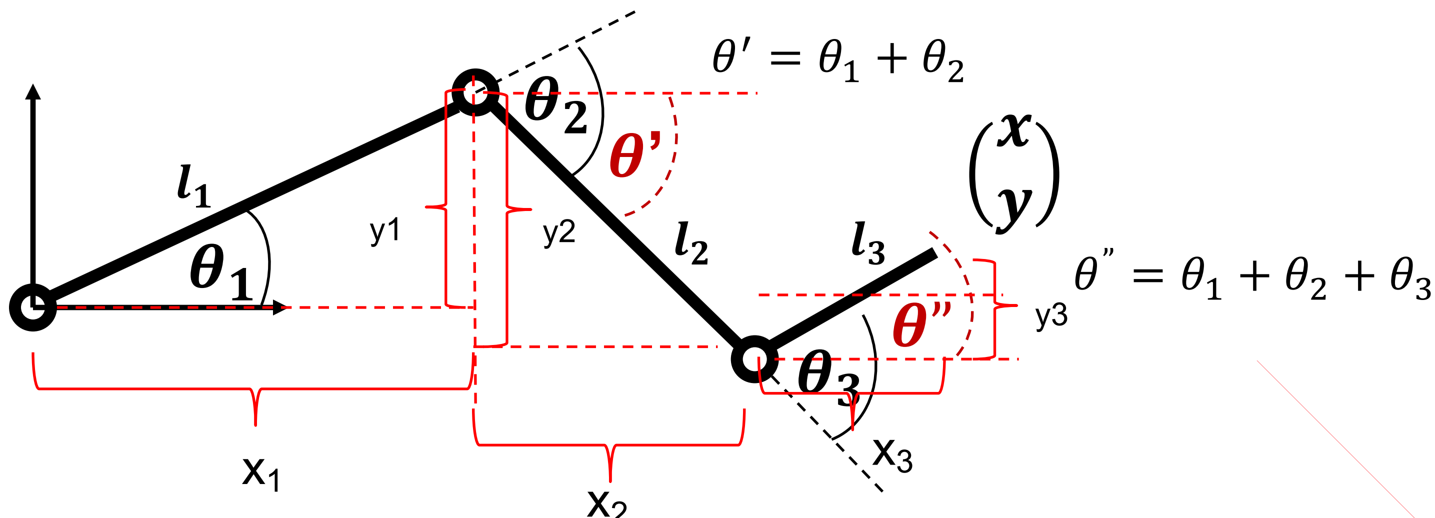
$x_2 = l_2 \cos(\theta_1 + \theta_2)$

x= $x_1 + x_2 = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$

y= $y + y_2 = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$

Trigonometric Solution

- This can be expanded to planar manipulators with more joints relatively easily
  - As before we have to be careful with signs of angles

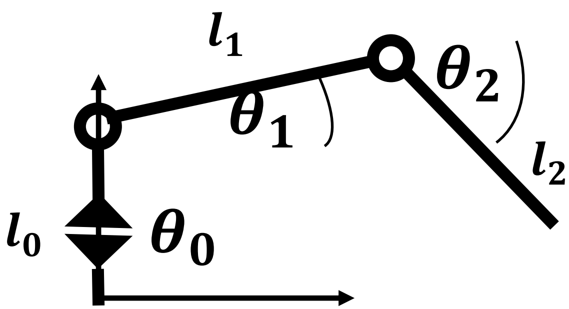


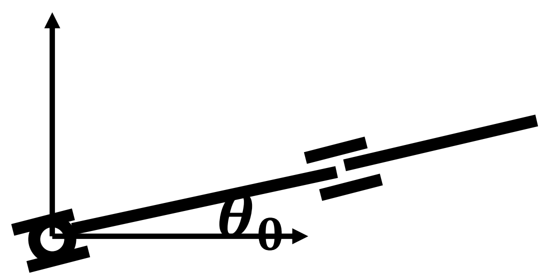
x= $x_1 + x_2 + x_3 = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3)$

y= $y_1 + y_2 + y_3 = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3)$

Trigonometric Solution

- We can also apply trigonometry to solve for the kinematic function in 3 dimensions
  - As before we have to be careful with signs of angles



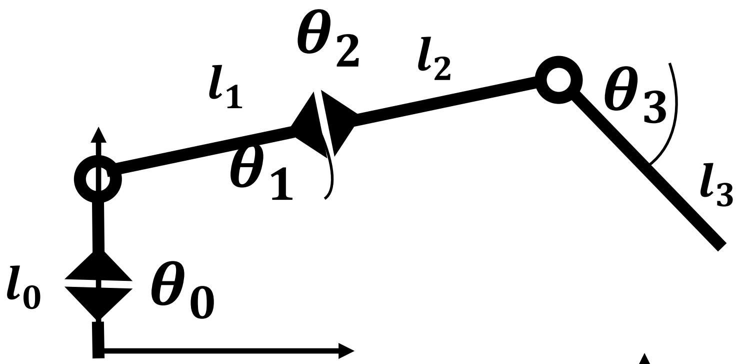


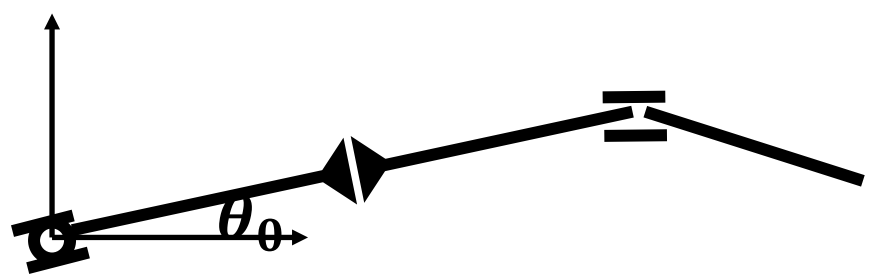
Top view

$$x = \cos(\theta_0)(l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3))$$
$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3)$$
$$z = \sin(\theta_0)(l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3))$$

Trigonometric Solution

- When adding more, identifying 2D planes in which usable triangles reside becomes difficult
  - Very easy to make mistakes





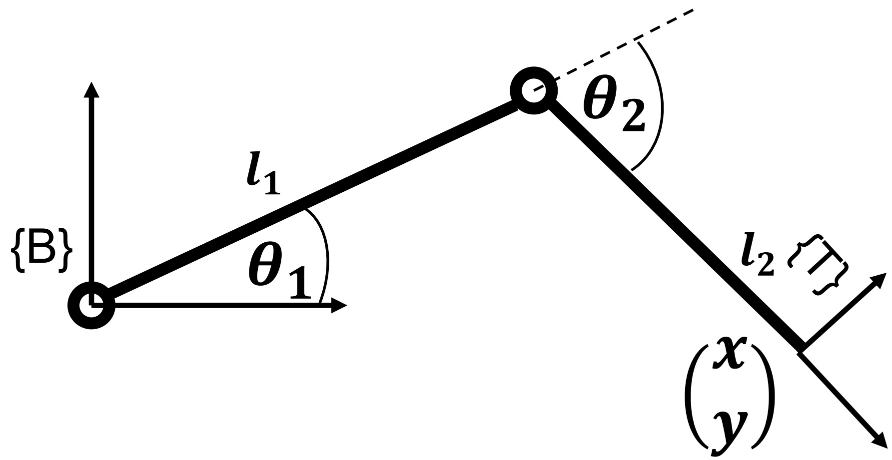
Top view

Linear Algebra Solution

- While the trigonometric solution works well for simple manipulators, it becomes increasingly difficult as the manipulators add degrees of freedom
  - Signs of joint angles make it easy to make mistakes
  - Solution can not be automated efficiently
- Linear Algebra and the transformations that we already used for representation of space provides another way to address the kinematic function
  - We can attach multiple coordinate frames to parts of the manipulator and then ask the question what their relative transformations are

Linear Algebra Solution

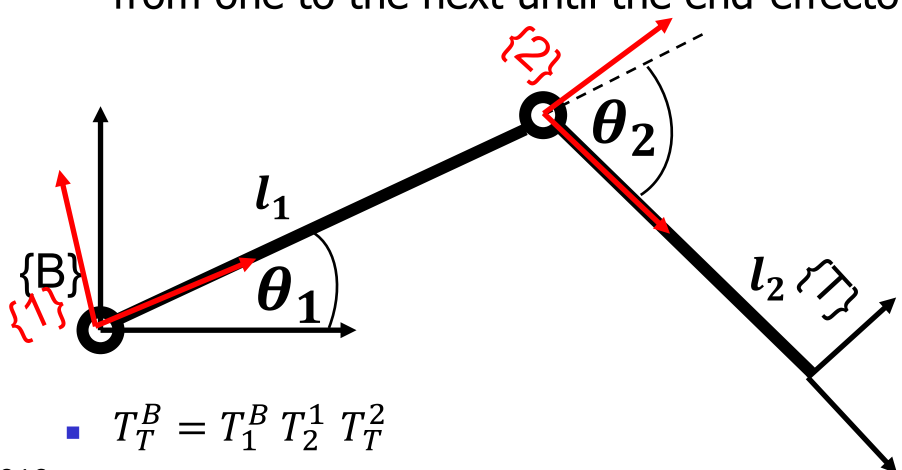
- Attaching a frame to the base of the robot {B} and one to the end-effector {T} yields a way to express the forward kinematic function as a transformation



- The location of the end-effector is the last column in T_T^B
  - Solving for the forward kinematics requires computing T_T^B
  - This solution not only provides the location but also the orientation

Linear Algebra Solution

- We can look at T_T^B as either the location of the end-effector frame in the base frame or as the movement from the base frame to the end-effector frame
  - The former is the view of the kinematics
  - The latter gives us a way to construct it in stages
    - Insert additional coordinate frames after each of the joints and move from one to the next until the end-effector frame is reached



    - $T_T^B = T_1^B \, T_2^1 \, T_T^2$

Outward Inward Iteration

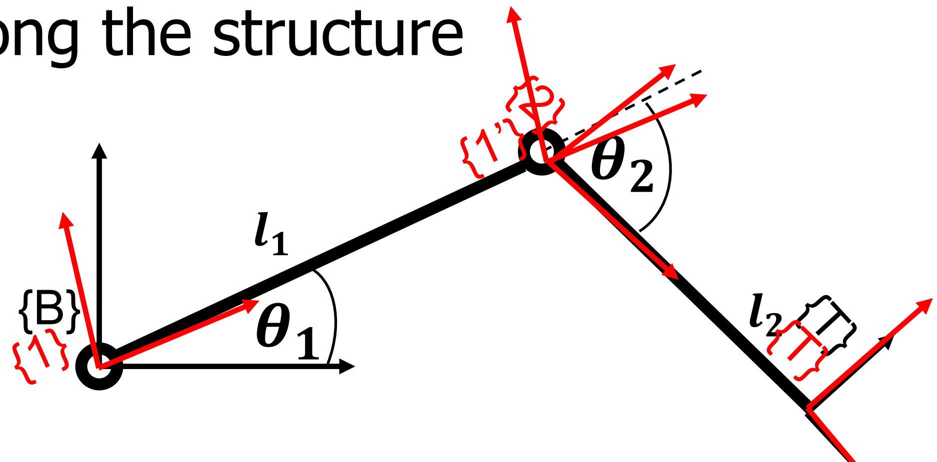
- The systematic way to solve the kinematics this way is called Outward Inward Iteration

  - Starting from the base frame, put down the movement transformations that move the coordinate frame “outward” along the kinematic chain

  - When arriving at the end-effector multiply the transformations right to left (inward) to obtain the final transform

Outward Inward Iteration

- Starting from the base frame we move the rame along the structure



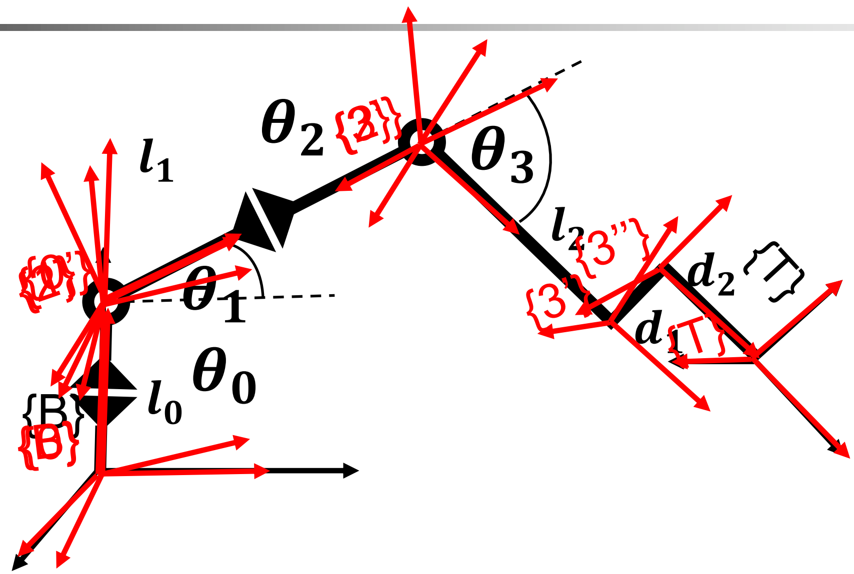
$$R_Z(\theta_1)D_X(l_1)R_Z(\theta_2)D_X(l_2) = T_T^B$$

- Outward Inward allows to look at only one joint at a time while ignoring everything before or after
  - Even complex manipulators can be solved this way

Outward Inward Iteration

- Outward Inward procedure can be automated if a description of the displacement and rotation between two consecutive frames is provided
  - Denavit Hartenberg provides such a description
    - Uses 2 parameters to describe the displacements between the axes of the joints along the link and along the axis
    - Uses 2 parameters to describe the rotational change in the axes and link direction (one around the original axis and one around the link
  - In this course we will not write down Denavit Hartenberg notations but instead will directly derive the sequence of Homogeneous transform for the end effector
    - Denavit Hartenberg is very often used in the specification of robots and can be transformed to the product of Homogeneous transforms

Outward Inward Iteration



$$T_T^B = \boldsymbol{R_Y(\theta_0)D_X(l_0)R_Z(\theta_1)R_X(\theta_2)D_X(l_1)}R_Z(\theta_3)D_X(l_2)D_Y(d_1)\ D_X(d_2)$$

- Start from the base frame
- Joints with an axis along the link can be moved to any point on the link (can think of it as at the beginning, middle, or end of the link)