Assignment 1: Forward Kinematics

16-384: Robot Kinematics and Dynamics

Prof. Jeff Ichnowski
Shahram Najam Syed
Yuemin Mao

Contents

1 Overview

Welcome to the first assignment for Robot Kinematics and Dynamics! The previous assignment ensured you were equipped with the essential preparatory materials to commence your deep dive into the course content from a robust starting point. This assignment aims to build a strong foundation in the core principles of kinematics, including:

- Degrees of freedom
- Homogeneous transformations
- Forward kinematics

Additionally, this will be your first opportunity to interact directly with the robots. We'll guide you through the initial steps of communicating effectively with the robotic arms, setting the stage for you to take more control in subsequent assignments.

1.1 Accessing the Robots

The robots are accessible in the Robotics Engineering Lab (REL) and are available for use immediately. Please be mindful of others and share the robotic resources fairly. We ask that you keep your sessions with the robots to a maximum of 30 minutes at a time to ensure that all students have equal opportunity to engage with the technology.

2 Background

This section outlines the essential concepts that will be covered in this assignment. You are expected to be familiar with these terms as they are fundamental to understanding robot kinematics and dynamics. Reference the attached summary of the relevant topics annexed in this document.

2.1 Key Concepts

- **Coordinate Frames**: Understand the use of coordinate frames in robotics, including origin points and orthonormal axes. All frames used will be right-handed.
- **Point and Vector**: Learn the representation of points and vectors within different coordinate frames. Points denote fixed positions in space, while vectors represent quantities like displacements and forces that have both magnitude and direction.
- **Rigid Body and Rigid Motion**: Explore the properties of rigid bodies, which maintain constant distances between points, and understand rigid motions like translations and rotations that can be applied to these bodies.
- **Mathematical Spaces**: Familiarize yourself with $\mathbb{R}^2$ for two-dimensional Cartesian space and $SE(2)$ for the combination of positional and rotational parameters in two dimensions.
- **Workspace**: Analyze the workspace of a robot, which is the set of all points that the robot's end effector can reach through any configuration of its joints.

2.2 Robots and Robot Diagrams

Gain a clear understanding of the structural and operational aspects of robots used in this class, focusing on:

- **Links and Joints**: Links are the rigid bodies making up the robot, and joints are the connections allowing constrained relative motions between these links.
- **Revolute and Prismatic Joints**: Understand the two main types of joints used in the class, where revolute joints allow rotational movement and prismatic joints permit translational movement along an axis.
- **Kinematic Chain**: Comprehend the configuration of a robot as a series of links connected by joints, forming a chain that determines the robot's motion capabilities.
- **Degrees of Freedom**: Determine the degrees of freedom in a system, which represent the number of independent movements allowed in the system.

2.3 Transforms in Robotics

- **Rigid Motions and Transforms in Two Dimensions**: Delve into how rotations and translations are used to describe the motion of points and the transformation between coordinate frames.

2.4 Learning Objectives

Through this assignment, you are expected to:

- Apply the concepts of coordinate frames to define the motion and orientation of robot components.
- Use transformations to calculate the positions of points and orientations of links in various robot configurations.
- Understand the physical limitations and capabilities of robots through the analysis of workspaces and degrees of freedom.

2.5 Course Logistics

In this class you would be using the following websites regularly:

- Canvas: The main course web page. This links to everything relevant to the class (including the next two links).
- Piazza: This is the best way to ask course staff questions, and see course announcements.
- Gradescope: This is where you will be turning in your homework. You should have already been added to the course. If you haven't, please contact the course staff via email or Piazza post.

3 Instructions

- The deadline for this project is 12th September, 2024 09:00 P.M.
- Zip your code into a single file named `<AndrewId>.zip`. See the complete submission checklist at the end, to ensure you have everything. Submit your PDF file to Gradescope.
- Each question (for points) is marked with a **points** heading.
- **Start early!** This homework may take a long time to complete.
- **During submission indicate the answer/page correspondence carefully when submitting on Gradescope.** If you skip a written question, just submit a blank page for it. This makes our work much easier to grade.
- If you have any questions or need clarifications, please post in Piazza or visit the TAs during the office hours.
- Unless otherwise specified, **all units are in radians, meters, and seconds**, where appropriate.

4 Theory Questions

1) **Matrix Analysis Review**

Given the Matrix $A = \begin{bmatrix} -\sin(\theta_1) - \sin(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) \\ \cos(\theta_1) + \cos(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix}$

(1) [1 point] Is A an element of $SO(2)$?

(2) [1 point] For what values of θ_1, θ_2 does $\text{rank}(A) = 1$?

(3) [1 point] What is the inverse of A?

(4) [1 point] Using (2) and (3), what values of θ can you *not* solve $Ax = b$, where x and b are column vectors?

(5) [1 point] When $\text{rank}(A) = 1$, what constraints are placed on b?

2) **Basic Transformations**

Let $t = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \theta = \frac{\pi}{4}$

(1) [1 point] Let H_1 be the homogeneous transformation that translates a point by t. What is H_1?

(2) [1 point] Let H_2 be the homogeneous transformation that rotates a point by θ. What is H_2?

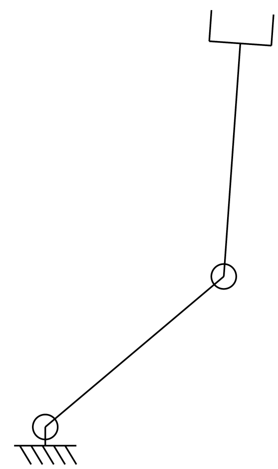
(3) [2 points] Show *algebraically* that $H_1 H_2$ is not equal to $H_2 H_1$.

(4) [1 point] Show *geometrically* that $H_1 H_2$ is not equal to $H_2 H_1$.

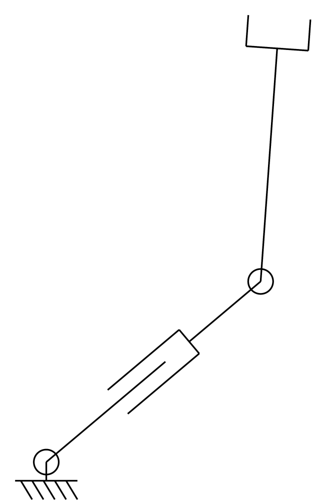
3) **Basic degrees of Freedom**

How many degrees of freedom do each of these robots have (assume $\mathbb{R}^2$)?

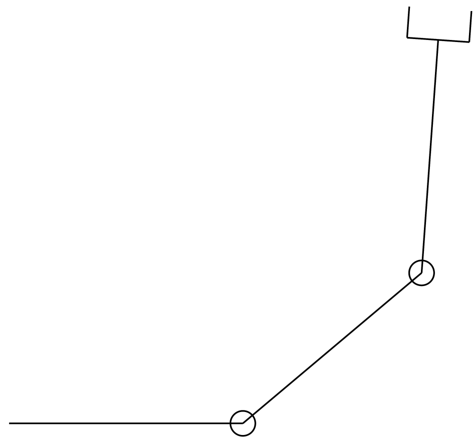
1. [1 point]



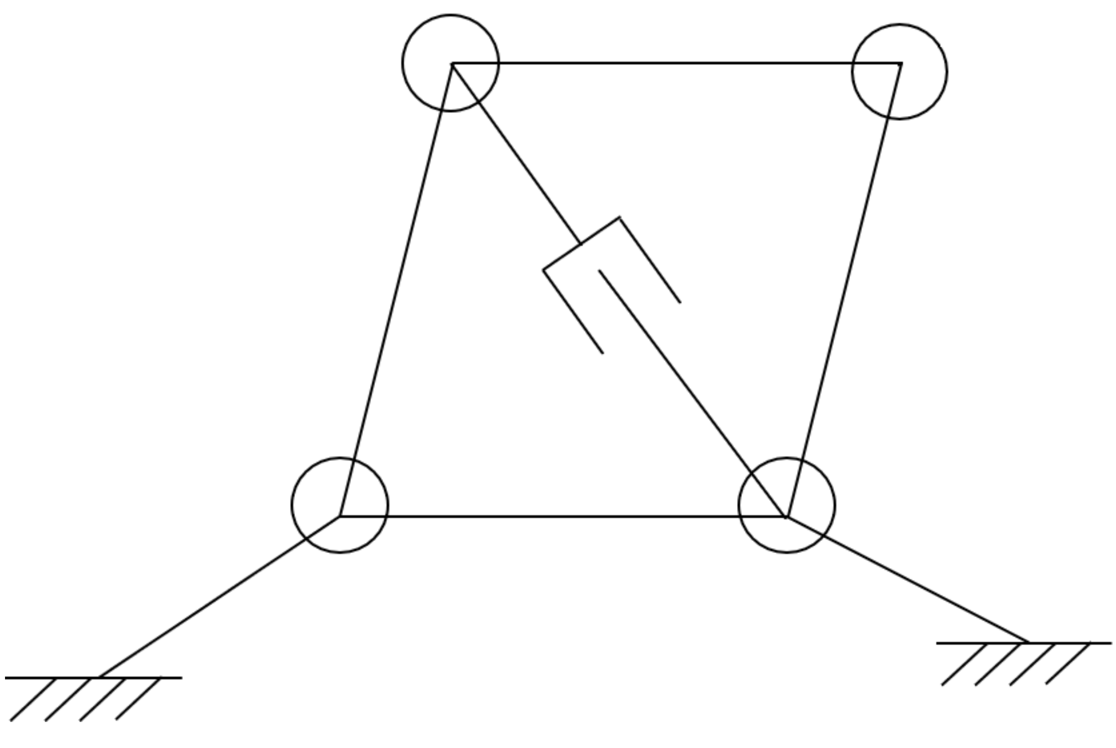
2. [1 point]



3. [1 point]



4. [2 point]



4) **Real-world DOF Determination**

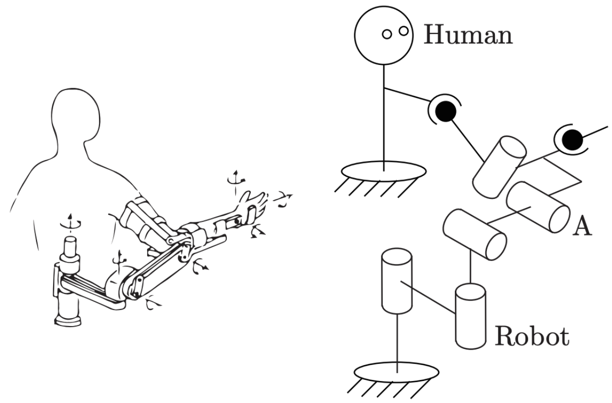
This exercise involves analyzing the degrees of freedom in human arms and their interaction with external systems like vehicles or rehabilitation robots. Understanding these concepts is critical in fields such as biomechanics, robotics, and ergonomic design.

1. **[Points 5] Degrees of Freedom in the Human Arm:** Determine the number of degrees of freedom from your torso to your palm (excluding fingers and thumb). Keep the shoulder joint stationary:
   (a) Add up the degrees of freedom at the shoulder, elbow, and wrist joints.
   (b) With your palm flat on a table and your elbow bent, explore and describe how many degrees of freedom you can utilize without moving the shoulder joint.

2. **[Points 5] Degrees of Freedom While Driving:** Assuming each of your arms has n degrees of freedom, analyze the combined system of your arms and the steering wheel while driving. Consider that your torso is stationary due to a seatbelt, and your hands are fixed on the steering wheel. Calculate and explain the total degrees of freedom available in the arms-plus-steering wheel system under these constraints.

3. **[Points 10] Human-Robot Interaction in Rehabilitation:** Referring to the figure below of a robot used for human arm rehabilitation. Determine the number of degrees of freedom in the chain formed by the human arm coupled with the

rehabilitation robot.



5) **Rotation Matrices**

Let $p = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$

1. [1 point] Let R be the rotation matrix representing a rotation by $\frac{\pi}{4}$. Find R.

2. [1 point] Let p' be the result of applying R to p. Find p'.

3. [1 point] Draw and label a set of axes, and plot p, and p' in this frame.

6) **Inverting Homogeneous Transformations** *5 points*

Given $H_i^j = \begin{bmatrix} R_i^j & d_i^j \\ \mathbf{0} & 1 \end{bmatrix}$, *algebraically* show that $(H_i^j)^{-1} = \begin{bmatrix} (R_i^j)^T & -(R_i^j)^T d_i^j \\ \mathbf{0} & 1 \end{bmatrix}$.

You do not need a formal proof, but you must show your steps clearly.

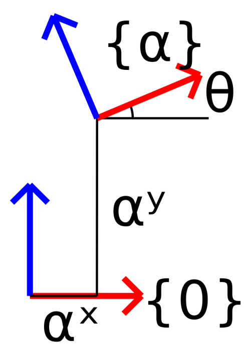
7) **Homogeneous Transformations**

Let $t = \begin{bmatrix} \alpha^x \\ \alpha^y \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$, $\theta = \frac{\pi}{4}$,

1. [1 point] Let T_1 be the homogeneous transformation that translates a point by t. What is T_1?

2. [1 point] Let T_2 be the homogeneous transformation that rotates a point by θ. What is T_2?

3. [5 points] Find H^0_α, using the frames shown in the figure. Then verify your solution by using this matrix to transform the following points p, q and vectors v, u (given in $\{\alpha\}$) to $\{\mathbf{0}\}$)



(i) What is H^0_α?

(ii) $p^\alpha = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

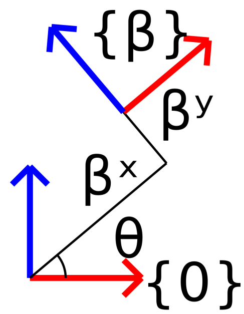
(iii) $q^\alpha = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

(iv) $v^\alpha = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

(v) $u^\alpha = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Ensure that the coordinates make sense in the new representation.

4. [5 point] Find H_β^0, using the frames shown in the figure.



Let $\theta = \frac{\pi}{4}$. Verify your solution by using this matrix to transform the following points p, q and vectors v, u (given in $\{\beta\}$) to $\{\mathbf{0}\}$)

(i) $p^\beta = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

(ii) $q^\beta = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

(iii) $v^\beta = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

(iv) $u^\beta = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

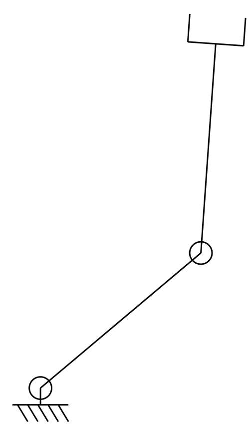
Ensure that the coordinates make sense in the new representation.

5. [1 point] Find H_β^α.

6. [2 points] Compute H^0_α and $(H^0_\alpha)^{-1}$. Verify that $(H^0_\alpha)^{-1}H^0_\alpha = I_{3\times3}$.

8) **Workspace and Frames**

Consider the following robot:



Draw and clearly label the following, **using the frame conventions given in the background for this assignment**.

1. [1 point] The base frame of the entire robot: $\{\mathbf{0}\}$.

2. [1 point] The starting frame for the first link: $\{\mathbf{1}\}$

3. [1 point] The frame at the end of the first link: $\{\mathbf{2}\}$

4. [1 point] The starting frame for the second link: $\{\mathbf{3}\}$

5. [1 point] The end effector frame: $\{\mathbf{4}\}$

6. [3 points] Assume $l_1 > l_2$. On a separate plot, shade the workspace (in $\mathbb{R}^2$), ignoring self-collision. Label the dimensions of this plot.

9) **Forward Kinematics of an RR Arm**

For the robot in the previous question, compute the following in terms of l_1, l_2, θ_1, and θ_2:

1. [1 point] H_1^0

2. [1 point] H_2^1

3. [1 point] H_3^2

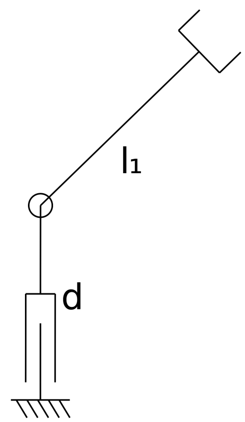
4. [1 point] H_4^3

5. [1 point] H_4^0, the transform from the end effector to the base frame (using the previous transforms).

6. The position of the origin of the end effector frame $\{4\}$, represented in the base frame $\{\mathbf{0}\}$, for the following values of $\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}$. Solving for both "nice" numbers and intermediate numbers will build intuition of RR arm forward kinematics.
   (i) [1 point] $\begin{bmatrix}0\\ 0\end{bmatrix}$
   (ii) [1 point] $\begin{bmatrix}0\\ \frac{\pi}{2}\end{bmatrix}$
   (iii) [1 point] $\begin{bmatrix}\frac{\pi}{2}\\ -\frac{\pi}{2}\end{bmatrix}$
   (iv) [1 point] $\begin{bmatrix}\frac{\pi}{3}\\ \frac{\pi}{2}\end{bmatrix}$

10) **Workspace and Frames of a PR Arm** *8 points*

Repeat problem 8 for the following arm:



Remember to use the frame conventions given in the background!

1. [1 point] The base frame of the entire robot: $\{\mathbf{0}\}$.

2. [1 point] The starting frame for the first link: $\{\mathbf{1}\}$

3. [1 point] The frame at the end of the first link: $\{\mathbf{2}\}$

4. [1 point] The starting frame for the second link: $\{\mathbf{3}\}$

5. [1 point] The end effector frame: $\{\mathbf{4}\}$

6. [3 points] For the workspace, assume the prismatic joint can go from $d = 0$ to $d = d_f$. Draw a separate image for $d_f = l_1$ and $d_f > 2l_1$.

11) **Forward Kinematics of a PR Arm** *9 points*

Repeat problem 9 with the arm in the previous problem.

a. [1 point] H_1^0

b. [1 point] H_2^1

c. [1 point] H_3^2

d. [1 point] H_4^3

e. [1 point] H_4^0, the transform from the end effector to the base frame (using the previous transforms).

f. [4 points] Substitute in the following values for the joint positions $\begin{bmatrix} d \\ \theta \end{bmatrix}$ for question 6 part f. Solving for both "nice" numbers and intermediate numbers will build intuition of PR arm forward kinematics.

(i) $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$

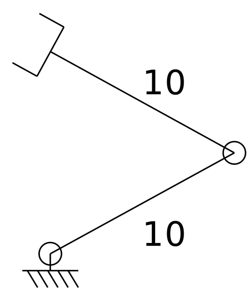
(ii) $\begin{bmatrix} 3 \\ \frac{\pi}{2} \end{bmatrix}$

(iii) $\begin{bmatrix} 1 \\ -\frac{\pi}{2} \end{bmatrix}$

(iv) $\begin{bmatrix} 3 \\ \frac{\pi}{4} \end{bmatrix}$

12) **Singularities**

Consider the following RR arm:



Draw labeled plots. Please draw each part separately.

For which $\begin{bmatrix} x \\ y \end{bmatrix}$ in the workspace can $\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$ have

a. [2 points] Only 1 value?

b. [2 points] Exactly 2 values?

c. [4 points] Infinite values?

5 Code Questions

In these problems, we'll analytically test some of the work from the previous section. Copy the Code Handout folder to a location of your choice.

1) Forward Kinematics *25 points*

Begin by navigating to the `ex_01` directory containing your Python scripts.

In this exercise, you must complete the `forward_kinematics_RR.py` script. When implemented correctly, this script will compute the forward kinematics for an RR robotic arm using specified joint angles `theta_1` and `theta_2`, along with link lengths `l1` and `l2` (these values will be provided within the script).

You will define transformation matrices in this file for several frames:

- `H_1_0`: Transformation from the base frame $\{\mathbf{0}\}$ to the first link's base $\{\mathbf{1}\}$.
- `H_2_1`: Transformation from the first link's base $\{\mathbf{1}\}$ to the first link's end $\{\mathbf{2}\}$.
- `H_3_2`: Transformation from the first link's end $\{\mathbf{2}\}$ to the second link's start $\{\mathbf{3}\}$.
- `H_4_3`: Transformation from the second link's start $\{\mathbf{3}\}$ to the end effector $\{\mathbf{4}\}$.
- `H_4_0`: Transformation from the base link $\{\mathbf{0}\}$ to the end effector $\{\mathbf{4}\}$.

The base frame for the entire robot is denoted as `frame_0`. The subsequent frames are defined at strategic points along the robotic arm, culminating in the end effector's frame, `frame_4`.

To ensure your implementation is correct:

- Initially test with simple cases like angles of 0 or $\pi/2$. Verify that the transformations align points correctly from the individual link frames to the base frame.
- Utilize the provided `sample_path.py` script, which simulates the robot's movement and plots the end effector's trajectory generated by your kinematics against the ground-truth end-effector positions. Confirm that these trajectories align closely.

2) Workspace Analysis *25 points*

Begin by navigating to the `ex_02` directory containing your Python scripts.

In this exercise, you will conduct a workspace analysis for a PR (Prismatic-Rotary) arm configuration.

First, run the `workspace_analysis_RR.py` script. This script computes the reachable workspace of a RR (Rotary-Rotary) arm by performing a complete sweep of the joint angles, given link lengths l_1 and l_2. Observe the generated figure which shows the end-effector positions and compare this to your analytical results from problem 8.

Your task is to develop a similar analysis for a PR arm configuration by creating a new script named `workspace_analysis_PR.py`. Utilize the RR analysis script as a reference to build upon. Ensure that the prismatic joint ranges from $d = 0$ to $d = d_f$, where d_f is the fully extended length of the prismatic joint. The function should accept the link length l_1 as input.

The expected deliverables for this task are:

- First figure: Set $l_1 = 1$, and $d_f = 1$. Save this as `workspace_pr_1.png`.
- Second figure: Set $l_1 = 0.5$, and $d_f = 2$. Save this as `workspace_pr_2.png`.

Verify that the generated workspaces align with the predictions made in problem 10 of your theoretical analysis.

6 Hands-On Questions

1) **Tracing Objects with the Franka Emika Arm** *50 points*

For this part of the homework, you will work with the Franka Emika Panda robot arm to trace and plot "rkd". Start by connecting to the robot, downloading your code, and opening the `ex_03` directory there.

1. **Franka Arm Connection**

   To establish a connection with the Franka Emika Panda robot arm, follow Franka Connection Tutorial.

2. **Testing Forward Kinematics Implementation**

   Import the forward kinematics function you developed earlier to `trace_object.py` and run the script. This script will move the robot to its initial position, set it in a selective guidance mode for 60 seconds, and start a pyplot window. After "Prepare the marker and press 'g' to close the gripper" is printed out, place a marker at the center of the gripper and press "g" to grasp the marker. Move the robot so that the marker points at the start of letter "r" on the white board, then press the spacebar to start recording the robot's motion. Move the arm manually to trace the "rkd" shape. Meanwhile, a live plot displaying the end effector position computed by your FK function will appear in the pyplot window. Once completed, press the spacebar again to stop recording and the motion data will be auto saved. If the plot represents "rkd" in the correct orientation, your implementation is correct, save the plot. Otherwise, revisit your forward kinematics function.

3. **Deliverables**

   - The motion data, saved in a `trace_data.pkl` file (pickle format). It stores the recorded joint angles and transforms.
   - Include the plot generated in 2 in your homework writeup.

Note: For this exercise, only two joints (joint 1 and joint 3) of the Franka arm are active to simulate an RR arm configuration, the remaining five joints are "locked" by applying high impedance. This setup greatly simplifies the control and kinematics of the Franka arm. For this reason, the "rkd" shape you get in the plot is expected to be a little distorted compared to the one on the white board.

2) **Submission**

To submit, run `create_submission.py`. This script checks that your files run without error and performs basic output consistency checks. Note, this script does not grade your submission but ensures it is prepared correctly. The script will generate a file called `<andrew_id>_hw1.zip`. Upload this to Canvas to complete your submission.

7 Submission Checklist

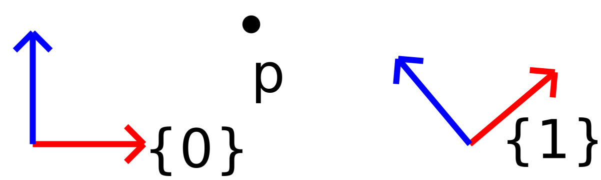
- ☐ Create a PDF of your answers to the written questions with the name <andrew_id>.pdf and submit it to Gradescope.
- ☐ Run `create_submission.py` in the python terminal.
- ☐ Upload <andrew_id>.zip to Gradescope.
- ☐ Coding questions are autograded, with no limit on number of submissions within the mandated deadline.

8 Summary of concepts

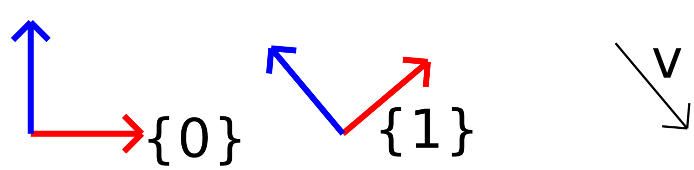
8.1 Preliminary Terms

Coordinate Frame: A coordinate frame consists of a special point called the *origin* and some number of orthonormal[1] axes. Frames provide a reference for measurements of positions and rotations of a body or point. Frame $\mathbf{i}$ is denoted $\{\mathbf{i}\}$. For the purposes of this class, all frames must be right-handed.

Point: A point is a fixed position in space. Its coordinates depend on the frame describing it. Point $\mathbf{p}$ relative to $\{\mathbf{0}\}$ is denoted $\mathbf{p^0}$. For example, in the figure below $\mathbf{p^0} = \left[\begin{smallmatrix} 2 \\ 1 \end{smallmatrix}\right]$ and $\mathbf{p^1} = \left[\begin{smallmatrix} -1 \\ 2 \end{smallmatrix}\right]$.



Vector: A direction and magnitude that is free in space. Among other things, vectors can describe offsets from a point, forces, or velocities. Just like a point, the direction of the vector depends upon in which frame it is referenced, but the magnitude remains constant in all frames. For example, in the figure below $\mathbf{v^0} = \left[\begin{smallmatrix} .7071 \\ -.7071 \end{smallmatrix}\right]$ and $\mathbf{v^1} = \left[\begin{smallmatrix} 0 \\ -1 \end{smallmatrix}\right]$ both have a magnitude of 1.



Rigid Body: A collection of points where the distance between two points and the handedness of the points remains constant while the collection undergoes a displacement.

Rigid Motion: A translation, rotation, or combination of the two that can be applied to a body without changing the distance between any two points in the body or changing

[1]Orthonormal axes are:
- unit vectors (length 1)
- mutually orthogonal (perpendicular to each other)

the handedness of the points.

$\mathbb{R}^2$: Two-dimensional Cartesian space; can be parameterized by x and y.

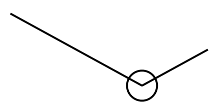
SE(2): It is both 1) the space corresponding to position and orientation of a rigid body in two dimensions and 2) the set of rigid motions in two dimensions. It is usually parameterized by x, y, and θ.

Workspace: The set of all points $\mathbf{p}$ such that there exists some joint configuration which places a robot's end effector at $\mathbf{p}$.

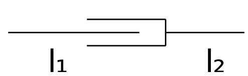
8.2 Robots and Robot Diagrams

For the purpose of this class, a **robot** or a **linkage** is a combination of *links* (which are rigid bodies) and *joints*, which connect two links with a certain constraint on relative motion. Links are generally represented by straight lines in robot diagrams. The two categories of joints we will use in this class are *rotational* and *prismatic* joints.

Revolute Joint: Creates an angular offset between two adjacent links. This offset is typically notated as θ, with a subscript that matches the joint index or name. Positive rotations go from the x axis towards the y axis (i.e., they follow the right hand rule). They are drawn as follows in robot diagrams:



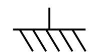
Prismatic Joint: Creates a translational offset between two adjacent links along a single axis. The notation for this offset varies. We will refer to this as d with a subscript to match the joint index or name, but in a robot with both rotational and prismatic joints sometimes $\theta or q$ is used for any joint. They are drawn as follows in robot diagrams. Note that this figure represents two separate links with one prismatic joint in between with a total length is $l_1 + l_2 + d$; however, depending on the system, l_1 and l_2 may be omitted and just combined into d. You can use either convention, but be sure to label any diagrams clearly.



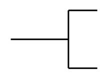
Kinematic Chain: An n-joint kinematic chain is a robot consisting of $n + 1$ links and n joints, connected in series. If the robot is attached to the world, the first link is called the base link. Often, the first link has zero length and is omitted. Many of our examples will have a zero length base link or "no base link." Joint i connects links $i - 1$ and

i and joint i moves link i. It is also possible to have a free floating robot of $n+1$ links connected by n joints.

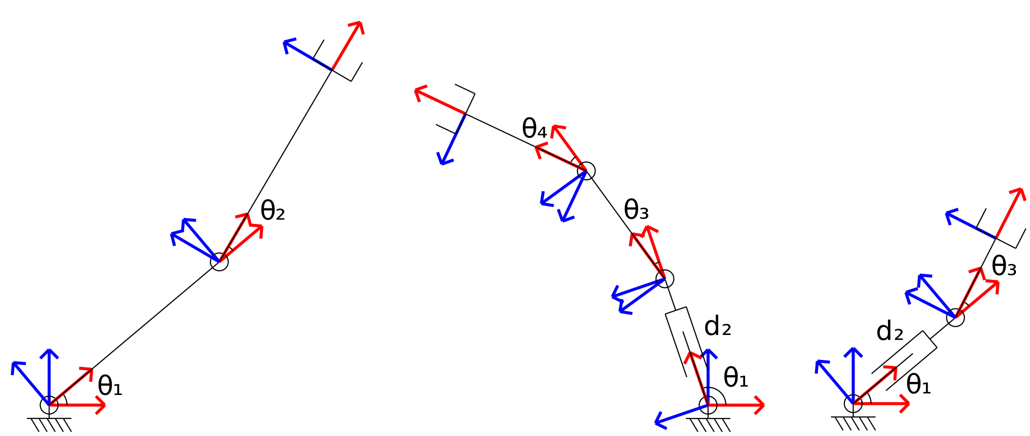
Fixed Base: Drawn as follows, this represents a rigid point of attachment to the world.



End Effector: Drawn as follows, this represents a gripper or other end effector attached to the end of a robot link (or directly to the output of a joint). We generally do not consider the end effector to add a degree of freedom to the robot, as it does not take part in the overall kinematics.



Frame Conventions: We define a *base* frame as a coordinate frame at the point where the robot is fixed to the world, labeled $\{0\}$ (NOTE: in the OLI material, this frame is omitted). We also define a coordinate frame at the base of each link (the *proximal* frame for that link). Optionally, we can also define a frame at the *end* of each link, where the next link is attached by a joint; this is the *distal* frame. Note that this is also omitted in the OLI material, but is used in this assignment to help define the order of all intermediate transformations. The convention in this class will be that the x axes of these frames points from the base to the end of the link. End effectors of planar robots will have a frame defined at the attachment point to the link, and the x axes will point *out* of the gripper (i.e., this frame will usually match the distal frame of the connecting link). Positive angles of rotation will be counter-clockwise (from the x axis towards the y axis). Below are several simple robots demonstrating these conventions.



8.3 Degrees of Freedom

In robot kinematics, a system's *degrees of freedom*, or *DOF*, is the number of parameters needed to completely define a particular configuaration. For example:

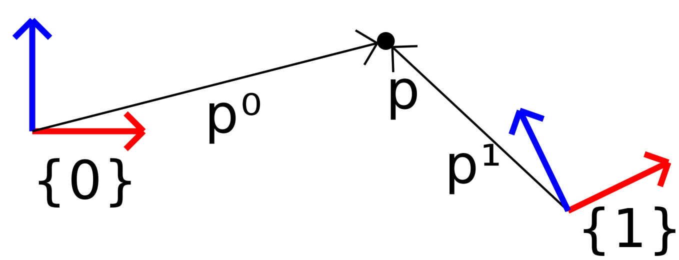
- A point in two dimensions (i.e., $\mathbb{R}^2$) has 2 DOF, since $\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right]$ completely defines the point's configuration.
- A rigid body in the plane has 3 DOF, since $\left[\begin{smallmatrix} x \\ y \\ \theta \end{smallmatrix}\right]$ completely defines its position and orientation.
- An fixed-base n-joint kinematic chain in SE(2) has n DOF, since n parameters for the joints completely define the robot's configuration. If the robot isn't attached to the base frame, the base link can move freely (in x, y, and θ), and the robot has $n+3$ DOF.

8.4 Rigid Motions / Transforms in Two Dimensions

There are two basic types of motions/transformation we are concerned with: rotations and translations. Rotations are represented with matrices, and translations are represented by vectors. These can be combined into compound motions/transformations.

Rotations and translations are used for two different concepts. The first is the motion of a point or rigid body within a coordinate frame:

The second concept is a *transform* between two coordinate frames. In other words, if you have the coordinates of a point or vector in $\{\mathbf{0}\}$, what are the coordinates of this same point or vector in $\{\mathbf{1}\}$:



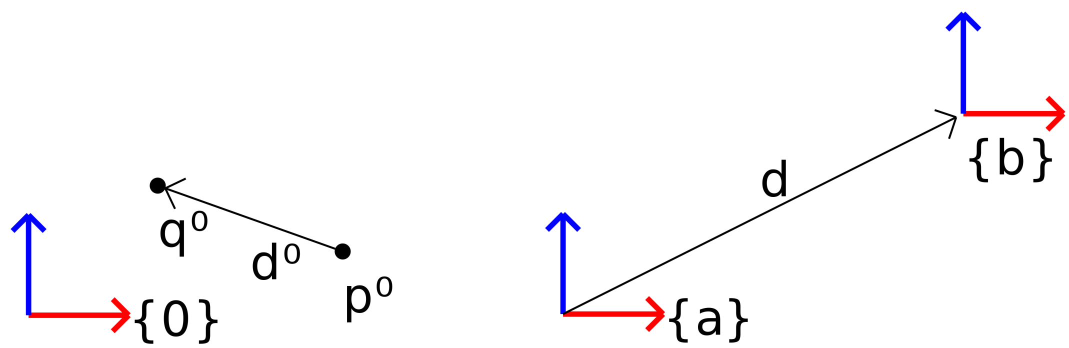
Notationally, transformations follow the convention of a superscript and a subscript, to describe which frames are transformed between. A transform $\mathbf{T}_{\mathbf{b}}^{\mathbf{a}}$ between two frames has several interpretations (notice the order of the superscripts in each case).

- The motion that moves $\{\mathbf{a}\}$ to $\{\mathbf{b}\}$.
- $\{\mathbf{b}\}$ as represented in $\{\mathbf{a}\}$.

- The transform that changes the representation of a point or vector from $\{\mathrm{b}\}$ to $\{\mathrm{a}\}$.

Note that the motion between two frames is the opposite of the transform that changes point representation between these frames. This is something which can often lead to errors! A good rule of thumb is that the top number always represents what frame this motion/transform is represented in. Here, the transform is in the frame of reference $\{\mathrm{a}\}$. Below, we explicitly consider translations, rotations, and combined motions.

8.4.1 Translation

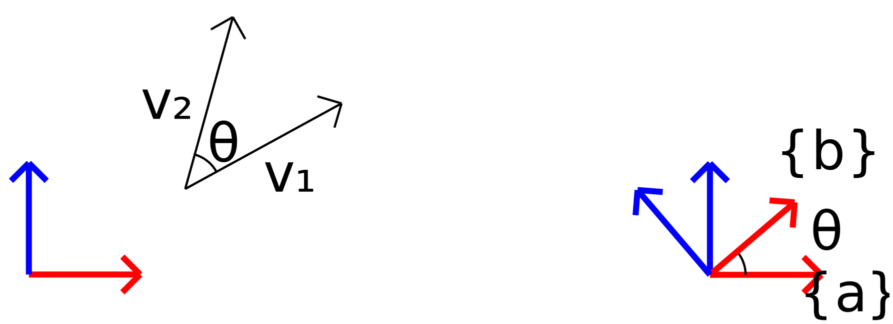


A translation $\mathbf{d} = \left[\begin{smallmatrix}\Delta x \\ \Delta y\end{smallmatrix}\right]$ expresses a change in position of a point or rigid body, or a position offset between coordinate frames. In terms of rigid motions, translating a point is the geometric addition of two positions; above left, $\mathbf{q}^0 = \mathbf{p}^0 + \mathbf{d}^0$. This is the motion that moves $\mathbf{p}$ to $\mathbf{q}$ in $\{0\}$.

For transformations, $\mathbf{d}_\mathrm{b}^\mathrm{a}$ is the offset from $\{\mathrm{a}\}$ to $\{\mathrm{b}\}$ (above right, $\mathbf{d} = \mathbf{d}_\mathrm{b}^\mathrm{a} = \left[\begin{smallmatrix}4 \\ 2\end{smallmatrix}\right]$). In other words, $\mathbf{d}_\mathrm{b}^\mathrm{a}$ is:

- The motion that moves (the origin) of $\{\mathrm{a}\}$ to the origin of $\{\mathrm{b}\}$.
- The representation of $\{\mathrm{b}\}$ in $\{\mathrm{a}\}$ (e.g., the coordinates of the origin of $\{\mathrm{b}\}$ in $\{\mathrm{a}\}$).
- The transform that changes the representation of a point p from $\{\mathrm{b}\}$ to $\{\mathrm{a}\}$: $\mathbf{p}^\mathrm{a} = \mathbf{d}_\mathrm{b}^\mathrm{a} + \mathbf{p}^\mathrm{b}$.

8.4.2 Rotation



A *rotation matrix* $\mathbf{R}$ represents rigid body rotations or rotational offsets between coordinate frames. In 2 dimensions, the rotation matrix corresponding to a rotation of θ is

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

For rigid motions, this is the rotation of a body or vector by θ; above left, the vector $v_2 = \mathbf{R}(\theta)v_1$.

For transformations, $\mathbf{R}^{\mathbf{a}}_{\mathbf{b}}$ is the rotation from $\{\mathrm{a}\}$ to $\{\mathrm{b}\}$ (above right, $\mathbf{R}^{\mathbf{a}}_{\mathbf{b}} = \mathbf{R}(\theta)$). $\mathbf{R}^{\mathbf{a}}_{\mathbf{b}}$ can be described as:

- The motion that rotates $\{\mathrm{a}\}$ to $\{\mathrm{b}\}$.
- The representation of $\{\mathrm{b}\}$ in $\{\mathrm{a}\}$; $\mathbf{R}^{\mathbf{a}}_{\mathbf{b}}$ can be found by expressing the axes of $\{\mathrm{b}\}$ in $\{\mathrm{a}\}$ as column vectors.
- The rotation that changes the representation of a vector v from $\{\mathrm{b}\}$ to $\{\mathrm{a}\}$: $\mathbf{v}^{\mathbf{a}} = \mathbf{R}^{\mathbf{a}}_{\mathbf{b}}\mathbf{v}^{\mathbf{b}}$.

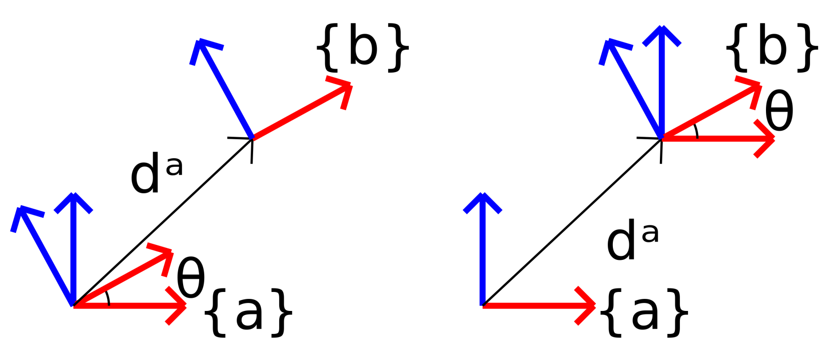
Furthermore, as shown in lecture, for any rotation matrix $\mathbf{R}$, $(\mathbf{R})^{-1} = (\mathbf{R})^{\top}$.

8.4.3 Compound Motions: Homogeneous Transforms

Any rigid motion/transform between two frames can be expressed by combining a single rotation and translation. A *homogeneous transform* is a single matrix which allows us to express this combined rotation and translation:

$$\mathbf{H}^{\mathbf{a}}_{\mathbf{b}} = \begin{bmatrix} \mathbf{R}^{\mathbf{a}}_{\mathbf{b}} & \mathbf{d}^{\mathbf{a}}_{\mathbf{b}} \\ 0\,0 & 1 \end{bmatrix}$$

As with previous notation, $\mathbf{H}^{\mathbf{a}}_{\mathbf{b}}$ is a motion from $\{\mathrm{a}\}$ to $\{\mathrm{b}\}$, or equivalently the transform that changes the coordinates of points from $\{\mathrm{b}\}$ to $\{\mathrm{a}\}$.



An important point here is that the *order* and *frame of reference* which you apply the rotation and transforms here matter. The following two descriptions are equivalent (and describe $\mathbf{H}^{\mathbf{a}}_{\mathbf{b}}$):

- First apply the rotation relative to $\{\mathrm{a}\}$, and then apply the translation relative to $\{\mathrm{a}\}$ (see above left).

- First apply the translation relative to $\{a\}$ and then apply to rotation relative to the *new* frame resulting from this translation (see above right).

Note that the other two potential descriptions - rotation then translation in the new frame, or translation then rotation in $\{a\}$ - are both equivalent but **do not describe** $\mathrm{H}^{\mathrm{a}}_{\mathrm{b}}$.

8.4.4 Points, Vectors, and Homogeneous Transforms

When using Homogeneous Transforms to operate on points and vectors, we must add an extra coordinate to these values. Points are padded with a 1, so $\mathbf{p}^{\mathbf{i}} = \left[\begin{smallmatrix} p_x \\ p_y \end{smallmatrix}\right]$ is now $\left[\begin{smallmatrix} p_x \\ p_y \\ 1 \end{smallmatrix}\right]$, whereas vectors are padded with a 0, so $\mathbf{v}^{\mathbf{i}} = \left[\begin{smallmatrix} v_x \\ v_y \end{smallmatrix}\right]$ is now $\left[\begin{smallmatrix} v_x \\ v_y \\ 0 \end{smallmatrix}\right]$. This allows the matrix dimensions to match when multiplying by such transforms.

If $\mathbf{H}^{\mathbf{0}}$ describes a motion in $\{\mathbf{0}\}$, then the point $\mathbf{q}^{\mathbf{0}}$ resulting from this motion starting at point $\mathbf{p}^{\mathbf{0}}$ is:

$$\begin{aligned} \mathbf{q}^{\mathbf{0}} &= \mathbf{H}^{\mathbf{0}} \begin{bmatrix} \mathbf{p}^{\mathbf{0}} \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}^{\mathbf{0}} & \mathbf{d}^{\mathbf{0}} \\ 0\,0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}^{\mathbf{0}} \\ 1 \end{bmatrix} \\ &= \mathbf{R}^{\mathbf{0}}\mathbf{p}^{\mathbf{0}} + \mathbf{d}^{\mathbf{0}} \end{aligned}$$

Note that the 1 we added to the bottom of the point results in the addition of $\mathbf{d}^{\mathbf{0}}$; because free vectors have a 0 added, they are only affected by the rotation portion of $\mathbf{H}^{\mathbf{0}}$.

Using $\mathbf{H}^{\mathbf{a}}_{\mathbf{b}}$ as a transformation between coordinate representations of points/vectors, we have

$$\left[\begin{smallmatrix} \mathbf{p}^{\mathbf{a}} \\ 1 \end{smallmatrix}\right] = \mathbf{H}^{\mathbf{a}}_{\mathbf{b}} \left[\begin{smallmatrix} \mathbf{p}^{\mathbf{b}} \\ 1 \end{smallmatrix}\right] \text{ and } \left[\begin{smallmatrix} \mathbf{v}^{\mathbf{a}} \\ 0 \end{smallmatrix}\right] = \mathbf{H}^{\mathbf{a}}_{\mathbf{b}} \left[\begin{smallmatrix} \mathbf{v}^{\mathbf{b}} \\ 0 \end{smallmatrix}\right].$$

8.4.5 Inverting $\mathrm{H}^{\mathrm{a}}_{\mathrm{b}}$

The inverse of $\mathbf{H}^{\mathbf{a}}_{\mathbf{b}}$ is $\mathbf{H}^{\mathbf{b}}_{\mathbf{a}}$, representing the opposite motion, or a reverse transform between the two frames.

To invert $\mathbf{H}^{\mathbf{a}}_{\mathbf{b}}$, we could perform the matrix inversion manually, but there is a faster method using the properties of $\mathbf{H}^{\mathbf{a}}_{\mathbf{b}}$. First, recall that $\mathbf{p}^{\mathbf{a}} = \mathbf{R}^{\mathbf{a}}_{\mathbf{b}}\, \mathbf{p}^{\mathbf{b}} + \mathbf{d}^{\mathbf{a}}_{\mathbf{b}}$. Solving for $\mathbf{p}^{\mathbf{b}}$, we can see that $\mathbf{p}^{\mathbf{b}} = (\mathbf{R}^{\mathbf{a}}_{\mathbf{b}})^\top \mathbf{p}^{\mathbf{a}} - (\mathbf{R}^{\mathbf{a}}_{\mathbf{b}})^\top \mathbf{d}^{\mathbf{a}}_{\mathbf{b}}$. Therefore, given $\mathbf{R}^{\mathbf{a}}_{\mathbf{b}}$ and $\mathbf{d}^{\mathbf{a}}_{\mathbf{b}}$, $\mathbf{R}^{\mathbf{b}}_{\mathbf{a}} = (\mathbf{R}^{\mathbf{a}}_{\mathbf{b}})^\top$ and $\mathbf{d}^{\mathbf{b}}_{\mathbf{a}} = -(\mathbf{R}^{\mathbf{a}}_{\mathbf{b}})^\top \mathbf{d}^{\mathbf{a}}_{\mathbf{b}}$. This finally gives us that

$$(\mathbf{H}^{\mathbf{a}}_{\mathbf{b}})^{-1} = \mathbf{H}^{\mathbf{b}}_{\mathbf{a}} = \begin{bmatrix} (\mathbf{R}^{\mathbf{a}}_{\mathbf{b}})^\top & (-(\mathbf{R}^{\mathbf{a}}_{\mathbf{b}})^\top \mathbf{d}^{\mathbf{a}}_{\mathbf{b}}) \\ (\mathbf{0})^\top & 1 \end{bmatrix}$$

Inspection can show that this holds true for $\mathbf{v}^{\mathbf{a}}$ and $\mathbf{v}^{\mathbf{b}}$ as well.

8.4.6 Composing Transforms

Homogeneous transforms are composable. For example:

$$\mathbf{H_2^0} = \mathbf{H_1^0 H_2^1}$$

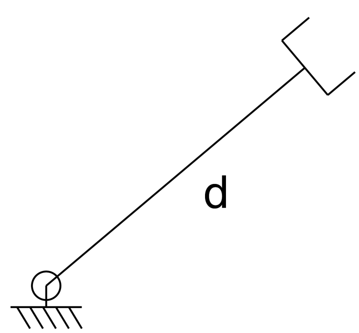
Note that in expressions like $\mathbf{p^1} = \mathbf{H_0^1 p^0}$, the superscripts "cancel" out, which gives you a quick check for your systems of equations. However, note that multiplication of transformations is **not** commutative and this cancellation only works diagonally one way: $\mathbf{H_2^0} \neq \mathbf{H_2^1 H_1^0}$. Using superscripts for the frame of points and vectors will help enforce this consistency.

Furthermore, $\left(\mathbf{H_i^j}\right)^{-1} = \mathbf{H_j^i}$, so if we are given $\mathbf{H_1^0}$, $\mathbf{H_1^2}$, and $\mathbf{H_3^2}$, we can find $\mathbf{H_3^0} = \mathbf{H_1^0}(\mathbf{H_1^2})^{-1}\mathbf{H_3^2}$.

8.5 Forward Kinematics

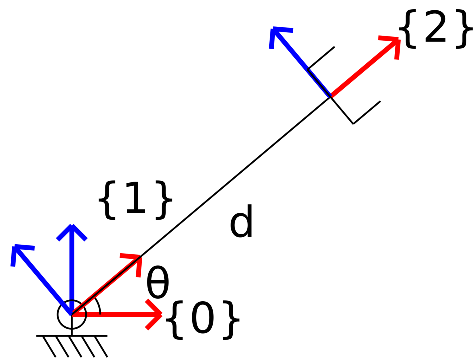
Forward kinematics is the idea of taking a robot's input parameters (joint angles, link lengths, etc.) and finding the position of the end effector. We can do this by writing homogeneous transforms that use the input parameters as variables in order to find the final end effector position. In effect, we want to define frames at each link and joint and describe their differences in terms of the robot parameters.

Let us consider the single link R arm of length d.



We wish to find the end effector position. First, we can find it analytically by noting that $x = d\cos(\theta)$ and $y = d\sin(\theta)$. In vector form, $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} d\cos(\theta) \\ d\sin(\theta) \end{bmatrix}$. While finding the forward kinematics algebraically will always work, it definitely gets more error prone with larger systems, and when composing multiple arms. The way around this is using homogeneous transforms.

We define one frame at the base, the base frame, usually labeled $\mathbf{0}$. Let $\{\mathbf{1}\}$ be another frame at the base of the link, but rotated by θ. Finally, let $\{\mathbf{2}\}$ be based at the end of the link, such that $\mathbf{p_e^2} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ in that frame is the end-effector position.



Note that:

- $\mathbf{H_1^0} = \begin{bmatrix} \mathbf{R}_\theta & \mathbf{0} \\ (\mathbf{0})^\top & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- $\mathbf{H_2^1} = \begin{bmatrix} 1 & 0 & d \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Therefore, $\mathbf{H_2^0} = \mathbf{H_1^0}\mathbf{H_2^1} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & d\cos(\theta) \\ \sin(\theta) & \cos(\theta) & d\sin(\theta) \\ 0 & 0 & 1 \end{bmatrix}$. Lastly, we can see that $\mathbf{H_2^0}\mathbf{p_e^2} = \mathbf{H_2^0}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} d\cos(\theta) \\ d\sin(\theta) \end{bmatrix}$, which matches our analytical solution.