Assignment 2: Jacobians

Robot Kinematics and Dynamics

Prof. Howie Choset

Contents

1 Overview

This assignment reinforces the following topics:

- Jacobians

2 Background

2.1 The Jacobian

The Jacobian transforms differential joint motions to differential changes in the workspace position and orientation of the robot. This means that it can be used to determine the instantaneous linear and angular velocity of a point on the robot given the current joint angles and joint velocities.

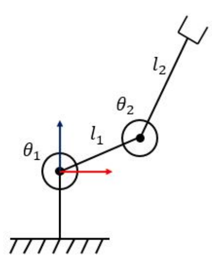
More formally, the Jacobian is a matrix which provides a linear mapping between joint velocities and motion of a coordinate frame attached to the robot (often the end effector frame). Denoted $\mathbf{J}$, we find it by taking the derivative of the forward kinematics to that frame. To be more concrete, the Jacobian fulfills the following expression

$$\dot{X} = J\dot{\Theta}$$

where $\dot{X}$ is the velocity of the frame in consideration ($\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$ or $\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$ depending on the workspace you are considering) and $\dot{\Theta} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_n \end{bmatrix}$.

Incidentally, this also means the dimension of the Jacobian for a robot with n joints whose workspace is SE(2) is $3 \times n$. Furthermore, we generally refer to $\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$ as the *linear velocity*, whereas $\dot{\theta}$ is generally referred to as the *angular velocity*[1].

First, we need the forward kinematics of a generic RR arm. Let f be the general expression for the forward kinematics of a given arm. We often denote f_x as the forward kinematics with respect to the base frame's x axis. Consider the following robot.



The forward kinematics (from the depicted base to end effector frame) of this arm are

$$\begin{aligned} f_x &= l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ f_y &= l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \\ f_\theta &= \theta_1 + \theta_2 \end{aligned}$$

[1] angular velocity is a measure of the speed of rotation, and has units radians/second

A note on finding the angular component of the forward kinematics in 2D: this can generally be done by inspection; if joint i is a rotational joint, θ_i is added, and if it is prismatic, it does not contribute to the end effector orientation.

Alternatively, if one computes the full homogeneous transform $H^{\texttt{base}}_{\texttt{end effector}}$, the translation components will give f_x and f_y, and the rotation matrix component can be used to extract f_θ given the definition of a rotation matrix $\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$.

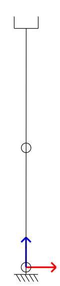
We compute the Jacobian by taking the partial derivatives of the forward kinematics with regard to each component. For example, the partial derivative of f with respect to a variable x is computed by treating all the other components as constant, and taking the first order derivative of f with respect to x. This is denoted $\frac{\partial f}{\partial x}$.

To find $\mathbf{J}$ (for the robot in the previous figure), we take the partial derivative of the forward kinematics with regard to each joint angle[2].

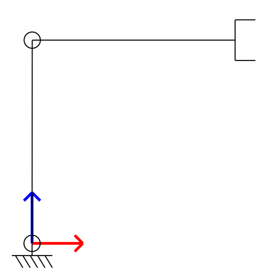
$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_x}{\partial \theta_1} & \frac{\partial f_x}{\partial \theta_2} \\ \frac{\partial f_y}{\partial \theta_1} & \frac{\partial f_y}{\partial \theta_2} \\ \frac{\partial f_\theta}{\partial \theta_1} & \frac{\partial f_\theta}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} -l_1\sin(\theta_1)-l_2\sin(\theta_1+\theta_2) & -l_2\sin(\theta_1+\theta_2) \\ l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2) & l_2\cos(\theta_1+\theta_2) \\ 1 & 1 \end{bmatrix}$$

2.2 Singularities

Informally, a *singularity* is a joint configuration in which a serial mechanism loses a degree of freedom. For example, the end effector of an RR arm in the following configuration can move instantaneously in the x direction, but not in the y direction.



However, in the following configuration, which is away from a singularity, the end effector can move instantaneously in any planar direction via some differential joint motion.



[2]Typically, for 2-DOF systems, we only consider the x and y terms of the Jacobian; however, this will be described in each question.

More formally, the singularity concept can be tied to the **rank** of the Jacobian. The rank of a matrix is the dimension of the space spanned by its rows (or columns). Therefore, for the Jacobian, the rank is the dimension of the space of possible velocities that can be achieved at the given configuration. Computing the Jacobian (considering x and y rows only) for these two configurations gives the following:

$$\begin{bmatrix} (-l_1-l_2) & -l_2 \\ 0 & 0 \end{bmatrix}$$

and

$$\begin{bmatrix} -l_1 & 0 \\ l_2 & l_2 \end{bmatrix}.$$

For the first (singular) configuration, the vectors defined by the columns are parallel - they both point in the -x direction - and therefore they span a 1 dimensional space. For the second, the vectors defined by the columns are at different angles ($\pi/4$ rad or 45 degrees), and therefore span a 2 dimensional space.

Using this terminology, a singular configuration can be thought of as one where the Jacobian loses rank compared to other configurations of the robot. Or as we intuitively noted above, where the kinematics are instantaneously more limited.

2.3 Forces, Torques, and Wrenches

The derivation of the Jacobian is based on differential motions, and therefore naturally relates velocities. However, it also provides a relationship between forces the robot can apply and joint torques. Before defining this relationship, we should formalize the definition of these terms.

A **torque** or **moment** is a force that causes rotation. Whereas a force is applied *along* a line, a moment can be thought of as being applied *about* a line – i.e., causing an object to rotate about that line.

The first concept we will introduce is that of a **wrench**. A wrench is a generalized notion of force. A wrench is a force applied to a body along a given line, and a moment about that line (a torque). In this class, we will represent a wrench by a column vector with the x and y force components followed by the moment: $\begin{bmatrix} f_x \\ f_y \\ \tau \end{bmatrix}$.

During the course, we will often speak of the wrench that is applied to the robot joints as a *joint torque*, and the wrench applied by the end effector to the world or by gravity to the links as the *end effector force* or *force due to gravity*. Technically, these are all wrenches, although due to the prevalence of revolute joints, you will often hear phrases such as "the joint torques which cause this end effector force."

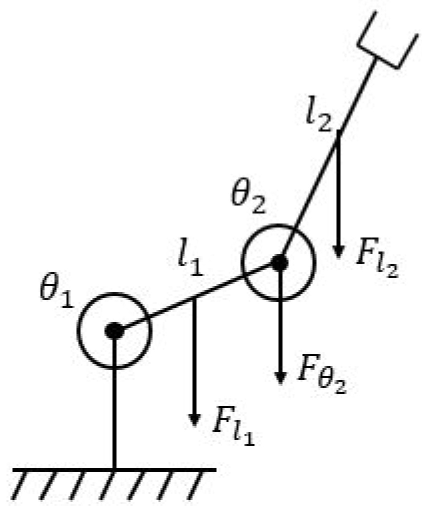
2.4 Jacobian Transpose

A second relationship that the Jacobian provides is between the wrench applied by or on the robot and the joint torques. More formally, the transpose of the Jacobian $(\mathbf{J})^\top$ provides a linear mapping between the wrench at a given location on the robot and the joint torques which generate this wrench:

$$\begin{bmatrix} \tau_{\theta_1} \\ \cdots \\ \tau_{\theta_n} \end{bmatrix} = (\mathbf{J})^\top \begin{bmatrix} f_x \\ f_y \\ \tau \end{bmatrix}.$$

Here, the wrench $\begin{bmatrix} f_x \\ f_y \\ \tau \end{bmatrix}$ is applied at the point that J was defined from. For example, the Jacobian derived from the kinematic map to the end effector can be used to compute the joint torques used to apply a given force at the end effector (e.g., lift an object of a given weight). Also, the Jacobian derived from the kinematic map to the center of mass of a link can be used to compute the joint torques necessary to counteract the effect of gravity on that link.

Joint torques can be added together to generate multiple forces due to the principle of superposition. This idea can be used to make a robot counteract the effect of gravity, making the robot seem weightless. To do this, one must simply compute the required torques to counteract the effect of each center of mass and then add these torques together. For instance, in the figure below, the Jacobian is calculated to the center of mass of l_1, l_2, and θ_2. Using these Jacobians, the joint torques needed to counteract each force are calculated. Lastly, for each joint, the calculated torques are added together. $\tau = \tau_{l_1} + \tau_{\theta_2} + \tau_{l_2}$.

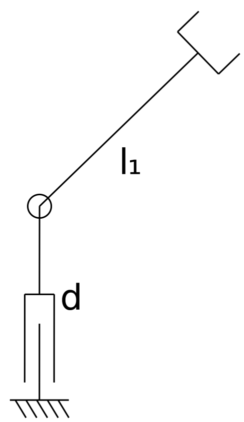


3 In Class Question

The following question will be done in class, as a part of a group. Your group's answer will still need to be turned in with the rest of your assignment, however unlike the rest of the work this is allowed to be done in groups.

1) Jacobian

Given the PR Arm illustrated below from HW1.



(1) [2 points] Determine the Jacobian of the arm.

$$\mathbf{J} = \begin{bmatrix} 0 & -l_1 \cos\theta \\ 1 & -l_1 \sin\theta \end{bmatrix}$$

(2) [2 points] Let d and θ be 0 and $\dot{d} = \dot{\theta} = v$, what is the contribution of each joint to the end-effector's linear velocity? To the angular velocity?

Linear:

$$\mathbf{J} \left[\begin{smallmatrix} v \\ 0 \end{smallmatrix}\right] = \begin{bmatrix} 0 \\ \\ v \end{bmatrix}$$

$$\mathbf{J} \left[\begin{smallmatrix} 0 \\ v \end{smallmatrix}\right] = \begin{bmatrix} -l_1 v \\ \\ 0 \end{bmatrix}$$

Angular:
No change wrt $\dot{d}$
When $\dot{\theta} = v$, $\dot{\theta_{eff}} = v$

(3) [2 points] For this problem, assume that forces due to gravity are pointing in the negative y direction. Assume each **link** has a mass of 0.25kg, and that each link's center

of mass can be assumed to be in the center of that link. What are the torques (as a function of the configuration $\Theta = \left[\begin{smallmatrix} d \\ \theta \end{smallmatrix}\right]$) necessary to counteract the effect of gravity on the center of mass of link 1?

$$\mathbf{J} = \begin{bmatrix} 0 & 0 \\ \\ 0.5 & 0 \end{bmatrix}$$
$$\tau = (\mathbf{J})^\top \left[\begin{smallmatrix} 0 \\ (9.8)(0.25) \end{smallmatrix}\right]$$
$$\tau = \begin{bmatrix} 1.23 \\ \\ 0 \end{bmatrix}$$

Of link 2?

$$\mathbf{J} = \begin{bmatrix} 0 & -\frac{l_1}{2}\cos\theta \\ \\ 1 & -\frac{l_1}{2}\sin\theta \end{bmatrix}$$
$$\tau = (\mathbf{J})^\top \left[\begin{smallmatrix} 0 \\ (9.8)(0.25) \end{smallmatrix}\right]$$
$$\tau = \begin{bmatrix} 2.45 \\ \\ -1.23 l_1 \sin\theta \end{bmatrix}$$

What torques are necessary to counteract the effect of gravity on all both links at the same time?

$$\Sigma\tau = \begin{bmatrix} 1.23 \\ \\ 0 \end{bmatrix} + \begin{bmatrix} 2.45 \\ \\ -1.23 l_1 \sin\theta \end{bmatrix}$$

$$\Sigma\tau = \begin{bmatrix} 3.68 \\ \\ -1.23 l_1 \sin\theta \end{bmatrix}$$

4 Written Questions

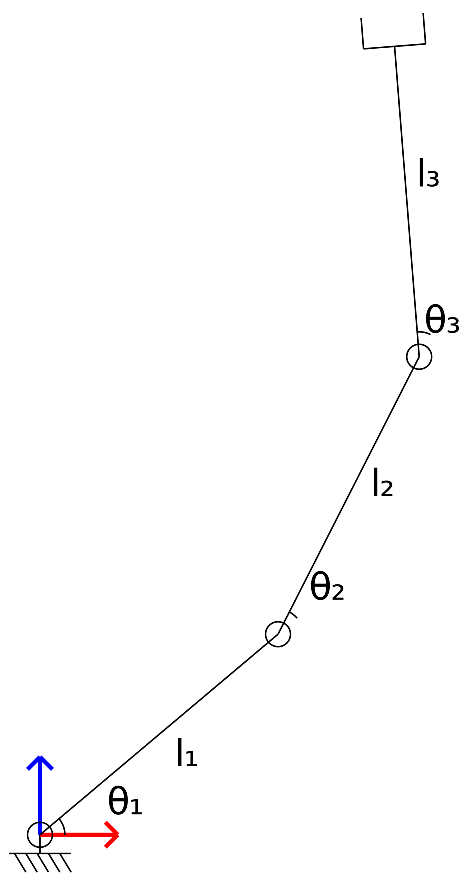
For the following problems, fully evaluate all answers unless otherwise specified.

Answers for written questions must be typed. We recommend LaTeX, Microsoft Word, OpenOffice, or similar. However, diagrams can be hand-drawn and scanned in.

Unless otherwise specified, **all units are in radians, meters, and seconds**, where appropriate.

1) Robot Analysis: RRR

Consider the following robot (assume no joint limits, and that the links do not intersect with the ground):



(1) [1 point] What is the dimension of the end-effector Jacobian? Assume the workspace is SE(2).

$\dim(\mathbf{J}) = 3 \times 3$

(2) [2 points] What does each row of $\mathbf{J}$ intuitively mean (in 1-2 sentences)?

Each row of $\mathbf{J}$represents the contribution of each configuration space variable to the end effector velocity in the row's corresponding dimension in space.

(3) [2 points] What does each column of $\mathbf{J}$ intuitively mean (in 1-2 sentences)?

Each col of $\mathbf{J}$represents the contribution of a single configuration space variable to the end effector velocity in all spatial dimensions.

(4) [2 points] Find f (the forward kinematic map to the end effector) for this arm for x, y, and θ as a function of θ_1, θ_2, and θ_3.

$$f\left(\begin{bmatrix}\theta_1\\ \theta_2\\ \theta_3\end{bmatrix}\right) = \begin{bmatrix} l_3c_{123} + l_2c_{12} + l_1c_1 \\ l_3s_{123} + l_2s_{12} + l_1s_1 \\ \theta_1 + \theta_2 + \theta_3 \end{bmatrix}$$

(5) [1 point] Find the partial derivatives of the previous answers with regard to θ_1, θ_2, and θ_3. Using these partial derivatives, what is $\mathbf{J}$?

$$\mathbf{J} = \begin{bmatrix} -l_3 s_{123} - l_2 s_{12} - l_1 s_1 & -l_3 s_{123} - l_2 s_{12} & -l_3 s_{123} \\ l_3 c_{123} + l_2 c_{12} + l_1 c_1 & l_3 c_{123} + l_2 c_{12} & l_3 c_{123} \\ 1 & 1 & 1 \end{bmatrix}$$

(6) [3 points] Let v be some nonzero positive scalar. When the arm is at position $\theta_1 = \theta_2 = \theta_3 = 0$, and $\dot{\theta}_1 = \dot{\theta}_2 = \dot{\theta}_3 = v$, what is the contribution of each joint to the end-effector's linear velocity? To the angular velocity?

Linear is idx 1 and 2, Angular is idx 3:

$\mathbf{J}(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}) \begin{bmatrix} v \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ v(l_1+l_2+l_3) \\ v \end{bmatrix}$

$\mathbf{J}(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}) \begin{bmatrix} 0 \\ v \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ v(l_2+l_3) \\ v \end{bmatrix}$

$\mathbf{J}(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}) \begin{bmatrix} 0 \\ 0 \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ v(l_3) \\ v \end{bmatrix}$

(7) [3 points] At configuration $\theta_1 = \theta_2 = \theta_3 = 0$, what $\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$ must be applied to instantaneously move the end-effector in the planar direction $\begin{bmatrix} -1 \\ 0 \end{bmatrix}$?

In this configuration, the arm exists in a singularity and cannot move in the x direction.

$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$?

$$\begin{bmatrix} \frac{1}{l_1+l_2+l_3} \\ 0 \\ 0 \end{bmatrix}$$

Any combination that satisfies the equality $(l_1 + l_2 + l_3)\dot{\theta}_1 + (l_2 + l_3)\dot{\theta}_2 + l_3\dot{\theta}_3 = k$ for any k

Now considering rotation, what joint velocities must be applied to move instantaneously in $\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ (where the third term is rotation)?

$$\dot{\theta} = \begin{bmatrix} 0 \\ \frac{1-l_3}{l_2} \\ 1 - \frac{1-l_3}{l_2} \end{bmatrix}$$

$(l_1 + l_2 + l_3)\dot{\theta}_1 + (l_2 + l_3)\dot{\theta}_2 + l_3\dot{\theta}_3 = k$ for any k

$\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 = 1$

(8) [3 points] What joints torques are necessary to exert a 5N force on the end effector in the $\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$ direction. Assume the configuration is $\left[\begin{smallmatrix}\theta_1\\\theta_2\\\theta_3\end{smallmatrix}\right]$.

$$\mathbf{J} = \begin{bmatrix} -l_3 s_{123} - l_2 s_{12} - l_1 s_1 & -l_3 s_{123} - l_2 s_{12} & -l_3 s_{123} \\ \\ l_3 c_{123} + l_2 c_{12} + l_1 c_1 & l_3 c_{123} + l_2 c_{12} & l_3 c_{123} \end{bmatrix}$$

$\mathbf{F} = \frac{\sqrt{2}}{2}\left[\begin{smallmatrix}5\\5\end{smallmatrix}\right]$

$\tau = (\mathbf{J})^{\top}\,\mathbf{F}$

$\tau = \frac{5\sqrt{2}}{2}\left[\begin{smallmatrix} l_3(c_{123}-s_{123})+l_2(c_{12}-s_{12})+l_1(c_1-s_1) \\ l_3(c_{123}-s_{123})+l_2(c_{12}-s_{12}) \\ l_3(c_{123}-s_{123}) \end{smallmatrix}\right]$

(9) [5 points] For this problem, assume that forces due to gravity are pointing in the negative y direction. Assume each **link** has a mass of 0.25kg, and that each link's center of mass can be assumed to be in the center of that link. What are the joint torques (as a function of the configuration $\Theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$) necessary to counteract the effect of gravity on the center of mass of link 1?

$$\mathbf{J} = \begin{bmatrix} -\frac{l_1}{2}s_1 & 0 & 0 \\ \frac{l_1}{2}c_1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} 0 \\ -9.8(0.25) \end{bmatrix}$$

$$\tau = (\mathbf{J})^\top(-\mathbf{F})$$

$$\tau = \begin{bmatrix} 1.23 l_1 c_1 \\ 0 \\ 0 \end{bmatrix}$$

Of link 2?

$$\mathbf{J} = \begin{bmatrix} -\frac{l_2}{2}s_{12} - l_1 s_1 & -\frac{l_2}{2}s_{12} & 0 \\ \frac{l_2}{2}c_{12} + l_1 c_1 & \frac{l_2}{2}c_{12} & 0 \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} 0 \\ -2.45 \end{bmatrix}$$

$$\tau = (\mathbf{J})^\top(-\mathbf{F})$$

$$\tau = \begin{bmatrix} 2.45 l_1 c_1 + 1.23 l_2 c_{12} \\ 1.23 l_2 c_{12} \\ 0 \end{bmatrix}$$

Of link 3?

$$\mathbf{J} = \begin{bmatrix} -\frac{l_3}{2}s_{123} - l_2 s_{12} - l_1 s_1 & -\frac{l_3}{2}s_{123} - l_2 s_{12} & -\frac{l_3}{2}s_{123} \\ \frac{l_3}{2}c_{123} + l_2 c_{12} + l_1 c_1 & \frac{l_3}{2}c_{123} + l_2 c_{12} & \frac{l_3}{2}c_{123} \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} 0 \\ -2.45 \end{bmatrix}$$

$$\tau = (\mathbf{J})^{\top}(-\mathbf{F})$$

$$\tau = \begin{bmatrix} 2.45 l_1 c_1 + 2.45 l_2 c_{12} + 1.23 l_3 c_{123} \\ 2.45 l_2 c_{12} + 1.23 l_3 c_{123} \\ 1.23 l_3 c_{123} \end{bmatrix}$$

What torques are necessary to counteract the effect of gravity on all three links at the same time?

$$\boldsymbol{\Sigma}\tau = \tau_1 + \tau_2 + \tau_3$$

$$\boldsymbol{\Sigma}\tau = \begin{bmatrix} 6.13 l_1 c_1 + 3.68 l_2 c_{12} + 1.23 l_3 c_{123} \\ 3.68 l_2 c_{12} + 1.23 l_3 c_{123} \\ 1.23 l_3 c_{123} \end{bmatrix}$$

(10) [2 points] Now assume the **joints** also each have mass; the joint mass is 0.35kg, centered on the axis of rotation. What are the new set of torques needed to counteract the effect of gravity on the entire robot (links and joints), as a function of the configuration $\Theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$?

Joint 1

$$\tau = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Joint 2

$$\mathbf{J} = \begin{bmatrix} -l_1 s_1 & 0 & 0 \\ l_1 c_1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} 0 \\ -3.43 \end{bmatrix}$$

$$\tau = (\mathbf{J})^\top (-\mathbf{F})$$

$$\tau = \begin{bmatrix} 3.43 l_1 c_1 \\ 0 \\ 0 \end{bmatrix}$$

Joint 3

$$\mathbf{J} = \begin{bmatrix} -l_2 s_{12} - l_1 s_1 & -l_2 s_{12} & 0 \\ l_2 c_{12} + l_1 c_1 & l_2 c_{12} & 0 \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} 0 \\ -3.43 \end{bmatrix}$$

$$\tau = (\mathbf{J})^\top (-\mathbf{F})$$

$$\tau = \begin{bmatrix} 3.43 l_1 c_1 + 3.43 l_2 c_{12} \\ 3.43 l_2 c_{12} \\ 0 \end{bmatrix}$$

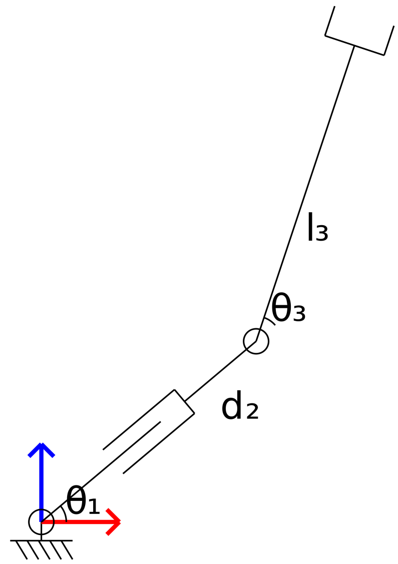
Everything all together (links and joints)

$$\mathbf{\Sigma}\tau = \tau_1 + \tau_2 + \tau_3$$

$$\mathbf{\Sigma}\tau = \begin{bmatrix} 12.99l_1c_1 + 7.11l_2c_{12} + 1.23l_3c_{123} \\ 7.11l_2c_{12} + 1.23l_3c_{123} \\ 1.23l_3c_{123} \end{bmatrix}$$

2) Robot Analysis: RPR

Consider the following robot (assume no joint limits, and that the links do not intersect with the ground):



Find the following:

(1) [2 points] The forward kinematic map for the end effector.

$$\mathbf{T} = \begin{bmatrix} c_{13} & -s_{13} & d_2c_1 + l_3c_{13} \\ s_{13} & c_{13} & d_2s_1 + l_3s_{13} \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{f} = \begin{bmatrix} d_2c_1 + l_3c_{13} \\ d_2s_1 + l_3s_{13} \\ \theta_1 + \theta_2 \end{bmatrix}$$

(2) [3 points] The Jacobian for the end effector.

$$\mathbf{J} = \begin{bmatrix} -d_2 s_1 - l_3 c_{13} & c_1 & -l_3 s_{13} \\ d_2 c_1 + l_3 c_{13} & s_1 & l_3 c_{13} \\ 1 & 0 & 1 \end{bmatrix} \text{ where } \Theta = \begin{bmatrix} \theta_1 \\ d_2 \\ \theta_3 \end{bmatrix}$$

(3) [5 points] The joint torques necessary to counteract the effects of gravity (where gravitational forces are in the negative y direction). For this, assume the entire length between the rotational joints is d_2, and this consists of a mass of 1.0kg centered at $d_2/2$ along this link. The rotational joints have masses of 0.2kg centered at their axis of rotation, and the distal link (of length l_3) has a mass of 0.5kg centered halfway along the link. The end effector is massless. Show your work by writing down the Jacobian and force vector.

Rotational joint (θ_1)

$$\mathbf{J}_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{F}_0 = \left[\begin{smallmatrix} 0 \\ -2.74 \end{smallmatrix}\right]$$

$$\mathbf{T} = (\mathbf{J}_0)^\top (-\mathbf{F}_0)$$

$$\mathbf{T} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Prismatic Joint (d_2)

$$\mathbf{J}_1 = \begin{bmatrix} -\frac{d_2 s_1}{2} & \frac{c_1}{2} & 0 \\ \frac{d_2 c_1}{2} & \frac{s_1}{2} & 0 \end{bmatrix} \quad \mathbf{F}_1 = \left[\begin{smallmatrix} 0 \\ -9.8 \end{smallmatrix}\right]$$

$$\mathbf{T} = (\mathbf{J}_1)^\top (-\mathbf{F}_1)$$

$$\mathbf{T} = \begin{bmatrix} 4.9 d_2 c_1 \\ 4.9 s_1 \\ 0 \end{bmatrix}$$

Rotational joint (θ_2)

$$\mathbf{J}_2 = \begin{bmatrix} -d_2 s_1 & c_1 & 0 \\ d_2 c_1 & s_1 & 0 \end{bmatrix} \quad \mathbf{F}_2 = \left[\begin{smallmatrix} 0 \\ -1.96 \end{smallmatrix}\right]$$

$$\mathbf{T} = (\mathbf{J}_2)^\top (-\mathbf{F}_2)$$

$$\mathbf{T} = \begin{bmatrix} 1.96 d_2 c_1 \\ 1.96 s_1 \\ 0 \end{bmatrix}$$

Link 2 (l_3)

$$\mathbf{J}_3 = \begin{bmatrix} -d_2 s_1 - \frac{l_3 s_{13}}{2} & c_1 & -\frac{l_3 s_{13}}{2} \\ d_2 c_1 + \frac{l_3 c_{13}}{2} & s_1 & \frac{l_3 c_{13}}{2} \end{bmatrix} \quad \mathbf{F}_3 = \left[\begin{smallmatrix} 0 \\ -4.9 \end{smallmatrix}\right]$$

$$\mathbf{T} = (\mathbf{J}_3)^\top (-\mathbf{F}_3)$$

$$\mathbf{T} = \begin{bmatrix} 4.9 d_2 c_1 + 2.45 l_3 c_{13} \\ 4.9 s_1 \\ 2.45 l_3 c_{13} \end{bmatrix}$$

Everything all together (links and joints)

$$\mathbf{J}_1 = \begin{bmatrix} -\frac{d_2 s_1}{2} & \frac{c_1}{2} & 0 \\ \frac{d_2 c_1}{2} & \frac{s_1}{2} & 0 \end{bmatrix} \quad \mathbf{F}_1 = \left[\begin{smallmatrix} 0 \\ -9.8 \end{smallmatrix}\right]$$

$$\mathbf{J}_2 = \begin{bmatrix} -d_2 s_1 & c_1 & 0 \\ d_2 c_1 & s_1 & 0 \end{bmatrix} \quad \mathbf{F}_2 = \left[\begin{smallmatrix} 0 \\ -2.94 \end{smallmatrix}\right]$$

$$\mathbf{J}_3 = \begin{bmatrix} -d_2 s_1 - \frac{l_3 s_{13}}{2} & c_1 & -\frac{l_3 s_{13}}{2} \\ d_2 c_1 + \frac{l_3 c_{13}}{2} & s_1 & \frac{l_3 c_{13}}{2} \end{bmatrix} \quad \mathbf{F}_3 = \left[\begin{smallmatrix} 0 \\ -4.9 \end{smallmatrix}\right]$$

$$\mathbf{T} = (\mathbf{J}_1)^\top (-\mathbf{F}_1) + (\mathbf{J}_2)^\top (-\mathbf{F}_2) + (\mathbf{J}_3)^\top (-\mathbf{F}_3)$$

$$\mathbf{T} = \begin{bmatrix} 11.76 d_2 c_1 + 2.45 l_3 c_{13} \\ 11.76 s_1 \\ 2.45 l_3 c_{13} \end{bmatrix}$$

5 Feedback

1) Feedback Form *5 points*

We are always looking to improve the class! To that end, we're looking for your feedback on the assignments. When you've completed the assignment, please fill out the feedback form.

6 Code Questions

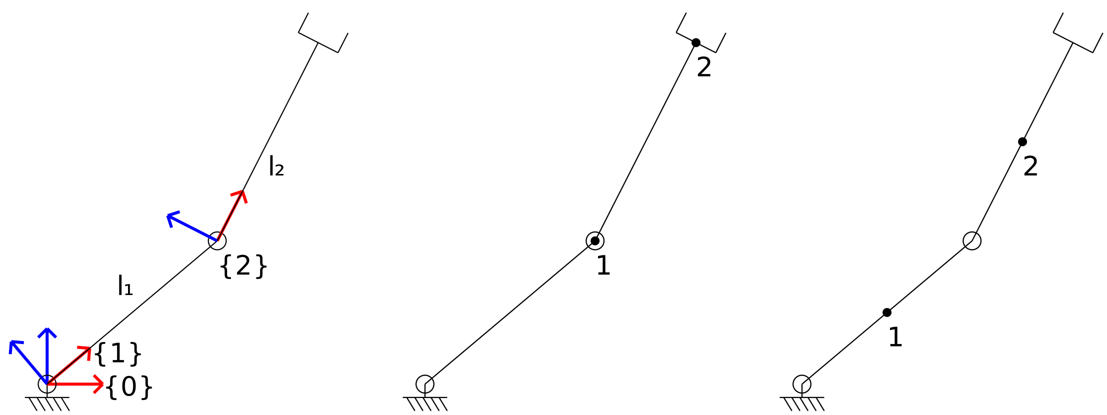
In these problems, we'll analytically test some of the work from the previous section.

Copy the Code Handout folder to some location of your choice. Open Matlab and navigate to that location. Whenever you work on the assignment, go into this directory and run `setup.m`.

1) Jacobians and Velocities *15 points*

Begin by opening the `ex_01` directory in Matlab.

In this exercise, you must fill out several files. When completed correctly, the first three functions will define the forward kinematics and Jacobian to several points on an RR arm; the final function will provide a self-check for the first three functions.



- `forward_kinematics_RR.m` - As in Exercise 1 from Assignment 1, the forward kinematics to several frames should be filled in. Note: the frames for this problem are a subset of those from the previous assignment. In the code, `H_i_j` refers to $\mathbf{H}_\mathbf{i}^\mathbf{j}$, and you will need to define $\mathbf{H}_\mathbf{1}^\mathbf{0}$ and $\mathbf{H}_\mathbf{2}^\mathbf{0}$ (where $\{\mathbf{0}\}$, $\{\mathbf{1}\}$, and $\{\mathbf{2}\}$ are shown in the above left figure).
- `jacobian_link_ends_RR.m` - Jacobian matrices to the end of the robot links should be filled in in this problem. In the code, `J_i` refers to the Jacobian to the end of link i (i.e., point i in the above middle figure).
- `jacobian_coms_RR.m` - Jacobian matrices to the centers of the robot links should be filled in in this problem. In the code, `J_i` refers to the Jacobian to the center of link i (i.e., point i in the above middle figure).
- `sample_path.m` - This function loads data taken from the robot, and draws the path taken by the center and end of each link. You should use the functions you have written above to fill in the position and velocity of these points.

After completing these tasks, run the `sample_path.m` function. You should see four separate lines, representing the path of the center and end of each link. These lines should be formed of arrows, which represent the velocity of each of these points. **Important:** if the arrows are not tangent to the path, there is an error in your code! Verify your code is correct before continuing. As before, when debugging, remember to use simple test cases such as $\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ or $\left[\begin{smallmatrix}0\\\pi/2\end{smallmatrix}\right]$ to check your matrices.

The deliverables for this problem are the four code files above, as well as the resulting figure from running `sample_path.m`. This figure should be saved as `path.fig`.

2) Forces and Gravity Compensation *15 points*

Begin by opening the `ex_02` directory in Matlab.

In this exercise, you must fill use the Jacobians found in the first problem to compute joint torques which result in forces applied by the robot. This will involve finishing two functions:

- `get_joint_torques.m` - In this function, you must determine the joint torques necessary for applying a given force with the end effector (e.g., centered at the end effector origin in a given direction with a given magnitude).
- `get_grav_comp_torques.m` - In this function, you will compute the joint torques which compensate for the force of gravity on each part of the robot, given the current joint angles. Note the direction and magnitude of acceleration due to gravity is passed into this function (assume units of m/s^2).

After completing these function, run the `sample_torques.m` function. You will see your computed torques for a test run of the robot, alongside ground truth joint torques. One figure will contain torques for applying a force with the end effector, and the other will contain torques for compensating for gravity. Verify that these match before continuing!

The deliverables for this problem are the two code files above, as well as the resulting figures from running `sample_torques.m`. These should be saved as `torques_ee.fig` and `torques_grav_comp.fig`, respectively.

7 Hands-On Questions

1) Gravity Compensation *15 points*

Begin by opening the ex_04 directory in Matlab.

In this section, we will connect to the robot, and will command torques which roughly cancel the weight of the elements of the robot.

First, open the file `gravity_compensation.m`, and complete the function. When correctly completed, this should command joint torques that are roughly equivalent to those applied by gravity on the actuators.

You will use function(s) from ex_02 to complete this exercise. To most effectively use your time on the robot itself, ensure you test out the logic you have added before connecting to the robot with some simple test cases.

TODO: The link lengths for this robot are: . Note: link lengths might change from robot to robot in labs moving forward. Change the `robot_info.link_lengths` variable in the file `robot_info.m`. You will need to change this variable when working on the different robots.

To complete this lab, you (or the TA) will move the robot through 5 configurations and ensure it is approximately weightless in each position. Do not be concerned if the robot moves slightly; the torque control on the robots is only approximate and so it is not expected that the robots perfectly balance.

The robot must pass through the following joint angle positions in order to successfully complete the lab:

- $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
- $\begin{bmatrix} \frac{\pi}{4} \\ \frac{\pi}{2} \end{bmatrix}$
- $\begin{bmatrix} \frac{\pi}{2} \\ 0 \end{bmatrix}$
- $\begin{bmatrix} \frac{3\pi}{4} \\ \frac{-\pi}{2} \end{bmatrix}$
- $\begin{bmatrix} \pi \\ 0 \end{bmatrix}$

You will be graded by the computed joint torques at the given locations.

IN PERSON: plug your computer into the router connected to the robots. Type `HebiLookup` into MATLAB, and ensure there are four modules (joint1 and joint2, each in module family 'Robot A' and 'Robot B') listed in the table of available modules. If not, ensure that you are indeed connected to the network. You may have to run HebiLookup.setLookupAddresses('*') after switching networks.

Note that there is a USB-Ethernet adapter provided if your computer does not have an Ethernet port. Also, these routers provide access to the internet and CMU networks, so MATLAB will be able to find the license server.

After connecting, you should be able to run `gravity_compensation`. Note that during this lab, you are actively controlling torques on the robot, which can be dangerous. **To stop the robot, press Control-C from MATLAB.** Be ready to press this key combination when running the script!

When you are done, press 'ctrl-c' to exit. At this point, a quick self-check will be run to ensure that you have indeed moved the robot to each of the locations of interest. If any of these checks fail, an error message will print to the screen indicating the problem; if they succeed, a success message will print, and the log for this problem will be ready for submission. Note that this check does not ensure that the computed torques are correct!

REMOTE ACCESS: Make sure that you pass `ex_02` before attempting remote access! Before you can begin, you must verify with a TA that `torques_ee.fig` and `torques_grav_comp.fig` are correct.

First, follow the directions on Piazza to begin Remote Access. Then, once you have MATLAB open on the remote desktop, type `HebiLookup` into the MATLAB command line, and ensure there are two modules (joint1 and joint2) listed in the table of available modules. If not, notify the TA to plug in wall ethernet cord to the desktop first and the robot ethernet cord second.

Run `gravity_compensation`, which will actively control torques of the robot. The robot will begin to move through a series of positions. Because the robot is actively controlling, it may move dangerously or jerk if the code is not written properly. If the TA sees any dangerous motion, they will emergency stop the robot, and you must work on fixing your code.

When you are done, press 'ctrl-c' to exit. At this point, a quick self-check will be run to ensure that the robot to each of the locations of interest. If any of these checks fail, an error message will print to the screen indicating the problem; if they succeed, a success message will print, and the log for this problem will be ready for submission. Note that this check does not ensure that the computed torques are correct!

MAKE SURE TO PUSH EVERYTHING TO YOUR PRIVATE GITHUB REPOSITORY!

2) Submission

To submit, run `create_submission.m`. It will first check that your submitted files run without error, and perform a small sanity check. Note, this is not going to grade your submission! The function will create a file called `handin-2.tar.gz`.

8 Submission Checklist

- ☐ Create a PDF of your answers to the written questions with the name `writeup.pdf`.
- ☐ Make sure you have `writeup.pdf` in the same folder as the `create_submission.m` script.
- ☐ Run `create_submission.m` in Matlab.
- ☐ Upload `handin-2.tar.gz` to Canvas.
- ☐ Upload `writeup.pdf` to Gradescope.
- ☐ After completing the entire assignment, fill out the feedback form[3].

[3] https://canvas.cmu.edu/courses/11823/quizzes/25758