Homework 3

EECS/BioE/MechE C106A/206A
Introduction to Robotics

Due: September 20, 2022

Note: Problems marked [bonus] will be eligible for a (very) small amount of extra credit, though you cannot receieve more than a full score on the homework as a whole. We encourage you not to spend exorbitant amounts of time on these questions, and as such, you may receieve partial credit for attempting them.

Note 2: For problems 1-3, there is no written submission. You must fill in the provided `hw3.py` file with your solutions to the problems in this assignment.

You will also need to use `kin_func_skeleton.py`, which you implemented as part of your Lab 3 prelab, and which you also use in Lab 3. To get started, you should place both `hw3.py` and `kin_func_skeleton.py` in the same directory, as `hw3.py` imports `kin_func_skeleton`.

Your deliverables for this assignment are:

1. `hw3.py` with your implementation of your solutions to this assignment.
2. `kin_func_skeleton.py` with your implementations from Lab 3 prelab.

Both the above files should be submitted together to the Gradescope assignment `Homework 3`. Your credit for this problem set will be awarded by the autograder.

Theory

For this problem set, you should recall the various formulas for finding the Twist corresponding to the type of joints that a robot has. We will consider the following types of joints:

1. **Prismatic Joints:** These are joints that implement a pure translation along an axis (a screw with infinite pitch $h = \infty$). The twist corresponding to such a joint is

$$\xi = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v \\ 0 \end{bmatrix}$$

where v is a unit vector in the direction of the translation.

2. **Revolute Joints:** These are joints that implement a pure rotation about some axis (a screw with zero pitch $h = 0$). These are the kinds of joints you saw in Lab 3 with the Baxter robot. The twist corresponding to such a joint is

$$\xi = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} -\omega \times q \\ \omega \end{bmatrix}$$

where ω is a unit vector in the direction of the rotation axis, and q is a point through which the rotation axis passes, all as seen with the robot in its initial configuration.

3. **Screw Joints:** These are joints that implement a simultaneous rotation and translation along the same axis (a screw with nonzero finite pitch h). The twist corresponding to such a joint is

$$\xi = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} -\omega \times q + h\omega \\ \omega \end{bmatrix}$$

where ω is a unit vector in the direction of the screw axis, q is a point through which the axis passes, and h is the pitch of the screw.

In each of the above cases, the point q can be chosen as any point on the axis. Every such point will result in the same twist. However some points may be more convenient from a computational viewpoint than others (for instance, if the axis passes through the origin, we usually want to pick $q = 0$ to simplify calculations).

If a robot has n joints (or n "degrees of freedom") then to compute the forward kinematics map of the robot, we must start off by finding the twists $\xi_1, \cdots, \xi_n$ corresponding to each of the joints of the robot.

The "joint angles" $\theta_1, \cdots, \theta_n$ specify the amount by which each joint on the robot is to be actuated. So for revolute and screw joints, θ is the angle through which we rotate the joint from its initial position. For prismatic joints, θ is the amount by which the joint is distended (the distance we translate by). The θ_i's are what we get to control to move the robot to our desired location. Think of these as our control inputs, which we can pick to move the motors on the robot to a particular position.

Forward Kinematics is the problem of finding where the end effector of the robot ends up when we pick particular values of $\theta_1, \cdots \theta_n$. In particular, we fix a spatial frame S to be stationary, and attach a tool frame T rigidly to the end effector of the robot. Then solving the forward kinematics map of the robot amounts to finding $g_{ST}(\theta) \in SE(3)$ as a function of $\theta = (\theta_1, \cdots \theta_n)$. Once we have the twists $\xi_1, \cdots \xi_n$ corresponding to each joint of the robot, we can use the *product of exponentials* formula to write down an expression for the forward kinematics map as

$$g_{ST}(\theta) = e^{\hat{\xi}_1 \theta_1} e^{\hat{\xi}_2 \theta_2} \cdots e^{\hat{\xi}_n \theta_n} g_{ST}(0)$$

where $g_{ST}(0) \in SE(3)$ is the pose of the tool frame with respect to the spatial frame when all joint angles are set to zero (the *initial configuration)*. Usually, we get to pick the initial configuration, so we know what $g_{ST}(0)$ is (or we can easily find it).

Problem 1. Forward Kinematics for 3DOF Manipulators

Figure 1 shows a 3 degrees-of-freedom manipulator in its initial configuration. The robot has three revolute joints. Take $L1 = 1$. In this problem, you will implement the forward kinematics map of this manipulator. Do this by writing down

(a) $g_{st}(0)$, the rigid pose corresponding to the initial configuration.

(b) The twists ξ_1, ξ_2, ξ_3 corresponding to each of the three joints of the manipulator.

(c) An expression for the forward kinematics map $g_{st}(\theta)$ where $\theta \in \mathbb{R}^3$ is the vector of joint angles $(\theta_1, \theta_2, \theta_3)$. You may leave your answer in terms of the exponentials of known matrices.

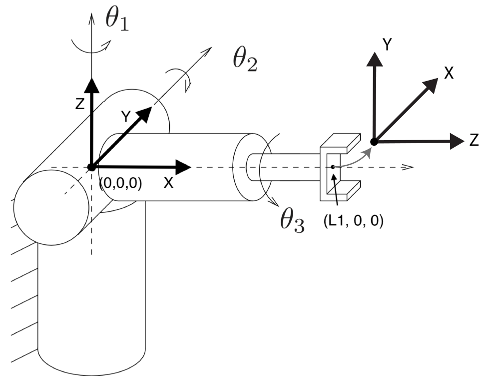


Figure 1: A three degree of freedom manipulator

Problem 2. Forward Kinematics with Screw Joints

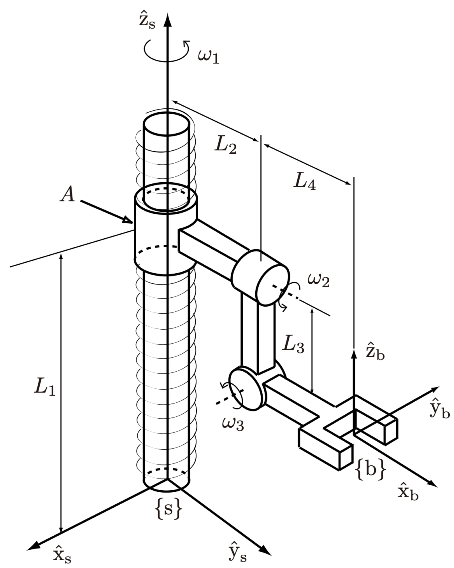


Figure 2: A 3 DOF manipulator.

Figure 2 shows the initial configuration of a robot arm whose first joint is a screw joint of pitch $h = 2$. The other two joints are revolute. The arm's link lengths are $L_1 = 10, L_2 = L_3 = 5$, and $L_4 = 3$. In this problem you will implement the forward kinematics map of this robot, in the following steps:

(a) Write down the 4×4 initial end effector configuration of the manipulator, $g_{sb}(0)$.

(b) Find the twists ξ_1, ξ_2, ξ_3 corresponding to each of the joints of the manipulator, and hence write down an expression for the forward kinematics map $g_{sb}(\theta)$. You may leave your answer in terms of the exponentials of known matrices.

Problem 3. Forward Kinematics for 6DOF Manipulators

Figure 3 shows the 6DOF Stanford arm in its initial configuration, with 5 revolute joints and one prismatic joint (joint 3). Take $l_0 = l_1 = 1$. You may also assume that in the initial configuration, q_w is a distance l_1 away from q_1. Implement the forward kinematics map of this robot by finding:

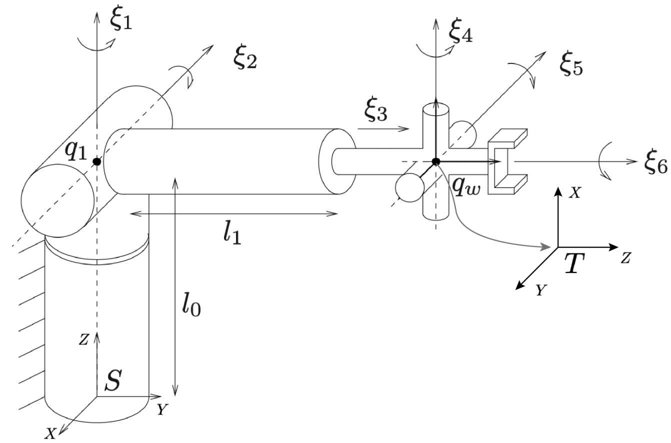


Figure 3: An idealized version of the Stanford arm.

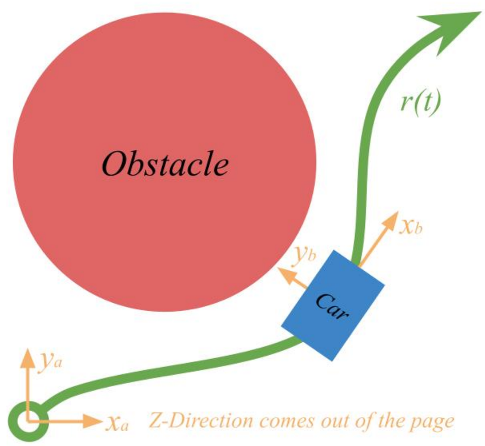
(a) $g_{st}(0)$, the rigid pose corresponding to the initial configuration.

(b) The twists $\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6$ corresponding to each of the six joints of the manipulator.

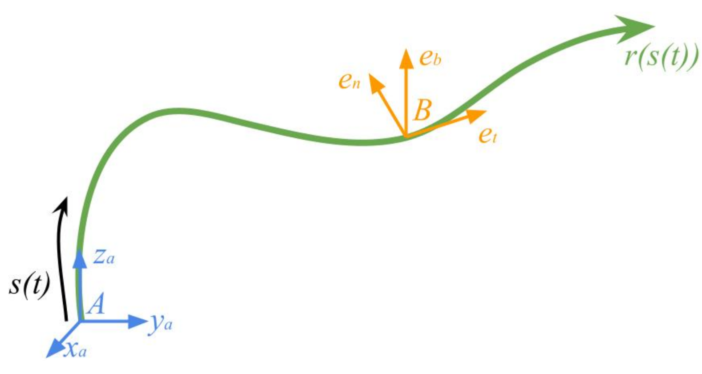
(c) An expression for the forward kinematics map $g_{st}(\theta)$ where $\theta \in \mathbb{R}^6$ is the vector of joint angles. You may leave your answer in terms of the exponentials of known matrices.

Problem 4. Bonus: Frenet-Serret Racing Inc.

You and your friend Toto have started a self-driving race car team. So far, you've devised a clever path-planning algorithm that helps your car find its way around obstacles and overtake other drivers in three dimensions.
Given a scene like the one below, your path planner is able to find a trajectory $r(t) \in \mathbb{R}^3$ such that your car avoids an obstacle.



Toto, however, has a strong knowledge of rigid body transformations. In addition to specifying a position for the car, he knows it's important to specify an orientation. How can you derive an orientation for your car using just a 3D trajectory $r(t)$? In this problem, we'll propose an answer to this question through the lens of *differential geometry*.
When describing the orientation associated with a trajectory in space, we may use the Frenet-Serret basis $\{e_t, e_n, e_b\}$, pictured below.



The unit vector tangent to the curve is called the *unit tangent vector*, e_t. It points in our vehicle's direction of motion. The unit vector pointing into the turn is called the *unit normal vector*, e_n, and the unit vector normal to these two is called the *unit binormal vector*, e_b. Thus, at each point in our trajectory $r(t)$, $\{e_t, e_n, e_b\}$ forms an orthonormal basis for $\mathbb{R}^3$ using a set of directions very natural to the vehicle's motion.
How do we solve for these vectors? First, we must re-parameterize the trajectory $r(t) \in \mathbb{R}^3$,

in terms of arc length, $s \in \mathbb{R}$. Given a trajectory $r(t) \in \mathbb{R}^3$ that is differentiable in time on $\mathbb{R}^+$, recall that the arc length of r is defined:

$$s(t) = \int_0^t \sqrt{\left(\frac{dx(\tau)}{d\tau}\right)^2 + \left(\frac{dy(\tau)}{d\tau}\right)^2 + \left(\frac{dz(\tau)}{d\tau}\right)^2} d\tau \tag{1}$$

Once we have re-parameterized $r(t)$ as $r(s(t))$, we may solve for the Frenet-Serret basis in the world frame.

(a) Suppose we have a trajectory $r(t) \in \mathbb{R}^3$ in the world frame.
We know from our knowledge of calculus that by taking the time derivative of $r(s(t))$, we will find a vector tangent to the curve $r(s(t))$. However, this vector may not have unit magnitude! How can we directly find a tangent vector that does have unit magnitude?
We may compute the *unit* tangent vector in the world frame as:

$$e_t = \frac{dr}{ds} \tag{2}$$

Show that $e_t = \frac{dr}{ds}$ has unit magnitude and is parallel to the tangent vector $\frac{dr}{dt}$. *Hint: start by taking the time derivative of $r(s(t))$ using the chain rule. Think about what the magnitude of this derivative represents.*

(b) Once we have computed the unit tangent vector, we may compute the unit normal vector in the world frame with the following formula:

$$e_n = \frac{1}{\kappa} \frac{de_t}{ds} \tag{3}$$

Where $\kappa \in \mathbb{R}$ is a constant known as *curvature*. Prove that the unit normal vector is orthogonal to the unit tangent vector. *Hint: The norm of the vector e_t is equal to 1 everywhere. How can we use this to help us show orthogonality?*

(c) Finally, we may compute the unit binormal vector with the following formula:

$$e_b = e_t \times e_n \tag{4}$$

We now define the *Frenet-Serret* frame $B = \{e_t, e_n, e_b\}$. Find the homogeneous rigid body transformation from the Frenet-Serret frame (frame B) into the world frame (frame A) as a function of arc length. Your answer should be of the form:

$$g_{ab}(s) = \begin{bmatrix} R(s) & p(s) \\ 0 & 1 \end{bmatrix} \tag{5}$$

You may leave your final answer in terms of $r(s(t))$.

(d) Explain in your own words why the Frenet-Serret basis might be useful for describing the trajectories of autonomous vehicles.