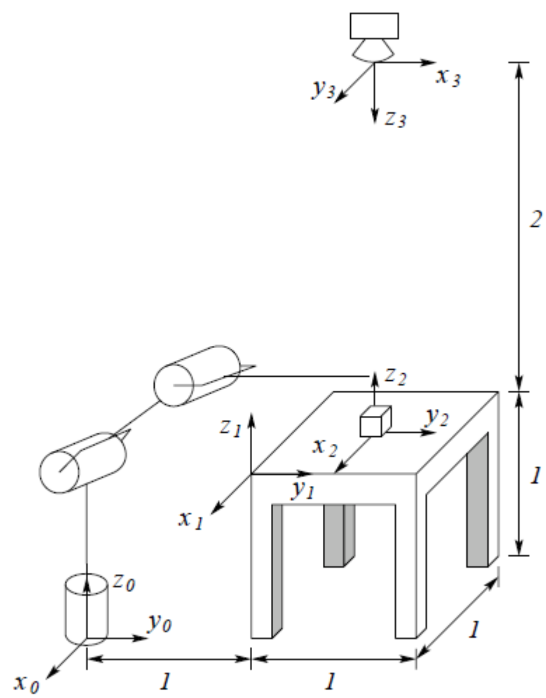
COMS W4733: Computational Aspects of Robotics

Homework 4 Solutions

Problem 1: Homogeneous Transformations (18 points)

In the diagram below, a robot with base frame 0 is located 1 m away from a table, which is 1 m high and 1 m square. Frame 1 is rigidly attached to the table at the corner closest to the robot, and the two frames' y axes are coincident. A cube is located at the center of the table, with frame 2 rigidly attached to the center of its bottom face. A camera is located 2 m directly above the center of the table, with frame 3 rigidly attached to it.



(a) The first two transformations only involve translations. The last transformation involves both a translation and a rotation. The rotation may be found in one of several ways. For example, we can first rotate frame 2 about z_2 by +90 degrees, aligning x_2 with x_3, followed by a rotation about x_2 by 180 degrees. Equivalently, we can rotate about z_2 by −90 degrees, aligning y_2 with y_3, followed by a rotation about y_2 by 180 degrees. Either way, we get the same result:

$$R_3^2 = R_z(+90)R_x(180) = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

$$R_3^2 = R_z(-90)R_y(180) = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

The rotation matrix R_3^2 can be substituted into the transformation A_3^2. The other two have no rotation displacements, so their rotation parts are the identity matrix. Finally, the translation components can be found by simply reading off measurements in the figure. Note that the translation for A_j^i is written with respect to frame i.

$$A_1^0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_2^1 = \begin{pmatrix} 1 & 0 & 0 & -0.5 \\ 0 & 1 & 0 & 0.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_3^2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(b) If the cube is rotated and moved, this will change A_2^1 and A_3^2. The provided rotation about the $z_2 = z_1$ axis and the position relative to frame 1 produce exactly the homogeneous transformation A_2^1:

$$A_2^1 = \begin{pmatrix} 0 & -1 & 0 & -0.2 \\ 1 & 0 & 0 & 0.2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

For A_3^2, the resultant cube rotation leaves only a difference of a $R_x(180)$ rotation between frames 2 and 3. The translation is the displacement that would move the cube back to the center of the table and then up to the camera: $(0.3, 0.3, 2)$.

$$A_3^2 = \begin{pmatrix} 1 & 0 & 0 & 0.3 \\ 0 & -1 & 0 & 0.3 \\ 0 & 0 & -1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

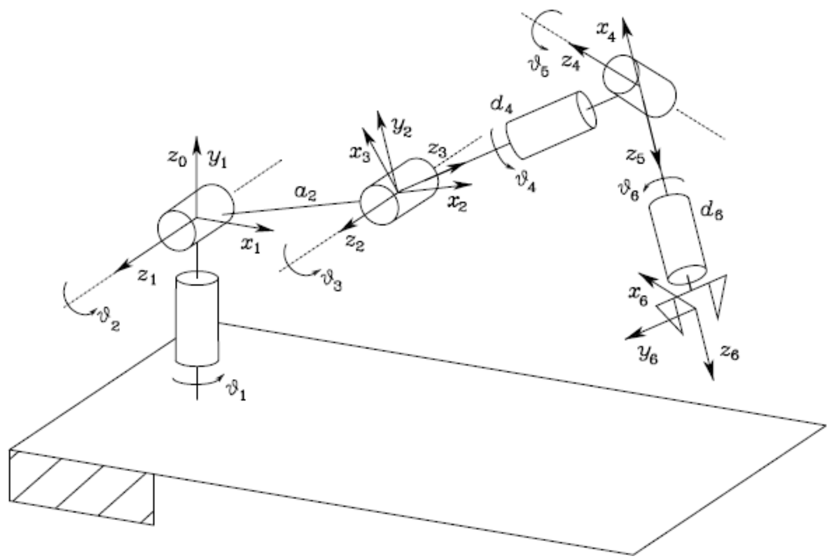
(c) The composite transformation can be found as

$$A_3^0 = A_1^0 A_2^1 A_3^2 = \begin{pmatrix} 0 & 1 & 0 & -0.5 \\ 1 & 0 & 0 & 1.5 \\ 0 & 0 & -1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

It doesn't matter which choice of intermediate frames we use, since the initial and final frames related by this transformation are the same in both cases.

Problem 2: Forward Kinematics (16 points)

We attach a spherical wrist to the end of the anthropomorphic arm as shown below. All coordinate frames are defined so as to conform to DH convention. Not all frame axes are shown; the origins of frame pairs 0 and 1, 2 and 3, and 4 and 5 are coincident. a_2, d_4, and d_6 are the robot's link lengths.



(a) To find the DH parameters, we proceed frame by frame and note the transformations between each successive pair.

- Frames 0, 1, and 2 are configured in exactly the same manner as the anthropomorphic arm in the lecture notes. Rows 1 and 2 of the DH table are unchanged.
- Frame 2 transforms into frame 3 by two rotations: a +90 degree rotation about x_3, and a θ_3 rotation about z_2. This gives us row 3.
- Frame 3 transforms into frame 4 by a translation along z_3 equal to d_4 and a rotation about x_4 by -90 degrees. Joint 4 also rotates about z_3 by θ_4. This gives us row 4.
- Frames 4, 5, 6 are otherwise identical to the lecture example of the spherical wrist. The remaining DH rows are unchanged.

Link	a_i	α_i	d_i	ϑ_i
1	0	$\pi/2$	0	ϑ_1
2	a_2	0	0	ϑ_2
3	0	$\pi/2$	0	ϑ_3
4	0	$-\pi/2$	d_4	ϑ_4
5	0	$\pi/2$	0	ϑ_5
6	0	0	d_6	ϑ_6

(b) Generating the composite transformation from the table and then plugging in 0 for all joint angles, we obtain the end-effector pose

$$T_6^0 = \begin{pmatrix} 1 & 0 & 0 & a_2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -d_4 - d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

This is a rotation of 180 degrees about the x_0 axis, a translation of a_2 along the x_0 axis, and a translation of $-d_4 - d_6$ along the z_0 axis.

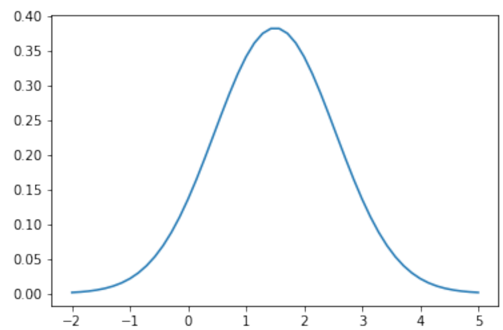
Problem 3: Bayes Filter (16 points)

We have a robot in an infinitely long hallway, located somewhere between $x = 0$ and $x = 1$. We will use a uniform distribution over this domain as our prior belief distribution $B(x_0)$.

(a) To compute $B'(x_1)$, we multiply the motion model with $B(x_0)$ and integrate over x_0. This reduces to just the integration of the Gaussian pdf but over the domain $[0, 1]$

$$\begin{aligned} B'(x_1) &= \int_0^1 \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x_1 - (x_0+1))^2\right) dx_0 \\ &= \frac{1}{2}\mathrm{erf}\left(\sqrt{2}(1 - 0.5x_1)\right) - \frac{1}{2}\mathrm{erf}\left(\frac{\sqrt{2}}{2}(1 - x_1)\right) \end{aligned}$$

This is a summation of two different error functions. The new pdf still looks like a Gaussian but is not quite the same. Note that the domain of the belief distribution now encompasses the entire infinite hallway, as there is a nonzero likelihood that the robot could have moved itself to any arbitrary location.

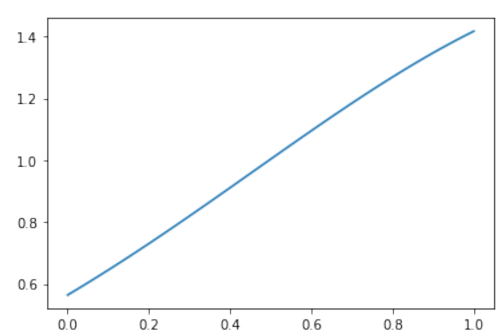


(b) To compute $B(x_1)$, we multiply $B'(x_1)$ by the observation model and normalize. Since $p(y_1 \mid x_1)$ is uniform over $[0, 1]$, this reduces to just keeping $B'(x_1)$ in this domain and then zeroing it out everywhere else. We compute the normalization factor as follows:

$$\eta = \int_0^1 B'(x_1)dx_1 = \int_0^1 \frac{1}{2}\mathrm{erf}\left(\sqrt{2}(1 - 0.5x_1)\right) - \frac{1}{2}\mathrm{erf}\left(\frac{\sqrt{2}}{2}(1 - x_1)\right) dx_1 \approx 0.241$$

We thus have the posterior pdf

$$B(x_1) = \begin{cases} 2.076\left(\mathrm{erf}\left(\sqrt{2}(1 - 0.5x_1)\right) - \mathrm{erf}\left(\frac{\sqrt{2}}{2}(1 - x_1)\right)\right) & 0 \le x_1 \le 1 \\ 0 & \text{otherwise} \end{cases}$$



Comparing $B(x_0)$ with $B(x_1)$, we are still completely confident that our robot is somewhere between $x = 0$ and $x = 1$. However, while we were ambivalent as to where inside this domain it is previously (hence the uniform distribution), there is now a higher likelihood that the robot is closer to the $x = 1$ side after having taken a noisy step in that direction.

Problem 4: Particle Filter Localization

4.4: Analysis and Discussion (20 points)

1. You should see that the particles generally approximate the robot's true state quite well. The particles start to diverge at the locations in which the robot senses no or only one landmark, but they quickly jump back to near the true location once new landmarks come within range.

2. For a larger value of `MAX_RANGE`, the robot is able to sense landmarks throughout its entire trajectory, so the uncertainty in our state estimate remains relatively constant. For smaller values, we lose sight of landmarks more often, causing the particles to diverge. If they are allowed to diverge for too long, all their weights tend toward 0, and the particle filter resets the entire estimator as it has lost track of where the robot is.

3. Having too few particles produces similar results as having a small sensing range. It is not so much a problem when there are landmarks within view to help establish the most likely particles, but once we lose sight of them, it is more likely for our entire particle set to have their weights decrease to 0.

4. Q expresses our uncertainty in the motion model. Too large means that we do not trust our motion model very much; the particles will tend to diverge more, even with landmarks nearby. Too small means that we are overly confident so that the particles will mostly stay converged near each other, which can be good for estimation but bad if our estimate starts to drift off (easier for weights to all go to 0 when that happens). R expresses our uncertainty in the observation model. Too large means that we do not trust our measurements very much, which allows for more particles to retain different weights relative to each other; less likely for particles to vanish, but a noisier state prediction history. Too small means that we overly trust our measurements, which does not produce too much trouble in this environment, but can lead us astray if our measurements were inherently very noisy.