CS W4733 Homework 3

Written part due Wednesday, October 28, noon. Programming part due Friday, October 30, 11:59PM

Part I: Individual Assignment, no teaming allowed

1. For each of the 4 robots below do the following: (12 points per robot, 48 points total)

   - (4 points) Draw a frame diagram showing each frame of the robot in the zero position
   - (4 points) Fiill in in the table of D-H parameters for each robot and show the resulting the transform matrices for each joint.

Joint	θ	d	a	α
1				
2				
3				
...	...	...	...	...
N				

   - (4 point) Describe each manipulator's work space.

   (a) Robot I: 3 Link R-R-R manipulator (see figure 1).

   (b) Robot II: 3 link R-R-R manipulator (see figure 2)

   (c) Robot III: 3 link R-P-R planar manipulator (see figure 3)

   (d) Robot IV: 3 link planar manipulator (see figure 4)

2. (a) (8 points) Solve the inverse kinematics for Manipulator B. Use the handout given in class for the forward kinematics matrix

   `http://www.cs.columbia.edu/~allen/F15/NOTES/forwardspong.pdf`

   (b) (4 points) Assume link parameters $a_1 = a_2 = a_3 = 1$. Using the inverse solution above, list ALL inverse solutions if the robot's end effector frame origin is at $P = [0\ ,\ 2\ ,\ 1\ ]$.

3. (a) (8 points) Solve the inverse kinematics for Manipulator D. Use the handout given in class for the forward kinematics matrix.

   (b) (4 points) Assume link parameters $a_1 = 2$ and $a_3 = 3$. Using the inverse solution above, list ALL inverse solutions if the robot's end effector frame origin is at $P = [1\ ,\ 0\ ,\ -1\ ]$.

Part II: Team Programing Assignment, you may work with your team partner(s) on this

1. (28 points) Create a 2-D occupancy grid for your robot environment, and map it out with cells that are either occupied or empty. You might think of this as how a roomba robot "learns" its environment so it can remember where to clean the floor and where obstacles are. Initially all cells in the grid are empty, and as the robot navigates the environment it fills in the grid spaces as free space or obstacles are encountered. The robot's behavior should be random, in that it goes in a direction that is randomly chosen until it hits an obstacle, at which point it changes its behavior to continue exploring the environment and creating a map. A good guide for this behavior is the "cover" demo program which covers a room using a combination of behaviors, such as random bounce, wall following, and spiraling. You can use any behavior you like, just output an occupancy grid map of the environment when the program finishes (see figures on last 2 pages of this document). Some details:

   - Assume a grid cell size of the diameter of the create robot.
   - Stopping conditions: if your robot has not updated any new cells after a certain amount of time has passed, you should consider stopping.

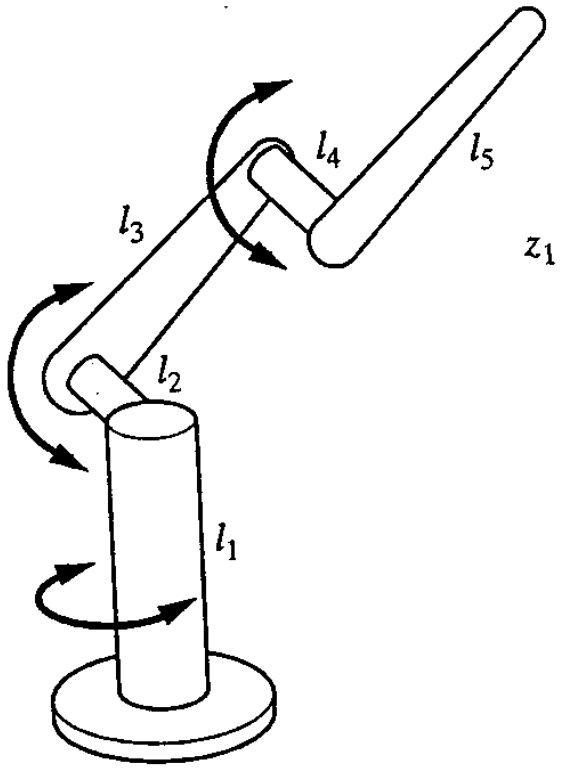


Figure 1: Robot I. Assume joint variables $\theta_1, \theta_2, \theta_3$

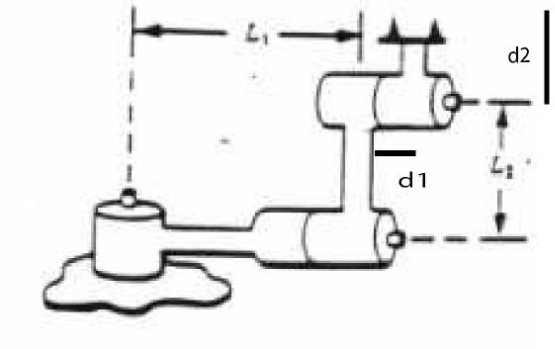


Figure 2: Robot II. Assume joint variables $\theta_1, \theta_2, \theta_3$

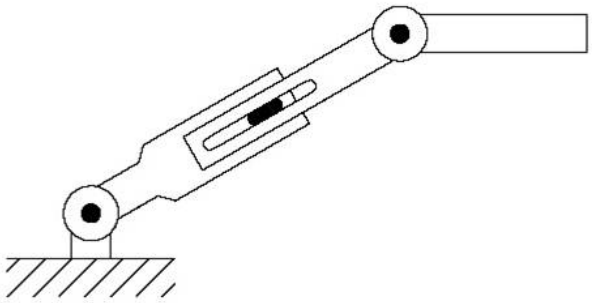


Figure 3: Robot III. Assume joint variables θ_1, d_2, θ_3, a link length of link 3 =l_3

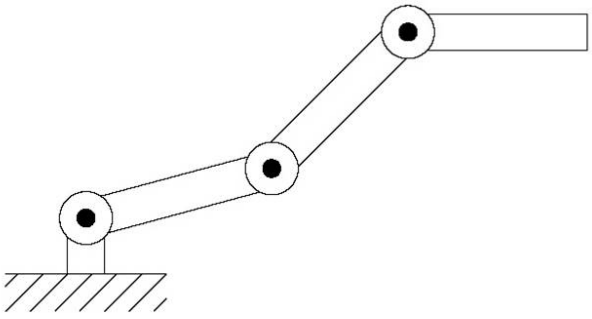
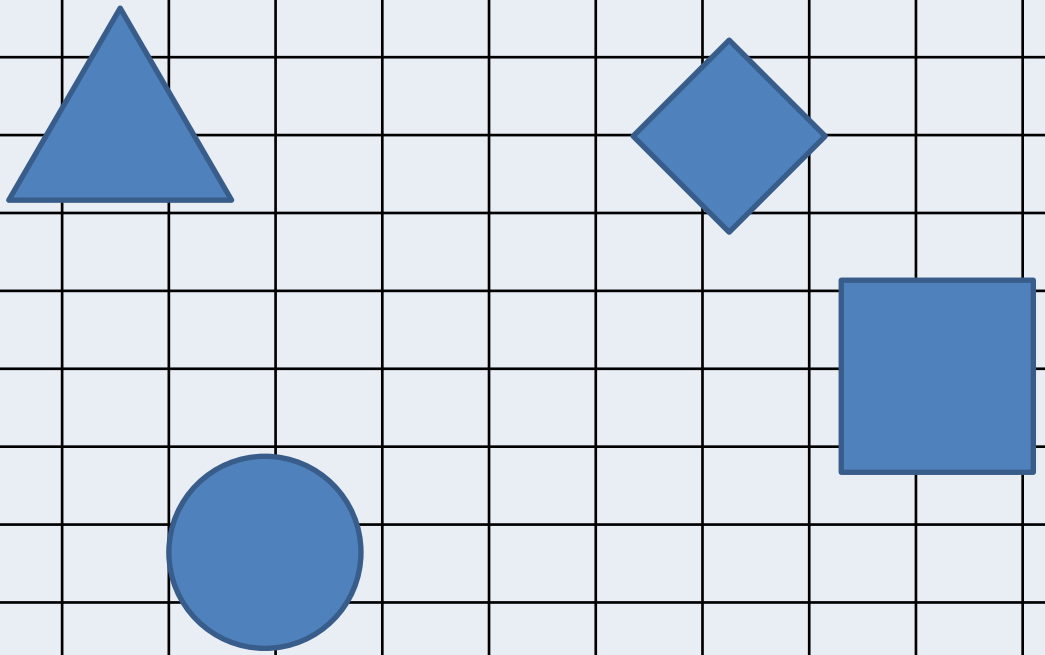
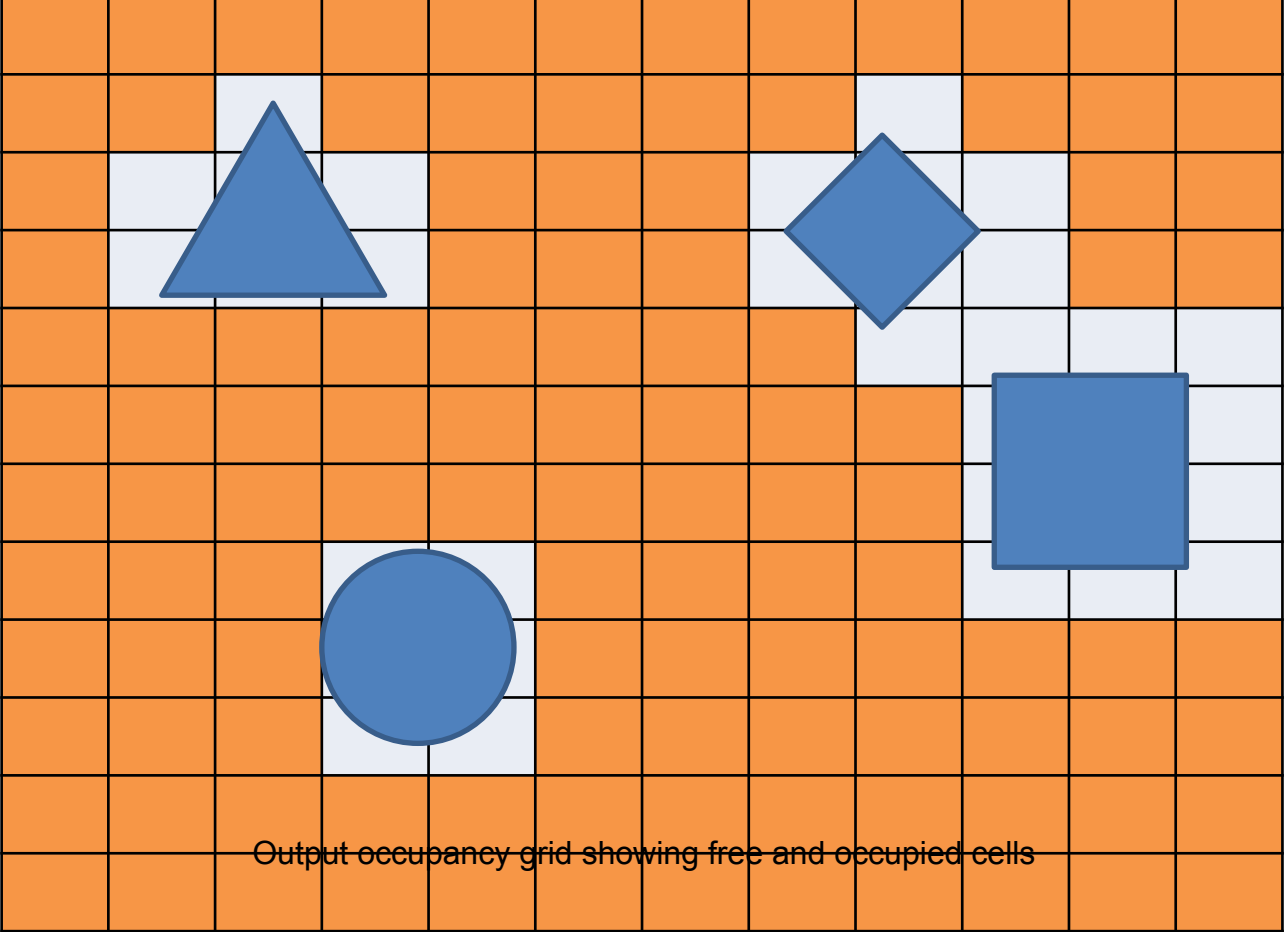


Figure 4: Robot IV. Assume joint variables $\theta_1, \theta_2, \theta_3$, and link lengths of l_1, l_2, l_3



Environment Before mapping


Output occupancy grid showing free and occupied cells