CS26N
Motion Planning for Robots, Digital Actors and Other Moving Objects
(Winter 2012)

Homework #4
Posted: February 29 – Due: March 7

How to complete this HW: First copy this file; then type your answers in the file immediately below each question; finally print this file and return it to the Wednesday class on the day indicated above. If you need to draw anything, you can do it by hand.

Your name: ……………………………………………………………………………………….

Your email address: ………………………………………………………………………….

Note: You may discuss this problem set with other students in the class (in fact, you are encouraged to do so).

Grading:

Question#	Max. grade	Your grade
I	20 (10 + 10)	
II	30 (15 + 15)	
III	20	
IV	30 (10 + 10 + 10)	
Total	100	

Problem I: Collision Detection for Haptic Interaction

Assume a 3D virtual world containing models of rigid objects. These objects are visualized on a graphic display. A haptic interface (similar to the one you have seen during the visit of the Robotics Lab with Prof. Khatib) is also available to "feel" them. Like a 3D mouse, this haptic interface allows you to move a point **P** in the virtual world. Whenever **P** is outside the objects, the null force is rendered on the haptic interface. Whenever **P** is inside an object, a force is rendered proportional to the penetration distance (the minimum distance by which **P** should be moved to take it out of the object).

Assume that the haptic system uses a BVH (bounding-volume hierarchy) checker to detect collisions.

1. Explain why, in this specific application, collision checking can be done very quickly by the BHV checker. [Hint: What are the two "objects" checked for collision?]

2. Imagine that one object is a very thin plate of metal. If one checks whether **P** at successive positions collides with the plate, it may be possible to miss a collision. For instance, the first successive positions of **P** are on one side of the plate, while all the subsequent positions are on the other side. Due to discretization, none falls *inside* the thin plate. How could you slightly change the collision checker to avoid missing collisions and detect the collision occurred?

Problem II: Configuration Space of a Legged Robot

Consider the planar robot shown in Figure 1. It consists of a "body" depicted by the blue oval and two identical "legs" depicted by thick lines. Each leg has two links and two revolute joints.

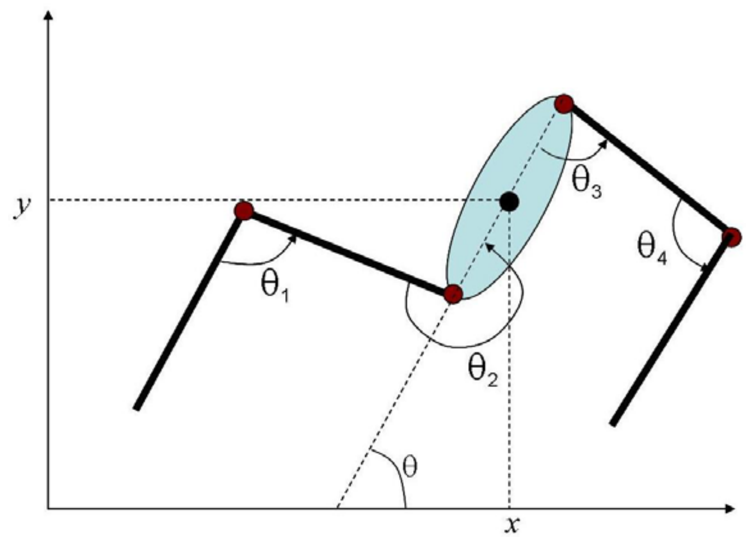

Figure 1: Two-leg robot

Recall that a configuration of a robot is a list of parameters that uniquely defines the position and orientation of each rigid body in the robot. So, if we assume that the robot of Figure 1 can move freely in the plane, a configuration of this robot can be represented by $(x, y, \theta, \theta_1, \theta_2, \theta_3, \theta_4)$, where x and y are the coordinates of the center of the body, θ is the angle in $[0, 2\pi]$ that defines the orientation of the body, θ_1 and θ_2 (each in $[0, 2\pi]$) are the angles associated with the revolute joints of the left leg, and θ_3 and θ_4 are the angles associated with the revolute joints of the right leg. Hence, the robot's configuration space has dimensionality 7.

1. Assume now that we fix the position of the endpoint of the left leg. What is the dimensionality of the subspace of configurations satisfying this constraint? How would you represent a configuration of the robot under this constraint? [Note: This representation may not be a subset of the 7 parameters specified above. You may use different parameters if you think it's more convenient.]

2. Assume now that we fix the positions of the two endpoints of the two legs. What is the dimensionality of the subspace of configurations satisfying this constraint? How would you represent a configuration of the robot under this constraint? [This question is rather difficult.]

Problem III: Coordination of two robots

In Figure 2 two square robots A and B move in a 2D workspace. Each side of a robot has length equal to 4 units. The robots cannot rotate. Each moves at fixed orientation along a distinct track so that its center remains on a bold line shown in the figure. The robots must not collide with each other.

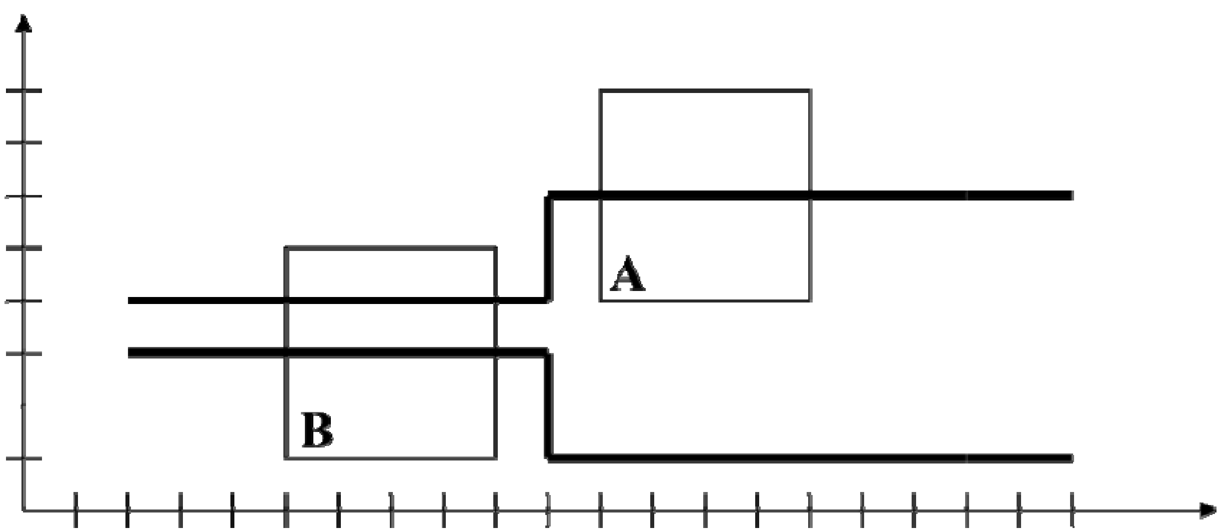

Figure 2: Two-robot system

Even though there are two robots, we can still represent both of them as a point in a 2D configuration space. To help you, we tell you that this configuration space is the 20×20 square shown in Figure 3. The horizontal axis corresponds to the position of robot A along its track and the vertical axis corresponds to the position of robot B. So, a point in this 2D configuration space uniquely defines the positions of the two robots. Assume that (0,0) would place each of the two robots at its leftmost position on its track. Similarly (20,20) would place each robot at its rightmost position.

The grid shown in Figure 3 is intended to help you answer the two questions of this problem precisely. Each small square of this grid has size 1 unit × 1 unit. The unit of length in Figure 3 is not shown at exactly the same scale as the unit of length in Figure 2.

1. In Figure 3, mark by a thick dot the point representing the configuration of the robots shown in Figure 2.

2. In Figure 3, draw the forbidden region where the two robots collide or touch each other.

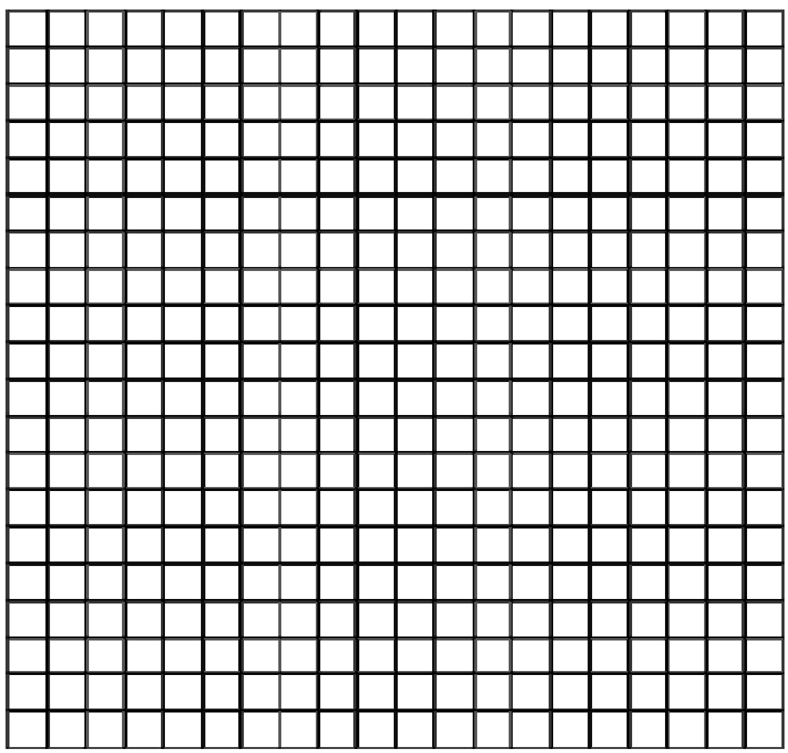

Figure 3: 2D configuration space of the two-robot system

Problem IV: Mobile robot

In this problem, we consider a cylindrical mobile robot with three wheels, two *driving* wheels and one *free* wheel in front. The driving wheels are the wheels that make the robot move. By rotating at different speeds they can make robot turn right or left. The free wheel can rotate passively about a vertical axis, so that during a turn of the robot it automatically adjusts its orientation. The purpose of the free wheel is only to prevent the robot from toppling over. (To improve stability there could be a second free wheel on the back.) In the following we will ignore the free wheel.

We model the robot by a disc in a planar workspace. Figure 4 shows this disc and the positions of the wheels. The driving wheels are shaded. The midpoint between them is exactly the center of the disc. Their angular velocities can be controlled independently of each other, with each wheel rotating at any velocity in [-ω, +ω]. The only two parameters directly controlled by the robot are the angular velocities of its two driving wheels.

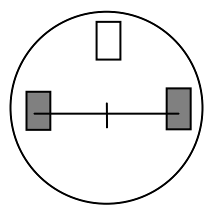

Figure 4: Mobile robot

1. Can this robot rotate at a fixed position? If yes, how?
2. Based on your answer to Question 1, how would you define a configuration of this robot? What would then the dimensionality of the configuration space? Explain (1) how a grid-based planner would plan a path for this robot and (2) assuming that this path is a polygonal line, how the robot would execute it.
3. Now, assume this robot is loaded with a large non-circular object (e.g., a rectangular box) that extends beyond the disc's boundary. How would you define a configuration of this loaded robot? What would then be the dimensionality of the configuration space? How could a path be planned for this robot?