RBE-550 Motion Planning
Daniel Montrallo Flickinger, PhD

Valet

DUE: 2022-10-31 @ 11:00 UTC

Abstract

A common path planning problem for autonomous vehicles involves maneuvering in tight spaces and cluttered environments, particularly while parking. Create a simulated world and park three vehicles of increasing ridiculousness into a compact space. Planners must take into account vehicle kinematics and collision.

1 | Introduction

Use kinematic planning under nonholonomic constraints to efficiently park several vehicles, like in Figure 1. The vehicles range from a diwheel configured delivery robot, to a standard car, to a truck with a trailer. Create a simulated world, and implement a kinematic path planner that takes into account the nonholonomic constraints, and obstacles.

Figure 1: The parking problem

2 | Methods

Refer to Section 13.1.2 in LaValle [2006] for examples of kinematics of wheeled systems, and Section 4.2.1 in LaValle [2006] for modeling configuration space of rigid bodies in two dimensions. A kinematic-only solution is acceptable, though a full simulation utilizing dynamic equations of motion and time-stepping will earn 10% extra credit on this assignment.

3 | Environment

Create kinematic models of cars with different steering mechanisms, then move them to park curbside. In each instance, a car starts at the Northwest corner of a bounded two dimensional field. Park the car is a compact space at the Southern border, flanked by vehicles both in front of and behind the target space. Additionally, there is an island in the central region. All wheels remain

in contact with the ground, and skidding is disallowed. (i.e., nonholonomic constraints are strictly enforced in this environment.) For parking the truck with trailer, omit one of the obstacle cars. See Figure 2 for a diagram. Dimensions are arbitrary, pick values that approximate the obstacles as depicted.
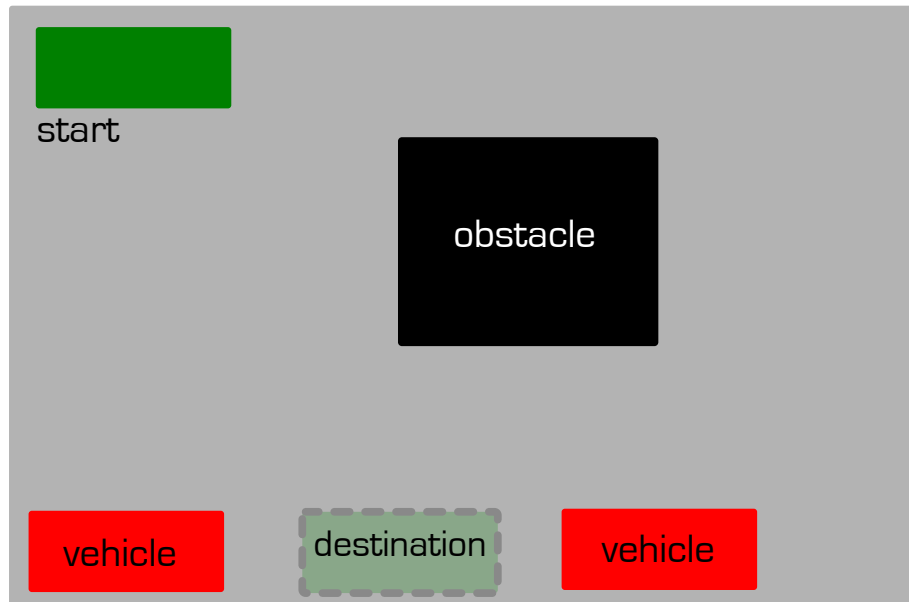

Figure 2: The parking lot

3.1 The Delivery Robot

First up is the delivery robot, seen in Figure 3. Is it delivering burritos? Is it delivering essential medicine? Who knows. All we know is that it should avoid Philadelphia, after what happened to HitchBOT. It has skid steering, but assume that it has diwheel kinematics for this simulation. Refer to Figure 4 for the kinematics.

3.2 The Car

Next, the trusty old police car in Figure 5 gets a try. If it can move under its own power, maybe it won't get scrapped for another month. Standard Ackerman steering (Explained) affords some maneuverability, but it can't execute zero radius turns. Wheelbase (L) is 2.8 m. Kinematics are shown in Figure 6.

Figure 3: The delivery robot

3.3 The Truck

Finally, it's truck time. The neighbors really want it gone, along with all the beat up old refrigerators and air conditioner condensers littered about the property. It also has Ackerman steering (Explained). Axle width is 1.75 m, wheelbase is 3.0 m, and the distance between the rear axle of the truck and the axle center of the trailer (d_1) is 5.0 m. The kinematics for a vehicle with multiple trailers is given in Figure 8.

4 Software

The most straightforward implementation is to write a basic kinematic simulation system from scratch with Python, MATLAB, or similar. Implement the kinematics, time-stepping, and basic graphical output, and use built-in utilities to generate plots of the path. Full-featured vehicle simulation systems are generally overkill for this assignment, and might take more time to set up. However, a research platform such as Carla Dosovitskiy et al. [2017] could be useful.

5 Results

For each vehicle, prepare a plot of the calculated path, for the center of the rear axle. Additionally, produce either a short video, or superimpose periodic snapshots onto an image to depict the movement of the vehicle.

Submit all source code, organized into functions, classes, associated files, etc. Additionally, submit flow diagrams or pseudocode for the planner algorithm. Also submit a brief report with all generated figures and videos.

6 Grading and Submission

This assignment is due 2022-10-31 @ 11:00 UTC. *Late submissions are not accepted.* Upload completed assignment components (as individual files, not a single ZIP or TAR file) to the course site on Canvas.

Weight	Type	Component
25%	tar.gz	full source code
10%	PDF	planner algorithms (diagram or pseudocode)
20%	PDF,video	path animation
25%	PDF	path depiction
20%	PDF	report
10%	note	EXTRA: dynamic simulation

7 References

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

Engineering Explained. Ackerman steering - explained. URL `https://www.youtube.com/watch?v=oYMMdjbmQXc`.

Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, May 2006. ISBN 9780521862059. URL `http://lavalle.pl/planning/`.

8   List of URLs
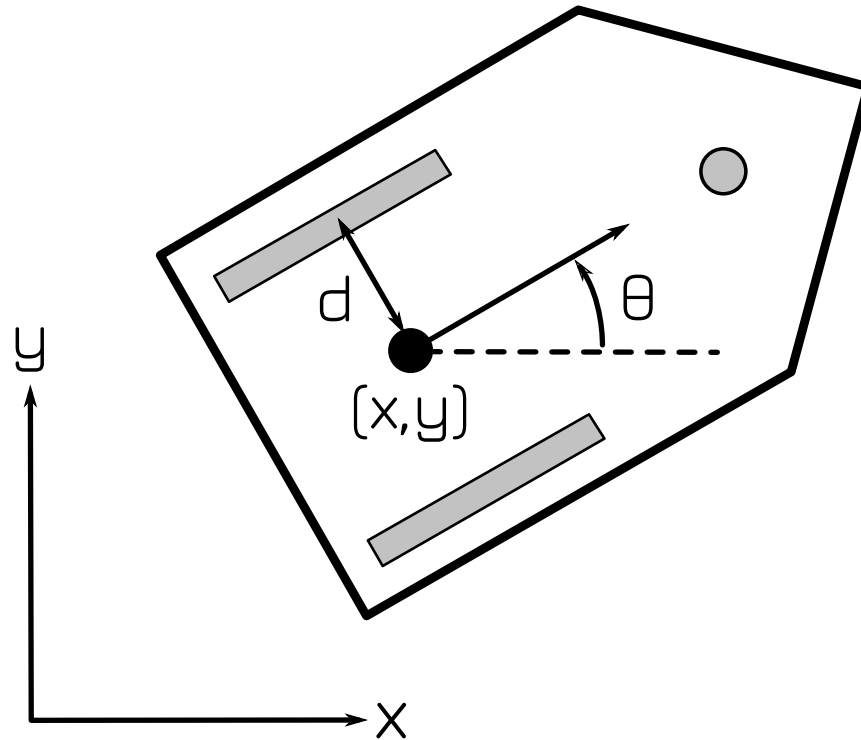
Last update: August 16, 2022

Figure 4: Differential drive (or skid steer) kinematics
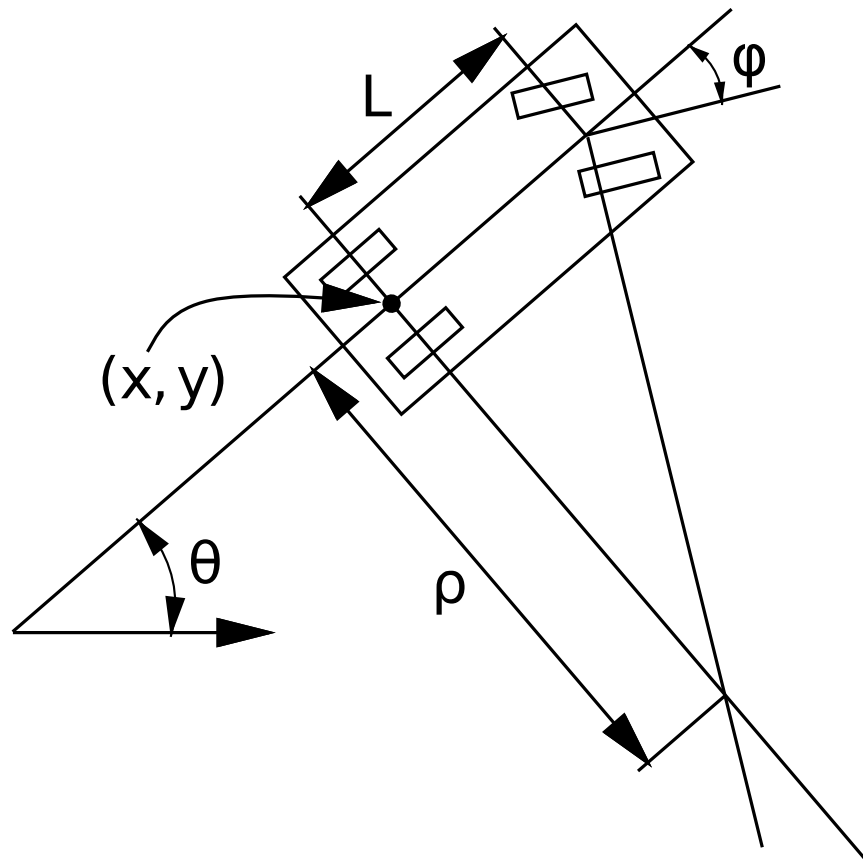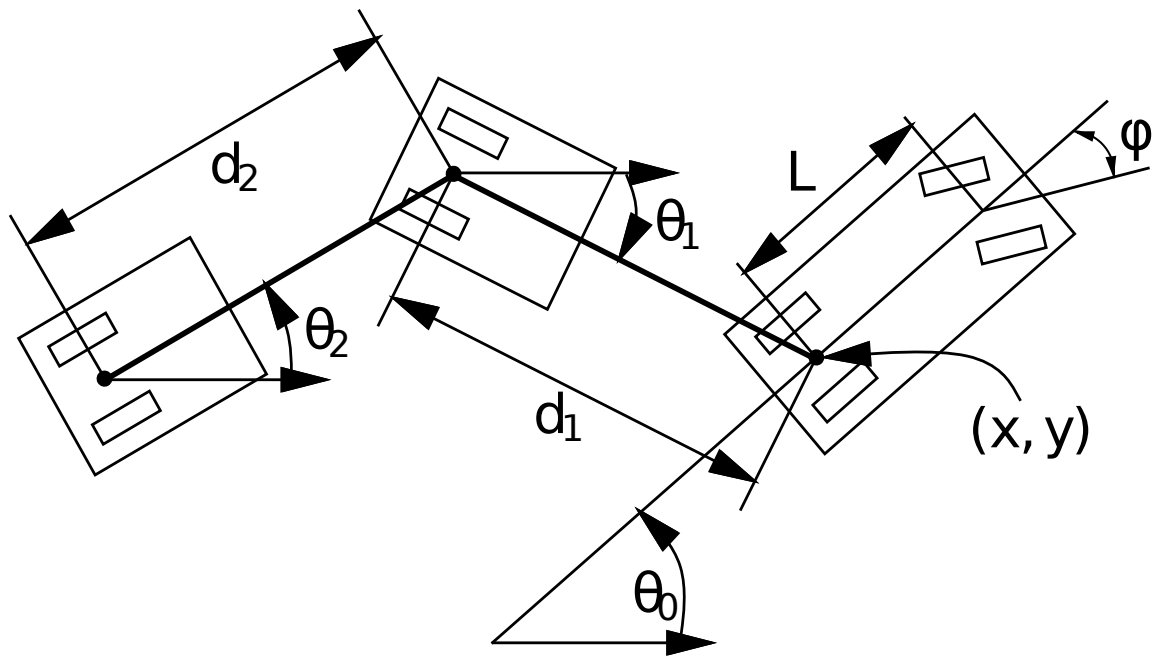

Figure 5: The car

Figure 6: Ackerman steering kinematics

Figure 7: The truck

Figure 8: Trailer kinematics