AE 5335 – Assignment 2

Collaboration Policy

This is an "open-book and open-notes" assignment, which means that consulting lecture notes/slides provided by the instructor and/or similar educational material from sources on the Internet is allowed. Using MATLAB® or similar software is allowed unless stated otherwise for specific problems. Discussions about this assignment with other students in the class are allowed and encouraged. However, the submitted assignment must reflect your own independent work. Plagiarism from any source is disallowed.

Statement of Academic Honesty: When submitting this assignment, you will be asked to state:

My signature below affirms that this submission is entirely my own work, and reflects my own understanding of the course content. This submission does not contain material copied or plagiarised from any other source including works of other students. I have read and I understand WPI's Academic Honesty Policy, and my conduct in preparing this submission has been in accordance with this Policy.

Instructions for Assignment Submission

- The submission format is a *single* PDF file, to be uploaded via *Canvas*. Multiple files will not be graded.
- Read WPI's Academic Honesty Policy, and the Student Guide to Academic Integrity.
- All deadlines provided are in Eastern Daylight Time. Late submissions will receive a grade penalty as follows.
  - Late by up to 48 hours after deadline: 10% penalty.
  - Late by 48 - 96 hours after deadline: 20% penalty.
  - Late by more than 96 hours after deadline: 30% penalty.
- The solution to each problem should be presented in the following format:
  - **Method:** e.g., "I did [insert basic outline of method] and obtained the following results." OR "I used the [insert name of command/tool/library] from [MATLAB or similar software]. This command/tool/library [insert brief explanation of what the tool does and/or how it works]."
  - **Results:** e.g. "The following plots/numbers/expressions indicate [what they indicate; insert plots / numbers, expressions etc]."
  - **Discussion:** e.g. "The [results above; details] match expected [behavior/values] because [reasons]." OR "The [results above; details] match expected [behavior/values], but may include minor numerical errors because [reasons]." OR "The [results above; details] are not sensible because [explain why you think the results are not correct]. My method appears to be correct, but I was not able to resolve [issue] despite trying [describe your attempts, including discussions, if any, with the instructor and/or TA]."
- Submissions in the form of software code (e.g., MATLAB® code) without an explanation of the method and discussion of results are not acceptable, and such submissions will be returned without grading. If and when code is included in the submission, it must be included as an appendix, not in the main submission.
- Illegible and/or untidy submissions will not be accepted or graded, and the students will forfeit points unless the work is entirely resubmitted. For these resubmissions, lateness penalties as above will apply.
- To scan handwritten work into a PDF file, use either a desktop scanner or a smartphone-based software application such as CamScanner, Microsoft Lens, or Adobe Scan. Watermarks, if any, left by free versions of these software applications are acceptable.
- Consult this page for examples of acceptable and unacceptable assignment submissions.

Problem 1. (20 points)

Consider the obstacle-ridden environment shown in Fig. 1 (nondimensional units). The areas shaded in gray are obstacles. For simplicity, all obstacles are rectangles, and the coordinates of vertices of these rectangles can be inferred using the gridlines shown in Fig. 1.

The objective is to find a shortest path from Start to Goal. The start and goal locations have coordinates $(9.5, 10.5)$ and $(-11.5, -11.5)$, respectively. The attached sample code provides a MATLAB® script to generate Fig. 1.

- Decompose the environment into obstacle-free cells. There is no restriction on your choice of shapes and sizes of these cells. The only requirement is that the decomposition be *algorithmic*, i.e., follow rules that can be programmed in software and not be chosen "by hand" for this specific environment.

  For the purposes of this assignment, you need not necessarily program your decomposition rules in software; a demonstration "by hand" of the application of your algorithm will suffice.
- Show your cell decomposition for this environment (overlay cells on Fig. 1).
- Define a cell topological graph, i.e., write the list of vertices and edges. If your graph has a large number of vertices (too many to write down), write down the rationale for defining vertices and edges. Provide a few examples.
- Find a shortest path in this graph, and plot it by overlaying on Fig. 1.

Problem 2. (20 points)

Find the shortest Dubins path between each of the following pairs of configurations. The minimum radius of turn is 150 m.

(i) Initial: $\boldsymbol{p}_0 = (0, 0)$ km, $\chi_0 = 0°$, final: $\boldsymbol{p}_\mathrm{f} = (1, 1, -0.1)$ km, $\chi_\mathrm{f} = 0°$.

(ii) Initial: $\boldsymbol{p}_0 = (1, 1, -0.1)$km, $\chi_0 = 0°$, final: $\boldsymbol{p}_\mathrm{f} = (1.25, 1)$km, $\chi_\mathrm{f} = 90°$.

Find the total length of the shortest path in each case.

Problem 3. (20 points)

Find a minimum-length polynomial trajectory that passes through the following sequence of waypoints. Each waypoint consists of a 2D (xy) position $\boldsymbol{p}$ and a heading angle χ specified in a NED axes system.

Waypoint 1: $\boldsymbol{p}_1 = (0, 0)$ and $\chi_1 = 10$.

Waypoint 2: $\boldsymbol{p}_2 = (10, -5)$ and $\chi_2 = 50°$.

Waypoint 3: $\boldsymbol{p}_3 = (30, 5)$ and $\chi_3 = 120°$.

Find the total length of the path. Plot the path and the radius of curvature of the path as a function of arc length.

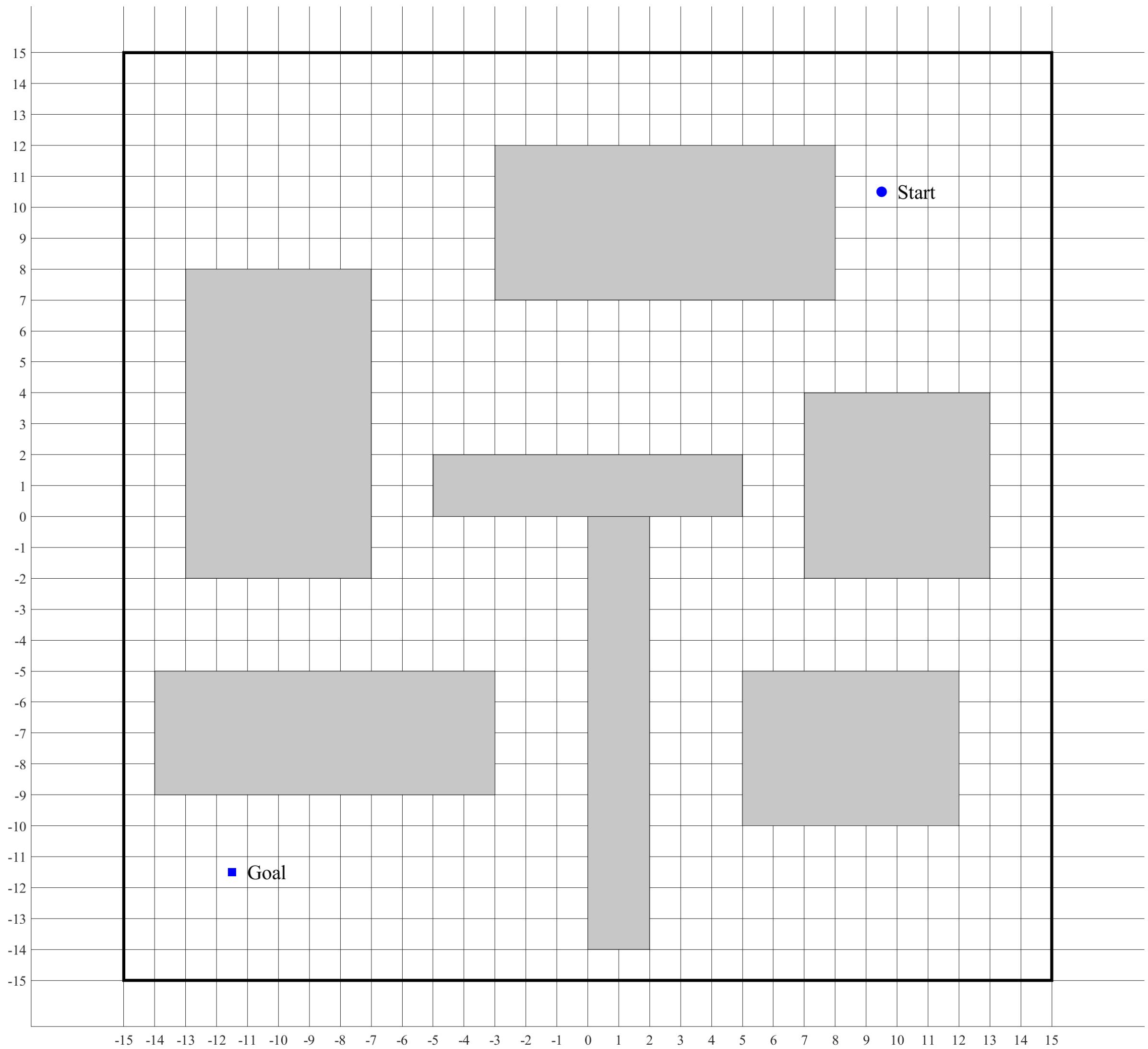

Figure 1: An obstacle-ridden environment.

Problem 4. (20 points)

Find a polynomial path $\boldsymbol{p}(t)$ that passes through the sequence of waypoints (locations and headings) given in the previous problem. In this case, the objective is to find a trajectory with lower curvature, i.e., higher radius of curvature than the minimum-length trajectory. To this end, we would like to minimize

$$\int_0^1 \left(\left(\frac{\mathrm{d}s(t)}{\mathrm{d}t} \right)^2 + \mu \left\| \frac{\mathrm{d}^2 \boldsymbol{p}(t)}{\mathrm{d}t^2} \right\|^2 \right) \mathrm{d}t,$$

where μ is a normalizing constant of your choice.

- Formulate and solve this problem as a quadratic program.
- Discuss the effects on optimal trajectories resulting from different choices of the constant μ.
- Compare this trajectory with the result of the previous problem, including a comparison of the radius of curvature.

Problem 5. (10 points)

Prepare a short ($\sim$ 500 words) summary of the following article provided on *Canvas*:

R. V. Cowlagi and P. Tsiotras, "Hierarchical motion planning with dynamical feasibility guarantees for mobile robotic vehicles," *IEEE Transactions on Robotics,* vol. 28, no. 2, pp. 379-395, April 2012.
https://ieeexplore.ieee.org/document/6075270

Explain what you consider as the main ideas, main findings, and/or weaknesses of the work reported in this article.

Problem 6. (10 points)

Prepare a short ($\sim$ 500 words) summary of the following article provided on *Canvas*:

R. Du and R. V. Cowlagi, "Interactive sensing and planning for a quadrotor vehicle in partially known environments," *Journal of Guidance, Control, and Dynamics,* vol. 42, no. 7, pp. 1601–1611, July 2019.
https://doi.org/10.2514/1.G003773

Explain what you consider as the main ideas, main findings, and/or weaknesses of the work reported in this article.