EC4.403: Robotics: Planning and Navigation **Spring 2021**

Assignment-2

Released: March 08th, 2021 *Deadline: March, 28th, 2021*

Instructions

- Submit your code files and a report (or everything as a single jupyter notebook if you wish) as a zip file, named as ⟨TeamName⟩.zip.
- The report should briefly describe your approach/algorithm used, your final results and any other findings.
- The deadline is March 28th, 23:55. This is a hard deadline and no submissions will be accepted after this.
- Plagiarism detectors will be run on all submissions, so please do not copy. If found, you would be given a straight zero for the assignment.

In this assignment we will pose local planning as a optimization problem with constraints and use predictive control technique called as Model Predictive Control(MPC) to execute the plan. You need to have some understanding of optimization [**1**]. Please use the resource attached for discrete MPC(available on Moodle, MPC subsection).

Question:
Implement a discrete MPC planner for omni-wheel robot. You must implement the MPC algorithm for a two cases (i) With Obstacles (ii) Without Obstacles. You can use solvers like cvxopt in python or any other equivalent in Matlab.

- Your planner for a robot needs to satisfy various constraints on speed, acceleration, obstacle avoidance to make it feasible. Additionally your plan needs to optimize some aspect in your environment like speed, time or safety. Your plan should be n steps into the future.
  You should frame Quadratic Programming QP for your planner to minimize goal reaching cost given by:
  $$(x_n - x_g)^2 + (y_n - y_g)^2$$
  robot model is given by
  $$x_{t+dt} = x_t + v_{xt} * dt$$
  $$y_{t+dt} = y_t + v_{yt} * dt$$
  velocity constraints are given by
  $$0 \le v_{xi} \le v_{\max} \quad \forall i \in [1, n]$$
  $$0 \le v_{yi} \le v_{\max} \quad \forall i \in [1, n]$$
- For (i)(with obstacles) : Extend the planner by adding circular obstacles. You can use euclidean distance constraint for obstacle avoidance. This adds Quadrtic constraint to Your problem. For example an static obstacle of radius r_1 which is located at (x_o, y_o) then your constraint is given by.
  $$(x_i - x_o)^2 + (y_i - y_o)^2 \ge r_1^2 \quad \forall i \in [1, n]$$

References

[1] https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf

Deliverables:

- Code for both the cases. You can use Python, MATLAB or whichever language you are comfortable with. Please ensure the code is well written, and we can ask you to explain certain snippets of it during the evaluations.
- A report summarising your understanding of the algorithms and explaining the results.
- Put all these files in a folder ⟨TeamName⟩, zip it, and submit it on moodle.

Feel free to reach out to the TA's for any queries. All the best, and may the Force be with you.