Version 2024.1

Assignment 5

Mathematics for Robotics

C. Gummaluru, A. Shoelling *University of Toronto*

This assignment is part of the *Fall 2024* offering of *ROB310: Mathematics for Robotics* at the University of Toronto. Please review the information below carefully. Submissions that do not adhere to the guidelines below may lose marks.

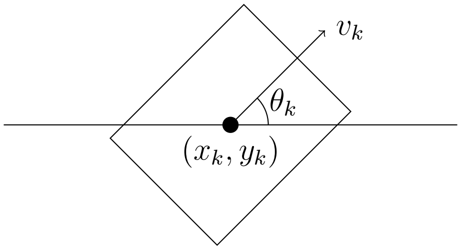
Release Date: .... October 28, 2024 - 00:01
Due Date: .... November 10, 2024 - 23:59
Weight: .... 3.75% or 0%
Late Penalties: .... -10% per day and -100% after 3 days

Submission Information:

- Download this document as a PDF and fill in your information below.
- Answer the questions to the best of your ability.
- Typeset or **neatly hand-write** your answers, labelling the questions identically to this document. We recommend using LaTeX, but will accept the use of other typesetting software (such as Microsoft Word or Google Docs).
- Save each question as a separate PDF document. Do NOT include the questions.
- Submit your PDFs on Crowdmark titled `lastname-firstname-rob310f24-a5-q#.pdf`.

Problem 5.1

Consider moving a robot along a straight line. We model the robot as a unicycle with sampled inputs, v_k, ω_k at time-step k, where v_k represents a pre-defined desired forward velocity and ω_k represents the turn rate input.



We use the tun rate command, ω_k, to move the vehicle onto the line and keep it on the line. The discrete-time dynamic equations describing the robot behaviour are given by:

$$\mathbf{x}_{k+1} = \begin{bmatrix} y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} y_k \\ \theta_k \end{bmatrix} + T \begin{bmatrix} v_k \sin\theta_k \\ \omega_k \end{bmatrix}$$

where θ_k is the heading of the robot at time-step k, T is the discrete time-step (in seconds), v_k is given, and ω_k is to be optimized. The turn rate is constrained by

$$\omega_{\min} \leq \omega_k \leq \omega_{\max}.$$

The goal of the optimization problem is to find a sequence, $\{\omega_k\}_{k=0,\ldots,N-1}$ that minimizes the following cost-function for a given initial condition:

$$\sum_{k=1}^{N} \mathbf{x}_k^\top \mathbf{Q} \mathbf{x}_k + \sum_{k=0}^{N-1} r\omega_k^2$$

where $\mathbf{Q} \in \mathbb{R}^{2,2}$ and $r \in \mathbb{R}$ are given weights that balances tracking accuracy and control effort with $\mathbf{Q} = \mathbf{Q}^\top$ being positive semi-definite and $r > 0$.

Part 5.1.a

We make a few simplifying assumptions:

1. the angles θ_k are sufficiently small so that $\sin\theta_k \approx \theta_k$
2. the pre-defined forward velocity, v_k, is constant with $v_k = v$ for all k

Write (1) in standard linear form:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k.$$

Part 5.1.b

Based on your answer to Part 5.1.a, formulate the optimization problem. Assume that we optimize over $\mathbf{x}_k, k = 1, \dots, N$ and $\mathbf{u}_k, k = 0, \dots, N-1$, with the system dynamics being equality constraints. Clearly state the objective function, equality constraints, and inequality constraints. You may end up with large matrices and vectors.

Part 5.1.c

Is the problem in Part 5.1.b convex? Provide a brief explanation.

Part 5.1.d

Solve the problem in Part 5.1.d using `Python`. Assume that $N = 1000$, $T = 0.01$ s, $v = 1\frac{\text{m}}{\text{s}}$, $\omega_{\text{min}} = -\pi/4\frac{\text{rad}}{s}$, $\omega_{\text{max}} = \pi/4\frac{\text{rad}}{\text{s}}$, $y_0 = 1.5$ m, $\theta_0 = 0$ rad, $r = 1$, and $\mathbf{Q} = \text{diag}(1, \dots, 1)$. Provide your code and plots of the sequences $\{y_k\}$ and $\{\theta_k\}$ over time.

Part 5.1.e

How does the result in Part 5.1.d change as you change the diagonal entries of $\mathbf{Q}$? Provide plots to support your answer.

Part 5.1.f

The optimal sequence $\{\omega_k\}$ found in Part 5.1.d is used to drive the robot with initial condition $y_0 = 1.5$ m, $\theta_0 = 0$ rad, but the robot dynamics are now corrupted by an unknown disturbance such that

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \begin{bmatrix} 0 \\ \theta\pi/25 \end{bmatrix}$$

Simulate the robot with the new dynamics in (5) and the optimal sequence $\{\omega_k\}$ found in Part 5.1.d. Provide your code and plots of the sequences $\{y_k\}$ and $\{\theta_k\}$ over time. What is the problem here?