Tasks

- Create a real-time MPC system in Simulink to manoeuvre a laboratory-scale gantry crane inside a constrained region
- Write a concise (one-page) report detailing your controller implementation and relevant design choices

Summary

In this coursework, you will implement a closed-loop MPC system in the provided Simulink template. The controller should be designed to handle arbitrary constraint sets that take the form of a wedge (e.g. two rectangles joined at a 90° angle) as shown in Figure 1. This will involve the crane moving inside the defined region from a starting point to a set target point. The sides of the region represent constraints that should not be violated by either the center of the cart position or the pendulum position. The end points for each line segment (e.g. the corners of the shape) will be provided to your controller at the beginning of the run, and then the shape will remain constant during the run.

Project Requirements

Target Point

The crane will start at the point (x_s, y_s), and the target point will be located at the point (x_t, y_t). The crane will be considered at the target point once the pendulum position and the cart position are within a circle of radius ϵ_t from the target point, and all inputs and rate states have decayed below a threshold ϵ_r. You will not be provided with any intermediate points between the start and target points, so it is your responsibility to decide how your controller traverses the region. Note that the start and target points can be located in the same side of the wedge.

Constraint Shape

The constraints are composed of two rectangles at 90° angle to each other, and at an arbitrary angle in the xy plane, as shown in Figure 1. The crane is allowed to move anywhere inside the region, but must not leave it. The constraints are defined by the 6 corners of the rectangle, which will be provided to you inside a 6x2 matrix of the form

$$c = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \\ x_5 & y_5 \\ x_6 & y_6 \end{bmatrix}.$$

These points will always be provided in the clockwise sequence shown in Figure 1, with point 2 always being on the outer vertex of the bend and point 5 always being on the inner vertex of the bend.

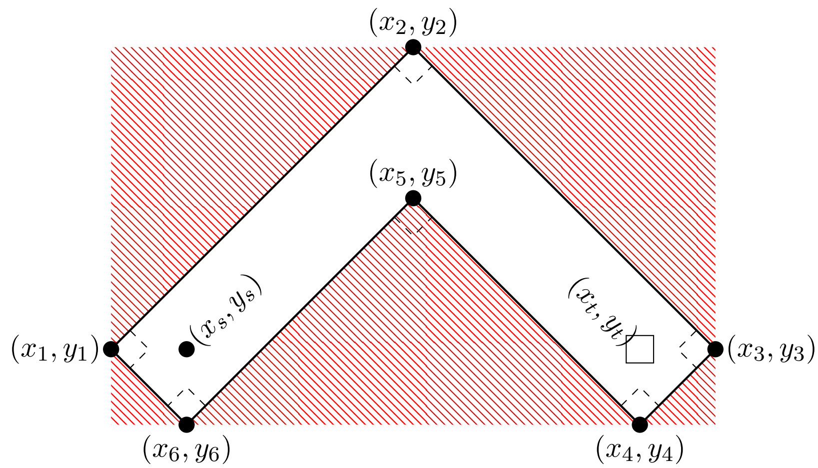

Figure 1: The shape the crane must stay inside. The allowed region is in white, forbidden region in red.

Approach

Physical System

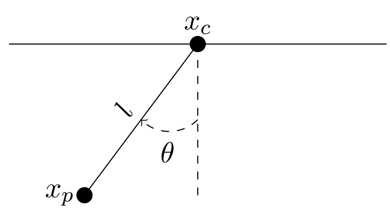

Figure 2: Pendulum and cart

This coursework will use the laboratory-scale gantry crane that you have been using throughout the course (and which you derived the model for in coursework 1). Note that all distances will be measured in meters and all angles will be measured in radians. The pendulum length will be fixed at 0.47m, and all constraints must be satisfied by both the pendulum and the cart positions. The pendulum is assumed to be a point-mass located at the end of a rigid string of length l=0.47m, as shown in Figure 2. Each angle is measured from the vertical line passing through the cart, with θ representing rotation about the y-axis (as shown in Figure 2), and ψ representing the rotation about the x axis.

Controller Framework

You will be responsible for writing 4 MATLAB functions for this project. The first function will run before the simulation starts, and can be used to compute matrices and other values needed during the simulation. The 3 other functions will run inside MATLAB function blocks in a Simulink model and actually perform the control and state estimation. The architecture that you are provided inside the Simulink file is shown in Figure 3. More information about using the framework can be found on Blackboard. Note that both the software and hardware portions of this coursework will be marked

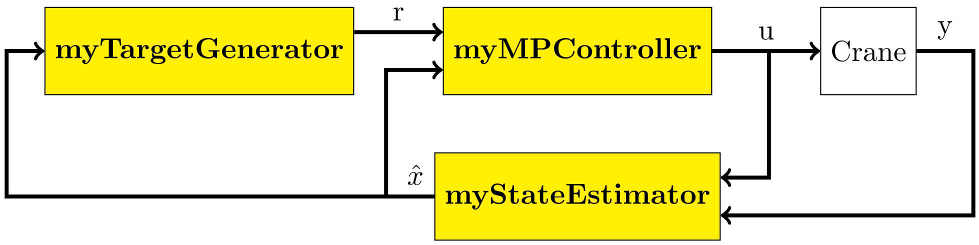

Figure 3: Block diagram of the complete system. Yellow blocks are the functions you will write for Simulink.

using MATLAB 2017a. When marking the software portion, the Simulink simulation will run with a computational timeout of 120 seconds. This means that after either 120 seconds of computation time or the 20 seconds of simulation time, the simulation will terminate and the results up to that point will be used.

The controller will run at a fixed rate of 20Hz ($T_s = \frac{1}{20}$s).

Submission

You will be submitting your functions on MATLAB Grader for this assignment. MATLAB Grader does not do any marking of your design, it will only test to make sure your controller can run, and you are using the correct input and output sizes. It will supply a representative input, so it can catch MATLAB programming errors, but will not actually perform any validation of your design. The tests on MATLAB Grader will utilize 2018b to test your code, so you are responsible for testing on 2017a on your own.

Hardware

The hardware testing will use the code you submitted to MATLAB Grader on the due date. You are given a 30-minute timeslot with the hardware where minor tweaks/tuning to your controller can be done. At the end of this timeslot, the controller will be tested with the default shape and its performance marked according to the rubric.

Marking

Overall Scoring

Overall, this project is worth 50 points with the breakdown given in Table 1.

Table 1: Scoring breakdown for the assignment

Software	Points	**Hardware**	Points	**Report**	Points
Default Shape	10	Obtain Equilibrium	5	Presentation	2
Other Shapes	20	Constraint Satisfaction	5	Language/Grammar	2
				Clarity/Expression	2
				Solution Description	4
Total	**30**		**10**		**10**

Software

Definition 1 (Software Equilibrium)**.** *Obtaining equilibrium in the software simulation means that:*

- *The x and y position states of the cart are within ϵ_t of the target point within 20 seconds,*
- *The x and y position states of the payload are within ϵ_t of the target point within 20 seconds,*
- *The x and y velocity states and θ and ψ angular velocity states are within ϵ_r of 0 within 20 seconds,*
- *The inputs are within ϵ_r of 0 within 20 seconds.*

All comparisons are made using the infinity norm.

Definition 2 (Software Successful Completion)**.** *Successful completion of a shape means that:*

- *The controller is at equilibrium (as defined in Definition 1).*
- *No constraints are violated*

The marking for the software component is done in 2 distinct parts:

- Successful completion of the default shape
- Successful completion of generated shapes

The default shape used to test the controller is

$$c_d = \begin{bmatrix} 0.00 & 0.05 \\ 0.25 & 0.30 \\ 0.50 & 0.05 \\ 0.45 & 0.00 \\ 0.25 & 0.20 \\ 0.05 & 0.00 \end{bmatrix}$$

with start point $(0.05, 0.05)$, target point $(0.45, 0.05)$, and tolerances $\epsilon_r = \epsilon_t = 0.02$. 10 points will be awarded for the successful completion of the default shape (with successful completion defined in Definition 2).

The remaining shapes will be generated by varying the size of the rectangles (e.g. narrowing them, etc), rotating the entire shape, moving the starting and target points, and changing the tolerances. Up to 20 points will be awarded based on the successful completion of these generated shapes and the speed at which the controller reaches equilibrium at the target point.

Hardware

Definition 3 (Hardware Equilibrium)**.** *Obtaining equilibrium in the hardware system means that:*

- *The x and y position states of the cart are within ϵ_t of the target point within 60 seconds,*
- *The x and y position states of the payload are within ϵ_t of the target point within 60 seconds,*
- *The x and y velocity states and θ and ψ angular velocity states are within ϵ_r of 0 within 60 seconds,*
- *The inputs are within ϵ_r of 0 within 60 seconds.*

All comparisons are made using the infinity norm.

All hardware marking will be done using the default shape (c_d) with the default parameters. 10 points will be awarded for the controller obtaining equilibrium (defined in Definition 3). Up to 5 points will be subtracted for entering the regions which violate the constraints. The points subtracted for being inside each violation region are shown in Figure 4. If a violation region is entered multiple times, the points will only be subtracted once.

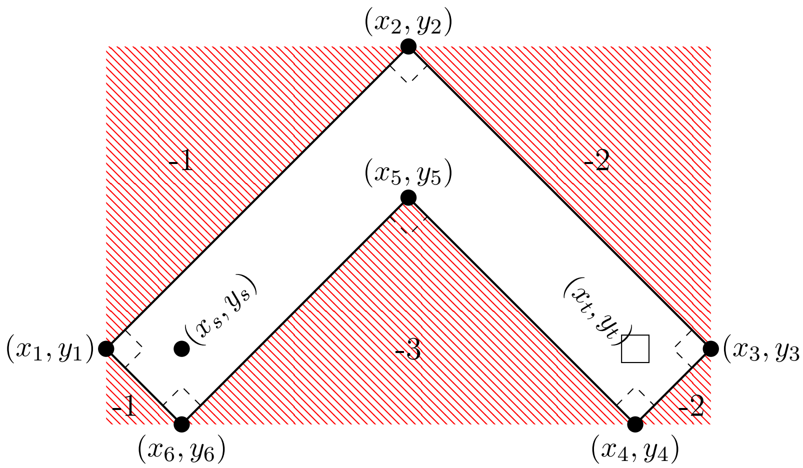

Figure 4: Penalties for the violation regions of the constraints

Report

You are required to write a short report detailing your solution to this assignment. This report must not be more than 1 page (2-column format) in length, and cannot contain any figures, tables or appendices. Any report longer than 1 page or containing those items will receive no marks. A LaTeX template will be provided on Blackboard for you to use when writing the report. You must use this template, and you are not allowed to modify it. If you utilize a technique from the literature that was not discussed in lectures, you must provide a citation to the paper it appears in.

The report should describe your solution to this project. This description can assume the reader has a basic knowledge of MPC, and the contents of the course; so it should focus on how you applied the material from the course to the project. The report must answer the following questions:

- How did you constrain the pendulum angle?
- How did you choose your cost function values?
- How did you handle the non-convexity of the constraints?
- How did you choose the target/path for the controller?

To earn additional marks, you should

- Describe how you used a method discussed in lectures but not contained in the first 3 MATLAB Grader courseworks.
- Describe how you used a method not discussed in lecture (and provide a citation to the paper).

Your report will be marked out of 10, following the rubric shown in Table 2.

Table 2: Marking rubric for the report

Element	**Weighting**
Presentation	2
Language/Grammar	2
Clarity/Expression	2
Description of solution method	4

Each of these elements will be marked as follows:

- Presentation
  Starts with 2 and one mark will be deducted for every category in which an obvious mistake in presentation is made (e.g. font size or type, spacing, no paragraphs, equation formatting, cross-referencing, bibliography formatting, inconsistency, list formatting, typos).

- Language/Grammar
  Starts with 2 and one mark will be deducted for every category in which an obvious mistake in language/grammar is made (e.g. spelling, grammar, punctuation, tense).

- Clarity/Expression
  Marks will be given based on the clarity of the text and the understandability of your report. The lower the mark the more paragraphs/sentences had to be re-read to comprehend the solution. High marks will require re-reading very few sentences, while low marks will require re-reading entire paragraphs or sections.

- Description of the Solution Method
  Marks will be awarded for providing a clear description of the implementation of your solution that demonstrates you have a deep understanding of the concepts. 2 marks will be awarded for your answers to the posed questions, while the other 2 marks will be awarded for describing your usage of ideas not covered in the prior assignments/lectures.