MCEN90028 Robotics Systems
Assignment 3 (~~8 marks~~ 10 marks)

Due date: ~~5:00 pm on Tuesday, 28 April, 2020~~
extended to Friday 1 May 2020 5pm

1 Description of Assignment

This assignment will focus on the construction of trajectories for a variable (it could be the task-space end-effector coordinates, for example) from a given initial pose and initial velocity to a given final pose and final velocity; through a set of via points. The following parameters are defined.

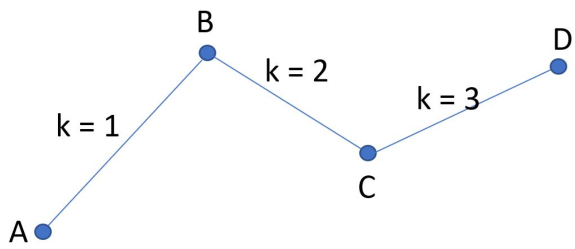

Figure 1: Example of a Trajectory with multiple segments

The trajectory to be generated can have k_{max} segments, labelled as segments $k = 1..k_{max}$. Each segment is constructed of polynomial of order P, where P is an odd number. The order of the polynomial, once selected, will be the same for all segments. In Figure 1, a trajectory will start at Point A and will end at Point D, with points B and C defined as "Via Points". A trajectory with k_{max} segments will have $k_{max} - 1$ via points.

A polynomial construction of each segment will be of the form:

$$x(t) = a_0 + a_1 t + a_2 t^2 + ... + a_P t^P. \tag{1}$$

For a cubic polynomial, $P = 3$, while for quintic polynomial, $P = 5$.

The coefficients $a_0, a_1, ...a_P$ are the coefficients of the polynomial and are the output of a trajectory generation function.

2 Assignment Tasks

Task 1 (3 marks)

In the first task, you are to construct a trajectory generation algorithm, coded in MATLAB:

$SegmentCoeff = TrajGen01(init, final, tfinal)$

where

- $SegmentCoeff$ is the output, which is a $(P+1) \times k_{max}$ matrix containing the coefficients of the polynomials for all segments of the trajectory;
- $init$ is the $((P+1)/2) \times k_{max}$ matrix containing the initial conditions for all the segments, for example, for cubic polynomials ($P = 3$), $init$ will contain the initial position, initial velocity (therefore, $((P+1)/2) = ((3+1)/2) = 2$ for all k_{max} segments), therefore $init$ will be a 2 by k_{max} matrix.
- Similarly, final is the $((P+1)/2) \times k_{max}$ matrix containing the final conditions for all the segments.
- $tfinal$ is an array of size $k_{max} \times 1$, containing the time duration of each segment of the trajectory.

In Task 1, it is assumed that all initial and final conditions are explicitly known and given - by designing a trajectory for a robot end-effector as shown in Figure 2, commencing at rest from Point A and finishing at rest at Point C. The task requires the robot to go from Point A to Point B and finally Point C. Coordinates are given in Figure 2 in meters. In this case, the TrajGen01 function needs to be run twice, once to produce the polynomial coefficients for the motion in the $x(t)$ direction and another for the $y(t)$ component of the trajectory. The other constraints necessary are explicitly provided (for Task 1) as:

- Velocity at Point B is 0.25m/s in the Y direction and 0.25m/s in the X direction.
- The time duration for segment AB is 3 seconds, for segment BC is 5 seconds.

For Task 1, in the report, provide:

- The resulting polynomials (with its resulting coefficients) for all 3 segments, in both x and y directions.
- The plot of the resulting motion:
  - $x(t)$ and $y(t)$ vs t, where time goes from $t = 0$ at Point A to $t = 8s$ at Point C.
  - $\dot{x}(t)$ and $\dot{y}(t)$ vs t, where time goes from $t = 0$ at Point A to $t = 8s$ at Point C.

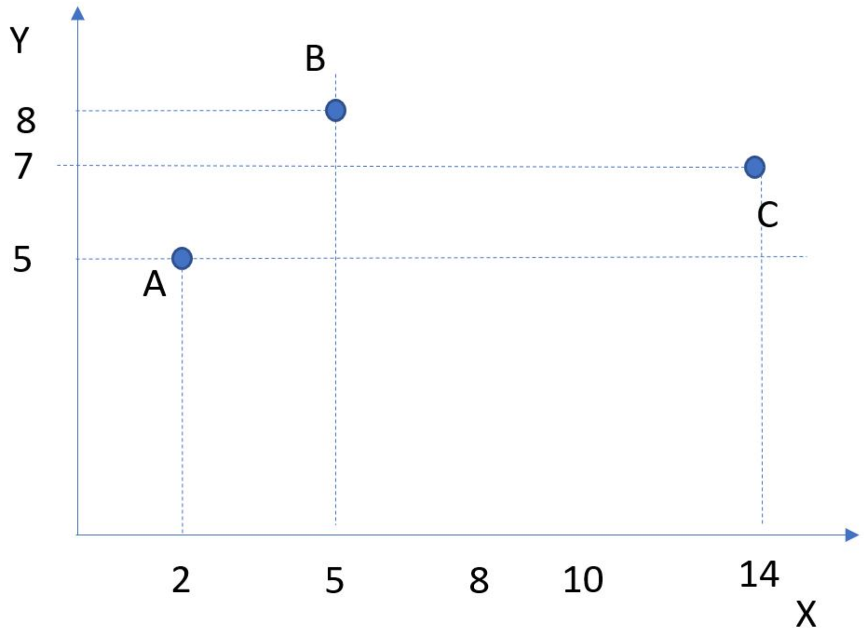

Figure 2: Locations of Point A, B, C for which the trajectory is to be designed for Task 1.

Task 2 (3 marks)

In Task 2, an obstacle (Obs1) is added at the coordinate given in Figure 3. This obstacle means that segment BC of the original trajectory designed in Task 1 is no longer viable. You are to add a new via point D and design new segments BD and DC. For Task 2, propose a method (of your choice) to detect potential collision with the given obstacle and calculate a location for via Point D, that would:

Condition 2A: ensure that straight-line segments BD and DC do not collide with Obs1.

Condition 2B: we would like to try for the total distance travelled to be minimum.

When constructing your answer, clearly point out how each of Conditions 2A and 2B are addressed. Once the location of Point D is calculated, use the MATLAB function in Task 1 to calculate the polynomial coefficients of the trajectory for segments BD and DC. You may choose the desired velocity at point D (in x and y directions) and the time durations for each segment, maintaining the total time duration from B to C at 5 seconds. Note: Keep the velocity at Point B to be the same as in Task 1.
In Task 2, construct a MATLAB file that determines if the original straight line trajectory comes into contact with the given Obs1 and calculates the location of the Via Point D, given the location of the starting and final points (B and C) and the location and radius of the obstacle Obs1. Keep this function generic so that it can work for any other 2D planning problem between two points with potentially 0 or 1 obstacle in the scenario. (in other words, if I throw in a different value for the

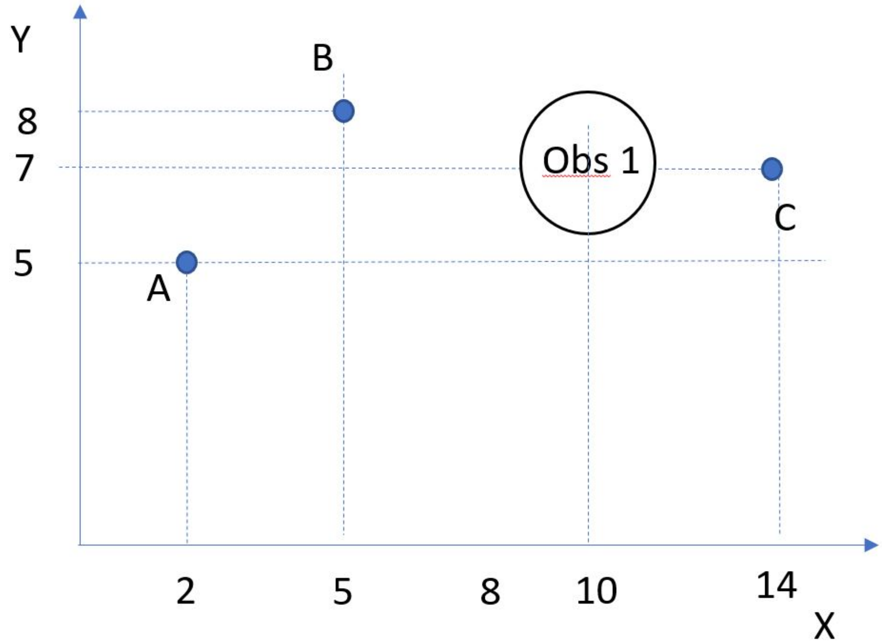

Figure 3: In Task 2, obstacle Obs1, with radius 1.5m, is added to the problem. A via point needs to be added.

locations of Point B, C and Obs1, and radius of Obs1 (assumed always circular), it should still work).
For the report, provide for Task 2:

- The explanation of the algorithm how Point D is calculated and justify your technique. Present any equations and pseudocode if needed. State the resulting location of the via point D.
  - Clearly address how Conditions 2A and 2B are addressed.
  - Plot Point D on the XY plane, showing Points A, B, C, D, the resulting path and the obstacle, in the correct proportion.
- Utilise the function from Task 1 and provide the resulting polynomials (with its resulting coefficients) for segments BD and DC, in both x and y directions.
- The plot of the resulting motion:
  - $x(t)$ and $y(t)$ vs t, where time goes from $t = 0$ at Point A to $t = 8s$ at Point C.
  - $\dot{x}(t)$ and $\dot{y}(t)$ vs t, where time goes from $t = 0$ at Point A to $t = 8s$ at Point C.

Task 3 (4 marks)

In Task 3, we repeat Task 2 but with the additional consideration of the structure of the robot. A planar 2 link robot (both revolute joints, with rotation axes at F and G aligned with the Z axis), as shown in Figure 4, is added to the problem. The trajectory being designed (from Point A to B to C) is to be traversed by the end-effector of the robot (Point T). The robot has a link length of $L_1 = 9m$ from the first to the second joint, and $L_2 = 9m$ from the second joint to the end-effector. Add the via Point D, respecting Conditions 2A and 2B, with the additional condition below:

Condition 3: the straight line formed by FG and GT must not collide the obstacle Obs1.

Condition 3 means that we simplify the "robot" into skeletal structure formed by 2 straight line, for the purpose of the assignment.

Note that in this task, robot can move freely in negative Y direction as well (X should not be treated as "floor"), and the velocity condition on point B is now lifted.

Similar to Task 2, construct a MATLAB file that calculates the location of the Via Point D, given the location of the starting and final points (B and C) and the location and radius of the obstacle Obs1 and the information of the robot kinematic.

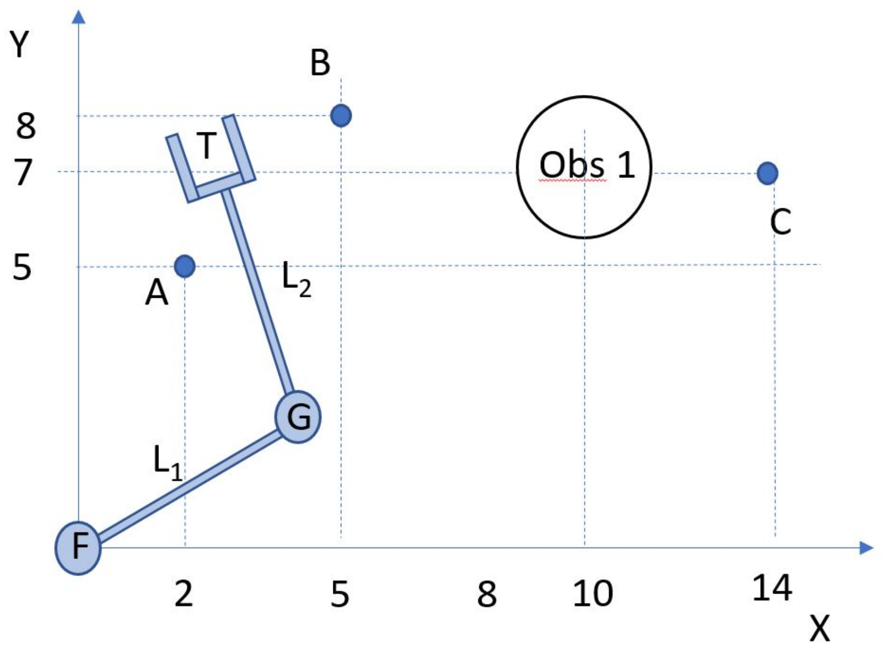

Figure 4: In Task 3, the structure of the robot is being taken into consideration in the planning.

For Task 3, provide the necessary Matlab code, as describe in Task 2, and provide in your report

- The explanation of the algorithm how Point D is calculated and justify your technique. Present any equations and pseudocode if needed. State the resulting location of the via point D.
  - Clearly explain how Conditions 2A, 2B and 3 are addressed.
  - Plot Point D on the XY plane, showing Points A, B, C, D, the resulting path and the obstacle, in the correct proportion.
- Utilise the function from Task 1 and provide the resulting polynomials (with its resulting coefficients) for segments BD and DC, in both x and y directions.
- The plot of the resulting motion:
  - $x(t)$ and $y(t)$ vs t, where time goes from $t = 0$ at Point A to $t = 8s$ at Point C.
  - $\dot{x}(t)$ and $\dot{y}(t)$ vs t, where time goes from $t = 0$ at Point A to $t = 8s$ at Point C.

Notes: In your report, detail the steps and understanding in producing your answers. Do not include the MATLAB codes in the body of the report. Include only the pseudo-code if necessary and explain the logic.

3 Submission

You need to submit one report per group plus the necessary MATLAB codes.

The report should be in an **appropriate engineering report format** and submitted as a PDF. The report should be **no more than 15 pages** (everything included) with **12pt font size**.

The title of your report should be "Assignment3_Report_AG[#]" (put your AG group number in "[#]", eg. AG07).

Compress your report and all your relevant MATLAB files as a .zip file with folder name **"Assignment3_AG[#]"** (eg. Assignment3_AG07) and submit to LMS.

If you utilised any methods taken from the literature (from a paper, textbook, etc), provide a reference to the source. Add a section at the end of the report titled "references". Learn how to cite literature source appropriately in your own time at: https://students.unimelb.edu.au/academic-skills/explore-our-resources/referencing

Note: only the last submission will be assessed.

Submission checklist:

- ☐ Report is no more than 15 pages
- ☐ Report has 12pt font size
- ☐ Report saved as PDF
- ☐ Titles of files are in the right format
- ☐ Compressed as a .zip file

4 Academic Integrity

We take academic integrity seriously. Please note that while the two assignment groups within one project group may discuss and share the robot design details, they should work on their assignments separately.

Details about academic integrity can be found on MCEN90028 Canvas page (under Subject Overview) or at `http://academicintegrity.unimelb.edu.au/`. Please check with the tutors or the lecturer if you are in doubt. Ignorance is not a valid reason for academic misconducts.