CS 7639 O01 Project 2: Air Traffic Control Simulation

Introduction

One day in 2050, you are the air traffic controller sitting at the control tower at ATL, enjoying your morning coffee. Suddenly, you receive a radio call saying that an airplane, A1, declares an emergency and needs to land first. Now, you look at the radar map in Figure 1 and realize that the aircraft is not the first in the landing queue. Your job is to land and park all three aircrafts in **minimal** time, without violating any airspace, runway and taxiway rule or regulation, with A1 being the first to land and park at P3. The optimal landing order after A1 lands is for you to determine as well.

Some Useful Coordinates Information

A1:	[0.5; 0.1; $\pi/2$].
A2:	[0; 0.3; π-0.2].
A3:	[0.1; -0.3; π-0.1].
P1:	[0.1; 0].
P2:	[0.2; 0].
P3:	[0.3; 0].

Airspace Rules and Regulations

General Rules

R1: Minimum separation of aircrafts in the air and on the runway: 0.25 m.

R2: R1 does not apply to aircrafts on the taxiway.

R3: Do not fly out of the simulation boundary.

R4: On the boundary between 2 areas, the limitations of the faster airspace apply.

R5: The center coordinates count as the position of the aircraft. Do not worry about the radius/width of the aircraft.

Class F(ast) Airspace

F1: $|w|_{max} = \pi/4$ rad/s.

F2: $v_{min} = 0.05$ m/s.

F3: $v_{max} = 0.08$ m/s.

Class S(low) Airspace

S1: $|w|_{max} = \pi/6$ rad/s.

S2: $v_{min} = 0.03$ m/s.

S3: $v_{max} = 0.04$ m/s.

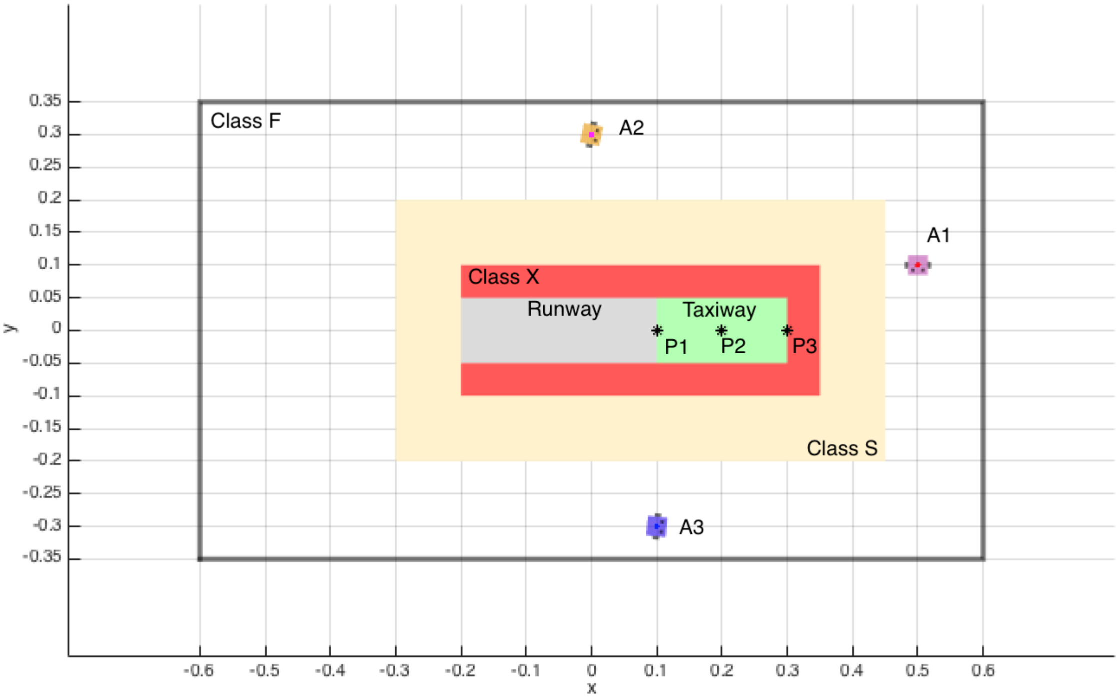

Figure 1: The radar map with coordinates shown.

Class X Airspace

X1: Entry is prohibited at any time.

Runway and Taxiway

T1: $|w|_{max} = \pi/2$ rad/s.

T2: $v_{min} = 0$ m/s.

T3: $v_{max} = 0.02$ m/s.

Procedures

P1: Download the latest version of the simulator so that everyone is on the same page: https://github.com/robotarium/robotarium-matlab-simulator

P2: Download the code template and setup files from the assignment on Canvas and place them in the simulator folder. Note that some files should be replaced by the ones from Canvas.

Goals

G1: **[80 pts]** Park all 3 aircrafts at the parking spots, with A1 at P3. Orientation does not matter.

G2: **[10 pts]** Achieve G1 by minimal number of iterations, with at most **1750** iterations.

Deliverables and Grading

Grading

1. You will receive 90 points by achieving G1 and G2.
2. (Competition) If your final iteration number is below the class average, you will receive 10 more points.
3. If you violate any rule at any time, or you do not achieve the goals, you are disqualified for the Competition.
4. 5 points are taken off every time you violate any rule in R, F, S or T.
5. 50 points are taken off every time you violate any rule in X.

Deliverables

1. The source code, a .m file.
2. The output data, a .mat file.
3. A pdf file containing a plot of the final positions and path traces of all 3 aircrafts.

Note

1. Do not upload your file to the real Robotarium since it will not work.
2. This project utilizes the simulator only. Do not worry about other physical factors like noise or saturation.
3. In each iteration, command "r.step()" only once, which has been done for you in the template code. You should not write any more "r.step()" in the for loop. All you need to do is to determine the input u, a 2 by 3 matrix, in each iteration. Failing to follow this note will result in grade penalties.
4. Note that this project is not only about constrained optimization, but also about verification and validation. Think about how you can make sure that your implementation does not violate any rule. What are some good ways to conduct verification?