Lab 5: Cal Band Pregame Formations
via Dynamic Programming
Big Game Week Special!

Due: Friday 12/21 at 2:00pm

1 Introduction

Our last lab features a Big Game Week theme. In this lab, you will design the algorithm which choreographs the Cal Band Pre Game Show formations[1]. Specifically, you will solve for the shortest path which takes each Cal Band member from an arbitrary position to their spot in the "Script Cal" formation shown above. Dynamic programming is the key ingredient for generating shows with 120 band members and 20-40 formations.

Problem Data

- The band members' positions are constrained to a grid: $(x_k^i, y_k^i) \in \{1, \cdots, n_x\} \times \{1, \cdots, n_y\}$.
- Their movement (u_k^i, v_k^i) is constrained to grid points. That is, they can move one unit north, south, east, west, or diagonally (45°) in each time step. Of course, they may also stand still. Mathematically we have $(u_k^i, v_k^i) \in \{-1, 0, 1\} \times \{-1, 0, 1\}$.
- Their movements are constrained to be within the grid. For example, in the northwest corner they can only move east, south, southeast, or stay still.
- There are M band members on the field.
- The time horizon is $k = 0, 1, \cdots, N$.
- The final state is the Script Cal formation, given in `scriptcal.mat`.
- The i^{th} Cal Band member's dynamics are given by:

[1] Cal Band uses a software called "Cal Chart" to perform this function. Believe it or not, but by the end of this assignment you will each have programmed the main functions of Cal Chart. Watching pre game will never be the same.

$$x^i_{k+1} = x^i_k + u^i_k \quad (1)$$
$$y^i_{k+1} = y^i_k + v^i_k \quad (2)$$

In compact matrix notation, we writes this as $X^i_{k+1} = X^i_k + U^i_k$, where $X^i_k = [x^i_k, y^i_k]^T$ and $U^i_k = [u^i_k, v^i_k]^T$.

- The cost-per-time-step is given by $\|U^i_k\|_2 = \sqrt{(u^i_k)^2 + (v^i_k)^2}$.

2 Formulation

Problem 1:
Formulate an optimization problem. Provide the following:

(a) Write down the objective function.
(b) Write down ALL the constraints. Label the physical meaning of each constraint.
(c) What is the decision variable? What is the state variable?

Problem 2:
(a) Define an appropriate value function, as done in lecture, for each Cal Band member i.
(b) Write down the principle of optimality equation and boundary condition.

3 Data

Download the file `scriptcal.mat` from bSpace. This file contains matrix `XN`, whose first column contains the Cal Band member index number. The 2nd and 3rd column contain the x and y coordinates of their spot in the Script Cal. Also download `Lab5_Skeleton.m` for skeleton code to solve the subsequent problems. The parameters are $M = 40$, $n_x = 25$, $n_y = 20$, $N = 24$.

Problem 3:
Plot the Script Cal. Use the marker and face color options. Include this figure and the code in your report.

4 Code

Problem 4:
Encode the Script Cal final condition in `XN` as the value function's boundary condition in the skeleton code. Provide that section of code in your report.

Problem 5:
Are the admissible control actions $U^i_k \in \{-1, 0, 1\} \times \{-1, 0, 1\}$ for all states? (Hint: think about the boundaries.) The admissible set $\mathcal{U}(X_k)$ must be defined for each X_k. Define the admissible control set `Uadmis{i,j}` in the code. Provide this code in your report.

Problem 6:
Implement the principle of optimality equations in your code. This includes (i) the cost-per-time-step calculation, (ii) principle of optimality equations, and (iii) optimal control. Provide the code in your report.

5 Animate

Run your code to solve for each band member's shortest path. To solve for all 40 members, the computation will require about 1 minute. For rapid code development and testing, you can set $m = 5$, then reset it to $m = 40$ once the code is correct. Note that DP returns a control law $u_k^{i*} = u_k^{i*}(x_k^i)$ which depends on the current state. We call this "state feedback control."

Problem 7:
Complete the skeleton code in the "Animate Solution" section. Note that plotting in a `for` loop produces animations. Use the `pause` function to control animation speed. Animate the formations for the following three initial conditions: (i) random, (ii) march on from student section side, (iii) rectangle. The rectangle initial conditions `x0rect, y0rect` are included in `scriptcal.mat`. *Comment on each of the three animations. Explain what you observe. Provide your code in the report.*

Problem 8:
Watch this video: http://www.youtube.com/watch?v=s4aTZO7wMlU

Deliverables

Submit the following on bSpace. Zip your code. Be sure that the function files are named exactly as specified (including spelling and case), and make sure the function declaration is exactly as specified.

`LASTNAME_FIRSTNAME_LAB5.PDF`
`LASTNAME_FIRSTNAME_LAB5.ZIP` which contains your respective Matlab files.