16.410 – Principles of Autonomy and Decision-Making
Problem Set. # 8

Please remember to include the time have spent for each problem in your solutions.

Problem 1: LP Formulation (20pts)

In one of the lunar bases that you are building, you are facing a problem of allocating robots for doing the science experiments autonomously. You would like to come up with a systematic procedure based on linear programming to solve this allocation problem, which is described more precisely below.

You have N robots to conduct a total of M science experiments autonomously. The percentage of experiment j that robot i can perform in unit time is given by the constant $T_{i,j}$. You are provided with all the $T_{i,j}$ for all $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$. Assume that two or more robots can work on an experiment together to make the experiment faster. In that case, the percentage of the experiment that is completed per unit time will be the sum of the contribution from each robot. Formulate Linear Programs to answer the following questions (you should formulate one linear program for each item).

- Formulate a linear program to minimize the total amount of time that the robots have spent doing the experiments, i.e., the sum (over all robots and all experiments) of the times that each robot has spent for doing all the experiments it has gotten involved in.
- Assume that you are given a cost $C_{i,j}$ for each robot and experiment pair, i.e., for all $i = 1, 2, \ldots, N$ and all $j = 1, 2, \ldots, M$. The constant $C_{i,j}$ denotes per unit time cost of robot i performing experiment j. For instance, if robot i has worked on experiment j for t time units then the incurred cost is $C_{i,j}\, t$. Formulate a linear program to minimize the total cost, i.e., the sum of all costs incurred by all robots from all experiments that they are involved in.
- Formulate a linear program that minimizes the total experiment time, i.e., first the time instance that all the robots are finished with all their experiments.

Problem 2: Solve LP problem (30pts)

Consider the following problem:

$$\begin{aligned} \max \quad & Z = x_1 + 2x_2 \\ \text{s.t.} \quad & x_1 + 3x_2 \leq 8 \\ & x_1 + x_2 \leq 4 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{aligned}$$

- Solve this problem graphically. Identify all corner-point feasible solutions.
- Solve this problem (by hand) using the simplex method.

Problem 3: Spacecraft Docking (50pts)

In this problem, we will consider the problem of steering a spacecraft to dock with another spacecraft (e.g., the International Space Station), at a given time, using minimum fuel. Some background information is provided for your reference/information, and is not necessary for solving the problem.

Consider a reference frame that is centered at the target spacecraft, which is assumed to be on a circular orbit, of period T; let $\omega = 2\pi/T$ be the angular velocity of the target spacecraft (to fix ideas, the orbital period of the ISS is about 90 minutes).

Let (x, y, z) be the relative coordinates of the spacecraft with respect to the target. The x axis is oriented in the radial direction from the center of the Earth to the target spacecraft. The y axis is oriented along the velocity vector of the target spacecraft (along-track). The z axis completes a right-handed triad (out of plane). In these coordinates, the dynamics of the spacecraft are well described by the following equations (known as Euler-Hill, or Clohessy-Wilshire equations).

$$X(t + \Delta t) = \Phi X(t) + \Phi[0, I]^T \Delta V(t), \tag{1}$$

where $X(t) = [x(t), y(t), z(t), \dot{x}(t), \dot{y}(t), \dot{z}(t)]^T$, is a vector containing position and velocity coordinates at time t, $\Delta V(t) = [\Delta v_x(t), \Delta v_y(t), \Delta v_z(t)]^T$ is a vector containing the instantaneous velocity changes at time t (we are considering impulsive maneuvers), and

$$\Phi = \begin{bmatrix} 4 - 3\cos(\omega\ \Delta t) & 0 & 0 & \sin(\omega\ \Delta t)/\omega & (2 - 2\cos(\omega\ \Delta t))/\omega & 0 \\ 6\sin(\omega\ \Delta t) - 6\omega\ \Delta t & 1 & 0 & 2(\cos(\omega\ \Delta t) - 1)/\omega & 4\sin(\omega\ \Delta t)/\omega - 3\Delta t & 0 \\ 0 & 0 & \cos(\omega\Delta t) & 0 & 0 & \sin(\omega\Delta t)/\omega \\ 3\omega\sin(\omega\ \Delta t) & 0 & 0 & \cos(\omega\ \Delta t) & 2\sin(\Delta t\omega) & 0 \\ 6\omega(\cos(\omega\ \Delta t) - 1) & 0 & 0 & -2\sin(\omega\ \Delta t) & 4\cos(\omega\ \Delta t) - 3 & 0 \\ 0 & 0 & -\omega\sin(\omega\Delta t) & 0 & 0 & \cos(\omega\Delta t) \end{bmatrix}$$

For your convenience, here is the same matrix in a more Java-friendly form:

```
4 - 3*Cos(dt*w),0,0,Sin(dt*w)/w,(2 - 2*Cos(dt*w))/w,0,
```

```
-6*dt*w + 6*Sin(dt*w),1,0,(2*(-1 + Cos(dt*w)))/w, -3*dt + (4*Sin(dt*w))/w,0,
0,0,Cos(dt*w),0,0,Sin(dt*w)/w,
3*w*Sin(dt*w),0,0,Cos(dt*w),2*Sin(dt*w),0,
6*w*(-1 + Cos(dt*w)),0,0,-2*Sin(dt*w),-3 + 4*Cos(dt*w),0,
0,0,-w*Sin(dt*w),0,0,Cos(dt*w)
```

Let us assume that the initial condition of the spacecraft is $X_0 = (1, -2, 0.5, 0, 0, 0)$. It is desired to take the spacecraft to dock with the target in one orbital period, i.e., at time $t = t_0 + T$. In order to dock, the spacecraft must come to a complete stop at the origin. Furthermore, it is desired to minimize the total amount of the "delta-Vs", i.e., the total amount of fuel needed to complete the maneuver.

For example, if you set $\Delta t = T/16$, i.e., $\omega \Delta t = \pi/8$, it is desired to minimize

$$J = \sum_{i=0}^{16} \left(|v_x(t_0 + i\Delta t)| + |v_z(t_0 + i\Delta t)| + |v_z(t_0 + i\Delta t)| \right).$$

(Notice that you should also consider the final delta-V, needed to come to a stop.)

1. Formulate the minimum-fuel docking problem as a Linear Program. In other words, choose a set of decision variables, and write the objective (or cost) and the constraints, in terms of affine functions of the decision variables.
2. Write a computer program that solves the problem, e.g., using the lpsolve library, as a function of the choice of Δt.
3. Plot the size of the problem (number of decision variables), the solution time, and the optimal cost, for at least 8 values of Δt ranging from $T/256$ to T. Plot the trajectory of the spacecraft for $\Delta t = T/256$, $\Delta t = T/16$ and, $\Delta t = T$.

You may use any programming language you wish. We have provided a code skeleton in Java (you can modify the code skeleton as you like, or choose not to use it). You may use a different solver if you wish. We have provided instructions for installing lp_solve.