AEROSP 740: Assignment 4

Note 1: Completed solutions should be uploaded into Gradescope in a single pdf file before 11:59 p.m. on the day due. **Put boxes around final answers (except for problems involving derivations/proofs, simulation figures and MATLAB codes). 1-2 Points could be deducted if not following the instruction**. Carefully explain and justify your solutions. Plots should be clearly labeled. Include print out of all the codes. Homework should be neat in appearance. You can develop your own code from scratch or follow coding hints given.

Note 2: When submitting the assignment, please follow the steps in the Gradescope and assign corresponding pages to each problem number. **5 points** deduction will be taken if not following this procedure.

Note 3: This assignment pdf together with other complementary documents are uploaded into Canvas: Files > Homeworks and Solutions > Homework 4.

Note 4: For other homework related policies, please consult the syllabus.

1. In this problem, we will code up our own quadratic programming solver by hand and use it to solve an MPC tracking problem for a mass-spring system.

   (a) (4 points) Implement the standard dual projected gradient algorithm for solving the QP,

   $$\text{Minimize } \frac{1}{2}U^{\mathrm{T}}HU + q^{\mathrm{T}}U \text{ subject to } AU \leq b.$$

   The function template should be in the form

   begin code
   ```
   function [U, lam] = myQP(H,  q, A, b, lam0);
   ```
   end code

   Here `lam0` is the guess for the vector of dual variables. As the termination criterion, use the maximum number of iterations and set it to 30. Use the vector of 1's as an initial guess for `lam0`. Test your function `myQP` and Matlab's QP solver `quadprog.m` for the following QP problem

   $$\text{Minimize } 3x_1^2 + x_2^2 + 2x_1x_2 + x_1 + 6x_2 + 2$$
   $$\text{subject to } 2x_1 + 3x_2 \geq 4, x_1 \geq 0, x_2 \geq 0.$$

   Give values of the solution (x_1, x_2) and of the dual variables (Lagrange multipliers) generated by both solvers.

   (b) (2 points) Consider now a mass-spring system the dynamics of which are described by the following differential equations,

   $$\dot{x}_1 = x_2$$
   $$\dot{x}_2 = -\frac{k}{m}x_1 + \frac{u}{m},$$

   where x_1 is the position of the mass, m is the mass, k is the spring stiffness and u is the control force. Implement an ode function that can be passed to `ode45.m` to perform continuous-time simulations. Use a template begin code

```
        function xdot = msd(t, x, u, m_msd, k_msd)
```

end code

(c) (1 point) Obtain a discrete-time model of the mass-spring system, assuming the sampling period, $T_s = 0.1$ sec and $m = 1$, $k = 1$. Give A_d and B_d matrices as your answer.

(d) (1 point) We will use a command tracking MPC formulation and impose constraints on the control input u and on the position x_1. Let the tracking error, e, be defined as $e = x_1 - r$, where r is the position command. Consider an augmented model in the incremental form,

$$\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ e \\ u \\ x_1 \end{bmatrix}_{k+1} = A \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ e \\ u \\ x_1 \end{bmatrix}_k + B\Delta u_k,$$

where $\Delta x_k = x_{k+1} - x_k$, $\Delta u_k = u_{k+1} - u_k$. Justify/derive expressions for matrices A and B. Hint: They are given in the following code

begin code

```
A=[Ad, zeros(2,3);
   1,  0,  1, 0, 0;
   0,  0,  0, 1, 0;
   1,  0,  0, 0, 1 ];

B   = [Bd; 0; 1; 0 ];
```

end code

(e) (1 point) Consider LQ-MPC formulation for the model in the incremental form. Define the weighting matrices Q and R to penalize tracking error and control increment, and build the terminal penalty matrix, P based on the solution of LQR problem for $(\Delta x,\ e)$ subsystem to recover LQR gain when constraints are inactive.

begin code

```
Q = diag([0, 0, 1, 0, 0]);
R = 1;
[K,Pdxu,E] = dlqr(A(1:3,1:3), B(1:3), diag([0,0,1]), 1);
P = blkdiag(Pdxu,zeros(2,2));
```

end code

Give your matrix P as the answer.

(f) (1 point) Suppose the constraints are given by $|x_1| \leq 0.2$ and $|u| \leq 0.2$ while other variables are unconstrained. Represent in your code the constraints to be used with the model in the augmented incremental form by defining variables `xlim` and `ulim`.

begin code

```
lN = 10;   % large number
xlim.max = [lN,  lN,  lN,    0.2,    0.2]; % [dx1, dx2, e, u, x_1]
xlim.min = -xlim.max;
ulim.max = [lN];
ulim.min = -ulim.max; % Note our "control" is "control increment",
                      % actual control is the fourth state
```

end code

Here lN stands for large number and used for defining constraints when a variable is unconstrained. You can try to set lN to Inf at the end and see if your code still runs.

(g) (3 points) Based on expressions given in class, develop a function that forms matrices you need for your QP. Use the following function template

begin code
```
[H, L, G, W, T, IMPC] = formQPMatrices(A, B, Q, R, P, xlim, ulim, N)
```
end code

where N is the prediction horizon. The QP has the form

$$\text{Minimize } \frac{1}{2}U^{\mathrm{T}}HU + q^{\mathrm{T}}U \text{ subject to } GU \leq W + Tx_0,$$

where x_0 is the intial augmented state, and

$$q = Lx_0.$$

The first control move is picked by the matrix IMPC, i.e., Δu_0=IMPC*U.

(h) (4 points) Implement the closed-loop simulations of your MPC controller that relies on function formQPMatrices to form QP matrices and myQP to solve the QP. Assume that $k = 1$ and $m = 1$. Choose the horizon $N = 15$. Initialize the vector of dual variables λ, lam0, to the vector of 1's for the first time step, while setting lam0 equal to the vector of the dual variables from the previous time step for subsequent time steps (warm start). Integrate the continuous-time dynamics of the mass spring system using ode45.m between the updates of the control. Plot time traces of the states when responding to steps in the position command, r, between -0.19 an 0.19 and one of the steps to -0.25 (outside of feasible range). Indicate the command by dashed black line and position constraints by red dashed lines on this plot. Plot also the time trace of the corresponding control input.

Assume the initial condition of the mass-spring system to be $[0, 0]^T$ and previous control is 0, and construct the initial augmented state accordingly.

Note that the step commands are $-0.19, 0.19, -0.25$. The command duration is up to you to choose, but you would like to make sure the system settles down and tracks desired feasible command before the command changes again. In this simulation, you can set each command to have a duration of 10-15 seconds.

(i) (3 points) Finally, test robustness of MPC controller. Reduce mass by 20 percent and increase stiffness by 20 percent in the simulation model (plant) but not in the model used by MPC controller for prediction. Give the same plots as in (1h). Increase the control authority a bit by changing the constraint $|u| \leq 0.2$ to $|u| \leq 0.25$ and repeat. Describe what you observe.

2. Modify the setup you developed in previous problem by handling constraints as soft and introducing a slack variable $\epsilon \in \mathbb{R}^5$ to relax them as described in the slides of Module 5. Note that the same slack variable with the dimension equal to the state dimension of the augmented model will be used to relax all constraints at once.

(a) (10 points) Modify the function formQPMatrices to account for the slack variable. The list of function arguments is extended with the slack penalty weight μ (i.e., slackPenalty)

begin code
```
function [H, L, G, W, T, IMPC] = ...
   formQPMatrices(A, B, Q, R, P, xlim, ulim, N, slackPenalty)
```
end code

Include the print out of the modified function code.

(b) (10 points) For the case with no plant-model mismatch, generate the same time traces as in the previous problem for two values of the slack penalty weight - small ($\mu = 1e-3$) and large ($\mu = 1e3$). Describe qualitatively the differences in the trajectories that you are observing.