Model Predictive Control

Model Predictive Control, Part 1

0.0/10.0 points (graded)

Given a discrete linear system, corresponding to a discretization of a second order system,

$$x[k+1] = \begin{bmatrix} 1 & 0 & .1 & 0 \\ 0 & 1 & 0 & .1 \\ 2 & -2 & 1.2 & 0 \\ 0 & 1.5 & -1 & .7 \end{bmatrix} x[k] + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & -2 \\ 0 & 1 \end{bmatrix} u[k]$$

Write a quadratic program (QP) to do model predictive control. From an initial state $x[0] = x_0$, find a sequence of inputs that minimizes the cost function: *[Math Processing Error]*

$$\sum_{k=1}^{N} ||u[k-1]||^2 + .01||x[k]||^2$$

such that the state is driven to the origin, $x[N] = 0^{4\times1}$. Please use the QP format given-- note that your solution will be graded on correctness of the control sequence. Ensure that the matrix "u" in your solution is $2 \times N$, where $u[k]$=u(:,k+1). You may implement this as a shooting or transcription type approach.

To avoid confusion, write the QP in the same format as MATLAB's quadprog:

$$\begin{aligned} \min .5 z^T H z + f^T z \\ \text{s.t.} Az \leq b \\ Bz = c \end{aligned}$$

```
% PROBLEM SETUP, DO NOT CHANGE
A_sys = [1 0 .1 0;0 1 0 .1; 2 -2 1.2 0; 0 1.5 -1 .7];
B_sys = [0 0;0 0;1 -2;0 1];
N = 10;
x0 = 10*randn(4,1);
```

```

        % QP SETUP HERE
        H =
        f =
        A =
        b =
        B =
        c =

        % SOLVE QP
```

Unanswered

```
        % Take the transcription based approach
                % order z = [u0;x1;u1;x2;...;uN-1;xN]
                num_vars = 6*N;
                H = zeros(num_vars);
                f = zeros(num_vars,1);
                A = zeros(0,num_vars);
                b = [];
                B = zeros(0,num_vars);
                c = [];

                % dynamic constraints
                for k=1:N,
                  H((1:2)+6*(k-1),(1:2)+6*(k-1)) = eye(2);
                  H((3:6)+6*(k-1),(3:6)+6*(k-1)) = .01*eye(4);
                  B_dyn = zeros(4,num_vars);
                  if k==1,
                        B_dyn(:,1:6) = [B_sys -eye(4)];
                        c = [c;-A_sys*x0];
                  else
                    start_ind = 6*(k-1) - 4;
                        B_dyn(:,start_ind + (1:10)) = [A_sys B_sys -eye(4)];
                        c = [c;zeros(4,1)];
              end
                  B = [B;B_dyn];
                end

                % goal constraint
                B = [B;zeros(4,num_vars-4) eye(4)];
                c = [c;zeros(4,1)];

    options = optimset('Display','Off');
                [z,fval,exitflag] = quadprog(H,f,A,b,B,c,[],[],[],options);

                % extract solution
                z_reshape = reshape(z,6,N);
                x_sol = z_reshape(3:end,:);
                u = z_reshape(1:2,:);
```

Run Code

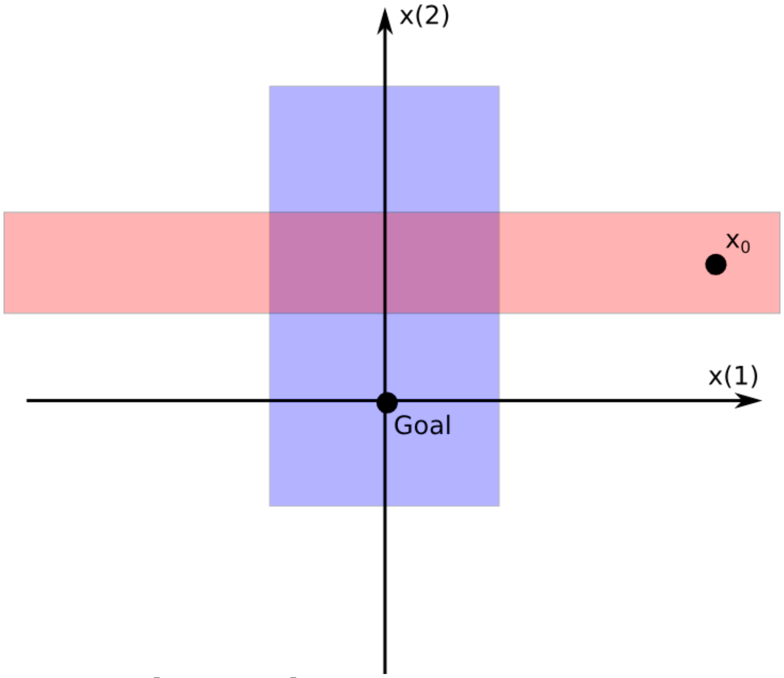
Submit You have used 0 of 3 attempts

Answers are displayed within the problem

Model Predictive Control, Part 2

0.0/10.0 points (graded)

Let's make the previous problem a a little more interesting. Suppose now that we have to stay inside a pair of intersecting corridors. In particular, for all $x[k]$, we must have $x(2) \in [2, 4]$ **OR** $x(1) \in [-2, 2]$, referring to the second and first element of the vector x respectively. Note, we are only enforcing this constraint at the discrete points, not between them!



For $x_0 = [5; 3; 0; 0]$, modify the previous program so that the first condition above holds for $k = 1, \ldots, m$ and the second condition holds for $k = m, \ldots, N$. Note the overlap on index m! Run your program for all possible values of m, and select the optimal u, as in Part 1, and the optimal point m^*.

Note that the program may not always be feasible! Be sure to check the solutions for feasibility.

```
        % PROBLEM SETUP, DO NOT CHANGE
        A_sys = [1 0 .1 0;0 1 0 .1; 2 -2 1.2 0; 0 1.5 -1 .7];
            B_sys = [0 0;0 0;1 -2;0 1];
            N = 10;
            x0 = [5;3;0;0];

        m_star =
        u =
```

```
9
```

Unanswered

```
% setup from part 1
num_vars = 6*N;
H = zeros(num_vars);
f = zeros(num_vars,1);
A = zeros(0,num_vars);
b = [];
B = zeros(0,num_vars);
c = [];

% dynamic constraints
for k=1:N,
  H((1:2)+6*(k-1),(1:2)+6*(k-1)) = eye(2);
  H((3:6)+6*(k-1),(3:6)+6*(k-1)) = .01*eye(4);
  B_dyn = zeros(4,num_vars);
  if k==1,
        B_dyn(:,1:6) = [B_sys -eye(4)];
        c = [c;-A_sys*x0];
  else
    start_ind = 6*(k-1) - 4;
        B_dyn(:,start_ind + (1:10)) = [A_sys B_sys -eye(4)];
        c = [c;zeros(4,1)];
end
  B = [B;B_dyn];
end

% goal constraint
B = [B;zeros(4,num_vars-4) eye(4)];
c = [c;zeros(4,1)];

opt_cost = inf;
opt_m = [];
opt_u = [];
for m=1:N,
        A = zeros(0,num_vars);
        b = [];
        for k=1:m,
                A_row = zeros(1,num_vars);
                A_row(6*(k-1)+4) = 1;
                A = [A;A_row;-A_row];
                b = [b;4;-2];
        end
        for k=m:N,
                A_row = zeros(1,num_vars);
                A_row(6*(k-1)+3) = 1;
                A = [A;A_row;-A_row];
                b = [b;2;2];
        end
```

```
    options = optimset('Display','Off');
                    [z,fval,exitflag] = quadprog(H,f,A,b,B,c,[],[],[],options);

                    if exitflag == 1,
                        if lt(fval,opt_cost)
                            opt_cost = fval;
                            opt_m = m;
                            % extract solution
                            z_reshape = reshape(z,6,N);
                            x_sol = z_reshape(3:end,:);
                        u_sol = z_reshape(1:2,:);
                        end
                    end
                end

    u = u_sol;
    m_star = opt_m;
```

Run Code

Submit You have used 0 of 3 attempts

Answers are displayed within the problem

Model Predictive Control, Part 3

0.0/7.0 points (graded)

For the QP from **Part 1**, which of the following statements are true?

- [ ] The program is guaranteed to return a control sequence $\boldsymbol{u}[k]$. ✔
- [ ] The program has a unique optimal control sequence $\boldsymbol{u}[k]$. ✔
- [ ] The first control element, $\boldsymbol{u}[0]$ is *continuous* with respect to the initial state $\boldsymbol{x}[0]$. ✔

Submit You have used 0 of 1 attempt

Answers are displayed within the problem