第三次实验课作业

2151094 宋正非

问题背景：

假设天问一下降速过程中的部分所需控制模型如下：

$$x_1(k+1) = 2x_1(k) + 2x_2(k)$$
$$x_2(k+1) =- 2x_1(k) - x_2(k) + x_3(k)$$
$$x_3(k+1) = 2x_2(k) + 1.5x_3(k) + u(k)$$

状态及输入的约束条件：$-20 \le x_1 \le 20$，$-10 \le x_2 \le 10$，$-20 \le x_3 \le 20$，$-10 \le u \le 10$

预测步数：10 步

初值：$x_0 = [2;1;1]$

代价函数系数矩阵为：$\mathrm{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$，$\mathrm{R} = 1$

问题：

结合课堂内容所学，请将其整理成**状态空间**的形式并采用**模型预测控制**方法求解最优解。请使用 Matlab 编程实现，其中 QP 求解器用**加速梯度投影法**。

解决：

根据问题描述，可以首先将该问题整理为状态空间的形式。状态空间模型可以表示为：

$$x(k+1) = Ax(k) + Bu(k)$$

其中，

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}, \quad A = \begin{bmatrix} 2 & 2 & 0 \\ -2 & -1 & 1 \\ 0 & 2 & 1.5 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

接下来，我们可以利用模型预测控制（MPC）方法求解最优解。首先，构造预测矩阵S 和预测输入矩阵H：

$$S = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{10} \end{bmatrix}, \quad H = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^9B & A^8B & \cdots & B \end{bmatrix}$$

然后，构造代价函数$J = \Delta U^T R \Delta U + (x_{ref} - Sx(0) - H\Delta U)^T Q(x_{ref} - Sx(0) - H\Delta U)$，其中$\Delta U$表示输入变化量，$x_{ref}$表示参考轨迹，$Q$和$R$表示代价系数矩阵。

代码部分：

```
%main.m
clear; clc; close all;
% 线性系统系数矩阵
A=[2 2 0; -2 -1 1;0 2 1.5]; B=[0; 0;1];
% 初始状态量-如果不能在下一步回到约束范围内，则会造成无解
x0=[2;1;1];
% 预测步长
Np=10;
% 优化目标参数，加权矩阵
Q=eye(3); R=1;
% 转化为用控制量 ut 表示的，关于状态量的推导方程的矩阵
At=[]; Bt=[]; temp=[];
% 转换后的加权矩阵
Qt=[]; Rt=[];
% 加权矩阵的计算过程，以及推导方程矩阵的叠加过程
for i=1:Np
At=[At; A^i];
Bt=[Bt zeros(size(Bt,1), size(B,2));
A^(i-1)*B temp];
temp=[A^(i-1)*B temp];
Qt=[Qt zeros(size(Qt,1),size(Q,1));
zeros(size(Q,1),size(Qt,1)) Q];
Rt=[Rt zeros(size(Rt,1),size(R,1));
zeros(size(R,1),size(Rt,1)) R];
end
% 转换后的优化目标函数矩阵，循环优化函数中 H 后的表达式为优化目标的另一项
H=2*(Bt'*Qt*Bt + Rt);
% 转换后的优化中的不等式约束左边系数矩阵，后面循环中的 bi 为不等式右边
Ai=[Bt;-Bt;eye(10);-eye(10)];
% 声明 u 来保存每一步采用的控制量
d=[20;10;20; 20;10;20;
20;10;20 ;20;10;20 ;20;10;20 ;20;10;20 ;20;10;20 ;20;10;20 ;20;10;20 ;20;10;
20];
u=[];
x=x0;
xk=x0;
[a,b]=eig(Ai*inv(H)*Ai');
lamada = max(diag(b))+1;
h=1;
S=zeros(80,1);
for k=1:50
% 一切准备就绪，进行二次优化
bi=[d-At*xk; d+At*xk;10*ones(10,1);10*ones(10,1)];
[ut]=admmfast(H,(2*xk'*At'*Qt*Bt)',h,Ai,bi,S,lamada);
% 采用优化得到的控制量的第一个元素作为实际作用的控制量，代入到原系统中得到下一个时
刻的状态量
u(k) = ut(1);
x(:, k+1) = A*x(:, k) + B*u(k);
xk = x(:, k+1);
end


figure();
plot(x');
legend('x_1','x_2','x_2');
```

```
figure();
plot(u);
legend('u');
```

```
%admmfast.m
function [z]=admmfast(Q,f,x,G,W,S,L)

k=1;
maxiter = 100;
iMG = inv(Q)*G';
iMc = inv(Q)*f*x;
GL=1/L*G;
bL=1/L*(W+S*x);

y0 = zeros(80,1);
y = zeros(80,1);
while k<maxiter
beta=max((k-1)/(k+2),0);
w=y+beta*(y-y0);
z=-(iMG*w+iMc);
s=GL*z-bL;
y0=y;
y=w+s;
k=k+1;
end
z= -inv(Q)*(f*x + G'*y);
end
```

得到结果：