1. (30’) Consider the LTI system



where  is the state,  is the input, and  is the output. Suppose that  is observable. Show that there exists a suitable matrix , such that all eigenvalues of  can be arbitrarily assigned.

1. (20’) Please show that whether the following system can be stabilized by a state feedback control . If it does, find a suitable .



1. (20’) Consider  and . For any matrix , show that  is controllable if and only if  is controllable.
2. (30’) Consider the LTI system



where  is the state,  is the input, and  is the output. Suppose is controllable and observable. Let one of its state observers be in the following form



Show that the observer is a special case of the following observer



i.e., , , and all eigenvalues of  have negative real parts.

1. (Practice by coding) Given a discrete time state space model with

Suppose your design objective is represented by a state weighting matrix and control weighting matrix

1. Compute value iterations (, k = 1, 2, . . .) using the Riccati recursion formula starting with. Find the converged matrix and the optimal control gain matrix . (You cannot directly use “dlqr” function in Matlab);
2. For initial state , compare the performance of the optimal gain matrix with respect to two stabilizing feedback gains designed using the eigenvalue assignment approach (you can pick desired eigenvalue sets of your choice, and you are allowed to use “place” function in Matlab). You need to plot trajectories of and (no need for ) for the comparison. You also need to compare the actual cost over 1000 steps. (hint: simulate the closed-loop system under different feedback gains for 1000 steps, and compare their costs ).
3. Compute the 10-horizon optimal cost for initial state (namely, . Find the 10-horizon optimal control sequence , k = 0, 1, . . . , 10.

（a）部分参考代码

% Define the system matrices

A = [1 2 3; 2 1 0; 1 1 3];

B = [1; 0; 1];

Q = diag([1 10 3]);

R = 2;

% Initialize P with Q

P = Q;

% Set a tolerance for convergence and a maximum number of iterations

tolerance = 1e-6;

max\_iter = 1000;

iter = 0;

% Perform Riccati recursion

while iter < max\_iter

iter = iter + 1;

P\_prev = P;

P = A' \* P \* A - A' \* P \* B \* inv(R + B' \* P \* B) \* B' \* P \* A + Q;

% Check for convergence

if norm(P - P\_prev, 'fro') < tolerance

break;

end

end

% Display the converged P matrix

disp('Converged P matrix:');

disp(P);

% Compute the optimal control gain matrix K\*

K = inv(R + B' \* P \* B) \* B' \* P \* A;

% Display the optimal control gain matrix

disp('Optimal control gain matrix K\*:');

disp(K);

The output:

Converged P matrix:

108.3257 65.4145 12.0360

65.4145 65.5894 13.6335

12.0360 13.6335 20.7680

Optimal control gain matrix K\*:

2.0060 2.2723 2.9613