Problem 3-B

# **CODE:**

% PROBLEM 3-B

% Clear workspace

close all

clear

clc

% Add parser and solver to path

addpath(genpath('C:\Users\tsamak\Downloads\MathWorks\Toolboxes\archives\required\YALMIP'))

addpath(genpath('C:\Users\tsamak\Downloads\MathWorks\Toolboxes\archives\required\SeDuMi'))

% Define the system matrices A, B, and C

A = [0, 1; -4, -5];

K = [0; 1];

M = [-1, -2];

H = 0;

% Define the LMI variables

P = sdpvar(2, 2);

gamma = sdpvar(1, 1);

% Define the LMI constraints

LMI1 = [P\*A+A'\*P, P\*K, M'; K'\*P, -gamma\*eye(1), H'; M, H, -gamma\*eye(1)] <= 0;

LMI2 = P >= 0;

% Set up the objective

Objective = gamma;

% Define the solver settings (use an LMI solver like YALMIP with a solver of your choice)

options = sdpsettings('verbose', 1, 'solver', 'sedumi');

% Solve the LMI problem

solution = optimize([LMI1, LMI2], Objective, options);

if solution.problem == 0

% Extract the optimal solution

optimal\_gamma = 1/value(gamma);

% Display the result

fprintf('Optimal gamma: %.4f\n', optimal\_gamma);

fprintf('Stability guaranteed: %.4f < delta(t) < %.4f\n', -optimal\_gamma, optimal\_gamma);

else

fprintf('LMI problem could not be solved.\n');

end

# **OUTPUT:**

Optimal gamma: 2.4173

Stability guaranteed: -2.4173 < delta(t) < 2.4173

# **SCREENSHOT:**

