clc; clear; close all;  
  
% Add paths  
addpath('Position Controller')  
addpath('Attitude Controller')  
addpath('functions')  
  
% Load workspaces  
load('infinite\_Position\_SNAC\_workspace2.mat','Position\_W','Position\_R','Position\_F','Position\_G','dt','grav')  
load('infinite\_Attitude\_SNAC\_workspace2.mat','Attitude\_W','Attitude\_R','Attitude\_F','Attitude\_G')  
  
% Load necessary variables for the position and attitude  
Position.Position\_W = Position\_W; % NN weights  
Position.Position\_G = Position\_G; % Position control dynamics  
Position.Position\_R = Position\_R; % Control penalizing matrix  
  
Attitude.Attitude\_W = Attitude\_W; % NN weights  
Attitude.Attitude\_G = Attitude\_G; % Attitude control dynamics  
Attitude.Attitude\_R = Attitude\_R; % Control penalizing matrix  
  
% Define simulation parameters  
parameters.dt = 0.001; % time step  
parameters.t\_f = 50; % final time  
parameters.grav = 9.81; % gravity (m/s^2)  
parameters.m = 1; % mass (kg)  
parameters.Ix = 0.3; % moments of inertia (kg\*m^2)  
parameters.Iy = 0.4; %  
parameters.Iz = 0.5; %  
  
% Define desired reference as function of time  
reference = @(t)...  
 [(1-exp(-0.01\*t))\*9.81\*cos(0.2\*t); % reference\_x  
 (1-exp(-0.01\*t))\*9.81\*sin(0.2\*t); % reference\_y  
 -1\*t]; % reference\_z  
  
% References can be waypoints (not function of time)  
% reference = @(t)...  
% [5; % reference\_x  
% 2.5; % reference\_y  
% -10]; % reference\_z  
  
% Define initial condition, each column is a new set of ICs  
IC = [0 5 -5 5 -5; % x  
 0 5 -5 -5 5; % y  
 0 0 0 0 0; % z  
 zeros(9,5)]; % velocity, angles, angular velocities  
  
% One set of IC  
% IC = [0;  
% 0;  
% 0;  
% zeros(9,1)];  
  
noise = 6; % 600% noise  
  
% Simulating for all IC, simulations saved in structures  
for i = 1:size(IC,2)  
 results = simulate(Position, Attitude, parameters, reference, IC(:,i),noise); %max 6  
 simulations.(['results\_', num2str(i)]) = results;  
 x.(['x\_',[num2str(i)]]) = results.x;  
 u.(['u\_',[num2str(i)]]) = results.u;  
 r\_smooth.(['r\_smooth\_',[num2str(i)]]) = results.r\_smooth;  
 r\_initial.(['r\_initial\_',[num2str(i)]]) = results.r\_initial;  
 angles\_ref.(['angles\_ref\_',[num2str(i)]]) = results.angles\_ref;  
 time = results.time;  
end

## Plotting

figure  
plot3(r\_initial.r\_initial\_1(1,:), r\_initial.r\_initial\_1(2,:), -r\_initial.r\_initial\_1(3,:), '--', 'Linewidth', 1.5)  
grid on  
hold on  
for i = 1:size(IC,2)  
plot3(x.(['x\_',[num2str(i)]])(1,:), x.(['x\_',[num2str(i)]])(2,:), -x.(['x\_',[num2str(i)]])(3,:), 'Linewidth', 1.5)  
end  
title('3D Trajectory')  
xlabel('x (m)'), ylabel('y (m)'), zlabel('z (m)')  
legend('Reference Trajectory', 'Path 1','Path 2','Path 3','Path 4','Path 5', 'Location', 'northeast');  
  
% Position Plotting %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
figure  
subplot(3,1,1)  
hold on  
grid on  
plot(time, r\_initial.r\_initial\_1(1,1:length(time)),'--', 'Linewidth', 1.5)  
for i = 1:size(IC,2)  
plot(time, x.(['x\_',[num2str(i)]])(1,1:length(time)), 'Linewidth', 1.5)  
end  
title('Position Tracking')  
ylabel('x (m)'), xlabel('time (s)')  
legend('Reference trajectory','Path 1','Path 2','Path 3','Location', 'northeast');  
  
subplot(3,1,2)  
hold on  
grid on  
plot(time, r\_initial.r\_initial\_1(2,1:length(time)),'--', 'Linewidth', 1.5)  
for i = 1:size(IC,2)  
plot(time, x.(['x\_',[num2str(i)]])(2,1:length(time)), 'Linewidth', 1.5)  
end  
ylabel('y (m)'), xlabel('time (s)')  
  
subplot(3,1,3)  
hold on  
grid on  
plot(time, -r\_initial.r\_initial\_1(3,1:length(time)),'--', 'Linewidth', 1.5)  
for i = 1:size(IC,2)  
plot(time, -x.(['x\_',[num2str(i)]])(3,1:length(time)), 'Linewidth', 1.5)  
end  
ylabel('z (m)'), xlabel('time (s)')  
  
% Control Plotting %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
figure  
subplot(4,1,1)  
hold on  
grid on  
plot(time(1:length(u.u\_1(1,:))), u.u\_1(1,:), 'Linewidth', 1.5)  
title('Quadcopter Controls')  
ylabel('$f\_t$ (N)','Interpreter','latex'), xlabel('time (s)')  
% ylim([9.7 10.1])  
  
subplot(4,1,2)  
hold on  
grid on  
plot(time(1:length(u.u\_1(1,:))), u.u\_1(2,:), 'Linewidth', 1.5)  
ylabel('$\tau\_x$ (Nm)','Interpreter','latex'), xlabel('time (s)')  
% ylim([-4E-4 4E-4])  
  
subplot(4,1,3)  
hold on  
grid on  
plot(time(1:length(u.u\_1(1,:))), u.u\_1(3,:), 'Linewidth', 1.5)  
ylabel('$\tau\_y$ (Nm)','Interpreter','latex'), xlabel('time (s)')  
% ylim([-4E-4 8E-4])  
  
subplot(4,1,4)  
hold on  
grid on  
plot(time(1:length(u.u\_1(1,:))), u.u\_1(4,:), 'Linewidth', 1.5)  
ylabel('$\tau\_z$ (Nm)','Interpreter','latex'), xlabel('time (s)')  
% ylim([-1.5E-5 1.5E-5])  
  
% Velocity Plotting %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
figure  
subplot(3,1,1)  
hold on  
grid on  
plot(time, r\_smooth.r\_smooth\_1(4,1:length(time)),'--', 'Linewidth', 1.5)  
plot(time, x.x\_1(4,1:length(time)), 'Linewidth', 1.5)  
title('Velocity Tracking')  
ylabel('u (m/s)'), xlabel('time (s)')  
legend('Reference trajectory', 'Simulated','Location', 'northeast');  
  
subplot(3,1,2)  
hold on  
grid on  
plot(time, r\_smooth.r\_smooth\_1(5,1:length(time)),'--', 'Linewidth', 1.5)  
plot(time, x.x\_1(5,1:length(time)), 'Linewidth', 1.5)  
ylabel('v (m/s)'), xlabel('time (s)')  
  
subplot(3,1,3)  
hold on  
grid on  
plot(time, -r\_smooth.r\_smooth\_1(6,1:length(time)),'--', 'Linewidth', 1.5)  
plot(time, -x.x\_1(6,1:length(time)), 'Linewidth', 1.5)  
ylabel('w (m/s)'), xlabel('time (s)')  
  
% Angle Plotting %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
figure  
subplot(3,1,1)  
hold on  
grid on  
plot(time(1:length(u.u\_1(1,:))), angles\_ref.angles\_ref\_1(1,:),'--', 'Linewidth', 1.5)  
plot(time, x.x\_1(7,1:length(time)), 'Linewidth', 1.5)  
title('Angle Tracking')  
ylabel('$\phi$ (rad)','Interpreter','latex'), xlabel('time (s)')  
legend('Reference trajectory', 'Simulated','Location', 'northeast');  
  
subplot(3,1,2)  
hold on  
grid on  
plot(time(1:length(u.u\_1(1,:))), angles\_ref.angles\_ref\_1(2,:),'--', 'Linewidth', 1.5)  
plot(time, x.x\_1(8,1:length(time)), 'Linewidth', 1.5)  
ylabel('$\theta$ (rad)','Interpreter','latex'), xlabel('time (s)')  
  
subplot(3,1,3)  
hold on  
grid on  
plot(time(1:length(u.u\_1(1,:))), angles\_ref.angles\_ref\_1(3,:),'--', 'Linewidth', 1.5)  
plot(time, x.x\_1(9,1:length(time)), 'Linewidth', 1.5)  
ylabel('$\psi$ (rad)','Interpreter','latex'), xlabel('time (s)')  
% ylim([-0.04 0.04])  
  
% Angular Velocity Plotting %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
figure  
subplot(3,1,1)  
hold on  
grid on  
plot(time(1:length(u.u\_1(1,:))), angles\_ref.angles\_ref\_1(4,:),'--', 'Linewidth', 1.5)  
plot(time, x.x\_1(10,1:length(time)), 'Linewidth', 1.5)  
title('Anglular Velocity Tracking')  
ylabel('p (rad/s)','Interpreter','latex'), xlabel('time (s)')  
legend('Reference trajectory', 'Simulated','Location', 'northeast');  
% ylim([-5E-3 5E-3])  
  
subplot(3,1,2)  
hold on  
grid on  
plot(time(1:length(u.u\_1(1,:))), angles\_ref.angles\_ref\_1(5,:),'--', 'Linewidth', 1.5)  
plot(time, x.x\_1(11,1:length(time)), 'Linewidth', 1.5)  
ylabel('q (rad/s)','Interpreter','latex'), xlabel('time (s)')  
% ylim([-5E-3 5E-3])  
  
subplot(3,1,3)  
hold on  
grid on  
plot(time(1:length(u.u\_1(1,:))), angles\_ref.angles\_ref\_1(6,:),'--', 'Linewidth', 1.5)  
plot(time, x.x\_1(12,1:length(time)), 'Linewidth', 1.5)  
ylabel('r (rad/s)','Interpreter','latex'), xlabel('time (s)')  
% ylim([-5E-3 5E-3])  
  
% Angular Velocity and Controls %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
figure  
sgtitle('Angular Velocity Stabilization')  
subplot(3,2,1)  
hold on  
grid on  
plot(time(1:length(u.u\_1(1,:))), angles\_ref.angles\_ref\_1(4,:),'--', 'Linewidth', 1.5)  
plot(time, x.x\_1(10,1:length(time)), 'Linewidth', 1.5)  
title('Anglular Velocities')  
ylabel('p (rad/s)','Interpreter','latex'), xlabel('time (s)')  
% ylim([-5E-2 5E-2]);  
xlim([0 1])  
  
subplot(3,2,3)  
hold on  
grid on  
plot(time(1:length(u.u\_1(1,:))), angles\_ref.angles\_ref\_1(5,:),'--', 'Linewidth', 1.5)  
plot(time, x.x\_1(11,1:length(time)), 'Linewidth', 1.5)  
ylabel('q (rad/s)','Interpreter','latex'), xlabel('time (s)')  
% ylim([-8E-2 8E-2]);  
xlim([0 1])  
  
subplot(3,2,5)  
hold on  
grid on  
plot(time(1:length(u.u\_1(1,:))), angles\_ref.angles\_ref\_1(6,:),'--', 'Linewidth', 1.5)  
plot(time, x.x\_1(12,1:length(time)), 'Linewidth', 1.5)  
ylabel('r (rad/s)','Interpreter','latex'), xlabel('time (s)')  
legend('Reference trajectory', 'Simulated','Location', 'northeast');  
xlim([0 1.5])  
  
subplot(3,2,2)  
hold on  
grid on  
title('Torque Controls')  
plot(time(1:length(u.u\_1(1,:))), u.u\_1(2,:), 'Linewidth', 1.5)  
ylabel('$\tau\_x$ (Nm)','Interpreter','latex'), xlabel('time (s)')  
% ylim([-0.2 0.02]);  
xlim([0 1])  
  
subplot(3,2,4)  
hold on  
grid on  
plot(time(1:length(u.u\_1(1,:))), u.u\_1(3,:), 'Linewidth', 1.5)  
ylabel('$\tau\_y$ (Nm)','Interpreter','latex'), xlabel('time (s)')  
% ylim([-0.35 6E-2]);  
xlim([0 1])  
  
subplot(3,2,6)  
hold on  
grid on  
plot(time(1:length(u.u\_1(1,:))), u.u\_1(4,:), 'Linewidth', 1.5)  
ylabel('$\tau\_z$ (Nm)','Interpreter','latex'), xlabel('time (s)')  
% ylim([-0.04 0.04]);  
xlim([0 1])

[*Published with MATLAB® R2022a*](https://www.mathworks.com/products/matlab)