

```

%Initial State = [x_pos, x_vel, y_pos, y_vel]
initial_state = [0 0 0 10];
time = 60;
time_step = 1;

%Problem 1
F = [1 time_step 0 0 ; 0 1 0 0 ; 0 0 1 time_step ; 0 0 0 1];
system_state = initial_state;
updated_state = initial_state;
for i=1:time-1
    updated_state = F * transpose(updated_state);
    updated_state = transpose(updated_state);
    system_state = [system_state; updated_state];
end
tiledlayout(2,1)
ax1 = nexttile;

plot(ax1, system_state(:,1),system_state(:,3))
hold on

%Problem 2
for j=1:4
    F = [1 time_step 0 0 ; 0 1 0 0 ; 0 0 1 time_step ; 0 0 0 1];
    q = 0.1;
    system_state = initial_state;

    updated_state = initial_state;
    for i=1:time
        noise = mvnrnd([0 0 0 0],[ ...
            (time_step^3)/3 (time_step^2)/2 0 0; ...
            (time_step^2)/2 time_step 0 0; ...
            0 0 (time_step^3)/3 (time_step^2)/2; ...
            0 0 (time_step^2)/2 time_step] ...
            *q);

        updated_state = F * transpose(updated_state)+transpose(noise);
        updated_state = transpose(updated_state);
        system_state = [system_state; updated_state];

    end

    plot(system_state(:,1),system_state(:,3));
    legend
    title(ax1,'Particle Trajectory');
    xlabel(ax1,'X Position');
    ylabel(ax1,'Y Position');
    hold on
end
hold off
measurement = [];
for i=1:time
    noise = mvnrnd([0 0 0 0],[25 0 0 0; 0 25 0 0; 0 0 25 0; 0 0 0 25]);
    measured_state = system_state(i,:) + noise;
    measurement = [measurement ; measured_state];
end
ax2 = nexttile;

plot(ax2, system_state(:,1),system_state(:,3));
hold on
plot(ax2, measurement(:,1), measurement(:,3), '.');
title(ax2,'Trajectory and Measurement Data for [Data5]')

```

```
xlabel(ax2,'X Position');  
ylabel(ax2,'Y Position');  
linkaxes([ax1,ax2],'x');
```

