```
% % Particle Filter
N = 1000;

Xt1 = cell(1,N); % input N Particles
Xtn = cell(5,1); % blank array to store particle poses at t=0 to t=20
for i = 1:N
    Xt1{i} = [0;0;0]; % assigning initial pose at t=0
end

Xtn{1} = Xt1;

t1=0; %first time step

  phi_l = 1.5; % left wheel commanded angular velocity
  phi_r = 2; % right wheel commanded angular velocity

  r = 0.25; %wheel radius
  w = 0.5; %wheel track width

  sig_l = 0.05; %left wheel speed uncertainty
  sig_r = 0.05; %rght wheel speed uncertainty
  sig_p = 0.10; %measurement uncertainty
  Xi=[];

  t_init = t1;

  count = 2; %  just so we're starting from index #2 when assigning new
particle positions

  for t=5:5:20

      dt = t-t_init; % time step given by time interval
      Xt2 = cell(1,N); % blank array for updated position

    for i=1:1:N

        xi = Xt1{i}; % extracting particle from array

        x = xi(1);
        y = xi(2); % extracting X, Y and theta from particle
        angle = xi(3);

        T_x1 = [cos(angle),-sin(angle),x % creating homogenous T for particle
xi
                sin(angle),cos(angle),y
                0, 0, 1];

    % calculating motion model on lie group
        phi_r_noise = phi_r + (sig_r*randn());
        phi_l_noise = phi_l + (sig_l*randn());

        omega_dot = [0,-(r/w)*(phi_r_noise-phi_l_noise),(r/
```

```
2)*(phi_r_noise+phi_l_noise)
                 (r/w)*(phi_r_noise-phi_l_noise),0,0
                 0,0,0];
    % converting lie group to euclidean space and updating particle
    % position using exp map

        T_x2 = T_x1 * expm(dt*omega_dot);
   % extracting X Y and theta from new particle pose and reassining it to
updated particle vector
        Xt2{i} = [T_x2(1,3);T_x2(2,3);atan2(T_x2(2,1),T_x2(1,1))] ;
    end

    Xtn{count} = Xt2; % transferring new pose to matrix containing all time
poses
   Xt1 = Xt2; % resetting starting pose
   t_init = t; % updating time step
   count = count+1; %updating counter for next iteration
  end
```

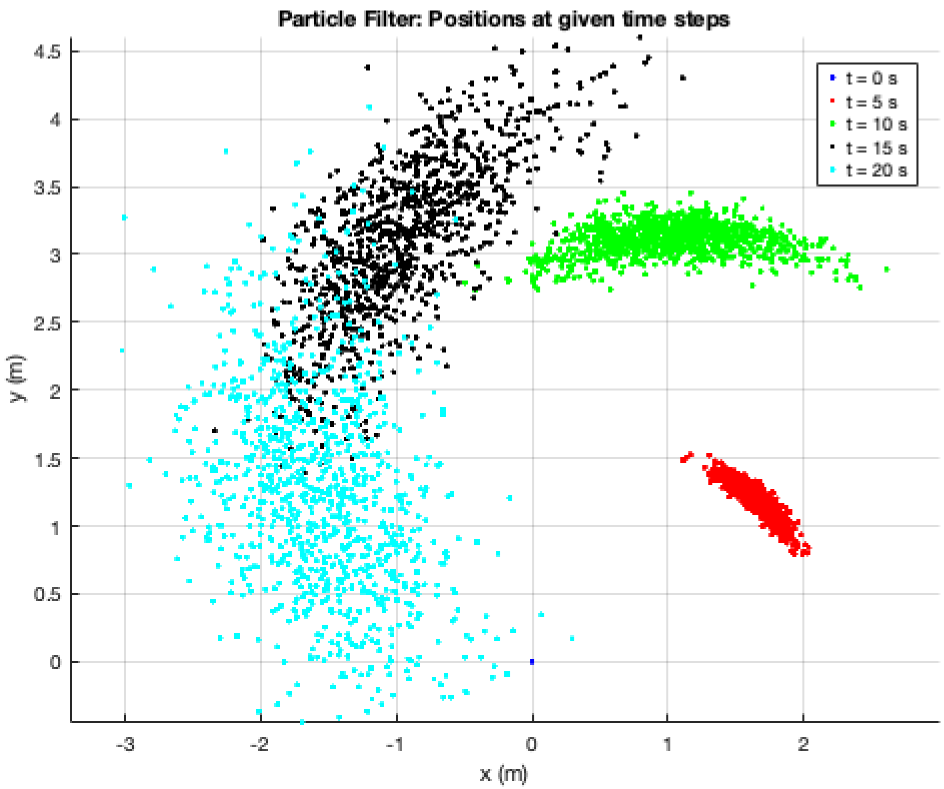
Plotting Code

```
% Calculating Mean for every position
num_iters = 5;
times = [0, 5, 10, 15, 20];

for t = 1:num_iters
    coords = [N,2];

    % Extract positions
    for i = 1:N
        positions(i, :) = Xtn{t}{i}(1:2)'; % extracting X,Y position from
parent array containing all iteration information
    end

    % Calculate mean and covariance
    t
    mean_pos = mean(positions, 1)
    cov_pos = cov(positions)

end

% Plot all particle sets on one plot
figure;
hold on;

colors = {'b', 'r', 'g', 'k', 'c'};
markers = {'.', '.', '.', '.', '.'};

for t = 1:num_iters
    positions = [N,2];

    % Extract positions
    for i = 1:N
```

```
        positions(i, :) = Xtn{t}{i}(1:2)';
    end

    % Plot particles
    plot(positions(:,1), positions(:,2), [colors{t},
markers{t}],'MarkerSize', 5, 'DisplayName', sprintf('t = %d s', times(t)));
end

xlabel('x (m)');
ylabel('y (m)');
title('Particle Filter: Positions at given time steps');
legend('Location', 'best');
grid on;
axis equal;
hold off;
```

```
t =

     1


mean_pos =

     0     0


cov_pos =

     0     0
     0     0


t =

     2


mean_pos =

    1.6467    1.1964


cov_pos =

    0.0200   -0.0159
   -0.0159    0.0157


t =

     3
```

```
mean_pos =

    1.0317    3.1078


cov_pos =

    0.2504    0.0062
    0.0062    0.0143


t =

     4


mean_pos =

   -0.9387    3.1189


cov_pos =

    0.2725    0.2214
    0.2214    0.3467


t =

     5


mean_pos =

   -1.5512    1.2755


cov_pos =

    0.2325   -0.1188
   -0.1188    0.6127
```



Published with MATLAB® R2024b