

Homework 3 - Monte Carlo Analysis

Jose Eduardo Laruta Espejo

Facultad de Ingeniería - Universidad Mayor de San Andrés

June 3, 2019

1 Projectile model

We are going to use the following model for our projectile:

$$\sum F = Ma \quad (1)$$

If we take the x and y components and the gravity effect into account we have:

$$Ma_x = -F_x \quad (2)$$

$$Ma_y = -F_y - F_g \quad (3)$$

$$F_x = F \cos \theta_0 \quad (4)$$

$$F_y = F \sin \theta_0 \quad (5)$$

$$F = Bv + D|v|^2 \quad (6)$$

our states are given by:

$$x_1 = y$$

$$x_2 = v_y$$

$$x_3 = x$$

$$x_4 = v_x$$

then, our state space form model is as follows:

$$\dot{x}_1 = x_2 \quad (7)$$

$$\dot{x}_2 = -\frac{(Bv + D|v|^2)\cos(\theta_0)}{M} - g \quad (8)$$

$$\dot{x}_3 = x_4 \quad (9)$$

$$\dot{x}_4 = -\frac{(Bv + D|v|^2)\sin(\theta_0)}{M} \quad (10)$$

where $v = \sqrt{x_2^2 + x_4^2}$ the current magnitude of the velocity and θ_0 the initial angle. The initial conditions are defined statistically by $E[\theta_0] = 45[deg]$, $E[v_0] = 200[m/s]$ and standard deviation of 3%. Then, the initial condition vector given θ_0 and v_0 are:

$$x_0 = \begin{bmatrix} 0 \\ v_0 \sin \theta_0 \\ 0 \\ v_0 \cos \theta_0 \end{bmatrix}$$

the model expressed in matlab code:

```

1 function xdot = projectile_model(t,x)
2     % Model constant and parameters
3     global theta0;
4     B = 0.01;           %N sec/m
5     D = 3.0e-5;         % N sec^2/m^2
6     M = 1;             % Kg
7     g = 9.8;           %m/s^2
8     % Compute useful values
9     v = sqrt((x(2)^2) + (x(4)^2));
10    theta = atan2(x(2), x(4));
11    F = B*v + D*abs(v)^2;
12    Fx = F * cos(theta);
13    Fy = F * sin(theta);
14    % State space
15    xdot(1) = x(2);
16    xdot(2) = -Fy - g;
17    xdot(3) = x(4);
18    xdot(4) = -Fx;
19    % reset velocity if hit the ground
20    if (x(1) <=0) && (t > 0)
21        xdot(1) = 0;
22        xdot(3) = 0;
23    end
24    xdot = xdot(:);
25 end

```

2 Monte Carlo Simulation

The code for the Simulation is as follows:

```

1
2 tspan = [0 40]; %reasonable time for steady state
3 % random initial condition
4 m_v0 = 200;           % mean for initial velocity
5 sigma_v0 = 0.03;      % sigma for initial velocity
6 m_theta0 = deg2rad(45); % mean for initial angle (radians)
7 sigma_theta0 = 0.03;  % sigma for initial angle
8 lambda = 3;           % kurtosis (3 = gaussian)
9 n_sig = 1.96;         % n_sigma
10 % for montecarlo simulation
11 n_trials = 10000;     % runs
12 stats_idx = 1;        % index for fill in the statistics
13 final_d = []          % empty array
14 hold on;

```

```

15 title('ballistic trajectory problem')
16 xlabel('range[m]')
17 ylabel('height[m]')
18 t_1 = [0 3300]; x_1 = [0 0];
19 plot(t_1,x_1,'-')
20 for q = 1:n_trials
21     % simulation parameters
22     v0 = m_v0 + sigma_v0*randn; % random initial velocity
23     theta0 = m_theta0 + sigma_theta0*randn; % random initial angle
24     % initial states for simulation;
25     x0 = [0; v0*sin(theta0); 0; v0*cos(theta0)]; % initial states
26     [t, x] = ode45('projectile_model', tspan, x0); % simulate
27     final_d = [final_d; x(length(t), 3)]; % save final distance
28
29     if mod(q, 100) == 0 % every 100 trials
30         % plot trajectory
31         plot(x(:,3), x(:,1));
32         % compute statistics
33         m_hat(stats_idx) = sum(final_d) / q
34         dif_sq = (final_d - m_hat(stats_idx)).^2;
35         p_hat = sum(dif_sq) / (q - 1);
36         debias = q/(q-1);
37         sigma(stats_idx) = sqrt(debias*p_hat); % save sigma
38         % confidence limits for sigma
39         sigma_low(stats_idx) = sigma(stats_idx)/(1 + n_sig * sqrt((lambda
40             -1)/q));
41         sigma_high(stats_idx) = sigma(stats_idx)/(1 - n_sig * sqrt((lambda
42             -1)/q));
43         % confidence limits for m
44         m_low(stats_idx) = m_hat(stats_idx) - (n_sig * sigma(stats_idx))/
45             sqrt(q);
46         m_high(stats_idx) = m_hat(stats_idx) + (n_sig * sigma(stats_idx))/
47             sqrt(q);
48         stats_idx = stats_idx + 1;
49     end
50 end
51 hold off
52 %%
53 % plot statistics
54 % mean distance
55 figure
56 t_plot = 1:length(sigma);
57
58 plot(t_plot, m_hat, t_plot, m_high, '+', t_plot, m_low, '+')
59 title('mean impact distance')
60 xlabel('cummulative set number x100')
61 ylabel('mean impact distance')
62 % sigma
63 figure
64 plot(t_plot, sigma, t_plot, sigma_high, '+', t_plot, sigma_low, '+')
65 title('sigma impact distance')
66 xlabel('cummulative set number x100')
67 ylabel('sigma impact distance')

```

after running the simulation the resulting plots are:

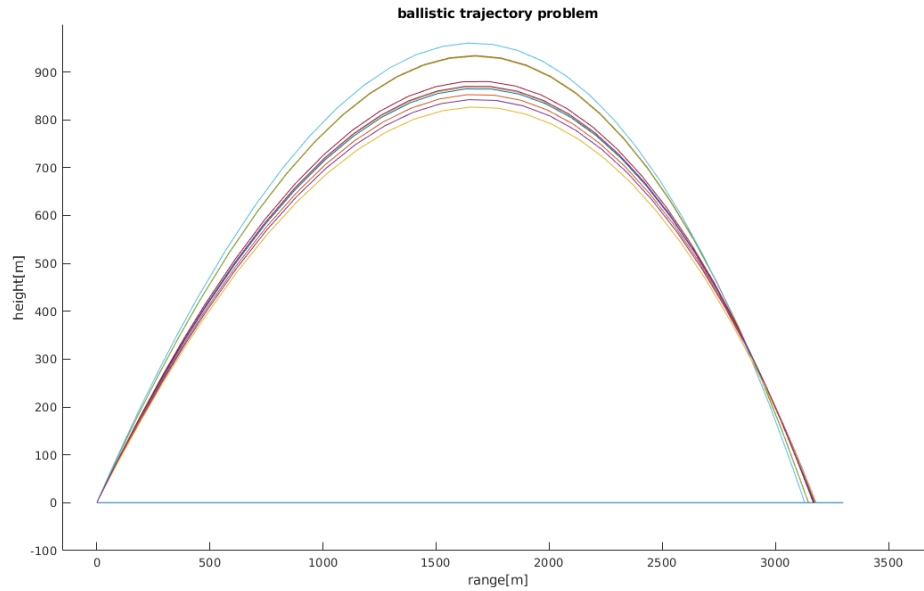


Figure 1: Simulation of the model

In the case of the simulation of the projectile trajectories, Fig 1 shows clearly the variation due to random initial conditions.

For the statistics we can see clearly the convergence of the mean as soon as the Monte Carlo trials begin to increase, the same for the standard deviation. In Fig 2 the mean is slowly converging and the interval of confidence decrease in size. The same goes for Sigma in Fig 3. Just for fun, the plots for 10000 iteration are shown in Figs 4 and 5, where we can see the convergence of the mean in 3159 m and sigma in 19.62 m.

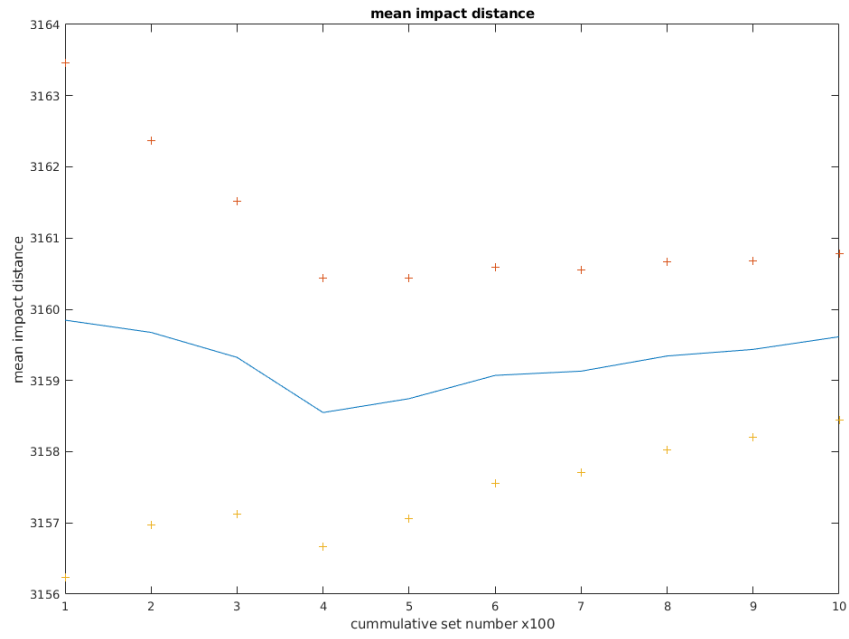


Figure 2: Mean convergence 1000 runs

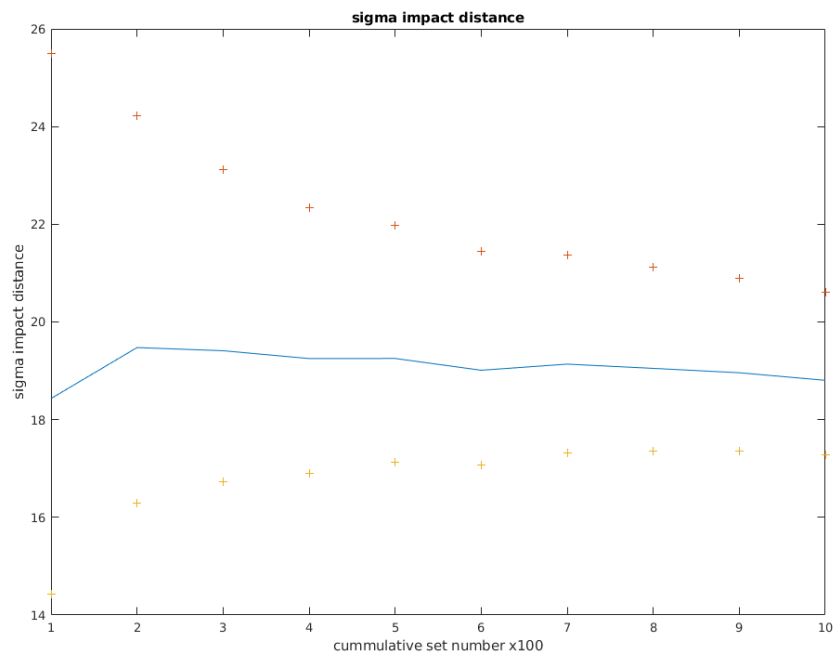


Figure 3: Sigma convergence 1000 runs

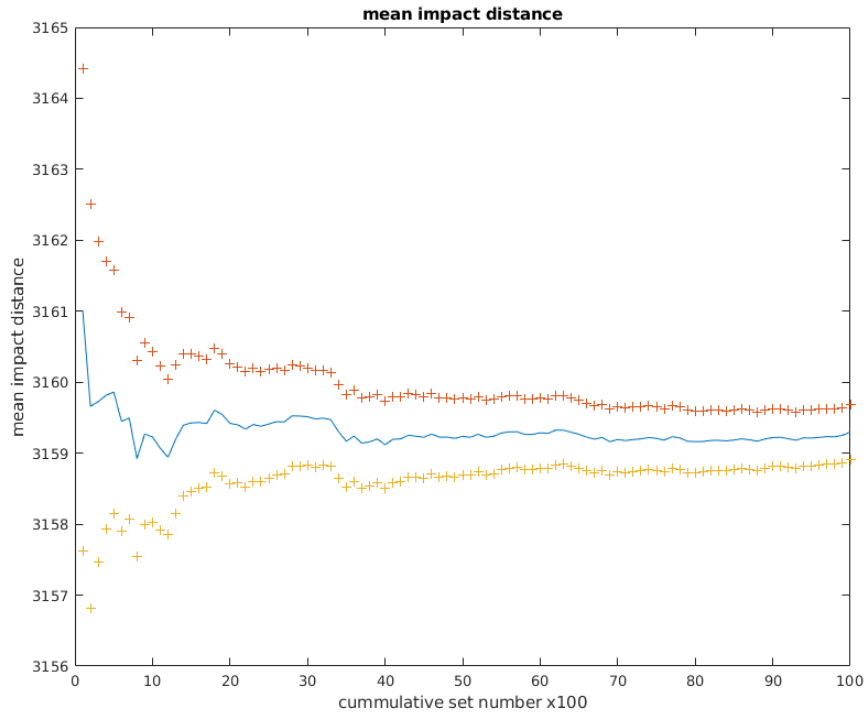


Figure 4: Mean convergence 10000 runs

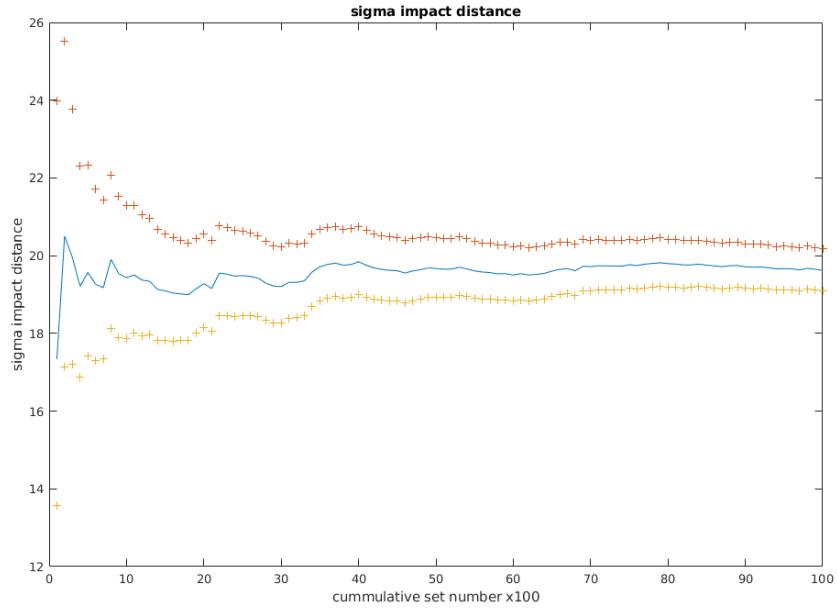


Figure 5: Sigma convergence 10000 runs