Asignatura	Datos del alumno	Fecha
Modelado de Sistemas Dinámicos	**Apellidos:** Avilés Cahill	27/02/2024
	Nombre: Adán	

Laboratorio: Modelos aleatorios y la simulación de Monte Carlo

Índice

Índice de figuras

Asignatura	Datos del alumno	Fecha
Modelado de Sistemas Dinámicos	**Apellidos:** Avilés Cahill	27/02/2024
	Nombre: Adán	

1. Introducción

En este laboratorio exploraremos en Matlab el potencial de la simulación de Monte Carlo en dos ejercicios diferentes, en el primer ejercicio habremos de calcular el volumen bajo la superficie definida por una función y ciertos criterios, y en segundo lugar, modelaremos el sistema de doble amortiguador-resorte y haremos un estudio de Monte Carlo para evaluar la incertidumbre provocada por los errores en las masas.

2. Actividad 1: Cálculo del volumen

Como se ha comentado, calcularemos el volumen definido por la función

$$f(x, y) = 3x^2 - \sin y$$

En el anillo de radio interior $r_i = 1$ y radio exterior $r_e = 2$. Utilizando el método de Monte Carlo, podemos aproximar este volumen por

$$\int\int_{Anillo} f(x, y)dxdy = \text{area\_anillo} \cdot \frac{n_e}{n}$$

Siendo n_e el número de puntos que caen en nuestro recinto deseado, y n el total de puntos simulados. En primer lugar, y por simplificar los cálculos, consideraremos coordenadas polares, esto es:

$$\begin{cases} x = r \cdot cos(\theta) \\ y = r \cdot sin(\theta) \end{cases}$$

Por tanto, nuestra integral para el cálculo pasa a ser:

$$\int\int_{Anillo} f(x, y)dxdy = \int_0^{2\pi}\int_1^2 f(r, \theta)rdrd\theta$$

Ahora, generaremos tres números aleatorios y haremos que estén acotados por nuestros límites, esto es:

- $1 <= r <= 2$
- $0 <= \theta <= 2\pi$
- $-1 <= z <= 12$, pues son el máximo y el mínimo posibles de $f(r, \theta)$ bajo las condiciones anteriores.

Como sabemos, queremos que nuestro punto aleatorio z esté por debajo de $f(r, \theta)$, pero hemos de tener en cuenta que cuando $z < 0$, esta condición ha de ser al inverso. Por tanto, podemos ejecutar el siguiente código en Matlab:

```
ni = 0;
n= 10000;
%Preparamos en memoria los valores para poder plotear los puntos
Xplot = zeros(1,n);
Yplot = zeros(1,n);
Zplot = zeros(1,n);
for i=1:n
    x = rand([1,3]); %Generamos 3 valores aleatorios
    x(1) = x(1)+1; % r va de 1 a 2
    x(2) = x(2)*2*pi; % theta va de 0 a 2pi
    x(3) = -1+x(3)*13; % f(r,theta) va de  -1 a 12
    % Almacenamos en coordenadas para poder pllotear
    Xplot(i) = x(1)*cos(x(2));
    Yplot(i) = x(1)*sin(x(2));
    Zplot(i) = x(3);
    % Computamos la funcion
    fx = 3*(x(1)*cos(x(2)))^2 -sin(x(1)*sin(x(2)));
    % Tener en cuenta que puede irpor arriba o por debajo por z=0
    % diferenciamos esos casos
    %si esta por debajo de cero, queremos que este por encima de fx
    if (x(3)<=0)
        if(x(3) > fx)
            ni=ni +1;
        end
    else
        if(x(3) < fx)
            ni=ni +1;
        end
```

```
    end
end
%Sabemos que V(s) = SxI donde S es el area del anillo e I la medida entre
% -1 y 12 => v(S) = (pi*r_1^2 -pi*r_2^2)*(12--1) = pi(2^2-1^2)*13 = pi*3*13
vol = pi*(3)*(13)*ni/n;

% Definimos la función en su forma polar para pdoer calcularla
fun_polar = @(r,theta) (3*(r.*cos(theta)).^2-sin(r.*sin(theta))).*r; %recordemos que dx
r1=1; r2=2;
theta1=0; theta2=2*pi;
I = integral2(fun_polar,r1, r2,theta1, theta2);
```

Con estos datos, obtenemos que el valor exacto del volumen es $I = 35{,}3429$ y nuestra aproximación $vol = 34{,}7228$, esta aproximación será más exacta con mayor cantidad de puntos.

En la siguiente imagen podemos ver la figura y la distribución de los puntos.

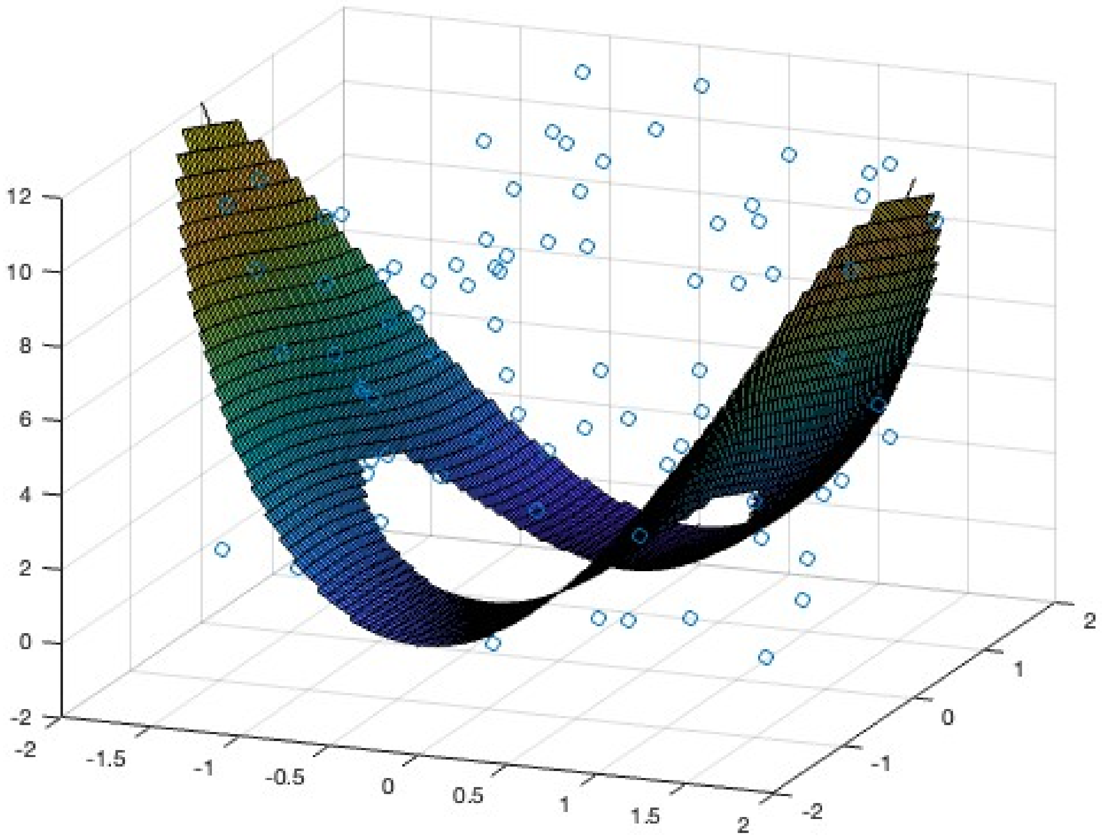

Figura 1: Curva y puntos

Imagen generada con

```
[X,Y] = meshgrid(-10:0.05:10,-10:0.05:10);
curva = X.^2+Y.^2;
Z = 3*X.^2 -sin(Y);
Z(curva<1 | curva > 4)=nan;
surf(X,Y,Z)
hold on
plot3(Xplot,Yplot,Zplot, 'o')
```

3. Actividad 2: Evaluación de la incertidumbre.

Modelaremos en Simulink el sistema de doble resorte, dejando "harcodeadas"todas las variables menos los pesos $m1$ y $m2$ que iremos cambiando con las simulaciones.

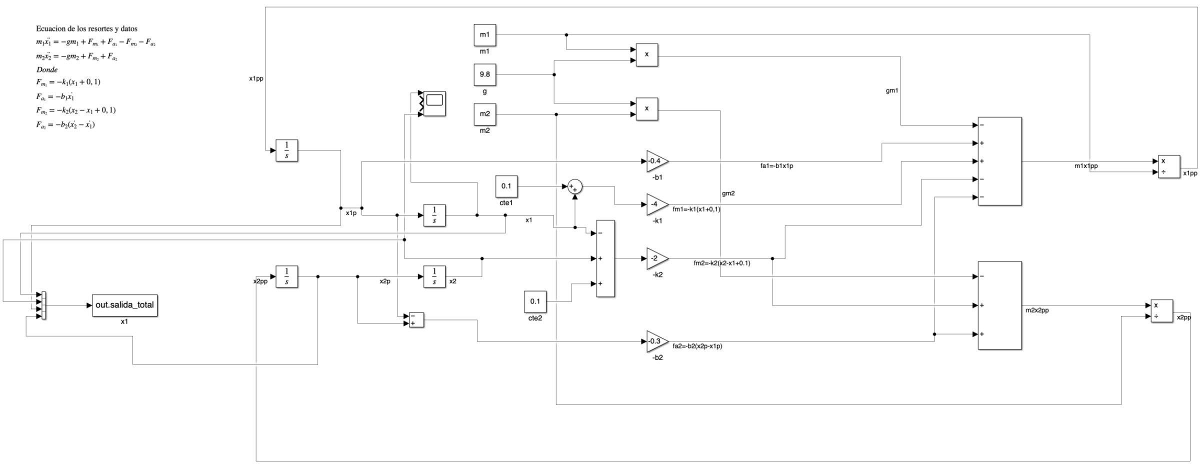

Figura 2: Sistema modelado en Simulink

En el modelado están explicadas las ecuaciones usadas y los pasos. Ahora, hemos de simular 100 pesos diferentes que siguen una distribución normal de media 1 y desviación 0.2. Utilizando el modelo de Simulink y la terminal de Matlab, haremos esta simulación 100 veces.

```
n=100
pos_1=zeros(1,n);
pos_2=zeros(1,n);
equilibrio_1 = zeros(1,n);
equilibrio_2 = zeros(1,n);
equilibrio_total = zeros(1,n);
for i=1:n
    m1 = normrnd(1,0.2);
    m2 = normrnd(1,0.2);
    rsim = sim('modelo_resorte.slx');
```

```
    pos_1(i) = rsim.salida_total(25,1);
    pos_2(i) = rsim.salida_total(25,2);
    % Equilibrio 1
    j=8; % Elijo t=8 pata que no sea estanco
    while abs(rsim.salida_total(j+1,1)-rsim.salida_total(j,1))>0.05 ...
            || abs(rsim.salida_total(j,3)) > 0.05
        j=j+1;
    end
    equilibrio_1(i) = j;
    % Equilibrio 2
    j=8;
    while abs(rsim.salida_total(j+1,2)-rsim.salida_total(j,2))>0.05 ...
            || abs(rsim.salida_total(j,4)) > 0.05
        j=j+1;
    end
    equilibrio_2(i) = j;
    % Equilibrio total
    j=8;
    while abs(rsim.salida_total(j+1,1)-rsim.salida_total(j,1))>0.05 ...
            || abs(rsim.salida_total(j,3)) > 0.05 ...
            || abs(rsim.salida_total(j+1,2)-rsim.salida_total(j,2))>0.05 ...
            || abs(rsim.salida_total(j,4)) > 0.05
        j=j+1;
    end
    equilibrio_total(i) = j;
end
```

Como se puede ver en el código, pos_1 y pos_2 son las posiciones de x_1 y x_2 en el segundo 25, almacenadas en un vector para poder ser represenatadas y ploteadas. Lo mismo hemos hecho con el equilibrio, donde exigimos que se cumplan ciertas condiciones, y una vez cumplidas, almacenamos el t que las satisfaga. Como para t pequeños se encontraba un falso equilibrio, hemos decidido coger como mínimo $t = 8$. Obteniendo que x_1 se encuentra en mean(pos_1) $= -5{,}3402$ de media al segundo 25, con distribución:

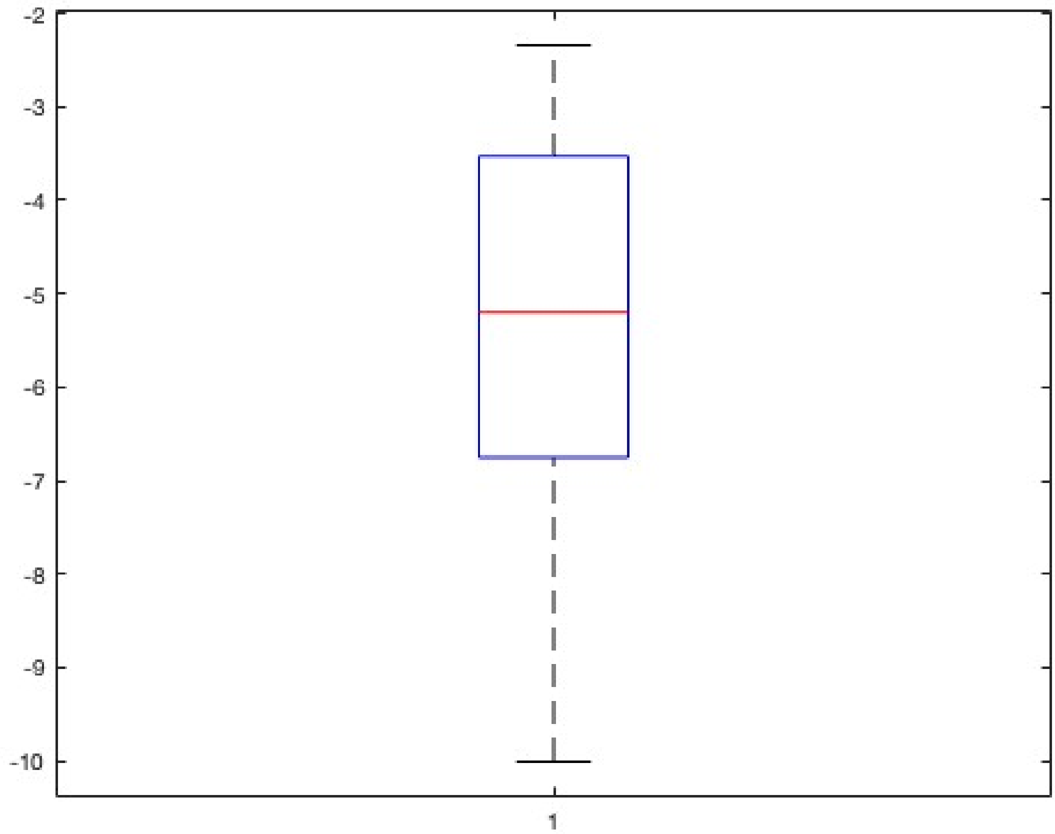

Figura 3: Posición de x1 en t=25 tras 100 repeticiones

Seguidamente, x_2 se encuentra en mean(pos_2) $= -11{,}0464$ de media al segundo 25, con distribución:

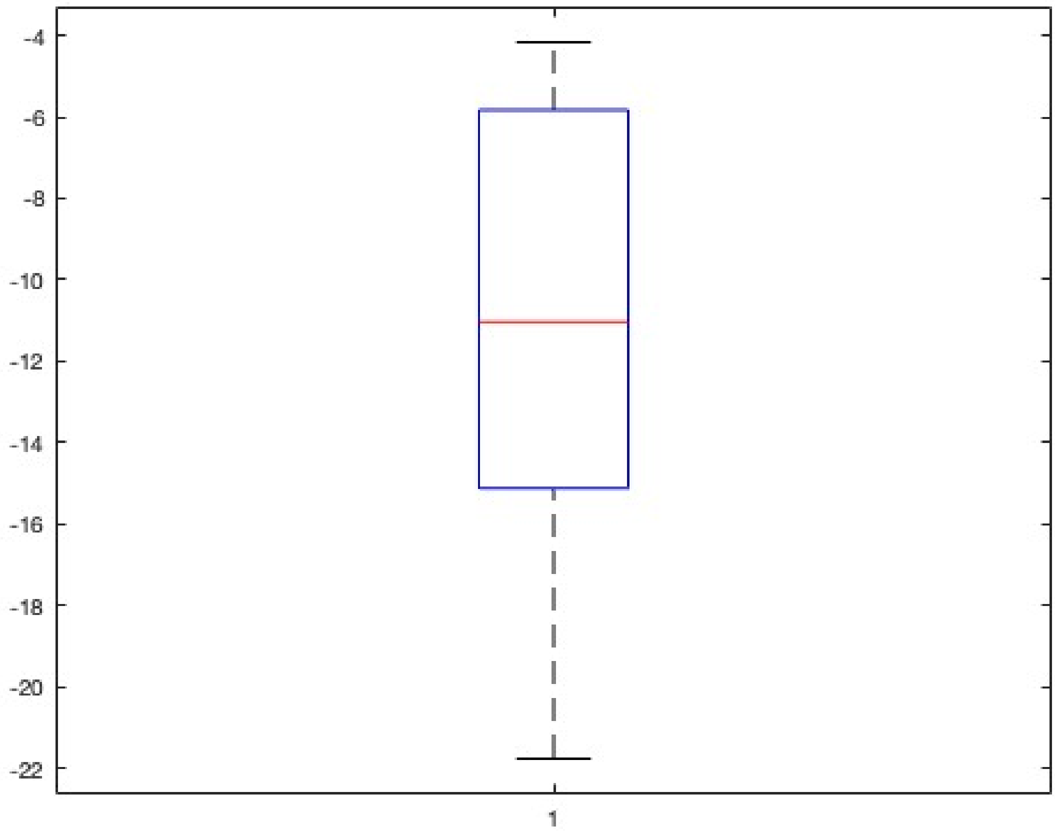

Figura 4: Posición de x2 en t=25 tras 100 repeticiones

Respecto al equilibrio, el para el subsistema primero, t_para_equilibrio_1 $= 63{,}2900$ de media, con distribución:

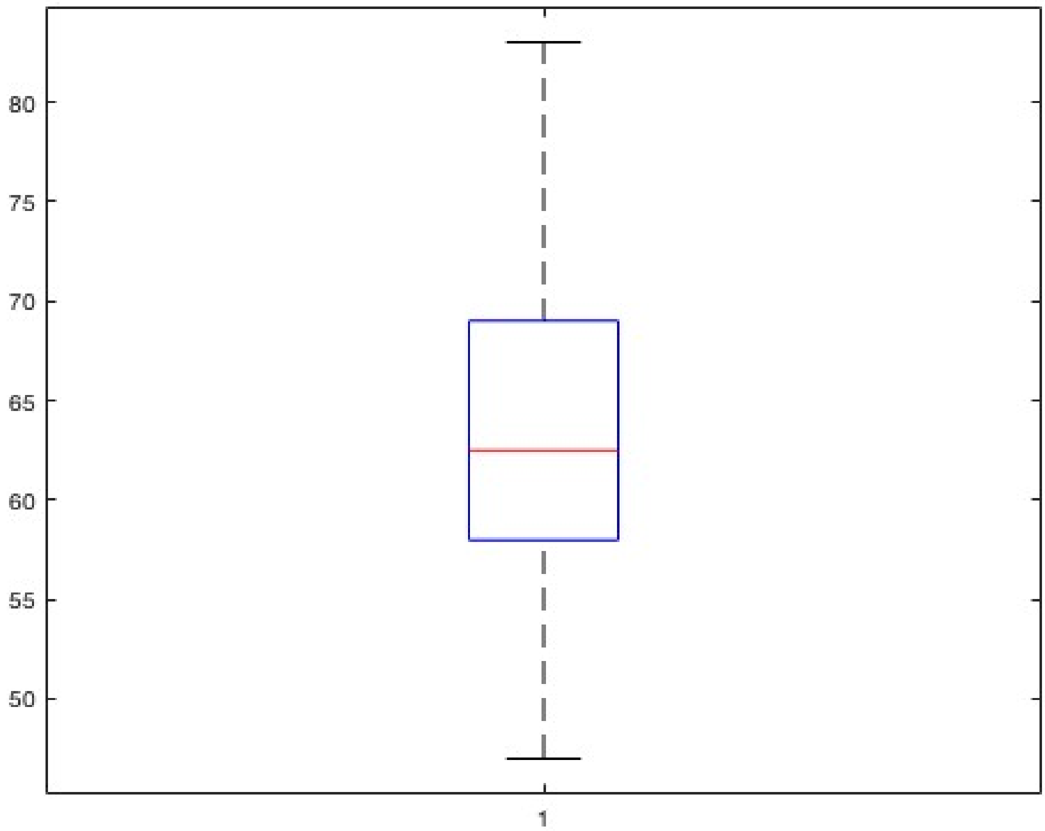

Figura 5: Distribución del tiempo de equilibrio de m1 tras 100 repeticiones

Finalmente el para el subsistema segundo, t_para_equilibrio_2 $= 73{,}7400$ de media, con distribución:

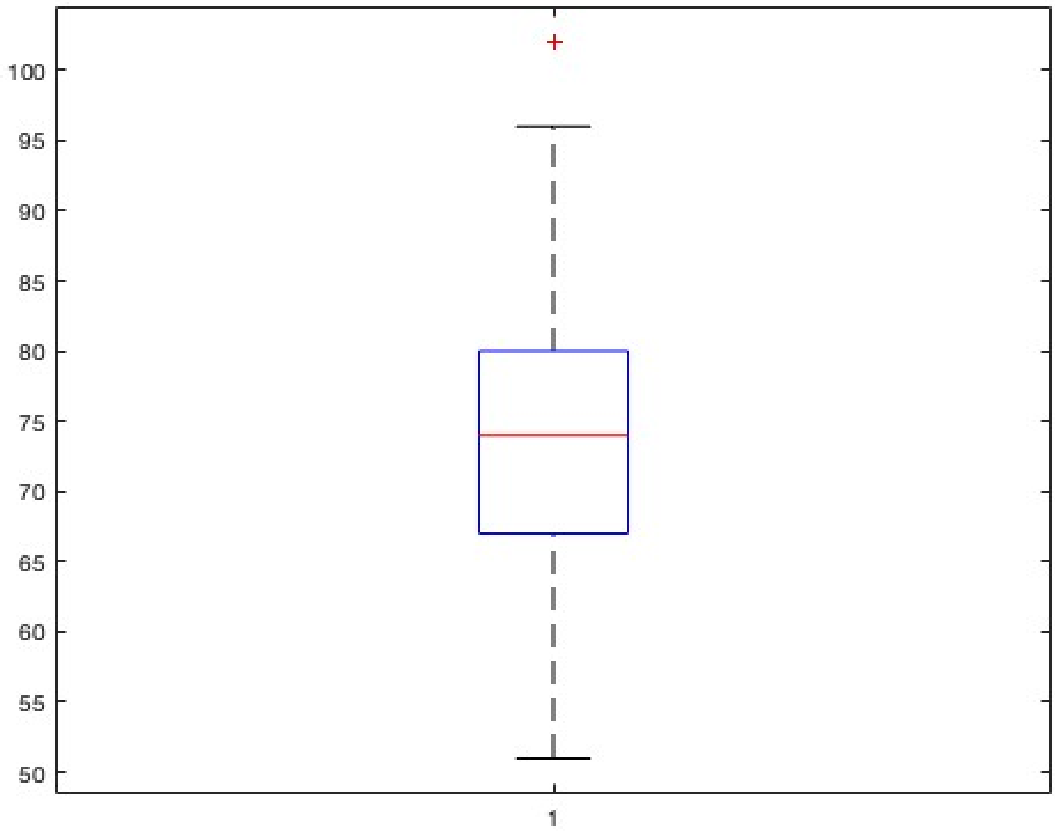

Figura 6: Distribución del tiempo de equilibrio de m2 tras 100 repeticiones

Con esto, podemos ver el potencial que tiene el método de Monte Carlo. Como curiosidad, yo personalmente lo he usado en mi trabajo para cálculos de Out Of Stock.