

Anexo: Códigos

En este apartado se incluyen todos los códigos escritos en entorno Matlab, necesarios para el desarrollo del estudio detallado en este proyecto.

Índice de los códigos:

1.	Procesamiento de datos de un paso	62
1.1.	Función buscaPasos	62
1.1.	Función fourier4	62
1.2.	Código para un paso derecho	64
1.3.	Código para un paso izquierdo.....	67
1.4.	Código PaseoFourier	69
2.	Procesamiento de datos de tres pasos	75
2.1.	Función buscaPasos	75
2.2.	Función fourier8	76
2.3.	Función spline	77
2.4.	Código para pasos derecho, izquierdo, derecho.....	78
2.5.	Código para pasos izquierdo, derecho izquierdo.....	81
2.6.	Código AjusteSplineDID.....	84

1. Procesamiento de datos de un paso

1.1. Función buscaPasos

```
function [Tini, Tfin] = buscaPasos(T, F, umMin, umMax, Nextra, Tmax)
% Función que busca los instantes inicial y final de los pasos dentro % de
% un registro donde hay muchos.
% Parámetros de entrada:
% T: vector de tiempos (s)
% F: vector de registro de fuerza (N)
% umMin: umbral mínimo para considerar un paso (N)
% umMax: umbral máximo al que deberían llegar todos los pasos (N)
% Nextra: número de puntos extra a considerar antes y después
% Tmax: máxima duración de un paso (s)

dT = T(2) - T(1);
Tini = [];
Tfin = [];
i = 1;
while(i < length(T))
    if(F(i) >= umMin)
        Ti = T(i);
        Tini = [Tini Ti-Nextra*dT];
        Tf = min(Ti + T(F(T) > Ti) <= umMin));
        Tfin = [Tfin Tf+Nextra*dT];
        i = find(T >= Tfin(end), 1)+1;
    else
        i = i + 1;
    end
end

% Quito los pasos que no superan el umbralMax
for i = 1:length(Tini)
    if(max(F(T >= Tini(i) & T <= Tfin(i))) <= umMax)
        Tini(i) = NaN;
        Tfin(i) = NaN;
    end
end

Tini = Tini(~isnan(Tini));
Tfin = Tfin(~isnan(Tfin));

% Finalmente, quito los pasos de duración excesiva
for i = 1:length(Tini)
    if(Tfin(i) - Tini(i) > Tmax)
        Tini(i) = NaN;
        Tfin(i) = NaN;
    end
end

Tini = Tini(~isnan(Tini));
Tfin = Tfin(~isnan(Tfin));

end
```

1.1. Función fourier4

```
function [res, gof] = fourier4(Tpaso, Ypaso)
%CREATEFIT(TPASO,YPASO)
% Create a fit.
```

```
%
% Data for 'untitled fit 1' fit:
%     X Input : Tpasso
%     Y Output: Ypasso
% Output:
%     fitresult : a fit object representing the fit.
%     gof : structure with goodness-of fit info.
%
% See also FIT, CFIT, SFIT.

% Auto-generated by MATLAB on 05-Jul-2019 19:40:09
[xData, yData] = prepareCurveData( Tpasso, Ypasso );

% Set up fittype and options.
ft = fittype( 'fourier4' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.Display = 'Off';
opts.Normalize = 'on';
opts.StartPoint = [0 0 0 0 0 0 0 0 0 1.8421157563932];

% Fit model to data.
[res, gof] = fit( xData, yData, ft, opts );

% Plot fit with data.
%figure( 'Name', 'untitled fit 1' );
%h = plot( res, xData, yData );
%legend( h, 'Ypasso vs. Tpasso', 'untitled fit 1', 'Location', 'NorthEast' );
% Label axes
%xlabel Tpasso
%ylabel Ypasso
%grid on
```

1.2. Código para un paso derecho

```
%Tratamiento de datos.
clear;

%Cargo datos experimentales
DATA1 = load('UnPasoD-1.mat');

%Llamada a la función 'buscaPasos'.
T = DATA1.Data1_time_AI_5;
F = double(DATA1.Data1_AI_5-0.0947);
umMin = 0.001;
umMax = 0.08;
Nextra = 1;
Tmax = 1.2;
[Tini, Tfin] = buscaPasos(T, F, umMin, umMax, Nextra, Tmax);

%Calibración
M = 509.6; %peso del sujeto en N
m = 0.1033; %media de datos estáticos
f = M/m; %factor de corrección
F = F * f;
DATA1.Data1_AI_5 = DATA1.Data1_AI_5 * f;

%Graficar DATA1 completo.
figure(1)
plot(T,F)
title('D')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%Graficar un paso.
paso = 1;
figure(2)
Tpaso = T(T >= Tini(paso) & T <= Tfin(paso));
Ypaso = F(T >= Tini(paso) & T <= Tfin(paso));
Tpaso = Tpaso - Tpaso(1);
Ypaso = Ypaso - Ypaso(1);
plot(Tpaso, Ypaso)
title('Un paso D')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%Integral.
q = trapz(Tpaso, Ypaso.*Tpaso);
b = trapz(Tpaso,Ypaso);
Tcdg = q/b;

%Graficar varios pasos.
figure(3); hold on
for i = 1:length(Tini)
    paso = i;
    Tpaso = DATA1.Data1_time_AI_5(DATA1.Data1_time_AI_5 >= Tini(paso) &
DATA1.Data1_time_AI_5 <= Tfin(paso));
    Ypaso = DATA1.Data1_AI_5(DATA1.Data1_time_AI_5 >= Tini(paso) &
DATA1.Data1_time_AI_5 <= Tfin(paso));
    Tpaso = Tpaso - Tpaso(1);
    Ypaso = Ypaso - Ypaso(1);
```

```

        q = trapz(Tpaso, Ypaso.*Tpaso);
        b = trapz(Tpaso,Ypaso);
        Tcdg = q/b;
        plot(Tpaso-Tcdg, Ypaso)
    end
    i
    hold off
    title('Varios pasos D')
    xlabel('Tiempo (s)');
    ylabel('Fuerza (N)');

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Identificación estadística %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Fourier de orden 4: 10 parámetros
    % a0 a1 b1 a2 b2 a3 b3 a4 b4 w
    parFou4 = zeros(length(Tini), 10);
    gof_sse = zeros(length(Tini),1);
    gof_r2 = zeros(length(Tini),1);

    figure(4); hold on
    for i = 1:length(Tini)
        paso = i;
        Tpaso = T(T >= Tini(paso) & T <= Tfin(paso));
        Ypaso = F(T >= Tini(paso) & T <= Tfin(paso));
        Tpaso = Tpaso - Tpaso(1);
        Ypaso = Ypaso - Ypaso(1);

        [res, gof] = fourier4(Tpaso,Ypaso);
        parFou4(i, :) = [res.a0, res.a1, res.b1, res.a2, res.b2, res.a3,
res.b3, res.a4, res.b4, res.w];
        gof_sse(i) = gof.sse;
        gof_r2(i) = gof.rsquare;
        plot(res, Tpaso, Ypaso)
    end
    hold off
    title('Aproximación de Fourier 4 términos')
    xlabel('Tiempo (s)');
    ylabel('Fuerza (N)');

    %Grafica tres pasos comparativa con la aproximación de fourier
    figure(5)
    h = plot( res, 'r', Tpaso, Ypaso, 'c*' )
    title('Comparativa de la aproximación de Fourier 4 términos')
    xlabel('Tiempo (s)');
    ylabel('Fuerza (N)');

    %Media de cada parametro de Fourier
    media = mean(parFou4)

    %Desviación típica de cada parametro de Fourier
    desv = std(parFou4)

    %Parametros de Fourier pseudoaleatorios
    pd_a0 = makedist ('Normal', 'mu', media(1), 'sigma', desv(1));
    pd_a0 = makedist ('Normal', 'mu', media(2), 'sigma', desv(2));
    pd_a0 = makedist ('Normal', 'mu', media(3), 'sigma', desv(3));

```

Registro, caracterización y modelado de las fuerzas verticales inducidas al andar.

```
pd_a0 = makedist ('Normal', 'mu', media(4), 'sigma', desv(4));
pd_a0 = makedist ('Normal', 'mu', media(5), 'sigma', desv(5));
pd_a0 = makedist ('Normal', 'mu', media(6), 'sigma', desv(6));
pd_a0 = makedist ('Normal', 'mu', media(7), 'sigma', desv(7));
pd_a0 = makedist ('Normal', 'mu', media(8), 'sigma', desv(8));
pd_a0 = makedist ('Normal', 'mu', media(9), 'sigma', desv(9));
pd_a0 = makedist ('Normal', 'mu', media(10), 'sigma', desv(10));

rand_a0 = random(pd_a0);
rand_a1 = random(pd_a0);
rand_b1 = random(pd_a0);
rand_a2 = random(pd_a0);
rand_b2 = random(pd_a0);
rand_a3 = random(pd_a0);
rand_b3 = random(pd_a0);
rand_a4 = random(pd_a0);
rand_b4 = random(pd_a0);
rand_w = random(pd_a0);

rand = [rand_a0 rand_a1 rand_b1 rand_a2 rand_b2 rand_a3 rand_b3 rand_a4
rand_b4 rand_w]
```

1.3. Código para un paso izquierdo

```
%Tratamiento de datos.
clear;

%Cargo datos experimentales
DATA1 = load('UnPasoI-1.mat');

%Llamada a la función 'buscaPasos'.
T = DATA1.Data1_time_AI_5;
F = double(DATA1.Data1_AI_5-0.0947);
umMin = 0.001;
umMax = 0.08;
Nextra = 1;
Tmax = 1.2;
[Tini, Tfin] = buscaPasos(T, F, umMin, umMax, Nextra, Tmax);

%Calibración
M = 509.6; %peso del sujeto en N
m = 0.1099; %media de datos estáticos
f = M/m; %factor de corrección
F = F * f;
DATA1.Data1_AI_5 = DATA1.Data1_AI_5 * f;

%Graficar DATA1 completo.
figure(1)
plot(T,F)
title('I')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%Graficar un paso.
paso = 1;
figure(2)
Tpaso = T(T >= Tini(paso) & T <= Tfin(paso));
Ypaso = F(T >= Tini(paso) & T <= Tfin(paso));
Tpaso = Tpasos - Tpasos(1);
Ypaso = Ypasos - Ypasos(1);
plot(Tpaso, Ypaso)
title('Un paso I')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%Integral.
q = trapz(Tpaso, Ypaso.*Tpaso);
b = trapz(Tpaso,Ypaso);
Tcdg = q/b;

%Graficar varios pasos.
figure(3); hold on
for i = 1:length(Tini)
    paso = i;
    Tpaso = DATA1.Data1_time_AI_5(DATA1.Data1_time_AI_5 >= Tini(paso) &
DATA1.Data1_time_AI_5 <= Tfin(paso));
    Ypaso = DATA1.Data1_AI_5(DATA1.Data1_time_AI_5 >= Tini(paso) &
DATA1.Data1_time_AI_5 <= Tfin(paso));
    Tpaso = Tpaso - Tpaso(1);
    Ypaso = Ypaso - Ypaso(1);
```

Registro, caracterización y modelado de las fuerzas verticales inducidas al andar.

```
    q = trapz(Tpaso, Ypaso.*Tpaso);
    b = trapz(Tpaso, Ypaso);
    Tcdg = q/b;
    plot(Tpaso-Tcdg, Ypaso)
end
i
hold off
title('Varios pasos I')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Identificación estadística %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fourier de orden 4: 10 parámetros
% a0 a1 b1 a2 b2 a3 b3 a4 b4 w
parFou4 = zeros(length(Tini), 10);
gof_sse = zeros(length(Tini),1);
gof_r2 = zeros(length(Tini),1);

figure(4); hold on
for i = 1:length(Tini)
    paso = i;
    Tpaso = T(T >= Tini(paso) & T <= Tfin(paso));
    Ypaso = F(T >= Tini(paso) & T <= Tfin(paso));
    Tpaso = Tpaso - Tpaso(1);
    Ypaso = Ypaso - Ypaso(1);

    [res, gof] = fourier4(Tpaso, Ypaso);
    parFou4(i, :) = [res.a0, res.a1, res.b1, res.a2, res.b2, res.a3,
res.b3, res.a4, res.b4, res.w];
    gof_sse(i) = gof.sse;
    gof_r2(i) = gof.rsquare;
    %plot(res)
end
hold off
title('Aproximación de Fourier 4 términos')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)')
%Grafica tres pasos comparativa con la aproximación de fourier
figure(5)
h = plot( res, Tpaso, Ypaso )
title('Comparativa de la aproximación de Fourier 4 términos')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');
%Media de cada parametro de Fourier
media = mean(parFou4)
%Desviación típica de cada parametro de Fourier
desv = std(parFou4)
```


1.4. Código PaseoFourier

```

%Modelado Fourier 4
clear;

%Carga datos experimentales
DATA1 = load('UnPasoD-1.mat');
DATA2 = load('UnPasoI-1.mat');

%DATA1
%Llamada a la función 'buscaPasos'.
T_d = DATA1.Data1_time_AI_5;
F_d = double(DATA1.Data1_AI_5-0.0947);
umMin = 0.001;
umMax = 0.08;
Nextra = 1;
Tmax = 1.2;
[Tini_d, Tfin_d] = buscaPasos(T_d, F_d, umMin, umMax, Nextra, Tmax);

%Calibración
M = 509.6; %peso del sujeto en N
m = 0.1033; %media de datos estáticos
f = M/m; %factor de corrección
F_d = F_d * f;
DATA1.Data1_AI_5 = DATA1.Data1_AI_5 * f;

%Definición de paso
paso = 1;
Tpaso_d = T_d(T_d >= Tini_d(paso) & T_d <= Tfin_d(paso));
Ypaso_d = F_d(T_d >= Tini_d(paso) & T_d <= Tfin_d(paso));
Tpaso_d = T_paso_d - T_paso_d(1);
Ypaso_d = Y_paso_d - Y_paso_d(1);

%Integral.
q = trapz(T_paso_d, Y_paso_d.*T_paso_d);
b = trapz(T_paso_d, Y_paso_d);
Tcdg = q/b;

%Graficar un paso, derecha e izquierda.
figure(1); hold on

Tpaso_d = zeros(length(Tini_d), 1);
Tpaso_d_ant = zeros(length(Tini_d), 1);

for i = 1:length(Tini_d)
    paso = i;
    Tpaso_d = DATA1.Data1_time_AI_5(DATA1.Data1_time_AI_5 >= Tini_d(paso) &
DATA1.Data1_time_AI_5 <= Tfin_d(paso));
    Ypaso_d = DATA1.Data1_AI_5(DATA1.Data1_time_AI_5 >= Tini_d(paso) &
DATA1.Data1_time_AI_5 <= Tfin_d(paso));
    Tpaso_d = Tpaso_d - Tpaso_d(1);
    Ypaso_d = Ypaso_d - Ypaso_d(1);
    q = trapz(T_paso_d, Y_paso_d.*T_paso_d);
    b = trapz(T_paso_d, Y_paso_d);
    Tcdg = q/b;
    plot(T_paso_d-Tcdg, Y_paso_d)
end

```

Registro, caracterización y modelado de las fuerzas verticales inducidas al andar.

```

i

%DATA2
T_i = DATA2.Data1_time_AI_5;
F_i = double(DATA2.Data1_AI_5-0.0947);
umMin = 0.001;
umMax = 0.08;
Nextra = 1;
Tmax = 1.2;
[Tini_i, Tfin_i] = buscaPasos(T_i, F_i, umMin, umMax, Nextra, Tmax);

%Calibración
F_i = F_i * f;
DATA2.Data1_AI_5 = DATA2.Data1_AI_5 * f;

%Definición de paso
paso = 1;
Tpaso_i = T_d(T_i >= Tini_i(paso) & T_i <= Tfin_i(paso));
Ypaso_i = F_d(T_i >= Tini_i(paso) & T_i <= Tfin_i(paso));
Tpaso_i = T_paso_i - T_paso_i(1);
Ypaso_i = Y_paso_i - Y_paso_i(1);

%Integral.
q = trapz(T_paso_i, Y_paso_i.*T_paso_i);
b = trapz(T_paso_i, Y_paso_i);
Tcdg = q/b;

%Graficar varios pasos
for i = 1:length(Tini_i)
    paso = i;
    Tpaso_i = DATA2.Data1_time_AI_5(DATA2.Data1_time_AI_5 >= Tini_i(paso) &
DATA2.Data1_time_AI_5 <= Tfin_i(paso));
    Ypaso_i = DATA2.Data1_AI_5(DATA2.Data1_time_AI_5 >= Tini_i(paso) &
DATA2.Data1_time_AI_5 <= Tfin_i(paso));
    Tpaso_i = T_paso_i - T_paso_i(1);
    Ypaso_i = Y_paso_i - Y_paso_i(1);
    q = trapz(T_paso_i, Y_paso_i.*T_paso_i);
    b = trapz(T_paso_i, Y_paso_i);
    Tcdg = q/b;
    plot(T_paso_i-Tcdg, Y_paso_i)
end
i
hold off
title('Un paso D-I')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Identificación estadística %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fourier de orden 4: 10 parámetros
% a0 a1 b1 a2 b2 a3 b3 a4 b4 w

%Fourier paso derecho
parFou4_d = zeros(length(Tini_d), 10);
gof_sse_d = zeros(length(Tini_d), 1);
gof_r2_d = zeros(length(Tini_d), 1);

```

```

%figure(2);
%hold on
for i = 1:length(Tini_d)
    paso = i;
    Tpaso_d = DATA1.Data1_time_AI_5(DATA1.Data1_time_AI_5 >= Tini_d(paso) &
DATA1.Data1_time_AI_5 <= Tfin_d(paso));
    Ypaso_d = DATA1.Data1_AI_5(DATA1.Data1_time_AI_5 >= Tini_d(paso) &
DATA1.Data1_time_AI_5 <= Tfin_d(paso));
    Tpaso_d = Tpaso_d - Tpaso_d(1);
    Ypaso_d = Ypaso_d - Ypaso_d(1);

    [res_d, gof_d] = fourier4(Tpaso_d,Ypaso_d);
    parFou4_d(i, :) = [res_d.a0, res_d.a1, res_d.b1, res_d.a2, res_d.b2,
res_d.a3, res_d.b3, res_d.a4, res_d.b4, res_d.w];
    gof_sse_d(i) = gof_d.sse;
    gof_r2_d(i) = gof_d.rsquare;
    %plot(res_d, Tpaso_d, Ypaso_d)

end

%Fourier paso dizquierdo
for i = 1:length(Tini_i)
    paso = i;
    Tpaso_i = DATA2.Data1_time_AI_5(DATA2.Data1_time_AI_5 >= Tini_i(paso) &
DATA2.Data1_time_AI_5 <= Tfin_i(paso));
    Ypaso_i = DATA2.Data1_AI_5(DATA2.Data1_time_AI_5 >= Tini_i(paso) &
DATA2.Data1_time_AI_5 <= Tfin_i(paso));
    Tpaso_i = Tpaso_i - Tpaso_i(1);
    Ypaso_i = Ypaso_i - Ypaso_i(1);

    [res_i, gof_i] = fourier4(Tpaso_i,Ypaso_i);
    parFou4_i(i, :) = [res_i.a0, res_i.a1, res_i.b1, res_i.a2, res_i.b2,
res_i.a3, res_i.b3, res_i.a4, res_i.b4, res_i.w];
    gof_sse_i(i) = gof_i.sse;
    gof_r2_i(i) = gof_i.rsquare;
    %plot(res_i, Tpaso_i, Ypaso_i)

end
hold off
title('Aproximación de un paso, Fourier 4 términos')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%Aproximación de fourier
figure(3)
hold on
plot( res_d, 'r')
plot( res_i, 'b')
title('Aproximación de Fourier 4 términos')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');
legend('Paso derecho','Paso izquierdo');

%Media de cada parametro de Fourier
media_d = mean(parFou4_d)
media_i = mean(parFou4_i)

%Desviación típica de cada parametro de Fourier

```

Registro, caracterización y modelado de las fuerzas verticales inducidas al andar.

```
desv_d = std(parFou4_d)
desv_i = std(parFou4_i)

%Parametros de Fourier pseudoaleatorios
%Distribución normal paso derecho
pd_a0_d = makedist ('Normal', 'mu', media_d(1), 'sigma', desv_d(1));
pd_a1_d = makedist ('Normal', 'mu', media_d(2), 'sigma', desv_d(2));
pd_b1_d = makedist ('Normal', 'mu', media_d(3), 'sigma', desv_d(3));
pd_a2_d = makedist ('Normal', 'mu', media_d(4), 'sigma', desv_d(4));
pd_b2_d = makedist ('Normal', 'mu', media_d(5), 'sigma', desv_d(5));
pd_a3_d = makedist ('Normal', 'mu', media_d(6), 'sigma', desv_d(6));
pd_b3_d = makedist ('Normal', 'mu', media_d(7), 'sigma', desv_d(7));
pd_a4_d = makedist ('Normal', 'mu', media_d(8), 'sigma', desv_d(8));
pd_b4_d = makedist ('Normal', 'mu', media_d(9), 'sigma', desv_d(9));
pd_w_d = makedist ('Normal', 'mu', media_d(10), 'sigma', desv_d(10));
%Generacion de numeros aleatorios paso derecho
rand_a0_d = random(pd_a0_d);
rand_a1_d = random(pd_a1_d);
rand_b1_d = random(pd_b1_d);
rand_a2_d = random(pd_a2_d);
rand_b2_d = random(pd_b2_d);
rand_a3_d = random(pd_a3_d);
rand_b3_d = random(pd_b3_d);
rand_a4_d = random(pd_a4_d);
rand_b4_d = random(pd_b4_d);
rand_w_d = random(pd_w_d);
rand_d = [rand_a0_d rand_a1_d rand_b1_d rand_a2_d rand_b2_d rand_a3_d
rand_b3_d rand_a4_d rand_b4_d rand_w_d]

%Distribución normal paso izquierdo
pd_a0_i = makedist ('Normal', 'mu', media_i(1), 'sigma', desv_i(1));
pd_a1_i = makedist ('Normal', 'mu', media_i(2), 'sigma', desv_i(2));
pd_b1_i = makedist ('Normal', 'mu', media_i(3), 'sigma', desv_i(3));
pd_a2_i = makedist ('Normal', 'mu', media_i(4), 'sigma', desv_i(4));
pd_b2_i = makedist ('Normal', 'mu', media_i(5), 'sigma', desv_i(5));
pd_a3_i = makedist ('Normal', 'mu', media_i(6), 'sigma', desv_i(6));
pd_b3_i = makedist ('Normal', 'mu', media_i(7), 'sigma', desv_i(7));
pd_a4_i = makedist ('Normal', 'mu', media_i(8), 'sigma', desv_i(8));
pd_b4_i = makedist ('Normal', 'mu', media_i(9), 'sigma', desv_i(9));
pd_w_i = makedist ('Normal', 'mu', media_i(10), 'sigma', desv_i(10));
%Generación de numeros aleatorios paso izquierdo
rand_a0_i = random(pd_a0_i);
rand_a1_i = random(pd_a1_i);
rand_b1_i = random(pd_b1_i);
rand_a2_i = random(pd_a2_i);
rand_b2_i = random(pd_b2_i);
rand_a3_i = random(pd_a3_i);
rand_b3_i = random(pd_b3_i);
rand_a4_i = random(pd_a4_i);
rand_b4_i = random(pd_b4_i);
rand_w_i = random(pd_w_i);
rand_i = [rand_a0_i rand_a1_i rand_b1_i rand_a2_i rand_b2_i rand_a3_i
rand_b3_i rand_a4_i rand_b4_i rand_w_i]
%Serie de Fourier
%Paso derecho
x = 1.6:0.01:5;
Fourier_d = rand_a0_d + rand_a1_d*cos(x*rand_w_d) +
rand_b1_d*sin(x*rand_w_d) + rand_a2_d*cos(2*x*rand_w_d) +
```

```

rand_b2_d*sin(2*x*rand_w_d) + rand_a3_d*cos(3*x*rand_w_d) +
rand_b3_d*sin(3*x*rand_w_d) + rand_a4_d*cos(4*x*rand_w_d) +
rand_b4_d*sin(4*x*rand_w_d);
%Paso izquierdo
Fourier_i = rand_a0_i + rand_a1_i*cos(x*rand_w_i) +
rand_b1_i*sin(x*rand_w_i) + rand_a2_i*cos(2*x*rand_w_i) +
rand_b2_i*sin(2*x*rand_w_i) + rand_a3_i*cos(3*x*rand_w_i) +
rand_b3_i*sin(3*x*rand_w_i) + rand_a4_i*cos(4*x*rand_w_i) +
rand_b4_i*sin(4*x*rand_w_i);
%Desplazamiento
pd_desp = makedist('Normal', 'mu', 0.75, 'sigma', 0.005);
%Representación gráfica de diez pasos, derecho izquierdo alternos
figure(4)
hold on

for i = 1:2:10
    paso = i;
    rand_d = [random(pd_a0_d) random(pd_a1_d) random(pd_b1_d)
random(pd_a2_d) random(pd_b2_d) random(pd_a3_d) random(pd_b3_d)
random(pd_a4_d) random(pd_b4_d) random(pd_w_d)];
    rand_i = [random(pd_a0_i) random(pd_a1_i) random(pd_b1_i)
random(pd_a2_i) random(pd_b2_i) random(pd_a3_i) random(pd_b3_i)
random(pd_a4_i) random(pd_b4_i) random(pd_w_i)];
    Fourier_d = rand_a0_d + rand_a1_d*cos(x*rand_w_d) +
rand_b1_d*sin(x*rand_w_d) + rand_a2_d*cos(2*x*rand_w_d) +
rand_b2_d*sin(2*x*rand_w_d) + rand_a3_d*cos(3*x*rand_w_d) +
rand_b3_d*sin(3*x*rand_w_d) + rand_a4_d*cos(4*x*rand_w_d) +
rand_b4_d*sin(4*x*rand_w_d);
    Fourier_i = rand_a0_i + rand_a1_i*cos(x*rand_w_i) +
rand_b1_i*sin(x*rand_w_i) + rand_a2_i*cos(2*x*rand_w_i) +
rand_b2_i*sin(2*x*rand_w_i) + rand_a3_i*cos(3*x*rand_w_i) +
rand_b3_i*sin(3*x*rand_w_i) + rand_a4_i*cos(4*x*rand_w_i) +
rand_b4_i*sin(4*x*rand_w_i);
    plot ( x*0.24+((i-1)*random(pd_desp)), Fourier_d, 'r',
x*0.24+(i*random(pd_desp)), Fourier_i, 'b')
end
hold off
title('Aproximación de un paseo, Fourier 4 términos')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');
legend('Paso derecho', 'Paso izquierdo');

```

Registro, caracterización y modelado de las fuerzas verticales inducidas al andar.

2. Procesamiento de datos de tres pasos

2.1. Función buscaPasos

```
function [Tini, Tfin] = buscaPasos(T, F, umMin, umMax, Nextra, Tmax)
% Función que busca los instantes inicial y final de los pasos dentro % de
% un registro donde hay muchos.
% Parámetros de entrada:
% T: vector de tiempos (s)
% F: vector de registro de fuerza (N)
% umMin: umbral mínimo para considerar un paso (N)
% umMax: umbral máximo al que deberían llegar todos los pasos (N)
% Nextra: número de puntos extra a considerar antes y después
% Tmax: máxima duración de un paso (s)

dT = T(2) - T(1);
Tini = [];
Tfin = [];
i = 1;
while(i < length(T))
    if(F(i) >= umMin)
        Ti = T(i);
        Tini = [Tini Ti-Nextra*dT];
        Tf = min(Ti + T(F(T > Ti) <= umMin));
        Tfin = [Tfin Tf+Nextra*dT];
        i = find(T == Tfin(end))+1;
    else
        i = i + 1;
    end
end

% Quito los pasos que no superan el umbralMax
for i = 1:length(Tini)
    if(max(F(T >= Tini(i) & T <= Tfin(i))) <= umMax)
        Tini(i) = NaN;
        Tfin(i) = NaN;
    end
end

Tini = Tini(~isnan(Tini));
Tfin = Tfin(~isnan(Tfin));

% Finalmente, quito los pasos de duración excesiva
for i = 1:length(Tini)
    if(Tfin(i) - Tini(i) > Tmax)
        Tini(i) = NaN;
        Tfin(i) = NaN;
    end
end

Tini = Tini(~isnan(Tini));
Tfin = Tfin(~isnan(Tfin));

end
```

2.2. Función fourier8

```
function [fitresult, gof] = fourier8(Tpaso, Ypaso)
%CREATEFIT(TPASO,YPASO)
% Create a fit.
%
% Data for 'untitled fit 1' fit:
%   X Input : Tpaso
%   Y Output: Ypaso
% Output:
%   fitresult : a fit object representing the fit.
%   gof : structure with goodness-of fit info.
%
% See also FIT, CFIT, SFIT.

% Auto-generated by MATLAB on 05-Jul-2019 23:35:20
[xData, yData] = prepareCurveData( Tpaso, Ypaso );

% Set up fittype and options.
ft = fittype( 'fourier8' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.Display = 'Off';
opts.Normalize = 'on';
opts.StartPoint = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.908056360050204];

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );

% Plot fit with data.
figure( 'Name', 'untitled fit 1' );
%h = plot( fitresult, xData, yData );
%legend( h, 'Ypaso vs. Tpaso', 'untitled fit 1', 'Location', 'NorthEast' );
% Label axes
xlabel Tpaso
ylabel Ypaso
grid on
```


2.3. Función spline

```
function [fitresult, god] = spline(Tpaso, Ypaso)
%CREATEFIT(TPASO,YPASO)
% Create a fit.
%
% Data for 'untitled fit 1' fit:
%   X Input : Tpaso
%   Y Output: Ypaso
% Output:
%   fitresult : a fit object representing the fit.
%   gof : structure with goodness-of fit info.
%
% See also FIT, CFIT, SFIT.

% Auto-generated by MATLAB on 09-Jul-2019 04:55:41
[xData, yData] = prepareCurveData( Tpaso, Ypaso );

% Set up fittype and options.
ft = fittype( 'smoothingspline' );
opts = fitoptions( 'Method', 'SmoothingSpline' );
opts.SmoothingParam = 0.99999;

% Fit model to data.
[fitresult, god] = fit( xData, yData, ft, opts );

% Plot fit with data.
figure( 'Name', 'untitled fit 1' );
%h = plot( fitresult, xData, yData, 'b' );
%legend( h, 'Data', 'Spline', 'Location', 'NorthEast' );
% Label axes
xlabel 'T (s)'
ylabel 'F (N)'
```

2.4. Código para pasos derecho, izquierdo, derecho

```
%Tratamiento de datos.
clear;

%Cargo datos experimentales
DATA1 = load('DID.mat');
DATA2 = load('DID-2.mat');

%DATA1
%Llamada a la función 'buscaPasos'.
T = DATA1.Data1_time_AI_10;
F = double(DATA1.Data1_AI_10-0.2);
umMin = 0.05;
umMax = 0.31;
Nextra = 1;
Tmax = 2;
[Tini, Tfin] = buscaPasos(T, F, umMin, umMax, Nextra, Tmax);

%Calibración
M = 509.6; %peso del sujeto en N
m = 0.2915; %media de datos estáticos
f = M/m; %factor de corrección
F = F * f;
DATA1.Data1_AI_10 = double(DATA1.Data1_AI_10 * f);

%Graficar DATA1 completo.
figure(1)
plot(T,F)
title('DID')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%Graficar tres pasos.
paso = 73;
figure(2)
Tpaso = T(T >= Tini(paso) & T <= Tfin(paso+1));
Ypaso = F(T >= Tini(paso) & T <= Tfin(paso+1));
Tpaso = Tpaso - Tpaso(1);
Ypaso = Ypaso - Ypaso(1);
plot(Tpaso, Ypaso)
title('Tres pasos DID')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%Integral.
q = trapz(Tpaso, Ypaso.*Tpaso);
b = trapz(Tpaso,Ypaso);
Tcdg = q/b;

%Graficar varios pasos.
figure(3); hold on
for i = 1:length(Tini)
    paso = i;
    Tpaso = DATA1.Data1_time_AI_10(DATA1.Data1_time_AI_10 >= Tini(paso) &
DATA1.Data1_time_AI_10 <= Tfin(paso+1));
    Ypaso = DATA1.Data1_AI_10(DATA1.Data1_time_AI_10 >= Tini(paso) &
DATA1.Data1_time_AI_10 <= Tfin(paso+1));
```

```

    Tpaso = Tpaso - Tpaso(1);
    Ypaso = Ypaso - Ypaso(1);
    q = trapz(Tpaso, Ypaso.*Tpaso);
    b = trapz(Tpaso, Ypaso);
    Tcdg = q/b;
    plot(Tpaso-Tcdg, Ypaso)
end
i

%DATA2
T = DATA2.Data1_time_AI_10;
F = double(DATA2.Data1_AI_10-0.2);
[Tini, Tfin] = buscaPasos(T, F, umMin, umMax, Nextra, Tmax);

%Calibración
F = F * f;
DATA2.Data1_AI_10 = double(DATA2.Data1_AI_10 * f);

for i = 1:length(Tini)
    paso = i;
    Tpaso = DATA2.Data1_time_AI_10(DATA2.Data1_time_AI_10 >= Tini(paso) &
DATA2.Data1_time_AI_10 <= Tfin(paso+1));
    Ypaso = DATA2.Data1_AI_10(DATA2.Data1_time_AI_10 >= Tini(paso) &
DATA2.Data1_time_AI_10 <= Tfin(paso+1));
    Tpaso = Tpaso - Tpaso(1);
    Ypaso = Ypaso - Ypaso(1);
    q = trapz(Tpaso, Ypaso.*Tpaso);
    b = trapz(Tpaso, Ypaso);
    Tcdg = q/b;
    plot(Tpaso-Tcdg, Ypaso)
end
i
hold off
title('Varios pasos DID')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Identificación estadística %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fourier de orden 4: 10 parámetros
% a0 a1 b1 a2 b2 a3 b3 a4 b4 a5 b5 a6 b6 a7 b7 a8 b8 w
parFou8 = zeros(length(Tini),18);
gof_sse = zeros(length(Tini),1);
gof_r2 = zeros(length(Tini),1);

%figure(4); hold on
for i = 1:length(Tini)
    paso = i;
    Tpaso = DATA2.Data1_time_AI_10(DATA2.Data1_time_AI_10 >= Tini(paso) &
DATA2.Data1_time_AI_10 <= Tfin(paso+1));
    Ypaso = DATA2.Data1_AI_10(DATA2.Data1_time_AI_10 >= Tini(paso) &
DATA2.Data1_time_AI_10 <= Tfin(paso+1));
    Tpaso = Tpaso - Tpaso(1);
    Ypaso = Ypaso - Ypaso(1);

```

Registro, caracterización y modelado de las fuerzas verticales inducidas al andar.

```
[res, gof] = fourier8(Tpaso,Ypaso);
parFou8(i, :) = [res.a0, res.a1, res.b1, res.a2, res.b2, res.a3,
res.b3, res.a4, res.b4, res.a5, res.b5, res.a6, res.b6, res.a7, res.b7,
res.a8, res.b8, res.w];
gof_sse(i) = gof.sse;
gof_r2(i) = gof.rsquare;
%plot(res, Tpaso, Ypaso)

end
%hold off
%title('Aproximación de Fourier 8 términos')
%xlabel('Tiempo (s)');
%ylabel('Fuerza (N)');

%Grafica tres pasos comparativa con la aproximación de fourier
figure(5)
h = plot( res, 'r', Tpaso, Ypaso, 'c*' )
title('Comparativa de la aproximación de Fourier 8 términos')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%Media de cada parametro de Fourier
media = mean(parFou8)

%Desviación típica de cada parametro de Fourier
desv = std(parFou8)
```

2.5. Código para pasos izquierdo, derecho izquierdo

```

%Tratamiento de datos.
clear;

%Carga datos experimentales
DATA1 = load('IDI-1.mat');
DATA2 = load('IDI-2.mat');

%DATA1
%Llamada a la función 'buscaPasos'.
T = DATA1.Data1_time_AI_10;
F = double(DATA1.Data1_AI_10-0.2);
umMin = 0.05;
umMax = 0.31;
Nextra = 1;
Tmax = 2;
[Tini, Tfin] = buscaPasos(T, F, umMin, umMax, Nextra, Tmax);

%Calibración
M = 509.6; %peso del sujeto en N
m = 0.2843; %media de datos estáticos
f = M/m; %factor de corrección
F = F * f;
DATA1.Data1_AI_10 = DATA1.Data1_AI_10 * f;

%Graficar DATA1 completo.
figure(1)
plot(T,F)
title('IDI')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%Graficar un paso.
paso = 1;
figure(2)
Tpaso = T(T >= Tini(paso) & T <= Tfin(paso+1));
Ypaso = F(T >= Tini(paso) & T <= Tfin(paso+1));
Tpaso = Tpasos - Tpasos(1);
Ypaso = Ypasos - Ypasos(1);
plot(Tpaso, Ypaso)
title('Tres pasos IDI')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%Integral.
q = trapz(Tpaso, Ypaso.*Tpaso);
b = trapz(Tpaso,Ypaso);
Tcdg = q/b;

%Graficar varios pasos.
figure(4); hold on
for i = 1:length(Tini)
    paso = i;
    Tpaso = DATA1.Data1_time_AI_10(DATA1.Data1_time_AI_10 >= Tini(paso) &
DATA1.Data1_time_AI_10 <= Tfin(paso+1));
    Ypaso = DATA1.Data1_AI_10(DATA1.Data1_time_AI_10 >= Tini(paso) &
DATA1.Data1_time_AI_10 <= Tfin(paso+1));

```

Registro, caracterización y modelado de las fuerzas verticales inducidas al andar.

```
    Tpaso = Tpaso - Tpaso(1);
    Ypaso = Ypaso - Ypaso(1);
    q = trapz(Tpaso, Ypaso.*Tpaso);
    b = trapz(Tpaso, Ypaso);
    Tcdg = q/b;
    plot(Tpaso-Tcdg, Ypaso)
end
i

%DATA2
T = DATA2.Data1_time_AI_10;
F = double(DATA2.Data1_AI_10-0.2);
[Tini, Tfin] = buscaPasos(T, F, umMin, umMax, Nextra, Tmax);

%Calibración
F = F * f;
DATA2.Data1_AI_10 = DATA2.Data1_AI_10 * f;

for i = 1:length(Tini)
    paso = i;
    Tpaso = DATA2.Data1_time_AI_10(DATA2.Data1_time_AI_10 >= Tini(paso) &
DATA2.Data1_time_AI_10 <= Tfin(paso+1));
    Ypaso = DATA2.Data1_AI_10(DATA2.Data1_time_AI_10 >= Tini(paso) &
DATA2.Data1_time_AI_10 <= Tfin(paso+1));
    Tpaso = Tpaso - Tpaso(1);
    Ypaso = Ypaso - Ypaso(1);
    q = trapz(Tpaso, Ypaso.*Tpaso);
    b = trapz(Tpaso, Ypaso);
    Tcdg = q/b;
    plot(Tpaso-Tcdg, Ypaso)
end
i
hold off
title('Varios pasos IDI')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Identificación estadística %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fourier de orden 4: 10 parámetros
% a0 a1 b1 a2 b2 a3 b3 a4 b4 a5 b5 a6 b6 a7 b7 a8 b8 w
parFou8 = zeros(length(Tini), 18);
gof_sse = zeros(length(Tini), 1);
gof_r2 = zeros(length(Tini), 1);

figure(4); hold on
for i = 1:length(Tini)
    paso = i;
    Tpaso = DATA2.Data1_time_AI_10(DATA2.Data1_time_AI_10 >= Tini(paso) &
DATA2.Data1_time_AI_10 <= Tfin(paso+1));
    Ypaso = DATA2.Data1_AI_10(DATA2.Data1_time_AI_10 >= Tini(paso) &
DATA2.Data1_time_AI_10 <= Tfin(paso+1));
    Tpaso = Tpaso - Tpaso(1);
    Ypaso = Ypaso - Ypaso(1);

    [res, gof] = fourier8(Tpaso, Ypaso);
```

```

        parFou8(i, :) = [res.a0, res.a1, res.b1, res.a2, res.b2, res.a3,
res.b3, res.a4, res.b4, res.a5, res.b5, res.a6, res.b6, res.a7, res.b7,
res.a8, res.b8, res.w];
        gof_sse(i) = gof.sse;
        gof_r2(i) = gof.rsquare;
        plot(res, Tpasso, Ypasso)

end
hold off
title('Aproximación de Fourier 8 términos')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)')

%Grafica tres pasos comparativa con la aproximación de fourier
figure(5)
h = plot( res, Tpasso, Ypasso )
title('Comparativa de la aproximación de Fourier 8 términos')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)')

%Media de cada parametro de Fourier
media = mean(parFou8)

%Desviación típica de cada parametro de Fourier
desv = std(parFou8)

```

2.6. Código AjusteSplineDID

```
%Tratamiento de datos.
clear;

%Cargo datos experimentales
DATA1 = load('DID.mat');

%DATA1
%Llamada a la función 'buscaPasos'.
T = DATA1.Data1_time_AI_10;
F = double(DATA1.Data1_AI_10-0.2);
umMin = 0.05;
umMax = 0.31;
Nextra = 1;
Tmax = 2;
[Tini, Tfin] = buscaPasos(T, F, umMin, umMax, Nextra, Tmax);

%Calibración
M = 509.6; %peso del sujeto en N
m = 0.2915; %media de datos estáticos
f = M/m; %factor de corrección
F = F * f;
DATA1.Data1_AI_10 = double(DATA1.Data1_AI_10 * f);

%Graficar tres pasos.
paso = 73;
figure(1)
Tpaso = T(T >= Tini(paso) & T <= Tfin(paso+1));
Ypaso = F(T >= Tini(paso) & T <= Tfin(paso+1));
Tpaso = Tpasos - Tpasos(1);
Ypaso = Ypasos - Ypasos(1);
plot(Tpaso, Ypaso)
title('Tres pasos DID')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%Integral.
q = trapz(Tpaso, Ypaso.*Tpaso);
b = trapz(Tpaso, Ypaso);
Tcdg = q/b;

%Graficar varios pasos.
figure(2); hold on
for i = 1:length(Tini)
    paso = i;
    Tpaso = DATA1.Data1_time_AI_10(DATA1.Data1_time_AI_10 >= Tini(paso) &
DATA1.Data1_time_AI_10 <= Tfin(paso+1));
    Ypaso = DATA1.Data1_AI_10(DATA1.Data1_time_AI_10 >= Tini(paso) &
DATA1.Data1_time_AI_10 <= Tfin(paso+1));
    Tpaso = Tpaso - Tpaso(1);
    Ypaso = Ypaso - Ypaso(1);
    q = trapz(Tpaso, Ypaso.*Tpaso);
    b = trapz(Tpaso, Ypaso);
    Tcdg = q/b;
    plot(Tpaso-Tcdg, Ypaso)
end
i
```



```

hold off
title('Varios pasos DID')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Identificación estadística %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SPLINE
god_sse = zeros(length(Tini),1);
god_r2 = zeros(length(Tini),1);

figure(3); hold on
for i = 1:length(Tini)
    paso = i;
    Tpaso = DATA1.Data1_time_AI_10(DATA1.Data1_time_AI_10 >= Tini(paso) &
DATA1.Data1_time_AI_10 <= Tfin(paso+1));
    Ypaso = DATA1.Data1_AI_10(DATA1.Data1_time_AI_10 >= Tini(paso) &
DATA1.Data1_time_AI_10 <= Tfin(paso+1));
    Tpaso = Tpaso - Tpaso(1);
    Ypaso = Ypaso - Ypaso(1);

    [fitresult, god] = spline(Tpaso,Ypaso);
    god_sse(i) = god.sse;
    god_r2(i) = god.rsquare;
    plot(fitresult, Tpaso, Ypaso)

end
hold off
title('Aproximación mediante Spline')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

%Grafica tres pasos comparativa con la aproximación mediante Spline
figure(4)
plot( fitresult, 'r', Tpaso, Ypaso, 'b' )
title('Comparativa de la aproximación mediante Spline')
xlabel('Tiempo (s)');
ylabel('Fuerza (N)');

```