

TREBALL FINAL DE MÀSTER

Màster Universitari en Enginyeria Industrial

CARACTERITZACIÓ DE LES PROPIETATS MECÀNIQUES D'UN ELASTÒMER UTILITZAT EN LA FABRICACIÓ PER FILAMENT FOS



Annexes

Autora: B rbara Adrover Monserrat

Director: Jordi Romeu Garbi

Codirector: Jos  Antonio Travieso Rodr guez

Convocat ria: Novembre 2020

Índex

| | | |
|------|--------------------------------------|----|
| I. | Taula característica impressora | 3 |
| ii. | Taula característica màquina tracció | 5 |
| iii. | Fitxa tècnica flexfill tpu 98a | 7 |
| iv. | Fitxa tècnica flexfill tpe 96a | 8 |
| v. | Plànols | 9 |
| vi. | Rutines matlab | 12 |

I. Taula característica impressora

La impressora utilitzada per a la fabricació de mostres amb la tecnologia FFF és la Ender-3 Pro. Les taules TXX, TXX i TXX detallen les característiques tècniques d'aquesta impressora.

Taula 1: Característiques tècniques Ender-3 Pro: Propietats.

Ender-3 Pro FDM 3D Propietats impressora

Tecnologia de modelat: FDM (Fused Deposition Modeling)

Dimensions zona impressió: 220*220*250mm

Velocitat impressió: ≤ 180 mm/s, normal 30-60mm/s

Filament: 1.75mm PLA, ABS, Wood, TPU, Gradient color, carbon fiber, etc.

Mode de treball: Online o targeta SD (offline)

Format arxius: STL, OBJ, AMF

Software "Slice": Cura, Repetier-Host, Simplify3D

Taula 2: Característiques tècniques Ender-3 Pro: Hardware.

Ender-3 pro FDM 3D Hardware

Dimensions màquina: 440-440-465mm

Dimensions paquet: 595*495*165mm

N.W: 6.9KG G.W: 8.9KG

Entrada d'alimentació: AC100-120V/6.8A 200-240V/3.4A/ 50/60Hz

Sortida: DC 24 V 270W

Taula 3: Característiques tècniques Ender-3 Pro: Extrusor

Ender-3 Pro FDM 3D Hardware Extrusor

Diàmetre extrusor: Estàndard 0.4mm, pot treballar amb 0.2, 0.3 mm

Precisió d'impressió: $\pm 0.1\text{mm}$

Temperatura llit.: $\leq 110^{\circ}\text{C}$

II. Taula característica màquina tracció

A continuació es detalla la informació de la màquina universal utilitzada per als assajos a tracció. S'utilitza una *Universal Testing Machine* estàndard del proveïdor Zwick/Roell. La figura FXX mostra la informació tècnica a temperatura ambient. La figura FXX mostra les dimensions de la màquina. I finalment, l'esquema de la figura FXX mostra les referències geomètriques de les que fa referència la figura FXX.

| Description | Value |
|---|---|
| Load frame | |
| Paint finish | RAL 7011 Iron gray and RAL7038 agate gray |
| Ambient temperature | +10 ... +35 °C |
| Humidity (non-condensing) | 20 ... 90 % |
| Drive | |
| Motor | AC servo motor with concentrated windings, Hlperface® motor feedback system |
| motor holding brake | yes |
| Control, set-value specification | digital (real-time Ethernet, EtherCAT®) |
| Controller/cycle time | adaptive / 1000 Hz |
| Repeat positioning accuracy on crosshead | ± 1.0 µm |
| Measurement and control electronics | |
| Number of slots available for measurement and control module | 5 synchronized module bus slots, 1 synchronized PCIe slot ¹⁾ |
| Force measurement | Class 0.5 from 1% of Fmax / Class 1 from 0.2 % of Fmax, as per EN ISO 7500-1, ASTM E4, JIS B 7721 |
| Calculated resolution (e.g. load cell in tensile/compression direction) | 24 bit |
| Measured-value sampling-rate, Internal | 400 kHz |
| Measured-value transmission rate to PC | 500 (optionally 2000) Hz |
| Zero-point correction | automatic, at start of measurement |
| Measurement signal runtime correction for all channels | yes |
| Interface to PC | Ethernet |
| Eco mode? | yes, power unit automatically switched off (time adjustable) |
| CE compliance? | yes, as per Machinery Directive 2006/42EC |
| Electrical supply data | |
| Mains supply frequency | 50/60 Hz |
| Electrical supply | 400 +/- 10 % V, 3Ph/PE |
| Power consumption | 24.5 kVA |

¹⁾ A high-quality DCSC measurement module for one load cell is included in delivery (occupies one module bus slot).

Options¹⁾

| Description |
|--|
| Display-equipped remote control for testControl II |
| Measured-value transmission-rate increase from standard 500 Hz to 2000Hz |
| Add-on table unit for electronics cabinet, with CPU-holder |
| ZwickRoell extensometer attachment (see catalog) |
| Front/rear safety device |
| Lead-screw guard |
| Slide test-area |

Figura 1: Informació tècnica de la màquina del proveïdor Zwick/Roell.

| Type Item Number | Standard 1015376 | H1-500 1021338 | H1+500 1021335 | H1+1000 1021339 | B1+400 1021340 | |
|---|---------------------|-------------------|-------------------|--------------------|-------------------|------------------------|
| Load frame | | | | | | |
| Test force F_N in tens./comp. direction | 1200 | 1200 | 1200 | 1200 | 1200 | kN |
| No. of guide columns | 4 | 4 | 4 | 4 | 4 | |
| No. of drive screws | 2 | 2 | 2 | 2 | 2 | |
| Stiffness of load frame at H2 = 1000 mm | 1972 | 1972 | 1972 | 1972 | 1034 | kN/ mm |
| Height - H1 | 3635 | 3135 | 4135 | 4635 | 3635 | mm |
| Width - B1 | 1470 | 1470 | 1470 | 1470 | 1870 | mm |
| Depth - T1 | 1063 | 1063 | 1063 | 1063 | 1063 | mm |
| Test-area width - B2 | 830 | 830 | 830 | 830 | 1230 | mm |
| Test-area height - H2 | 2340 | 1840 | 2840 | 3340 | 2340 | mm |
| Test stroke - H3 (with hydraulic grips) | 1080 | 580 | 1580 | 2080 | 1080 | mm |
| Weight without attachments | 6700 | 6200 | 7000 | 7200 | 7600 | kg |
| with hydraulic grips | 8200 | 7600 | 8400 | 8600 | 9000 | kg |
| Specific floor loading (with hydraulic grips) | 2.5 | 2.3 | 2.5 | 2.4 | 2.7 | kg/ cm ² |
| Noise level at maximum test-speed | <68 | <68 | <68 | <68 | <68 | dBA |
| Drive | | | | | | |
| Crosshead speed | 0.00005 to 400 | 0.00005 to 400 | 0.00005 to 400 | 0.00005 to 400 | 0.00005 to 400 | mm/ min |
| Increased crosshead return speed (at reduced force) | 550 | 550 | 550 | 550 | 550 | mm/ min |
| Displacement resolution of drive | 0.000254 | 0.000254 | 0.000254 | 0.000254 | 0.000254 | µm |

Figura 2: Dades físiques de la màquina utilitzada.

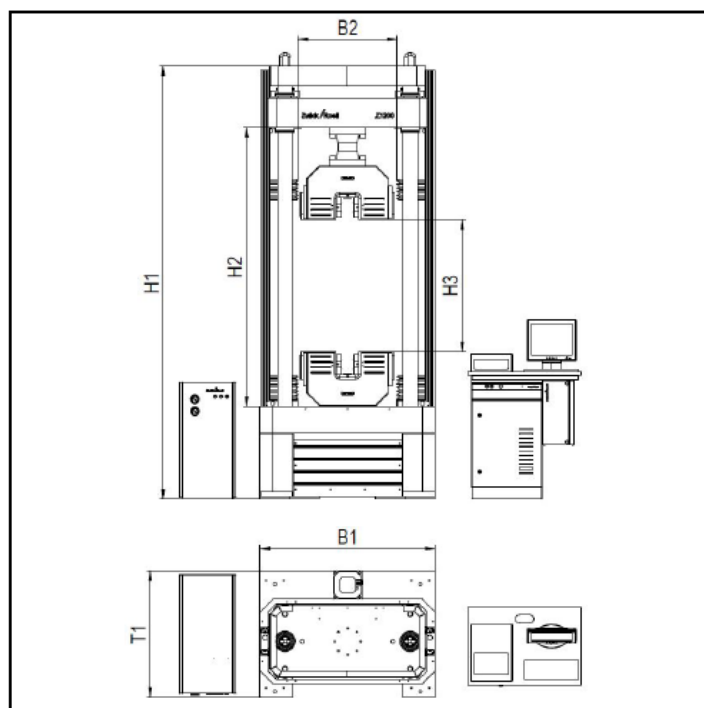


Figura 3: Plànol de la màquina del proveïdor Zwick/Roell.

III. Fitxa tècnica Flexfill TPU 98A

La fitxa tècnica del material s'ha obtingut de la empresa Filamentum. Aquesta mostra els paràmetres recomanables d'impressió.

| Physical properties | Typical Value | Test Method | Test Condition |
|---|--------------------------------------|---|--------------------------------|
| Material density | 1,23 g/cm ³ | ISO 1183-1 | |
| Diameter tolerance | ± 0,10 mm | | |
| Weight | 500 g of filament (+ 250 g spool) | | |
| Mechanical properties | Typical Value | Test Method | Test Condition |
| Tensile strength | 53,7 MPa | DIN 53504 | at break, 200 mm/min |
| Tensile stress | 12,1 MPa | DIN 53504 | 10% elongation, 200 mm/min |
| | 22,1 MPa | DIN 53504 | 50% elongation, 200 mm/min |
| | 28,4 MPa | DIN 53504 | 100% elongation, 200 mm/min |
| | 37,8 MPa | DIN 53504 | 300% elongation, 200 mm/min |
| Elongation at break | 318 % | DIN 53504 | 200 mm/min |
| Hardness | 98 Shore A | ISO 7619-1 | |
| | 60 Shore D | ISO 7619-1 | |
| Tear strength | 170 kN/m | ISO 34-1 | 500 mm/min |
| Abrasion | 23 mm ³ | ISO 4649 | method A |
| Chemical properties | Typical Value | Test Method | Test Condition |
| Polymer base | polyurethane | | |
| Oils, greases and ozone resistance | good | | 25 °C |
| Water, acetone, acids, alkalis, alcohols and car fluids resistance | bad | | 25 °C |
| Printing properties | Recommended | Notes | |
| Print temperature | 220-240 °C | Recommended settings! It may differ according to the printer and the object. Try your own optimization before printing. | |
| Hot pad | 50-60 °C | | |
| Bed adhesive | Magigoo | For easy removing of the object. | |
| Speed of printing | 20-30 mm/s | | |
| Fan speed | 70-100 % | | |

IV. Fitxa tècnica Flexfill TPE 96A

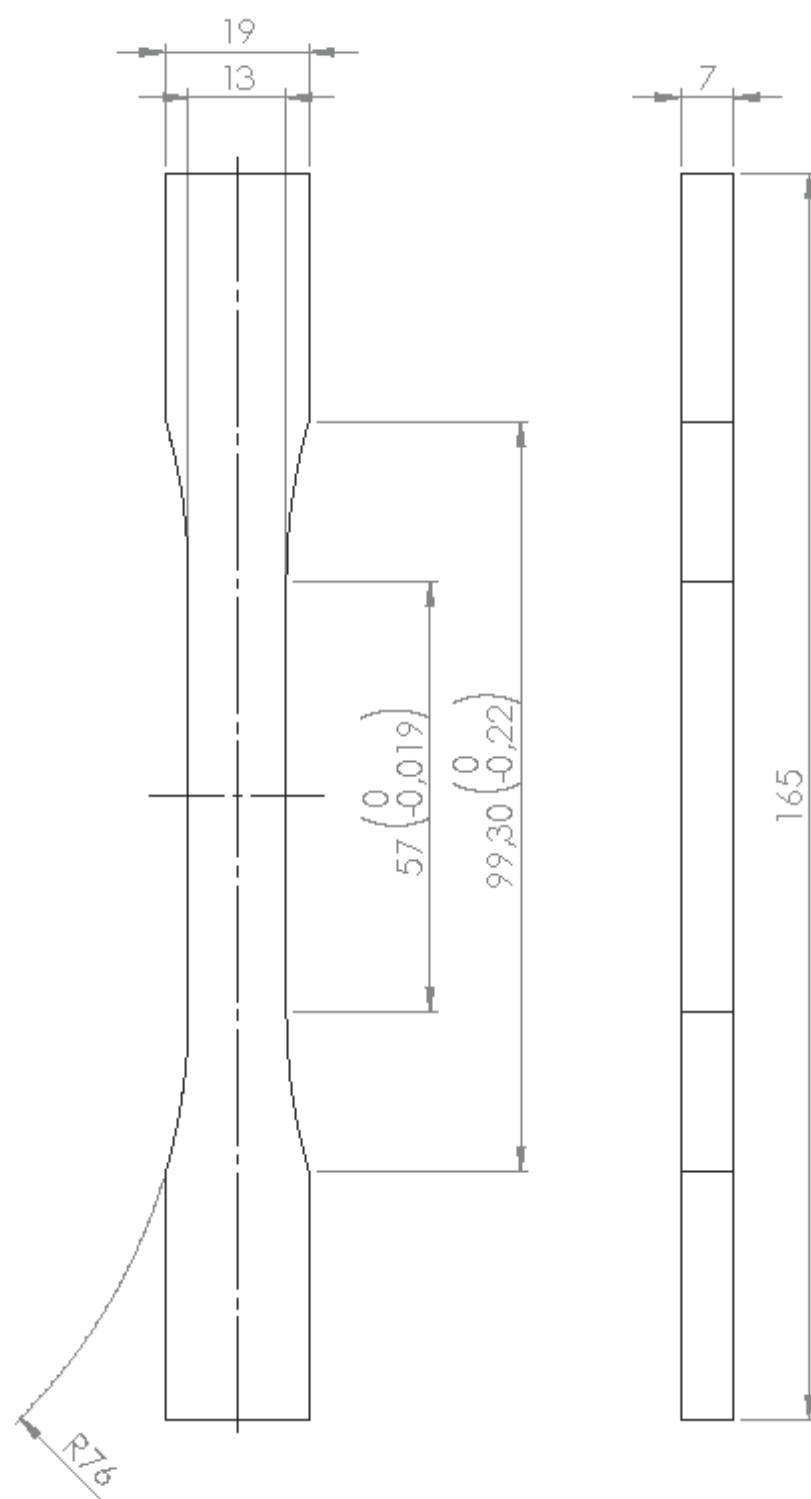
La fitxa tècnica del material s'ha obtingut de la empresa Filamentum. Aquesta mostra els paràmetres recomanables d'impressió.

| Physical Properties | Typical Value | Test Method | Test Condition |
|---|--------------------------------------|---|----------------|
| Material density | 1,15 g/cm ³ | ISO 1183 | 20 °C |
| Melt flow index | > 25 g/10 min | laboratory method | 230 °C, 5 kg |
| Diameter tolerance | ± 0,10 mm | | |
| Weight | 500 g of filament (+ 250 g spool) | | |
| Mechanical properties | Typical Value | Test Method | Test Condition |
| Tensile strength | > 5 MPa | laboratory method | at break |
| Elongation at break | > 150 % | laboratory method | at break |
| Hardness | 95 Shore A | laboratory method | |
| | 40 Shore D | laboratory method | |
| Chemical properties | Typical Value | Test Method | Test Condition |
| Polymer base | polyolefin | | |
| Resistance against water, acetone, acids, alkalis, alcohols | good | | 25 °C |
| Resistance against oils, greases, car fluids, ozone | bad | | 25 °C |
| Printing properties | Recommended | Notes | |
| Print temperature | 225-245 °C | Recommended settings! It may differ according to the printer and the object. Try your own settings before printing. | |
| Hot pad | 50-60 °C | | |
| Bed adhesive | Magigoo PP, 3DLac | Use brim for better bed adhesion. | |
| Type of bed | PEI, mirror/glass | | |
| Speed of printing | 15-25 mm/s | | |

V. Plànols

A aquest apartat es presenten els plànols dissenyats:

- Provetes per a tracció segons la norma ASTM D368.
- Adaptador per assajos a tracció per cel·la de 2kN.



**Provetes per a tracció segons
la norma ASTM D368**

DIBUIXAT: Bàrbara Adrover

PLÀNOL 1/2

REVISAT: J. Antonio Travieso

DIN-A4

DATA: 17/04/2020

ESCALA: 1:1

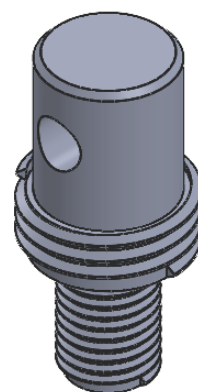
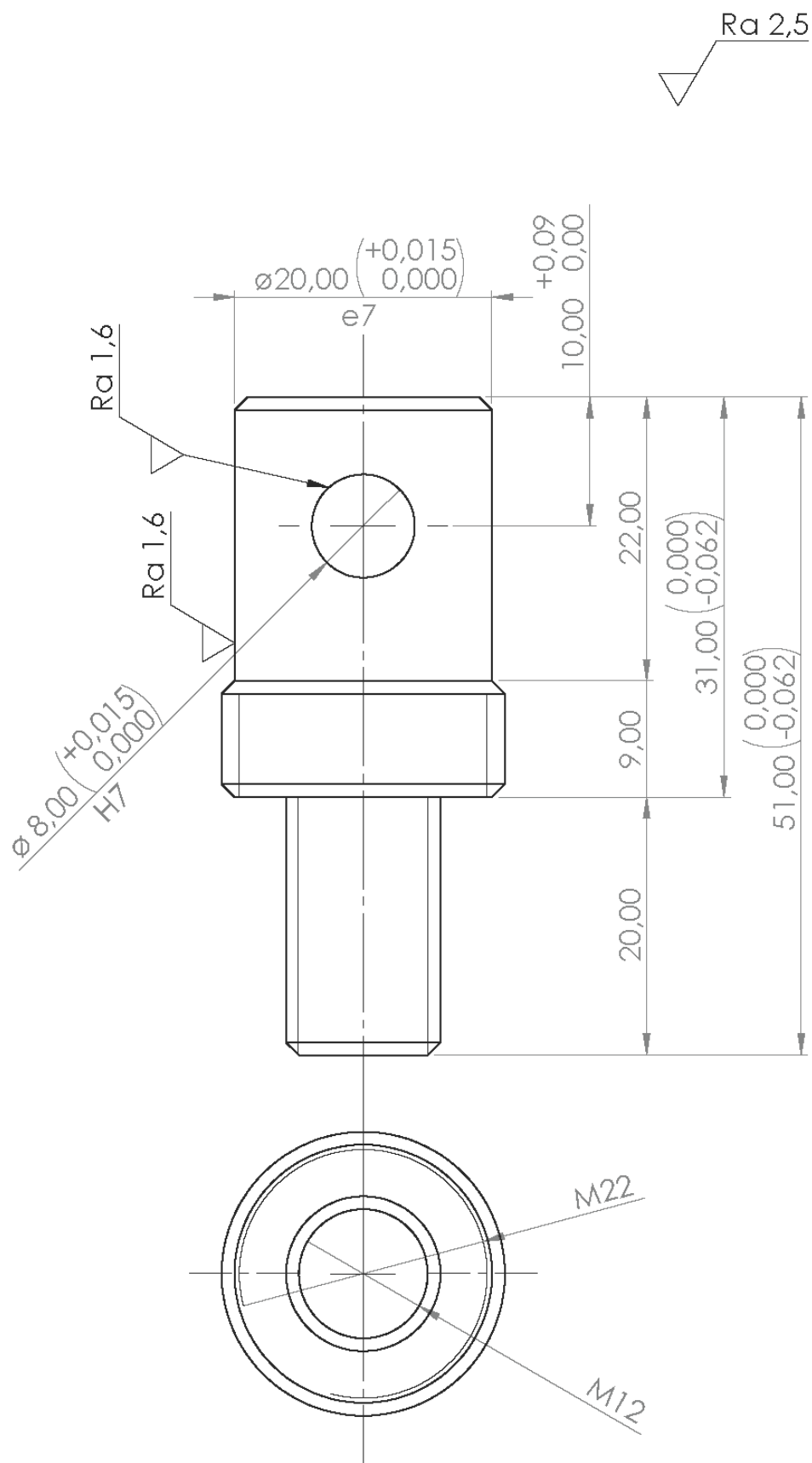


**UNIVERSITAT POLITÈCNICA
DE CATALUNYA**
BARCELONATECH

MATERIAL: Flexfill

QUANTITAT: 16 unitats





Escala: 1:1

**Adaptador assajos a tracció per
cèlula de 2kN**



**UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH**

DIBUIXAT: Bàrbara Adrover Monserrat

REVISAT: Jordi Llumà

DATA: 15/09/2020

MATERIAL: Alumini

QUANTITAT: 1 unitat

PLÀNOL: 2/2

DIN-A4

ESCALA: 2:1



VI. Rutines Matlab

A continuació es presenta les rutines utilitzades. Aquestes creen els mallats de punts, on s'estudia el moviment de les mostres, generen l'arxiu de deformacions en funció de com es mou la mostra i finalment generen la corba desitjada.

```
function sincronitza_traccio(Dades10,V_freq)

%load data in case you did not load it into workspace yet
if exist('Dades10','var')==0
    metro=dir('*.asc');
    if length(metro)==1
        Dades10name=metro(1).name;
    else
        [Dades10name,PathDades10] = uigetfile('*.asc','Open CATMAN
file');
        if Dades10name==0
            disp('You did not select a file!')
            return
        end
        cd(PathDades10);
    end
    Dades10=llegeix_catman(Dades10name); % Ja inverteix la Força
end

%define the size of the data set
N_forca=size(Dades10);
N_forca=N_forca(1);

% Converteix la matriu en vectors
F_temps=Dades10(1:N_forca,1);
Forca=Dades10(1:N_forca,2);
Voltatge=Dades10(1:N_forca,3);

%% Busca punt de sincronia
%
% Filtra Voltatge
pas=5; % Selecciona l'interval de filtratge com 2*pas+1
for i=(1+pas):(N_forca-pas)
    Voltatge(i)=median(Dades10((i-pas):(i+pas),3));
end
Voltatge(:)=Voltatge(:)-min(Voltatge(:)); %Corregeix voltatge amb
offset

% Busca esglao en Voltatge
[N_F_inici_sincro,N_F_final_sincro]=busca_esglao(Voltatge);

%% Llegeix imatge i calcula la intensitat
cd('Procesat video'); % Canvia a directori de processat
kk=dir('*.tif'); % obté el llistat d'imatges
N_video=size(kk);N_video=N_video(1); % Determina el nombre
d'imatges
for i=1:N_video
    filenamelist(i,:)=string(kk(i).name);
end
clear kk; %Neteja la matriu de noms
filenamelist=char(sort(filenamelist).');
```

```

% Pregunta la freqüència de video si no s'entra com a parametre, ni
hi ha fitxer -video.txt.
Dades_video=strcat('../',Dades10name(1:length(Dades10name)-4),'-
video.txt');
if exist(Dades_video,'file') % Hi ha fitxer que descriu video. Llegeix
dades.
    file10 = fopen(Dades_video,'r','ieee-be');
% Obre arxiu auxiliar de video
    textscan(file10, '%s', 1, 'Delimiter', '\t', 'ReturnOnError',
false); % Salta 1 línia
    V_freq=fscanf(file10,'%f',1);
% Llegeix freqüència de video
    fclose(file10);
% Tanca arxiu
else % No hi ha fitxer. Des de dades.
    if exist('V_freq','var')==0 % La freqüència no s'ha posat com a
parametre.
        answer = inputdlg({'FPS ?'},'Freq. video',1,{'59.94006'});
% Demana freq, suggereix la del video HD
        V_freq = str2double(cell2mat(answer(1,1)));
        clear answer;
    end
    mida_foto=size(imread(convertStringsToChars(filenamelist(1,:))));
% Llegeix mida de les imatges.
    Imatge_X=mida_foto(2); % Determina amplada imatge
    Imatge_Y=mida_foto(1); % Determina alçada imatge
    sortida=fopen(Dades_video,'w+'); % Obre arxiu auxiliar de video
    fprintf(sortida,'%s \n%f \n','FPS:',V_freq); % Des de freq
    fprintf(sortida,'%s \n%i \n','Pixels horitzontal:',Imatge_X); %
Des de amplada imatge
    fprintf(sortida,'%s \n%i \n','Pixels verticals:',Imatge_Y); % Des de
alçada imatge
    fclose(sortida); % Tanca arxiu
end
% Crea la base de temps del video
Video_temps=zeros(N_video,1);
for i=1:N_video
    Video_temps(i)=(i-1)/V_freq;
end

% Busca esglai en Intensitat de video
% Crea perfil d'intensitats de les 10 primeres imatges
intensitat=zeros(N_video,1);
for i=1:10
    intensitat(i)=
sum(sum(sum(imread(convertStringsToChars(filenamelist(i,:))))));
%calcula la intensitat de la imatge i
end
ref=median(intensitat(1:10))/2; % Estableix llindar a la meitat de
la mediana
int_incr=5; % Defineix el pas d'acceleració de búsqueda
while intensitat(i)>ref % Busca caiguda ràpida
    i=i+int_incr;
    intensitat(i)=
sum(sum(sum(imread(convertStringsToChars(filenamelist(i,:))))));
%calcula la intensitat de la imatge i
end
i=i-int_incr; % Retrocedeix int_incr fotogrames
while intensitat(i)>ref % Afina búsqueda de la caiguda
    i=i+1;

```

```

    intensitat(i)=
    sum(sum(sum(imread(convertStringsToChars(filenameelist(i,:))))));
    %calcula la intensitat de la imatge i
end
Video_inici_sincro=i-1; % Defineix el darrer fotograma il·luminat
i=i+5; % Avançada 5 dades (per evitar soroll)
intensitat(i)=
sum(sum(sum(imread(convertStringsToChars(filenameelist(i,:))))));
%calcula la intensitat de la imatge i
while intensitat(i)<ref %Busca pujada rÃ pid
    i=i+int_incr;
    intensitat(i)=
sum(sum(sum(imread(convertStringsToChars(filenameelist(i,:))))));
%calcula la intensitat de la imatge i
end
i=i-int_incr; % Retrocedeix int_incr fotogrames
while intensitat(i)<ref % Afina busqueda de la pujada
    i=i+1;
    intensitat(i)=
sum(sum(sum(imread(convertStringsToChars(filenameelist(i,:))))));
%calcula la intensitat de la imatge i
end
Video_final_sincro=i; % Defineix el primer fotograma il·luminat de
nou

% Calcula defasatge de temps.
delta_temps=((F_temps(N_F_inici_sincro)+F_temps(N_F_final_sincro))-
(Video_temps(Video_inici_sincro)+Video_temps(Video_final_sincro)))/2;

%% Busca inici de tracciÃ
%
Forca_max=max(Forca);
Forca_min=Forca_max;
N_Forca_min=N_F_inici_sincro;
%Busca mÃ xim i minim de forca.
for i=N_F_inici_sincro:N_forca
    if Forca(i)==Forca_max
        N_Forca_max=i;
        break
    end
    if Forca(i)<Forca_min
        N_Forca_min=i;
        Forca_min=Forca(i);
    end
end
%Busca mÃ-nim de forÃsa, abans que el mÃ xim
rao=50; %estableix el rang on es busca l'inici de cÃ rrega.
Forca_limit=Forca_min+(Forca_max-Forca_min)/rao;
% Busca inici de tracciÃ com la diferÃncia mÃ xima entre la derivada
per la dreta i l'esquerra. Agafa kk_pas+1 punts per calcular el
pendent.
kk_pas=19;
i=N_Forca_min+kk_pas;
N_F_inici=i;
F_increment=0;
%kk=0;
while Forca(i)<Forca_limit
    kk_mes=polyfit(F_temps(i:i+kk_pas),Forca(i:i+kk_pas),1);
    kk_menys=max(polyfit(F_temps(i-kk_pas:i),Forca(i-kk_pas:i),1),0);
    kk=kk_mes(1)-kk_menys(1);

```

```

        if kk>F_increment
            F_increment=kk;
            N_F_inici=i;
        end
        i=i+1;
    end
    N_V_inici=round((F_temps(N_F_inici)-delta_temps)*V_freq+1);
    clear kk kk_mes kk_menys F_increment kk_pas rao;
    %% Busca ruptura
    % Busca la màxima caiguda de força
    N_F_ruptura=N_Forca_max;
    delta_F=Forca(N_F_ruptura)-Forca(N_F_ruptura+1);
    for i=N_Forca_max+1:N_forca-1
        if Forca(i)<Forca_min
            break
        end
        kk=Forca(i)-Forca(i+1);
        if kk>delta_F
            delta_F=kk;
            N_F_ruptura=i;
        end
    end
    clear delta_F kk;
    % Busca darrer punt per sobre de la força estable del final
    Forca_inf=median(Forca(N_forca-10:N_forca)); % Defineix valor estable
    com mediana dels darrers 10 punts.
    N_F_ruptura_total=N_forca-10;
    for i=N_F_ruptura:N_forca-10
        if Forca(i)<=Forca_inf
            N_F_ruptura_total=i;
            break
        end
    end
    end

    % Busca el fotograma de ruptura.
    N_V_ruptura=round((F_temps(N_F_ruptura)-delta_temps)*V_freq+1)-5; %
    Defineix primer fotograma a buscar
    N_V_ruptura_total=min(N_video,round((F_temps(N_F_ruptura_total)-
    delta_temps)*V_freq+1)+1); % Defineix darrer fotograma a buscar
    if N_V_ruptura>N_V_ruptura_total-10
        N_V_ruptura=N_V_ruptura_total-10;
        kkkk='Video mes curt que dades'
    end
    foto=zeros(N_V_ruptura_total-N_V_ruptura+1,1); %Crea la matriu on
    guardar resultats
    imatge0=imread(convertStringsToChars(filenamelist(N_V_ruptura,:))); %
    Llegeix la imatge N_V_ruptura

    % Busca la màxima diferència de fotogrames.
    for i=N_V_ruptura+1:N_V_ruptura_total
        imatge1=imread(convertStringsToChars(filenamelist(i,:))); %
    Llegeix la imatge i
        kk=imatge1-imatge0;
        foto(i-N_V_ruptura)=sum(sum(sum(sum(bsxfun(@times, kk, kk)))));
        imatge0=imatge1;
    end

    % Defineix el punt d'esglai
    kk=0.8*max(foto);
    for i=N_V_ruptura+1:N_V_ruptura_total
        if foto(i-N_V_ruptura)>kk

```

```

        N_V_ruptura=i-1;
        break
    end
end

% Ajusta manualment els fotogrames
N_V_ruptura=ajusta_trencament(N_V_ruptura,filenamelist);
clear foto imatge0 imatge1 kk;
%% Genera i desa llistat d'imatges
filenamelist=filenamelist(N_V_inici:N_V_ruptura,:); % Genera el nom de
la primera imatge
save('filenamelist.mat','filenamelist');

%% Desa dades
cd ('..');
Dades10name=strcat(Dades10name(1:length(Dades10name)-4),'-
sincro.txt');
sortida=fopen(Dades10name,'w+');
fprintf(sortida,'%s \n','Dada de sincronisme');
fprintf(sortida,'%s \n%i \n','Inici video',Video_inici_sincro);
fprintf(sortida,'%s \n%i \n','Final video',Video_final_sincro);
fprintf(sortida,'%s \n%i \n','Inici forÀça',N_F_inici_sincro);
fprintf(sortida,'%s \n%i \n','Final forÀça',N_F_final_sincro);
fprintf(sortida,'%s \n%f \n\n','Delta de temps(s)',delta_temps);
fprintf(sortida,'%s \n','Dada d'inici assaigs');
fprintf(sortida,'%s \n%i \n','Inici video',N_V_inici);
fprintf(sortida,'%s \n%i \n\n','Inici forÀça',N_F_inici);
fprintf(sortida,'%s \n','Dada de ruptura');
fprintf(sortida,'%s \n%i \n','Ruptura video',N_V_ruptura);
fprintf(sortida,'%s \n%i \n\n','Ruptura forÀça',N_F_ruptura);
fclose(sortida);
close all
end

function [N_V_ruptura] = ajusta_trencament(N_V_ruptura,filenamelist)
    scrsz = get(0,'ScreenSize');
    dibuix=figure('Position',[scrsz(3)/10 scrsz(4)/10 (8*scrsz(3)/10
(8*scrsz(4)/10)]);
    tria=0;
    %uicontrol('Style','pushbutton','String','<','Position',[20 20
50 20],'Callback', @decrementa_fotograma);
    %uicontrol('Style','pushbutton','String','>','Position',
[(8*scrsz(3)/10)-70 20 50 20],'Callback', @incrementa_fotograma);
    while tria~=2
        dibuixa_trencament(N_V_ruptura,filenamelist);
        tria=menu('Fotograma','+', 'OK', '-');
        if tria==1
            N_V_ruptura=N_V_ruptura+1;
        end
        if tria==3
            N_V_ruptura=N_V_ruptura-1;
        end
        if tria==2
            close all;
            return
        end
    end
    close dibuix
end

```



```
function dibuixa_trencament(N_V_ruptura,filenamelist)

subplot(2,2,3),imshow(imread(convertStringsToChars(filenamelist(N_V_ruptura-1,:)))); % Dibuixa la imatge anterior
title('Fotograma anterior (telca -)'); % Posa nom i text

subplot(2,2,1:2),imshow(imread(convertStringsToChars(filenamelist(N_V_ruptura,:)))); % Dibuixa la darrera imatge sencera
title(['Darrer fotograma no fracturat ',num2str(N_V_ruptura),' (telca RETORN)']) % Posa nom i text

subplot(2,2,4),imshow(imread(convertStringsToChars(filenamelist(N_V_ruptura+1,:)))); % Dibuixa la primera imatge trencada
title('Primer fotograma fracturat (telca +)'); % Posa nom i text
text
end

function [Vinici,Vfinal]=busca_esglao(Volt)
ref=median(Volt(1:10))/2; % Estableix llindar a la meitat de la mediana
i=1;
while Volt(i)>ref % Busca caiguda
i=i+1;
end
Vinici=i-1;
i=i+5; % Avançsa 5 dades (per evitar soroll)
while Volt(i)<ref %Busca pujada
i=i+1;
end
Vfinal=i;
End

function autogrid(increment)
%Defineix increment per defecte com a 1500
if exist('increment','var')==0
increment=1500;
end

% Llegeix llistat de fotogrames
cd('Procesat video'); % Va a procesat Video
kkk=load('filenamelist'); % file with the list of
filenames to be processed
filenamelist=kkk.filenamelist;
clear kkk;
fi=length(filenamelist); % Determina el nombre total de
fotos a processar.

% Genera totes les malles
for i=1:increment:fi
foto=imread(convertStringsToChars(filenamelist(i,:)));
%Llegeix fotograma
carpeta=num2str(i-1); % Arrodoneix nombre
mkdir(carpeta); % Crea la carpeta
cd(carpeta); % Va a la carpeta
disp(carpeta);
rect_grid(foto);
cd ('..');
end
close all; % Tanca les figures
end
```

```

function rect_grid(im_grid)

    % Function to select a rectangular grid and to add these to an
    existing one
    % wirtten by Chris
    %
    tria=0;
    mida=size(im_grid);      %Busca dimensions de la foto
    CORSIZE=25;              % Estableix el marge
    X1=CORSIZE;Y1=X1;X2=mida(2)-CORSIZE;Y2=mida(1)-CORSIZE;    %
    Determina limits de la malla.
    xspacing = 50;yspacing = 50;                                % Defineix espaiat

    while tria~=1
        % Dibuixa foto
        hold off
        imshow(im_grid,'InitialMagnification',100); %show chosen Image
        hold on
        plot([X1,X1,X2,X2,X1],[Y1,Y2,Y2,Y1,Y1]);
        hold off

        title(sprintf('Define the region of interest. Pick (single
        click) a point in the LOWER LEFT region of the gage section.\n Do the
        same for a point in the UPPER RIGHT portion of the gage section.'))

        % Agafem el primer punt
        [x(1,1),y(1,1)]=ginput(1);
        hold on
        plot(x(1,1),y(1,1),'+b')

        % Agafem la 2a cantonada
        [x(2,1),y(2,1)]=ginput(1);
        hold on
        plot(x(2,1),y(2,1),'+b')

        % Dibuixem
        drawnow

        % Ordenem i restringim a estar dins dels marges.
        xmin = min(x);if xmin<X1;xmin=X1;end
        xmax = max(x);if xmax>X2;xmax=X2;end
        ymin = min(y);if ymin<Y1;ymin=Y1;end
        ymax = max(y);if ymax>Y2;ymax=Y2;end

        % Round xmin,xmax and ymin,ymax "up" based on selected spacing
        numXelem = ceil((xmax-xmin)/xspacing)-1;
        numYelem = ceil((ymax-ymin)/yspacing)-1;

        xmin_new = (xmax+xmin)/2-((numXelem/2)*xspacing);
        xmax_new = (xmax+xmin)/2+((numXelem/2)*xspacing);
        ymin_new = (ymax+ymin)/2-((numYelem/2)*yspacing);
        ymax_new = (ymax+ymin)/2+((numYelem/2)*yspacing);

        % Create the analysis grid and show user
        [x,y] =
        meshgrid(xmin_new:xspacing:xmax_new,ymin_new:yspacing:ymax_new);
        [rows, columns] = size(x);
        %zdummy = 200.*ones(rows,columns);
        imshow(im_grid)
    end
end

```

```

        title(['Selected grid has ',num2str(rows*columns), '
rasterpoints']) % plot a title onto the image
        hold on;
        plot(x,y,'+b')

        % Pregunta si Ãs correcte
        tria = menu(sprintf('Es correcte?'),...
            'Si','No, repetiex');

        close all
    end

    % Save settings and grid files in the image directory for
    visualization/plotting later
    x=reshape(x,[],1);
    y=reshape(y,[],1);
    grid_x=x;
    grid_y=y;
    save settings.dat xspacing yspacing xmin_new xmax_new ymin_new
    ymax_new -ascii -tabs
    save grid_x.dat grid_x -ascii -tabs
    save grid_y.dat grid_y -ascii -tabs
    hold off;
end

function autoflexfill(increment)
    %Defineix increment per defecte com a 1500
    if exist('increment','var')==0
        increment=1500;
    end

    % Llegeix llistat de fotogrames
    kkk=load('filenamelist'); % file with the list of
    filenames to be processed
    filenamelist=kkk.filenamelist;
    clear kkk;
    fi=length(filenamelist); % Determina el nombre total de
    fotos a processar.

    % Processa totes les malles
    for i=1:increment:fi
        carpeta=num2str(i-1); % Arrodoneix nombre
        disp(carpeta); % Mostra la carpeta a processar
        grid_x=load(strcat(carpeta,'/grid_x.dat')); % llegeix
        la malla x feta per l'autogrid
        grid_y=load(strcat(carpeta,'/grid_y.dat')); %
        llegeix la malla y feta per l'autogrid
        llista=filenamelist(i:min(i+increment,fi)); % Assigna els
        fotogrames a processar
        automate_image_mp_2017b(1,grid_x,grid_y,llista);
        deformacio_uniforme;
        movefile('validx.dat',carpeta); % Mou validx.dat a
        la carpeta
        movefile('validy.dat',carpeta); % Mou validy.dat a
        la carpeta
        movefile('deformacions.txt',carpeta); % Mou
        deformacions.txt a la carpeta
    end
end

```

```

function [corba_traccio]=genera_corba_traccio(ex,ey,forca,V_freq);

%load data in case you did not load it into workspace yet
if exist('videox')==0
    if exist('./Procesat video/deformacions.txt')
        videoxname='./Procesat video/deformacions.txt';
    else
        [videoxname,Pathvideox] = uigetfile('*.txt','Open
deformacions.txt');
        if videoxname==0
            disp('You did not select a file!')
            return
        end
        cd(Pathvideox);
    end
    kk=importdata(videoxname,' ');
    ex=kk.data(:,1);
    ey=kk.data(:,2);
end
%define the size of the data set
N=size(ex);
N=N(1);
% Assigna freqÀncia de video de HD si no en rep cap
metro=dir('*-video.txt'); % cerca fitxer d'informaciÀ del
vÀdeo
if length(metro)==1
    file10 = fopen(metro(1).name,'r','ieee-be');
% Obre arxiu auxiliar de video
    textscan(file10, '%s', 1,'Delimiter', '\t', 'ReturnOnError',
false); % Salta 1 línia
    V_freq=fscanf(file10,'%f',1);
% Llegeix frequencia de video
    textscan(file10, '%s', 1,'Delimiter', '\t', 'ReturnOnError',
false); % Salta 1 línia
    mida_x=fscanf(file10,'%f',1);
% Llegeix amplada imatge
    textscan(file10, '%s', 1,'Delimiter', '\t', 'ReturnOnError',
false); % Salta 1 línia
    mida_y=fscanf(file10,'%f',1);
% Llegeix alçada imatge
    fclose(file10);
% Tanca arxiu
else
    V_freq=59.94006;
end

% Llegeix forÀça
if exist('forca')==0
    metro=dir('*.asc');
    if length(metro)==1
        Dades10name=metro(1).name;
    else
        [Dades10name,PathDades10] = uigetfile('*.asc','Open CATMAN
file');
        if Dades10name==0
            disp('You did not select a file!')
            return
        end
        cd(PathDades10);
    end
    Dades10=llegeix_catman(Dades10name); % Ja inverteix la ForÀça

```

```

end

%define the size of the data set
N_forca=size(Dades10);
N_forca=N_forca(1);

% Converteix la matriu en vectors
F_t=Dades10(1:N_forca,1);
Forca=Dades10(1:N_forca,2);
clear Dades10;

% Demana la secció i la primera dada de for
seccio = llegeix_seccio();
F_inici = llegeix_inici_forca();

% Genera la base de temps del video i en calcula la tensió.
% També grava les dades.
kklarg=size(Dades10name);
fitxersortida=strcat(Dades10name(1:(kklarg(2)-4)),'-corba-
cor.txt');
fitxerid=fopen(fitxersortida,'w+');
fprintf(fitxerid,'%s \t%s \t%s \t%s \n','ex','ey','s (MPa)','t
(s)');
v_temps=zeros(N,1);sigma=zeros(N,1);
j=F_inici;F_t_i=F_t(F_inici);
F_temps=F_t-F_t_i;
for i=1:N
    v_temps(i)=(i-1)/V_freq;
    while F_temps(j)<=v_temps(i)
        j=j+1;
    end
    sigma(i)=(Forca(j-1)+(Forca(j)-Forca(j-1))*(v_temps(i)-
F_temps(j-1))/(F_temps(j)-F_temps(j-1)))/seccio;
    fprintf(fitxerid,'%f \t%f \t%f \t%f
\n',ex(i),ey(i),sigma(i),v_temps(i));
end
% Tanca tot
fclose(fitxerid);
close all;
clear ;
end

function [seccio]=llegeix_seccio()
    metro=dir('*-metro.txt');
    if length(metro)==1
        file10 = fopen(metro(1).name,'r','ieee-be');
% Obre arxiu de metrologia
        textscan(file10, '%s', 1,'Delimiter', '\t',
'ReturnOnError', false); % Salta l línia
        seccio=fscanf(file10,'%f',1);
% Llegeix seccio
        fclose(file10);
% Tanca arxiu Cadman
    else
        answer = inputdlg({'Secció de la proveta en mm²
?'}, 'Càlcul de tensions',1,{ '1' });
        seccio = str2double(cell2mat(answer(1,1)));
        clear answer;
        metro=dir('*-sincro.txt'); % Busca nom de la mostra
        if length(metro)==1

```

```

        metro=strcat(metro(1).name(1:end-10),'metro.txt'); %
Genera nom del fitxer de metrologia
        fitxerid=fopen(metro,'w+');
        fprintf(fitxerid,'%s \n','Secció de la proveta en
mm² : ');
        fprintf(fitxerid,'%f \n',seccio);
        fclose(fitxerid);
    end
end
end

function [F_inici]=llegeix_inici_forca()
    sincro=dir('*-sincro.txt');
    if length(sincro)==1
        kk=importdata(sincro(1).name,'',16); %llegeix el dada
d'inici de l'assaigs
        F_inici=kk(1).data;
    else
        answer = inputdlg({'Numero de dada de força de l'inici
de l'assaigs ?'},'Sincronisme de temps',1,{ '1' });
        F_inici = str2num(cell2mat(answer(1,1)));
        clear answer;
    end
end
end

```