

TRABAJO FIN DE MÁSTER

Análisis y optimización de errores cinemáticos en una máquina CNC de 5 ejes

Realizado por
Yasser Zekalmi Richa

Para la obtención del título de
Máster en Ingeniería Mecánica

Dirigido por
José Antonio Albajez
María José Oliveros



Universidad
Zaragoza



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza



DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe remitirse a seceina@unizar.es dentro del plazo de depósito)

D./D^a. YASSER ZEKALMI RICHA ,

en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de Estudios de la titulación de Máster Universitario en Ingeniería Mecánica

 (Título del Trabajo)

ANÁLISIS Y OPTIMIZACIÓN DE ERRORES CINEMÁTICOS EN UNA MÁQUINA CNC DE 5 EJES

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 31/05/2023

Firmado por ZEKALMI RICHA, YASSER
(AUTENTICACIÓN) el día 31/05/2023
con un certificado emitido por AC DNIE
005

Fdo:

Índice

A. Anexos	6
A.1. Simulaciones adicionales	6
A.1.1. Cambiando la forma de los errores	6
A.2. Estrategia no perpendicular	6
A.3. Script de limpieza para la extracción de datos del CAM	6
A.4. Código del algoritmo de optimización en Matlab	19
A.4.1. Función Main	19
A.4.2. Función PSO general	39
A.4.3. Función Modelo cinemático HAAS	44
A.4.4. Función Mover herramienta	47
A.4.5. Función Check colisiones	49
A.4.6. Función Coste colisiones	50
A.4.7. Función Coste total	51
A.4.8. Función Gráficas errores cinemáticos	52
A.4.9. Función Pruebas	57

Índice de figuras

1.	Resultados cambiando la forma de los errores geométricos. (Elaboración propia).	6
2.	Resultados cambiando la estrategia a una no perpendicular. (Elaboración propia).	6

Índice de tablas

A. Anexos

A.1. Simulaciones adicionales

A.1.1. Cambiando la forma de los errores

En la figura 1 se demuestra que cambiar la forma de los errores geométricos, ampliando su valor y desfasando las senoidales, resulta en cambios en la forma de los perfiles.

sol inicial gbest, sin control Δ

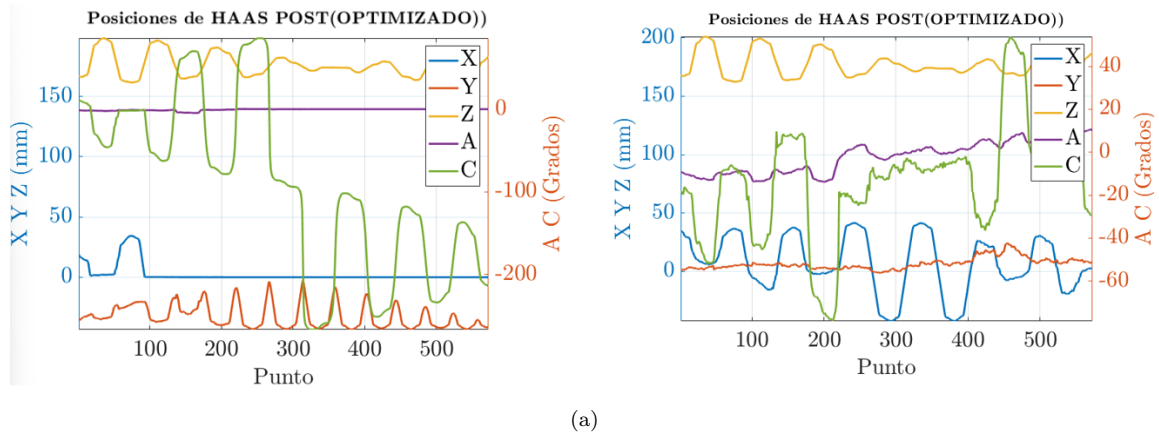


Figura 1: Resultados cambiando la forma de los errores geométricos. (Elaboración propia).

A.2. Estrategia no perpendicular

En la figura 2 a la izquierda se ven los perfiles originales de la estrategia no perpendicular y a la derecha los optimizados. Cómo se ve la forma de los perfiles no representa los zig-zags de las pasadas, como sí lo hacían los perfiles de la estrategia perpendicular.

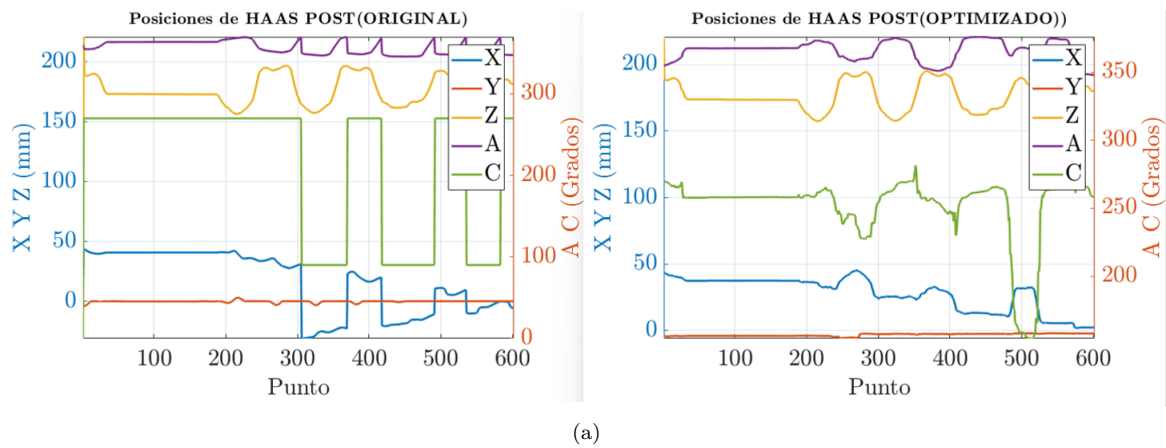


Figura 2: Resultados cambiando la estrategia a una no perpendicular. (Elaboración propia).

A.3. Script de limpieza para la extracción de datos del CAM

tfm

July 17, 2023

[]:

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mping
import re
import random

from matplotlib import rcParams

import os

from scipy.stats import linregress

##plotly

import plotly.express as px
import plotly.graph_objs as go
from plotly.subplots import make_subplots
from plotly.offline import plot, iplot, init_notebook_mode

from scipy.interpolate import interp1d
from scipy.interpolate import interp2d

plt.style.use(['default'])
plt.style.use(['science', 'notebook', 'grid'])

#### Tema negro matplotlib #####

# plt.style.use('dark_background')

import matplotlib
matplotlib.rcParams['mathtext.fontset'] = 'stix'
```

```

matplotlib.rcParams['font.family'] = 'STIXGeneral'

plt.rcParams['axes.prop_cycle'] = plt.cycler(color=['#1F77B4', '#FF7F0E',
↪ '#2CA02C', '#D62728', '#9467BD', '#8C564B', '#E377C2', '#7F7F7F', '#BCBD22',
↪ '#17BECF'])

#### Funcion girar / a \\ para directorios

def pa(path):
    pathi=list(path)
    c=0
    for i in path:
        if i==path[2]:
            pathi[c]='/'
            c+=1
    str_f=''.join(pathi)

    return str_f

def regre(x,y):
    m, inter, r, p_value, std_err = linregress(x, y)
    return m, inter, r

```

```
[2]: os.chdir(pa(r'C:\Users\User\Desktop\__TFM\1-pruebas'))
```

```
[3]: 12/1.26
```

```
[3]: 9.523809523809524
```

```
[1]: # ls
```

1 Script de limpieza GOTO

```

[5]: df=pd.read_csv('goto_HAAS_true.csv', sep=';', header=None)
df.fillna('o', inplace=True)

df[:10]

```

```

[5]:
0  GOTO/49.6707,-41.9729,203.0414,0.5540418,-0.14...
1                                PAINT/COLOR,211
2                                RAPID
3  GOTO/24.8124,-35.6422,166.2304

```

```

4                                PAINT/COLOR,42
5                                FEDRAT/MMPM,250.0000
6                                GOTO/24.3259,-35.5423,165.7134
7                                GOTO/23.7051,-35.4517,165.3657
8                                GOTO/23.0106,-35.3768,165.2002
9                                GOTO/22.2943,-35.3193,165.1978

```

```
[6]: df.iloc[23][0][:4]
```

```
[6]: 'GOTO'
```

```
[7]: mat=np.zeros((len(df),6))

fi=[1,2,3,4,5,6]

mat[0][:]=fi
mat
```

```
[7]: array([[1., 2., 3., 4., 5., 6.],
           [0., 0., 0., 0., 0., 0.],
           [0., 0., 0., 0., 0., 0.],
           ...,
           [0., 0., 0., 0., 0., 0.],
           [0., 0., 0., 0., 0., 0.],
           [0., 0., 0., 0., 0., 0.]])
```

```
[8]: mat=[[1,2,3,4],[1,2,3,5],[1,1,1,1]]
mat[2][1]=3
mat
```

```
[8]: [[1, 2, 3, 4], [1, 2, 3, 5], [1, 3, 1, 1]]
```

```
[9]: mat=np.zeros((len(df),6))

j=0

df_i=pd.DataFrame(columns=["X", "Y", "Z", "i","j","k"])

for i in range(0, len(df), 1):
    if df.iloc[i][j][:4]=='GOTO':
        vec=df.iloc[i][j][5:].split(',')
        fila=np.zeros((1,6))
        for k in range(0, len(vec),1):
            fila[0][k]=(float(vec[k]))
        # print(fila[0])

        df_i.loc[len(df_i)]=fila[0]
```

```
df_i[:-1]
```

```
[9]:
```

	X	Y	Z	i	j	k
0	49.6707	-41.9729	203.0414	0.554042	-0.141099	0.820444
1	24.8124	-35.6422	166.2304	0.000000	0.000000	0.000000
2	24.3259	-35.5423	165.7134	0.000000	0.000000	0.000000
3	23.7051	-35.4517	165.3657	0.000000	0.000000	0.000000
4	23.0106	-35.3768	165.2002	0.000000	0.000000	0.000000
..	...	...	...	...	...	...
568	21.9120	35.1194	180.6836	0.000000	0.000000	0.000000
569	22.2996	35.0560	181.2853	0.000000	0.000000	0.000000
570	22.5386	34.9732	181.9572	0.000000	0.000000	0.000000
571	22.5847	34.8731	182.6660	0.000000	0.000000	0.000000
572	22.4157	34.7628	183.3539	0.000000	0.000000	0.000000

```
[573 rows x 6 columns]
```

```
[10]: # df_i.to_excel('goto_HAAS_true_v2.xlsx', header=False, index=False)
```

2 Sript limpieza POST

```
[11]: df=pd.read_csv('post_HAAS_true.csv', sep=';', header=None)
df.fillna('o', inplace=True)

df.iloc[24][0].split('X')

df
```

```
[11]:
```

0	0
0	G54
1	G00 X-28.42 Y68.09
2	G00 A-34.87 C75.71
3	G43 H07 Z200.03
4	S1000 M03
..	...
585	G54 G90 A0 C0
586	G53 X-500. Y0.0
587	M30
588	%
589	El listado Trayectoria para herramientas conti...

```
[590 rows x 1 columns]
```

```
[12]: x=[]
y=[]
```

```

z=[]
b=[]
c=[]

dicto={'X': [0], 'Y': [0], 'Z': [0], 'A': [0], 'C': [0]}

for i in range(0, len(df), 1):
    for j in range(0, len(dicto), 1 ):
        keys = list(dicto.keys())
        var=keys[j]

        if len(df.iloc[i][0].split(var))==1:
            dicto[var].append(dicto[var][-1])

        if len(df.iloc[i][0].split(var))!=1:
            my_string = df.iloc[i][0].split(var)[1]

            # captura patrones de numeros asi
            pattern = r'([-\\d.]+)'

            # pasarlos a formato numero

            matches = re.findall(pattern, my_string)
            # print(matches)

            val=float(matches[0])
            dicto[var].append(val)

df_post=pd.DataFrame(dicto)
df_post[560:587]

```

```

[12]:
      X      Y      Z      A      C
560 -26.21  51.74  160.88 -26.30  239.41
561 -30.58  54.96  160.69 -24.95  246.13
562 -33.80  60.08  159.98 -23.91  253.63
563 -35.45  67.09  158.72 -23.26  261.74
564 -35.20  75.70  156.93 -23.05  270.20
565 -33.00  85.38  154.60 -23.29  278.65
566 -29.07  95.51  151.78 -23.96  286.74
567 -26.59  100.58  150.19 -24.44  290.57
568 -25.92  101.84  149.78 -24.57  291.49
569 -25.58  102.46  149.57 -24.64  291.95
570 -25.58  102.46  149.57 -24.64  291.95
571 -25.09  103.81  149.66 -24.64  291.95
572 -24.84  104.48  149.79 -24.64  291.95
573 -24.61  105.13  149.99 -24.64  291.95

```

574	-24.38	105.74	150.28	-24.64	291.95
575	-24.18	106.30	150.69	-24.64	291.95
576	-24.01	106.75	151.22	-24.64	291.95
577	-23.90	107.05	151.86	-24.64	291.95
578	-23.86	107.16	152.57	-24.64	291.95
579	-23.86	107.16	192.45	-24.64	291.95
580	-23.86	107.16	0.00	-24.64	291.95
581	-23.86	107.16	0.00	-24.64	291.95
582	-23.86	107.16	0.00	-24.64	291.95
583	-23.86	107.16	0.00	-24.64	291.95
584	-23.86	107.16	0.00	-24.64	291.95
585	-23.86	107.16	0.00	0.00	0.00
586	-23.86	107.16	0.00	0.00	0.00

```
[ ]:
```

```
[13]: # df_post.to_excel('post_HAAS_true_v2.xlsx', header=False, index=False)
```

```
[ ]:
```

3 Script de Limpieza REVIEW

4 Sacar POST

```
[27]: df=pd.read_excel('estrategia_c3_review.xlsx', header=None)
# df.fillna('o', inplace=True)

# df
```

```
[28]: # # mat=[]

# # linea=[]

# var=df[0][1873].split('mom_pos')[1][4:]

# print(df[0][1873].split('mom_pos')[1][4:])

col=df.columns

df=df.dropna()
df
```

```

[28]:
0
0 E MOM__part_attributes 0
2 V mom_attr_PART_NX_NON_MASTER = CAM
3 V mom_nx_materialmultipleassigned = FALSE
4 V mom_attr_PART_NX_MaterialMultipleAssigned = ...
5 V mom_nx_materialmissingassignments = TRUE
...
110066 0 mom_o_buffer = M30
110067 V mom_skip_handler_to_event =
110068
110069 V mom_event_error =
110070 V mom_warning_info =

[110070 rows x 1 columns]

```

```

[29]: x=[]
y=[]
z=[]
b=[]
c=[]

dicto={'0': [0], '1': [0], '2': [0], '3': [0], '4': [0]}

for i in range(0, len(df), 1):
    for j in range(0, len(dicto), 1 ):
        keys = list(dicto.keys())
        var='mom_pos['+keys[j]+']'

        # if len(df.iloc[i][0].split(var))==1:
        #     dicto[keys[j]].append(dicto[keys[j]][-1])

        if len(df.iloc[i][0].split(var))!=1:
            my_string = df.iloc[i][0].split(var)[1]

            # captura patrones de numeros asi
            pattern = r'([-\\d.]+)'

            # pasarlos a formato numero

            matches = re.findall(pattern, my_string)
            # print(matches)

            val=float(matches[0])
            dicto[keys[j]].append(val)

df_post=pd.DataFrame(dicto)

```

```
[30]: cols=df_post.columns
```

```
[31]: # Limpiar los REPETIDOS

delta=np.zeros(len(df_post)-1)

for i in range(0, len(df_post)-1, 1):
    delta[i]=df_post[cols[0]][i+1]-df_post[cols[0]][i]
    if delta[i]==0:
        df_post=df_post.drop(i)

# print(delta)
```

```
[32]: len(df_post)
```

```
[32]: 1016
```

```
[34]: # df_post.to_excel('post_estrategia_c3_review.xlsx', header=False, index=False)
```

5 Script de Limpieza REVIEW

6 sacar GOTO

```
[7]: df=pd.read_excel('estrategia_c3_review.xlsx', header=None)
# df.fillna('o', inplace=True)
```

```
[8]: col=df.columns

df=df.dropna()
df
```

```
[8]:
0
0 E MOM__part_attributes 0
2 V mom_attr_PART_NX_NON_MASTER = CAM
3 V mom_nx_materialmultipleassigned = FALSE
4 V mom_attr_PART_NX_MaterialMultipleAssigned = ...
5 V mom_nx_materialmissingassignments = TRUE
...
110066 0 mom_o_buffer = M30
110067 V mom_skip_handler_to_event =
110068
110069 V mom_event_error =
110070 V mom_warning_info =
```

[110070 rows x 1 columns]

```
[10]: df.iloc[300][0]
```

```
[10]: 'V mom_sequence_increment = 10'
```

```
[11]: x=[]
y=[]
z=[]
b=[]
c=[]

dicto={'0': [0], '1': [0], '2': [0], '0i': [0], '1j': [0], '2k': [0]}

for i in range(0, len(df), 1):
    for j in range(0, 3, 1):
        keys = list(dicto.keys())
        # var='mom_contact_goto['+keys[j]+']'

        var='mom_mcs_goto['+keys[j]+']'

        # if len(df.iloc[i][0].split(var))==1:
        #     dicto[keys[j]].append(dicto[keys[j]][-1])

        if len(df.iloc[i][0].split(var))!=1:
            my_string = df.iloc[i][0].split(var)[1]

            # captura patrones de numeros asi
            pattern = r'([-\\d.]+)'

            # pasarlos a formato numero

            matches = re.findall(pattern, my_string)
            # print(matches)

            val=float(matches[0])
            dicto[keys[j]].append(val)

for j in range(3, len(dicto), 1):
    keys = list(dicto.keys())
    # var='mom_contact_vector['+keys[j][0]+']'
    var='mom_tool_axis['+keys[j][0]+'']
```

```

# if len(df.iloc[i][0].split(var))==1:
#     dicto[keys[j]].append(dicto[keys[j]][-1])

if len(df.iloc[i][0].split(var))!=1:
    my_string = df.iloc[i][0].split(var)[1]

    # captura patrones de numeros asi
    pattern = r'([-\\d.]+)'

    # pasarlos a formato numero

    matches = re.findall(pattern, my_string)
    # print(matches)

    val=float(matches[0])
    dicto[keys[j]].append(val)

df_post=pd.DataFrame(dicto)

```

```
[12]: df_post
```

```

[12]:
      0      1      2      0i  1j      2k
0    0.000000  0.000000  0.000000  0.000000  0.0  0.000000
1    0.000000  0.000000  0.000000  0.000000  0.0  1.000000
2   -23.805387 -43.227978  219.601068 -0.089465  0.0  0.995990
3   -20.977585 -43.227978  188.120066 -0.089465  0.0  0.995990
4   -20.812555 -43.067032  187.521060 -0.089465  0.0  0.995990
...
1028 -19.955244  42.318703  153.819723 -0.112666  0.0  0.993633
1029 -20.383723  42.695658  153.508040 -0.112666  0.0  0.993633
1030 -20.838860  43.139927  153.460642 -0.112666  0.0  0.993633
1031 -21.085394  43.353937  153.982948 -0.112666  0.0  0.993633
1032 -28.525728  43.353937  219.601068 -0.112666  0.0  0.993633

```

[1033 rows x 6 columns]

```
[13]: cols=df_post.columns
cols
```

```
[13]: Index(['0', '1', '2', '0i', '1j', '2k'], dtype='object')
```

```
[14]: df_post
len(df_post)
```

```
[14]: 1033
```

```
[15]: # Limpiar los REPETIDOS

delta=np.zeros(len(df_post)-1)

df_poste=df_post

for i in range(0, len(delta), 1):
    delta[i]=df_poste[cols[0]][i+1]-df_poste[cols[0]][i]
    if abs(delta[i])<=1e-7:
        df_poste=df_poste.drop(i)

# print(delta)

[16]: print(len(df_poste))

1016

[18]: # df_poste.to_excel('goto_estrategia_c3_review.xlsx', header=False, index=False)

[ ]:

[ ]:
```

7 Fin

```
[ ]:

[2]: file=os.listdir(pa(r'C:\Users\User\Desktop\__TFM\1-pruebas'))
os.chdir(pa(r'C:\Users\User\Desktop\__TFM\1-pruebas'))

[3]: file

[3]: ['espiral.m',
      'p.m',
      'paraboloide-solid-o.stl',
      'prueba.m',
      'S.csv',
      'sur-pboloide.ply',
      'tray-cam.csv',
      'tray-cam.xlsx']

[4]: points=pd.read_csv('S.csv', sep=',', header=None)
col=points.columns

points['r']=np.ones(len(points[col[0]]))
points['g']=np.ones(len(points[col[0]]))
```

```
points['b']=np.ones(len(points[col[0]]))
```

```
col=points.columns  
points
```

```
[4]:
```

	0	1	2	r	g	b
0	-39.936	-2.208	31.645792	1.0	1.0	1.0
1	-39.936	-2.112	31.573888	1.0	1.0	1.0
2	-39.936	-2.016	31.501984	1.0	1.0	1.0
3	-39.936	-1.920	31.430080	1.0	1.0	1.0
4	-39.936	-1.824	31.358176	1.0	1.0	1.0
...	...	...	...	...	...	...
545436	39.936	1.824	20.918176	1.0	1.0	1.0
545437	39.936	1.920	20.966080	1.0	1.0	1.0
545438	39.936	2.016	21.013984	1.0	1.0	1.0
545439	39.936	2.112	21.061888	1.0	1.0	1.0
545440	39.936	2.208	21.109792	1.0	1.0	1.0

[545441 rows x 6 columns]

```
[5]: points[col[0]]
```

```
[5]:
```

0	-39.936
1	-39.936
2	-39.936
3	-39.936
4	-39.936
...	...
545436	39.936
545437	39.936
545438	39.936
545439	39.936
545440	39.936

Name: 0, Length: 545441, dtype: float64

```
[6]: from pyntcloud import PyntCloud  
  
d = {'x': points[col[0]], 'y': points[col[1]], 'z': points[col[2]],  
      'red' : points[col[3]], 'green' : points[col[4]], 'blue' :  
      ↪points[col[5]]}  
  
cloud = PyntCloud(pd.DataFrame(data=d))  
cloud.to_file("sur-pboloide.ply")
```

```
[ ]:
```

A.4. Código del algoritmo de optimización en Matlab

A.4.1. Función Main

```
set(groot, 'defaultFigureColor', 'white');
set(groot, 'defaultTextInterpreter', 'latex');
% set(groot, 'defaultAxesTickLabelInterpreter', 'default');
% set(groot, 'defaultColorbarTickLabelInterpreter', 'default');

set(groot, 'defaultAxesFontSize', 13)
set(groot, 'defaultLineLineWidth', 1)
set(groot, 'defaultLineMarkerSize', 7)

%% ##### FUNCION MAIN #####

clear
close all

% LEO P OBJ y los proyecto en la superficie teorica
% Projectando los puntos objetivo sobre la superficie teorica

% p_obj=xlsread('p_haas_goto_tra_2.xlsx');
% p_post=xlsread('p_haas_post_tra_2.xlsx');

% p_post=xlsread('p_haas_post_ejes.xlsx');
% p_obj=xlsread('p_haas_goto_ejes.xlsx');

% ----- estrategia perpendicular

% p_post=xlsread('post_HAAS_true.xlsx');

% p_obj=xlsread('goto_HAAS_true.xlsx');

p_post=xlsread('post_review_true.xlsx');
p_obj=xlsread('goto_review_true.xlsx');

% ----- estrategia no perpendicular

% p_post=xlsread('post_estrategia_c3_review.xlsx');
% p_obj=xlsread('goto_estrategia_c3_review.xlsx');

tabla=load('Tabla errores MH Fexac.mat');

% tabla=load('Tabla errores MH Fexac modifi.mat');
% tabla=tabla.tabla;

% p_post=[p_post(1:300,:); p_post];
% p_obj=[p_obj(1:300,:); p_obj];

%%
% excluyo el primer y ultimo dato ver los mas cercanos a

p_obj=p_obj(2:end-1, :);
p_post=p_post(2:end-1, :);
```

```

% [vertices] = stlread("Herramienta-quad-hibrid.STL");

[vertices] = stlread("Herramienta-quad.STL");

vertices=vertices.Points;

r=5;

%
p_obj_teo=zeros(size(p_obj,1),3);
n_obj_teo=zeros(size(p_obj,1),3);

for i=1:size(p_obj,1)
    px=p_obj(i,1);
    py=p_obj(i,2);

    p_teo=pboide(px,py);

    p_obj_teo(i,:)=p_teo;
    n_obj_teo(i,:)=normal_surf(p_obj_teo(i,:));

    if p_teo(3)==0
        p_obj_teo(i,:)=p_obj(i, 1:3);
        n_obj_teo(i,:)=p_obj(i, 4:6);
    end

%     if p_teo(3)==0 && i~=1
%         p_obj_teo(i,:)=p_obj(i, 1:3);
%         n_obj_teo(i,:)=normal_surf(p_obj_teo(i-1,:));
%     end
%
%     if p_teo~=0
%
%     end

end

obj_tot=p_obj_teo+r*n_obj_teo;

p_obj_teo(end,:)=p_obj(end,1:3);
% p_obj_teo=p_obj(:,1:3);

%% ##### Algoritmo --- _PSO_ ---- #####

nom='si_zeros_gbest_no_control_full_aaa';

a=1;
b=a+100;
% b=size(obj_tot,1);

tic

error_pso=zeros(size(obj_tot,1), 5);

gbest_tot=zeros(size(obj_tot,1), 5);
cost_tot=zeros(size(obj_tot,1), 1);

```

```

normal=n_obj_teo(a,:);
[angulos]=pto_rot(normal);

tx=angulos(1);
tz=angulos(3);

%----- sol_inicial=[p_post(a, 1:3), tx, tz];

% sol_inicial=p_post(a,:);

% sol_inicial=zeros(1,5);

% sol_inicial=[p_post(a,1:3), 0, 0];

sol_inicial=[0, 2, 2, 0, 0];

min_dz_tot=NaN(size(obj_tot,1),1);
col_tot=NaN(size(obj_tot,1),1);

% p_obj_teo=p_obj(:,1:3);

hfigpso=figure;

error_pso(1:5,2)=0;

yyaxis left

e=plot(error_pso(:,2), 'LineWidth', 2);
hold on
grid on

ylabel('Error Distancia')

yyaxis right

rug=error_pso(:,5);
rug(rug>1e-4)=NaN;
r=plot(rug, 'LineWidth', 2);

ylabel('Error Rugosidad')

legend('$distancia$', '$rugosidad$', 'Interpreter', 'latex')
xlabel('Punto')

name=strcat('PS0_ejecucion_',num2str(nom),'.png');

for_fig(hfigpso, 20, 0.69);
view(0,90);

% title(strcat('PS0_ejecucion_',num2str(nom)), 'FontSize', 10)

error_deltas=zeros(size(p_obj_teo));

for ind=a:b

```

```

%     disp(sol_inicial);

[gbest, error_real, choque_obj, error_ac, error_aca, particula]=
    pso_general(p_obj_teo, n_obj_teo, ind, sol_inicial, tabla, vertices);

[points_final, min_dz, delta_z, colisiona]=situ_herramienta(vertices,
    gbest.pos, tabla, p_obj_teo(ind,:), n_obj_teo(ind,:), ind, 5);
[coste_col, min_dz]=coste_colisiones(gbest.pos, p_obj_teo(ind,:),
    n_obj_teo(ind,:), vertices, 5, tabla, ind);
min_dz_tot(ind,:)=min_dz;
col_tot(ind,:)=coste_col;

error_deltas(ind,:)=error_ac;

disp(error_aca)
disp(['error delta: ', num2str(error_ac)])

%     disp(gbest.pos)
%     fprintf('Pto: -> %d Costes: %d : %d : %d : %d -- R: %d\n', ind, gbest.
cost, error_real, choque_obj, error_ac, min_dz)

error_pso(ind,:)=[ gbest.cost, error_real, choque_obj, error_ac, min_dz
    ];

gbest_tot(ind, :)=gbest.pos;
cost_tot(ind)=gbest.cost;

% ----- Figura de errores

set(e, 'YData', error_pso(:,2));
rug=error_pso(:,5);
rug(rug>1e-4)=NaN;
set(r, 'YData', rug);

for_fig(hfigpso, 20, 0.69);

xlim([a, ind+1])

drawnow

%     pos=modelo_haas(p_post(i,:), tabla);

%     sol_inicial=[p_post(i,1:3), gbest.pos(4), gbest.pos(5)];

sol_inicial=gbest.pos;

%     sol_inicial=p_post(ind,:);

%     sol_inicial=zeros(1,5);

end

frame = getframe(hfigpso);

```

```

imwrite(frame2im(frame), strcat('C:\Users\User\Desktop\__TFM\4-resultados\
    images\' ,name));

toc

%% ----- guarda los resultados en csv

chdir('C:\Users\User\Desktop\__TFM\4-resultados\matrices')

writematrix(gbest_tot, strcat('gbest_tot_', nom, '.csv'))
writematrix(min_dz_tot, strcat('min_dz_tot_', nom, '.csv'))
writematrix(error_pso, strcat('error_pso_', nom, '.csv'))

chdir('C:\Users\User\Desktop\__TFM\1-pruebas')

%% ##### Graficas de INTERES
#####

%----- grafica de XYZAC -----

% --- f1 ---

% chdir('C:\Users\User\Desktop\__TFM\4-resultados\matrices')
% gbest_tot = readmatrix('gbest_tot_si_post_control_xyzac.csv');

% gbest_tot=p_post;
% nom='POST (ORIGINAL)';

% close all

% p_a=p_post(:,4);
%
% indiz=(p_a<300);
%
% p_a(indiz)=p_a(indiz)+360;
% p_post(:,4)=p_a;

chdir('C:\Users\User\Desktop\__TFM\1-pruebas')

hfig=figure;

col=lines(100);

p_tam=size(p_obj,1);
it=round(linspace(a, b+1, b-a+1));
grid on
hold on

yyaxis left

plot(it, gbest_tot(a:b,1), '-', 'DisplayName', 'X', 'LineWidth', 2, 'Color',
    col(1,:))

```

```

plot(it, gbest_tot(a:b,2), '-', 'DisplayName', 'Y', 'LineWidth', 2, 'Color',
     col(2,:))
plot(it, gbest_tot(a:b,3), '-', 'DisplayName', 'Z', 'LineWidth', 2, 'Color',
     col(3,:))

ylabel(' X Y Z (mm)')

title({'\textbf{\fontsize{7}{1}\selectfont Posiciones de HAAS POST(
      OPTIMIZADO))'}, 'Interpreter', 'latex');

angles = gbest_tot(a:b,4);
gA = mod(angles, 360);
gA(gA > 180) = gA(gA > 180) - 360;

angles = gbest_tot(a:b,5);
gC = mod(angles, 360);
gC(gC > 180) = gC(gC > 180) - 360;

yyaxis right

% plot(it, gbest_tot(a:b, 4), '-', 'DisplayName', 'A', 'LineWidth', 2, 'Color
    ', col(4,:))
% plot(it, gbest_tot(a:b, 5), '-', 'DisplayName', 'C', 'LineWidth', 2, 'Color
    ', col(5,:))

plot(it, (gbest_tot(a:b, 4)), '-', 'DisplayName', 'A', 'LineWidth', 2, 'Color
    ', col(4,:))
plot(it, (gbest_tot(a:b, 5)), '-', 'DisplayName', 'C', 'LineWidth', 2, 'Color
    ', col(5,:))

ylabel(' A C (Grados)')

xlim([a,b])

% xlim([0, 100])

legend

xlabel('Punto')

for_fig(hfig, 20, 0.69)

axis tight

frame = getframe(hfig);
name=strcat('XYZAC-', num2str(nom), '.png');
imwrite(frame2im(frame), strcat('C:\Users\User\Desktop\__TFM\4-resultados\
    images\'', name));

```

```

%% ----- grafica POST (Original)

% gbest_tot=p_post;
% nom='POST (ORIGINAL)';

% p_post(344:355, 5)=p_post(344:355, 5)-360;
% p_post(357:358, 4)=p_post(357:358, 4)+360;
% p_post(356, 5)=p_post(355, 5);

% close all

chdir('C:\Users\User\Desktop\__TFM\1-pruebas')

hfig=figure;

col=lines(100);

% it=round(linspace(a, b, b-a));
grid on
hold on

yyaxis left

plot(it, p_post(a:b,1), '-', 'DisplayName', 'X', 'LineWidth', 2, 'Color',
     col(1,:))
plot(it, p_post(a:b,2), '-', 'DisplayName', 'Y', 'LineWidth', 2, 'Color',
     col(2,:))
plot(it, p_post(a:b,3), '-', 'DisplayName', 'Z', 'LineWidth', 2, 'Color',
     col(3,:))

ylabel(' X Y Z (mm)')

title({'\textbf{\fontsize{7}{1}\selectfont Posiciones de HAAS POST(ORIGINAL)'}},
      'Interpreter', 'latex');

angles = p_post(a:b,4);
gA = mod(angles, 360);
gA(gA > 180) = gA(gA > 180) - 360;

angles = p_post(a:b,5);
gC = mod(angles, 360);
gC(gC > 180) = gC(gC > 180) - 360;

yyaxis right

plot(it, (p_post(a:b, 4)), '-', 'DisplayName', 'A', 'LineWidth', 2, 'Color',
     col(4,:))

```

```

plot(it, (p_post(a:b, 5)), '-', 'DisplayName', 'C', 'LineWidth', 2, 'Color',
      col(5,:))

ylabel(' A C (Grados)')

xlim([a,b])

axis tight

% xlim([0, 100])

legend

xlabel('Punto')

for_fig(hfig, 20, 0.69)

%% ----- Grafica de los deltas post-optimizado

%
% % delta=p_post-gbest_tot;
%
% delta=zeros(size(gbest_tot, 1), 5);
%
% for i=1:length(delta)-1
%
%     delta(i,:)=gbest_tot(i+1, :) - gbest_tot(i, :);
% end
%
%
% hfig=figure;
%
% yyaxis left
%
%
% plot(it, delta(a:b,1), '-', 'LineWidth', 2, 'DisplayName', '$\Delta X$', '
%     Color', col(1,:))
% hold on
% grid on
%
% plot(it, delta(a:b,2), '-', 'LineWidth', 2, 'DisplayName', '$\Delta Y$', '
%     Color', col(2,:))
% plot(it, delta(a:b,3), '-', 'LineWidth', 2, 'DisplayName', '$\Delta Z$', '
%     Color', col(3,:))
%
%
%
% leg=legend;
% xlabel('Punto')
% ylabel('$\Delta XYZ(mm)$')
% % title({'\textbf{\fontsize{7}{1}\selectfont POST(ORIGINAL) - POST(
%     OPTIMIZADO)'}}, 'Interpreter', 'latex');
%

```

```

% title({'\textbf{\fontsize{7}{1}\selectfont $\Delta$ GBEST'}}, 'Interpreter', 'latex');
%
%
% y1_max = max(max((delta(a:b, 1:3))));
%
% ylim([-y1_max, y1_max])
%
%
% yyaxis right
%
% plot(it, delta(a:b,4), '-','LineWidth', 2, 'DisplayName','$\Delta$ A$', 'Color', col(4,:))
% plot(it, delta(a:b,5), '-','LineWidth', 2, 'DisplayName','$\Delta$ C$', 'Color', col(5,:))
%
% ylabel('$\Delta$ AC(Grados)$')
%
% legend('Location','eastoutside')
%
%
% y2_max = max(max((delta(a:b, 4:5))));
%
% ylim([-y2_max, y2_max])
%
%
%
% % set(leg, 'Color', 'None')
% for_fig(hfig, 30, 0.5)
% view(0,90)
%
%
% frame = getframe(hfig);
% name=strcat('XYZAC-delta-',num2str(nom),'.png');
% imwrite(frame2im(frame), strcat('C:\Users\User\Desktop\__TFM\4-resultados\images\',name));
%
%
%
%
% % ----- deltas original
%
%
%
% delta_og=zeros(size(p_post, 1), 5);
%
% for i=1:length(delta_og)-1
%
%     delta_og(i,:)=p_post(i+1, :) - p_post(i, :);
% end
%
%
% hfig=figure;
%
% yyaxis left
%
%
%
```

```

% plot(it, delta_og(a:b,1), '-', 'LineWidth', 2, 'DisplayName', '$\Delta X$',
    'Color', col(1,:))
% hold on
% grid on
%
% plot(it, delta_og(a:b,2), '-', 'LineWidth', 2, 'DisplayName', '$\Delta Y$',
    'Color', col(2,:))
% plot(it, delta_og(a:b,3), '-', 'LineWidth', 2, 'DisplayName', '$\Delta Z$',
    'Color', col(3,:))
%
%
%
% leg=legend;
% xlabel('Punto')
% ylabel('$\Delta XYZ(mm)$')
% % title({'\textbf{\fontsize{7}{1}\selectfont POST(ORIGINAL) - POST(OPTIMIZADO)}'}, 'Interpreter', 'latex');
%
% title({'\textbf{\fontsize{7}{1}\selectfont $\Delta$ GBEST ORIGINAL'}}, 'Interpreter', 'latex');
%
%
% y1_max = max(max(abs(delta_og(a:b, 1:3)))));
%
% ylim([-y1_max, y1_max])
%
%
% yyaxis right
%
% plot(it, delta_og(a:b,4), '-', 'LineWidth', 2, 'DisplayName', '$\Delta A$',
    'Color', col(4,:))
% plot(it, delta_og(a:b,5), '-', 'LineWidth', 2, 'DisplayName', '$\Delta C$',
    'Color', col(5,:))
%
% ylabel('$\Delta AC(Grados)$')
%
% legend('Location','eastoutside')
%
%
% y2_max = max(max(abs(delta_og(a:b, 4:5)))));
%
% ylim([-y2_max, y2_max])
%
%
%
% % set(leg, 'Color', 'None')
% for_fig(hfig, 30, 0.5)
% view(0,90)
%
%
% frame = getframe(hfig);
% name=strcat('XYZAC-delta-og-',num2str(nom),'.png');
% imwrite(frame2im(frame), strcat('C:\Users\User\Desktop\__TFM\4-resultados\images\' ,name));

```

```

%% %----- grafica de CNC -----

% --- f2 ---

figure

az=70;
ceni=30;

name=strcat(nom,num2str(i),'.png');
name_gif=strcat(nom, num2str(az), '_', num2str(ceni));

% name_gif=strcat('ORIGINAL_', num2str(az), '_', num2str(ceni));
% gbest_tot=p_post;

%-----

ui= 0:0.007:1;
vi= 0:0.007:1;
l=length(ui);
S = zeros(l*1,3);

cont=1;
% A=zeros(1,3);

for i=1:l
    u=ui(i);
    for j=1:l
        v=vi(j);
        p=[96*u-48, 96*v-48, 144*u*v-84*u-84*v+73];
        if ((p(1)^2+p(2)^2) > 40^2)
            p(3)=NaN;
        end

        if p(3)~=0
            S((i-1)*1+j,:)=p;
            % A(cont,:)=p;
            % cont=cont+1;
        end
    end
end

nanRows = any(isnan(S), 2);
S(nanRows, :) = [];

% tam=30*ones(size(S,1),1);

% ----- surf_teo

x = S(:,1);
y = S(:,2);
z = S(:,3)+150;

```

```

% Interpola al grid

[X,Y] = meshgrid(linspace(min(x),max(x),300), linspace(min(y),max(y),300));
Z = griddata(x,y,z,X,Y);

s=surf(X,Y,Z, 'FaceColor','green', 'EdgeColor', 'none');

%----- surf herram

% -----

% clear x y z X Y Z ui vi S

grayColor = [0.3 0.3 0.3];
set(s, 'FaceColor', grayColor);

% shading interp
light          % create a light
% lighting gouraud % preferred method for lighting curved surfaces
% lightangle(70, 30)

material dull

hold on
grid on

% -----
% s=surf(X,Y,Z);
% colormap gray

% s.EdgeColor='none';

plot3(p_obj_teo(a:b,1), p_obj_teo(a:b,2), p_obj_teo(a:b,3),'.-','LineWidth',
      3,'MarkerSize', 10)

title(name_gif)

xlabel('$X(mm)$', 'interpreter', 'latex')
ylabel('$Y(mm)$', 'interpreter', 'latex')
zlabel('$Z(mm)$', 'interpreter', 'latex')

% ----- Dibujo la herramienta

% disp(vertices)

chdir('C:\Users\User\Desktop\__TFM\1-pruebas');

% disp(pwd)
[points_final, min_dz, delta_z, colisiona]=situ_herramienta(vertices,
    gbest_tot(a,:), tabla, p_obj_teo(a,:), n_obj_teo(a,:), a, 5);

% [points_final, min_dz, delta_z]=situ_herramienta(vertices, gbest_tot(a,:),
    tabla, p_obj_teo(a,:), n_obj_teo(a,:), a,5);

```

```

fig=scatter3(points_final(:,1), points_final(:,2), points_final(:,3), 10,
    delta_z, 'filled');
colormap(jet(300))
colormap(flipud(colormap));
colorbar

caxis([-1, 30]);

% ----- ajustes grafica _view_

set(gcf, 'Position', [700, 300, 900, 600]);

view(az, cen1);    % 3D
% view(az, cen1);  % XY
% view(az, cen1);  % YZ

xlim([-60,60])
ylim([-60,60])
zlim([160,230])

numTicks = 6;
xticks(linspace(-60, 60, numTicks));
yticks(linspace(-60, 60, numTicks));
zticks(linspace(160, 230, numTicks));
%----- superficie y mas -----
i=1;
frame = getframe(gcf);
% name=strcat('cnc-5-',num2str(i),'.png');
% imwrite(frame.cdata, strcat('C:\Users\User\Desktop\__TFM\4-resultados\
    images\',name));

chdir('C:\Users\User\Desktop\__TFM\4-resultados\gifs');

writerObj = VideoWriter(name_gif, 'MPEG-4');
writerObj.FrameRate = 5; % Specify the frame rate (frames per second)
open(writerObj);

min_dz_tot(min_dz_tot > 1e-4)=NaN;

legend(strcat('Rugosidad: ',num2str(min_dz_tot(a)), ' mm'))
legend('Location', 'northeast');

%
% -----

% axis tight manual
gif = 'animation.gif';

for i=a+1:b
    chdir('C:\Users\User\Desktop\__TFM\1-pruebas');

```

```

[points_final, min_dz, delta_z]=situ_herramienta(vertices, gbest_tot(i
   ,:), tabla, p_obj_teo(i,:), n_obj_teo(i,:), i,5);

% [points_final, min_dz, delta_z]=situ_herramienta(vertices, gbest_tot(i
   ,:), tabla, p_obj_teo, ind, 5);

set(fig, 'XData', points_final(:,1));
set(fig, 'YData', points_final(:,2));
set(fig, 'ZData', points_final(:,3));
set(fig, 'CData', delta_z);

min_dz_tot(min_dz_tot > 1e-4)=NaN;

legend(strcat('Rugosidad: ', num2str(min_dz_tot(i)), ' mm'))
legend('Location', 'northeast');

numTicks = 6;
xticks(linspace(-60, 60, numTicks));
yticks(linspace(-60, 60, numTicks));
zticks(linspace(160, 230, numTicks));

hold on
grid on

dis_0=dis_eu([p_obj_teo(i-1,1), p_obj_teo(i-1,2), 0], [p_obj_teo(i,1),
    p_obj_teo(i,2), 0]);

fac=1e-4;
ps=fac*dis_0^(16/9/dis_0);

% ----- GUARDO PNG y GIF

% frame = getframe(gcf);
% name=strcat('cnc-5-', num2str(i), '.png');
% imwrite(frame.cdata, strcat('C:\Users\User\Desktop\__TFM\4-resultados\
images\', name));

path='C:\Users\User\Desktop\__TFM\4-resultados\gifs';

%----- intento gif
path_antes=pwd;
chdir(path);

pw=pwd;
% disp(pwd)

drawnow

frame = getframe(gcf);
writeVideo(writerObj, frame);

end

```

```

close(writerObj);

% ----- escribir el gif

%% ----- Figura de rugosidad MIN DZ

chdir('C:\Users\User\Desktop\__TFM\1-pruebas');

% --- f3 ---

fac=0.9;

hfig=figure;

subplot(3, 3, [1 2 4 5 7 8]);

rugosidad=min_dz_tot(a:ind);

rugosidad(rugosidad > 1e-4)=NaN;

rugosidad(rugosidad > fac*mean(rugosidad))=NaN;

% rugosidad(rugosidad > 2e-8)=NaN;

scatter3(p_obj_teo(a:ind,1), p_obj_teo(a:ind,2), p_obj_teo(a:ind,3), 60,
    rugosidad, 'filled');

grid on
hold on

zeroIndices = min_dz_tot == 0;

colormap(parula(500))
colorbar

caxis([min(rugosidad), max(rugosidad)])

title('Rugosidad (mm)', 'interpreter', 'latex')

xlabel('$X(mm)$', 'interpreter', 'latex')
ylabel('$Y(mm)$', 'interpreter', 'latex')
zlabel('$Z(mm)$', 'interpreter', 'latex')

s=surf(X,Y,Z, 'FaceColor','blue', 'EdgeColor', 'none');
grayColor = [0.5 0.5 0.5];
set(s, 'FaceColor', grayColor);

axis tight

```

```

% shading interp
light          % create a light
% lighting gouraud % preferred method for lighting curved surfaces
% lightangle(70, 30)

material dull

for_fig(hfig, 20, 0.69)

%.-----

subplot(3, 3, [3 6 9]);

aa=size(rugosidad, 1);

% scatter(linspace(1,aa,aa), rugosidad)

picturewidth = 33; % set this parameter and keep it forever
hw_ratio = 0.44; % feel free to play with this ratio

set(findall(hfig, '-property', 'FontSize'), 'FontSize', 23) % adjust fontsize
    to your document

set(hfig, 'Units', 'centimeters', 'Position', [3 3 picturewidth hw_ratio*
    picturewidth])

nbins=10;
h1=histogram(rugosidad, nbins);

txt=strcat('Moda: ', num2str(mode(rugosidad)));
text(0.3*max(rugosidad), 0.9*max(h1.Values), txt)

% xticks(linspace(min(rugosidad), max(rugosidad), 10));

set(findall(hfig, '-property', 'FontSize'), 'FontSize', 18) % adjust fontsize
    to your document

set(findall(hfig, '-property', 'Box'), 'Box', 'on') % optional
set(findall(hfig, '-property', 'Interpreter'), 'Interpreter', 'latex')
set(findall(hfig, '-property', 'TickLabelInterpreter'), 'TickLabelInterpreter',
    'latex')

pos = get(subplot(3, 3, [3 6 9]), 'Position'); % Get the position of the
    second subplot
pos(3) = 0.2; % Change the width (third element of the 'Position' array)
set(subplot(3, 3, [3 6 9]), 'Position', pos); % Set the new position for the
    second subplot

view(0,90)

```

```

xlabel('Error')

% set(gca, 'Position', [ picturewidth, 0.3*picturewidth])

% pos = get(hfig,'Position');
% set(hfig,'PaperPositionMode','Auto','PaperUnits','centimeters','PaperSize',
    ',[pos(3), pos(4)]')

%----- surf herram

% -----

% clear x y z X Y Z ui vi S

frame = getframe(gcf);
name=strcat('rugosidad-',num2str(a),'-',num2str(b),'.png');
imwrite(frame2im(frame), strcat('C:\Users\User\Desktop\__TFM\4-resultados\
    images\',strcat(nom,'rugosidad.png')));

%% ----- Figura de TEO vs REAL (OPTIMIZADO)

% --- f4 ---

p_real=zeros(size(p_obj(:,1:3),1),4);

for i=1:size(p_obj,1)
    p_real(i,:)=modelo_haas(gbest_tot(i,:), tabla);
end

ax=1;
ay=1;
az=1;

error=(ax*(p_obj_teo(:,1)-p_real(:,1)).^2+ay*(p_obj_teo(:,2)-p_real(:,2))
    .^2+az*(p_obj_teo(:,3)-p_real(:,3)).^2).^0.5;
% error(zeroIndices)=0;

% disp(mean(error(a:ind)))
% error(error(a:ind) > fac*mean(error(a:ind)))=NaN;

hfig=figure;

subplot(3, 3, [1 2 4 5 7 8]);

x=p_obj_teo(a:ind,1);
y=p_obj_teo(a:ind,2);
z=p_obj_teo(a:ind,3);

vals=error(a:ind);

```

```

fac=1.6;

indo=vals<= fac*mean(vals);

scatter3(x(indo), y(indo), z(indo), 60, vals(indo), 'filled');

grid on
hold on

colormap(parula(500))
colorbar

% caxis([-fac*mean(error(a:ind)), fac*mean(error(a:ind))])

title('Teorico vs Real(Optimizado)', 'interpreter', 'latex')

caxis([min(vals(indo)), max(vals(indo))])

xlabel('$X(mm)$', 'interpreter', 'latex')
ylabel('$Y(mm)$', 'interpreter', 'latex')
zlabel('$Z(mm)$', 'interpreter', 'latex')

s=surf(X,Y,Z, 'FaceColor','blue', 'EdgeColor', 'none');
grayColor = [0.5 0.5 0.5];
set(s, 'FaceColor', grayColor);

axis tight

% shading interp
light % create a light
% lighting gouraud % preferred method for lighting curved surfaces
% lightangle(70, 30)

material dull
%----- surf herram

% -----

% clear x y z X Y Z ui vi S

for_fig(hfig, 20, 0.69)

%.-----

rugosidad=error(a:ind);

subplot(3, 3, [3 6 9]);

aa=size(rugosidad, 1);

% scatter(linspace(1,aa,aa), rugosidad)

```

```

pos = get(subplot(3, 3, [3 6 9]), 'Position'); % Get the position of the
second subplot
pos(3) = 0.2; % Change the width (third element of the 'Position' array)
set(subplot(3, 3, [3 6 9]), 'Position', pos); % Set the new position for the
second subplot

view(0,90)

picturewidth = 33; % set this parameter and keep it forever
hw_ratio = 0.44; % feel free to play with this ratio

set(hfig, 'Units', 'centimeters', 'Position', [3 3 picturewidth hw_ratio*
picturewidth])

nbins=20;
histogram(rugosidad, nbins)

h1=histogram(rugosidad, nbins);

txt=strcat('Moda: ', num2str(mode(rugosidad)));
text(0.3*max(rugosidad), 0.9*max(h1.Values), txt)

xlabel('Error')

set(findall(hfig, '-property', 'FontSize'), 'FontSize', 18) % adjust fontsize
to your document

set(findall(hfig, '-property', 'Box'), 'Box', 'on') % optional
set(findall(hfig, '-property', 'Interpreter'), 'Interpreter', 'latex')
set(findall(hfig, '-property', 'TickLabelInterpreter'), 'TickLabelInterpreter',
'latex')

frame = getframe(gcf);
imwrite(frame2im(frame), strcat('C:\Users\User\Desktop\__TFM\4-resultados\
images\', strcat(nom, 'optimizado.png')));

% caxis([0, 1e-6]);
%
%
% title('$CAM vs REAL (Optimizado)$', 'interpreter', 'latex')
%
% xlabel('$X(mm)$', 'interpreter', 'latex')
% ylabel('$Y(mm)$', 'interpreter', 'latex')
% zlabel('$Z(mm)$', 'interpreter', 'latex')
%
% % numTicks = 6;
% % xticks(linspace(-60, 60, numTicks));
% % yticks(linspace(-60, 60, numTicks));
% % zticks(linspace(160, 230, numTicks));
%
% frame = getframe(gcf);
% name=strcat('CAM vs REAL (Optimizado)-', num2str(a), '-', num2str(b), '.png');
% imwrite(frame2im(frame), strcat('C:\Users\User\Desktop\__TFM\4-resultados\
images\', name));
%

```

```

% % --- f6 ---
%
%
% p_real=zeros(size(p_obj(:,1:3),1),4);
%
% for i=1:size(p_obj,1)
%     p_real(i,:)=modelo_haas(gbest_tot(i,:), tabla);
% end
%
% ax=1;
% ay=1;
% az=1;
%
% error=(ax*(obj_tot(:,1)-p_real(:,1)).^2+ay*(obj_tot(:,2)-p_real(:,2)).^2+
%     az*(obj_tot(:,3)-p_real(:,3)).^2).^0.5;
% % error(zeroIndices)=0;
%
%
%
% figure
% fig=scatter3(p_obj_teo(a:ind,1), p_obj_teo(a:ind,2), p_obj_teo(a:ind,3),
%     20, min_dz_tot(a:ind), 'filled');
%
% grid on
% hold on
%
% colormap parula
% colorbar
%
% title('$CAM vs REAL (Optimizado)$', 'interpreter', 'latex')
%
% xlabel('$X(mm)$', 'interpreter', 'latex')
% ylabel('$Y(mm)$', 'interpreter', 'latex')
% zlabel('$Z(mm)$', 'interpreter', 'latex')
%
% % numTicks = 6;
% % xticks(linspace(-60, 60, numTicks));
% % yticks(linspace(-60, 60, numTicks));
% % zticks(linspace(160, 230, numTicks));
%
% frame = getframe(gcf);
% name=strcat('TEO vs REAL (Optimizado)-',num2str(a),'-',num2str(b),'.png');
% imwrite(frame2im(frame), strcat('C:\Users\User\Desktop\__TFM\4-resultados\
%     images\' ,name));
%%
% picturewidth = 20; % set this parameter and keep it forever
% hw_ratio = 0.69; % feel free to play with this ratio
% set(findall(hfig,'-property','FontSize'),'FontSize',17) % adjust fontsize
%     to your document
%
% set(findall(hfig,'-property','Box'),'Box','on') % optional
% set(findall(hfig,'-property','Interpreter'),'Interpreter','latex')
% set(findall(hfig,'-property','TickLabelInterpreter'),'TickLabelInterpreter
%     ','latex')
% set(hfig,'Units','centimeters','Position',[3 3 picturewidth hw_ratio*
%     picturewidth])
% pos = get(hfig,'Position');

```

```

% set(hfig,'PaperPositionMode','Auto','PaperUnits','centimeters','PaperSize',
    ',[pos(3), pos(4)]')
% print(hfig,fname,'-dpdf','-painters','-fillpage')
% print(hfig,fname,'-dpng','-painters')

%%

hfig=figure;

plot3(n_obj_teo(:,1), n_obj_teo(:,2), n_obj_teo(:,3), 'Linewidth', 2)

grid on
hold on

title('Normales teoricas [i, j, k]', 'interpreter', 'latex')

% axis equal

xlabel('$i$', 'interpreter', 'latex')
ylabel('$j$', 'interpreter', 'latex')
zlabel('$k$', 'interpreter', 'latex')

for_fig(hfig, 20, 0.7)

```

A.4.2. Función PSO general

```

function [gbest, error_real, choque_obj, error_ac, error_aca, particula]=
    pso_general(obj_tot, n_obj_teo, ind, sol_inicial, tabla, vertices)
%% ##### PSO #####

% Tenemos 5 parametros [x,y,z,tx,tz]

r=5;

rng('shuffle');

n_obj=n_obj_teo(ind,:);
obj=obj_tot(ind,:);

cost_func=@(p_obj, n_obj, X, vertices, r, tabla, sol_inicial, ind) coste(
    p_obj, n_obj, X, vertices, r, tabla, sol_inicial, ind);

niter=100;
npar=70;

w=0.5;
c1=0.3;
c2=2;

if ind>=10
    npar=40;

```

```

end

% w=1;
% c1=1.2;
% c2=1.5;

%% Punto(s) objetivo

% ind=700;

% tx=0;
% tz=0;

% sol_inicial=[obj(1),obj(2),obj(3) ,tx,tz];

% sol_inicial=[0,0,0,0,0];

costo=5000;

dxyz=70;
dac=85;

% dxyz=20;
% dac=25;

deltas_join=[dxyz, dxyz, dxyz, dac, dac];

lb=sol_inicial - deltas_join;
ub=sol_inicial + deltas_join;

% lb=[-70, -70, -70, 0, 0];
% ub=-[70, 70, 70, 999, 999];

%% EL STL leerlo siempre en la funcion m s global

% [vertices] = stlread("Herramienta-quad-hibrid.STL");

r=5;
% pto_rotacion=obj;

%% ##### Inicializacion #####

% if punto==1

%     particula.pos=[];
%     particula.vel=[];
%     particula.cost=[];
%     particula.best.pos=[];
%     particula.best.cost=[];
%     particula = repmat(particula, npar, 1);

```

```

% if punto==2
%     particula.vel=[];
% end

clear particula

gbest.pos=sol_inicial;
gbest.cost=costo;

particula.pos=[];
particula.vel=[];
particula.cost=[];
particula.best.pos=[];
particula.best.cost=[];
particula = repmat(particula, npar, 1);

for i=1:npar
    a=1e-2;
    b=1e-2;

    % por defecto coger sol_inicial
    psox=sol_inicial(1)+a*unifrnd(lb(1),ub(1),1);
    psoy=sol_inicial(2)+a*unifrnd(lb(2),ub(2),1);
    psoz=sol_inicial(3)+a*unifrnd(lb(3),ub(3),1);
    tx=sol_inicial(4)+b*unifrnd(lb(4),ub(4),1);
    tz=sol_inicial(5)+b*unifrnd(lb(5),ub(5),1);

    %     tx=0;
    %     tz=0;

    % caso de que sean todos ceros randomizar

    if sum(sol_inicial)==0
        a=30;
        b=20;

        % por defecto coger sol_inicial
        psox=sol_inicial(1)+a*unifrnd(lb(1),ub(1),1);
        psoy=sol_inicial(2)+a*unifrnd(lb(2),ub(2),1);
        psoz=sol_inicial(3)+a*unifrnd(lb(3),ub(3),1);
        tx=sol_inicial(4)+b*unifrnd(lb(4),ub(4),1);
        tz=sol_inicial(5)+b*unifrnd(lb(5),ub(5),1);
    end

    particula(i).pos=[psox, psoy, psoz, tx, tz];

    particula(i).vel=0.2*particula(i).pos;
    %     particula(i).cost=cost_func(obj, particula(i).pos, lb, ub);
    [particula(i).cost, error_real, choque_obj, error_ac, error_aca]=
        cost_func(obj, n_obj, particula(i).pos, vertices, r, tabla,
            sol_inicial, ind);
    particula(i).best.pos=particula(i).pos;
    particula(i).best.cost=particula(i).cost;

    % Buscar el global

```

```

        if particula(i).best.cost < gbest.cost
            gbest.cost=particula(i).best.cost;
            gbest.pos=particula(i).best.pos;
        end
    end

%% ##### Loop del PSO total
%% #####

% tolerancias
j=1;
tol=10^-5;

tol_best=tol;

% gbest_tot=[];

coo=0;

gbest_test=0;

psi=c1+c2;
xi=2/abs(2-psi-sqrt(psi^2-4*psi));

niter_t=niter;

disp(j<=niter_t && tol_best>=tol)

while j<=niter_t && tol_best>=tol
    for i=1:npar

        if gbest_test>=1e4

            w=0.7;
            c1=0.1;
            c2=2;

            b=1.6;
            a=1.4;

            psox=sol_inicial(1)+a*unifrnd(lb(1),ub(1),1);
            psoy=sol_inicial(2)+a*unifrnd(lb(2),ub(2),1);
            psoz=sol_inicial(3)+a*unifrnd(lb(3),ub(3),1);
            tx=sol_inicial(4)+b*unifrnd(lb(4),ub(4),1);
            tz=sol_inicial(5)+b*unifrnd(lb(5),ub(5),1);

            %         tx=0;
            %         tz=0;

            particula(i).pos=[psox, psoy, psoz, tx, tz];
            particula(i).vel=0.5*particula(i).pos;

            %         if conta>=0.5*npar*niter
            %             gbest_test=0;

```

```

%         end
        gbest_test=0;

        niter_t=20+niter;
end

particula(i).vel= xi*(w*particula(i).vel + c1*rand([1, 5]).*(
    particula(i).best.pos- particula(i).pos) + c2*rand([1, 5]).*(
    gbest.pos-particula(i).pos));

particula(i).pos= particula(i).pos + particula(i).vel;

% reduccion de la velocidad para los angulos

%         kk=0.4;

%         particula(i).vel(4:5)=kk*particula(i).vel(4:5);

% se imponen las constantes

%         tx=unifrnd(lb(4),ub(4),1);
%         tz=unifrnd(lb(5),ub(5),1);

%         particula(i).pos(4)=tx;
%         particula(i).pos(5)=tz;

% Actualizar el coste y pos de cada particula

%         particula(i).cost=cost_func(obj, particula(i).pos, lb, ub);
[particula(i).cost, error_real, choque_obj, error_ac, error_aca]=
    cost_func(obj , n_obj, particula(i).pos, vertices, r, tabla,
    sol_inicial, ind);

if particula(i).cost < particula(i).best.cost
    particula(i).best.cost=particula(i).cost;
    particula(i).best.pos=particula(i).pos;
end

% Actualizar marca global

if particula(i).best.cost < gbest.cost
    gbest.cost=particula(i).best.cost;
    gbest.pos=particula(i).best.pos;
end

if mod(j, 10)==0
    gbest_test=gbest_test+gbest.cost;
end

```

```

%           gbest_test=gbest.cost;

disp([num2str(ind),num2str(j),' -> Cost: ', num2str(gbest.cost),'
      --> X: ', string(gbest.pos)])
end

% Se reduce la inercia en 95%

w=0.4*w;

%       wmax=0.9;
%       wmin=0.2;
%       w=wmax-(wmax-wmin)*j/niter;

% Guardar el gbest en cada iter

%       gbest_tot(j).cost=gbest.cost;
%       gbest_tot(j).pos=gbest.pos;
%
tol_best=gbest.cost;

j=j+1;
end

% toc
end

```

A.4.3. Función Modelo cinemático HAAS

```

function [T_P, T_T] = modelo_haas(pto_nominal, tabla)

%       pto_nominal=p_post(1,:);

%% Factores para cambiar la magnitud e influencia de los errores

ff=0.5;

ffx=1;
ffy=1;
ffz=1;
ffa=1;
ffc=1;

struct2cell(tabla);
%Posiciones nominales de los ejes de la MH
x = pto_nominal(1);
y = pto_nominal(2);
z = pto_nominal(3);
A=deg2rad(pto_nominal(4));
C=deg2rad(pto_nominal(5));

A=A+ deg2rad(0);

C=C+ deg2rad(0);

```

```

%Errores de perpendicularidad de la MH
xPy = tabla.Tbl_xPy*1;
xPz = tabla.Tbl_xPz*1;
yPz = tabla.Tbl_yPz*1;

%Matrices de rotaci n del modelo de error de la MH
xRx = ffx*ff*interp_T(tabla.Tbl_xRx, x);
xRy = ffx*ff*interp_T(tabla.Tbl_xRy, x);
xRz = ffx*ff*interp_T(tabla.Tbl_xRz, x);

%      R_X = [1 -xRz xRy; xRz 1 -xRx; -xRy xRx 1];

iR_x = [1 xRz -xRy 0;
        -xRz 1 xRx 0;
        xRy -xRx 1 0;
        0 0 0 1];%inversa de la matriz

yRx = ffy*ff*interp_T(tabla.Tbl_yRx, y);
yRy = ffy*ff*interp_T(tabla.Tbl_yRy, y);
yRz = ffy*ff*interp_T(tabla.Tbl_yRz, y);

%      R_Y= [1 -yRz yRy; yRz 1 -yRx; -yRy yRx 1];

iR_y = [1 yRz -yRy 0;
        -yRz 1 yRx 0;
        yRy -yRx 1 0;
        0 0 0 1];%inversa de la matriz

zRx = ffz*ff*interp_T(tabla.Tbl_zRx, z);
zRy = ffz*ff*interp_T(tabla.Tbl_zRy, z);
zRz = ffz*ff*interp_T(tabla.Tbl_zRz, z);

%      R_z = [1 -zRz zRy; zRz 1 -zRx; -zRy zRx 1];

iR_z = [1 zRz -zRy 0;
        -zRz 1 zRx 0;
        zRy -zRx 1 0;
        0 0 0 1];%inversa de la matriz

%Vectores de traslaci n del modelo de error de la MH
xTx = ffx*ff*interp_T(tabla.Tbl_xTx, x);
xTy = ffx*ff*interp_T(tabla.Tbl_xTy, x);
xTz = ffx*ff*interp_T(tabla.Tbl_xTz, x);

%      T_X = [-x+xTx; xTy-x*xPy; xTz-x*xPz];

yTx = ffy*ff*interp_T(tabla.Tbl_yTx, y);
yTy = ffy*ff*interp_T(tabla.Tbl_yTy, y);
yTz = ffy*ff*interp_T(tabla.Tbl_yTz, y);

%      T_Y = [yTx; -y+yTy; yTz-y*yPz];

zTx = ffz*ff*interp_T(tabla.Tbl_zTx, z);
zTy = ffz*ff*interp_T(tabla.Tbl_zTy, z);
zTz = ffz*ff*interp_T(tabla.Tbl_zTz, z);

```

```

%      T_Z = [zTx; zTy; -z+zTz];

%%  modeloooooooooooo

%% Matrices de rotacion y Traslacion

Tx=[1, 0, 0, x+Tx;
    0, 1, 0, xTy-x*xPy;
    0, 0, 1, xTz-x*xPz;
    0, 0, 0, 1];

Ty=[1, 0, 0, yTx;
    0, 1, 0, yTy;
    0, 0, 1, yTz-y*yPz;
    0, 0, 0, 1];

R_A = [1, 0, 0, 0;
        0, cos(A), -sin(A), 0;
        0, sin(A), cos(A), 0;
        0, 0, 0, 1];

R_C = [cos(-C), -sin(-C), 0, 0;
        sin(-C), cos(-C), 0, 0;
        0, 0, 1, 0;
        0, 0, 0, 1];

Tz=[1, 0, 0, zTx;
    0, 1, 0, zTy;
    0, 0, 1, z+zTz;
    0, 0, 0, 1];

%% Errores de Giro

th_za=ffa*e_sen(1e-4, A, 0, 0);
th_ya=ffa*e_sen(1e-4, A, 0, 0);
eya=ffa*e_sen(1e-4, A, 45, 1e-2);
eza=ffa*e_sen(1e-4, A, 45, 1e-2);

th_yc=ffc*e_sen(1e-4, C, 0, 0);
th_xc=ffc*e_sen(1e-4, C, 0, 0);
exc=ffc*e_sen(1e-4, C, 45, 1e-2);
eyc=ffc*e_sen(1e-4, C, 45, 1e-2);

%      th_za=e_sen(1e-4, A, 30, 0);
%      th_ya=e_sen(1e-4, A, 30, 0);
%      eya=e_sen(1e-4, A, 0, 10e-2);
%      eza=e_sen(1e-4, A, 0, 10e-2);
%
%      th_yc=e_sen(1e-4, C, 50, 0);
%      th_xc=e_sen(1e-4, C, 50, 0);
%      exc=e_sen(1e-4, C, 55, 10e-2);
%      eyc=e_sen(1e-4, C, 0, 10e-2);

```

```

E_A = [1, -th_za, th_ya, 0;
        th_za, 1, 0, eya;
        -th_ya, 0, 1, eza;
        0, 0, 0, 1];

E_C = [1, 0, th_yc, exc;
        0, 1, -th_xc, eyc;
        -th_yc, th_xc, 1, 0;
        0, 0, 0, 1];

% E_A=eye(4);

% E_C=eye(4);

% M=Tz*(R_C)*(R_B)*(Txy);

% M=Tz*R_C*(Txy);

% M=Tz*R_C*Ty*R_A*(Tx);

% M=iR_z*E_C*Tz*R_C*iR_y*Ty*E_A*R_A*iR_x*Tx;

% M=iR_z*Tz*E_C*R_C*iR_y*Ty*E_A*R_A*iR_x*Tx;

%%% ----- Funciona orienta

% M=Tz*inv(R_C)*inv(R_A)*Ty*Tx;

% M=inv(R_C)*inv(R_A)*Ty*Tx*Tz;

M=E_C*(R_C')*E_A*(R_A')*iR_y*Ty*iR_x*Tx*iR_z*Tz;

% disp(R_C)
% disp(R_A)

T_P=M(:,4);

T_T=M(:,3);

% T_T(1)=-T_T(1);

% pn=(fke(pto_nominal));

% disp(T_P)
% disp(pn)

end

```

A.4.4. Función Mover herramienta

```

% close all

function points_final=mover_herramienta(points, pto_rotacion, dx, dy, dz,
    theta_x, theta_z)

% points = [0,0,0; 10, 20, 0];
% pto_rotacion = [0, 0, 0];
% dx=10;
% dy=20;
% dz=30;
%
% theta_x = 90;
% theta_z = 90;

one=ones(size(points,1),1);
points=[points(:,1)+dx*one, points(:,2)+dy*one, points(:,3)+dz*one];

% disp(points)

theta_x_rad = deg2rad(theta_x);
theta_z_rad = deg2rad(theta_z);

% Matrices de rotacion

% theta_x_rad=theta_x_rad;
theta_z_rad=-theta_z_rad;

R_x = [1 0 0;
    0 cos(theta_x_rad) -sin(theta_x_rad);
    0 sin(theta_x_rad) cos(theta_x_rad)];

R_z = [cos(theta_z_rad) -sin(theta_z_rad) 0;
    sin(theta_z_rad) cos(theta_z_rad) 0;
    0 0 1];

% Se desplazan al centro de giro

ptos_centrado = points - pto_rotacion;

% % Roto en x
% ptos_rotados_x = ptos_centrado * inv(R_x);
%
% % Roto en z
% ptos_rotados_xz = ptos_rotados_x * inv(R_z);

% Roto en x
ptos_rotados_x = inv(R_x)*transpose(ptos_centrado);

% Roto en z
ptos_rotados_xz = inv(R_z)*(ptos_rotados_x) ;

% Se desplazan a donde estaban

```

```

points_final = ptos_rotados_xz + transpose(pto_rotacion);

% % se visualiza
% scatter3(points(:,1), points(:,2), points(:,3), 'b', 'filled'); % Original
    points in blue
% hold on;
% scatter3(points_final(:,1), points_final(:,2), points_final(:,3), 'r', '
    filled'); % Rotated points in red
% legend('Original Points', 'Rotated Points');
% xlabel('X');
% ylabel('Y');
% zlabel('Z');

points_final=transpose(points_final);

% disp(R_z*R_x);
end

```

A.4.5. Función Check colisiones

```

function [colisiona, min_dz, XY, delta_z]=check_colisiones(points_final,
    p_obj_teo, C, r)

% Chequeo para conjunto de puntos
% p_obj=P(ind, :);
% C=Q(ind, :);
% r=5;

tol=1e-4;

z_herramienta=points_final(:,3);
pboi=pboide(points_final(:,1), points_final(:,2));

z_pboi=pboi(:,3);

indo=pboi(:,3)==0;

delta_z=z_herramienta-z_pboi;

% disp(delta_z)

delta_z(indo)=500;

% Chequeo de 4 puntos con la esfera

delta=1e-5;

cx=p_obj_teo(1);
cy=p_obj_teo(2);
n=3;

XY=malla(cx, cy, n, delta);

% disp(XY)

z_esf=zeros(size(XY,1), 1);
zz_esf=pboide(XY(:,1), XY(:,2));

```

```

% % disp(zz_esf)

z_pboi_esf=zz_esf(:,3);

for i=1:size(XY,1)
    p_es=esfera(XY(i,1), XY(i,2), r, C);
    z_esf(i)=p_es(3);
end

delta_z_esf=z_esf-z_pboi_esf;

% disp(min( delta_z))

join_z=[delta_z; delta_z_esf];

% join_z=[delta_z];
min_dz=min(join_z);

colisiona=0;

if min_dz<0
    colisiona=1;
end

if min_dz>0
    colisiona=0;
end
end

```

A.4.6. Función Coste colisiones

```

function [coste_col, min_dz]=coste_colisiones(X, obj, n_obj_teo, vertices, r
    , tabla, ind)

tol=1e-4;

% C=[X(1), X(2), X(3)]+ 5*n_obj_teo;
% ptos_1=[vertices.Points; 0, 0, 1];
% points_final=mover_herramienta(ptos_1, C, C(1), C(2), C(3), X(4), X(5));
% [colisiona, min_dz, ~]=check_colisiones(points_final, obj, C, r);

% [~, min_dz, ~]=situ_herramienta(vertices, X, tabla, obj, n_obj_teo, ind,
    5);

[points_final, min_dz, delta_z, colisiona]=situ_herramienta(vertices, X,
    tabla, obj, n_obj_teo, ind, 5);

% if mod(ind, 10)==0
% disp(['compro_',num2str(colisiona)])
% end

coste_col=0;

if (colisiona==1) && (abs(min_dz)<tol)
    coste_col=0;
end

```

```

if (colisiona==1) && (abs(min_dz)>tol)
%     coste_col=abs(min_dz);
    coste_col=100;
end

% if (colisiona==1) && (abs(min_dz)<tol)
%     coste_col=0;
% end
%
% if (colisiona==1) && (abs(min_dz)>tol)
% %     coste_col=abs(min_dz)*10;
%     coste_col=100;
% end

end

```

A.4.7. Función Coste total

```

function [error_tot, error_real, choque_obj, error_ac, error_aca]=coste(obj,
    n_obj_teo, X, vertices, r, tabla, sol_inicial, ind)

% pto_rotacion=X(1:3);

% suavizado en A C

error_aca=abs(X(4:5)-sol_inicial(4:5));

error_ac=sum((error_aca.^2).^0.5);

% dt_ac=15;
% dt_xyz=10;

% if (error_aca(4)>= dt_ac) || (error_aca(5) >= dt_ac)
%     error_ac=200;
% end
%
% if (error_aca(1)>= dt_xyz) || (error_aca(2) >= dt_xyz) || (error_aca(3) >=
    dt_xyz)
%     error_ac=200;
% end

p_mod = modelo_haas(X, tabla);

A=[p_mod(1), p_mod(2), p_mod(3)];

##### Que no choque en la posicion objetivo

Q_cen=A+ 5*n_obj_teo;

% disp([num2str(A), ' : ', num2str(Q_cen), ' .. ', num2str(n_obj_teo)])

% p_obj_choque=[obj(1), obj(2), obj(3), X(4), X(5)];

```

```

p_obj_choque=[Q_cen(1), Q_cen(2), Q_cen(3), X(4), X(5)];

% [choque_obj, min]=coste_colisiones(p_obj_choque, obj, n_obj_teo, vertices,
    r, tabla, ind);

[points_final, min_dz, delta_z, colisiona]=situ_herramienta(vertices, X,
    tabla, obj, n_obj_teo, ind, 5);

choque_obj=10*colisiona;

%% ##### Suma de errores #####

error_real=dis_eu(obj, p_mod(1:3));

% error_tot=sqrt(1e3*error_real^2 + 1000*choque_mod^2 + 1000*choque_obj^2);
% error_tot=(1e3*error_real^2 + 100*choque_obj^2 + 1e-2*error_ac^2)^0.5;
error_tot=(1e3*error_real^2 + 100*choque_obj^2 + 0*error_ac)^0.5;

end

```

A.4.8. Función Gráficas errores cinemáticos

```

% struct2cell(tabla);
% Posiciones nominales de los ejes de la MH

% x = pto_nominal(1);
% y = pto_nominal(2);
% z = pto_nominal(3);
% A=deg2rad(pto_nominal(4));
% C=deg2rad(pto_nominal(5));

syms x y z A C

% Errores de perpendicularidad de la MH

% xPy = tabla.Tbl_xPy*1;
% xPz = tabla.Tbl_xPz*1;
% yPz = tabla.Tbl_yPz*1;

syms xPy xPz yPz

% Matrices de rotación del modelo de error de la MH

% xRx = interp_T(tabla.Tbl_xRx, x);
% xRy = interp_T(tabla.Tbl_xRy, x);

```

```

% xRz = interp_T(tabla.Tbl_xRz, x);

syms xRx xRy xRz

%-----

R_X = [1 -xRz xRy; xRz 1 -xRx; -xRy xRx 1];

iR_x = [1 xRz -xRy 0;
        -xRz 1 xRx 0;
        xRy -xRx 1 0;
        0 0 0 1];%inversa de la matriz

% yRx = interp_T(tabla.Tbl_yRx, y);
% yRy = interp_T(tabla.Tbl_yRy, y);
% yRz = interp_T(tabla.Tbl_yRz, y);

syms yRx yRy yRz

R_Y = [1 -yRz yRy; yRz 1 -yRx; -yRy yRx 1];

iR_y = [1 yRz -yRy 0;
        -yRz 1 yRx 0;
        yRy -yRx 1 0;
        0 0 0 1];%inversa de la matriz

% zRx = interp_T(tabla.Tbl_zRx, z);
% zRy = interp_T(tabla.Tbl_zRy, z);
% zRz = interp_T(tabla.Tbl_zRz, z);

syms zRx zRy zRz

R_z = [1 -zRz zRy; zRz 1 -zRx; -zRy zRx 1];

iR_z = [1 zRz -zRy 0;
        -zRz 1 zRx 0;
        zRy -zRx 1 0;
        0 0 0 1];%inversa de la matriz

% Vectores de traslaci n del modelo de error de la MH

% xTx = interp_T(tabla.Tbl_xTx, x);
% xTy = interp_T(tabla.Tbl_xTy, x);
% xTz = interp_T(tabla.Tbl_xTz, x);

syms xTx xTy xTz

T_X = [-x+xTx; xTy-x*xPy; xTz-x*xPz];

% yTx = interp_T(tabla.Tbl_yTx, y);

```

```
% yTy = interp_T(tabla.Tbl_yTy, y);
% yTz = interp_T(tabla.Tbl_yTz, y);
```

```
syms yTx yTy yTz
```

```
T_Y = [yTx; -y+yTy; yTz-y*yPz];
```

```
% zTx = interp_T(tabla.Tbl_zTx, z);
% zTy = interp_T(tabla.Tbl_zTy, z);
% zTz = interp_T(tabla.Tbl_zTz, z);
```

```
syms zTx zTy zTz
```

```
T_Z = [zTx; zTy; -z+zTz];
```

```
% modeloooooooooooo
```

```
% Matrices de rotacion y Traslacion
```

```
Tx=[1, 0, 0, x+xTx;
    0, 1, 0, xTy-x*xPy;
    0, 0, 1, xTz-x*xPz;
    0, 0, 0, 1];
```

```
Ty=[1, 0, 0, yTx;
    0, 1, 0, yTy;
    0, 0, 1, yTz-y*yPz;
    0, 0, 0, 1];
```

```
R_A = [1, 0, 0, 0;
        0, cos(A), -sin(A), 0;
        0, sin(A), cos(A), 0;
        0, 0, 0, 1];
```

```
R_C = [cos(C), -sin(C), 0, 0;
        sin(C), cos(C), 0, 0;
        0, 0, 1, 0;
        0, 0, 0, 1];
```

```
Tz=[1, 0, 0, zTx;
    0, 1, 0, zTy;
    0, 0, 1, z+zTz;
    0, 0, 0, 1];
```

```
% Errores de Giro
```

```
% th_za=e_sen(1e-4, A, 0, 0);
% th_ya=e_sen(1e-4, A, 0, 0);
% eya=e_sen(1e-4, A, 0, 1e-2);
```

```

% eza=e_sen(1e-4, A, 0, 1e-2);
%
% th_yc=e_sen(1e-4, C, 0, 0);
% th_xc=e_sen(1e-4, C, 0, 0);
% exc=e_sen(1e-4, C, 0, 1e-2);
% eyc=e_sen(1e-4, C, 0, 1e-2);

syms th_za th_ya eya eza

syms th_yc th_xc exc eyc

E_A = [1, -th_za, th_ya, 0;
        th_za, 1, 0, eya;
        -th_ya, 0, 1, eza;
        0, 0, 0, 1];

E_C = [1, 0, th_yc, exc;
        0, 1, -th_xc, eyc;
        -th_yc, th_xc, 1, 0;
        0, 0, 0, 1];

M=iR_z*Tz*E_C*R_C*iR_y*Ty*E_A*R_A*iR_x*Tx;

disp(simplify(M(:,4)))

arr=latex(M(:,4));

%% ----- creacion de graficas de errores

tabla=load('Tabla errores MH Fexac.mat');

na=fieldnames(tabla);

names=string(na);

part=split(names(2),'_');

ind_R=[2,3,4,8,9,10,14,15,16];

ind_T=[5,6,7,11,12,13,17,18,19];

%----- figura de errores en xy
hfig=figure;

for i=1:size(ind_R,2)
    part=split(names(ind_R(i)),'_');
    err=part(2);

    xy=linspace(-40, 200, 400);
    a = interp_T(tabla.(names(ind_R(i))), xy);

    plot(xy, a, 'DisplayName', err, 'LineWidth', 2)

```

```

        hold on
        grid on
end

title('Errores de rotacion (R)')
legend('Location', 'eastoutside')
xlabel('X, Y, Z (mm)')
ylabel('Error (mm)')

for_fig(hfig, 20, 0.69)
view(0,90)

picturewidth = 30; % set this parameter and keep it forever
hw_ratio = 0.5; % feel free to play with this ratio
set(findall(hfig, '-property', 'FontSize'), 'FontSize', 23) % adjust fontsize
    to your document

set(findall(hfig, '-property', 'Box'), 'Box', 'on') % optional
set(findall(hfig, '-property', 'Interpreter'), 'Interpreter', 'latex')
set(findall(hfig, '-property', 'TickLabelInterpreter'), 'TickLabelInterpreter',
    'latex')
set(hfig, 'Units', 'centimeters', 'Position', [3 3 picturewidth hw_ratio*
    picturewidth])

%%----- Errores Lineales T

tabla=load('Tabla errores MH Fexac.mat');

na=fieldnames(tabla);

names=string(na);

part=split(names(2), '_');

ind_R=[2,3,4,8,9,10,14,15,16];

ind_T=[5,6,7,11,12,13,17,18,19];

%----- figura de errores en xy
hfig=figure;

for i=1:size(ind_T,2)
    part=split(names(ind_T(i)), '_');
    err=part(2);

    xy=linspace(-40, 200, 400);
    a = interp_T(tabla.(names(ind_T(i))), xy);

    plot(xy, a, 'DisplayName', err, 'LineWidth', 2)
    hold on
    grid on
end

title('Errores lineales (T)')

```

```

legend('Location', 'eastoutside')
xlabel('X, Y, Z (mm)')
ylabel('Error (mm)')

for_fig(hfig, 20, 0.69)
view(0,90)

picturewidth = 30; % set this parameter and keep it forever
hw_ratio = 0.5; % feel free to play with this ratio
set(findall(hfig, '-property', 'FontSize'), 'FontSize', 23) % adjust fontsize
    to your document

set(findall(hfig, '-property', 'Box'), 'Box', 'on') % optional
set(findall(hfig, '-property', 'Interpreter'), 'Interpreter', 'latex')
set(findall(hfig, '-property', 'TickLabelInterpreter'), 'TickLabelInterpreter',
    'latex')
set(hfig, 'Units', 'centimeters', 'Position', [3 3 picturewidth hw_ratio*
    picturewidth])

% ----- figuras errores de giro

hfig=figure;

xx=linspace(0, 360, 500);

th_yi=e_sen(1e-4, xx, 0, 0);
e_yi=e_sen(1e-4, xx, 45, 1e-2);

yyaxis left
plot(xx, th_yi, 'LineWidth', 2, 'DisplayName', '$ \theta_{i0j}$')
hold on
grid on

xlabel('A, C (mm)')
ylabel('Error (mm)')

yyaxis right
plot(xx, e_yi, 'LineWidth', 2, 'DisplayName', '$ e_{i0j}$')

axis tight

title('Errores debidos a rotaciones')

legend
for_fig(hfig, 30, 0.5)

```

A.4.9. Función Pruebas

```

close all

chdir('C:\Users\User\Desktop\__TFM\1-pruebas');
tabla=load('Tabla errores MH Fexac.mat');

```

```

path='C:\Users\User\Desktop\__TFM\4-resultados\images\fin';

tic

set(groot, 'defaultTextInterpreter', 'latex');
% set(groot, 'defaultAxesTickLabelInterpreter', 'default');
% set(groot, 'defaultColorbarTickLabelInterpreter', 'default');

set(groot, 'defaultAxesFontSize', 10)
set(groot, 'defaultLineLineWidth', 1)
set(groot, 'defaultLineMarkerSize', 7)

ui= 0:0.01:1;
vi= 0:0.01:1;
l=length(ui);
S = zeros(l*1,3);

cont=1;
% A=zeros(1,3);

for i=1:l
    u=ui(i);
    for j=1:l
        v=vi(j);
        p=[96*u-48, 96*v-48, 144*u*v-84*u-84*v+73];
        if ((p(1)^2+p(2)^2) > 40^2)
            p(3)=NaN;
        end

        if p(3)~=0
            S((i-1)*l+j,:)=p;
            % A(cont,:)=p;
            % cont=cont+1;
        end
    end
end

% rows_with_nan = any(isnan(S), 2);
%
% S(rows_with_nan, :) = [];

% writematrix(S,'S.csv');

S(:,3)=S(:,3)+150;

%% ##### Leo los puntos CAM y aplico ModeloMH
FEXAC
chdir('C:\Users\User\Desktop\__TFM\1-pruebas');

% --- f1

hfig=figure;

plot3(S(:,1),S(:,2),S(:,3), ' .', 'MarkerSize', 10, 'DisplayName', 'Teorica')
hold on

```

```

axis equal

% gm = importGeometry("paraboloide-solid-o.STL");

%----- 3 ejes----- pdegplot(gm)

% p_obj=xlsread('tray-goto-3-ejes-finn.xlsx');
% p_obj=xlsread('tray-goto-3-ejes-finn.xlsx');

% p_obj=p_obj(10:10:end, :);
% load p_surf_3_ejes.mat

%----- 5 ejes ----- pdegplot(gm)

% p_obj=xlsread('p_haas_goto_tra_2.xlsx');
% p_post=xlsread('p_haas_post_tra_2.xlsx');
% load p_surf_tray_2.mat

% ----- HAAS

% load p_surf_5_ejes.mat
% p_surf=p_surf(2:end-1, :);
% p_post=xlsread('p_haas_post_ejes.xlsx');
% p_obj=xlsread('p_haas_goto_ejes.xlsx');

tabla=load('Tabla errores MH Fexac.mat');

% excluyo el primer y ultimo dato ver los mas cercanos a

% p_obj=p_obj(2:end-10, :);
% p_post=p_post(2:end-10, :);

% p_post=xlsread('post_HAAS_true.xlsx');

p_post=xlsread('post_review_true.xlsx');

p_obj=xlsread('goto_HAAS_true.xlsx');

tabla=load('Tabla errores MH Fexac.mat');

% excluyo el primer y ultimo dato ver los mas cercanos a

p_obj=p_obj(2:end-14, :);
p_post=p_post(2:end-14, :);

tx=0;
tz=0;
% L=1000;

p_real=zeros(size(p_obj(:,1:3),1),4);

for i=1:size(p_obj,1)

```

```

        p_real(i,:)=modelo_haas(p_post(i,:), tabla);
end

a=1;
b=size(p_obj,1);

% p_surf=pboide(x(1),x(2));
%
% dista=dis_eu(p_surf, pp);

plot3(p_obj(a:b,1), p_obj(a:b,2), p_obj(a:b,3),'.-','LineWidth',2, '
    DisplayName', 'Goto')

hold on

% plot3(p_post(a:b,1), p_post(a:b,2), p_post(a:b,3),'.-','LineWidth',2)

legend

ptos_girados=zeros(size(p_post,1),4);

for i=1:size(p_post,1)

    %     pto=fke(p_post(i,:));

    [pto, ori]=modelo_haas(p_post(i,:), tabla);
    pto=transpose(pto);

    ptos_girados(i, 1:4)=pto;
end

plot3(ptos_girados(2:end-4,1), ptos_girados(2:end-4,2), ptos_girados(2:end
    -4,3),'.-','LineWidth', 2,'MarkerSize', 10, 'DisplayName', 'M(post)');

% p_ob=xlsread('3-ejes-post.xlsx');
% hold on
%
% plot3(p_ob(a:b,1), p_ob(a:b,2), p_ob(a:b,3),'.-','LineWidth',2)

% plot3(pp(1), pp(2), pp(3),'o-red','MarkerSize',10);
% plot3(p_surf(1), p_surf(2), p_surf(3),'o-blue','MarkerSize',10);

% title("3 ejes GOTO vs POST")

xlabel('$X(mm)$', 'interpreter', 'latex')
ylabel('$Y(mm)$', 'interpreter', 'latex')
zlabel('$Z(mm)$', 'interpreter', 'latex')

grid on

```

```

for_fig(hfig, 20, 0.69)

r=5;

%% Puntos en Q a una distancia d, de P a lo largo de la direcci n normal n

% hfig=figure;
%
% P=p_obj(:,1:3);
%
% n=p_obj(:,4:6);
%
% d=10;
%
% Q=P+d*n;
%
% k=788;
%
% quiver3(P(:,1),P(:,2),P(:,3),Q(:,1),Q(:,2),Q(:,3),'LineWidth', 3);
%
%
% xlabel('$X(mm)$', 'interpreter', 'latex')
% ylabel('$Y(mm)$', 'interpreter', 'latex')
% zlabel('$Z(mm)$', 'interpreter', 'latex')
%% #####

p_obj_teo=zeros(size(p_obj,1),3);
n_obj_teo=zeros(size(p_obj,1),3);

for i=1:size(p_obj,1)
    px=p_obj(i,1);
    py=p_obj(i,2);

    p_teo=pboide(px,py);

    p_obj_teo(i,:)=p_teo;
    n_obj_teo(i,:)=normal_surf(p_obj_teo(i,:));

    if p_teo(3)==0
        p_obj_teo(i,:)=p_obj(i, 1:3);
        n_obj_teo(i,:)=p_obj(i, 4:6);
    end

    % if p_teo(3)==0 && i~=1
    %     p_obj_teo(i,:)=p_obj(i, 1:3);
    %     n_obj_teo(i,:)=normal_surf(p_obj_teo(i-1,:));
    % end
    %
    % if p_teo~=0
    %
    % end

end

obj_tot=p_obj_teo+r*n_obj_teo;

```

```

p_obj_teo(end,:)=p_obj(end,1:3);
% p_obj_teo=p_obj(:,1:3);

%% Gráficas de numeración de puntos

% --- f2
close

tipo_err='XYZAC';

ax=1;
ay=1;
az=1;

error=(ax*(p_obj(:,1)-p_real(:,1)).^2+ay*(p_obj(:,2)-p_real(:,2)).^2+az*(
    p_obj(:,3)-p_real(:,3)).^2).^0.5;

disp(sum(error))

hfig=figure;
grid on

scatter3(p_obj(:,1), p_obj(:,2), p_obj(:,3), 20, 'filled')

np=length(p_obj);

str=string(1:np);

xf=NaN(1,np);
yf=NaN(1,np);
zf=NaN(1,np);

f=41;

for i=1:f:np
    xf(i)=p_obj(i,1);
    yf(i)=p_obj(i,2);
    zf(i)=p_obj(i,3);
end

hold on

color=[1, 0, 0.3];

colors= repmat(color, np, 1);

textscatter3(xf, yf, zf, str, 'FontSize', 20, 'ColorData', colors, '
    MaxTextLength', 10, 'BackgroundColor', [1,1,1]);

% textscatter3(xf, yf, zf, str);

% colorbar;
% colormap(parula);

```

```

x = S(:,1);
y = S(:,2);
z = S(:,3);

% Interpola al grid

% [X,Y] = meshgrid(linspace(min(x),max(x),300), linspace(min(y),max(y),300))
;
% Z = griddata(x,y,z,X,Y);
%
% s=surf(X,Y,Z, 'FaceColor','blue', 'EdgeColor', 'none');
% grayColor = [0.5 0.5 0.5];
% set(s, 'FaceColor', grayColor);
%
%
%
% axis tight
%
%
% % shading interp
% light % create a light
% % lighting gouraud % preferred method for lighting curved surfaces
% % lightangle(70, 30)
%
% material dull

axis tight

xlabel('$X(mm)$', 'interpreter', 'latex')
ylabel('$Y(mm)$', 'interpreter', 'latex')
zlabel('$Z(mm)$', 'interpreter', 'latex')

title(strcat({'\textbf{\fontsize{10}{1}\selectfont Numeracion de los puntos
}'}), 'Interpreter', 'latex');

for_fig(hfig, 20, 0.69)
axis tight

view(45,66)

%% Gr ficas de errores CAM vs REAL

% --- f2

tipo_err='XYZAC';

ax=1;
ay=1;
az=1;

error=(ax*(p_obj(:,1)-p_real(:,1)).^2+ay*(p_obj(:,2)-p_real(:,2)).^2+az*(
p_obj(:,3)-p_real(:,3)).^2).^0.5;

```

```

disp(sum(error))

hfig=figure;
grid on

scatter3(p_obj(:,1), p_obj(:,2), p_obj(:,3), [], error, 'filled')

colorbar;
colormap(parula);

xlabel('$X(mm)$', 'interpreter', 'latex')
ylabel('$Y(mm)$', 'interpreter', 'latex')
zlabel('$Z(mm)$', 'interpreter', 'latex')

title(strcat({'\textbf{\fontsize{10}{1}\selectfont CAM vs Real }'}, tipo_err)
, 'Interpreter', 'latex');

for_fig(hfig, 20, 0.69)
axis tight

view(45,66)

fname=strcat('cam-vs-real-eu-', tipo_err);
path=('C:\Users\User\Desktop\__TFM\4-resultados\images\fin\ analisis-errores '
);
sav_fig(hfig, fname, path)

chdir('C:\Users\User\Desktop\__TFM\4-resultados\images\fin\ analisis-errores '
)
writematrix(error, strcat(fname, '.xlsx'))
chdir('C:\Users\User\Desktop\__TFM\1-pruebas')
%% ----- ahora con diferencias separadas

hfig=figure;

fig_pto_d(p_real(:,1:3), p_obj(:, 1:3))

for_fig(hfig, 20, 0.69)
view(0,90)

title(strcat({'\textbf{\fontsize{10}{1}\selectfont CAM vs Real }'}, tipo_err)
, 'Interpreter', 'latex');

fname=strcat('cam-vs-real-di-', tipo_err);
path=('C:\Users\User\Desktop\__TFM\4-resultados\images\fin\ analisis-errores '
);
sav_fig(hfig, fname, path)

dcm_obj = datacursormode(gcf);
set(dcm_obj, 'UpdateFcn', @(src, event) sprintf('Index: %d', event.DataIndex
));

```

```

%% Separacion de cuadrantes

hfig=figure;

x=p_obj_teo(:,1);
y=p_obj_teo(:,2);

ind_1= find(x<=0 & y<=0);
ind_2= find(x<=0 & y>=0);
ind_3= find(x>=0 & y<=0);
ind_4= find(x>=0 & y>=0);

color=zeros(length(p_obj_teo), 3);

color(ind_1,:)=repmat([1,0,0], length(ind_1), 1);
color(ind_2,:)=repmat([0,1,0], length(ind_2), 1);
color(ind_3,:)=repmat([0,0,1], length(ind_3), 1);
color(ind_4,:)=repmat([0.5,0.5,0], length(ind_4), 1);

scatter3(p_obj(:,1), p_obj(:,2), p_obj(:,3), 30, color, 'filled')

for_fig(hfig, 20, 0.69)
view(45,66)

axis tight

xlabel('$X(mm)$', 'interpreter', 'latex')
ylabel('$Y(mm)$', 'interpreter', 'latex')
zlabel('$Z(mm)$', 'interpreter', 'latex')

title(strcat({'\textbf{\fontsize{10}{1}\selectfont Cuadrantes }'}, '
    Interpreter', 'latex'));

%% RESULTADOS DE METRICAS analisis de errores

% disp(error)

n=length(error);

mae_=@(error) 1/n*sum(abs(error));
dest_=@(error) (sum(error.^2)/(n-1))^0.5;

aaa_=[mae_(error(ind_1)), dest_(error(ind_1));
      mae_(error(ind_2)), dest_(error(ind_2));
      mae_(error(ind_3)), dest_(error(ind_3));
      mae_(error(ind_4)), dest_(error(ind_4));];

disp(aaa_)

%% errores teorico vs CAM

% % calculo los puntos en la superficie teorica a menor distancia de los

```

```

% % otros puntos

% p_surf=zeros(size(p_obj,1), 3);
%
% for j=1:size(p_obj,1)
%
%     p_surf(j,:)=pso_dist_surf_pto(p_obj, j);
% end

% hecho

%% grafica errores teorico vs CAM

% --- f3

% ax=1;
% ay=1;
% az=1;
%
% toli=1e-4;
%
% error=(ax*(p_obj(:,1)-p_surf(:,1)).^2+ay*(p_obj(:,2)-p_surf(:,2)).^2+az*(
%     p_obj(:,3)-p_surf(:,3)).^2).^0.5;
%
% error(error>tol)=NaN;
%
% % for j=1:length(error)
% %
% %     if error(j)> toli
% %         error(j)=0;
% %     end
% % end
%
% hfig=figure;
% grid on
%
% scatter3(p_obj(:,1), p_obj(:,2), p_obj(:,3), 45, error, 'filled')
%
% colorbar;
%
% colormap(parula);
% % caxis([0, 0.5e-4]);
%
%
% xlabel('$X(mm)$', 'interpreter', 'latex')
% ylabel('$Y(mm)$', 'interpreter', 'latex')
% zlabel('$Z(mm)$', 'interpreter', 'latex')
%
% title({'\textbf{\fontsize{10}{1}\selectfont CAM vs Teorico}'}, '
%     Interpreter', 'latex');
% % title('CAM vs Te rico');
%
%
% for_fig(hfig, 20, 0.69)
%

```

```

% axis tight
%
% fname='cam-vs-teo';
% path=('C:\Users\User\Desktop\__TFM\4-resultados\images\fin');
%
% sav_fig(hfig, fname, path)
%
%
%
%
% %% ----- Figura de NTEO vs NCAM
%
%
% % --- f4 ---
% hfig=figure;
%
% dn=abs(n_obj_teo-p_obj(:, 4:6));
%
% plot(dn(:,1), 'LineWidth', 2)
% grid on
% hold on
% plot(dn(:,2), 'LineWidth', 2)
% plot(dn(:,3), 'LineWidth', 2)
%
%
% legend('$dn_{x}$', '$dn_{y}$', '$dn_{z}$', 'Interpreter', 'latex')
%
% xlabel('Punto')
% ylabel('Diferencia')
% % zlabel('$Z(mm)$', 'interpreter', 'latex')
%
% title({'\textbf{\fontsize{10}{1}\selectfont dNormales (CAM vs Teo)'}}, '
% Interpreter', 'latex');
% axis tight
%
%
% for_fig(hfig, 20, 0.69)
% view(0,90)
%
%
%
% fname='dnormales';
% path=('C:\Users\User\Desktop\__TFM\4-resultados\images\fin');
%
% sav_fig(hfig, fname, path)
%
%
% %% ----- Comparo XYZ (A vs C)
%
% load XYZA_err.mat
%
% e_a=error;
%
% load XYZC_err.mat
%
% e_c=error;
%
%
%
```

```

% hfig=figure;
%
% plot(e_a, 'DisplayName','Error XYZA', 'Linewidth', 2)
%
% hold on
% grid on
%
% plot(e_c, 'DisplayName','Error XYZC', 'Linewidth', 2)
%
% legend('Location', 'northwest')
%
% xlabel('Punto')
% ylabel('Error (mm)')
%
% for_fig(hfig, 20, 0.69)
% view(0,90)
%
% fname='error_a_vs_c';
% path=('C:\Users\User\Desktop\__TFM\4-resultados\images\fin');
%
% sav_fig(hfig, fname, path)
% %% errores de MHfexac por separado
%
% % hfig=hfig=figure;
% %
% % hold on
% % grid on
% %
% % load('Tabla errores MH Fexac.mat')
% %
% % yy=linspace(-100,500,100);
% % plot(yy, interp_T(Tbl_yTy, yy))
% % scatter(Tbl_yTy(1,:),Tbl_yTy(2,:));
% %
% % xlabel('$Y(mm)$', 'interpreter', 'latex')
% % ylabel('Errores yTy (mm)', 'interpreter', 'latex')
% %
% % title({'\textbf{\fontsize{12}{10}\selectfont Errores yTy}'}, '
% %       Interpreter', 'latex');
%
%
%
% toc

```