PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

DISEÑO MECÁNICO DE UN SIMULADOR DE MARCHA NORMAL BASADO EN LA PLATAFORMA STEWART-GOUGH

ANEXOS

Tesis para optar el Título de Ingeniero Mecánico,

que presenta el bachiller:

GONZALO EDUARDO SEVILLANO GAINZA

ASESOR: Ing. Dante Elías Giordano

Lima, Febrero 2014

ANEXOS

PROGRAMA DE CÁLCULO PARA LA OBTENCIÓN DEL ESPACIO DE TRABAJO EOC, USANDO LA CINEMÁTICA INDIRECTA

Programa para ser usado en Matlab

```
%Workspace (Espacio de Trabajo a Orientación constante);
%Creando matriz para guardar valores;
X=zeros(1,10);
Z=zeros(1,10);
%Utlizando contadores: m y n;
m=0;
n=0;

% Utilizando función for, para la iteración;
for a=-500:5:500
for c=400:800

%Ajustando las dimensiones preliminaries de la plataforma, radios y angulos de los
hexagonos;
rf = 215;
gk = 15;
rm = 150;
hk = 20;

x = a;
y = 0;
z = c;
t = [ x y z ];
T = [ t
      t
      t
      t
      t
      t ];
%ángulos de rotación [alfa beta gamma] rotaciones con respecto a [x y z];
alfa1= 0;
beta1= 30;
gamma1= 0;

alfa=(alfa1)*3.1416/180;
```

```
beta=(beta1)*3.1416/180;
gamma=(gamma1)*3.1416/180;

c1 = cos(alfa);
s1 = sin(alfa);
c2 = cos(beta);
s2 = sin(beta);
c3 = cos(gamma);
s3 = sin(gamma);

%definiendo matriz de rotación Euler;
R=[c3*c2  c3*s2*s1-s3*c1  s3*s1+c3*s2*c1
   s3*c2  c3*c1+s3*s2*s1  s3*s2*c1-c3*s1
   -s2    c2*s1           c2*c1 ];

k=3.1416/6;
g = gk*3.1416/180;
h = hk*3.1416/180;

 %Obteniendo los puntos fijos de los actuadores  en la plataforma fija;
  b1 = [ -rf*cos(k+g) -rf*sin(k+g) 0];
  b2 = [ rf*cos(k+g) -rf*sin(k+g) 0];
  b3 = [ rf*cos(k-g) -rf*sin(k-g) 0];
  b4 = [ rf*sin(g) rf*cos(g) 0];
  b5 = [ -rf*sin(g) rf*cos(g) 0];
  b6 = [ -rf*cos(k-g) -rf*sin(k-g) 0];

B= [ b1
     b2
     b3
     b4
     b5
     b6 ];

% Definiendo los puntos pi, coordenadas de las uniones superiores con respecto al SRF.
```

```
 p1 = [ -rm*sin(h) -rm*cos(h) 0];
 p2 = [ rm*sin(h) -rm*cos(h) 0];
 p3 = [ rm*cos(k-h) rm*sin(k-h)  0];
 p4 = [ rm*cos(k+h) rm*sin(k+h) 0];
 p5 = [ -rm*cos(k+h) rm*sin(k+h) 0];
 p6 = [ -rm*cos(k-h) rm*sin(k-h)  0];

Pb1 = (R*p1')' + t;
Pb2 = (R*p2')' + t;
Pb3 = (R*p3')' + t;
Pb4 = (R*p4')' + t;
Pb5 = (R*p5')' + t;
Pb6 = (R*p6')' + t;

P = [ Pb1
      Pb2
      Pb3
      Pb4
      Pb5
      Pb6 ];

% analizando un actuador

  L = P -B ;

l1 = [ L(1,1) L(1,2) L(1,3)];
l2 = [ L(2,1) L(2,2) L(2,3)];
l3 = [ L(3,1) L(3,2) L(3,3)];
l4 = [ L(4,1) L(4,2) L(4,3)];
l5 = [ L(5,1) L(5,2) L(5,3)];
l6 = [ L(6,1) L(6,2) L(6,3)];

  D = [ sqrt(l1*l1')
        sqrt(l2*l2')
        sqrt(l3*l3')
        sqrt(l4*l4')
        sqrt(l5*l5')
```

```
        sqrt(l6*l6') ];


s1 = [ L(1,1) L(1,2) 0];
s2 = [ L(2,1) L(2,2) 0];
s3 = [ L(3,1) L(3,2) 0];
s4 = [ L(4,1) L(4,2) 0];
s5 = [ L(5,1) L(5,2) 0];
s6 = [ L(6,1) L(6,2) 0];

    S = [ sqrt(s1*s1')
        sqrt(s2*s2')
        sqrt(s3*s3')
        sqrt(s4*s4')
        sqrt(s5*s5')
        sqrt(s6*s6') ];

q=50;

k=3.1416/6;
qk = q*3.1416/180;

N=[ P(1,3)*tan(qk)
    P(2,3)*tan(qk)
    P(3,3)*tan(qk)
    P(4,3)*tan(qk)
    P(5,3)*tan(qk)
    P(6,3)*tan(qk)];

if N>=S
if  D(1,1)>530 & D(2,1)>530 & D(3,1)>530 & D(4,1)>530 & D(5,1)>530 & D(6,1)>530
if  D(1,1)<=530.5 | D(2,1)<=530.5 | D(3,1)<=530.5 | D(4,1)<=530.5 | D(5,1)<=530.5 |
D(6,1)<=530.5
m=m+1;
m
X(1,m+n)=a;
Z(1,m+n)=c;
```

```
else
if  D(1,1)<730 & D(2,1)<730 & D(3,1)<730 & D(4,1)<730 & D(5,1)<730 & D(6,1)<730
if  D(1,1)>=729.5 | D(2,1)>=729.5 | D(3,1)>=729.5 | D(4,1)>=729.5 | D(5,1)>729.5 |
D(6,1)>729.5
n=n+1;
X(1,m+n)=a;
Z(1,m+n)=c;
n
end
end
end
end
end
end
end
plot(X,Z,'r.')

%Graficando el Espacio de trabajo ejes X-Z;
axis ([-450 450  0 800]);
grid on
box on
title('Espacio de trabajo X-Z');
xlabel('X mm')
ylabel('Z mm')
```

PROGRAMA DE CÁLCULO PARA LA OBTENCIÓN DE LAS VELOCIDADES, USANDO LA CINEMÁTICA INDIRECTA

Programa para ser usado en Matlab

```
%Definiendo los puntos bi, coordenadas de las uniones inferiores con respecto al SRF.
rf = 215;
gk = 30;
rm = 150;
hk = 20;

x =
y =
z =
t = [ x y z ];
T = [ t
    t
    t
    t
    t
    t ];
%angulos de rotacion [alfa beta gamma] rotaciones con respecto a [x y z]
alfa1= input  ('Angulo de rotacion en el eje X        ');
beta1= input  ('Angulo de rotacion en el eje Y        ');
gamma1= input ('Angulo de rotacion en el eje Z        ') ;

alfa=(alfa1)*3.1416/180;
beta=(beta1)*3.1416/180;
gamma=(gamma1)*3.1416/180;

c1 = cos(alfa);
s1 = sin(alfa);
c2 = cos(beta);
s2 = sin(beta);
c3 = cos(gamma);
s3 = sin(gamma);

%Definiendo matriz de rotacion Euler;
R=[c3*c2  c3*s2*s1-s3*c1  s3*s1+c3*s2*c1
   s3*c2  c3*c1+s3*s2*s1  s3*s2*c1-c3*s1
```

```
   -s2    c2*s1           c2*c1 ];

k=3.1416/6;
g = gk*3.1416/180
h = hk*3.1416/180

  b1 = [ -rf*cos(k+g) -rf*sin(k+g) 0];
  b2 = [ rf*cos(k+g) -rf*sin(k+g) 0];
  b3 = [ rf*cos(k-g) -rf*sin(k-g) 0];
  b4 = [ rf*sin(g) rf*cos(g) 0];
  b5 = [ -rf*sin(g) rf*cos(g) 0];
  b6 = [ -rf*cos(k-g) -rf*sin(k-g) 0];


B= [ b1
     b2
     b3
     b4
     b5
     b6 ];

% Definiendo los puntos pi, coordenadas de las uniones superiores con respecto al
SRF;

 p1 = [ -rm*sin(h) -rm*cos(h) 0];
 p2 = [ rm*sin(h) -rm*cos(h) 0];
 p3 = [ rm*cos(k-h) rm*sin(k-h)  0];
 p4 = [ rm*cos(k+h) rm*sin(k+h) 0];
 p5 = [ -rm*cos(k+h) rm*sin(k+h) 0];
 p6 = [ -rm*cos(k-h) rm*sin(k-h)  0];

Pb1 = (R*p1')' + t;
Pb2 = (R*p2')' + t;
Pb3 = (R*p3')' + t;
Pb4 = (R*p4')' + t;
Pb5 = (R*p5')' + t;
Pb6 = (R*p6')' + t;
```

```
P = [ Pb1
      Pb2
      Pb3
      Pb4
      Pb5
      Pb6 ]

% Analizando un actuador;

  L = P -B ;

l1 = [ L(1,1) L(1,2) L(1,3)];
l2 = [ L(2,1) L(2,2) L(2,3)];
l3 = [ L(3,1) L(3,2) L(3,3)];
l4 = [ L(4,1) L(4,2) L(4,3)];
l5 = [ L(5,1) L(5,2) L(5,3)];
l6 = [ L(6,1) L(6,2) L(6,3)];

  D = [ sqrt(l1*l1')
        sqrt(l2*l2')
        sqrt(l3*l3')
        sqrt(l4*l4')
        sqrt(l5*l5')
        sqrt(l6*l6') ];
   D

N = [ l1/D(1,1)
      l2/D(2,1)
      l3/D(3,1)
      l4/D(4,1)
      l5/D(5,1)
      l6/D(6,1)];

n1 = [ N(1,1) N(1,2) N(1,3)];
n2 = [ N(2,1) N(2,2) N(2,3)];
n3 = [ N(3,1) N(3,2) N(3,3)];
```

```
n4 = [ N(4,1) N(4,2) N(4,3)];
n5 = [ N(5,1) N(5,2) N(5,3)];
n6 = [ N(6,1) N(6,2) N(6,3)];


U = P-T ;

r1 = [ U(1,1) U(1,2) U(1,3)];
r2 = [ U(2,1) U(2,2) U(2,3)];
r3 = [ U(3,1) U(3,2) U(3,3)];
r4 = [ U(4,1) U(4,2) U(4,3)];
r5 = [ U(5,1) U(5,2) U(5,3)];
r6 = [ U(6,1) U(6,2) U(6,3)];

%Hallando la velocidad lineal por actuador;
Cd=135;
Tc=120/Cd;
To=0.4*Tc;
ac=0.8/(To*To/4);
Vmax=ac*(To/2);
tp=Vmax/6;
Vmax1=tp*0;
Vmax1=tp*1;
Vmax2=tp*2;
Vmax3=tp*3;
Vmax4=tp*2;
Vmax5=tp*1;
Vmax6=tp*0;

Xv = Vmax1
Yv = 0
Zv = 0
Xw = 0
Yw = 10.23
Zw = 0

qv = [ Xv
```

```
        Yv
        Zv
        Xw
        Yw
        Zw ];

tv = [ qv(1,1) qv(2,1) qv(3,1)];

w = [ qv(4,1) qv(5,1) qv(6,1)];

V1 = [ n1 cross(r1,n1)]*qv;
V2 = [ n2 cross(r2,n2)]*qv;
V3 = [ n3 cross(r3,n3)]*qv;
V4 = [ n4 cross(r4,n4)]*qv;
V5 = [ n5 cross(r5,n5)]*qv;
V6 = [ n6 cross(r6,n6)]*qv;

Vm1 = [V1;V2;V3;V4;V5;V6]/1000;
dmt=0.04;
Ar=3.1416*dmt*dmt/4;
Qc1=Vm1*Ar

qp1 = tv+cross(w,r1);
qp2 = tv+cross(w,r2);
qp3 = tv+cross(w,r3);
qp4 = tv+cross(w,r4);
qp5 = tv+cross(w,r5);
qp6 = tv+cross(w,r6);

Qp = [ qp1;qp2;qp3;qp4;qp5;qp6 ];

%Velocidad angular de cada uno de los actuadores

w1 = cross(n1,qp1)/D(1,1);
w2 = cross(n2,qp1)/D(2,1);
w3 = cross(n3,qp1)/D(3,1);
w4 = cross(n4,qp1)/D(4,1);
```

```
w5 = cross(n5,qp1)/D(5,1);
w6 = cross(n6,qp1)/D(6,1);
%Graficando;
X=[B(1,1),B(2,1),B(3,1),B(4,1),B(5,1),B(6,1),B(1,1),Pb1(1,1),Pb2(1,1),Pb3(1,1),Pb2(1,1
),B(2,1),B(3,1),Pb3(1,1),Pb4(1,1),Pb5(1,1),Pb4(1,1),B(4,1),B(5,1),Pb5(1,1),Pb6(1,1),Pb
1(1,1),Pb6(1,1),B(6,1)];
Y=[B(1,2),B(2,2),B(3,2),B(4,2),B(5,2),B(6,2),B(1,2),Pb1(1,2),Pb2(1,2),Pb3(1,2),Pb2(1,2
),B(2,2),B(3,2),Pb3(1,2),Pb4(1,2),Pb5(1,2),Pb4(1,2),B(4,2),B(5,2),Pb5(1,2),Pb6(1,2),Pb
1(1,2),Pb6(1,2),B(6,2)];
Z=[B(1,3),B(2,3),B(3,3),B(4,3),B(5,3),B(6,3),B(1,3),Pb1(1,3),Pb2(1,3),Pb3(1,3),Pb2(1,3
),B(2,3),B(3,3),Pb3(1,3),Pb4(1,3),Pb5(1,3),Pb4(1,3),B(4,3),B(5,3),Pb5(1,3),Pb6(1,3),Pb
1(1,3),Pb6(1,3),B(6,3)];

Fig1= plot3(X,Y,Z,'b')
```

PROGRAMA DE CÁLCULO DE LAS FUERZAS ESTÁTICAS SOBRE LOS ACTUADORES, CONCOCIENDOSE LAS CARGAS. POSICIÓN Y ORIENTACIÓN DEL EFECTOR FINAL.

Programa para ser usado en Matlab

```
%Workspace;
%Definiendo los puntos bi, coordenadas de las uniones inferiores con respecto al SRF.
rf = 215;
gk = 30;
rm = 150;
hk = 20;
x = input ('Ingrese la posicion X del centro de lap plataforma     ');
y = input ('Ingrese la posicion Y del centro de la plataforma      ');
z = input ('Ingrese la posicion Z del centro de la plataforma      ');
t = [ x y z ];
T = [ t
      t
      t
      t
      t
      t ];
%angulos de rotacion [alfa beta gamma] rotaciones con respecto a [x y z]
alfa1= input  ('Angulo de rotacion en el eje X        ');
beta1= input  ('Angulo de rotacion en el eje Y        ');
gamma1= input ('Angulo de rotacion en el eje Z        ') ;
alfa=(alfa1)*3.1416/180;
beta=(beta1)*3.1416/180;
gamma=(gamma1)*3.1416/180;
c1 = cos(alfa);
s1 = sin(alfa);
c2 = cos(beta);
s2 = sin(beta);
c3 = cos(gamma);
s3 = sin(gamma);
%definiendo matriz de rotacion
R=[c3*c2  c3*s2*s1-s3*c1  s3*s1+c3*s2*c1
   s3*c2  c3*c1+s3*s2*s1  s3*s2*c1-c3*s1
   -s2    c2*s1           c2*c1 ];
k=3.1416/6;
g = gk*3.1416/180
```

```
h = hk*3.1416/180
  b1 = [ -rf*cos(k+g) -rf*sin(k+g) 0];
  b2 = [ rf*cos(k+g) -rf*sin(k+g) 0];
  b3 = [ rf*cos(k-g) -rf*sin(k-g) 0];
  b4 = [ rf*sin(g) rf*cos(g) 0];
  b5 = [ -rf*sin(g) rf*cos(g) 0];
  b6 = [ -rf*cos(k-g) -rf*sin(k-g) 0];

B= [ b1
     b2
     b3
     b4
     b5
     b6 ];

% Definiendo los puntos pi, coordenadas de las uniones superiores con respecto al
SRF.
 p1 = [ -rm*sin(h) -rm*cos(h) 0];
 p2 = [ rm*sin(h) -rm*cos(h) 0];
 p3 = [ rm*cos(k-h) rm*sin(k-h)  0];
 p4 = [ rm*cos(k+h) rm*sin(k+h) 0];
 p5 = [ -rm*cos(k+h) rm*sin(k+h) 0];
 p6 = [ -rm*cos(k-h) rm*sin(k-h)  0];

Pb1 = (R*p1')' + t;
Pb2 = (R*p2')' + t;
Pb3 = (R*p3')' + t;
Pb4 = (R*p4')' + t;
Pb5 = (R*p5')' + t;
Pb6 = (R*p6')' + t;

P = [ Pb1
      Pb2
      Pb3
      Pb4
      Pb5
      Pb6 ]
```

```
% analizando un actuador

  L = P -B ;

l1 = [ L(1,1) L(1,2) L(1,3)];
l2 = [ L(2,1) L(2,2) L(2,3)];
l3 = [ L(3,1) L(3,2) L(3,3)];
l4 = [ L(4,1) L(4,2) L(4,3)];
l5 = [ L(5,1) L(5,2) L(5,3)];
l6 = [ L(6,1) L(6,2) L(6,3)];

  D = [ sqrt(l1*l1')
        sqrt(l2*l2')
        sqrt(l3*l3')
        sqrt(l4*l4')
        sqrt(l5*l5')
        sqrt(l6*l6') ];
   D
 n1 =l1/D(1,1)
 n2 =l2/D(2,1)
 n3 =l3/D(3,1)
 n4 =l4/D(4,1)
 n5 =l5/D(5,1)
 n6 =l6/D(6,1)

 Pr1=Pb1-t
 Pr2=Pb2-t
 Pr3=Pb3-t
 Pr4=Pb4-t
 Pr5=Pb5-t
 Pr6=Pb6-t

 m1= cross (Pr1,n1);
 m2= cross (Pr2,n2);
 m3= cross (Pr3,n3);
 m4= cross (Pr4,n4);
 m5= cross (Pr5,n5);
```

```
m6= cross (Pr6,n6);

% Analizando estáticamente;
% Equilibrio de fuerzas;
M= [n1(1,1) n2(1,1) n3(1,1) n4(1,1) n5(1,1) n6(1,1)
    n1(1,2) n2(1,2) n3(1,2) n4(1,2) n5(1,2) n6(1,2)
    n1(1,3) n2(1,3) n3(1,3) n4(1,3) n5(1,3) n6(1,3)
    m1(1,1) m2(1,1) m3(1,1) m4(1,1) m5(1,1) m6(1,1)
    m1(1,2) m2(1,2) m3(1,2) m4(1,2) m5(1,2) m6(1,2)
    m1(1,3) m2(1,3) m3(1,3) m4(1,3) m5(1,3) m6(1,3)]

 W=[0;0;1000;0;0;0];

 F=inv(M)*W;


X=[B(1,1),B(2,1),B(3,1),B(4,1),B(5,1),B(6,1),B(1,1),Pb1(1,1),Pb2(1,1),Pb3(1,1),Pb2(1,1),B(2,1),B(3,1),Pb3(1,1),Pb4(1,1),Pb5(1,1),Pb4(1,1),B(4,1),B(5,1),Pb5(1,1),Pb6(1,1),Pb1(1,1),Pb6(1,1),B(6,1)];
Y=[B(1,2),B(2,2),B(3,2),B(4,2),B(5,2),B(6,2),B(1,2),Pb1(1,2),Pb2(1,2),Pb3(1,2),Pb2(1,2),B(2,2),B(3,2),Pb3(1,2),Pb4(1,2),Pb5(1,2),Pb4(1,2),B(4,2),B(5,2),Pb5(1,2),Pb6(1,2),Pb1(1,2),Pb6(1,2),B(6,2)];
Z=[B(1,3),B(2,3),B(3,3),B(4,3),B(5,3),B(6,3),B(1,3),Pb1(1,3),Pb2(1,3),Pb3(1,3),Pb2(1,3),B(2,3),B(3,3),Pb3(1,3),Pb4(1,3),Pb5(1,3),Pb4(1,3),B(4,3),B(5,3),Pb5(1,3),Pb6(1,3),Pb1(1,3),Pb6(1,3),B(6,3)];

Fig1= plot3(X,Y,Z,'g')
```

FOTOS DEL ENSAMBLE DE LA PLATAFORMA Y COMPONENTES