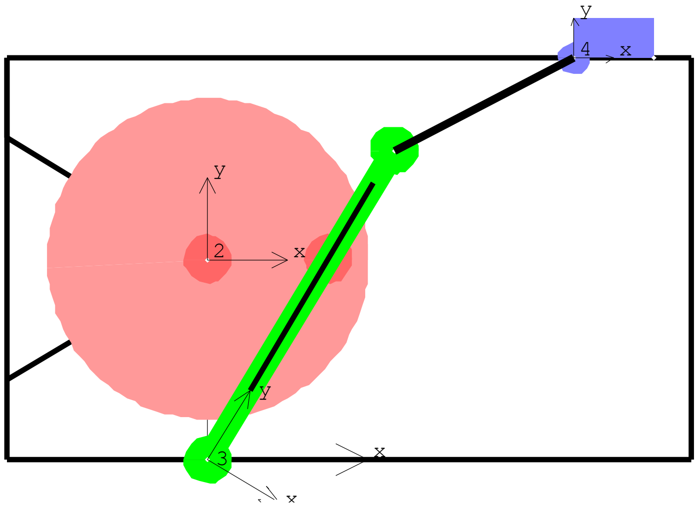
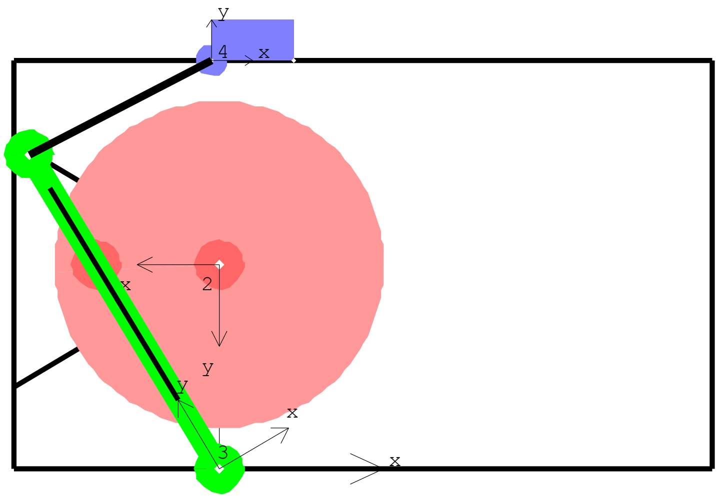
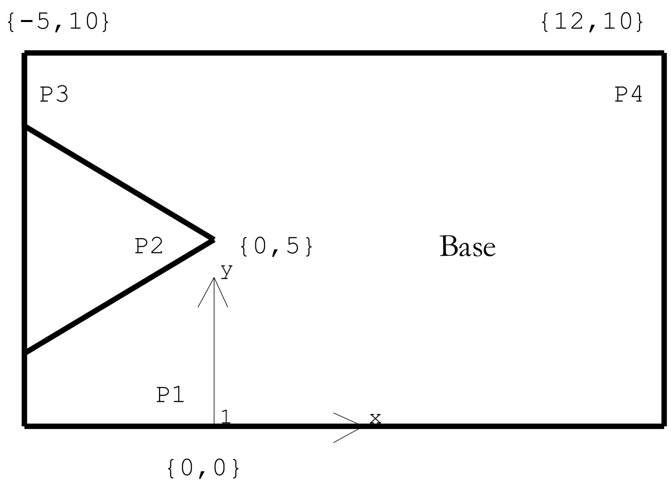
ANALISIS CINEMATICO MECANISMOS

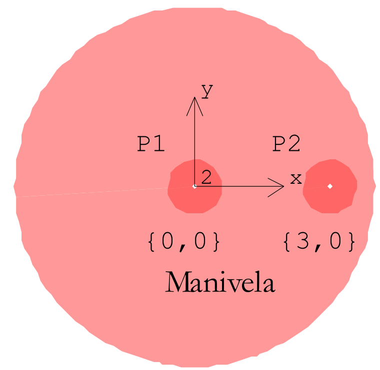
EJEMPLO MECANISMO RETORNO RAPIDO

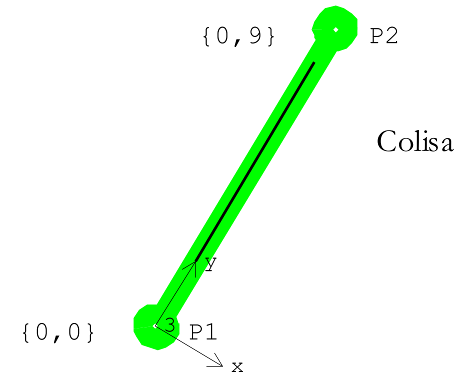




ANALISIS CINEMATICO MECANISMOS
(1) DEFINICION CUERPOS Y PARAMETROS

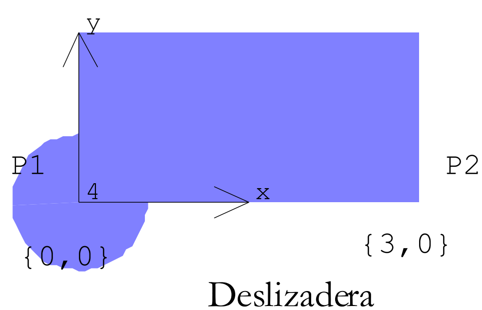






CUERPOS :

base = 1;
manivela = 2;
colisa = 3;
deslizadera = 4;

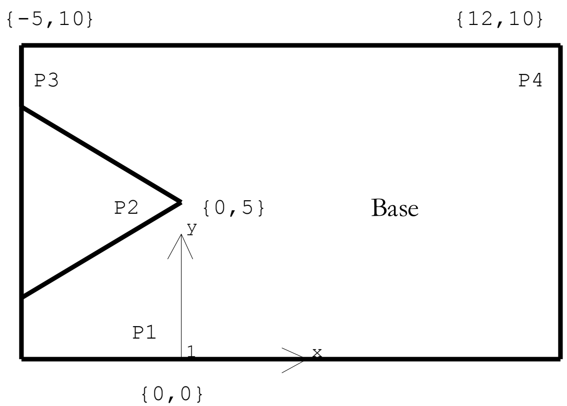


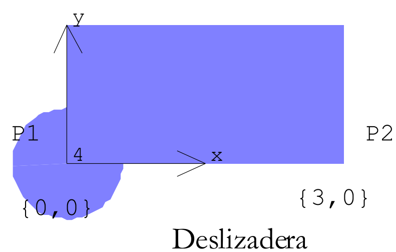
PARAMETROS :

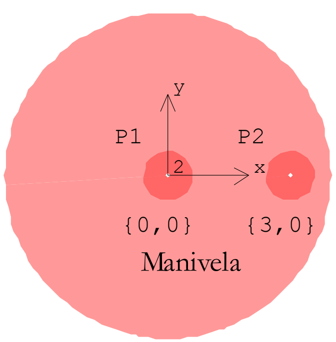
longitud = .;
velocidad = .;
anguloI = .;

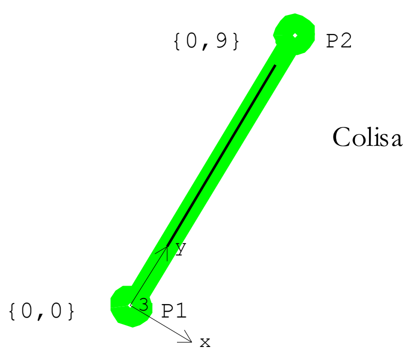
ANALISIS CINEMATICO MECANISMOS
(1) DEFINICION CUERPOS Y PARAMETROS

DEFINICION DE LOS CUERPOS :







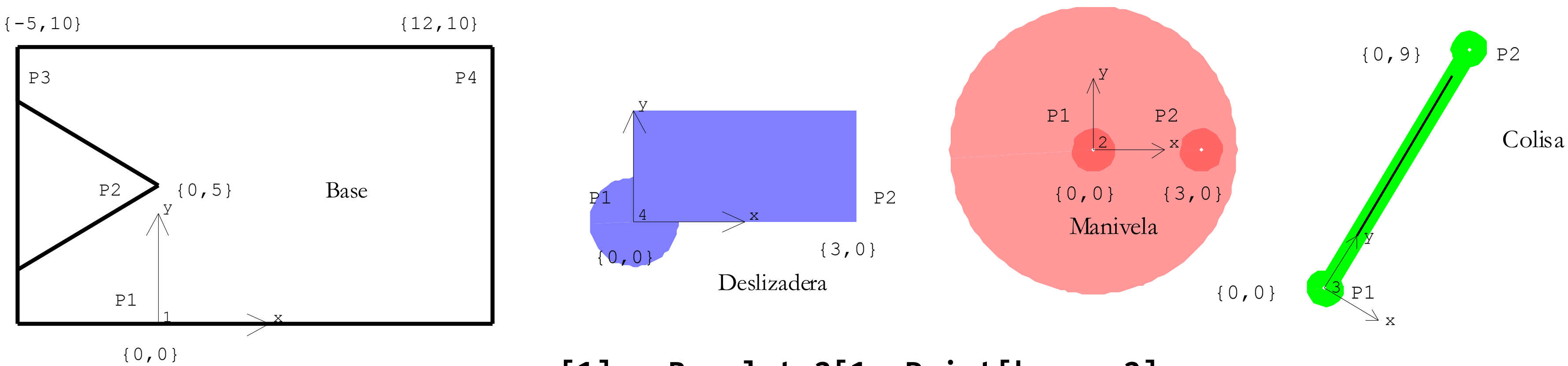



```
bd[base] = Body[base,
  PointList->{
    (*P1*) { 0,  0 },
    (*P2*) { 0,  5 },
    (*P3*) {-5, 10 },
    (*P4*) {12, 10 }}];
bd[manivela] = Body[manivela,
  PointList->{
    (*P1*) { 0, 0 },
    (*P2*) { longitud, 0 }},
  InitialGuess->{{0, 5}, 0}];
bd[colisa] = Body[colisa,
  PointList->{
    (*P1*) { 0, 0 },
    (*P2*) { 0, 9 }},
  InitialGuess->{{0, 0}, -0.5}];
bd[deslizadera] = Body[deslizadera,
  PointList->{
    (*P1*){ 0, 0 },
    (*P2*){ 2, 0 }},
  InitialGuess->{{6, 10}, 0}];
```

```
SetBodies[bd[base], bd[manivela], bd[colisa], bd[deslizadera]]
```

ANALISIS CINEMATICO MECANISMOS

(2A) RESTRICCIONES - **IMPULSOR MANIVELA**



DEFINICION DE LAS RESTRICCIONES :

```
cs[1] = Revolute2[1, Point[base, 2],
                     Point[manivela,  1] ];
cs[2] = RotationLock1[2, manivela, base,
                      velocidad T + anguloI ];
cs[3] = Revolute2[3, Point[base, 1],
                     Point[colisa, 1] ];
cs[4] = PointOnLine1[4, Point[manivela, 2],
                        Line[colisa, 1, 2] ];
cs[5] = RelativeDistance1[5, Point[colisa, 2],
                     Point[deslizadera, 1], 5 ];
cs[6] = Translate2[6, Line[base, 3, 4],
                      Line[deslizadera, 1, 2] ];
SetConstraints[cs[1],cs[2],cs[3],cs[4],cs[5],cs[6]]
```

```
SetBodies[bd[base], bd[manivela], bd[colisa], bd[deslizadera]]
```

ANALISIS CINEMATICO MECANISMOS

(2) ANALISIS POSICION - ECUACIONES RESTRICCION

VECTOR COORDENADAS GENERALIZADAS

$$\mathbf{q}_i = \left[\mathbf{r}_i , \quad \phi_i \right]^T$$

```
MatrixForm[Location[All]]
X2
Y2
Th2
X3
Y3
Th3
X4
Y4
Th4
```

ECUACION VECTORIAL DE LAS RESTRICCIONES

$$\Phi(\mathbf{q}, t) = \begin{bmatrix} \Phi^K(\mathbf{q}, t) \\ \Phi^D(\mathbf{q}, t) \end{bmatrix} = \mathbf{0}$$

```
MatrixForm[Constraints[All]]
-X2 = 0
5 - Y2 = 0
0. - 6.28319 T + Th2 = 0
-X3 = 0
-Y3 = 0
9 (-X2 + X3 - 3 Cos[Th2]) Cos[Th3] - 9 (Y2 - Y3 + 3 Sin[Th2]) Sin[Th3] = 0
                                  2                          2
-25 + (Y3 - Y4 + 9 Cos[Th3])  + (X3 - X4 - 9 Sin[Th3]) = 0
34 Sin[Th4] = 0
2 (10 - Y4) Cos[Th4] + 2 (5 + X4) Sin[Th4] = 0
```

ANALISIS CINEMATICO MECANISMOS
(2) ANALISIS POSICION - ECUACIONES RESTRICCION

JACOBIANO DE LAS RESTRICCIONES:

$$\Phi_{\mathbf{q}}(\mathbf{q},t) = \left[\frac{\partial \Phi_i(\mathbf{q},t)}{\partial q_j}\right]_{nx \times nc}$$

```
Jacobian[All,All]
{{-1, 0, 0, 0, 0, 0, 0, 0, 0}, {0, -1, 0, 0, 0, 0, 0, 0, 0},
   {0, 0, 1, 0, 0, 0, 0, 0, 0}, {0, 0, 0, -1, 0, 0, 0, 0, 0},
   {0, 0, 0, 0, -1, 0, 0, 0, 0},
  {-9 Cos[Th3], -9 Sin[Th3], 27 Cos[Th3] Sin[Th2] - 27 Cos[Th2] Sin[Th3],
    9 Cos[Th3], 9 Sin[Th3], -9 Cos[Th3] (Y2 - Y3 + 3 Sin[Th2]) -
     9 (-X2 + X3 - 3 Cos[Th2]) Sin[Th3], 0, 0, 0},
  {0, 0, 0, 2 (X3 - X4 - 9 Sin[Th3]), 2 (Y3 - Y4 + 9 Cos[Th3]),
    -18 Cos[Th3] (X3 - X4 - 9 Sin[Th3]) -
     18 (Y3 - Y4 + 9 Cos[Th3]) Sin[Th3], -2 (X3 - X4 - 9 Sin[Th3]),
    -2 (Y3 - Y4 + 9 Cos[Th3]), 0}, {0, 0, 0, 0, 0, 0, 0, 0, 34 Cos[Th4]},
   {0, 0, 0, 0, 0, 0, 2 Sin[Th4], -2 Cos[Th4],
    2 (5 + X4) Cos[Th4] - 2 (10 - Y4) Sin[Th4]}}
Dimensions[Jacobian[All,All]]
{9, 9}
```

ANALISIS CINEMATICO MECANISMOS

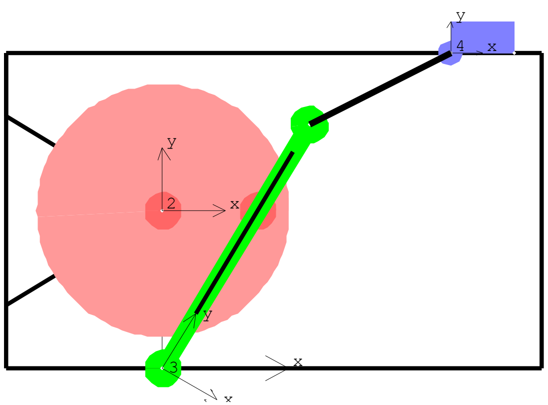
(3) ANALISIS POSICION - RESULTADOS

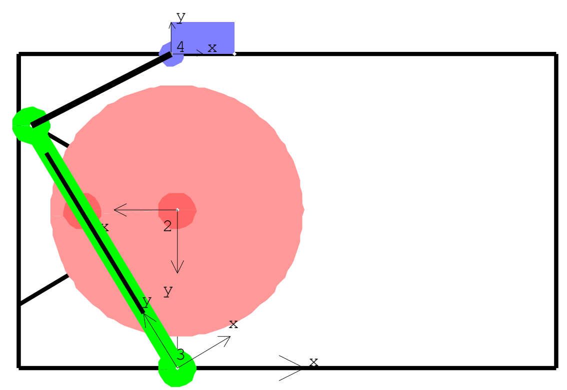
```
longitud = 3;
velocidad = 2*N[Pi];
anguloI = 0.0;
CheckSystem[]
True
SolveMech[.0]
{T -> 0., X2 -> 0., Y2 -> 5., Th2 -> 0., X3 -> 0., Y3 -> 0.,
   Th3 -> -0.54042, X4 -> 9.07905, Y4 -> 10., Th4 -> 0.}
```

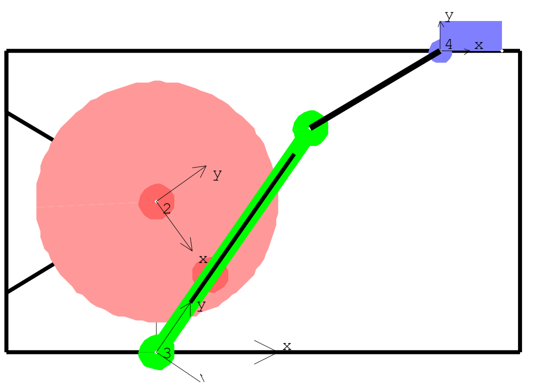
```
MatrixForm[SolveMech[.0]]
T -> 0.
X2 -> 0.
Y2 -> 5.
Th2 -> 0.
X3 -> 0.
Y3 -> 0.
Th3 -> -0.54042
X4 -> 9.07905
Y4 -> 10.
Th4 -> 0.
```

```
MatrixForm[SolveMech[.5]]
T -> 0.5
X2 -> 0.
Y2 -> 5.
Th2 -> 3.14159
X3 -> 0.
Y3 -> 0.
Th3 -> 0.54042
X4 -> -0.181877
Y4 -> 10.
Th4 -> 0.
```

```
MatrixForm[SolveMech[.85]]
T -> 0.85
X2 -> 0.
Y2 -> 5.
Th2 -> 5.34071
X3 -> 0.
Y3 -> 0.
Th3 -> -0.600822
X4 -> 9.37314
Y4 -> 10.
Th4 -> 0.
```



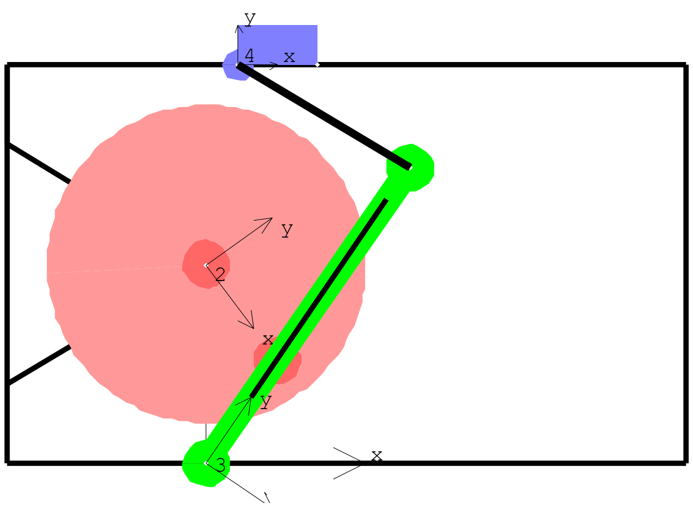




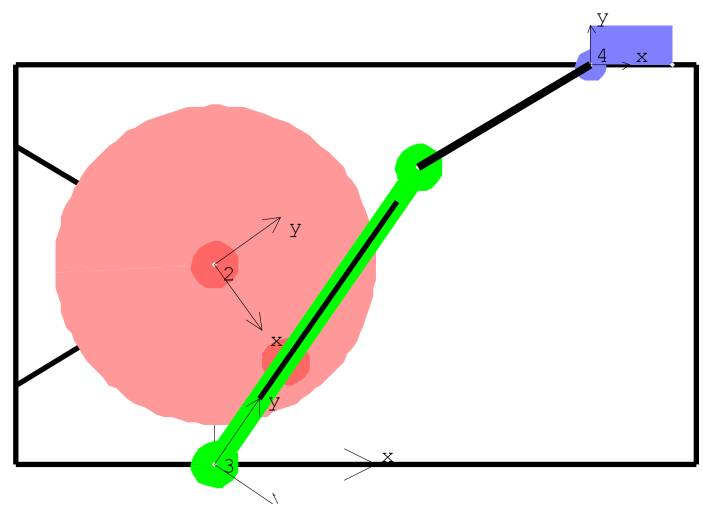
```
longitud = 3;
velocidad = 2*N[Pi];
anguloI = 0.0;

cs[1] = Revolute2[1, Point[base, 2],
                     Point[manivela,  1] ];
cs[2] = RotationLock1[2, manivela, base,
                      velocidad T + anguloI ];
cs[3] = Revolute2[3, Point[base, 1],
                     Point[colisa, 1] ];
cs[4] = PointOnLine1[4, Point[manivela, 2],
                        Line[colisa, 1, 2] ];
cs[5] = RelativeDistance1[5, Point[colisa, 2],
                     Point[deslizadera, 1], 5 ];
cs[6] = Translate2[6, Line[base, 3, 4],
                      Line[deslizadera, 1, 2] ];
SetConstraints[cs[1],cs[2],cs[3],cs[4],cs[5],cs[6]]
```

ANALISIS CINEMATICO MECANISMOS
(3) ANALISIS POSICION - FORMAS ENSAMBLADO

MatrixForm[SolveMech[.85]]
T -> 0.85
X2 -> 0.
Y2 -> 5.
Th2 -> 5.34071
X3 -> 0.
Y3 -> 0.
Th3 -> 5.68236
X4 -> 0.802632
Y4 -> 10.
Th4 -> 0.

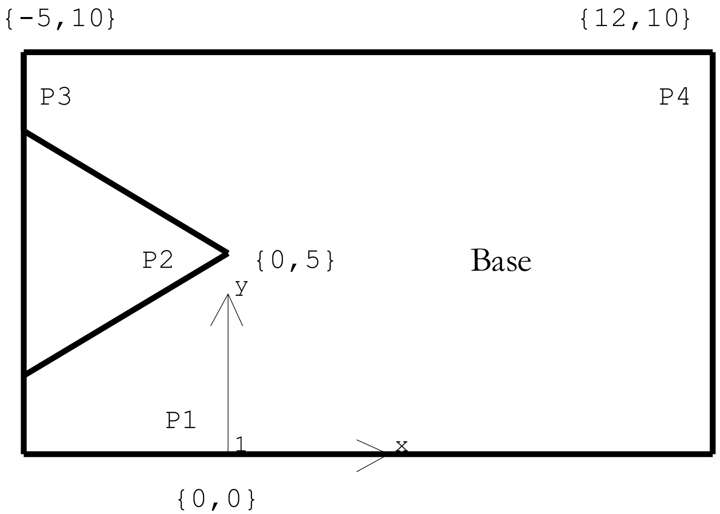


MatrixForm[SolveMech[.85]]
T -> 0.85
X2 -> 0.
Y2 -> 5.
Th2 -> 5.34071
X3 -> 0.
Y3 -> 0.
Th3 -> -0.600822
X4 -> 9.37314
Y4 -> 10.
Th4 -> 0.



ANALISIS CINEMATICO MECANISMOS

(4) DEFINICION GRAFICA CUERPO BASE

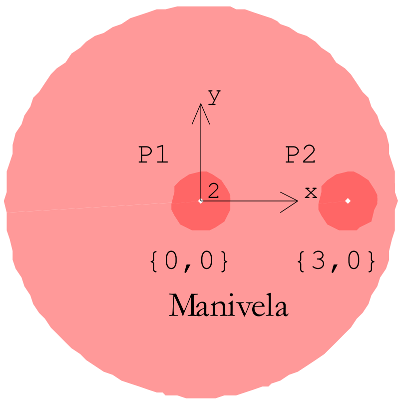



```
baseT = Graphics[
  { Text["{0,0}",   { 0, 0}, { 0  , 1.8}],
    Text["P1",      { 0, 0}, { 2.0  ,-1.8}],
    Text["{0,5}",   { 0, 5}, {-1.4, 0  }],
    Text["P2",      { 0, 5}, { 3.0, 0  }],
    Text["{-5,10}", {-5,10}, {-0.7,-1.8}],
    Text["P3",      {-5,10}, {-1.6, 1.8}],
    Text["{12,10}", {12,10}, { 0.7,-1.8}],
    Text["P4"     , {12,10}, { 1.6, 1.8}],
    Text[FontForm["Base",
             {"Garamond", 12}], {7, 5}] }];
```

```
baseG = Graphics[ {Thickness[.005],
                   Point[{0,0}],
                   Edge[base, {3, 4, {12,0},
                              {-5,0}, {-5,10},
                              {-5,8}, 2, {-5,2}} ],
                   Thickness[.001],
                   RGBColor[0,0,0],
                   LocalAxes[{base},4]} ];
```

```
plot1 = Show[{baseG,baseT},
        PlotRange->{{-7,15}, {-2,12}},
        AspectRatio->Automatic];
```

ANALISIS CINEMATICO MECANISMOS
(4) DEFINICION GRAFICA MANIVELA

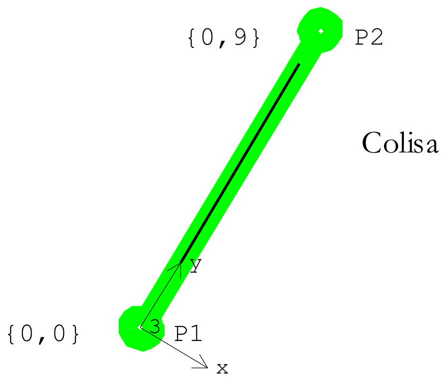



```
manivelaG = Graphics[ {Thickness[.01],
              RGBColor[1,.6,.6],
              Disk[ Location[manivela, 1], 4.0 ] },
              { RGBColor[1,.4,.4],
                Disk[ Location[manivela, 2], 0.6],
                Disk[ Location[manivela, 1], 0.6]},
              { PointSize[.02],
                Vertex[manivela, {1, 2}] },
                Thickness[.001],
                RGBColor[0,0,0],
                LocalAxes[{manivela},2] } ];
```

```
manivelaT = Graphics[
            Text["{0,0}", Location[manivela, 1], {0  , 2.4}],
            Text["P1",    Location[manivela, 1], {2  ,-2.4}],
            Text["{3,0}", Location[manivela, 2], {0  , 2.4}],
            Text["P2",    Location[manivela, 2], {2  ,-2.4}],
            Text[FontForm["Manivela", {"Garamond", 12}], {1, 3}] }];
```

```
plot2 = Show[ {manivelaG, manivelaT} /. SolveMech[0.],
        PlotRange->{{-6,6}, {-1,10}},
        AspectRatio->Automatic];
```

ANALISIS CINEMATICO MECANISMOS
(4) DEFINICION GRAFICA COLISA

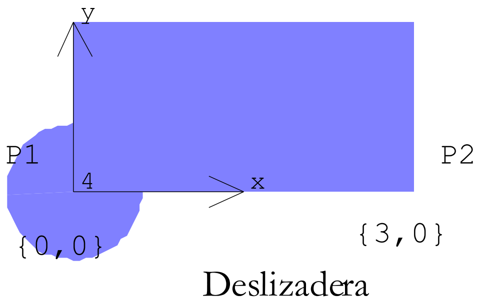



```
colisaG = Graphics[{
                  { RGBColor[0,1,0],
                    Disk[ Location[base, 1],   0.6],
                    Disk[ Location[colisa, 2], 0.6],
                    Bar[ Line[colisa, 1, 2], 0.3, 0.3 ] },
                  PointSize[.02],
                  Vertex[colisa, {1, 2}],
                  Thickness[.005],
                  Edge[colisa, {{0,2}, {0,8}} ],
                  Thickness[.001],
                  RGBColor[0,0,0],
                  LocalAxes[{colisa},2]  }  ];
```

```
colisaT = Graphics[
                  {Text["{0,0}", Location[colisa, 1],{ 1.6, 0}],
                   Text["P1",    Location[colisa, 1],{-2.6, 0}],
                   Text["{0,9}", Location[colisa, 2],{ 1.6, 0}],
                   Text["P2",    Location[colisa, 2],{-2.6, 0}],
                   Text[FontForm["Colisa", {"Garamond", 12}], {7, 5}] }];
```

```
plot3 = Show[ {colisaG, colisaT} /. SolveMech[0.],
        PlotRange->{{-4,10}, {-2,10}},
        AspectRatio->Automatic];
```

ANALISIS CINEMATICO MECANISMOS
(4) DEFINICION GRAFICA DESLIZADERA




```
deslizaderaG = Graphics[{
                {RGBColor[.5,.5,1],
                 Disk[ Location[deslizadera, 1], 0.4 ],
                 Bar[ Line[deslizadera, 1, 2], 1, 0 ]},
                 Vertex[deslizadera, {1, 2}],
                 Thickness[.001],
                 RGBColor[0,0,0],
                 LocalAxes[{deslizadera},1]  }  ];
```

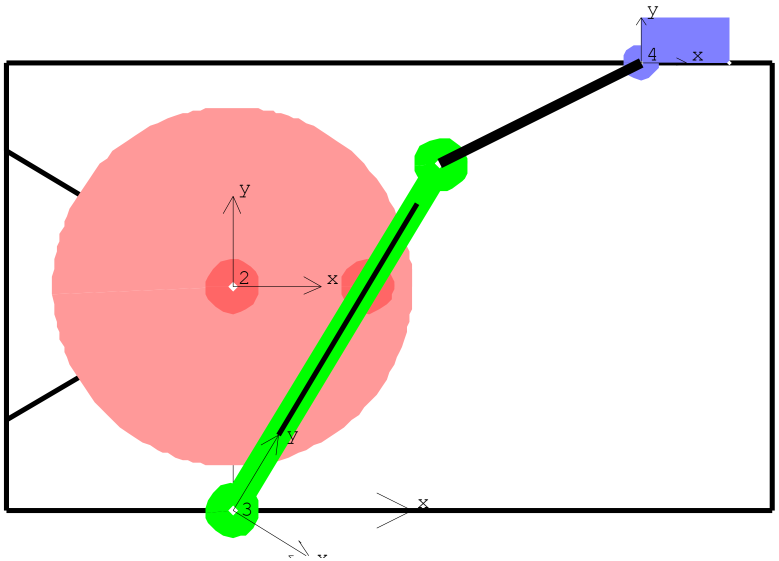
```
deslizaderaT = Graphics[
                        {Text["{0,0}", Location[deslizadera, 1], {0   , 2.0}],
                         Text["P1"   , Location[deslizadera, 1], {2.0 ,-1.8}],
                         Text["{3,0}", Location[deslizadera, 2], {0   , 1.5}],
                         Text["P2"   , Location[deslizadera, 2], {-2.0,-1.8}],
                       Text[FontForm["Deslizadera", {"Garamond", 12}], {10.5, 9.5}] }];
```

```
plot4 = Show[ {deslizaderaG, deslizaderaT} /. SolveMech[0.],
        PlotRange->{{8,12}, {9,12}},
        AspectRatio->Automatic];
```

```
conexionG = Graphics[{
                    Thickness[.010],
                    Line[{Location[colisa, 2], Location[deslizadera, 1]}]}];
```

ANALISIS CINEMATICO MECANISMOS
(4) ANALISIS POSICION - RESULTADOS

```
?SolveMech
SolveMech[t] attempts to find a location solution to the current model at
   time t.  SolveMech returns a list of rules containing the global
   coordinates of each body.  SolveMech[{t1, t2,...tn}] returns a nested
   list of solution rules containing solution points at all of the ti.
   SolveMech accepts the Solution option to determine what order of
   solution to seek.  Interpolation->True causes SolveMech to interpolate
   the solution rules returned.
SolveMech[.0]
{T -> 0., X2 -> 0., Y2 -> 5., Th2 -> 0., X3 -> 0., Y3 -> 0.,
  Th3 -> -0.54042, X4 -> 9.07905, Y4 -> 10., Th4 -> 0.}
Show[{ baseG, manivelaG, colisaG, deslizaderaG, conexionG} /. SolveMech[.0],
     PlotRange -> {{-6,15},{-1,12}}, AspectRatio -> Automatic];
```



ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - FUNCIONES **SALIDA RESULTADOS**

```
?Location
Location is a setting for the Solution option for
   certain Mech functions.  Setting
   Solution->Location causes such functions to seek
   a solution for the location of the current
   mechanism.  Location[point] returns the global
   coordinates, {X, Y} or {X, Y, Z}, of the
   specified point.

Location[Point[colisa, 2]]
{X3 - 9 Sin[Th3], Y3 + 9 Cos[Th3]}

Location[colisa, 2]
{X3 - 9 Sin[Th3], Y3 + 9 Cos[Th3]}

%/.SolveMech[.3]
%%/.SolveMech[.4]
{-1.05511, 8.93794}
{-3.03987, 8.47108}
```

ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - FUNCIONES **SALIDA RESULTADOS**

```
?Distance
Distance[point1, point2] returns the absolute
   distance between the two points.

Distance[Point[manivela, 1], Point[deslizadera, 2]]
                                 2
Sqrt[(-X2 + X4 + 2 Cos[Th4])  +
                               2
   (-Y2 + Y4 + 2 Sin[Th4]) ]

%/.SolveMech[.1]
%%/.SolveMech[.2]
11.0022
9.38401

Distance[Point[manivela, 1], Point[manivela, 2]]
3
```

ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - FUNCIONES **SALIDA RESULTADOS**

```
?PointToLineDistance
PointToLineDistance[point, axis] returns the
   shortest distance from the point to the axis.

PointToLineDistance[Point[manivela, 1], Line[colisa, 1, 2]]
9 (-X2 + X3) Cos[Th3] - 9 (Y2 - Y3) Sin[Th3]
--------------------------------------------
                     9
%/.SolveMech[.2]
%%/.SolveMech[.25]
0.58617
          -16
1.14807 10
```

ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - FUNCIONES **SALIDA RESULTADOS**

```
?Angle
Angle is an option for Mech graphics functions that
   specifies the range of the revolution angle of
   revolved objects.  Angle->{a, b} revolves the
   object from a to b radians.  The default setting
   is Angle->{0, 2 Pi}.  In Mech2D, Angle[vector]
   returns the direction angle of the vector object
   relative to the global X-axis.

Angle[Line[colisa, 2, deslizadera, 1]]
ArcTan[-X3 + X4 + 9 Sin[Th3], -Y3 + Y4 - 9 Cos[Th3]]

(%/.SolveMech[.3])/N[Degree]
12.2638
```

ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - FUNCIONES **SALIDA RESULTADOS**

```
?Rotation
Rotation[bnum] returns the angle of rotation
   associated with the orientation of 2D body bnum,
   or the list {ang, axis} specifying the angle and
   axis of rotation associated with 3D body bnum.
Rotation[colisa]
Th3

?IntersectionPoint
IntersectionPoint[axis1, axis2] returns the point of
   intersection of axis1 and axis2 in global 2D
   space, or the point of intersection of axis1 and
   a plane that is normal to axis2 in global 3D
   space.  In 3D, axis2 is typically a Plane object.
IntersectionPoint[Line[colisa, 2, deslizadera, 1],
                  Line[base, 0, {0,1}]] /. LastSolve[]
{0., 9.16729}
```

ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - FUNCIONES **SALIDA RESULTADOS**

```
?ProjectOnLine
ProjectOnLine[vector1, vector2] projects vector1
   onto vector2 and returns the vector component of
   vector1 that is in the direction of vector2.
   ProjectOnLine[point, axis] returns the
   coordinates of the point orthogonally projected
   onto the axis.

ProjectOnLine[Point[manivela, 2],
              Line[colisa, 2, deslizadera, 1]]/.LastSolve[]
{-1.15799, 8.91557}
```

ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - FUNCIONES **ALGEBRA VECTORIAL**

```
?Direction
Direction[vector] returns the direction vector of a
   Mech vector object in global coordinates.
   Direction is an option for Limit. With Direction
   -> 1, the limit is taken from below. With
   Direction -> -1, the limit is taken from above.
   Direction -> Automatic uses Direction -> -1
   except for limits at Infinity, where it is
   equivalent to Direction -> 1.

Direction[Line[colisa, 2,deslizadera, 1]]
{-X3 + X4 + 9 Sin[Th3], -Y3 + Y4 - 9 Cos[Th3]}

%/.SolveMech[.45]
{4.62876, 1.89067}
```

ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - FUNCIONES **ALGEBRA VECTORIAL**

```
?Magnitude
Magnitude[vector] returns the length of the vector
   or Mech vector object.  Magnitude is also an
   option for Force and Moment.  Setting
   Magnitude->Relative causes the magnitude of the
   applied load to be equal to the length of the
   load vector times the given magnitude.  The
   default setting is Magnitude->Absolute.

Magnitude[Line[colisa, 2, deslizadera, 1]]/.LastSolve[]
5.

Magnitude[{9, 12}]
15
```

ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - FUNCIONES **ALGEBRA VECTORIAL**

```
?Unit
Unit[vector] returns a unit vector pointed in the
   direction of the Mech vector object.

Unit[{3, 5}]
    3         5
{--------, --------}
 Sqrt[34]  Sqrt[34]

?Cross
Cross[vector1, vector2] returns the cross product of
   the direction vectors of the two Mech vector
   objects.

Cross[Line[colisa, 2, deslizadera, 1], {0, 3}]
3 (-X3 + X4 + 9 Sin[Th3])

%/.SolveMech[.5]
13.3458
```

ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - FUNCIONES **ALGEBRA VECTORIAL**

```
?RotationMatrix
RotationMatrix[bnum] returns the 2D/3D rotation
   matrix associated with body bnum.

RotationMatrix[ deslizadera ]
{{Cos[Th4], -Sin[Th4]}, {Sin[Th4], Cos[Th4]}}

Location[deslizadera, 0] +
RotationMatrix[ deslizadera ] . {3, 8}
{X4 + 3 Cos[Th4] - 8 Sin[Th4],
  Y4 + 8 Cos[Th4] + 3 Sin[Th4]}

Location[deslizadera, {3, 8}]
{X4 + 3 Cos[Th4] - 8 Sin[Th4],
  Y4 + 8 Cos[Th4] + 3 Sin[Th4]}

%/.LastSolve[]
{2.81812, 18.}
```

ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - FUNCIONES **ALGEBRA VECTORIAL**

```
?PointToLocal
PointToLocal[bnum, point] returns the local
   coordinates of the point object in the coordinate
   system of body bnum.

Location[deslizadera, {3, 8}]
{X4 + 3 Cos[Th4] - 8 Sin[Th4],
  Y4 + 8 Cos[Th4] + 3 Sin[Th4]}

PointToLocal[ deslizadera, % ] /. LastSolve[]
{3., 8.}
```

ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - FUNCIONES **CONVERSION COORDENADAS**

```
?PolarToXY
PolarToXY[{radius, angle}] converts the given point
   from polar to Cartesian coordinates and returns
   {x, y}.

PolarToXY[{4, Pi/3}]
{2, 2 Sqrt[3]}

?XYToPolar
XYToPolar[{x, y}] converts the given point from
   Cartesian to polar coordinates and returns
   {radius, angle}.

PolarToXY[{4, Pi/3}]
{2, 2 Sqrt[3]}

XYToPolar[%]
{4, ArcTan[2, 2 Sqrt[3]]}

XYToPolar[ Location[deslizadera, 2] ]/.LastSolve[]
{10.1639, 1.39095}
```

ANALISIS CINEMATICO MECANISMOS

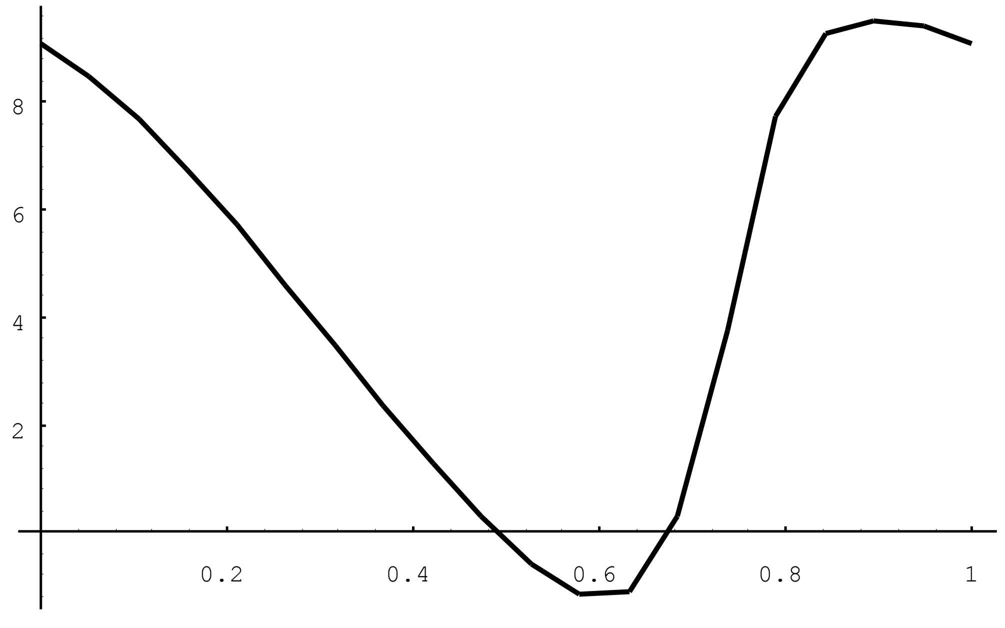
(5) ANALISIS POSICION - GENERACION **GRAFICOS RESULTADOS**

```
SetGuess[]
{T -> 1., X2 -> 0, Y2 -> 5, Th2 -> 0, X3 -> 0, Y3 -> 0, Th3 -> -
0.5,   X4 -> 6, Y4 -> 10, Th4 -> 0}
longitud = 3;
velocidad = 2*N[Pi];
anguloI = 0.0;
CheckSystem[]
True
?First
First[expr] gives the first element in expr.

First[ posicionT = SolveMech[{0,1}, 20] ]
{T -> 0, X2 -> 0., Y2 -> 5., Th2 -> 0., X3 -> 0., Y3 -> 0.,   Th3
-> -0.54042, X4 -> 9.07905, Y4 -> 10., Th4 -> 0.}
```

ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - GENERACION **GRAFICOS RESULTADOS**

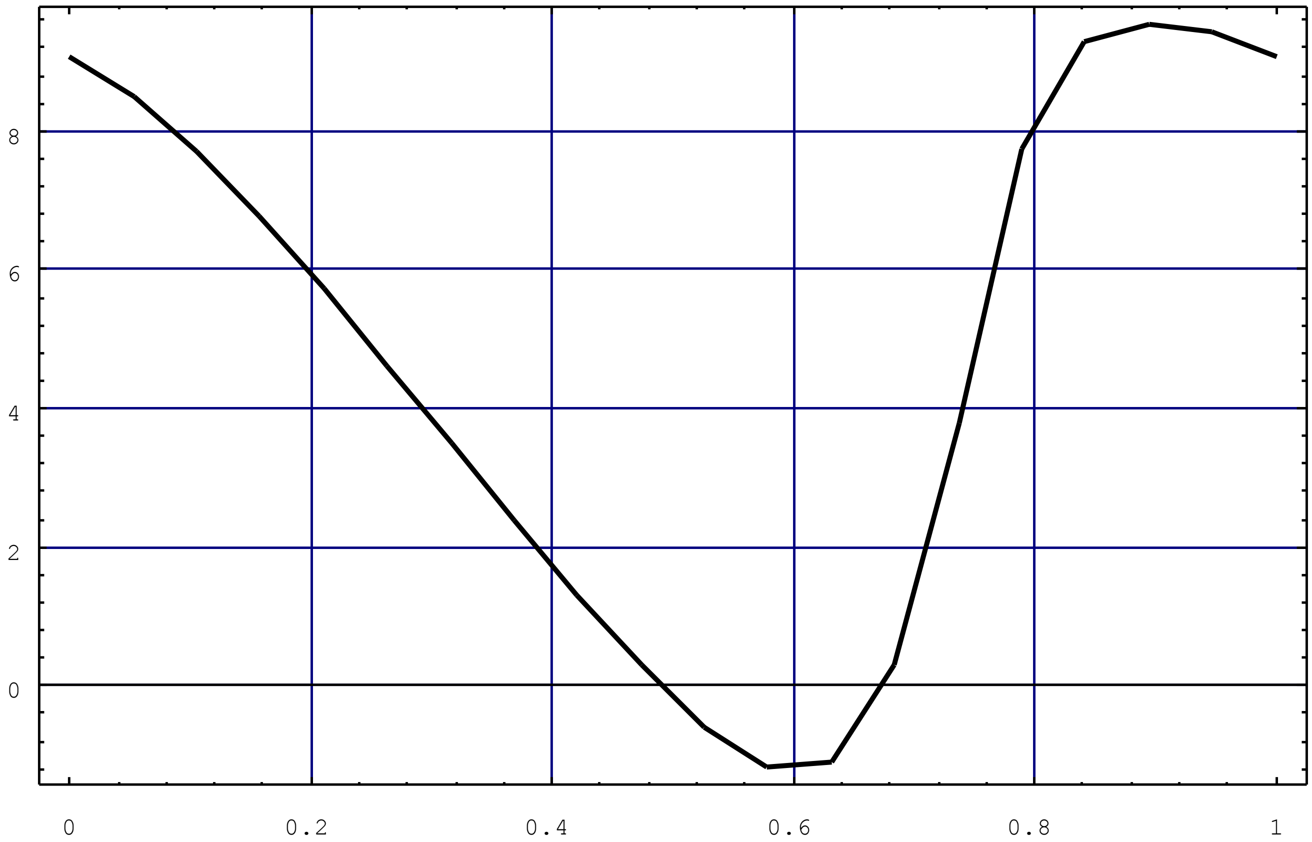
```
?ListPlot
ListPlot[{y1, y2, ...}] plots a list of values. The
   x coordinates for each point are taken to be 1,
   2, .... ListPlot[{{x1, y1}, {x2, y2}, ...}] plots
   a list of values with specified x and y
   coordinates.
ListPlot[ {T, X4}/.posicionT, PlotJoined->True];
```



ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - GENERACION **GRAFICOS RESULTADOS**

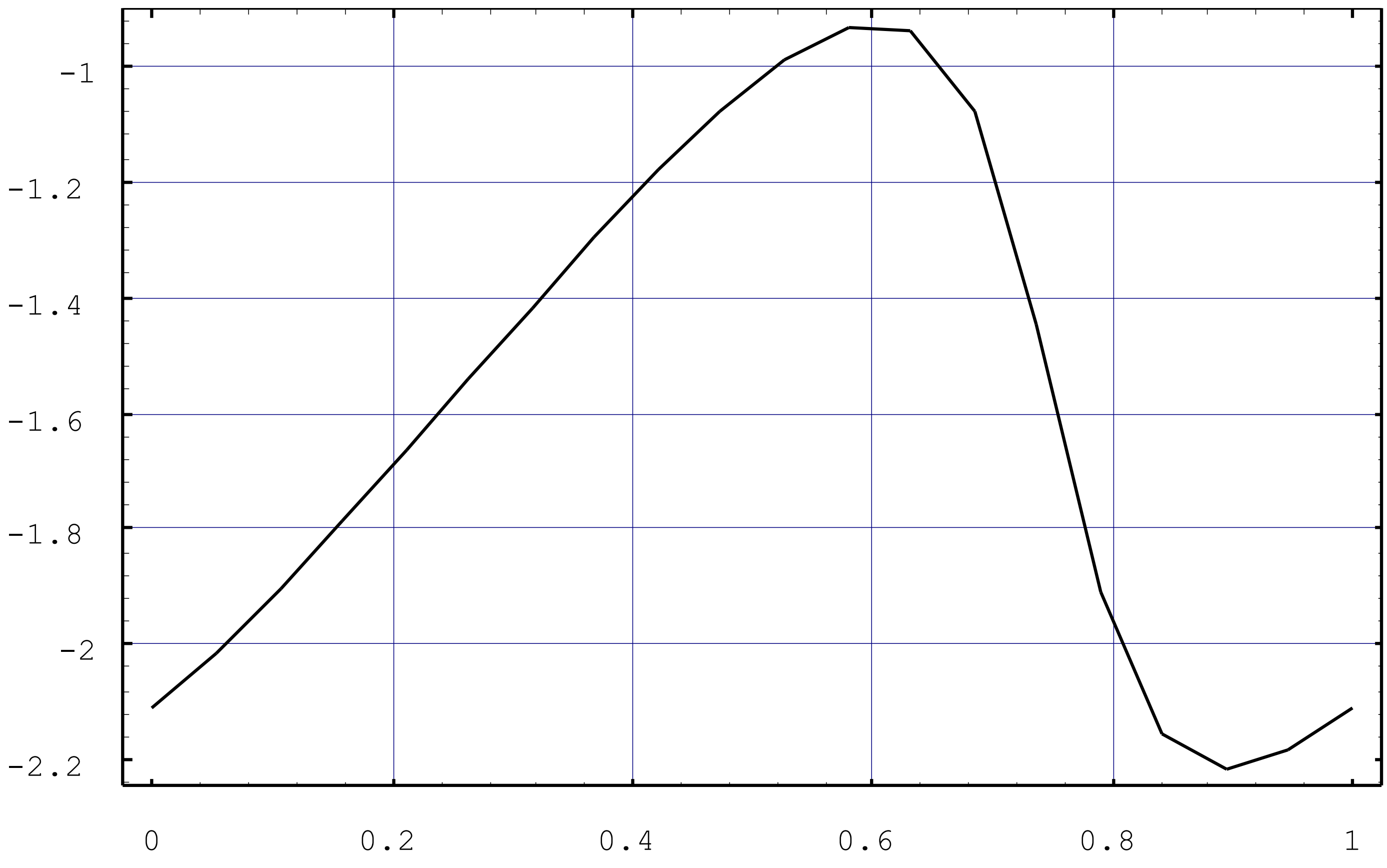
```
ListPlot[ {T, X4}/.posicionT, PlotJoined->True,
                              GridLines -> Automatic,
                              Frame -> True];
```



ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - GENERACION **GRAFICOS RESULTADOS**

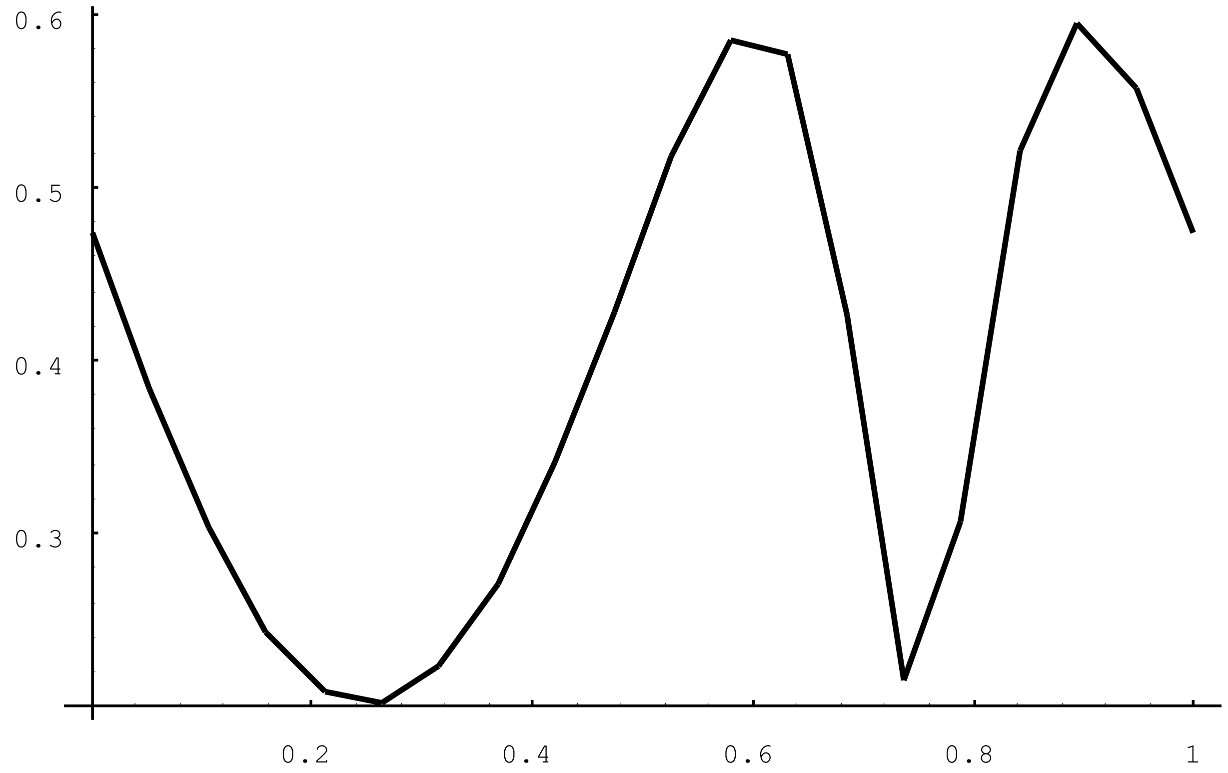
```
Aplot = ListPlot[{T,Angle[colisa, 2, 1]}/.posicionT,
        PlotJoined->True, Frame->True,
        GridLines->Automatic, PlotStyle->{Thickness[.002]}];
```



ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - GENERACION **GRAFICOS RESULTADOS**

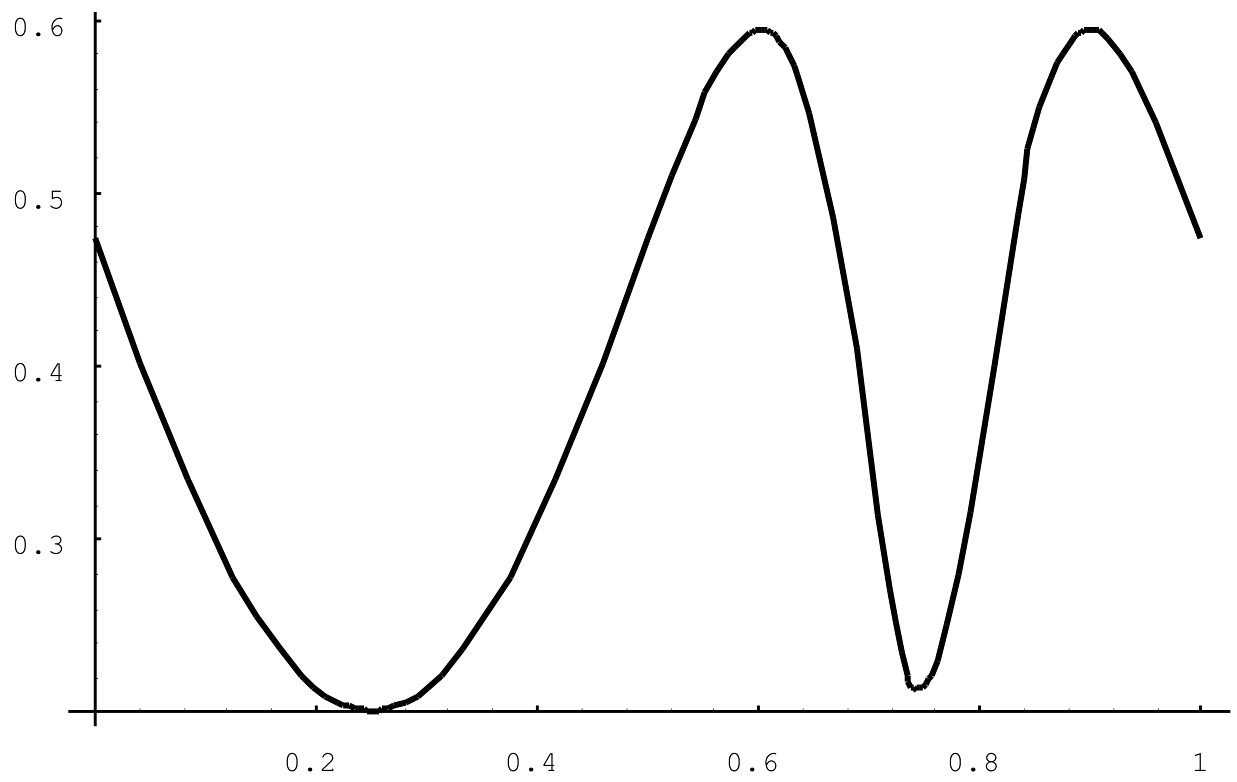
```
ListPlot[ {T, Angle[Line[colisa, 2, deslizadera, 1]]}/.posicionT,
                    PlotJoined->True];
```



ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - GENERACION **GRAFICOS RESULTADOS**

```
TimeInterpolate[ Angle[Line[colisa, 2, deslizadera, 1]],
  posicionT ]
InterpolatingFunction[{0, 1.}, <>]
Plot[%[T], {T, 0, 1}];
```



ANALISIS CINEMATICO MECANISMOS

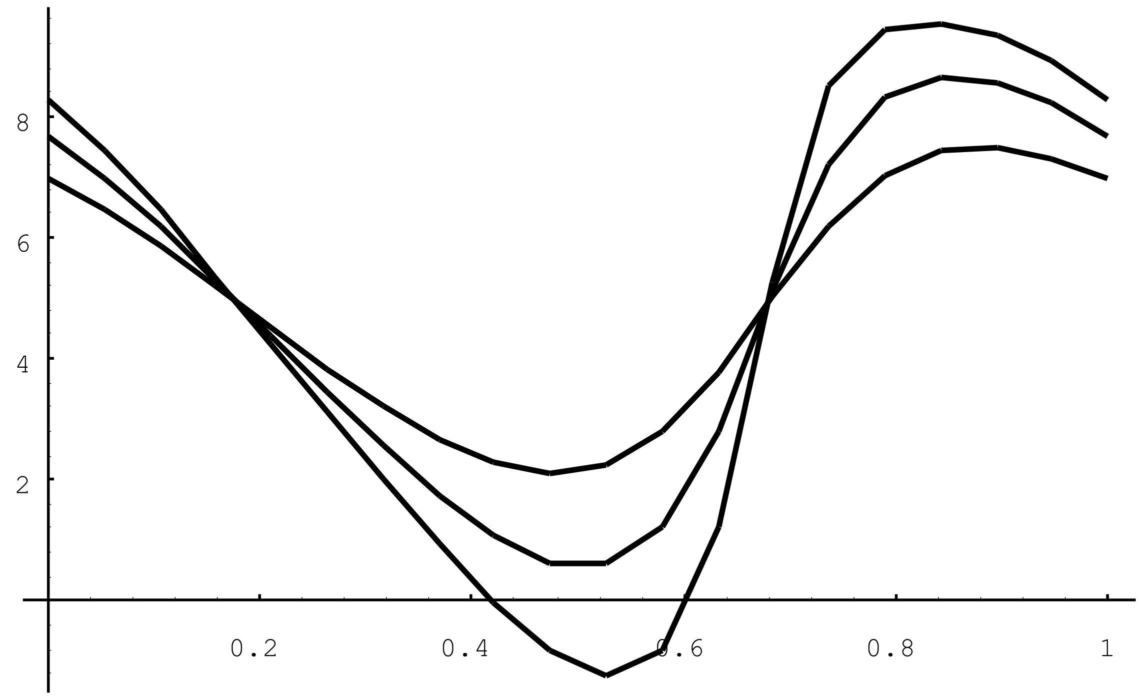
(5) ANALISIS POSICION - GENERACION **GRAFICOS RESULTADOS**

```
SetGuess[]
{T -> 1., X2 -> 0, Y2 -> 5, Th2 -> 0, X3 -> 0, Y3 -> 0, Th3 -> -
0.5,   X4 -> 6, Y4 -> 10, Th4 -> 0}
longitud = 3;
velocidad = 2*N[Pi];
anguloI = 0.0;
CheckSystem[]
True

positionT2 = Table[SetGuess[];
                   SolveMech[{0, 1}, 20],
                   {longitud, 1.5, 3., 0.75}];
```

ANALISIS CINEMATICO MECANISMOS

(5) ANALISIS POSICION - GENERACION **GRAFICOS RESULTADOS**

```
Show[ListPlot[#, PlotJoined->True, DisplayFunction->Identity]&/@
            ({T, X4}/.positionT2), DisplayFunction->$DisplayFunction ];
```



ANALISIS CINEMATICO MECANISMOS

(6) **ANALISIS VELOCIDADES**

```
SetGuess[]
{T -> 0.6, X2 -> 0, Y2 -> 5, Th2 -> 0, X3 -> 0,
  Y3 -> 0, Th3 -> -0.5, X4 -> 6, Y4 -> 10, Th4 -> 0}
SolveMech[0.15, Solution->Velocity]
{T -> 0.15, X2 -> 0., Y2 -> 5., Th2 -> 0.942478,
  X3 -> 0., Y3 -> 0., Th3 -> -0.233107,
  X4 -> 6.92194, Y4 -> 10., Th4 -> 0., X2d -> 0.,
  Y2d -> 0., Th2d -> 6.28319, X3d -> 0., Y3d -> 0.,
  Th3d -> 2.27897, X4d -> -18.7395, Y4d -> 0.,
  Th4d -> 0.}
```

ANALISIS CINEMATICO MECANISMOS

(6) ANALISIS VELOCIDAD - ECUACIONES VELOCIDAD

VECTOR VELOCIDADES GENERALIZADAS

$$\dot{\mathbf{q}}_i = \left[\dot{\mathbf{r}}_i, \quad \dot{\phi}_i\right]^T$$

```
MatrixForm[Velocity[All]]
X2d
Y2d
Th2d
X3d
Y3d
Th3d
X4d
Y4d
Th4d
```

ECUACION VECTORIAL DE LAS VELOCIDADES:

$$\Phi_{\mathbf{q}}\dot{\mathbf{q}} = -\Phi_{\mathbf{t}} \equiv \nu$$

```
MatrixForm[VelocityTerms[All]]
0
0
-6.28319
0
0
0
0
0
0
Dimensions[VelocityTerms[All]]
{9}
```

ANALISIS CINEMATICO MECANISMOS

(6) ANALISIS VELOCIDAD - ECUACIONES VELOCIDAD

ECUACION VECTORIAL DE LAS VELOCIDADES:

$$\dot{\Phi} = \Phi_{\mathbf{q}}\dot{\mathbf{q}} + \Phi_{\mathbf{t}} = \mathbf{0}$$

```
Jacobian[All,All].Velocity[All]+VelocityTerms[All]
{-X2d, -Y2d, -6.28319 + Th2d, -X3d, -Y3d,
   -9 X2d Cos[Th3] + 9 X3d Cos[Th3] - 9 Y2d Sin[Th3] + 9 Y3d Sin[Th3] +
    Th3d (-9 Cos[Th3] (Y2 - Y3 + 3 Sin[Th2]) -
      9 (-X2 + X3 - 3 Cos[Th2]) Sin[Th3]) +
    Th2d (27 Cos[Th3] Sin[Th2] - 27 Cos[Th2] Sin[Th3]),
   2 Y3d (Y3 - Y4 + 9 Cos[Th3]) - 2 Y4d (Y3 - Y4 + 9 Cos[Th3]) +
    2 X3d (X3 - X4 - 9 Sin[Th3]) - 2 X4d (X3 - X4 - 9 Sin[Th3]) +
   Th3d (-18 Cos[Th3] (X3 - X4 - 9 Sin[Th3]) -
       18 (Y3 - Y4 + 9 Cos[Th3]) Sin[Th3]), 34 Th4d Cos[Th4],
   -2 Y4d Cos[Th4] + 2 X4d Sin[Th4] +
   Th4d (2 (5 + X4) Cos[Th4] - 2 (10 - Y4) Sin[Th4])}
```

ANALISIS CINEMATICO MECANISMOS

(6) ANALISIS VELOCIDADES - **SALIDA RESULTADOS**

```
?DDistanceDT
DDistanceDT[point1, point2] returns the time derivative of the absolute
distance between the two points.

DDistanceDT[Point[manivela, 1], Point[deslizadera, 1]]/.LastSolve[]
-15.1909

Distance[Point[manivela, 1], Point[deslizadera, 1]]
                  2             2
Sqrt[(-X2 + X4)  + (-Y2 + Y4) ]

Dt[%, T]
2 (-X2 + X4) (-X2d + X4d) + 2 (-Y2 + Y4) (-Y2d + Y4d)
-----------------------------------------------------
                                2             2
          2 Sqrt[(-X2 + X4)  + (-Y2 + Y4) ]

%/.LastSolve[]
-15.1909
```

ANALISIS CINEMATICO MECANISMOS

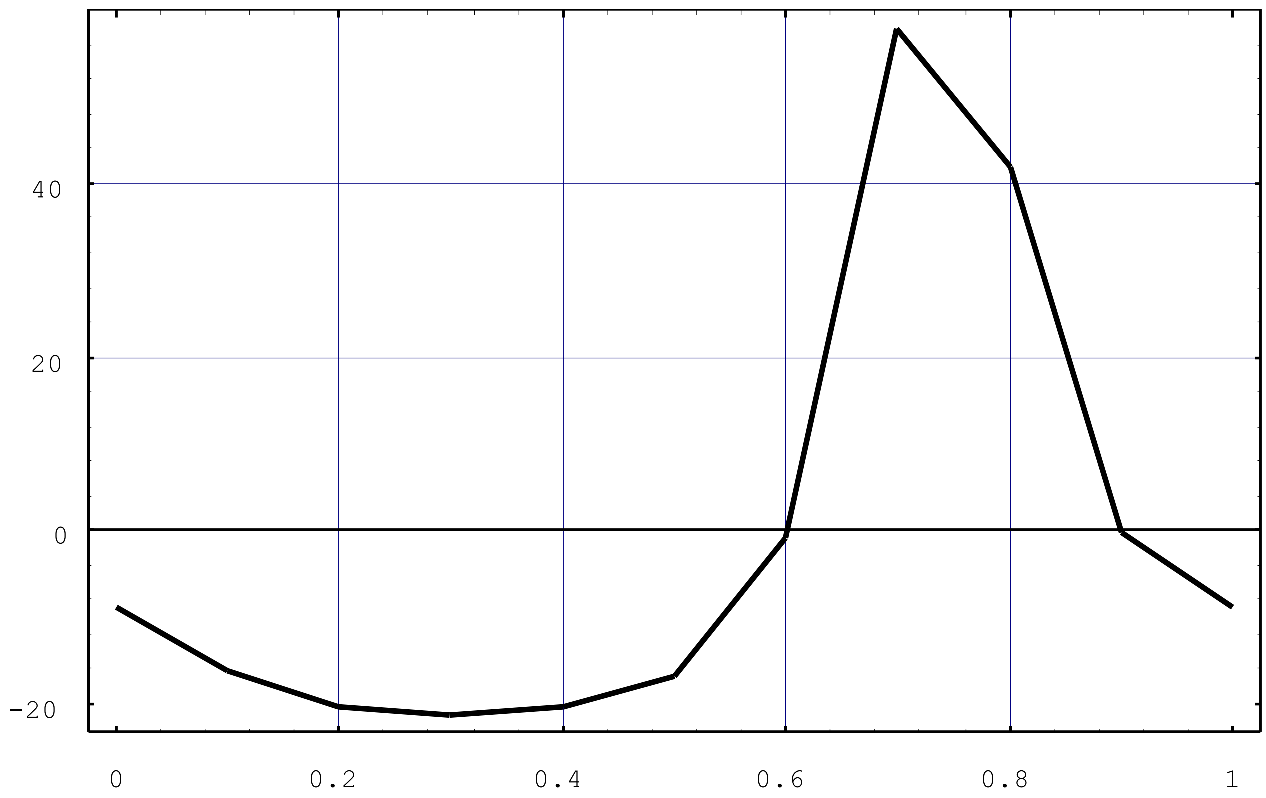
(6) ANALISIS VELOCIDADES - **GRAFICOS RESULTADOS**

```
postab = SolveMech[{0,1},11];

veltab = SolveMech[postab, Solution->Velocity];

?Velocity
Velocity is a setting for the Solution option for
   certain Mech functions.  Setting
   Solution->Velocity causes such functions to seek
   a solution for the location and velocity of the
   current mechanism.  Velocity[point] returns the
   global velocity vector, {dX/dT, dY/dT (,dZ/dT)},
   of the specified point.
```

ANALISIS CINEMATICO MECANISMOS

(6) ANALISIS VELOCIDADES - **GRAFICOS RESULTADOS**

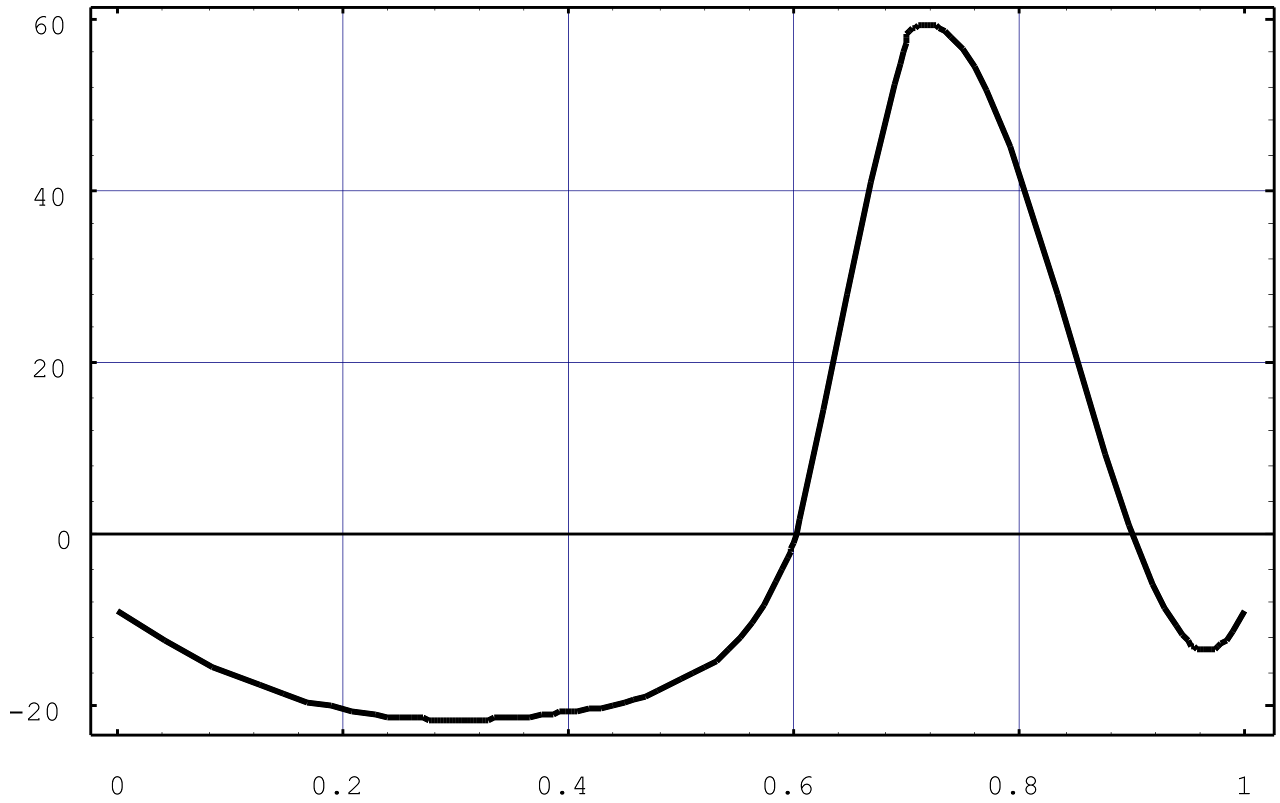
```
ListPlot[{T, Velocity[deslizadera,2][[1]]}/.veltab,
         PlotJoined->True, Frame->True,
         GridLines->Automatic];
```



ANALISIS CINEMATICO MECANISMOS

(6) ANALISIS VELOCIDADES - **GRAFICOS RESULTADOS**

```
pl = TimeInterpolate[Velocity[deslizadera,2][[1]], veltab];

Plot[ pl[T], {T, 0, 1.}, Frame->True, GridLines->Automatic];
```



ANALISIS CINEMATICO MECANISMOS
(7) **ANALISIS ACELERACIONES**

```
SetGuess[]
{T -> 1., X2 -> 0, Y2 -> 5, Th2 -> 0, X3 -> 0,
  Y3 -> 0, Th3 -> -0.5, X4 -> 6, Y4 -> 10,
  Th4 -> 0, X2d -> 0., Y2d -> 0., Th2d -> 6.28319,
  X3d -> 0., Y3d -> 0., Th3d -> 1.6632,
  X4d -> -8.88405, Y4d -> 0., Th4d -> 0.}

SolveMech[0.15,Solution->Acceleration]
{T -> 0.15, X2 -> 0., Y2 -> 5., Th2 -> 0.942478,
  X3 -> 0., Y3 -> 0., Th3 -> -0.233107,
  X4 -> 6.92194, Y4 -> 10., Th4 -> 0., X2d -> 0.,
  Y2d -> 0., Th2d -> 6.28319, X3d -> 0., Y3d -> 0.,
  Th3d -> 2.27897, X4d -> -18.7395, Y4d -> 0.,
  Th4d -> 0., X2dd -> 0., Y2dd -> 0., Th2dd -> 0.,
  X3dd -> 0., Y3dd -> 0., Th3dd -> 1.64018,
  X4dd -> -40.9024, Y4dd -> 0., Th4dd -> 0.}
```

ANALISIS CINEMATICO MECANISMOS

(7) ANALISIS ACELERACION - ECUACIONES ACELERACION

VECTOR VACELERACIONES GENERALIZADAS

$$\ddot{\mathbf{q}}_i = \left[\ddot{\mathbf{r}}_i , \quad \ddot{\phi}_i \right]^T$$

MatrixForm[Acceleration[All]]

```
X2dd
Y2dd
Th2dd
X3dd
Y3dd
Th3dd
X4dd
Y4dd
Th4dd
```

ECUACION VECTORIAL DE LAS ACELERACIONES

$$\Phi_{\mathbf{q}} \ddot{\mathbf{q}} = -(\Phi_{\mathbf{q}} \dot{\mathbf{q}})_{\mathbf{q}} \dot{\mathbf{q}} - 2 \Phi_{\mathbf{q}t} \dot{\mathbf{q}} - \Phi_{\mathbf{tt}} = \boldsymbol{\gamma}$$

AccelerationTerms[All]

```
{0, 0, 0, 0, 0, -9 Th3d Y2d Cos[Th3] + 9 Th3d Y3d Cos[Th3] +    9 Th3d X2d Sin[Th3] - 9 Th3d X3d Sin[Th3] +
    Th2d (Th3d (-27 Cos[Th2] Cos[Th3] - 27 Sin[Th2] Sin[Th3]) +       Th2d (27 Cos[Th2] Cos[Th3] + 27 Sin[Th2] Sin[Th3])) +
   Th3d (-9 Y2d Cos[Th3] + 9 Y3d Cos[Th3] + 9 X2d Sin[Th3] -       9 X3d Sin[Th3] + Th2d (-27 Cos[Th2] Cos[Th3] -
         27 Sin[Th2] Sin[Th3]) +       Th3d (-9 (-X2 + X3 - 3 Cos[Th2]) Cos[Th3] + 9 (Y2 - Y3 + 3 Sin[Th2]) Sin[Th3])),
   X3d (2 X3d - 2 X4d - 18 Th3d Cos[Th3]) +  X4d (-2 X3d + 2 X4d + 18 Th3d Cos[Th3]) +  Y3d (2 Y3d - 2 Y4d - 18 Th3d Sin[Th3]) +
   Y4d (-2 Y3d + 2 Y4d + 18 Th3d Sin[Th3]) +    Th3d (-18 X3d Cos[Th3] + 18 X4d Cos[Th3] - 18 Y3d Sin[Th3] +
18 Y4d Sin[Th3] + Th3d (162 Cos[Th3]  -         18 Cos[Th3] (Y3 - Y4 + 9 Cos[Th3]) +
    18 (X3 - X4 - 9 Sin[Th3]) Sin[Th3] + 162 Sin[Th3] )),
         2  -34 Th4d  Sin[Th4], 2 Th4d X4d Cos[Th4] + 2 Th4d Y4d Sin[Th4] +    Th4d (2 X4d Cos[Th4] + 2 Y4d Sin[Th4] +
       Th4d (-2 (10 - Y4) Cos[Th4] - 2 (5 + X4) Sin[Th4]))}
```

Dimensions[AccelerationTerms[All]]

{9}

ANALISIS CINEMATICO MECANISMOS

(7) ANALISIS ACELERACION - ECUACIONES ACELERACION

ECUACION VECTORIAL DE LAS ACELERACIONES

$$\Phi_{\mathbf{q}}\ddot{\mathbf{q}} + (-\boldsymbol{\gamma}) = \mathbf{0}$$

MatrixForm[Jacobian[All,All].Acceleration[All]+AccelerationTerms[All]]

```
-X2dd-Y2ddTh2dd-X3dd-Y3dd-9 X2dd Cos[Th3] + 9 X3dd Cos[Th3] - 9 Th3d Y2d Cos[Th3] +   9 Th3d Y3d Cos[Th3] + 9 Th3d X2d Sin[Th3] - 9 Th3d X3d Sin[Th3] -
  9 Y2dd Sin[Th3] + 9 Y3dd Sin[Th3] +   Th3dd (-9 Cos[Th3] (Y2 - Y3 + 3 Sin[Th2]) -      9 (-X2 + X3 - 3 Cos[Th2]) Sin[Th3]) +   Th2dd (27 Cos[Th3] Sin[Th2] -
 27 Cos[Th2] Sin[Th3]) +   Th2d (Th3d (-27 Cos[Th2] Cos[Th3] - 27 Sin[Th2] Sin[Th3]) +      Th2d (27 Cos[Th2] Cos[Th3] + 27 Sin[Th2] Sin[Th3])) +
 Th3d (-9 Y2d Cos[Th3] + 9 Y3d Cos[Th3] + 9 X2d Sin[Th3] -      9 X3d Sin[Th3] + Th2d (-27 Cos[Th2] Cos[Th3] -        27 Sin[Th2] Sin[Th3]) +
   Th3d (-9 (-X2 + X3 - 3 Cos[Th2]) Cos[Th3] +        9 (Y2 - Y3 + 3 Sin[Th2]) Sin[Th3]))2 Y3dd (Y3 - Y4 + 9 Cos[Th3]) - 2 Y4dd (Y3 - Y4 + 9 Cos[Th3]) +
 X3d (2 X3d - 2 X4d - 18 Th3d Cos[Th3]) +   X4d (-2 X3d + 2 X4d + 18 Th3d Cos[Th3]) +   2 X3dd (X3 - X4 - 9 Sin[Th3]) - 2 X4dd (X3 - X4 - 9 Sin[Th3]) +
  Y3d (2 Y3d - 2 Y4d - 18 Th3d Sin[Th3]) +   Y4d (-2 Y3d + 2 Y4d + 18 Th3d Sin[Th3]) +   Th3dd (-18 Cos[Th3] (X3 - X4 - 9 Sin[Th3]) -
   18 (Y3 - Y4 + 9 Cos[Th3]) Sin[Th3]) +   Th3d (-18 X3d Cos[Th3] + 18 X4d Cos[Th3] - 18 Y3d Sin[Th3] +
18 Y4d Sin[Th3] + Th3d (162 Cos[Th3]  -        18 Cos[Th3] (Y3 - Y4 + 9 Cos[Th3]) +
  18 (X3 - X4 - 9 Sin[Th3]) Sin[Th3] + 162 Sin[Th3] ))     Cos[Th4] - 34 Th4d  Sin[Th4]2 Th4d X4d Cos[Th4] - 2 Y4dd Cos[Th4] + 2 X4dd Sin[Th4] +

   2 Th4d Y4d Sin[Th4] + Th4dd    (2 (5 + X4) Cos[Th4] - 2 (10 - Y4) Sin[Th4]) +   Th4d (2 X4d Cos[Th4] + 2 Y4d  Sin[Th4] +
   Th4d (-2 (10 - Y4) Cos[Th4] - 2 (5 + X4) Sin[Th4]))
```

ANALISIS CINEMATICO MECANISMOS

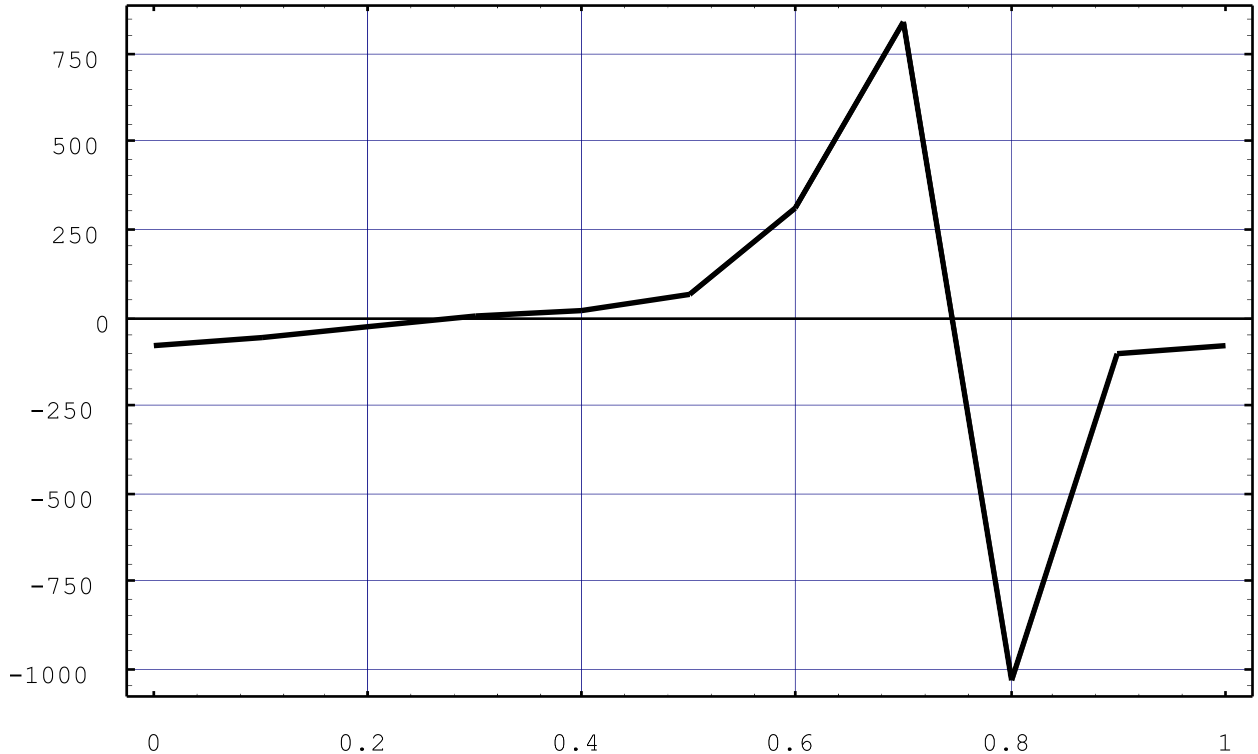
(7) ANALISIS ACELERACIONES - **SALIDA RESULTADOS**

```
?D2DistanceDT2
D2DistanceDT2[point1, point2] returns the second
   time derivative of the absolute distance between
   the two points.

D2DistanceDT2[Point[manivela, 1], Point[deslizadera, 1]]/.LastSolve[]
-19.056

?Dt
Dt[f, x] gives the total derivative of f with
   respect to x. Dt[f] gives the total differential
   of f. Dt[f, {x, n}] gives the nth total
   derivative with respect to x. Dt[f, x1, x2, ...]
   gives a mixed total derivative.

Dt[Distance[Point[manivela, 1], Point[deslizadera, 1]],{T, 2}]/.
  LastSolve[]
-19.056
```

ANALISIS CINEMATICO MECANISMOS

(7) ANALISIS ACELERACIONES - **GRAFICOS RESULTADOS**

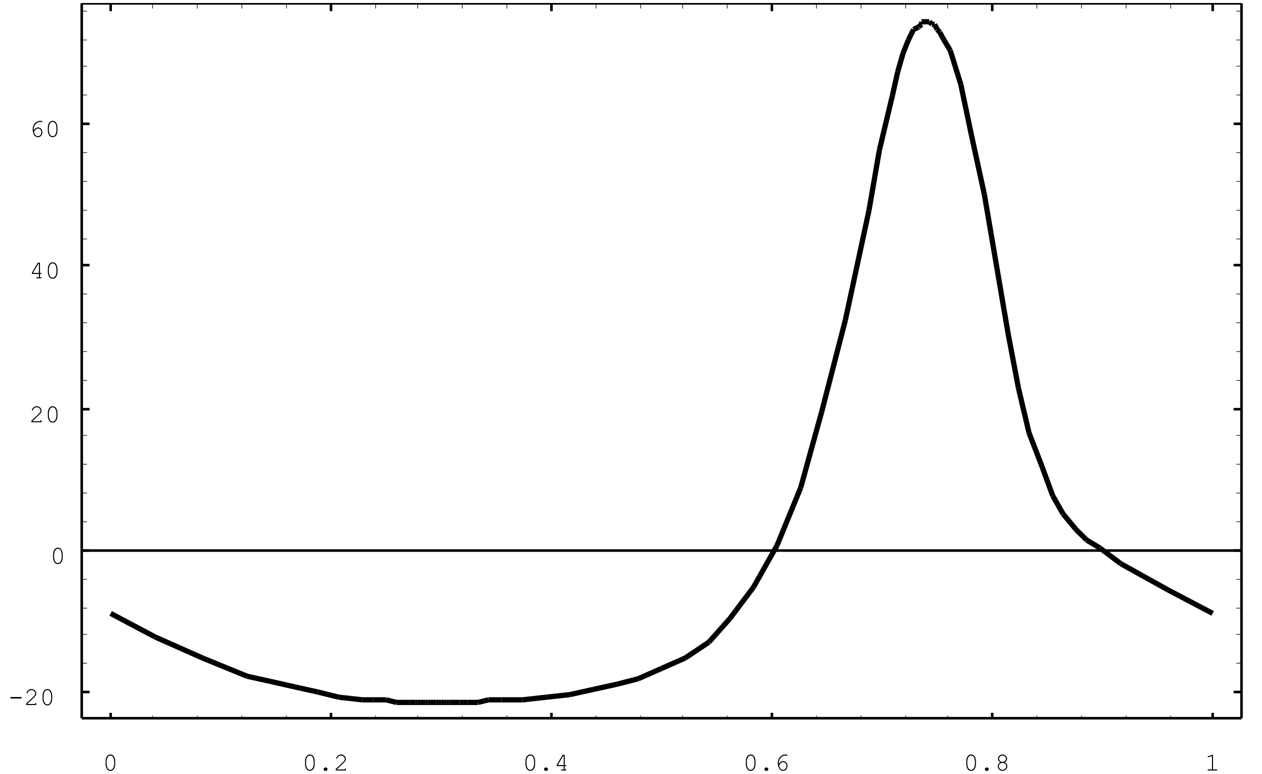
```
acctab = SolveMech[veltab, Solution->Acceleration];

ListPlot[{T, X4dd}/.acctab, PlotJoined->True, Frame->True,
                            GridLines->Automatic, PlotRange->All];
```



ANALISIS CINEMATICO MECANISMOS

(7) ANALISIS ACELERACIONES - **GRAFICOS RESULTADOS**

```
pl = TimeInterpolate[Velocity[deslizadera,2][[1]],acctab];

Plot[ pl[T], {T, 0, 1.}, Frame->True ];
```



ANALISIS CINEMATICO MECANISMOS

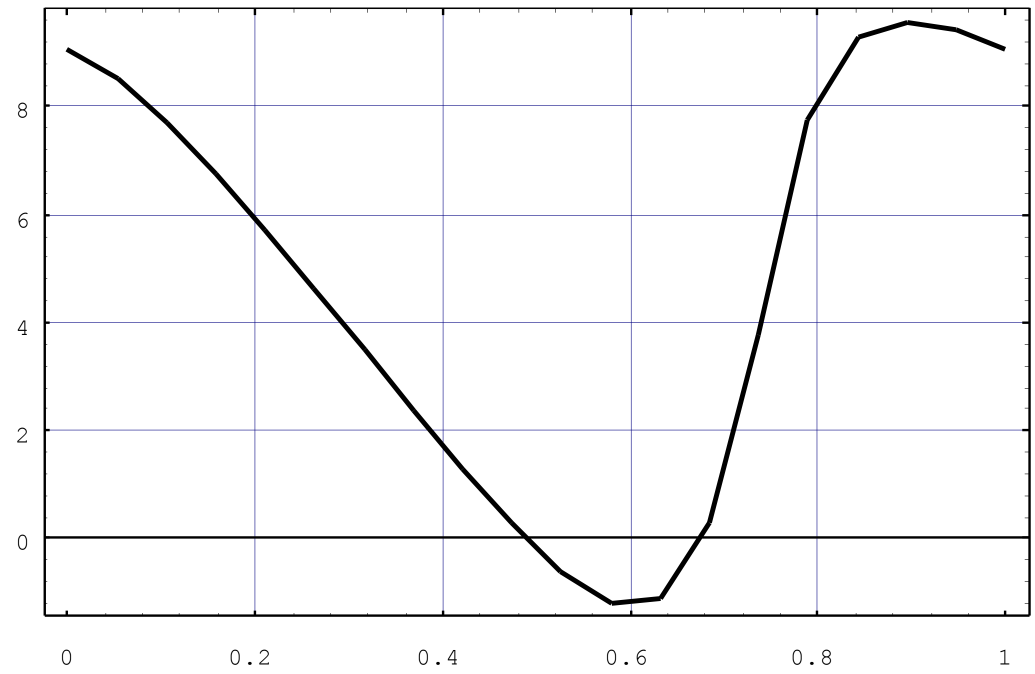
OBTENCION **VALORES MAXIMO Y MINIMO X4** (1)

```
longitud = 3;
velocidad = 2*N[Pi];
anguloI = 0.0;
CheckSystem[]
True
SetGuess[]
{T -> 0.3, X2 -> 0, Y2 -> 5, Th2 -> 0, X3 -> 0,
 Y3 -> 0, Th3 -> -0.5,   X4 -> 6, Y4 -> 10, Th4 -> 0}
First[ posicionT = SolveMech[{0,1}, 20] ]
{T -> 0, X2 -> 0., Y2 -> 5., Th2 -> 0., X3 -> 0., Y3 -> 0.,
   Th3 -> -0.54042, X4 -> 9.07905, Y4 -> 10., Th4 -> 0.}
```

ANALISIS CINEMATICO MECANISMOS

OBTENCION **VALORES MAXIMO Y MINIMO X4** (2)

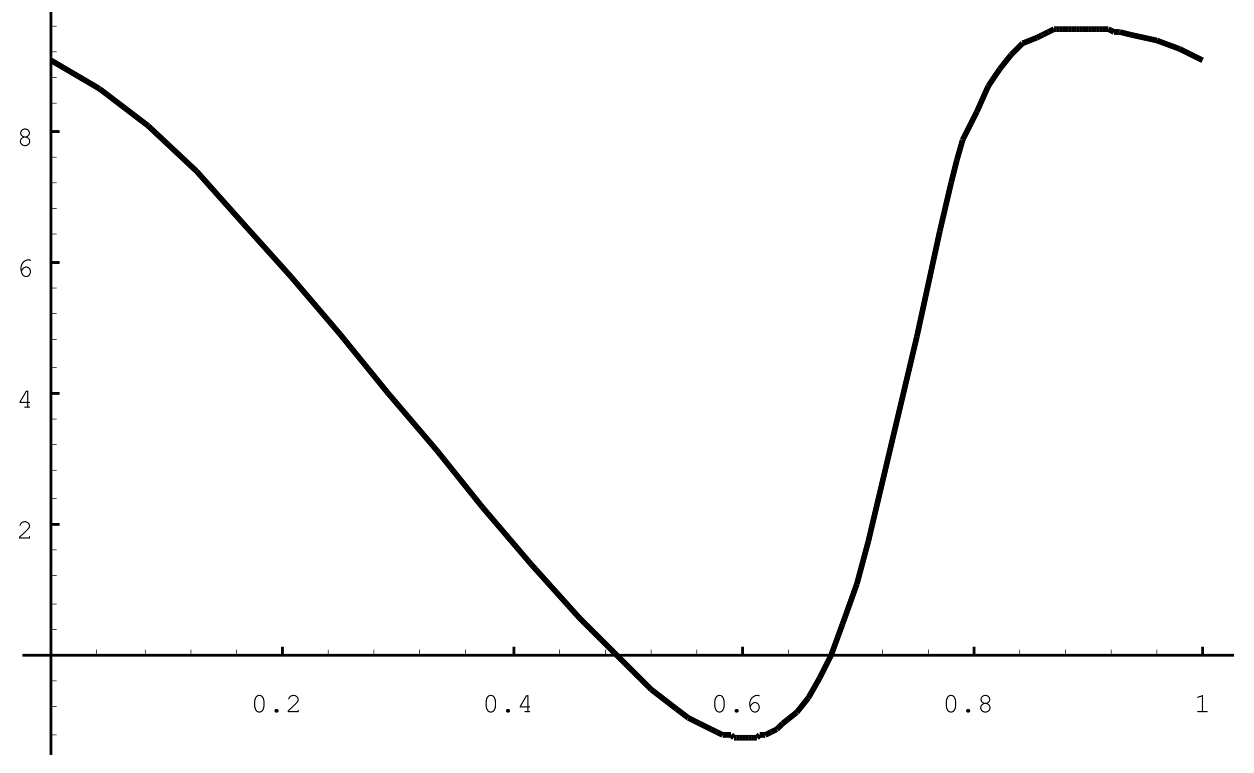
```
ListPlot[ {T, X4}/.posicionT, PlotJoined->True, GridLines -> Automatic,
                              Frame -> True];
```



ANALISIS CINEMATICO MECANISMOS

OBTENCION **VALORES MAXIMO Y MINIMO X4** (3)

```
X4t= TimeInterpolate[ X4, posicionT]
InterpolatingFunction[{0, 1.}, <>]
Plot[X4t[T], {T, 0, 1}];
```



ANALISIS CINEMATICO MECANISMOS

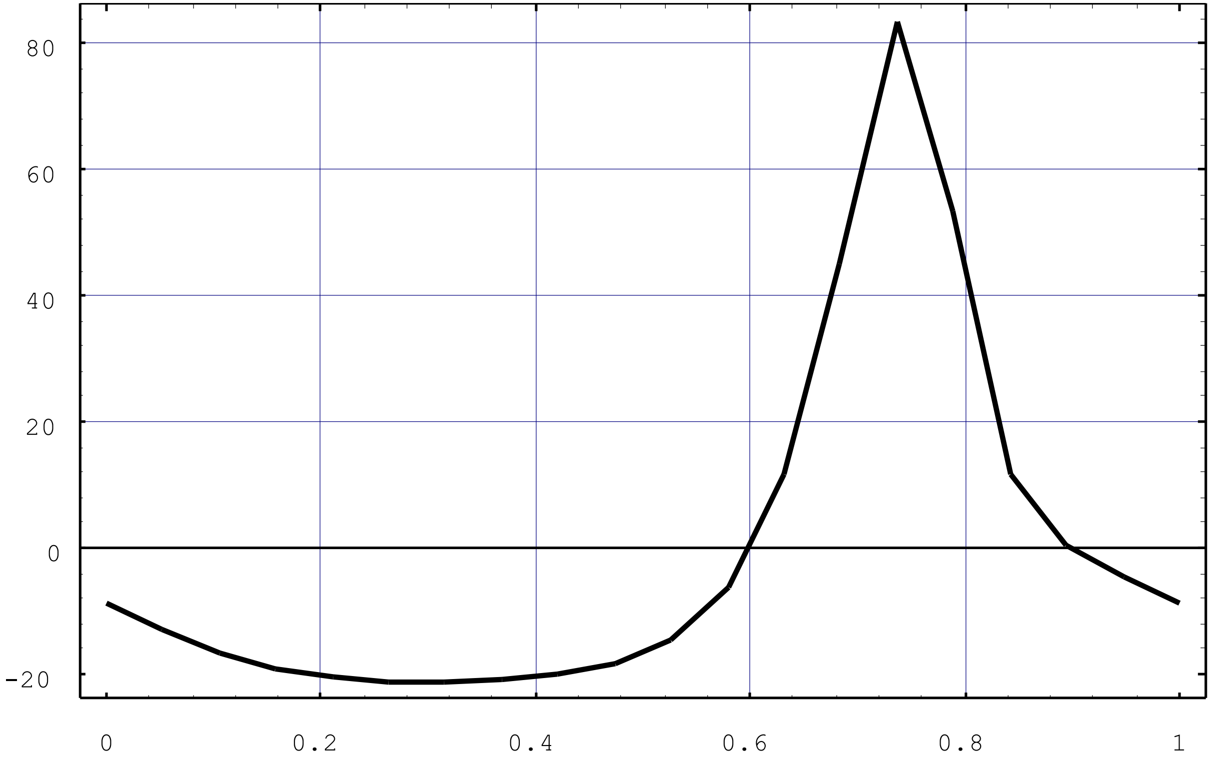
OBTENCION **VALORES MAXIMO Y MINIMO X4** (4)

```
SetGuess[]
{T -> 1., X2 -> 0, Y2 -> 5, Th2 -> 0, X3 -> 0,
 Y3 -> 0, Th3 -> -0.5,   X4 -> 6, Y4 -> 10, Th4 -> 0}
SolveMech[0.15, Solution->Velocity]
{T -> 0.15, X2 -> 0., Y2 -> 5., Th2 -> 0.942478,
 X3 -> 0., Y3 -> 0.,   Th3 -> -0.233107, X4 -> 6.92194,
 Y4 -> 10., Th4 -> 0., X2d -> 0.,   Y2d -> 0.,
Th2d -> 6.28319, X3d -> 0., Y3d -> 0., Th3d -> 2.27897,
  X4d -> -18.7395, Y4d -> 0., Th4d -> 0.}
velocidadT = SolveMech[posicionT, Solution->Velocity];
```

ANALISIS CINEMATICO MECANISMOS

OBTENCION **VALORES MAXIMO Y MINIMO X4** (5)

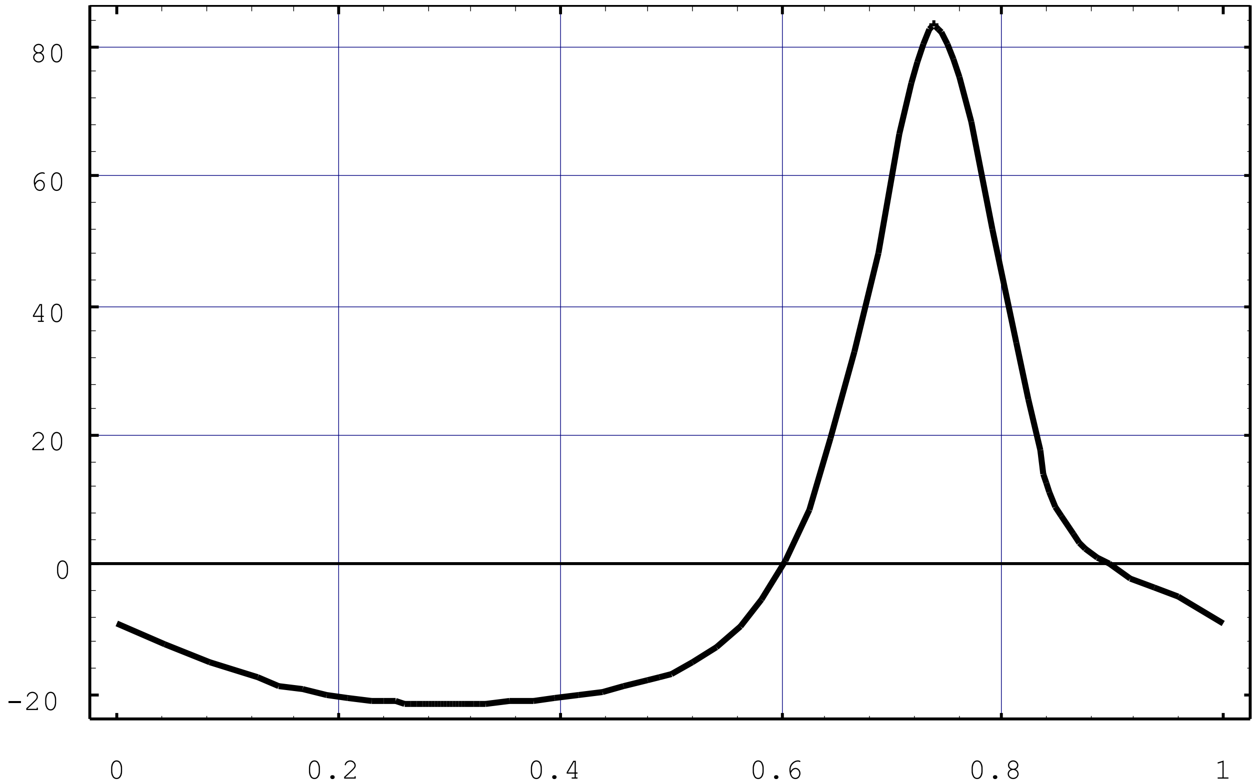
```
ListPlot[{T, X4d}/.velocidadT,
  PlotJoined->True, Frame->True, GridLines->Automatic];
```



ANALISIS CINEMATICO MECANISMOS

OBTENCION **VALORES MAXIMO Y MINIMO X4** (6)

```
X4dt = TimeInterpolate[X4d, velocidadT];
Plot[ X4dt[T], {T, 0, 1.}, Frame->True, GridLines->Automatic];
```



ANALISIS CINEMATICO MECANISMOS

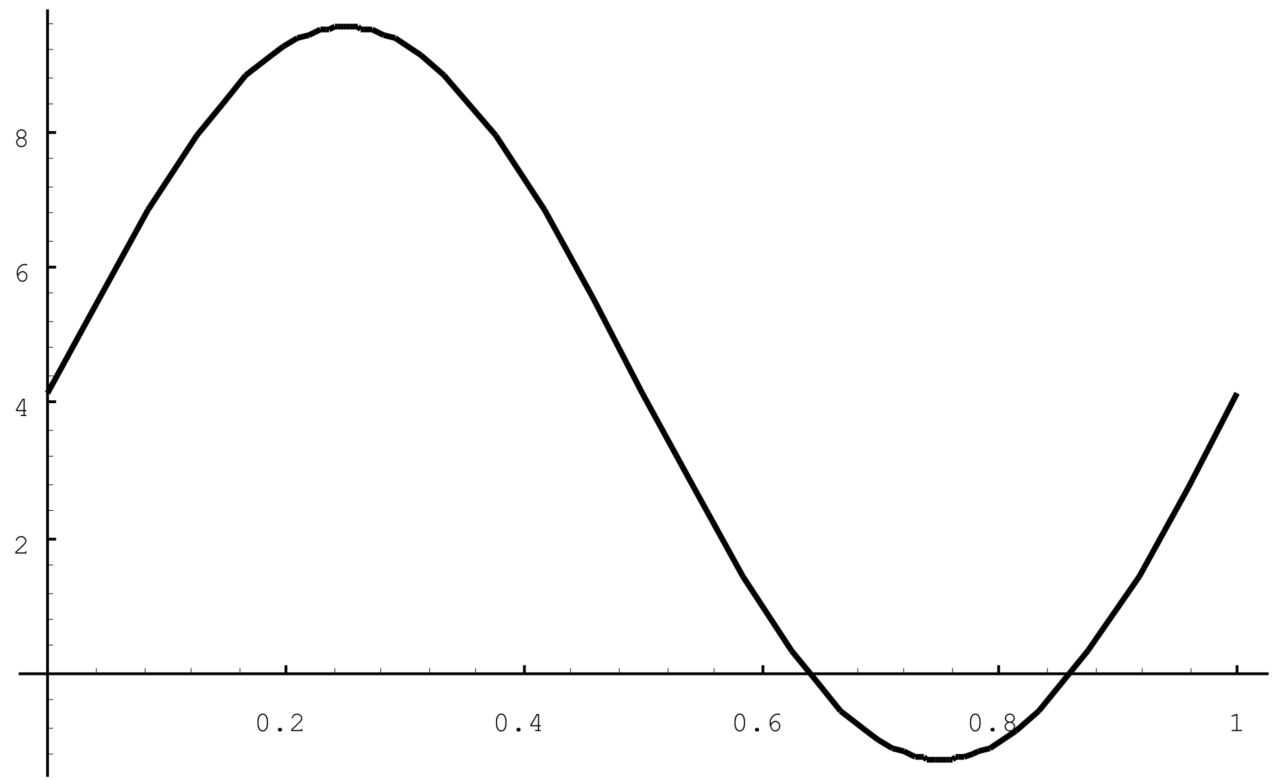
OBTENCION **VALORES MAXIMO Y MINIMO X4** (7)

```
FindRoot[X4dt[T],{T,0.7}]
{T -> 0.602307}
FindRoot[X4dt[T],{T,1}]
{T -> 0.897082}
X4t[0.602307]
-1.26516
X4t[0.897082]
9.54378
(-X4t[0.602307] + X4t[0.897082]) /2
5.40447
X4t[0.602307] + (X4t[0.897082] - X4t[0.602307])/2
4.13931
```

ANALISIS CINEMATICO MECANISMOS

OBTENCION **VALORES MAXIMO Y MINIMO X4** (8)

```
Plot[X4t[0.602307] + (X4t[0.897082] - X4t[0.602307])/2 +
        (X4t[0.897082] - X4t[0.602307])/2 Sin[2 N[Pi] T], {T,0,1}]
```

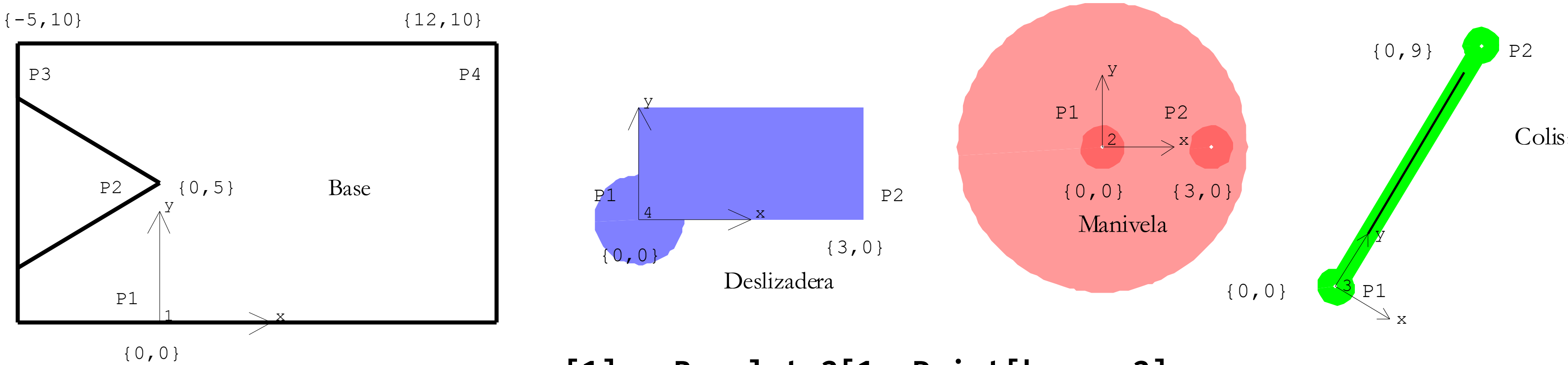


ANALISIS CINEMATICO MECANISMOS

OBTENCION **VALORES MAXIMO Y MINIMO X4** (9)

```
X4t[0.602307] + (X4t[0.897082] - X4t[0.602307])/2 +
             (X4t[0.897082] - X4t[0.602307])/2 Sin[2 N[Pi] 0.0]
4.13931
X4t[0.602307] + (X4t[0.897082] - X4t[0.602307])/2 +
             (X4t[0.897082] - X4t[0.602307])/2 Sin[2 N[Pi] 0.25]
9.54378
X4t[0.602307] + (X4t[0.897082] - X4t[0.602307])/2 +
             (X4t[0.897082] - X4t[0.602307])/2 Sin[2 N[Pi] 0.5]
4.13931
X4t[0.602307] + (X4t[0.897082] - X4t[0.602307])/2 +
             (X4t[0.897082] - X4t[0.602307])/2 Sin[2 N[Pi] 0.75]
-1.26516
X4t[0.602307] + (X4t[0.897082] - X4t[0.602307])/2 +
             (X4t[0.897082] - X4t[0.602307])/2 Sin[2 N[Pi] 1.0]
4.13931

(-X4t[0.602307] + X4t[0.897082]) /2
5.40447
X4t[0.602307] + (X4t[0.897082] - X4t[0.602307])/2
4.13931
```

ANALISIS CINEMATICO MECANISMOS
(2B) RESTRICCIONES - **IMPULSOR DESLIZADERA**



DEFINICION DE LAS RESTRICCIONES :

```
cs[1] = Revolute2[1, Point[base, 2],
                     Point[manivela,  1] ];
cs[2] = Revolute2[2, Point[base, 1],
                     Point[colisa, 1] ];
cs[3] = PointOnLine1[3, Point[manivela, 2],
                        Line[colisa, 1, 2] ];
cs[4] = RelativeDistance1[4, Point[colisa, 2],
                         Point[deslizadera, 1], 5 ];
cs[5] = Translate2[5, Line[base, 3, 4],
                      Line[deslizadera, 1, 2] ];
cs[6] = RelativeX1[6,Point[deslizadera,1],
                  4.13931 + 5.40447 Sin[2 N[Pi] T]];
```

```
SetBodies[bd[base], bd[manivela], bd[colisa], bd[deslizadera]]
```

ANALISIS CINEMATICO MECANISMOS

(2B) ANALISIS POSICION - ECUACIONES RESTRICCION

```
MatrixForm[Constraints[All]]
-X2 = 0
5 - Y2 = 0
-X3 = 0
-Y3 = 0
9 (-X2 + X3 - 3 Cos[Th2]) Cos[Th3] - 9 (Y2 - Y3 + 3 Sin[Th2]) Sin[Th3] = 0
                            2                          2
-25 + (Y3 - Y4 + 9 Cos[Th3])  + (X3 - X4 - 9 Sin[Th3]) = 0
34 Sin[Th4] = 0
2 (10 - Y4) Cos[Th4] + 2 (5 + X4) Sin[Th4] = 0
-4.13931 + X4 - 5.40447 Sin[6.28319 T] = 0
```

```
longitud = 3;
CheckSystem[]
True
SetGuess[ X2 -> 0., Y2 -> 5., Th2 -> 1.88496, Th3 -> 0.117504,
          X4 -> 3.83079, Y4 -> 10.]
{T -> 0, X2 -> 0., Y2 -> 5., Th2 -> 1.88496, X3 -> 0, Y3 -> 0,
   Th3 -> 0.117504, X4 -> 3.83079, Y4 -> 10., Th4 -> 0}
SolveMech[.0]
{T -> 0., X2 -> 0., Y2 -> 5., Th2 -> 1.7945, X3 -> 0., Y3 -> 0.,
   Th3 -> 0.083778, X4 -> 4.13931, Y4 -> 10., Th4 -> 0.}
```

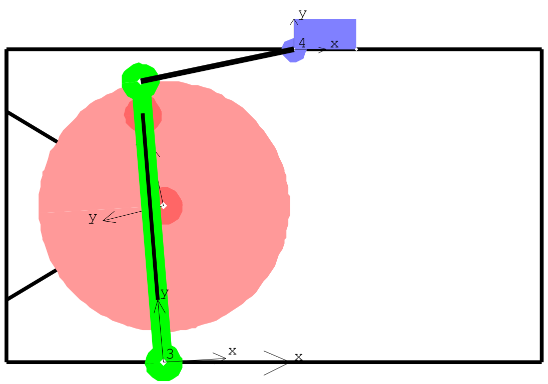
ANALISIS CINEMATICO MECANISMOS

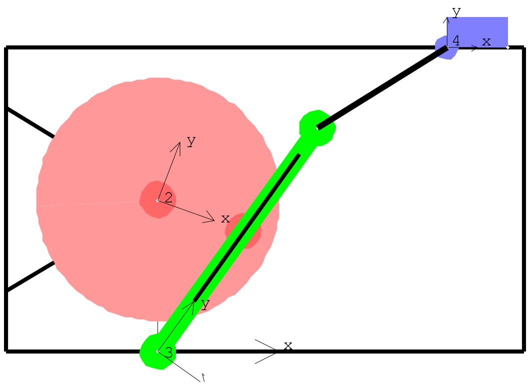
(3B) ANALISIS POSICION - RESULTADOS

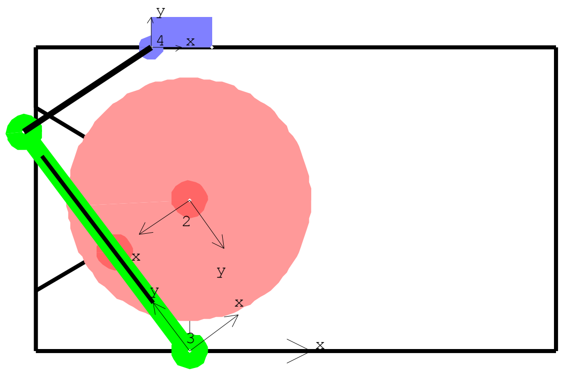
```
Show[{ baseG,
       manivelaG,
       colisaG,
       deslizaderaG,
       conexionG} /. SolveMech[.0],
     PlotRange->{{-6,15},{-1,12}},
     AspectRatio->Automatic];
```

```
Show[{ baseG,
       manivelaG,
       colisaG,
       deslizaderaG,
       conexionG} /. SolveMech[.22],
     PlotRange->{{-6,15},{-1,12}},
     AspectRatio->Automatic];
```

```
Show[{ baseG,
       manivelaG,
       colisaG,
       deslizaderaG,
       conexionG} /. SolveMech[.74],
     PlotRange->{{-6,15},{-1,12}},
     AspectRatio->Automatic];
```

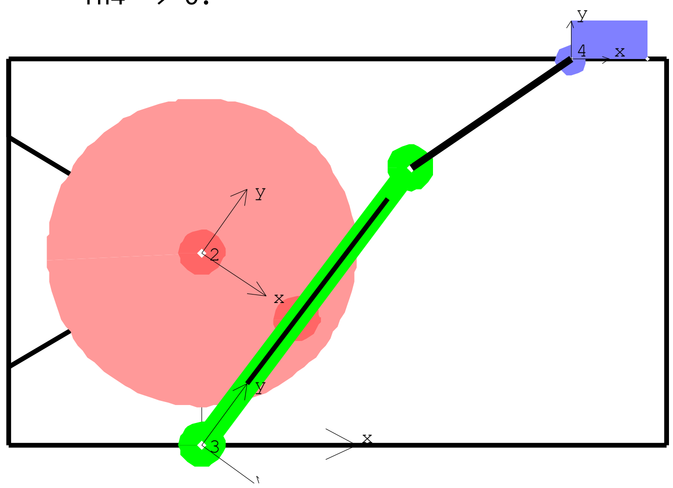






ANALISIS CINEMATICO MECANISMOS
(3B) ANALISIS POSICION - PUNTOS BLOQUEO

```
MatrixForm[SolveMech[.245]]
T -> 0.245
X2 -> 0.
Y2 -> 5.
Th2 -> -0.611293
X3 -> 0.
Y3 -> 0.
Th3 -> -0.643121
X4 -> 9.54111
Y4 -> 10.
Th4 -> 0.
```




```
(Angle[Vector[manivela,2]]/. LastSolve[])/N[Degree]
-35.0245
(N[Angle[Vector[colisa,2]]/.LastSolve[]])/N[Degree]
53.1519
- (Angle[Vector[manivela,2]]/. LastSolve[])/N[Degree] +
   (N[Angle[Vector[colisa,2]]/.LastSolve[]])/N[Degree]
88.1764
```

```
MatrixForm[SolveMech[.249]]
Mech::noconverge: Newton's method
failed to converge in 15 iterations.
T -> 0.249
X2 -> 0.
Y2 -> 5.
Th2 -> -1.17501
X3 -> 0.
Y3 -> 0.
Th3 -> -0.643842
X4 -> 9.54367
Y4 -> 10.
Th4 -> 0.
```