```
> restart:
  with(LinearAlgebra):
```

Università degli studi di Roma La Sapienza
Meccanica applicata alle macchine per
allievi iscritti al Corso di Laurea in Ingegneria Meccanica
A.A. 2008 - 2009
Formule di riferimento per l'esercitazione
DEDUZIONE DELLE EQUAZIONI DI VINCOLO
NELLE COORDINATE NATURALI

Si ha [q] := [xB,yB,xC,yC]T che si partiziona nei due vettori
[y] := [yB,xC,yC] delle variabili dipendenti e
[x] :=[xB] delle variabili indipendenti (corrispondono ai moventi)

In questo esempio si ha:
n = 4 coordinate lagrangiane
m = 3 coordinate di vincolo dedotte dalla definizione delle 4 cerniere
F = n - m = 1 gradi di libertà, ovvero dimensioni del vettore [x] delle coordinate lagrangiane moventi.

DFINIZIONE DELLE COORDINATE NATURALI

la manovella 2 è definita dalle coordinate x2 ed y2 del baricentro e l'angolo di rotazione della manovella stessa (assoluto rispetto all'asse x)
la posizione della manovella è rappresentata dallo spostamento definito da un piano soliodale con il telaio ad un piano sovrapposto al riferimento solidale con la manovella.
Lo stesso vale per le posizioni della biella e dell'asta cedente.

DEFINIZIONE DEI VINCOLI nelle coordinate naturali

Definizione delle equazioni di vincolo sulle 4 cerniere

CERNIERA A, COMUNE AI CORPI 1 E 2

```
> psi := Vector(3):
```

```
psi[1] := xB^2 + yB^2 -l[2]^2;
psi[2] := (xC-xB)^2 + (yC-yB)^2 -l[3]^2;
psi[3] := (xC-l[1])^2 + yC^2 -l[4]^2;
```

$$\psi_1 := xB^2 + yB^2 - l_2^2$$

$$\psi_2 := (xC - xB)^2 + (yC - yB)^2 - l_3^2$$

$$\psi_3 := (xC - l_1)^2 + yC^2 - l_4^2$$

Vettore dei vincoli Psi(q) = 0

```
> psi;
```

$$\begin{bmatrix} xB^2 + yB^2 - l_2^2 \\ (xC - xB)^2 + (yC - yB)^2 - l_3^2 \\ (xC - l_1)^2 + yC^2 - l_4^2 \end{bmatrix}$$

le coordinate lagrangiane sono in questo caso le 9 coordinate assolute (tre angoli e sei coordinate dei baricentri delle aste
essendo 8 le equazioni di vincolo abbiamo un solo parametro libero (corrispondente al grado di libertà del sistema

INCOGNITE E SET di EQUAZIONI

```
> set_lagrangiane := {yB,xC,yC,xB};
set_incognite := {yB,xC,yC};
set_espressioni := {}:
for i from 1 to 3 do set_espressioni := set_espressioni union
{psi[i]} od:
set_espressioni;
qq := Vector([ yB,xC,yC,xB ]);
uu := SubVector(qq,[1..3]);
vv := SubVector(qq,[4]);
```

$$set\_lagrangiane := \{xB, xC, yB, yC\}$$

$$set\_incognite := \{xC, yB, yC\}$$

$$\left\{xB^2 + yB^2 - l_2^2,\ (xC - xB)^2 + (yC - yB)^2 - l_3^2,\ (xC - l_1)^2 + yC^2 - l_4^2\right\}$$

$$qq := \begin{bmatrix} yB \\ xC \\ yC \\ xB \end{bmatrix}$$

$$uu := \begin{bmatrix} yB \\ xC \\ yC \end{bmatrix}$$

$$vv := \begin{bmatrix} xB \end{bmatrix}$$

Espressione dello jacobiano del vettore delle equazioni di vincolo e suoi minori di interesse

```
> psi_q := Matrix(3,4,linalg[ jacobian ](psi,[yB,xC,yC,xB]));
  psi_u := Matrix(3,3,SubMatrix(psi_q,1..3,1..3));
  psi_v := Matrix(3,1,SubMatrix(psi_q,1..3,4..4));
```

$$psi\_q := \begin{bmatrix} 2\,yB & 0 & 0 & 2\,xB \\ -2\,yC + 2\,yB & 2\,xC - 2\,xB & 2\,yC - 2\,yB & -2\,xC + 2\,xB \\ 0 & 2\,xC - 2\,l_1 & 2\,yC & 0 \end{bmatrix}$$

$$psi\_u := \begin{bmatrix} 2\,yB & 0 & 0 \\ -2\,yC + 2\,yB & 2\,xC - 2\,xB & 2\,yC - 2\,yB \\ 0 & 2\,xC - 2\,l_1 & 2\,yC \end{bmatrix}$$

$$psi\_v := \begin{bmatrix} 2\,xB \\ -2\,xC + 2\,xB \\ 0 \end{bmatrix}$$

```
> d_psi_u := subs(yB=yB(t),xC=xC(t),yC=yC(t),xB=xB(t),psi_u);
  d_psi_v := subs(yB=yB(t),xC=xC(t),yC=yC(t),xB=xB(t),psi_v);
```

$$d\_psi\_u := \begin{bmatrix} 2\,yB(t) & 0 & 0 \\ -2\,yC(t) + 2\,yB(t) & 2\,xC(t) - 2\,xB(t) & 2\,yC(t) - 2\,yB(t) \\ 0 & 2\,xC(t) - 2\,l_1 & 2\,yC(t) \end{bmatrix}$$

$$d\_psi\_v := \begin{bmatrix} 2\,xB(t) \\ -2\,xC(t) + 2\,xB(t) \\ 0 \end{bmatrix}$$

```
> psi_u_pu := Matrix(3,3):
  for ii from 1 to 3 do for jj from 1 to 3 do psi_u_pu[ii,jj]:=
  diff(d_psi_u[ii,jj],t) od od:
  psi_u_pu;
  psi_v_pu := Matrix(3,1):
  for ii from 1 to 3 do for jj from 1 to 1 do psi_v_pu[ii,jj]:=
  diff(d_psi_v[ii,jj],t) od od:
  psi_v_pu;
```

$$\left[\left[2\left(\frac{d}{dt}yB(t)\right),0,0\right],\left[-2\left(\frac{d}{dt}yC(t)\right)+2\left(\frac{d}{dt}yB(t)\right),2\left(\frac{d}{dt}xC(t)\right)-2\left(\frac{d}{dt}xB(t)\right),2\left(\frac{d}{dt}yC(t)\right)-2\left(\frac{d}{dt}yB(t)\right)\right],\left[0,2\left(\frac{d}{dt}xC(t)\right),2\left(\frac{d}{dt}yC(t)\right)\right]\right]$$

$$\left[\begin{array}{c}2\left(\frac{d}{dt}xB(t)\right)\\-2\left(\frac{d}{dt}xC(t)\right)+2\left(\frac{d}{dt}xB(t)\right)\\0\end{array}\right]$$

Determinante del minore dello jacobiano psi_y

```
> Det_psi_u := simplify(Determinant(d_psi_u));
```

$$Det\_psi\_u := -8\,yB(t)\left(yC(t)\,xB(t)-yC(t)\,l_1-yB(t)\,xC(t)+yB(t)\,l_1\right)$$

Matrice inversa del minore psi_y

```
> Inv_psi_u := MatrixInverse(d_psi_u);
```

$$Inv\_psi\_u := \left[\left[\frac{1}{2\,yB(t)},0,0\right],\left[-\frac{1}{2}\frac{(-yC(t)+yB(t))\,yC(t)}{yB(t)\left(-yC(t)\,xB(t)+yC(t)\,l_1+yB(t)\,xC(t)-yB(t)\,l_1\right)},\frac{1}{2}\frac{yC(t)}{-yC(t)\,xB(t)+yC(t)\,l_1+yB(t)\,xC(t)-yB(t)\,l_1},\frac{1}{2}\frac{-yC(t)+yB(t)}{-yC(t)\,xB(t)+yC(t)\,l_1+yB(t)\,xC(t)-yB(t)\,l_1}\right],\right.$$

$$\left[\frac{1}{2}\,\frac{(-yC(t)+yB(t))\,(xC(t)-l_1)}{yB(t)\,(-yC(t)\,xB(t)+yC(t)\,l_1+yB(t)\,xC(t)-yB(t)\,l_1)},\right.$$

$$-\frac{1}{2}\,\frac{xC(t)-l_1}{-yC(t)\,xB(t)+yC(t)\,l_1+yB(t)\,xC(t)-yB(t)\,l_1},$$

$$\left.\frac{1}{2}\,\frac{xC(t)-xB(t)}{-yC(t)\,xB(t)+yC(t)\,l_1+yB(t)\,xC(t)-yB(t)\,l_1}\right]\Bigg]$$

VALORI NOTI - semplificazione delle espressioni

```
> v_punto := Vector([VBx]);
```

$$v\_punto := \left[\; VBx \;\right]$$

Matrice inversa

```
> In_psi_u := Matrix(3,3):
  for ii from 1 to 3 do for jj from 1 to 3 do
  In_psi_u[ii,jj] := subs(yB(t)=yB,yC(t)=yC,xB(t)=xB,xC(t)=xC,
  Inv_psi_u[ii,jj]):
  od: od:
  In_psi_u;
```

$$\left[\left[\frac{1}{2\,yB},0,0\right],\right.$$

$$\left[-\frac{1}{2}\,\frac{(-yC+yB)\,yC}{yB\,(-yC\,xB+yC\,l_1+yB\,xC-yB\,l_1)},\frac{1}{2}\,\frac{yC}{-yC\,xB+yC\,l_1+yB\,xC-yB\,l_1},\right.$$

$$\left.\frac{1}{2}\,\frac{-yC+yB}{-yC\,xB+yC\,l_1+yB\,xC-yB\,l_1}\right],$$

$$\left[\frac{1}{2}\,\frac{(-yC+yB)\,(xC-l_1)}{yB\,(-yC\,xB+yC\,l_1+yB\,xC-yB\,l_1)},-\frac{1}{2}\,\frac{xC-l_1}{-yC\,xB+yC\,l_1+yB\,xC-yB\,l_1},\right.$$

$$\left.\left.\frac{1}{2}\,\frac{xC-xB}{-yC\,xB+yC\,l_1+yB\,xC-yB\,l_1}\right]\right]$$

Matrici psi_u e psi_v

```
> psi_u;
```

```
psi_v;
```

$$\begin{bmatrix} 2\,yB & 0 & 0 \\ -2\,yC + 2\,yB & 2\,xC - 2\,xB & 2\,yC - 2\,yB \\ 0 & 2\,xC - 2\,l_1 & 2\,yC \end{bmatrix}$$

$$\begin{bmatrix} 2\,xB \\ -2\,xC + 2\,xB \\ 0 \end{bmatrix}$$

matrici psi_u_punto e psi_v_punto

```
> psi_u_punto := Matrix(3,3):
for ii from 1 to 3 do for jj from 1 to 3 do
psi_u_punto[ii,jj] := subs(diff(yB(t),t)=VBy,diff(yC(t),t)=VCy,
diff(xB(t),t)=VBx,diff(xC(t),t)=VCx,psi_u_pu[ii,jj]):
od: od:
psi_u_punto;
psi_v_punto := Matrix(3,1):
for ii from 1 to 3 do
psi_v_punto[ii,1] := subs(diff(yB(t),t)=VBy,diff(yC(t),t)=VCy,
diff(xB(t),t)=VBx,diff(xC(t),t)=VCx,psi_v_pu[ii,1]):
od:
psi_v_punto;
```

$$\begin{bmatrix} 2\,VBy & 0 & 0 \\ -2\,VCy + 2\,VBy & 2\,VCx - 2\,VBx & 2\,VCy - 2\,VBy \\ 0 & 2\,VCx & 2\,VCy \end{bmatrix}$$

$$\begin{bmatrix} 2\,VBx \\ -2\,VCx + 2\,VBx \\ 0 \end{bmatrix}$$

CALCOLO DELLE VELOCITA'

```
> Rap := MatrixMatrixMultiply(In_psi_u,psi_v);
  for jj from 1 to 3 do Rap[jj,1] := simplify(Rap[jj,1]) od:
  Rap;
  u_punto := - MatrixVectorMultiply(Rap,v_punto):
  for jj from 1 to 3 do u_punto[jj] := simplify(u_punto[jj]) od:
  u_punto;
```

$$Rap := \begin{bmatrix} \frac{xB}{yB} \\ -\frac{(-yC+yB)\,yC\,xB}{yB\,(-yC\,xB+yC\,l_1+yB\,xC-yB\,l_1)} + \frac{1}{2}\,\frac{yC\,(-2\,xC+2\,xB)}{-yC\,xB+yC\,l_1+yB\,xC-yB\,l_1} \\ \frac{(-yC+yB)\,(xC-l_1)\,xB}{yB\,(-yC\,xB+yC\,l_1+yB\,xC-yB\,l_1)} - \frac{1}{2}\,\frac{(xC-l_1)\,(-2\,xC+2\,xB)}{-yC\,xB+yC\,l_1+yB\,xC-yB\,l_1} \end{bmatrix}$$

$$\begin{bmatrix} \frac{xB}{yB} \\ -\frac{yC\,(yC\,xB-yB\,xC)}{yB\,(yC\,xB-yC\,l_1-yB\,xC+yB\,l_1)} \\ \frac{(xC-l_1)\,(yC\,xB-yB\,xC)}{yB\,(yC\,xB-yC\,l_1-yB\,xC+yB\,l_1)} \end{bmatrix}$$

$$\begin{bmatrix} -\frac{xB\,VBx}{yB} \\ \frac{yC\,(yC\,xB-yB\,xC)\,VBx}{yB\,(yC\,xB-yC\,l_1-yB\,xC+yB\,l_1)} \\ -\frac{(xC-l_1)\,(yC\,xB-yB\,xC)\,VBx}{yB\,(yC\,xB-yC\,l_1-yB\,xC+yB\,l_1)} \end{bmatrix}$$

CALCOLO DELLE ACCELERAZIONI

```
> v_2_pu := Vector([ABx]);
  H1 := MatrixVectorMultiply(psi_v,v_2_pu);
  H2 := MatrixVectorMultiply(psi_u_punto,u_punto);
  H3 := MatrixVectorMultiply(psi_v_punto,v_punto);
  H0 := H1 + H2 + H3:
  u_2_pu := - MatrixVectorMultiply(In_psi_u,H0);
```

$$v\_2\_pu := \left[\begin{array}{c} ABx \end{array}\right]$$

$$H1 := \left[\begin{array}{c} 2\,xB\,ABx \\ (-2\,xC + 2\,xB)\,ABx \\ 0 \end{array}\right]$$

$$H2 := \left[\left[-\frac{2\,VBy\,xB\,VBx}{yB}\right],\right.$$

$$\left[-\frac{(-2\,VCy + 2\,VBy)\,xB\,VBx}{yB} + \frac{(2\,VCx - 2\,VBx)\,yC\,(yC\,xB - yB\,xC)\,VBx}{yB\,(yC\,xB - yC\,l_1 - yB\,xC + yB\,l_1)}\right.$$

$$\left. - \frac{(2\,VCy - 2\,VBy)\,(xC - l_1)\,(yC\,xB - yB\,xC)\,VBx}{yB\,(yC\,xB - yC\,l_1 - yB\,xC + yB\,l_1)}\right],$$

$$\left.\left[\frac{2\,VCx\,yC\,(yC\,xB - yB\,xC)\,VBx}{yB\,(yC\,xB - yC\,l_1 - yB\,xC + yB\,l_1)} - \frac{2\,VCy\,(xC - l_1)\,(yC\,xB - yB\,xC)\,VBx}{yB\,(yC\,xB - yC\,l_1 - yB\,xC + yB\,l_1)}\right]\right]$$

$$H3 := \left[\begin{array}{c} 2\,VBx^2 \\ (-2\,VCx + 2\,VBx)\,VBx \\ 0 \end{array}\right]$$

$$u\_2\_pu := \left[\left[-\frac{1}{2}\,\frac{2\,xB\,ABx - \frac{2\,VBy\,xB\,VBx}{yB} + 2\,VBx^2}{yB}\right],\right.$$

$$\left[\frac{1}{2}\,\frac{(-yC + yB)\,yC\,\left(2\,xB\,ABx - \frac{2\,VBy\,xB\,VBx}{yB} + 2\,VBx^2\right)}{yB\,(-yC\,xB + yC\,l_1 + yB\,xC - yB\,l_1)}\right.$$

$$-\frac{1}{2}\,\frac{1}{-yC\,xB + yC\,l_1 + yB\,xC - yB\,l_1}\left(yC\left((-2\,xC + 2\,xB)\,ABx\right.\right.$$

$$-\frac{(-2\,VCy + 2\,VBy)\,xB\,VBx}{yB} + \frac{(2\,VCx - 2\,VBx)\,yC\,(yC\,xB - yB\,xC)\,VBx}{yB\,(yC\,xB - yC\,l_1 - yB\,xC + yB\,l_1)}$$

$$\left.\left.-\frac{(2\,VCy - 2\,VBy)\,(xC - l_1)\,(yC\,xB - yB\,xC)\,VBx}{yB\,(yC\,xB - yC\,l_1 - yB\,xC + yB\,l_1)} + (-2\,VCx + 2\,VBx)\,VBx\right)\right)$$

$$-\frac{1}{2}\,\frac{1}{-yC\,xB + yC\,l_1 + yB\,xC - yB\,l_1}\left((-yC\right.$$

$$+ yB) \left(\frac{2\, VCx\, yC\, (yC\, xB - yB\, xC)\, VBx}{yB\, (yC\, xB - yC\, l_1 - yB\, xC + yB\, l_1)} \right.$$

$$\left. \left. \left. - \frac{2\, VCy\, (xC - l_1)\, (yC\, xB - yB\, xC)\, VBx}{yB\, (yC\, xB - yC\, l_1 - yB\, xC + yB\, l_1)} \right) \right) \right],$$

$$\left[-\frac{1}{2}\, \frac{(-yC + yB)\, (xC - l_1) \left(2\, xB\, ABx - \frac{2\, VBy\, xB\, VBx}{yB} + 2\, VBx^2 \right)}{yB\, (-yC\, xB + yC\, l_1 + yB\, xC - yB\, l_1)} \right.$$

$$+ \frac{1}{2}\, \frac{1}{-yC\, xB + yC\, l_1 + yB\, xC - yB\, l_1} \left((xC - l_1) \left((-2\, xC + 2\, xB)\, ABx \right. \right.$$

$$- \frac{(-2\, VCy + 2\, VBy)\, xB\, VBx}{yB} + \frac{(2\, VCx - 2\, VBx)\, yC\, (yC\, xB - yB\, xC)\, VBx}{yB\, (yC\, xB - yC\, l_1 - yB\, xC + yB\, l_1)}$$

$$\left. \left. - \frac{(2\, VCy - 2\, VBy)\, (xC - l_1)\, (yC\, xB - yB\, xC)\, VBx}{yB\, (yC\, xB - yC\, l_1 - yB\, xC + yB\, l_1)} + (-2\, VCx + 2\, VBx)\, VBx \right) \right)$$

$$- \frac{1}{2}\, \frac{1}{-yC\, xB + yC\, l_1 + yB\, xC - yB\, l_1} \left((xC \right.$$

$$- xB) \left(\frac{2\, VCx\, yC\, (yC\, xB - yB\, xC)\, VBx}{yB\, (yC\, xB - yC\, l_1 - yB\, xC + yB\, l_1)} \right.$$

$$\left. \left. \left. \left. - \frac{2\, VCy\, (xC - l_1)\, (yC\, xB - yB\, xC)\, VBx}{yB\, (yC\, xB - yC\, l_1 - yB\, xC + yB\, l_1)} \right) \right) \right] \right]$$

VALORI NUMERICI

```
> theta2 := evalf(20*(Pi/180));
  omega2 := evalf(400*(2*Pi/60));
  alpha2 := 0;
  l[1] := 80;
  l[2] := 20;
  l[3] := 50;
  l[4] := 70;
  xB := l[2]*cos(theta2);
  VB_mod := omega2 * l[2];
  AB_mod := omega2^2 * l[2];
```

```
xB_punto := - VB_mod * sin(theta2);
xB_2_punti := - AB_mod * cos(theta2);
VBx := xB_punto;
ABx := xB_2_punti;
```

$$\theta 2 := 0.3490658504$$

$$\omega 2 := 41.88790204$$

$$\alpha 2 := 0$$

$$l_1 := 80$$

$$l_2 := 20$$

$$l_3 := 50$$

$$l_4 := 70$$

$$xB := 18.79385242$$

$$VB\_mod := 837.7580408$$

$$AB\_mod := 35091.92674$$

$$xB\_punto := -286.5301252$$

$$xB\_2\_punti := -32975.62461$$

$$VBx := -286.5301252$$

$$ABx := -32975.62461$$

```
> yB_0 := 10;
  xC_0 := 40;
  yC_0 := 40;
  u_0 := Vector([yB_0,xC_0,yC_0]);
  u_1 := Vector([0,0,0]):
  u_2 := Vector([0,0,0]):
  u_3 := Vector([0,0,0]):
  u_4 := Vector([0,0,0]):
  ps_num := Vector([0,0,0]):
  I_psi_num := Matrix(3,3):
  I_psi_num := evalf(eval(Inv_psi_u,[yB(t) = yB_0,xC(t) = xC_0,yC
  (t) = yC_0])):
  ps_num[1] := subs(yB = yB_0,xC = xC_0,yC = yC_0,psi[1]):
  ps_num[2] := subs(yB = yB_0,xC = xC_0,yC = yC_0,psi[2]):
  ps_num[3] := subs(yB = yB_0,xC = xC_0,yC = yC_0,psi[3]):
  ps_num:
  u_1 := u_0 - MatrixVectorMultiply(I_psi_num,ps_num);
  I_psi_num := evalf(eval(Inv_psi_u,[yB(t) = u_1[1],xC(t) = u_1
  [2],yC(t) = u_1[3]])):
```

```
ps_num[1] := subs(yB = u_1[1],xC = u_1[2],yC = u_1[3],psi[1]):
ps_num[2] := subs(yB = u_1[1],xC = u_1[2],yC = u_1[3],psi[2]):
ps_num[3] := subs(yB = u_1[1],xC = u_1[2],yC = u_1[3],psi[3]):
ps_num:
u_2 := u_1 - MatrixVectorMultiply(I_psi_num,ps_num);
I_psi_num := evalf(eval(Inv_psi_u,[yB(t) = u_2[1],xC(t) = u_2
[2],yC(t) = u_2[3]])):
ps_num[1] := subs(yB = u_2[1],xC = u_2[2],yC = u_2[3],psi[1]):
ps_num[2] := subs(yB = u_2[1],xC = u_2[2],yC = u_2[3],psi[2]):
ps_num[3] := subs(yB = u_2[1],xC = u_2[2],yC = u_2[3],psi[3]):
ps_num:
u_3 := u_2 - MatrixVectorMultiply(I_psi_num,ps_num);
I_psi_num := evalf(eval(Inv_psi_u,[yB(t) = u_3[1],xC(t) = u_3
[2],yC(t) = u_3[3]])):
ps_num[1] := subs(yB = u_3[1],xC = u_3[2],yC = u_3[3],psi[1]):
ps_num[2] := subs(yB = u_3[1],xC = u_3[2],yC = u_3[3],psi[2]):
ps_num[3] := subs(yB = u_3[1],xC = u_3[2],yC = u_3[3],psi[3]):
ps_num:
u_4 := u_3 - MatrixVectorMultiply(I_psi_num,ps_num);
```

$$yB\_0 := 10$$

$$xC\_0 := 40$$

$$yC\_0 := 40$$

$$u\_0 := \begin{bmatrix} 10 \\ 40 \\ 40 \end{bmatrix}$$

$$u\_1 := \begin{bmatrix} 7.33955555999999998 \\ 37.2236989649999970 \\ 58.4736989600000002 \end{bmatrix}$$

$$u\_2 := \begin{bmatrix} 6.85737618529999970 \\ 35.4462869929999940 \\ 54.1893144510000013 \end{bmatrix}$$

$$u\_3 := \begin{bmatrix} 6.84042386026999960 \\ 35.4424339330999914 \\ 53.9876281916000026 \end{bmatrix}$$

$$u\_4 := \begin{bmatrix} 6.84040285426175920 \\ 35.4424286544461893 \\ 53.9872469708526026 \end{bmatrix}$$

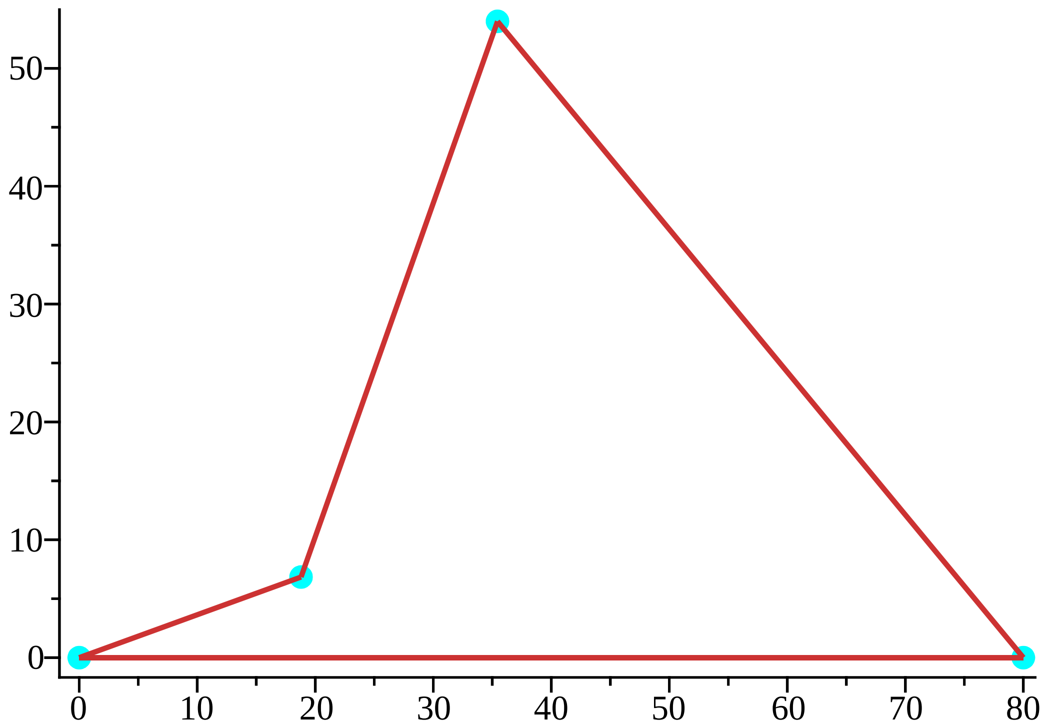
GRAFICO DEL QUADRILATERO

```
> with(geometry):
  l_co := 0.5;
  rag_cer := 1;
  quadri := {};

  point(A1P,0,0):
  point(D1P,evalf(l[1]),0):
  point(B1P,xB,u_4[1]):
  point(C1P,u_4[2],u_4[3]):

  circle(A1PC,[A1P,rag_cer]):
  circle(B1PC,[B1P,rag_cer]):
  circle(C1PC,[C1P,rag_cer]):
  circle(D1PC,[D1P,rag_cer]):

  segment(corp1,[A1P,D1P]):
  segment(corp2,[A1P,B1P]):
  segment(corp3,[B1P,C1P]):
  segment(corp4,[C1P,D1P]):

  quadri := quadri union {A1PC(color='cyan',thickness='3',filled=
  'true'),corp1(color='orange',thickness='2')}:
  quadri := quadri union {B1PC(color='cyan',thickness='3',filled=
  'true'),corp2(color='orange',thickness='2')}:
  quadri := quadri union {C1PC(color='cyan',thickness='3',filled=
  'true'),corp3(color='orange',thickness='2')}:
  quadri := quadri union {D1PC(color='cyan',thickness='3',filled=
  'true'),corp4(color='orange',thickness='2')}:

  draw(quadri,axes='framed');
```

$$l\_co := 0.5$$

$rag\_cer := 1$

$quadri := \{\ \}$




```
> u_pu := Vector([0,0,0]):
  u_pu[1] := evalf(subs(yB=u_4[1],xC=u_4[2],yC=u_4[3],u_punto[1]))
  :
  u_pu[2] := evalf(subs(yB=u_4[1],xC=u_4[2],yC=u_4[3],u_punto[2]))
  :
  u_pu[3] := evalf(subs(yB=u_4[1],xC=u_4[2],yC=u_4[3],u_punto[3]))
  :
  u_pu;
```

$$\begin{bmatrix} 787.2350507 \\ 582.1638808 \\ 480.4803008 \end{bmatrix}$$

Individuazione del quadrilatero articolato (ovvero degli infiniti quadrilateri articolati)

Verifica della lunghezza delle aste ed assegnazione delle velocità

```
> distance(A1P,B1P);
  distance(B1P,C1P);
  distance(C1P,D1P);
  evalf(distance(D1P,A1P));
  midpoint(GG2,corp2);
  detail(C1P);
  detail(GG2);
  VBy := u_pu[1];
  VCx := u_pu[2];
  VCy := u_pu[3];
```

20.00000000

50.00000000

70.00000000

80.00000000

GG2

name of the object	*C1P*
form of the object	*point2d*
coordinates of the point	[35.4424286544461893, 53.9872469708526026]

name of the object	*GG2*
form of the object	*point2d*
coordinates of the point	[9.396926210, 3.420201427]

$VBy := 787.2350507$

$VCx := 582.1638808$

$VCy := 480.4803008$

```
> scala_V := 0.02;
  G2_pos_x := evalf(xB/2);
  G2_pos_y := evalf(u_4[1] / 2);
  ext_V2_x := G2_pos_x + scala_V * (VBx/2):
  ext_V2_y := G2_pos_y + scala_V * (u_pu[1]/2):
  point(G2P,G2_pos_x,G2_pos_y):
  point(EV2P,ext_V2_x,ext_V2_y):
  segment(V2,[G2P,EV2P]):
```

```
G4_pos_x := u_4[2] + (l[1]-u_4[2])/2;
G4_pos_y := u_4[3]/2;
ext_V4_x := G4_pos_x + scala_V * (u_pu[2]/2):
ext_V4_y := G4_pos_y + scala_V * (u_pu[3]/2):
point(G4P,G4_pos_x,G4_pos_y):
point(EV4P,ext_V4_x,ext_V4_y):
segment(V4,[G4P,EV4P]):
G3_pos_x := xB + (u_4[2]-xB)/2;
G3_pos_y := u_4[1] + (u_4[3]-u_4[1])/2;
VG3x := VBx  + (VCx - VBx)/2;
VG3y := u_pu[1] + (VCy - VBy)/2;
ext_V3_x := G3_pos_x + scala_V * VG3x:
ext_V3_y := G3_pos_y + scala_V * VG3y:
point(G3P,G3_pos_x,G3_pos_y):
point(EV3P,ext_V3_x,ext_V3_y):
segment(V3,[G3P,EV3P]):
quadri := quadri union {V2(color='green',thickness='2')}:
quadri := quadri union {V4(color='green',thickness='2')}:
quadri := quadri union {V3(color='green',thickness='2')}:
draw(quadri);
```

$$scala\_V := 0.02$$

$$G2\_pos\_x := 9.396926210$$

$$G2\_pos\_y := 3.420201427$$

$$G4\_pos\_x := 57.72121432$$

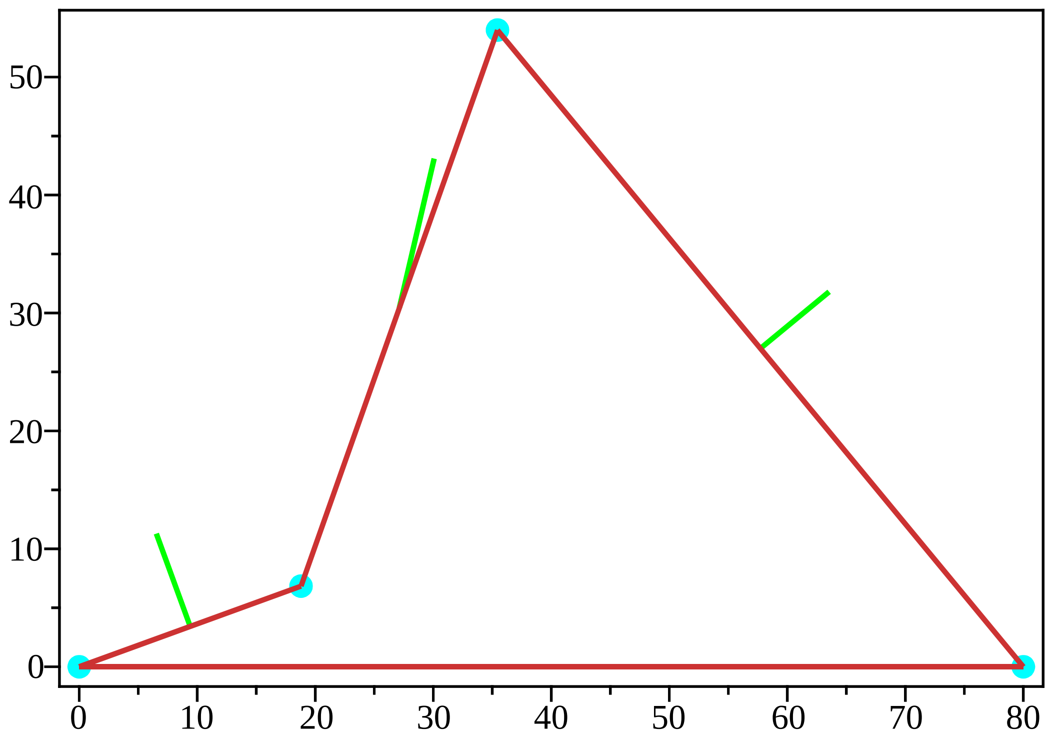
$$G4\_pos\_y := 26.99362348$$

$$G3\_pos\_x := 27.11814053$$

$$G3\_pos\_y := 30.41382491$$

$$VG3x := 147.8168778$$

$$VG3y := 633.8576756$$



Calcolo delle accelerazioni

```
> ABx;
  ABy := eval(u_2_pu[1],[yB=u_4[1] , xC=u_4[2], yC=u_4[3] ]);
  ACx := eval(u_2_pu[2],[yB=u_4[1] , xC=u_4[2], yC=u_4[3] ]);
  ACy := eval(u_2_pu[3],[yB=u_4[1] , xC=u_4[2], yC=u_4[3] ]);
```

$$-32975.62461$$

$$ABy := -12002.14618$$

$$ACx := -26385.73271$$

$$ACy := -32330.97387$$

```
> scala_A := 0.0005;
  ext_A2_x := G2_pos_x + scala_A * (ABx/2);
  ext_A2_y := G2_pos_y + scala_A * (ABy/2);
  point(EA2P,ext_A2_x,ext_A2_y);
  segment(AC2,[G2P,EA2P]);
  ext_A4_x := G4_pos_x + scala_A * (ACx/2);
  ext_A4_y := G4_pos_y + scala_A * (ACy/2);
  point(EA4P,ext_A4_x,ext_A4_y);
  segment(AC4,[G4P,EA4P]);
  ext_A3_x := G3_pos_x + scala_A * (ABx + (ACx - ABx)/2 );
  ext_A3_y := G3_pos_y + scala_A * (ABy + (ACy - ABy)/2);
  point(EA3P,ext_A3_x,ext_A3_y);
  segment(AC3,[G3P,EA3P]);
  quadri := quadri union {AC2(color='red',thickness='3')}:
  quadri := quadri union {AC4(color='red',thickness='3')}:
  quadri := quadri union {AC3(color='red',thickness='3')}:
  draw(quadri);
```

scala_A := 0.0005

ext_A2_x := 1.153020060

ext_A2_y := 0.419664882

EA2P

AC2

ext_A4_x := 51.12478114

ext_A4_y := 18.91088001

EA4P

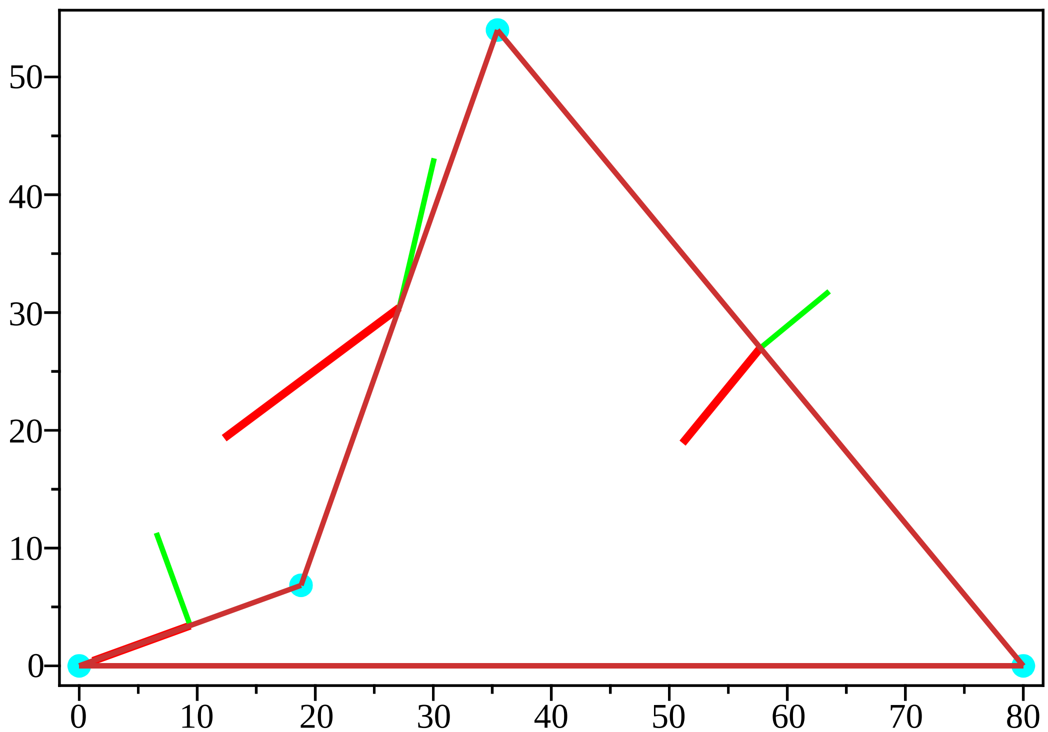
AC4

ext_A3_x := 12.27780120

ext_A3_y := 19.33054489

EA3P

AC3



50
40
30
20
10
0
0
10
20
30
40
50
60
70
80