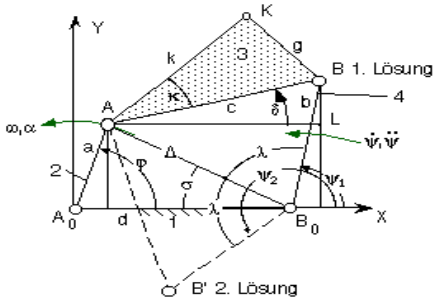
MDA-Chapter 3: Design of Mechanisms - Project WS2009-Problem1 - Solution

Munich Univ. of Applied Sciences - Prof. Dr. Oskar Wallrapp, FK06

created by Prof. O.Wallrapp - 11.12.2009
updated:

Crank-rocker mechanism: Basic functions




```
> restart;
  with(plots): with(plottools): with(StringTools):    # for plots
  and animation

> alias(Degree = Pi/180);
  1.0*Degree;
```

$$Degree$$

$$0.005555555556\,\pi$$

```
> X := d - a*cos(phi):
  Y := a * sin(phi):
  sigma := arctan(Y/X):
  Delta := sqrt(X^2 + Y^2):
  lambda := simplify(arccos((Delta^2 + b^2 - c^2) / (2 * b *
  Delta)) ):
> fpsi := Pi - (sigma + lambda);
```

$$fpsi := -\arctan\left(\frac{a\sin(\phi)}{d-a\cos(\phi)}\right)+\arccos\left(\frac{1}{2}\,\frac{-d^2+2\,d\,a\cos(\phi)-b^2+c^2-a^2}{b\sqrt{d^2-2\,d\,a\cos(\phi)+a^2}}\right)$$

```
> fpsi2 := Pi - (sigma - lambda);
```

$$fpsi2 := 2\,\pi-\arctan\left(\frac{a\sin(\phi)}{d-a\cos(\phi)}\right)-\arccos\left(\frac{1}{2}\,\frac{-d^2+2\,d\,a\cos(\phi)-b^2+c^2-a^2}{b\sqrt{d^2-2\,d\,a\cos(\phi)+a^2}}\right)$$

```
> fphii := Pi + arccos((d^2 + (c - a)^2 - b^2) / (2 * (c - a) *
  d)):
  fphia  := arccos((d^2 + (c + a)^2 - b^2) / (2 * (c + a) * d)):
  fphi0 := fphii - fphia;
```

$$fphi0 := \pi+\arccos\left(\frac{1}{2}\,\frac{d^2+(c-a)^2-b^2}{(c-a)\,d}\right)-\arccos\left(\frac{1}{2}\,\frac{d^2+(c+a)^2-b^2}{(c+a)\,d}\right)$$

```
> fpsii := Pi - arccos((d^2 + b^2 - (c - a)^2) / (2 * b * d)):
  fpsia := Pi - arccos((d^2 + b^2 - (c + a)^2) / (2 * b * d)):
  fpsi0 := fpsii - fpsia;
```

$$fpsi0 := -\arccos\left(\frac{1}{2}\,\frac{d^2+b^2-(c-a)^2}{b\,d}\right)+\arccos\left(\frac{1}{2}\,\frac{d^2+b^2-(c+a)^2}{b\,d}\right)$$

```
> mumin1:=arccos((c^2 + b^2 - (d - a)^2)/(2 * b * c));
```

$$mumin1 := \arccos\left(\frac{1}{2}\,\frac{c^2+b^2-(d-a)^2}{b\,c}\right)$$

```
> mumin2:= Pi-arccos((c^2 + b^2 - (d + a)^2)/(2 * b * c));
```

$$mumin2 := \pi-\arccos\left(\frac{1}{2}\,\frac{c^2+b^2-(d+a)^2}{b\,c}\right)$$

```
> mu := arccos((c^2+b^2-Delta_f^2)/(2*b*c)); fmu := subs(Delta_f
  = Delta, mu);
```

$$\mu := \arccos\left(\frac{1}{2}\,\frac{c^2+b^2-Delta\_f^2}{b\,c}\right)$$

$$fmu := \arccos\left(\frac{1}{2}\,\frac{c^2+b^2-(d-a\cos(\phi))^2-a^2\sin(\phi)^2}{b\,c}\right)$$

```
> Grashof:=min(a, b, c, d) + max(a, b, c, d);
```

$$Grashof := \min(a,b,c,d)+\max(a,b,c,d)$$

```
> fdelta := arctan((b * sin(psi) - a * sin(phi)), (d + b * cos
  (psi) - a * cos(phi)));
```

$$fdelta := \arctan(b\sin(\psi)-a\sin(\phi),\, d+b\cos(\psi)-a\cos(\phi))$$

```
> xK := A0x + a * cos(phi + gamma0) + k * cos(kappa + delta +
  gamma0);
  yK := A0y + a * sin(phi + gamma0) + k * sin(kappa + delta +
  gamma0);
```

$$xK := A0x+a\cos(\phi+\gamma0)+k\cos(\kappa+\delta+\gamma0)$$

$$yK := A0y+a\sin(\phi+\gamma0)+k\sin(\kappa+\delta+\gamma0)$$

1 Dead point construction via ALT

- 1. Find the length a, b, c for giben toggle parameters φ0, ψ0 and μmin -> max (VDI 2130) use Table 3-1

```
> parALT := [phi0 = 147.6 * Degree, psi0 = 11 * Degree, auxbeta =
  70 * Pi/180, d = 200];
```

$$parALT := \left[\phi0 = 0.8200000000\,\pi,\ \psi0 = \frac{11}{180}\,\pi,\ auxbeta = \frac{7}{18}\,\pi,\ d = 200\right]$$

```
> gamma0 := 1/2 * (phi0 - psi0):
> rA := (d * sin(psi0/2))/(2 * sin(gamma0))):
> rB := rA/cos(gamma0):
> theta := Pi - auxbeta - phi0/2:
> a := 2 * rA * cos(theta):
```

```
> c := 2 * rB * cos(theta - gamma0) - a:
> e := (a + c) * cos(auxbeta):
> b := sqrt(d^2 + (a+c)^2 - 2 * d * e):
> a_num := evalf(subs(parALT,a));
```

$$a\_num := 16.64858618$$

```
> b_num := evalf(subs(parALT,b));
  c_num := evalf(subs(parALT,c));
  parALT;
```

$$b\_num := 189.1233007$$

$$c\_num := 30.61935964$$

$$\left[\phi 0 = 0.8200000000\,\pi,\ \psi 0 = \frac{11}{180}\,\pi,\ auxbeta = \frac{7}{18}\,\pi,\ d = 200\right]$$

• further evaluations using the results from ALT:

```
> unassign('a','b','c','d','gamma0');
  par_erg1 := [A0x=-200, A0y=0, gamma0=0, a = a_num, b=b_num, c=
  c_num, d=200, k=0, kappa=0];
```

$$par\_erg1 := [A0x = -200,\ A0y = 0,\ \gamma 0 = 0,\ a = 16.64858618,\ b = 189.1233007,\ c = 30.61935964,\ d = 200,\ k = 0,\ \kappa = 0]$$

```
> evalf(subs(par_erg1, fpsi0/Degree));   # check of toggle angles
  psi0 and phi0
```

$$11.00000029$$

```
> evalf(subs(par_erg1, fphi0/Degree));
```

$$147.6000001$$

```
> evalf(subs(par_erg1, mumin1/Degree));
```

$$74.53921473$$

```
> mumin_erg1 := evalf(subs(par_erg1,mumin2/Degree));   # minimal
  transmission angle
```

$$mumin\_erg1 := 27.94267067$$

```
> phia_erg1 := evalf(subs(par_erg1, fphia));          # angles of
  toggle point phi a and phi i
  evalf(%/Degree);
```

$$phia\_erg1 := 1.221730477$$

$$70.00000003$$

```
> phii_erg1 := evalf(subs(par_erg1, fphii));
  evalf(%/Degree);
```

$$phii\_erg1 := 3.797836455$$

$$217.6000001$$

```
> psia_erg1 := evalf(subs(par_erg1,phi=phia_erg1, fpsi)); #
  angles of toggle point psi a and psi i:
  evalf(%/Degree);
```

$$psia\_erg1 := 2.904518980$$

$$166.4166790$$

```
> psii_erg1 := evalf(subs(par_erg1,phi=phii_erg1, fpsii)); #
  angle of toggle point a:
  evalf(%/Degree);
```

$$psii\_erg1 := 3.096505202$$

$$177.4166793$$

```
> # angles of the first point of diagram
  phi1_erg1 := evalf(phia_erg1 - 147.6/41 * 21*Degree );
  evalf(%/Degree);
```

$$phi1\_erg1 := -0.097738438$$

$$-5.599999992$$

```
> psi1_erg1 := evalf(subs(par_erg1, phi=phi1_erg1, fpsi));
  evalf(%*180/Pi);
```

$$psi1\_erg1 := 2.988740367$$

$$171.2422091$$

```
> # Build-in angle epsilon:
  epsilon := evalf(psia_erg1/Degree + 7 + 90);
```

$$\varepsilon := 263.4166790$$

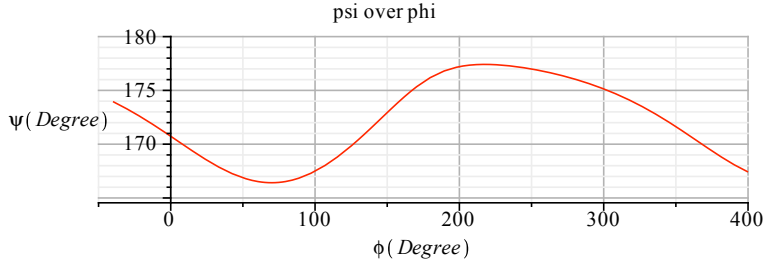
```
> alpha_Rq := epsilon*Degree-Pi/2-psi;
  evalf(subs(psi=fpsi,par_erg1, phi=phia_erg1,  alpha_Rq)/Degree)
  ;   # test for "a"
```

$$alpha\_Rq := 0.9634259940\,\pi - \psi$$

$$6.999999847$$

Test of alpha (for phi = phia) = 7 is correct:

```
> phi:= phiG*Degree:
  plo_psi1 := plot(subs(par_erg1, fpsi)/Degree, phiG=-40..400,
  gridlines=true,labels=['phi(Degree)','psi(Degree)'],title="psi
  over phi", view=[-50..400, 165..180]):
  phi := 'phi':
  display(plo_psi1);
```




```
> Grashof:= min(subs(par_erg1, a_num), subs(par_erg1, b_num),
  subs(par_erg1, c), subs(par_erg1,d)) + max(subs(par_erg1,
  a_num), subs(par_erg1, b_num), subs(par_erg1, c_num), subs
  (par_erg1, d));
```

$$Grashof := 216.6485862$$

```
> evalf(subs(par_erg1,c_num + b_num));
```

$$219.7426603$$

We have a good solution: full revolution of the crank
Transmission angle mu_min = 27.9 degrees.

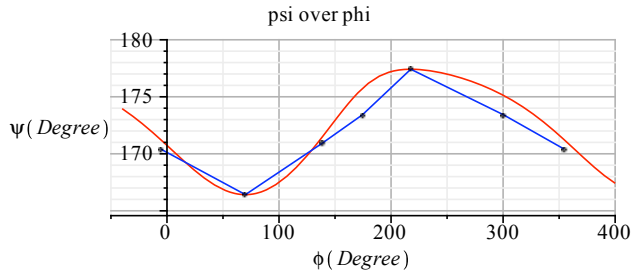
Animation of crank-rocker-mechanism with par1(psi=fpsi)

Check of the other points of psi(phi)

```
> val_alpha := [3, 7, 2.5, 0, -4, 0, 3 ];      # alpha in Degree
```
$$val\_alpha := [3, 7, 2.5, 0, -4, 0, 3]$$
```
> val_Dphi := [0, 0.21, 0.4, 0.5, 0.62, 0.85, 1]*360;  # Delta
  values in Degree
```
$$val\_Dphi := [0, 75.60, 144.0, 180.0, 223.20, 306.00, 360]$$

we write down the data about toggle point "a", => phia.
New unkowns are phia for the phi-axis, eps for the build-in angle epsilon.

```
> val_phi := evalm(evalf(evalm(evalm(val_Dphi - 0.21*360)*
  Pi/180)) + phia);  # shift to TP-a
```
$$val\_phi := [-1.319468915 + phia, phia, 1.193805209 + phia, 1.822123739 + phia, 2.576105976 + phia, 4.021238597 + phia, 4.963716393 + phia]$$
```
> # psi = eps - gamma0 - Pi/2 - alpha
  val_psi := evalf(evalm( evalf(subs(par_erg1, (eps - gamma0 -
  Pi/2))) - val_alpha*Degree ));
```
$$val\_psi := [-1.623156205 + eps, -1.692969375 + eps, -1.614429558 + eps, eps - 1.570796327, -1.500983157 + eps, eps - 1.570796327, -1.623156205 + eps]$$
```
> nfct := nops(val_Dphi);    # number of points
```
$$nfct := 7$$
```
> #test of phi with erg1 for point "a"
  evalf(evalm(subs(phia=phia_erg1, evalm(val_phi))/Degree));
  evalf(phi1_erg1/Degree);
```
$$[-5.599999992, 70.00000003, 138.4000000, 174.4000000, 217.6000000, 300.4000000, 354.4000001]$$
$$-5.599999992$$
```
> #test of psi with erg1 for point "a"
  val_psi[2]; evalf(subs(eps=epsilon*Degree,%));
  psia_erg1;
```
$$-1.692969375 + eps$$
$$2.904518977$$
$$2.904518980$$
```
> xx := Vector(nfct): yy := Vector(nfct):
  for i from 1 to nfct do
    xx[i] := evalf(subs(phia=phia_erg1     ,val_phi[i])/Degree)
  ;
    yy[i] := evalf(subs(eps=epsilon*Degree ,val_psi[i])/Degree)
  ;
  od:
  [xx, yy]:
> plo_Req := plot(xx,yy, color=blue):
  display(plo_psil, plo_Req,
    point([xx[1], yy[1]],lmax/50),
    point([xx[2], yy[2]],lmax/100),
    point([xx[3], yy[3]],lmax/100),
    point([xx[4], yy[4]],lmax/100),
    point([xx[5], yy[5]],lmax/100),
    point([xx[6], yy[6]],lmax/100),
    point([xx[7], yy[7]],lmax/100),
  gridlines=true);
```



There are big errors at the other points expect a and i.

2 Find a mechanism for a list of function values psi of phi

Given is
function fpsi(phi , par) , where par = a, b, c, d
parReq := {d=200};
Values of phi and psi with unknows phia and eps for 7 points

```
> par_erg1;
```
$$[A0x = -200, A0y = 0, \gamma 0 = 0, a = 16.64858618, b = 189.1233007, c = 30.61935964, d = 200, k = 0, \kappa = 0]$$
```
> fpsi;
```
$$-\arctan\left(\frac{a\sin(\phi)}{d - a\cos(\phi)}\right) + \arccos\left(\frac{1}{2}\,\frac{-d^2 + 2\,d\,a\cos(\phi) - b^2 + c^2 - a^2}{b\sqrt{d^2 - 2\,d\,a\cos(\phi) + a^2}}\right)$$
```
> parReq := {d=200};
```
$$parReq := \{d = 200\}$$
```
> evalm(val_phi);
```
$$[-1.319468915 + phia, phia, 1.193805209 + phia, 1.822123739 + phia, 2.576105976 + phia, 4.021238597 + phia, 4.963716393 + phia]$$
```
> evalm(val_psi);
```
$$[-1.623156205 + eps, -1.692969375 + eps, -1.614429558 + eps, eps - 1.570796327, -1.500983157 + eps, eps - 1.570796327, -1.623156205 + eps]$$

We setup j = 1..nfct requirements to satisfy psi(phi(j), a, b, c) = val_psi(j)(eps) , where nfct = 7

```
> Rq := [
      subs(phi=(val_phi[1]), parReq, fpsi),
      subs(phi=(val_phi[2]), parReq, fpsi),
      subs(phi=(val_phi[3]), parReq, fpsi),
```

```
    subs(phi=(val_phi[4]), parReq, fpsi),
    subs(phi=(val_phi[5]), parReq, fpsi),
    subs(phi=(val_phi[6]), parReq, fpsi),
    subs(phi=(val_phi[7]), parReq, fpsi)
];
```

$$Rq := \left[-\arctan\left(\frac{a\sin(-1.319468915+phia)}{200-a\cos(-1.319468915+phia)}\right) + \arccos\left(\frac{1}{2}\,\frac{-40000+400\,a\cos(-1.319468915+phia)-b^2+c^2-a^2}{b\sqrt{40000-400\,a\cos(-1.319468915+phia)+a^2}}\right),\right.$$
$$-\arctan\left(\frac{a\sin(phia)}{200-a\cos(phia)}\right) + \arccos\left(\frac{1}{2}\,\frac{-40000+400\,a\cos(phia)-b^2+c^2-a^2}{b\sqrt{40000-400\,a\cos(phia)+a^2}}\right),$$
$$-\arctan\left(\frac{a\sin(1.193805209+phia)}{200-a\cos(1.193805209+phia)}\right) + \arccos\left(\frac{1}{2}\,\frac{-40000+400\,a\cos(1.193805209+phia)-b^2+c^2-a^2}{b\sqrt{40000-400\,a\cos(1.193805209+phia)+a^2}}\right),$$
$$-\arctan\left(\frac{a\sin(1.822123739+phia)}{200-a\cos(1.822123739+phia)}\right) + \arccos\left(\frac{1}{2}\,\frac{-40000+400\,a\cos(1.822123739+phia)-b^2+c^2-a^2}{b\sqrt{40000-400\,a\cos(1.822123739+phia)+a^2}}\right),$$
$$-\arctan\left(\frac{a\sin(2.576105976+phia)}{200-a\cos(2.576105976+phia)}\right) + \arccos\left(\frac{1}{2}\,\frac{-40000+400\,a\cos(2.576105976+phia)-b^2+c^2-a^2}{b\sqrt{40000-400\,a\cos(2.576105976+phia)+a^2}}\right),$$
$$-\arctan\left(\frac{a\sin(4.021238597+phia)}{200-a\cos(4.021238597+phia)}\right) + \arccos\left(\frac{1}{2}\,\frac{-40000+400\,a\cos(4.021238597+phia)-b^2+c^2-a^2}{b\sqrt{40000-400\,a\cos(4.021238597+phia)+a^2}}\right),$$
$$\left.-\arctan\left(\frac{a\sin(4.963716393+phia)}{200-a\cos(4.963716393+phia)}\right) + \arccos\left(\frac{1}{2}\,\frac{-40000+400\,a\cos(4.963716393+phia)-b^2+c^2-a^2}{b\sqrt{40000-400\,a\cos(4.963716393+phia)+a^2}}\right)\right]$$

```
> evalf(subs(par_erg1, eps=epsilon*Degree, phia=phia_erg1, Rq[2]=
  val_psi[2])); # test
```

$$2.904518980 = 2.904518977$$

Find an exact solution for 5 points using fsolve

```
> erg2 := fsolve({Rq[1]=val_psi[1], Rq[2]=val_psi[2], Rq[3]=
  val_psi[3], Rq[4]=val_psi[4],Rq[6]=val_psi[6]},
  {phia=phia_erg1, eps=epsilon, a=16.6, b=189.1, c=30.6});
```

$$erg2 := \{a = 13.48849722,\ b = 198.4495821,\ c = 17.74308151,\ eps = 4.678488194,\ phia = 1.548128144\}$$

```
> evalf(subs(erg2, parReq, fphi0)/Degree);
```

$$165.3868552$$

```
> evalf(subs(erg2, parReq, fpsi0)/Degree);
```

$$7.840702706$$

```
> evalf(subs(erg2, parReq, fphia)/Degree);
```

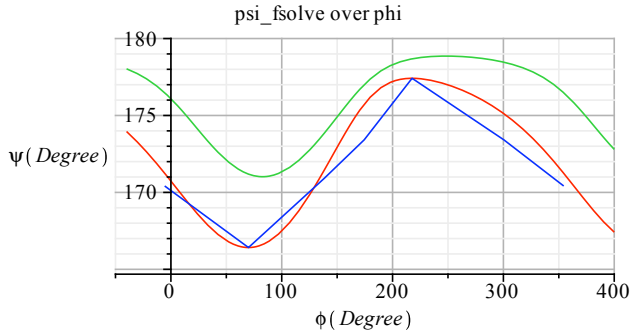
$$82.67316451$$

```
> evalf(subs(erg2, parReq, fpsia)/Degree);
```

$$171.0198145$$

```
> phi:= phiG*Pi/180:
  plo_psi2 := plot( { subs(parReq,erg2, fpsi)*180/Pi, subs
  (par_erg1, fpsi)*180/Pi},
  phiG=-40..400,gridlines=true,labels=['phi(Degree)','psi(Degree)
  '],title="psi_fsolve over phi", view=[-50..400, 165..180]);
  phi := 'phi':
  display(plo_psi2, plo_Req);
```

$$plo\_psi2 := PLOT(\ldots)$$



it's a bad solution: it satisfied the 5 points, but phi0 = 165.4 and psi0 = 7.8 degrees are bad values w. r.t. 147.6 and 11 degrees, respectively

Let's try an optimization!

```
> fct := [
  simplify( subs(phi=(val_phi[1]), parReq, fpsi) ),
  simplify( subs(phi=(val_phi[2]), parReq, fpsi) ),
  simplify( subs(phi=(val_phi[3]), parReq, fpsi) ),
```

```
simplify( subs(phi=(val_phi[4]), parReq, fpsi) ),
simplify( subs(phi=(val_phi[5]), parReq, fpsi) ),
simplify( subs(phi=(val_phi[6]), parReq, fpsi) ),
simplify( subs(phi=(val_phi[7]), parReq, fpsi) )
];
```

$$fct := \left[\arctan\left(\frac{a \sin(-1.319468915 + phia)}{-200. + a \cos(-1.319468915 + phia)} \right) \right.$$

$$+ \arccos\left(\left(0.5000000000 \left(-40000. + 400.\, a \cos(-1.319468915 + phia) - 1.\, b^2 + c^2 \right.\right.\right.$$

$$\left.\left.\left. - 1.\, a^2\right)\right) \Big/ \left(b \sqrt{40000. - 400.\, a \cos(-1.319468915 + phia) + a^2}\right)\right),$$

$$\arctan\left(\frac{a \sin(phia)}{-200 + a \cos(phia)} \right) + \arccos\left(\frac{1}{2} \frac{-40000 + 400\, a \cos(phia) - b^2 + c^2 - a^2}{b \sqrt{40000 - 400\, a \cos(phia) + a^2}} \right),$$

$$\arctan\left(\frac{a \sin(1.193805209 + phia)}{-200. + a \cos(1.193805209 + phia)} \right)$$

$$+ \arccos\left(\left(0.5000000000 \left(-40000. + 400.\, a \cos(1.193805209 + phia) - 1.\, b^2 + c^2 \right.\right.\right.$$

$$\left.\left.\left. - 1.\, a^2\right)\right) \Big/ \left(b \sqrt{40000. - 400.\, a \cos(1.193805209 + phia) + a^2}\right)\right),$$

$$\arctan\left(\frac{a \sin(1.822123739 + phia)}{-200. + a \cos(1.822123739 + phia)} \right)$$

$$+ \arccos\left(\left(0.5000000000 \left(-40000. + 400.\, a \cos(1.822123739 + phia) - 1.\, b^2 + c^2 \right.\right.\right.$$

$$\left.\left.\left. - 1.\, a^2\right)\right) \Big/ \left(b \sqrt{40000. - 400.\, a \cos(1.822123739 + phia) + a^2}\right)\right),$$

$$\arctan\left(\frac{a \sin(2.576105976 + phia)}{-200. + a \cos(2.576105976 + phia)} \right)$$

$$+ \arccos\left(\left(0.5000000000 \left(-40000. + 400.\, a \cos(2.576105976 + phia) - 1.\, b^2 + c^2 \right.\right.\right.$$

$$\left.\left.\left. - 1.\, a^2\right)\right) \Big/ \left(b \sqrt{40000. - 400.\, a \cos(2.576105976 + phia) + a^2}\right)\right),$$

$$\arctan\left(\frac{a \sin(4.021238597 + phia)}{-200. + a \cos(4.021238597 + phia)} \right)$$

$$+ \arccos\left(\left(0.5000000000 \left(-40000. + 400.\, a \cos(4.021238597 + phia) - 1.\, b^2 + c^2 \right.\right.\right.$$

$$\left.\left.\left. - 1.\, a^2\right)\right) \Big/ \left(b \sqrt{40000. - 400.\, a \cos(4.021238597 + phia) + a^2}\right)\right),$$

$$\arctan\left(\frac{a \sin(4.963716393 + phia)}{-200. + a \cos(4.963716393 + phia)} \right)$$

$$+ \arccos\left(\left(0.5000000000 \left(-40000. + 400.\, a \cos(4.963716393 + phia) - 1.\, b^2 + c^2 \right.\right.\right.$$

$$\left.\left.\left. - 1.\, a^2\right)\right) \Big/ \left(b \sqrt{40000. - 400.\, a \cos(4.963716393 + phia) + a^2}\right)\right) \Big]$$

```
> evalm(val_psi);
```

$$[-1.623156205 + eps, -1.692969375 + eps, -1.614429558 + eps, eps - 1.570796327, -1.500983157 + eps, eps - 1.570796327, -1.623156205 + eps]$$

```
> ferror := 0:
  for i from 1 to nfct do
  ferror := ferror + (Rq[i] - evalm(val_psi[i]))^2:
  od:

> pinit := {phia = phia_erg1, eps = epsilon*Degree, par_erg1[4],
  par_erg1[5], par_erg1[6] };
> evalf(subs(pinit, ferror));
```

$$pinit := \{a = 16.64858618, b = 189.1233007, c = 30.61935964, eps = 1.463425994\,\pi, phia = 1.221730477\}$$

$$0.003013099510$$

```
> with(Optimization):
  parOpt := Minimize(ferror,initialpoint = pinit, assume =
  nonnegative );
```

$$parOpt := [0.0004580967183947066 37, [a = 9.490636678246277 21, b = 165.8202510817748 13, c = 45.13294180668707 87, eps = 4.590484940885145 80, phia = 0.7372892178502898 91]]$$

This is a better solution.

```
> parOpt[2];
```

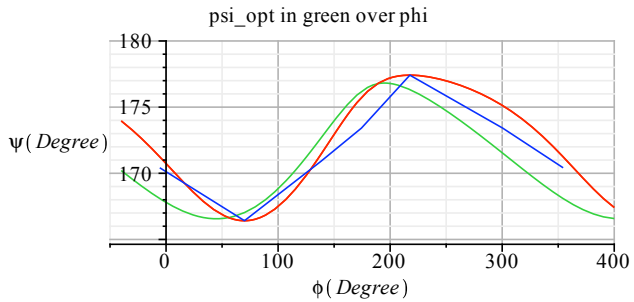
$$[a = 9.490636678246277 21, b = 165.8202510817748 13, c = 45.13294180668707 87, eps = 4.590484940885145 80, phia = 0.7372892178502898 91]$$

```
> parReq;
```

$$\{d = 200\}$$

```
> phi:= phiG*Pi/180:
  plo_psi3 := plot( {subs(parReq,parOpt[2], fpsi)*180/Pi,   subs
  (par_erg1, fpsi)*180/Pi },
  phiG=-40..400,gridlines=true,labels=['phi(Degree)','psi(Degree)
  '],title="psi_opt in green over phi", view=[-50..400, 165..180]
  );
  phi := 'phi':
  display(plo_psi3, plo_psi1, plo_Req);
```

$$plo\_psi3 := PLOT(\ldots)$$

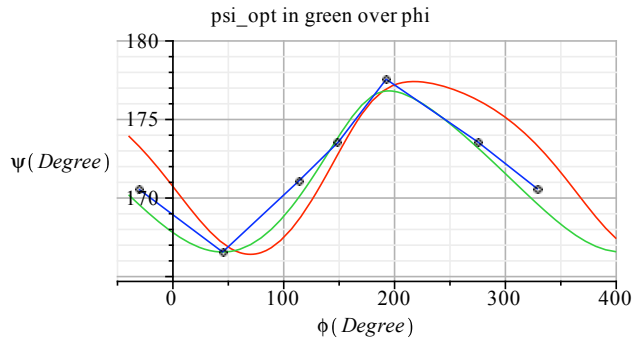


Test prints and a new plot, where the new build-in angle epsilon is used: the alpha values are shifted to the new solution.

```
> evalf(subs(parReq,parOpt[2], fpsi0)/Degree);     # new psi0
```
10.25653504
```
> evalf(subs(parReq,parOpt[2], fphi0)/Degree);     # new phi0
```
150.0943073
```
> phia_erg3 := evalf(subs(parReq,parOpt[2], fphia));    # new phia
  evalf(%/Degree);
```
phia_erg3 := 0.7829680989

44.86076755
```
> psia_erg3 := evalf(subs(parReq,parOpt[2], fpsia));    # new psia
  evalf(%/Degree);
```
psia_erg3 := 2.907084704

166.5636842
```
> epsilon_erg3 := evalf(psia_erg3/Degree + 7 + 90);
```
epsilon_erg3 := 263.5636842
```
> xx := Vector(nfct): yy := Vector(nfct):
  for i from 1 to nfct do
    xx[i] := evalf(subs(phia=phia_erg3     ,val_phi[i])/Degree);
    yy[i] := evalf(subs(eps=epsilon_erg3*Degree ,val_psi[i])
  /Degree);
  od:
  [xx, yy]:
> plo_Req3 := plot(xx,yy, color=blue):
  display(plo_psi3, plo_Req3,
    point([xx[1], yy[1]],lmax/50),
    point([xx[2], yy[2]],lmax/100),
    point([xx[3], yy[3]],lmax/100),
    point([xx[4], yy[4]],lmax/100),
    point([xx[5], yy[5]],lmax/100),
    point([xx[6], yy[6]],lmax/100),
    point([xx[7], yy[7]],lmax/100),
  gridlines=true);
```

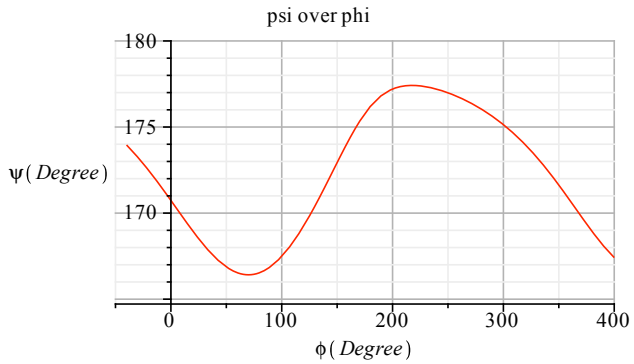


For further calculations let use the parameters of the ALT solution !

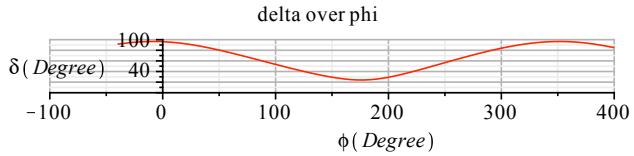
```
> par := par_erg1;
```
par := [*A0x* = -200, *A0y* = 0, *γ0* = 0, *a* = 16.64858618, *b* = 189.1233007, *c* = 30.61935964, *d* = 200, *k* = 0, κ = 0]

3 Kinematic Analysis

```
> par;
```
[*A0x* = -200, *A0y* = 0, *γ0* = 0, *a* = 16.64858618, *b* = 189.1233007, *c* = 30.61935964, *d* = 200, *k* = 0, κ = 0]
```
> phi:= phiG*Degree:
  plo_psi1 := plot(subs(par, fpsi)/Degree, phiG=-40..400,
  gridlines=true,labels=['phi(Degree)','psi(Degree)'],title="psi
  over phi", view=[-50..400, 165..180]):
  phi := 'phi':
  display(plo_psi1);
```




```
> phi:= phiG*Degree:
  plot(subs(psi=fpsi, par, fdelta)/Degree,phiG=-40..400,
  gridlines=true,labels=['phi(Degree)','delta(Degree)'],title=
  "delta over phi",view=[-100..400, 0..100]);
  phi := 'phi':
```




```
> mumin_erg := evalf(subs(par,mumin2/Degree))* degrees;
```

$$mumin\_erg := 27.94267067\ degrees$$

```
> phi0_erg := evalf(subs(par,fphi0/Degree))* degrees;
```

$$phi0\_erg := 147.6000001\ degrees$$

```
> psi0_erg := evalf(subs(par,fpsi0/Degree))* degrees;
```

$$psi0\_erg := 11.00000029\ degrees$$

```
> epsilon := evalf(psia_erg1/Degree + 7 + 90) * degrees;
```

$$\varepsilon := 263.4166790\ degrees$$

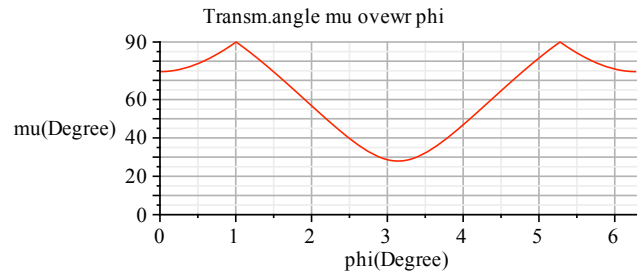
```
> mu_f2 := proc(phi)
          global par;
          if (evalf(subs(par,arccos((1/2)*(c^2+b^2-d^2+2*d*a*cos
  (phi)-a^2)/(b*c)))) > evalf(Pi/2)) then subs(par, Pi - arccos(
  (1/2)*(c^2+b^2-d^2+2*d*a*cos(phi)-a^2)/(b*c))) else subs(par,
  arccos((1/2)*(c^2+b^2-d^2+2*d*a*cos(phi)-a^2)/(b*c))) end if;
  end proc:

> evalf(mu_f2(0));      # Test for phi = 0
```

$$1.300954720$$

```
> # Note: ' proc '  allows to evaluate the proc containing if
  statements in the plot !!
```

```
> plot('mu_f2(phi)'/Degree, phi =0..2*Pi, labels=["phi(Degree)",
  "mu(Degree)"],title="Transm.angle mu ovewr phi",gridlines=true,
  view=[0..2*Pi,0..90]);
  phi := 'phi':
```



4 Linear velocity vK (phi, omega=phip=const) of points K / H

```
> omega := 'omega';
  omega_par := 2*Pi/1;
```

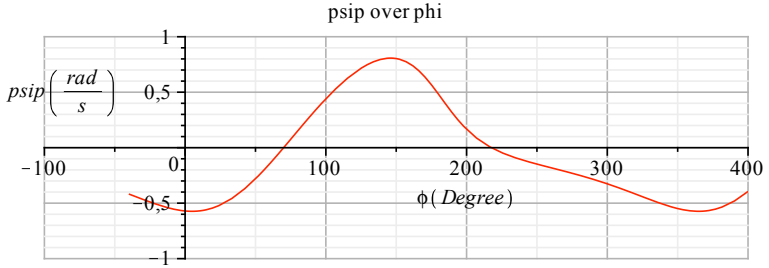
$$\omega := \omega$$

$$omega\_par := 2\,\pi$$

```
> fpsi;
```

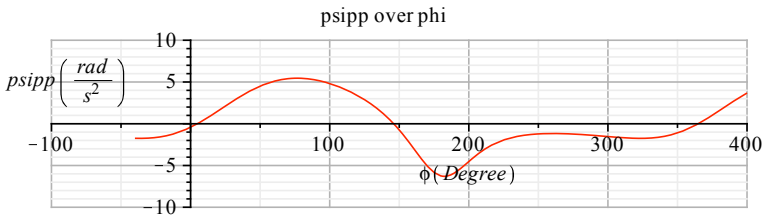
$$-\arctan\left(\frac{a\sin(\phi)}{d-a\cos(\phi)}\right)+\arccos\left(\frac{1}{2}\,\frac{-d^2+2\,d\,a\cos(\phi)-b^2+c^2-a^2}{b\sqrt{d^2-2\,d\,a\cos(\phi)+a^2}}\right)$$

```
> psi_p := simplify(diff(subs(par, fpsi), phi)) * omega:
> phi:= phiG*Degree:
  plot(subs(omega=omega_par, psi_p), phiG=-40..400,gridlines=
  true,labels=['phi(Degree)','psip(rad/s)'],title="psip over
  phi",view=[-100..400, -1..1]);
  phi := 'phi':
```




```
> psi_pp := simplify(diff(psi_p, phi) * omega):
> phi:= phiG*Degree:
  plot(subs(omega=omega_par, psi_pp), phiG=-40..400,gridlines=
```

```
true,labels=['phi(Degree)','psipp(rad/s^2)'],title="psipp over
phi",view=[-100..400, -10..10]);
phi := 'phi':
```



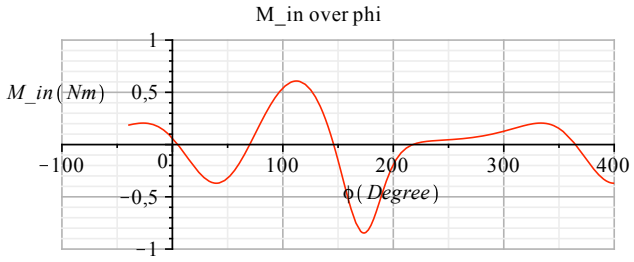
5 Input torque Min to drive the machine

sum of power = 0 = + Min omega - Irocker psipp psip

```
> par_dyn := {Irocker = 1.6 };
```

$par\_dyn := \{Irocker = 1.6\}$

```
> M_in := Irocker*psi_pp*psi_p / omega:
> phi:= phiG*Degree:
plot(subs(omega=omega_par, par, par_dyn, M_in), phiG=-40..400,
gridlines=true,labels=['phi(Degree)','M_in(Nm)'],title="M_in
over phi",view=[-100..400, -1..1]);
phi := 'phi':
```




```
> simplify(diff(M_in, phi)):
erg := fsolve(subs(omega=omega_par, par_dyn, %=0), phi=110*
Pi/180.);
Min_max := evalf(subs(phi=erg, omega=omega_par, par_dyn,  M_in)
);
```

$erg := 1.959610057$

$Min\_max := 0.6082271679$

```
> simplify(diff(M_in, phi)):
erg := fsolve(subs(omega=omega_par, par_dyn, %=0), phi=160*
Pi/180.);
Min_max := evalf(subs(phi=erg, omega=omega_par, par_dyn,  M_in)
);
```

$erg := 3.026570062$

$Min\_max := -0.8464968570$

End of Problem 1