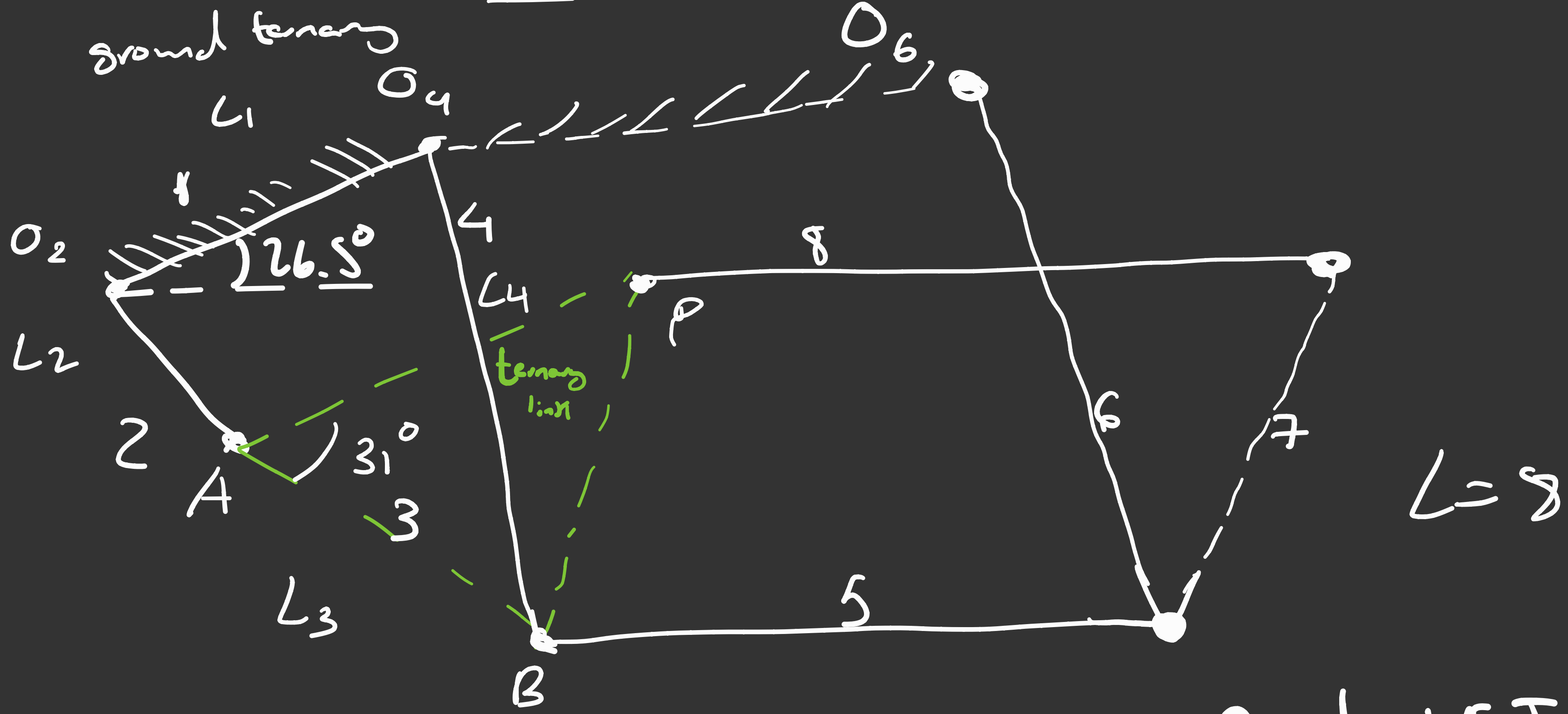
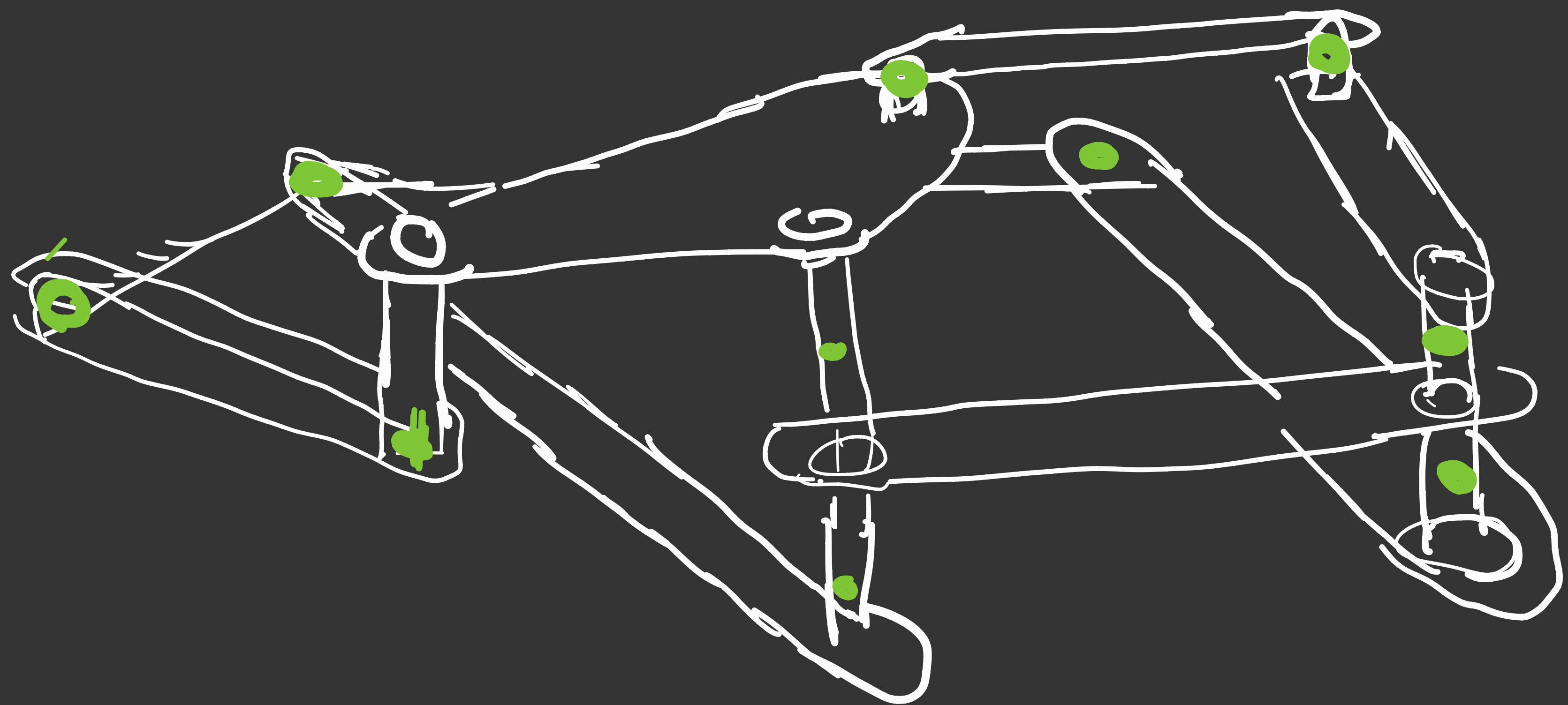
8-bar



no half Joints

Mobility Calculation with Gruebler's equation.



3D layout to count joints

$J_1 = 10$

$M = 3L - 3 - 2J_1 - J_2$

$M = 3(8) - 3 - 2(10) - 0 = 24 - 3 - 20 = 1$

$\boxed{M = 1}$

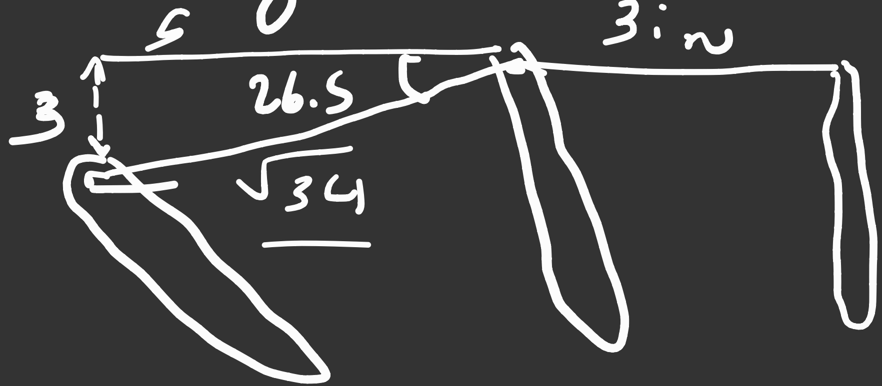
(matches rotational input)

Position analysis

$\sqrt{34}$ in $\approx$ 148 mm

info on ground link

Interest in Coupler Point P



Notice that link 4 and 6 move in parallel orientation are connected by link 5.

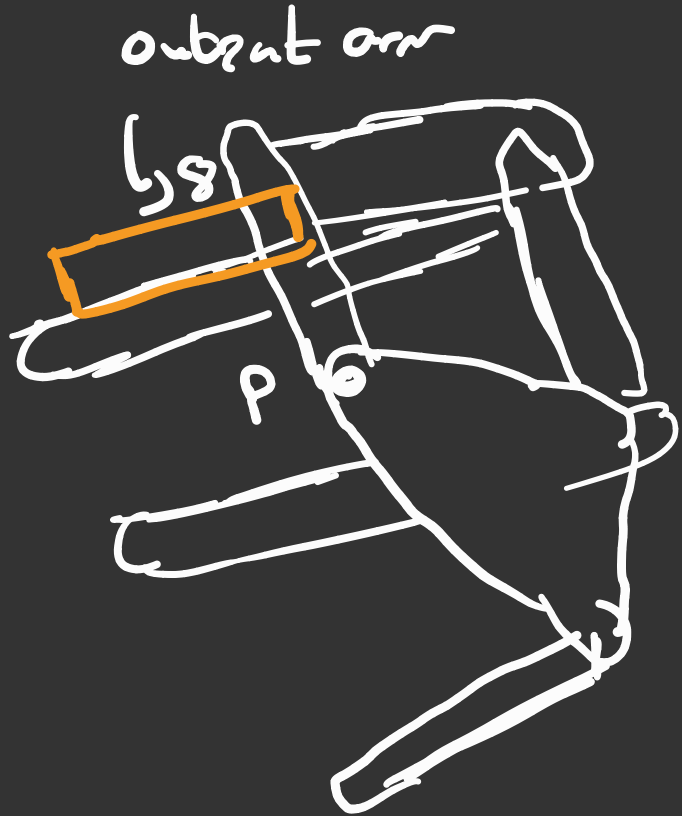
link 7 is identical to side BP of the ternary link

link 7 is connected to 5 and 8.

links 5 and 8 are setup to follow curvilinear motion with no rotation about an axis

$\boxed{\text{Coupler point P}}$ is on link 8.

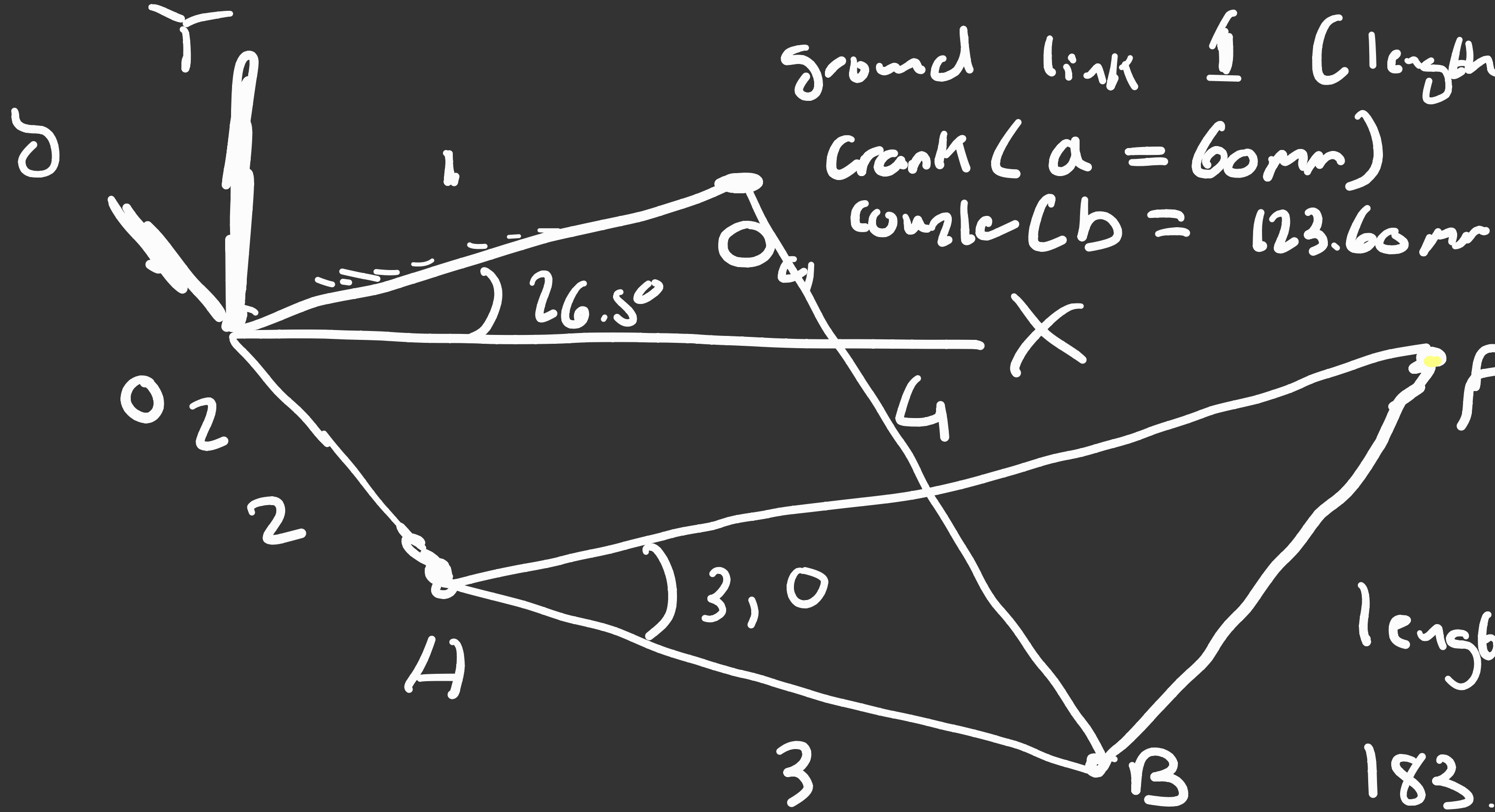
Side view



All points on link 8 move together (rigid) curvilinearly. So all points exhibit the same kinematic profile. (same velocity and acc).

Therefore, analyzing only 4 links is a sufficient kinematic study.

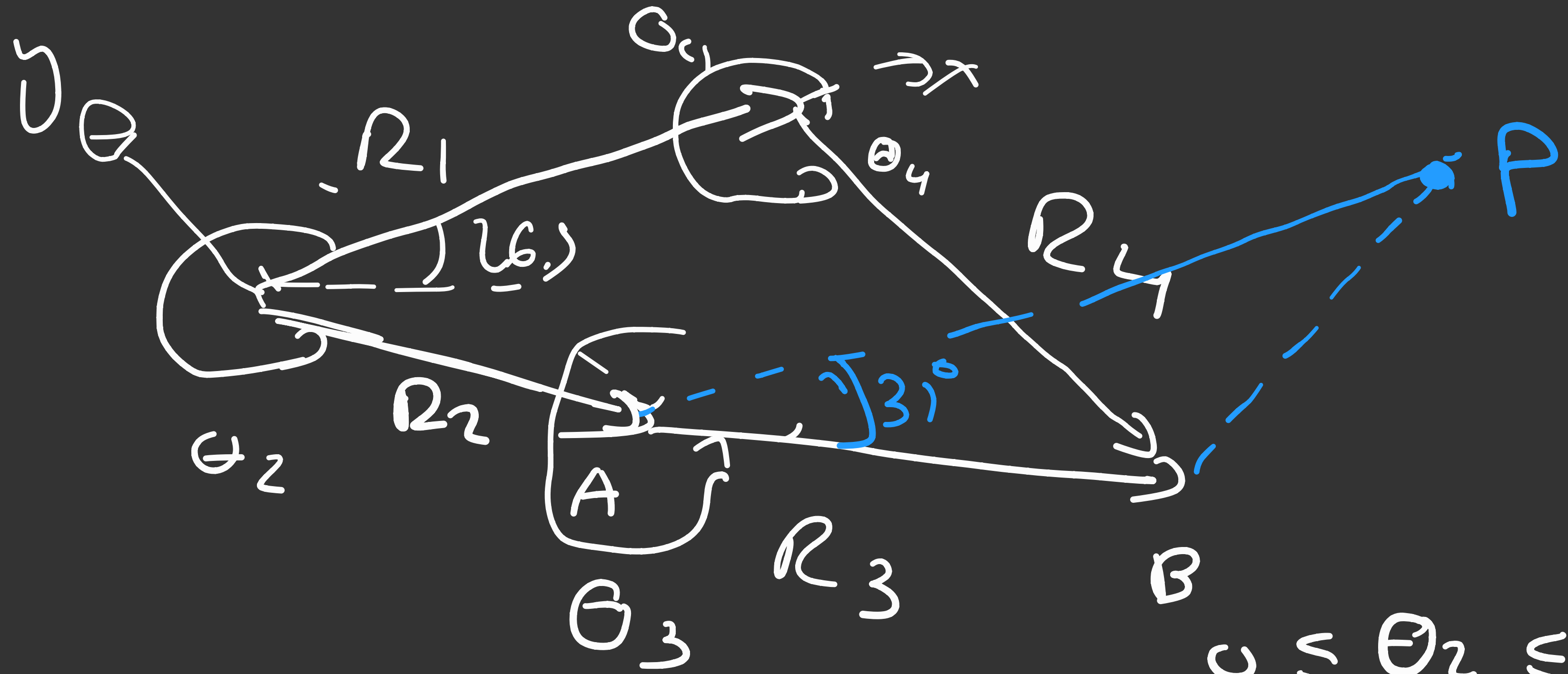
ground link 1 (length $d = 148\,mm$)
crank ($a = 60\,mm$)
coupler ($b = 123.60\,mm$)
rocker ($c = 115\,mm$)



length AP = 183.60 mm

use local coordinate frame and apply a rotation matrix to revert back to global XY

Goal: Point P position

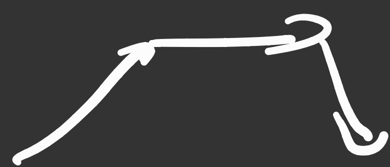


$$0 \le \theta_2 \le 360^\circ$$

$\theta_1 = 0$ in local frame.

$$R_2 + R_3 - R_4 - R_1 = 0$$

$$a e^{j\theta_2} + b e^{j\theta_3} - c e^{j\theta_4} - d e^{j\theta_1} = 0$$



Algorithmically, apply four bar position function which resolves vector components into cos and sin and solves for θ_3 and θ_4 in the local frame.

$$a(\cos\theta_2 + j\sin\theta_2) + b(\cos\theta_3 + j\sin\theta_3) - c(\cos\theta_4 + j\sin\theta_4)$$

$$-d(\cos\theta_1 + j\sin\theta_1) = 0$$

$\cos(\theta_1) = \cos(0) = 1$

$\sin(\theta_1) = \sin(0) = 0$

so $\quad a\cos\theta_2 + b\cos\theta_3 - c\cos\theta_4 = d$

$\quad a\sin\theta_2 + b\sin\theta_3 - c\sin\theta_4 = 0$

Nonlinear system of equations. 2 unknowns, $\boxed{\theta_3, \theta_4}$

used.

$$\xrightarrow[\text{crossed}]{\theta_2,\ \text{lengths}} \boxed{\text{fourbar}} \longrightarrow \theta_3, \theta_4$$

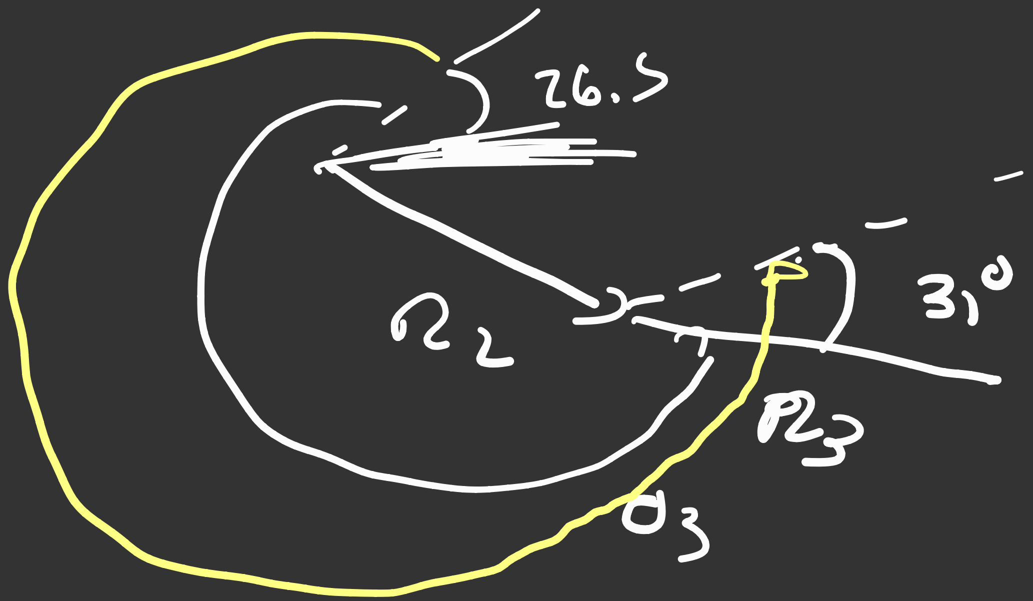
Shortest link, link 2 (input).

as it rotates, it crosses under ternary link 3 and ground. hence the configuration is <u>crossed</u>.

To find position of P.

$$\vec{R}_{PO_2} = \underbrace{\vec{R}_{AO_2}}_{\hookrightarrow R_2} + \vec{R}_{P/A}$$

$R_{PA} \Rightarrow$ angular position of $R_{PA} = \theta_{3_{local}} + 31^\circ$ $\Rightarrow$ ternary angle



$$R_{AO_2} = a\cos(\theta_2) + aj\sin\theta_2$$

$$R_{PA} = AP\cos(\theta_{3local} + 31) + AP \cdot j \cdot \sin(\theta_{3local} + 31)$$

$$\underbrace{R_{PO_2}}_{\text{local}} = \underbrace{R_{AO_2}}_{\text{local}} + \underbrace{R_{P/A}}_{\text{local}}$$

$26.5^\circ \rightarrow \alpha$

Finally return to global frame.

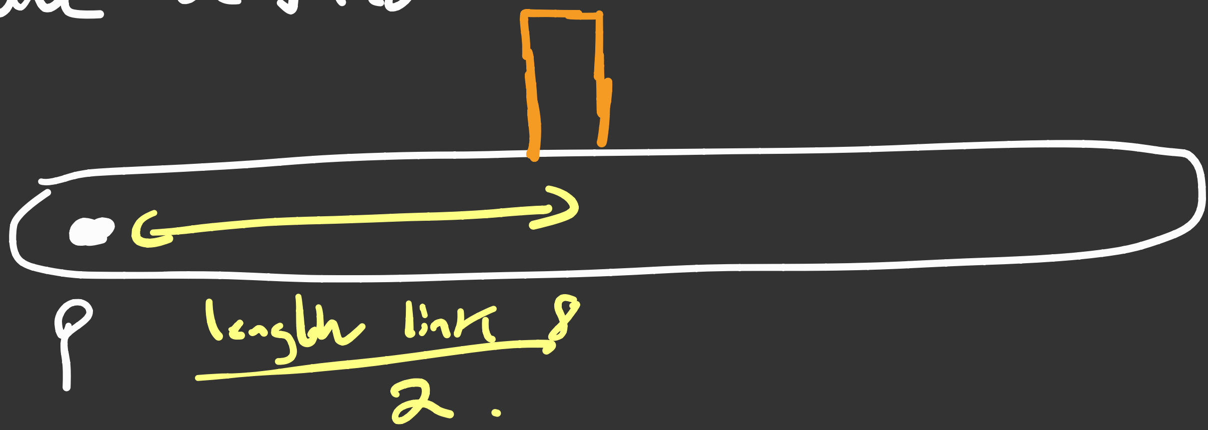
2x2

$$\text{rotation\_matrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

$$RPO_{2\,global} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} Re(RPO_{2\,local}) \\ Im(RPO_{2\,local}) \end{bmatrix}$$

$$\hookrightarrow \begin{bmatrix} X_{point\,P} \\ Y_{point\,P} \end{bmatrix}$$

Point P and the output cantilever (placed at the center.) will always have a fixed horizontal distance between them.



Velocity analysis

$R_2 + R_3 - R_4 - R_1 = 0$

in the Same local frame

$$a e^{j\theta_2} + b e^{j\theta_3} - c e^{j\theta_4} - d e^{j\theta_1} = 0$$

$$a \omega_2 j e^{j\theta_2} + b \omega_3 j e^{j\theta_3} - c \omega_4 j e^{j\theta_4} - d \cancel{\omega_1} e^{j\theta_1} = 0$$

$\omega_1 = 0$

$\omega_2 = \text{constant}$

↳ from motor

$\omega_2 = 100\,\text{rpm}$

$$a\omega_2 j(\cos\theta_2 + j\sin\theta_2) + b\omega_3 j(\cos\theta_3 + j\sin\theta_3)$$

$$-c\omega_4 j(\cos\theta_4 + j\sin\theta_4) = 0$$

$$a\omega_2(j\cos\theta_2 - \sin\theta_2) + b\omega_3(j\cos\theta_3 - \sin\theta_3)$$

$$-c\omega_4(j\cos\theta_4 - \sin\theta_4) = 0$$

unknown ω_3, and ω_4

$$-a\omega_2\sin\theta_2 - b\omega_3\sin\theta_3 + c\omega_4\sin\theta_4 = 0$$

$$a\omega_2\cos\theta_2 + b\omega_3\cos\theta_3 - c\omega_4\cos\theta_4 = 0$$

direct substitution to obtain ω_3 and ω_4

use compound angle theorem.

so $\boxed{\omega_3 = \frac{a\omega_2}{b} \frac{\sin(\theta_4 - \theta_2)}{\sin(\theta_3 - \theta_4)}}$

$$\vec{R}_{PO_2} = \underline{\vec{R}_{AO_2}} + \vec{R}_{P/A}$$

$\hookrightarrow R_2$

$$\boxed{\omega_4 = \frac{a\omega_2}{c} \frac{\sin(\theta_2 - \theta_3)}{\sin(\theta_4 - \theta_3)}}$$

$$\boxed{\vec{V}_{PO_2} = \vec{V}_{AO_2} + \vec{V}_{P/A}}$$

velocity of point A in local frame

$$V_{AO_2} = a\omega_2 j e^{j\theta_2} = a\omega_2(-\sin\theta_2 + j\cos\theta_2)$$

$$V_{PA} = AP\omega_3\left(j e^{j(\theta_3 + 31^\circ)}\right) = AP\omega_3\left(-\sin(\theta_3 + 31^\circ) + j\cos(\theta_3 + 31^\circ)\right)$$

$V_{PO_2\,local} = V_{AO_2} + V_{PA}$

$\rightarrow$ 2x2 $\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$ $\underline{\alpha = 26^\circ}$

$\rightarrow$ output velocity

$V_{PO_2\,global}$ = Rotation Matrix $\times$ $\begin{bmatrix} Real(V_{PO_2\,local}) \\ Im(V_{PO_2\,local}) \end{bmatrix}$

a plot of V_{out} vs. θ_2 can be generated

Acceleration analysis:

no coriolis

(no sliding link on rotating link

$$\frac{d}{dt}(V_2 + V_3 - V_4 - V_1) = 0$$

$$A_2 + A_3 - A_4 - A_1 = 0$$

$$\frac{d}{dt}$$

$$\left(a\omega_2 j e^{j\theta_2} + b\omega_3 j e^{j\theta_3} - c\omega_4 j e^{j\theta_4} - d\cancel{\omega_1} e^{j\theta_1} = 0\right)$$

$$\left(a\alpha_2 j e^{j\theta_2} - a\omega_2^2 e^{j\theta_2}\right) + \left(b\alpha_3 j e^{j\theta_3} - b\omega_3^2 e^{j\theta_3}\right)$$

$$-\left(c\alpha_4 j e^{j\theta_4} - c\omega_4^2 e^{j\theta_4}\right) = 0$$

$$\alpha_2 = 0$$

so $-a\omega_2^2 e^{j\theta_2} + b\alpha_3 j e^{j\theta_3} - b\omega_3^2 e^{j\theta_3} - c\alpha_4 j e^{j\theta_4} + c\omega_4^2 e^{j\theta_4}$

$= 0$

$$-a\omega_2^2(\cos\theta_2 + j\sin\theta_2) + b\alpha_3(j\cos\theta_3 - \sin\theta_3)$$

$$-b\omega_3^2(\cos\theta_3 + j\sin\theta_3) - c\alpha_4(j\cos\theta_4 - \sin\theta_4)$$

$$+c\omega_4^2(\cos\theta_4 + j\sin\theta_4) = 0$$

$$-a\omega_2^2\cos\theta_2 - b\alpha_3\sin\theta_3 - b\omega_3^2\cos\theta_3 + c\alpha_4\sin\theta_4$$

$$+c\omega_4^2\cos\theta_4 = 0$$

$$-a\omega_2^2\sin\theta_2 + b\alpha_3\cos\theta_3 - b\omega_3^2\sin\theta_3 - c\alpha_4\cos\theta_4$$

$$+c\omega_4^2\sin\theta_4 = 0$$

$$-b\alpha_3 \sin\theta_3 + c\alpha_4 \sin\theta_4 = a\omega_2^2 \cos\theta_2 + b\omega_3^2 \cos\theta_3 - c\omega_4^2 \cos\theta_4$$

$$b\alpha_3 \cos\theta_3 - c\alpha_4 \cos\theta_4 = a\omega_2^2 \sin\theta_2 + b\omega_3^2 \sin\theta_3 - c\omega_4^2 \sin\theta_4$$

Algebraically

$$\boxed{\alpha_3 = \frac{CD - AF}{AE - BD}}$$

where $A = c\sin\theta_4$ $B = b\sin\theta_3$

$D = c\cos\theta_4$ $E = b\cos\theta_3$

$$C = a\alpha_2 \sin\theta_2 + a\omega_2^2 \cos\theta_2 + b\omega_3^2 \cos\theta_3 - c\omega_4^2 \cos\theta_4$$

$$F = a\alpha_2 \cos\theta_2 - a\omega_2^2 \sin\theta_2 - b\omega_3^2 \sin\theta_3 + c\omega_4^2 \sin\theta_4$$

↳ MATLAB syntax

See textbook

Or 2×2 matrix

$\therefore$ Preferred approach analytically

$$\begin{array}{ccc} M & \vec{\alpha} & K \\ \begin{bmatrix} -b\sin\theta_3 & c\sin\theta_4 \\ b\cos\theta_3 & -c\cos\theta_4 \end{bmatrix} & \begin{bmatrix} \alpha_3 \\ \alpha_4 \end{bmatrix} = & \begin{bmatrix} -a\omega_2^2\cos\theta_2 + b\omega_3^2\cos\theta_3 - c\omega_4^2\cos\theta_4 \\ a\omega_2^2\sin\theta_2 + b\omega_3^2\sin\theta_3 - c\omega_4^2\sin\theta_4 \end{bmatrix} \end{array}$$

$$M\vec{\alpha} = K$$

$$\vec{\alpha} = M^{-1}K$$

Plot a vs. $\theta 2$ from 0 to 360

$\hookrightarrow \theta_2$

acceleration

$$\vec{A}_{PO_2} = \vec{A}_{AO_2} + A_{PA}$$

$$A_{AO_2} = \left(\alpha a_2 j e^{j\theta_2} - a\omega_2^2 e^{j\theta_2}\right) = -a\omega_2^2\left(\cos(\theta_2) + j\sin(\theta_2)\right)$$

$$A_{PA} = |AP|\left(\alpha_3 j e^{j\theta_3} - \omega_3^2 e^{j\theta_3}\right)$$

$$= AP\alpha_3\left(-\sin(\theta_3+31) + j\cos(\theta_3+31)\right)$$

$$+ \; -AP\omega_3^2\left(\cos(\theta_3+31) + j\sin(\theta_3+31)\right)$$

$$A_{PO_2\,global} = \underset{2\times2}{\text{rot matrix}} \times A_{PO_2\,local}$$

$$A_{PO_2\,local} = A_{AO_2\,local} + A_{PA\,local}$$

Mechanical considerations:

Torque ratio $(m_t) = \dfrac{1}{\text{angular velocity ratio}} = \dfrac{1}{m_v}$

$m_v = \dfrac{\omega_3}{\omega_2}$, So $m_t = \dfrac{\omega_2}{\omega_3}$

$\boxed{\text{Generate plot to study torque profile}}$

m_t vs. θ_2

mechanical advantage?

$m_A = \dfrac{F_{out}}{F_{in}}$

however the input is purely a torque acting at the axis of rotation of link 2 input power is from motor.

Selected motor: Greartisan 12V @ 100rpm (likely reduced with PWM)
1.1A rating

$$P = VI = 12 \times 1.1 = 13.2W$$

$$P_{in} = T_{in}\omega_2 \qquad \omega_2 = 100\,rpm \times \frac{2\pi}{60} = 10.472\ rad/s$$

$$\frac{P_{in}}{\omega_2} = \frac{13.2\ W}{10.472\ rad/s} = 1.2605\ Nm$$

assume a torque of 1.2605 Nm is desired
by manually exerting a force at the end of
input link 2, @ a = 60 mm away from the axis of
rotation. (i.e by hand)

$$T = f \cdot r$$

$$\frac{T}{r} = f$$

$$\frac{1.2605 \text{ Nm}}{\frac{60}{1000} \text{ m}} = 21.008 \text{ N}$$ needed to replicate
motor torque.

This can be used for mechanical advantage
yet it would not be useful, as motorizing is a more

Viable approach.

Assume no power loss!

So $P_{in} = P_{out}$

$P_{motor} = V_{out} \cdot F_{coupler}$

$\hookrightarrow$ coupler

$$\frac{P_{motor}}{V_{out}} = F_{coupler}$$

$$F(\theta_2) = \overset{\rightarrow W}{P_{motor}} \cdot \frac{1}{\underbrace{V_{out}(\theta_2)}_{\hookrightarrow \frac{mm}{s}}}$$

$F_{coupler}$ vs. θ_2 plot.

Table of Contents

Prepare Workspace

```
clc; close all;
clear
```

Position

```
L =  [148, 60, 123.60, 115]; % Link lengths
d = L(1); % define link lengths as letter variables
a = L(2);
b = L(3);
c = L(4);
delta = 1; % crossed configuration
t1 = 0; % all angles defined relative to theta_1

t3 =[]; % empty vector for t3 and t4
t4= [];
t2 = 0:1:360;
for n = 1:length(t2) % loop to obtain theta_3 and theta_4
    output =(four_bar(L,t1,deg2rad(n),delta)); % run fourbar position
 function
    t3(n) = output(3);
    t4(n) = output(4);
end

P = 183.60 ; % length of AP on ternary link
% With the local frame
RA_O2 = a*cos(deg2rad(t2)) + a*sin(deg2rad(t2))*1i; % define A's
 position with respect to origin
R_PA = P*(cos(t3 + deg2rad(31)) + sin(t3 + deg2rad(31))*1i); %
 Position P with respect to A

R_PO2 = RA_O2 + R_PA; % P position with respect to O2 (coupler)

in_coor_pos = [real(R_PO2); imag(R_PO2)]; % Partition into real and
 imaginary components
rot_angle = 26.5 ; % angle for coordinate system transformation
% Apply rot_mat rotational matrix to obtain the coupler's position in
 the
% global frame
```

```
out_coor = rot_mat(rot_angle) * in_coor_pos;

X = out_coor(1,:); % Index for x-pos
Y = out_coor(2,:); % index for y-pos
```

Velocity

```
omega_2 =  convangvel(100,'rpm','rad/s'); % 100 rpm

% From algebra, define omega_3 and omega_4 (angular velocities)
omega_3 = (a./b).*(omega_2).*(sin(t4-deg2rad(t2))./sin(t3-t4));
omega_4 = (a./c).*(omega_2).* (sin(deg2rad(t2)-t3)./sin(t4-t3));

% Define A's velocity with respect to O2, P's velocity with respect to
 A
VA_O2 = a.*omega_2.*-sin(deg2rad(t2)) +
 a.*omega_2.*cos(deg2rad(t2)).*1i;
V_PA = P.*omega_3.*(-sin(t3 + deg2rad(31)) + cos(t3 +
 deg2rad(31))*1i);
V_PO2 = VA_O2 + V_PA; % local frame velocity of coupler point with
 respect to O2


% Convert to global frame
in_coor_vel = [real(V_PO2); imag(V_PO2)];
out_coor_vel = rot_mat(rot_angle) * in_coor_vel;
V_x = out_coor_vel(1,:);
V_y = out_coor_vel(2,:);
```

Acceleration

```
% Matrix approach. Define vectors for accelerations and angular
 (alpha)
% acceleration
A_AO2 = [];
A_PA = [];
A_PO2 = [];
alpha_3 = [];
alpha_4 = [];

% Solve the linear system for M*(alpha_vec) = k for every instance of
% theta_2
for n = 1:length(t2)
    M = [-b.*sin(t3(n)) c.*sin(t4(n)); b.*cos(t3(n)) -c.*cos(t4(n))];
    k = [a.*omega_2.^2.*cos(deg2rad(t2(n))) +
 b.*omega_3(n).^2.*cos(t3(n)) - c*omega_4(n).^2*cos(t4(n));
 a*omega_2.^2*sin(deg2rad(t2(n))) + b*omega_3(n).^2*sin(t3(n)) -
 c*omega_4(n).^2*sin(t4(n))];
    alpha_vec = inv(M)*k;
    alpha_3(n) = alpha_vec(1);
    alpha_4(n) = alpha_vec(2);
    A_AO2(n) = -a.*(omega_2).^2.*(cos(deg2rad(t2(n))) +
 1i.*sin(deg2rad(t2(n))));
```

```
    A_PA(n) = P.*alpha_3(n).*(-sin(t3(n)+deg2rad(31)) + 1i.*cos(t3(n)
 + deg2rad(31))) + -P.*(omega_3(n)).^2.*(cos(t3(n)+ deg2rad(31)) +
 1i.*sin(t3(n)+ deg2rad(31)));
    A_PO2(n) =  A_AO2(n) + A_PA(n);
end

% Convert to global frame
in_coor_acc = [real(A_PO2); imag(A_PO2)];
out_coor_acc = rot_mat(rot_angle) * in_coor_acc;
A_x = out_coor_acc(1,:);
A_y = out_coor_acc(2,:);
```

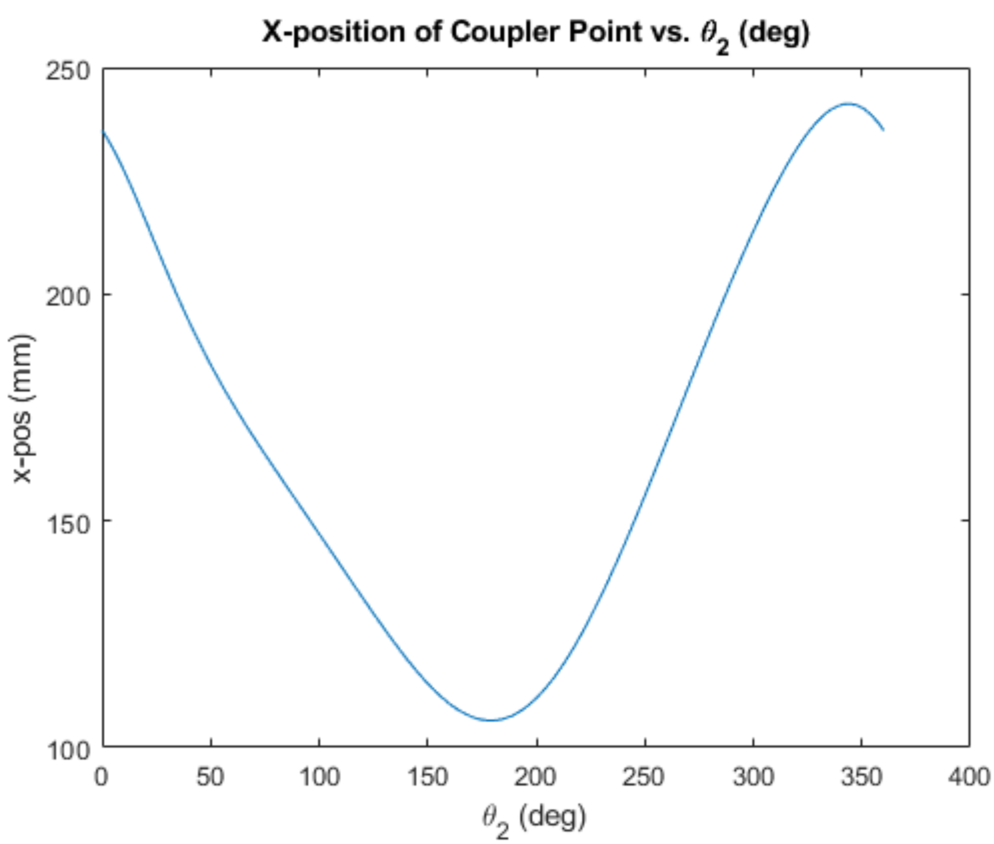
Mechanical Considerations

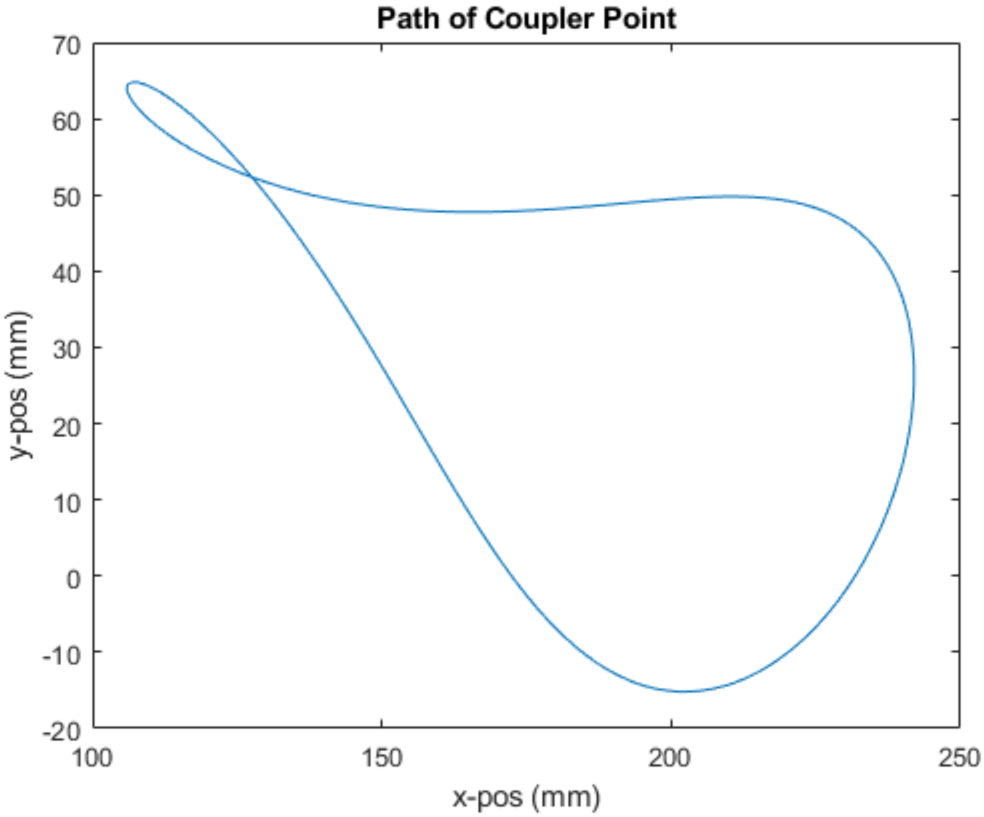
```
angular_velocity_ratio = omega_3./omega_2;
torque_ratio = 1./(angular_velocity_ratio);
P_motor = 12 * 1.1; % voltage * current

V_mag = vecnorm([V_x; V_y].',2,2).';
F_coupler = P_motor./V_mag * 1000; % Watt-second/mm to N conversion
```

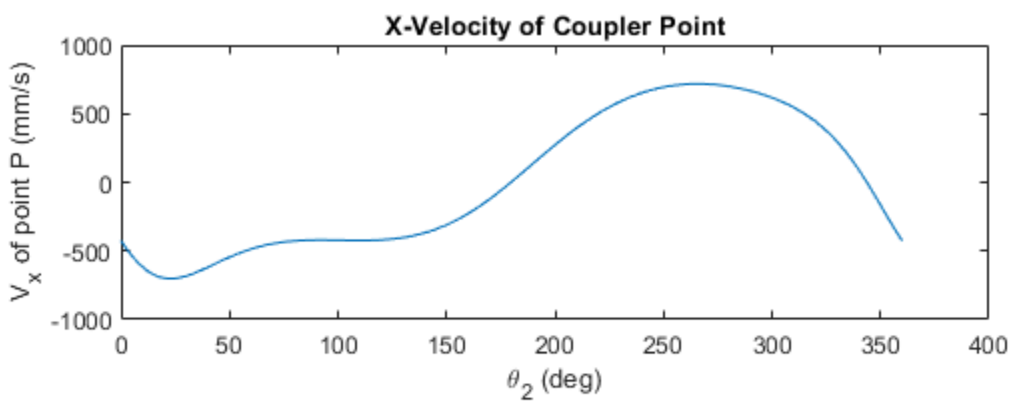
Plot results

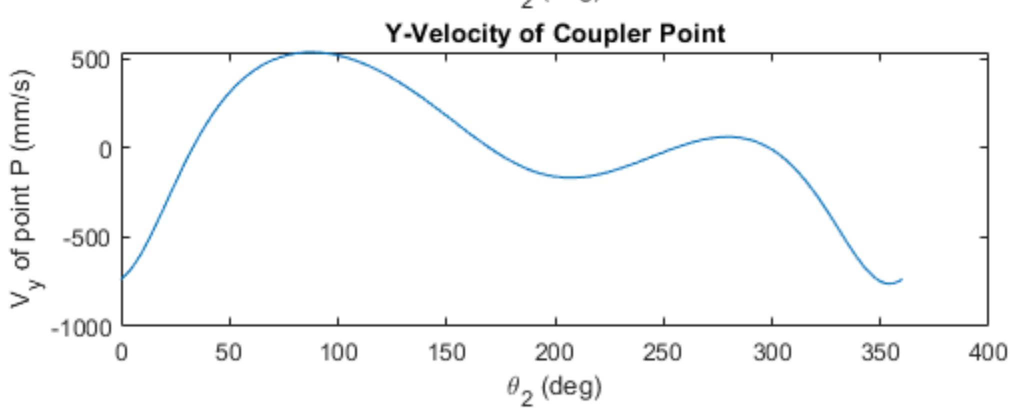
```
figure(1)
plot(t2,X)
xlabel('\theta_2 (deg)');ylabel('x-pos (mm)'); title('X-position of
 Coupler Point vs. \theta_2 (deg)')

figure(2)
plot(X,Y)
xlabel('x-pos (mm)');ylabel('y-pos (mm)'); title('Path of Coupler
 Point')

figure(3)
subplot(2,1,1)
plot(t2,V_x)
xlabel('\theta_2 (deg)');ylabel('V_x of point P (mm/s)'); title('X-
Velocity of Coupler Point')

subplot(2,1,2)
plot(t2,V_y)
xlabel('\theta_2 (deg)');ylabel('V_y of point P (mm/s)'); title('Y-
Velocity of Coupler Point')

figure(4)
subplot(2,1,1)
plot(t2,A_x)
xlabel('\theta_2 (deg)');ylabel('A_x of point P (mm/s^2)'); title('X-
Acceleration of Coupler Point')

subplot(2,1,2)
plot(t2,A_y)
```

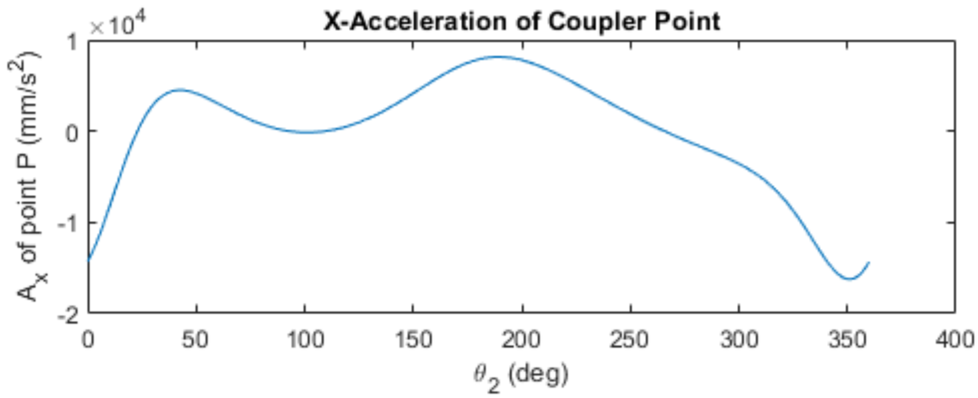
```
xlabel('\theta_2 (deg)');ylabel('A_y of point P (mm/s^2)'); title('Y-
Acceleration of Coupler Point')



figure(5)
subplot(3,1,1)
plot(t2,angular_velocity_ratio)
xlabel('\theta_2 (deg)');ylabel('angular velocity ratio (m_v)');
 title('Angular Velocity Ratio vs. \theta_2')

subplot(3,1,2)
plot(t2,torque_ratio)
xlabel('\theta_2 (deg)');ylabel('torque ratio (m_T)'); title('Torque
 Ratio vs. \theta_2')

subplot(3,1,3)
plot(t2,F_coupler)
xlabel('\theta_2 (deg)');ylabel('Coupler force F_c (N)');
 title('Coupler force (N) vs. \theta_2')
```



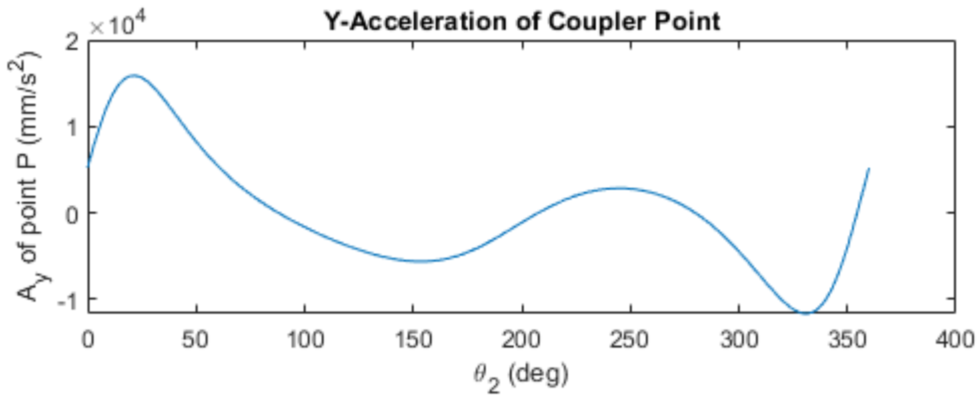

Path of Coupler Point
y-pos (mm)
x-pos (mm)
70
60
50
40
30
20
10
0
-10
-20
100
150
200
250


X-Velocity of Coupler Point
V_x of point P (mm/s)
θ_2 (deg)
1000
500
0
-500
-1000
0
50
100
150
200
250
300
350
400

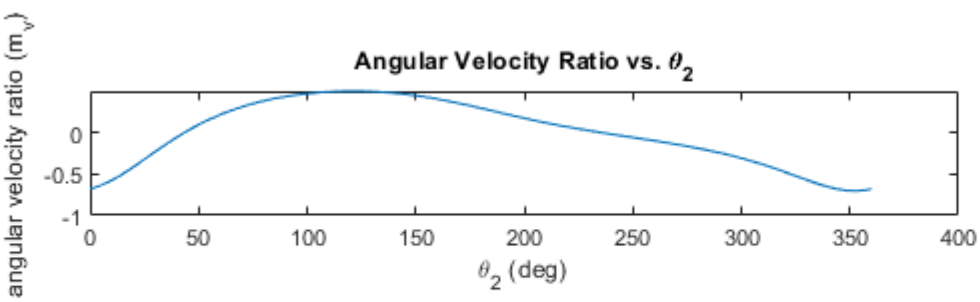

Y-Velocity of Coupler Point
V_y of point P (mm/s)
θ_2 (deg)
500
0
-500
-1000
0
50
100
150
200
250
300
350
400



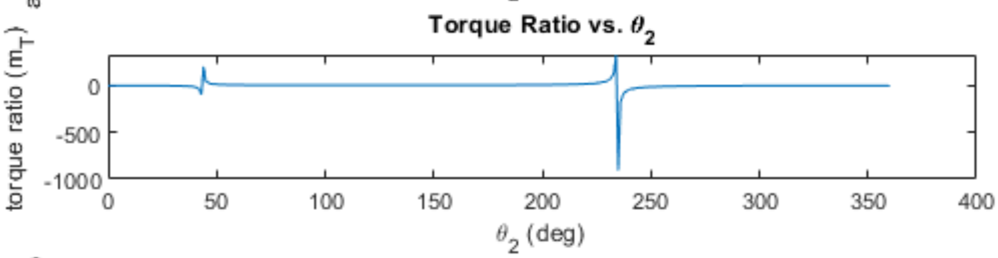
X-Acceleration of Coupler Point
×10^4
A_x of point P (mm/s^2)
θ_2 (deg)



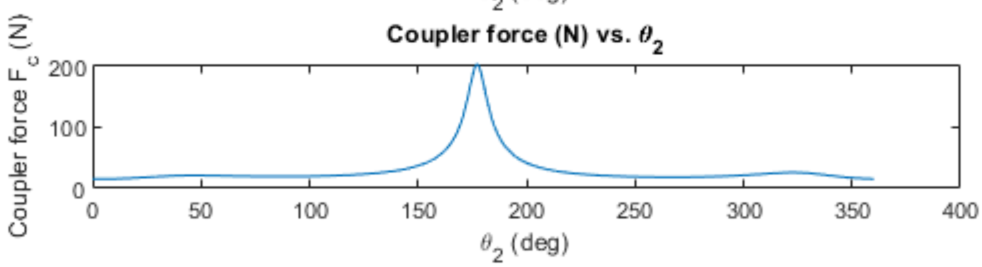
Y-Acceleration of Coupler Point
×10^4
A_y of point P (mm/s^2)
θ_2 (deg)



Angular Velocity Ratio vs. θ_2
angular velocity ratio (m_V)
θ_2 (deg)



Torque Ratio vs. θ_2
torque ratio (m_T)
θ_2 (deg)



Coupler force (N) vs. θ_2
Coupler force F_c (N)
θ_2 (deg)

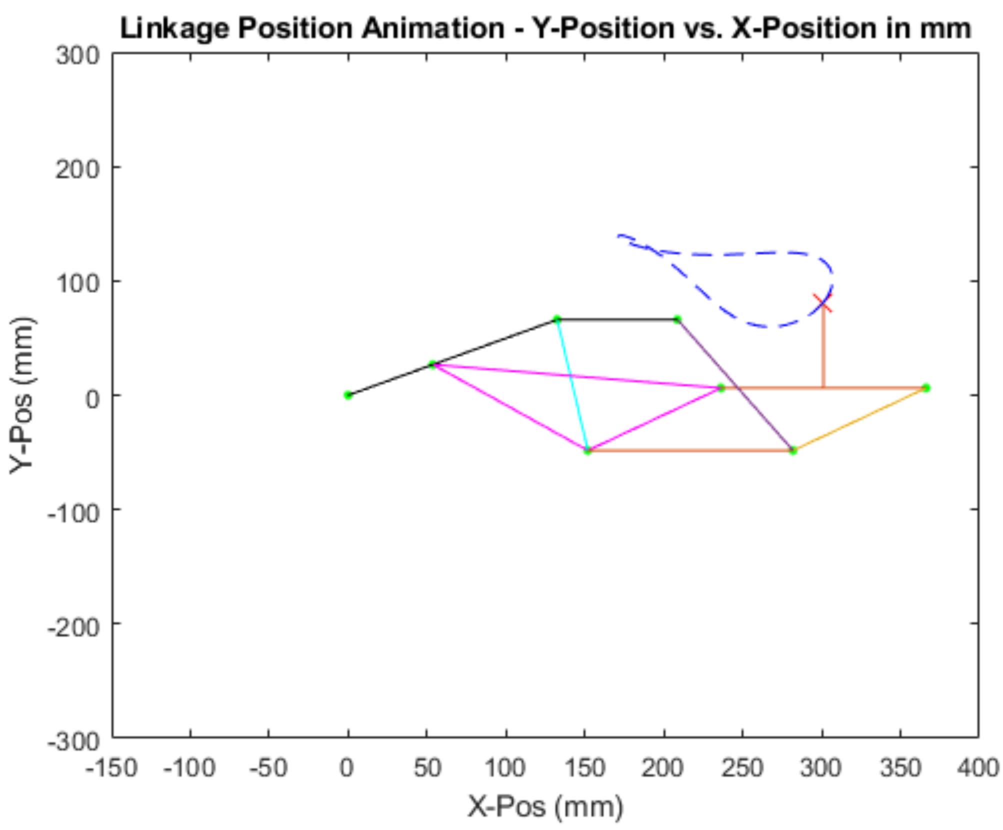
Animation

```
% Setup joint positions, grounds, and rotate coordinates to global as
% needed
figure(6)

t2 = 0:1:360;
t2  = deg2rad(t2);

O2x = 0;
O2y = 0;
O2 = [0; 0];


O4= [O2(1,:) + d*cos(deg2rad(26.5)); O2(2,:) + d*sin(deg2rad(26.5))];
O4x = O4(1,:);
O4y = O4(2,:);

O6= [O4x + 76.2 ; O4y];

O6x = O6(1,:);
O6y = O6(2,:);



A = [a*cos(t2);a*sin(t2)]; % rotate A last to avoid double rotating
 (simplest syntax)


B = rot_mat(rot_angle)*[ A(1,:) + b*cos(t3); A(2,:) + b*sin(t3)];
Bx = B(1,:);
By = B(2,:);


point_P = rot_mat(rot_angle)*[A(1,:) + P*cos(t3 + deg2rad(31)); A(2,:)
 + P*sin(t3+deg2rad(31))];
point_Px = point_P(1,:);
point_Py = point_P(2,:);

B_para = [Bx + 130; By];

Bx_para = B_para(1,:);
By_para = B_para(2,:);
P_para = [point_Px + 130 ;point_Py];
Px_para = P_para(1,:);
Py_para = P_para(2,:);



A = rot_mat(rot_angle) * [a*cos(t2);a*sin(t2)]; % rotate A last to
 avoid double rotating (simplest syntax)
Ax = A(1,:);
Ay = A(2,:);
```

```
xTrace = point_Px + (130/2);
yTrace = point_Py(n) + 75;

vidStr = 'linkanimation' % name of the video
nFrames = length(t2);
duration = 10; % length of video in secs

vidObj = VideoWriter(vidStr,'MPEG-4'); % create video object in mp4
vidObj.FrameRate = floor(nFrames/duration);% set frame rate
vidObj.Quality = 100; % max vid quality

open(vidObj);
% loop for animation
for n = 1:3: length(t2)

  plot(O2x,O2y,'g.','MarkerSize',10); hold on
  plot(O4x,O4y,'g.','MarkerSize',10)
  plot(Ax(n), Ay(n),'g.', 'MarkerSize',10);
  plot(Bx(n),By(n),'g.','MarkerSize',10)
  plot(point_Px(n),point_Py(n),'g.','MarkerSize',10)
  plot(Bx_para(n),By_para(n),'g.','MarkerSize',10)
  plot(Px_para(n),Py_para(n),'g.','MarkerSize',10)
  plot(O6x,O6y,'g.','MarkerSize',10)
  plot((point_Px(n) + (130/2)), (point_Py(n) + 75), 'rx','markersize',
 10)


  plot([O2x Ax(n)],[O2y Ay(n)], 'r-')
  plot([O2x O4x],[O2y O4y], 'k-')
  plot([O4x O6x],[O4y O6y], 'k-')

  plot([Ax(n) Bx(n)],[Ay(n) By(n)], 'm-')
  plot([O4x Bx(n)],[O4y By(n)], 'c-')
  plot([Ax(n) point_Px(n)], [Ay(n) point_Py(n)],'m-')
  plot([Bx(n) point_Px(n)], [By(n) point_Py(n)],'m-')
  plot([Bx(n) Bx_para(n)], [By(n) By_para(n)], 'color','#D95319')
  plot([point_Px(n) Px_para(n)], [point_Py(n)
 Py_para(n)],'color','#D95319')
  plot([Bx_para(n) Px_para(n)], [By_para(n)
 Py_para(n)],'color','#EDB120')
  plot([O6x Bx_para(n)],[O6y By_para(n)],'color','#7E2F8E')
  plot([(point_Px(n) + (130/2)) (point_Px(n) + (130/2))], [point_Py(n)
 (point_Py(n) + 75)], 'color', '#D95319')

  xTrace = point_Px(1:n) + (130/2);
  yTrace = point_Py(1:n) + 75;
  plot(xTrace ,yTrace,'b--'); hold off


  title('Linkage Position Animation - Y-Position vs. X-Position in
 mm')
  xlim([-150 400])
```

```
  ylim([-300 300])
  xlabel('X-Pos (mm)')
  ylabel('Y-Pos (mm)')

 drawnow  %Force plot to update before code advances
 writeVideo(vidObj, getframe(gcf));
%  Draw arm trace (parallel to coupler point P)
%  if n ==  length(t2)
%      hold on
%      for n = 1:length(t2)
%         plot(point_Px(n) + (130/2) ,point_Py(n) + 75,'b.')
%         drawnow
%      end
%      hold off
end

close(vidObj);
```

```
vidStr =

    'linkanimation'
```



Published with MATLAB® R2020b