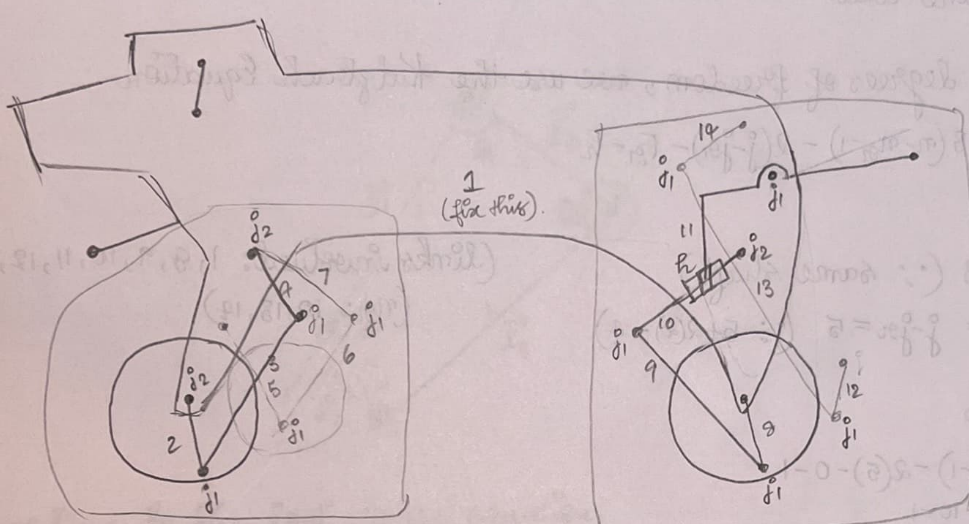
ME2220: INDIVIDUAL PROJECT

PHASE-II

NAME: MANASVI GAMPA
ROLL.NO: ME23BTECH11035

PULL ALONG DOG

Kinematic Diagram depicting the individual mechanisms present in the toy:



Mechanism 1 Mechanism 2

Mechanism 1:

Mechanism 1 consists of the front legs. Both legs are connected via shafts passing through the body of the toy; one moving effectively causing movement of the other. Each leg is a crank-rocker mechanism.

To find the degrees of freedom, we have the Kutzbach Equation.

$$F = 3(n - n_r - 1) - 2(j - j_r) - F_r - h$$

Here,

$n = 7$ (links involved: 1, 2, 3, 4, 5, 6, 7)

$j_{eff} = j - j_r = 4$ ($\because \underline{4} + 2(2) - \underline{4}$)

$n_r = 3$ ($\because$ same shaft)

$$\Rightarrow F = 3(7-3-1) - 2(4) - 0 - 0$$

$$F = 3(3) - 2(4)$$

$$F = 1$$

Mechanism 2:

Mechanism 2 consists of the rear legs + tail combination. Both legs are connected via shafts passing through the body of the toy; one moving effectively causing movement of the other. Each leg is a crank-rocker mechanism. To the rocker of one of the legs, a slotted link mechanism is arranged which effectively causes the up-down motion of the tail.

To find the degrees of freedom, we use the Kutzbach Equation

$$F = 3(n - n_r - 1) - 2(j - j_r) - F_r - h$$

Here,

$n = 8$

$n_r = 3$ ($\because$ same shaft)

$j_{eff} = j - j_r = 5$ ($\because 5 + 2(2) - 4$)

$h = 1$

$F_r = 0$

(links involved: 1, 8, 9, 10, 11, 12, 13, 14)

(n_r: 12, 13, 14)

$\Rightarrow F = 3(8-3-1) - 2(5) - 0 - 1$

$F = 3(4) - 10 - 1$

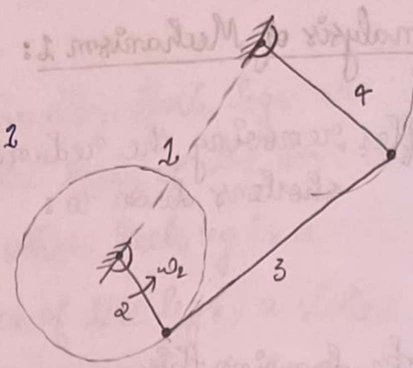
$F = 1$

$\therefore$ The whole toy as a whole has 3 degrees of freedom, 1 due to mechanism 1, 1 due to mechanism 2, 1 due to free movement of the ears (connected using same shaft).

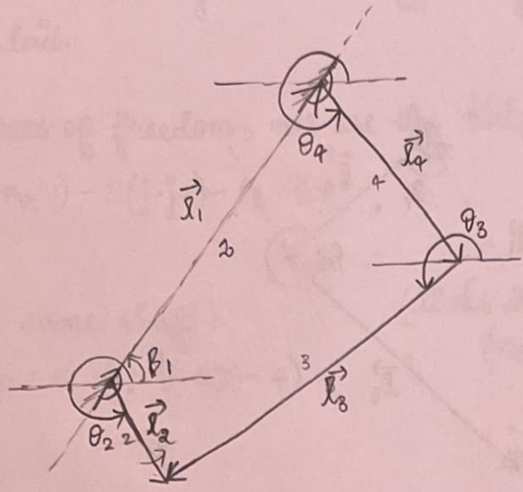
Analysis of Mechanism 1:

After removing the redundant links, mechanism 1 shortens down to:

(1 is fixed link i.e frame)



Re-drawing this,



Knowns: $l_1, l_2, l_3, l_4, \beta_1$ (fixed)

θ_2: input angle

θ_3 & θ_4 determined via displacement analysis

According to the loop closure equation,

$$\vec{l_1} + \vec{l_4} + \vec{l_3} = \vec{l_2}$$

$$l_1 e^{i\beta_1} + l_4 e^{i\theta_4} + l_3 e^{i\theta_3} = l_2 e^{i\theta_2}$$

Seperating real & imaginary components, we have,

$$l_1\cos\beta_1 + l_4\cos\theta_4 + l_3\cos\theta_3 = l_2\cos\theta_2$$

$$l_1\sin\beta_1 + l_4\sin\theta_4 + l_3\sin\theta_3 = l_2\sin\theta_2$$

Displacement Analysis:

Squaring & adding to eliminate θ_3,

$$l_3^2 = \left[l_2\cos\theta_2 - (l_1\cos\beta_1 + l_4\cos\theta_4)\right]^2 + (l_2\sin\theta_2 - l_1\sin\beta_1 - l_4\sin\theta_4)^2$$

$$\Rightarrow l_3^2 = l_2^2\cos^2\theta_2 + l_1^2\cos^2\beta_1 + l_4^2\cos^2\theta_4 - 2l_1l_2\cos\beta_1\cos\theta_2 - 2l_2l_4\cos\theta_2\cos\theta_4$$
$$+ 2l_1l_4\cos\beta_1\cos\theta_4$$
$$+ l_2^2\sin^2\theta_2 + l_1^2\sin^2\beta_1 + l_4^2\sin^2\theta_4 - 2l_1l_2\sin\beta_1\sin\theta_2 + l_1l_4\sin\beta_1\sin\theta_4$$
$$- 2l_2l_4\sin\theta_2\sin\theta_4$$

$$l_3^2 = l_1^2 + l_2^2 + l_4^2 - 2l_2 l_4 \cos\theta_2 \cos\theta_4 + 2l_1 l_4 \cos\beta_1 \cos\theta_4 + 2l_1 l_4 \sin\beta_1 \sin\theta_4$$
$$- 2l_2 l_4 \sin\theta_2 \sin\theta_4 - 2l_1 l_2 \cos\beta_1 \cos\theta_2 - 2l_1 l_2 \sin\beta_1 \sin\theta_2$$

$$\Rightarrow \sin\theta_4\left(2l_2 l_4 \sin\theta_2 - 2l_1 l_4 \sin\beta_1\right) + \cos\theta_4\left(2l_2 l_4 \cos\theta_2 - 2l_1 l_4 \cos\beta_1\right)$$
$$= \left(l_1^2 + l_2^2 + l_4^2 - l_3^2\right) - 2l_1 l_2\left(\cos\theta_2 \cos\beta_1 + \sin\theta_2 \sin\beta_1\right)$$

$$2l_4\left(l_2 \sin\theta_2 - l_1 \sin\beta_1\right)\sin\theta_4 + 2l_4\left(l_2\cos\theta_2 - l_1\cos\beta_1\right)\cos\theta_4$$
$$= \left(l_1^2 + l_2^2 + l_4^2 - l_3^2\right) - 2l_1 l_2\left(\cos\theta_2 \cos\beta_1 + \sin\theta_2 \sin\beta_1\right)$$

$\Rightarrow$ $a\sin\theta_4 + b\cos\theta_4 = c.$

where,

$a = (l_2 \sin\theta_2 - l_1 \sin\beta_1)$

$b = (l_2 \cos\theta_2 - l_1 \cos\beta_1)$

$$c = \frac{(l_1^2 + l_2^2 + l_4^2 - l_3^2)}{2l_4} - \frac{l_1 l_2}{l_4}\cos(\theta_2 - \beta_1)$$

From the obtained equation,

$$\frac{2a\tan(\theta_4/2)}{1+\tan^2(\theta_4/2)} + \frac{b\left(1-\tan^2(\theta_4/2)\right)}{1+\tan^2(\theta_4/2)} = c$$

$$(-b-c)\tan^2(\theta_4/2) + 2a\tan(\theta_4/2) + (b-c) = 0$$

$$\tan(\theta_4/2) = \frac{-2a \pm \sqrt{(2a)^2 + 4(b^2-c^2)}}{-2b-2c}$$

$$\Rightarrow \theta_4 = 2\tan^{-1}\left(\frac{a \pm \sqrt{a^2+b^2-c^2}}{b+c}\right)$$

(two orientations)

$$\boxed{\theta_4^{(1)} = 2\tan^{-1}\left(\frac{a - \sqrt{a^2+b^2-c^2}}{b+c}\right), \quad \theta_4^{(2)} = 2\tan^{-1}\left(\frac{a + \sqrt{a^2+b^2-c^2}}{b+c}\right)}$$

Similarly, we obtain $\theta_3^{(1)}, \theta_3^{(2)}$,

a, b remain the same

$$c' = \frac{(l_1^2 + l_2^2 + l_3^2 - l_4^2)}{2l_4} - \frac{l_1 l_2}{l_3}\cos(\theta_2 - \beta_1)$$

$$\boxed{\theta_3^{(1)} = 2\tan^{-1}\left(\frac{a + \sqrt{a^2 + b^2 - c'^2}}{b + c'}\right), \quad \theta_3^{(2)} = 2\tan^{-1}\left(\frac{a - \sqrt{a^2 + b^2 - c'^2}}{b + c'}\right)}$$

Velocity Analysis:

From loop closure, we have,

$$l_1\cos\beta_1 + l_4\cos\theta_4 + l_3\cos\theta_3 = l_2\cos\theta_2 \quad (1)$$

$$l_1\sin\beta_1 + l_4\sin\theta_4 + l_3\sin\theta_3 = l_2\sin\theta_2 \quad (2)$$

Differentiating wrt time, (Here $\beta_1 \to$ fixed) i.e const

$$-l_4\sin\theta_4\,\omega_4 - l_3\sin\theta_3\,\omega_3 = -l_2\sin\theta_2\,\omega_2 \quad (3)$$

$$l_4\cos\theta_4\,\omega_4 + l_3\cos\theta_3\,\omega_3 = l_2\cos\theta_2\,\omega_2 \quad (4)$$

$\Rightarrow$ (3) $\times\cos\theta_3$ + (4) $\times\sin\theta_3$

$$\Rightarrow l_4\omega_4(\cos\theta_4\sin\theta_3 - \sin\theta_4\cos\theta_3) + l_2\omega_2(\sin\theta_2\cos\theta_3 - \sin\theta_3\cos\theta_2) = 0$$

$$\Rightarrow l_4\omega_4\sin(\theta_3 - \theta_4) + l_2\omega_2\sin(\theta_2 - \theta_3) = 0$$

$$\Rightarrow \boxed{\omega_4 = \frac{l_2\omega_2\sin(\theta_2 - \theta_3)}{l_4\sin(\theta_4 - \theta_3)}}$$

(where, l_2, l_4, θ_2 are known initially, ω_2 is taken as some value, θ_3, θ_4 are obtained from the displacement analysis above)

Similarly, we obtain,

$$\Rightarrow \boxed{\omega_3 = \frac{l_2\omega_2\sin(\theta_2 - \theta_4)}{l_3\sin(\theta_3 - \theta_4)}}$$

Acceleration Analysis:

from (3) & (4), Differentiating wrt time,

$$-l_4(\cos\theta_4\,\omega_4^2+\alpha_4\sin\theta_4)-l_3(\cos\theta_3\,\omega_3^2+\alpha_3\sin\theta_3)=-l_2(\cos\theta_2\,\omega_2^2+\alpha_2\sin\theta_2) \quad (5)$$

$$l_4(-\sin\theta_4\,\omega_4^2+\cos\theta_4\,\alpha_4)+l_3(-\sin\theta_3\,\omega_3^2+\alpha_3\cos\theta_3)=l_2(-\sin\theta_2\,\omega_2^2+\alpha_2\cos\theta_2) \quad (6)$$

Here, l_4, l_3, l_2 are known

θ_2 is given $\Rightarrow$ θ_3 & θ_4 are determined via displacement analysis.

ω_2 is fixed $\Rightarrow$ ω_3 & ω_4 are determined via velocity analysis

Hence, by solving (5) & (6), we can determine α_3, α_4.

Using matrix multiplication & numerical ways, we obtain,

$$\alpha_4=\frac{A_y\cos\theta_3-A_x\sin\theta_3}{l_4\sin(\theta_4-\theta_3)}$$

$$\alpha_3=\frac{A_x\sin\theta_4-A_y\cos\theta_4}{l_3\sin(\theta_3-\theta_4)}$$

where,

$$A_x=\alpha_2 l_2\cos\theta_2-\omega_2^2 l_2\sin\theta_2+\omega_4^2 l_4\sin\theta_4+\omega_3^2 l_3\sin\theta_3$$

$$A_y=\alpha_2 l_2\sin\theta_2+\omega_2^2 l_2\cos\theta_2-\omega_4^2 l_4\cos\theta_4-\omega_3^2 l_3\cos\theta_3$$

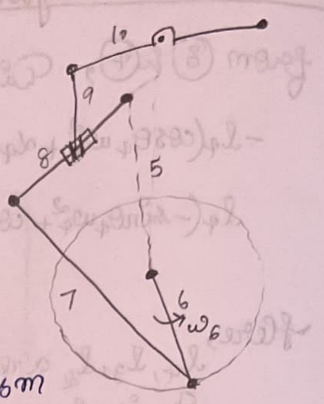
Physical measurements observed from the toy:

$$\left.\begin{aligned} l_1 &= 8.5\text{cm}\\ l_2 &= 2\text{cm}\\ l_3 &= 7.5\text{cm}\\ l_4 &= 5\text{cm}\end{aligned}\right\} \quad s+l<p+q$$

↳ satisfies Grashof's criterion

$\beta_1 = 80°$

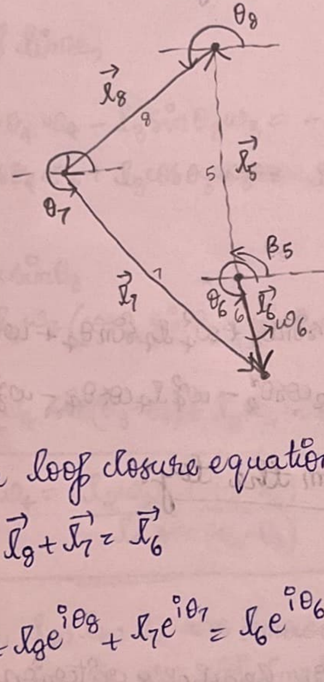
Analysis of Mechanism 2:



After removing the redundant links, mechanism 2 shortens down to:

Mechanism 2 comprises of a four bar (crank-rocker) mechanism and a slotted link mechanism.

Analysis of the rear-leg i.e crank-rocker mechanism:



Knowns: $l_5, l_6, l_7, l_8, \beta_5 = \text{const.}$

θ_6: input angle

θ_7 & θ_8 determined via displacement analysis

According to the loop closure equation,

$$\vec{l_5} + \vec{l_8} + \vec{l_7} = \vec{l_6}$$

$$l_5 e^{i\beta_5} + l_8 e^{i\theta_8} + l_7 e^{i\theta_7} = l_6 e^{i\theta_6}$$

Seperating the real & imaginary components,

$$l_5 \cos\beta_5 + l_8 \cos\theta_8 + l_7 \cos\theta_7 = l_6 \cos\theta_6 \quad (7)$$

$$l_5 \sin\beta_5 + l_8 \sin\theta_8 + l_7 \sin\theta_7 = l_6 \sin\theta_6. \quad (8)$$

Displacement Analysis:

(Similar to mechanism 1

$l_1 \rightarrow l_5$

$l_2 \rightarrow l_6$

$l_3 \rightarrow l_7$

$l_4 \rightarrow l_8$)

we obtain,

$$a\sin\theta_8 + b\cos\theta_8 = c$$

where,

$$a = (l_6\sin\theta_6 - l_5\sin\beta_5)$$
$$b = (l_6\cos\theta_6 - l_5\cos\beta_6)$$
$$c = \frac{l_5^2 + l_6^2 + l_8^2 - l_7^2}{2l_8} - \frac{l_5 l_6}{l_8}\cos(\theta_6 - \beta_6)$$

obtain θ_8

Similarly,

for θ_7,

a, b remain the same

$$c' = \frac{l_5^2 + l_6^2 + l_7^2 - l_8^2}{2l_7} - \frac{l_5 l_6}{l_7}\cos(\theta_6 - \beta_5)$$

Velocity Analysis:

(Similar to mechanism 1)

from ⑦, ⑧, differentiating wrt time,

$$-l_8\sin\theta_8\omega_8 - l_7\sin\theta_7\omega_7 + l_6\sin\theta_6\omega_6 = 0 \quad (9)$$
$$l_8\cos\theta_8\omega_8 + l_7\cos\theta_7\omega_7 - l_6\cos\theta_6\omega_6 = 0 \quad (10)$$

we obtain,

$$\omega_8 = \frac{l_6\omega_6\sin(\theta_6 - \theta_7)}{l_8\sin(\theta_8 - \theta_7)} \quad ; \quad \omega_7 = \frac{l_6\omega_6\sin(\theta_6 - \theta_8)}{l_7\sin(\theta_7 - \theta_8)}$$

Acceleration Analysis:

(Similar to mechanism 2)

from ⑨ & ⑩,

$$-l_8(\cos\theta_8\omega_8^2 + \alpha_8\sin\theta_8) - l_7(\cos\theta_7\omega_7^2 + \alpha_7\sin\theta_7) + l_6(\cos\theta_6\omega_6^2 + \alpha_6\sin\theta_6) = 0 \quad (11)$$
$$l_8(-\sin\theta_8\omega_8^2 + \cos\theta_8\alpha_8) + l_7(-\sin\theta_7\omega_7^2 + \alpha_7\cos\theta_7) - l_6(-\sin\theta_6\omega_6^2 + \alpha_6\cos\theta_6) = 0 \quad (12)$$

Here, l_8, l_6, l_7 are known

θ_6 is given $\Rightarrow \theta_7$ & θ_8 are determined via displacement analysis

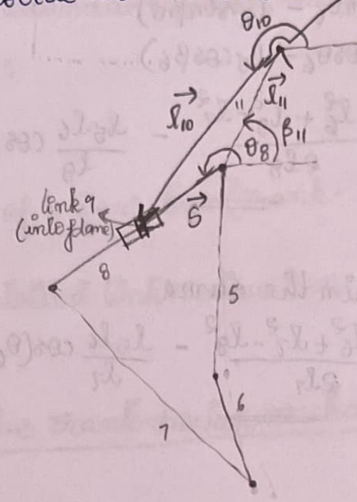
$\underbrace{\omega_6 = \omega_2}$ is fixed $\Rightarrow \omega_7$ & ω_8 are determined via velocity analysis

both wheels (legs) move via same pulling force

Hence, by solving ⑪ & ⑫, we can determine α_7, α_8.

Analysis of the tail mechanism i.e slotted link mechanism:

Considering that link 9 lies into the plane, we can simplify the system further to provide an end resultant of:



Knowns: $l_{11}, l_{10}, \beta_{11}$ (fixed)
θ_8 from crank-rocker in mechanism 2

Unknowns: S, θ_{10}

According to the loop closure equation,

$$\vec{l}_{11} + \vec{l}_{10} = \vec{S}$$

$$l_{11} e^{i\beta_{11}} + l_{10} e^{i\theta_{10}} = S e^{i\theta_8}$$

Separating the real & imaginary components,

$$l_{11}\cos\beta_{11} + l_{10}\cos\theta_{10} = S\cos\theta_8 \quad (13)$$

$$l_{11}\sin\beta_{11} + l_{10}\sin\theta_{10} = S\sin\theta_8 \quad (14)$$

Displacement Analysis:

Squaring & adding,

$$l_{10}^2 = (S\cos\theta_8 - l_{11}\cos\beta_{11})^2 + (S\sin\theta_8 - l_{11}\sin\beta_{11})^2$$

$$l_{10}^2 = S^2\cos^2\theta_8 + l_{11}^2\cos^2\beta_{11} - 2Sl_{11}\cos\theta_8\cos\beta_{11} + S^2\sin^2\theta_8 + l_{11}^2\sin^2\beta_{11} - 2Sl_{11}\sin\theta_8\sin\beta_{11}$$

$$l_{10}^2 = S^2 + l_{11}^2 - 2Sl_{11}\cos(\theta_8 - \beta_{11})$$

$$\Rightarrow S^2 - S[2l_{11}\cos(\theta_8 - \beta_{11})] + l_{11}^2 - l_{10}^2 = 0.$$

i.e. $aS^2 - bS + c = 0$

where, $a = 1$

$b = 2l_{11}\cos(\theta_8 - \beta_{11})$

$c = l_{11}^2 - l_{10}^2$

$$\Rightarrow S = \frac{b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$\Rightarrow S = \frac{2l_{11}\cos(\theta_8-\beta_{11}) \pm \sqrt{4l_{11}^2\cos^2(\theta_8-\beta_{11}) - 4(l_{11}^2-l_{10}^2)}}{2}$$

$$\boxed{S = l_{11}\cos(\theta_8-\beta_{11}) \pm \sqrt{l_{11}^2\cos^2(\theta_8-\beta_{11}) - l_{11}^2+l_{10}^2}}$$

Here,

$S^{(1)} = l_{11}\cos(\theta_8-\beta_{11}) - \sqrt{l_{11}^2\cos^2(\theta_8-\beta_{11}) - l_{11}^2+l_{10}^2}$ $\Rightarrow$ corresponds to minimum sliding distance

$S^{(2)} = l_{11}\cos(\theta_8-\beta_{11}) + \sqrt{l_{11}^2\cos^2(\theta_8-\beta_{11}) - l_{11}^2+l_{10}^2}$ $\Rightarrow$ corresponds to maximum sliding distance.

$\Rightarrow$ Length of the slot $= S^{(2)} - S^{(1)}$

$$= 2\sqrt{l_{11}^2\cos^2(\theta_8-\beta_{11}) - l_{11}^2+l_{10}^2}$$

which can be verified to be of an certain range (θ_8) and can be verified physically (i.e in toy).

$\Rightarrow$ Squaring & adding,

$$l_{11}^2\cos^2\beta_{11} + l_{10}^2\cos^2\theta_{10} + 2l_{11}l_{10}\cos\beta_{11}\cos\theta_{10} + l_{11}^2\sin^2\beta_{11} + l_{10}^2\sin^2\theta_{10} + 2l_{11}l_{10}\sin\beta_{11}\sin\theta_{10} = S^2$$

$$= l_{11}^2 + l_{10}^2 + 2l_{11}l_{10}\cos(\theta_{10}-\beta_{11}) = S^2$$

$$\Rightarrow \cos(\theta_{10}-\beta_{11}) = \frac{S^2 - l_{11}^2 - l_{10}^2}{2l_{11}l_{10}}$$

$$\Rightarrow \boxed{\theta_{10} = \beta_{11} + \cos^{-1}\left(\frac{S^2 - l_{11}^2 - l_{10}^2}{2l_{11}l_{10}}\right)}$$

Hence, based on the values of $S^{(1)}$ & $S^{(2)}$ we obtain $\theta_{10}^{(1)}$ & $\theta_{10}^{(2)}$ respectively

Velocity Analysis:

from (13) & (14), differentiating wrt time,

$$-l_{10}\sin\theta_{10}\omega_{10} = V\cos\theta_8 - S\sin\theta_8\omega_8 \quad (15) \qquad (\beta_{11} \text{ is const}).$$

$$l_{10}\cos\theta_{10}\omega_{10} = V\sin\theta_8 + S\cos\theta_8\omega_8 \quad (16)$$

$\Rightarrow$ (15) $\times \cos\theta_8$ + (16) $\times \sin\theta_8 \Rightarrow$

$$-l_{10}\omega_{10}\sin\theta_{10}\cos\theta_8 + l_{10}\omega_{10}\cos\theta_{10}\sin\theta_8 = V\cos^2\theta_8 + V\sin^2\theta_8$$

$$l_{10}\omega_{10}\sin(\theta_8-\theta_{10}) = V \quad \Rightarrow \quad \boxed{V = l_{10}\omega_{10}\sin(\theta_8-\theta_{10})}$$

Similarly,

$$\Rightarrow (15)\times\sin\theta_8 - (16)\times\cos\theta_8 \Rightarrow$$

$$+l_{10}\omega_{10}\sin\theta_8\sin\theta_{10} + l_{10}\omega_{10}\cos\theta_8\cos\theta_{10} = +s\omega_8\sin^2\theta_8 + s\omega_8\cos^2\theta_8$$

$$l_{10}\omega_{10}\cos(\theta_8-\theta_{10}) = s\omega_8$$

$$\Rightarrow \boxed{\omega_{10} = \frac{s\omega_8}{l_{10}\cos(\theta_8-\theta_{10})}}$$

So, $\boxed{v = s\omega_8\tan(\theta_8-\theta_{10})}$

Here, θ_8 is known, ω_8 is known; s, θ_{10} is found from displacement analysis.

Acceleration Analysis:

from (15) & (16), differentiating wrt time,

$$-l_{10}(\cos\theta_{10}\,\omega_{10}^2 + \sin\theta_{10}\,\alpha_{10}) = a\cos\theta_8 - v\omega_8\sin\theta_8 - (v\sin\theta_8\,\omega_8 + s\cos\theta_8\,\omega_8^2 + s\sin\theta_8\,\alpha_8) \quad (17)$$

$$l_{10}(-\sin\theta_{10}\,\omega_{10}^2 + \cos\theta_{10}\,\alpha_{10}) = a\sin\theta_8 + v\omega_8\cos\theta_8 + (v\cos\theta_8\,\omega_8 - s\sin\theta_8\,\omega_8^2 + s\cos\theta_8\,\alpha_8) \quad (18)$$

From (17) & (18) using numerical methods, we obtain,

$$a = s\left[\sec^2(\theta_8-\theta_{10})(\omega_{10}-\omega_8)\omega_8 + \tan(\theta_8-\theta_{10})\alpha_8\right]$$

$$\alpha_{10} = \frac{s\alpha_8\cos(\theta_8-\theta_{10}) + s\omega_8(\omega_8-\omega_{10})\sin(\theta_8-\theta_{10})}{l_{10}\cos^2(\theta_8-\theta_{10})}$$

Here,
$\alpha_8, \theta_8, \omega_8$ is known; s, θ_{10} is found from displacement analysis, ω_{10}, v is known from velocity analysis.

$\therefore$ we can obtain a, α_{10}.

Physical measurements observed from the toy:

$l_5 = 8.5cm$
$l_6 = 2cm$
$l_7 = 7cm$
$l_8 = 6cm$

$s + l < p + q \rightarrow$ satisfies grashof criterion

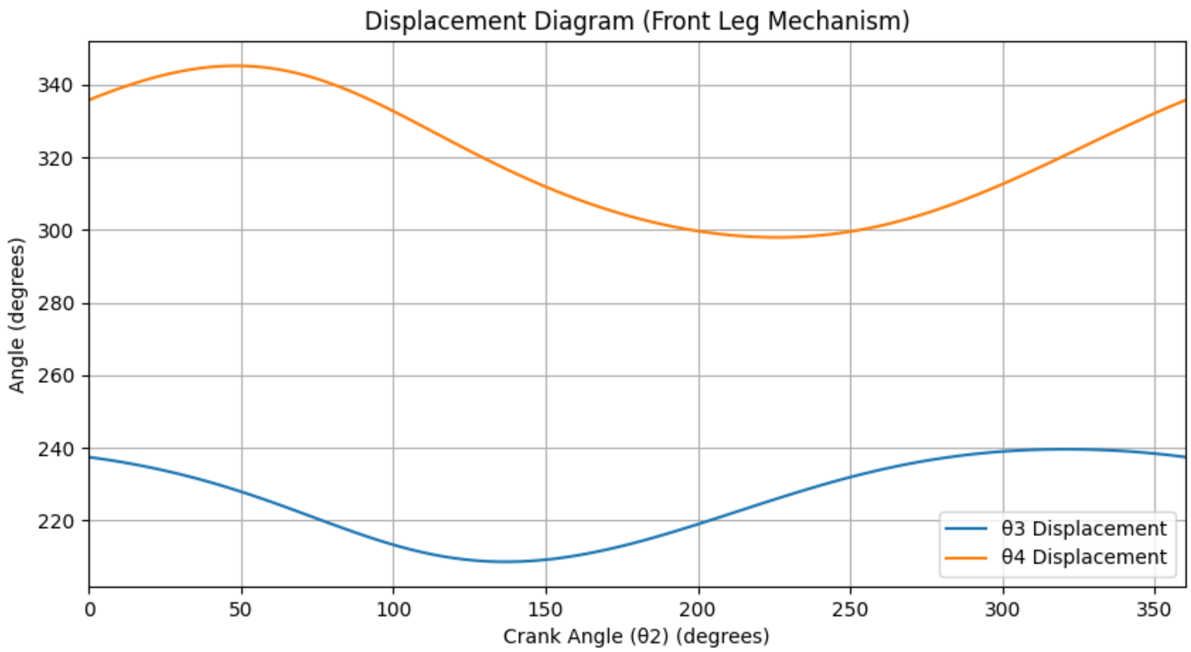
$\beta_5 = 70^\circ$
$\beta_4 = 90^\circ$
$l_{11} = 2cm$
$l_{10} = 5cm$

DISPLACEMENT, VELOCITY & ACCELERATION DIAGRAMS

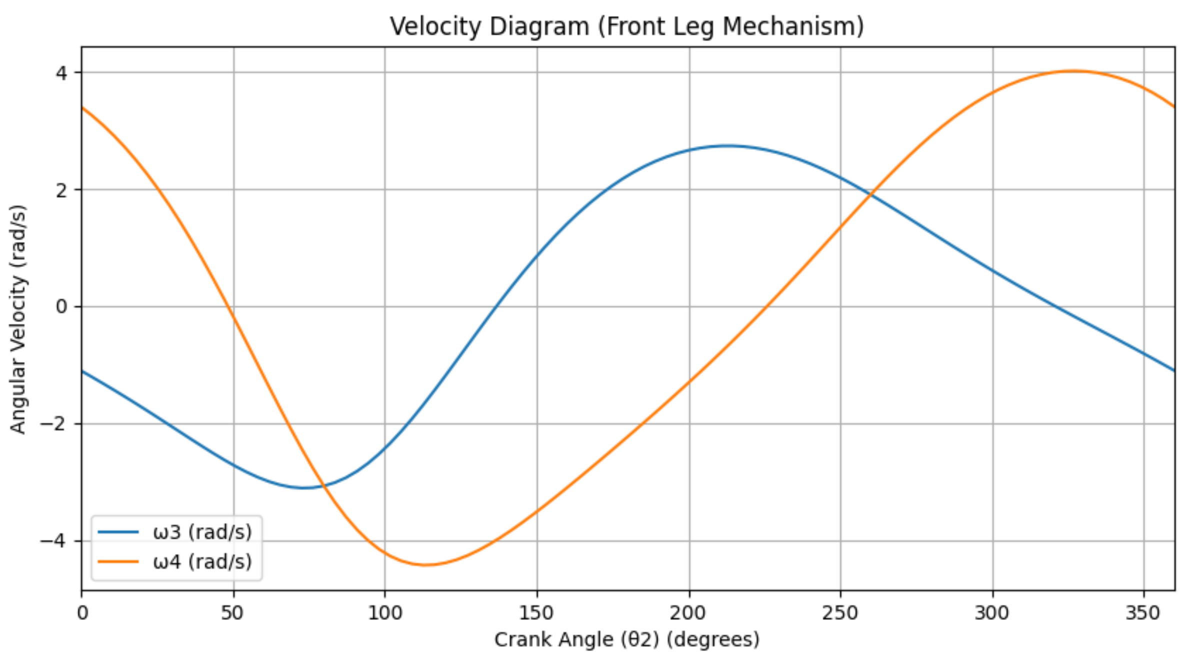
MECHANISM-1:

FRONT LEG MECHANISM

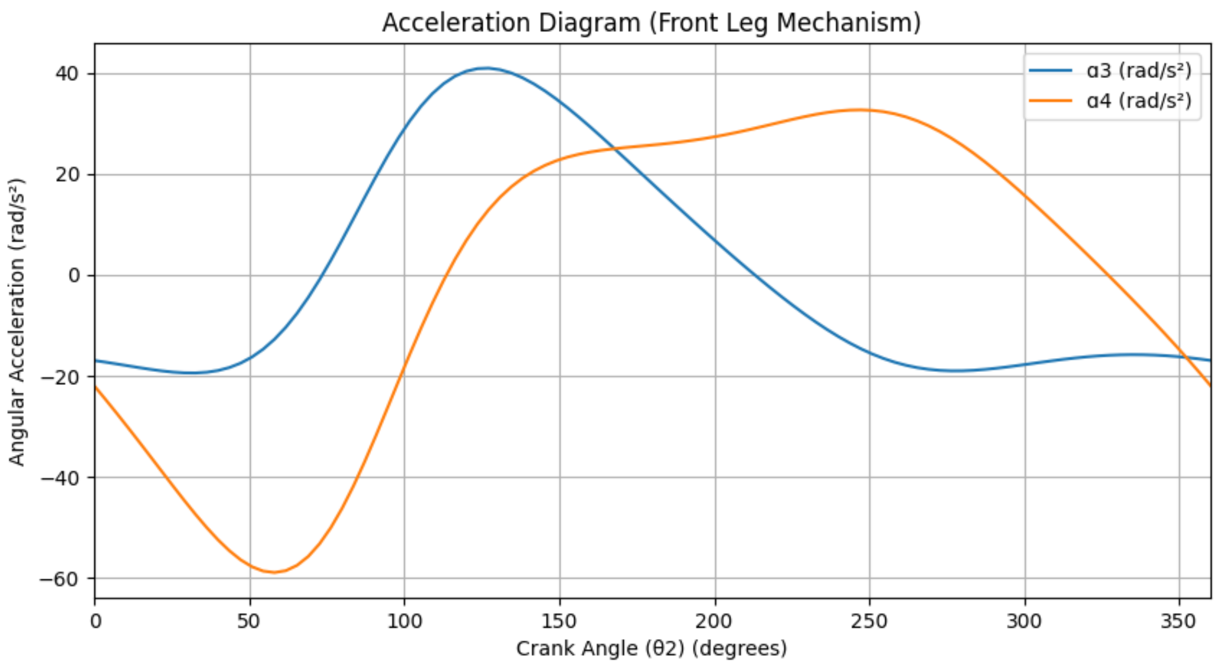
Displacement Diagram:



Velocity Diagram:



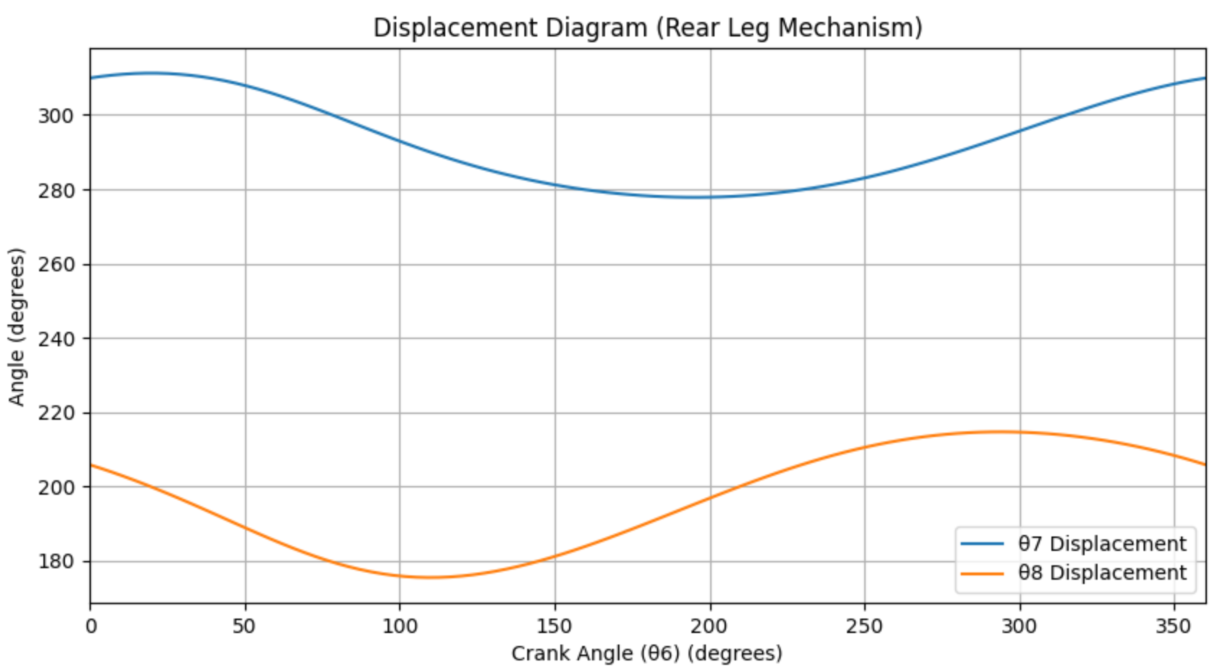
Acceleration Diagram:



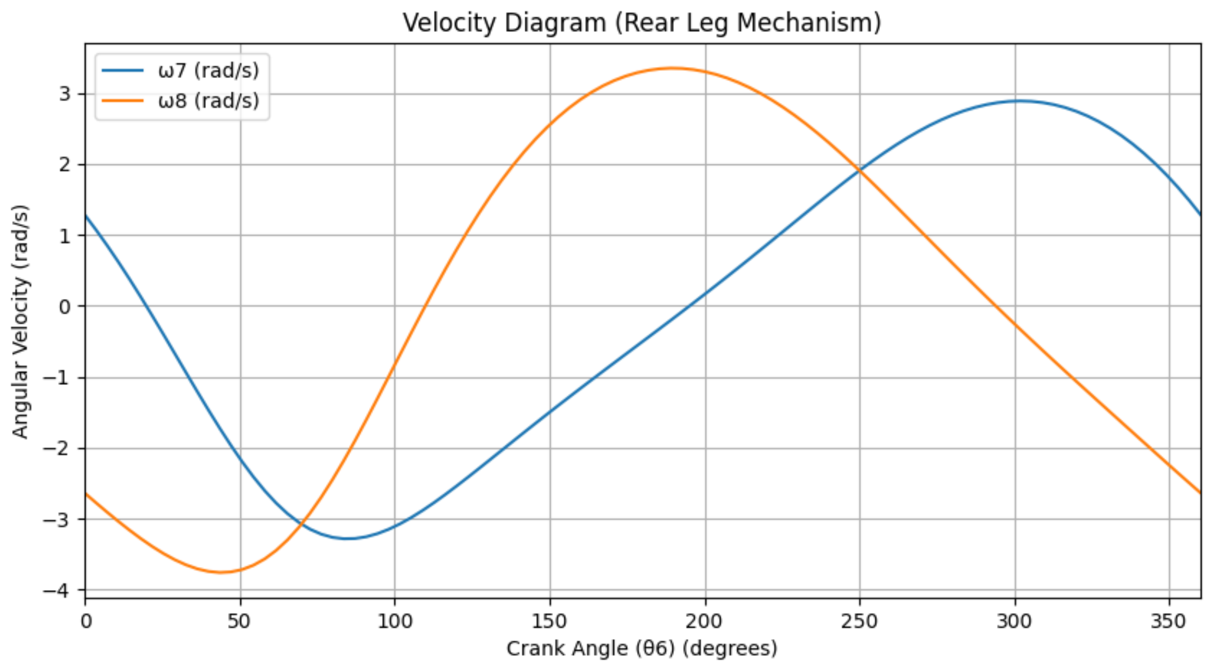
MECHANISM-2:

REAR LEG MECHANISM

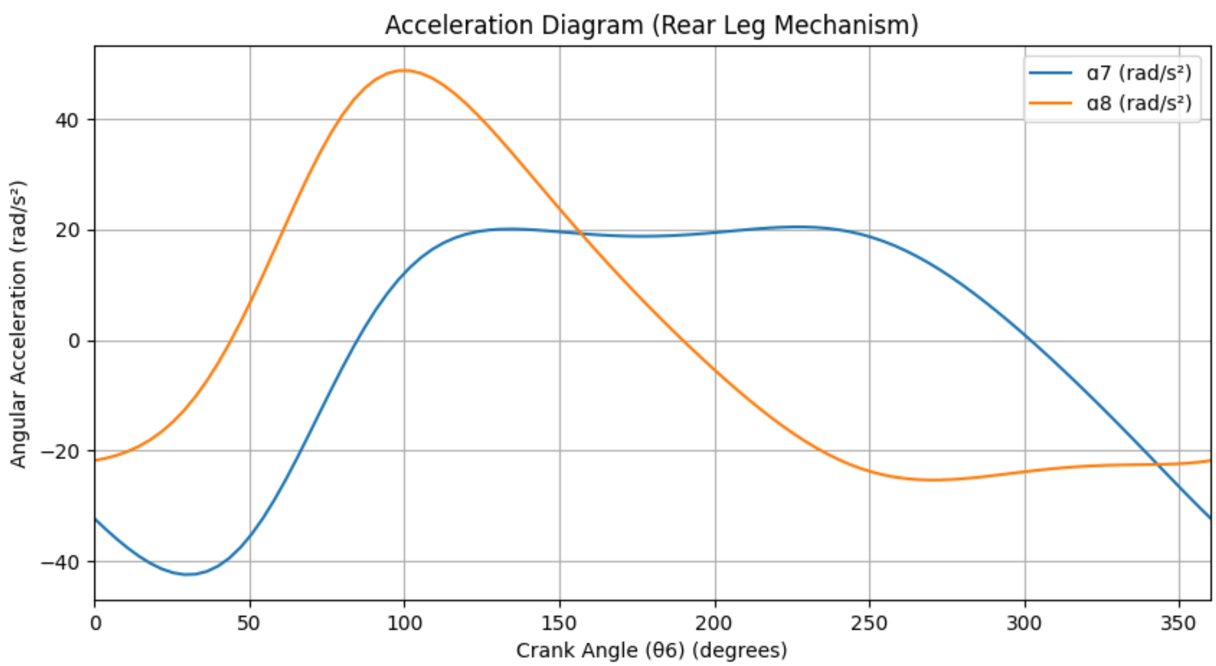
Displacement Diagram:



Velocity Diagram:

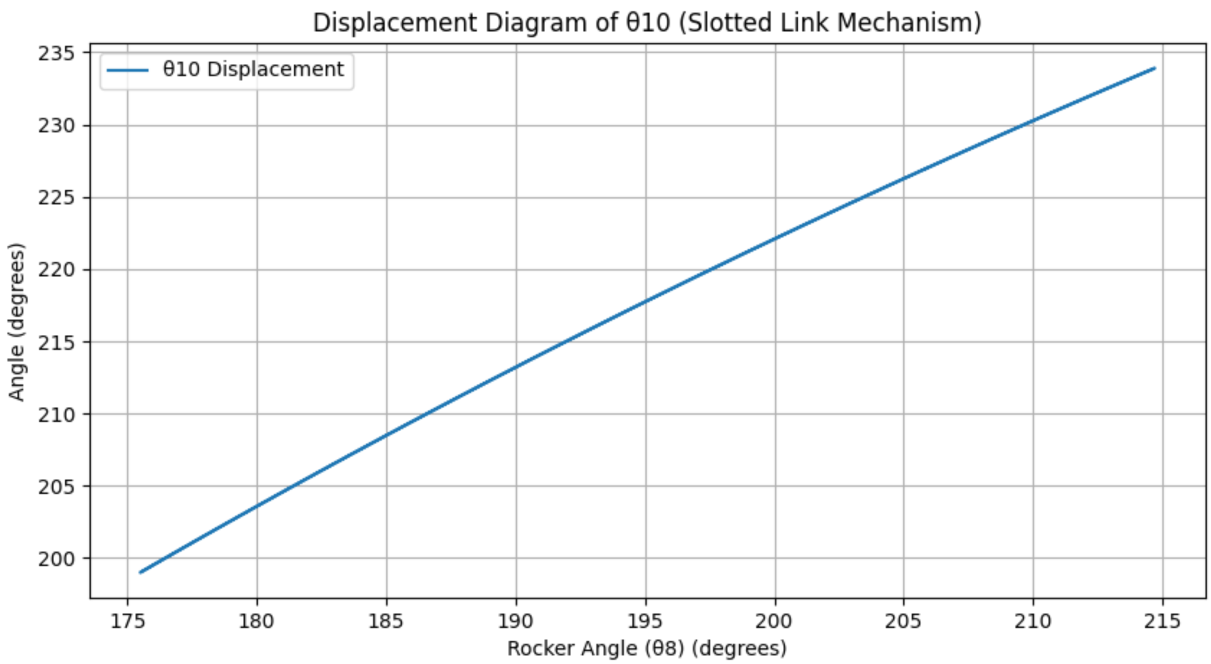


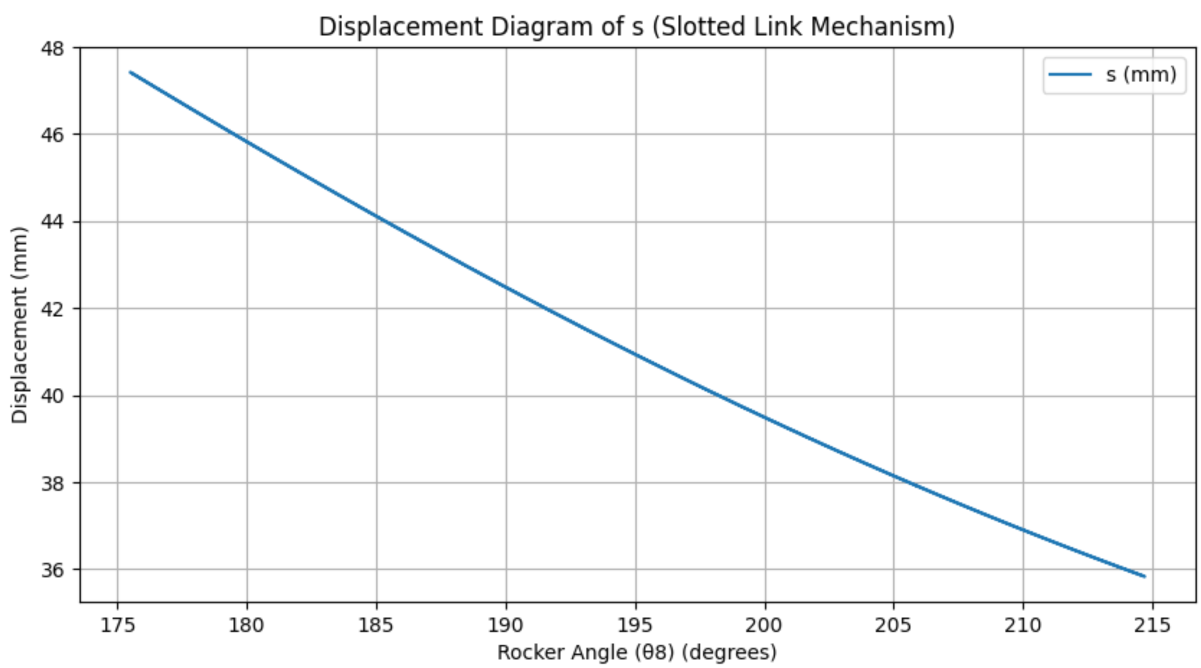
Acceleration Diagram:



SLOTTED LINK MECHANISM

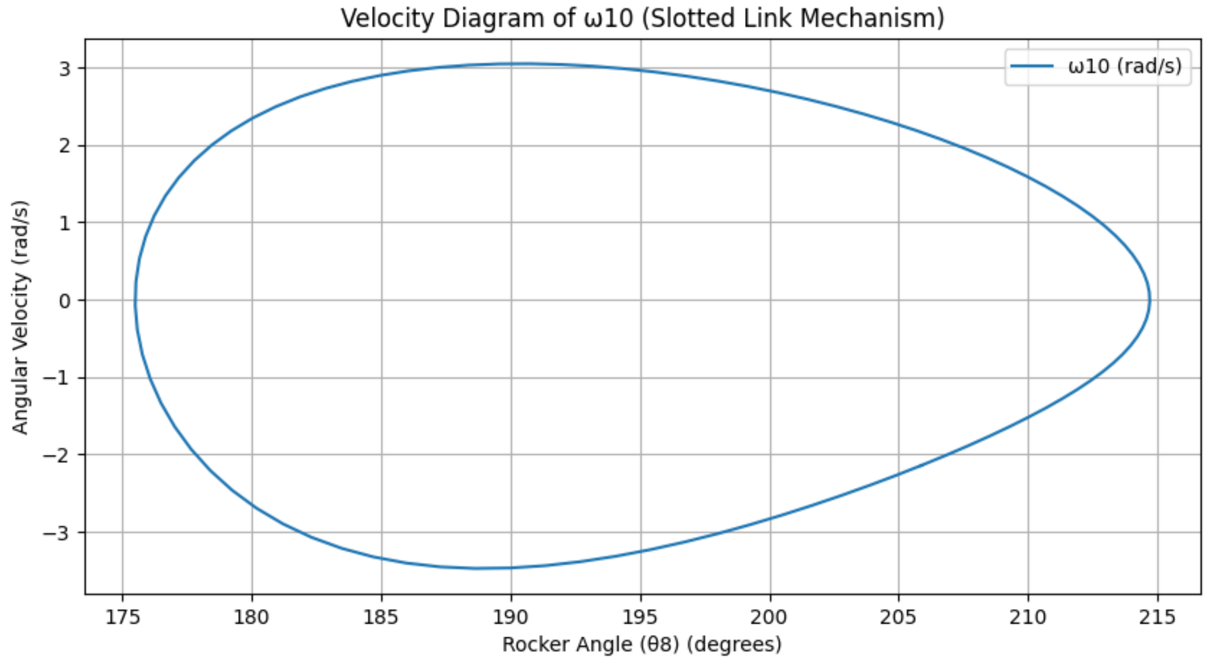
Displacement Diagrams:

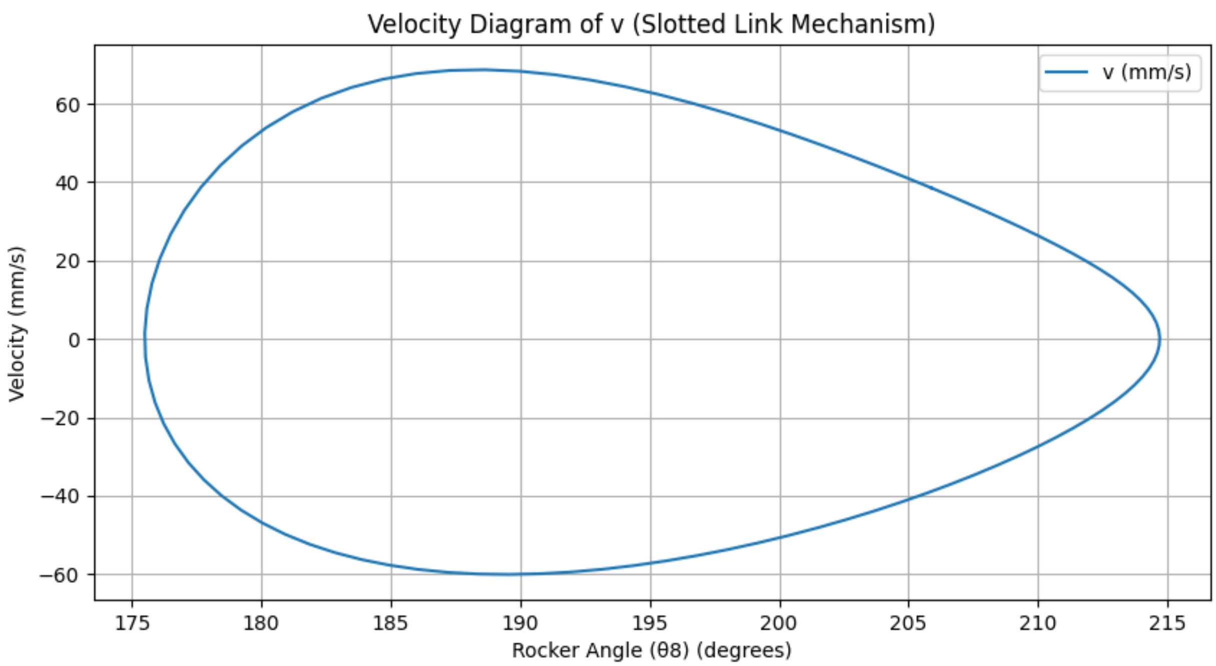




As $\theta 8$ increases, the sliding length decreases (slider inside the slot moves to its right extreme, i.e. closest to the revolute pair joining link8 and link11) whilst $\theta 10$ increases causing the tail to move up and vice-versa.

Velocity Diagrams:

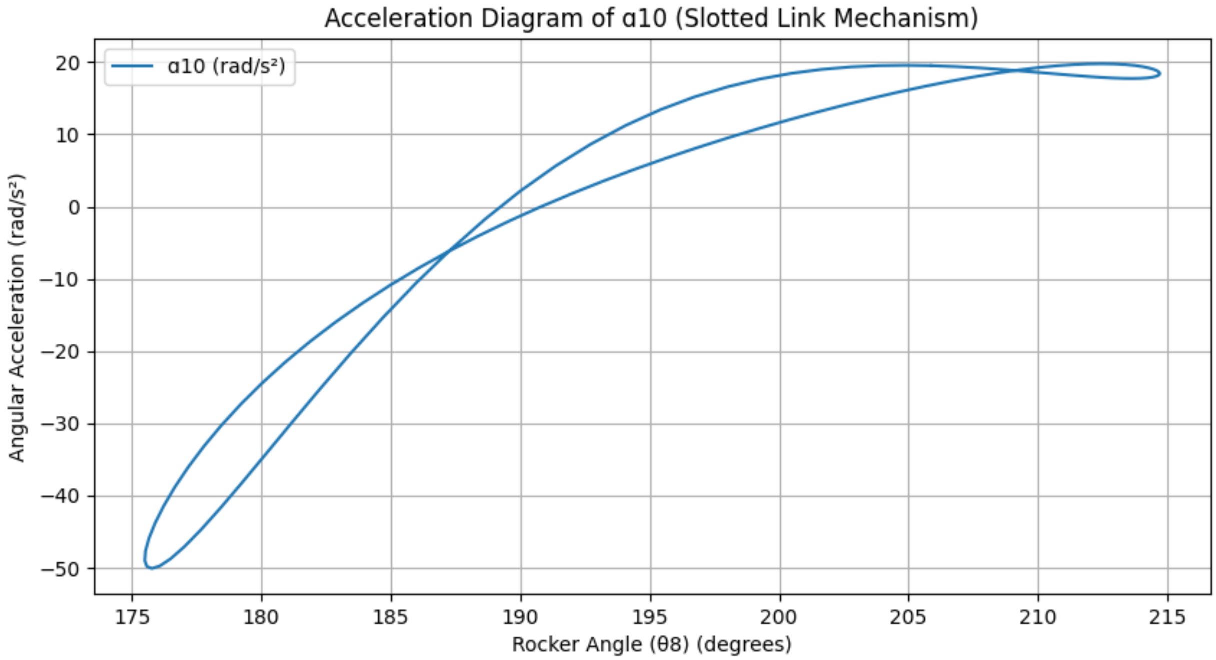


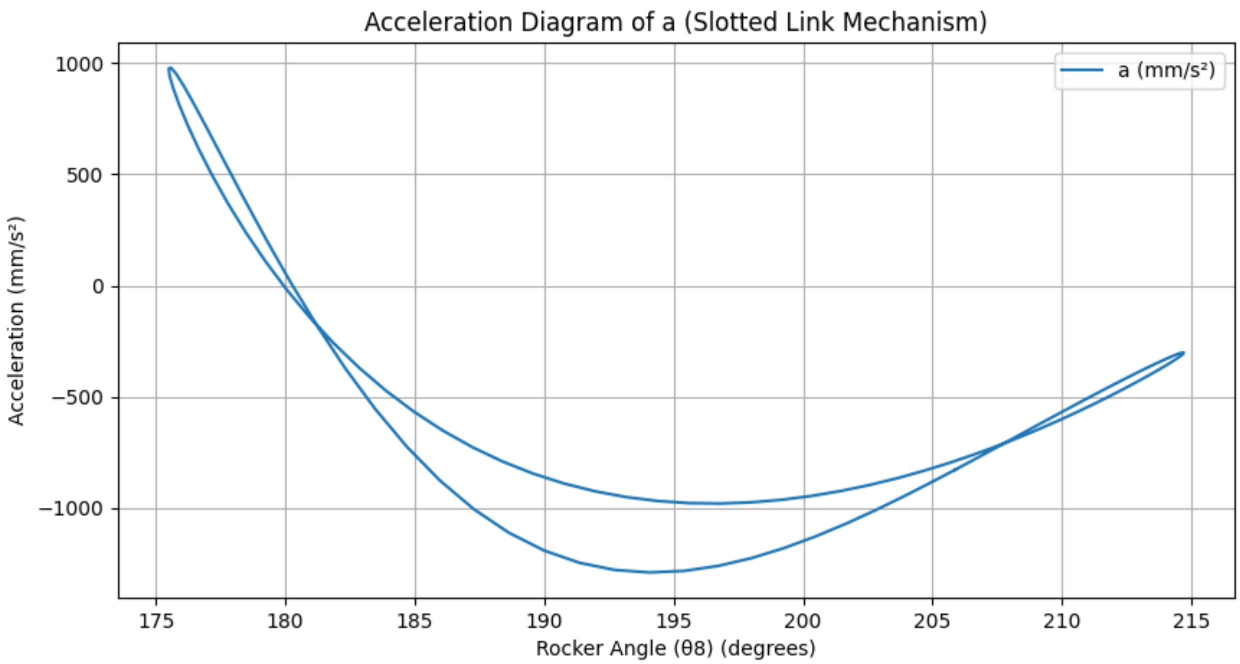


(An Interesting Observation)

Here we see a closed loop for the velocity diagrams. This happens as for one crank rotation ($\theta 6$), the rocker ($\theta 8$) reaches its extremes (lowest and highest angle) twice which is accompanied with a change in the direction of velocity (causing both positive and negative values of velocity for the same θ) with a momentary pause at the extremes leading velocity to be zero. However, the direction of the loops of $\omega 10$ and v are different as $\theta 10$ and s are inversely proportional to each other.

Acceleration Diagrams:



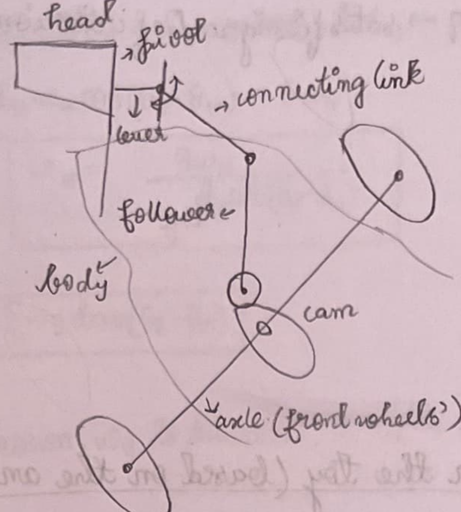


(An Interesting Observation)
Here we see a closed loop for the acceleration diagrams. This happens as for one crank rotation ($\theta 6$), the rocker ($\theta 8$) reaches its extremes (lowest and highest angle) twice which is accompanied with a change in the acceleration (magnitude/sign vise). But, note that the acceleration curves are smooth, accompanied with a change in direction at the extremes.

Practical observations seen in the toy (based on the analysis conducted):

The kinematic analysis of the pull along dog toy provides insights into how the mechanisms contribute to the toy's movement. The displacement diagrams show how the front and rear legs oscillate in response to the crank (wheel) motion, mimicking a walking motion. The velocity and acceleration diagrams highlight the variations in speed and dynamic forces acting on the legs and slotted link. Notably, the rear rocker, which incorporates the slotted link, exhibits an additional vertical motion component. This explains why, in the actual toy, the rear legs not only rotate but also slightly lift, creating a more lifelike walking effect. The nature of the slotted link's displacement suggests a smooth, periodic up-and-down motion, which can be observed in the toy's tail as the toy moves. This refined movement enhances the realism of the toy's walking simulation.

Synthesis of the additional mechanism that I'd like to add:



follower↓ head→
follower↑ head←

(detailed description of its working, in PHASE-I submission).

Type synthesis:

The mechanism is intended to achieve side-to-side tilting of the head, using a cam-follower mechanism. The cam-follower system converts rotational motion of the cam into oscillatory motion of the follower transmitted via a connecting link to the pivot-lever system, ensuring that the head undergoes a tilting motion.

Number synthesis:

Components involved:

Body = 1
Cam = 1
Follower = 1
Coupler = 1
output (lever + head) = 1

5 links

Joints:

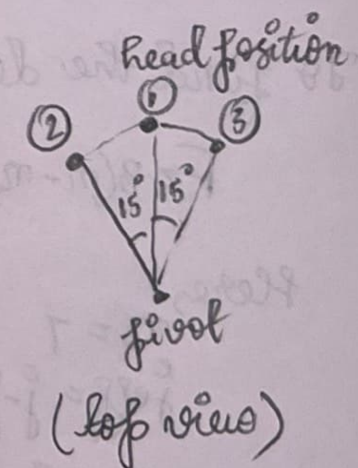
Revolute pairs (j_1) = 3
higher pairs (h) = 1

$$\text{Degrees of freedom} = 3(n-1) - 2j - h$$
$$= 3(5-1) - 2(3) - 1$$
$$= 5$$

But, effective degrees of freedom = 1 ($\because$ 4 redundant DoF).

Dimensional synthesis:

Taking 3 points in the x-z plane with a maximum tilt angle of around ±15°, we can use the Frudenstein equations for function generation to effectively find the link lengths adhering to practical scenarios.



CODES USED FOR PLOT GENERATION

MECHANISM-1:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import fsolve

#link lengths (in mm)
l1 = 85
l2 = 20
l3 = 75
l4 = 50

#angular velocity of the crank (rad/s)
omega2 = 10  #assumming omega2 to be 10, for analysis
alpha2 = 0  #constant omega2

#inclination angle of front mechanism's fixed link
beta1 = np.radians(80)

#initialize arrays for storing results
theta2_vals = np.linspace(0, 2*np.pi, 100)  #crank angles from 0 to 360 degrees
theta3_vals = []
theta4_vals = []
omega3_vals = []
omega4_vals = []
alpha3_vals = []
alpha4_vals = []

#function to solve displacement equations
def equations(vars, theta2, l2, l3, l4,l1, beta):
    theta3, theta4 = vars
    eq1 = l2 * np.cos(theta2) - l3 * np.cos(theta3) - l4 * np.cos(theta4) - l1 * np.cos(beta)
    eq2 = l2 * np.sin(theta2) - l3 * np.sin(theta3) - l4 * np.sin(theta4) - l1 * np.sin(beta)
    return [eq1, eq2]

#function to solve acceleration equations
def compute_equations(vars1, l2, l3, l4, theta2, theta3, theta4, omega2, omega3, omega4,alpha2):
    alpha3, alpha4 = vars1
    acc1 = -l4 * (np.cos(theta4) * omega4**2 + alpha4 * np.sin(theta4)) - \
           l3 * (np.cos(theta3) * omega3**2 + alpha3 * np.sin(theta3)) + \
           l2 * (np.cos(theta2) * omega2**2 + alpha2 * np.sin(theta2))

    acc2 = l4 * (-np.sin(theta4) * omega4**2 + np.cos(theta4) * alpha4) + \
           l3 * (-np.sin(theta3) * omega3**2 + np.cos(theta3) * alpha3) - \
           l2 * (-np.sin(theta2) * omega2**2 + np.cos(theta2) * alpha2)

    return [acc1, acc2]
```

```python
#loop through crank angles
for theta2 in theta2_vals:
    #displacement analysis
    initial_guess = [np.radians(190), np.radians(280)]
    theta3, theta4 = fsolve(equations, initial_guess, args=(theta2, l2, l3, l4,l1, beta1))
    theta3_vals.append(theta3)
    theta4_vals.append(theta4)

    #velocity analysis
    omega3 = (l2 * omega2 * np.sin(theta2 - theta4)) / (l3 * np.sin(theta3 - theta4))
    omega4 = (l2 * omega2 * np.sin(theta2 - theta3)) / (l4 * np.sin(theta4 - theta3))

    omega3_vals.append(omega3)
    omega4_vals.append(omega4)

    #acceleration analysis
    ini_guess_acc = [15, 20]
    alpha3, alpha4 = fsolve(compute_equations, ini_guess_acc, args=(l2, l3, l4, theta2, theta3,
theta4, omega2, omega3, omega4,alpha2))

    alpha3_vals.append(alpha3)
    alpha4_vals.append(alpha4)

#convert angles to degrees for plotting
theta2_vals_deg = np.degrees(theta2_vals)
theta3_vals_deg = np.degrees(theta3_vals)
theta4_vals_deg = np.degrees(theta4_vals)

#plot displacement diagram
plt.figure(figsize=(10, 5))
plt.plot(theta2_vals_deg, theta3_vals_deg, label='θ3 Displacement')
plt.plot(theta2_vals_deg, theta4_vals_deg, label='θ4 Displacement')
plt.xlim(0,360)
plt.xlabel("Crank Angle (θ2) (degrees)")
plt.ylabel("Angle (degrees)")
plt.title("Displacement Diagram (Front Leg Mechanism)")
plt.legend()
plt.grid()
plt.savefig('mech1_dis.png', format='png')
plt.show()

#plot velocity diagram
plt.figure(figsize=(10, 5))
plt.plot(theta2_vals_deg, omega3_vals, label='ω3 (rad/s)')
plt.plot(theta2_vals_deg, omega4_vals, label='ω4 (rad/s)')
plt.xlim(0,360)
plt.xlabel("Crank Angle (θ2) (degrees)")
plt.ylabel("Angular Velocity (rad/s)")
plt.title("Velocity Diagram (Front Leg Mechanism)")
plt.legend()
plt.grid()
plt.savefig('mech1_vel.png', format='png')
plt.show()
```

```python
#plot acceleration diagram
plt.figure(figsize=(10, 5))
plt.plot(theta2_vals_deg, alpha3_vals, label='α3 (rad/s²)')
plt.plot(theta2_vals_deg, alpha4_vals, label='α4 (rad/s²)')
plt.xlim(0,360)
plt.xlabel("Crank Angle (θ2) (degrees)")
plt.ylabel("Angular Acceleration (rad/s²)")
plt.title("Acceleration Diagram (Front Leg Mechanism)")
plt.legend()
plt.grid()
plt.savefig('mech1_acc.png', format='png')
plt.show()
```

MECHANISM-2:

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import fsolve

#link lengths (in mm)
l5 = 85
l6 = 20
l7 = 70
l8 = 60
l10 = 50
l11 = 20

#angular velocity of the crank (rad/s)
omega6 = 10  #equal to omega2, as both wheels are driven via the same drawstring
alpha6 = 0  #Constant omega2

#inclination angles
beta5 = np.radians(70)
beta11 = np.radians(90)

#initialize arrays for storing results
theta6_vals = np.linspace(0, 2*np.pi, 100)  #crank angles from 0 to 360 degrees
theta7_vals = []
theta8_vals = []
omega7_vals = []
omega8_vals = []
alpha7_vals = []
alpha8_vals = []

s_vals = []
theta10_vals = []
v_vals = []
omega10_vals = []
a_vals = []
```

```python
alpha10_vals = []

#function to solve displacement equations for Rear Leg Mechanism
def equations(vars, theta6, l6, l7, l8,l5, beta):
    theta7, theta8 = vars
    eq1 = l6 * np.cos(theta6) - l7 * np.cos(theta7) - l8 * np.cos(theta8) - l5 * np.cos(beta)
    eq2 = l6 * np.sin(theta6) - l7 * np.sin(theta7) - l8 * np.sin(theta8) - l5 * np.sin(beta)
    return [eq1, eq2]

#function to compute acceleration equations Rear Leg Mechanism
def compute_equations(vars1, l6, l7, l8, theta6, theta7, theta8, omega6, omega7, omega8,alpha6):
    alpha7, alpha8 = vars1
    acc1 = -l8 * (np.cos(theta8) * omega8**2 + alpha8 * np.sin(theta8)) - \
           l7 * (np.cos(theta7) * omega7**2 + alpha7 * np.sin(theta7)) + \
           l6 * (np.cos(theta6) * omega6**2 + alpha6 * np.sin(theta6))

    acc2 = l8 * (-np.sin(theta8) * omega8**2 + np.cos(theta8) * alpha8) + \
           l7 * (-np.sin(theta7) * omega7**2 + np.cos(theta7) * alpha7) - \
           l6 * (-np.sin(theta6) * omega6**2 + np.cos(theta6) * alpha6)

    return [acc1, acc2]

#Slotted Link Mechanism Analysis
def displacement_equations(vars, l10, l11, beta11, theta8):
    s, theta10 = vars
    dis_eq1 = l11 * np.cos(beta11) + l10 * np.cos(theta10) - s * np.cos(theta8)
    dis_eq2 = l11 * np.sin(beta11) + l10 * np.sin(theta10) - s * np.sin(theta8)
    return [dis_eq1, dis_eq2]

def velocity_equations(vars, l10, s, theta10, theta8, omega8):
    omega10, v = vars
    vel_eq1 = -l10 * np.sin(theta10) * omega10 - v * np.cos(theta8) + s * np.sin(theta8) * omega8
    vel_eq2 = l10 * np.cos(theta10) * omega10 - v * np.sin(theta8) - s * np.cos(theta8) * omega8
    return [vel_eq1, vel_eq2]

def acceleration_equations(vars, l10, s, theta10, theta8, omega10, omega8, alpha8, v):
    alpha10, a = vars
    acc_eq1 = (-l10 * (np.cos(theta10) * omega10**2 + np.sin(theta10) * alpha10)
               - (a * np.cos(theta8) - v * omega8 * np.sin(theta8) + (v * np.sin(theta8) * omega8 +
s * np.cos(theta8) * omega8**2 + s * np.sin(theta8) * alpha8)))
    acc_eq2 = (l10 * (-np.sin(theta10) * omega10**2 + np.cos(theta10) * alpha10)
               - (a * np.sin(theta8) + v * omega8 * np.cos(theta8) - (v * np.cos(theta8) * omega8 -
s * np.sin(theta8) * omega8**2 + s * np.cos(theta8) * alpha8)))
    return [acc_eq1, acc_eq2]


#loop through crank angles
for theta6 in theta6_vals:
    #displacement analysis of rear leg mechanism
    initial_guess = [np.radians(290), np.radians(200)]
    theta7, theta8 = fsolve(equations, initial_guess, args=(theta6, l6, l7, l8,l5, beta5))
    theta7_vals.append(theta7)
    theta8_vals.append(theta8)
```

```python
    #velocity analysis of rear leg mechanism
    omega7 = (l6 * omega6 * np.sin(theta6 - theta8)) / (l7 * np.sin(theta7 - theta8))
    omega8 = (l6 * omega6 * np.sin(theta6 - theta7)) / (l8 * np.sin(theta8 - theta7))

    omega7_vals.append(omega7)
    omega8_vals.append(omega8)

    #acceleration analysis of rear leg mechanism
    ini_guess_acc = [15, 20]
    alpha7, alpha8 = fsolve(compute_equations, ini_guess_acc, args=(l6, l7, l8, theta6, theta7,
theta8, omega6, omega7, omega8,alpha6))

    alpha7_vals.append(alpha7)
    alpha8_vals.append(alpha8)

    #Slotted link analysis
    initial_guess_slider = [35, np.radians(200)]
    s, theta10 = fsolve(displacement_equations, initial_guess_slider, args=(l10, l11, beta11,
theta8))
    s_vals.append(s)
    theta10_vals.append(theta10)

    ini_guess_vel_slider = [10, 15]
    omega10, v = fsolve(velocity_equations, ini_guess_vel_slider, args=(l10, s, theta10, theta8,
omega8))
    v_vals.append(v)
    omega10_vals.append(omega10)

    ini_guess_acc_slider = [10, 100]
    alpha10, a = fsolve(acceleration_equations, ini_guess_acc_slider, args=(l10, s, theta10,
theta8, omega10, omega8, alpha8, v))
    alpha10_vals.append(alpha10)
    a_vals.append(a)


#convert angles to degrees for plotting
theta6_vals_deg = np.degrees(theta6_vals)
theta7_vals_deg = np.degrees(theta7_vals)
theta8_vals_deg = np.degrees(theta8_vals)
theta10_vals_deg = np.degrees(theta10_vals)

#plot displacement diagram of rear leg mechanism
plt.figure(figsize=(10, 5))
plt.plot(theta6_vals_deg, theta7_vals_deg, label='θ7 Displacement')
plt.plot(theta6_vals_deg, theta8_vals_deg, label='θ8 Displacement')
plt.xlim(0,360)
plt.xlabel("Crank Angle (θ6) (degrees)")
plt.ylabel("Angle (degrees)")
plt.title("Displacement Diagram (Rear Leg Mechanism)")
plt.legend()
plt.grid()
plt.savefig('mech2_dis.png', format='png')
```

```
plt.show()

#plot velocity diagram of rear leg mechanism
plt.figure(figsize=(10, 5))
plt.plot(theta6_vals_deg, omega7_vals, label='ω7 (rad/s)')
plt.plot(theta6_vals_deg, omega8_vals, label='ω8 (rad/s)')
plt.xlim(0,360)
plt.xlabel("Crank Angle (θ6) (degrees)")
plt.ylabel("Angular Velocity (rad/s)")
plt.title("Velocity Diagram (Rear Leg Mechanism)")
plt.legend()
plt.grid()
plt.savefig('mech2_vel.png', format='png')
plt.show()

#plot acceleration diagram of rear leg mechanism
plt.figure(figsize=(10, 5))
plt.plot(theta6_vals_deg, alpha7_vals, label='α7 (rad/s²)')
plt.plot(theta6_vals_deg, alpha8_vals, label='α8 (rad/s²)')
plt.xlim(0,360)
plt.xlabel("Crank Angle (θ6) (degrees)")
plt.ylabel("Angular Acceleration (rad/s²)")
plt.title("Acceleration Diagram (Rear Leg Mechanism)")
plt.legend()
plt.grid()
plt.savefig('mech2_acc.png', format='png')
plt.show()


# For slotted link mechanism
#plot displacement diagram of theta10
plt.figure(figsize=(10, 5))
plt.plot(theta8_vals_deg, theta10_vals_deg, label='θ10 Displacement')
plt.xlabel("Rocker Angle (θ8) (degrees)")
plt.ylabel("Angle (degrees)")
plt.title("Displacement Diagram of θ10 (Slotted Link Mechanism)")
plt.legend()
plt.grid()
plt.savefig('mech2_1_dis_1.png', format='png')
plt.show()

#plot displacement diagram of s
plt.figure(figsize=(10, 5))
plt.plot(theta8_vals_deg, s_vals, label='s (mm)')
plt.xlabel("Rocker Angle (θ8) (degrees)")
plt.ylabel("Displacement (mm)")
plt.title("Displacement Diagram of s (Slotted Link Mechanism)")
plt.legend()
plt.grid()
plt.savefig('mech2_1_dis_2.png', format='png')
plt.show()

#plot velocity diagram of ω10
```

```python
plt.figure(figsize=(10, 5))
plt.plot(theta8_vals_deg, omega10_vals, label='ω10 (rad/s)')
plt.xlabel("Rocker Angle (θ8) (degrees)")
plt.ylabel("Angular Velocity (rad/s)")
plt.title("Velocity Diagram of ω10 (Slotted Link Mechanism)")
plt.legend()
plt.grid()
plt.savefig('mech2_1_vel_1.png', format='png')
plt.show()

#plot velocity diagram of v
plt.figure(figsize=(10, 5))
plt.plot(theta8_vals_deg, v_vals, label='v (mm/s)')
plt.xlabel("Rocker Angle (θ8) (degrees)")
plt.ylabel("Velocity (mm/s)")
plt.title("Velocity Diagram of v (Slotted Link Mechanism)")
plt.legend()
plt.grid()
plt.savefig('mech2_1_vel_2.png', format='png')
plt.show()

#plot acceleration diagram of α10
plt.figure(figsize=(10, 5))
plt.plot(theta8_vals_deg, alpha10_vals, label='α10 (rad/s²)')
plt.xlabel("Rocker Angle (θ8) (degrees)")
plt.ylabel("Angular Acceleration (rad/s²)")
plt.title("Acceleration Diagram of α10 (Slotted Link Mechanism)")
plt.legend()
plt.grid()
plt.savefig('mech2_1_acc_1.png', format='png')
plt.show()

#plot acceleration diagram of a
plt.figure(figsize=(10, 5))
plt.plot(theta8_vals_deg, a_vals, label='a (mm/s²)')
plt.xlabel("Rocker Angle (θ8) (degrees)")
plt.ylabel("Acceleration (mm/s²)")
plt.title("Acceleration Diagram of a (Slotted Link Mechanism)")
plt.legend()
plt.grid()
plt.savefig('mech2_1_acc_2.png', format='png')
plt.show()
```