Week 5 and 6 Crank-Slider Kinematics

Written by: Caiden Bonney

Created: 2/19/25

Modified: 2/26/25

Background:

In this lab, we are analyzing the kinematics of a crank-slider mechanism. We will use the vector loop method to determine the position, velocity, and acceleration of each link symbolically and numerically. This includes finding angular velocities, slider motion, and center of gravity kinematics. Additionally, we will simulate and animate the mechanism using MATLAB and Simulink.

Required Files:

- ME326_05_Activity4_BonneyCaiden_MatLabFile
- ME326_05_Activity4_BonneyCaiden_SimulinkFile

Table of Contents

Introduction

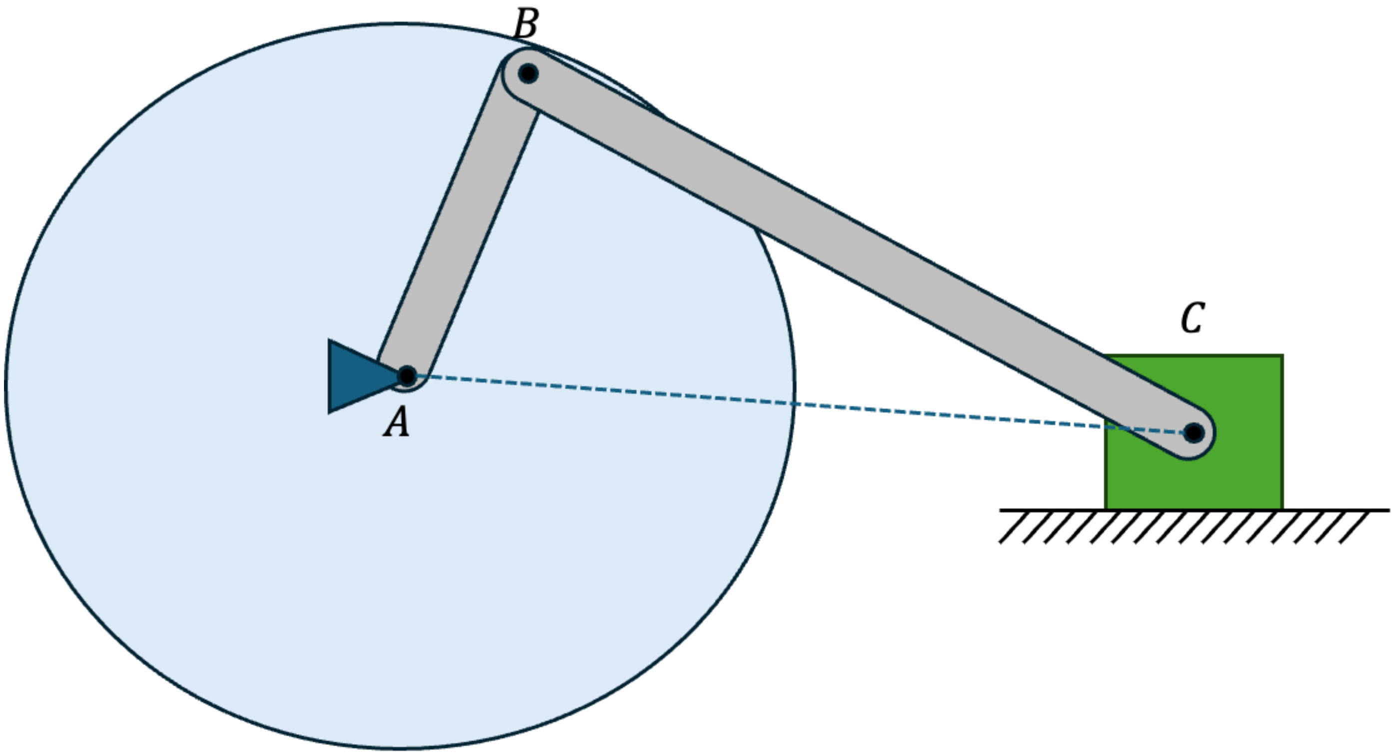
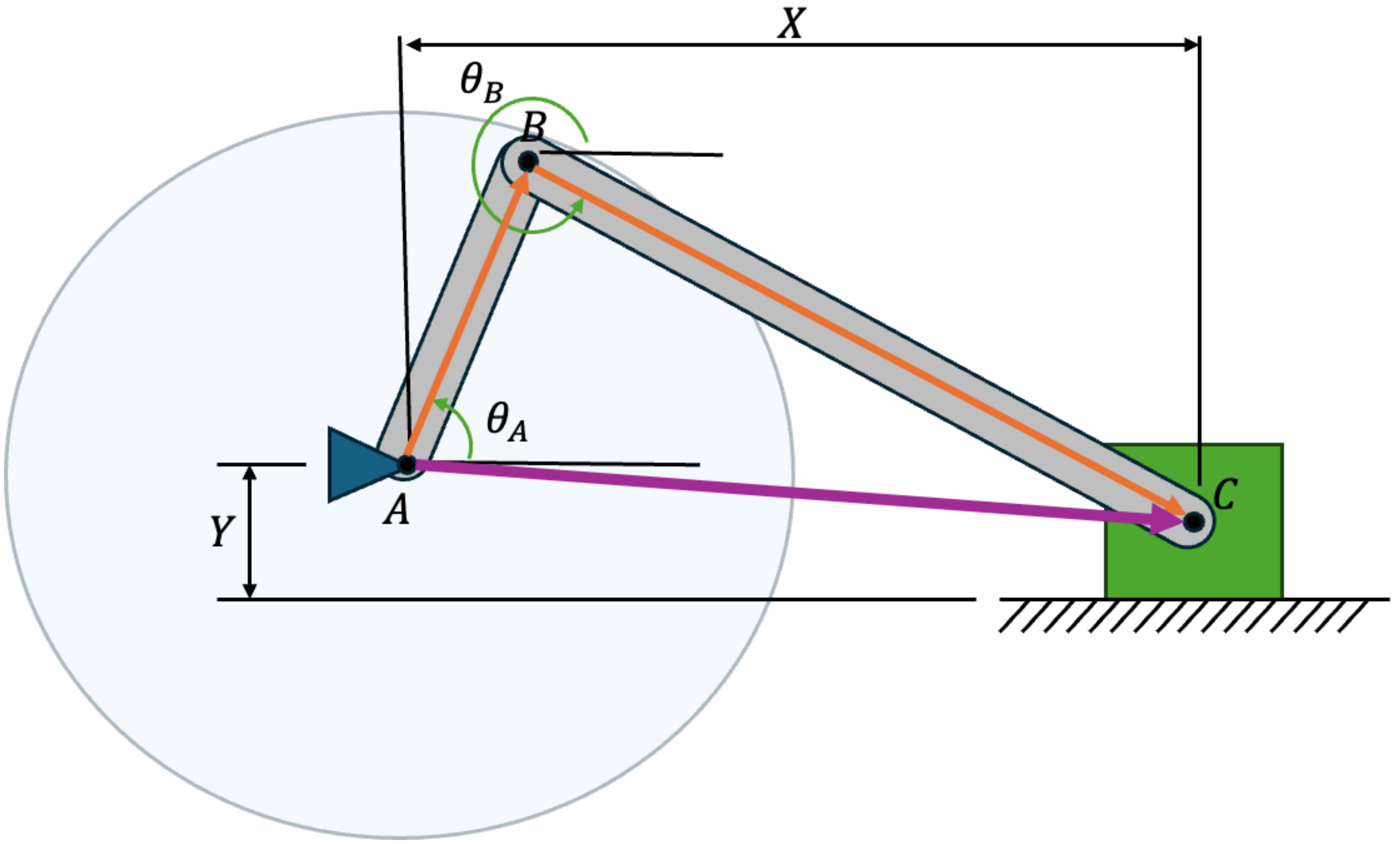


Figure Depicts Model of Mechanism That Converts Rotary Motion Into Linear Motion

The model consists of 4 links:

1. The input link (AB), which connects to the input of the mechanism (the motor)
2. The coupler link (BC)
3. The slider (C)
4. The fixed ground, which can be thought of as link AC.

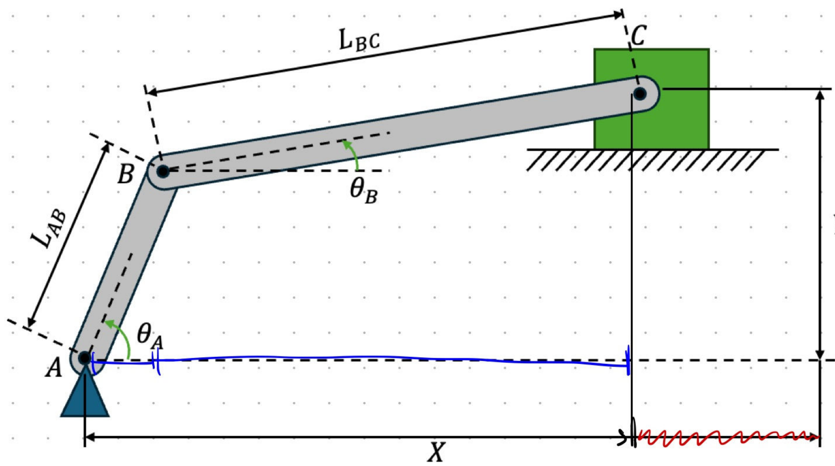
Due to the joints at A, B, and C the distance vector from A to B plus the distance vector from B to C must be equivalent to the distance vector from A to C. Defining the angles of bar AB and bar BC from the positive x direction (to the right) and noting that these angles change over means that the length of AC changes over time as well. The change in the length of AC in the horizontal direction is equivalent to the motion of the slider C. A schematic displaying these angles is shown below.



In this case, X represents the horizontal length of AC and Y represents the vertical length of AC which remains constant throughout the motion.

Applying previously learned concepts of relative velocity and acceleration, along with kinematic constraints such as slider C only sliding horizontally, as it is fixed to the fixed ground, the motion of point C in relation to theta A can be modeled. This process is shown in Part A and animated through the use of Simulink in Part B.

Part A Task 1: Symbolic Kinematic Analysis



Task 1: Symbolic Kinematic Analysis

Suppose that, for the mechanism in Figure 4, the following is always known:

- The link lengths L_{AB}, L_{BC}
- The slider vertical offset Y
- The input angle θ_A at any time (meaning we also know ω_A, α_A)

Using the Vector Loop method, symbolically find:

- The angle θ_B
- The horizontal offset X
- The angular velocity of link BC (ω_B) and the slider velocity (v_C) -- You may find these values in terms of θ_B and X
- The angular acceleration of link BC (α_B) and the slider acceleration (a_C) -- You may find these values in terms of θ_B, X, ω_B, and v_C

Show your computations in detail. First, put the velocity and acceleration equations in matrix form $A\chi_d = b$. Then, you should isolate expressions for:

- ω_B
- v_C
- α_B
- a_C

Position

$$\vec{r}_{AB}(L_{AB}, \theta_A) + \vec{r}_{BC}(L_{BC}, \theta_B) = \vec{r}_{AC}(X, Y)$$

$$\hat{i} \quad L_{AB}\cos\theta_A + L_{BC}\cos\theta_B = X \quad (1)$$
$$\hat{j} \quad L_{AB}\sin\theta_A + L_{BC}\sin\theta_B = Y \quad (2)$$

$$\theta_B = \sin^{-1}\left(\frac{Y - L_{AB}\sin\theta_A}{L_{BC}}\right)$$

$$X = L_{AB}\cos\theta_A + L_{BC}\cos\left(\sin^{-1}\left(\frac{Y - L_{AB}\sin\theta_A}{L_{BC}}\right)\right)$$

Velocity

$$\vec{V}_{AB}(L_{AB}, \omega_A) + \vec{V}_{BC}(L_{BC}, \omega_B) = \vec{V}_{AC}(\dot{X}, \cancel{\dot{Y}}^{\,0})$$

$$\hat{i}: -L_{AB}\sin\theta_A\,\omega_A - L_{BC}\sin\theta_B\,\omega_B = \dot{X} \quad (1)$$
$$\hat{j}: L_{AB}\cos\theta_A\,\omega_A + L_{BC}\cos\theta_B\,\omega_B = 0 \quad (2)$$

$\hat{i}: \; L_{BC}\sin\theta_B\,\omega_B + \dot{X} = -L_{AB}\sin\theta_A\,\omega_A \qquad (1)$

$\hat{j}: \; L_{BC}\cos\theta_B\,\omega_B = -L_{AB}\cos\theta_A\,\omega_A \qquad (2)$

$$\begin{bmatrix} L_{BC}\sin\theta_B & 1 \\ L_{BC}\cos\theta_B & 0 \end{bmatrix}\begin{bmatrix} \omega_B \\ \dot{X} \end{bmatrix} = \begin{bmatrix} -L_{AB}\sin\theta_A\,\omega_A \\ -L_{AB}\cos\theta_A\,\omega_A \end{bmatrix}$$

$$\begin{bmatrix} \omega_B \\ \dot{X} \end{bmatrix} = \begin{bmatrix} L_{BC}\sin\theta_B & 1 \\ L_{BC}\cos\theta_B & 0 \end{bmatrix}^{-1}\begin{bmatrix} -L_{AB}\sin\theta_A\,\omega_A \\ -L_{AB}\cos\theta_A\,\omega_A \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$\begin{bmatrix} L_{BC}\sin\theta_B & 1 \\ L_{BC}\cos\theta_B & 0 \end{bmatrix}^{-1} = \frac{1}{\cancelto{0}{(L_{BC}\sin\theta_B)(0)} - (1)(L_{BC}\cos\theta_B)}\begin{bmatrix} 0 & -1 \\ -L_{BC}\cos\theta_B & L_{BC}\sin\theta_B \end{bmatrix}$$

$$\begin{bmatrix} \omega_B \\ \dot{X} \end{bmatrix} = \frac{1}{-L_{BC}\cos\theta_B}\begin{bmatrix} 0 & -1 \\ -L_{BC}\cos\theta_B & L_{BC}\sin\theta_B \end{bmatrix}\begin{bmatrix} -L_{AB}\sin\theta_A\,\omega_A \\ -L_{AB}\cos\theta_A\,\omega_A \end{bmatrix}$$

$$\begin{bmatrix} \omega_B \\ \dot{X} \end{bmatrix} = \frac{1}{-L_{BC}\cos\theta_B}\begin{bmatrix} L_{AB}\cos\theta_A\,\omega_A \\ (-L_{BC}\cos\theta_B)(-L_{AB}\sin\theta_A\,\omega_A) + (L_{BC}\sin\theta_B)(-L_{AB}\cos\theta_A\,\omega_A) \end{bmatrix}$$

$$\begin{bmatrix} \omega_B \\ \dot{X} \end{bmatrix} = \begin{bmatrix} \dfrac{L_{AB}\cos\theta_A\,\omega_A}{-L_{BC}\cos\theta_B} \\ \dfrac{(-L_{BC}\cos\theta_B)(-L_{AB}\sin\theta_A\,\omega_A) + (L_{BC}\sin\theta_B)(-L_{AB}\cos\theta_A\,\omega_A)}{-L_{BC}\cos\theta_B} \end{bmatrix}$$

$$\begin{bmatrix} \omega_B \\ \dot{X} \end{bmatrix} = \begin{bmatrix} \dfrac{L_{AB}\cos\theta_A\,\omega_A}{-L_{BC}\cos\theta_B} \\ \dfrac{(\cancel{-L_{BC}\cos\theta_B})(-L_{AB}\sin\theta_A\,\omega_A)}{\cancel{-L_{BC}\cos\theta_B}} + \dfrac{(\cancel{L_{BC}}\sin\theta_B)(-L_{AB}\cos\theta_A\,\omega_A)}{\cancel{-L_{BC}}\cos\theta_B} \end{bmatrix}$$

$$\begin{bmatrix} \omega_B \\ \dot{X} \end{bmatrix} = \begin{bmatrix} \dfrac{L_{AB}\cos\theta_A\,\omega_A}{-L_{BC}\cos\theta_B} \\ -L_{AB}\sin\theta_A\,\omega_A + L_{AB}\cos\theta_A\,\omega_A\tan\theta_B \end{bmatrix} \qquad \frac{\sin\theta_B}{\cos\theta_B} = \tan\theta_B$$

✓ $\omega_B = \dfrac{L_{AB}\cos\theta_A\,\omega_A}{-L_{BC}\cos\theta_B}$

✓ $\dot{X} = -L_{AB}\sin\theta_A\,\omega_A + L_{AB}\cos\theta_A\,\omega_A\tan\theta_B$

$\dot{X} = V_C$
(on flat surface)

Acceleration

$$\vec{a}_{AB}(L_{AB}, \alpha_A) + \vec{a}_{BC}(L_{BC}, \alpha_B) = \vec{a}_{AC}(\ddot{X}, \cancel{\ddot{Y}}^{0})$$

$$\hat{i}: -L_{AB}(\cos\theta_A \omega_A^2 + \sin\theta_A \alpha_A) - L_{BC}(\cos\theta_B \omega_B^2 + \sin\theta_B \alpha_B) = \ddot{X} \quad (1)$$

$$\hat{j}: L_{AB}(-\sin\theta_A \omega_A^2 + \cos\theta_A \alpha_A) + L_{BC}(-\sin\theta_B \omega_B^2 + \cos\theta_B \alpha_B) = 0 \quad (2)$$

$$\hat{i}: L_{BC}\sin\theta_B \alpha_B + \ddot{X} = -L_{AB}\cos\theta_A \omega_A^2 - L_{AB}\sin\theta_A \alpha_A - L_{BC}\cos\theta_B \omega_B^2 \quad (1)$$

$$\hat{j}: L_{BC}\cos\theta_B \alpha_B = L_{AB}\sin\theta_A \omega_A^2 - L_{AB}\cos\theta_A \alpha_A + L_{BC}\sin\theta_B \omega_B^2 \quad (2)$$

$$\begin{bmatrix} L_{BC}\sin\theta_B & 1 \\ L_{BC}\cos\theta_B & 0 \end{bmatrix}\begin{bmatrix} \alpha_B \\ \ddot{X} \end{bmatrix} = \begin{bmatrix} -L_{AB}\cos\theta_A \omega_A^2 - L_{AB}\sin\theta_A \alpha_A - L_{BC}\cos\theta_B \omega_B^2 \\ L_{AB}\sin\theta_A \omega_A^2 - L_{AB}\cos\theta_A \alpha_A + L_{BC}\sin\theta_B \omega_B^2 \end{bmatrix}$$

$$\begin{bmatrix} \alpha_B \\ \ddot{X} \end{bmatrix} = \begin{bmatrix} L_{BC}\sin\theta_B & 1 \\ L_{BC}\cos\theta_B & 0 \end{bmatrix}^{-1}\begin{bmatrix} -L_{AB}\cos\theta_A \omega_A^2 - L_{AB}\sin\theta_A \alpha_A - L_{BC}\cos\theta_B \omega_B^2 \\ L_{AB}\sin\theta_A \omega_A^2 - L_{AB}\cos\theta_A \alpha_A + L_{BC}\sin\theta_B \omega_B^2 \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$\begin{bmatrix} L_{BC}\sin\theta_B & 1 \\ L_{BC}\cos\theta_B & 0 \end{bmatrix}^{-1} = \frac{1}{-L_{BC}\cos\theta_B}\begin{bmatrix} 0 & -1 \\ -L_{BC}\cos\theta_B & L_{BC}\sin\theta_B \end{bmatrix}$$

$$\begin{bmatrix} \alpha_B \\ \ddot{X} \end{bmatrix} = \frac{1}{-L_{BC}\cos\theta_B}\begin{bmatrix} 0 & -1 \\ -L_{BC}\cos\theta_B & L_{BC}\sin\theta_B \end{bmatrix}\begin{bmatrix} -L_{AB}\cos\theta_A \omega_A^2 - L_{AB}\sin\theta_A \alpha_A - L_{BC}\cos\theta_B \omega_B^2 \\ L_{AB}\sin\theta_A \omega_A^2 - L_{AB}\cos\theta_A \alpha_A + L_{BC}\sin\theta_B \omega_B^2 \end{bmatrix}$$

$$\begin{bmatrix} \alpha_B \\ \ddot{X} \end{bmatrix} = \frac{1}{-L_{BC}\cos\theta_B}\begin{bmatrix} -(L_{AB}\sin\theta_A \omega_A^2 - L_{AB}\cos\theta_A \alpha_A + L_{BC}\sin\theta_B \omega_B^2) \\ (-L_{BC}\cos\theta_B)(-L_{AB}\cos\theta_A \omega_A^2 - L_{AB}\sin\theta_A \alpha_A - L_{BC}\cos\theta_B \omega_B^2) + (L_{BC}\sin\theta_B)(L_{AB}\sin\theta_A \omega_A^2 - L_{AB}\cos\theta_A \alpha_A + L_{BC}\sin\theta_B \omega_B^2) \end{bmatrix}$$

$$\begin{bmatrix} \alpha_B \\ \ddot{X} \end{bmatrix} = \begin{bmatrix} \dfrac{L_{AB}\sin\theta_A \omega_A^2 - L_{AB}\cos\theta_A \alpha_A + L_{BC}\sin\theta_B \omega_B^2}{L_{BC}\cos\theta_B} \\ (-L_{AB}\cos\theta_A \omega_A^2 - L_{AB}\sin\theta_A \alpha_A - L_{BC}\cos\theta_B \omega_B^2) - (\tan\theta_B)(L_{AB}\sin\theta_A \omega_A^2 - L_{AB}\cos\theta_A \alpha_A + L_{BC}\sin\theta_B \omega_B^2) \end{bmatrix}$$

$$\alpha_B = \frac{L_{AB}\sin\theta_A \omega_A^2 - L_{AB}\cos\theta_A \alpha_A + L_{BC}\sin\theta_B \omega_B^2}{L_{BC}\cos\theta_B}$$

$$\ddot{X} = (-L_{AB}\cos\theta_A \omega_A^2 - L_{AB}\sin\theta_A \alpha_A - L_{BC}\cos\theta_B \omega_B^2) - (\tan\theta_B)(L_{AB}\sin\theta_A \omega_A^2 - L_{AB}\cos\theta_A \alpha_A + L_{BC}\sin\theta_B \omega_B^2)$$

Part A Task 2: Symbolic Kinematic Analysis -- Center of Gravity velocity and acceleration.

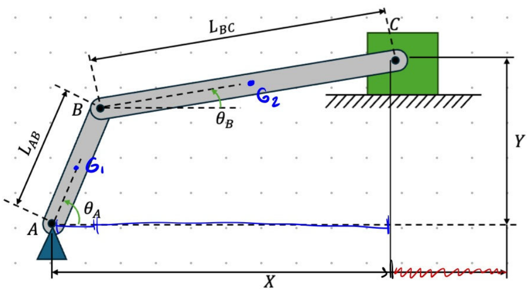
Task 2: Symbolic Kinematic Analysis -- Center of Gravity velocity and acceleration.

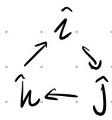
Following the results of the previous task, find the following (symbolically) using the relative velocity method (do not put in matrix form):

- The velocity of the center of gravity of link AB ($\vec{v}_{G1}$) and The velocity of the center of gravity of link BC ($\vec{v}_{G2}$).
- The acceleration of the center of gravity of link AB ($\vec{a}_{G1}$) and The acceleration of the center of gravity of link BC ($\vec{a}_{G2}$).

Assume that the center of gravity of each link is located exactly at the link's geometric center.

Show your computations in detail.





$$\vec{V}_{G1} = \cancel{\vec{V}_A}^{\,0} + \vec{\omega}_A \times \vec{r}_{G1/A}$$

$$\vec{\omega}_A = \omega_A \hat{k}$$
$$\vec{r}_{G1/A} = \frac{L_{AB}}{2}(\cos\theta_A \hat{i} + \sin\theta_A \hat{j})$$

$$\vec{V}_{G1} = \omega_A \hat{k} \times \left(\frac{L_{AB}}{2}\cos\theta_A \hat{i} + \frac{L_{AB}}{2}\sin\theta_A \hat{j}\right)$$
$$= \omega_A \frac{L_{AB}}{2}\cos\theta_A \hat{j} - \omega_A \frac{L_{AB}}{2}\sin\theta_A \hat{i}$$

$$\vec{V}_{G1} = -\omega_A \frac{L_{AB}}{2}\sin\theta_A \hat{i} + \omega_A \frac{L_{AB}}{2}\cos\theta_A \hat{j}$$

$$\vec{V}_{G2} = \vec{V}_B + \vec{\omega}_B \times \vec{r}_{G2/B}$$

$$\vec{V}_B = \vec{\omega}_A \times \vec{r}_{B/A}$$
$$= \omega_A \hat{k} \times (L_{AB}\cos\theta_A \hat{i} + L_{AB}\sin\theta_B \hat{j})$$
$$= \omega_A L_{AB}\cos\theta_A \hat{j} - \omega_A L_{AB}\sin\theta_B \hat{i}$$

$$\vec{\omega}_B = \omega_B \hat{k}$$
$$\vec{r}_{G2/B} = \frac{L_{BC}}{2}(\cos\theta_B \hat{i} + \sin\theta_B \hat{j})$$

$$\vec{V}_{G2} = \omega_A L_{AB}\cos\theta_A \hat{j} - \omega_A L_{AB}\sin\theta_B \hat{i} + \omega_B \hat{k} \times \left(\frac{L_{BC}}{2}\cos\theta_B \hat{i} + \frac{L_{BC}}{2}\sin\theta_B \hat{j}\right)$$
$$= \omega_A L_{AB}\cos\theta_A \hat{j} - \omega_A L_{AB}\sin\theta_B \hat{i} + \omega_B \frac{L_{BC}}{2}\cos\theta_B \hat{j} - \omega_B \frac{L_{BC}}{2}\sin\theta_B \hat{i}$$

$$\vec{V}_{G2} = \left[-\omega_A L_{AB}\sin\theta_B - \omega_B \frac{L_{BC}}{2}\sin\theta_B\right]\hat{i} + \left[\omega_A L_{AB}\cos\theta_A + \omega_B \frac{L_{BC}}{2}\cos\theta_B\right]\hat{j}$$

$$\vec{a}_{G1} = \cancel{\vec{a}_A}^{\,0} + \vec{\alpha}_A \times \vec{r}_{G1/A} + \vec{\omega}_A \times (\vec{\omega}_A \times \vec{r}_{G1/A})$$

$$\alpha_A = \alpha_A \hat{k}$$
$$\vec{r}_{G1/A} = \frac{L_{AB}}{2}(\cos\theta_A \hat{i} + \sin\theta_A \hat{j})$$
$$\vec{\omega}_A = \omega_A \hat{k}$$

$$\vec{a}_{G1} = \alpha_A \hat{k} \times \left(\frac{L_{AB}}{2}\cos\theta_A \hat{i} + \frac{L_{AB}}{2}\sin\theta_A \hat{j}\right) + \omega_A \hat{k} \times \left(\omega_A \frac{L_{AB}}{2}\cos\theta_A \hat{j} - \omega_A \frac{L_{AB}}{2}\sin\theta_A \hat{i}\right)$$

$$= \alpha_A \frac{L_{AB}}{2}\cos\theta_A \hat{j} - \alpha_A \frac{L_{AB}}{2}\sin\theta_A \hat{i} - \omega_A^2 \frac{L_{AB}}{2}\cos\theta_A \hat{i} - \omega_A^2 \frac{L_{AB}}{2}\sin\theta_A \hat{j}$$

$$\vec{a}_{G1} = \left[-\alpha_A \frac{L_{AB}}{2}\sin\theta_A - \omega_A^2 \frac{L_{AB}}{2}\cos\theta_A\right]\hat{i} + \left[\alpha_A \frac{L_{AB}}{2}\cos\theta_A - \omega_A^2 \frac{L_{AB}}{2}\sin\theta_A\right]\hat{j}$$

$$\vec{a}_{G2} = \vec{a}_B + \vec{\alpha}_B \times \vec{r}_{G2/B} + \vec{\omega}_B \times (\vec{\omega}_B \times \vec{r}_{G2/B})$$

$$\vec{a}_B = \cancel{\vec{a}_A}^{\,0} + \vec{\alpha}_A \times \vec{r}_{B/A} + \vec{\omega}_A \times (\vec{\omega}_A \times \vec{r}_{B/A})$$

$$= \alpha_A \hat{k} \times (L_{AB}\cos\theta_A \hat{i} + L_{AB}\sin\theta_B \hat{j}) + \omega_A \times (\omega_A L_{AB}\cos\theta_A \hat{j} - \omega_A L_{AB}\sin\theta_B \hat{i})$$

$$= \alpha_A L_{AB}\cos\theta_A \hat{j} - \alpha_A L_{AB}\sin\theta_B \hat{i} - \omega_A^2 L_{AB}\cos\theta_A \hat{i} - \omega_A^2 L_{AB}\sin\theta_B \hat{j}$$

$$\vec{a}_B = [-\alpha_A L_{AB}\sin\theta_B - \omega_A^2 L_{AB}\cos\theta_A]\hat{i} + [\alpha_A L_{AB}\cos\theta_A - \omega_A^2 L_{AB}\sin\theta_B]\hat{j}$$

$$\vec{\alpha}_B = \alpha_B \hat{k}$$

$$\vec{r}_{G2/B} = \frac{L_{BC}}{2}(\cos\theta_B \hat{i} + \sin\theta_B \hat{j})$$

$$\vec{\omega}_B = \omega_B \hat{k}$$

$$\vec{a}_{G2} = [-\alpha_A L_{AB}\sin\theta_B - \omega_A^2 L_{AB}\cos\theta_A]\hat{i} + [\alpha_A L_{AB}\cos\theta_A - \omega_A^2 L_{AB}\sin\theta_B]\hat{j} + \alpha_B \hat{k} \times \left(\frac{L_{BC}}{2}\cos\theta_B \hat{i} + \frac{L_{BC}}{2}\sin\theta_B\right)$$
$$+ \omega_B \times \left(\omega_B \frac{L_{BC}}{2}\cos\theta_B \hat{j} - \omega_B \frac{L_{BC}}{2}\sin\theta_B \hat{i}\right)$$

$$= [-\alpha_A L_{AB}\sin\theta_B - \omega_A^2 L_{AB}\cos\theta_A]\hat{i} + [\alpha_A L_{AB}\cos\theta_A - \omega_A^2 L_{AB}\sin\theta_B]\hat{j} + \alpha_B \frac{L_{BC}}{2}\cos\theta_B \hat{j} - \alpha_B \frac{L_{BC}}{2}\sin\theta_B \hat{i} - \omega_B^2 \frac{L_{BC}}{2}\cos\theta_B \hat{i} - \omega_B^2 \frac{L_{BC}}{2}\sin\theta_B \hat{j}$$

$$\vec{a}_{G2} = \left[-\alpha_A L_{AB}\sin\theta_B - \omega_A^2 L_{AB}\cos\theta_A - \alpha_B \frac{L_{BC}}{2}\sin\theta_B - \omega_B^2 \frac{L_{BC}}{2}\cos\theta_B\right]\hat{i} + \left[\alpha_A L_{AB}\cos\theta_A - \omega_A^2 L_{AB}\sin\theta_B + \alpha_B \frac{L_{BC}}{2}\cos\theta_B - \omega_B^2 \frac{L_{BC}}{2}\sin\theta_B\right]\hat{j}$$

Task 3: Numerical Verification

From this task onward, use the following numerical values:

$$L_{AB} = 20cm, L_{BC} = 60cm, Y = 30cm$$

Suppose that, at some instance, $\theta_A = 30^\circ, \omega_A = 0.2rad/s, \alpha_A = -0.1rad/s^2$, find:

- The angle θ_B
- The horizontal offset X
- The angular velocity of link BC (ω_B) and the slider velocity (v_C) (From your isolated expressions)
- The angular acceleration of link BC (α_B) and the slider acceleration (a_C) (From your isolated expressions)

- The velocity of the center of gravity of link AB ($\vec{v}_{G1}$) and The velocity of the center of gravity of link BC ($\vec{v}_{G2}$).
- The acceleration of the center of gravity of link AB ($\vec{a}_{G1}$) and The acceleration of the center of gravity of link BC ($\vec{a}_{G2}$).

Create a MATLAB Live Script and perform your calculations for Task 3 within MATLAB. Output each of these by removing the semicolon for these variables.

Task 3

```
theta_B = asind((Y-L_AB*sind(theta_A))/(L_BC))

X = L_AB*cosd(theta_A) + L_BC*cosd(asind((Y-L_AB*sind(theta_A))/(L_BC)))

omega_B = (L_AB*cosd(theta_A)*omega_A)/(-L_BC*cosd(theta_B))

X_dot = -L_AB*sind(theta_A)*omega_A + L_AB*cosd(theta_A)*omega_A*tand(theta_B)

alpha_B = (L_AB*sind(theta_A)*omega_A^2 - L_AB*cosd(theta_A)* alpha_A + L_BC*sind(theta_B)*omega_B^2)/(L_BC*cosd(theta_B))

X_double_dot = -L_AB*cosd(theta_A)*omega_A^2 - L_AB*sind(theta_A)* alpha_A - L_BC*cosd(theta_B)*omega_B^2 -  (tand(theta_B))*(L_AB*sind(theta_A)*omega_

v_G1 = [-omega_A*(L_AB/2)*sind(theta_A); omega_A*(L_AB/2)*cosd(theta_A); 0]

v_G2 = [-omega_A*L_AB*sind(theta_B) - omega_B*(L_BC/2)*sind(theta_B); -omega_A*L_AB*sind(theta_B) - omega_B*(L_BC/2)*sind(theta_B); 0]

a_G1 = [-(L_AB/2)*alpha_A*sind(theta_A)-(L_AB/2)*omega_A^2*cosd(theta_A); (L_AB/2)*alpha_A*cosd(theta_A) - (L_AB/2)*omega_A^2*sind(theta_A);0]

a_G2 = [-L_AB*alpha_A*sind(theta_A)-L_AB*omega_A^2*cosd(theta_A) - (L_BC/2)*alpha_B*sind(theta_B) - (L_BC/2)*omega_B^2*cosd(theta_B); L_AB*alpha_A*cosd
```

```
theta_B = 19.4712
X = 0.7389
omega_B = -0.0612
X_dot = -0.0078
alpha_B = 0.0390
X_double_dot = -0.0069
v_G1 = 3x1
    -0.0100
     0.0173
          0

v_G2 = 3x1
    -0.0072
    -0.0072
          0

a_G1 = 3x1
     0.0015
    -0.0107
          0

a_G2 = 3x1
    -0.0019
    -0.0107
          0
```

Part A Task 3: Numerical Verification

1. Position, Velocity and Acceleration ICs

```
clear all;
clc;
close all;

L_AB = 0.20; % [m]
L_BC = 0.60; % [m]
Y = 0.30; % [m]
theta_A = 30; % [deg]
omega_A = 0.2; % [rad/s]
alpha_A = -0.1; % [rad/s^2]
```

2. Link CG Velocity and Acceleration

```
theta_B = asind((Y-L_AB*sind(theta_A))/(L_BC))
```

```
theta_B =
19.4712
```

```
% x_c (x position of point C)
x_c = L_AB*cosd(theta_A) + L_BC*cosd(asind((Y-L_AB*sind(theta_A))/(L_BC)))
```

```
x_c =
0.7389
```

```
omega_B = (L_AB*cosd(theta_A)*omega_A)/(-L_BC*cosd(theta_B))
```

```
omega_B =
-0.0612
```

```
% v_c (velocity of point C)
v_c = -L_AB*sind(theta_A)*omega_A + L_AB*cosd(theta_A)*omega_A*tand(theta_B)
```

```
v_c =
-0.0078
```

```
alpha_B = (L_AB*sind(theta_A)*omega_A^2 - L_AB*cosd(theta_A)* alpha_A +
L_BC*sind(theta_B)*omega_B^2)/(L_BC*cosd(theta_B))
```

```
alpha_B =
0.0390
```

```
% a_c (acceleration of point C)
a_c = -L_AB*cosd(theta_A)*omega_A^2 - L_AB*sind(theta_A)* alpha_A -
L_BC*cosd(theta_B)*omega_B^2 -  (tand(theta_B))*(L_AB*sind(theta_A)*omega_A^2 -
L_AB*cosd(theta_A)* alpha_A + L_BC*sind(theta_B)*omega_B^2)
```

```
a_c =
-0.0069
```

```
v_G1 = [-omega_A*(L_AB/2)*sind(theta_A); omega_A*(L_AB/2)*cosd(theta_A); 0]
```

```
v_G1 = 3×1
   -0.0100
    0.0173
         0
```

```
v_G2 = [-omega_A*L_AB*sind(theta_B) - omega_B*(L_BC/2)*sind(theta_B);
-omega_A*L_AB*sind(theta_B) - omega_B*(L_BC/2)*sind(theta_B); 0]
```

```
v_G2 = 3×1
   -0.0072
   -0.0072
         0
```

```
a_G1 = [-(L_AB/2)*alpha_A*sind(theta_A)-(L_AB/2)*omega_A^2*cosd(theta_A); (L_AB/
2)*alpha_A*cosd(theta_A) - (L_AB/2)*omega_A^2*sind(theta_A);0]
```

```
a_G1 = 3×1
    0.0015
   -0.0107
         0
```

```
a_G2 = [-L_AB*alpha_A*sind(theta_A)-L_AB*omega_A^2*cosd(theta_A) -
(L_BC/2)*alpha_B*sind(theta_B) - (L_BC/2)*omega_B^2*cosd(theta_B);
L_AB*alpha_A*cosd(theta_A) - L_AB*omega_A^2*sind(theta_A) + (L_BC/
2)*alpha_B*cosd(theta_B) - (L_BC/2)*omega_B^2*sind(theta_B);0]
```

```
a_G2 = 3×1
   -0.0019
   -0.0107
         0
```

Part B: Crank Slider Kinematics Modeling

1. SolidWorks Screenshot

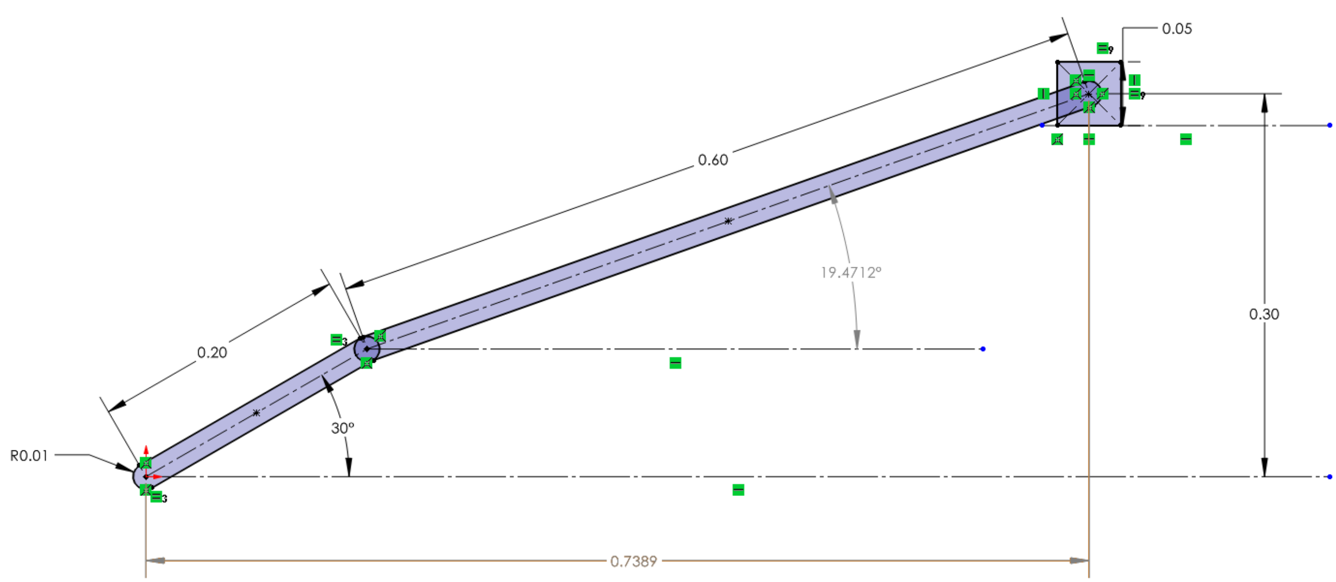


Figure Depicting SolidWorks Model Crank Slider Mechanism

The input initial conditions of Task 3 and the resulting theta_B and X are shown in the figure, as calculated by SolidWorks. These values match the results found in Task 3.

2. Creating ICs for Simulink (A and B matrix method)

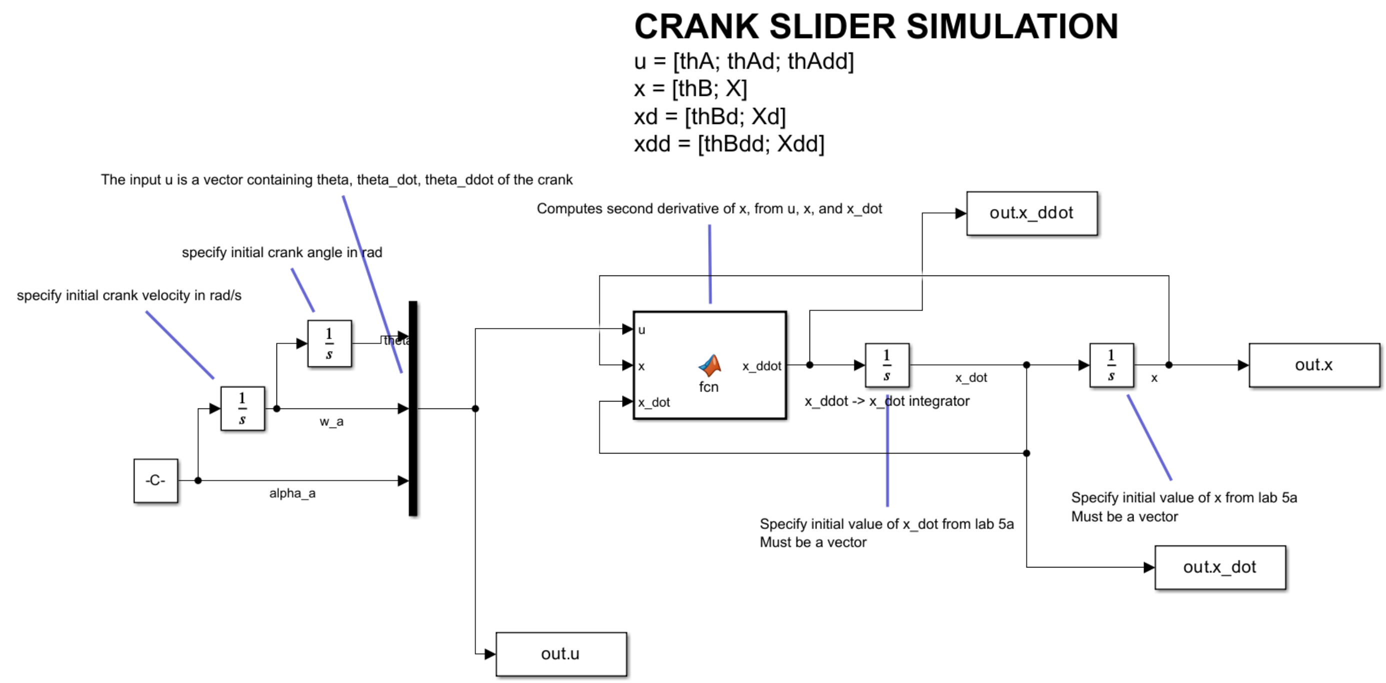
```
%Use angles for thetaA and thetaB from Task 3
cA = cosd(theta_A); %cos(thetaA)
sA = sind(theta_A);

cB = cosd(theta_B);
sB = sind(theta_B);

A = [L_BC*sB 1;
     L_BC*cB 0];
b_vel = [-L_AB*sA*omega_A;
         -L_AB*cA*omega_A];

x_dot = A\b_vel;
thb_d_task4 = x_dot(1);
vc_task4 = x_dot(2);

b_accel = [-L_AB*cA*(omega_A^2) - L_AB*sA*alpha_A - L_BC*cB*(omega_B^2);
           L_AB*sA*(omega_A^2) - L_AB*cA*alpha_A + L_BC*sB*(omega_B^2)];

x_ddot = A\b_accel;
thb_dd_task4 = x_ddot(1);
ac_task4 = x_ddot(2);
```

4. Defining the Structure

```

CrankSlider = struct();
CrankSlider.L_ab = L_AB;
CrankSlider.L_bc = L_BC;
CrankSlider.cA = cA;
CrankSlider.sA = sA;
CrankSlider.cB = cB;
CrankSlider.sB = sB;

CrankSlider.tha_rad = theta_A/180 * pi();
CrankSlider.w_a = omega_A;
CrankSlider.alpha_a = alpha_A;

CrankSlider.thb = theta_B/180 * pi();
CrankSlider.thb_d = omega_B;
CrankSlider.thb_dd = alpha_B;

CrankSlider.X = x_c;
CrankSlider.vc = v_c;
CrankSlider.ac = a_c;
```

```
CrankSlider.unknowns_vec = [CrankSlider.thb; CrankSlider.X];
CrankSlider.xd = x_dot;
CrankSlider.x_ddot = x_ddot;

crank_slider_out = sim('ME326_05_Activity4_BonneyCaiden_SimulinkFile');
```

5. Simulink Model

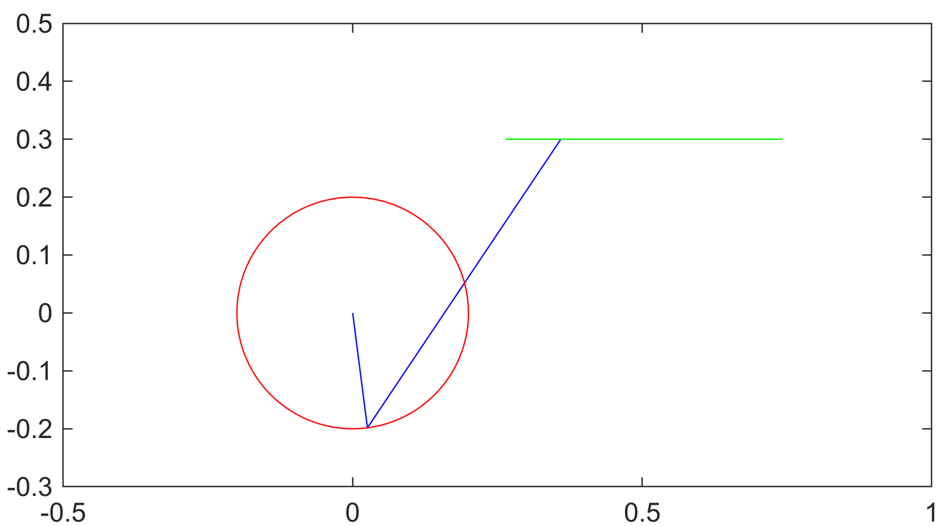


6. Animation Pre-Processing

```
%% Simulation
%u = [th1 th1d ]; % input vector FOR STATIONARY
u = [crank_slider_out.u(:,1), crank_slider_out.u(:,2)]; %MOVING
%x = [th2 r3]; % state vector FOR STATIONARY
x = [crank_slider_out.x(:,1), crank_slider_out.x(:,2)];

%% Animation
N = size (x ,1); % Number of rows of data
cA_coord = cos(u(: ,1)); % cos( thetaA )
sA_coord = sin(u(: ,1)); % sin( thetaA )
cB_coord = cos(x(: ,1)); % cos( thetaB )
sB_coord = sin(x(: ,1)); % sin( thetaB )
% X and Y components for each point on the crank - slider mechanism .
%                Pt. A                Pt. B                                       Pt.
C
x_coord = [ zeros(N ,1), cA_coord * CrankSlider.L_ab, cA_coord * CrankSlider.L_ab +
cB_coord * CrankSlider.L_bc];
y_coord = [ zeros(N ,1), sA_coord * CrankSlider.L_ab, ones(N, 1) * Y];
```

6 and 7. Animating Mechanism and Points B and C

```
% Loop through the frames and plot them all.
for n = 1:N
    fig = figure (1); % Make sure the animation stays put
    plot ( x_coord (n ,:) ,y_coord (n ,:), 'b'); % Plot the whole mechanism at once
    hold on;
    storage_matrix_x = x_coord;
    storage_matrix_y = y_coord;
    plot(storage_matrix_x(1:n,2),storage_matrix_y(1:n,2),"r"); %POINT B
    plot(storage_matrix_x(1:n,3),storage_matrix_y(1:n,3),"g"); %POINT C
    hold off;
    axis('equal'); % Makes x- and y- scalle equal
    xlim([-0.5 1]);
    ylim([-0.3 0.5]);
    drawnow ('limitrate'); % Forces MATLAB to update plot
    frame = getframe(fig);
    im{n} = frame2im(frame);
end
```




```
% % The commented out for loop below places each frame of the gif into a 52 column
figure
% % however matlab cannot display that many subplots to an adequate
% % resolution
%
```

```
% for n = 1:N
%     subplot(N,52,n)
%     imshow(im{n});
% end
%
% By creating a grid of the images and saving the stitched image the same
% result occurs
%
%
*************************************************************************************
******************************************************************* %

% Define grid size
    % Square grid
    % rows = ceil(sqrt(N)); % Number of rows
    % cols = ceil(N / rows); % Number of columns

    % Keeping the same 52 columns as previously denoted
    cols = 52; % Number of columns
    rows = ceil(N / cols); % Number of rows

% Convert cell array to matrix (assuming all frames are the same size)
frame_size = size(im{1}); % Get the size of one frame (height x width x channels)
H = frame_size(1); % Frame height
W = frame_size(2); % Frame width

% Initialize an empty image (white background)
stitched_image = uint8(255 * ones(rows * H, cols * W, 3)); % 3 for RGB

% Loop to place each image into the correct position
for n = 1:N
    % Get row and column index
    r = floor((n-1) / cols) + 1; % Row index
    c = mod(n-1, cols) + 1; % Column index

    % Get pixel indices for placement
    row_start = (r-1) * H + 1;
    row_end = r * H;
    col_start = (c-1) * W + 1;
    col_end = c * W;

    % Place the image in the grid
    stitched_image(row_start:row_end, col_start:col_end, :) = im{n};
end

% Display the stitched image
figure(2);
imshow(stitched_image);
```

```
% Save the final stitched image
imwrite(stitched_image, 'crank_slider_all_frames.png');
%
****************************************************************************************
********************************************************************** %

% Save the crank_slider as an animated gif
filename = 'crank_slider.gif'; % Specify the output file name
for n = 1:N
    [A,map] = rgb2ind(im{n},256);
    if n == 1
        imwrite(A,map,filename,'gif','LoopCount',Inf,'DelayTime',0.25);
    else
        imwrite(A,map,filename,'gif','WriteMode','append','DelayTime',0.25);
    end
end
```

8. Plotting Velocity of Point B and C

```
%we need the derivative of the coordinates of point B and point C
dBx = (-sA_coord * CrankSlider.L_ab) .* crank_slider_out.u(:,2);
dBy = (cA_coord * CrankSlider.L_ab) .* crank_slider_out.u(:,2);
dCx = (-sA_coord * CrankSlider.L_ab) .* crank_slider_out.u(:,2) + (-sB_coord *
CrankSlider.L_bc) .* crank_slider_out.x_dot(:,1);

%ADD FIGURE TITLES AND AXES TITLES!
figure(3)
set(gcf, 'Position', [100, 100, 800, 600]); % Sets width to 800 pixels and height
to 600 pixels

subplot(2,1,1);
plot(crank_slider_out.tout, dBx)

hold on
plot(crank_slider_out.tout, dBy);
title('Velocity of Point B');
legend('x velocity', 'y velocity', 'Location', 'northwest')
xlabel('Time, t [s]');
ylabel('Velocity, v [m/s]');
% Sets the y limits to be symmetric about the furthest limit
ymax = max(abs(ylim));
ylim([-ymax ymax]);
```

```
hold off;

subplot(2,1,2);
plot(crank_slider_out.tout, dCx);

hold on;
plot(crank_slider_out.tout, zeros(N,1));
title('Velocity of Point C');
legend('x velocity', 'y velocity', 'Location', 'northwest')
xlabel('Time, t [s]');
ylabel('Velocity, v [m/s]');
% Sets the y limits to be symmetric about the furthest limit
ymax = max(abs(ylim));
ylim([-ymax ymax]);

hold off;
```