ME 206 – Statics and Dynamics

Experiment 4 - Design a planar four-bar mechanism with one link (crank) that can rotate (at a constant angular velocity) completely, and the opposite link (rocker) rotates partially.

Group 12

Roll Number	Name	Percentage Contribution(%)
24110238	Kavya Parmar	25
24110239	Vinit Parmar	25
24110240	Parshva Patel	25
24110243	Parth Singh	25

Introduction:

A four–bar mechanism is one of the most fundamental planar mechanisms used in mechanical systems to generate controlled motion. It consists of four rigid links connected in a closed loop by revolute joints, forming one of the simplest yet most versatile mechanical linkages. Depending on the relative lengths of its links, the mechanism can exhibit various types of motion, such as complete rotation, oscillation, or complex coupler motions. Due to its simplicity, reliability, and predictable behaviour, four–bar linkages are widely used in engines, pumps, robotic arms, suspension systems, manufacturing equipment, and numerous real-world machines.

In this experiment, we focus on a specific category of four-bar mechanisms known as the crank–rocker mechanism, in which one link (the crank) is capable of a full rotation, while the opposite link (the rocker) oscillates back and forth. When the crank rotates at a constant angular velocity, the coupler link connecting the crank and rocker exhibits complex translational and rotational motion. Studying this motion is crucial for understanding kinematic relationships, velocity transmission characteristics, and performance of mechanisms designed for real engineering applications.

The objective of this experiment is to design and fabricate a functional four-bar crank–rocker mechanism and experimentally determine the velocity of the centre of mass of the coupler link, along with its angular velocity, for at least twelve evenly spaced crank angles. These experimentally obtained values are then compared with theoretical calculations derived from classical kinematic loop equations and velocity analysis. This allows us to observe the mobility of the mechanism, understand the dependency of coupler motion on crank position, and visualise how theoretical planar kinematics manifest in a real physical mechanism.

Through this study, we aim to bridge the gap between theory and practice by demonstrating how the fundamental kinematic principles—vector loop closure, instantaneous centers, and velocity analysis—are applied to a real mechanical system. The experiment offers an illustrative way to see how relative motion between rigid bodies is transmitted through joints, how coupler trajectories vary, and how errors or deviations arise in practical setups. Overall, this experiment enhances our understanding of mechanism design, motion transfer, and the dynamic behaviour of multi-link mechanical systems.

Experimental Design:

Aim of the Experiment

To design and fabricate a planar four-bar crank–rocker mechanism and experimentally determine:

1. The velocity of the centre of mass of the coupler link
2. The angular velocity of the coupler link

for at least twelve input crank angles (0, π/6, π/3, …) and to compare these results with analytical velocity predictions obtained from kinematic modelling.

Apparatus and Equipment

- 3D-printed four-bar linkage (Hyper-PLA material)
- MG996R servo motor (for constant angular velocity of crank)
- Arduino UNO with motor control code
- Nuts and bolts (3D-printed) for joint connections
- Measuring scale and digital calliper
- Smartphone camera (for recording motion)
- Tracker video analysis software
- Laptop with Python (NumPy, Matplotlib, Pandas) for plot generation

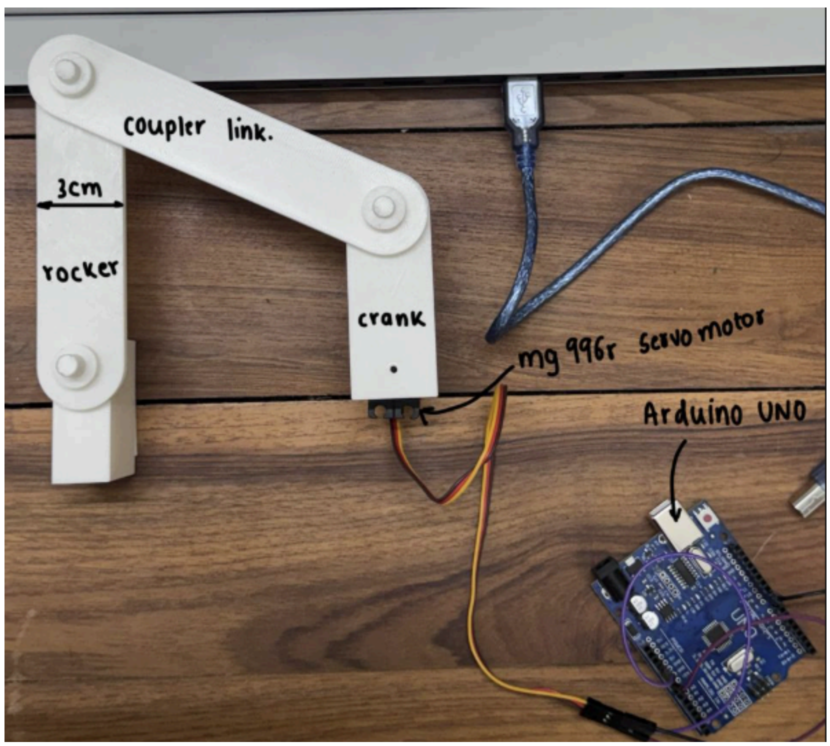
Setup Description

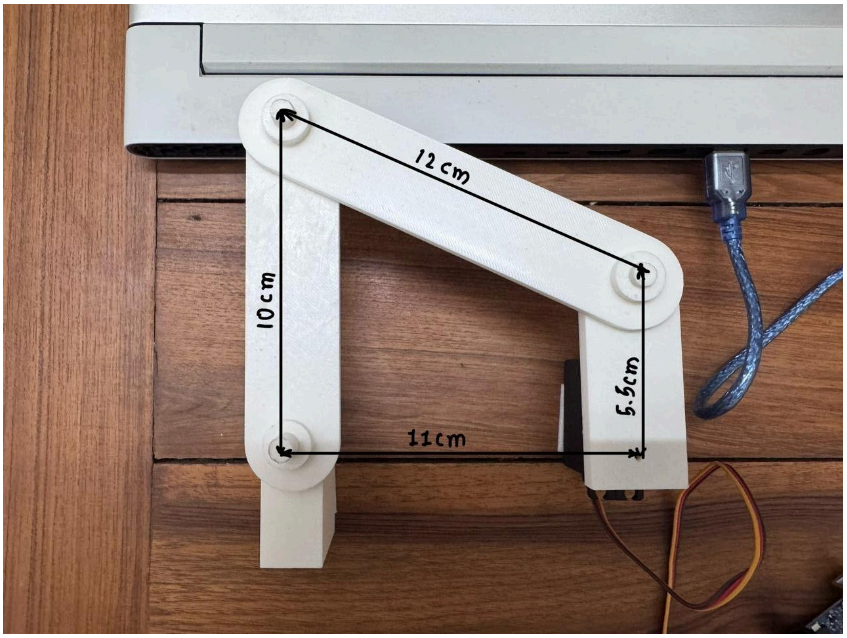
The experimental setup consists of a fully 3D-printed four-bar crank–rocker mechanism, with all four links, circular joint flanges, and even the nuts and bolts fabricated using Hyper PLA filament. The base of the mechanism is fixed to a flat wooden surface, providing a rigid ground link. The **crank link (5.5 cm)** is directly mounted on the shaft of the **MG996R servo motor**, which is controlled using an Arduino UNO to ensure constant and repeatable angular velocity. A simple Arduino code was used to rotate the servo uniformly over the desired angular range.

The **coupler link (12 cm)** connects the crank and rocker through two 3D-printed revolute joints. The **rocker link (10 cm)** is connected to the opposite end and is constrained to oscillate. The distance between the crank pivot and the rocker pivot forms the fixed **ground link (11 cm)**.

During operation, the servo motor continuously rotates the crank, causing the remaining links to move in a closed kinematic loop. The entire motion was recorded using a smartphone camera positioned above the mechanism. To convert pixels to physical distances, a calibration scale was placed near the setup. The recorded video was later analysed using Tracker to obtain the coupler’s position at twelve different crank angles.

This simple yet effective setup creates a clear demonstration of theoretical planar kinematics and provides accurate experimental data for comparison.





Fabrication Details

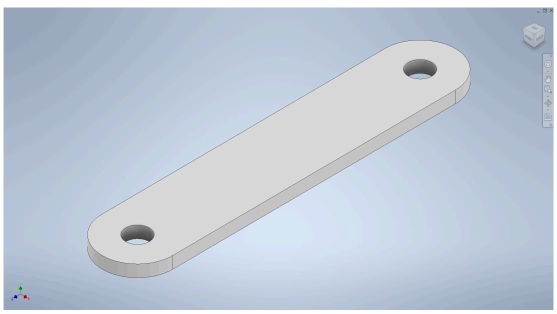
The entire mechanism, including all links, joints, connecting bolts, and mounting components, was fully fabricated using 3D printing. We used Hyper PLA filament, which provides good stiffness and dimensional accuracy, making it suitable for kinematic experiments.

Steps in Fabrication:

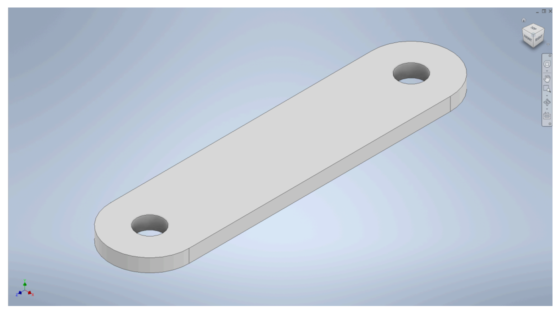
1. CAD Modelling:
   Each link (ground, crank, coupler, rocker) was designed on CAD software with revolute joint holes and flanges.
2. 3D Printing:
   All parts were printed on an FDM 3D printer using Hyper PLA at 0.2 mm layer height and 20–30% infill.
   The printed joints allowed smooth rotation while maintaining minimal backlash.
3. Joint Assembly:
   Custom 3D-printed bolts and nuts were inserted into the joint holes to form rotational joints.
   The tolerances allowed free motion without play.

4. Servo Motor Integration:
 The crank was directly mounted on the MG996R servo horn and secured with screws.
 This ensured accurate transmission of rotational input.
5. Final Assembly:
 The mechanism was mounted onto a wooden base to create a fixed ground link.
 After assembly, the mobility and smoothness of each joint were manually tested.
6. Electronics Setup:
 An Arduino Uno was used to program the MG996R servo to rotate at a nearly constant angular velocity (ω).
 A simple code with time-controlled PWM ensured uniform motion.

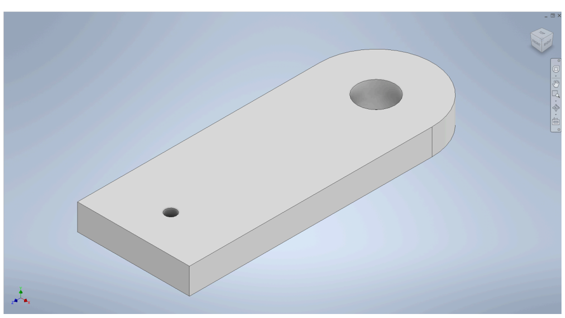
The fabrication approach ensured the mechanism was lightweight, rigid, and cost-effective, while allowing for high repeatability during experimental observations.



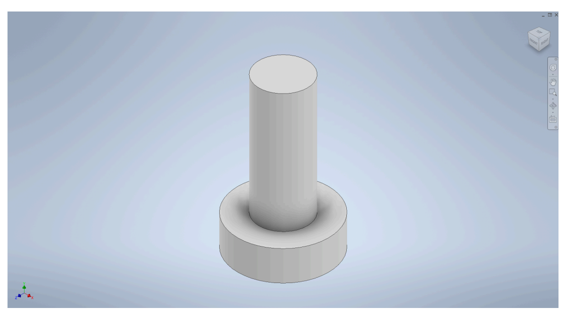
(Coupler Link)



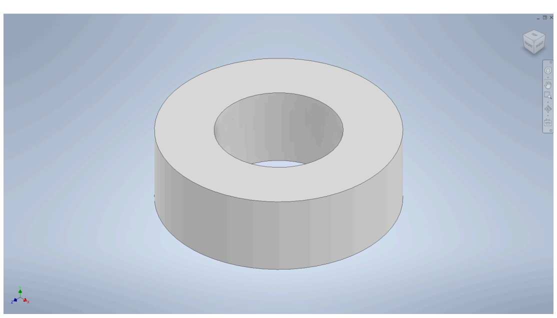
(Rocker)

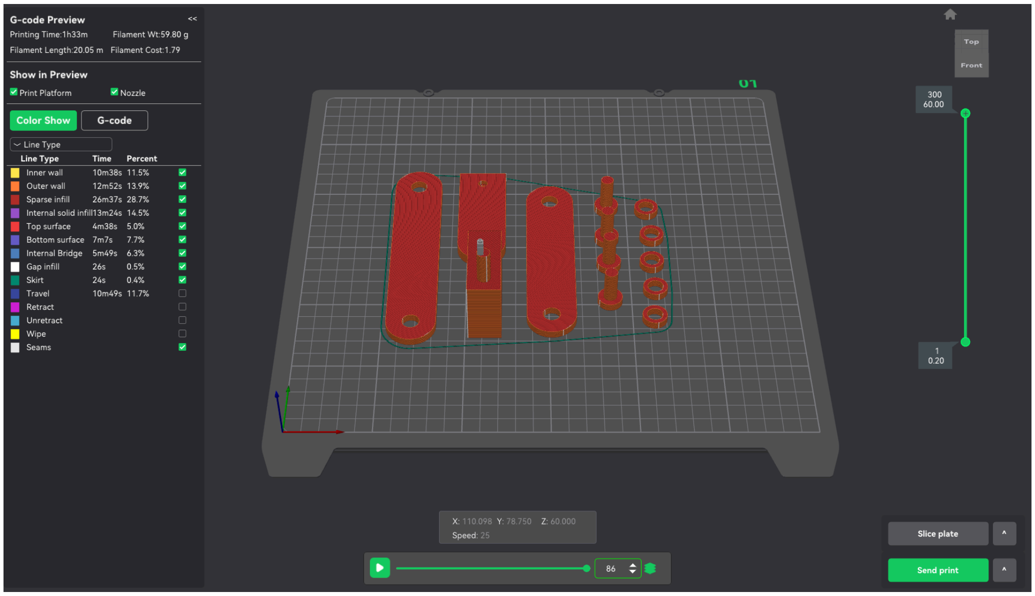


(Crank)



(Bolt)





(3D Printing)

Measurement Techniques:

To accurately determine the linear velocity of the centre of mass of the coupler and the angular velocity of the coupler for different crank angles, a combination of video-based motion tracking and controlled servo motor input was used. The following measurement techniques ensured consistent, repeatable, and quantifiable results:

1. Servo-Controlled Constant Angular Velocity Input

An **MG996R servo motor** was used to rotate the crank at a near-constant angular velocity. The servo was controlled using an **Arduino UNO**, programmed using a simple PWM-based code that sets the servo to sweep across the full crank rotation range.

- The angular velocity of the crank was estimated using:
  - The programmed delay between servo positions
  - The known servo rotation step
  - Time recorded in the video

This ensured a consistent and repeatable motion input across all trials.

2. High-Resolution Video Capture

The motion of the mechanism was recorded using a smartphone camera placed above the setup to obtain a clear top view.

- The camera was kept **fixed at a constant height** using a support to minimise parallax error.
- Adequate lighting ensured that the centres of the crank, coupler, and rocker joints were clearly visible.
- A calibration scale (a ruler placed on the board) was included in the frame to convert pixel distances to real-world distances (cm or m).

The resulting video captured the crank rotation and corresponding coupler motion with sufficient clarity for motion-tracking.

3. Tracker Software for Motion Tracking

The recorded video was imported into **Tracker**, an open-source physics analysis tool. Tracker was used to track:

- The crank pivot
- The coupler–crank joint
- The coupler–rocker joint
- The approximate centre of mass of the coupler

Steps in Tracker:

1. The video was calibrated using the ruler visible in the frame.
2. A coordinate system was defined with the ground link aligned horizontally.
3. The centre of the coupler COM was manually marked at each frame.
4. The software was automatically generated:
    - Position–time data
    - Velocity components (Vx, Vy)
    - Angular position and angular velocity

By advancing the video frame by frame, the crank angle corresponding to each frame was determined.

4. Crank Angle Identification

For every frame used in analysis, the crank angle θ was identified by measuring the angle between:

- The line joining the crank pivot to the coupler–crank joint
- The horizontal reference through the crank base pivot

Tracker's built-in angle-measurement tool was used for this.

Angles corresponding to:

$0, \pi/6, \pi/3, \ldots 11\pi/6$

were selected manually from the video by identifying frames closest to these angles.

5. Centre of Mass Velocity Calculation

The COM of the coupler was tracked manually. Tracker automatically computed:

- Instantaneous linear velocity vector
- Magnitude of velocity

Using:

$$V_{com} = \sqrt{Vx^2 + Vy^2}$$

These values were recorded for each of the twelve crank angles.

6. Angular Velocity of the Coupler

To compute angular velocity, two points on the coupler were tracked:

1. The crank–coupler joint
2. The rocker–coupler joint

From these points, Tracker automatically computed:

- The coupler orientation angle $\varphi(t)$
- Angular velocity $\omega(t) = d\varphi/dt$

This method eliminates dependency on exact link dimensions and instead relies on the visual motion, making it robust against minor printing tolerances.

7. Data Extraction and Post-Processing

The resulting position and velocity data were exported as CSV files and further processed using Python:

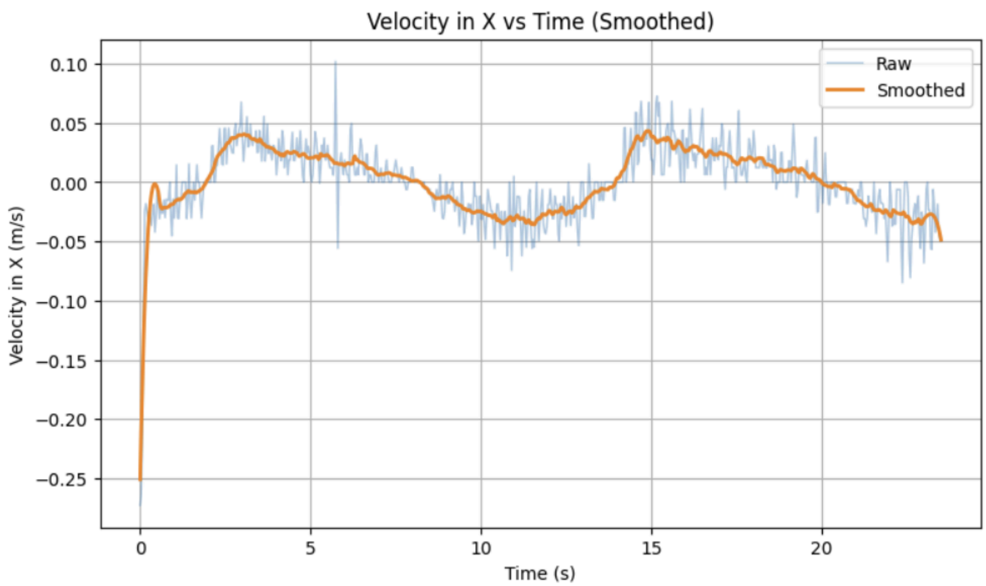
- **NumPy** for numerical calculations
- **Pandas** for structuring and indexing the dataset
- **Matplotlib** for plotting velocity vs. crank angle graphs

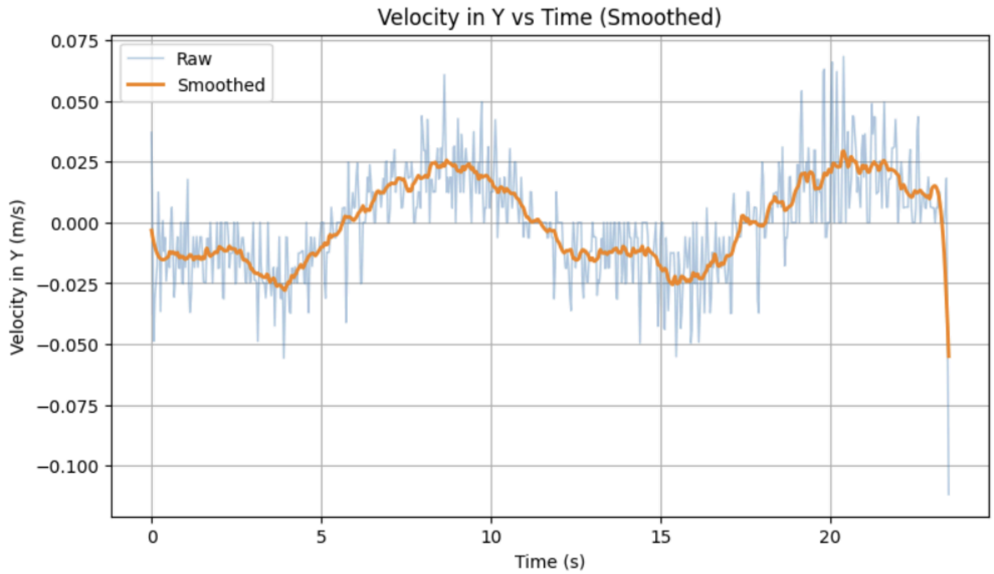
This post-processing ensured smooth curves, allowing for a comparison between theoretical and experimental results.

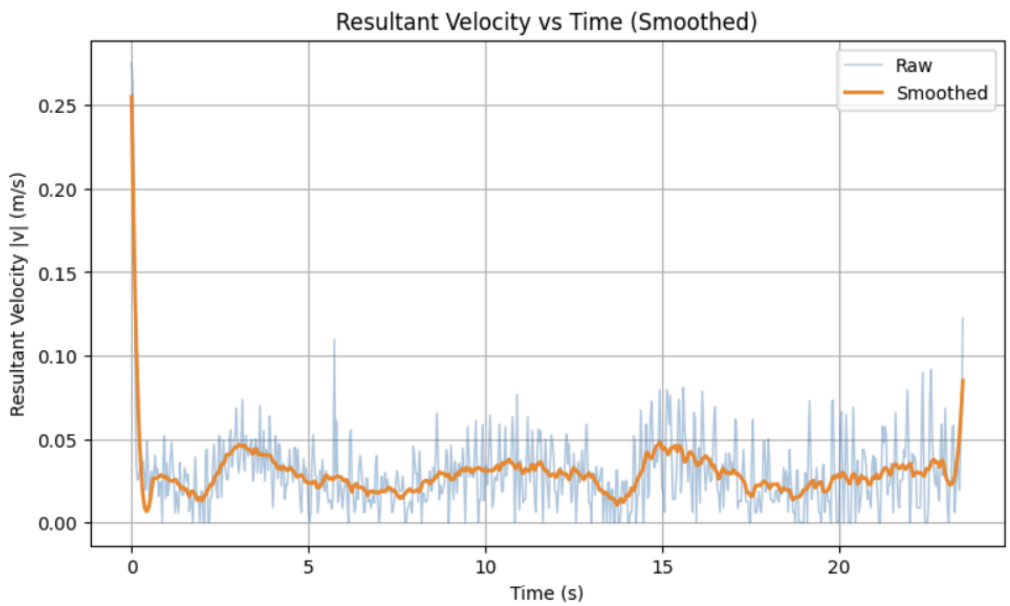
Mathematical and Theoretical Modelling:

1) Velocity of Centre of Mass of Coupler link:

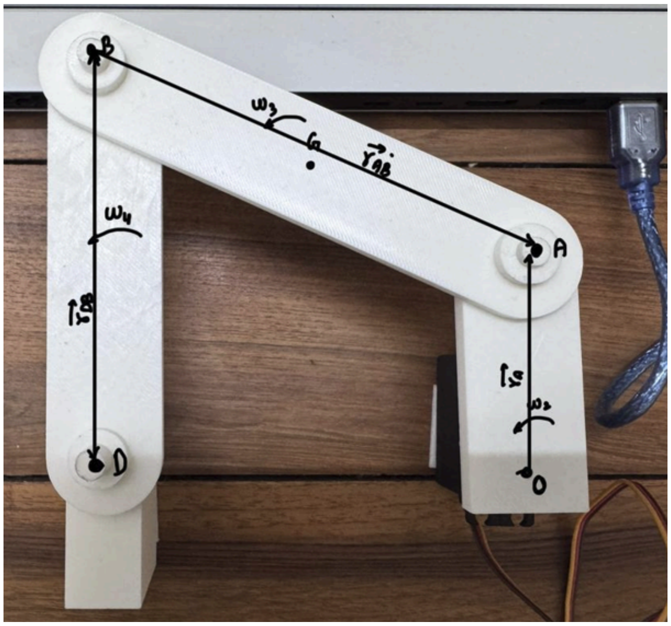
- **Experimental:**
  Through video tracker software, we were able to determine the position of the centre of mass of the coupler link over time. Using simple code, we plotted a velocity v/s time graph for the centre of mass of the coupler link.




Velocity in Y vs Time (Smoothed)
Raw
Smoothed
Velocity in Y (m/s)
Time (s)
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
0
5
10
15
20


Resultant Velocity vs Time (Smoothed)
Raw
Smoothed
Resultant Velocity |v| (m/s)
Time (s)
0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20

- **Analytical:**



Here $\omega_{2,}$ which is the input of the servo motor, is known.

$\therefore V_A = \omega_2 \times r_A$

$\therefore V_B = V_A + V_{B/A}$

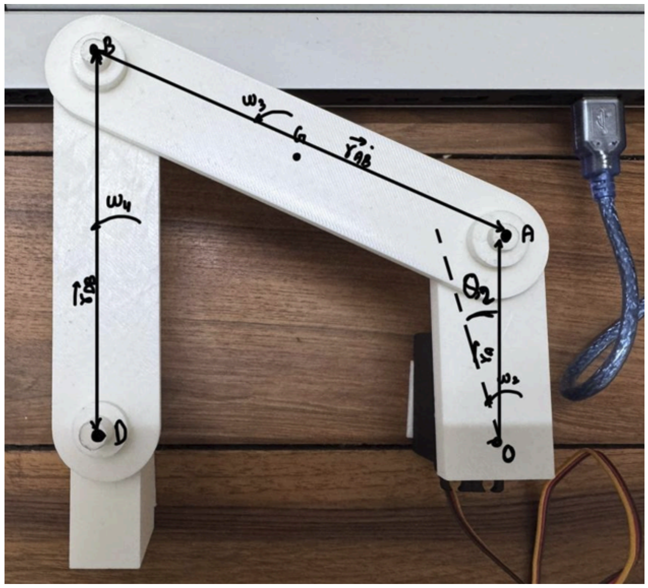
$\therefore V_B = V_A + (\omega_3 \times r_{AB})$ ——--------------Eqn-1
$\therefore V_B = \omega_4 \times r_B$ —--------------------Eqn-2

By equating the x and y components of Equation 1 and Equation 2, a system of two linear equations is established with two unknowns: the magnitudes of the angular velocities ω_3 and ω_4.

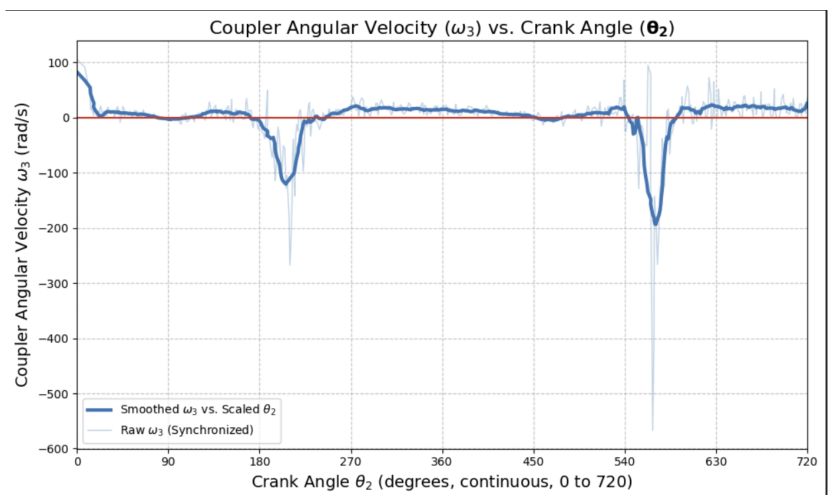
$\therefore V_G = V_A + V_{G/A}$

$\therefore V_G = V_A + (\omega_3 \times r_{AG})$

2) Angular Velocity of Coupler Link:



- **Experimentally:**
  Through video tracker software, we were able to determine the position of a point coupler link over time. Using simple code, we plotted a ω_3 v/s angle of crank ($\boldsymbol{\Theta}_2$) graph for the centre of mass of the coupler link.



- **Analytically:**
  To find the angular velocity of the coupler link, we will use the following equation:

$\therefore V_A = \omega_2 \times r_A$ (ω_2 is the input from the servo motor, which is known)

$\therefore V_A = \omega_2 \times (- r_a \sin\boldsymbol{\Theta}_2\, i + r_a \cos\boldsymbol{\Theta}_2\, j)$

$\therefore V_B = V_A + V_{B/A}$

$\therefore\ V_B = V_A + (\omega_3 \times r_{AB})$ —--------------Eqn-3

$\therefore\ V_B = \omega_4 \times r_{BD}$ —-------------------Eqn-4

By equating the x and y components of Equation 3 and Equation 4, a system of two linear equations is established with two unknowns: the magnitudes of the angular velocities ω_3.

r_A, r_{AB}, r_{BD} are subjected to change as the angle Θ_2 changes.

Result and Discussion

1. Mass Measurement

The individual links were fabricated using Hyper-PLA material. Before assembly, the mass of each link was measured using a digital scale to assist in potential dynamic force analysis.

- **Material:** Hyper PLA (High Precision Polylactic Acid)
- **Infill Density:** 25%

2. Angular Velocity of the coupler link:

The motion of the mechanism was recorded and analyzed analytically; the crank angle (Θ_2) was identified for twelve distinct positions. For each angle, the angular velocity of the coupler was calculated.

Table 1: Analytically calculated Angular Velocity

Crank Angle (Θ_2)	Coupler Angular Velocity (ω) [rad/s]
0	-96.7
$\frac{\pi}{6}$	-92.7
$\frac{\pi}{3}$	-76.8
$\frac{\pi}{2}$	-52.4
$\frac{2\pi}{3}$	-24.9

$\frac{5\pi}{6}$	3.9
π	30.6
$\frac{7\pi}{6}$	51.9
$\frac{4\pi}{3}$	63.8
$\frac{3\pi}{2}$	63.8
$\frac{5\pi}{2}$	52.4
$\frac{11\pi}{6}$	46.3

3. Discussion

The experimental results obtained from Tracker were plotted against the theoretical values derived from the vector loop equations.

- **Velocity Trends:** The velocity of the coupler's COM exhibited a periodic fluctuation characteristic of the crank-rocker mechanism. As the crank rotates at a constant velocity, the coupler accelerates and decelerates as the rocker approaches its limit positions.
- **Comparison:** The experimental curve largely follows the theoretical trajectory. However, slight deviations were observed near the toggle positions (where the crank and coupler are collinear), likely due to joint looseness or minor backlash in the 3D-printed parts.
- **Motion Quality:** The visualisation of the data confirms that the servo motor maintained a reasonably constant angular velocity, though minor ripples were observed in the Tracker data, which may be attributed to the PWM signal steps of the Arduino.

4. Sources of Discrepancy

While the experiment successfully demonstrated the kinematic principles, the following sources of error contributed to the percentage difference between theoretical and experimental values:

1. **Joint Friction and Backlash:** Although Hyper-PLA allows for smooth rotation, the 3D-printed revolute joints possess inherent friction and minor clearances (backlash). This causes the physical mechanism to "lag" slightly behind the theoretical position.

2. **Parallax Error:** Despite positioning the camera at a fixed height, slight lens distortion or non-perpendicular viewing angles could introduce errors in the pixel-to-length conversion during Tracker analysis.

3. **Servo Motor Jitter:** The MG996R is a hobby-grade servo. While programmed for constant velocity, it operates on a feedback loop that can cause micro-oscillations (jitter) rather than perfectly smooth rotation.

4. **Marker Tracking Manual Error:** The COM was tracked manually in some frames, introducing human error in clicking the exact pixel centroid frame-by-frame.

5. Sources of Improvement

To enhance the accuracy of future iterations of this experiment:

- **Improved Joints:** Replacing 3D-printed pins with metal ball bearings would significantly reduce friction and play.

- **High-Frame-Rate Capture:** Using a camera with a higher frame rate (60 or 120 fps) would provide more data points for the velocity calculation, smoothing out the derivative noise in Tracker.

- **Automated Tracking:** Using high-contrast fiducial markers (e.g., bright colored dots) on the joints would allow Tracker to use "Autotracker" exclusively, removing human error from manual point selection.

6. Software/Tools Used

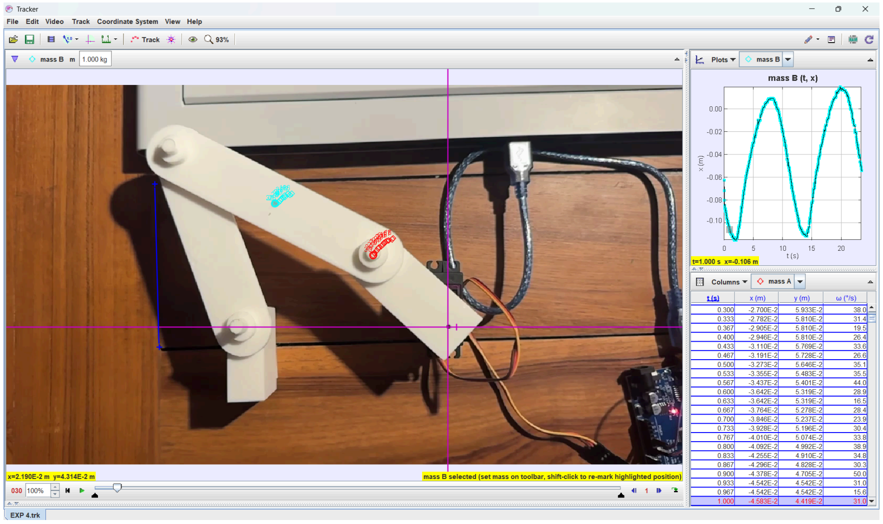
- **Hardware Interface:** Arduino IDE (for MG996R Servo control).

Code:

```
#include <Servo.h>

const int SERVO_PIN = 9;

int ROTATION_VALUE = 205;

unsigned long startTime = 0;

bool isRotating = false;

Servo motor;

void setup() {

  Serial.begin(9600);
```

```
  Serial.println("MG996R Continuous Rotation Control Started.");

  Serial.println("Target speed: 10 RPM (1 rotation every 6 seconds).");

  motor.attach(SERVO_PIN);

  motor.write(ROTATION_VALUE);

  Serial.print("Initial rotation value set to: ");

  Serial.println(ROTATION_VALUE);

  Serial.println("--- START CALIBRATION ---");

  Serial.println("Watch the servo shaft. It should take 6 seconds for one full rotation (after
    calibration).");

  Serial.println("Currently set to 105 for faster spin to confirm motor is working.");

  Serial.println("If it's too fast, move the ROTATION_VALUE closer to 90 (e.g., 94, 93, 92).");

  Serial.println("If it's too slow, move the ROTATION_VALUE further from 90 (e.g., 96, 97,
    98).");

  Serial.println("To reverse direction, use a value below 90 (e.g., 85).");

  Serial.println("--- END CALIBRATION ---");

  startTime = millis();

  isRotating = true;

}

void loop() {

  if (isRotating && (millis() - startTime >= 60000)) {

    Serial.println("\n*** 10 Rotations should have been completed by now if calibrated
    correctly. ***");

    startTime = millis();

  }

}
```

- **Motion Analysis:** Tracker Video Analysis and Modelling Tool.



- **Data Processing & Plotting:** Python 3.x, utilizing:
    - *NumPy* for vector calculations.
    - *Pandas* for data structuring.
    - *Matplotlib* for generating comparison graphs.
- **Fabrication:** CAD Software for linkage design.

Acknowledgement:

We thank those who assisted and participated in completing our experiment 3 for the Statics and Dynamics course. Most of all, we sincerely thank Professor K.R. Jayaprakash for his remarkable mentorship and expert guidance. His clear explanations and considerate assistance have significantly enhanced our understanding of the subject and helped us overcome the experiment's intricacies. We would especially like to thank the teaching assistants, whose valuable advice and insightful observations significantly contributed to the smooth operation of our work. Moreover, we thank the Indian Institute of Technology Gandhinagar for providing the opportunities that enabled us to perform this experiment.

Lastly, we would like to thank our fellow students and colleagues for their cooperation, feedback, and support, which contributed to the experiment's accomplishment.