Design and Fabrication of Experiment for Dynamic Analysis of Mechanisms

submitted in partial fulfilment of the requirements

for the degree of

BACHELOR OF TECHNOLOGY

in

MECHANICAL ENGINEERING

by

AAKASH ME16B001

A. AKHIL ME16B003

Supervisor(s)

Dr. Sriram Sundar

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI

December 2019

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Place: Tirupati
Date: 06-12-2019

Signature
Aakash
ME16B001

Place: Tirupati
Date: 06-12-2019

Signature
A. Akhil
ME16B003

BONA FIDE CERTIFICATE

This is to certify that the thesis titled **Design and Fabrication of Experiment for Dynamic Analysis of Mechanisms**, submitted by **Aakash** and **A. Akhil**, to the Indian Institute of Technology, Tirupati, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by them under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: Tirupati
Date: 08-12-2019

Dr. Sriram Sundar
Guide
Assistant Professor
Department of Mechanical Engineering
IIT Tirupati - 517501

ACKNOWLEDGEMENTS

Thanks to all those who helped us during the course of our B.Tech. project at IIT Tirupati. We are thankful to and fortunate enough to get constant encouragement, support and guidance from Dr. Sriram Sundar of Mechanical Engineering Department which helped us in successfully completing first phase of our project work. We would also like to extend our sincere esteems to all staff in laboratory for their timely support.

To our families, thank you for encouraging us in all of our pursuits and inspiring us to follow our dreams. We are especially grateful to our parents, who supported us emotionally and financially.

ABSTRACT

KEYWORDS: Slider crank mechanism; Four bar mechanism; Six-bar mechanism; Dynamic analysis; Kinematic analysis; Simulation.

This report explains the detailed process of design and analysis of an experimental setup, which consists of a 3-in-1 mechanism. The setup is designed to perform static, kinematic and dynamic analysis of mechanisms. The design and analysis were performed analytically and numerically by using various commercial software's. The results thus obtained were used to determine the link geometry of the mechanism and in selecting the suitable sensors which in turn will be used for fabrication and instrumentation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

The following list describes the significance of various abbreviations and acronyms used throughout the report.

KDM	Kinematics and Dynamics of Machinery
PMSM	Permanent Magnet Synchronous Motor
CAD	Computer Aided Design
CSV	Comma Seperated Values
FBD	Free Body Diagram

NOTATION

The following notations are used throughout the report.

$\vec{r}_{A/B}$	Position vector of point A with respect to point B, m
ω_{AB}	Angular velocity of link AB, $\frac{rad}{s}$
$\vec{V}_{A/B}$	Velocity vector of point A with respect to point B, $\frac{m}{s}$
$\vec{a}_{A/B}$	Acceleration vector of point A with respect to point B, $\frac{m}{s^2}$
α_{AB}	Angular acceleration of link AB, $\frac{rad}{s^2}$

CHAPTER 1

INTRODUCTION

1.1 Motivation and objective

The study of the *"Kinematics and Dynamics of Machinery" (IITT course code: ME2206)* lies at the very core of a mechanical engineering background. Although, little has changed in the way the subject is presented, our methodology brings the subject alive and current. We present the design and fabrication of a novel experimental setup for carrying out static, kinematic and dynamic analysis of three different mechanisms in a single setup (Figure 1.1). The mechanism is designed to be configurable to three different types of mechanisms namely - double crank, slider crank and a six bar mechanism depending on the use case as shown in Figure 1.2. The mechanism has retrofitted parts (different link lengths and sliders) to facilitate multiple experiments in the same setup.

The learner gets to "play" with the mechanism parameters and immediately understand their effects. This will enhance one's grasp of the concepts and the development of analytical skills. Hence greatly supplementing and reinforcing the theoretical understanding of the undergraduate students taking the course.

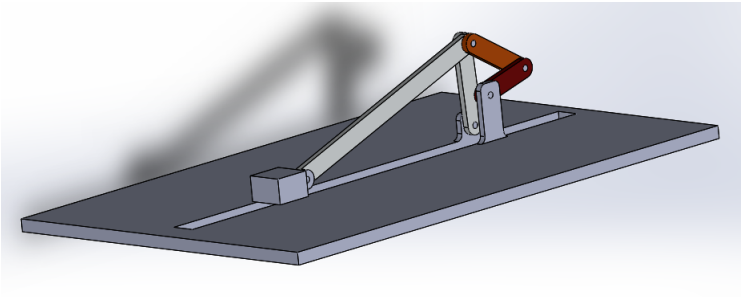


Figure 1.1: CAD model of the mechanism

1.2 Present method(s) and challenges

In the present scenario finding standard mechanisms integrated with different sensor measurement and data acquisition is hard to find. Almost all the mechanisms that are

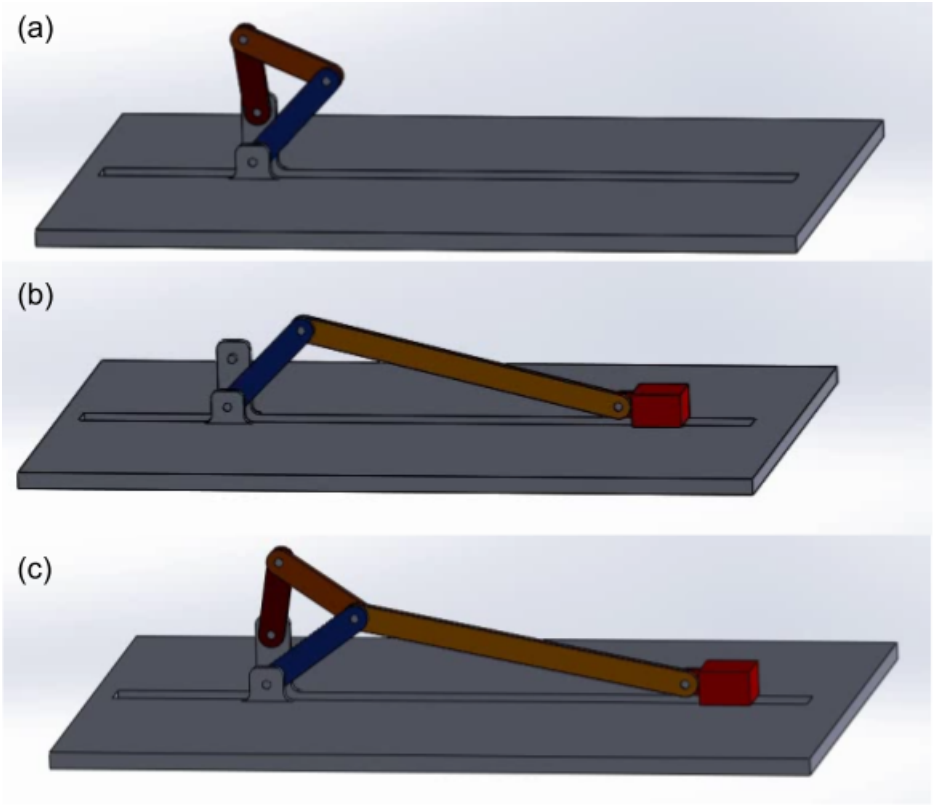


Figure 1.2: Three possible configurations of the mechanism: (a) Four bar, (b) Slider crank and (c) Six bar

available off-the-shelf do not have any data measurement capability and thus can only be used for demonstrations purpose and not as an experimental setup.

In chapter 2 we present thoroughly all the methodologies used in designing and modelling the mechanisms. This includes both the analysis carried out by analytical as well as numerical methods. In the chapter that follows we bring out the various design considerations that has to be taken into account while manufacturing the product. The final two sections present the result(s) and the conclusion(s).

CHAPTER 2

ANALYSIS OF MECHANISMS

2.1 Kinematic analysis

This section presents the methodology employed in order to simulate the kinematics of the various mechanisms. In order to carry out the preliminary analysis of the mechanism, *GeoGebra* an interactive math and geometry software was used. This provided the coarse link lengths required for having the least deviant constant velocity mechanism as shown in Figure 2.4. Once the coarse link lengths were obtained, we developed an in-house Python code that can do iterations on the link lengths so as to minimise any fluctuations in the constant velocity mechanism. The visual output of the Python simulation can be seen in Figure 2.5 while rest of the simulation data is store in a CSV file.

2.1.1 Mathematical formulation

This section brings out the equations of motion associated with the three different mechanisms (Norton, 2011; Uicker *et al.*, 2011). These equations are used to carry out the kinematic and dynamics method theoretically, which then in turn be compared with the numerical solutions obtained by other simulation tools.

Slider crank mechanism

The equations of motion pertaining to the slider crank mechanism (Figure 2.1) are presented in this subsection.

$$\vec{V}_{B/A} = \vec{V}_B - \vec{V}_A \tag{2.1}$$

For link OA we have

$$\vec{V}_{A/O} = \vec{\omega}_{AO} \times \vec{r}_{A/O} = \vec{V}_A \tag{2.2}$$

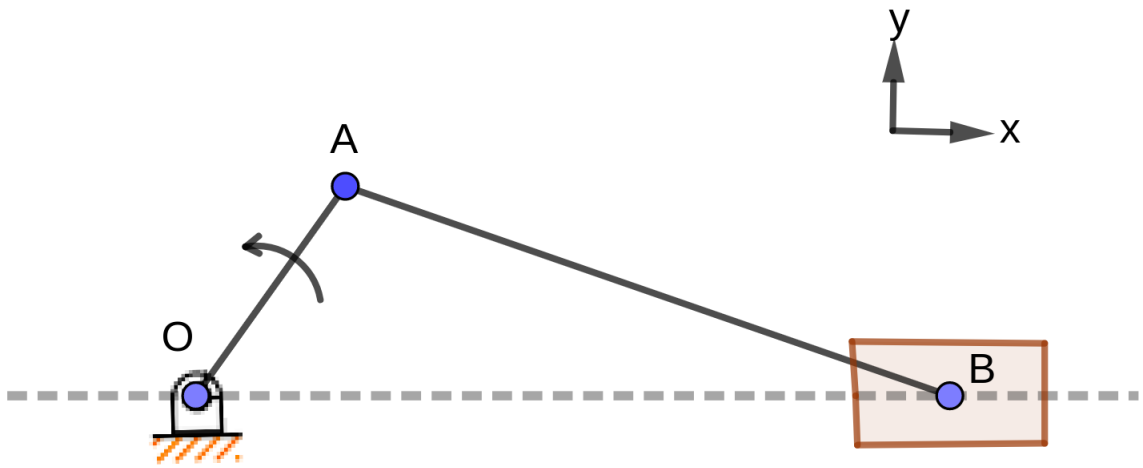


Figure 2.1: Slider crank mechanism

Since 'O' is ground link we have $\vec{V}_O = 0 \implies \vec{V}_{A/O} = \vec{V}_A - \vec{V}_O = \vec{V}_A$. As point B lies on the slider and is moving leftwards it will have velocity along the negative x-direction i.e. $\vec{V}_B = -V_B\hat{i}$; because $\vec{\omega}_{AO} = \omega_{AO}\hat{k}$. For link AB we have

$$\vec{V}_{B/A} = \vec{\omega}_{BA} \times \vec{r}_{B/A} \tag{2.3}$$

Substituting equation 2.3 in equation 2.1 we obtain

$$\vec{\omega}_{BA} \times \vec{r}_{B/A} = \vec{V}_B - \vec{V}_A \tag{2.4}$$

Using equation 2.4 we can find ω_{BA} and V_B. For finding the accelerations we can write

$$\vec{a}_{A/O} = \vec{a}_A = \vec{a}_A^t + \vec{a}_A^n \tag{2.5a}$$

$$\vec{a}_A^t = \vec{\alpha_{A/O}} \times \vec{r_{A/O}} \tag{2.5b}$$

$$\vec{a}_A^n = \vec{\omega_{AO}} \times \vec{\omega_{AO}} \times \vec{r_{A/O}} \tag{2.5c}$$

where, $\vec{a}_A^t$ and $\vec{a}_A^n$ denote the tangential and normal acceleration of point A. As point B lies on the slider, we have $\vec{a}_B = a_B\hat{i}$.

$$\vec{a}_{B/A} = \vec{a}_{B/A}^t + \vec{a}_{B/A}^n = \vec{a}_B - \vec{a}_A \tag{2.6}$$

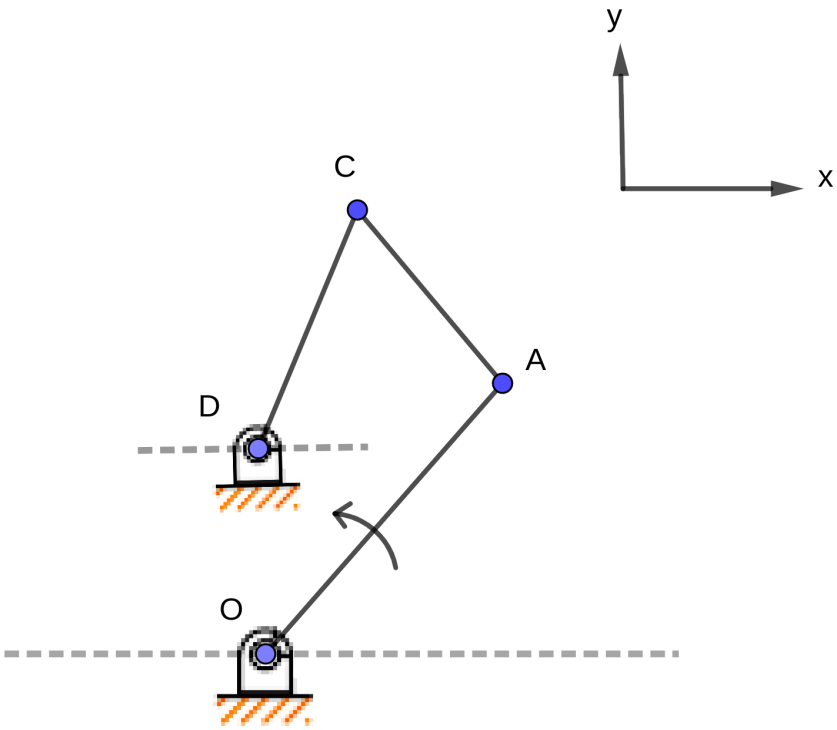


Figure 2.2: Four bar mechanism

Using the above equation we can find α_{BA} and $\vec{a}_B$.

Four bar mechanism (with four revolute joints)

This subsection lists the mathematical equations pertaining to the four bar mechanism (Figure 2.2).

$$\vec{V}_{A/O} = \vec{\omega}_{AO} \times \vec{r}_{A/O} = \vec{V}_A \tag{2.7a}$$

$$\vec{V}_{C/D} = \vec{\omega}_{CD} \times \vec{r}_{C/D} = \vec{V}_C \tag{2.7b}$$

$$\vec{V}_{A/C} = \vec{\omega}_{AC} \times \vec{r}_{A/C} = \vec{V}_A - \vec{V}_C = \vec{\omega}_{AO} \times \vec{r}_{A/O} - \vec{\omega}_{CD} \times \vec{r}_{C/D} \tag{2.8}$$

Using the above equation we can find ω_{CD} and ω_{AC}.

$$\vec{a}_{A/O} = \vec{a}_A = \vec{a}_A^t + \vec{a}_A^n \tag{2.9a}$$

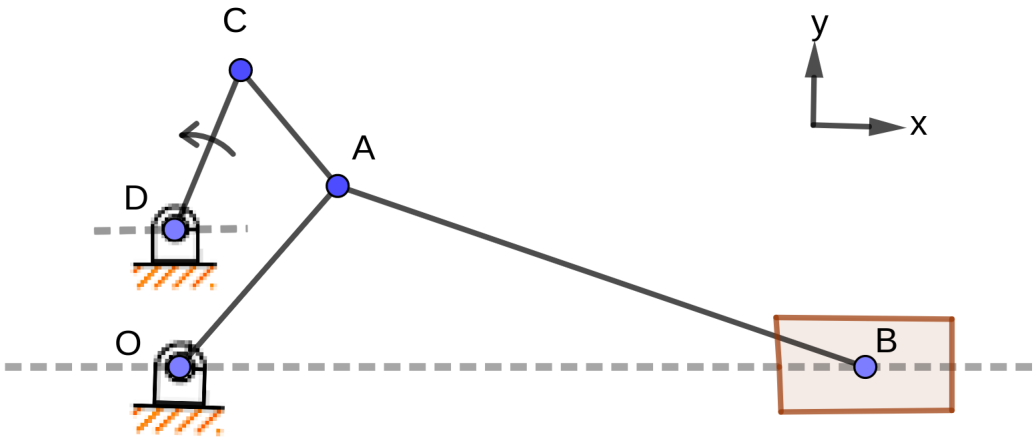


Figure 2.3: Six bar mechanism

$$\vec{a}_{C/D} = \vec{a}_C = \vec{a}_C^t + \vec{a}_C^n \tag{2.9b}$$

$$\vec{a}_{A/C} = \vec{a}_{A/C}^t + \vec{a}_{A/C}^n = \vec{a}_A - \vec{a}_C \tag{2.10}$$

Here $\vec{a}^t = \vec{\alpha} \times \vec{r}$ and $\vec{a}^n = \vec{\omega} \times \vec{\omega} \times \vec{r}$. Using the above equations we can find α_{AC} and α_{CD}

Six bar mechanism

This subsection presents the mathematical equations associated with the six bar mechanism (Figure 2.3).

$$\vec{V}_{C/D} = \vec{\omega}_{CD} \times \vec{r}_{C/D} = \vec{V}_C \tag{2.11}$$

$$\vec{V}_{A/C} = \vec{\omega}_{AC} \times \vec{r}_{A/C} = \vec{V}_A - \vec{V}_C = \vec{\omega}_{AO} \times \vec{r}_{A/O} - \vec{\omega}_{CD} \times \vec{r}_{C/D} \tag{2.12}$$

Using the above equation we can find ω_{OA} and ω_{AC}.

$$\vec{V}_{B/A} = \vec{V}_B - \vec{V}_A \tag{2.13}$$

For link OA we have

$$\vec{V}_{A/O} = \vec{\omega}_{AO} \times \vec{r}_{A/O} = \vec{V}_A \tag{2.14}$$

Since 'O' is ground link we have $\vec{V}_O = 0 \implies \vec{V}_{A/O} = \vec{V}_A - \vec{V}_O = \vec{V}_A$. As point B lies on the slider and is moving leftwards it will have velocity along the negative x-direction i.e. $\vec{V}_B = -V_B\hat{i}$; because $\vec{\omega}_{AO} = \omega_{AO}\hat{k}$. For link AB we have

$$\vec{V}_{B/A} = \vec{\omega}_{BA} \times \vec{r}_{B/A} \tag{2.15}$$

Substituting equation 2.15 in equation 2.13 we obtain

$$\vec{\omega}_{BA} \times \vec{r}_{B/A} = \vec{V}_B - \vec{V}_A \tag{2.16}$$

Using equation 2.16 we can find ω_{BA} and V_B.

For finding the accelerations we can write

$$\vec{a}_{C/D} = \vec{a}_C = \vec{a}^t_C + \vec{a}^n_C \tag{2.17}$$

$$\vec{a}_{A/C} = \vec{a}^t_{A/C} + \vec{a}^n_{A/C} = \vec{a}_A - \vec{a}_C \tag{2.18}$$

Here $\vec{a}^t = \vec{\alpha} \times \vec{r}$ and $\vec{a}^n = \vec{\omega} \times \vec{\omega} \times \vec{r}$. Using the above equations we can find α_{AC} and α_{OA}

$$\vec{a}_{A/O} = \vec{a}_A = \vec{a}^t_A + \vec{a}^n_A \tag{2.19a}$$

$$\vec{a}^t_A = \vec{\alpha_{A/O}} \times \vec{r_{A/O}} \tag{2.19b}$$

$$\vec{a}^n_A = \vec{\omega_{AO}} \times \vec{\omega_{AO}} \times \vec{r_{A/O}} \tag{2.19c}$$

where, $\vec{a}^t_A$ and $\vec{a}^n_A$ denote the tangential and normal acceleration of point A. As point B lies on the slider, we have $\vec{a}_B = a_B\hat{i}$.

$$\vec{a}_{B/A} = \vec{a}^t_{B/A} + \vec{a}^n_{B/A} = \vec{a}_B - \vec{a}_A \tag{2.20}$$

Using the above equation we can find α_{BA} and $\vec{a}_B$.

2.1.2 GeoGebra modelling

This subsection provides an insight into the preliminary analysis done by using off-the-shelf software - GeoGebra for determining initial link lengths (Hohenwarter *et al.*, 2013). First the mechanism was created inside GeoGebra geometry with sliders for varying the link lengths manually. Later the slider velocity was plotted by using graphical analysis of the mechanism inside GeoGebra itself (as shown in Figure 2.4). In order to find the required link lengths yielding constant velocity mechanism, each of the link lengths were varied to get the constant velocity mechanism with least variation while keeping the others fixed. The manual iterations are shown in Table 2.1.

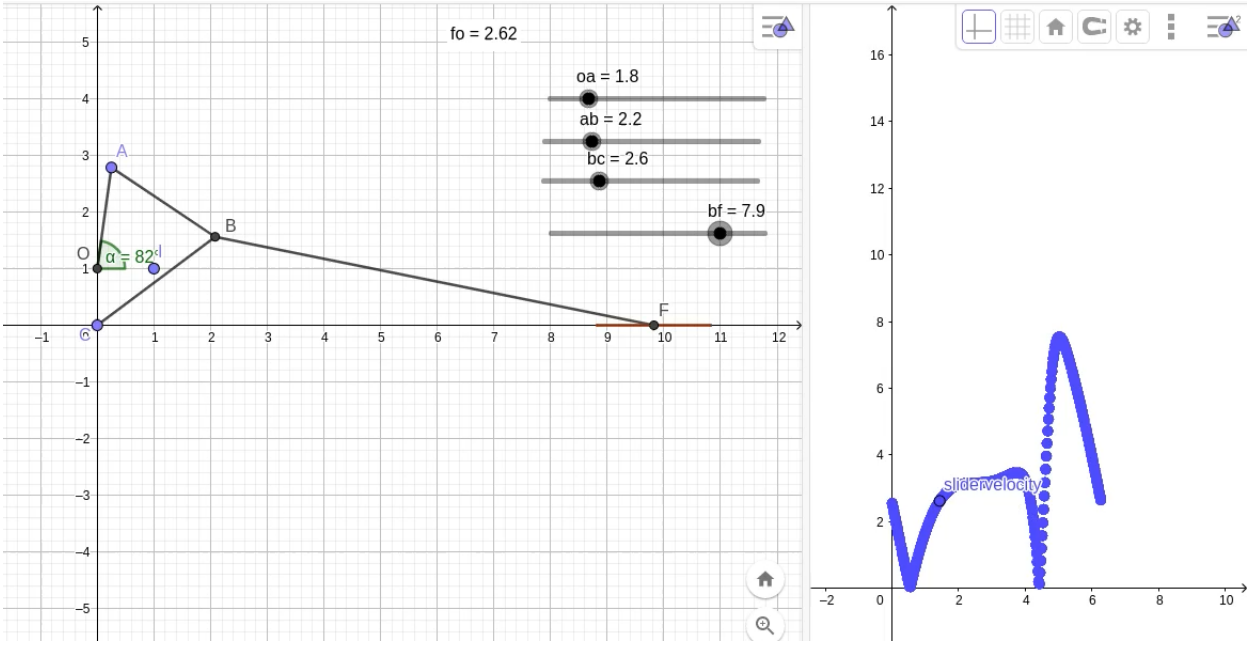


Figure 2.4: Snapshot from GeoGebra showing the mechanism (left) and the velocity profile (right)

2.1.3 Python simulation

This section deals with further refining the preliminary links lengths which were obtained from GeoGebra modelling. The initial lengths are fed to an in-house Python script (see appendix) to get optimum link lengths for nearly constant slider velocity (Python Core Team, 2018). The algorithm performs the iterations by varying each of the link by step size of $0.1 \times$ unit length. It also stores the peak to valley distance for each of the obtained velocity profiles which can be used to select the link lengths giving minimum velocity variation for the constant velocity range. Figure 2.5 shows the mechanism animation as

OC	OA	AB	BC	BF	Mean slider velocity	Peak to valley of slider velocity	ω of crank
1	2	1.6	2	6	2.25	0.5	2
1	2	1.4	2	6	2.28	0.81	2
1	2	1.2	2	6	2.37	1.24	2
1	2	2	2	6	2.385	0.77	2
1	2.2	1.6	2	6	2.31	0.83	2
1	1.8	1.6	2	6	2.375	0.75	2
1	2	1.6	2.2	6	2.545	0.37	2
1	2	1.6	1.8	6	1.92	0.76	2
1	2	1.6	2.4	6	2.87	0.26	2
1	2	1.6	2.6	6	3.2	0.25	2
1	2	1.6	2.8	6	3.52	0.36	2
1	2	1.6	2.6	7	3.135	0.15	2
1	2	1.6	2.6	7.5	3.2	0.12	2
1	2	1.6	2.6	8	3.24	0.09	2

All lengths are in *units*, velocity are in $units/s$, angular velocity ω is in rad/s.

Table 2.1: GeoGebra results

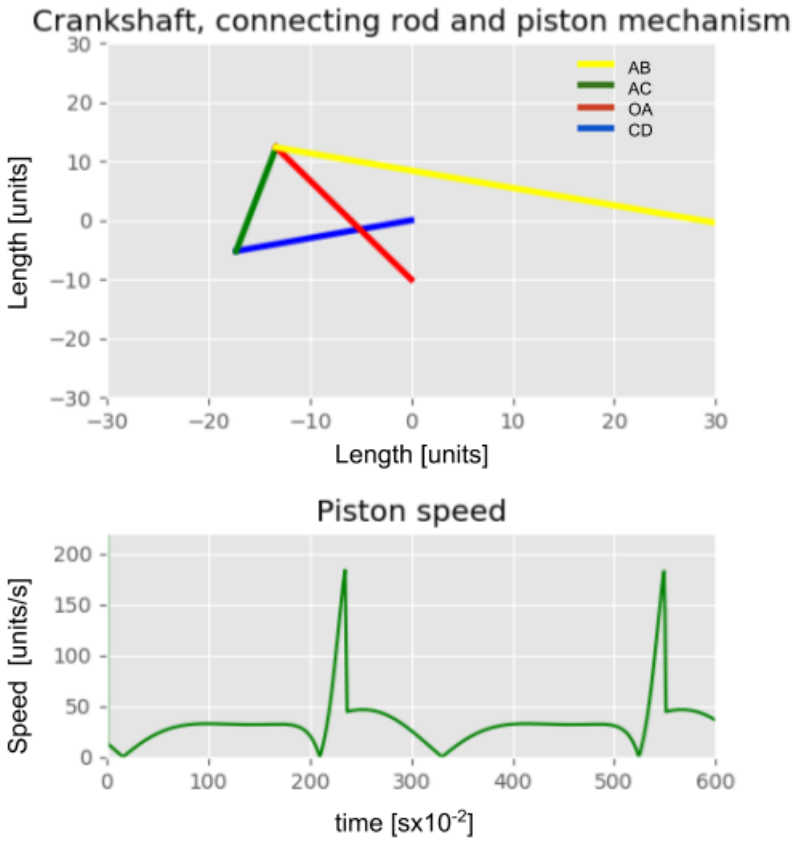


Figure 2.5: Output window from Python for kinematic simulation

well as the velocity plot having a near constant velocity for an interval.

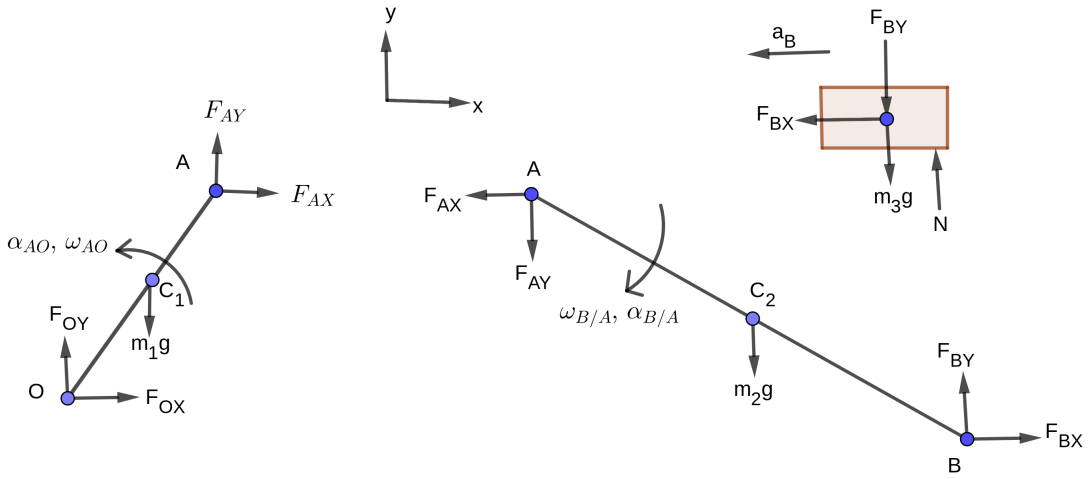


Figure 2.6: Free Body Diagram for Slider Crank Mechanism

2.2 Dynamic analysis

For performing the dynamic analysis of the mechanism we used MathWorks Simscape Mechanical package. A 3D model was created in the package itself and analysed the same for various forces, torques, acceleration and velocities by taking the material as aluminium (Figure 2.9, **??**). This analysis is necessary in order to ease the process of selection of actuator, material and the sensor system.

2.2.1 Mathematical formulation

This subsection brings out the mathematical equations for performing dynamic analysis of all the three possible mechanism (Ghosh and Mallik, 2002).

Slider crank mechanism

Force and Torque balance for link *OA* using Figure 2.6

$$F_{OX} + F_{AX} = 0 \tag{2.21a}$$

$$F_{OY} + F_{AY} = m_1 g \tag{2.21b}$$

$$I_{AO}\vec{\alpha}_{AO} = \vec{r}_{C_1O} \times m_1 g(-\hat{j}) + \vec{r}_{A/O} \times \vec{F}_A \tag{2.21c}$$

where, $\vec{F}_A = \vec{F}_{AX}\hat{i} + \vec{F}_{AY}\hat{j}$. Similarly for link AB we have

$$F_{AX} = F_{BX} \tag{2.22a}$$

$$F_{BY} = F_{AY} + m_2 g \tag{2.22b}$$

$$I_{BA}\vec{\alpha}_{B/A} = \vec{r}_{C_2/A} \times m_2 g(-\hat{j}) + \vec{r}_{B/A} \times \vec{F}_B \tag{2.22c}$$

where, $\vec{F}_B = \vec{F}_{BX}\hat{i} + \vec{F}_{BY}\hat{j}$.

Force balance for the slider

$$F_{BY} + m_3 g = N \tag{2.23a}$$

$$m_3 a_B = F_{BX} \tag{2.23b}$$

Four bar mechanism (with four revolute joints)

Force and Torque balance for link *OA* using Figure 2.7

$$\sum \vec{F}_X = 0 \tag{2.24a}$$

$$\sum \vec{F}_Y = 0 \tag{2.24b}$$

$$\sum \vec{T} = I_1 \vec{\alpha}_{AO} \tag{2.24c}$$

Force and Torque balance for link *AC* using Figure 2.7

$$\sum \vec{F}_X = 0 \tag{2.25a}$$

$$\sum \vec{F}_Y = 0 \tag{2.25b}$$

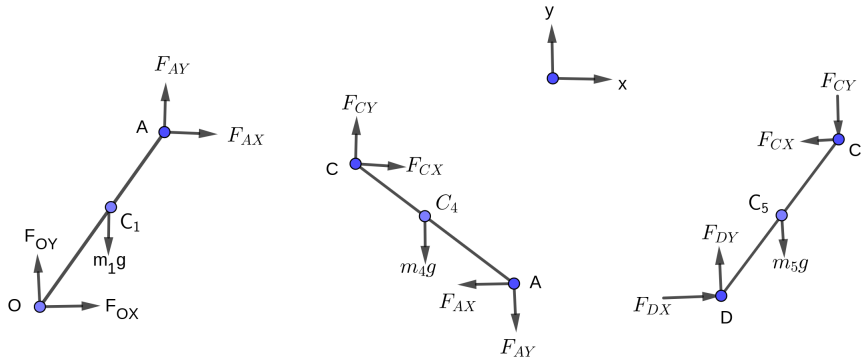


Figure 2.7: Free Body Diagram for four-bar mechanism

$$\sum \vec{T} = I_4 \vec{\alpha}_{AC} \tag{2.25c}$$

Force and Torque balance for link *CD* using Figure 2.7

$$\sum \vec{F}_X = 0 \tag{2.26a}$$

$$\sum \vec{F}_Y = 0 \tag{2.26b}$$

$$\sum \vec{T} = I_5 \vec{\alpha}_{CD} \tag{2.26c}$$

Six bar mechanism

Force and Torque balance for all links using Figure 2.8

$$\sum \vec{F}_X = 0, \sum \vec{F}_Y = 0, \sum \vec{T} = I\vec{\alpha} \tag{2.27}$$

$$F_{A_1X} + F_{A_2X} + F_{A_4X} = 0 \tag{2.28a}$$

$$F_{A_1Y} + F_{A_2Y} + F_{A_4Y} = 0 \tag{2.28b}$$

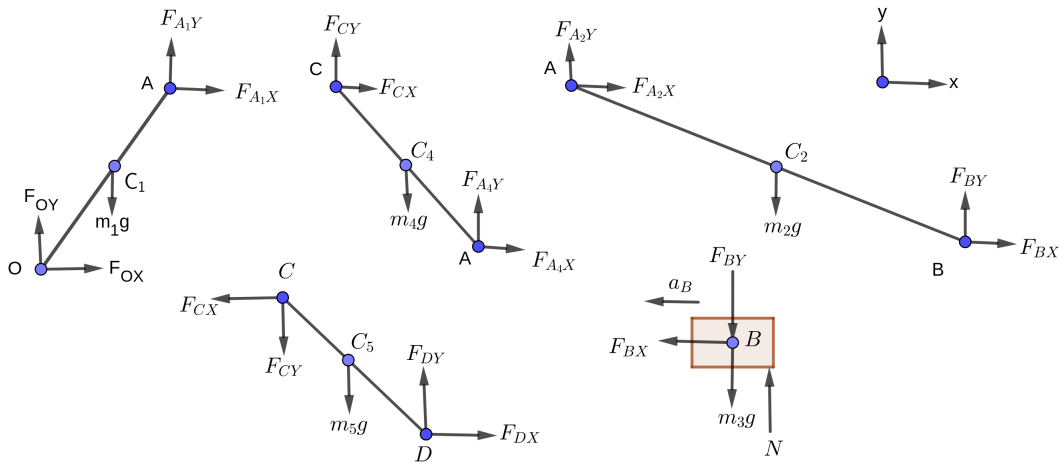


Figure 2.8: Free Body Diagram for six-bar mechanism

Force balance for Slider using Figure 2.8

$$F_{BY} + m_3 g = N \tag{2.29a}$$

$$m_3 a_B = F_{BX} \tag{2.29b}$$

2.2.2 Using commercial software: MathWorks Simscape

This sections deals with the dynamic analysis of the mechanism using Mathworks Simscape. A MathWorks Simscape model (Figure 2.9) was created based on the link lengths that were obtained from the kinematic analysis.The link width is taken as $3cm$ and link thickness as $1cm$ while taking aluminum as material for all links and slider. (MAT, 2017)

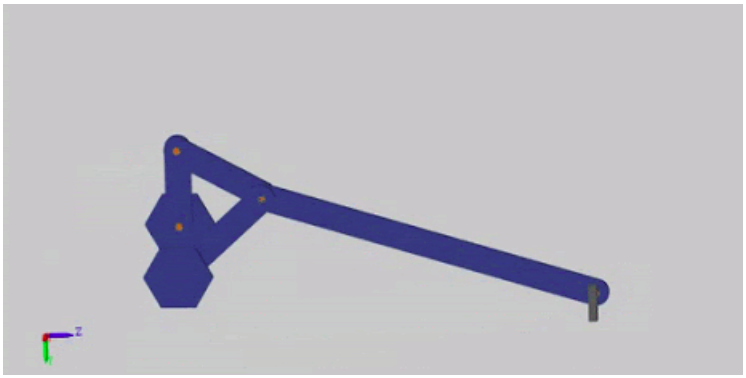
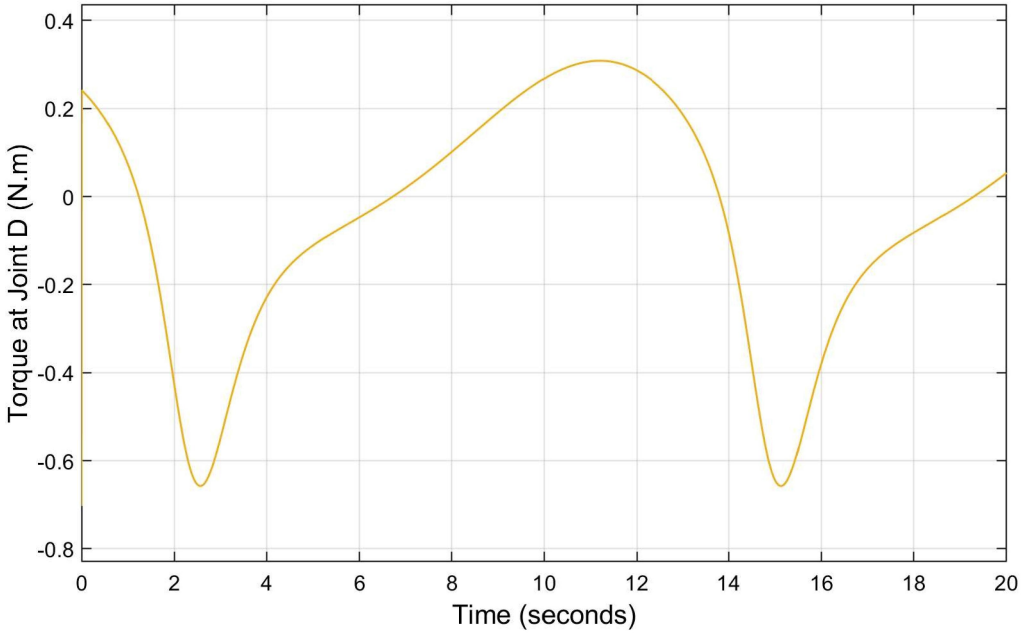
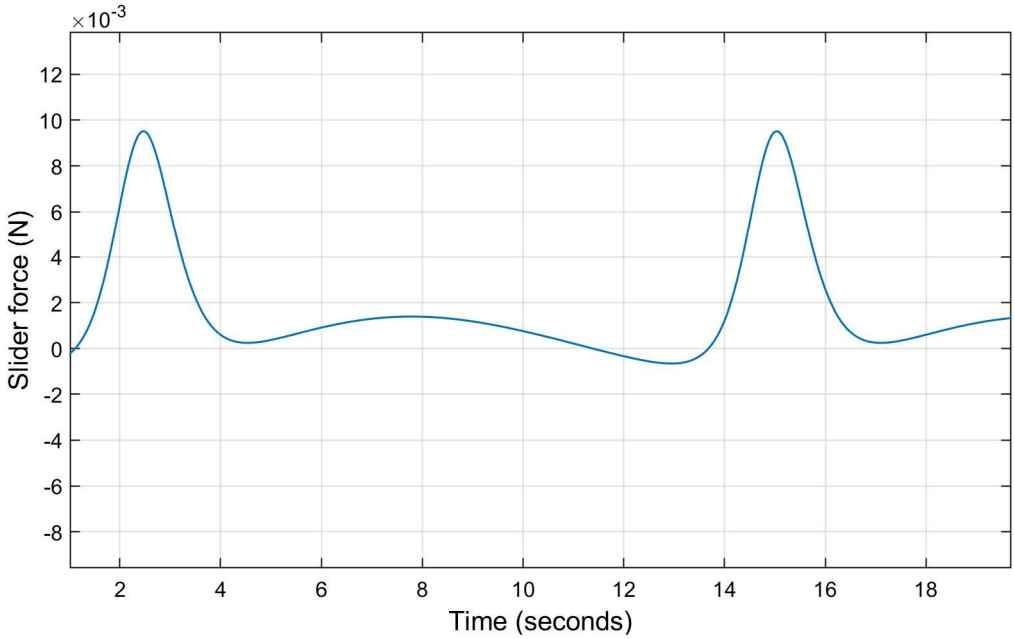


Figure 2.9: CAD model in MATLAB Simscape



(a) Input torque



(b) Slider force

Figure 2.10: Results obtained from Simscape

CHAPTER 3

MANUFACTURING

3.0.1 Sensor selection

Based on the results of Kinematic, Dynamic analysis using MathWorks Simscape, We selected 2-Phase Incremental Optical Rotary Encoder to detect angular velocity and angular acceleration of links, 3-Axis Accelerometer is to detect linear acceleration and linear velocity of slider.

2-Phase Incremental Optical Rotary Encoder has 600 Pulse Per Revolution (PPR).

3-Axis Accelerometer has range of $-16g$ to $16g$, with a sampling rate of 8kHz.

3.0.2 Manufacturing plan

This section focuses on the materials and fabrication processes that will be used for carrying out the manufacturing in the second phase of the project. Figure 3.2 gives complete insight on the various stages involved in manufacturing the experiment design.

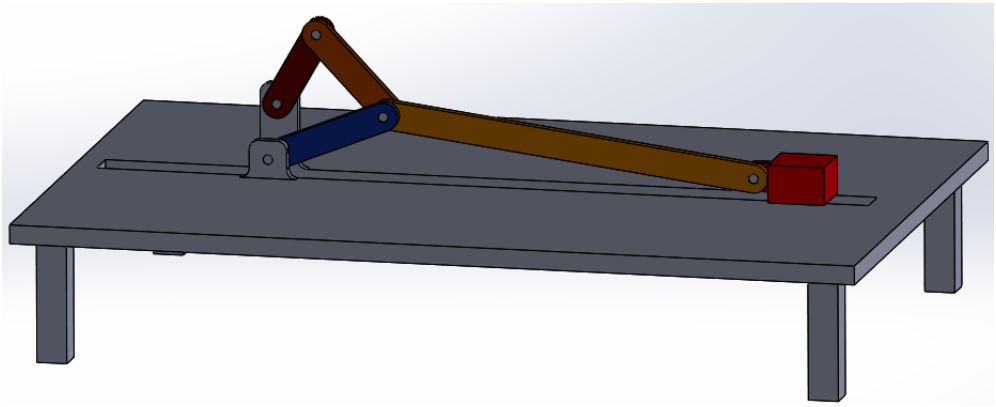


Figure 3.1: 3D Model of the experiment on (table)

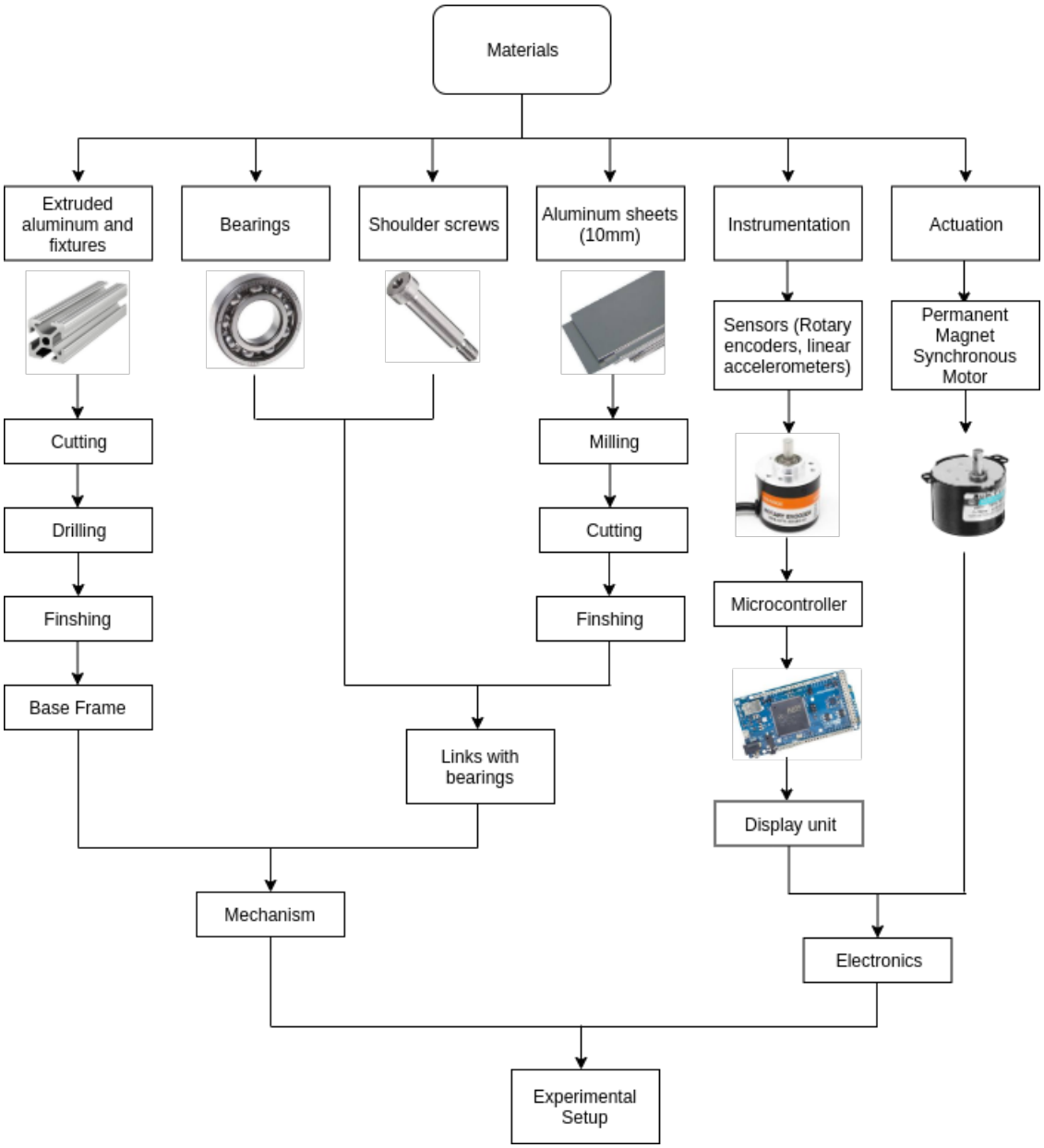


Figure 3.2: Flowchart showing manufacturing

CHAPTER 4

RESULTS

This final section presents obtained results. By utilizzing the kinematic analysis we obtained the link lengths resulting in constant velocity mechanism as:

Link	Link length (cm)
OA	13
CD	9
AC	11
AB	39.5
DO	5

Table 4.1: Final link lengths

Furthermore plots for slider velocity and acceleration were obtained using Simscape.

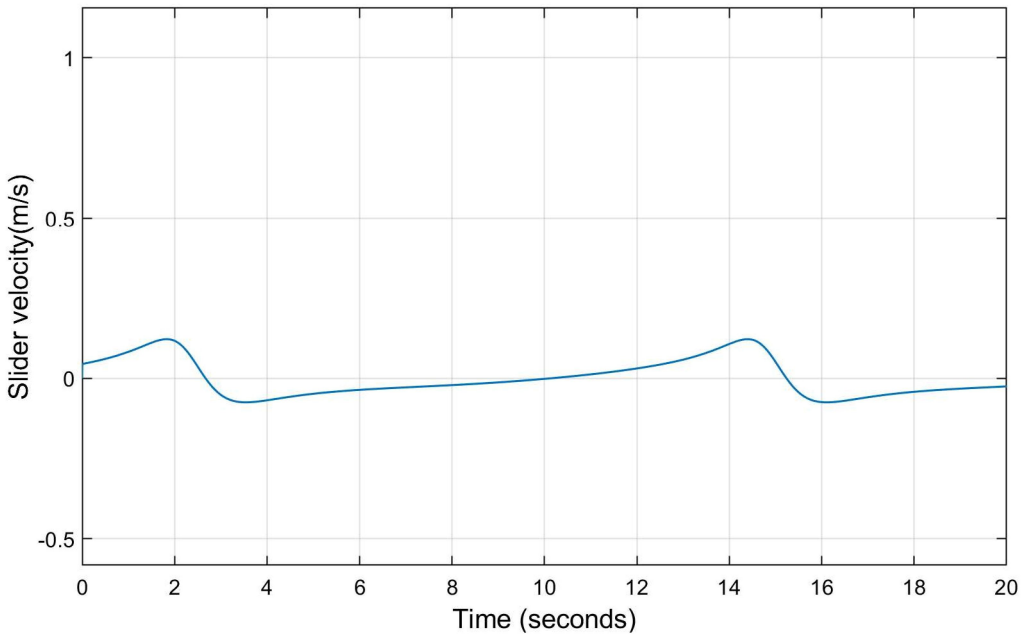


Figure 4.1: Slider Velocity

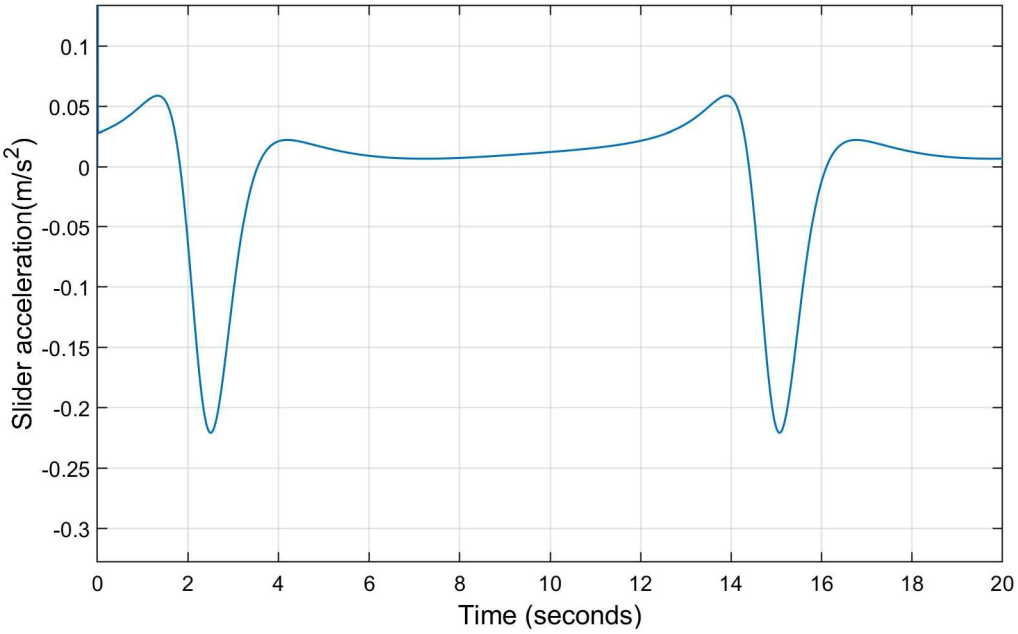


Figure 4.2: Slider acceleration

CHAPTER 5

CONCLUSION AND FUTURE WORK

The complete analysis of the mechanism(s) was performed using applicable method and software. 3D CAD model of the mechanism was created to facilitate better visualisation. Various design considerations were addressed and manufacturing workflow and plan were accomplished as a prerequisite to manufacturing. Also, the BoM was prepared and materials acquisition from the vendor is in process.

In the near future we shall start the manufacturing process based on the work that has been already accomplished by now. After completing the manufacturing rigorous testing will be done on the mechanism and corrections be made accordingly. In the end we will work on preparing the documentation for the product.

APPENDIX A

CODE LISTING

The code used for carrying out this work is presented here. Due due to the high volume of the code involved in this project, it is also hosted on GitHub for clearity at https://github.com/AakashSYadav/BTP with complete documentation.

A.1 Six-bar mechanism (Python)

```
# http://firsttimeprogrammer.blogspot.com/2015/02/crankshaft-
    connecting-rod-and-piston.html

# import matplotlib.animation as animation
# import matplotlib.pyplot as plt
# from sympy import symbols, Eq, solve
import numpy as np
import time
import math
import csv

# plt.style.use('ggplot')
link_a_pivot = (0,-10)
link_p_pivot = (0,0)
# x, y, h = symbols('x y h')

class My_mechanism(object):

    # The init function
    def __init__(self,a,b,p,q,omega):
        self.a = a                              # Rod a
        self.b = b                              # Rod b
        self.p = p                              # Rod p
        self.q = q                              # Rod q
        self.omega = omega            # Angular speed of rod a in rad
    /s
```

```
25          self.theta0 = 0                 # Initial angular position of
    rod a theta()
26          self.set0 = (0,-100)            # for selecting one of the two
    solutions
27          self.k = 0.00001                # Initial time for animation
28          self.c_position = []            # Piston position for animation
29          self.c_speed = []               # storing piston speed
30          self.c_time = []                # storing the time intervals
31          self.pos_old = 0  # old position of the piston for c_dot
    calculation
32
33      # Angular position of rod a as a function of time
34      def theta(self,t):
35          # if not all(t):
36          #     raise ValueError('Each time t must be greater than 0')
37          theta = self.theta0 + self.omega*t
38          return theta
39
40      def rod_p_position(self,t):
41          p_y = self.p*np.sin(self.theta(t))
42          p_x = self.p*np.cos(self.theta(t))
43          return p_x,p_y
44
45      def rod_q_position(self,t):
46          px,py = self.rod_p_position(t)
47          ax,ay = link_a_pivot
48          c = ((self.p**2-px**2-py**2) - (self.a**2-ax**2-ay**2)) /
    (2*(ax-px))
49          d = (py-ay)/(ax-px)
50          D = (d*(px-c)+py)**2 - (1+d**2)*(py**2 + (px-c)**2 - self.p
    **2)
51          if D<0:
52              print("complex")
53              raise Exception('complex')
54          D2 = D**0.5
55          y0 = (d*(px-c)+py+D2)/(1+d**2)
56          y1 = (d*(px-c)+py-D2)/(1+d**2)
57          x0 = c+d*y0
58          x1 = c+d*y1
59          set1=(x0,y0)
```

```
60          set2=(x1,y1)
61          # print(set1,set2)
62
63          dist1 = math.hypot(set1[0] - self.set0[0], set1[1] - self.
    set0[1])
64          dist2 = math.hypot(set2[0] - self.set0[0], set2[1] - self.
    set0[1])
65          if dist1<dist2:
66              self.set0=set1
67              return set1[0],set1[1]
68          else:
69              self.set0=set2
70              return set2[0],set2[1]
71
72      def piston_position(self,t):
73          q_x,q_y = self.rod_q_position(t)
74          h0 = q_x+(self.b**2 - (q_y-link_a_pivot[1])**2)**0.5
75          return h0
76
77      # Piston speed
78      def c_dot(self,t):
79          c_x = self.piston_position(t)
80          # c_dot = (c_x-self.pos_old)/0.05
81          c_dot = abs(c_x-self.pos_old)/0.01  # dt = 0.01
82          self.pos_old = c_x
83          return c_dot
84
85      def velocity_params(self):
86          while self.k<3.5: # to get one complete cycle
87              speed = self.c_dot(self.k)
88              self.k+=0.01
89
90              """
91              Uncomment the below 3 lines and comment the rest
92              to collect data for a single set of params using "c_out_2.
    py"
93              """
94              # with open('oneP_best2.csv', mode='a') as label_file:
95              #     label_writer = csv.writer(label_file, delimiter=',',
     quotechar='"', quoting=csv.QUOTE_MINIMAL)
```

```
            #     label_writer.writerow([self.k,speed])

            if self.k>0.4 and self.k<2.1:# store the ROI
                self.c_speed.append(speed)
                self.c_time.append(self.k)

        # print(self.c_speed)
        maxi = []
        mini = []
        for i in range(1,len(self.c_speed)-1):
            # local maxima
            if self.c_speed[i]>self.c_speed[i+1] and self.c_speed[i]>
    self.c_speed[i-1]:
                maxi.append(self.c_speed[i])
                # print("max",self.c_speed[i])
            # local minima
            if self.c_speed[i]<self.c_speed[i+1] and self.c_speed[i]<
    self.c_speed[i-1]:
                mini.append(self.c_speed[i])
                # print("min",self.c_speed[i])
        # print(self.a,self.b,self.p,self.q,max(maxi),min(mini))
        with open('data_best.csv', mode='a') as label_file:
           label_writer = csv.writer(label_file, delimiter=',',
    quotechar='"', quoting=csv.QUOTE_MINIMAL)
            label_writer.writerow([self.a,self.b,self.p,self.q,max(
    maxi),min(mini)])
        mini.clear()
        maxi.clear()
        self.c_speed.clear()
        self.time.clear()
```

```
from crankshaft import *
import time

initial_time = time.time()
# #a,b,p,q,omega
# best
for a in range(20,35):
    for b in range(70,80):
        for p in range(15,25):
```

```
            for q in range(15,30):
                m = My_mechanism(a,b,p,q,2)
                try:
                    m.velocity_params()
                except Exception:
                    continue

print("Processing time : ",time.time()-initial_time,"s")
```

REFERENCES

1. (2017). *MATLAB version 9.3.0.713579 (R2017b)*. The Mathworks, Inc., Natick, Massachusetts.

2. **A. Ghosh** and **A. K. Mallik**, *Theory of mechanisms and machines*. Affiliated East-West Press Private Limited, 2002.

3. **M. Hohenwarter**, **M. Borcherds**, **G. Ancsin**, **B. Bencze**, **M. Blossier**, **A. Delobelle**, **C. Denizet**, **J. Éliás**, **A. Fekete**, **L. Gál**, **Z. Konečný**, **Z. Kovács**, **S. Lizelfelner**, **B. Parisse**, and **G. Sturr** (2013). GeoGebra 4.4. http://www.geogebra.org.

4. **R. L. Norton**, *Kinematics and dynamics of machinery*. McGraw-Hill Higher Education, 2011.

5. **Python Core Team** (2018). *Python: A dynamic, open source programming language*. Python Software Foundation. URL https://www.python.org/.

6. **J. J. Uicker**, **G. R. Pennock**, **J. E. Shigley**, *et al.*, *Theory of machines and mechanisms*, volume 1. Oxford University Press New York, NY, 2011.