Copy No: 01

SUMMER TRAINING REPORT

ON

"PREDICTION OF POSITION AND VELOCITY OF A PARATROOPER IN 2D"

Prepared by	Madhvan Sharma B. Tech, 4th Semester Delhi Technological University, New Delhi
Supervised by	Sandip Kumar, Scientist 'D' 30/6/2021
Approved by	Mahendra Pratap, Scientist 'F' 30/6/21 Group Director, Aircrew Ejection Parachute

DRDO

AERIAL DELIVERY RESEARCH & DEVELOPMENT ESTABLISHMENT

P.B. # 51, STATION ROAD, AGRA CANTT-282001.

19 April 2021 to 30 June 2021

REPORT DOCUMENTATION PAGE

1.	Security Classification: Open	2.	Distributions: HRD ADRDE Para Group ADRDE
3.	Report status: New	4.	Report Type: Training Document
5.	Report No: DRDO – ADRDE – SE – 2021	6.	Part or Volume No: I
7.	Title and subtitle: Prediction of Position and Velocity of a Paratrooper in 2D	8.	Control No: Open
9.	No of Pages: **18**	10.	Project No: **NA**
11.	Period Covered (if progress report): **-**		
12.	Personal Author (s): **Madhvan Sharma**		
13.	Affiliation of Author (s): **DTU, New Delhi**		
14.	Document Issuing Authority (s): **ADRDE**		
15.	Origination Unit: ADRDE, PO Box 51, Station Road, Agra Cantt Agra		
16.	Sponsor(s) Name: NA		
17.	**Training duration:** 19 April 2021 to 30 June 2021	18.	Publication/Issue date: **June 2021**
19.	Publisher/Distributor: DTU		
20.	Form of Distribution: Hard Copy: Yes Digital: No		
21.	No. of References: 04		
22.	Abstract: This project report is describes the brief introduction about parachute-payload trajectory and velocity prediction in 2 dimension.		
23.	Keywords/Descriptors: Software, Python, parachute, Position and Velocity, Paratrooper		
24.	Supplementary Elements: Nil		

AMENDMENT RECORD SHEET

The incorporation of each amendment to this document is to be recorded by filling the columns against each amendment number:

S. No.	Issue/Amendment No.	Date of Issue/ Amendment	Amended page nos. (for amendments)	Signature of the 'Approving authority'	Remarks

TABLE OF CONTENTS

CERTIFICATE

This is to certify that the Project Report compiled by Mr. Madhvan Sharma which is entitled "Prediction of Position and Velocity of a Paratrooper in 2D" is an authentic record of the effort carried out by him during the period of his summer training from 19th April 2021 to 30th June 2021. The report is aimed towards the partial fulfillment of the requirement of training certificate duly accredited by Aerial Delivery Research and Development Establishment (ADRDE) Agra, under the guidance of Mr. Sandeep Kumar, scientist 'D'.

Mr. Sandip Kumar
Scientist D
ADRDE, Agra

ACKNOWLEDGEMENT

Through this acknowledgement, I express my sincere gratitude to all those people who have been associated with this project and have helped me with it and made it a worthwhile experience.

Firstly, I extend my thanks to the various people who have shared their opinions and experiences through which I received the required information crucial for my report.

I am highly indebted to **Sh Sandip Kumar, Scientist 'D'** and **Sh Mahendra Pratap, Scientist 'F'** my project guides, and their team for guidance and constant supervision as well as for providing necessary information regarding the project execution and also for their support in completing the project on time.

My thanks and appreciations also goes to my colleagues in developing the prototype parachute systems for payload delivery and people who have willingly helped me out with their best abilities.

Finally, I express my sincere thanks to **Sh A. K. Saxena, Outstanding Scientist & Director ADRDE** who has permitted me to carry out this project.

PREFACE

The purpose of a parachute is to decelerate and provide stability to a payload during flight. Paratrooper jump from aircraft with help of parachute for defined mission operation, therefore, parachute must be reliable and land on ground as desired position. Prediction of trajectory of paratrooper is a very complex phenomenon due to uncertainty of external wind, gust and atmospheric conditions presents. The aerodynamic and stability characteristics of the parachute system are governed by the geometry of the parachute and these external forces. The effects of deployment and opening force are major events occurs and responsible for integrity of the paratrooper. The design is carried out to overcome these developed opening characteristics and accordingly sufficient margin of safety selected.

A paratrooper is a military parachutist and trained for an operation, and usually functioning as part of an airborne force. Military parachutists (troops) and parachutes were first used on a large scale during World War II for troop distribution and transportation but nowadays paratroopers are often used in surprise attacks, to seize strategic objectives such as airfields or bridges.

In this project an effort has been made to predict the positioning and velocity of a paratrooper without considering the external forces. This report first gives a brief introduction about the parachute design forces, trajectory parameters and a computer program has been developed in python to plot the trajectory of a paratrooper.

ABOUT AERIAL DELIVERY RESEARCH & DEVELOPMENT ESTABLISHMENT

ADRDE (Aerial Delivery Research and Development Establishment) is one of the most reputed research establishments in the field of lighter than air technologies (i.e. Aerostat, Balloon barrage system, Floatation system, parachutes etc). This establishment is an ISO 9001:2015 certified and a part of Defence Research and Development Organization (DRDO). ADRDE founded in year 1967 keeping in mind to develop parachute system. There are 350 employees including 150 officers handling the various projects. ADRDE is working as technology development and testing centre where the various systems designed and validated through testing and trials before induction. Now our country has self-reliance in development of parachutes for various aero dynamics and aerospace applications and saved foreign exchange. During the last three decades, ADRDE has successfully completed many projects on man carrying parachute, cargo and heavy equipment dropping systems, aircraft brake parachute, delivery parachute systems, recovery parachute, inflatable structures and related equipment’s etc.

Over a span of 30 years, more than 88 products development by ADRDE have gone into bulk production. It may be said that efforts at ADRDE have brought the appreciable level of self-reliance in the field of parachute and allied systems with the latest state of art.

1. INTRODUCTION

The purpose of a parachute is to decelerate and provide stability to a payload in flight. The aerodynamic and stability characteristics of the parachute system are governed by the geometry of the parachute as such careful consideration is paid to this in the design process. The effects of deployment and opening force are critical in the safe operation of the parachute and the integrity of the payload. The opening characteristics also feature heavily in the selection of geometry and other parameters in the design process.

All the components of the parachute are made of textile materials and their confluence joints are textile loops or metal components depending on the load applied on it. The main parts of a parachute are shown in Figure 1.

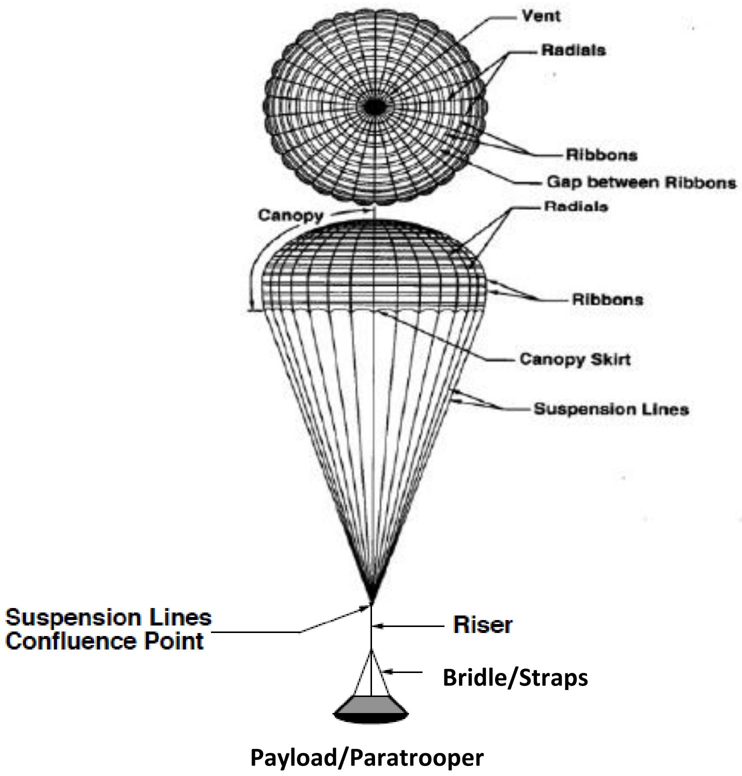

Figure 1: Main parts of a parachute system

There are various types of parachutes like round canopy, square canopy, ribbon type, solid round canopy, spherical canopy etc. A paratrooper uses a round or ram air type parachute (see Figure 2). Strength and reliability are the basic requirements of textile fabric. Parachute strength has been estimated through design, various testing and trials. Many of the unforeseen eventualities will require rigorous testing. For designing a parachute following are required:

(i) Mass of payload (M)

(ii) Altitude of dropping the parachute (ρ) and

(iii) Terminal velocity during landing (v)

A payload may be luggage, aircraft or jumper depends of application, types of parachute are selected. A paratrooper is a military parachutist i.e. someone trained to jump into an operation, and usually functioning as part of an airborne force. Military parachutists (troops) and parachutes were first used on a large scale during World War II for troop distribution and transportation. Paratroopers are often used in surprise attacks, to seize strategic objectives such as airfields or bridges. To summarize, calculation of parachute deployment, inflation, and deceleration requires the numerical solution to the equations of motion for a viscous, turbulent, separated airflow. To make a mathematical model that is feasible, a simplified method is presented in this report to predict the trajectory of a paratrooper.

Figure 2: Paratrooper with parachute during landing

2. SOFTWARE AND TECHNOLOGIES USED

2.1 PYTHON 3.9

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Rather than having all of its functionality built into its core, Python was designed to be highly extensible (with modules). This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

2.2 GEKKO Optimization Suite

GEKKO is a Python package for machine learning and optimization of mixed-integer and differential algebraic equations. It is coupled with large-scale solvers for linear, quadratic, nonlinear, and mixed integer programming (LP, QP, NLP, MILP, MINLP). Modes of operation include parameter regression, data reconciliation, real-time optimization, **dynamic simulation,** and nonlinear predictive control. GEKKO is an object-oriented Python library to facilitate local execution of AP Monitor.

2.3 NumPy

NumPy is the fundamental package for scientific computing with Python. It is the most used package for a computer engineer wherever long and complex calculations are required. NumPy is an open source project aiming to enable numerical computing with Python. It was created in 2005, building on the early work of the Numeric and Numarray libraries. Nearly every scientist working in Python draws on the power of **NumPy**.

NumPy brings the computational power of languages like C and Fortran to Python, a language much easier to learn and use. With this power comes simplicity: a solution in NumPy is often clear and elegant.

2.4 MATPLOTLIB (Visualization with Python)

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib. Matplotlib was originally written by John D. Hunter. Since then it has an active development community and is distributed under a BSD-style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012 and was further joined by Thomas Caswell.

2.5 IDLE

IDLE (short for Integrated Development and Learning Environment) is an integrated development environment for Python, which has been bundled with the default implementation of the language. It is packaged as an optional part of the Python packaging with many Linux distributions. It is completely written in Python and the Tkinter GUI toolkit (wrapper functions for Tcl/Tk).

IDLE is intended to be a simple IDE and suitable for beginners, especially in an educational environment. To that end, it is cross-platform, and avoids feature clutter.

3. PARAMETERS AND FORMULAS USED

3.1 Parameters

We will be only considering these parameters:

- Time at which paratrooper pulls the parachute
- Altitude of the Airplane
- Velocity of the Airplane
- Drag coefficient while Free fall
- Drag coefficient while Parachute is open
- Gravitational constant
- Mass of the paratrooper and parachute

3.2 Formulas

Assumptions made in the below formulas:

- If the final position is to the **right** of the initial position, then the horizontal position is considered **positive**.
- If the final position is to the **top** of the initial position, then the vertical position is considered **positive.**

Basically, we have **five** variables which are need to be found:

(i) $\mathbf{x}$ (Horizontal position)

(ii) $\mathbf{y}$ (Vertical position)

(iii) $\mathbf{v_x}$ (Horizontal velocity)

(iv) $\mathbf{v_y}$ (Vertical velocity)

(v) $\mathbf{v}$ (Resultant velocity)

Since, there are five variables to be calculated. We require five equations/formulas to solve them.

These are as follows:

3.2.1 Instantaneous Velocity

For any given moment, vertical and horizontal velocity is given by equation (1) and (2).

$$v_x = \lim_{\Delta t \to 0} \frac{\Delta x}{\Delta t} = \frac{dx}{dt} \tag{1}$$

and

$$v_y = \lim_{\Delta t \to 0} \frac{\Delta y}{\Delta t} = \frac{dy}{dt} \tag{2}$$

3.2.2 Drag Coefficient

It is known that air resistance forces are proportional to **v** and $\mathbf{v^2}$. Many sources attempt to treat all of the non-velocity influences on the drag force separately. So, we have a **constant/variable** that represents the influence of the shape of the object on the air resistance (drag) force, a constant/variable that represents the effect of the density of the fluid on the air resistance force, etc. This results in a very large, very complicated looking expression for the air resistance force. My choice is to lump all of these other factors into one constant - let's call it "**c**". So, the shape of the object influences the value of "c", the density of the fluid influences "c", and so on.

This means that we can write the air resistance force (or drag force) as

$\mathbf{F_{drag} = cv}$ for very small, slow objects,

or

$\mathbf{F_{drag} = cv^2}$ for human-size objects

Therefore, for the scope of this project, we will be considering $\mathbf{F_{drag} = cv^2}$**.** Simplifying this formula, we get equation (3) and (4).

$$m \frac{dv_x}{dt} = -cv_x^2 \tag{3}$$

and

$$m \frac{dv_y}{dt} = cv_y^2 - mg \tag{4}$$

3.2.3 Resultant Velocity

The resultant velocity is given by equation (5).

$$v = \sqrt{v_x^2 + v_y^2} \tag{5}$$

Now we have **5 equations** and **5 variables** (v_x, v_y, v, x, and y). Thus, it can be written as codes in Python programming language to get the results.

4. SIMULATION

For executing the code, taking an example input data and displaying results from that input data.

4.1 Input Data

We will be predicting the position and velocity of a paratrooper in two dimensions (horizontal and vertical) from the time of initial jump through the first 90 seconds. At 60 seconds after the jump, the paratrooper pulls the chute and the drag coefficient increases to slow the descent. The airplane is flying at a constant altitude of 5000 m and at velocity of 50 m/s when the paratrooper jumps. The drag coefficient is 0.2 $N\text{-}s^2/m^2$ while free-falling (without parachute) and 0.56 $N\text{-}s^2/m^2$ with the round canopy parachute (Knacke, 1992). The gravitational constant is 9.8 m/s^2. The mass is considered 80 kg for the paratrooper and the parachute.

From the above inputs, we can estimate the size of parachute assuming rate descent 8 m/s landing speed of human (Knacke, 1992), it is given by equation (6).

$$Mg = ½ \rho v^2 C_d S \tag{6}$$

Substituting the above value in equation (6), we get

$$80 \times 9.80 = \frac{1}{2} \times 1.1256 \times 8^2 \times 0.56 \times \pi/4\ D^2$$

So, $D = 7$ m

Therefore, size of parachute is 7m diameter to achieve the 8 m/s landing speed for a normal human being.

4.2 Code

A computer program has been developed to predict the position and velocity value of a paratrooper during the jump. The detailed program is appended at **Appendix 'A'**. A comments is listed wherever possible in the code to ease the process of editing it for different values.

NOTE: Lines starting with **#** in the below code are comments and they are just to help anyone who wants to simulate the code for different values.

5. RESULTS AND DISCUSSION

We get the following plots from the simulation of the above computer program:

5.1 Parachute Trajectory (y vs x Graph)

A paratrooper is jumped from the aircraft at the speed of the aircraft at the beginning and start free falling under gravitational force acting on his body. The trajectory followed by the paratrooper is estimated as per code developed (ref Appendix 'A'), the path as shown in the Figure 3.

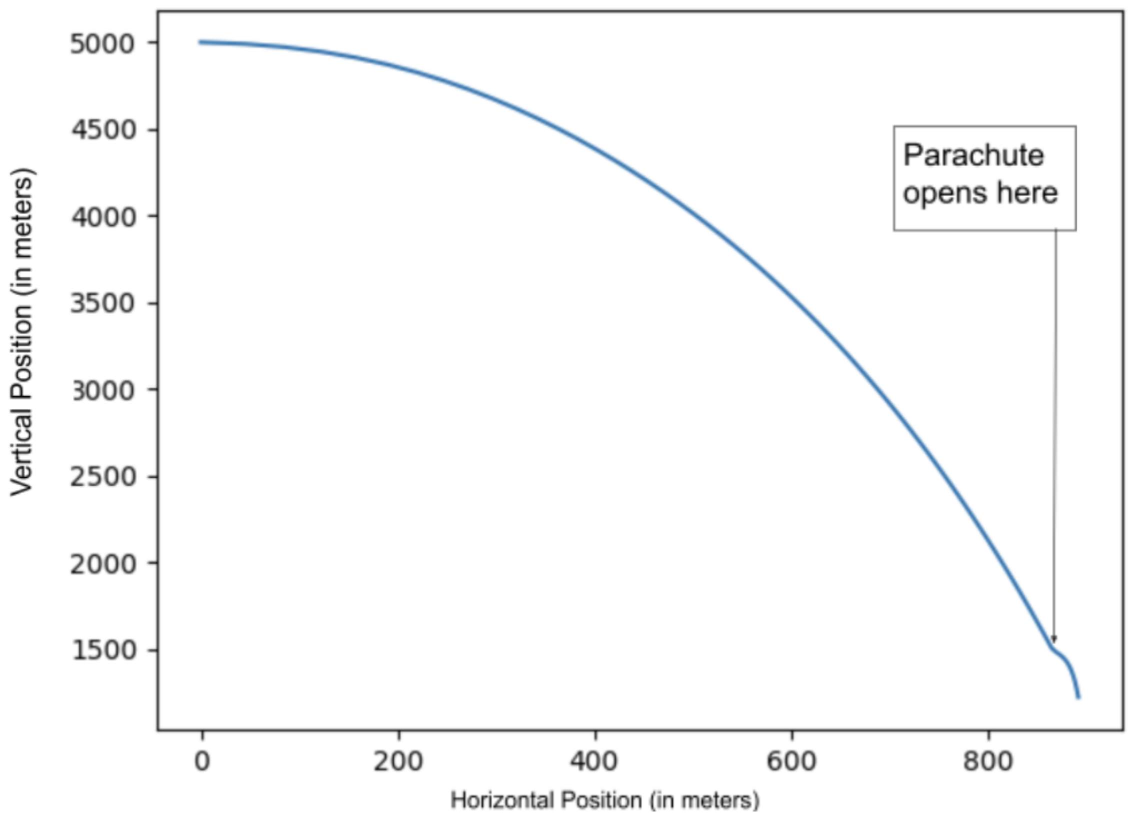

Figure 3: Paratrooper position in 2D

OBSERVATIONS

1. We can clearly see that the paratrooper jumps out of an airplane moving at 50 m/s and at an altitude of 5000 meters, he open the parachute at an altitude of approximately **1507.5 meters** (after 60 seconds) and a slow decrease in altitude after that due to parachute deceleration acting the paratrooper.

2. From the above graph, we can clearly see that after 90 seconds, the paratrooper will be at an altitude of 1225.8 meters and at horizontal position of 891.8 meters.

5.2 Position vs Time Graph

When paratrooper jumps, he is to free fall under gravity force then open the parachute when reaches at desired altitude (1500 m) so that he can easily land at specified location as shown in Figure 4. If he opens at high altitude, external wind may drift him out of the range which is not desired.

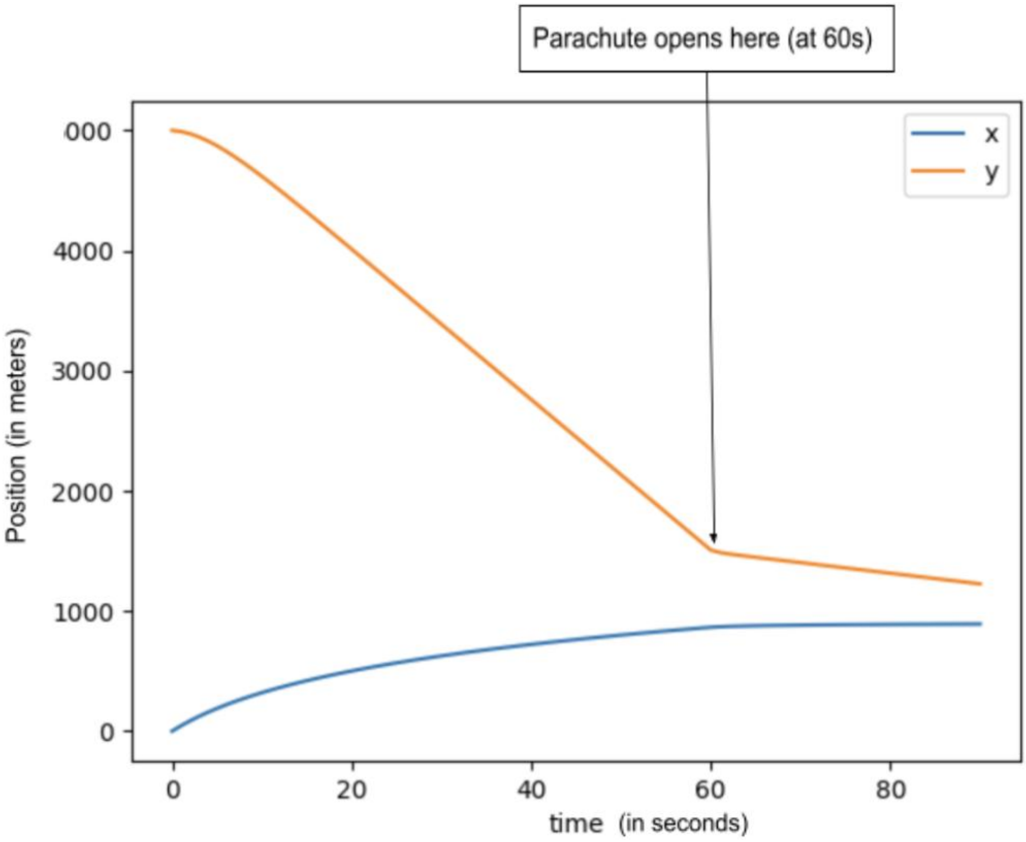

Figure 4: Paratrooper position with respect to time

OBSERVATIONS

1. Here, if we focus on the Vertical Position (ORANGE LINE), we can clearly see that after 60 seconds, there is a decrease in slope (which represents velocity in y-direction). Therefore, the rate of descent is decreased after 60 seconds reaches to terminal speed due to parachute drag force applied on the paratrooper.

5.3 Velocity vs Time Graph

The paratrooper jump at the speed of the 50 m/s from the aircraft and free fall till reaches the desired altitude and then he opens the parachute and decelerates at the rate of 6 m/s terminal speed. If he is not able to open the parachute an altimeter (altitude sensor) trigger the switch and open the parachute automatically. The velocity plot from jump to the terminal speed is shown in Figure 5.

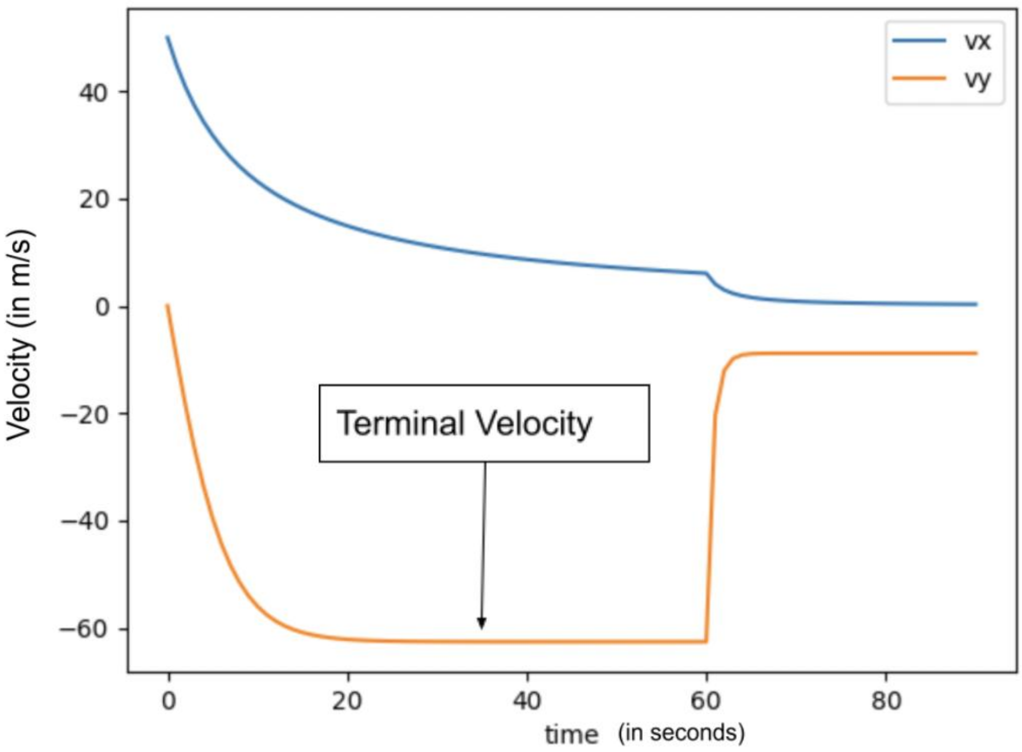

Figure 5: Velocity plot of paratrooper in time domain

OBSERVATIONS

1. Here, if we look at the Horizontal Velocity (BLUE LINE), we observe that it decreases continuously due to the drag force. We can also see that after 60 seconds, the horizontal velocity decreases faster.

2. If we look at the Vertical Velocity (ORANGE LINE), we can see that it increases at a high rate up to 17 seconds. After that, the paratrooper achieves own terminal speed of approximately 62.5 m/s i.e. paratrooper accelerate due to gravity force. This velocity is maintained up to 60 seconds after which the parachute is opened and paratrooper reaches at terminal velocity of 6 m/s at the landing speed.

3. From the above graph, we can clearly see that after 90 seconds, the paratrooper will be at a vertical velocity of -8.86 m/s and at a horizontal velocity of 0.34 m/s.

6. CONCLUSIONS

By using this Program, we can conclude the following:

- After 90 seconds, the paratrooper will be at an altitude of 1225.8 meters and at a distance of 891.8 meters from the initial position.
- After 90 seconds, the paratrooper will have a vertical DESCENT velocity of 8.86 m/s and at a horizontal velocity of 0.34 m/s which is at par with classical calculated rate of descent velocity of 8 m/s.

7. FUTURE SCOPE FOR IMPROVEMENT

- The simulation **only** works for 2 dimensions (x-axis and y-axis). This can be further improve to include the **z-axis also** and thus can simulate the real world parachute drop in 3D.
- We are providing the pre-processed value of '**c**' (i.e. parametric value). The program can be modified to calculate drag coefficient at an instance of time so that we can get more accurate results.

REFERENCES

1. Python documentation
2. NumPy documentation
3. Matplotlib documentation
4. Gekko Documentation
5. Gekko Python Tutorials
6. Dynamic Modeling in GEKKO
7. Dynamic Simulation using GEKKO
8. GEKKO optimisation suite overview
9. NASA - Forces during recovery

10. NASA - Velocity during recovery

11. Knacke, T. W.(1992), “Parachute recovery systems design manual,” NWC TP 6575, Para publishing, Santa Barbara, California, ISBN 0-915516-85-3.

Appendix 'A'

```
#importing gekko module to simulate the event
from gekko import GEKKO
#importing numpy module for calculating complex differential equations
import numpy as np
#importing matplotlib for plotting the result once they are generated
import matplotlib.pyplot as plt

#number of points in time discretization
#since we are calculating for first 90 second i.e. 0s to 90s therefore total seconds = 91
n = 91

#Initialize GEKKO Model
m = GEKKO()

#define time discretization
#Here will write the start and end time of the simulation i.e. start = 0s, end = 90s
m.time = np.linspace(0,90,n)

#make array of drag coefficients, changing at time 60
#here we are setting drag coefficient to 0.2 is current time is 60s(i.e. chute NOT opened)
else we are setting it to 10
drag = [(0.2 if t<=60 else 10) for t in m.time]

#define constants
# gravitational constant = 9.81
g = m.Const(value=9.81)
#mass given in the input is 80kg
mass = m.Const(value=80)

#define drag parameter
d = m.Param(value=drag)

#initialize variables which are to be considered
#x = position in x direction
#y = position in y direction
#vx = velocity in x direction
#vy = velocity in y direction
#v = resultant velocity
#Fx = Force in x direction
#Fy = Force in y direction
x,y,vx,vy,v,Fx,Fy = [m.Var(value=0) for i in range(7)]

#initial conditions
#this is the altitude from which paratrooper jumped from Airplane
y.value = 5000
#speed of Airplane
vx.value = 50

#Equations
# force balance
```

```
m.Equation(Fx == -d * vx**2)
m.Equation(Fy == -mass*g + d*vy**2)
#F = ma
m.Equation(Fx/mass == vx.dt())
m.Equation(Fy/mass == vy.dt())
#vel = dxdt
m.Equation(vx == x.dt())
m.Equation(vy == y.dt())
#total velocity
m.Equation(v == (vx**2 + vy**2)**.5)

#Set global options
m.options.IMODE = 4 #dynamic simulation

#Solve simulation
m.solve(remote=True)

#%% Plot results
plt.figure()
plt.plot(x.value,y.value)
plt.xlabel('x')
plt.ylabel('y')

plt.figure()
plt.plot(m.time,x.value,label='x')
plt.plot(m.time,y.value,label='y')
plt.xlabel('time')
plt.legend()

plt.figure()
plt.plot(m.time,vx.value,label='vx')
plt.plot(m.time,vy.value,label='vy')
plt.xlabel('time')
plt.legend()

plt.show()
```