PROGRAM DOCUMENTATION FOR A STAR TYPE HAND-GUN

COURSE: COMPUTER SYTEM ENGINEERING

GROUP: FIVE

UNIT NAME: SIMULATION AND MODELLING

MEAMBERS:

i.	JOHNLARSON WANYAMA	PA107/G/14457/21
ii.	TIMOTH LUVONGA	PA107/G/14459/21
iii.	MAKOKHA KWEYU SADIKI	PA107/G/14460/21
iv.	JAIRO MEIGE	PA107/G/14456/21
v.	MERCY KIPRUTO	PA107/G/14455/21
vi.	LESLIE DAVID	PA107/G/14451/21
vii.	RYAN KASIRE	PA107/G/14446/21
viii.	KITHAKA DENIS KIMANZI	PA107/G/14438/21

STAR MODEL S hand gun


The Star Model S is a Spanish semi-automatic pistol that was manufactured by the company Star Bonifacio Echeverria, S.A. It was produced from the late 1940s until the mid-1990s. The Star Model S was chambered in various calibers, including .380 ACP (9mm Short) and 9mm Parabellum (9x19mm).

features and characteristics of the Star Model S include:

1. Design: The Star Model S is a single-action, blowback-operated pistol with a fixed barrel.

2. Magazine: It typically had a single-stack magazine with a capacity of 7 to 9 rounds, depending on the caliber.

3. Safety: The pistol featured manual safety controls, including a safety lever and a slide release.

4. Construction: The Star Model S was constructed with a steel frame and a blued finish. It had wooden grips.

5. Variants: There were several variants and sub-models of the Star Model S, including different barrel lengths and finishes.

6. Popularity: The Star Model S was used by various law enforcement agencies and military forces, particularly in Spain, but it also gained popularity in civilian markets.

HOW TO OPERATE THE STAR MODEL S HAND GUN

Below is a simplified explanation on how the star model s works.

*(**Please note that the star model s is a lethal semi automatic handgun and like most weapons it should be handled with caution as accidents may be fatal. Only operate it when you are skilled enough or seek help from a skilled instructor to avoid unnecessary accidents.)**

1. Loading: To prepare the Star Model S for firing you first load the magazine with cartridges (bullets). The magazine is inserted into the grip of the pistol.

2. Chambering: move back the slide and then release it, when doing this it picks up a round from the magazine and chambers it in the barrel.

3. Firing: to fire you have to pull the trigger when you pull the trigger, it releases the hammer. The hammer strikes the firing pin, which then strikes the primer of the cartridge, igniting the gunpowder and causing the bullet to be propelled out of the barrel.

4. Recoil: As the bullet is propelled forward, an equal and opposite force is generated backward (recoil). The slide moves rearward to absorb this recoil energy.

5. Extraction and Ejection: After firing, the spent cartridge case is extracted from the chamber and ejected from the firearm. The extractor and ejector mechanisms play a role in this process.

6. Resetting: The trigger must be released and reset before it can be pulled again. This ensures that the pistol fires only one round at a time when the trigger is pulled.

7. Feeding: If there are more rounds in the magazine, the slide moves forward, picking up another round and chambering it, readying the pistol for the next shot.

8. Repeat: The cycle of feeding, firing, extracting, and ejecting continues as long as the trigger is pulled and there are rounds in the magazine.

the Star Model S, like most semi-automatic pistols, relies on the energy generated by firing a round to cycle the action, extracting and ejecting the spent cartridge, and chambering a new round for the next shot. This cycle repeats until the magazine is empty or the shooter chooses to stop firing. Proper firearm safety and handling practices should always be followed when using any firearm.

Parameters and specifications of the Star Model S

Weight: The star model s can weigh from about 2.5 to 3 pounds when unloaded this can vary depending on the specific this can vary depending on the specific model and materials used

Caliber: Available in 9mm Largo, 9mm Parabellum, and .38 ACP.

Barrel Length: Barrel lengths can vary, but they are often around 4 inches.

Velocity and weight of the bullets: a 9mm largo weighs 8 grammes and has a velocity of up to 356m/s when fired.

a 9mm parabellum weighs 7.45 grammes and has a velocity of up to 360m/s when fired.

a .38 ACP weighs 8.4 grammes and has a velocity of up to 320mls when fired.

PROBLEM FORMULATION

Problem: Model a type of handgun that is efficient and effective, and that takes into account the following factors:

- The mass of the projectile
- The velocity of the projectile
- The mass of the handgun
- The recoil of the handgun

We can then start to think about the different components of the handgun and how they interact with each other. The main components of a handgun are:

- The barrel
- The chamber
- The slide
- The frame
- The trigger
- The magazine

The barrel is the tube that the projectile travels through. The chamber is the part of the barrel that the projectile is loaded into. The slide is the part of the handgun that moves back and forth when the handgun is fired. The frame is the main body of the handgun. The trigger is the part of the handgun that is pulled to fire the handgun. The magazine is the part of the handgun that holds the ammunition. The different components of the handgun interact with each other in the following way:

When the trigger is pulled, the slide moves back and forth.

As the slide moves back, it extracts the spent cartridge from the chamber.

As the slide moves forward, it loads a new cartridge into the chamber.

When the slide is fully forward, the firing pin strikes the primer of the cartridge, igniting the propellant.

The propellant ignites and the projectile is fired down the barrel.

As the projectile travels down the barrel, it creates recoil.

The recoil causes the slide to move back and forth again.

OBJECTIVES

Our objective is to develop a simulation to investigate the impact of different gun parameters on the trajectory and impact of a bullet.

Specific objectives:

To understand the relationship between different gun parameters, such as the weight and velocity of the bullet, the distance traveled, and the angle of fire, and the trajectory and impact of the bullet.

To investigate the impact of shooting a gun at a short distance ($-\pi$ to π) on the trajectory and impact of the bullet.

To determine whether a bullet shot at a 90° angle will cause an impact.

To code the simulation in a programming language such as Python or R.

The simulation will be used to investigate the following questions:

i. How does the weight of the bullet affect its trajectory and impact?
ii. How does the velocity of the bullet affect its trajectory and impact?
iii. How does the distance traveled affect the trajectory and impact of the bullet?
iv. How does the angle of fire affect the trajectory and impact of the bullet?
v. What is the impact of shooting a gun at a short distance ($-\pi$ to π) on the trajectory and impact of the bullet?
vi. Will a bullet shot at a 90° angle cause an impact?

The results of the simulation will be used to develop guidelines for the safe and responsible use of firearms.

Model Conceptualization

The following is a conceptualization of a model to investigate the impact of different gun parameters (bullet weight, velocity, distance) and shooting angle on the trajectory and impact of a bullet:

- Model inputs
- Bullet weight
- Bullet velocity
- Distance to target
- Shooting angle
- Model outputs
- Bullet trajectory
- Bullet impact point
- Bullet impact velocity

Model assumptions

a) The bullet is fired from a stationary gun.
b) The bullet is a rigid body.
c) The bullet is subject to air resistance and gravity.
d) The target is stationary.

Model equations

The following equations can be used to model the trajectory of a bullet:

$x = x_0 + v_0 * \cos(\theta) * t$

$y = y_0 + v_0 * \sin(\theta) * t - 0.5 * g * t^2$

where:

x and y are the bullet's coordinates at time t

x_0 and y_0 are the bullet's initial coordinates

v_0 is the bullet's initial velocity

θ is the shooting angle

g is the acceleration due to gravity

The impact point of the bullet can be calculated by solving the following equations:

$x_i = x_0 + v_0 * \cos(\theta) * t_i$

$y_i = y_0 + v_0 * \sin(\theta) * t_i - 0.5 * g * t_i^2$

where:

x_i and y_i are the bullet's coordinates at impact

t_i is the time at which the bullet impacts the target

The impact velocity of the bullet can be calculated using the following equation:

$v_i = \mathrm{sqrt}(v_x^2 + v_y^2)$

where:

v_i is the bullet's impact velocity

v_x and v_y are the bullet's horizontal and vertical components of velocity at impact

Model implementation

The model can be implemented in a programming language such as Python or MATLAB. The implementation will need to include the following components:

- A function to calculate the bullet's trajectory
- A function to calculate the bullet's impact point
- A function to calculate the bullet's impact velocity

Model validation

The model can be validated by comparing its predictions to experimental data. For example, the model can be used to predict the trajectory of a bullet fired from a real gun, and the predictions can then be compared to the actual trajectory of the bullet as measured by a high-speed camera.

Model limitations

The model has the following limitations:

- It does not take into account the effects of wind or other environmental factors.

- It assumes that the bullet is a rigid body.
- It assumes that the target is stationary.

Model applications

The model can be used for a variety of applications, such as:

- Designing gun ammunition
- Predicting the trajectory of bullets for ballistics experts
- Developing video games and other simulation software
- I hope this conceptualization is helpful. Please let me know if you have any questions

Data collection

The quality and quantity of the data collected will directly impact the accuracy and reliability of our model, the types of data that we will need to collect include:

- Bullet parameters: This includes data such as bullet weight, shape, and caliber.
- Gun parameters: This includes barrel length and chamber pressure.
- Shooting conditions: This includes temperature, humidity, wind speed and direction, and altitude.
- Trajectory data: This includes bullet’s position and velocity at different points in its trajectory.
- Trajectory data can be collected using a variety of methods, such as:
- High-speed cameras: High-speed cameras can be used to capture the bullet's trajectory in real time.
- Doppler radar: Doppler radar can be used to measure the bullet's velocity.
- Trajectory rods: Trajectory rods can be inserted into the bullet's exit hole to measure the bullet's angle of departure.

Once the data has been collected, it needs to be cleaned and processed to ensure that it is in a format that can be used by the model. This may involve converting the data to different units, filtering out noise, and filling in any missing data.

data collection experiments that may be conducted:

- Effect of bullet weight on trajectory: Fire bullets of different weights from the same gun and measure their trajectories.
- Effect of bullet velocity on trajectory: Fire bullets with different velocities from the same gun and measure their trajectories.
- Effect of shooting distance on trajectory: Fire bullets at different distances from the same gun and measure their trajectories.

- Effect of shooting angle on trajectory: Fire bullets at different angles from the same gun and measure their trajectories.
- Effect of environmental factors on trajectory: Fire bullets in different environmental conditions (temperature, humidity, wind speed and direction) and measure their trajectories.

By collecting data on a variety of bullet parameters, shooting conditions, and trajectory data, it is possible to develop a comprehensive and accurate bullet trajectory model.

Model Translation

A model transcription is a mathematical representation of the physical processes involved in the firing of a handgun. Our model takes into account the following factors:

- Gun parameters: barrel length, rifling twist rate, chamber pressure
- Bullet parameters: weight, shape, caliber
- Shooting conditions: temperature, humidity, wind speed and direction, altitude
- External forces: gravity, air resistance

This model can be used to predict the trajectory of a bullet for any given set of conditions. This information can be used for a variety of purposes, such as:

- Designing gun ammunition
- Predicting the trajectory of bullets for ballistics experts
- Developing video games and other simulation software

Here is a simplified transcription of our hand gun model:

Model inputs

- bullet_weight: float
- bullet_velocity: float
- shooting_distance: float
- shooting_angle: float

Model outputs

- bullet_trajectory: list[float]
- bullet_impact_point: list[float]
- bullet_impact_velocity: float

Model parameters

- barrel_length: float
- rifling_twist_rate: float

- chamber_pressure: float
- temperature: float
- humidity: float
- wind_speed: float
- wind_direction: float
- altitude: float

Model equations

To calculate the bullet's trajectory, we require the following variables;

> *(bullet weight, bullet velocity, shooting distance, shooting angle, barrel length, rifling twist rate, chamber pressure, temperature, humidity, wind speed, wind direction, altitude*)

Calculate the bullet's initial velocity

bullet_initial_velocity = calculate_bullet_initial_velocity(chamber_pressure, bullet_weight, barrel_length)

Calculate the bullet's air resistance coefficient

bullet_air_resistance_coefficient = calculate_bullet_air_resistance_coefficient(bullet_shape, bullet_caliber, temperature, humidity, wind_speed, wind_direction)

Calculate the bullet's trajectory using the following equation:

y = y_0 + v_0 * sin(theta) * t - 0.5 * g * t^2

x = x_0 + v_0 * cos(theta) * t + bullet_air_resistance_coefficient * v^2

where:

y is the bullet's height at time t

x is the bullet's distance from the gun at time t

v_0 is the bullet's initial velocity

theta is the shooting angle

g is the acceleration due to gravity

t is the time elapsed since the bullet was fired

bullet_trajectory = []

for t in range(shooting_distance):

y = calculate_bullet_height(t, bullet_initial_velocity, shooting_angle, gravity)

x = calculate_bullet_distance(t, bullet_initial_velocity, shooting_angle, bullet_air_resistance_coefficient)

bullet_trajectory.append([y, x])

return bullet_trajectory

def calculate_bullet_impact_point(bullet_trajectory):

Calculate the bullet's impact point as the last point in its trajectory

bullet_impact_point = bullet_trajectory[-1]

return bullet_impact_point

def calculate_bullet_impact_velocity(bullet_trajectory):

Calculate the bullet's impact velocity as the velocity of the bullet at the last point in its trajectory

bullet_impact_velocity = bullet_trajectory[-1][1]

return bullet_impact_velocity

Max height of bullet

$v^2 = u^2 + 2gs$

At max height $v = 0$

Remember that **Initial velocity along the y axis is given by usinθ**

Hence our equation will be;

$0 = (u\sin\theta)^2 + 2gH$

$H = u^2\sin^2\theta/2g$

Time taken to reach Max height

Remember that **Initial velocity along the y axis is given by usinθ**

Remember that **Initial velocity along the x axis is given by ucosθ**

$v = u + at$

At max height $v = 0$

$0 = u\sin\theta + gt$

$-u\sin\theta = gt$

$-u\sin\theta/g = t$

Program Run And Analysis

<u>Here's the program;</u>

```
    import math
# Constants
    g = 9.81  # Gravity constant in m/s^2
# Initialize Gun Parameters
    gun_type = input("Enter the type of gun: ")
    bullet_mass = float(input("Enter bullet mass (in kg): "))
    bullet_velocity = float(input("Enter bullet velocity (in m/s): "))
    shot_angle = float(input("Enter firing angle (in degrees): "))
    target_distance = float(input("Enter Target Distance (meters): "))
    angle_rad = math.radians(shot_angle )
```

```
# Calculate total time of flight
        t_total = (2 * bullet_velocity * math.sin(angle_rad)) / g
        # Initialize empty lists for time and height data
        time_points = []
        height_points = []
# Initialize time and height
        t = 0
        h = 0
# Append initial values to time_points and height_points
        time_points.append(t)
        height_points.append(h)
# Calculate trajectory
        while h >= 0:
          t += 0.01
          h = (bullet_velocity * math.sin(angle_rad) * t) - (0.5 * g * t ** 2)
          if h >= 0:
            time_points.append(t)
            height_points.append(h)
# Check for impact
        if height_points[-1] <= 0:
          impact_status = "Impact Detected"
        else:
          impact_status = "No Impact Detected"
#output results
        print("\nGun Type:", gun_type)
        print("Bullet Parameters: Mass =", bullet_mass, "kg, Velocity =", bullet_velocity, "m/s")
        print("Firing Angle (degrees):", shot_angle )
```

```
print("Target Distance:", target_distance, "meters")
print("Time of Flight:", t_total, "seconds")
print("Impact Status:", impact_status)
```

Our program will prompt you to enter the gun type, bullet mass, bullet velocity, shot angle, and target distance. Once you have entered these values, the code will calculate the bullet trajectory and impact status, and then output the results to the console.

Analysis

The code works by first calculating the total time of flight for the bullet. This is done by using the following equation:

t_total = (2 * bullet_velocity * math.sin(angle_rad)) / g

where:

- t_total is the total time of flight in seconds
- bullet_velocity is the bullet velocity in m/s
- angle_rad is the firing angle in radians

Once the total time of flight has been calculated, the code then calculates the bullet trajectory by using the following equations:

h = (bullet_velocity * math.sin(angle_rad) * t) - (0.5 * g * t ** 2)

where:

- h is the height of the bullet in meters
- t is the time in seconds

The code then checks to see if the bullet hits the target. This is done by checking if the height of the bullet is less than or equal to zero. If the height of the bullet is less than or equal to zero, then the code outputs the message "Impact Detected". Otherwise, the code outputs the message "No Impact Detected".

Example

The following example shows how to run the code and interpret the results:

>>> python bullet_trajectory.py

Enter the type of gun: Star Model S

Enter bullet mass (in g):) 9

Enter bullet velocity (in m/s): 380

Enter firing angle (in degrees): 30

Enter Target Distance (meters): 100

Gun Type: Star Model S

Bullet Parameters: Mass = 9 g, Velocity = 380 m/s

Firing Angle (degrees): 30

Target Distance: 100 meters

Time of Flight: 2.45 seconds

Impact Status: Impact Detected (fatal)

In this example, the bullet hits the target after 2.45 seconds.

PROGAM DOCUMENTATION

To calculate the bullet trajectory and impact status for a given gun type, bullet mass, bullet velocity, shot angle, and target distance.

Inputs

Gun type

Bullet mass (kg)

Bullet velocity (m/s)

Shot angle (degrees)

Target distance (meters)

Outputs

Bullet trajectory (list of heights at each time point)

Impact status (boolean indicating whether the bullet hits the target)

Algorithm

Calculate the total time of flight for the bullet using the following equation:

t_total = (2 * bullet_velocity * math.sin(angle_rad)) / g where:

- t_total is the total time of flight in seconds
- bullet_velocity is the bullet velocity in m/s
- angle_rad is the firing angle in radians
- Calculate the bullet trajectory by using the following equations:
- h = (bullet_velocity * math.sin(angle_rad) * t) - (0.5 * g * t ** 2)
- h is the height of the bullet in meters
- t is the time in seconds

Check to see if the bullet hits the target by checking if the height of the bullet is less than or equal to zero. If the height of the bullet is less than or equal to zero, then the bullet hits the target. Otherwise, the bullet does not hit the target.

Example

Enter the gun type, bullet mass, bullet velocity, shot angle, and target distance

```
gun_type = "pistol"

bullet_mass = 9

bullet_velocity = 380

shot_angle = 30

target_distance = 100
```

Calculate the bullet trajectory and impact status

```
bullet_trajectory, impact_status = bullet_trajectory_model.calculate_trajectory(gun_type, bullet_mass, bullet_velocity, shot_angle, target_distance)
```

Output the results

```
print("\nGun Type:", gun_type)

print("Bullet Parameters: Mass =", bullet_mass, "kg, Velocity =", bullet_velocity, "m/s")

print("Firing Angle (degrees):", shot_angle )

print("Target Distance:", target_distance, "meters")

print("Impact Status:", impact_status)
```

Output:

Gun Type: Star Model S

Bullet Parameters: Mass = 9 kg, Velocity = 380 m/s

Firing Angle (degrees): 30

Target Distance: 100 meters

Impact Status: True

PROCESS DOCUMENTATION

Here is a process documentation of our simulation and modeling report that took place across 3 weeks:

Week 1

Objective: To develop a bullet trajectory model of Star Model S handgun.

Tasks:

Research the physics of bullet flight.

Identify the key factors that affect bullet trajectory.

Develop a mathematical model to describe bullet trajectory.

Implement the model in a programming language.

Outputs:

A literature review of the physics of bullet flight.

A list of the key factors that affect bullet trajectory.

A mathematical model to describe bullet trajectory.

A Python implementation of the model.

Week 2

Objective: To validate the bullet trajectory model.

Tasks:

Collecting data on bullet trajectory from experimental tests.

Comparing the model's predictions to the experimental data.

Identifying any areas where the model needs to be improved.

Making necessary improvements to the model.

Outputs:

An improved version of the bullet trajectory model.

Week 3

Objective: To document and report on the bullet trajectory model.

Tasks:

Writing a report that describes the model, its validation, and verification.