

“魔板”加持下的平抛运动定理验证实验

吴承宇 20230616^{*} 陈麒泽 20231204[†] 胡铭轩 20231207[†] 卢柯忻 20230534[†]
张函毓 20231237[†] 章嘉乐 20231226[†]

2024 年 5 月 30 日

摘要

通过朗威 (DISLab) 的“魔板”系统, 可以即时获取平抛物体的运动轨迹. 通过以时间, x 轴坐标和 y 轴坐标为变量的数据, 可以验证平抛运动的各项定理. 本实验通过“魔板”系统, 以及“魔板”系统配套的实验仪器, 验证了平抛运动的各项定理.

在中学课本中, 平抛运动的定理的验证是通过复写纸, 通过速度的分解来分别得出数值和平行方向的运动特点, 进而验证平抛运动的模型. 然而, 传统实验中, 初始抛出小球的斜面需要人用手来控制小球的释放, 这可能导致一系列的系统误差.

通过信息技术, 本组成员方便的获取了每 0.02s 的实验数据, 以通过回归的方式训练出描述轨迹的回归方程, 以验证平抛运动的定律.

关键词 平抛运动; 实验改进; DISLab “魔板”实验; 回归模型; 验证实验

目录

| | |
|----------------|---|
| 1 问题分析 | 3 |
| 2 实验准备 | 3 |
| 3 实验步骤 | 4 |
| 4 实验数据分析 | 4 |
| 4.1 水平方向线性回归分析 | 5 |
| 4.2 竖直方向线性回归分析 | 6 |
| 4.3 能量学角度分析 | 7 |
| 5 实验结论 | 7 |
| 6 实验改进 | 7 |
| 7 实验拓展 | 7 |
| A 实验数据 | 9 |
| A.1 数据格式 | 9 |

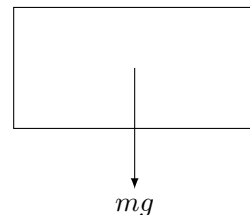
^{*}Ethan Goh (<7086cmd@gmail.com>). 本文的主要撰写者, 同时承担数据分析, 软件配置操作.

[†]基本上贡献相等. 搭建了实验模型, 参与了实验设计和文章撰写.

| | | |
|----------|--------------------------|-----------|
| B | 数据分析代码 | 9 |
| B.1 | 绘图 | 9 |
| B.1.1 | 绘制单个散点图 | 9 |
| B.1.2 | 绘制所有散点图 | 10 |
| B.1.3 | 绘制 $x - t$ 回归图 | 10 |
| B.1.4 | 绘制 $y - t$ 回归图 | 11 |
| B.2 | 回归 | 11 |
| C | 仓库 | 12 |

1 问题分析

对于平抛运动的轨迹, 理论上可以通过受力分析进行:



因此, 在水平方向上, 不计空气阻力则有 $F_x = 0$, 在竖直方向上, 有 $F_y = -mg$. 由牛顿第二定律, 可以得到:

$$\begin{cases} a_{\perp} = -g \\ a_{\parallel} = 0 \end{cases} \quad (1)$$

因此, 对其进行积分, 可以得到:

$$\begin{cases} v_{\perp} = -gt + v_{0\perp} \\ v_{\parallel} = v_{0\parallel} \end{cases} \quad (2)$$

因此, 可以得到平抛运动的轨迹方程:

$$\begin{cases} x = v_{0\parallel}t \\ y = v_{0\perp}t - \frac{1}{2}gt^2 \end{cases} \quad (3)$$

因此, 可以通过实验验证平抛运动的定理.

2 实验准备

本实验通过朗威 (DISLab) 的“魔板” (Magic Board) 系统进行实验. “魔板”系统通过提供一个较大的竖直平台, 可以快速以 50 Hz 等频率记录运动块的方位, 通过绘制散点图, 线型图等方式可以得到运动块的运动轨迹.

魔板系统可以视为一个抛出器与位置传感器, 整体如图 1 所示.

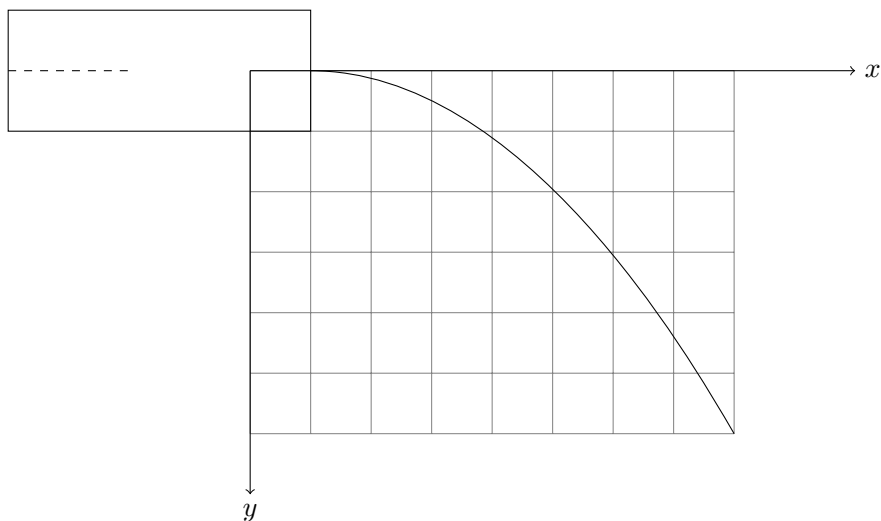


图 1: “魔板”示意图

3 实验步骤

本次实验设计步骤如下:

1. 组装“魔板”系统, 并连接至电脑.
2. 在计算机中打开“魔板”对应软件, 并将运动快放到抛出原点.
3. 按照 3 级弹簧压缩量, 以及 0° 的角度多次抛出是运动块.
4. 导出实验数据, 通过 `scikit-learn` 库进行回归模型的分析, 并总结模型.

该软件的界面如图 2 所示.

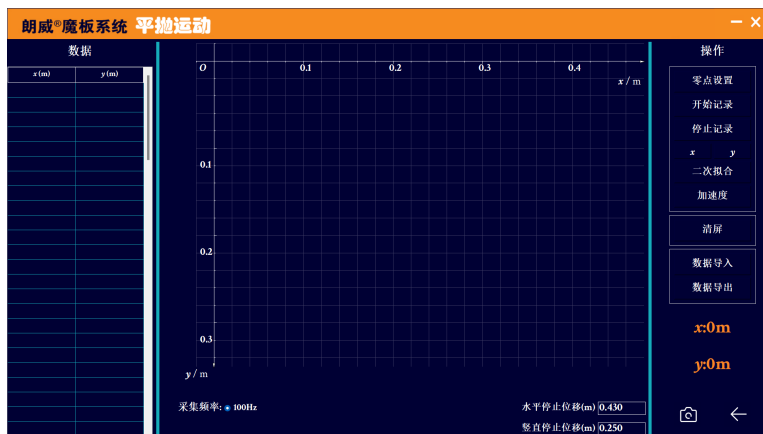


图 2: “魔板”软件界面

通过多次实验, 笔者获得了 4 组数据, 并保存在 `data` 文件夹中.

4 实验数据分析

以 `2024-05-17(16-57-14).csv` 中的数据为例, 通过 `matplotlib` 库, 笔者绘制出如下散点图:

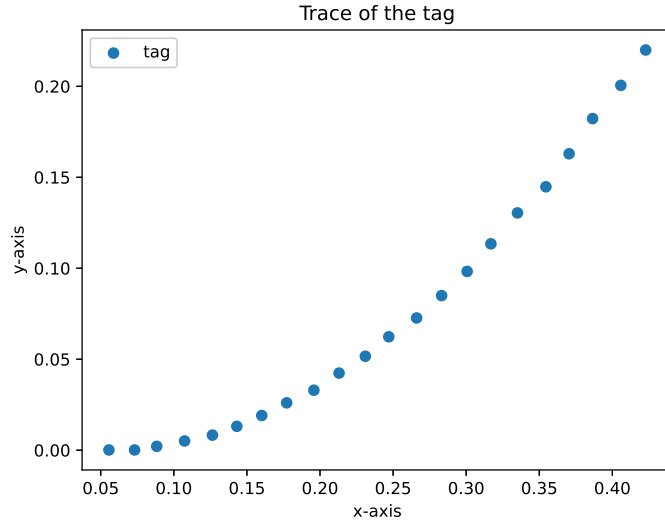


图 3: 实验数据散点图

因此, 我们对于所有的数据进行绘制, 得到如下的散点图:

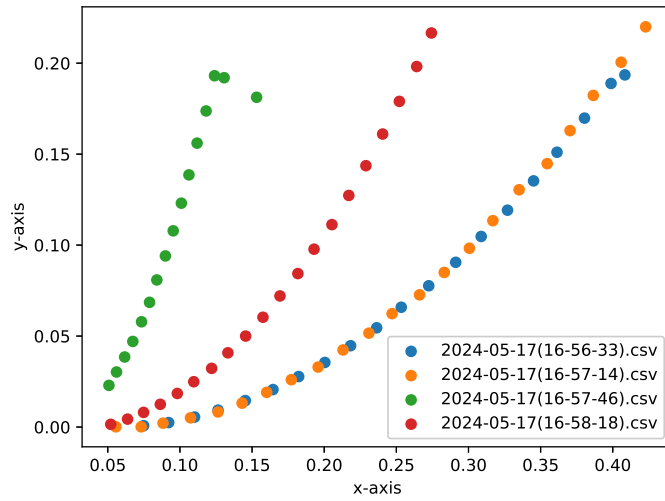


图 4: 实验犬必怒数据散点图

通过散点图的绘制, 发现实验数据 `2024-05-17(16-56-33).csv`, `2024-05-17(16-57-14).csv` 和 `2024-05-17(16-58-18).csv` 两份数据较有准确性, 因此采用这两组数据进行线性回归拟合来进行数据分析.

因此, 笔者通过 `scikit-learn` 的 `LinearRegression` 模型分别对于 $x-t$ (水平方向) 和 $y-t$ (竖直方向) 进行回归分析.

4.1 水平方向线性回归分析

通过匀加速运动的公式, 我们假设:

$$\hat{x} = \theta_2 t^2 + \theta_1 t + \theta_0 \quad (4)$$

则容易发现, $a = 2\theta_2$, $v_0 = \theta_1 = 0$. 通过回归分析, 可以得到如下的参数:

| 数据 | $\theta_2 / \text{ms}^{-2}$ | $\theta_1 / \text{ms}^{-1}$ | θ_0 / m | Loss |
|-------------------------------|-----------------------------|-----------------------------|-----------------------|------------|
| data/2024-05-17(16-57-14).csv | 4.88 | 0.08 | -0.00 | 0.00000775 |
| data/2024-05-17(16-56-33).csv | 4.44 | 0.12 | -0.00 | 0.00014052 |
| data/2024-05-17(16-58-18).csv | 4.81 | -7.28 | 2.75 | 0.00000277 |
| data/2024-05-17(16-57-46).csv | 0.93 | 0.33 | -0.29 | 0.00112636 |

表 1: 水平方向线性回归分析

数据 data/2024-05-17(16-57-46).csv 和 data/2024-05-17(16-58-18).csv 存在较大误差, 因此不予考虑. 回归模型的误差函数小, 拟合效果较好.

取数据 data/2024-05-17(16-57-14).csv 为例, 绘制出拟合图像如下:

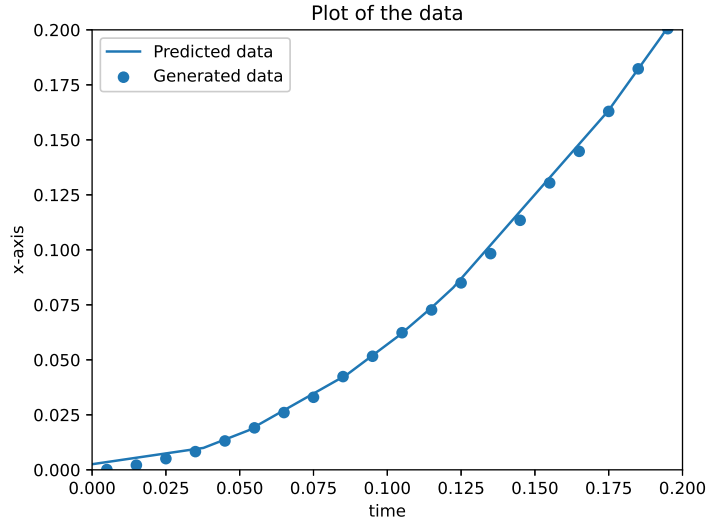


图 5: 水平方向线性回归分析

则水平方向的重力加速度 $g = 2\theta_2 = 9.76 \text{ ms}^{-2}$. 根据数据, 30° N 位置的重力加速度一般为 9.794 ms^{-2} , 因此实验数据较为准确.

4.2 竖直方向线性回归分析

根据平抛运动的定理, 竖直运动基本上为匀速直线运动. 因此, 我们假设:

$$\hat{y} = \theta_1 t + \theta_0 \quad (5)$$

通过对于数据的回归分析, 可以得到如下的参数:

| 数据 | $\theta_1 / \text{ms}^{-1}$ | θ_0 / m | Loss |
|-------------------------------|-----------------------------|-----------------------|------------|
| data/2024-05-17(16-57-14).csv | 1.75 | 0.06 | 0.00001409 |
| data/2024-05-17(16-56-33).csv | 1.79 | 0.05 | 0.00006406 |
| data/2024-05-17(16-58-18).csv | 1.18 | -0.87 | 0.00000526 |
| data/2024-05-17(16-57-46).csv | 0.60 | -0.21 | 0.00025321 |

表 2: 竖直方向线性回归分析

则容易发现, 数据 `data/2024-05-17(16-57-46).csv` 存在较大误差, 因此不予考虑. 回归模型的误差函数同样小, 拟合效果较好.

以 `data/2024-05-17(16-57-14).csv` 为例, 绘制出拟合图像如下:

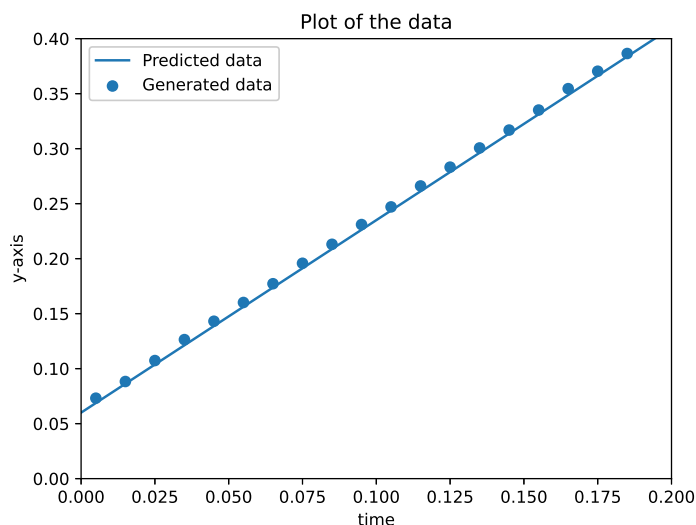


图 6: 竖直方向线性回归分析

4.3 能量学角度分析

根据动能定理 $\frac{1}{2}mv^2 = mgh$, 可以通过功能转换的方法来验证数据的准确性. 该实验装置 50 Hz 的频率, 可以使得接近于瞬时速度. 但由于笔者精力有限, 未能进行更多的数据分析.

5 实验结论

物体做平抛运动时, 在水平方向上是匀速直线运动, 在竖直方向上是匀加速直线运动. 通过“魔板”系统, 可以方便的获取实验数据, 并通过回归模型进行分析, 以验证平抛运动的定理.

6 实验改进

在该实验中, 通过“魔板”系统, 可以方便的获取实验数据, 并通过回归模型进行分析, 以验证平抛运动的定理. 但是, 使用“魔板”系统时, 也会存在记录时间和放出时间的误差, 这可能导致了诸如 16-57-46 组数据的问题. 在实际使用中, 可以通过更加精确的时间配合, 甚至自动进行数据记录, 以减少误差.

同时, “魔板”系统自身存在一定摩擦. 在实验中, 可以通过水平方向上的力场进行控制. 以及, 进行平抛运动的物体较轻, 且非流线型, 这可能导致了空气阻力对实验产生的影响.

7 实验拓展

该实验也可以用于考虑空气阻力的抛体运动, 以及复合场中的抛体运动进行分析. 尽管本身“魔板”系统为磁吸装置, 但是可以通过更换不同的感应物块, 以及更换不同的磁场, 以进行不同的实验.

参考文献

- [1] Peter Chudinov. Vertical projectile motion taking into account the wind, 2024.
- [2] Ethan Goh. Mathematical modeling of shot put throwing distance. March 2024.
- [3] Dimitrios Gousopoulos. Investigating students' scientific reasoning through heuristic and analytical thought processes, 2023.
- [4] Andrew Jacobsen and Ashok Cutkosky. Online linear regression in dynamic environments via discounting, 2024.
- [5] Ramkumar Natarajan, Hanlan Yang, Qintong Xie, Yash Oza, Manash Pratim Das, Fahad Islam, Muhammad Suhail Saleem, Howie Choset, and Maxim Likhachev. Preprocessing-based kinodynamic motion planning framework for intercepting projectiles using a robot manipulator, 2024.
- [6] David White, Lam Tran, and Lin Ma. A statistical analysis of drug seizures and opioid overdose deaths in ohio from 2014 to 2018. *Journal of Student Research*, 10(1), March 2021.
- [7] 人民教育出版社. 物理必修第一册, page 43. 人民教育出版社, 2019.
- [8] 人民教育出版社. 物理必修第二册, pages 10–13. 人民教育出版社, 2019.
- [9] 李俊成. 谈平抛运动问题的解题思想和方法. 物理教学, (2):3, 2013.
- [10] 胡一德. 探究平抛运动的规律. 中国现代教育装备, 2011.

附录

A 实验数据

实验数据保存在 `data` 文件夹中, 其中包含了 4 组数据, 分别为:

- 2024-05-17(16-56-33).csv
- 2024-05-17(16-57-14).csv
- 2024-05-17(16-57-46).csv
- 2024-05-17(16-58-18).csv

A.1 数据格式

数据格式如下:

| 时间 | x | y |
|------|------|------|
| 0.00 | 0.00 | 0.00 |
| 0.02 | 0.00 | 0.00 |
| 0.04 | 0.00 | 0.00 |
| 0.06 | 0.00 | 0.00 |
| 0.08 | 0.00 | 0.00 |

表 3: 数据格式

B 数据分析代码

代码分为绘图和回归两部分, 代码如下:

B.1 绘图

B.1.1 绘制单个散点图

```
1 import matplotlib.pyplot as plt
2 import pandas as pd
3
4 df = pd.read_csv('data/2024-05-17(16-57-14).csv')
5
6 plt.scatter(df['x'], df['y'], label='tag')
7 plt.xlabel('x-axis')
8 plt.ylabel('y-axis')
9 plt.title('Trace of the tag')
10 plt.legend()
11 plt.savefig('figures/plot1.eps')
```

B.1.2 绘制所有散点图

```
1 import matplotlib.pyplot as plt
2 import pandas as pd
3 import os
4
5 for file in os.listdir('data'):
6     if file.endswith('.csv'):
7         df = pd.read_csv(f'data/{file}')
8         plt.scatter(df['x'], df['y'], label= file)
9
10 plt.xlabel('x-axis')
11 plt.ylabel('y-axis')
12 plt.legend()
13 plt.savefig('figures/plot-all.eps')
```

B.1.3 绘制 $x - t$ 回归图

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4
5 # Generate some data
6 np.random.seed(0)
7 # 1000 random numbers from a normal distribution in [0, 10]
8 X = np.random.normal(0, 10, 1000)
9
10 df = pd.read_csv('data/2024-05-17(16-57-14).csv')
11
12 X = np.sort(X)
13
14 # \dfrac{1}{2} 4.88 * x ^ 2 + 0.08 * x - 0.00
15
16 y = 4.88 * X ** 2 + 0.08 * X - 0.00
17
18 # Plot the data
19
20 plt.plot(X, y, label='Predicted data')
21 plt.scatter(df['t'], df['y'], label='Generated data')
22 plt.title('Plot of the data')
23 plt.xlabel('time')
24 plt.ylabel('x-axis')
25 plt.xlim(0, 0.2)
26 plt.ylim(0, 0.2)
27 plt.legend()
28 plt.savefig('figures/plot-x-regression.eps')
```

B.1.4 绘制 $y - t$ 回归图

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4
5 # Generate some data
6 np.random.seed(0)
7 # 1000 random numbers from a normal distribution in [0, 10]
8 X = np.random.normal(0, 1, 1000)
9
10 df = pd.read_csv('data/2024-05-17(16-57-14).csv')
11
12 X = np.sort(X)
13
14 # 1.75 * x + 0.06
15
16 y = 1.75 * X + 0.06
17
18 # Plot the data
19
20 plt.plot(X, y, label='Predicted data')
21 plt.scatter(df['t'], df['x'], label='Generated data')
22 plt.title('Plot of the data')
23 plt.xlabel('time')
24 plt.ylabel('y-axis')
25 plt.xlim(0, 0.2)
26 plt.ylim(0, 0.4)
27 plt.legend()
28 plt.savefig('figures/plot-y-regression.eps')
```

B.2 回归

```
1 from sklearn.linear_model import LinearRegression
2 import pandas as pd
3 import os
4
5 def analysis(file_name):
6     df = pd.read_csv(file_name)
7
8     df['t_2'] = df['t'] ** 2
9
10    df.rename(columns={'t': 't_1', 'y': 'target'}, inplace=True)
11
12    X = df[['t_2', 't_1']]
13    y = df['target']
14
```

```

15     model = LinearRegression()
16     model.fit(X, y)
17
18     loss = ((model.predict(X) - y) ** 2).sum()
19
20     # print(f"{file_name} & {model.coef_[0]:.2f} & {model.coef_[1]:.2f} & {model.intercept_:~.2f} & {loss:.8f} \\\\")
21
22     df = pd.read_csv(file_name)
23
24     df.rename(columns={'x': 'target'}, inplace=True)
25
26     X = df[['t']]
27     y = df['target']
28
29     model = LinearRegression()
30     model.fit(X, y)
31
32     loss = ((model.predict(X.values) - y) ** 2).sum()
33
34     print(f"{file_name} & {model.coef_[0]:.2f} & {model.intercept_:~.2f} & {loss:.8f} \\\\")
35
36
37
38 for file in os.listdir('data'):
39     if file.endswith('.csv'):
40         analysis(f'data/{file}')

```

C 仓库

本文的仓库地址为 <https://github.com/7086cmd/expr-2024-report.git>.