Coefficient Estimation and Simulation for Measured Data

Batuhan Berk Şahin
dept. of Computer Engineering
Ankara University
batuberkshn@gmail.com

***Abstract*—This report explores coefficient estimation using the Least Squares Method and its application in simulating free-fall motion. The Least Squares Method results in estimated coefficients. These coefficients are then used to simulate the motion of a free-falling object, highlighting the practical implications of coefficient estimation and error management in engineering applications.**

I. Introduction

This report was written for the third assignment of the Modeling and Simulation course. This assignment consists of two parts. The first is independent of the second. Therefore, the report is prepared in two parts.

II. Requirements

There are no different requirements than before. The **R2023b** version of MATLAB will be sufficient to run the required source files.

III. Preparation of the Assignment

A. Estimation the Coefficients of the Polynomial

1) Method of the Estimation: In the first part of the assignment, some measurements were taken on an observed system. These measurements have a Gaussian error distribution. We are also informed that this system is modeled as a second order polynomial.

As a result of the research, many methods were found on how to estimate the coefficients of a system modeled as a quadratic polynomial.

The top set of these methods is Regression analysis and a subset is Polynomial Regression. In this assignment, **Least Squares Method** was preferred among Polynomial Regression methods.

For a given n (x, y) lines, the parameter estimation equation according to the least square method is as follows [1]:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^k \\ 1 & x_2 & x_2^2 & \dots & x_2^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^k \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} \tag{1}$$

Therefore:

$$y = Xa \tag{2}$$

This equation can be solved by multiplying both sides by the transpose of X^T

$$X^T y = X^T X a \tag{3}$$

$$a = (X^T X)^{-1} X^T y \tag{4}$$

We can make the necessary calculations for a and estimate the coefficients.

2) Finalizing the estimation: According to the derived equations of the Least Square Method, an algorithm was designed to solve the equation via MATLAB.

The algorithm basically handles matrix calculations and is designed as follows:

Algorithm 1 Estimation of Coefficients using Least Squares Method

0: Initialize arrays `xi` and `Yi` with measurement data
0: Create a matrix `X` with ones in the first column and powers of `xi` in the next two columns
0: **for** i from 1 to 2 **do**
0: Update the $i+1$-th column of `X` with xi^i
0: **end for**
0: Calculate the coefficients by solving the linear system
0: `coefficients` $= (X^T \cdot X)^{-1} \cdot X^T \cdot Yi$ =0

`This pseudocode generated by ChatGPT`

As a result of all this, when we process the noisy data we have, the coefficients we estimate are as follows:

TABLE I
Estimation of Coefficients using Least Squares Method

Coefficient	**Value**
n^0	200.144250
n^1	50.284380
n^2	-4.935419

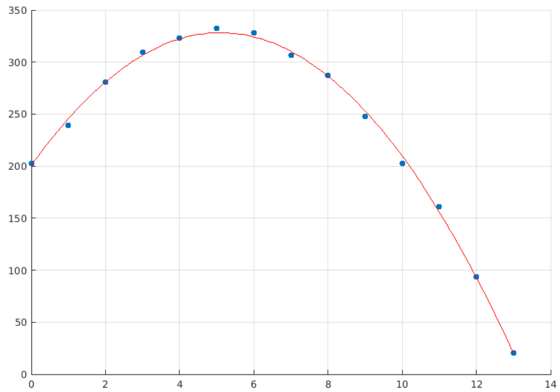

Fig. 1. Estimated Second-Order Polynomial

B. Implementing the Estimated Polynomial in One-Dimensional Motion

The polynomial estimated in the previous part of the assignment is given by Fig. 1.

In this part of the assignment we need to investigate the motion of a free-falling body in a frictionless environment. Fig. 1 and its coefficients, we can equate it to the motion of a free-falling body.

The equation for the height of a body in free fall in a frictionless environment with respect to time is as follows:

$$h(t) = h_{\text{initial}} + v_{\text{initial}} \cdot t - \frac{1}{2}gt^2 \tag{5}$$

Since Equation (5) is also a quadratic polynomial, we can equate the coefficient we find.

$$\begin{aligned} h_{\text{initial}} &: 200.144250\text{m} \\ v_{\text{initial}} &: 50.284380\text{m/s} \\ \frac{1}{2}g &: -4.935419\text{m/s²} \end{aligned}$$

Hence, it can be adapted to free fall motion. From this polynomial, the time-dependent motion function can be obtained.

The final equation is as follows:

$$h(t) = 200.144250 + 50.284380 \cdot t - 4.935419 \cdot t^2 \tag{6}$$

where:

$$\begin{aligned} h(t) &: \text{Height at time } t \\ t &: \text{Time} \end{aligned}$$

With the time-dependent height equation, noisy error can simulate measured values.

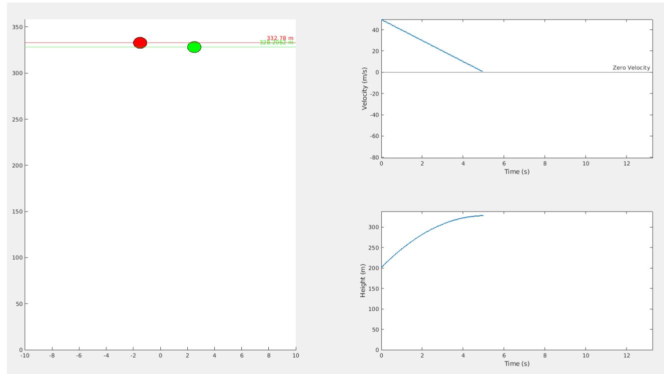

Fig. 2. Screenshot of Simulation

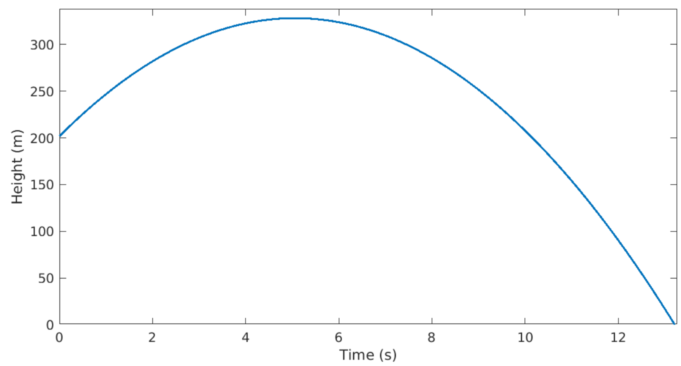

Fig. 3. Height-Time Graph

IV. Simulation

The simulation part was carried out in an animated way as in the previous assignments. Unlike the previous assignments, the animation part runs converging to real time (30 frames per second). Of course, there are 1 second breaks in between to examine the measured and simulated objects.

The simulation set includes, animation of measured and simulated objects, height-time (Fig. 3) plot of the simulated object, velocity-time (Fig. 4) plot of the simulated object. A Zero Velocity axis has also been added to the velocity-time plot.

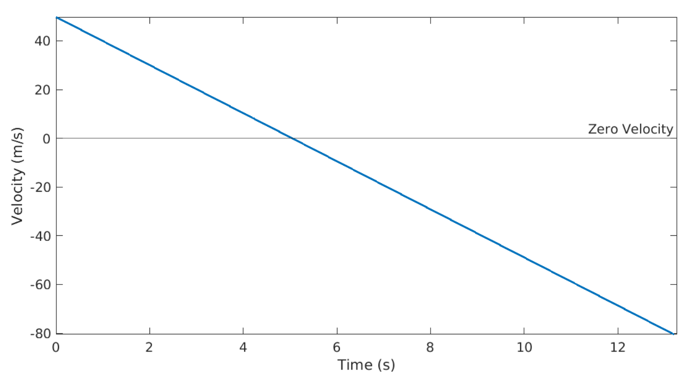

Fig. 4. Velocity-Time Graph

V. Conclusion

In this assignment, basically regression analysis, coefficient estimation of a quadratic polynomial with least square method was learned and practiced. Systems contain a margin of error and it is our job as engineers to understand and classify them.

The measured and predicted height values are easily understandable through the simulation. It is up to the engineer to make sense of these erroneous measurements and what can be done afterwards to make an estimate.

References

[1] Least Square Method Fitting, https://mathworld.wolfram.com/LeastSquaresFittingPolynomial.html