ECE 5390 Practicum Assignment 1

Modeling from Experimental Data

Griffin Davis

January 27, 2022

# 1 Introduction

Practicum 1 explores the modeling of first and second order systems given the data points of the system's step response. System modeling is completed and plotted using MATLAB.

The rest of this report is organized as follows. Section 2 covers the search method for parametric identification of the system. Section 3 discusses the results of the search in reference to a first order system. Section 4 discusses the results of the search in reference to a second order system. Section 5 discusses the practicum as a whole and draws conclusions.

# 2 Parametric Identification Through N-Search

Practicum 1 calls for the approximation of a first and second order system through parametric identification of terms *k*,α,and β. A brute force method using N-search is used to determine the parameters yielding the minimum Least Squares Error.

Before searching, an estimate of αand *k* for the first order system is determined using equations:

The search involves iterating a set number of times, tweaking the magnitude by which parameters *k* and α—or β in the second order—are changed and either increasing or decreasing values by that magnitude in accordance with the operation yielding the lower least squares error.

MATLAB code of the search methods for the first and second order systems can be found in **Appendix A** and **Appendix B**.

# 3 First Order System Modeling

The N-Search method discussed in section 2 assuming a first order system yields a transfer function with parameters:

The above values result in a least squares error of 0.101436. Further tests with a greater number of iterations and more precise alterations in magnitude determined this to be the lowest obtainable least squares error within a reasonable search time.

The experimental model can be seen plotted against the provided data in **Figure 1**.

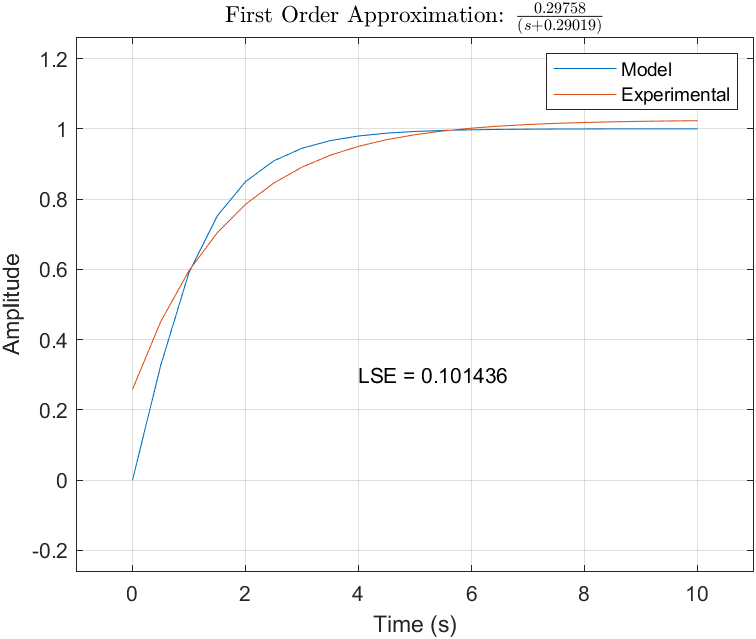


Fig. 1: First Order Approximation of Step Response

# 4 Second Order System Modeling

Applying the search method to a second order assumption first requires an assumption of β. β is first tested as equal to α, but this assumption results in an exponential step response rather than a critically damped, logarithmic response. α is doubled and the equivalence tested again for a least squares error of 0.0636809. The parameters are identified as:

Further tests were explored and again the least squares error of ~0.064 was the lowest obtainable error within a reasonable search time.

The experimental model can be seen plotted against the provided data in **Figure 2**.

# 5 Discussion and Conclusion

Due to the lower least squares error of 0.0636809 found in the second order approximation of the step response, the system is more likely to be a second order system than a first order system. It can also be seen in **Figures 1 & 2** that the second order system reaches steady state at close to the same period of *4*τ than the first order system. The second order system comes closer to an initial start of 0 as well and thus more accurately represents a system step response.

Practicum 1 is a great introduction to system modeling and parametric identification through the relatively complex algorithm required to accurately calculate the parameters through N-search.

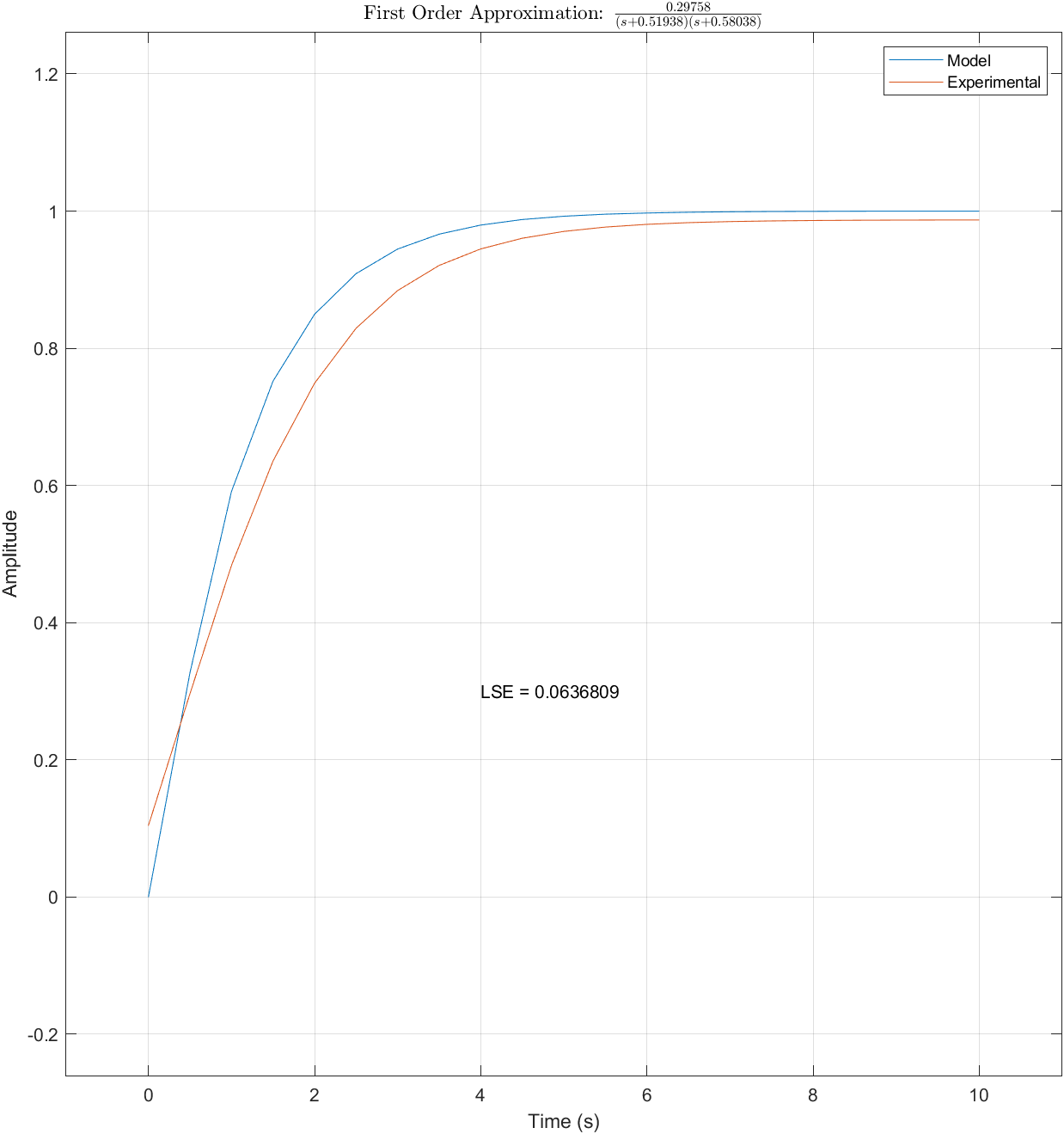


Fig. 2: Second Order Approximation of Step Response

# Appendices

## Appendix (A): First Order Search

for i=1:1000

if LSE == 0

break

end

if mod(i,2)

% work on alpha

alpha\_pos = alpha + aMag\_pos;

alpha\_neg = alpha - aMag\_neg;

aLSE\_pos = getLSE(alpha\_pos, k);

aLSE\_neg = getLSE(alpha\_neg, k);

if aLSE\_pos > prevLSE && aLSE\_neg > prevLSE

aMag\_pos = aMag\_pos - mag\_order;

aMag\_neg = aMag\_neg - mag\_order;

else

if aLSE\_pos > aLSE\_neg

aMag\_pos = aMag\_pos - mag\_order;

alpha = alpha\_neg;

else

aMag\_neg = aMag\_neg - mag\_order;

alpha = alpha\_pos;

end

end

else

% work on k

k\_pos = k + kMag\_pos;

k\_neg = k - kMag\_neg;

kLSE\_pos = getLSE(alpha, k\_pos);

kLSE\_neg = getLSE(alpha, k\_neg);

if kLSE\_pos > prevLSE && kLSE\_neg > prevLSE

kMag\_pos = kMag\_pos - mag\_order;

kMag\_neg = kMag\_neg - mag\_order;

else

if kLSE\_pos > kLSE\_neg

kMag\_pos = kMag\_pos - mag\_order;

k = k\_neg;

else

kMag\_neg = kMag\_neg - mag\_order;

k = k\_pos;

end

end

end

prevLSE = LSE;

LSE = getLSE(alpha, k);

end

## Appendix (B): Second Order Search

for i=1:1000

if LSE == 0

break

end

m = mod(i,3);

if m==2

% work on alpha

alpha\_pos = alpha + aMag\_pos;

alpha\_neg = alpha - aMag\_neg;

aLSE\_pos = getLSE(alpha\_pos, k);

aLSE\_neg = getLSE(alpha\_neg, k);

if aLSE\_pos > prevLSE && aLSE\_neg > prevLSE

aMag\_pos = aMag\_pos - mag\_order;

aMag\_neg = aMag\_neg - mag\_order;

else

if aLSE\_pos > aLSE\_neg

aMag\_pos = aMag\_pos - mag\_order;

alpha = alpha\_neg;

else

aMag\_neg = aMag\_neg - mag\_order;

alpha = alpha\_pos;

end

end

elseif m==1

% work on k

k\_pos = k + kMag\_pos;

k\_neg = k - kMag\_neg;

kLSE\_pos = getLSE(alpha, k\_pos);

kLSE\_neg = getLSE(alpha, k\_neg);

if kLSE\_pos > prevLSE && kLSE\_neg > prevLSE

kMag\_pos = kMag\_pos - mag\_order;

kMag\_neg = kMag\_neg - mag\_order;

else

if kLSE\_pos > kLSE\_neg

kMag\_pos = kMag\_pos - mag\_order;

k = k\_neg;

else

kMag\_neg = kMag\_neg - mag\_order;

k = k\_pos;

end

end

else

% work on beta

beta\_pos = beta + bMag\_pos;

beta\_neg = beta - bMag\_neg;

bLSE\_pos = get2LSE(alpha, beta\_pos, k);

bLSE\_neg = get2LSE(alpha, beta\_neg, k);

if bLSE\_pos > prevLSE && bLSE\_neg > prevLSE

bMag\_pos = bMag\_pos - mag\_order;

bMag\_neg = bMag\_neg - mag\_order;

else

if bLSE\_pos > bLSE\_neg

bMag\_pos = bMag\_pos - mag\_order;

b = beta\_neg;

else

bMag\_neg = bMag\_neg - mag\_order;

b = beta\_pos;

end

end

end

prevLSE = LSE;

LSE = get2LSE(alpha, beta, k);

end