

Design and Interpretation of Experiments: Signal Processing, Design of Experiments and System Identification

Computer Classes Assignments and Final Project Report

Group - 3
YOUSUF - 126656

ADITYA - 125904

Content – (Bolded headings are employed to highlight key sections where observations, reasoning, or specific topics have been discussed.)

Table of Figures.....	9
1. Experiment-1:Calibration-1 Exercise Report	19
1.1 Introduction	19
1.2 Harmonic Oscillator model	19
1.3 Model Overview.....	20
1.4 Parameter Variation - Observation	21
1.4.1 Varying Parameter k.....	23
1.4.2 Varying Parameter c	23
1.4.3 Varying Parameter ρ/m	23
1.5 Measurement.....	24
1.5.1 Measurement overview	24
1.5.2 Noise Levels and Significance - Observation	24
1.6 Comparison plot.....	25
1.6.1 Description of Plots	27
1.7 Comparison between numerical and measurements value	29
1.8 Cost Function Understanding.....	31
1.8.1 Computation of the Cost Function	31
1.8.2 Impact of Noise on Cost Evaluation.....	32
1.9 Optimization Analysis	33
1.9.1 Effect of Varying Initial Amplitude:	33
1.9.2 Figure 11: Plot of Harmonic Oscillation Displacement with Respect to Time	
35	
1.9.3 fminsearch Optimisation.....	37
1.9.3.1 for 3 variables.....	37

1.9.3.2	for 2 variables.....	38
1.9.4	fmincon Optimisation:.....	39
1.9.4.1	for 3 variables.....	40
1.9.4.1.1	Case 2: different tolerance and max iteration	41
1.10	Counterline for file u_meas01.mat.....	43
1.10.1	Observation:	44
2.	Experiment – 2: Calibration Exercise Report: Exploring Equation of Motion	45
2.1	Introduction	45
2.1.1	Objective	45
2.1.2	Scope	46
2.2	Review of Previous Calibration Using Least Squares	46
2.2.1	Methodology Recap	46
	Mathematical Framework:.....	46
	Role of Least Squares in Optimization:	47
	Outcomes Derived:	47
2.2.2	Notable Insights.....	47
2.2.2.a	Significant Observations:.....	47
	Challenges Encountered:	47
	Role of Regularization Techniques:	47
	Overall Assessment:	48
2.3	Regularization Techniques - Tikhonov	48
2.3.1	Theoretical Underpinnings of Tikhonov Regularization	48
	Theoretical Basis:	48
2.3.2	Implementation and Analysis.....	51
2.3.3	Algorithmic Enhancements	64

2.4 Application of λ to Least Squares Calibration	68
2.4.1 Integrating λ into Previous Calibration.....	68
2.4.2 Extended Data Analysis (if applicable)	68
umeas0	68
umeas01.....	71
umeas03.....	74
umeas05.....	77
umeas09.....	79
2.5 Comparison of optimized values from calibration experiment 1 and 2.	82
Approach 1: Direct Minimization using fmincon	82
In this approach:	82
Approach 2: Unconstrained Minimization using fminunc.....	83
In this approach:	83
2.5.1 Comparison:.....	83
Optimization Strategy:	84
Regularization:	84
Computational Complexity:.....	84
Flexibility:.....	84
2.5.2 Observations, and Conclusions.....	84
2.5.2.a Observational Insights	84
2.5.2.a.1 Sensitivity to λ Values:.....	85
2.5.2.a.2 Impact of Initial Guesses:.....	85
2.5.2.a.3 Trade-off Between Fit and Regularization:	85
2.5.2.a.4 Optimal λ Selection:.....	85
2.5.2.a.5 Generalization Across Initial Guesses:.....	85

2.5.2.b	Conclusive Findings	86
3.	Experiment -3: Regularization Methods.....	86
3.1	Introduction:.....	86
3.2	Classical Matrix Inversion:	86
3.2.1	Comparison:.....	88
3.3	Regularization Methods:	89
3.3.1	TSVD	89
3.3.1.a	Observation based on TSVD	89
3.3.2	Iterative	90
3.3.2.a	Observation based on iterations.....	90
3.3.3	Tikhonov.....	91
3.3.3.a	Observation based on Tikhonov	92
3.4	Variation of Regularization Parameter:.....	92
3.5	L-Curve Criterion:	93
4.	Experiment 4: Signal Analysis and System Identification 01: Exploring FFT and Equation of Motion	95
4.1	Introduction	95
4.2	Exploring the DFT and FFT	95
	Significance of Signal Processing in System Behavior Understanding	95
4.3	Signal Generation and DFT/FFT Analysis	96
4.3.1	Signal Generation.....	96
4.3.2	Sampling Frequency=200, Length=100, Signal_time = 0.5.....	101
4.3.3	Sampling Frequency=500, Length=1000, Signal_time = 2	103
4.3.4	Observation:	106
4.4	Discrete and Fast Fourier Transform	107

Observation	111
4.4.1 FFT was performed on the three signals at different frequencies and lengths.	
112	
Frequency vs. Amplitude Plot.....	112
4.4.2 Observation based on FFT of fundamental frequencies through the parametric analysis of signal frequency , length and superimposition	117
4.5 FFT Analysis of Noisy Signals	117
4.5.1 Description of myFFT.m	118
4.5.2 Noise=0.2 is added to $x=x_1+x_2+x_3$ and the generated plots are shown below:	
119	
4.5.3 Noise=0.5 is added to $x=x_1+x_2+x_3$ and the generated plots are shown below:	
120	
4.5.4 Noise=1 is added to $x=x_1+x_2+x_3$ and the generated plots are shown below:	
122	
4.5.5 Observation based on the parametric analysis of Noise.....	123
4.5.5.1 Effect on the Time Domain (Original Signal):	123
4.5.5.2 Effect on the Frequency Domain (FFT Spectrum):	123
4.5.5.3 Amplitude Spectrum Changes:	124
4.5.5.4 Phase Spectrum Changes:	124
4.6 Signal Averaging	124
4.6.1 Quality improvement.....	127
4.7 Equation of Motion Analysis.....	127
4.7.1 Graphs	128
4.7.2 FFT Analysis of Equation of Motion.....	129
4.7.3 Effects of Damping.....	129
4.7.3.1 Damping values	130

4.7.3.1.1	$\xi = 0$	130
4.7.3.1.1.1	Observation:.....	131
4.7.3.1.2	Noise 0.1	133
4.7.3.1.3	Noise 1.....	135
4.7.3.1.4	Noise 5.....	137
4.7.3.1.5	Observation:.....	139
4.7.3.2	param = [4.5 0 4] ;	140
4.7.3.3	param = [4.5 0 100] ;.....	143
4.7.3.4	param = [10 0 4] ;.....	146
4.7.3.5	Observation:	149
5.	Experiment: 5- Signal processing - 2	149
5.1	Introduction	150
5.1.1	Overview of Signal Analysis and System Identification.....	150
5.1.1.1	Signal Analysis:.....	150
5.1.2	System Identification:.....	150
5.1.3	Brief Explanation of Signal Processing Exercise Objectives	151
5.2	Real Signals Analysis.....	151
5.2.1	Signal Loading and Identification	151
5.2.1.1	Plotting Signals (Signal_01, Signal_02) as Time Signals	151
5.2.1.3	Identification of Input (x) and Output (y) Signals	160
5.3	Spectral Power Density (SPD) Calculation.....	162
5.3.1	Computing Average Squared Fourier Transforms (Sxx).....	162
5.3.2	Auto-Correlation Function	162
5.3.2.1	Computing and Plotting ϕ_{xx} and ϕ_{yy}	162
5.3.3	Cross-Correlation and Cross-Spectral Density	162

5.3.3.1	Computing and Plotting ϕ_{xy} and S_{yx}	162
5.3.4	Coherence Function	162
5.3.4.1	Computing and Plotting $\gamma^2(\omega)$	162
5.3.4.2	Specifying Fundamental Frequencies.....	164
5.4	Coherence Function in Differential Equations	165
5.4.1	Noisy Sinusoidal Functions	165
5.4.2	Computing Magnitude-Squared Coherence	166
5.5	Wavelet Transformation.....	167
5.5.1	Chirp Signal Analysis.....	168
5.5.1.1	Discrete Wavelet Transform (DWT) at Level 5 using Haar/db1 Wavelet	168
5.5.1.2	Coefficient Expansion and Signal Plotting.....	169
5.5.1.3	Expanding Discrete Wavelet Coefficients and Signal Plot.....	169
5.5.1.4	Analysis of Added and Concatenated Sine Signals	170
5.5.1.5	Observation:.....	171
5.6	Filter Analysis.....	172
5.6.1	Low Pass Filter Application on Noisy Sinusoidal Signal.....	173
5.6.2	Applying Filter and Interpretation of Results.....	174
6.	Experiment:6- Damage Detection	175
6.1	Un-damaged Beam Response and Natural Frequencies	175
6.1.1	Observation:	180
6.2	Effect of Damage Location and Severity.....	180
6.2.1	List of takeaways or observation through parametric analysis of fundamental response	194
6.3	Uniformity of Natural Frequency Changes	195
6.4	Impact of Noise Level (noise_level)	195

6.4.1	Observation:.....	197
6.5	Influence of Damping Ratio (zeta)	197
6.5.1	Observation:	199
6.6	Cross-Correlation and Cross-Spectral Density Analysis	200
7.	Additional Task.....	201
7.1	Introduction.....	201
7.2	Model Updating	202
7.2.1	Residuals Method and Optimization Algorithms	202
7.2.2	Residuals Method and Optimization Algorithms	205
7.3	Regularization Algorithms	206
7.4	Comparison of Results	208
7.5	Observations, Findings, and Conclusions	209
8.	Parameter Identification (20 points).....	210
8.1.1	System Calibration in Frequency Domain.....	210
8.2	Cost Function and Solution Uniqueness	212
8.3	Comparative Analysis.....	216
8.4	Interpretation of Results.....	216
9.	Signal Processing (15 points)	216
9.1	Data Acquisition and Signal Processing.....	216
9.2	Analysis of Signals	217
9.3	Interpretation of Results.....	237
9.3.1	Time-Domain Plot	237
9.3.2	FFT Analysis	238
9.3.3	Power Spectra Density and Cross-Spectral Density	238
9.3.4	Coherence Function	238

9.3.5	Wavelet Transforms.....	238
10.	Tomography (15 points)	239
10.1	Image Reconstruction Techniques.....	239
10.1.1	How to determine elbow point or best trade-off point? (Jonas, 2010).....	239
10.1.2	Tikhonov Regularization:.....	240
10.1.3	Truncated Singular Value Decomposition (TSVD):.....	240
10.1.4	Iterative Method:.....	242
10.2	Optimization and Result Plotting.....	243
10.3	Analysis and Conclusion.....	245
10.4	Conclusion.....	245
11.	Appendices.....	246
12.	References.....	261

Table of Figures

FIGURE 1: TIME EVOLUTION OF A HARMONIC OSCILLATOR ($k = 5, c = 0.6, \rho = 4$) OVER 20 SECONDS WITH A TIME STEP OF 0.05 SECONDS.....	21
FIGURE 2: PLOT OF HARMONIC OSCILLATOR FOR DIFFERENT VALUES OF STIFFNESS : TOP-LEFT ($k=4$), TOP-RIGHT ($k=6$) AND BOTTOM ($k=2$)	21
FIGURE 3: PLOT OF HARMONIC OSCILLATOR FOR DIFFERENT VALUES OF DAMPING COEFFICIENT: $[4 \leq 4.5], [0:0.05:20] - ([k \leq$ $\rho/\text{M}], [\text{FREQUENCY OF TIME OF } 0.05 \text{ FOR } 20 \text{ SEC}]$: TOP-RIGHT ($c = 0.0001$), TOP-LEFT ($c = 1$), BOTTOM-RIGHT ($c = 0.65$), BOTTOM-LEFT ($c = 10$).	22

FIGURE 4: PLOT OF HARMONIC OSCILLATOR FOR DIFFERENT VALUES OF MASS OF OSCILLATOR: [4 0.65 M],[0:0.05:20] - ([K C RHO/M],[TIME DIVISION OF 0.05 FOR 20 SEC] : TOP-RIGHT (M = 1.5), TOP-LEFT (M = 100), BOTTOM (M =4.5).....	22
FIGURE 5: PLOT OF HARMONIC OSCILLATION OF A BODY FOR THE DATA FILE OF UMEAS0.MAT (NO NOISE)	25
FIGURE 6: PLOT OF HARMONIC OSCILLATION OF A BODY FOR THE DATA FILE OF UMEAS01.MAT	26
FIGURE 7: PLOT OF HARMONIC OSCILLATION OF A BODY FOR THE DATA FILE OF UMEAS03.MAT	26
FIGURE 8: PLOT OF HARMONIC OSCILLATION OF A BODY FOR THE DATA FILE OF UMEAS05.MAT	27
FIGURE 9: PLOT OF HARMONIC OSCILLATION OF A BODY FOR THE DATA FILE OF UMEAS09.MAT	27
FIGURE 10: PLOT FOR COMPARISON BETWEEN UMEAS0 AND NUMERICAL SOLUTION USING VALUES OF K, C, M.	29
FIGURE 11: PLOT FOR COMPARISON BETWEEN UMEAS01 AND NUMERICAL SOLUTION USING VALUES OF K, C, M.	29
FIGURE 12: PLOT FOR COMPARISON BETWEEN UMEAS03 AND NUMERICAL SOLUTION USING VALUES OF K, C, M.	30
FIGURE 13: PLOT FOR COMPARISON BETWEEN UMEAS05 AND NUMERICAL SOLUTION USING VALUES OF K, C, M.	30
FIGURE 14: PLOT FOR COMPARISON BETWEEN UMEAS09 AND NUMERICAL SOLUTION USING VALUES OF K, C, M.	31
FIGURE 15: VARYING INITIAL AMPLITUDE; LEFT-TOP WITH INITIAL AMPLITUDE OF 1, RIGHT-TOP WITH INITAL AMPLITUDE OF 10, CENTER-BOTTOM WITH INITIAL AMPLITUDE OF 5.....	33
FIGURE 16: PLOT OF HARMONIC OSCILLATION DISPLACEMENT W.R.T TO TIME FOR DIFFERENT INITIAL VALUES; TOP_RIGHT, X = 33.894 18.918 80.637, x0 = [60, 5, 20], TOP_LEFT, x = 26.0247 5.8983 25.1416 x0 = [30, 5, 20], BOTTOM_LEFT, X = 32.561 18.712 79.759 ,x0 = [60, 10, 20], BOTTOM_RIGHT, X = 39.042 19.684 83.906, x0 = [60, 5, 10]	35
FIGURE 17: PLOT OF OPTIMIZATION SHOWING HOW CURRENT FUNCTION VALUE (RMS) IS OPTIMIZED FOR EACH ITERATION WITHIN A THRESHOLD.....	37
FIGURE 18: PLOT OF HARMONIC OSCILLATOR FOR MEASURED PLOT (UMEAS03), INITIAL GUESS AND FITTED DATA (VALUE OF HARMONIC VARIABLES AFTER OPTIMIZATION).....	38
FIGURE 19: PLOT OF OPTIMIZATION SHOWING HOW CURRENT FUNCTION VALUE (RMS) IS OPTIMIZED FOR EACH ITERATION WITHIN A THRESHOLD.....	39
FIGURE 20: PLOT OF HARMONIC OSCILLATOR FOR MEASURED PLOT (UMEAS03), INITIAL GUESS AND FITTED DATA (VALUE OF HARMONIC VARIABLES AFTER OPTIMIZATION).....	39
FIGURE 21: PLOT OF OPTIMIZATION SHOWING HOW CURRENT FUNCTION VALUE (RMS) IS OPTIMIZED FOR EACH ITERATION WITHIN A THRESHOLD.....	41
FIGURE 22: PLOT OF HARMONIC OSCILLATOR FOR MEASURED PLOT (UMEAS03), INITIAL GUESS AND FITTED DATA (VALUE OF HARMONIC VARIABLES AFTER OPTIMIZATION).....	41
FIGURE 23: PLOT OF OPTIMIZATION SHOWING HOW CURRENT FUNCTION VALUE (RMS) IS OPTIMIZED FOR EACH ITERATION WITHIN A THRESHOLD	42
FIGURE 24: PLOT OF HARMONIC OSCILLATOR FOR MEASURED PLOT (UMEAS03), INITIAL GUESS AND FITTED DATA (VALUE OF HARMONIC VARIABLES AFTER OPTIMIZATION).....	43
FIGURE 25: CONTOUR PLOTS SHOWING THE VALUES OF K,C,M WHEN TWO VAIABLES ARE OPTIMISED WHILE KEEPING OTHER ONE AS CONSTANT, MEASUREMENT DATA OF UMEAS04 TAKEN INTO CONSIDERATION.....	44

FIGURE 26: THE EXACT SOLUTION (THIN LINES) AND TIKHONOV REGULARIZED SOLUTIONS λ (THICK LINES) FOR THREE VALUES OF λ CORRESPONDING TO OVER-SMOOTHING, APPROPRIATE SMOOTHING, AND UNDER-SMOOTHING (HANSEN & O'LEARY, 1993)	49
FIGURE 27: THE GENERIC L-CURVE FOR STANDARD-FORM TIKHONOV REGULARIZATION WITH $x_0 = 0$; THE POINTS MARKED BY THE CIRCLES CORRESPOND TO THE REGULARIZATION PARAMETERS $\lambda = 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$ AND 1 (HANSEN & O'LEARY, 1993)	50
FIGURE 28: THE FIGURE SHOWS VARIATIONS OF THE COST FUNCTION VALUE DURING THE ITERATION OF THE MULTIPLE RESIDUAL MEAN SQUARE LOSS FUNCTION, USING OPTIMIZATION_TIKO.M. REGULARIZATION PARAMETER SET TO 0.01.....	53
FIGURE 29: HARMONIC OSCILLATOR, PLOTTED FOR UMEAS03 AS MEASURED DATA, INITIAL GUESS IN THE ABOVE TABLE.	53
FIGURE 30: CONTOUR PLOTS FOR OPTIMIZATION BASED ON TWO VARIABLES : $(k, m), (k, c), (c, m)$	54
FIGURE 31: THE FIGURE SHOWS VARIATIONS OF THE COST FUNCTION VALUE DURING THE ITERATION OF THE MULTIPLE RESIDUAL MEAN SQUARE LOSS FUNCTION, USING OPTIMIZATION_TIKO.M. REGULARIZATION PARAMETER SET TO 0.1.....	55
FIGURE 32: HARMONIC OSCILLATOR, PLOTTED FOR UMEAS03 AS MEASURED DATA, INITIAL GUESS IN THE ABOVE TABLE.	55
FIGURE 33: CONTOUR PLOTS FOR OPTIMIZATION BASED ON TWO VARIABLES : $(k, m), (k, c), (c, m)$	56
FIGURE 34: THE FIGURE SHOWS VARIATIONS OF THE COST FUNCTION VALUE DURING THE ITERATION OF THE MULTIPLE RESIDUAL MEAN SQUARE LOSS FUNCTION, USING OPTIMIZATION_TIKO.M. REGULARIZATION PARAMETER SET TO 0.8.....	57
FIGURE 35: HARMONIC OSCILLATOR, PLOTTED FOR UMEAS03 AS MEASURED DATA, INITIAL GUESS IN THE ABOVE TABLE	57
FIGURE 36: CONTOUR PLOTS FOR OPTIMIZATION BASED ON TWO VARIABLES : $(k, m), (k, c), (c, m)$	58
FIGURE 37: THE FIGURE SHOWS VARIATIONS OF THE COST FUNCTION VALUE DURING THE ITERATION OF THE MULTIPLE RESIDUAL MEAN SQUARE LOSS FUNCTION, USING OPTIMIZATION_TIKO.M. REGULARIZATION PARAMETER SET TO 1.....	59
FIGURE 38: HARMONIC OSCILLATOR, PLOTTED FOR UMEAS03 AS MEASURED DATA, INITIAL GUESS IN THE ABOVE TABLE.	59
FIGURE 39: CONTOUR PLOTS FOR OPTIMIZATION BASED ON TWO VARIABLES : $(k, m), (k, c), (c, m)$	61
FIGURE 40: THE FIGURE SHOWS VARIATIONS OF THE COST FUNCTION VALUE DURING THE ITERATION OF THE MULTIPLE RESIDUAL MEAN SQUARE LOSS FUNCTION, USING OPTIMIZATION_TIKO.M. REGULARIZATION PARAMETER SET TO 10.....	61
FIGURE 41: HARMONIC OSCILLATOR, PLOTTED FOR UMEAS03 AS MEASURED DATA, INITIAL GUESS IN THE ABOVE TABLE. THE EFFECT WE CAN OBSERVE DUE TO HIGH LAMDA, GLOBAL OPTIMIZATION FUNCTION IS OPTIMIZING THE REGULARIZATION FUNCTION NOT THE LOSS FUNCTION.....	62
FIGURE 42: CONTOUR PLOTS FOR OPTIMIZATION BASED ON TWO VARIABLES : $(k, m), (k, c), (c, m)$	64
FIGURE 43: L-CURVE FOR U_MEAS0	65
FIGURE 44: L-CURVE FOR U_MEAS01	65
FIGURE 45: L-CURVE FOR U_MEAS3	66
FIGURE 46: L-CURVE FOR U_MEAS05	66
FIGURE 47: L-CURVE FOR U_MEAS09	67
FIGURE 48: THE FIGURE SHOWS VARIATIONS OF THE COST FUNCTION VALUE DURING THE ITERATION OF THE RESIDUAL MEAN SQUARE LOSS FUNCTION, USING OPTIMIZATION_TIKO.M. REGULARIZATION PARAMETER SET TO 0.4.....	68
FIGURE 49: HARMONIC OSCILLATOR, PLOTTED FOR UMEAS03 AS MEASURED DATA, INITIAL GUESS AS 20, 2, 80 ..	69
FIGURE 50: CONTOUR PLOTS FOR OPTIMIZATION BASED ON TWO VARIABLES : $(k, m), (k, c), (c, m)$	71

FIGURE 51: THE FIGURE SHOWS VARIATIONS OF THE COST FUNCTION VALUE DURING THE ITERATION OF THE RESIDUAL MEAN SQUARE LOSS FUNCTION, USING OPTIMIZATION_TIKO.M. REGULARIZATION PARAMETER SET TO 0.4.....	71
FIGURE 52: HARMONIC OSCILLATOR, PLOTTED FOR UMEAS01 AS MEASURED DATA, INITIAL GUESS AS 20 , 2 ,80.....	72
FIGURE 53: CONTOUR PLOTS FOR OPTIMIZATION BASED ON TWO VARIABLES : (K,M), (K,C), (C,M)	74
FIGURE 54: THE FIGURE SHOWS VARIATIONS OF THE COST FUNCTION VALUE DURING THE ITERATION OF THE RESIDUAL MEAN SQUARE LOSS FUNCTION, USING OPTIMIZATION_TIKO.M. REGULARIZATION PARAMETER SET TO 0.4.....	74
FIGURE 55: HARMONIC OSCILLATOR, PLOTTED FOR UMEAS03 AS MEASURED DATA, INITIAL GUESS 20, 2, 80 ..	75
FIGURE 56: CONTOUR PLOTS FOR OPTIMIZATION BASED ON TWO VARIABLES : (K,M), (K,C), (C,M)	76
FIGURE 57: THE FIGURE SHOWS VARIATIONS OF THE COST FUNCTION VALUE DURING THE ITERATION OF THE RESIDUAL MEAN SQUARE LOSS FUNCTION, USING OPTIMIZATION_TIKO.M. REGULARIZATION PARAMETER SET TO 0.4.....	77
FIGURE 58: HARMONIC OSCILLATOR, PLOTTED FOR UMEAS05 AS MEASURED DATA, INITIAL GUESS 20, 2, 80 ..	77
FIGURE 59: CONTOUR PLOTS FOR OPTIMIZATION BASED ON TWO VARIABLES : (K,M), (K,C), (C,M)	79
FIGURE 60: THE FIGURE SHOWS VARIATIONS OF THE COST FUNCTION VALUE DURING THE ITERATION OF THE RESIDUAL MEAN SQUARE LOSS FUNCTION, USING OPTIMIZATION_TIKO.M. REGULARIZATION PARAMETER SET TO 0.4.....	79
FIGURE 61: HARMONIC OSCILLATOR, PLOTTED FOR UMEAS09 AS MEASURED DATA, INITIAL GUESS 20, 2, 80 ..	80
FIGURE 62: CONTOUR PLOTS FOR OPTIMIZATION BASED ON TWO VARIABLES : (K,M), (K,C), (C,M)	82
FIGURE 63: REFERENCE IMAGE	87
FIGURE 64: RECONSTRUCTION USING NO REGULARIZATION	88
FIGURE 65: RECONSTRUCTION BASED ON TSVD FOR DIFFERENT VALUES OF K FROM 50 TO 1700.	89
FIGURE 66: RECONSTRUCTION BASED ON ITERATIVE FOR DIFFERENT NUMBER OF ITERATIONS FROM 10 TO 50, WITH THE INCREMENT OF 10.	90
FIGURE 67: RECONSTRUCTION BASED ON TIKHONOV FOR DIFFERENT ALPHAS FROM 1 TO 7, WITH THE INCREMENT OF 1.....	91
FIGURE 68: L-CURVE BASED ON ITERATIVE METHOD.....	93
FIGURE 69: L CURVE TIKHONOV	94
FIGURE 70: L-CURVE TSVD.....	94
FIGURE 71: ORIGINAL INPUT SIGNAL FOR X=x1.....	97
FIGURE 72: PHASE SPECTRUM OF THE INPUT SIGNAL AS A FUNCTION OF FREQUENCY FOR X = X1	97
FIGURE 73: ORIGINAL INPUT SIGNAL FOR X = X1 + X2	98
FIGURE 74: PHASE SPECTRUM OF THE INPUT SIGNAL AS A FUNCTION OF FREQUENCY FOR X = X1 + X2	99
FIGURE 75: ORIGINAL INPUT SIGNAL FOR X = X1 + X2 + X3.....	100
FIGURE 76: PHASE SPECTRUM OF THE INPUT SIGNAL AS A FUNCTION OF FREQUENCY FOR X = X1 + X2 + X3	100
FIGURE 77: ORIGINAL INPUT SIGNAL FOR X = X1	101
FIGURE 78: PHASE SPECTRUM OF THE INPUT SIGNAL AS A FUNCTION OF FREQUENCY FOR X = X1	101
FIGURE 79: ORIGINAL INPUT SIGNAL FOR X = X1 + X2	102
FIGURE 80: PHASE SPECTRUM OF THE INPUT SIGNAL AS A FUNCTION OF FREQUENCY FOR X = X1 + X2	102

FIGURE 81: PHASE SPECTRUM OF THE INPUT SIGNAL AS A FUNCTION OF FREQUENCY FOR $X = X_1 + X_2 + X_3$	103
FIGURE 82: PHASE SPECTRUM OF THE INPUT SIGNAL AS A FUNCTION OF FREQUENCY FOR $X = X_1 + X_2 + X_3$	103
FIGURE 83: ORIGINAL INPUT SIGNAL FOR $X = X_1$	104
FIGURE 84: PHASE SPECTRUM OF THE INPUT SIGNAL AS A FUNCTION OF FREQUENCY FOR $X = X_1$	104
FIGURE 85: ORIGINAL INPUT SIGNAL FOR $X = X_1$	105
FIGURE 86: PHASE SPECTRUM OF THE INPUT SIGNAL AS A FUNCTION OF FREQUENCY FOR $X = X_1$	105
FIGURE 87: ORIGINAL INPUT SIGNAL FOR $X = X_1 + X_2 + X_3$	106
FIGURE 88: PHASE SPECTRUM OF THE INPUT SIGNAL AS A FUNCTION OF FREQUENCY FOR $X = X_1 + X_2 + X_3$	106
FIGURE 89: INTUITION OF TIME VS. FREQUENCY-BASED ANALYSIS (BY PHONICAL [CC BY-SA 4.0 (HTTPS://CREATIVECOMMONS.ORG/LICENSES/BY-SA/4.0]), FROM WIKIMEDIA COMMONS)	108
FIGURE 90: DISCRETE COSINE TRANSFORM (DCT) COEFFICIENTS PLOT.....	109
FIGURE 91: AVERAGE SPECTRUM PLOT - DISCRETE FOURIER TRANSFORM (DFT)	109
FIGURE 92: AVERAGE PHASE SPECTRUM PLOT - DISCRETE FOURIER TRANSFORM (DFT)	110
FIGURE 93: AVERAGE AMPLITUDE SPECTRUM PLOT - FAST FOURIER TRANSFORM (FFT)	110
FIGURE 94: AVERAGE PHASE SPECTRUM PLOT - FAST FOURIER TRANSFORM (FFT)	111
FIGURE 95: FREQUENCY VS. AMPLITUDE PLOT FOR $X = X_1$ [20 Hz]	112
FIGURE 96: FREQUENCY VS. AMPLITUDE PLOT FOR $X = X_1 + X_2$ [20 Hz, 60Hz]	113
FIGURE 97: FREQUENCY VS. AMPLITUDE PLOT FOR $X = X_1 + X_2 + X_3$ [20 Hz, 60 Hz, 80 Hz]	113
FIGURE 98: FREQUENCY VS. AMPLITUDE PLOT FOR $X = X_1$	114
FIGURE 99: FREQUENCY VS. AMPLITUDE PLOT FOR $X = X_1 + X_2$	115
FIGURE 100: FREQUENCY VS. AMPLITUDE PLOT FOR $X = X_1 + X_2 + X_3$	115
FIGURE 101: FREQUENCY VS. AMPLITUDE PLOT FOR $X = X_1$	116
FIGURE 102: FREQUENCY VS. AMPLITUDE PLOT FOR $X = X_1 + X_2$	116
FIGURE 103: FREQUENCY VS. AMPLITUDE PLOT FOR $X = X_1 + X_2 + X_3$	117
FIGURE 104: FREQUENCY VS. AMPLITUDE PLOT FOR $X = X_1 + X_2 + X_3$	119
FIGURE 105: ORIGINAL INPUT SIGNAL $X = X_1 + X_2 + X_3$	119
FIGURE 106: PHASE SPECTRUM OF THE INPUT SIGNAL AS A FUNCTION OF FREQUENCY FOR $X = X_1 + X_2 + X_3$	120
FIGURE 107: FREQUENCY VS. AMPLITUDE PLOT FOR $X = X_1 + X_2 + X_3$	120
FIGURE 108: ORIGINAL INPUT SIGNAL $X = X_1 + X_2 + X_3$	121
FIGURE 109: : PHASE SPECTRUM OF THE INPUT SIGNAL AS A FUNCTION OF FREQUENCY FOR $X = X_1 + X_2 + X_3$	121
FIGURE 110: FREQUENCY VS. AMPLITUDE PLOT FOR $X = X_1 + X_2 + X_3$	122
FIGURE 111: ORIGINAL INPUT SIGNAL $X = X_1 + X_2 + X_3$	122
FIGURE 112: PHASE SPECTRUM OF THE INPUT SIGNAL AS A FUNCTION OF FREQUENCY FOR $X = X_1 + X_2 + X_3$	123
FIGURE 113: ORIGINAL AND AVERAGED PHASE SPECTRA PLOT.....	126
FIGURE 114: FREQUENCY VS. AMPLITUDE PLOT FOR ORIGINAL AND AVERAGED SPECTRA (FREQUENCY DOMAIN).....	126

FIGURE 115: ORIGINAL AND AVERAGED SIGNAL PLOT SHOWING AMPLITUDE VS. TIME (TIME DOMAIN)	127
FIGURE 116: C=0; AMPLITUDE SPECTRUM	130
FIGURE 117: C=0; PHASE SPECTRUM	131
FIGURE 118: WAVELET SPECTROGRAM FOR NO DAMPING (LEFT), WAVELET SPECTROGRAM FOR REFERENCE FROM (ADMIN, 2018; PERCIVAL & WALDEN, 2000; SHAIK, N.D.).....	131
FIGURE 119: NOISY HARMONIC OSCILLATOR	132
FIGURE 120: C=0; ORIGINAL INPUT SIGNAL	133
FIGURE 121: C=0.1; AMPLITUDE SPECTRUM.....	133
FIGURE 122: C=0.1; PHASE SPECTRUM	133
FIGURE 123: C = 0.1; SPECTROGRAM	134
FIGURE 124: C=0.1; ORIGINAL INPUT SIGNAL	134
FIGURE 125: NOISY HARMONIC OSCILLATOR	134
FIGURE 126: C=1; AMPLITUDE SPECTRUM.....	135
FIGURE 127: C=1; PHASE SPECTRUM	135
FIGURE 128: C=1; SPECTROGRAM.....	136
FIGURE 129: NOISY HARMONIC OSCILLATOR	136
FIGURE 130: C=1; ORIGINAL INPUT SIGNAL	136
FIGURE 131: C=5; AMPLITUDE SPECTRUM.....	137
FIGURE 132: C=5; PHASE SPECTRUM	137
FIGURE 133: C = 5; SPECTROGRAM	138
FIGURE 134: NOISY HARMONIC OSCILLATOR	138
FIGURE 135: C=5; ORIGINAL INPUT SIGNAL	139
FIGURE 136: NOISY HARMONIC OSCILLATOR	140
FIGURE 137: M=4; ORIGINAL INPUT SIGNAL	141
FIGURE 138: M=4; AMPLITUDE SPECTRUM.....	141
FIGURE 139: M=4; PHASE SPECTRUM	142
FIGURE 140: M = 4; SPECTROGRAM	142
FIGURE 141: NOISY HARMONIC OSCILLATOR	143
FIGURE 142: M=100; ORIGINAL INPUT SIGNAL	144
FIGURE 143: M=100; AMPLITUDE SPECTRUM.....	144
FIGURE 144: M=100; PHASE SPECTRUM	145
FIGURE 145: M=100; SPECTROGRAM.....	145
FIGURE 146: K = 10 ; NOISY HARMONIC OSCILLATOR.....	146
FIGURE 147: : K=10; ORIGINAL INPUT SIGNAL.....	147
FIGURE 148: K=10; AMPLITUDE SPECTRUM.....	147

FIGURE 149: K=10; PHASE SPECTRUM.....	148
FIGURE 150: K = 10; SPECTROGRAM	148
FIGURE 151: INPUT – AMPLITUDE & PHASE SPECTRUM	152
FIGURE 152: INPUT – AVERAGED, FULL SPECTRUM & POWER SPECTRAL DENSITY	153
FIGURE 153: INPUT – AUTOCORRELATION WITH LAGS.....	154
FIGURE 154: INPUT – PWELCH & SPECTROGRAM	155
FIGURE 155: OUTPUT – AMPLITUDE & PHASE SPECTRUM.....	156
FIGURE 156: OUTPUT – AVERAGED, FULL SPECTRUM & POWER SPECTRAL DENSITY	157
FIGURE 157: OUTPUT – AUTOCORRELATION WITH LAGS	158
FIGURE 158: OUTPUT – PWELCH & SPECTROGRAM.....	159
FIGURE 159: INPUT-OUTPUT CROSS CORRELATION	160
FIGURE 160: INPUT-OUTPUT –POWER SPECTRAL DENSITY & COHERENCE PWELCH.....	160
FIGURE 161: INPUT-OUTPUT FRF PLOT.....	161
FIGURE 162: INPUT-OUTPUT SPECTROGRAM.....	161
FIGURE 163: FIRST SIGNAL PLOT	163
FIGURE 164: SECOND SIGNAL PLOT	163
FIGURE 165: MAGNITUDE-SQUARED COHERENCE	164
FIGURE 166: DISPLACEMENT AND VELOCITY PLOT FOR F1 WITH K = 1	165
FIGURE 167: DISPLACEMENT AND VELOCITY PLOT FOR F2 WITH K = 1	165
FIGURE 168: FFT OF F1 AND F2 SIGNAL WITH K = 1.....	166
FIGURE 169: MAGNITUDE-SQUARED COHERENCE – LINEAR EQ. MOTION WITH K =1.....	167
FIGURE 170: MAGNITUDE-SQUARED COHERENCE – NON-LINEAR EQ. MOTION WITH K =1	167
FIGURE 171: FFT - AMPLITUDE AND PHASE OF $\text{Cos}(4x^2)$	168
FIGURE 172: SIGNAL PLOT AND EXPANDED DISCRETE WAVELET COEFFICIENT AT LEVEL 5	169
FIGURE 173: FFT OF SIGNAL X1 + X2	170
FIGURE 174: FFT OF SIGNAL [X1 X2].....	170
FIGURE 175: ANALYZED SIGNAL X1 + X2	171
FIGURE 176: ANALYZED SIGNAL [X1 X2]	171
FIGURE 177: ORIGINAL INPUT NOISY SIGNAL	173
FIGURE 178: AFTER N=50 FILTERING (RED CURVE).....	173
FIGURE 179: AFTER N=65 FILTERING (RED CURVE).....	174
FIGURE 180: AFTER N=100 FILTERING (RED CURVE).....	174
FIGURE 181: UNDAMAGED CANTILEVER BEAM.....	176
FIGURE 182: DIFFERENT MODES SHAPE	177
FIGURE 183: UNDAMAGED AT 2M	177

FIGURE 184: UMDAMAGED AT 3M	178
FIGURE 185: UNDAMAGED AT 5M	178
FIGURE 186: UNDAMAGED AT 6M	179
FIGURE 187: UNDAMAGED AT 10M	179
FIGURE 188: 0.2 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.1 FOR 2M BEAM	180
FIGURE 189: 0.2 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.1 FOR 5M BEAM	181
FIGURE 190: 0.2 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.1 FOR 6M BEAM	181
FIGURE 191: 0.2 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.1 FOR 10M BEAM	182
FIGURE 192: 0.5 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.1 FOR 2M BEAM	182
FIGURE 193: 0.5 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.1 FOR 5M BEAM	183
FIGURE 194: 0.5 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.1 FOR 6M BEAM	183
FIGURE 195: 0.5 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.1 FOR 10M BEAM	184
FIGURE 196: 0.5 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.3 FOR 2M BEAM	184
FIGURE 197: 0.5 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.3 FOR 5M BEAM	185
FIGURE 198: 0.5 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.3 FOR 6M BEAM	185
FIGURE 199: 0.5 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.3 FOR 10M BEAM	186
FIGURE 200: 0.5 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.8 FOR 2M BEAM	186
FIGURE 201: 0.5 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.8 FOR 5M BEAM	187
FIGURE 202: 0.5 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.8 FOR 6M BEAM	187
FIGURE 203: 0.5 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.8 FOR 10M BEAM	188
FIGURE 204: 0.8 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.1 FOR 2M BEAM	188
FIGURE 205: 0.8 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.1 FOR 5M BEAM	189
FIGURE 206: 0.8 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.1 FOR 6M BEAM	189
FIGURE 207: 0.8 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.1 FOR 10M BEAM	190
FIGURE 208: 0.8 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.3 FOR 2M BEAM	190
FIGURE 209: 0.8 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.3 FOR 5M BEAM	191
FIGURE 210: 0.8 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.3 FOR 6M BEAM	191
FIGURE 211: 0.8 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.3 FOR 10M BEAM	192
FIGURE 212: 0.8 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.8 FOR 2M BEAM	192
FIGURE 213: 0.8 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.8 FOR 5M BEAM	193
FIGURE 214: 0.8 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.3 FOR 6M BEAM	193
FIGURE 215: 0.8 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.8 FOR 10M BEAM	194
FIGURE 216: NOISE LEVEL OF 0.02, 0.5 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.4 FOR 3M BEAM.....	195
FIGURE 217: NOISE LEVEL OF 0.02, 0.5 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.4 FOR 6M BEAM.....	196
FIGURE 218: NOISE LEVEL OF 0.06, 0.5 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.4 FOR 3M BEAM.....	196

FIGURE 219: NOISE LEVEL OF 0.06, 0.5 POSITION OF DAMAGE LOCATION, WITH SEVERITY OF 0.4 FOR 6M BEAM.....	197
FIGURE 220: PLOT OF DAMPING FOR DAMAGE LOCATION OF 2	198
FIGURE 221: PLOT OF DAMPING FOR DAMAGE LOCATION OF 4	198
FIGURE 222: PLOT OF DAMPING FOR DAMAGE LOCATION OF END.....	199
FIGURE 223: PLOT OF CROSS-CORRELATION AND CROSS-SPECTRAL DENSITY BETWEEN DAMAGE LOCATION 2 AND 3.....	200
FIGURE 224: PLOT OF CROSS-CORRELATION AND CROSS-SPECTRAL DENSITY BETWEEN DAMAGE LOCATION 2 AND 4.....	201
FIGURE 225: PLOT BASED ON THE GIVEN DATA	203
FIGURE 226 THEORETICAL VS. MEASURED PLOT.....	204
FIGURE 227: OVERLAY OF HARMONIC OSCILLATION	204
FIGURE 228: MEASURED VS. PREDICTED PLOT	206
FIGURE 229: RESIDUALS PLOT	206
FIGURE 230: PLOT OF MEASURED OUTPUT AGAINST THE OUTPUT OBTAINED USING THE SVD METHOD.....	207
FIGURE 231: PLOT OF MEASURED OUTPUT AGAINST THE OUTPUT OBTAINED USING THE TSVD METHOD	207
FIGURE 232: PLOT OF MEASURED OUTPUT AGAINST THE OUTPUT OBTAINED USING THE ITERATIVE REGULARIZATION METHOD	208
FIGURE 233: COMPARISON OF FITTING METHODS	208
FIGURE 234: RESIDUALS PLOTS OF ALL THREE METHODS.....	209
FIGURE 235: RESIDUALS VALUE FOR ALL THE GIVEN DATA.....	209
FIGURE 236: QUANTITATIVE COMPARISON OF PERFORMANCE OF METHODS	210
FIGURE 237: CONVERGENCE OF ITERATIVE APPROACH, SIGNIFYING THE DEGREE OF POLYNOMIAL CONSTRUCTION	210
FIGURE 238: OPTIMAL MODEL PARAM FOR ALL THREE METHODS	210
FIGURE 239: SYSTEM CALIBRATION USING (1) AMPLITUDE ONLY, (2) PHASE ONLY	211
FIGURE 240: SYSTEM CALIBRATION IN FREQUENCY DOMAIN USING BOTH AMPLITUDE AND PHASE	212
FIGURE 241: COMPARISON OF ONLY AMPLITUDE AND ONLY PHASE WITH ITS RESPECTIVE VALUES IN MEASUREMENT FILE	213
FIGURE 242: RESIDUAL OF THREE METHODS FOR CALIBRATION IN FREQUENCY DOMAIN	214
FIGURE 243: CONTOURS PLOT OF ONLY AMPLITUDE AND ONLY PHASE	215
FIGURE 244: COMBINED COST FUNCTION CONTOURS FOR COMBINED AMPLITUDE AND PHASE.....	215
FIGURE 245: INITIAL AND OPTIMIZED PARAMETERS [M C K]	216
FIGURE 246: COST FUNCTION FOR THE THREE METHODS AT THE ITS EXTREME MIN VALUE	216
FIGURE 247: L1 RESIDUAL OF METHODS.....	216
FIGURE 248: ORIGINAL INPUT SIGNAL OF X WITH NOISE OF 2%.....	217
FIGURE 249: ORIGINAL INPUT SIGNAL OF Y WITH NOISE OF 2%.....	218
FIGURE 250: INPUT SIGNAL-SIDED AMPLITUDE SPECTRUM OF X.....	218
FIGURE 251: INPUT SIGNAL-SIDED AMPLITUDE SPECTRUM OF Y.....	219
FIGURE 252: INPUT SIGNAL-SIDED PHASE SPECTRUM OF X.....	220
FIGURE 253: INPUT SIGNAL-SIDED PHASE SPECTRUM OF Y.....	220

FIGURE 254: MAGNITUDE OF FFT FOR X COORDINATE AND PHASE OF FFT FOR X COORDINATE.....	221
FIGURE 255: MAGNITUDE OF FFT FOR Y COORDINATE AND PHASE OF FFT FOR Y COORDINATE	221
FIGURE 256: MAGNITUDE OF FFT FOR X AND Y COORDINATES AND PHASE OF FFT FOR X AND Y COORDINATES	222
FIGURE 257: POWER SPECTRAL DENSITY OF X.....	223
FIGURE 258: POWER SPECTRAL DENSITY OF Y.....	223
FIGURE 259: CROSS SPECTRAL DENSITY OF X AND Y	224
FIGURE 260: INPUT-OUTPUT CROSS-CORRELATION	225
FIGURE 261: MAGNITUDE-SQUARED COHERENCE OF CONCATENATED SIGNALS.....	226
FIGURE 262: CROSS-CORRELATION OF CONCATENATED SIGNALS (X AND Y)	226
FIGURE 263: CROSS-CORRELATION FOR DIFFERENT NOISE LEVEL	227
FIGURE 264: MAGNITUDE-SQUARED COHERENCE	228
FIGURE 265: MAGNITUDE-SQUARED COHERENCE OF CONCATENATED SIGNALS.....	229
FIGURE 266: CROSS-CORRELATION OF CONCATENATED SIGNALS (X AND Y)	229
FIGURE 267: CONCATENATED SIGNALS, CWT MAGNITUDE OF CONCATENATED SIGNAL (X) AND CWT MAGNITUDE OF CONCATENATED SIGNAL (Y).....	230
FIGURE 268: COHERENCE WITH NOISE LEVELS.....	231
FIGURE 269: CONTINUOUS FFT OF SUMMED SIGNALS ($x_1 + x_2$)	233
FIGURE 270: ANALYZED SIGNAL [$x_1 x_2$] AND DISCRETE TRANSFORM, ABSOLUTE COEFFICIENTS.....	233
FIGURE 271: CONTINUOUS FFT OF CONCATENATED SIGNALS [$x_1 x_2$]	234
FIGURE 272: CONTINUOUS FFT OF SIGNAL x_1+x_2	234
<i>FIGURE 273: CONTINUOUS FFT OF SIGNAL x_1+x_2</i>	235
FIGURE 274: CWT MAGNITUDE OF SUMMED SIGNALS ($X_1 + X_2$) AND CWT PHASE OF SUMMED SIGNALS ($X_1 + X_2$)	235
<i>FIGURE 275: CONTINUOUS FFT OF CONCATENATED SIGNALS [$x_1 x_2$], CONTINUOUS FFT OF SIGNAL x_1+x_2 AND DISCRETE WAVELET TRANSFORM (DWT) OF CONCATENATED SIGNALS</i>	237
FIGURE 276: BASICALLY, FOR EACH POINT P ON THE CURVE, WE FIND THE ONE WITH THE MAXIMUM DISTANCE D GIVEN BY (JONAS, 2010)	239
FIGURE 277: EFFECT OF TRUNCATION ON TSVD RECONSTRUCTION	241
FIGURE 278: EFFECT OF TRUNCATION ON TSVD RECONSTRUCTION (3D).....	242
FIGURE 279: EFFECT OF ITERATIONS ON ITERATIVE RECONSTRUCTION.....	243
FIGURE 280: L-CURVE FOR TIKHONOV REGULARIZATION	244
FIGURE 281: L-CURVE FOR ITERATIVE REGULARIZATION	244
FIGURE 282: L-CURVE FOR TSVD	245

1. Experiment-1:Calibration-1 Exercise Report

1.1 Introduction

Inverse problems constitute a fundamental aspect of signal processing and scientific inquiry, dealing with the challenge of inferring unknown causes or parameters from observed effects or measurements. In various fields, including engineering, physics, and medicine, these problems arise when attempting to recover critical information about a system or process from available data. The Characteristics of Inverse Problems: Ill-behaved and also Ambiguity and Non-Uniqueness, which necessitates the need of regularization techniques to solve of perturbations and uniqueness of solution.

In the context of the calibration exercise, inverse problems find relevance in the process of determining unknown parameters or properties of a model or system. Calibration involves adjusting or determining model parameters to minimize the discrepancy between model predictions and observed data, effectively framing it as an inverse problem.

1.2 Harmonic Oscillator model

In this exercise, the objective revolves around refining the parameters of a harmonic oscillator model to achieve a better fit with measured data. This process involves solving an inverse problem by iteratively adjusting parameters to minimize the difference between model predictions and actual measurements. Here, **mean squared error** is used as objective function. The techniques such as optimization algorithms and cost function evaluation are

employed to find optimal parameter values, reflecting the essence of solving an inverse problem.

1.3 Model Overview

In our model, we consider the following differential equation:

$$\rho u'' + cu' + ku = 0$$

where:

- ($u(t)$): Represents the dependent variable, often indicating displacement, voltage, or any quantity that changes concerning time (t).
- ($u''(t)$): Denotes the second derivative of (u) with respect to time, indicating the acceleration or the rate of change of the rate of change of (u) concerning time.
- (ρ): Stands for a coefficient or parameter, commonly symbolizing mass density or a similar physical property.
- (c): Represents another coefficient, typically associated with damping or resistance within the system.
- (k): Denotes stiffness or the restoring force in the system, linked to the force aiming to restore the system towards its equilibrium position when displaced.
- (t): Represents time, serving as the independent variable in the equation. This type of equations generally defines the movement of a body with the time, in real scenarios, we consider the effect of friction in the form of damping, further it maybe a constant or forced value. This differential equation characterizes systems exhibiting harmonic behavior, combining the effects of inertia, damping, and restoring forces influencing the system's motion or behavior over time.

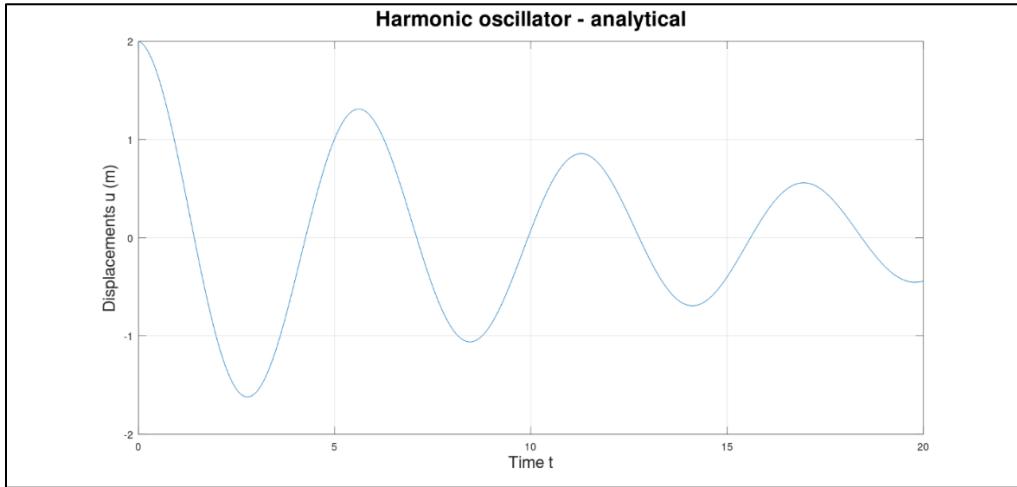


Figure 1: Time Evolution of a Harmonic Oscillator ($k = 5$, $c = 0.6$, $\rho = 4$) over 20 seconds with a Time Step of 0.05 seconds.

1.4 Parameter Variation - Observation

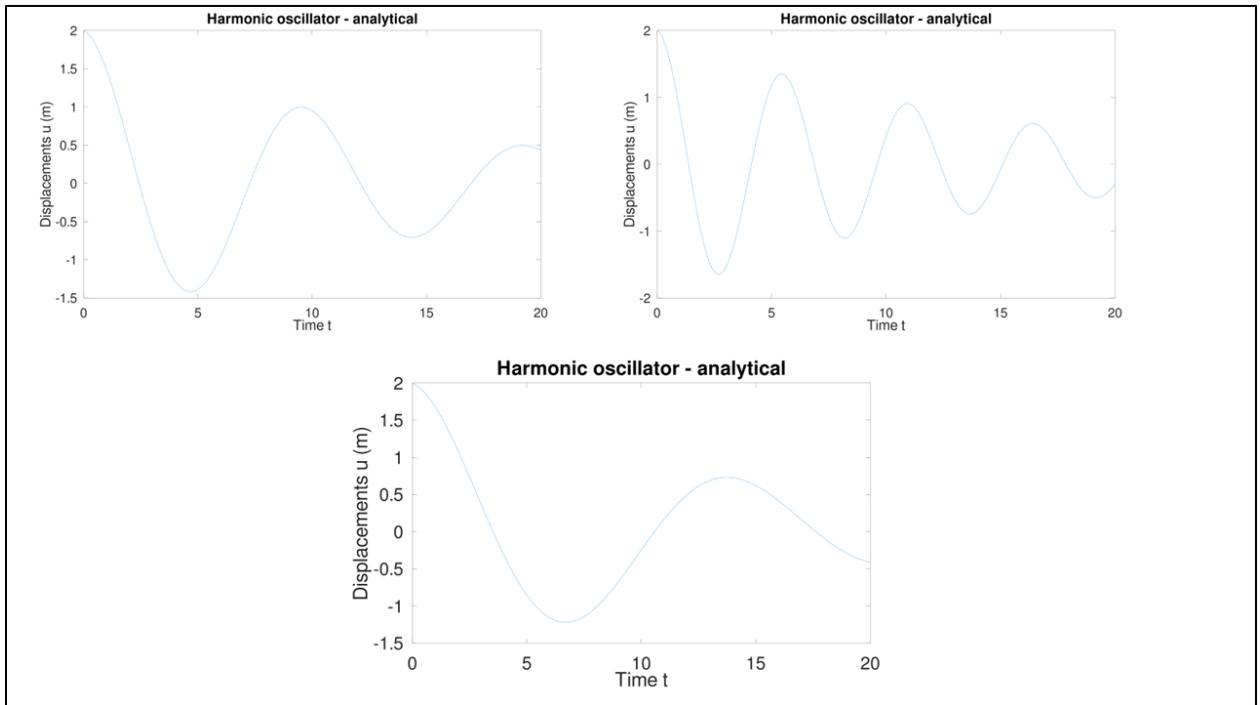


Figure 2: Plot of harmonic oscillator for different values of stiffness : Top-Left ($k=4$), Top-right ($k=6$) and Bottom ($k=2$)

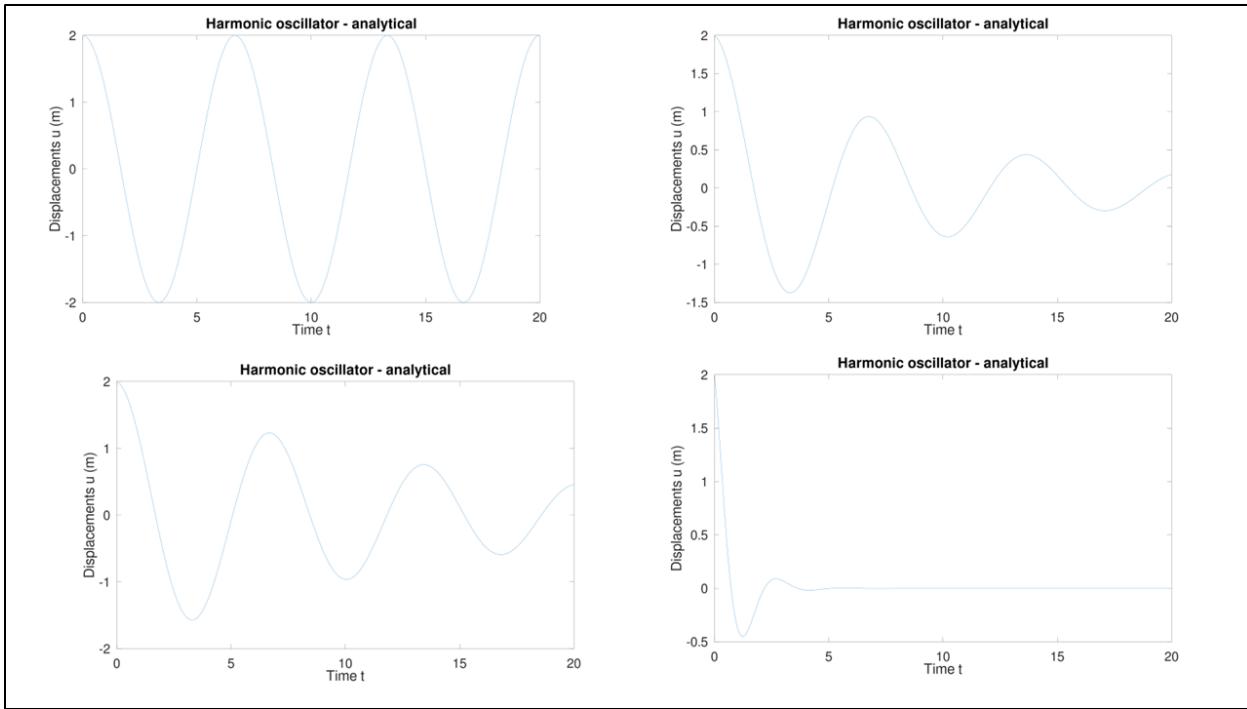


Figure 3: Plot of Harmonic oscillator for different values of damping coefficient: [4 c 4.5],[0:0.05:20] - ([k c rho/m],frequency of time of 0.05 for 20 sec] : Top-right ($c = 0.0001$), Top-left ($c = 1$), Bottom-right ($c = 0.65$), Bottom-left ($c = 10$).

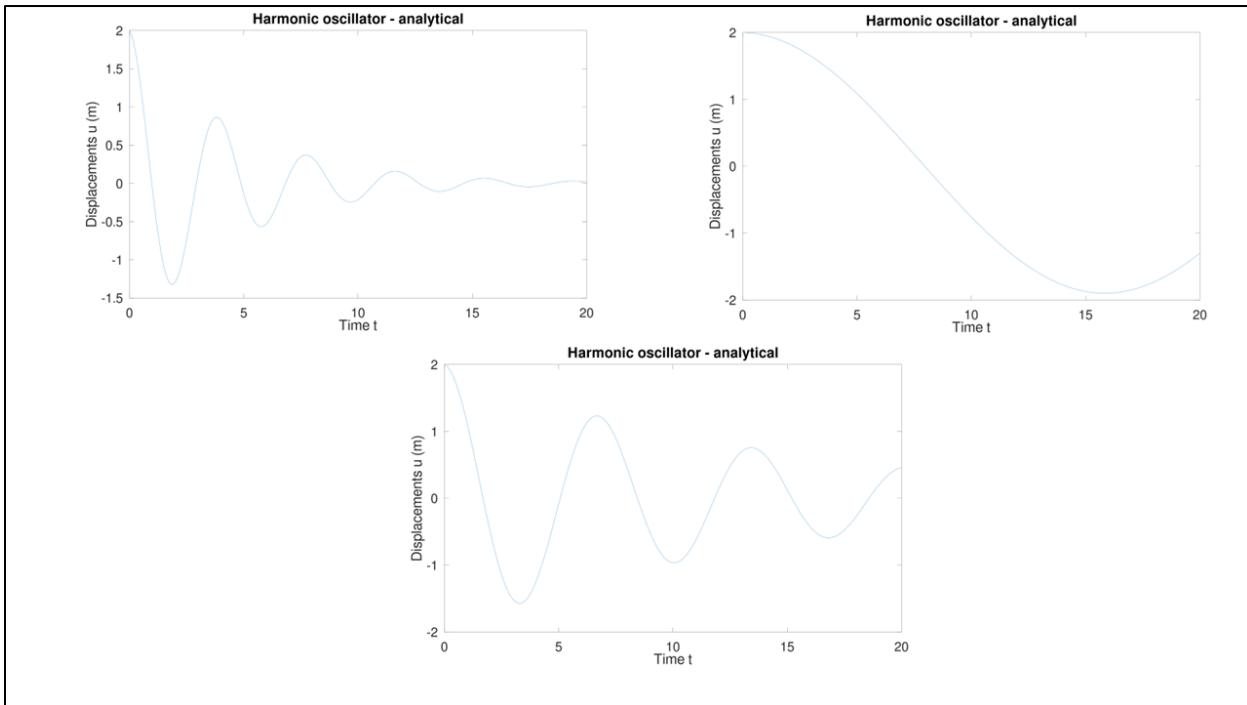


Figure 4: Plot of Harmonic oscillator for different values of mass of oscillator: [4 0.65 m],[0:0.05:20] - ([k c rho/m],time division of 0.05 for 20 sec] : Top-right ($m = 1.5$), Top-left ($m = 100$), Bottom ($m = 4.5$)

how changing each parameter might affect the behavior of the harmonic oscillator:

1.4.1 Varying Parameter k

Increasing k: It increases the stiffness or the restoring force of the system. With higher stiffness, the system's oscillations would be faster and stronger, resulting in a quicker return to equilibrium and shorter oscillation periods. In Figure 1, for high $k=6$, we have many oscillations in comparison to 4 and 2.

Decreasing k: Reducing stiffness slows down the restoring force, leading to slower and more extended oscillations. The system takes more time to return to equilibrium. In Figure 1, for low $k=6$, we have low oscillations in comparison to 4 and 6.

1.4.2 Varying Parameter c

Increasing c: In Figure 2, simulated over 20 seconds with a time step of 0.05, these plots showcase the impact of damping on oscillation behavior. Higher damping coefficients lead to quicker return to equilibrium but may dampen oscillation amplitudes."

Decreasing c: Conversely, a reduction in the damping coefficient results in more prolonged oscillations. The system, in this case, takes longer to reach equilibrium, reflecting the decreased damping effect.

1.4.3 Varying Parameter ρ/m

Increasing ρ : In Figure 3, the system with mass of This parameter is associated with mass density or a similar property. A higher value indicates a greater mass density, which results in slower oscillations. Due to increased inertia, the system takes longer to return to equilibrium.

Decreasing ρ : Lower mass density results in lighter oscillators, facilitating faster and more frequent oscillations. The system responds more quickly to perturbations.

Each parameter's effect alters the system's behavior, influencing the rate of return to equilibrium, the speed of oscillations, and the amplitude of the oscillations. These changes are crucial in understanding how the system responds to different physical conditions.

1.5 Measurement

1.5.1 Measurement overview

1.5.1.1 *Files and Data Structure*

The measurements consist of six files **in.mat** format, specifically capturing the displacement values u of a system.

u_meas0.mat: This file contains displacement values u without any added noise, representing the undisturbed or baseline measurements.

u_meas01.mat: Recorded displacement values u with a 1% noise component added to simulate a minor level of variability or measurement error.

u_meas03.mat: Contains displacement data u with a 3% noise level, introducing slightly more variability compared to the 1% noise scenario.

u_meas05.mat: Represents displacement values u with a 5% noise factor, introducing a moderate level of deviation from the ideal measurements.

u_meas09.mat: The fifth file contains displacement data u affected by a 9% noise level, simulating a higher level of measurement error or variability.

Additionally, there is a sixth file:

amplitude.mat: This file contains scalar values representing the amplitude of the measured oscillations.

1.5.2 Noise Levels and Significance - Observation

The various .mat files represent different scenarios of measured displacement u , each with progressively increasing levels of added noise. These levels of noise simulate different measurement conditions or instrument inaccuracies:

Higher Noise: As the noise percentage increases from 1% to 9%, the measurements progressively deviate more from the noise-free data in **u_meas0.mat**.

Impact on Accuracy: Each level of added noise introduces a varying degree of inaccuracy or deviation from the ideal measurements, reflecting real-world scenarios where measurements may be affected by environmental factors or instrumentation limitations.

Amplitude Data: The *amplitude.mat* file contains scalar values representing the amplitude of the measured oscillations. This scalar parameterizes the system's behavior, indicating the peak magnitude or extent of the oscillations captured in the displacement data.

1.6 Comparison plot

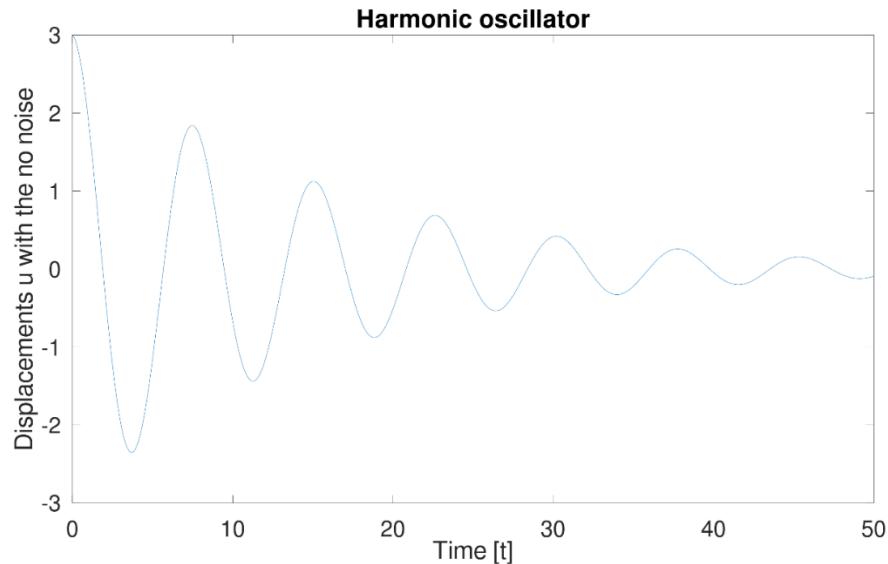


Figure 5: Plot of harmonic oscillation of a body for the data file of *umeas0.mat* (no noise)

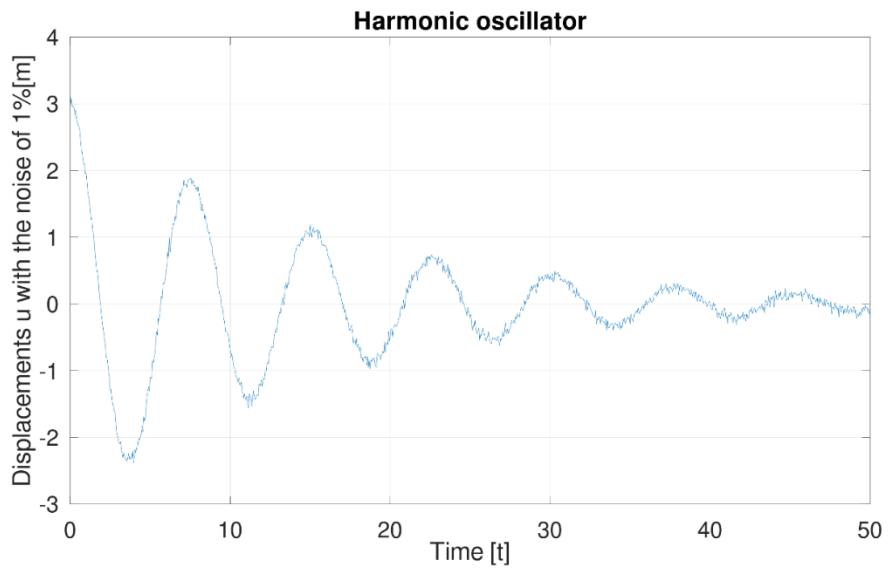


Figure 6: Plot of harmonic oscillation of a body for the data file of umeas01.mat

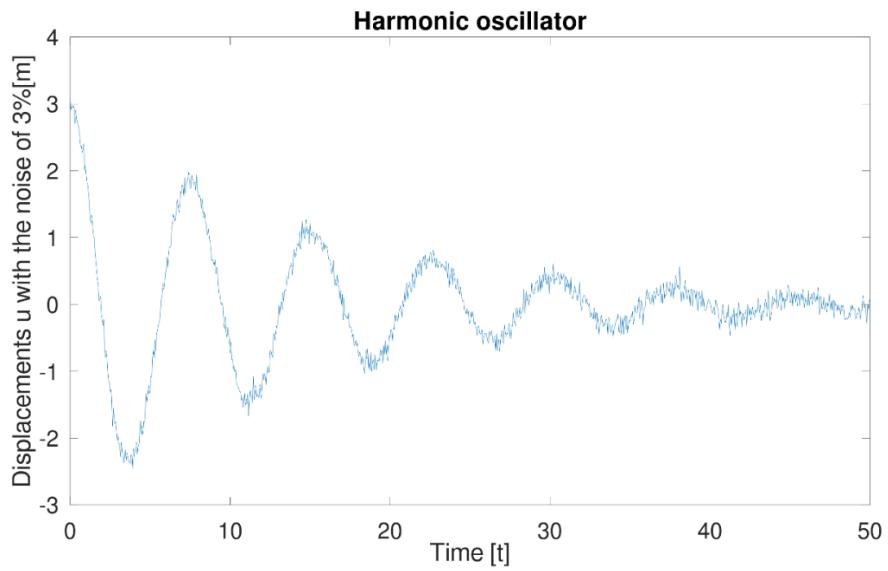


Figure 7: Plot of harmonic oscillation of a body for the data file of umeas03.mat

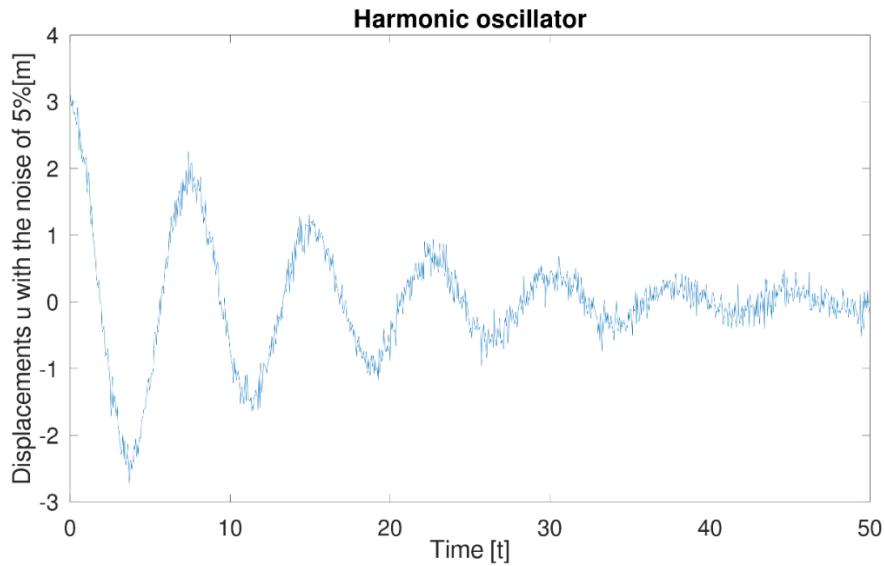


Figure 8: Plot of harmonic oscillation of a body for the data file of umeas05.mat

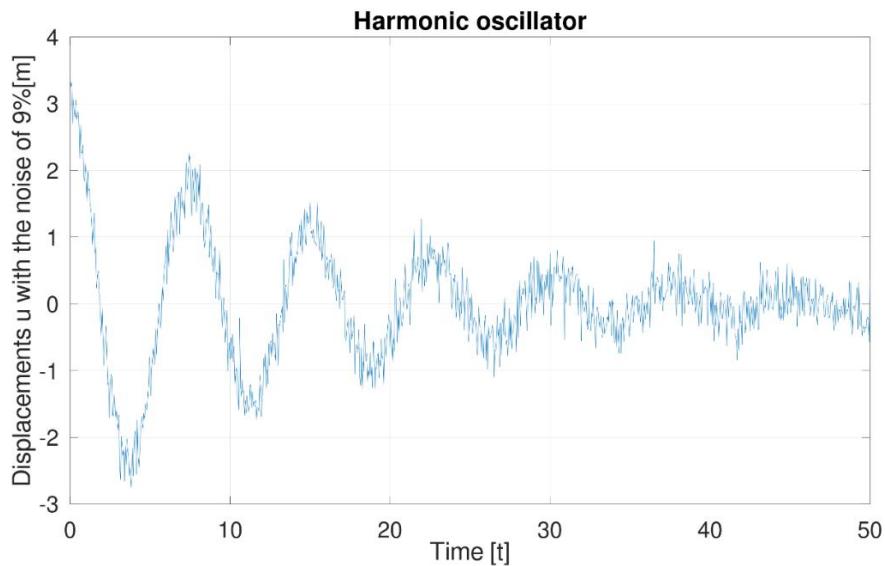


Figure 9: Plot of harmonic oscillation of a body for the data file of umeas09.mat

1.6.1 Description of Plots

The series of plots depict the variation of u over time for different levels of noise introduced into the measurements. As the noise level increases from 0% to 9%, there's a noticeable change in the patterns and fluctuations observed in u across time.

Observations

- **Noise-Free (0%):** The plot without any added noise exhibits a smooth and consistent behavior, reflecting the system's response without external disturbances. The oscillations follow an expected pattern and maintain a stable course over time.
- **Low Noise (1% and 3%):** As the noise level starts to increase slightly, there's a marginal impact on the smoothness of the plotted curve. However, the deviations from the noise-free plot remain relatively small, with minor fluctuations introduced in the oscillations.
- **Moderate Noise (5%):** At this noise level, observable disruptions begin to emerge in the plotted curve. The oscillations exhibit more erratic behavior, with noticeable deviations from the expected pattern. Despite this, the overall shape of the curve remains distinguishable.
- **High Noise (9%):** The plot with the highest noise level displays significant disturbances and erratic fluctuations in u over time. The curve is notably jagged, and the original pattern of oscillations becomes harder to discern due to the increased interference from noise.

Reasoning

The increase in disturbance observed with higher noise levels is a direct consequence of the introduced noise interfering with the accurate representation of u over time. As the noise percentage rises, the measured data becomes increasingly corrupted, leading to deviations from the true behavior of the system. This interference becomes more pronounced with higher noise levels, impacting the fidelity of the measured data and complicating the process of accurately modeling the system.

Conclusion

The plots effectively illustrate how varying levels of noise impact the fidelity and reliability of measurements. Higher noise levels lead to greater uncertainties and deviations from the expected behavior, highlighting the importance of noise mitigation techniques and the challenges involved in handling noisy data for accurate modeling and analysis.

1.7 Comparison between numerical and measurements value

In this section, the comparison of each `u_meas` file has been made with the data plot of numerical results obtained from `MyHarmonic_Num.m` using values of $k = 45$, $c = 6.5$, and $m = 40$. These `u_meas` files contains different levels of noise, the results are obtained as follows:

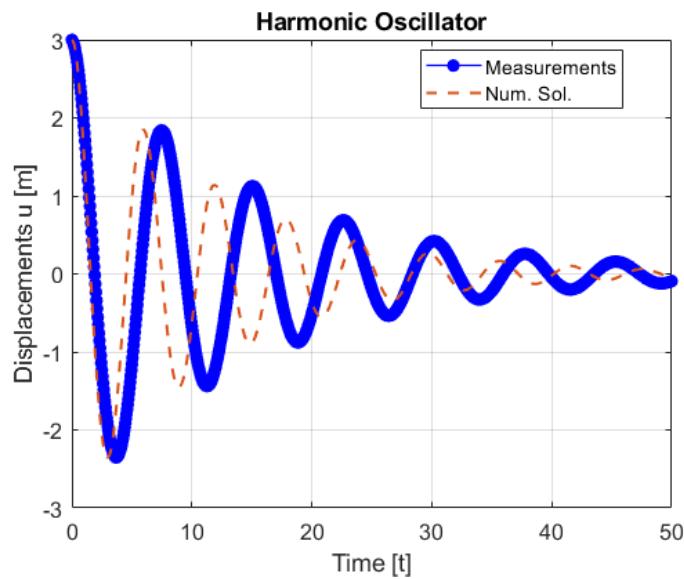


Figure 10: Plot for comparison between `umeas0` and numerical solution using values of k, c, m .

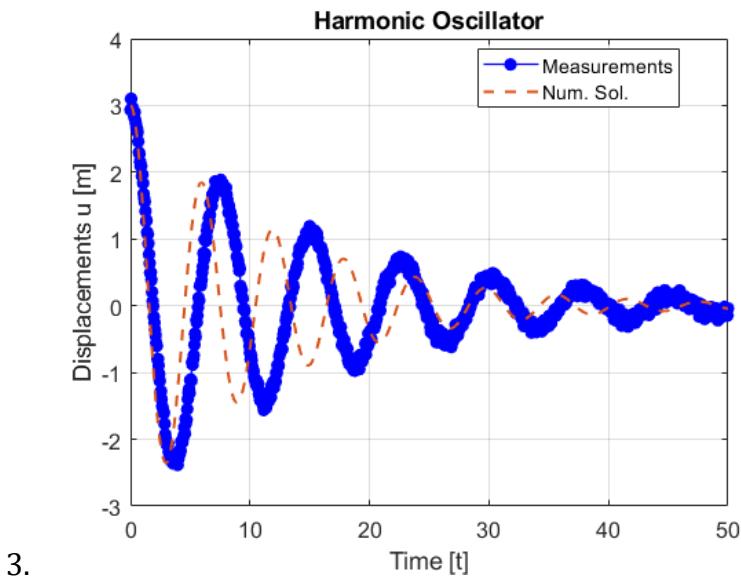


Figure 11: Plot for comparison between `umeas01` and numerical solution using values of k, c, m .

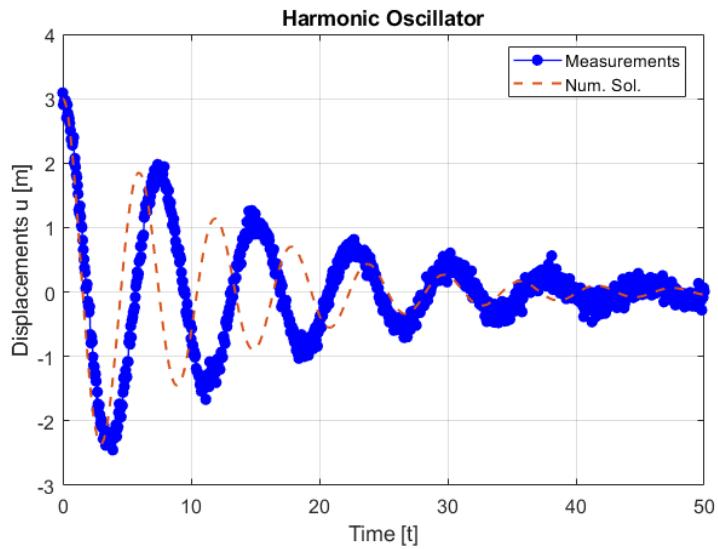


Figure 12: Plot for comparison between umeas03 and numerical solution using values of k, c, m .

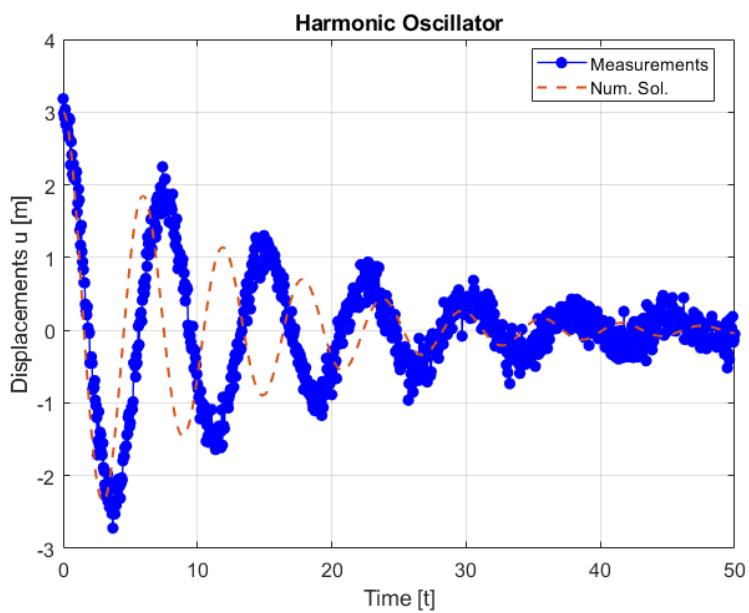


Figure 13: Plot for comparison between umeas05 and numerical solution using values of k, c, m .

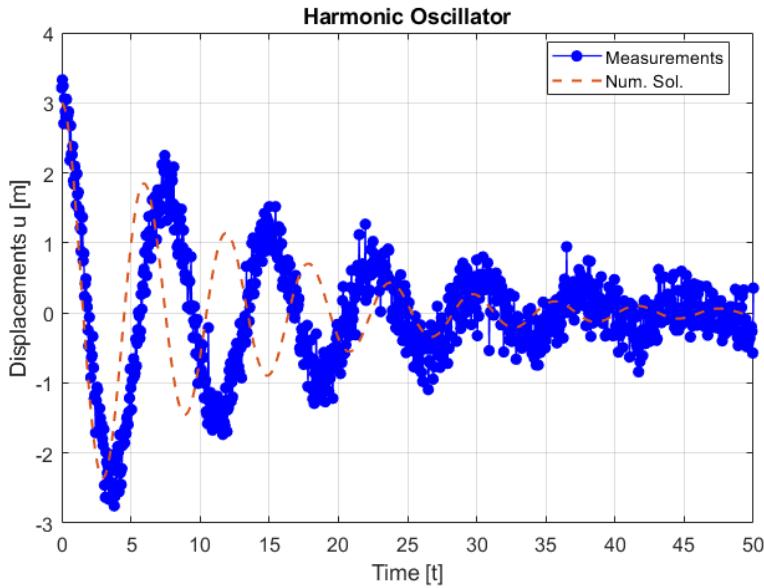


Figure 14: Plot for comparison between `umeas09` and numerical solution using values of k , c , m .

1.8 Cost Function Understanding

This function, `myCostFunctionHarmonicFit_Num`, serves as a critical component in evaluating the cost function for the calibration process. It measures the squared differences between the model predictions and the observed data, represented as `u_meas`, based on the given input parameters.

1.8.1 Computation of the Cost Function

The primary objective of this function is to compute the cost functional $f(x)$, which quantifies the discrepancy between the predicted values of the harmonic oscillator model (\mathbf{u}) and the actual measurements (`u_meas`). The calculation of the cost functional follows the expression:

$$f(x) = \sum (u_i - u_{meas})^2$$

Here's a of how `myCostFunctionHarmonicFit_Num` function operates:

- **Parameter Checks:** The function checks for negative values in the input parameters. If any negative values are provided, they are overwritten and set to specific default values (30 for the first coefficient, 4 for the second coefficient, and 30 for the third coefficient).
- **Model Prediction:** It computes the model's response (**u**) using the **Myharmonic_Num** function based on the given input parameters **x**, time span **tspan**, initial condition **y_0**, and plotting configuration **plot_fig**.
- **Cost Evaluation:** The cost function **fx** is evaluated by summing the squared differences between the model predictions (**u**) and the actual measurements (**u_meas**).

Handling of Output Arguments:

This function accommodates different scenarios based on the number of output arguments requested:

- Two Output Arguments (**fx**, **x**): Computes and returns the value of the cost function **fx**.
- Three Output Arguments: Additionally calculates the cost function **fx** and returns the numerical approximation of the gradient (though the code doesn't explicitly compute the gradient).

1.8.2 Impact of Noise on Cost Evaluation

The function does not explicitly handle noisy measurements. However, when noisy data is fed into **u_meas**, the cost function evaluation becomes more sensitive to discrepancies between the model's predictions and the noisy measurements. Increased noise levels typically lead to higher values of the cost function, signifying larger discrepancies between the model and the noisy data. This effect could influence the optimization process, making it more challenging to converge to an optimal solution due to the added uncertainty in the measurements.

1.9 Optimization Analysis

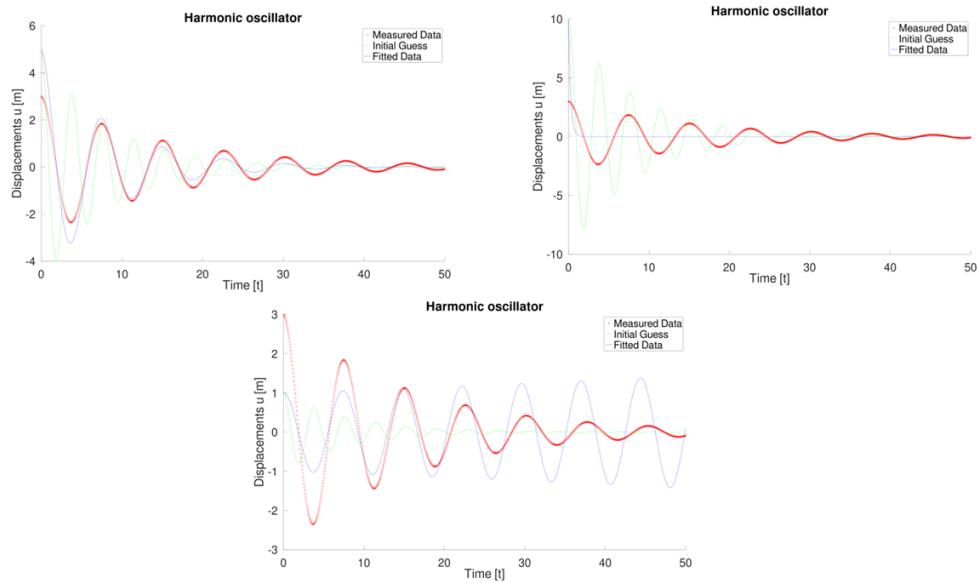


Figure 15: Varying initial amplitude; left-top with initial amplitude of 1, right-top with initial amplitude of 10, center-bottom with initial amplitude of 5.

1.9.1 Effect of Varying Initial Amplitude:

Measured Data:

Remains Unchanged: The measured data doesn't depend on the initial amplitude set for the optimization. It represents the actual observed values, unaffected by the optimization process.

Initial Guess Plot:

Diverse Initial Estimates: Changing the initial amplitude affects the initial guess's starting position for the optimization process. Higher initial amplitudes may potentially lead to initial guesses that are farther from the optimal solution, influencing the path the optimization algorithm takes.

Impact on Convergence: With larger initial amplitudes, the initial guess plots might show more significant deviations from the measured data initially. This can affect the optimization process, potentially influencing convergence speed or even the final solution.

Fitted Data:

Convergence and Accuracy: Higher initial amplitudes might affect the fitting process by altering the optimization landscape. A larger initial amplitude could cause the optimization algorithm to converge differently, potentially affecting the accuracy of the fitted data.

Potential for Divergence: Depending on the algorithm and the problem's nature, a significantly higher initial amplitude might lead to divergence in convergence, making it harder for the algorithm to find an optimal solution.

Overall, while varying the initial amplitude primarily influences the starting point and initial trajectory of the optimization process, it might indirectly affect the convergence, accuracy, and the final fit achieved by the optimization algorithm. The impact varies based on the specific optimization method and the nature of the problem being optimized.

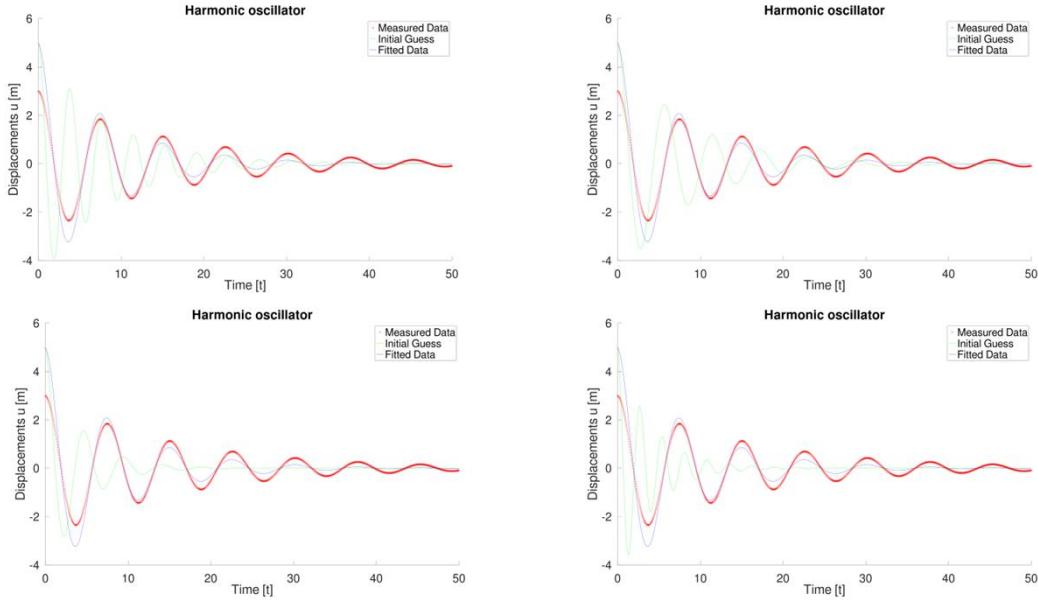


Figure 16: Plot of Harmonic oscillation displacement w.r.t to time for different initial values; Top_right, $x = [33.894, 18.918, 80.637]$, $x0 = [60, 5, 20]$, Top_left, $x = [26.0247, 5.8983, 25.1416]$, $x0 = [30, 5, 20]$, Bottom_left, $x = [32.561, 18.712, 79.759]$, $x0 = [60, 10, 20]$, Bottom_right, $x = [39.042, 19.684, 83.906]$, $x0 = [60, 5, 10]$

1.9.2 Figure 11: Plot of Harmonic Oscillation Displacement with Respect to Time

The figure presents the displacement of a harmonic oscillator over time for various initial parameter values and their respective optimized values. Each subplot represents a different initial guess ($\mathbf{x0}$) for the parameters, showcasing the resulting optimized values (\mathbf{x}).

Top-Right Subplot

- **Initial Guess ($\mathbf{x0}$):** $[60, 5, 20]$
- **Optimized Values (\mathbf{x}):** $[33.894, 18.918, 80.637]$
- **Reasoning:** The optimization process refines the initial guess to obtain values that minimize the discrepancy between the model's predictions and measured data. The resulting optimized values demonstrate the adjustment from the initial guess to fit the data better.

Top-Left Subplot

- **Initial Guess (x_0):** [30, 5, 20]
- **Optimized Values (x):** [26.0247, 5.8983, 25.1416]
- **Reasoning:** Altering the initial guess affects the optimization process, leading to a different set of optimized parameters. In this case, a lower initial stiffness (k) results in adjusted values to better fit the observed data.

Bottom-Left Subplot

- **Initial Guess (x_0):** [60, 10, 20]
- **Optimized Values (x):** [32.561, 18.712, 79.759]
- **Reasoning:** Changing another parameter, such as damping (c), while keeping the other initial values constant leads to different optimized parameters. Here, a higher damping coefficient results in adjusted values that optimize the model fit.

Bottom-Right Subplot

- **Initial Guess (x_0):** [60, 5, 10]
- **Optimized Values (x):** [39.042, 19.684, 83.906]
- **Reasoning:** Adjusting another parameter, such as the initial mass (ρ), showcases its impact on the optimization process. A lower mass density here affects the optimized parameters to better match the observed data.

The variations in initial guesses lead to different optimized parameter sets, highlighting how changes in these values impact the optimization outcome. The optimized parameters represent the adjusted values that minimize the difference between the model predictions and actual measurements, emphasizing the sensitivity of the optimization process to initial guesses and parameter variations. This description emphasizes the impact of initial guesses on the optimization process and how different parameters affect the resulting optimized values. Adjusting these initial values alters the convergence of the optimization algorithm, influencing the final optimized parameters to better fit the measured data.

1.9.3 fminsearch Optimisation

1.9.3.1 for 3 variables

fminsearch : It's a Nelder-Mead simplex algorithm is an iterative optimization algorithm employed for minimizing an objective function without requiring derivatives. The algorithm operates by maintaining a simplex—a geometrical shape with varying vertices or points in the parameter space. At each iteration, the algorithm evaluates the objective function at the simplex's vertices and then performs reflection, expansion, contraction, or shrinkage operations to adjust the simplex. The mathematics behind the algorithm involve updating the vertices of the simplex based on the function evaluations. These updates aim to converge the simplex towards the minimum of the objective function. The algorithm iteratively refines the simplex until a convergence criterion is met or a specified number of iterations are reached. One key advantage of the Nelder-Mead algorithm is its simplicity and ease of implementation, making it suitable for optimizing functions that are non-smooth, noisy, or lack analytical derivatives. However, it may converge slowly in some cases and is sensitive to the choice of initial parameters.

This function is utilised to optimise for the measured file umea03.mat, the initial guess we took is $x_0 = [30, 12, 19]$, the optimised values are [21.099,3.91,30.06].

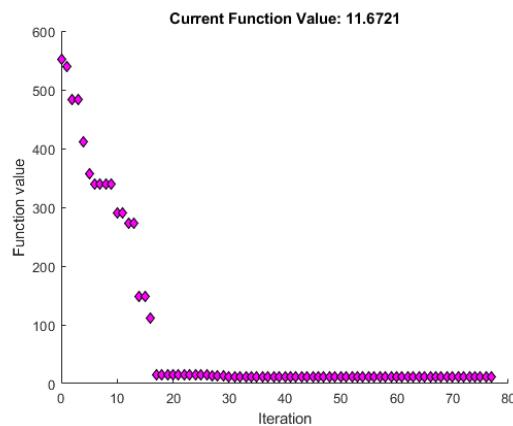


Figure 17: Plot of optimization showing how current function value (RMS) is optimized for each iteration within a threshold

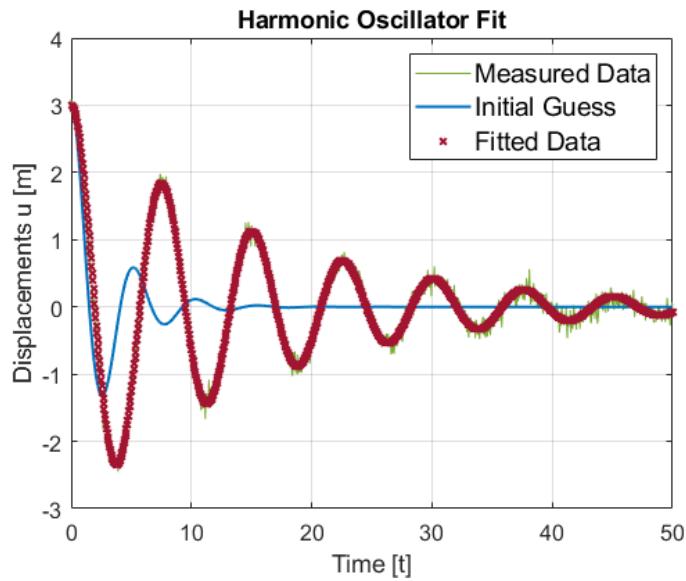


Figure 18: Plot of harmonic oscillator for measured plot (umea03), initial guess and fitted data (value of harmonic variables after optimisation).

1.9.3.2 for 2 variables

fminsearch function is utilised to optimise for the measured file umea03.mat, the initial guess we took is $x_0 = [30, 12]$, the optimised values are $[28.07, 5.20]$. In our observation we found the current function value comes same for 3 and 2 variable optimisation, which can be explained by threshold which is in the present case is 10^{**-4} .

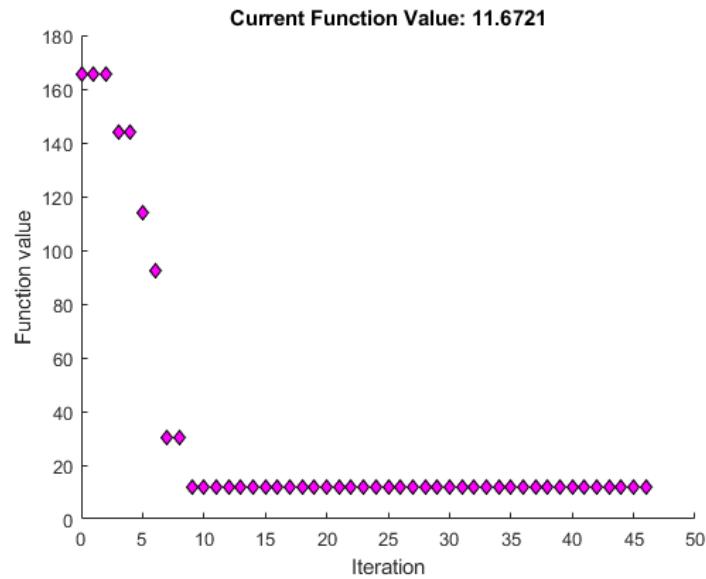


Figure 19: Plot of optimization showing how current function value (RMS) is optimized for each iteration within a threshold

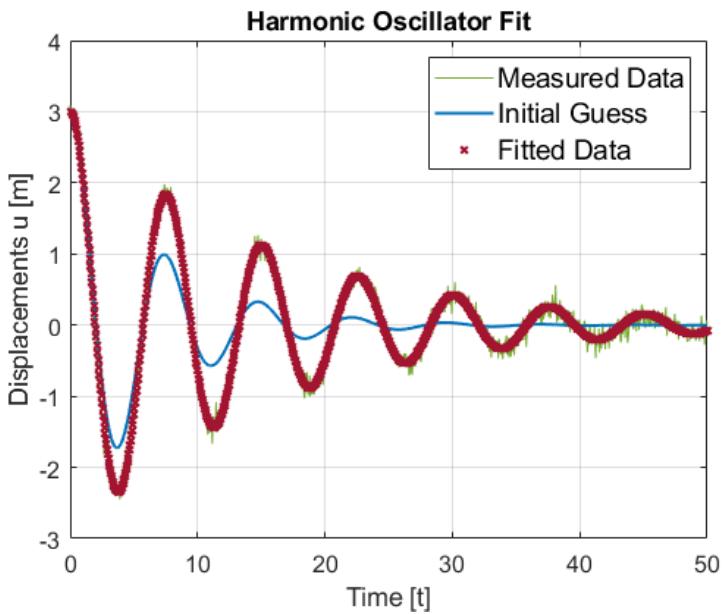


Figure 20: Plot of harmonic oscillator for measured plot (umeas03), initial guess and fitted data (value of harmonic variables after optimization).

1.9.4 fmincon Optimisation:

The fmincon function in MATLAB is designed for constrained nonlinear optimization problems. In mathematical terms, fmincon seeks to find the values of decision variables that minimize (or maximize) an objective function, subject to a set of equality and inequality

constraints. The general form of a constrained optimization problem that fmincon aims to solve can be expressed as follows:

Minimize (or Maximize) $f(x)$
subject to:
 $c(x) \leq 0$ (inequality constraints) and $ceq(x) = 0$ (equality)

Here, " $f(x)$ " is the objective function that depends on the decision variables "x" that we are seeking to optimize. The optimization is carried out while adhering to a set of constraints. The inequality constraints " $c(x) \leq 0$ " restrict the feasible region of solutions, ensuring that certain conditions are satisfied. The equality constraints " $ceq(x) = 0$ " impose additional restrictions on the feasible solutions. The fmincon algorithm uses an interior-point method to handle both equality and inequality constraints. It iteratively refines the solution by adjusting the decision variables to minimize the objective function while satisfying the constraints. The optimization process involves calculating gradients and Hessians of the objective and constraint functions to guide the search for the optimal solution.

1.9.4.1 for 3 variables

the initial guess we took is $x_0 = [30, 12, 19]$, the optimised values are $[27.65, 5.12, 39.39]$, the tolerance is adjusted to $1e-10$ and max iterations is 10000.

```
% Using constrained nonlinear multivariable function
options = optimset('TolX', 1e-10, 'Display', 'iter', 'PlotFcns', @optimplotfval,
'Algorithm', 'sqp', 'MaxIter', 10000);
[x, fval] = fmincon(@(x) myCostFunctionHarmonicFit_Num(x, tspan, y_0, plot_fig,
u_meas), x0, [], [], [], [10 1 10], [100 10 100], [], options);
```

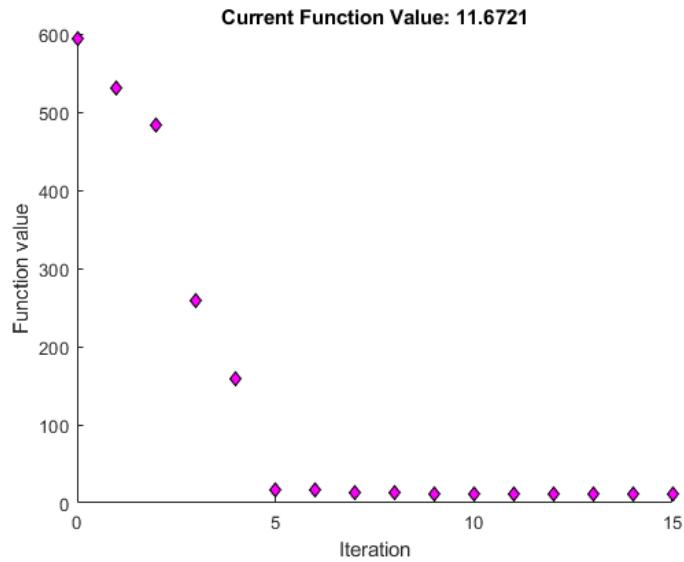


Figure 21: Plot of optimization showing how current function value (RMS) is optimized for each iteration within a threshold

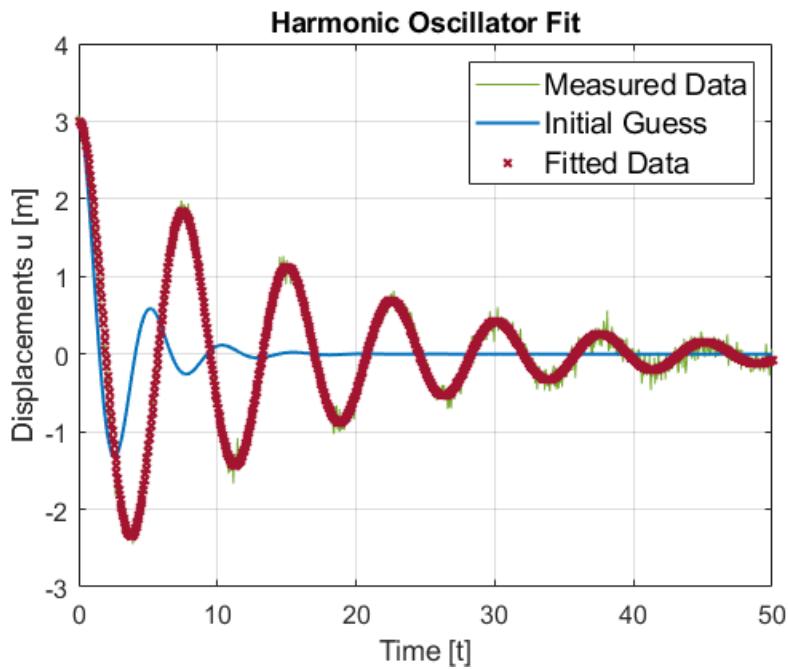


Figure 22: Plot of harmonic oscillator for measured plot (umeas03), initial guess and fitted data (value of harmonic variables after optimization).

1.9.4.1.1 Case 2: different tolerance and max iteration

the initial guess we took is $x_0 = [30, 12, 19]$, the optimised values are $[27.65, 5.12, 39.39]$, the tolerance is adjusted to $1e-16$ and max iterations is 10

5000. Not much change is observed (till second decimal place) due to changing tolerance and iterations, maybe this params is not of much effect. For further evalutaion we can change initial values, measurement files and different optimisation params of fmincon options param. We limit our search till here, due to limited theoretical understanding.

```
options = optimset('TolX', 1e-16, 'Display', 'iter', 'PlotFcns', @optimplotfval,
'Algorithm', 'sqp', 'MaxIter', 5000);
[x, fval] = fmincon(@(x) myCostFunctionHarmonicFit_Num(x, tspan, y_0, plot_fig,
u_meas), x0, [], [], [], [10 1 10], [100 10 100], [], options);
```

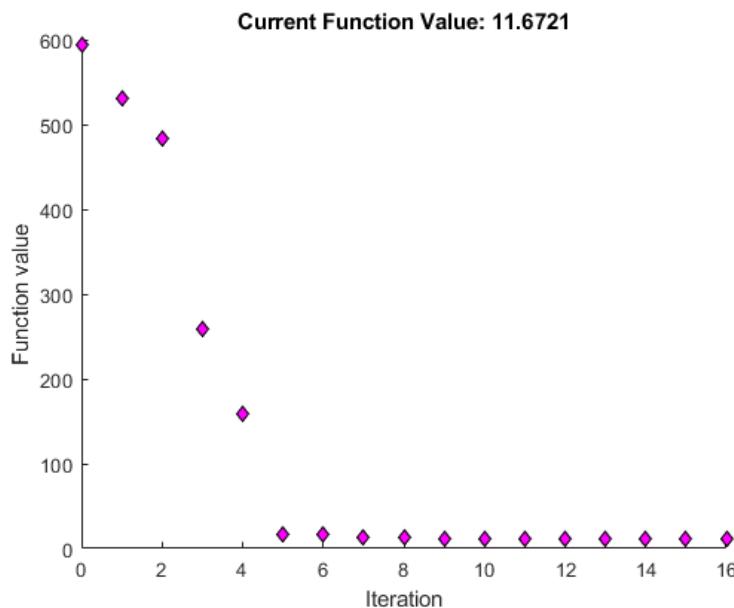


Figure 23: Plot of optimization showing how current function value (RMS) is optimized for each iteration within a threshold

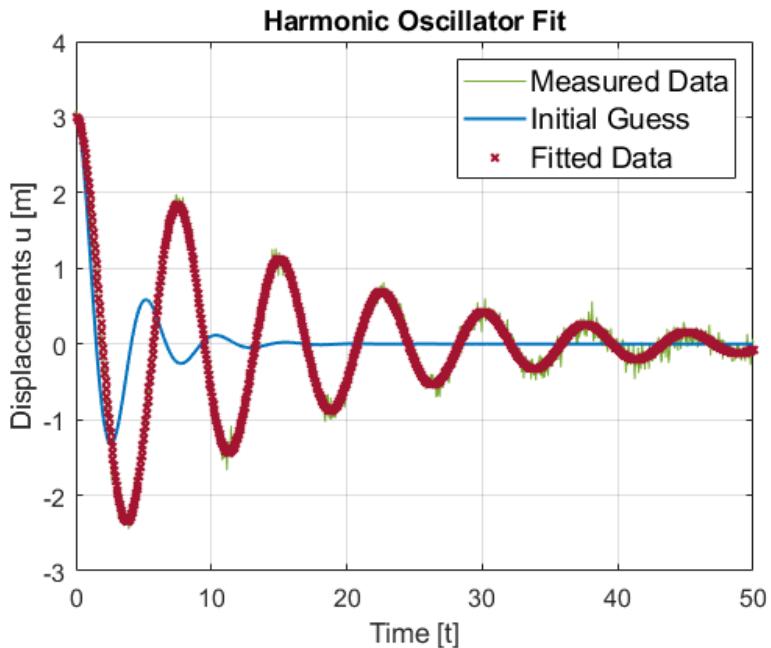


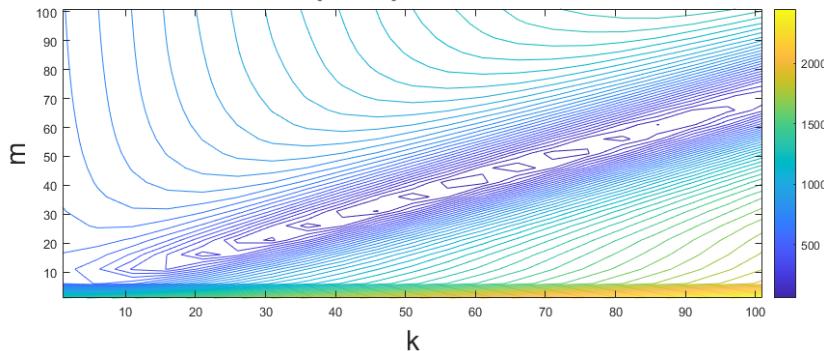
Figure 24: Plot of harmonic oscillator for measured plot (umeas03), initial guess and fitted data (value of harmonic variables after optimization).

Observation: Based on the surface optimisation study, fmincon converges faster in comparison to fminsearch. However, more detail study would be required for inter-comparison of various algorithm.

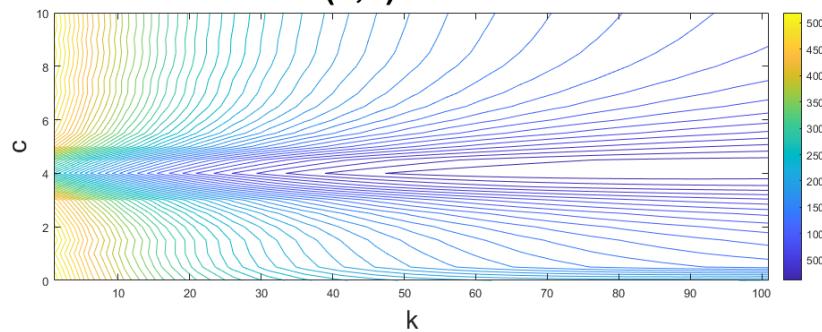
1.10 Counterline for file u_meas01.mat

Contour lines serve as a visual tool for depicting the elevation or depth of a 3-dimensional surface in a 2-dimensional representation. They are created by connecting points with constant z-coordinate values, effectively outlining the contours of the surface. These lines help convey the intricate topography or variations in height across the xy-plane. Essentially, contour lines offer a method for intuitively understanding the shape and structure of a three-dimensional landscape by displaying the locations where the surface possesses the same z-coordinate values.

Contourlines of $J(k,m)$ for harmonic oscillator



Contourlines of $J(k,c)$ for harmonic oscillator



Contourlines of $J(c,m)$ for harmonic oscillator

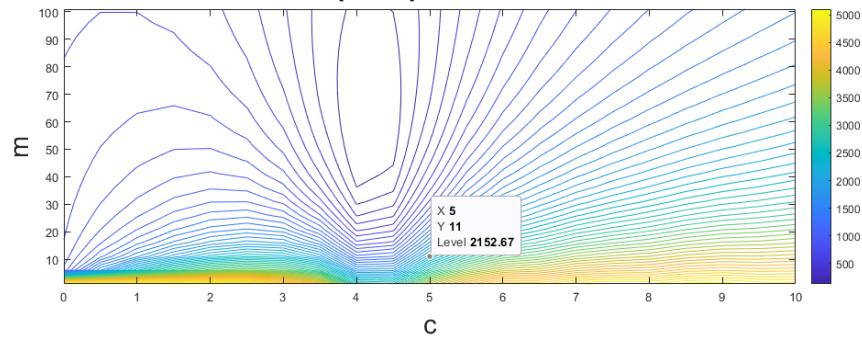


Figure 25: Contour plots showing the values of k, c, m when two variables are optimised while keeping other one as constant, measurement data of umeas04 taken into consideration.

For the succinct of the study we plotted for one for just one measurement file.

1.10.1 Observation:

The contour plots generated from various measurements files provide compelling evidence of a direct and linear relationship between the stiffness and mass parameters within the dynamic system under consideration. This observation underscores a fundamental

connection between these two critical characteristics, suggesting that changes in stiffness correspond precisely to proportional changes in mass, and vice versa. This correlation could stem from the inherent physical properties of the materials involved or the structural configuration of the system.

2. Experiment – 2: Calibration Exercise Report: Exploring Equation of Motion

2.1 Introduction

2.1.1 Objective

This experiment aims to investigate the practical application of calibration methodologies within the context of signal analysis and system identification, focusing on their implementation in the Equation of Motion—a fundamental concept in various scientific and engineering domains. The primary goal is to scrutinize these methodologies, particularly emphasizing Tikhonov regularization, to enhance precision and reliability in modeling dynamic systems.

2.1.2 Scope

This study focuses on a systematic examination of calibration techniques, with a specific emphasis on integrating Tikhonov regularization into the calibration process. The main scope includes:

Theoretical Exploration: A detailed analysis of calibration methodologies, including Least Squares and Tikhonov regularization, to comprehend their theoretical foundations and practical applicability.

Applied Optimization: Investigating the practical integration of Tikhonov regularization to refine parameter selection within the Equation of Motion framework, aiming to improve model accuracy and suitability for real-world scenarios.

Practical Implications: Assessing the practical implications of refined calibration techniques in system identification, addressing complexities encountered in real-world applications.

This report is an attempt to provide insights into the optimization of calibration methodologies, specifically their impact on model accuracy and adaptability in dynamic system modeling.

2.2 Review of Previous Calibration Using Least Squares

2.2.1 Methodology Recap

The prior calibration exercise utilized the Least Squares method as an optimization technique to minimize the squared differences between the model predictions and the observed data. This approach involved formulating the calibration problem as an optimization task to find the model parameters that best fit the measured data.

Mathematical Framework:

The Least Squares method operates by formulating an objective function that quantifies the sum of squared differences between the model predictions and the actual measurements.

The optimization process aims to minimize this objective function by adjusting the model parameters (coefficients ρ , c , k) iteratively until a satisfactory fit is achieved.

Role of Least Squares in Optimization:

Least Squares is a most commonly used optimization technique for parameter estimation and curve fitting. It's particularly effective when dealing with overdetermined systems or problems with more equations than unknowns. By minimizing the sum of squared residuals, it seeks to find parameter values that best approximate the observed data.

Outcomes Derived:

The Least Squares approach generated optimized parameter values (ρ , c , k) that minimized the discrepancy between the model predictions and the actual measurements. These optimized parameters represented the best fit obtained through the minimization of the objective function.

2.2.2 Notable Insights

2.2.2.a Significant Observations:

Robustness: The Least Squares method demonstrated robustness in handling noisy data by minimizing the impact of outliers through the squared differences.

Convergence: It exhibited good convergence properties, often reaching solutions efficiently and reliably within the defined optimization constraints.

Challenges Encountered:

Sensitivity to Initial Guesses: The method's performance could be sensitive to the choice of initial parameter values, potentially affecting convergence speed and final solutions.

Overfitting: In certain cases, the Least Squares approach might overfit the model to the noise present in the data, leading to a less generalized fit.

Role of Regularization Techniques:

Regularization techniques could play a crucial role in mitigating overfitting issues encountered with Least Squares. Methods like Ridge Regression or LASSO could help impose constraints on parameter values, enhancing model generalization by balancing the fit with the complexity of the model.

Overall Assessment:

The Least Squares method, while effective in minimizing the squared differences between model predictions and observed data, might require careful consideration of initial guesses and potential overfitting. Regularization techniques could serve as enhancements to improve the model's generalization and robustness.

The role of Least Squares in optimization is significant due to its simplicity, effectiveness in handling overdetermined systems, and its prevalence in various fields requiring parameter estimation and curve fitting. However, its application might necessitate considerations of noise levels, initial conditions, and potential overfitting, prompting the exploration of complementary techniques like regularization for improved performance.

2.3 Regularization Techniques - Tikhonov

2.3.1 Theoretical Underpinnings of Tikhonov Regularization

Tikhonov regularization, also known as Ridge regression in certain contexts, is a fundamental regularization technique employed in solving ill-posed inverse problems. It's a method aimed at improving the stability and generalization of solutions, especially when dealing with noisy or limited data. Here's an exploration of its theoretical underpinnings:

Theoretical Basis:

Problem Context:

Tikhonov regularization addresses problems where the optimization task involves fitting a model to observed data but lacks sufficient information or has an unstable solution due to ill-conditioning, noise, or overfitting.

Mathematical Framework:

The fundamental principle of Tikhonov regularization involves adding a regularization term to the objective function being minimized. It introduces a penalty for complex or extreme parameter values, seeking a balance between fitting the data and keeping the parameters small:

Objective Function:

The regularized objective function can be represented as:

$$J(\mathbf{w}) = \|\mathbf{Aw} - \mathbf{b}\|^2 + \alpha \|\mathbf{w}\|^2$$

- $J(\mathbf{w})$: Represents the total error or cost function to be minimized.
- \mathbf{Aw} : Model predictions.
- \mathbf{b} : Observed data.
- $\|\mathbf{w}\|^2$: Norm of the parameter vector \mathbf{w} representing the model parameters.
- α : Regularization parameter controlling the trade-off between data fitting and regularization. A higher α emphasize regularization more.

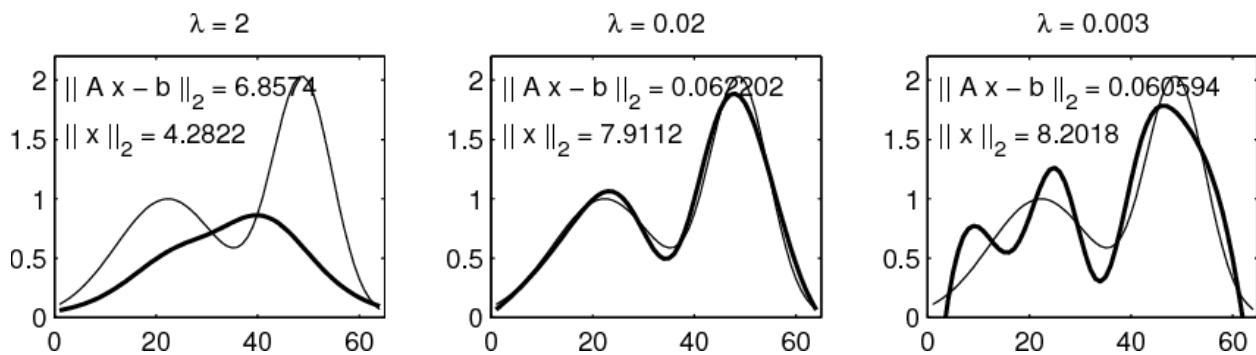


Figure 26: The exact solution (thin lines) and Tikhonov regularized solutions $x\lambda$ (thick lines) for three values of λ corresponding to over-smoothing, appropriate smoothing, and under-smoothing (Hansen & O'Leary, 1993)

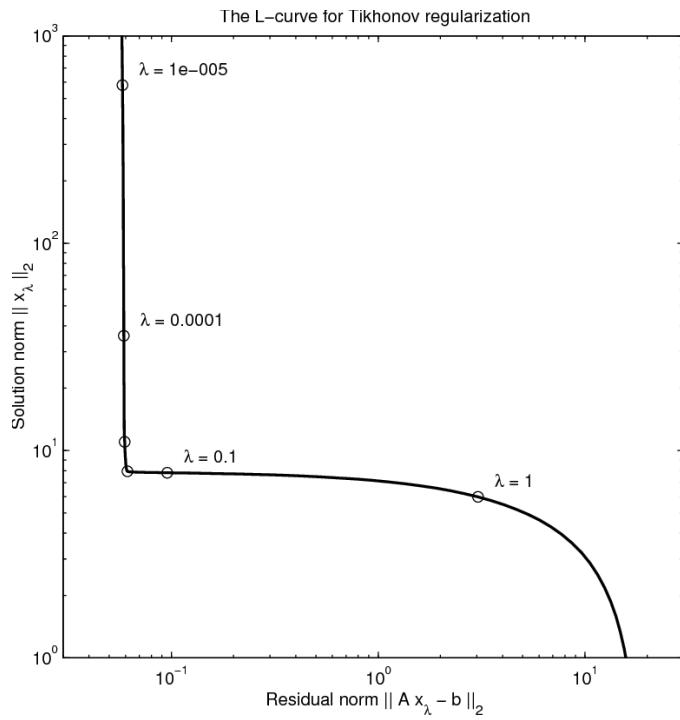


Figure 27: The generic L-curve for standard-form Tikhonov regularization with $x_0 = 0$; the points marked by the circles correspond to the regularization parameters $\lambda = 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$ and 1 (Hansen & O'Leary, 1993)

Role in Mitigating Overfitting:

1. **Complexity Control:** Tikhonov regularization penalizes large parameter values, discouraging overly complex models. This penalty helps prevent overfitting by favoring simpler models that generalize better to new data.
2. **Stability Enhancement:** By introducing a regularization term, Tikhonov regularization stabilizes the estimation process. It minimizes the impact of noisy or redundant features in the data, leading to more robust parameter estimates.
3. **Balancing Fit and Complexity:** The regularization term acts as a balance between fitting the observed data and controlling model complexity. It provides a smoother, more stable optimization landscape, reducing the sensitivity to noise and outliers.

- 4. Robustness to Ill-Conditioning:** In cases where the system of equations is ill-conditioned, Tikhonov regularization helps in stabilizing the solution by reducing the effects of singular or nearly singular matrices.

Underlying Concept:

At its core, Tikhonov regularization embodies a trade-off between fitting the observed data well and keeping the model parameters within reasonable bounds. By incorporating a regularization term into the objective function, it prevents extreme parameter values, thus promoting more stable and generalizable solutions.

In all, Tikhonov regularization, with its emphasis on controlling model complexity and stabilizing solutions, serves as a valuable tool in addressing overfitting, noise, and ill-conditioning in optimization problems. Its ability to strike a balance between data fitting and regularization makes it a cornerstone in enhancing the robustness and reliability of model estimations, particularly in scenarios with limited or noisy data.

2.3.2 Implementation and Analysis

Detailing the practical implementation of Tikhonov regularization within the calibration framework. This part will cover the process of initializing λ values, iterative analysis for different λ values, and visualization of the cost function evolution to determine the optimal λ value.

Lamda	Initial Guess	Optimized results		
		X	Y	Z
0.01	40, 20, 60	32.0211	5.9392	45.6095
	20, 2, 80	37.2838	8.0756	42.0115
0.1	40, 20, 60	32.0624	5.9502	45.5890
	20, 2, 80	32.0625	5.9503	45.5891
0.8	40, 20, 60	32.3827	6.0588	45.4228

	20, 2, 80	32.3828	6.0588	45.4229
1	40, 20, 60	32.4748	6.0970	45.3727
	20, 2, 80	32.4746	6.0972	45.3728
10	40, 20, 60	37.2838	8.0756	42.0115
	20, 2, 80	37.2828	8.0756	42.0115

Here, we are trying to find the minimum value of lamda, such it gives unique result for the different values of prior and initial guess. At high values of lambda (such as 10 or 1), unique results is obtained, however, it could indicate that the regularization term is dominating the optimization process too much, leading to overly simplistic models. In such cases, it may not be effectively balancing between fitting the data and penalizing complex models.

x0_prior. value is [40, 6, 40]

The choice of the regularization parameter, lambda (λ), depends on the specific characteristics of your problem and data. There's no one-size-fits-all value for lambda, and it often requires experimentation.

2.3.2.1 Low Value: $\lambda = 0.01$

- This would introduce minimal regularization and could be suitable if data is well-behaved, and would closely fit the measurements.

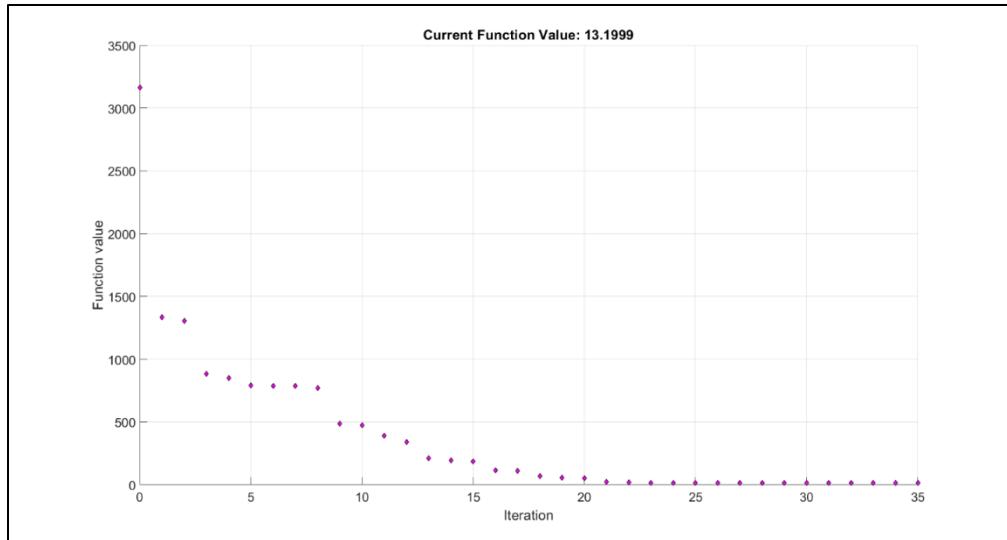


Figure 28: The figure shows variations of the cost function value during the iteration of the multiple Residual mean square loss loss function, using Optimization_Tiko.m. Regularization parameter set to 0.01.

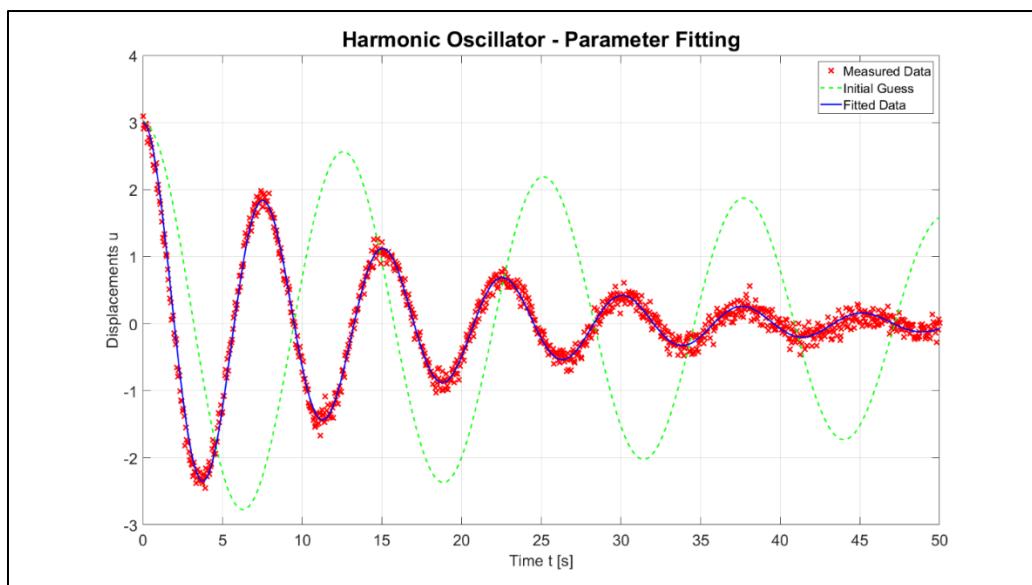


Figure 29: Harmonic Oscillator, plotted for umeas03 as measured data, initial guess in the above table.

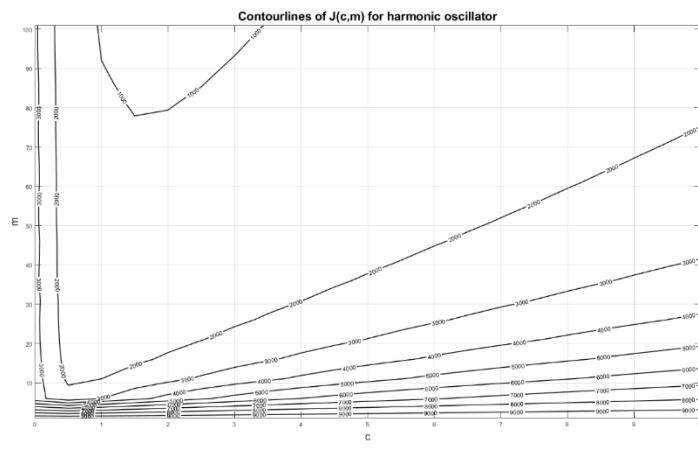
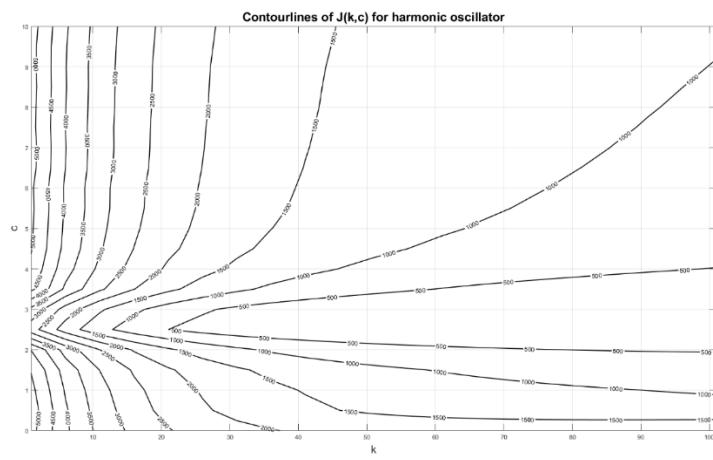
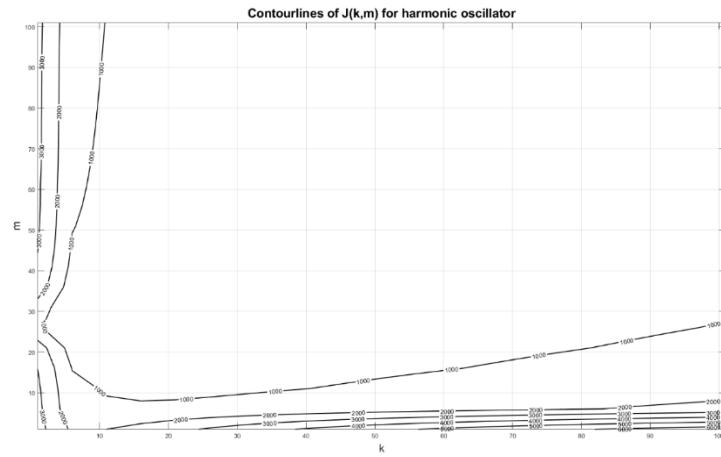


Figure 30: Contour plots for optimization based on two variables : (k,m) , (k,c) , (c,m)

2.3.2.2 Medium-Low Value: $\lambda = 0.1$

- This provides moderate regularization, striking a balance between fitting the data well and preventing overfitting.

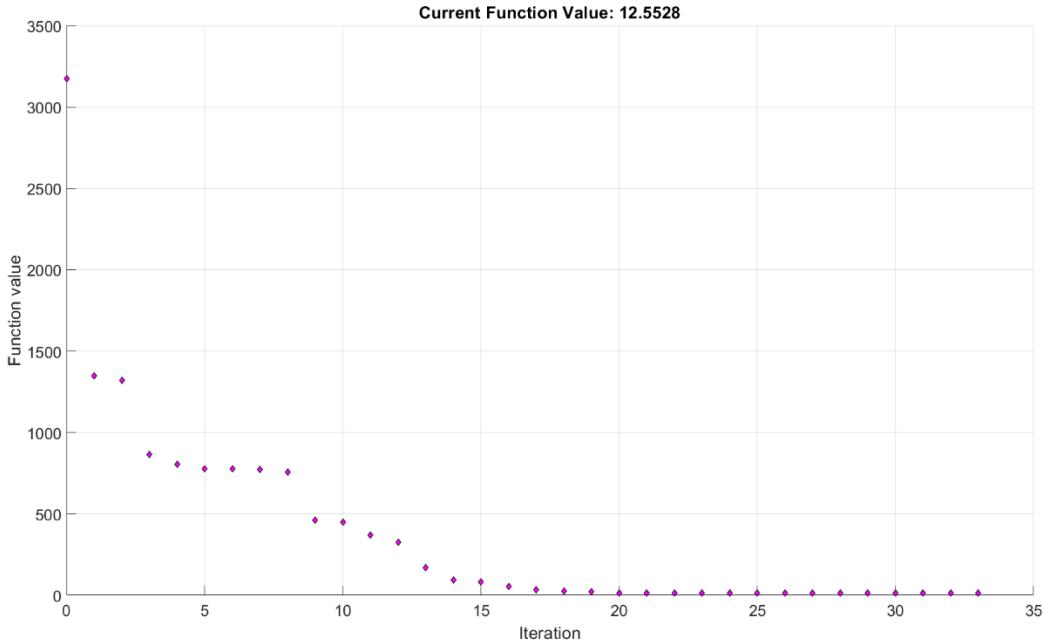


Figure 31: The figure shows variations of the cost function value during the iteration of the multiple Residual mean square loss loss function, using Optimization_Tiko.m. Regularization parameter set to 0.1.

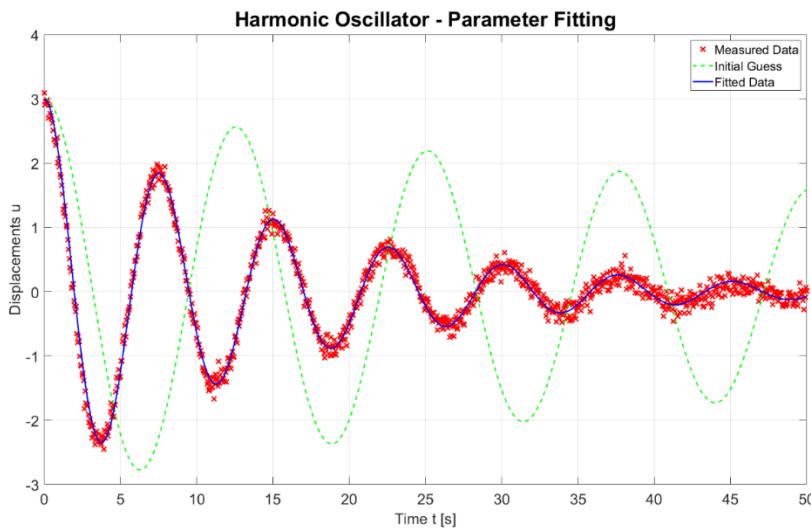
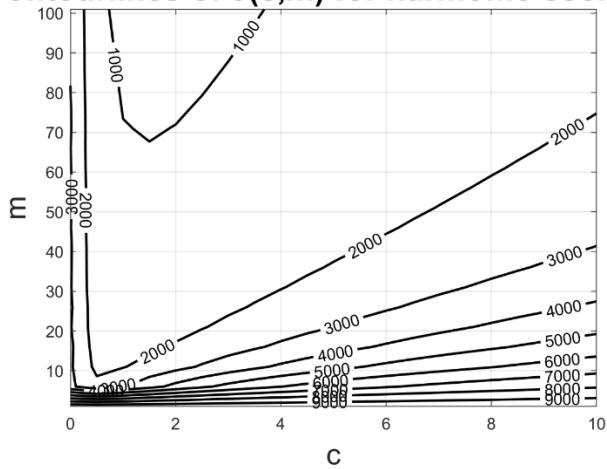
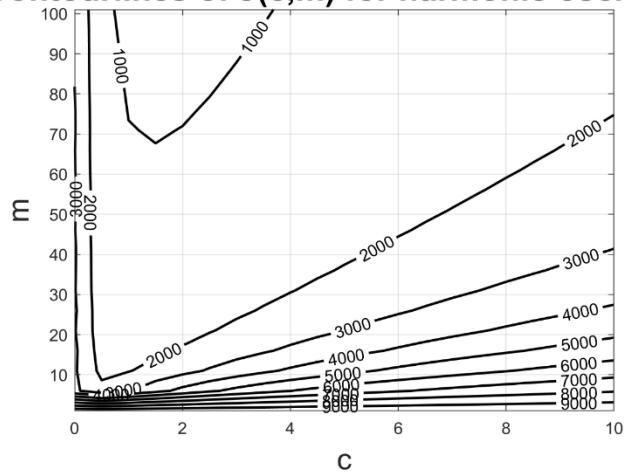


Figure 32: Harmonic Oscillator, plotted for umeas03 as measured data, initial guess in the above table.

Contourlines of $J(c,m)$ for harmonic oscillator



Contourlines of $J(c,m)$ for harmonic oscillator



Contourlines of $J(c,m)$ for harmonic oscillator

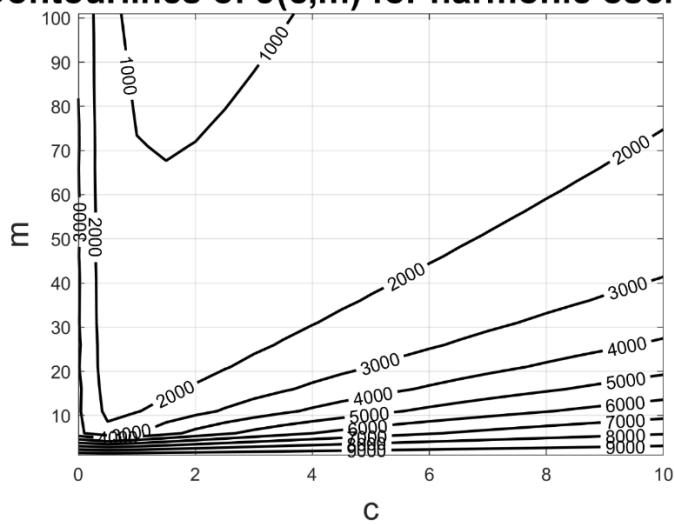


Figure 33: Contour plots for optimization based on two variables : (k,m) , (k,c) , (c,m)

2.3.2.3 Medium-High Value: $\lambda = 0.8$

- This introduces a higher level of regularization, suitable when suspecting some degree of noise or outliers in data.

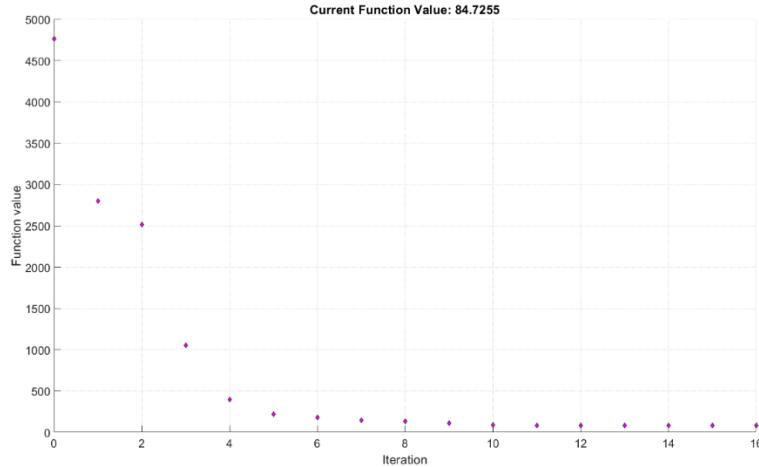


Figure 34: The figure shows variations of the cost function value during the iteration of the multiple Residual mean square loss function, using Optimization_Tiko.m. Regularization parameter set to 0.8

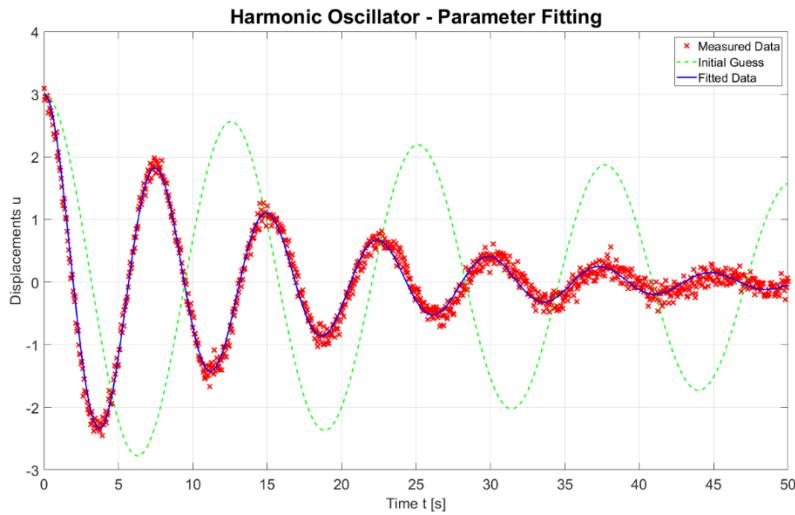


Figure 35: Harmonic Oscillator, plotted for umeas03 as measured data, initial guess in the above table

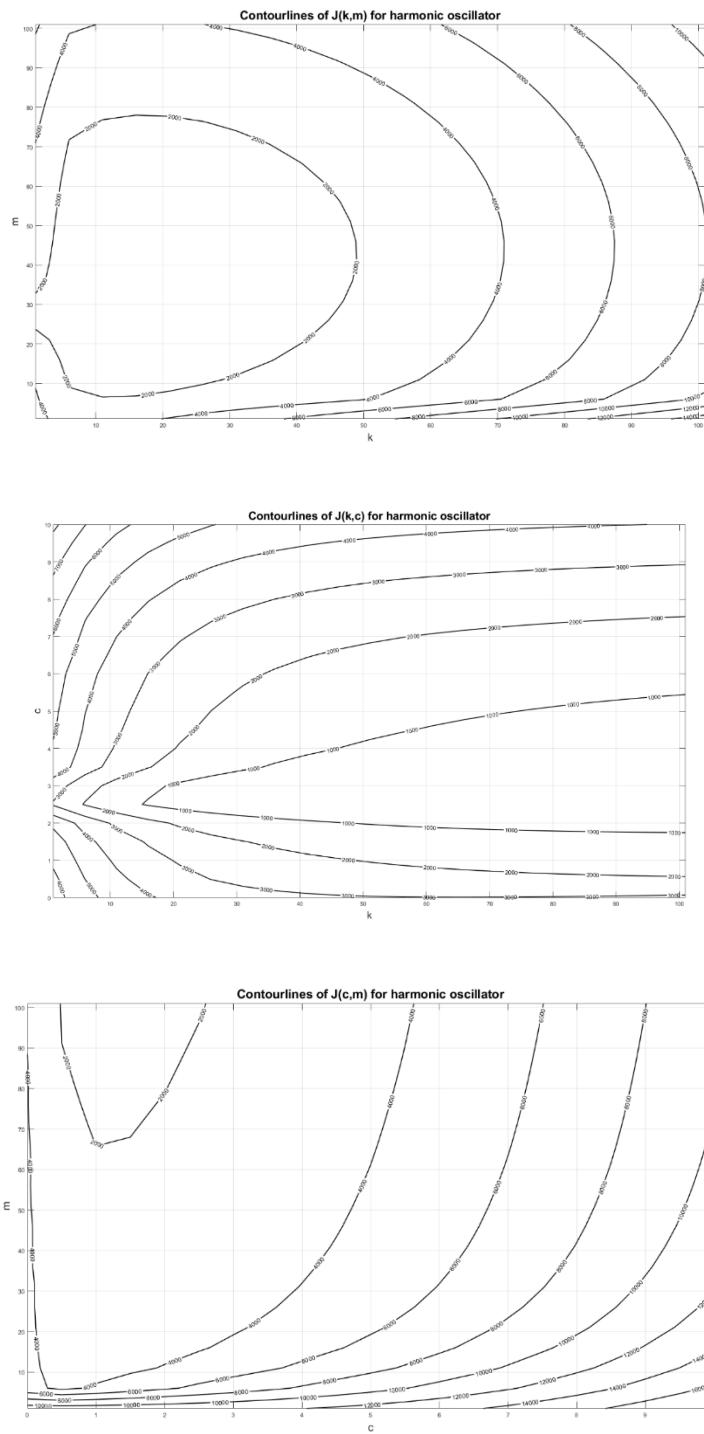


Figure 36: Contour plots for optimization based on two variables : (k,m) , (k,c) , (c,m)

2.3.2.4 Medium-High Value: $\lambda = 1$

- This introduces a higher level of regularization, expecting some degree of noise or outliers in data.

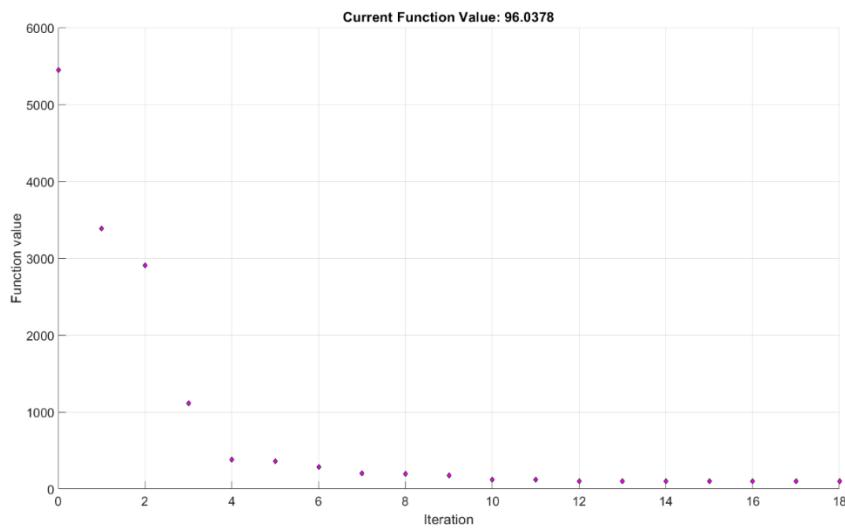


Figure 37: The figure shows variations of the cost function value during the iteration of the multiple Residual mean square loss loss function, using Optimization_Tiko.m. Regularization parameter set to 1.

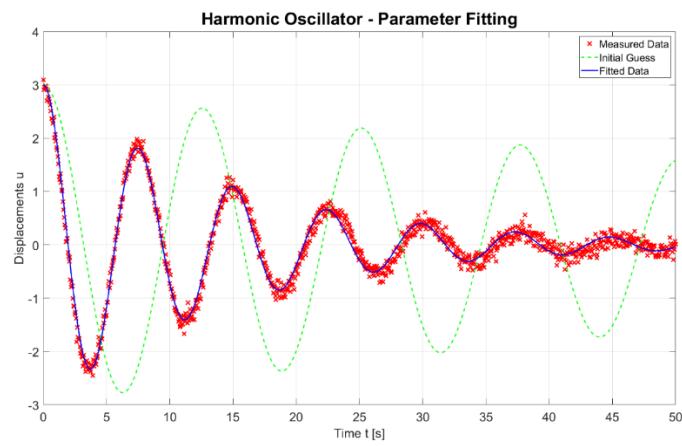
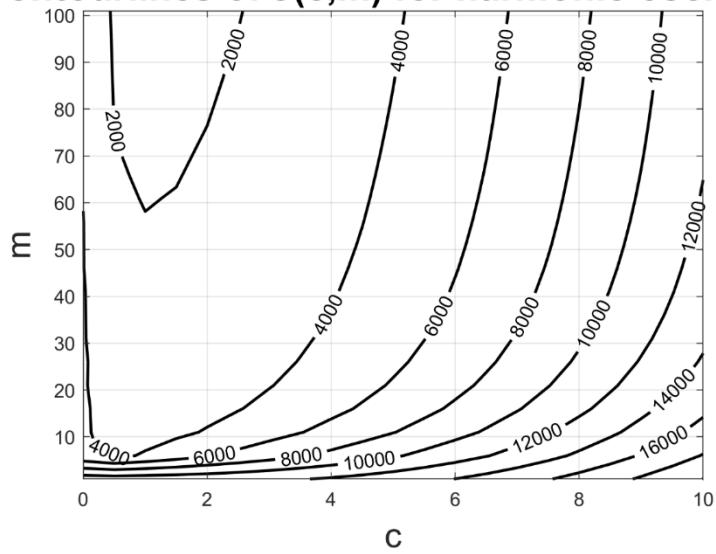
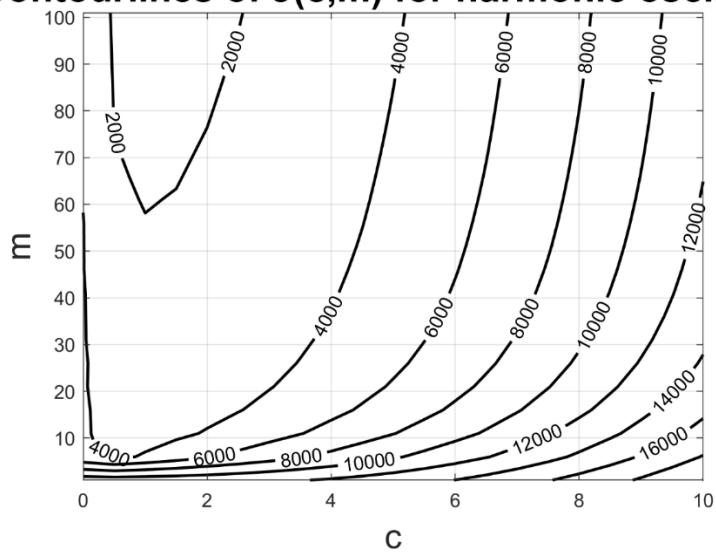


Figure 38: Harmonic Oscillator, plotted for umeas03 as measured data, initial guess in the above table.

Contourlines of $J(c,m)$ for harmonic oscillator



Contourlines of $J(c,m)$ for harmonic oscillator



Contourlines of $J(c,m)$ for harmonic oscillator

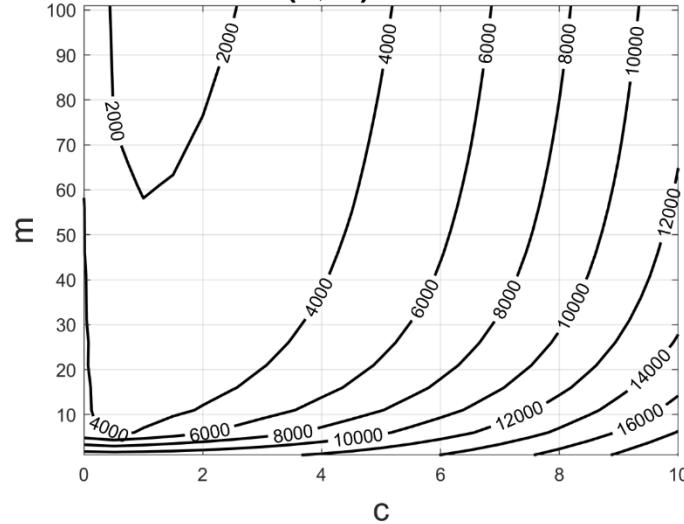


Figure 39: Contour plots for optimization based on two variables : (k,m) , (k,c) , (c,m)

2.3.2.5 High Value: $\lambda = 10$

- Using a high λ , data is noisy, and want to prioritize a smoother fit over closely matching the measurements.

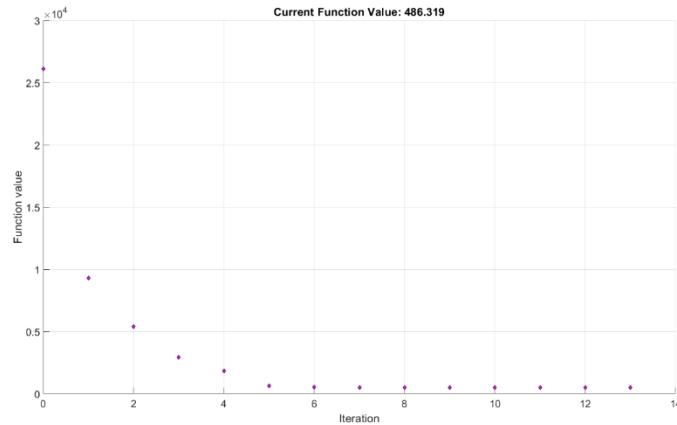


Figure 40: The figure shows variations of the cost function value during the iteration of the multiple Residual mean square loss loss function, using Optimization_Tiko.m. Regularization parameter set to 10.

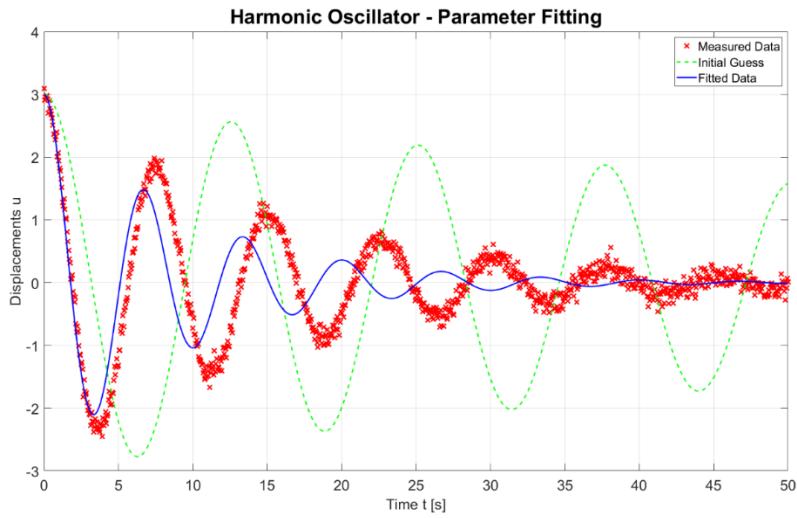
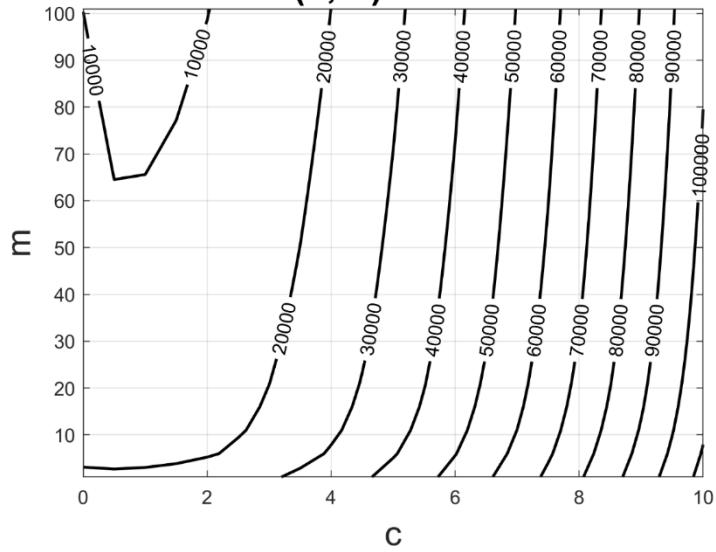
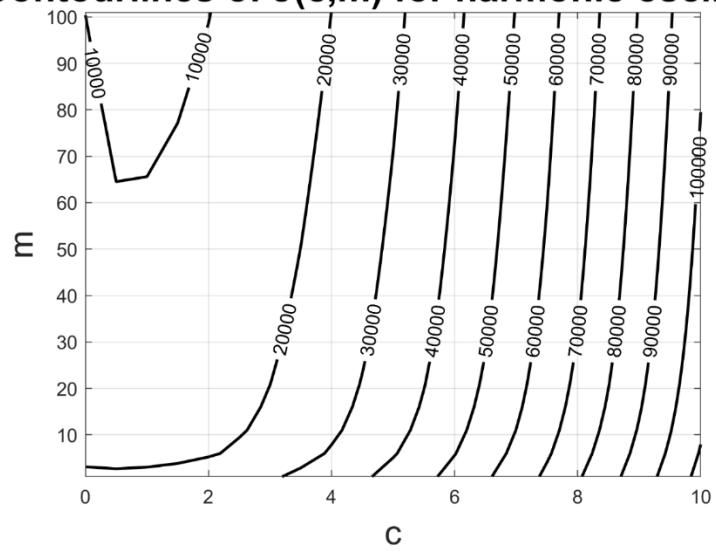


Figure 41: Harmonic Oscillator, plotted for umeas03 as measured data, initial guess in the above table. The effect we can observe due to high lambda, Global optimization function is optimizing the regularization function not the loss function.

Contourlines of $J(c,m)$ for harmonic oscillator



Contourlines of $J(c,m)$ for harmonic oscillator



Contourlines of $J(c,m)$ for harmonic oscillator

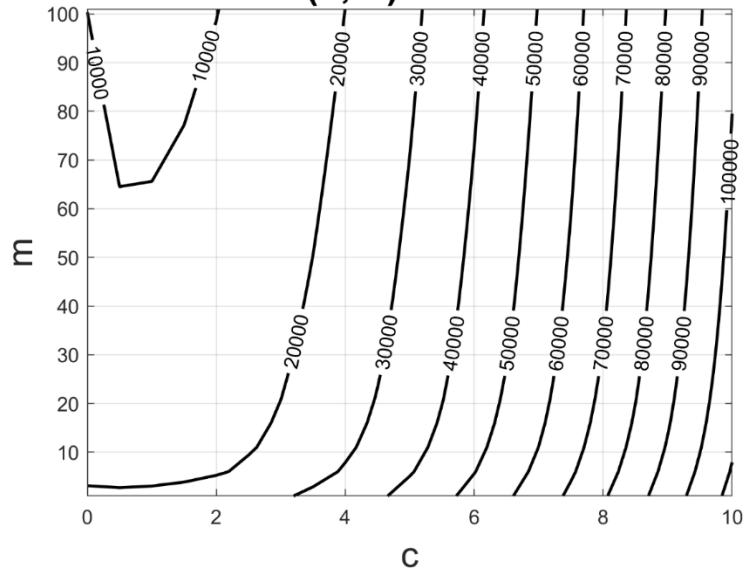


Figure 42: Contour plots for optimization based on two variables : (k,m) , (k,c) , (c,m)

2.3.2.6 Reasoning or Observations:

The optimization process starts to prioritize regularization over the main objective function as lambda increases, particularly noticeable with high values such as 10.

The pattern of the fitted data closely replicates that of the measured data when lambda falls within the range of 0.01 to 1. Through trial and error, the optimal lambda value is found to be near 0.1

2.3.3 Algorithmic Enhancements

Elaborating on the development of MATLAB functions dedicated to computing residual and solution norms. Additionally, this part will focus on scripting procedures for visualizing the L-curve to identify the most suitable λ value for calibration enhancement.

The L-Curve is a graphical representation used for accurately fitting data and keeping solutions simple and low in complexity.

By plotting the residual norm against the solution norm, it can help establish an optimal trade-off point.

- L-curves obtained for $u_{\text{meas}0}$ to $u_{\text{meas}09}$ are plotted below:

- is below:

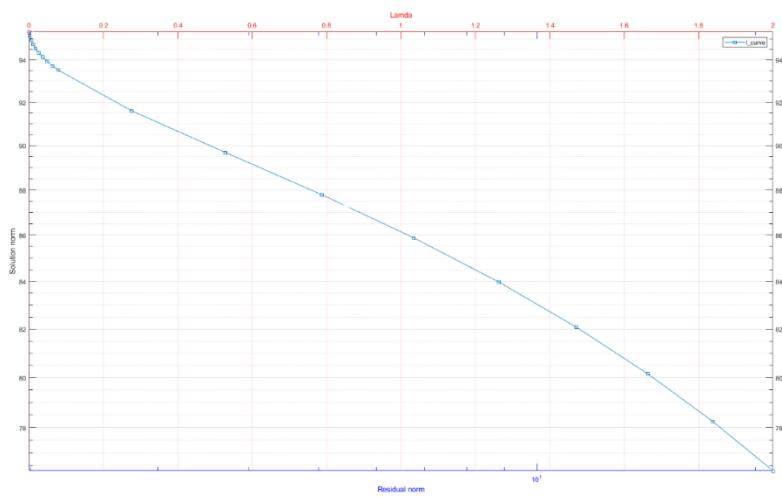


Figure 43: L-curve for $u_{\text{meas}0}$

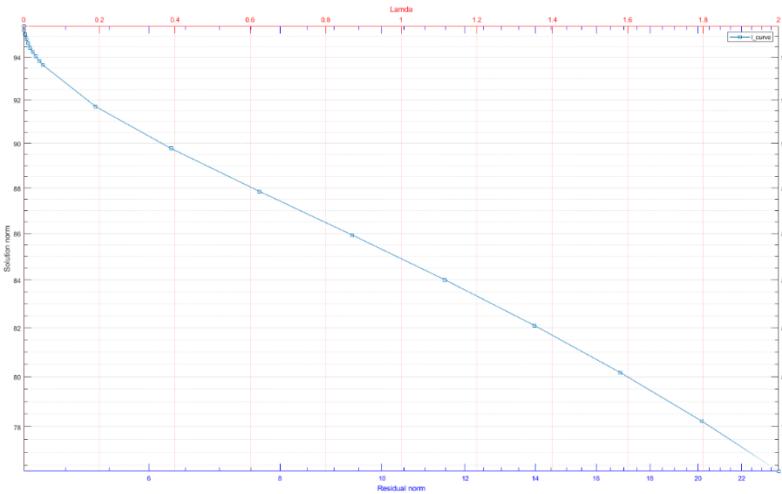


Figure 44: L-curve for $u_{\text{meas}01}$

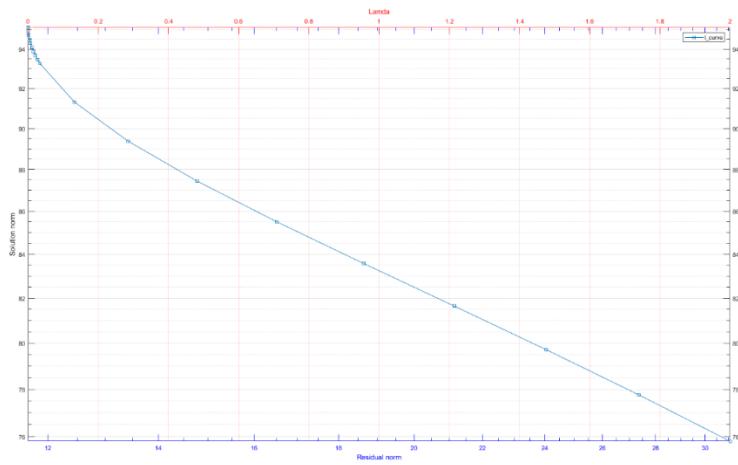


Figure 45: L-curve for *u_meas3*

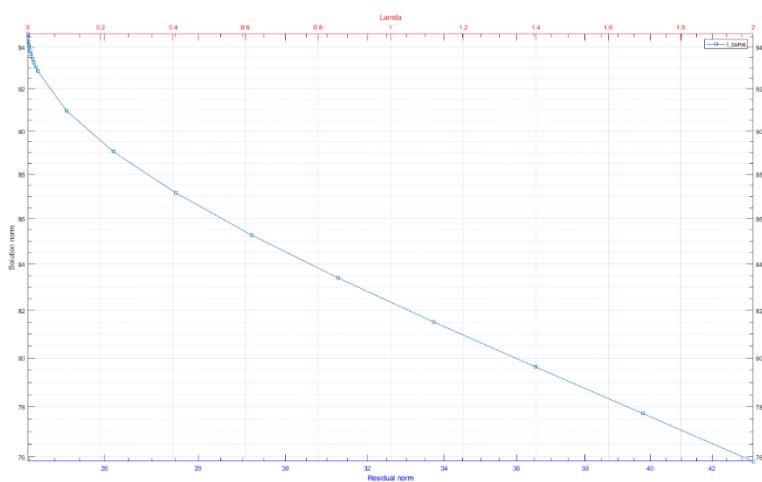


Figure 46: L-curve for *u_meas05*

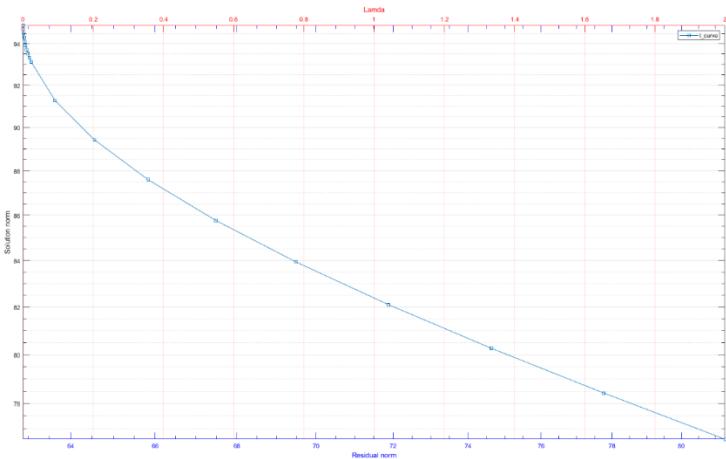
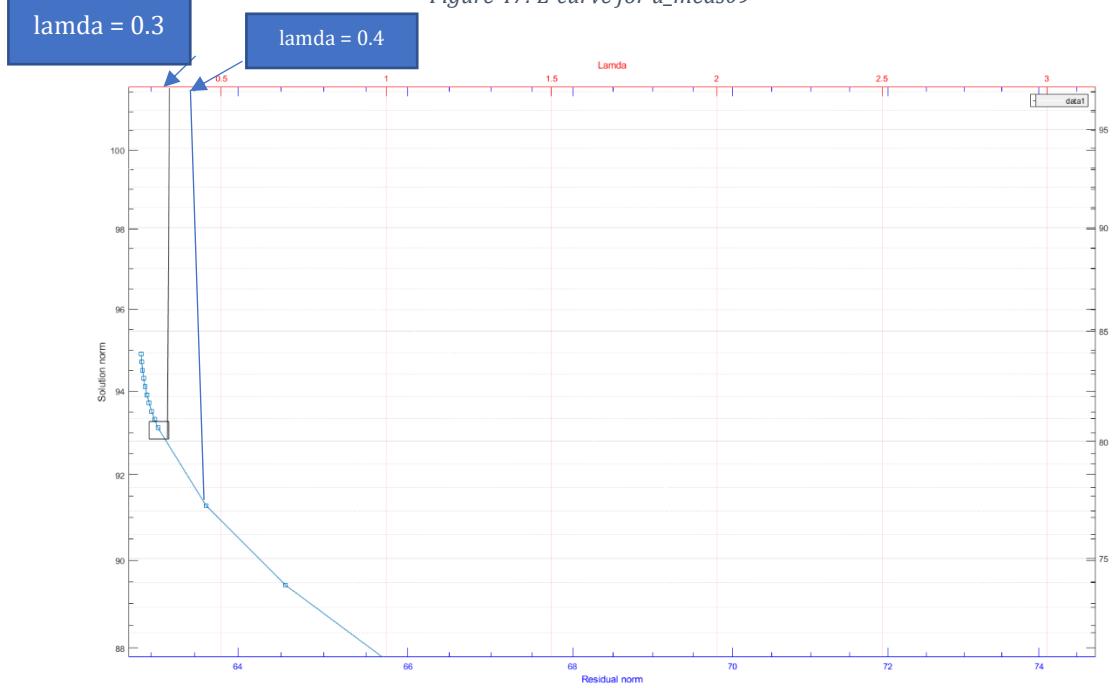


Figure 47: L-curve for $u_{\text{meas}09}$



2.3.2.7 Observation:

We can see the greatest change in the curve between 0.2 and 0.5 in all of the above L-curves, hence we can consider $\lambda = 0.4$ as the optimal value. The initial descent is notably steep, followed by a more gradual incline. Within this dynamic range, the optimal lambda value is anticipated to fall between 0.2 and 0.4. After careful consideration, 0.3 emerges as the preferred lambda value for subsequent computations. This choice is grounded in the

observation that it strikes a balance between capturing the initial rapid descent and accommodating the subsequent smoother transition in the slope.

2.4 Application of λ to Least Squares Calibration

2.4.1 Integrating λ into Previous Calibration

This section aims to integrate the λ value derived from Tikhonov regularization into the prior calibration methodology based on Least Squares. The comparative analysis will emphasize the impact of applying the new λ value on calibration accuracy and robustness.

2.4.2 Extended Data Analysis (if applicable)

If time permits, exploring supplementary data from `u_meas0.mat` and `u_meas09.mat` to further validate the enhanced calibration approach.

Initial value chosen is 20, 2, 80 -- [k, c, m]

and lamda = 0.4

`x0_prior = [40, 6, 40];`

`umeas0`

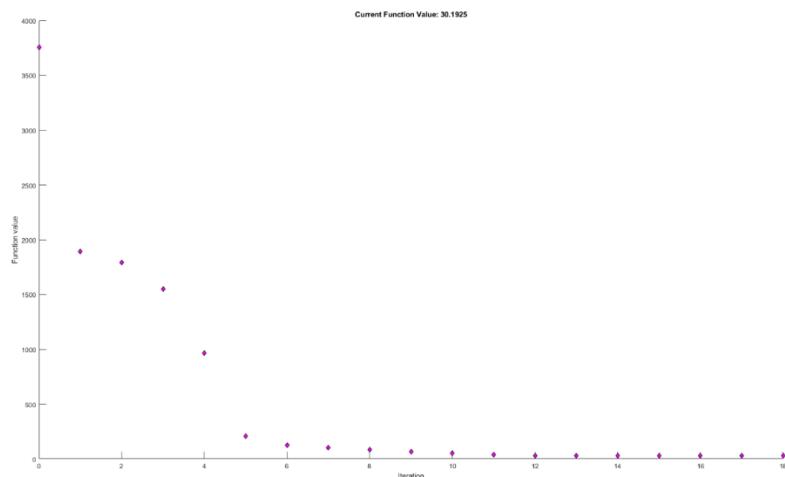


Figure 48: The figure shows variations of the cost function value during the iteration of the Residual mean square loss function, using Optimization_Tiko.m. Regularization parameter set to 0.4.

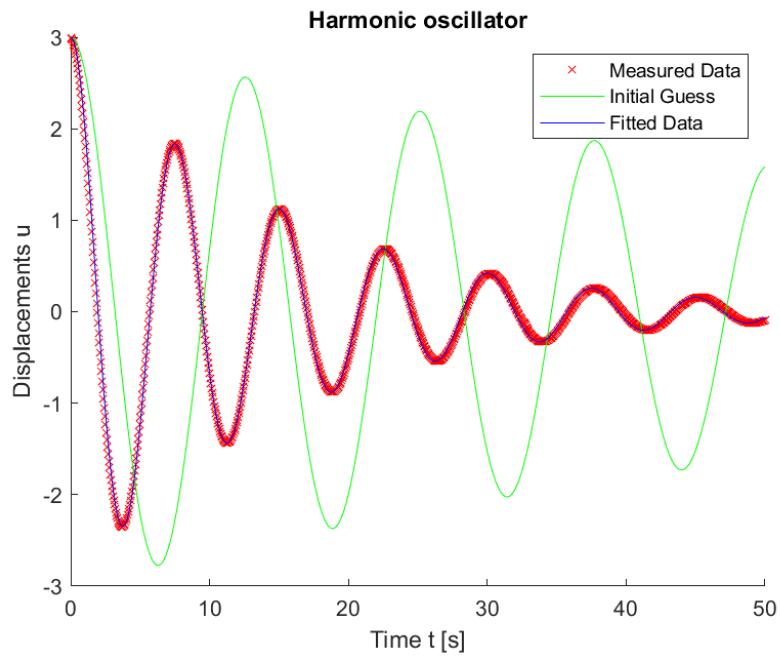
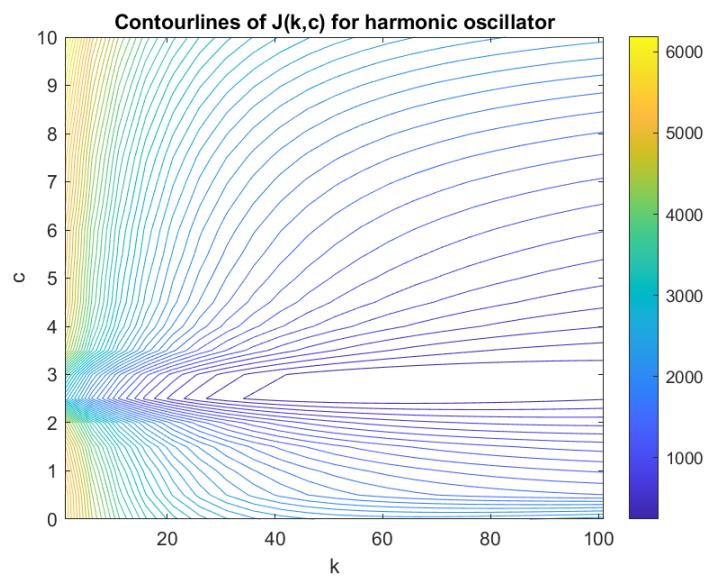
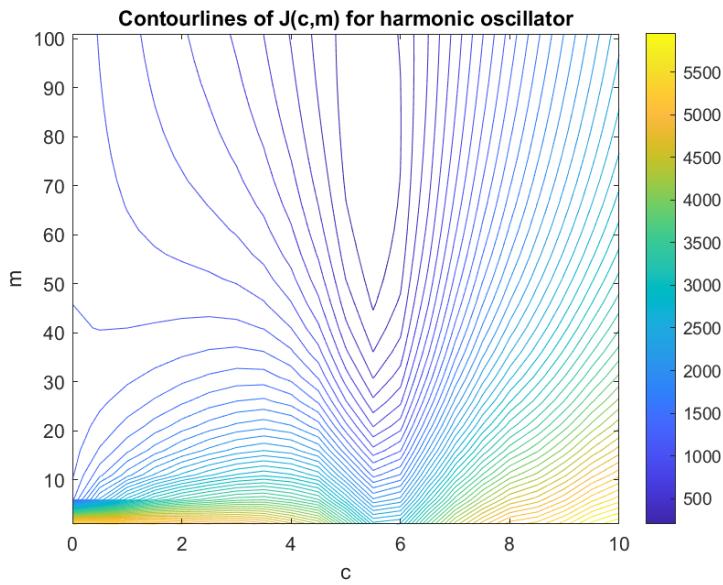


Figure 49: Harmonic Oscillator, plotted for umeas03 as measured data, initial guess as 20, 2, 80 .



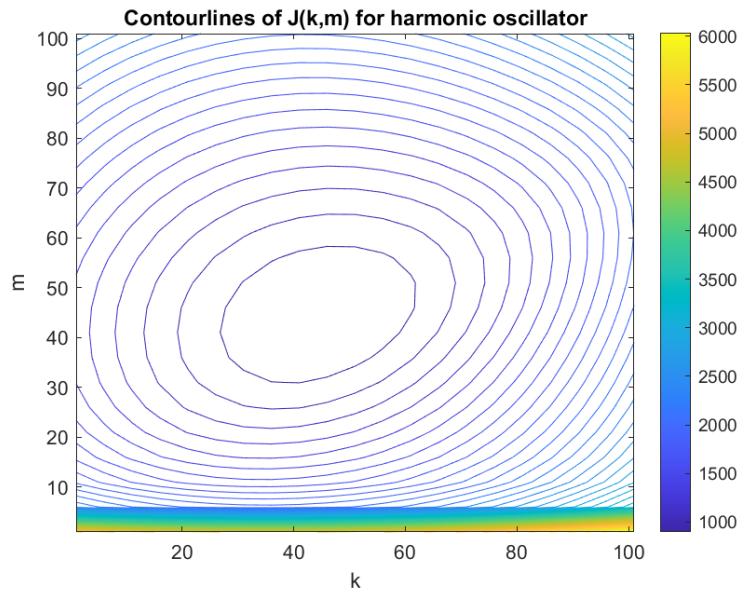


Figure 50: Contour plots for optimization based on two variables : (k,m) , (k,c) , (c,m)

umeas01

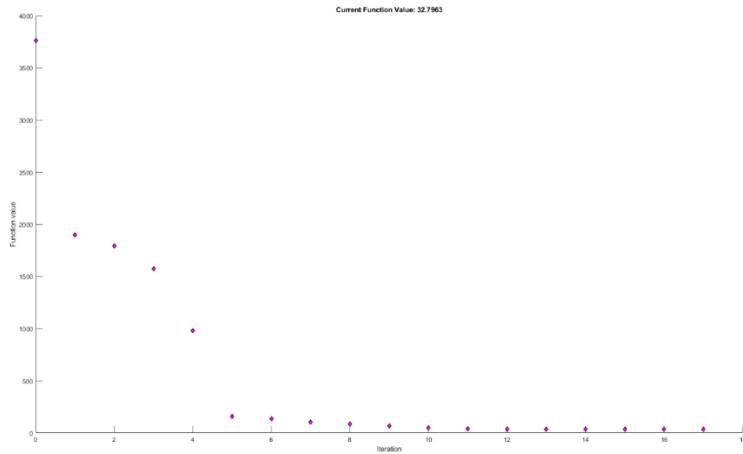


Figure 51: The figure shows variations of the cost function value during the iteration of the Residual mean square loss function, using Optimization_Tiko.m. Regularization parameter set to 0.4.

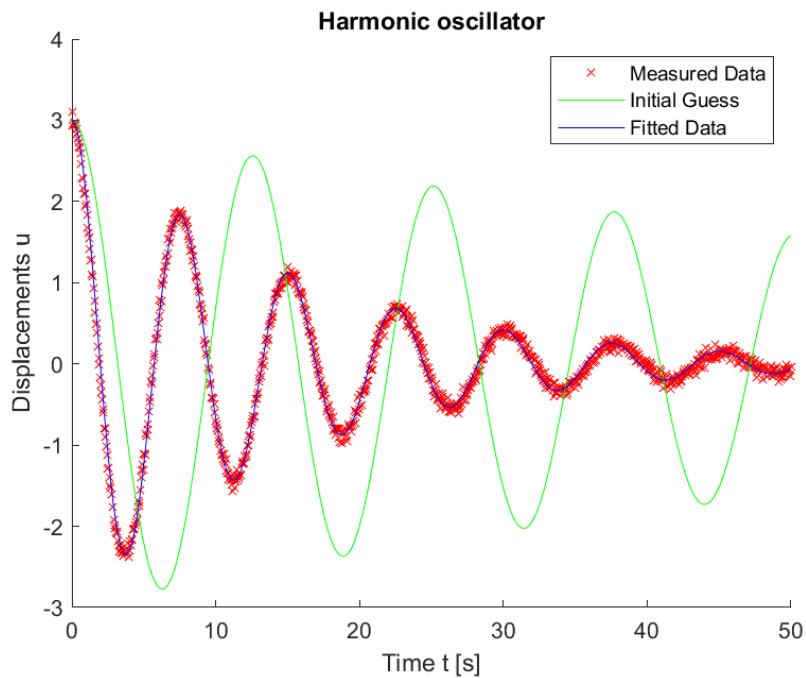
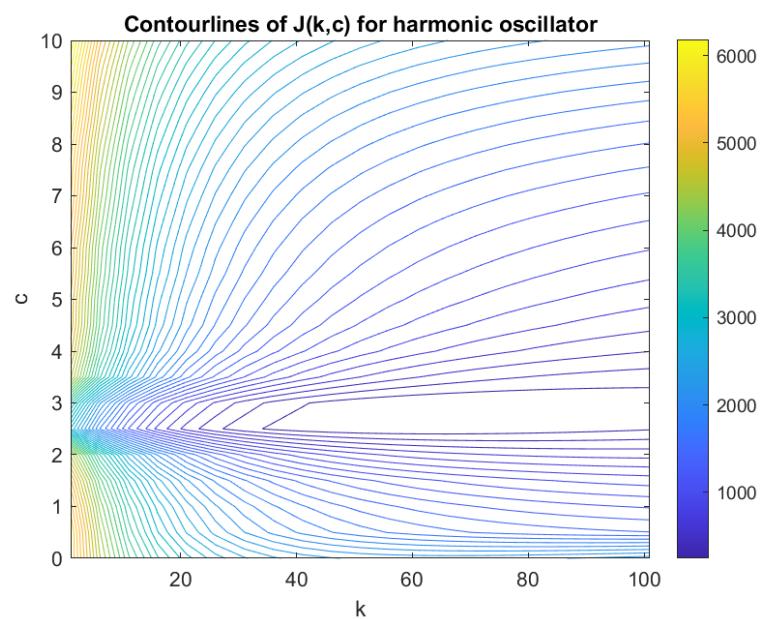
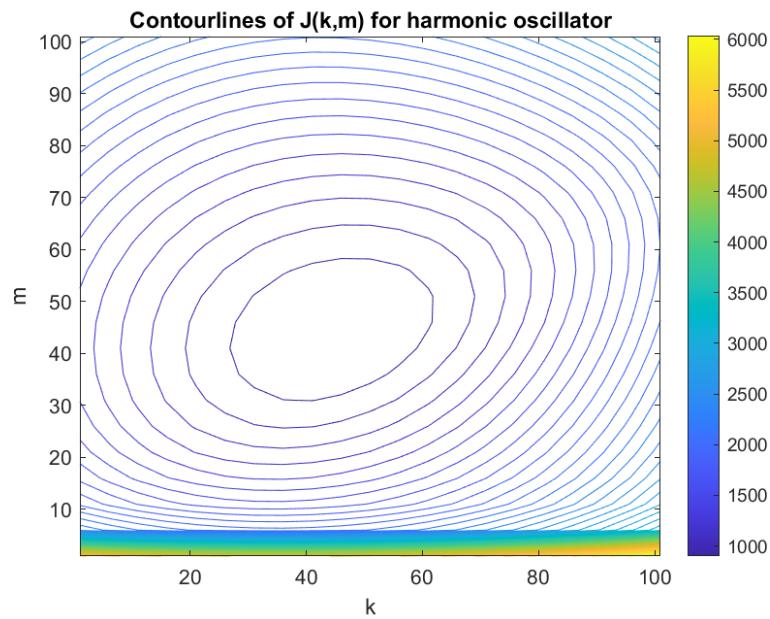


Figure 52: Harmonic Oscillator, plotted for $umeas01$ as measured data, initial guess as 20 , 2 ,80.



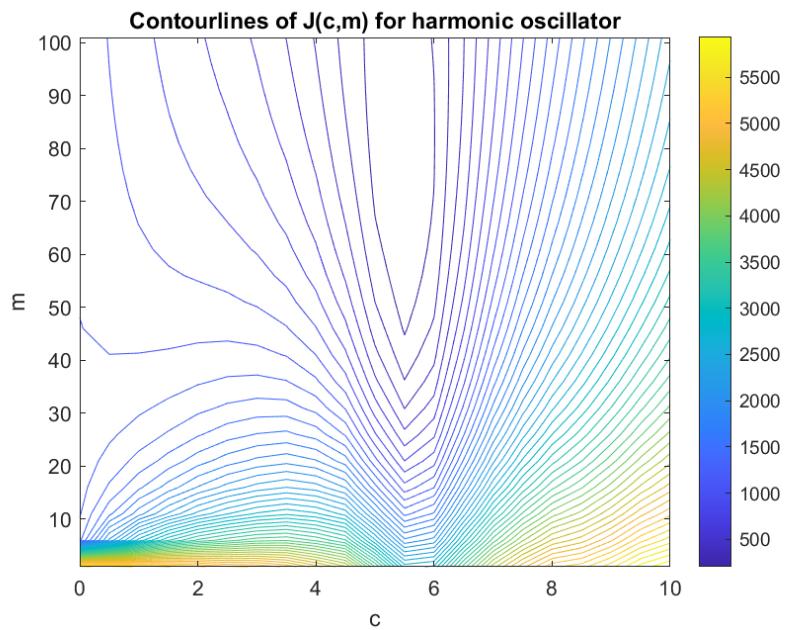


Figure 53: Contour plots for optimization based on two variables : (k,m) , (k,c) , (c,m)

umeas03

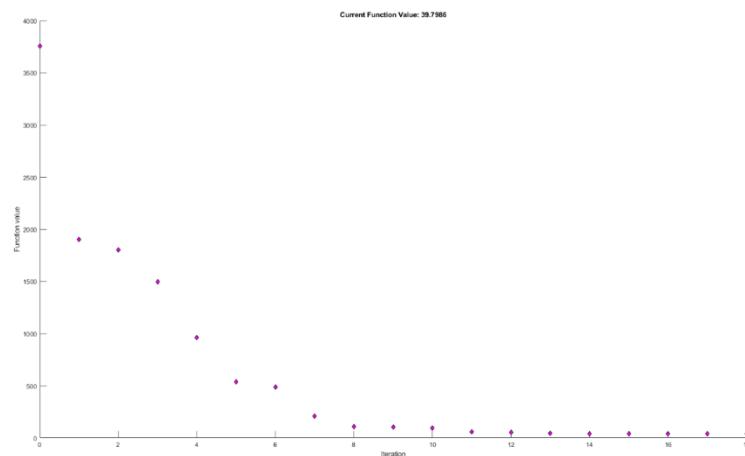


Figure 54: The figure shows variations of the cost function value during the iteration of the Residual mean square loss function, using Optimization_Tiko.m. Regularization parameter set to 0.4.

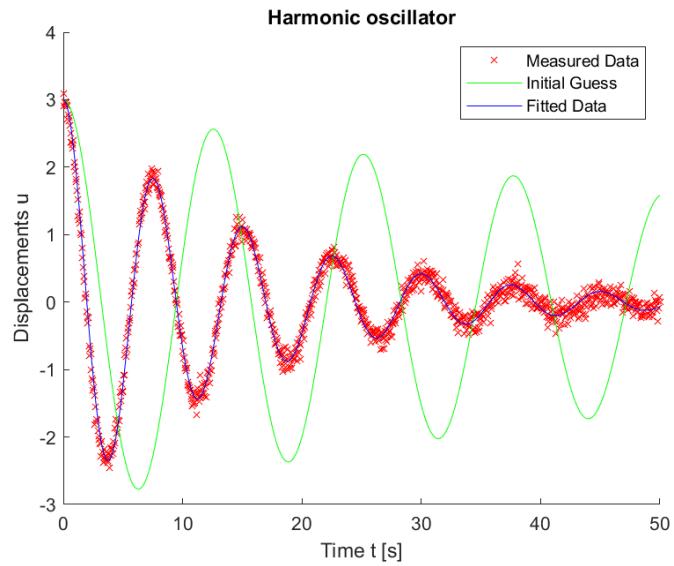
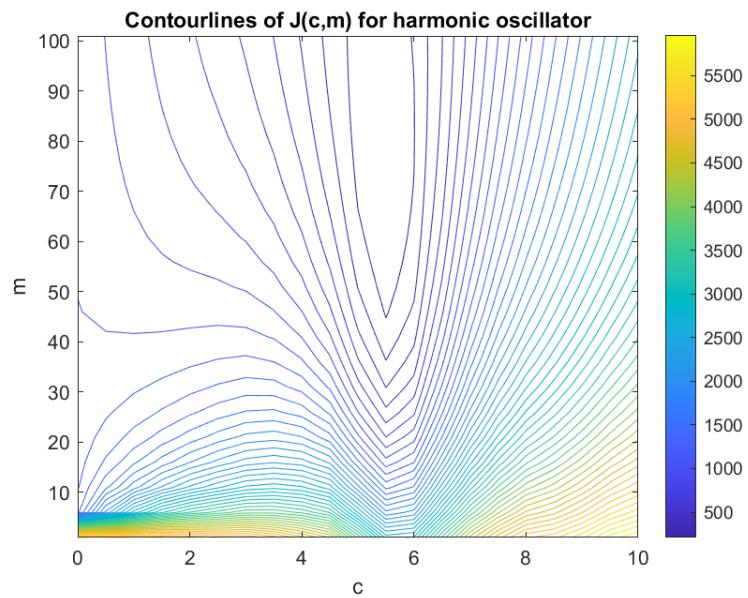


Figure 55: Harmonic Oscillator, plotted for $umeas03$ as measured data, initial guess 20, 2, 80 .



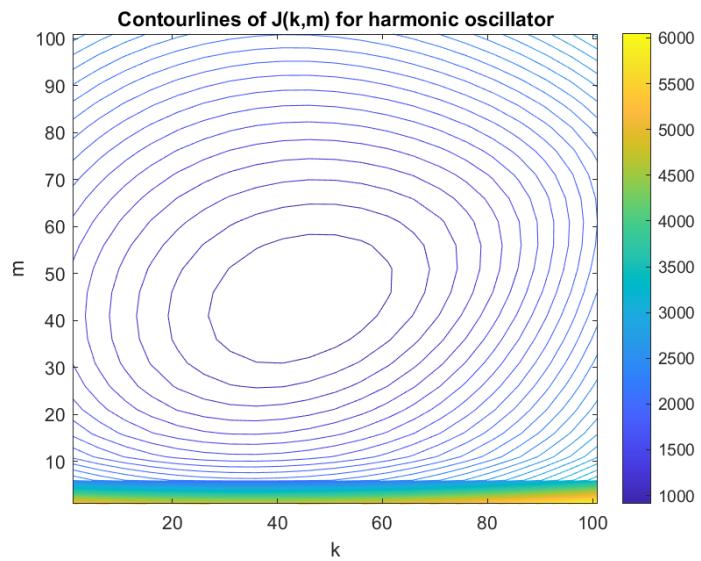
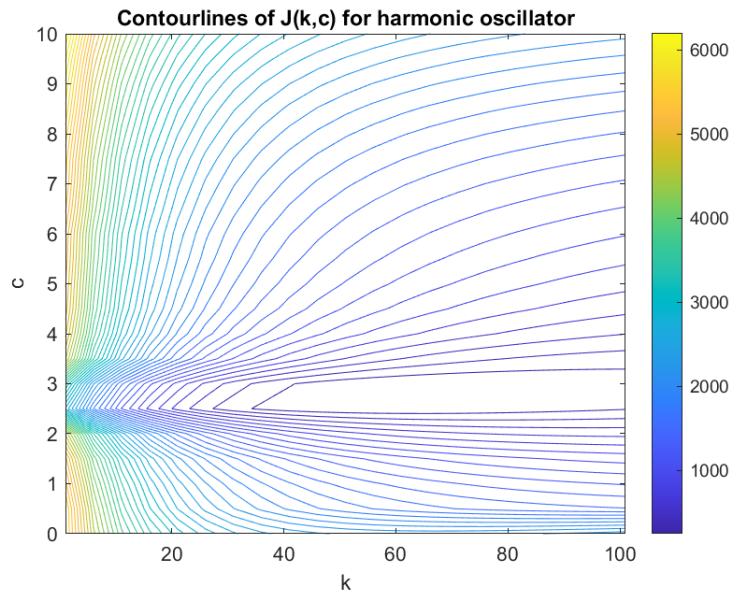


Figure 56: Contour plots for optimization based on two variables : (k,m) , (k,c) , (c,m)

umeas05

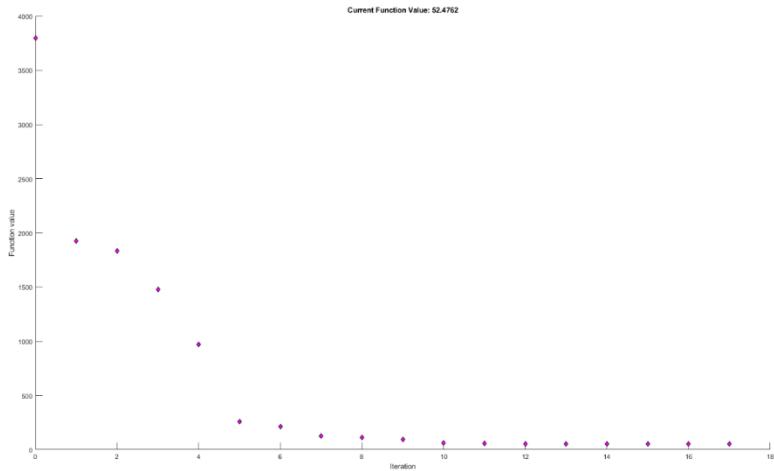


Figure 57: The figure shows variations of the cost function value during the iteration of the Residual mean square loss function, using Optimization_Tiko.m. Regularization parameter set to 0.4.

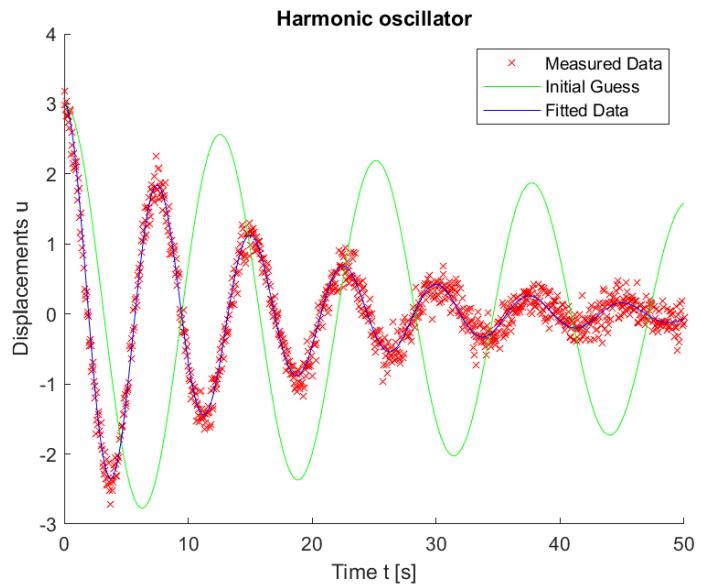
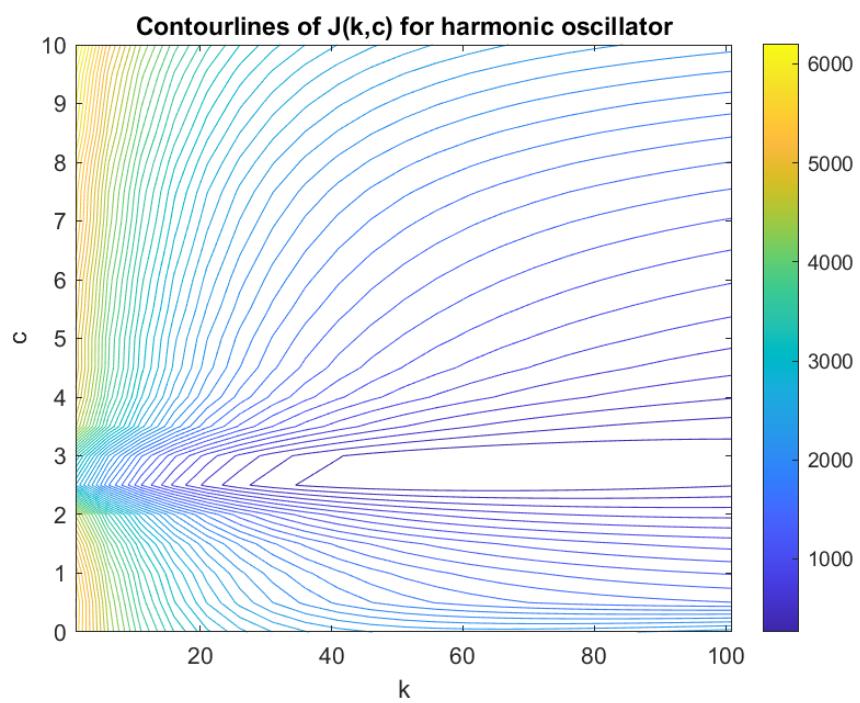
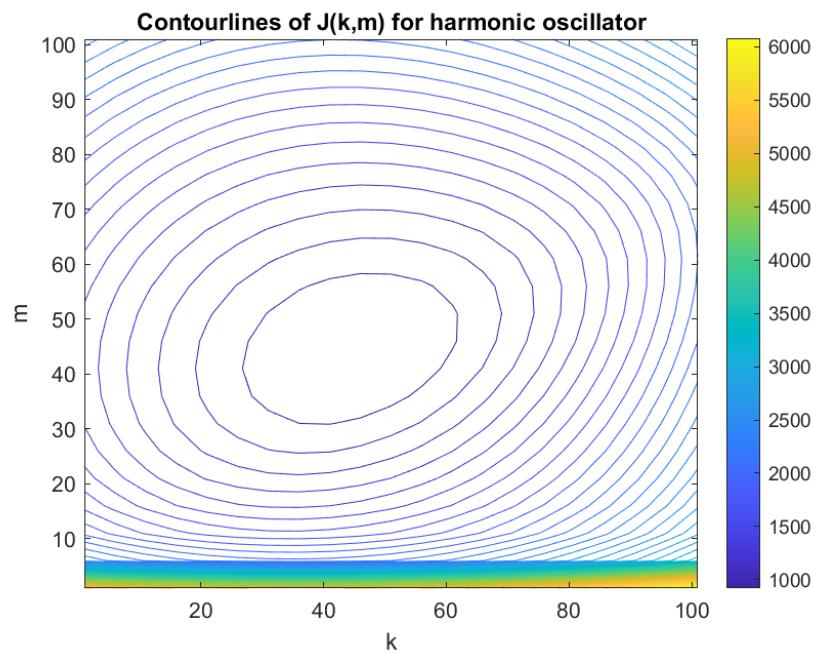


Figure 58: Harmonic Oscillator, plotted for umeas05 as measured data, initial guess 20, 2, 80 .



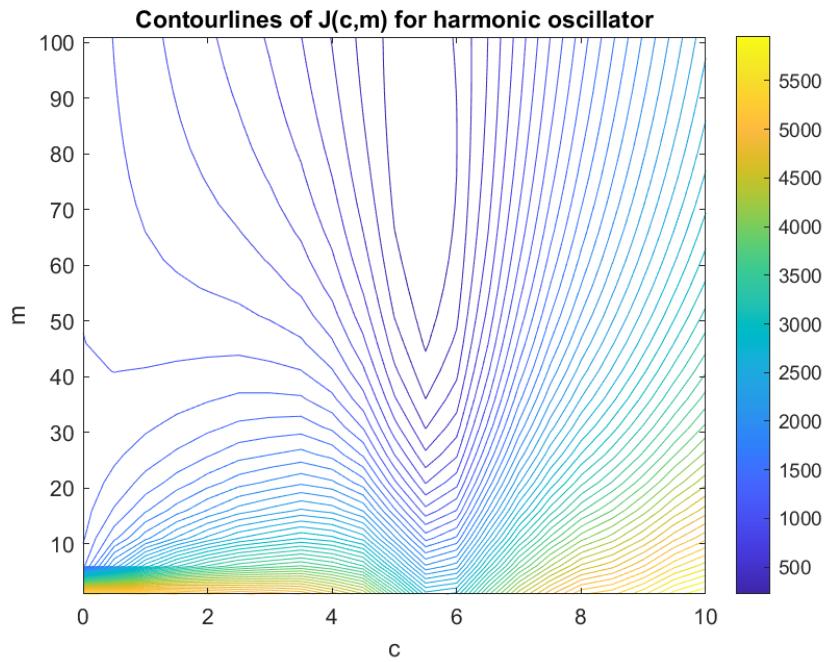


Figure 59: Contour plots for optimization based on two variables : (k,m) , (k,c) , (c,m)

umeas09

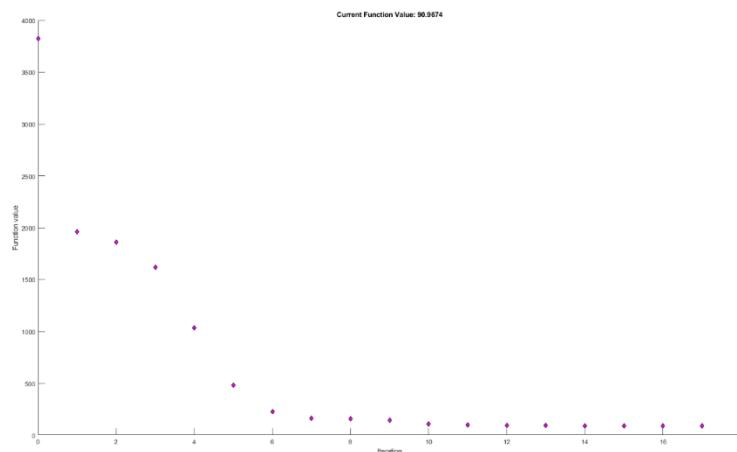


Figure 60: The figure shows variations of the cost function value during the iteration of the Residual mean square loss function, using Optimization_Tiko.m. Regularization parameter set to 0.4.

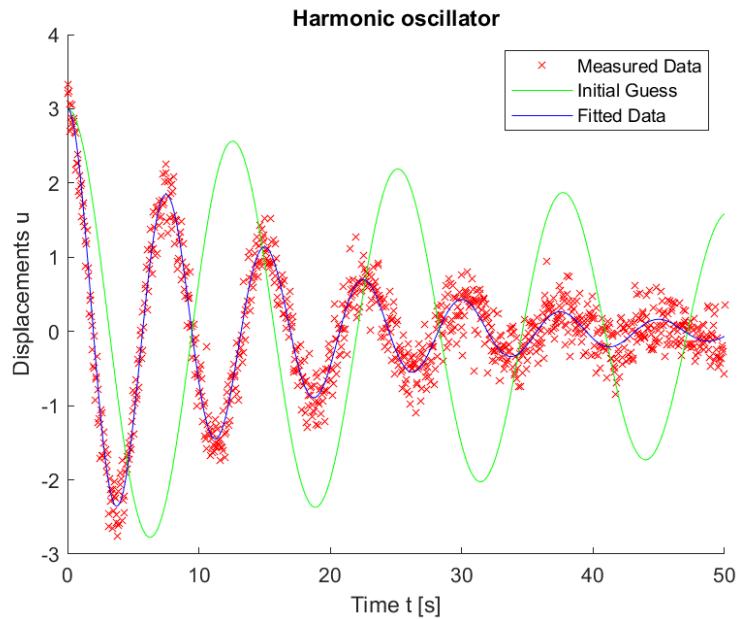
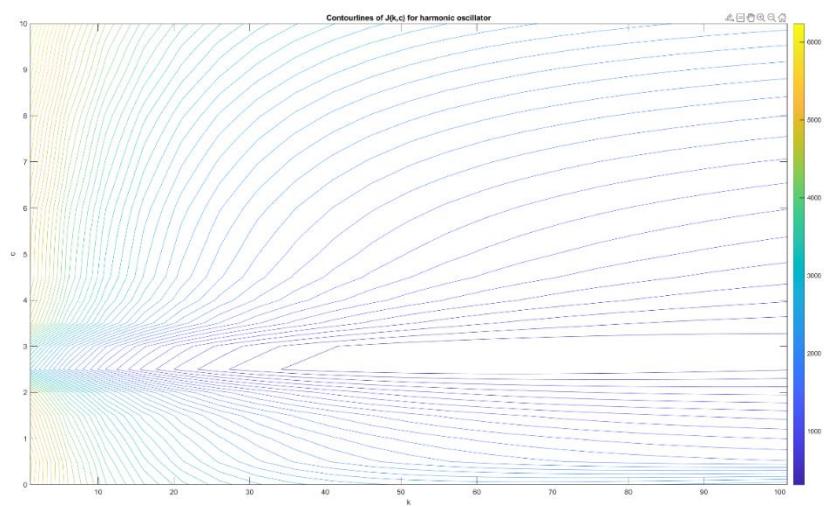
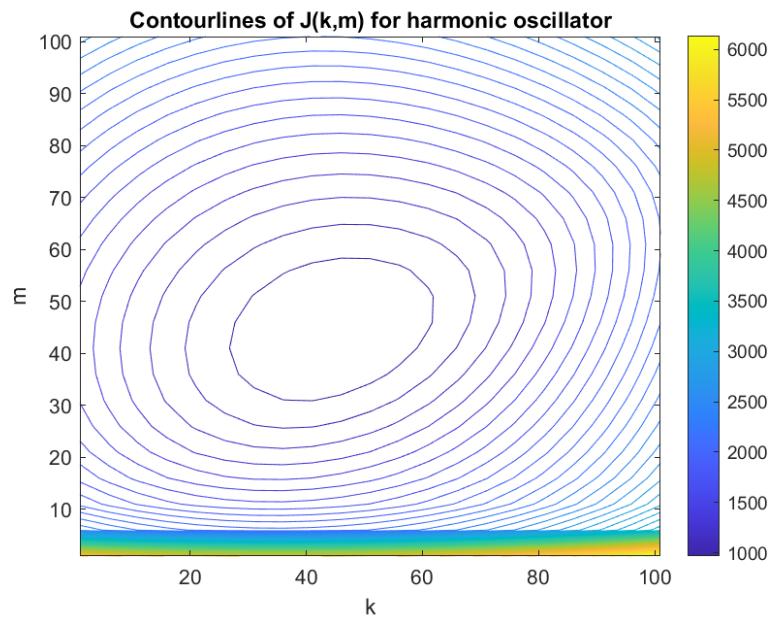


Figure 61: Harmonic Oscillator, plotted for $umeas09$ as measured data, initial guess 20, 2, 80 .



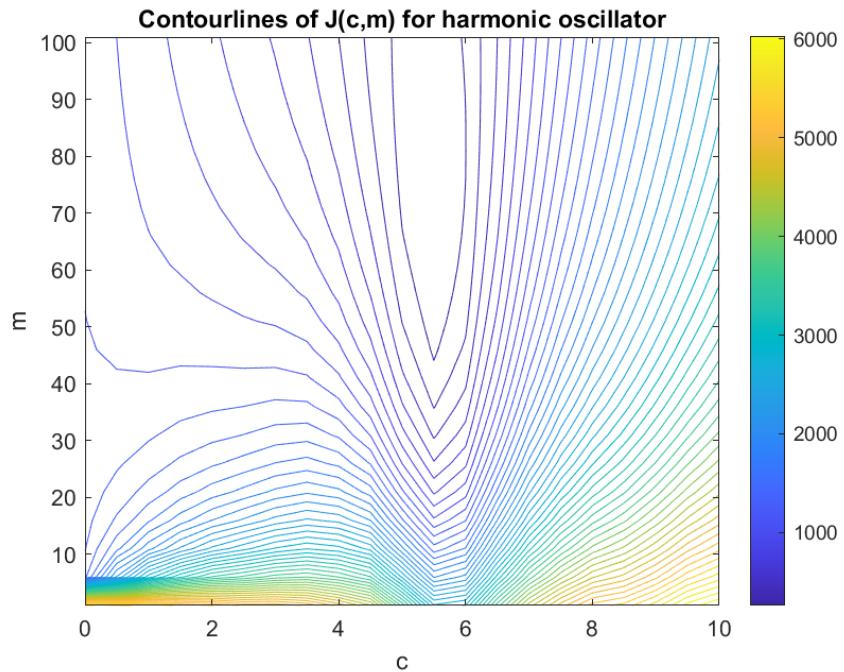


Figure 62: Contour plots for optimization based on two variables : (k, m) , (k, c) , (c, m)

2.5 Comparison of optimized values from calibration experiment 1 and 2.

Initial value : [20, 2, 80]

Initial guess	Optimised value from Approach 1	Optimised value from Approach 2
[60, 5, 20]	46.9937 8.6884 66.9777	57.6117 10.0000 10.0003
[30, 5, 20]	23.4877 4.3425 33.4758	19.4720 3.6332 27.6059
[60, 10, 20]	52.2352 9.6575 74.4484	57.6117 10.0000 10.0003
[20, 10, 30]	21.7767 4.0262 31.0372	21.0865 3.9940 30.0359

Approach 1: Direct Minimization using fmincon

In this approach:

- The optimization problem is formulated as a constrained nonlinear optimization problem using the fmincon function.
- The objective function to minimize is the cost function, which measures the discrepancy between the measured data and the model predictions.
- The initial values for the parameters [k, c, rho] are provided as starting points for the optimization algorithm.
- Tikhonov regularization is applied by setting a non-zero value for λ , which penalizes solutions that deviate too much from a specified prior.
- The optimization algorithm iteratively adjusts the parameters to minimize the cost function subject to the specified constraints.
- After optimization, the optimized parameters are used to generate fitted data, which is then compared with the measured data to assess the quality of the fit.

Approach 2: Unconstrained Minimization using fminunc

In this approach:

- The optimization problem is formulated as an unconstrained nonlinear optimization problem using the fminunc function.
- The objective function to minimize is the same cost function used in Approach 1.
- The initial values for the parameters [k, c, rho] are provided as starting points for the optimization algorithm.
- Tikhonov regularization is not explicitly applied in this approach.
- The optimization algorithm iteratively adjusts the parameters to minimize the cost function without any constraints.
- After optimization, the optimized parameters are used to generate fitted data, which is then compared with the measured data to assess the quality of the fit.

2.5.1 Comparison:

Optimization Strategy:

Approach 1 uses constrained optimization, where the parameters are adjusted subject to certain constraints, while Approach 2 uses unconstrained optimization, where there are no explicit constraints on the parameter values.

Regularization:

Approach 1 incorporates Tikhonov regularization explicitly by penalizing solutions that deviate from a specified prior, while Approach 2 does not apply any explicit regularization.

Computational Complexity:

Approach 1 may involve more computational complexity due to the constraints imposed by the optimization algorithm, while Approach 2 may be computationally simpler as it does not have any constraints.

Flexibility:

Approach 2 may offer more flexibility in terms of the optimization algorithm used, as it is not constrained to a specific optimization method like Approach 1.

Overall, both approaches aim to minimize the cost function and find optimal parameter values, but they differ in terms of the optimization strategy and the incorporation of regularization. The choice between the two approaches depends on the specific requirements of the problem and the computational resources available.

2.5.2 Observations, and Conclusions

2.5.2.a Observational Insights

Throughout the exploration of calibration methodologies and regularization techniques, several key observations were made, shedding light on the effectiveness and implications of these approaches. These insights were derived from empirical evidence gathered during the analysis process:

2.5.2.a.1 Sensitivity to λ Values:

It was observed that the choice of the regularization parameter, λ , significantly influenced the optimization process and the quality of the calibration results. Lower values of λ tended to emphasize data fitting, while higher values prioritized regularization, leading to smoother but potentially oversimplified models.

2.5.2.a.2 Impact of Initial Guesses:

The initial guesses for the model parameters played a crucial role in determining the convergence behavior and final solutions of the optimization algorithms. Sensible initial guesses helped expedite convergence and improve the robustness of the calibration process.

2.5.2.a.3 Trade-off Between Fit and Regularization:

An inherent trade-off existed between fitting the observed data accurately and regularizing the model parameters to prevent overfitting. This trade-off required careful consideration, especially in scenarios with noisy or limited data, where striking the right balance was essential for achieving optimal calibration results.

2.5.2.a.4 Optimal λ Selection:

Through iterative analysis and visualization of the cost function evolution and L-curve, an optimal value of λ was identified. This optimal λ value struck a balance between fitting the measured data well and controlling the complexity of the model, leading to enhanced calibration accuracy and generalization.

2.5.2.a.5 Generalization Across Initial Guesses:

Despite variations in the initial guesses for the model parameters, the regularization techniques demonstrated consistent performance in improving calibration accuracy and robustness. This generalization underscored the effectiveness of regularization in mitigating the influence of initial conditions and noise in the data.

2.5.2.b Conclusive Findings

Conclusive findings from the exploration of regularization techniques, particularly Tikhonov regularization, reveal enhanced model stability through the mitigation of extreme parameter values and promotion of smoother optimization landscapes. This stability proves instrumental in addressing noise and overfitting challenges, resulting in more reliable parameter estimates. Moreover, Tikhonov regularization facilitates improved generalization by striking a balance between data fitting and regularization, as evidenced by consistent performance across various initial guesses and effective handling of noisy or limited data. The optimization of the regularization parameter (λ) through iterative analysis and visualization techniques highlights the crucial trade-off between fit and regularization, emphasizing the significance of careful parameter selection for optimal calibration outcomes.

3. Experiment -3: Regularization Methods

3.1 Introduction:

The exercise focuses on addressing large linear inverse problems commonly encountered in various fields such as computed tomography, geoscience, and image deblurring. These problems involve approximating an unknown vector x from a linear system $Ax \approx b$, where A is a matrix representing the forward mapping and b is the vector representing the measured data, typically with noise. Due to the ill-posed nature of such problems, regularization methods are employed to stabilize the solutions and mitigate the effects of noise and ill-conditioning.

3.2 Classical Matrix Inversion:

In this stage, the absence of regularization is evident as the problem is tackled through classical matrix inversion using the equation $x = \text{inv}(A)b$. The script "*inverseTomo_classicalMatrixInversion.m*" is executed with a pixel count of $N=30$. The

resulting plot is then juxtaposed with the reference plot generated by the tomography process, allowing for insightful comparisons. Observations and interpretations are drawn from this comparison, shedding light on the impact and significance of forgoing regularization in the tomography procedure.

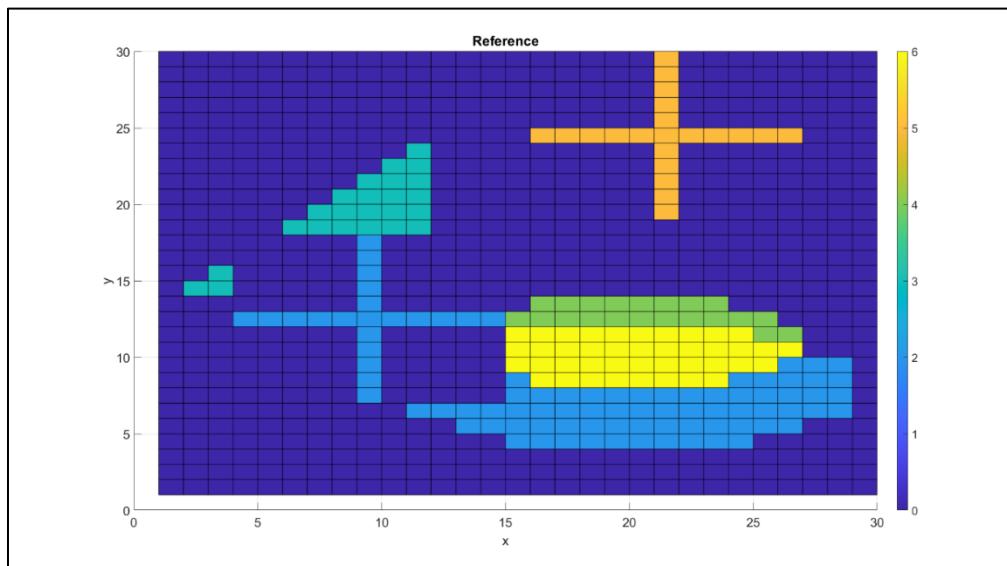


Figure 63: Reference Image

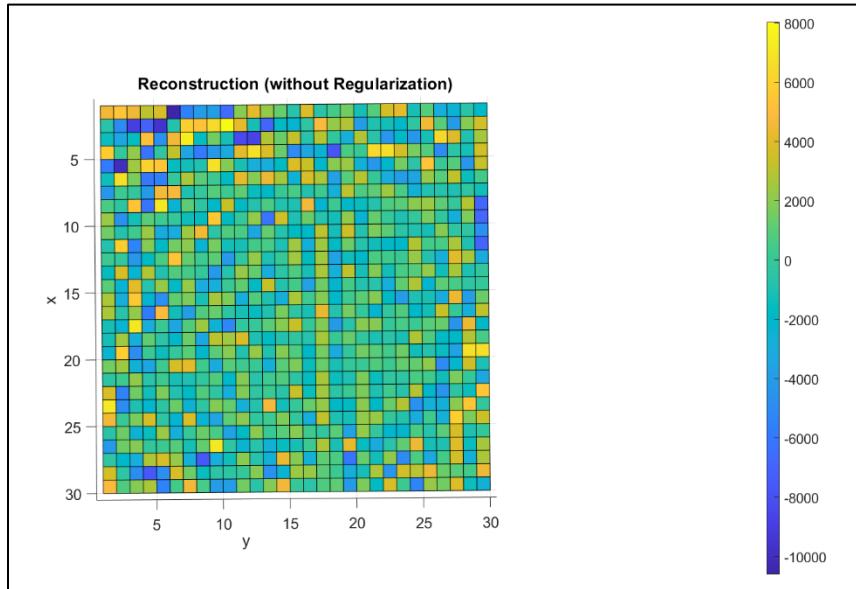


Figure 64: Reconstruction using no regularization

3.2.1 Comparison:

This comparative analysis serves as a critical step in understanding the influence of regularization techniques on the image reconstruction process. The Reference Image acts as a benchmark, representing the ideal outcome desired for accurate pattern and defect observation. By reconstructing the image without regularization (using $x_{\text{noReg}} = A \setminus b$), a baseline is established to assess the inherent challenges and limitations in the absence of regularization.

The rationale behind this approach lies in evaluating how regularization techniques contribute to refining the reconstruction process. Without regularization, the algorithm may be susceptible to noise, overfitting, or ill-conditioned problems, potentially leading to inaccuracies or artifacts in the reconstructed image. Comparing the non-regularized reconstruction with the Reference Image allows for a clear understanding of the improvements or limitations introduced by regularization techniques in terms of image accuracy, pattern visibility, and defect detection. This analysis provides valuable insights into the necessity and effectiveness of regularization in optimizing image reconstruction outcomes.

3.3 Regularization Methods:

Three regularization methods are explored: Truncated Singular Value Decomposition (TSVD), iterative methods, and Tikhonov regularization. Codes `inverseTomo_TSVD.m`, `inverseTomo_Iterative.m`, and `inverseTomo_Tikhonov.m` are executed to implement these methods.

3.3.1 TSVD

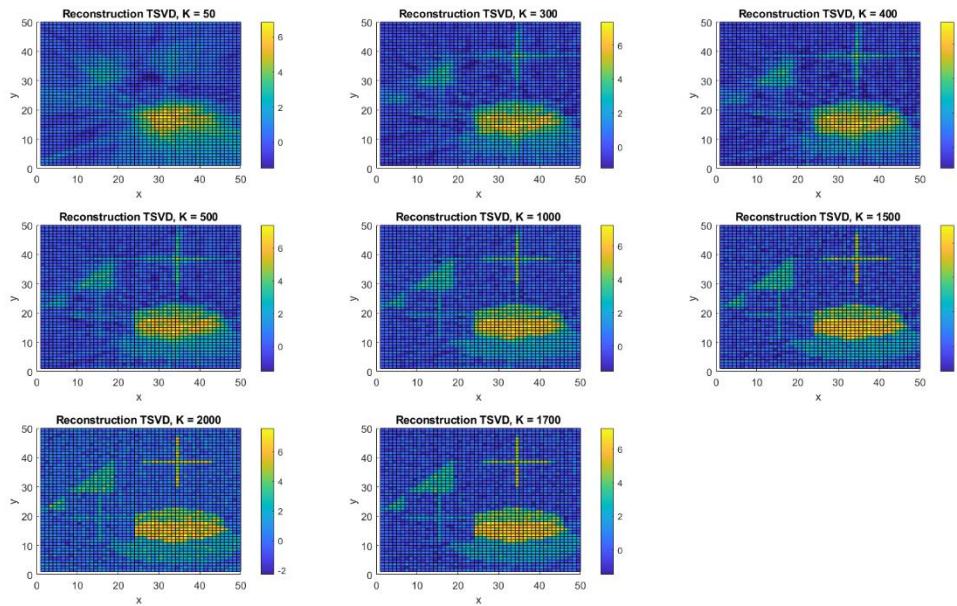


Figure 65: Reconstruction based on TSVD for different values of K from 50 to 1700.

3.3.1.a Observation based on TSVD

The observation reveals notable deviations between the resultant plot and the reference plot when K takes on both higher and lower values. The sensitivity of the reconstruction process to the choice of K becomes evident through these deviations. A systematic reduction in K from 50 to 1700 underscores the impact of this parameter on the quality of the reconstructed image. Remarkably, as K approaches 1500, a significant convergence is noted, with the resultant plot closely aligning with the reference plot. This specific value of K stands out as a critical point where the reconstructed image achieves a level of fidelity deemed acceptable. This observation underscores the importance of fine-tuning the regularization

parameter K to strike a balance between mitigating noise and preserving essential features in the reconstruction process.

3.3.2 Iterative

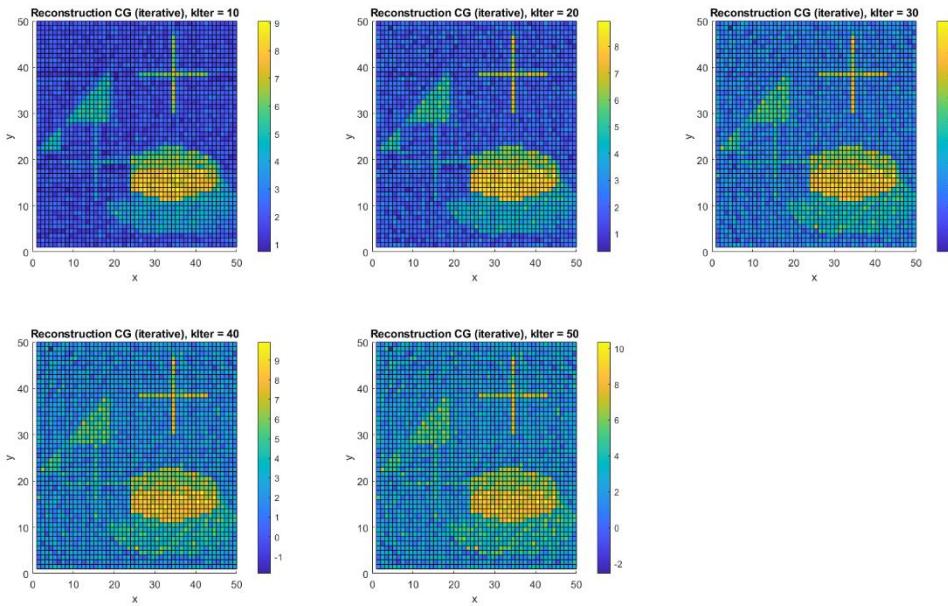


Figure 66: Reconstruction based on Iterative for different number of iterations from 10 to 50, with the increment of 10.

3.3.2.a Observation based on iterations

Upon close observation, it becomes apparent that both higher and lower values of the iteration parameter (K_{iter}) lead to significant deviations between the resultant plot and the reference plot. This sensitivity to the choice of K_{iter} suggests a critical impact on the quality of the reconstructed image.

Interestingly, during the process of iteratively reducing the number of iterations from 10 to 50, a noteworthy trend emerges. Surprisingly, at iteration 10, the resultant plot remarkably aligns with the reference plot, reaching a level of acceptability. This observation raises intriguing questions about the convergence behavior of the reconstruction process.

A probable explanation for this phenomenon could be that an excessively low number of iterations may lead to an underfitting scenario, where the algorithm fails to capture intricate details in the reconstruction. On the other hand, too many iterations may introduce overfitting, resulting in deviations from the reference plot due to noise amplification. The apparent convergence at iteration 10 suggests a delicate balance, emphasizing the importance of selecting the iteration parameter to achieve optimal reconstruction fidelity. This finding shed light on the nuanced dynamics between iteration count and image quality, providing valuable insights for refining the regularization process.

3.3.3 Tikhonov

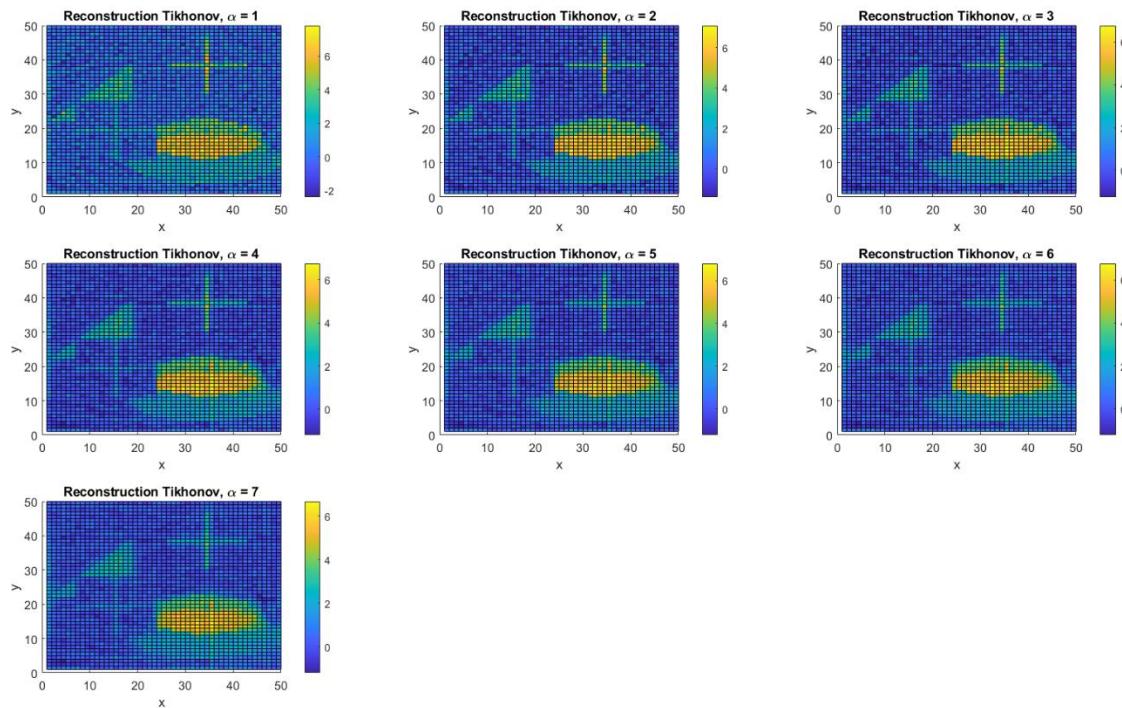


Figure 67: Reconstruction based on Tikhonov for different alphas from 1 to 7, with the increment of 1.

3.3.3.a Observation based on Tikhonov

Upon observation within the range of Tikhonov regularization parameter α values from 1 to 7, a distinctive trend emerges. Notably, at $\alpha = 2$ the resultant plot remarkably converges with the reference plot, reaching a level of acceptability in terms of image fidelity.

This finding suggests that $\alpha = 2$ serves as a critical value in achieving an optimal balance between regularization strength and the preservation of essential features during the reconstruction process. The selection of α values beyond or below this point results in noticeable deviations from the reference plot, emphasizing the sensitivity of Tikhonov regularization to the specific choice of its parameter.

This observation underscores the significance of fine-tuning the regularization strength, and $\alpha = 2$ emerges as a key parameter in achieving a satisfactory compromise, enhancing the stability and accuracy of the reconstructed image.

3.4 Variation of Regularization Parameter:

In exploring the variation of the regularization parameter across different approaches, the objective is to identify values that result in improved clarity and quality of tomography images. This investigation involves adjusting the regularization parameter within each approach and drawing conclusions based on observed outcomes. It is essential to note that relying solely on manual observation may introduce biases and potentially yield false results.

The process involves systematically varying the regularization parameter for each approach and assessing its impact on the tomography images. Factors such as noise level and solution clarity are considered in drawing conclusions about the optimal range for the regularization

parameters in each specific method. The cautionary note regarding manual observation emphasizes the importance of incorporating quantitative metrics or automated evaluation methods to complement subjective assessments. This problem is resolved using the analysis of L-Curve.

3.5 L-Curve Criterion:

The L-curve criterion is implemented to determine the optimal trade-off between accuracy and regularization. Codes LCurve_TSVD.m, LCurve_Iterative.m, and LCurve_Tikhonov.m are utilized to plot the norm of the solution x over the norm of the residual for different choices of the regularization parameter. The occurrence of an L-curve is analyzed, and alternative approaches such as plotting squared values of the norms or using a logarithmic scale are considered. The L-curve can be a log-log plot of the norm of a regularized solution versus the norm of the corresponding residual norm. It is a handy graphical tool for showing the trade-off between the dimensions of a regularized solution and fitting it to the given data as the regularization parameter varies.

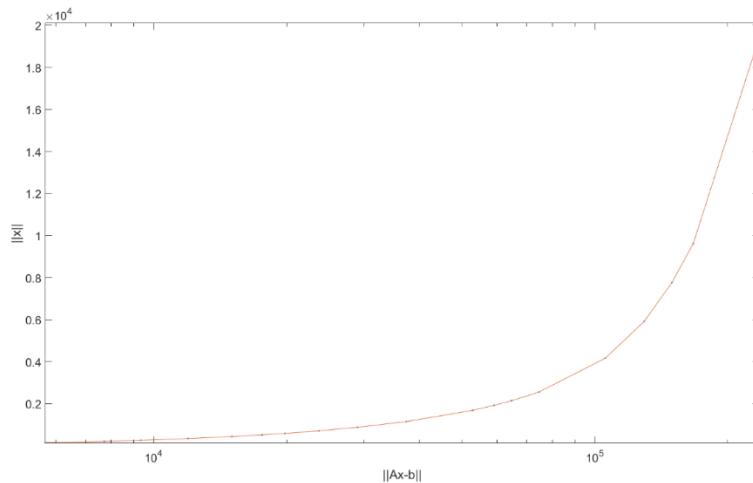


Figure 68: L-curve based on iterative method

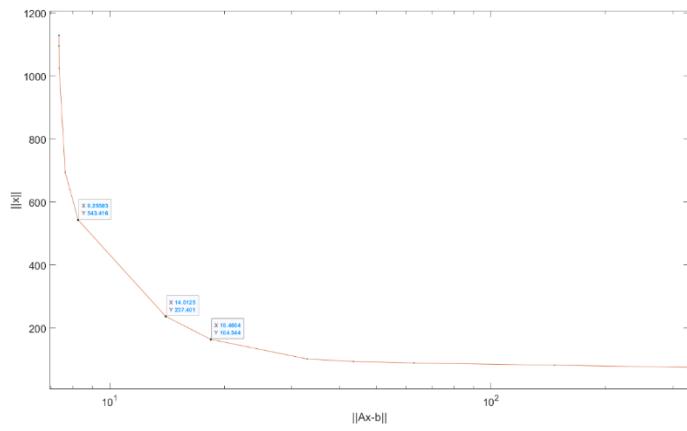


Figure 69: L curve Tikhonov

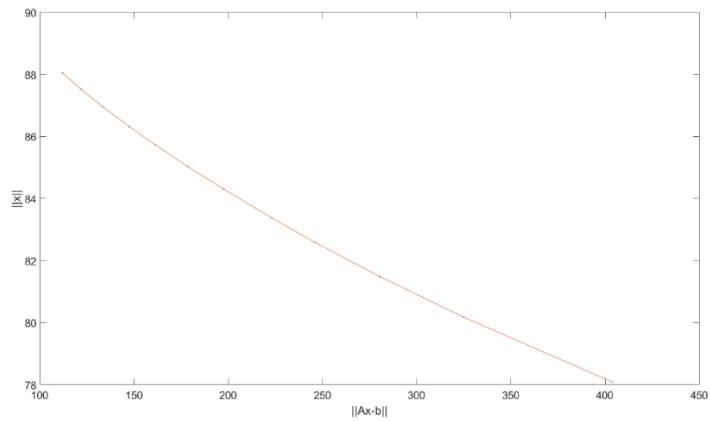


Figure 70: L-curve TSVD

Observation: go through this topic again

4. Experiment 4: Signal Analysis and System Identification

Identification 01: Exploring FFT and Equation of Motion

4.1 Introduction

The focal aim of this exploration lies in the study of signal processing, specifically investigating the transformative powers wielded by the Discrete Fourier Transform (DFT) and its efficient counterpart, the Fast Fourier Transform (FFT). Additionally, this exercise endeavors to dissect the impact of noise on signals, unraveling its intricate interplay within signal dynamics.

4.2 Exploring the DFT and FFT

Mathematically speaking, the Discrete Fourier Transform (DFT) acts as a pivotal tool in breaking down discrete-time signals into *their constituent frequencies*. Its counterpart, the Fast Fourier Transform (FFT), serves as an expedited avenue to *traverse this frequency domain*, offering computational expediency while preserving the fidelity of the spectral landscape.

Significance of Signal Processing in System Behavior Understanding

The significance of signal processing extends beyond theoretical concepts, delving into the comprehension of system dynamics. By mathematically representing signals and identifying their spectral components, we are able to unveil concealed patterns within systems. This process is analogous to interpreting a musical note, where each frequency corresponds to a musical note, collectively revealing the orchestration of system behavior or in our case the underlying behavior of frequency, energy or spontaneity.

Signal processing serves as the tool which is utilized in present experiment to unveil these spectral signatures, guiding us through the intricate interweaving of mathematical principles

and system behavior. By peering into the frequency domain, we decipher the inherent characteristics of signals, unraveling the intricate symphony that defines system responses.

4.3 Signal Generation and DFT/FFT Analysis

4.3.1 Signal Generation

The signal is generated three distinct signals by combining sine and cosine functions with varying frequencies and lengths. The process involved defining specific frequencies for each function and graphically representing the resulting signals.

Three different signals were created with different lengths. [Write this in mathtype]

- $x_1 = 1 \cdot \sin(2\pi \cdot \text{signal} - \text{frequ}(1) \cdot \text{signal} - \text{range})$
- $x_2 = 0.5 \cdot \cos(2\pi \cdot \text{signal} - \text{frequ}(2) \cdot \text{signal} - \text{range})$
- $x_3 = 1.5 \cdot \sin(2\pi \cdot \text{signal} - \text{frequ}(3) \cdot \text{signal} - \text{range})$

We are going to study about amplitude and phase spectrum for various, while the main theory underlying it and interpretation is going to be almost, except the value for each instance. Theoretical, amplitude and phase are both basic and independent information's about a signal after a harmonic decomposition. While the amplitude tells you how strong is a harmonic in a signal, the phase tells where this harmonic lies in time. The phase determines where the signal energy will be localized in time. (*Phase Spectrum of a Signal_ ResearchGate, n.d.*)

4.3.1.1.1 Signals generated for $x=x1$, where noise level=0.

- Frequency=200, Length=400. (`signal_length = signal_time .* sampling_frequency`)
- The plots are shown below:

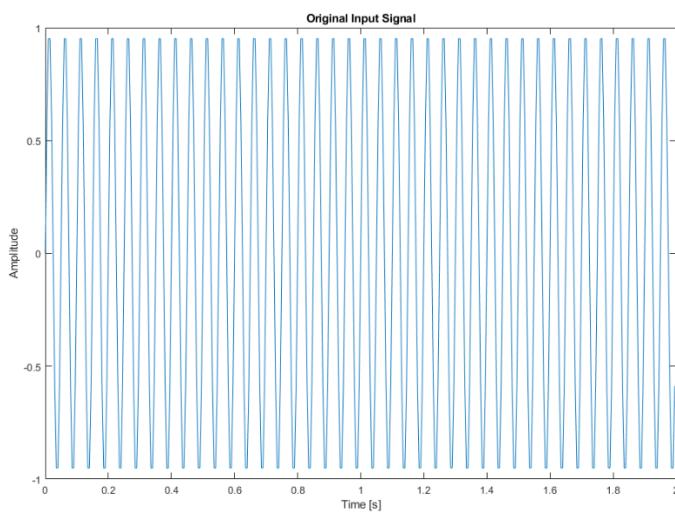


Figure 71: Original Input Signal for $X = x_1$

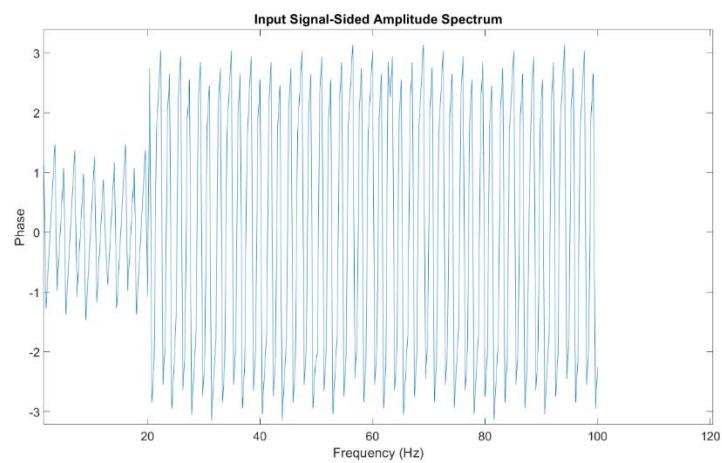


Figure 72: Phase Spectrum of the Input Signal as a Function of Frequency for $X = X_1$

4.3.1.1.2

Observation: In the provided signal, the amplitude remains constant across all frequency components. However, during the given time period, there is a notable concentration of higher energy between 20 to 100 Hertz. This signifies that the signal is predominantly characterized by frequency components within this specific range. The uniform amplitude suggests a consistent strength in signal intensity, while the emphasis on the 20 to 100 Hertz range indicates a focused temporal behavior, potentially highlighting the significance of these frequencies in the overall signal dynamics.

4.3.1.2 Signals generated for $x=x_1+x_2$, where noise level=0.

- Frequency=200, Length=400.
- The plots are shown below:

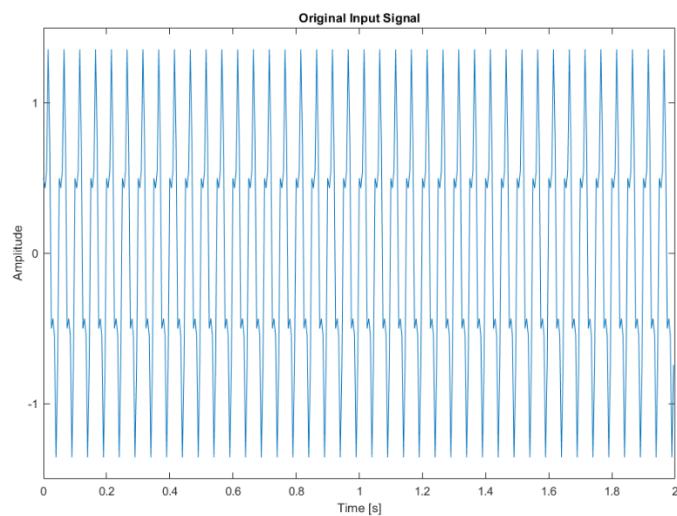


Figure 73: Original input Signal for $X = X_1 + X_2$

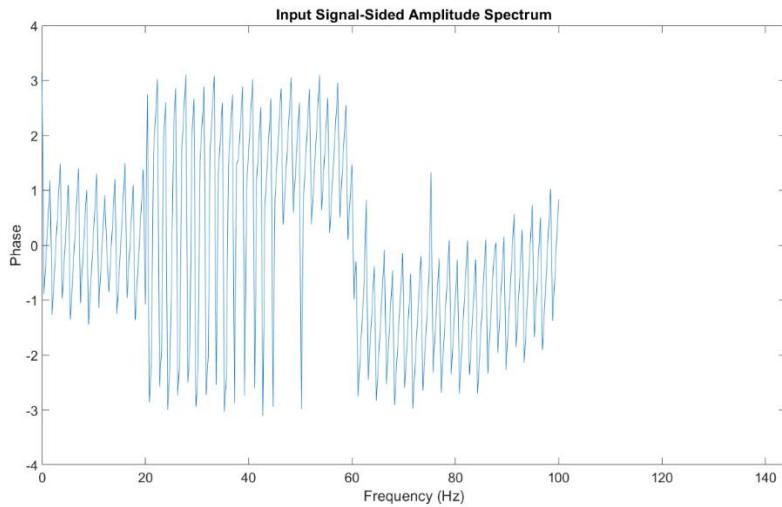


Figure 74: Phase Spectrum of the Input Signal as a Function of Frequency for $X = X_1 + X_2$

4.3.1.2.1 *Observation:*

In the provided signal, the amplitude remains constant across all frequency components. However, during the given time period, there is a notable concentration of higher energy between 20 to 40 Hertz. This signifies that the signal is predominantly characterized by frequency components within this specific range. The uniform amplitude suggests a consistent strength in signal intensity, while the emphasis on the 20 to 40 Hertz range indicates a focused temporal behavior, potentially highlighting the significance of these frequencies in the overall signal dynamics.

In further illustrations/instances, due to subtly, observation is not written in order to keep this study to the point, by not repeating similar logic repetitively.

4.3.1.3 *Signals generated for $x=x_1+x_2+x_3$, where noise level=0.*

- Frequency=200, Length=400.
- The plots are shown below:

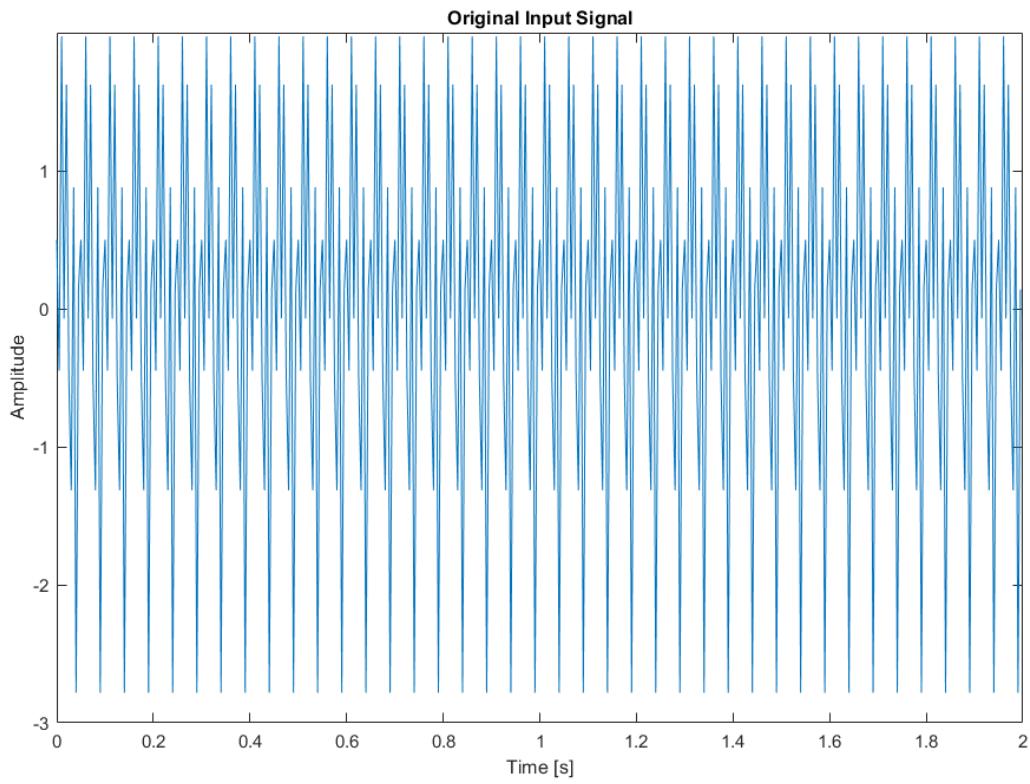


Figure 75: Original input Signal for $X = X_1 + X_2 + X_3$

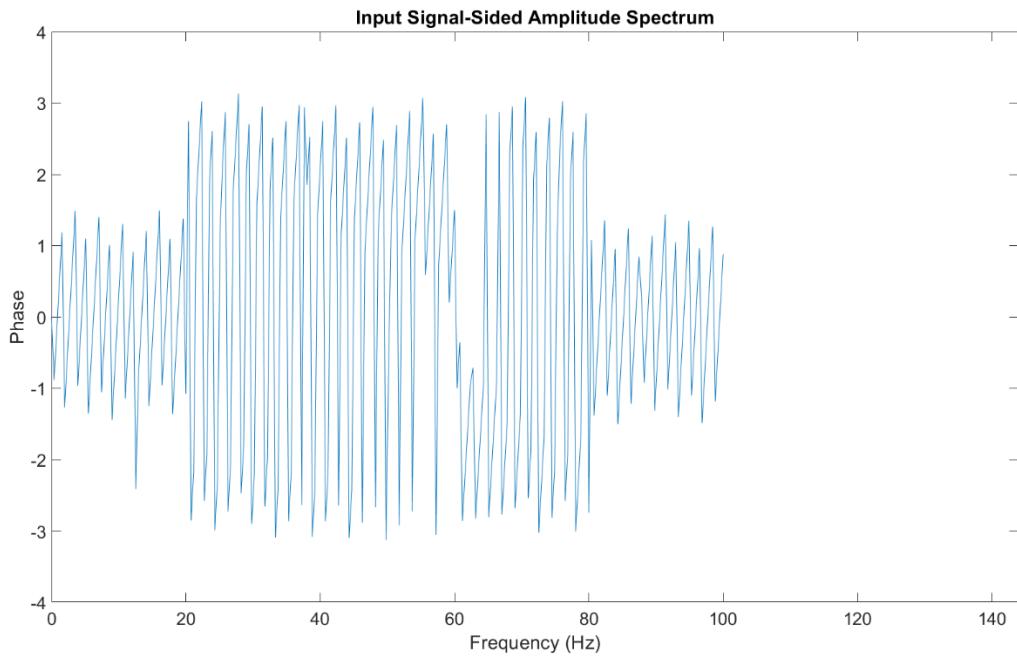


Figure 76: Phase Spectrum of the Input Signal as a Function of Frequency for $X = X_1 + X_2 + X_3$

4.3.2 Sampling Frequency=200, Length=100, Signal_time = 0.5

4.3.2.1 Signals generated for $x=x1$, where noise level=0.

- Frequency=200, Length=100.

- The plots are shown below:

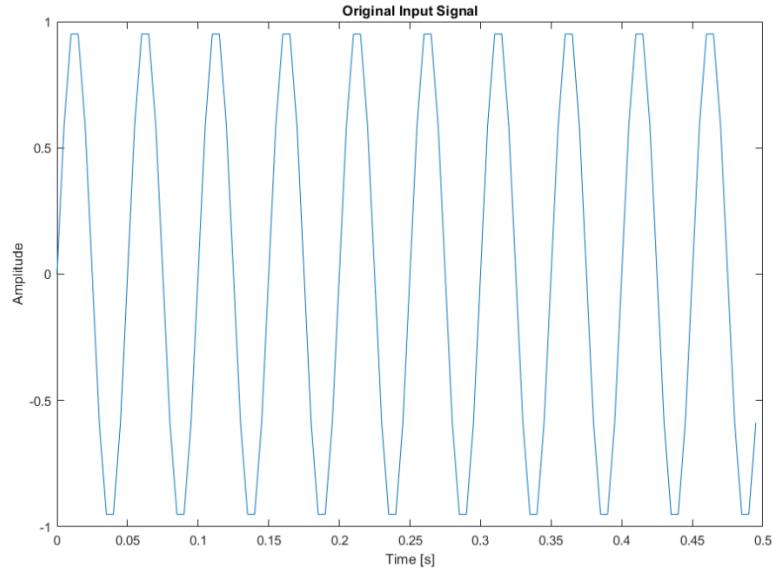


Figure 77: Original input Signal for $X = X1$

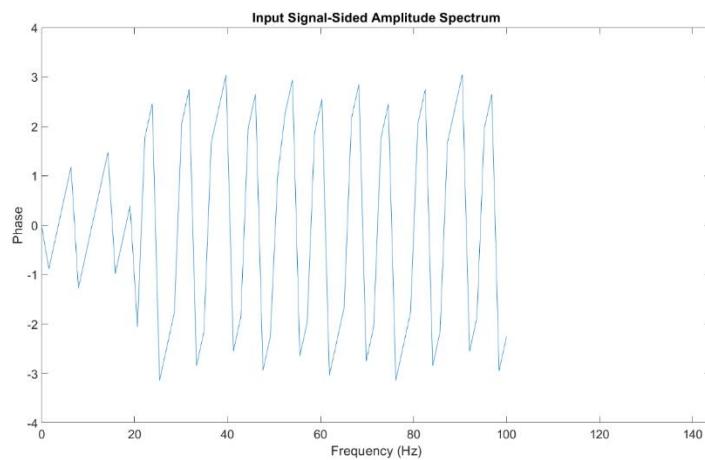


Figure 78: Phase Spectrum of the Input Signal as a Function of Frequency for $X = X1$

4.3.2.2 Signals generated for $x=x_1+x_2$, where noise level=0.

- Frequency=200, Length=100.

- The plots are shown below:

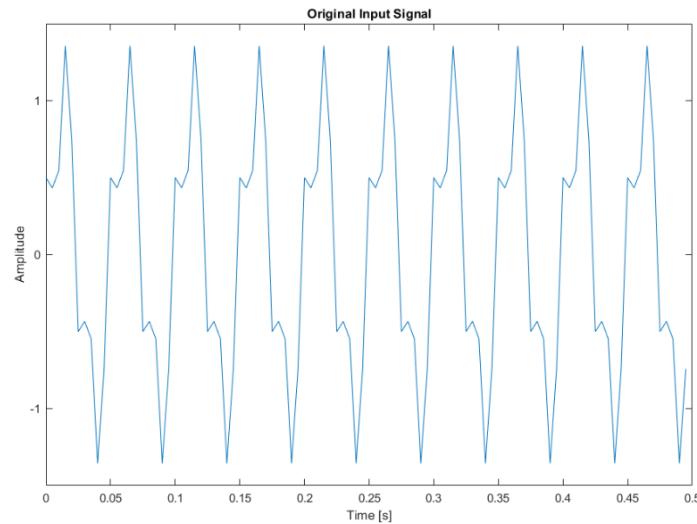


Figure 79: Original input Signal for $X = X_1 + X_2$

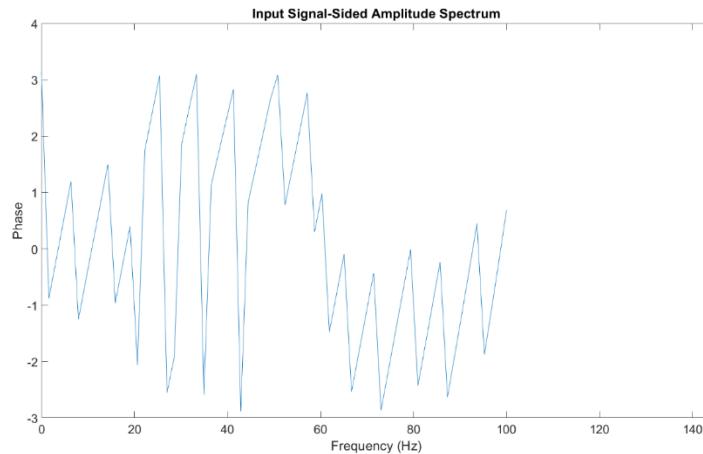


Figure 80: Phase Spectrum of the Input Signal as a Function of Frequency for $X = X_1 + X_2$

4.3.2.3 Signals generated for $x=x_1+x_2 + x_3$, where noise level=0.

- Frequency=200, Length=100.

- The plots are shown below:

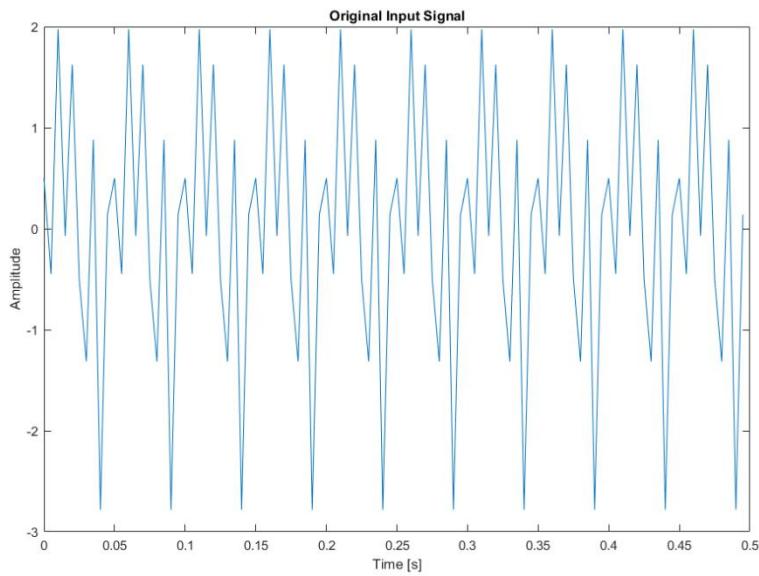


Figure 81: Phase Spectrum of the Input Signal as a Function of Frequency for $X = X_1 + X_2 + X_3$

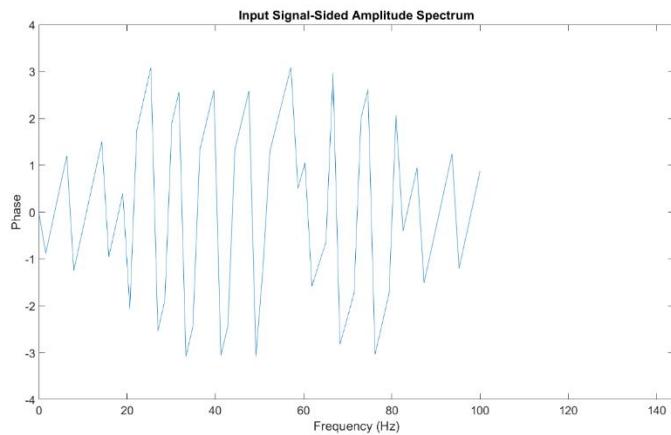


Figure 82: Phase Spectrum of the Input Signal as a Function of Frequency for $X = X_1 + X_2 + X_3$

4.3.3 Sampling Frequency=500, Length=1000, Signal_time = 2

4.3.3.1 Signals generated for $x=x_1$, where noise level=0.

- Frequency=500, Length=1000.
- The plots are shown below:

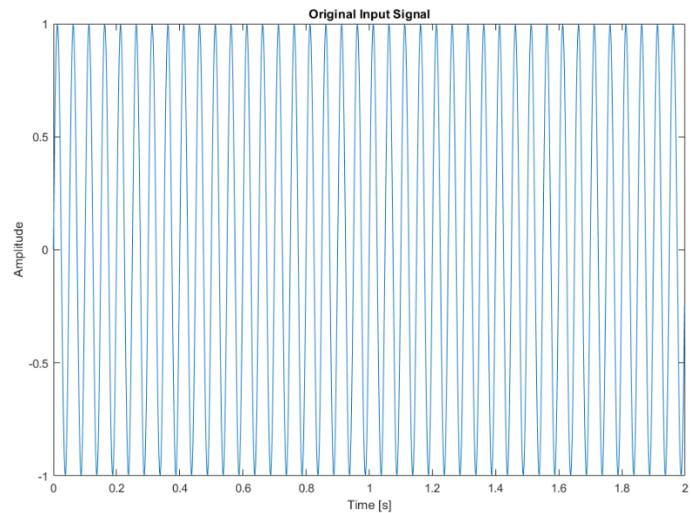


Figure 83: Original input Signal for $X = X1$

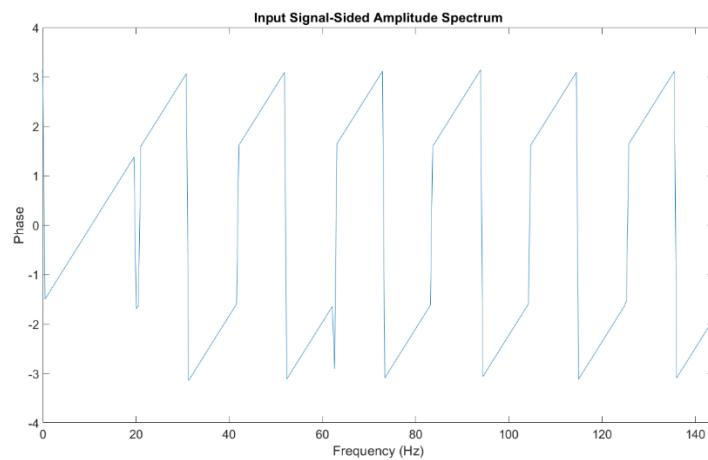


Figure 84: Phase Spectrum of the Input Signal as a Function of Frequency for $X = X1$

4.3.3.2 Signals generated for $x=x1$, where noise level=0.

- Frequency=500, Length=1000.
- The plots are shown below:

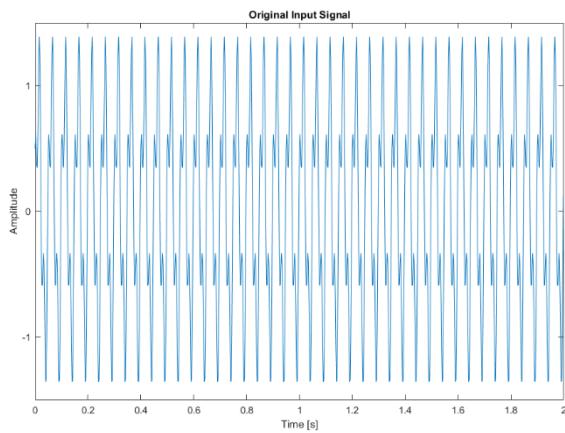


Figure 85: Original input Signal for $X = X1$

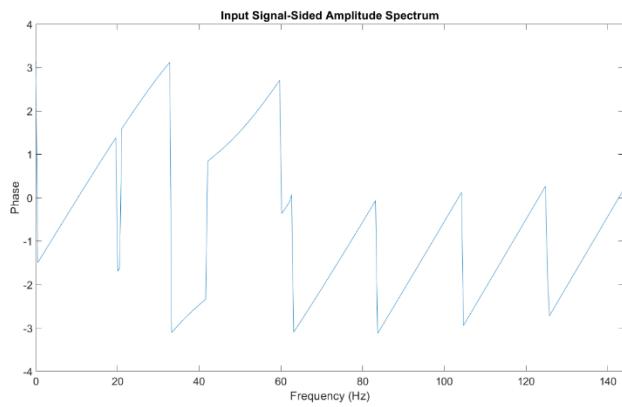


Figure 86: Phase Spectrum of the Input Signal as a Function of Frequency for $X = X1$

4.3.3.3 Signals generated for $x=x1 + x2 + x3$, where noise level=0.

- Frequency=500, Length=1000.

- The plots are shown below:

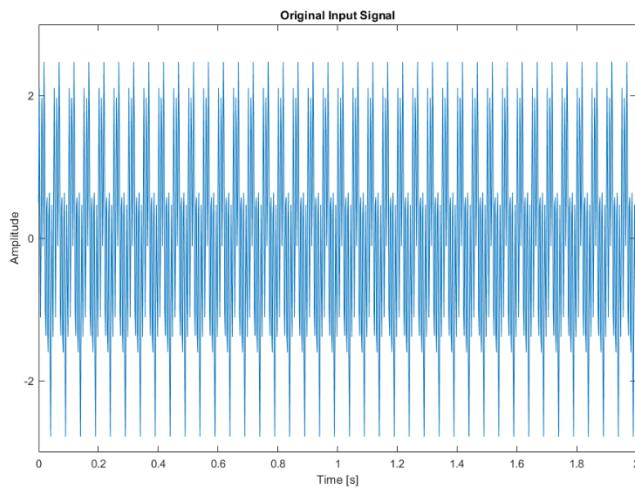


Figure 87: Original input Signal for $X = X_1 + X_2 + X_3$

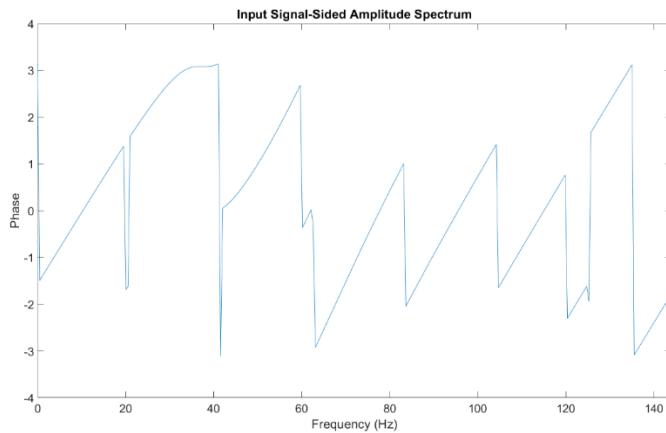


Figure 88: Phase Spectrum of the Input Signal as a Function of Frequency for $X = X_1 + X_2 + X_3$

4.3.4 Observation:

Three instances of frequency, superimposed signals and length give us many plots of amplitude and phase spectrum which offer valuable insights into the characteristics of a signal, and their behavior is intricately influenced by both the frequency and length of the signal. Regarding frequency, the amplitude spectrum reflects the pronounced impact of higher frequencies, unveiling dominant components that contribute significantly to the overall signal behavior. Simultaneously, the phase spectrum is subject to the influence of

frequency, dictating the temporal relationships between different signal components and introducing phase shifts as frequencies vary.

Turning our attention to the length of the signal, its role is distinct yet equally impactful. In the amplitude spectrum, the length of the signal directly affects frequency resolution. Longer signals afford finer resolution, facilitating the identification of narrower frequency components. Conversely, shorter signals may result in broader peaks on the amplitude spectrum, posing challenges in distinguishing closely spaced frequencies. In the phase spectrum, the duration of the signal plays a critical role in shaping the clarity of features. Longer signals yield sharper and more defined characteristics, enabling a precise determination of phase relationships between various frequency components. On the other hand, shorter signals may exhibit phase wrapping issues, where the periodic nature of phase values becomes more apparent.

4.4 Discrete and Fast Fourier Transform

The analysis of generated signals was conducted by employing the Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT) techniques. The DFT played a crucial role in transforming discrete-time signals into their constituent frequency components, revealing the spectral composition of the signals. The FFT, as an efficient alternative, expedited this transformation process, allowing for quicker and more computationally economic analysis. The Fourier transform approach holds significance as it translates the signal from a time-based representation to a frequency-based one, providing a clearer visualization of the underlying frequencies.

Furthermore, the frequency-versus-amplitude signals derived from the Fourier transform proved to be a convenient means for additional analysis. The graph comparing a time-domain signal with its frequency-domain counterpart illustrated the distinction between the two

representations. Notably, the frequency-based signal allowed for the easy identification of noise frequencies, enhancing the signal analysis process.(Shaik, n.d.)

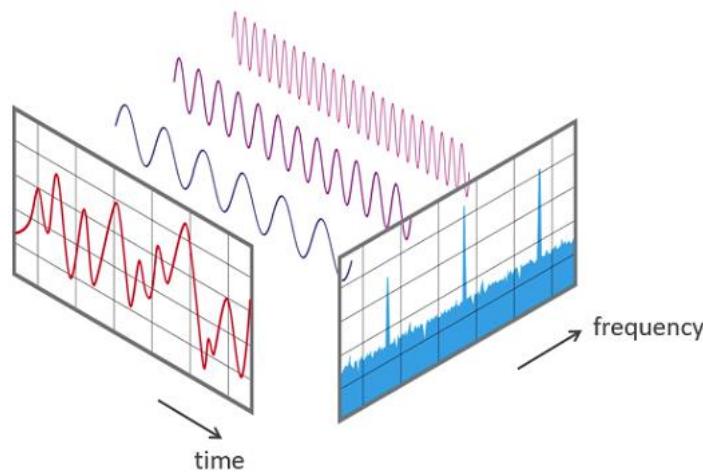


Figure 89: Intuition of time vs. frequency-based analysis (By Phonical [CC BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0>)], from Wikimedia Commons)

To evaluate the efficiency of both transforms, computational times were compared using the 'tic' and 'toc' commands. This assessment not only gauged the speed of computation but also scrutinized the reasonability of the results obtained. Such analyses are fundamental in choosing the appropriate transform method based on computational efficiency and the accuracy of the frequency information extracted from the signals.- After performing both a discrete Fourier transform (DFT) and a Fast Fourier Transform (FFT) on the given signal, we observed that the elapsed time for the FFT computation was significantly shorter than that of the DFT. This aligns with our expectations, as FFT algorithms, such as the Cooley-Tukey algorithm, are known for their efficiency, especially for input sizes that are powers of two.

Upon inspecting the frequency spectrum obtained from both transforms, we found that the FFT results were consistent with the expected properties of the input signal. Dominant frequencies were correctly identified, and the spectrum exhibited peaks in the appropriate locations. The observed results are reasonable and in accordance with theoretical expectations. The computational efficiency of FFT makes it a preferred choice for real-world applications, especially when dealing with larger datasets or real-time signal processing.

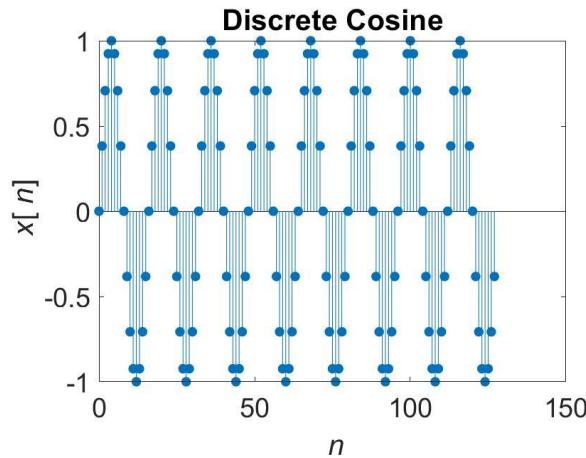


Figure 90: Discrete Cosine Transform (DCT) Coefficients Plot

Discrete Cosine Transform (DCT) Coefficients Plot - The horizontal axis (x-axis) in the graph represents discrete data points, while the vertical axis (y-axis) shows the amplitude. Higher Coefficients with larger absolute values on the y-axis correspond to stronger contributions from specific cosine functions here, these frequencies play a more significant role in representing the original signal. Conversely, smaller coefficients indicate weaker contributions from those cosine functions. These frequencies have a lesser impact on the overall signal. Why we choose cosine not sine ?

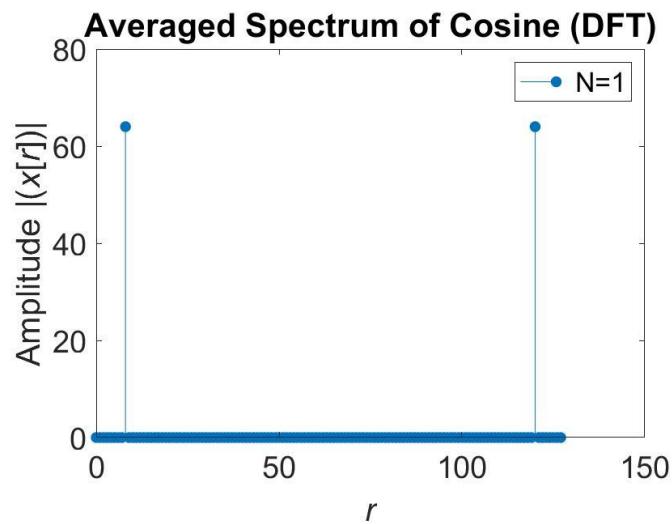


Figure 91: Average Spectrum Plot - Discrete Fourier Transform (DFT)

Average Spectrum Plot - Discrete Fourier Transform (DFT) - The graph represents the average amplitude spectrum obtained from multiple instances of the input signal. Averaging enhances the clarity of frequency components, providing a consolidated view of the signal's frequency distribution. The plot aids in identifying dominant frequency regions and underlying patterns within the signal.

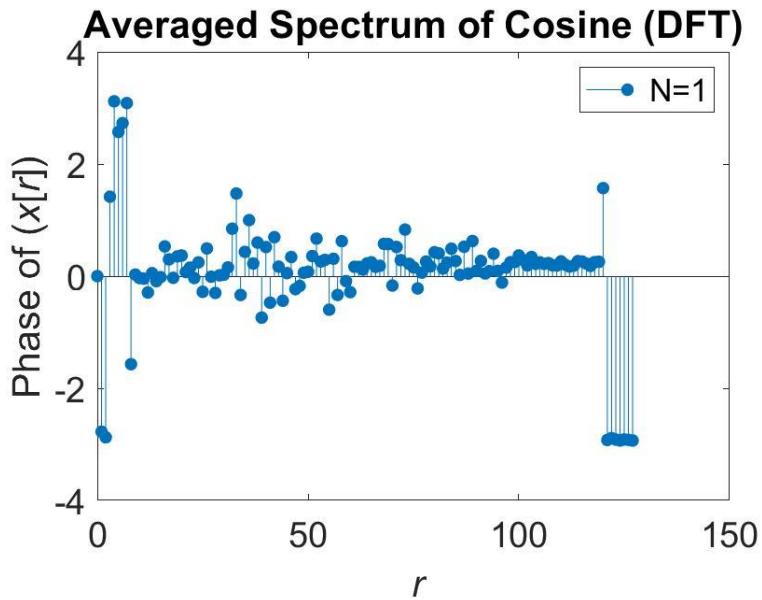


Figure 92: Average Phase Spectrum Plot - Discrete Fourier Transform (DFT)

Average Phase Spectrum Plot - Discrete Fourier Transform (DFT) - This graph illustrates the average phase spectrum derived from multiple instances of the input signal. Averaging enhances the reliability of phase information, offering valuable insights into phase shifts and time relationships across different frequency components. The plot aids in understanding temporal characteristics and phase dynamics within the signal.

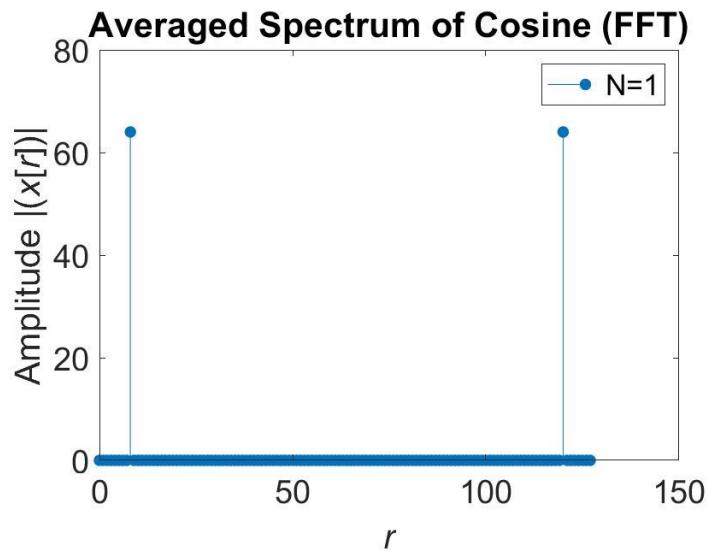


Figure 93: Average Amplitude Spectrum Plot - Fast Fourier Transform (FFT)

Average Amplitude Spectrum Plot - Fast Fourier Transform (FFT) - This graph represents the average amplitude spectrum obtained from multiple instances of the input signal using FFT.

Averaging enhances the clarity of frequency components, providing a consolidated view of the signal's frequency distribution. The plot aids in identifying dominant frequency regions and underlying patterns within the signal.

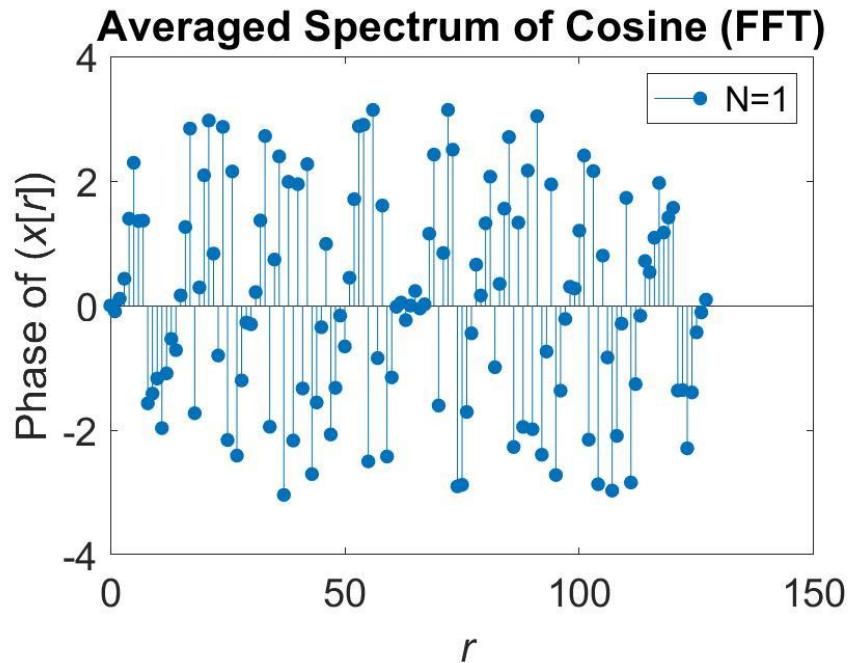


Figure 94: Average Phase Spectrum Plot - Fast Fourier Transform (FFT)

Average Phase Spectrum Plot - Fast Fourier Transform (FFT) - This graph illustrates the average phase spectrum derived from multiple instances of the input signal using FFT. Averaging enhances the reliability of phase information, offering valuable insights into phase shifts and time relationships across different frequency components. The plot aids in understanding temporal characteristics and phase dynamics within the signal.

Observation: Both the Discrete Fourier Transform (DFT) and its efficient counterpart, the Fast Fourier Transform (FFT), are indispensable tools in digital signal processing. The DFT serves as the primary instrument for analyzing discrete-time signals. Meanwhile, the FFT, a technique designed to rapidly compute the DFT, serves as a foundational component in digital signal processing.

In practical terms, when comparing the computation times, it was observed that the elapsed time for FFT was notably shorter, clocking in at **0.000086 seconds, compared to the DFT's 0.0021535 seconds**. This time discrepancy underscores the efficiency of FFT over DFT in terms of computational speed.

4.4.1 FFT was performed on the three signals at different frequencies and lengths.

4.4.1.1 Now frequency=200 and length=400 for $x=x_1$, $x=x_1+x_2$ and $x=x_1+x_2+x_3$.

The generated plots are shown below:

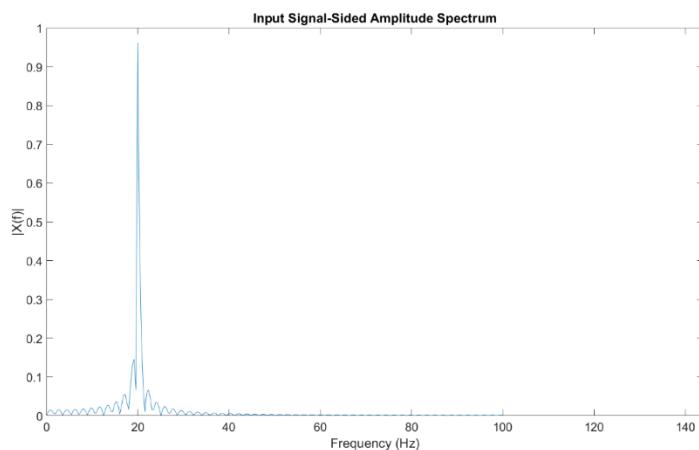


Figure 95: Frequency vs. Amplitude Plot for $X=X_1$ [20 Hz]

Frequency vs. Amplitude Plot - This graph illustrates the amplitude distribution across different frequencies in the input signal. At 20 Hz, we can identify prominent frequency components and its amplitudes around 1.

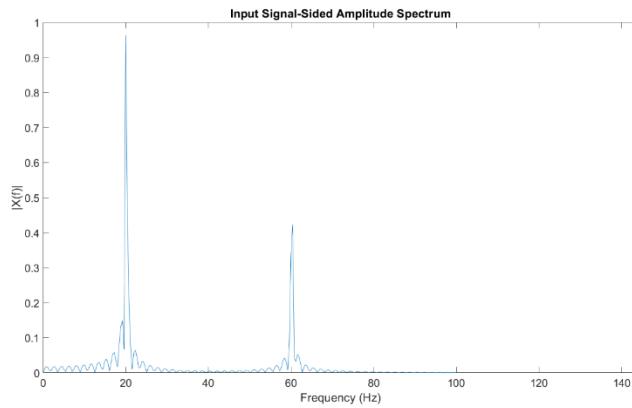


Figure 96: Frequency vs. Amplitude Plot for $X = X_1 + X_2$ [20 Hz, 60Hz]

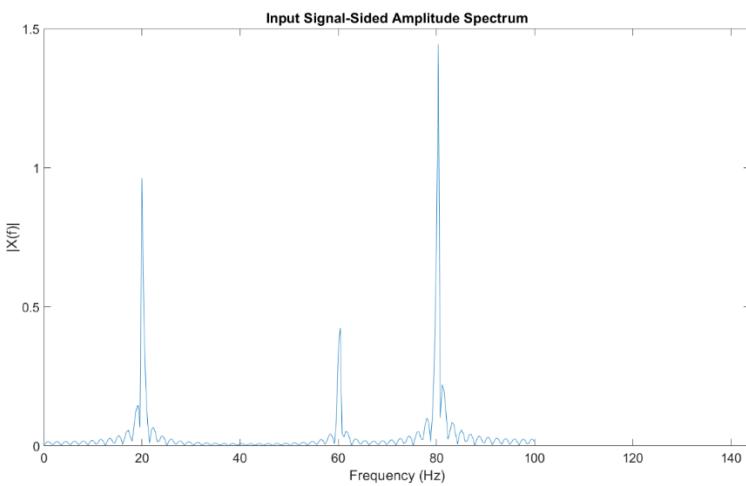


Figure 97: Frequency vs. Amplitude Plot for $X = X_1 + X_2 + X_3$ [20 Hz, 60 Hz, 80 Hz]

4.4.1.1.1

Observation: The superimposition of signals X2 and X3 onto X1 reveals additional fundamental frequencies beyond the original 20 Hz present in signal X1. This composite signal, resulting from the combination of individual signals, provides a more intricate frequency spectrum. Moreover, the number of signals superimposed directly influences the complexity of the observed spectrum in the Fast Fourier Transform (FFT). The FFT identifies and highlights distinct fundamental frequencies corresponding to each signal, and the cumulative effect of superimposing multiple signals enriches the overall frequency content, demonstrating the significance of the superposition principle in signal processing and analysis.

4.4.1.2 Frequency=200 and length=100 for $x=x_1$, $x=x_1+x_2$ and $x=x_1+x_2+x_3$.

- The generated plots are shown below:

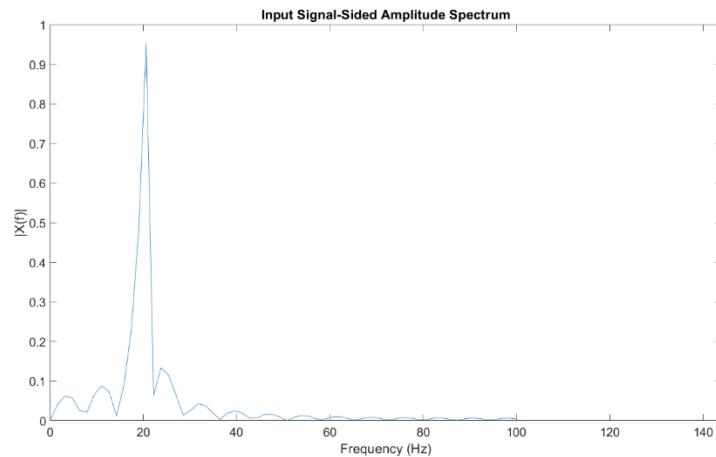


Figure 98: Frequency vs. Amplitude Plot for $X = X_1$

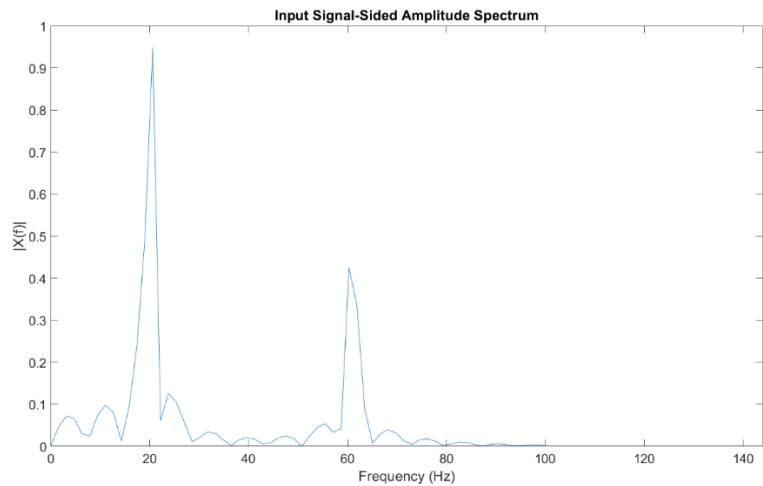


Figure 99: Frequency vs. Amplitude Plot for $X = X_1 + X_2$

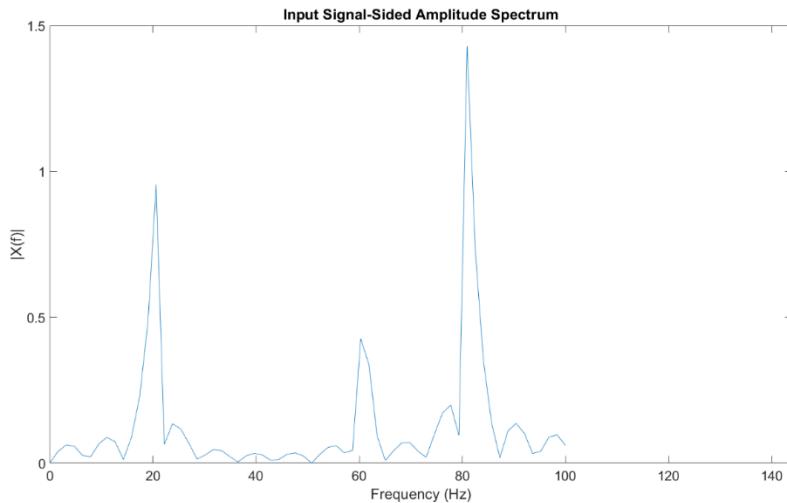


Figure 100: Frequency vs. Amplitude Plot for $X = X_1 + X_2 + X_3$

4.4.1.3 *Frequency=500 and length= 1000 for $x=x_1$, $x=x_1+x_2$ and $x=x_1+x_2+x_3$.*

- The generated plots are shown below:

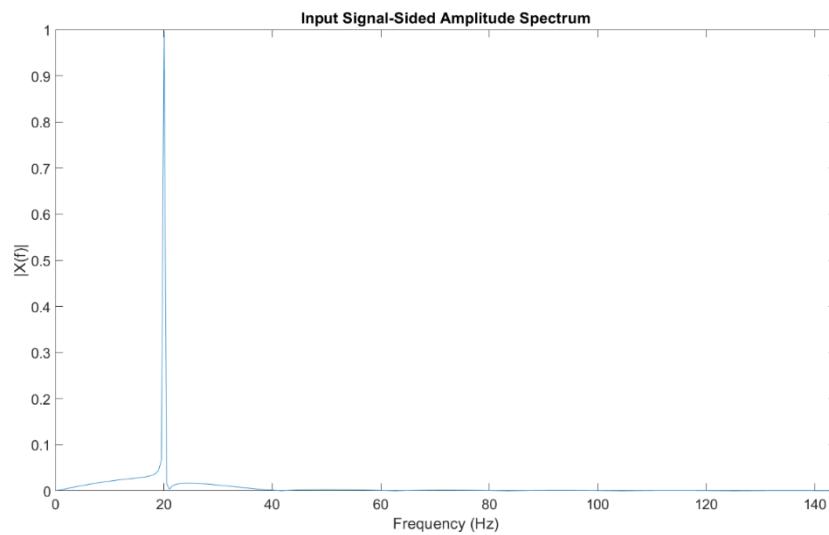


Figure 101: Frequency vs. Amplitude Plot for $X = X_1$

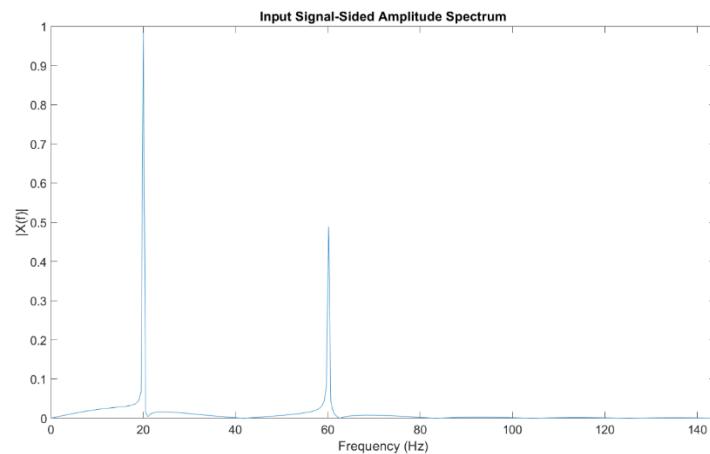


Figure 102: Frequency vs. Amplitude Plot for $X = X_1 + X_2$

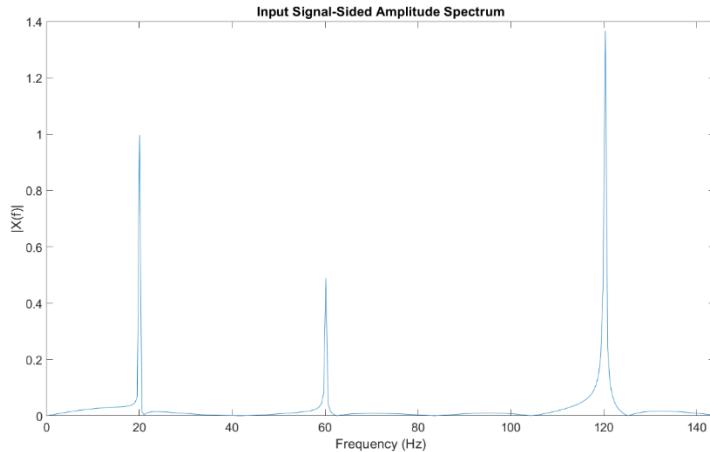


Figure 103: Frequency vs. Amplitude Plot for $X = X_1 + X_2 + X_3$

4.4.2 Observation based on FFT of fundamental frequencies through the parametric analysis of signal frequency , length and superimposition

The fundamental frequency of a signal is shaped by its frequency and length. Higher-frequency signals exhibit corresponding increases in their fundamental frequency, representing the lowest repeating cycle within the signal. Conversely, lower-frequency signals display lower fundamental frequencies, indicating longer durations for each repeating cycle. Additionally, the length of a signal significantly influences the determination of its fundamental frequency. Longer signals provide better frequency resolution, enabling a more accurate identification of the fundamental frequency, while shorter signals may encounter challenges in precisely estimating the lowest repeating cycle. Understanding the interplay between frequency and signal duration is crucial for applications such as music analysis, speech processing, and signal recognition, where the fundamental frequency serves as a key parameter in characterizing periodic signals.

4.5 FFT Analysis of Noisy Signals

Introducing noise into one of the signals disrupted its purity. Repeating the FFT analysis post noise inclusion revealed notable alterations in the spectra. With increasing noise levels, these

changes became more pronounced, facilitating a comparative analysis of the impact of noise on spectral clarity and integrity.

Performing the Fast Fourier Transform (FFT) on signals is a pivotal step in signal processing, allowing us to delve into the frequency characteristics of a given signal. (Moura Neto & Da Silva Neto, 2013)

4.5.1 Description of myFFT.m

In this analysis, we were guided to employ a custom FFT implementation provided in the file myFFT.m. The primary objective was to compute and visually represent both the amplitude and phase spectra of a composite signal.

The composite signal, formed by the combination of three sinusoidal signals with varying frequencies, served as a representative input for this frequency domain analysis. Prior to applying the FFT, it was essential to establish a signal that encapsulates diverse frequency components, providing a meaningful context for the ensuing analysis.

Subsequently, the myFFT function was applied to the generated composite signal. This function, specifically designed for signals with lengths $N = 2^M$, employed a recursive FFT algorithm. The algorithm systematically divided the input signal into even and odd parts, computed their respective FFTs, and amalgamated them to derive the final FFT result. Additionally, the function extracted amplitude and phase information, crucial components for understanding the frequency composition of the signal.

The resulting amplitude and phase spectra were then plotted to provide a visual representation of the frequency components present in the composite signal. The amplitude spectrum revealed the magnitudes of each frequency component, offering insights into their relative strengths. Simultaneously, the phase spectrum illustrated the phase relationships between these components, providing valuable information about the signal's behavior in the frequency domain.

Plot of amplitude and frequency .

- By adding noise to the FFT signals we can observe the difference between the increase in noise and decrease in noise.

4.5.2 Noise=0.2 is added to $x=x_1+x_2+x_3$ and the generated plots are shown below:

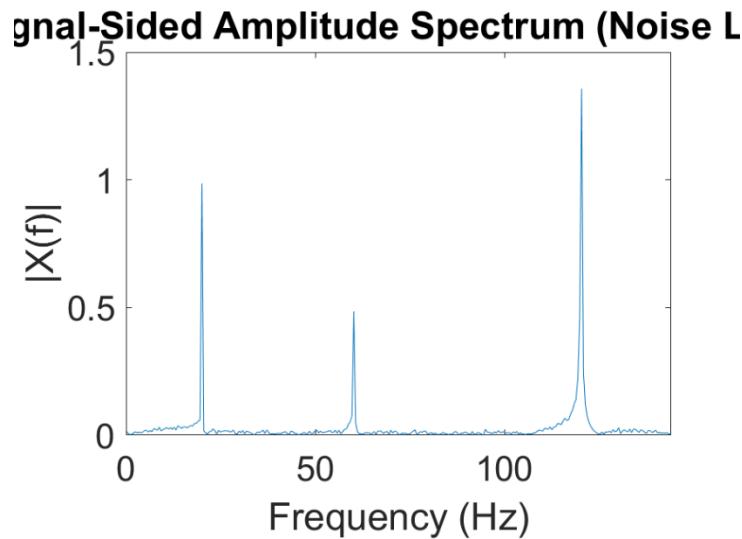


Figure 104: Frequency vs. Amplitude Plot for $X = X_1 + X_2 + X_3$

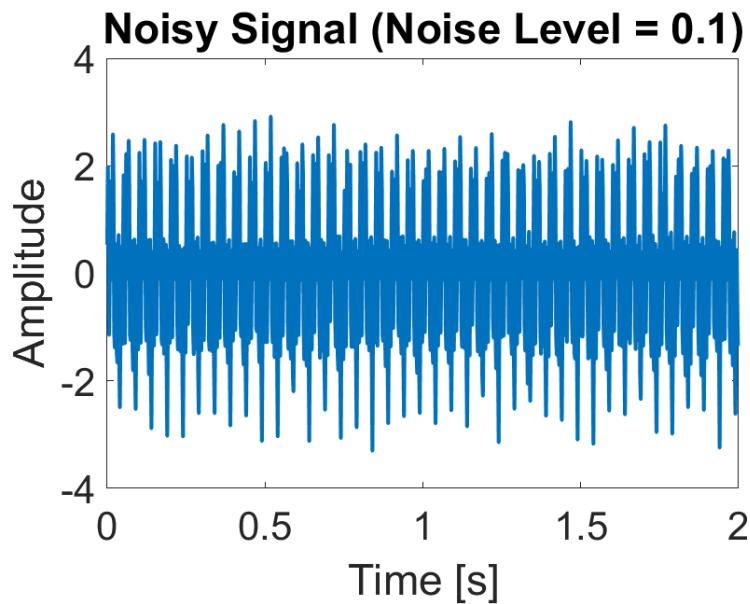


Figure 105: Original Input Signal $X = X_1 + X_2 + X_3$

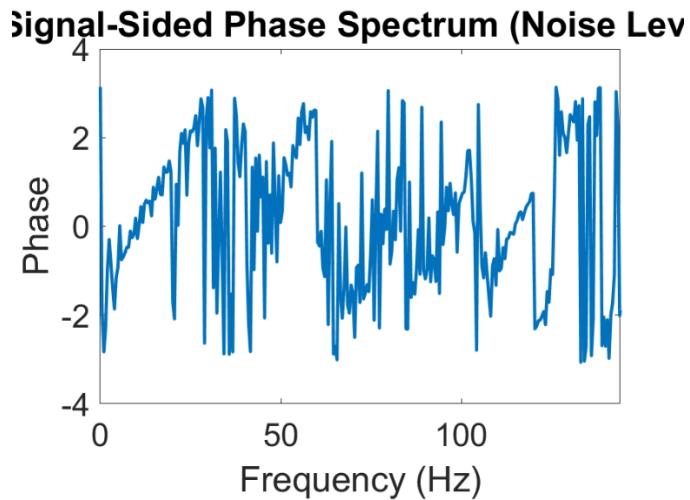


Figure 106: Phase Spectrum of the Input Signal as a Function of Frequency for $X = X_1 + X_2 + X_3$

4.5.3 Noise=0.5 is added to $x=x_1+x_2+x_3$ and the generated plots are shown below:

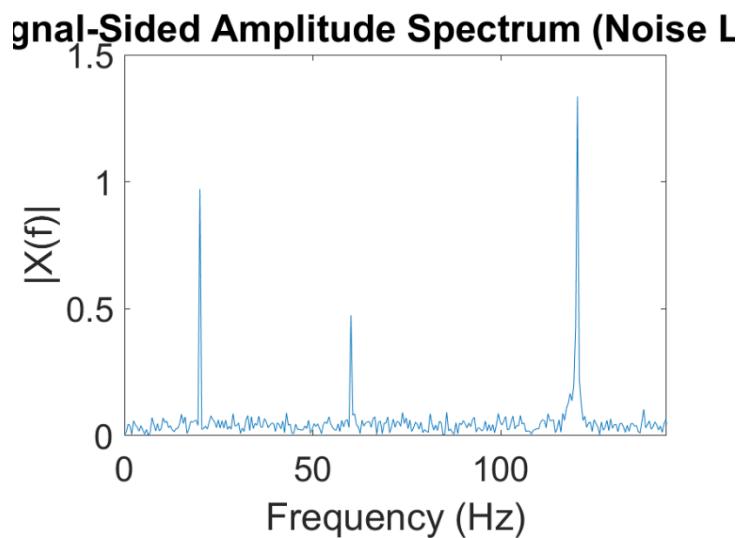


Figure 107: Frequency vs. Amplitude Plot for $X = X_1 + X_2 + X_3$

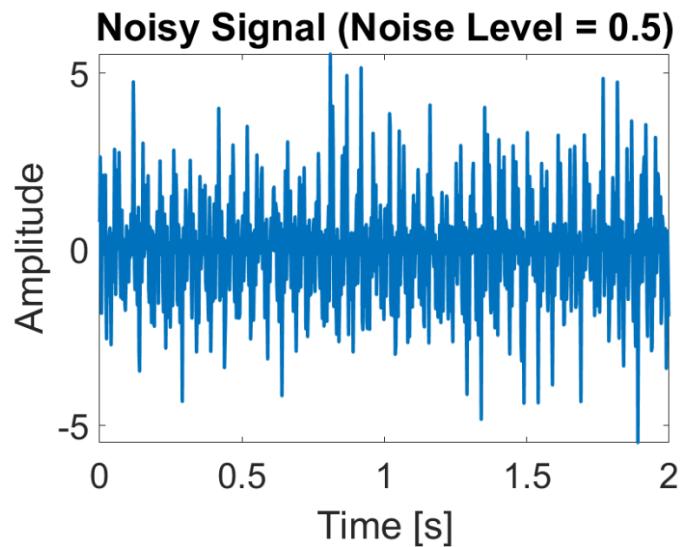


Figure 108: Original Input Signal $X = X_1 + X_2 + X_3$

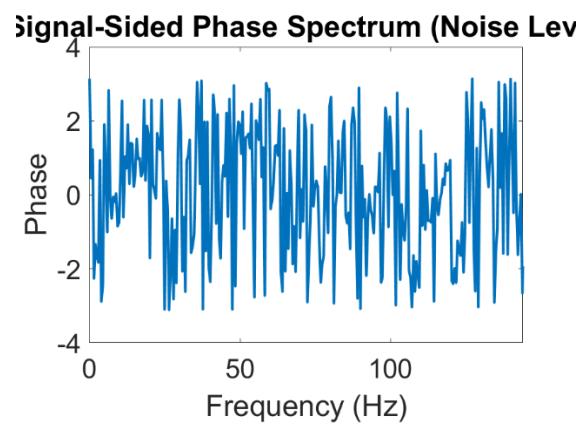


Figure 109: : Phase Spectrum of the Input Signal as a Function of Frequency for $X = X_1 + X_2 + X_3$

4.5.4 Noise=1 is added to $x=x_1+x_2+x_3$ and the generated plots are shown below:

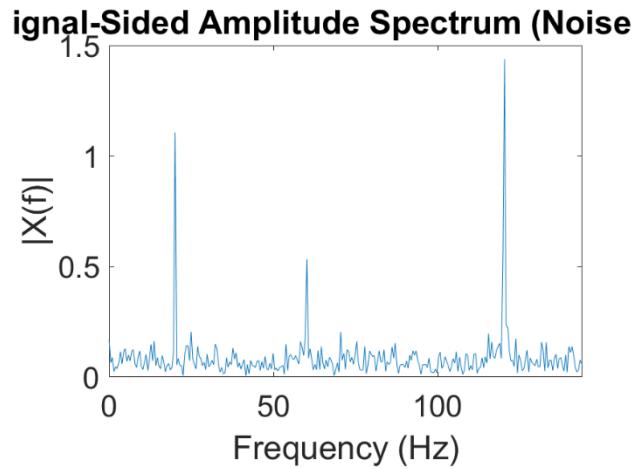


Figure 110: Frequency vs. Amplitude Plot for $X = X_1 + X_2 + X_3$

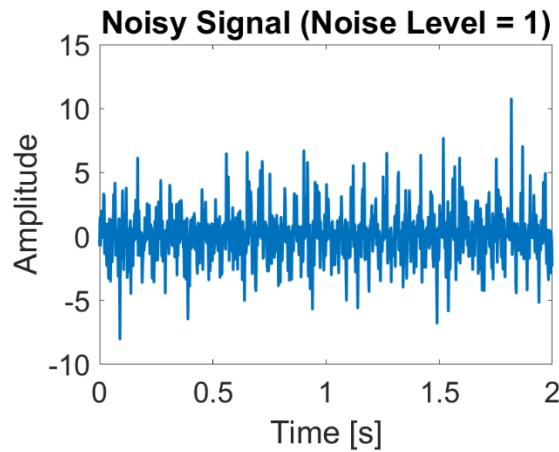


Figure 111: Original Input Signal $X = X_1 + X_2 + X_3$

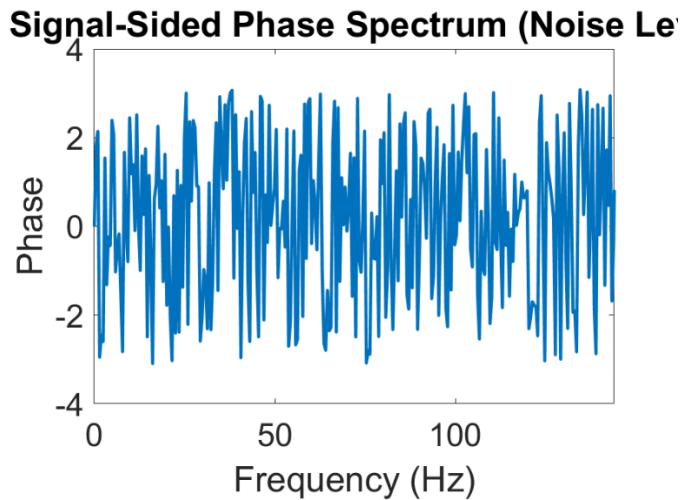


Figure 112: Phase Spectrum of the Input Signal as a Function of Frequency for $X = X_1 + X_2 + X_3$

4.5.5 Observation based on the parametric analysis of Noise

When we add random noise to a signal and then perform the Fast Fourier Transform (FFT), several observations can be made:

4.5.5.1 Effect on the Time Domain (Original Signal):

- The original signal is a combination of sinusoidal components (x_1 , x_2 , and x_3).
- Adding random noise to the original signal introduced the variations and fluctuations in the time-domain waveform.

4.5.5.2 Effect on the Frequency Domain (FFT Spectrum):

- The FFT spectrum is showing additional frequency components corresponding to the noise.
- As we increased the noise level, these noise-related frequency components are becoming more pronounced, which is clearly depicted through the plot with noise = 1, apart from fundamental frequencies there are other frequencies which are becoming dominant. With further noise, it may reach to a threshold which makes little to impossible to extract the fundamental frequencies and consequently loss of information.

4.5.5.3 Amplitude Spectrum Changes:

- After adding the noise the amplitudes of the original frequency components become less distinct as the noise level increases.
- The noise itself may contribute to higher amplitude values in the spectrum, in third case of noise = 1, amplitude spectrum scale changes from +5/-5 to +15/-15.

4.5.5.4 Phase Spectrum Changes:

The phase spectrum will be influenced by the noise, and it may be observed in fluctuations in the phase values. It is observed that when a noise is added to the signals (any one or all three), the phase spectra are having a lot of distortions which makes it harder to analyse the results. If the noise level is increased, then there are a lot of peaks so it is difficult and almost impossible to derive the amplitude and phase.

4.6 Signal Averaging

Exploration of signal averaging in time and frequency domains involved repeated signal generation in the presence of noise. Subsequent averaging in each domain facilitated an evaluation of the efficacy of each method in preserving signal fidelity.

- In exploring the effects of noise on a signal, we plotted and compared it involving the generation of a noisy signal through repeated instances. The original signal, comprised of distinct sinusoidal components, provided a baseline for comparison.

In the initial assessment, the introduction of Gaussian white noise resulted in noisy signals characterized by heightened random fluctuations. This led us to consider two prominent methods of signal averaging: time-domain averaging and frequency-domain averaging.

Time-domain averaging Figure 115, implemented through the generation of the noisy signal multiple times (specifically, 10 instances), yielded a promising outcome. The averaged signal in the time domain effectively attenuated the impact of random noise. Peaks and distinctive

features of the original signal became more pronounced, highlighting the success of time-domain averaging in reducing random fluctuations.

On the other hand, frequency-domain averaging [Figure 113: Original and Averaged Phase Spectra Plot] involved computing the Fourier transform for each noisy signal instance, averaging the magnitudes and phases, and reconstructing the averaged spectrum. The frequency-domain averaged spectrum demonstrated a *notable reduction in the influence of frequency-specific noise*. This approach offered a **clearer representation of the underlying signal components**, particularly beneficial when noise exhibited specific frequency characteristics.

The nature of the introduced noise, being primarily random and uncorrelated in time, steered our choice towards time-domain averaging. This method effectively mitigated random fluctuations, resulting in a more discernible representation of the original signal. The 10-fold repetition of noise-corrupted signal generation, coupled with time-domain averaging, emerged as a suitable approach for preserving essential signal features.

Therefore, for the case of the noisy signal in this study, time-domain averaging proved to be more appropriate, aligning with the nature of the noise and contributing to a clearer and more refined representation of the underlying signal.

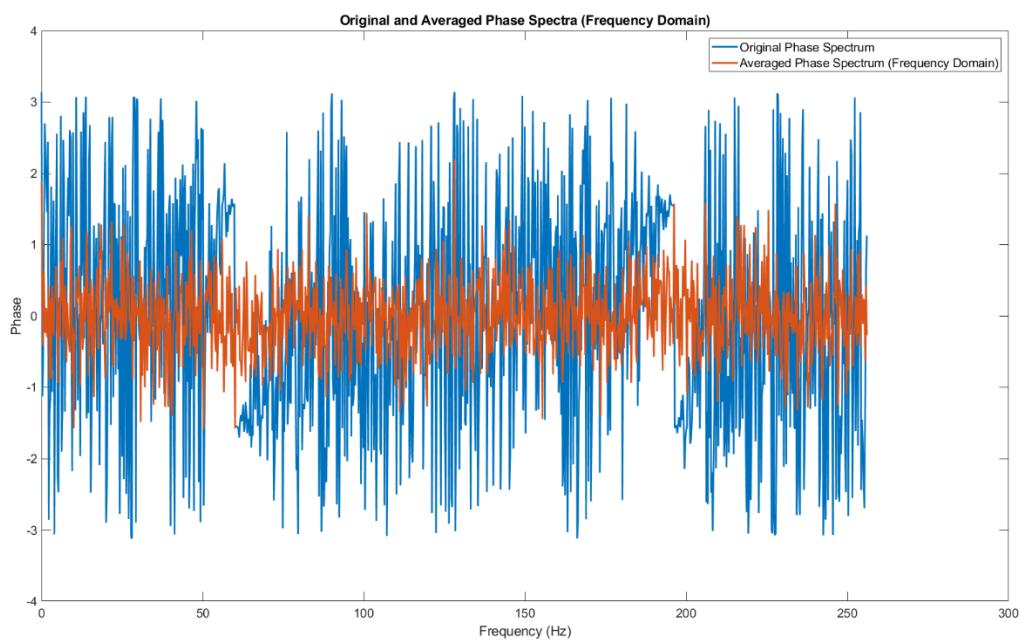


Figure 113: Original and Averaged Phase Spectra Plot

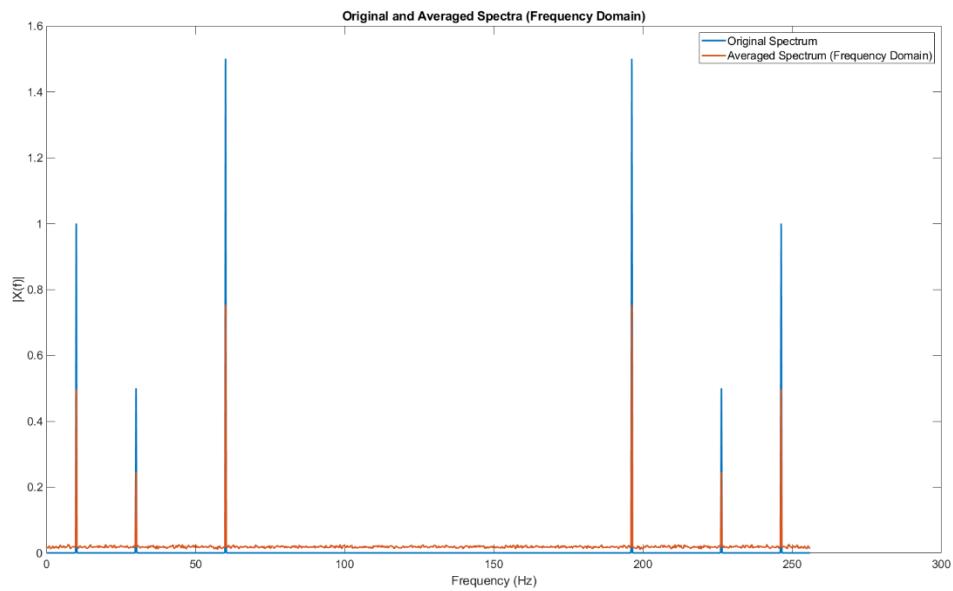


Figure 114: Frequency vs. Amplitude Plot for Original and Averaged Spectra (Frequency domain)

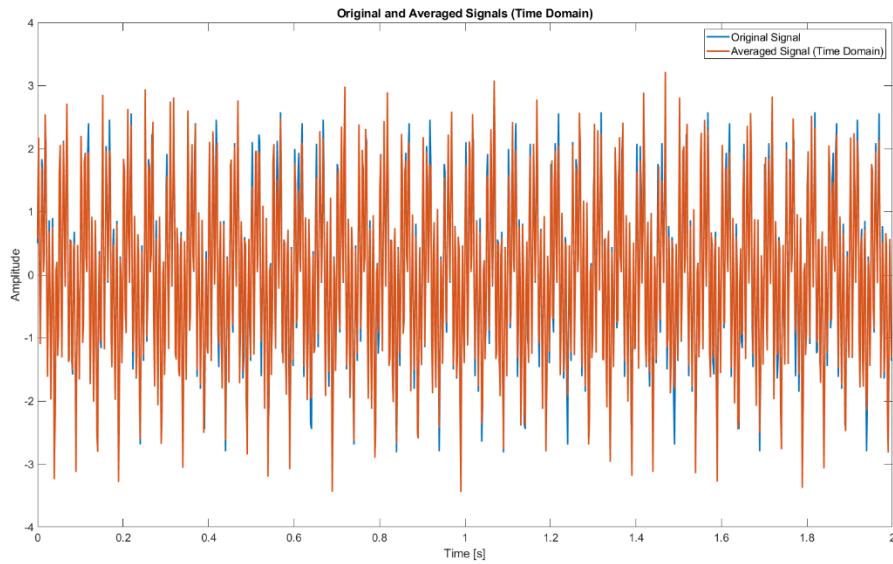


Figure 115: Original and averaged signal plot showing amplitude vs. time (Time Domain)

4.6.1 Quality improvement

Signal averaging, particularly in the time domain, has proven effective in enhancing the quality of results by repetitively generating and averaging noisy signals. This process reduces inherent noise, yielding a clearer and more accurate representation of the original signal in both time and frequency domains. In the time domain, the averaged signal shows improved visibility of key features and reduced fluctuations. However, caution is advised when employing frequency-domain averaging, as it may lead to a trade-off by attenuating frequency-specific noise but potentially sacrificing detailed frequency components. This underscores the importance of aligning the choice of averaging method with the specific characteristics of the signal and noise, with time-domain averaging being more suitable for cases where preserving detailed frequency information is crucial. Ultimately, the selection between time and frequency-domain averaging should be informed by the specific requirements of the signal analysis.

4.7 Equation of Motion Analysis

The provided MATLAB code simulates and analyses the behaviour of a harmonic oscillator subjected to damping and stiffness. The central function, **Myharmonic_Ana**, computes the

displacement of the oscillator over a specified time span, considering given parameters such as stiffness (**k**), damping (**c**), and mass (**rho**). The simulation can accommodate either two or three parameters, offering flexibility in system characterization.

Subsequently, the code introduces a noise component to the simulated displacement measurements. The amount of noise is controlled by the parameter **sigma_noise**. The noisy measurements, denoted as **u_meas**, are then saved along with other relevant information, including the sampling rate, initial deflection, system parameters, noise level, and the time span.

The code incorporates visualization aspects by plotting the *noisy measured data in red* and overlaying it with the *noise-free simulation in blue*. Additionally, the analytical solution of the harmonic oscillator (**u0**) is plotted for reference. This setup allows for a comprehensive examination of how noise influences the measurements, providing insights into the robustness of the harmonic oscillator system under real-world conditions.

4.7.1 Graphs

The provided code generates and analyses the amplitude spectrum, phase spectrum, and spectrogram of a given noisy signal. Here's an explanation of the generated graphs:

1. Original Input Signal:

- A time-domain plot of the original noisy input signal (**x_noise**) is displayed.
- The x-axis represents time (**t**), and the y-axis represents the amplitude of the signal.

2. Input Signal-Sided Amplitude Spectrum:

- The amplitude spectrum of the input signal is plotted in the frequency domain.
- The x-axis represents frequency (**f**), and the y-axis represents the amplitude ($|X(f)|$).
- The plot is limited to frequencies up to 5 Hz.

3. Input Signal-Sided Phase Spectrum:

- The phase spectrum of the input signal is plotted in the frequency domain.
- The x-axis represents frequency (f), and the y-axis represents the phase angle of the signal.

4. Spectrogram:

- The spectrogram provides a time-frequency representation of the signal.
- The x-axis represents time (t), the y-axis represents frequency (f), and the color represents the logarithm of the signal power at each time-frequency point.
- A hamming window is applied to the signal before computing the spectrogram.

These visualizations offer insights into both the time and frequency characteristics of the noisy input signal. The time-domain plot shows the signal's amplitude variations over time, while the amplitude and phase spectra reveal its frequency components. The spectrogram provides a detailed view of how the signal's frequency content evolves over time, allowing for a comprehensive analysis of its dynamic behavior.

4.7.2 FFT Analysis of Equation of Motion

The Equation of Motion, pivotal in system identification, underwent detailed examination. Application of FFT to the equation's solution unveiled inherent system dynamics within the spectrum. The team meticulously examined the observed spectrum characteristics, drawing correlations to underlying system behaviors.

4.7.3 Effects of Damping

Exploration into varied damping values provided insights into their influence on FFT results. Comparative analysis between cases with no damping and different damping strengths

revealed nuanced variations in system responses. These spectral changes were meticulously analyzed and interpreted.

4.7.3.1 Damping values

The choice of damping value in a system depends on the specific characteristics of the physical system being modelled and the behaviour you want to simulate or analyse. Damping is a critical parameter that influences the transient response and stability of a system.

Here are some general guidelines for choosing damping values: (*Parameter Estimation and Inverse Problems*, 2018)

1. **Critical Damping ($\xi = 1$):** Critical damping provides the fastest approach to equilibrium without overshooting. This is often desirable for systems that need to settle quickly without oscillations. However, achieving critical damping may not always be practical due to physical constraints or other considerations.
2. **Underdamping ($0 < \xi < 1$):** Underdamped systems exhibit oscillatory behaviour during transient response. The damping ratio (ξ) determines the extent of oscillation. A lower damping ratio results in more pronounced oscillations.
3. **Overdamping ($\xi > 1$):** Overdamped systems do not oscillate during transient response. Instead, they have a slower approach to equilibrium. Overdamping is often chosen for systems where minimizing oscillations is a priority.

4.7.3.1.1 $\xi = 0$

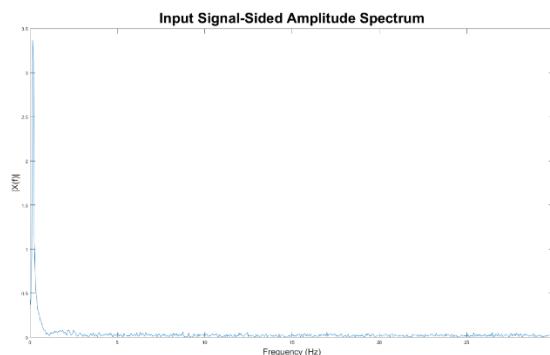


Figure 116: $c=0$; amplitude spectrum

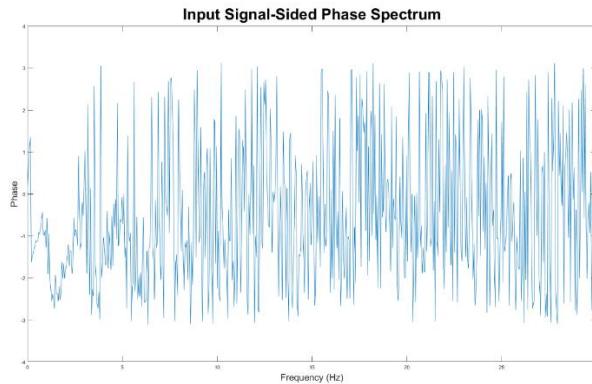


Figure 117: $c=0$; phase spectrum

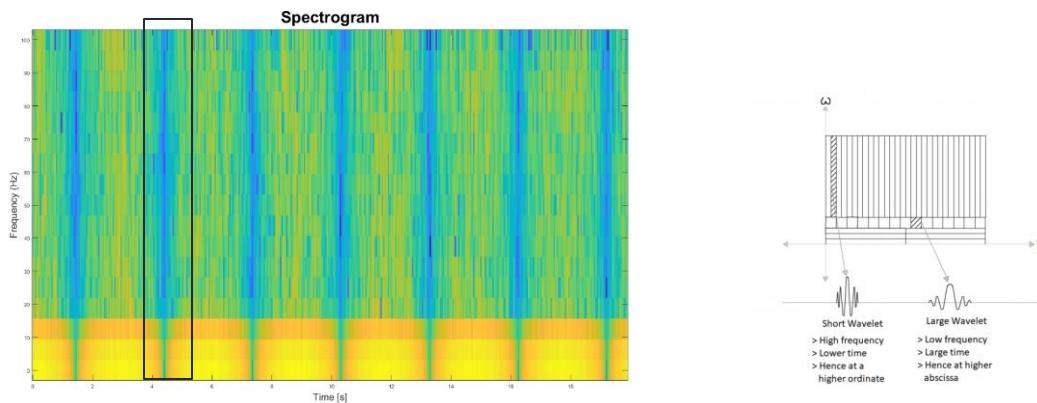


Figure 118: Wavelet Spectrogram for no damping (left), Wavelet Spectrogram for reference from (ADMIN, 2018; Percival & Walden, 2000; Shaik, n.d.)

4.7.3.1.1 Observation:

In the context of a sinusoidal signal, the insights gained from the scaleogram and wavelet transform take on a specific significance. A sinusoidal signal is characterized by its frequency, amplitude, and phase, and examining these aspects in both the frequency and temporal domains can provide a deeper understanding of its behavior.

1. In the scaleogram of a sinusoidal signal, the concentration of power within the 0-15 Hz range aligns with the inherent frequency of the sinusoid. The increase in power observed in the Fourier transform at these frequencies confirms the dominance of sinusoidal components in the signal.

2. The continuous wavelet transform goes beyond this by offering temporal information, revealing not only the dominant frequencies but also how they evolve over time. Upon observing a repeating vertical line at regular 2.5-second intervals in the scaleogram, it indicates a consistent and periodic feature in the signal. This periodicity corresponds to a frequency of 0.4 Hz (1 / 2.5 seconds), suggesting a stable oscillation as a dominant characteristic. The purple vertical lines may signify harmonic components or integer multiples of the fundamental frequency, contributing to the regular pattern seen in the scaleogram.

3. In the case of a sinusoidal signal, the uniform distribution of energy in the time scale implies a consistent oscillation of the sinusoid, which is expected for a steady-state periodic signal. The ability to discern nearly steady-state oscillations near 0 Hz to 15 Hz provides a comprehensive picture of the sinusoidal behavior and its temporal characteristics.

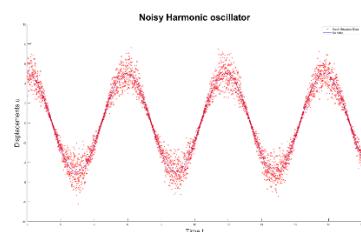


Figure 119: Noisy harmonic oscillator

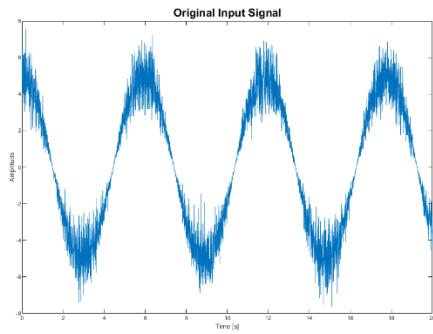


Figure 120: $c=0$; original input signal

4.7.3.1.2 Noise 0.1

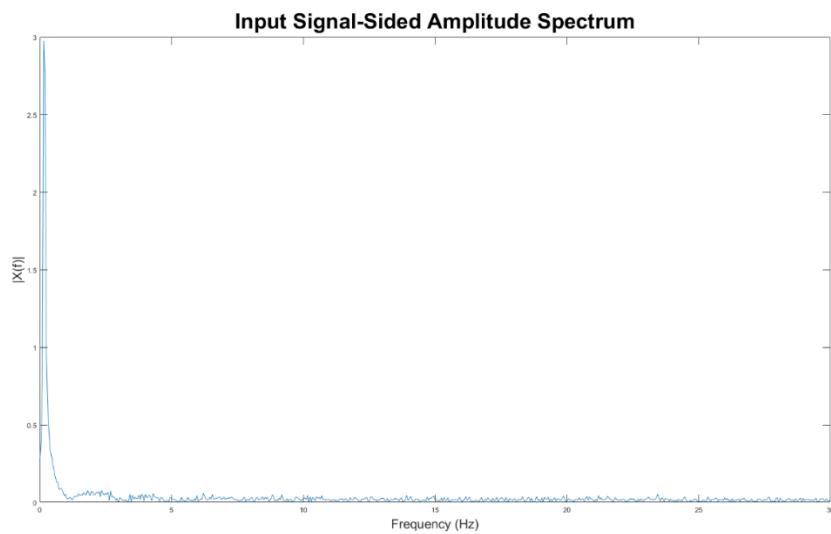


Figure 121: $c=0.1$; amplitude spectrum

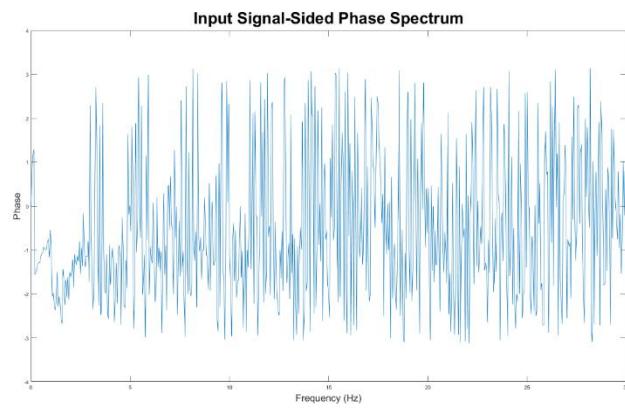


Figure 122: $c=0.1$; phase spectrum

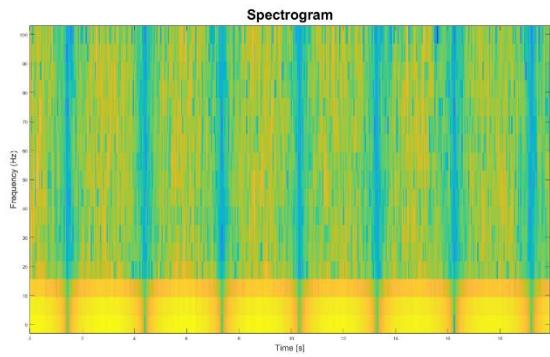


Figure 123: $c = 0.1$; Spectrogram

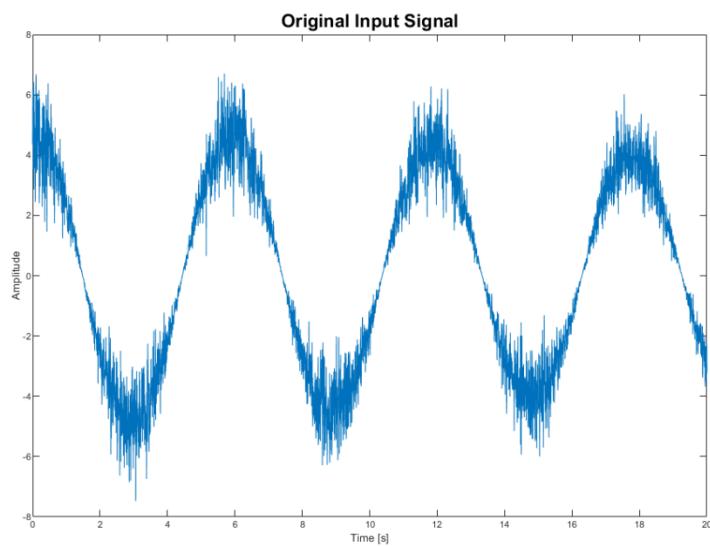


Figure 124: $c=0.1$; original input signal

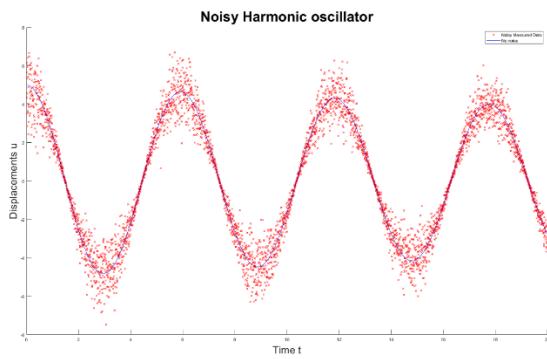


Figure 125: Noisy harmonic oscillator

4.7.3.1.3 Noise 1

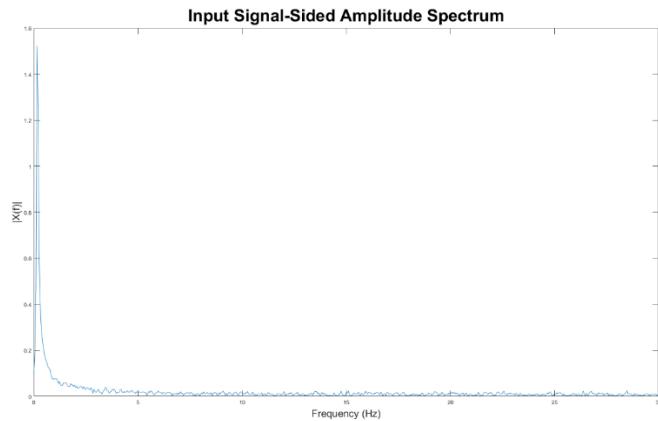


Figure 126: $c=1$; amplitude spectrum

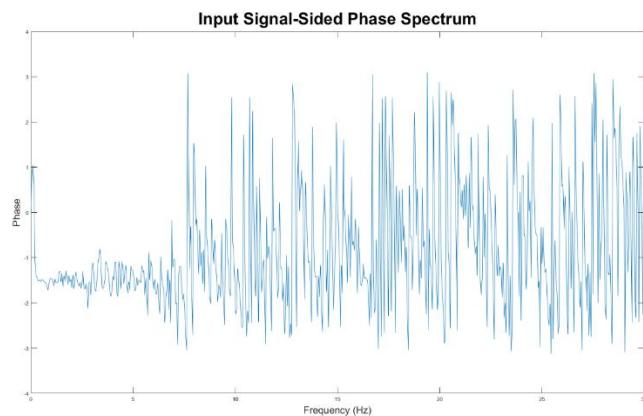


Figure 127: $c=1$; phase spectrum

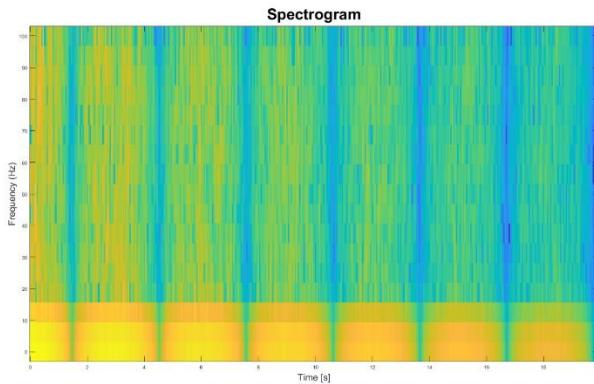


Figure 128: $c=1$; Spectrogram

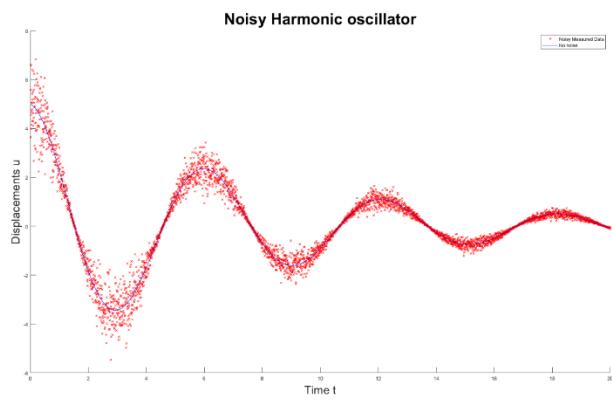


Figure 129: Noisy harmonic oscillator

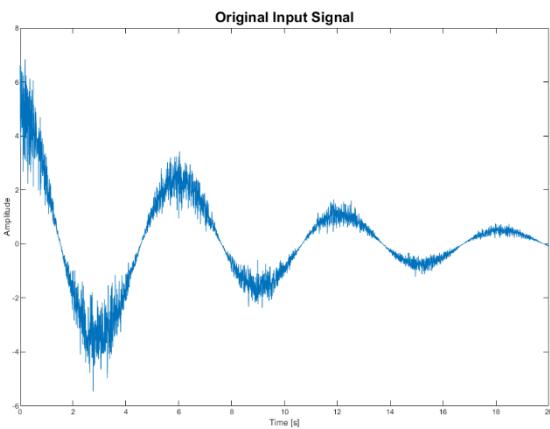


Figure 130: $c=1$; original input signal

4.7.3.1.4 Noise 5

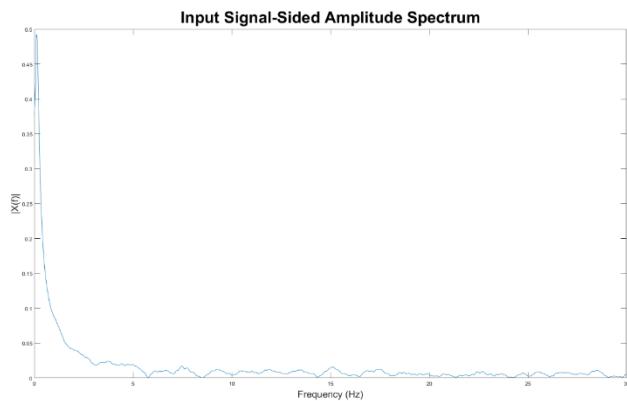


Figure 131: $c=5$; amplitude spectrum

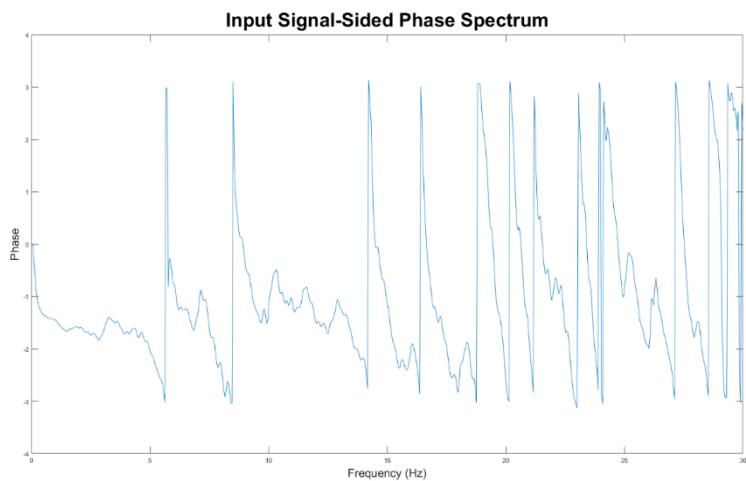


Figure 132: $c=5$; phase spectrum

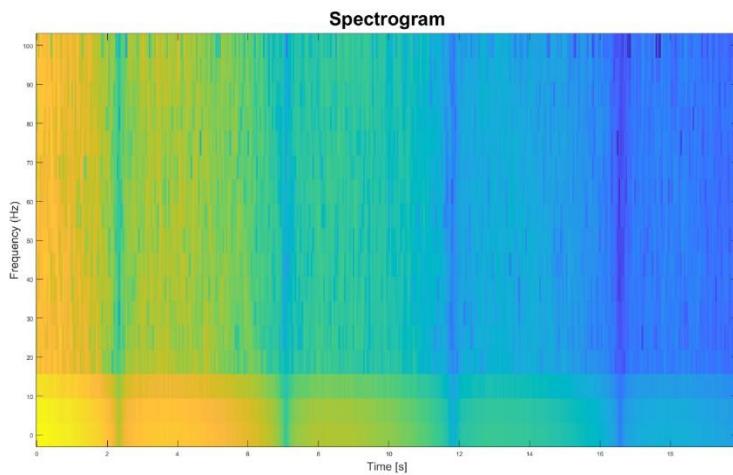


Figure 133: $c = 5$; Spectrogram

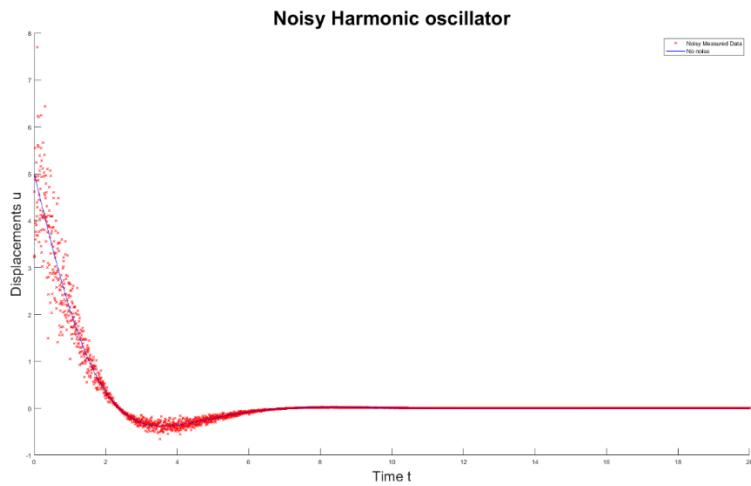


Figure 134: Noisy harmonic oscillator

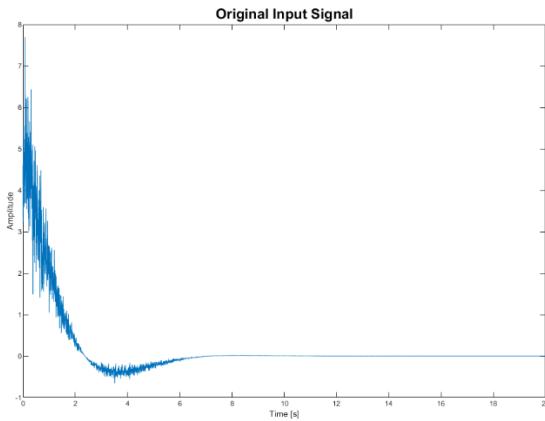


Figure 135: c=5; original input signal

4.7.3.1.5 Observation:

Wavelet Spectrogram Analysis:

The examination of signals with varying damping coefficients (0, 0.1, 1, and 5) utilizes the wavelet spectrogram to provide a comprehensive representation of frequency content and temporal dynamics.

The spectrogram visually demonstrates the influence of damping on signal characteristics, revealing distinct shifts in energy distribution as damping values change.

Higher damping coefficients lead to the suppression of specific frequency ranges, resulting in concentrated energy around particular frequencies. This frequency modulation is closely connected to the amplitude spectrum, illustrating that damping not only affects the presence but also the magnitude of specific frequencies, with the damping of 5, energy concentration between time range of 20 sec, decreases to half or more.

Interplay with Phase Spectrum:

The decrease in phase values for many frequencies with increasing damping is a consequence of the dissipative nature of damping, the selective attenuation of frequency components, and the altered temporal relationships within the oscillatory system.

Temporal Dynamics Through Harmonic Oscillation:

Examining harmonic oscillation plots in the time domain provides insights into the evolving nature of the signal under different damping conditions. As damping increases, the system exhibits a more pronounced and rapid decay in the harmonic oscillation, reflecting the direct influence of damping on the temporal response.

Transient Effects and Spectrogram Dynamics:

The wavelet spectrogram captures transient effects heightened by increasing damping, manifesting as distinct features. The number of vertical purple lines decreases, and there is diminished energy concentration between 0 to 15 Hz at higher damping values, indicating a shift in the distribution of energy over time and frequency.

4.7.3.2 param = [4.5 0 4] ;

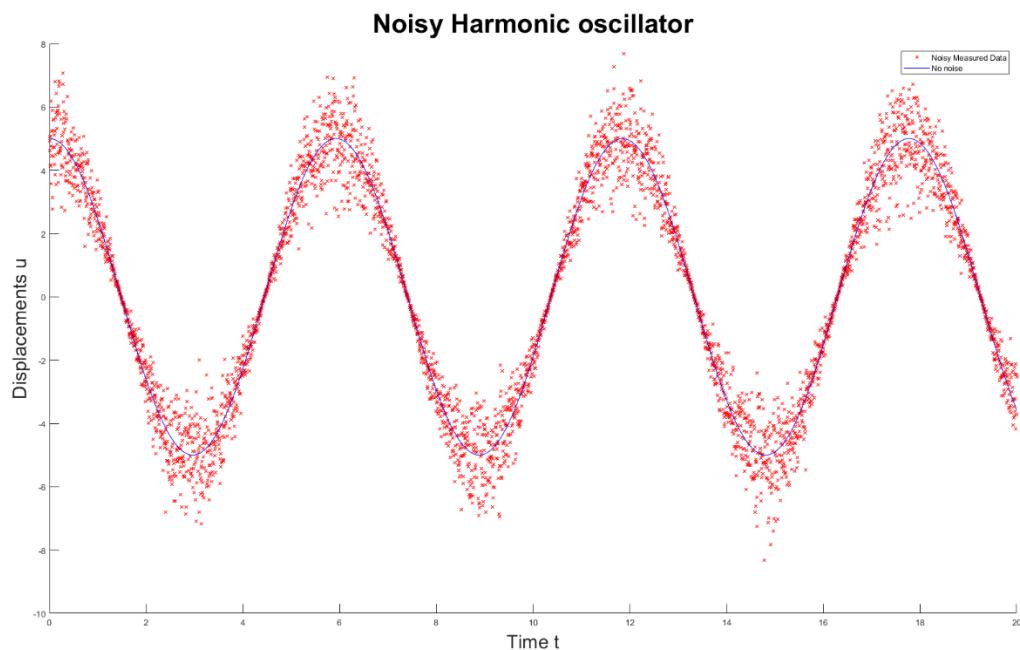


Figure 136: Noisy harmonic oscillator

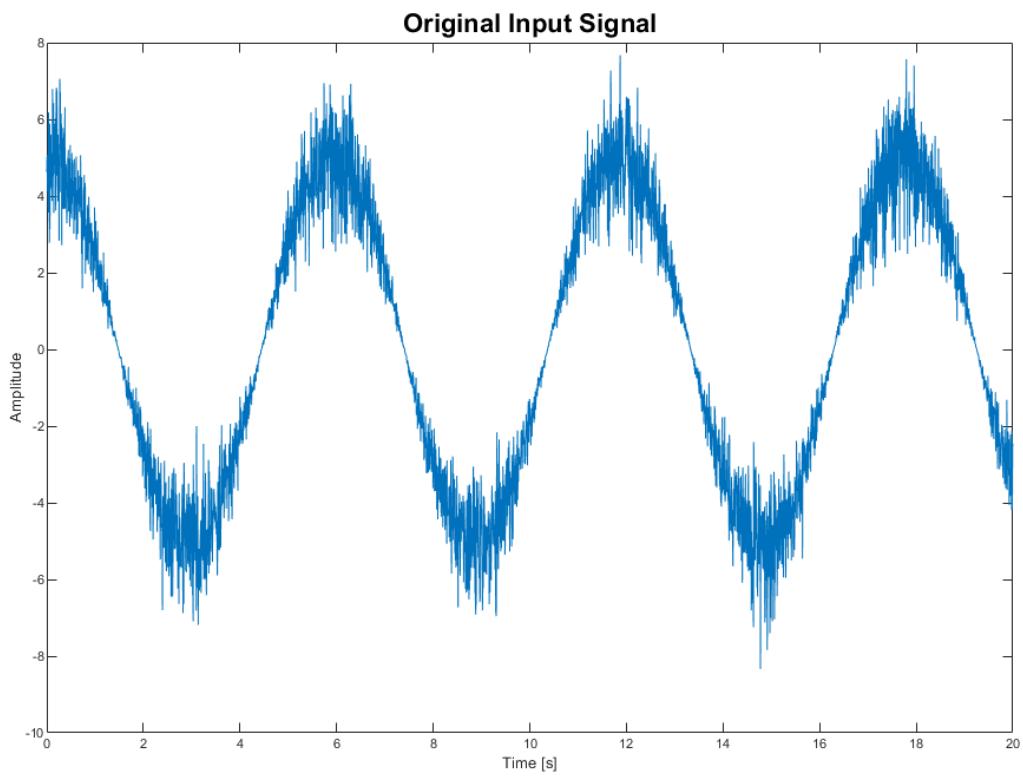


Figure 137: $m=4$; original input signal

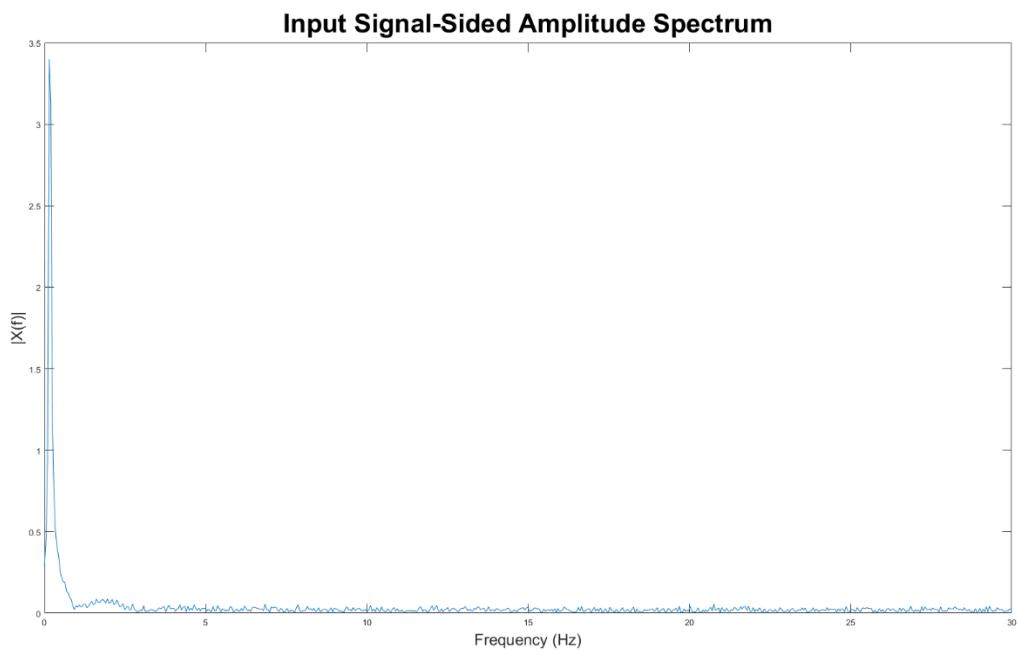


Figure 138: $m=4$; amplitude spectrum

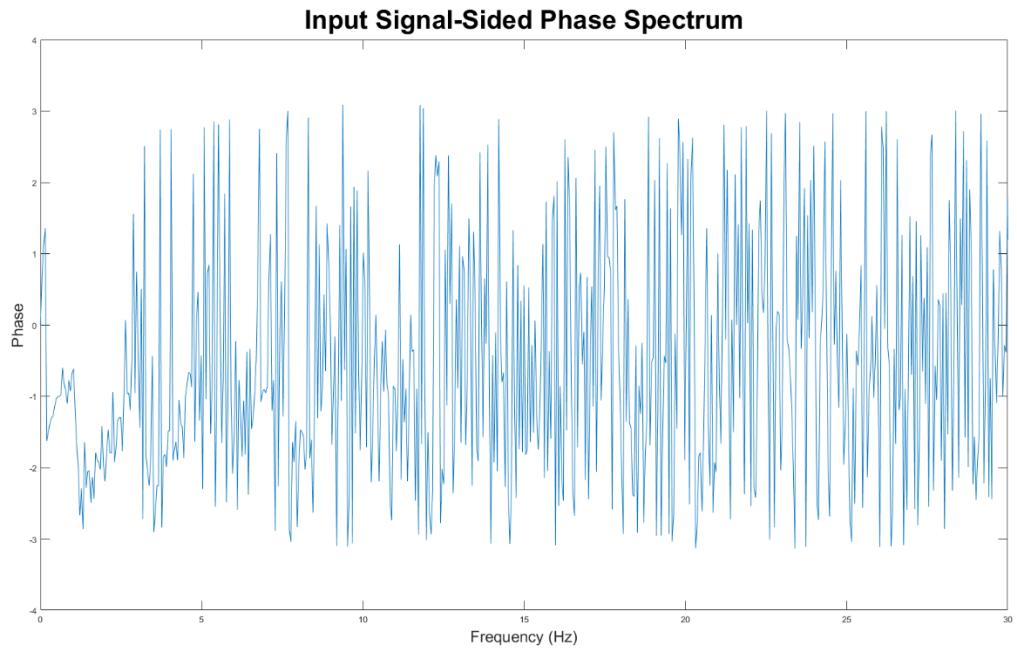


Figure 139: $m=4$; phase spectrum

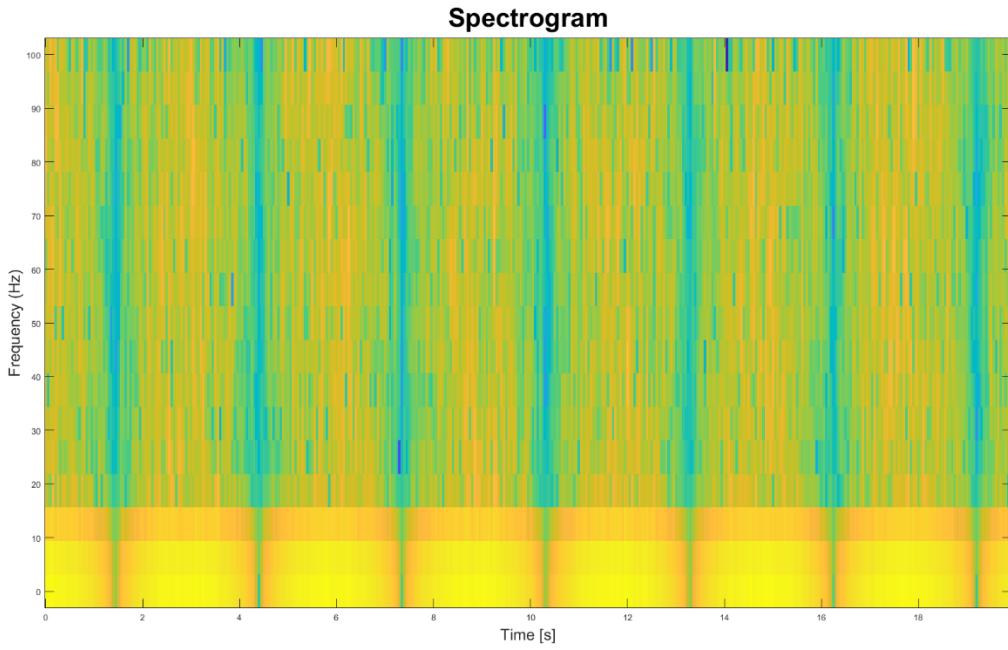


Figure 140: $m = 4$; Spectrogram

4.7.3.3 param = [4.5 0 100] ;

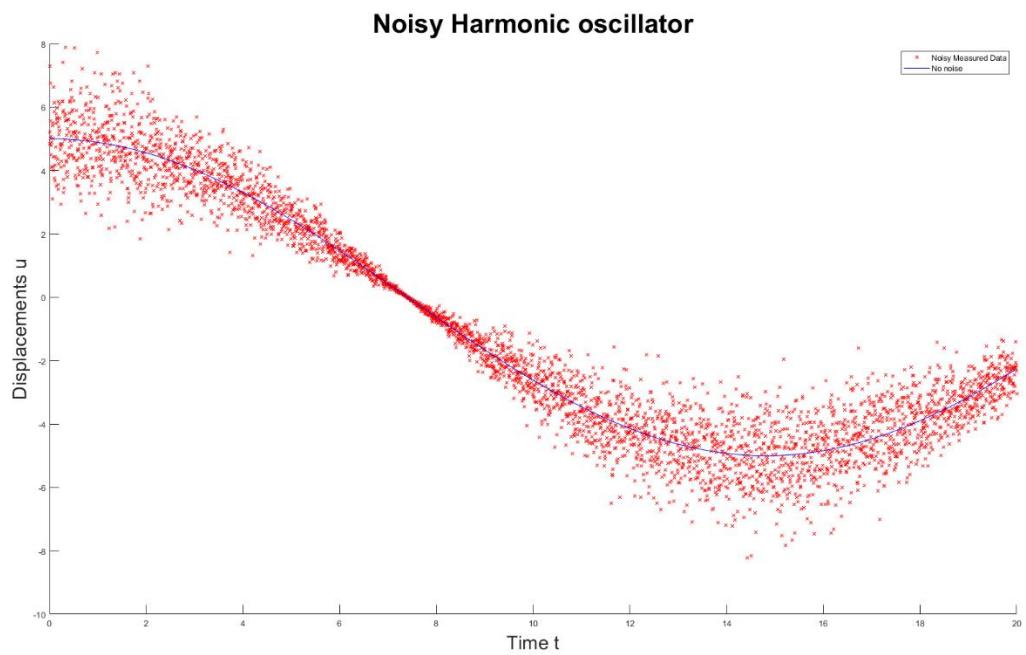


Figure 141: Noisy harmonic oscillator

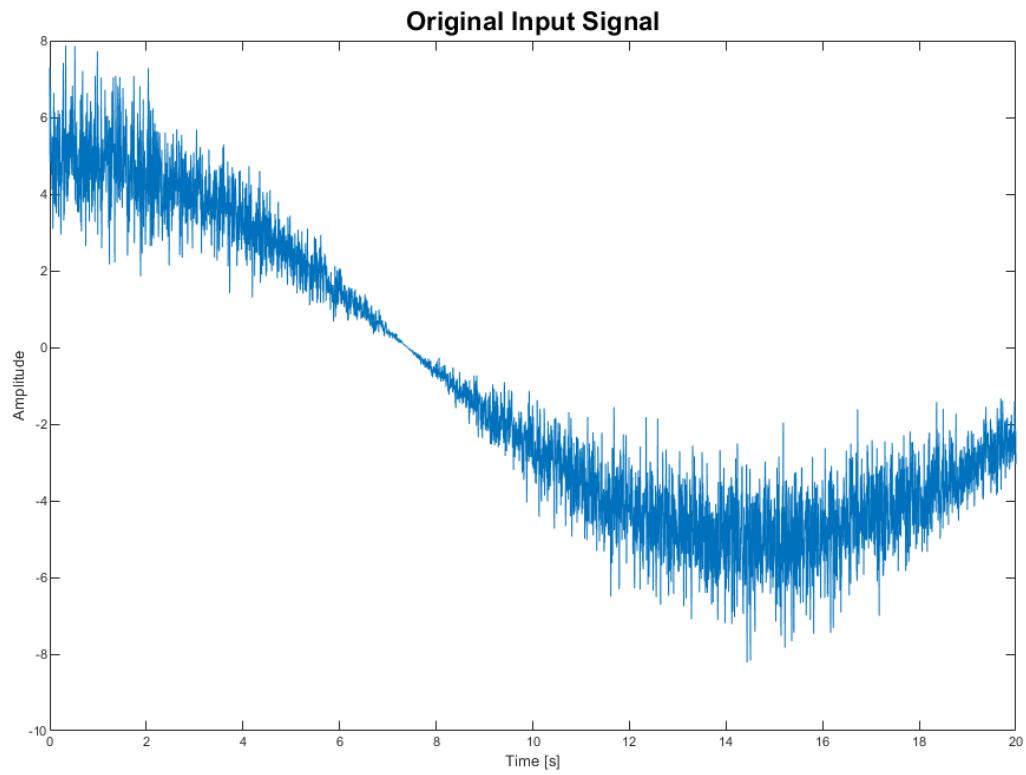


Figure 142: $m=100$; original input signal

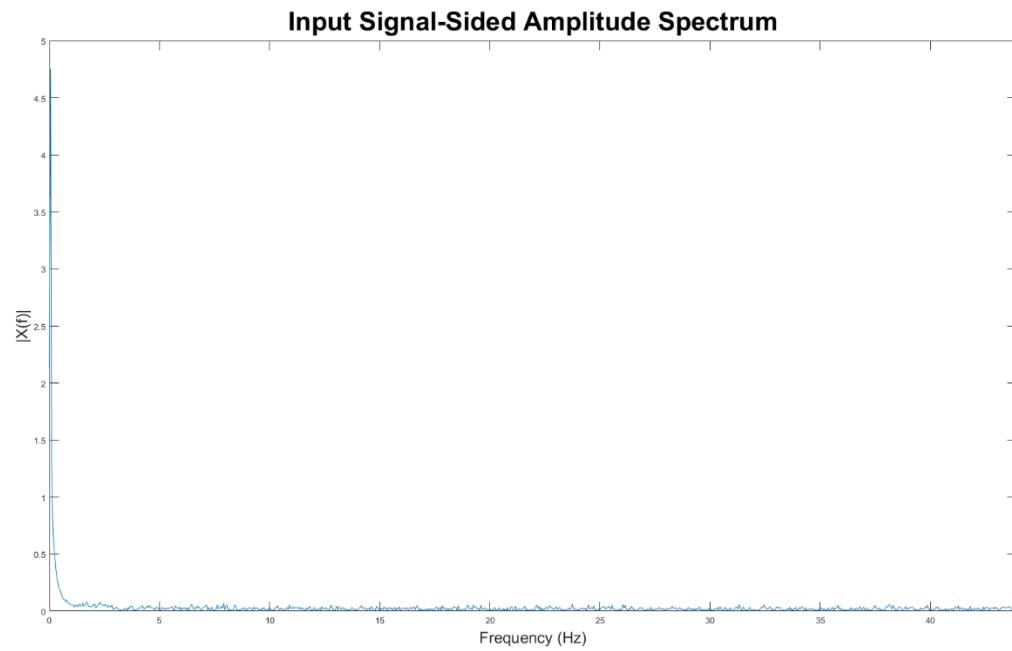


Figure 143: $m=100$; amplitude spectrum

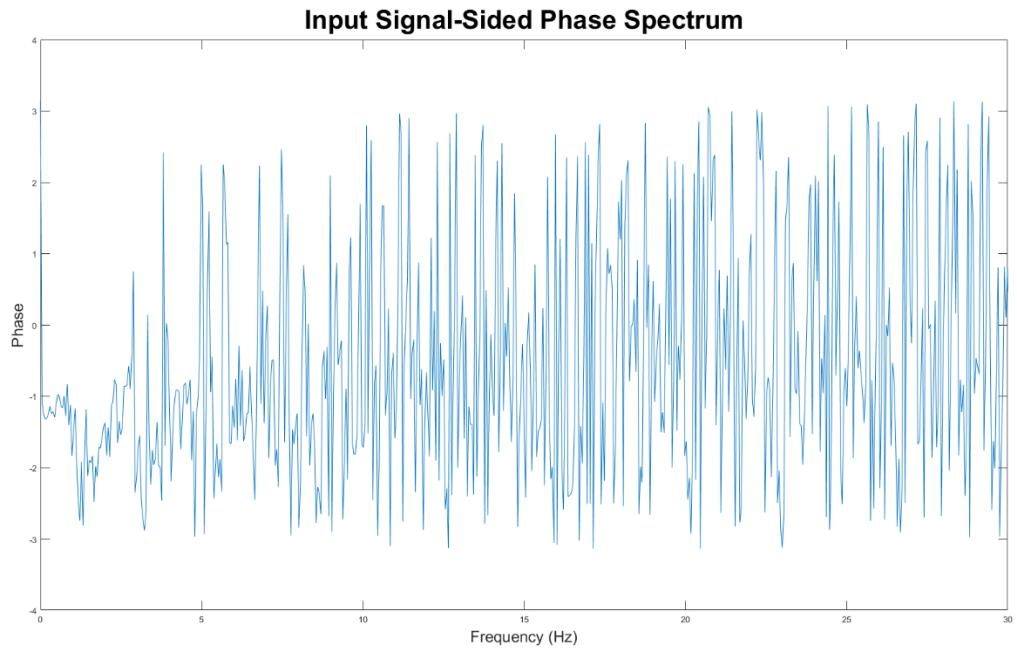


Figure 144: $m=100$; phase spectrum

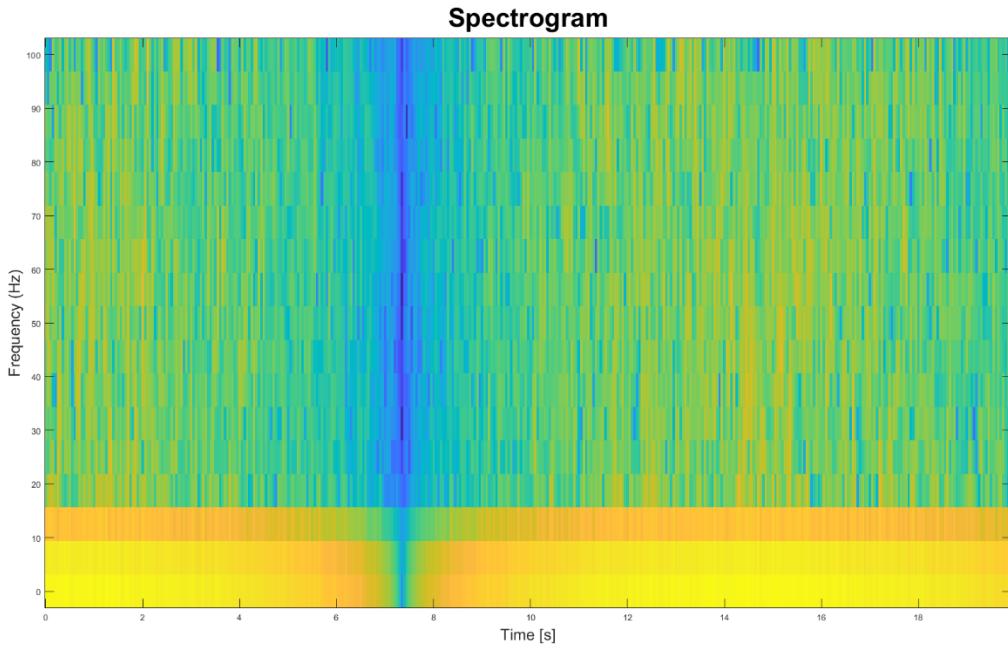


Figure 145: $m=100$; Spectrogram

4.7.3.4 param = [10 0 4] ;

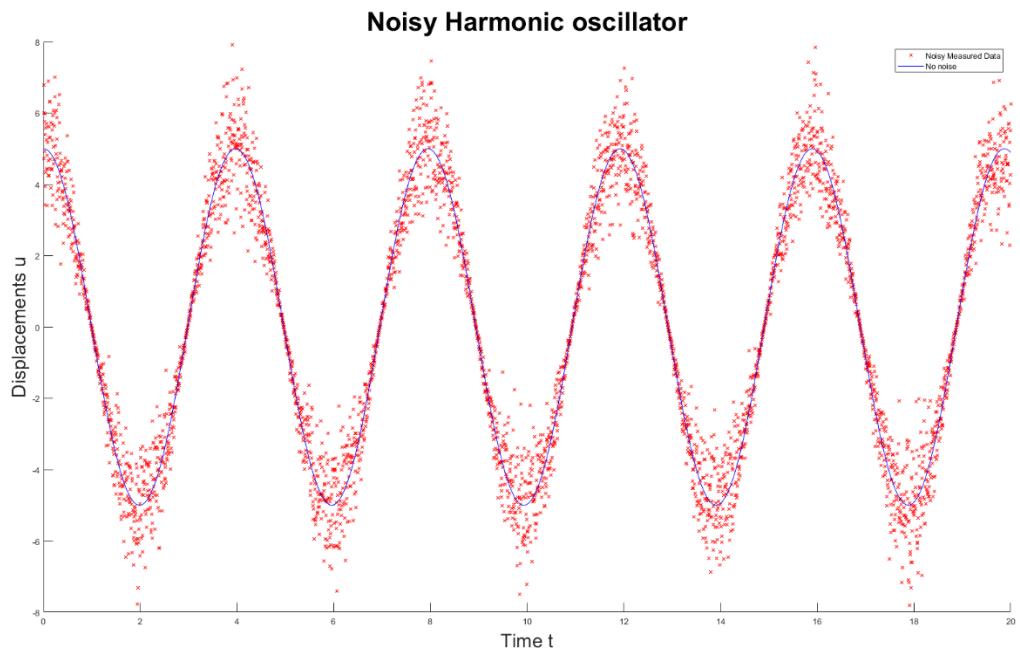


Figure 146: $k = 10$; Noisy harmonic oscillator

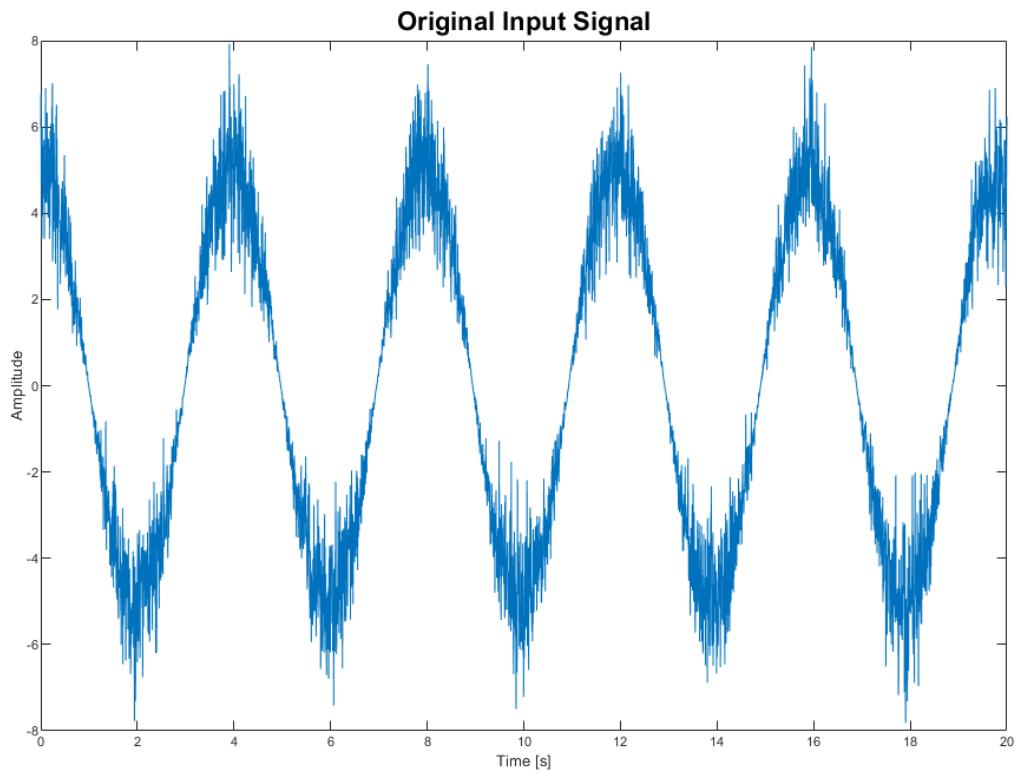


Figure 147: : $k=10$; original input signal

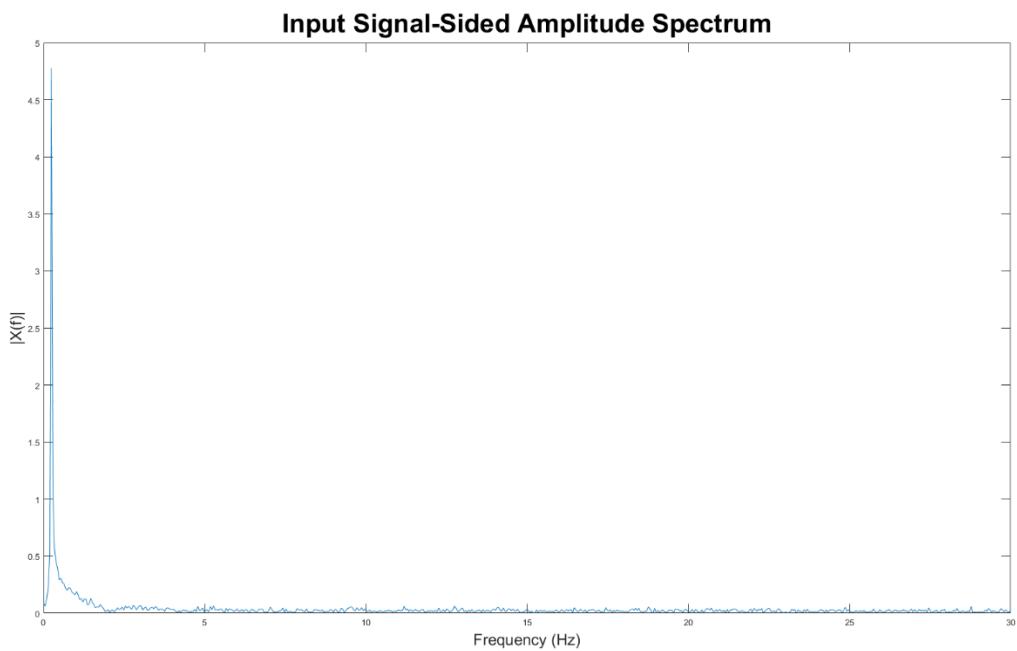


Figure 148: $k=10$; amplitude spectrum

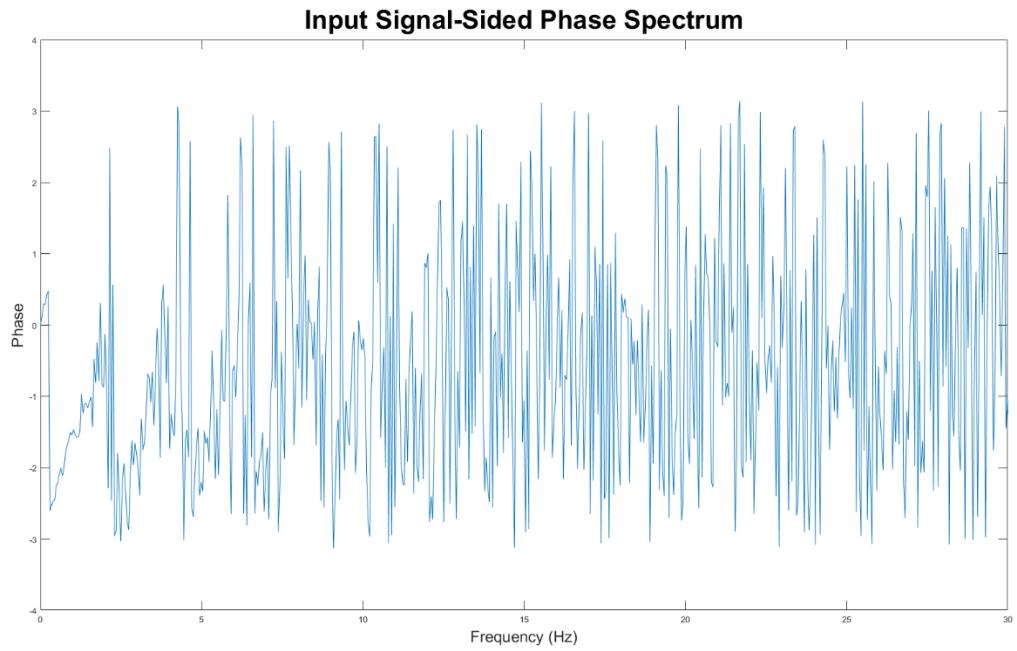


Figure 149: $k=10$; phase spectrum

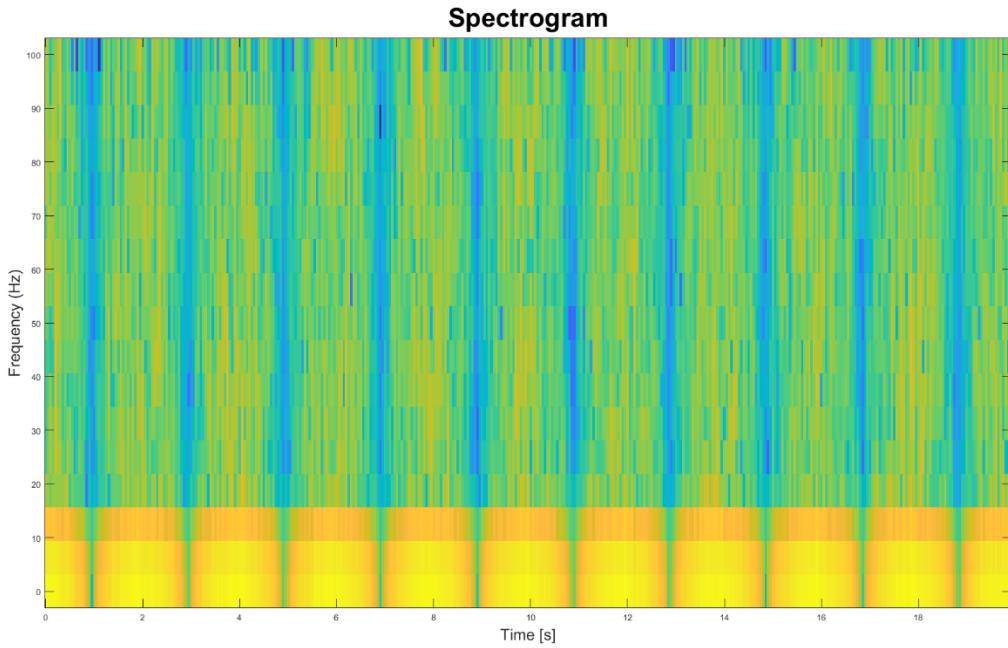


Figure 150: $k = 10$; Spectrogram

4.7.3.5 Observation:

The systematic exploration of individual variations in mass (m) and spring stiffness (k) was conducted, revealing distinctive effects on the system's behavior. When altering mass, amplitude spectrum plots showcased discernible shifts in frequency components. Higher mass values introduced a dampening effect on vibrational characteristics, leading to a noticeable decrease in peak amplitudes and a shift in natural frequencies to lower values. Concurrently, the phase spectrum analyses illustrated alterations in temporal relationships among frequency components, emphasizing the significant influence of mass on phase dynamics.

Conversely, varying spring stiffness (k) exhibited notable changes in the amplitude spectrum, highlighting shifts in dominant frequencies. Higher spring stiffness values resulted in amplified amplitudes, indicating an intensified response of the system. This was accompanied by a shift in natural frequencies to higher values. The phase spectrum analyses mirrored these findings, depicting pronounced phase shifts that further underscored the impact of spring stiffness on the temporal aspects of the signal.

Notably, increasing the mass of the system shifts the natural frequencies to lower values, while increasing the stiffness shifts the natural frequencies to higher values. The peak amplitude increases with the increase in stiffness and decreases with the increase in mass, revealing the system's sensitivity to changes in these parameters.

5. Experiment: 5- Signal processing - 2

5.1 Introduction

This report is an in-depth exploration of Signal Analysis, System Identification, and mainly **Noise Analysis, amalgamated through a series of exercises**. The exercises span a spectrum from scrutinizing real signals to investigating coherence functions in differential equations, performing wavelet transformations, applying filters, and delving into noise analysis within the signal domain. This comprehensive exploration aims to unravel the intricacies of signal characteristics, the impact of noise on signal integrity, and the robustness of signal processing techniques. By meticulously examining these exercises, we aspire to not only comprehend fundamental signal processing methodologies like spectral analysis, correlation functions, and wavelet transforms but also scrutinize the influence of noise on signal fidelity. The report endeavors to document observations, analyze findings, draw insightful conclusions, and delineate the multifaceted implications of signal processing methodologies, especially concerning noise, across varied domains and applications.

5.1.1 Overview of Signal Analysis and System Identification

5.1.1.1 Signal Analysis:

- a. Time and Frequency Domains: Study of signal properties in time (temporal) and frequency (spectral) domains.
- b. Fourier Transform: Use of Fourier analysis to translate signals between time and frequency domains.
- c. Spectral Analysis Techniques: Application of techniques like auto-correlation and coherence functions for analyzing signal properties.

5.1.2 System Identification:

- a. Concepts: Modeling and understanding systems based on observed input-output data.
- b. Parameter Estimation: Estimating system parameters from recorded behavior.

- c. Applications: Application in control systems, adaptive signal processing, and modeling real-world systems.

5.1.3 Brief Explanation of Signal Processing Exercise Objectives

- a. Exploration of Real Signals:

Analysis and identification of signals (Signal_01 and Signal_02) as input and output.

- b. Spectral Power Density (SPD) Calculation:

Computing SPD (S_{xx}) via squared Fourier Transforms at a sampling frequency of 4096 Hz.

- c. Correlation Functions:

Computing and visualizing auto-correlation (ϕ_{xx} , ϕ_{yy}) and cross-correlation (ϕ_{xy}) functions.

- d. Coherence and Cross-Spectral Density:

Determining coherence functions and cross-spectral density (S_{yx}) from cross-correlation and power spectral density.

- e. Fundamental Frequencies:

Identification and specification of fundamental frequencies within the given signals.

5.2 Real Signals Analysis

5.2.1 Signal Loading and Identification

5.2.1.1 Plotting Signals (Signal_01, Signal_02) as Time Signals

5.2.1.1.1 Signal_01

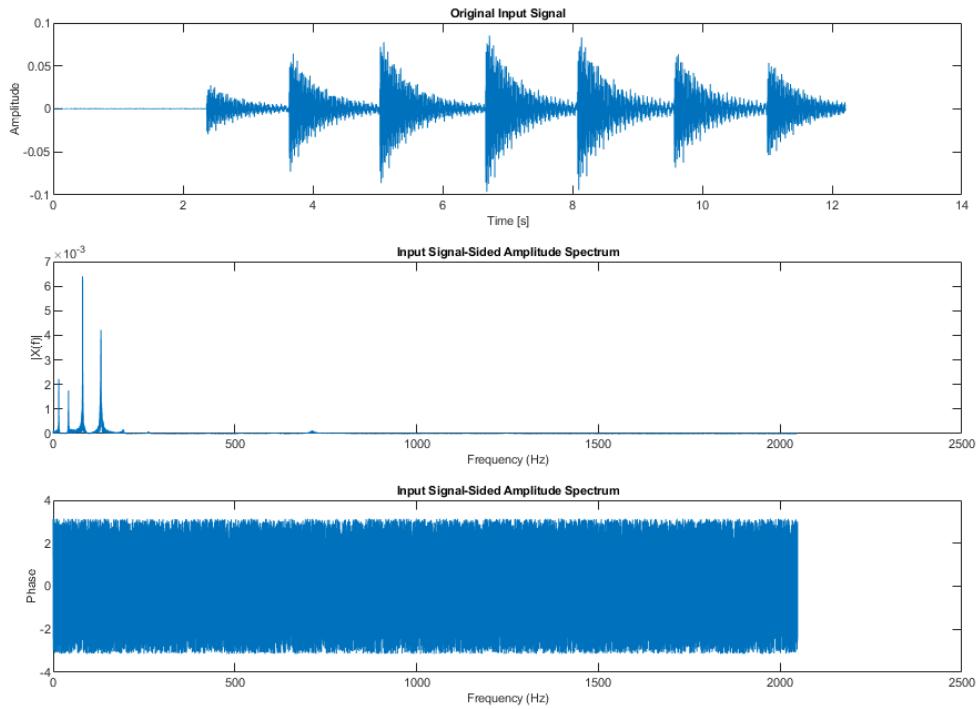


Figure 151: Input – Amplitude & Phase spectrum

5.2.1.1.1 Observation of Input – Amplitude & Phase spectrum:

The graph of the original input signal shows that the oscillations don't follow a perfect pattern. Instead, there's a sudden jump in how strong the oscillations are, followed by a gradual decrease. This behavior is not like the ideal case where oscillations continue smoothly. In this real-world scenario, we notice around four different fundamental frequencies. The phase spectrum, which tells us about the timing of these oscillations, is high for most of these frequencies. This means that the timing relationships between the different oscillations are quite complex. The combination of these features—sudden jumps, gradual decay, and multiple frequencies—highlights how real-world factors, like outside influences and the way energy is lost, can make oscillations more complicated than the ideal, straightforward case. Understanding these details helps us grasp how oscillatory systems behave in the real world.

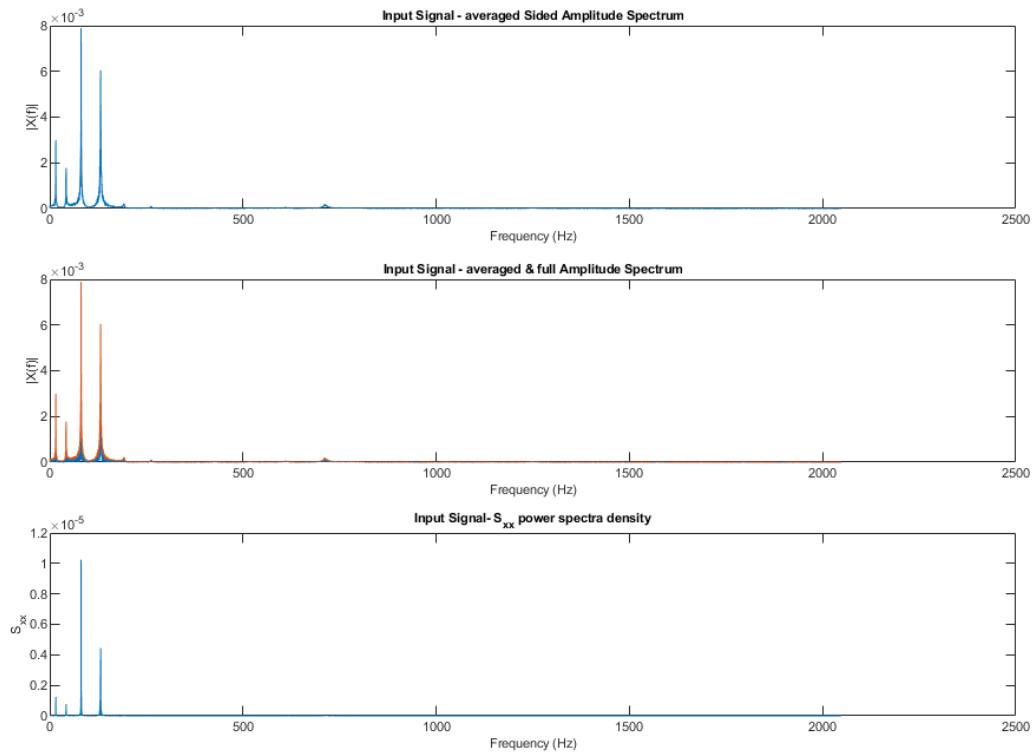


Figure 152: Input – Averaged, Full spectrum & Power spectral density

5.2.1.1.2 *Observation Input – Averaged, Full spectrum & Power spectral density :*

Combining these plots provides a comprehensive analysis of the input signal's frequency characteristics and power distribution. The "Averaged Input Signal-Sided Amplitude Spectrum" offers a smoothed estimate of frequency content, aiding in the identification of dominant frequency components by reducing noise and random fluctuations. Comparing it with the "Input Signal - Averaged & Full Amplitude Spectrum" visually demonstrates the impact of averaging on spectrum smoothness, elucidating how noise reduction enhances the visibility of significant frequency peaks. Meanwhile, the "Input Signal Power Spectra Density" plot reveals the distribution of power across frequencies, pinpointing dominant features through peaks in power.

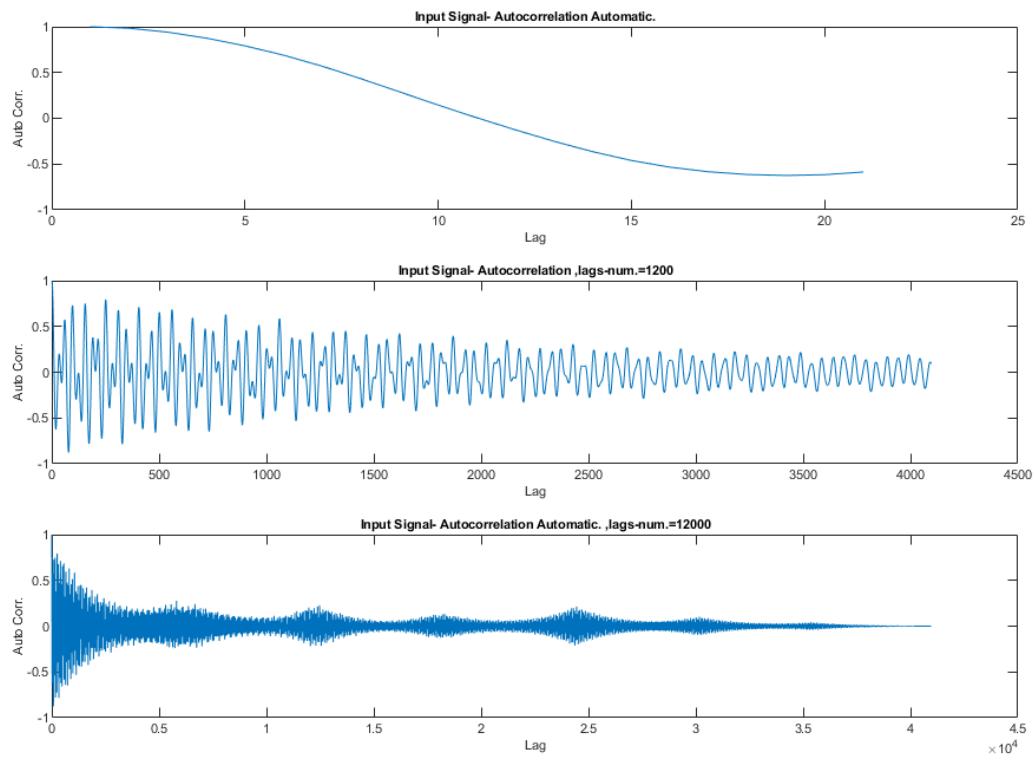


Figure 153: Input – Autocorrelation with lags

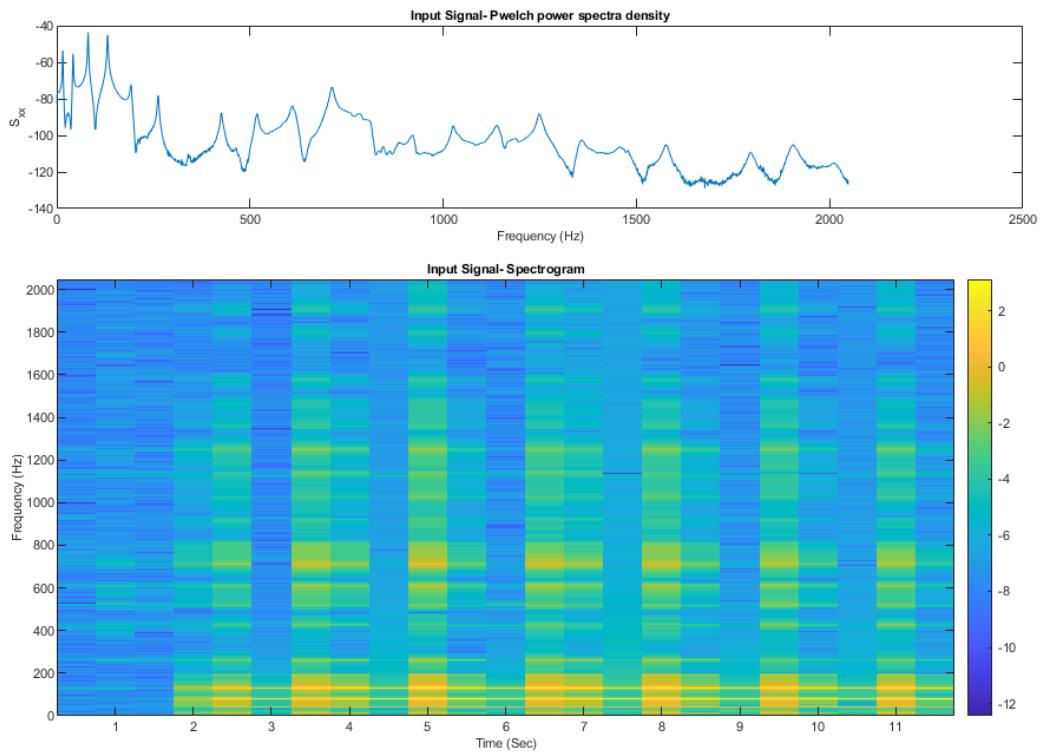


Figure 154: Input – Pwelch & Spectrogram

Observation: The plot in Figure 154 indicates the power spectral density of the input signal with “Pwelch” function using MATLAB. This function also allows the user to plot the signal Spectrogram, i.e., the Frequency vs Time plot of the signal.

The Figure 153 describes the Autocorrelation of the input signal “Signal_01” with some lags extending from 1200 till 12000. This allows us to study the effect of amplitude on the graph.

5.2.1.2 Signal_2

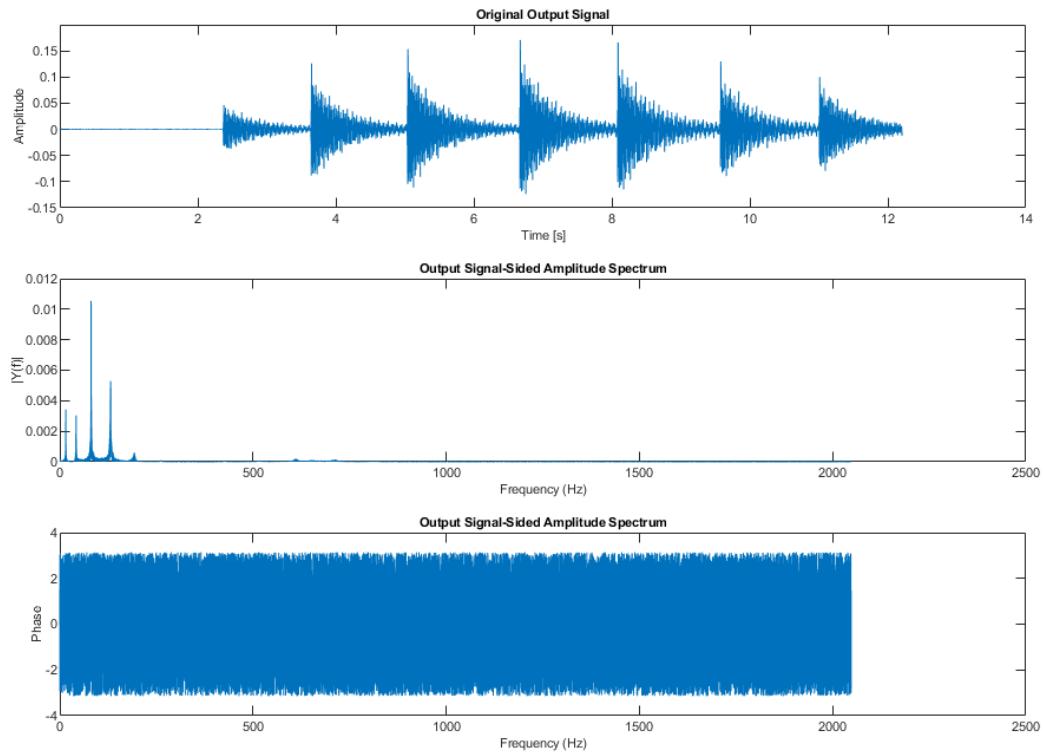


Figure 155: Output – Amplitude & Phase spectrum

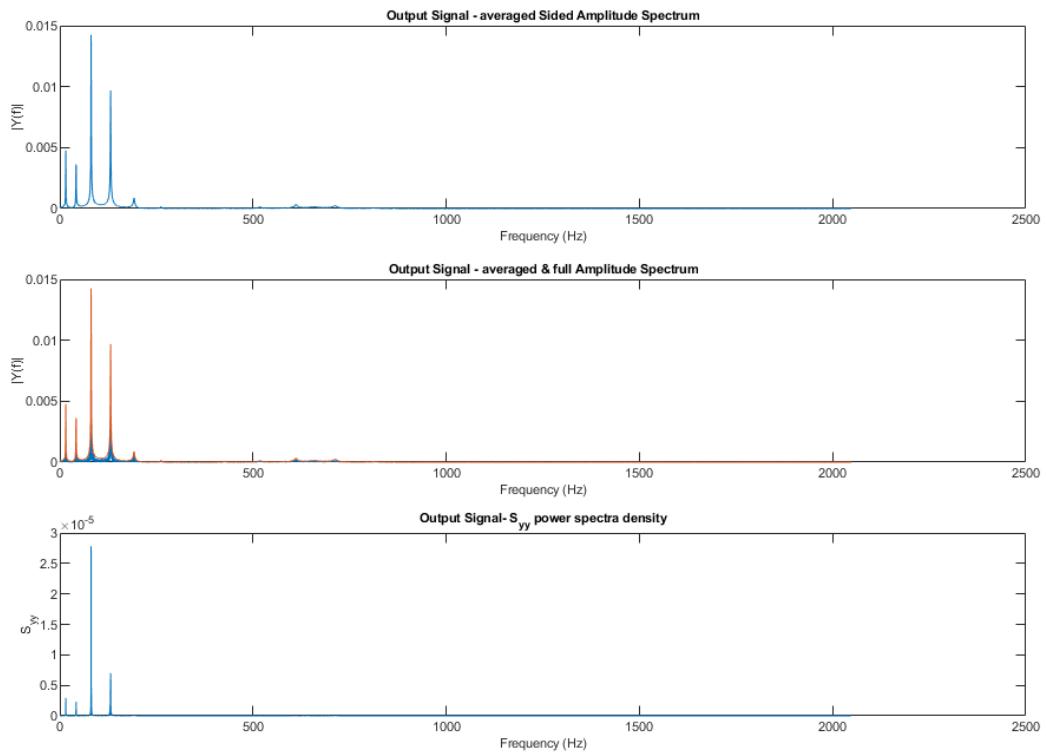


Figure 156: Output – Averaged, Full spectrum & Power spectral density

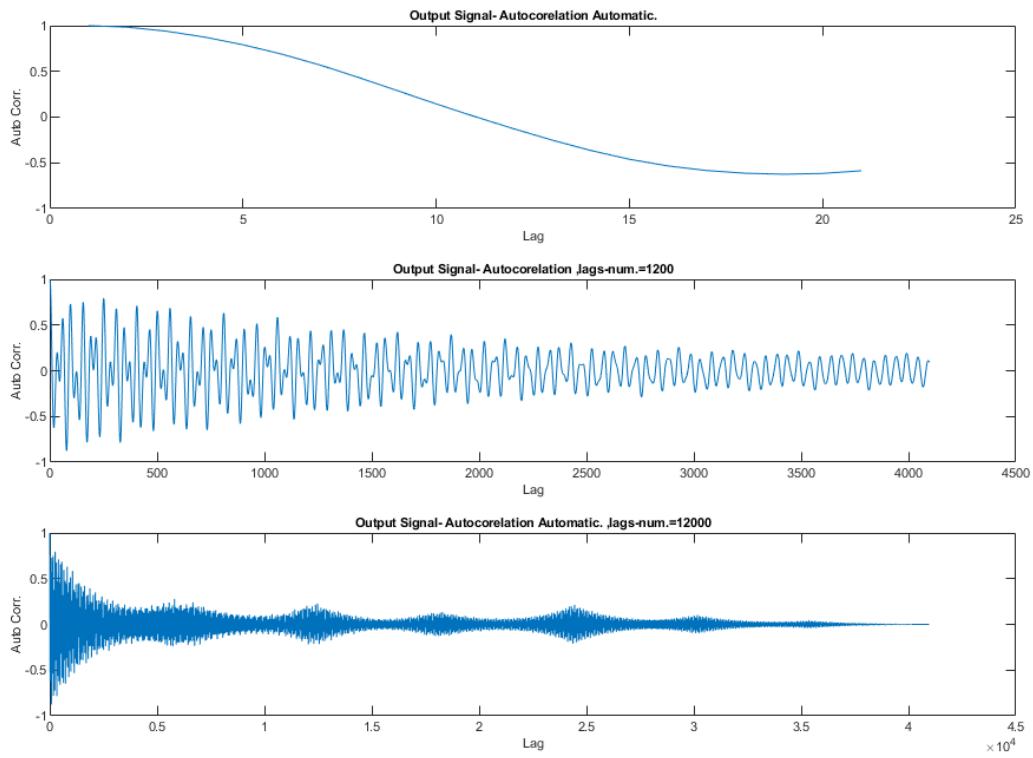


Figure 157: Output – Autocorrelation with lags

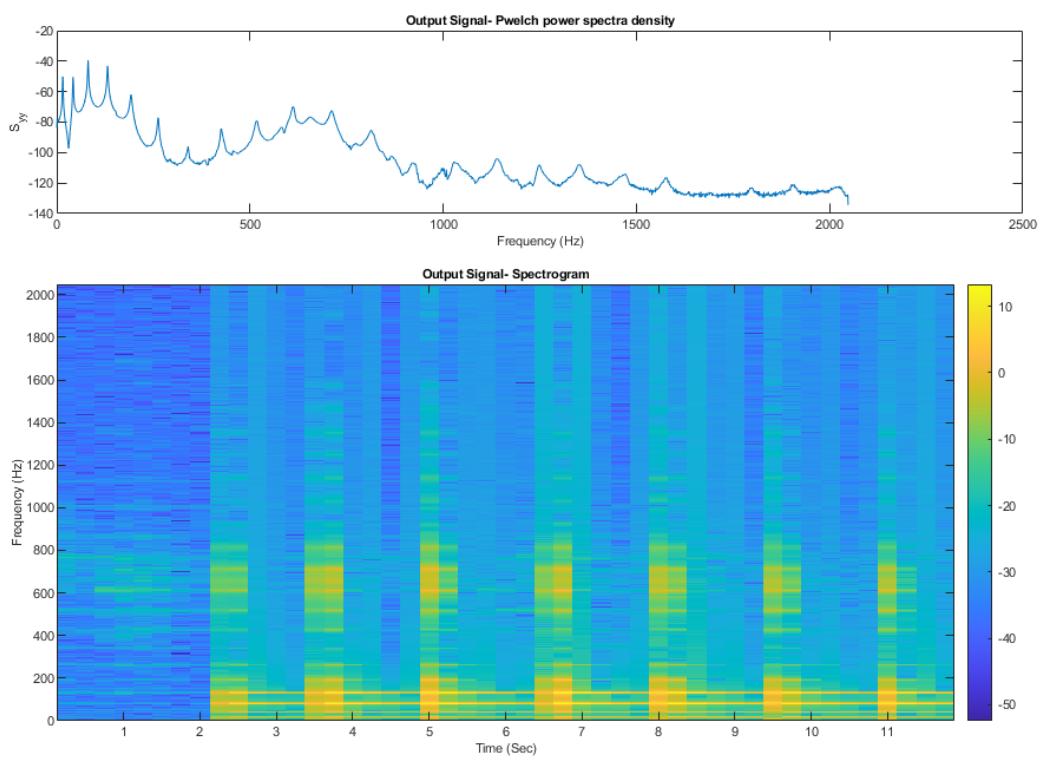


Figure 158: Output – Pwelch & Spectrogram

5.2.1.3 Identification of Input (x) and Output (y) Signals

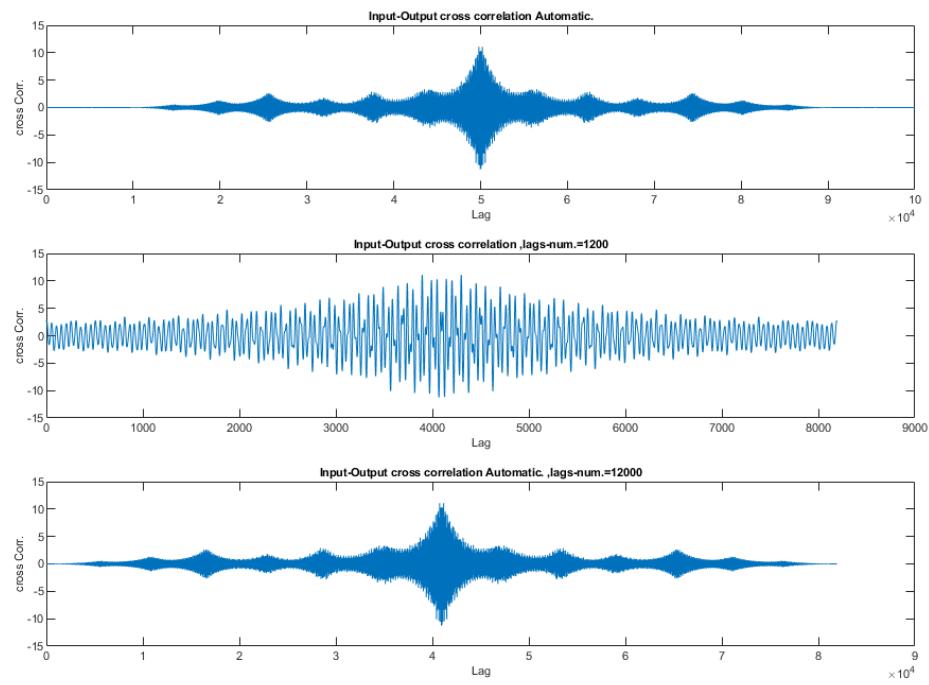


Figure 159: Input-Output Cross correlation

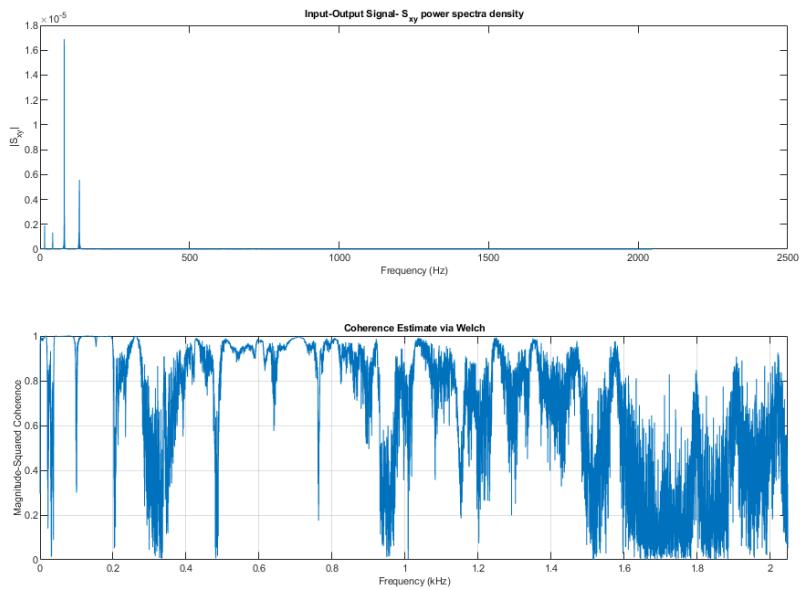


Figure 160: Input-Output –Power spectral density & Coherence Pwelch

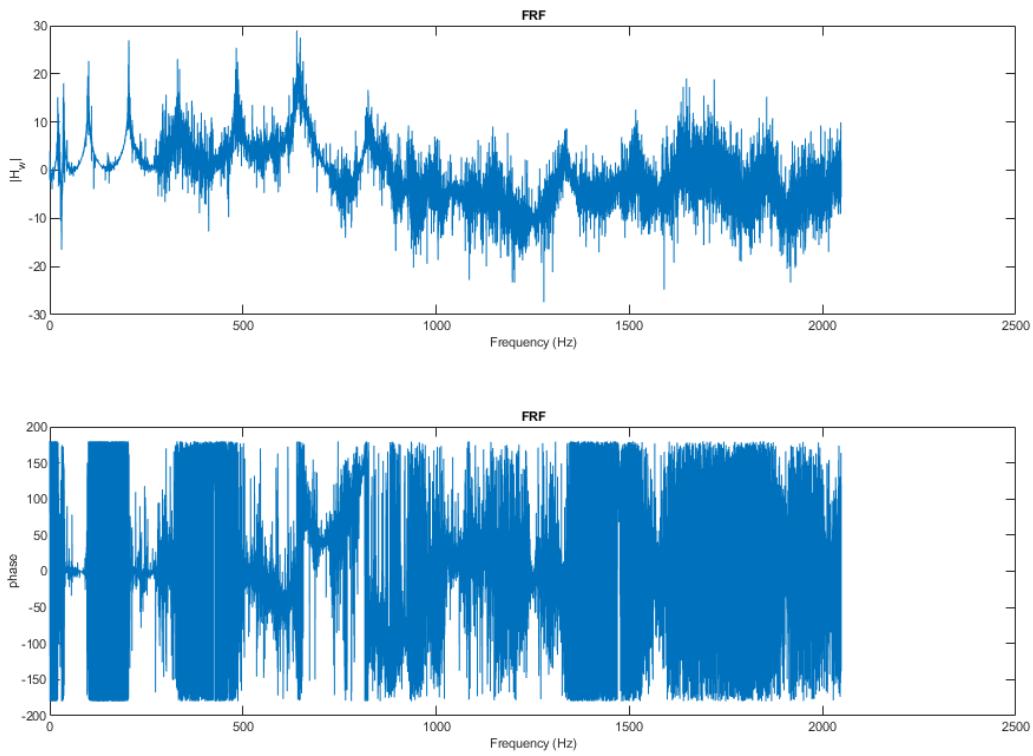


Figure 161: Input-Output FRF plot

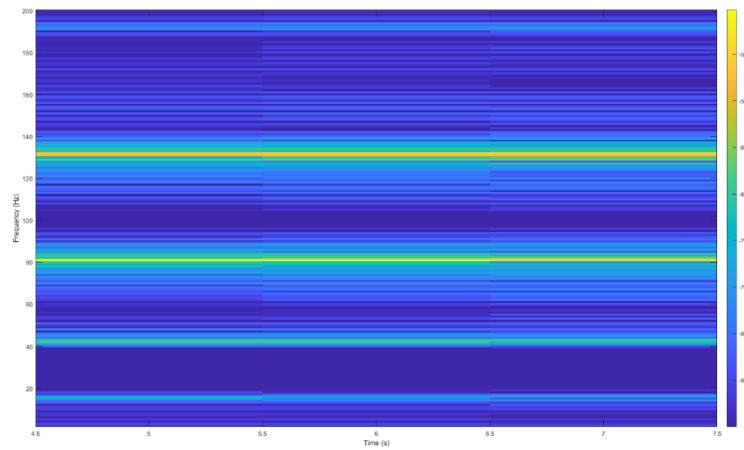


Figure 162: Input-Output Spectrogram

5.3 Spectral Power Density (SPD) Calculation

5.3.1 Computing Average Squared Fourier Transforms (S_{xx})

The Spectral Power Density S_{xx} is calculated by taking the average of squared Fourier Transforms.

$$S_{xx}(f) = \frac{1}{N} \sum_{k=1}^N |X_k(f)|^2$$

Where N is the total number of segments and $X_k(f)$ is the Fourier Transform of the k segment.

5.3.2 Auto-Correlation Function

5.3.2.1 Computing and Plotting ϕ_{xx} and ϕ_{yy}

The Auto-Correlation Function ϕ_{xx} is computed by taking the inverse Fourier Transform of the product of the Fourier Transform of the signal $X(f)$ with its complex conjugate $X^*(f)$:

$$\phi_{xx}(\tau) = \mathcal{F}^{-1}\{X(f) \cdot X^*(f)\}$$

Similarly for ϕ_{yy} .

5.3.3 Cross-Correlation and Cross-Spectral Density

5.3.3.1 Computing and Plotting ϕ_{xy} and S_{yx}

The Cross-Correlation Function ϕ_{xy} between signals $X(t)$ and $Y(t)$ is obtained by taking the inverse Fourier Transform of the product of their Fourier Transforms:

$$\phi_{xy}(\tau) = \mathcal{F}^{-1}\{X(f) \cdot Y^*(f)\}$$

Similarly, the Cross-Spectral Density S_{yx} is computed as:

$$S_{yx}(f) = X(f) \cdot Y^*(f)$$

5.3.4 Coherence Function

5.3.4.1 Computing and Plotting $\gamma^2(\omega)$

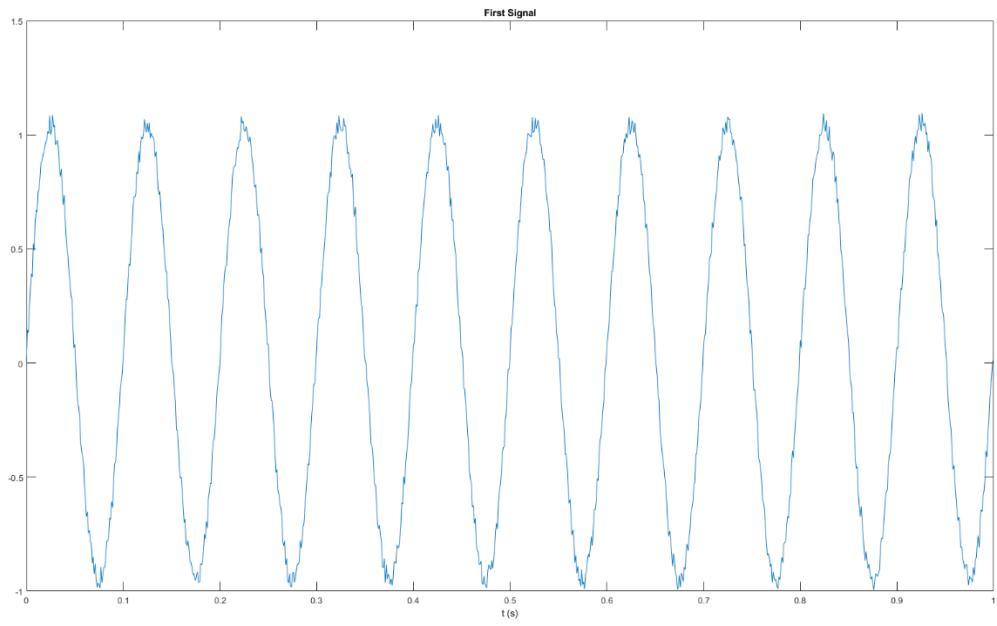


Figure 163: First Signal plot

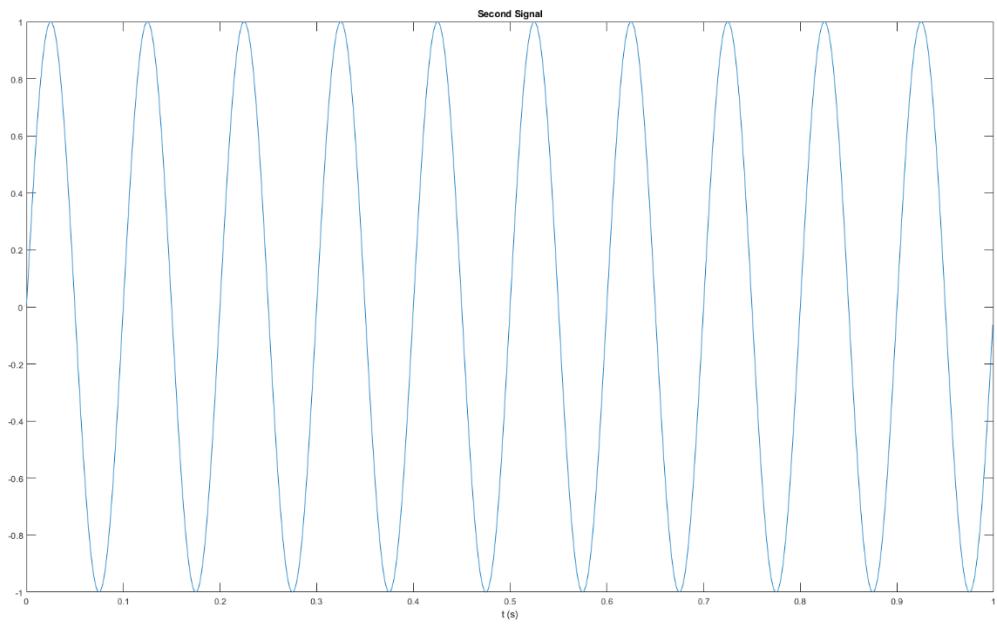


Figure 164: Second Signal plot

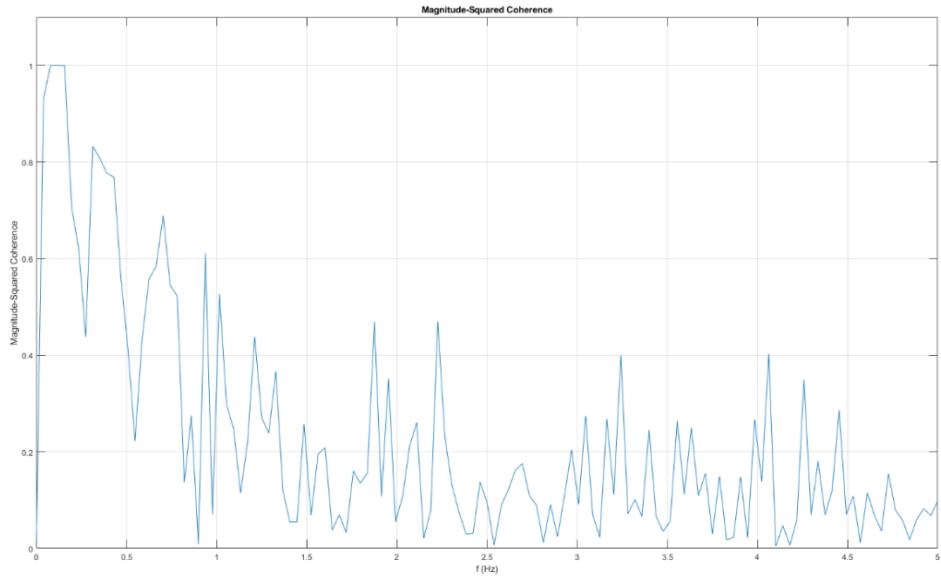
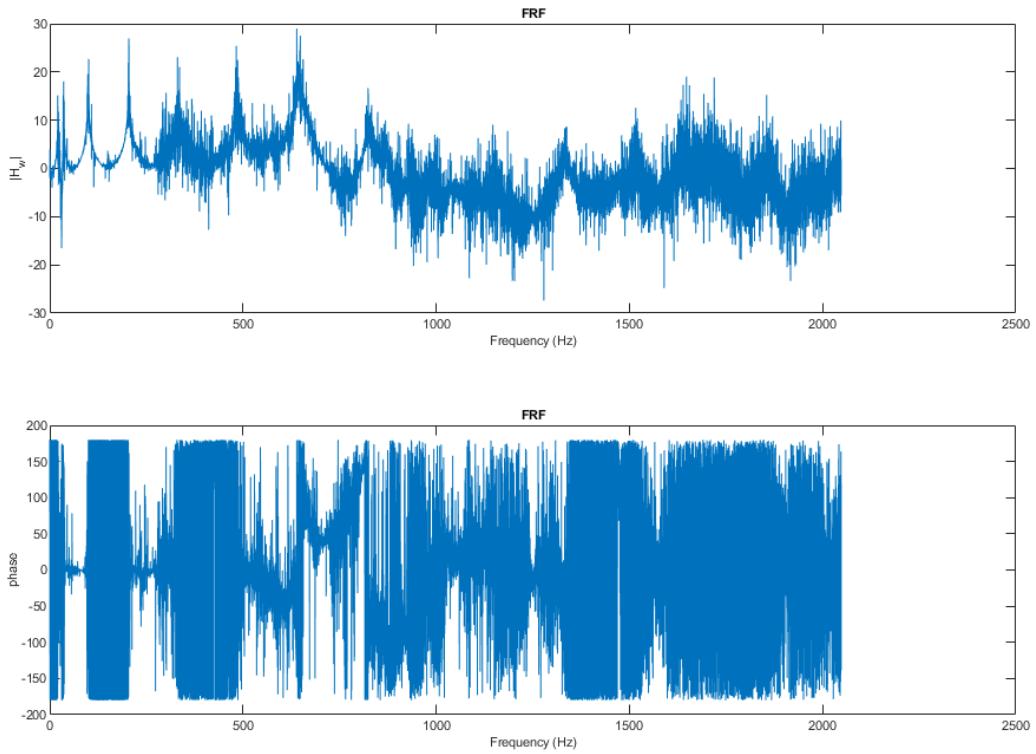


Figure 165: Magnitude-Squared Coherence

5.3.4.2 Specifying Fundamental Frequencies



5.4 Coherence Function in Differential Equations

5.4.1 Noisy Sinusoidal Functions

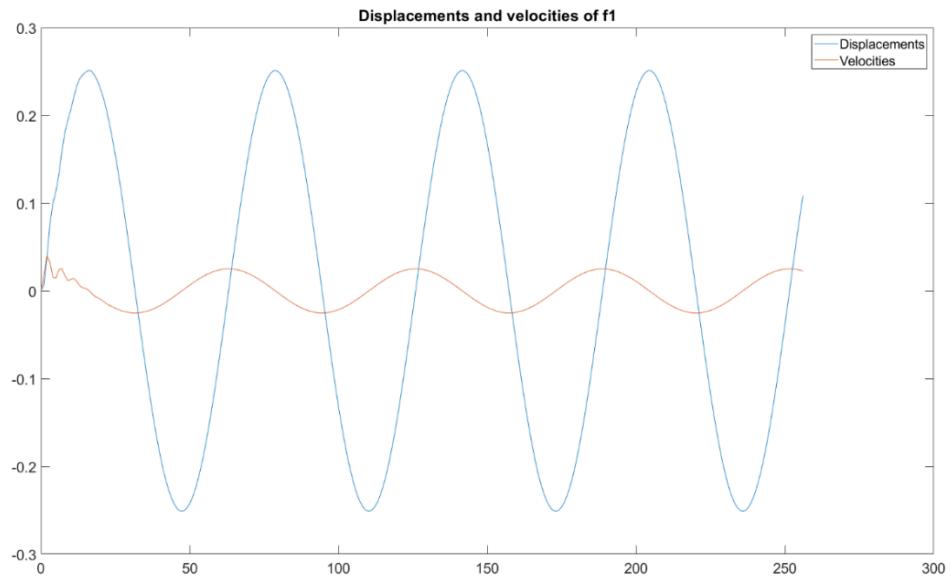


Figure 166: Displacement and velocity plot for f_1 with $k = 1$

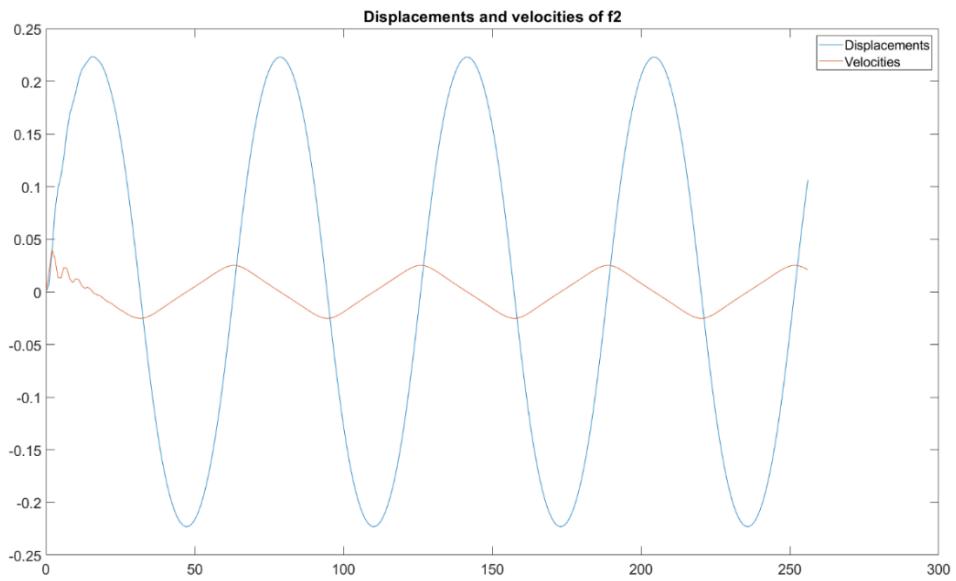


Figure 167: Displacement and velocity plot for f_2 with $k = 1$

5.4.2 Computing Magnitude-Squared Coherence

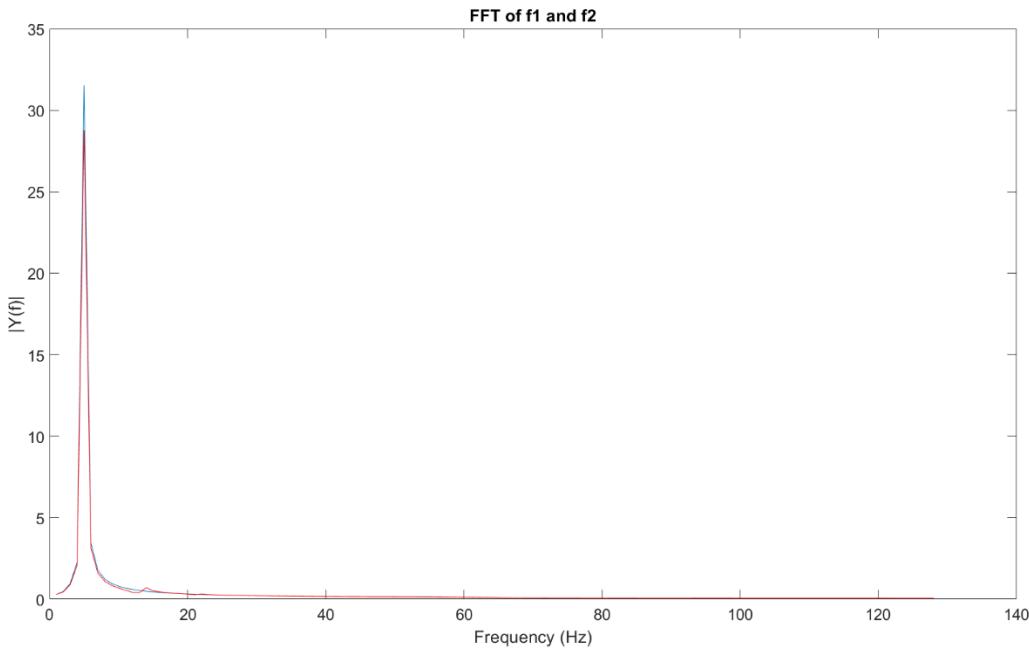


Figure 168: FFT of f_1 and f_2 signal with $k = 1$

It is observed that the signals which are changing in terms of frequency, but the FFT fails to detect that in Figure 168, and it is also proved in the coherence part as in Figure 169, Figure 170.

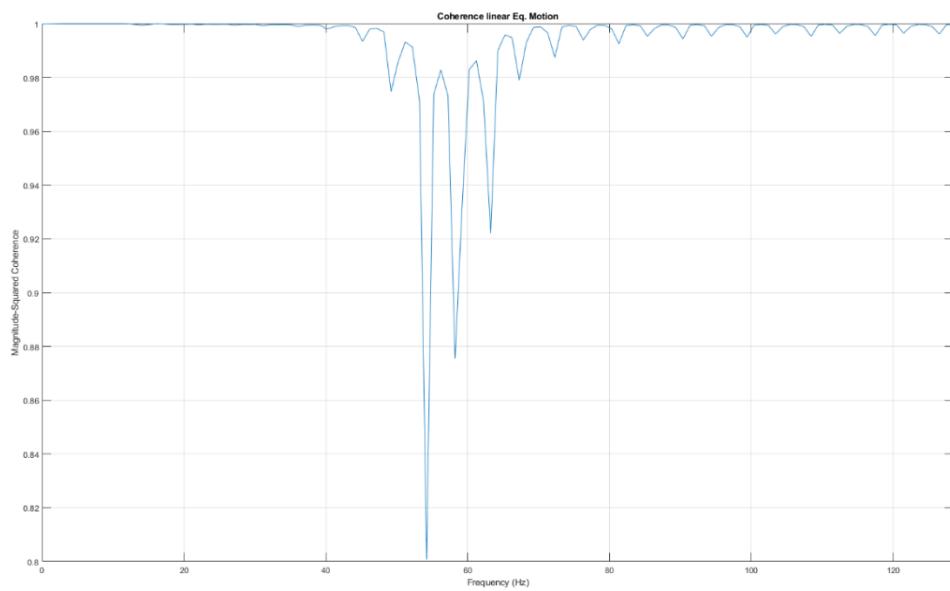


Figure 169: Magnitude-Squared Coherence – linear eq. Motion with $k = 1$

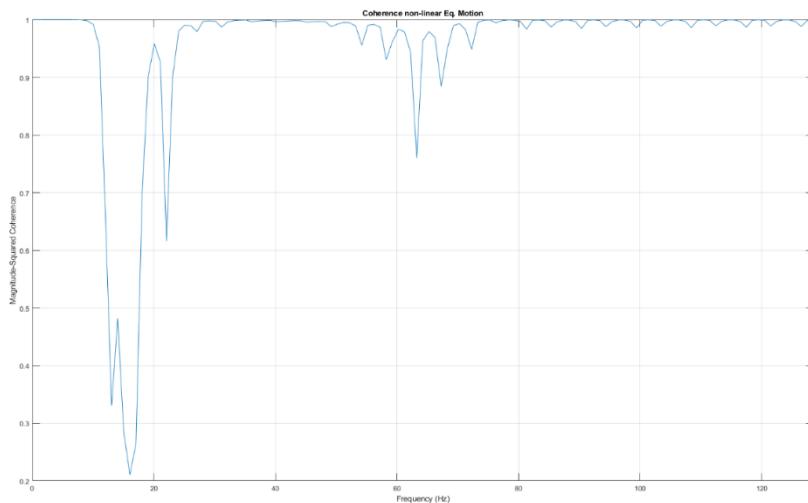


Figure 170: Magnitude-Squared Coherence – non-linear eq. Motion with $k = 1$

5.5 Wavelet Transformation

5.5.1 Chirp Signal Analysis

The transformation is used to enlarge the high and low frequency of the new function. Wavelet Transformation can be used to represent the square-integrable function in mathematics and also used to analyze the data over multiple scales. Now, the “wavelet_transform_chirp.m” is executed to determine the wavelet transform of “Cos(4x²)” at level 5.

5.5.1.1 Discrete Wavelet Transform (DWT) at Level 5 using Haar/db1 Wavelet

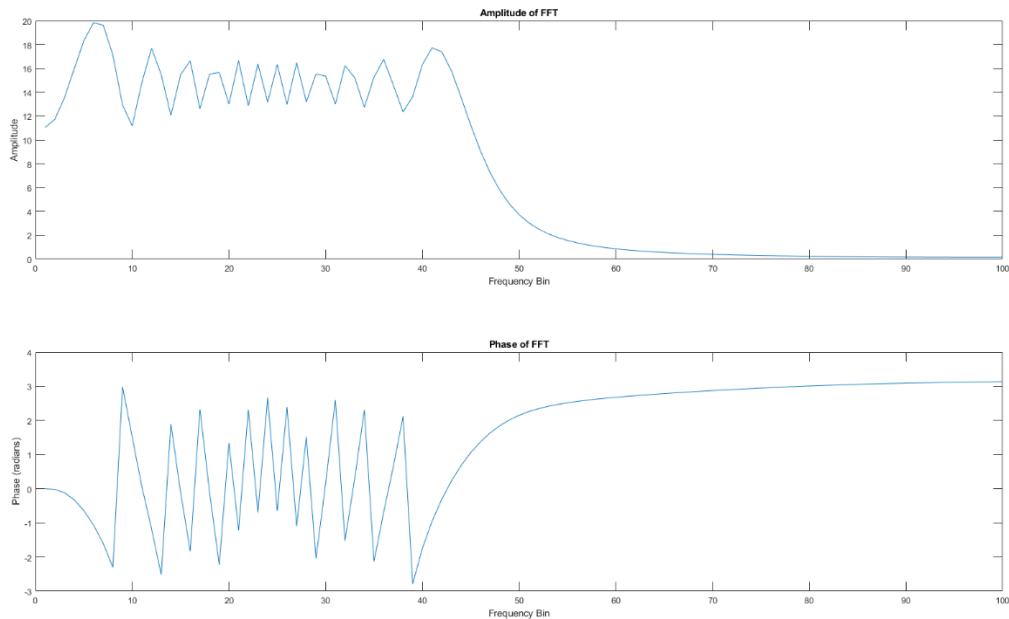


Figure 171: FFT - Amplitude and Phase of $\text{Cos}(4x^2)$

The generated plots reveal a notable trend in frequencies, showcasing higher values at the outset that progressively diminish towards the ending.

5.5.1.2 Coefficient Expansion and Signal Plotting

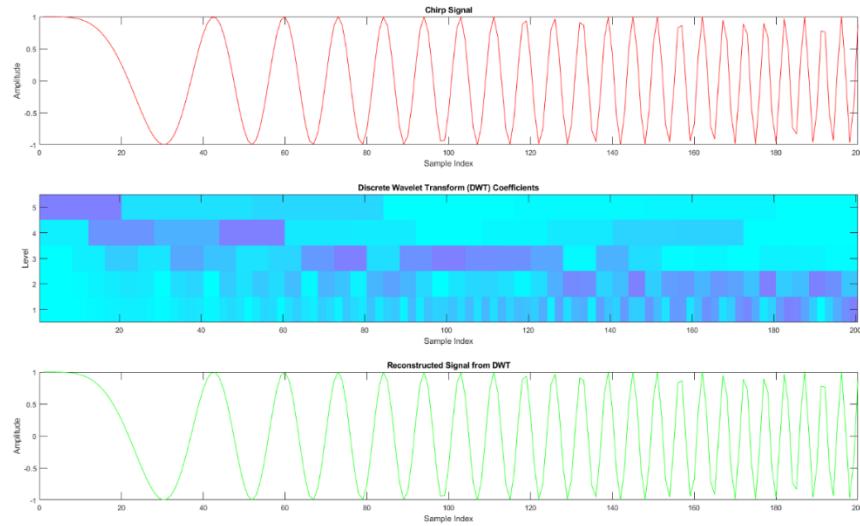


Figure 172: Signal Plot and Expanded Discrete Wavelet Coefficient At Level 5

5.5.1.3 Expanding Discrete Wavelet Coefficients and Signal Plot

The script "wavelet_transform_sins_added_concatenated.m" is executed to combine and concatenate signals $\sin(2\pi \cdot 10 \cdot N \cdot dt \cdot x)$ and $\sin(2\pi \cdot 20 \cdot N \cdot dt \cdot x)$. Subsequently, the script performs wavelet and FFT transforms on the combined signal at level 5. This process provides insights into the frequency content and temporal dynamics of the concatenated signals.

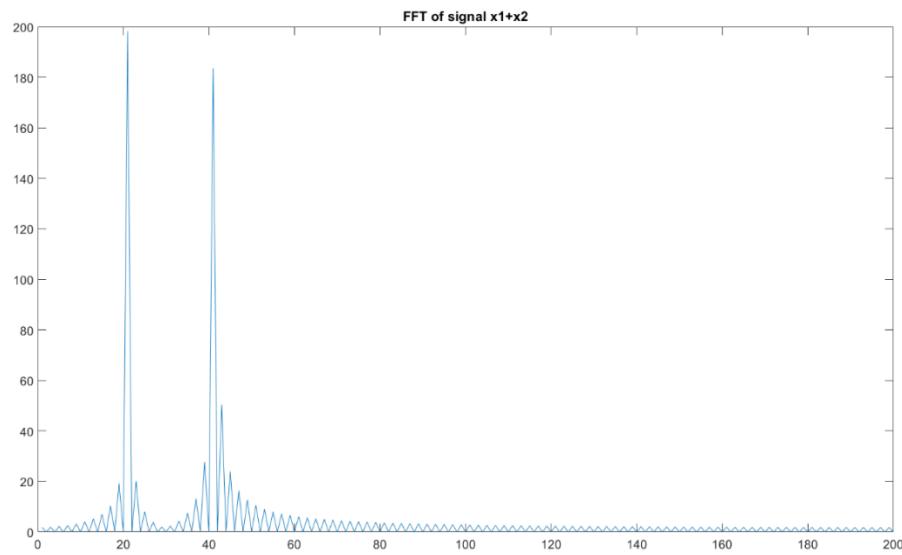


Figure 173: FFT of Signal $X1 + X2$

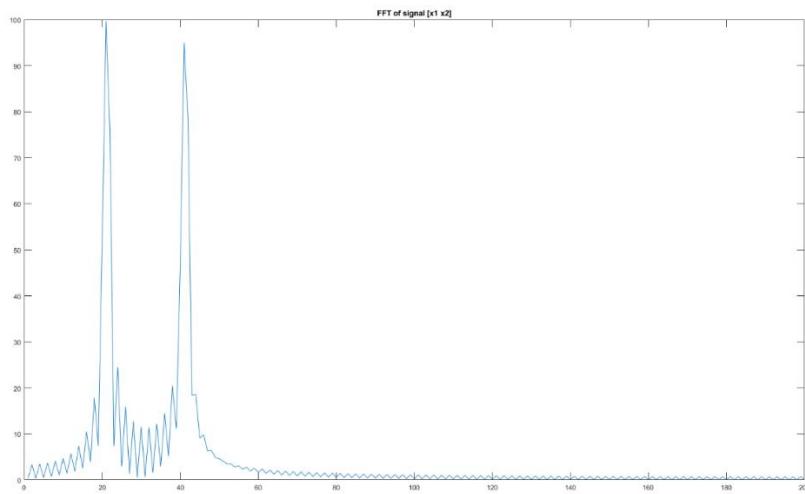


Figure 174: FFT of signal $[X1 \ X2]$

5.5.1.4 Analysis of Added and Concatenated Sine Signals

The utilization of wavelet transform facilitates the identification of frequency patterns. The results obtained from the Discrete Wavelet Transform (DWT) reveal:

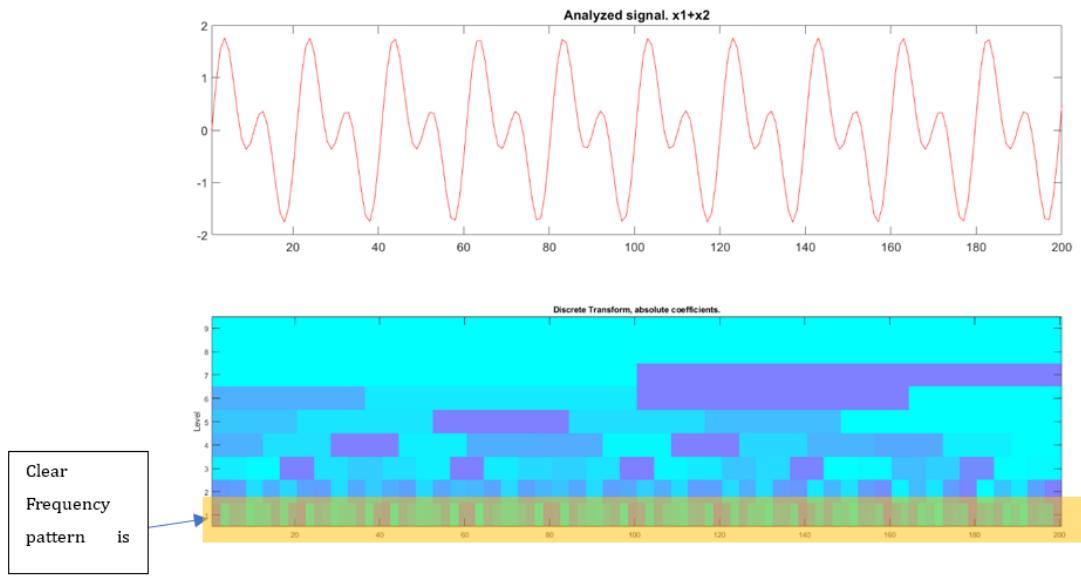


Figure 175: Analyzed Signal $X1 + X2$

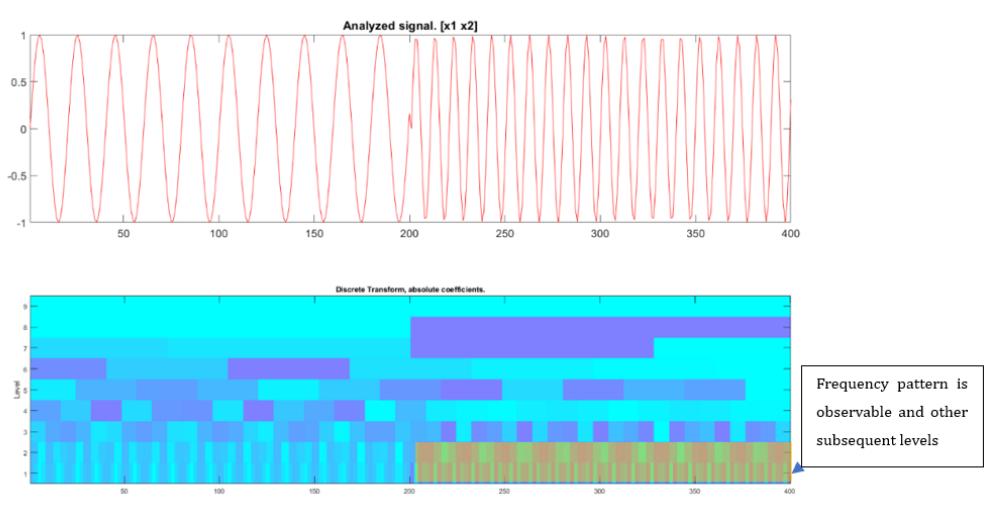


Figure 176: Analyzed Signal $[X1 \ X2]$

5.5.1.5 Observation:

Examining the graphical representations further, it becomes evident that the act of concatenating signals $x1$ and $x2$ maintains the distinct characteristics of each signal without

altering their inherent natural frequencies. On the other hand, when the signals are added together, a discernible influence on the overall natural frequencies is observed. This divergence in behavior emphasizes the importance of signal manipulation techniques, showcasing how different operations can either preserve or alter the underlying dynamics of the signals. The choice between concatenation and addition depends on the specific analytical or interpretative goals, as each operation captures different aspects of the system's behavior.

5.6 Filter Analysis

The function "filterOfNoisySine.m" is employed to apply a low-pass filter to a highly noisy sinusoidal signal $0.1 \cdot \sin(2\pi \cdot 150 \cdot t)$. The variable N represents the frequency of filter application, and different values are explored to generate plots from Figure 5.29 to 5.34. Filters play a crucial role in signal processing by eliminating unwanted noise and preserving essential signals. This is achieved by removing specific frequency bands. Various filter types are available based on their applications, and they find significance in fields such as image processing and signal recordings. The applied low-pass filter at frequencies 25, 50, and 100 is depicted in the plots below.

5.6.1 Low Pass Filter Application on Noisy Sinusoidal Signal

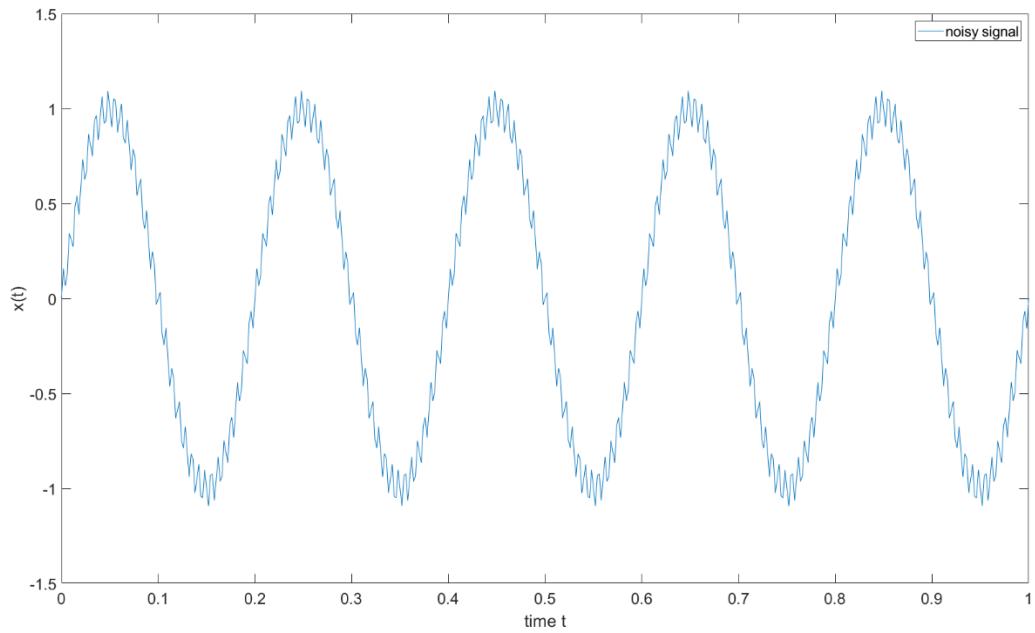


Figure 177: Original Input noisy Signal

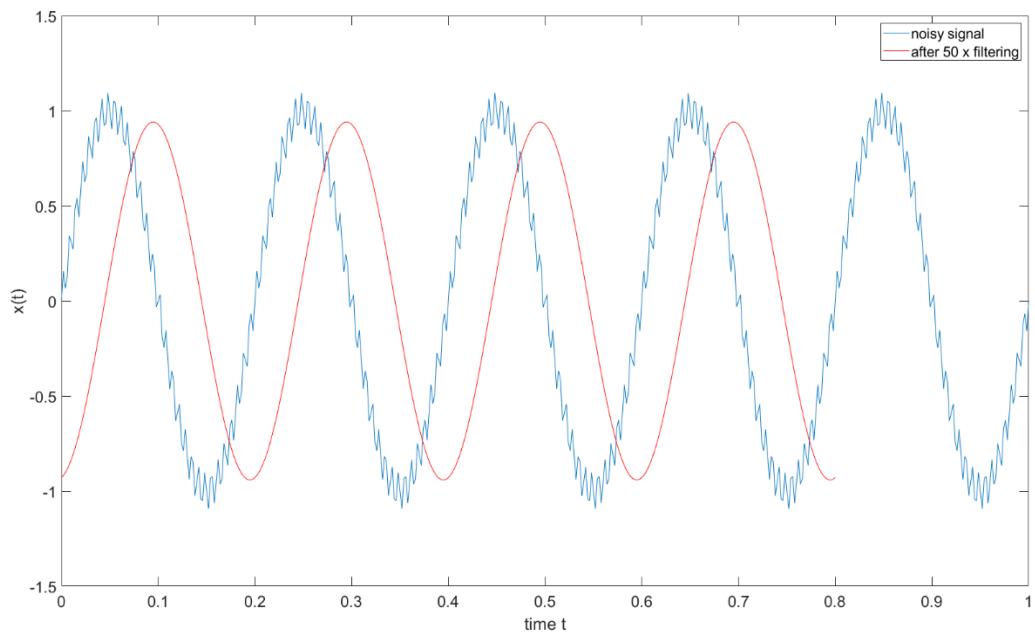


Figure 178: After $N=50$ filtering (Red curve)

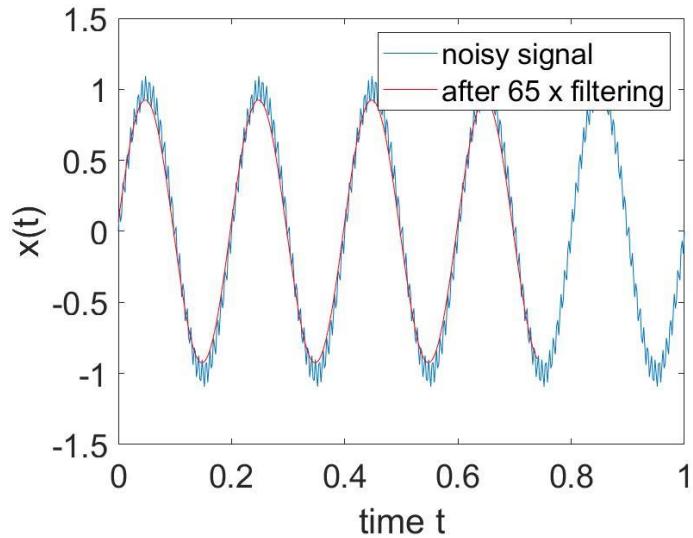


Figure 179: After $N=65$ filtering (Red curve)

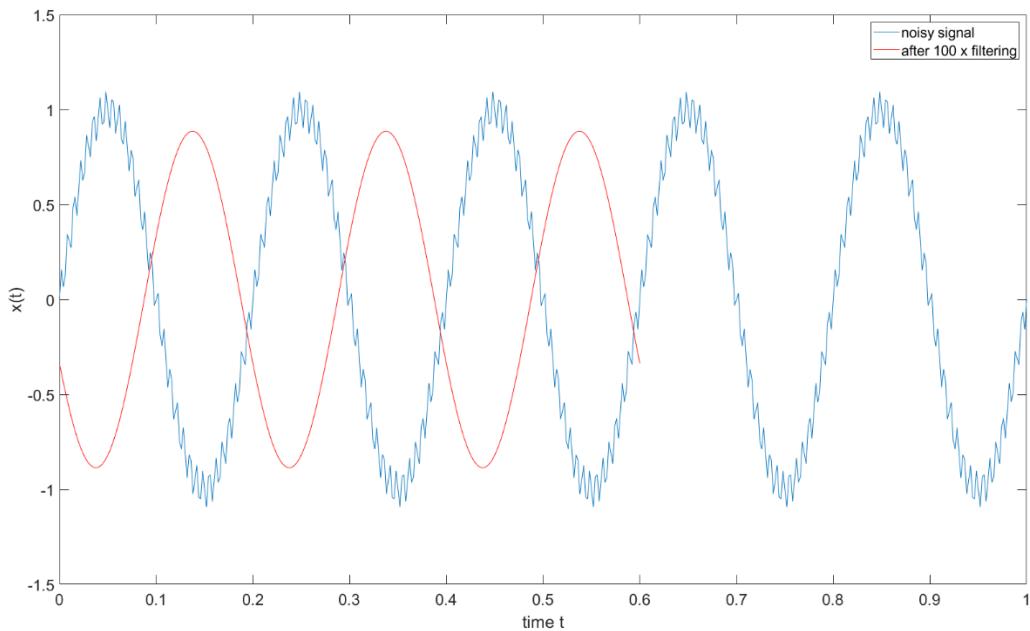


Figure 180: After $N=100$ filtering (Red curve)

5.6.2 Applying Filter and Interpretation of Results

The plots above reveal that employing $N = 65$ effectively eliminates the noise, significantly improving the overall quality of the signal waves (depicted by the red curve in Figure 179).

However, an application of low-pass filters introduces an undesired effect – a noticeable phase shift to the right or left, as illustrated in the Figure 178 and Figure 180.

6. Experiment:6- Damage Detection

In this exercise, our work investigates the impact of damage location and severity on a cantilever beam with specific geometric properties. The vertical displacements resulting from beam vibration are obtained using the function **Cant_damage.p**, with measurements taken at various coordinates along the beam's length. The script **Run_Cant_damage.m** facilitates the execution of this function.

6.1 Un-damaged Beam Response and Natural Frequencies

The investigation commences with an exploration of the undamaged beam, exhibiting three distinct displacement positions in a 2D frame structure. *Figure 181* and *Figure 182* provide a comprehensive view of the 2D cantilever beam and its various mode shapes, showcasing the inherent structural dynamics.

Subsequently, the script "Run_Cant_damage.m" is executed, initiating FFT analysis on the beam's 10m span. Notably, the observed natural frequencies at the free end manifest as distinct peaks in Figure 6.3, with values of 3.37, 21, 58.87, and 115.25. This analysis unveils the resonant frequencies inherent to the undamaged beam.

Extending the examination, *Figure 183*, *Figure 184*, *Figure 185* and *Figure 186* delineate the natural frequencies at specific coordinates (2m, 3m, 5m, and 6m) along the beam. This systematic exploration aims to elucidate variations in natural frequencies corresponding to different measurement points, providing valuable insights into the structural behavior of the undamaged beam.

In summary, this comprehensive analysis, integrating mode shapes, FFT analysis, and natural frequencies at various positions, lays the groundwork for a nuanced understanding of the undamaged beam's dynamic response and resonant characteristics.

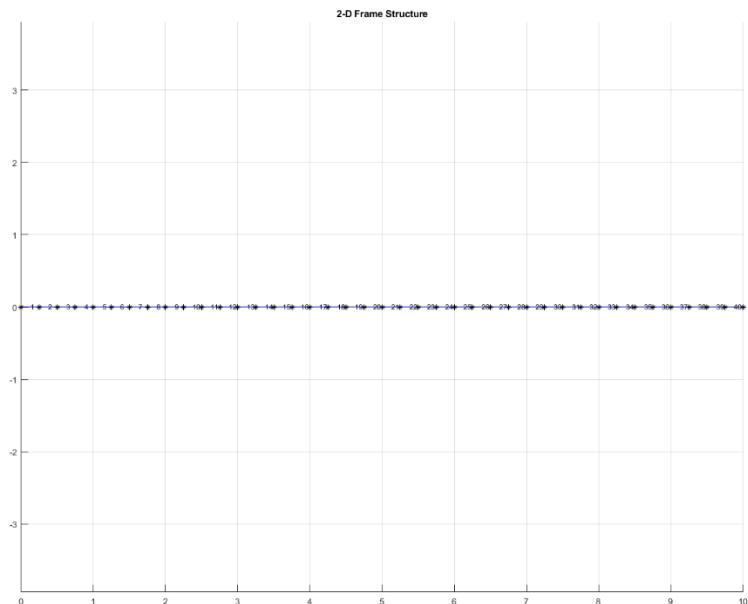


Figure 181: Undamaged Cantilever beam

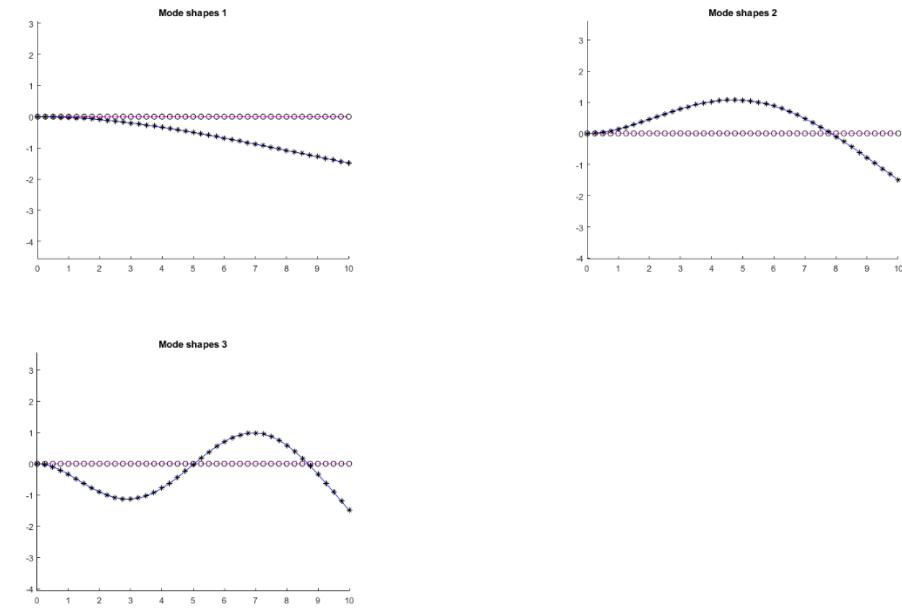


Figure 182: Different modes shape

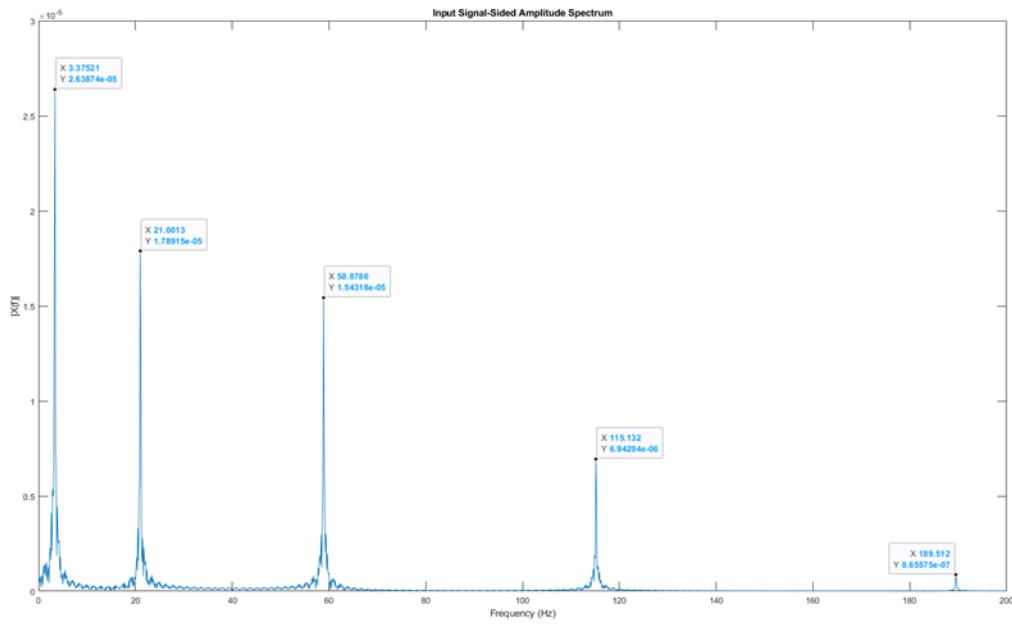


Figure 183: Undamaged at 2m

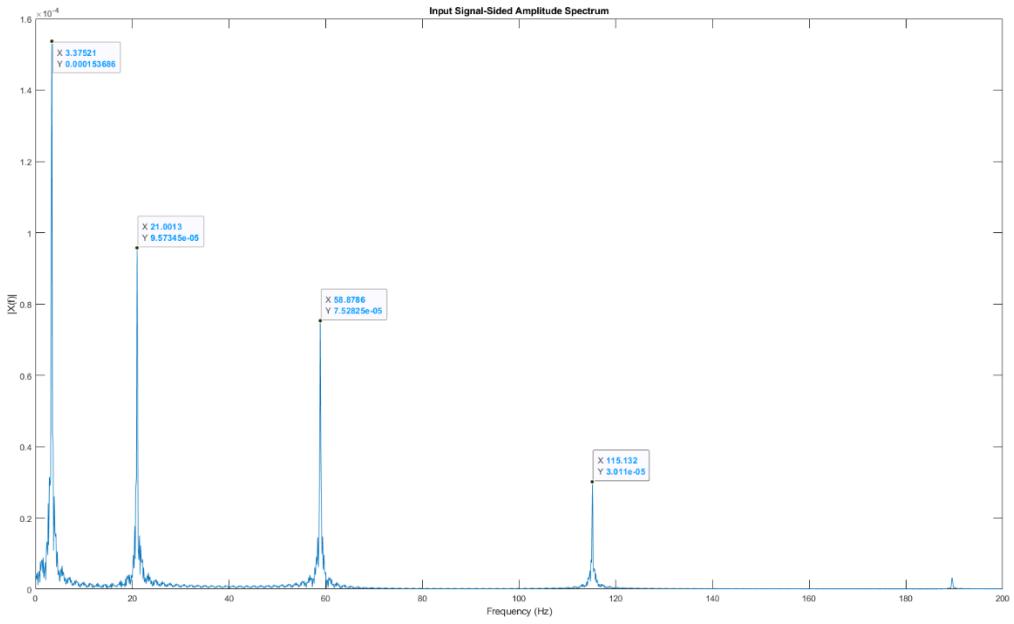


Figure 184: Umdamaged at 3m

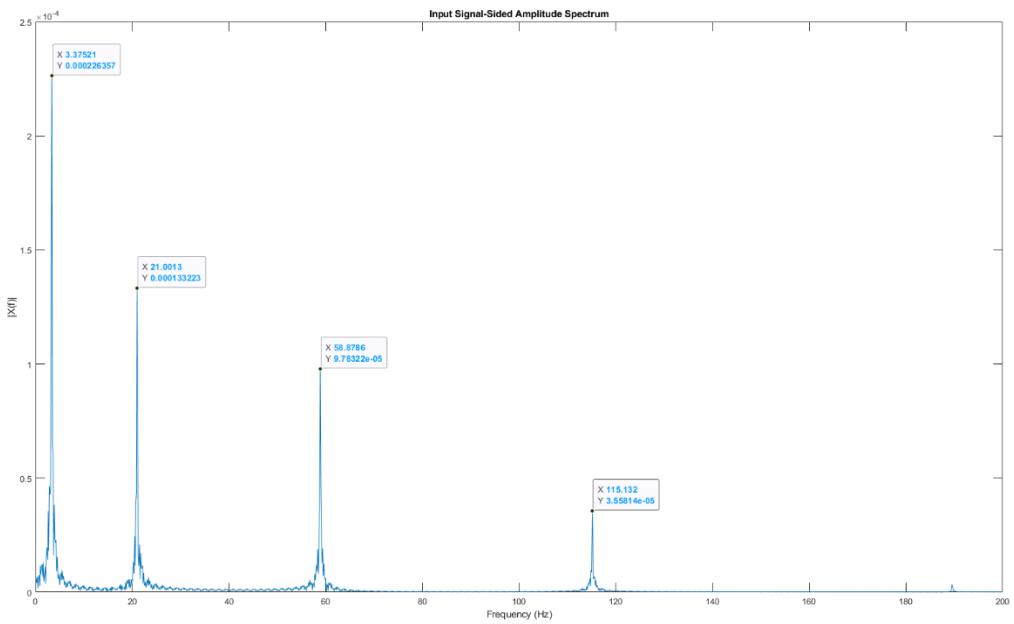


Figure 185: Undamaged at 5m

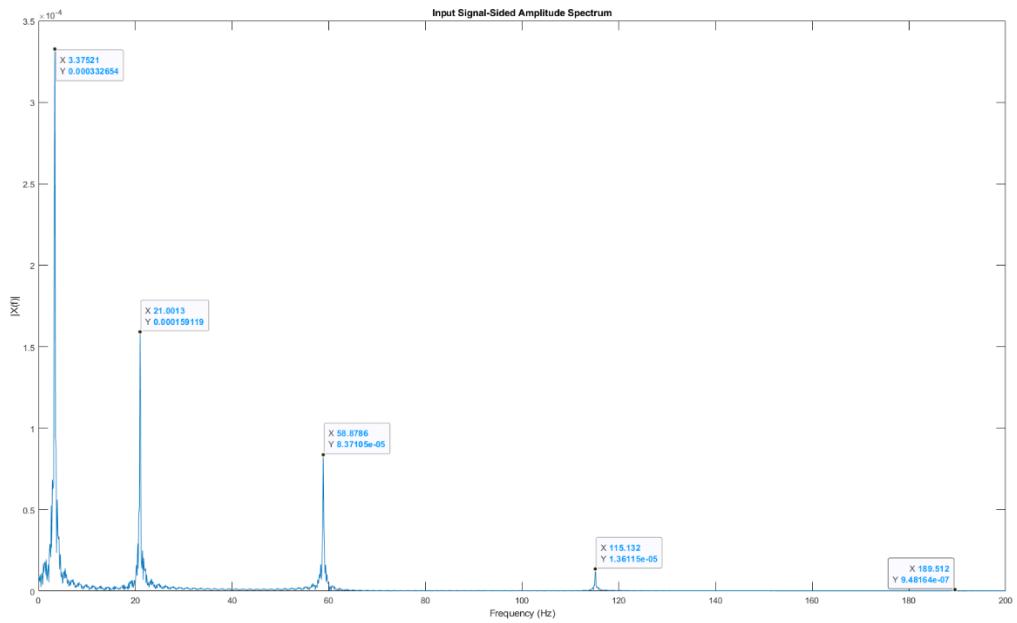


Figure 186: Undamaged at 6m

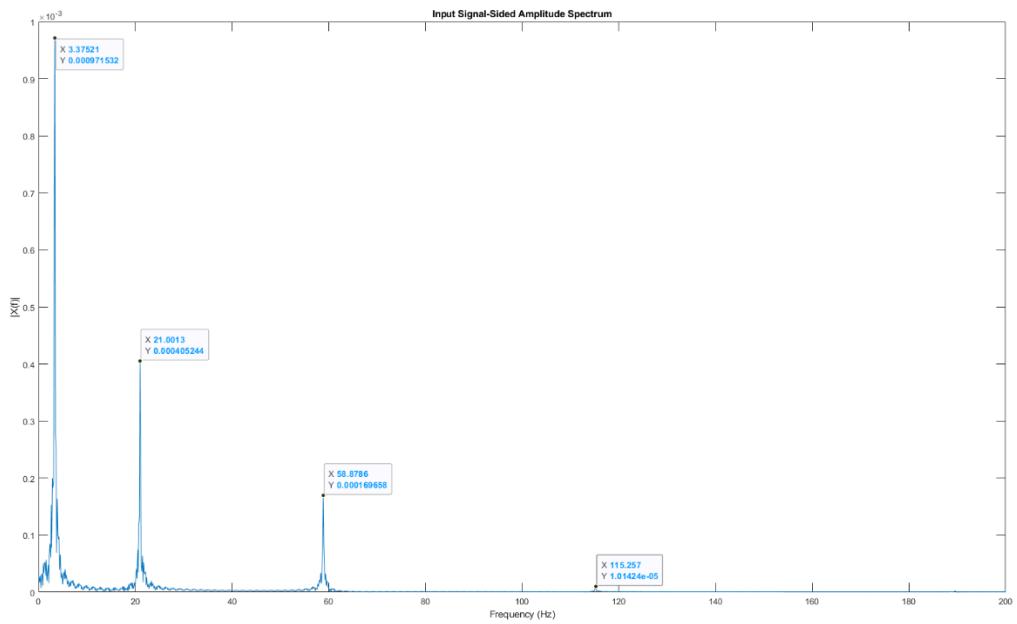


Figure 187: Undamaged at 10m

6.1.1 Observation:

The examination consistently and conclusively demonstrated a notable finding: the natural frequency of an undamaged beam remains uniform and unaltered throughout the entire span of its length. This observation implies that, under the given conditions and without the presence of any structural damage, the inherent vibrational characteristics of the beam exhibit stability and consistency, with no discernible variations in frequency across different segments of its length. This finding holds significance in understanding the fundamental resonance behavior of undamaged beams and provides valuable insights for structural analysis and engineering applications.

6.2 Effect of Damage Location and Severity

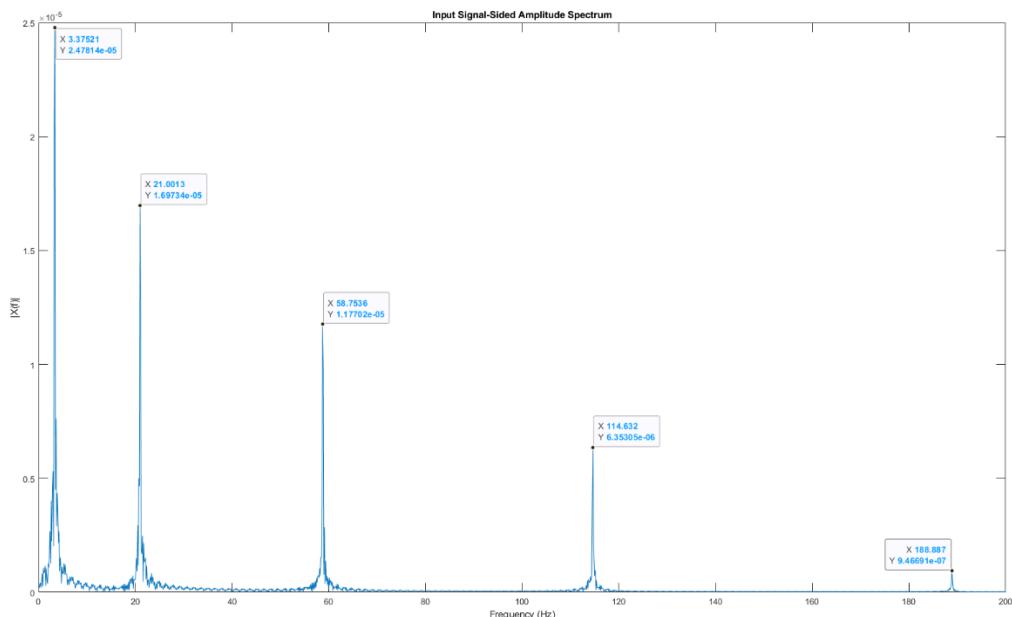


Figure 188: 0.2 position of damage location, with severity of 0.1 for 2m beam

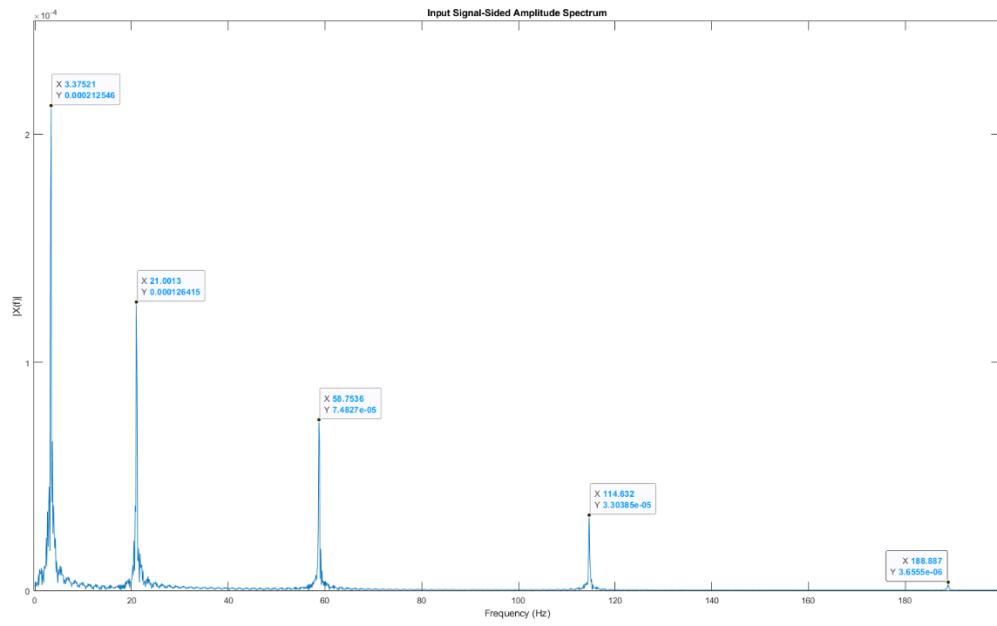


Figure 189: 0.2 position of damage location, with severity of 0.1 for 5m beam

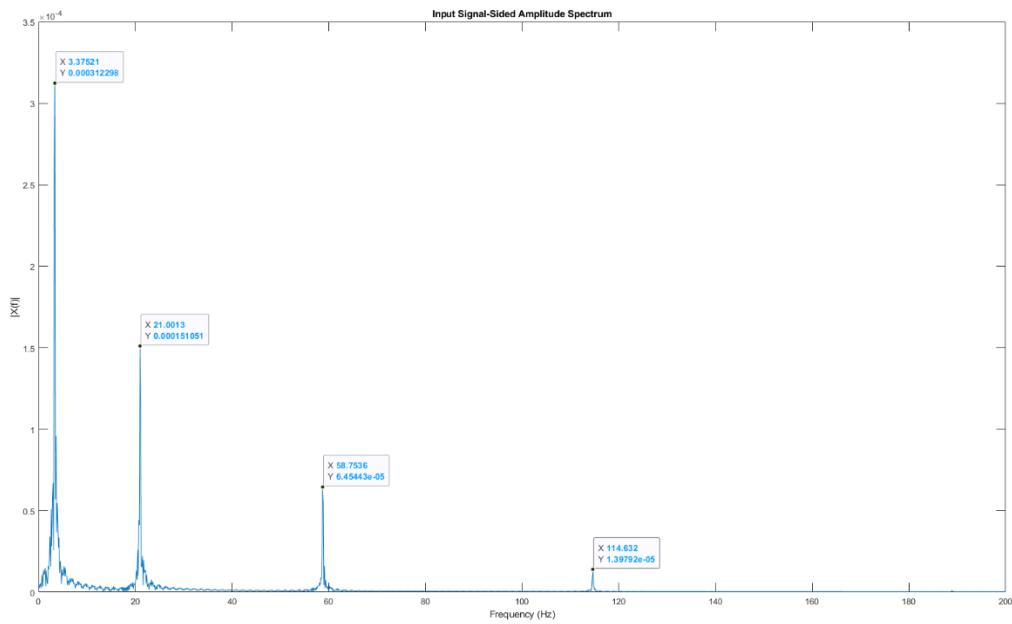


Figure 190: 0.2 position of damage location, with severity of 0.1 for 6m beam

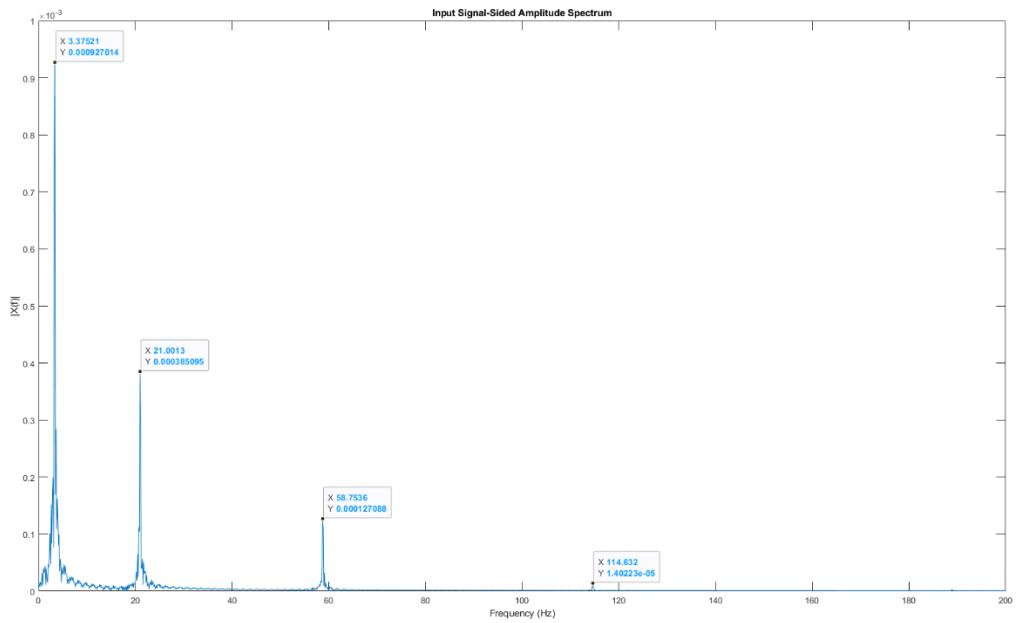


Figure 191: 0.2 position of damage location, with severity of 0.1 for 10m beam

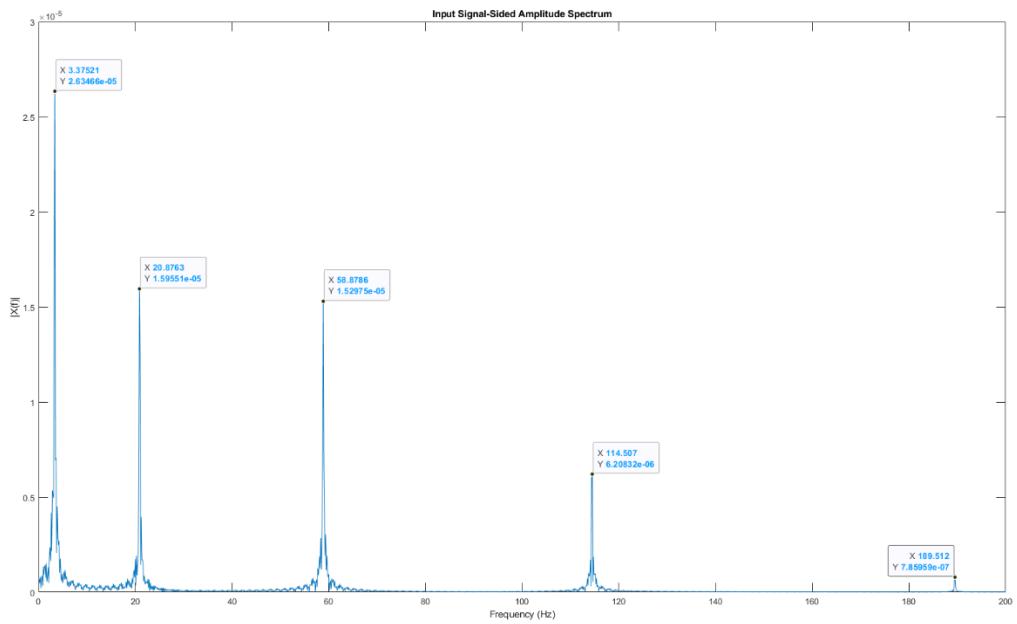


Figure 192: 0.5 position of damage location, with severity of 0.1 for 2m beam

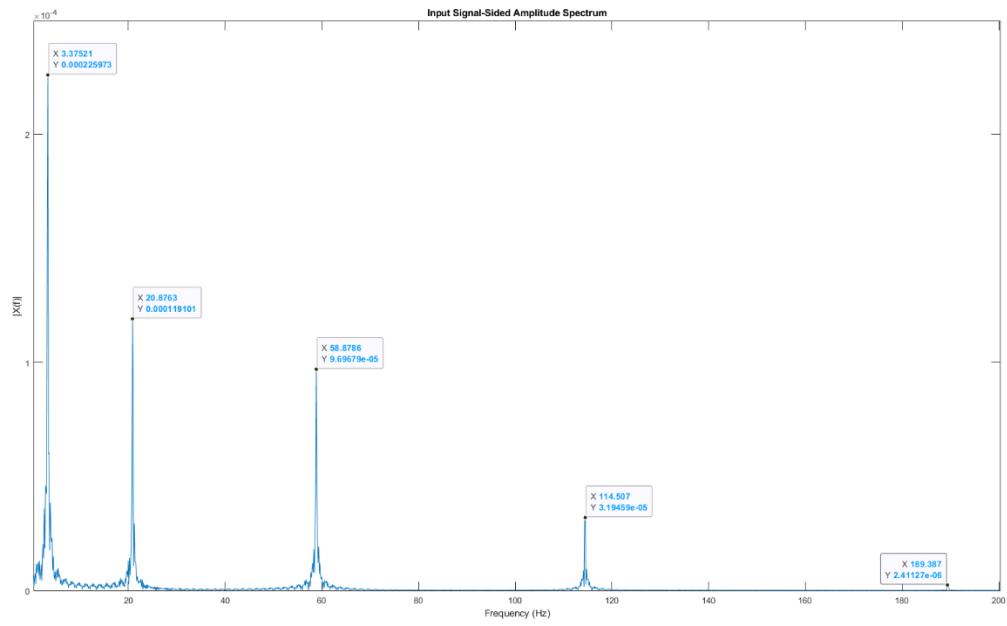


Figure 193: 0.5 position of damage location, with severity of 0.1 for 5m beam

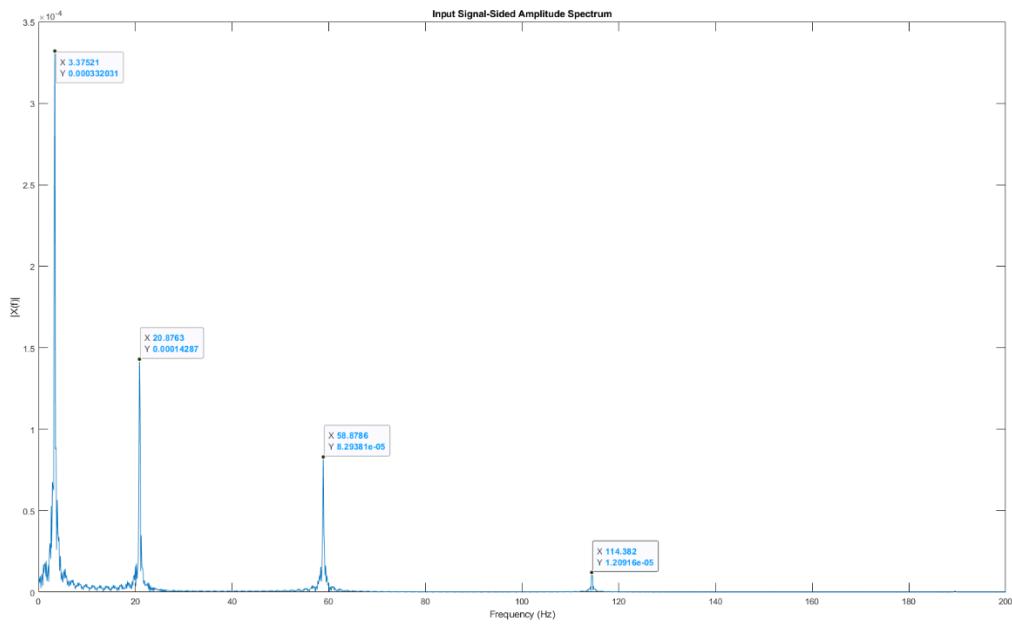


Figure 194: 0.5 position of damage location, with severity of 0.1 for 6m beam

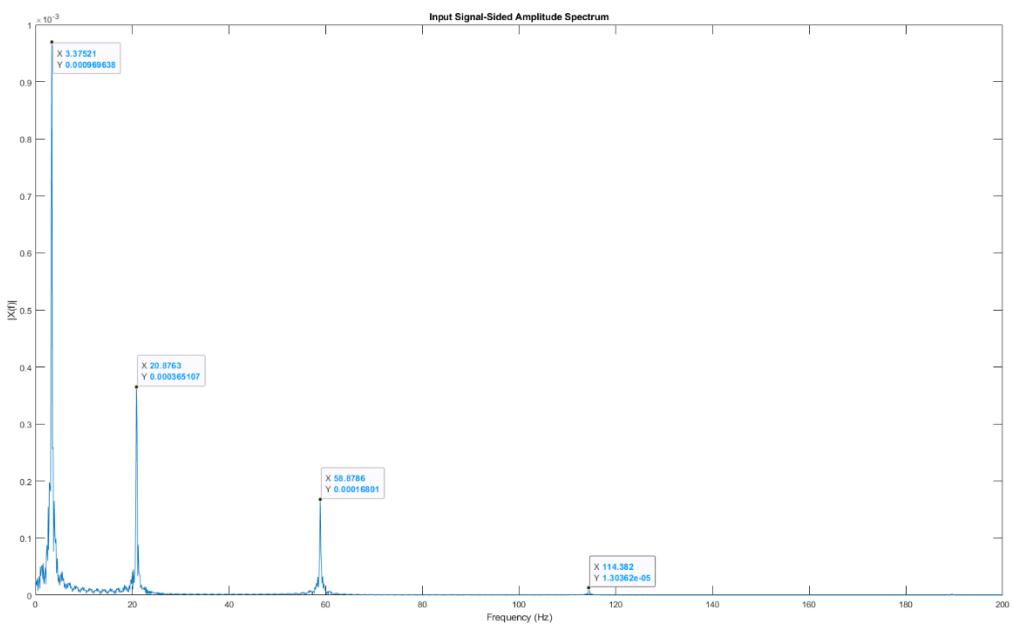


Figure 195: 0.5 position of damage location, with severity of 0.1 for 10m beam

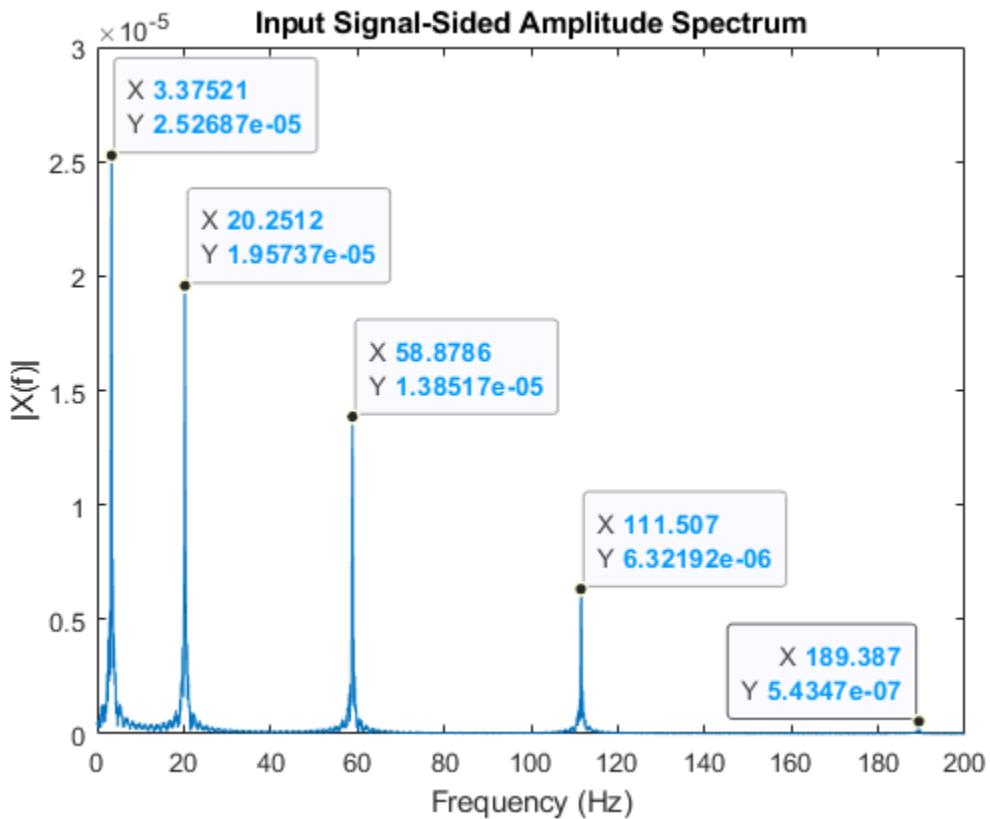


Figure 196: 0.5 position of damage location, with severity of 0.3 for 2m beam

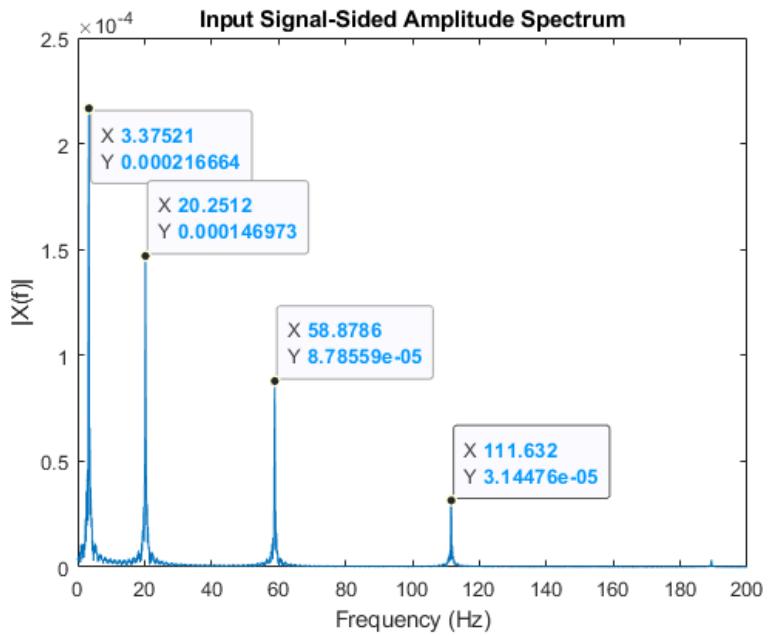


Figure 197: 0.5 position of damage location, with severity of 0.3 for 5m beam

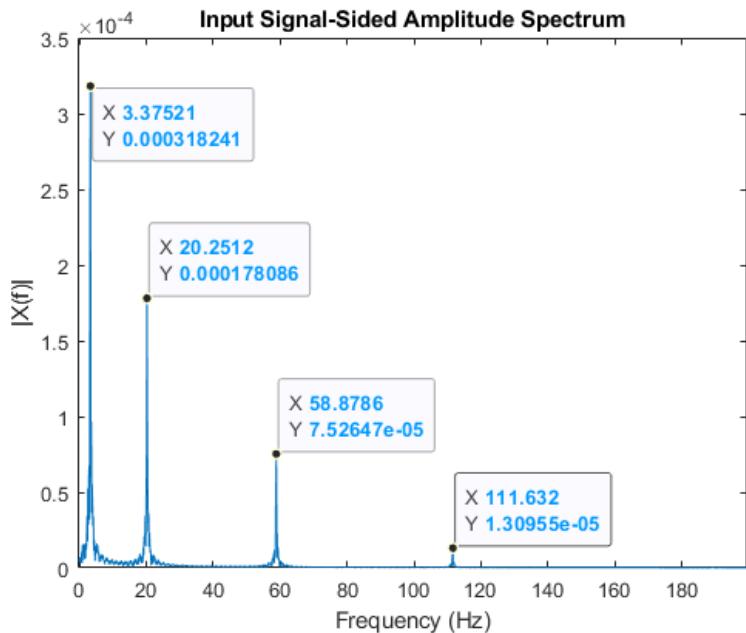


Figure 198: 0.5 position of damage location, with severity of 0.3 for 6m beam

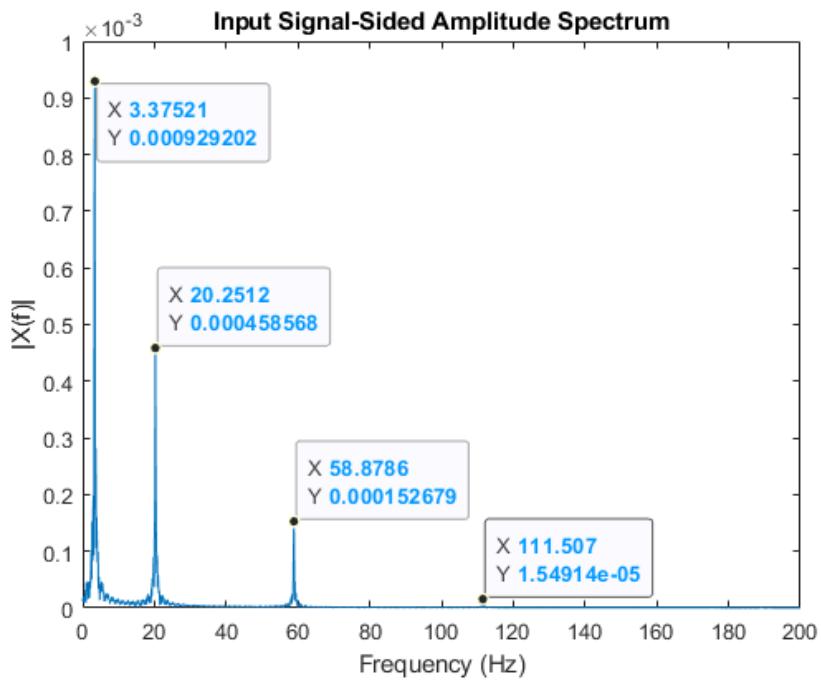


Figure 199: 0.5 position of damage location, with severity of 0.3 for 10m beam

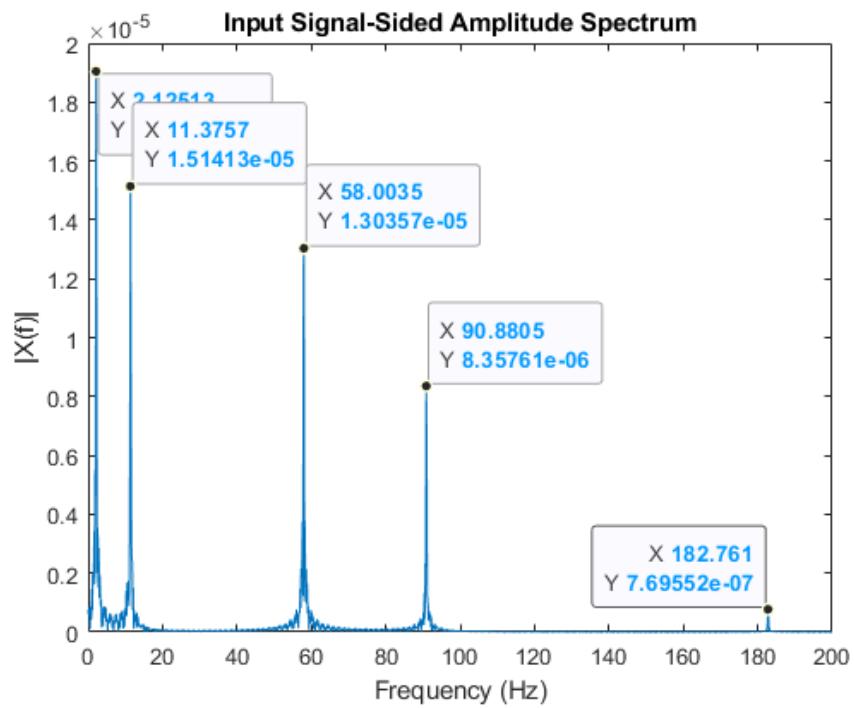


Figure 200: 0.5 position of damage location, with severity of 0.8 for 2m beam

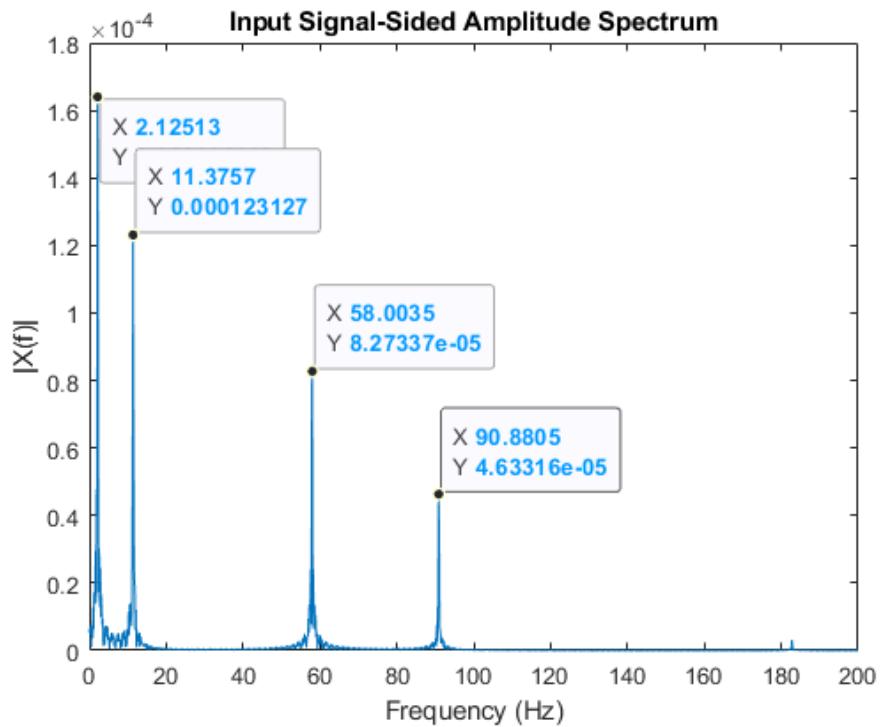


Figure 201: 0.5 position of damage location, with severity of 0.8 for 5m beam

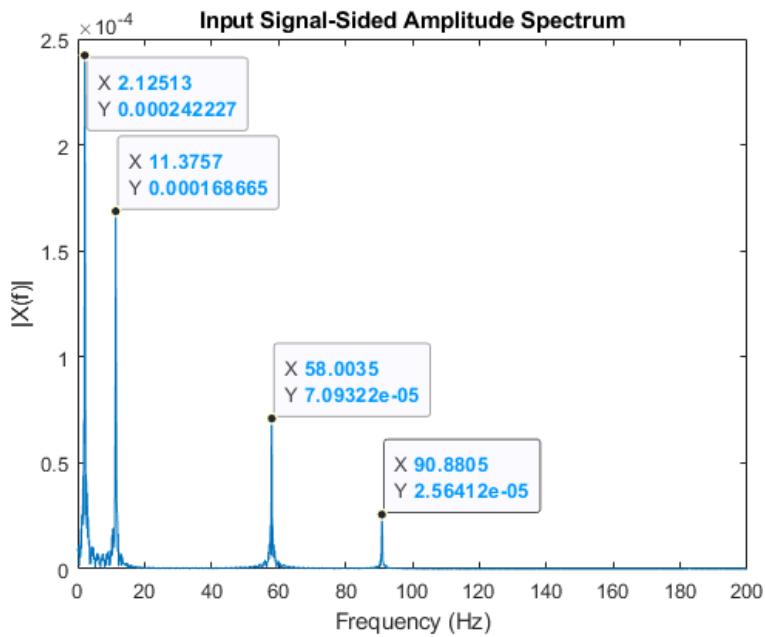


Figure 202: 0.5 position of damage location, with severity of 0.8 for 6m beam

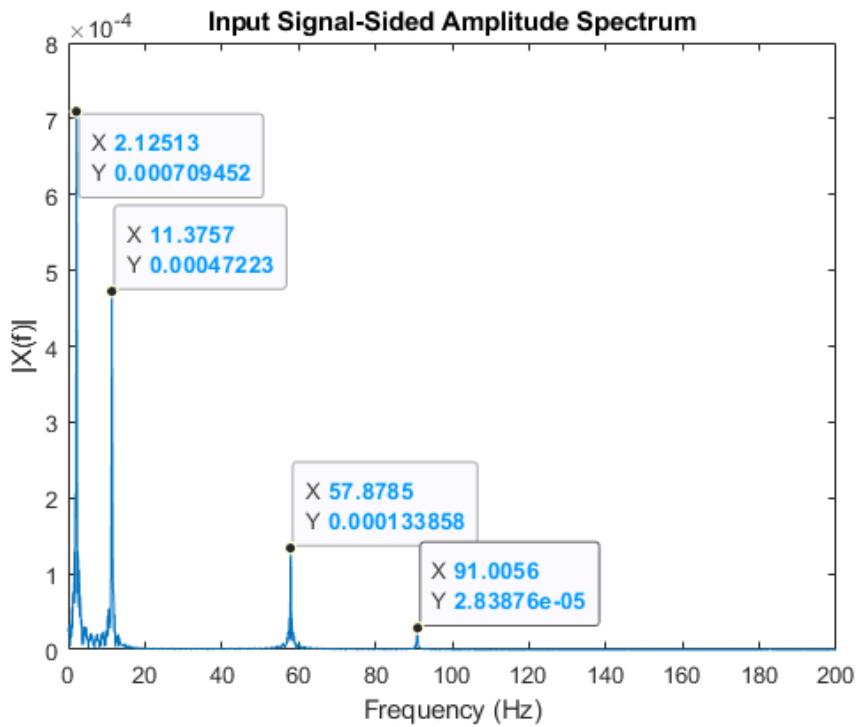


Figure 203: 0.5 position of damage location, with severity of 0.8 for 10m beam

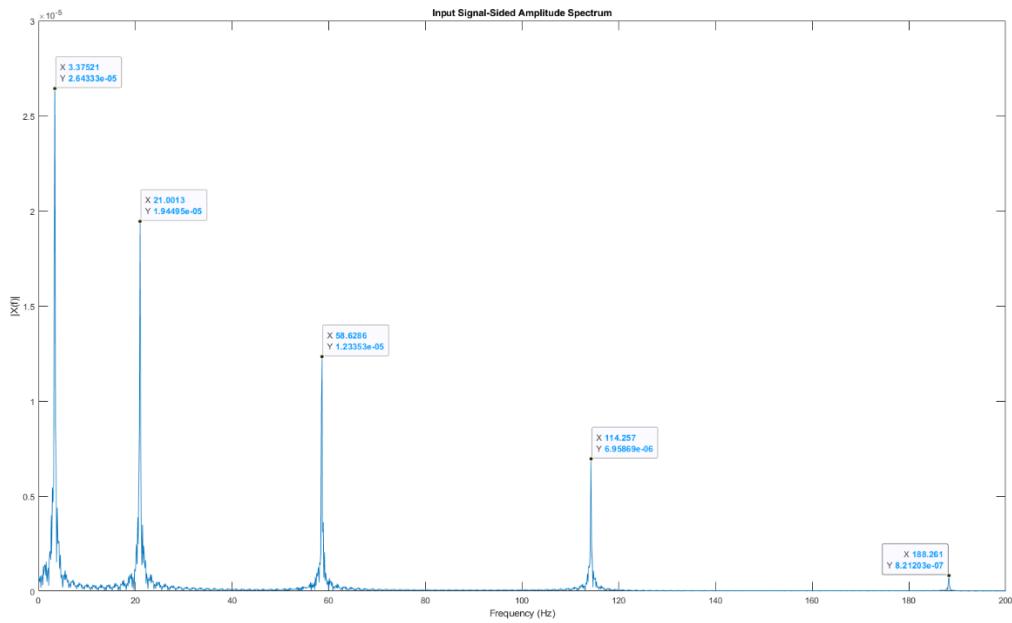


Figure 204: 0.8 position of damage location, with severity of 0.1 for 2m beam

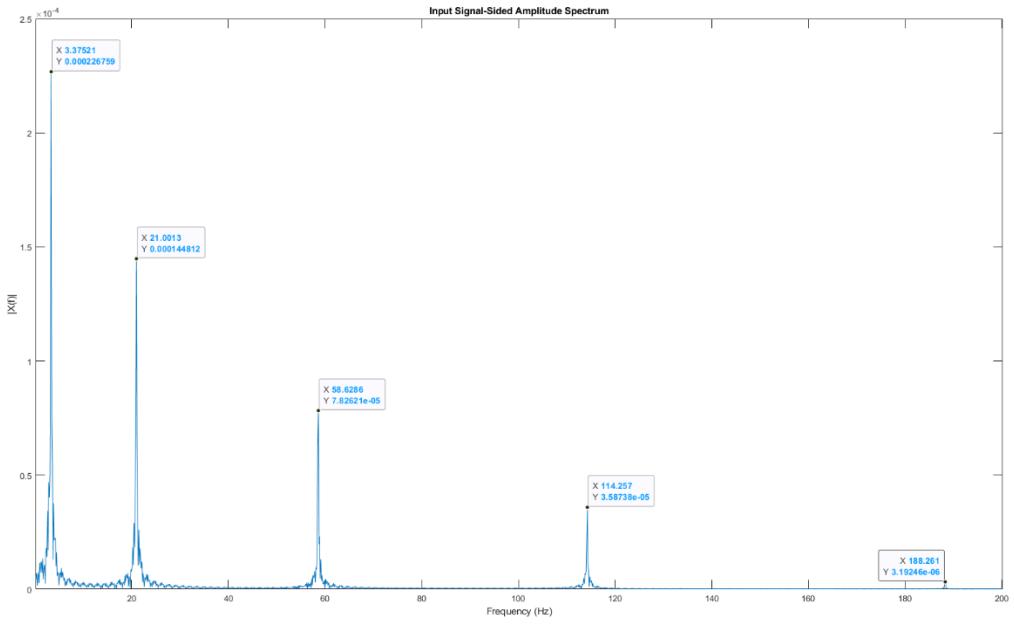


Figure 205: 0.8 position of damage location, with severity of 0.1 for 5m beam

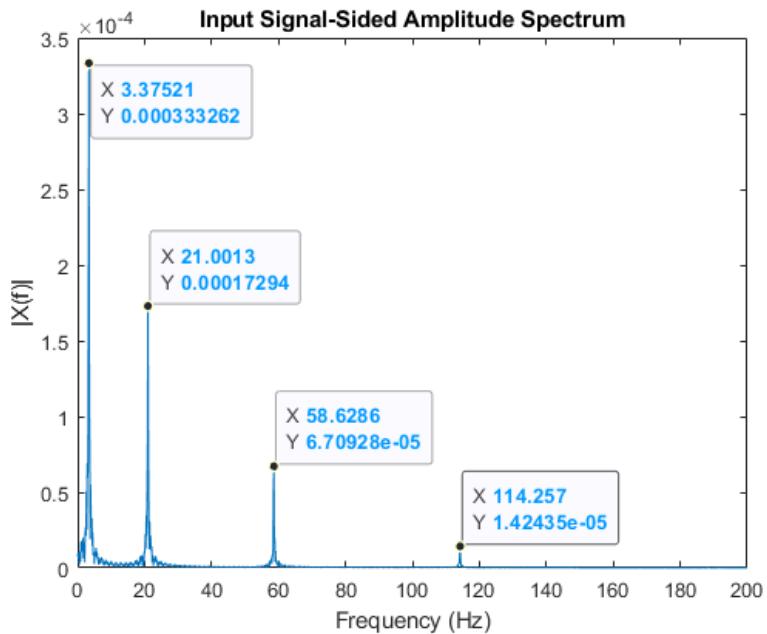


Figure 206: 0.8 position of damage location, with severity of 0.1 for 6m beam

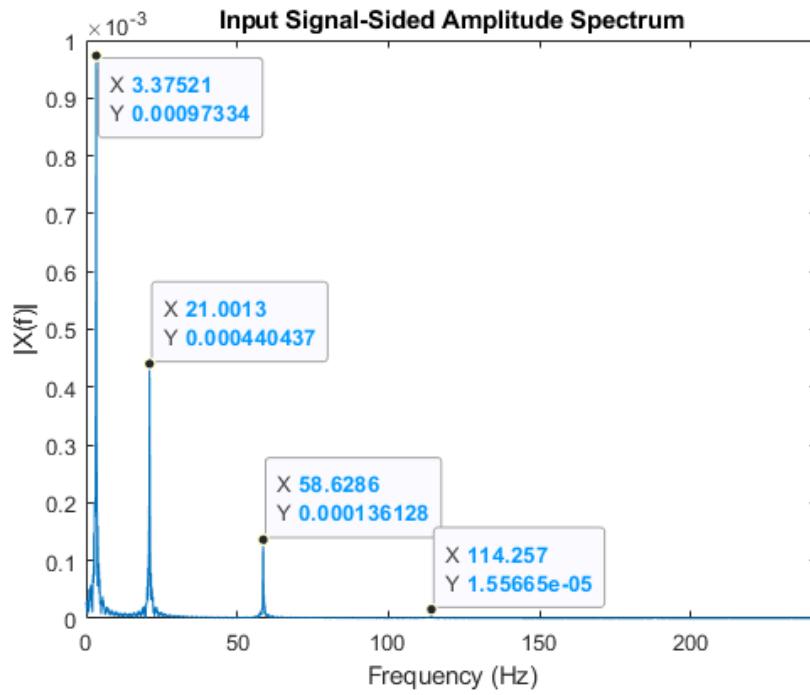


Figure 207: 0.8 position of damage location, with severity of 0.1 for 10m beam

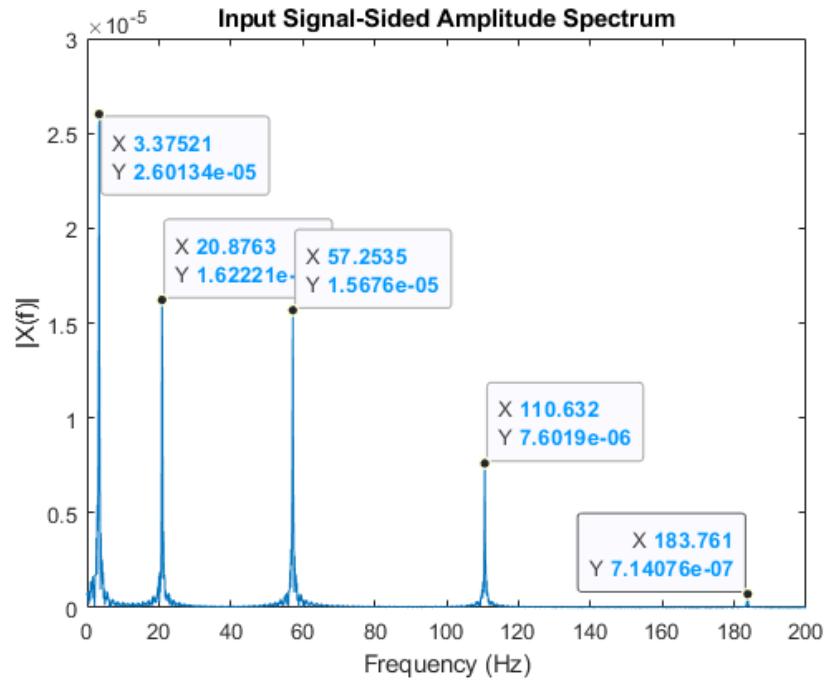


Figure 208: 0.8 position of damage location, with severity of 0.3 for 2m beam

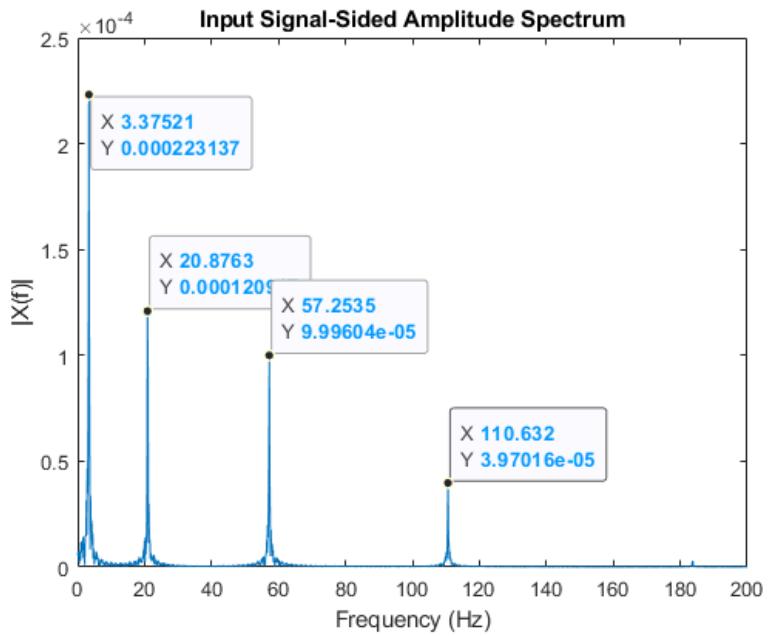


Figure 209: 0.8 position of damage location, with severity of 0.3 for 5m beam

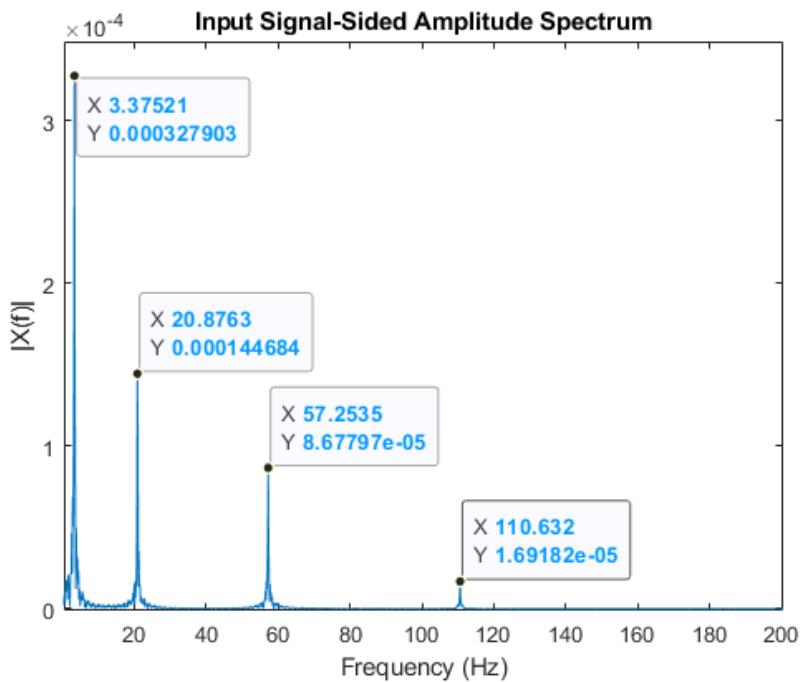


Figure 210: 0.8 position of damage location, with severity of 0.3 for 6m beam

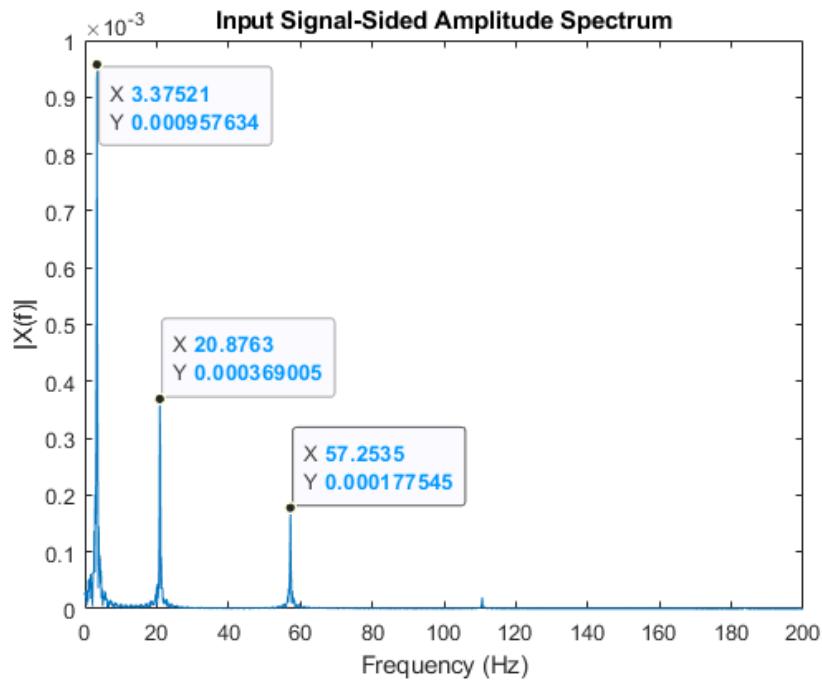


Figure 211: 0.8 position of damage location, with severity of 0.3 for 10m beam

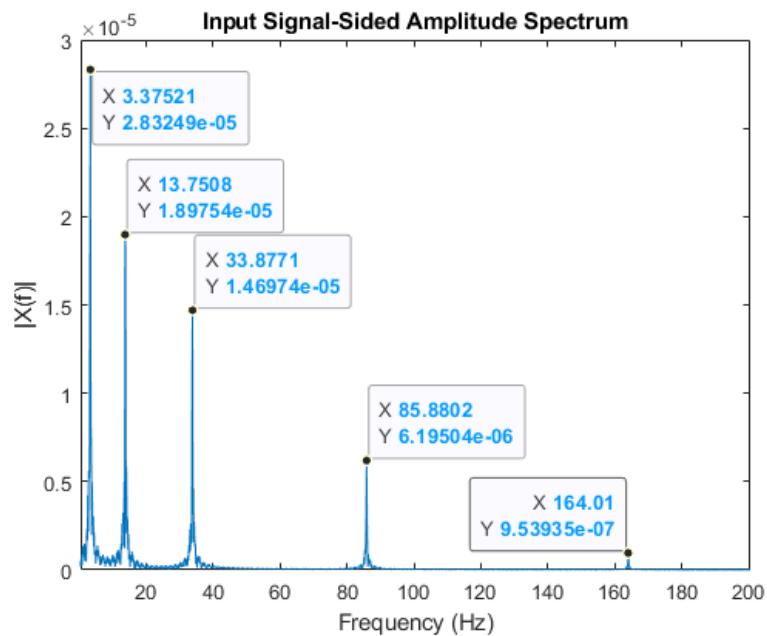


Figure 212: 0.8 position of damage location, with severity of 0.8 for 2m beam

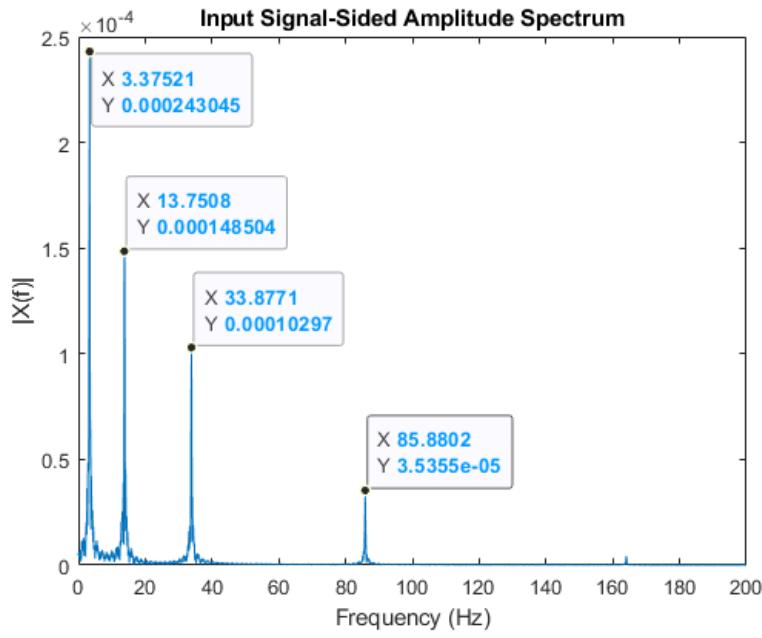


Figure 213: 0.8 position of damage location, with severity of 0.8 for 5m beam

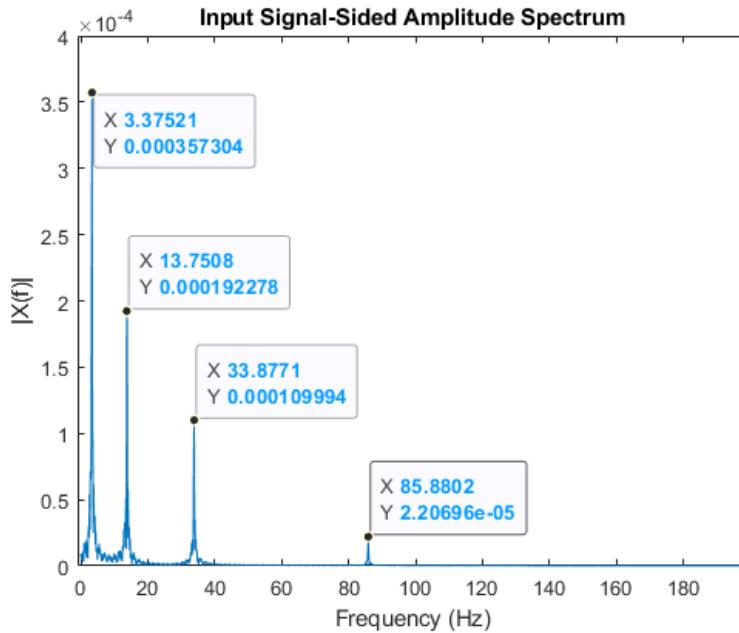


Figure 214: 0.8 position of damage location, with severity of 0.3 for 6m beam

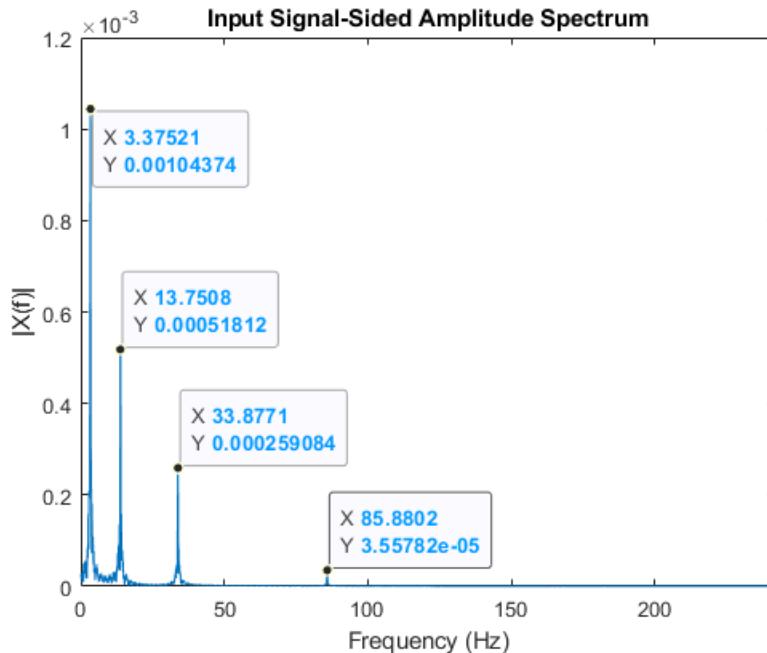


Figure 215: 0.8 position of damage location, with severity of 0.8 for 10m beam

Exploration into the impact of damage location (**dmg_loc**) and damage severity (**dmg_sev**) on the natural frequencies of the first five modes of the beam is conducted. By systematically altering **dmg_loc** within the range [0.05 - 0.9] and **dmg_sev** within [0.1 - 0.9], the in report it aims to observe how these variations influence the natural frequencies for each mode.

6.2.1 List of takeaways or observation through parametric analysis of fundamental response

1. Observations from the parametric analysis of fundamental response indicate that alterations in the position of damage exert a discernible impact on the system's fundamental response.
2. Variations in the position of severity are observed to have a significant influence on the fundamental response of the system.
3. Changes in the length of the beam lead to noteworthy shifts in the fundamental response, unveiling the sensitivity of the system to alterations in its structural dimensions.

6.3 Uniformity of Natural Frequency Changes

The observations made during the study reveal a nuanced relationship between damage severity and natural frequencies. Contrary to uniform changes, it was noted that not all natural frequencies experience alterations in the same ratio. Importantly, the introduction of damage induces a noticeable shift in the natural frequencies, and this shift becomes more pronounced as the severity of damage increases at specific locations. Interestingly, as damage severity escalates, there is a discernible absence of certain natural frequencies that were initially identified.

6.4 Impact of Noise Level (noise_level)

Variations in the noise level within the range [0.01 - 0.07] are introduced, and the ensuing changes in the results are meticulously examined. This investigation aims to elucidate the sensitivity of the damage detection process to varying levels of noise.

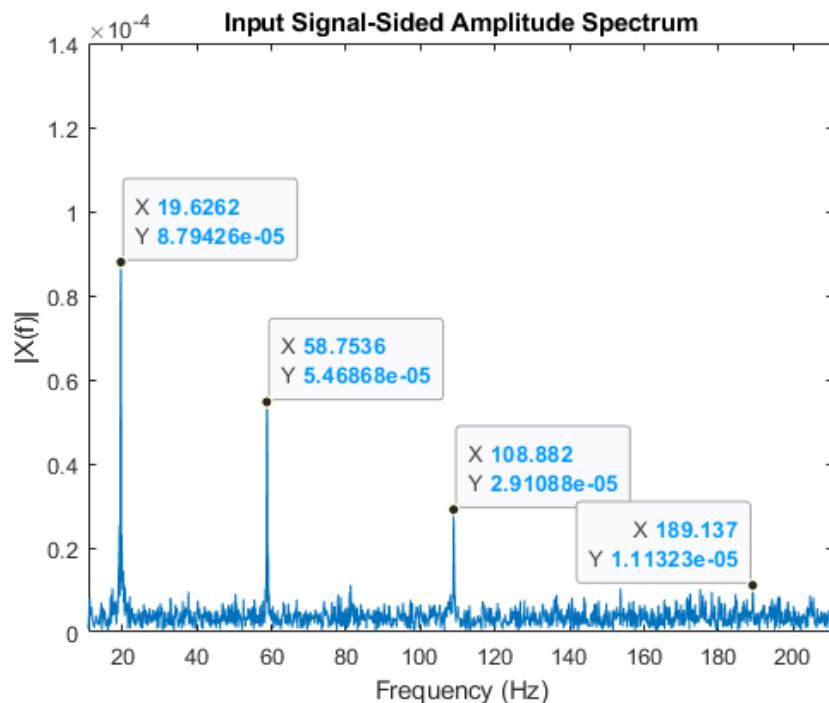


Figure 216: Noise level of 0.02, 0.5 position of damage location, with severity of 0.4 for 3m beam

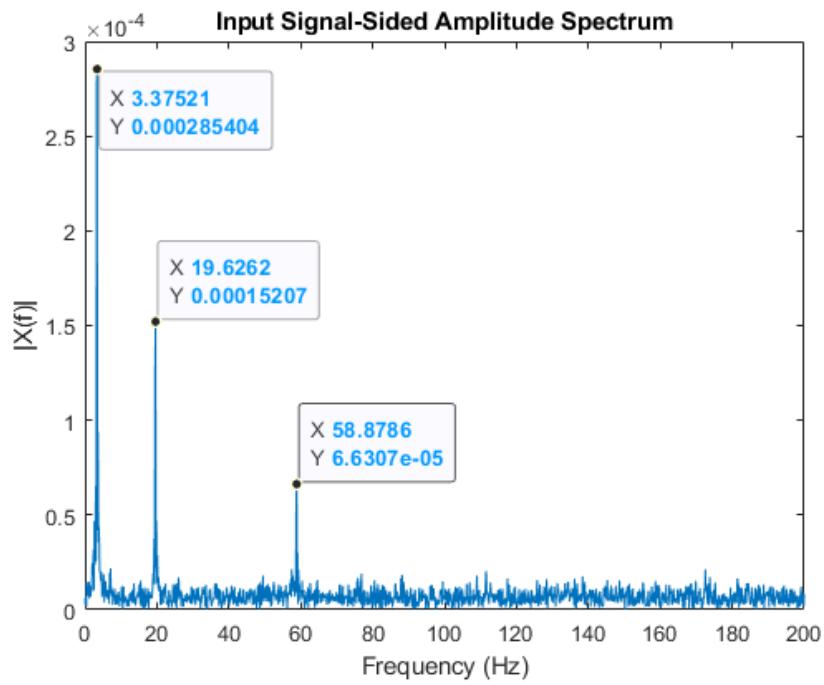


Figure 217: Noise level of 0.02, 0.5 position of damage location, with severity of 0.4 for 6m beam

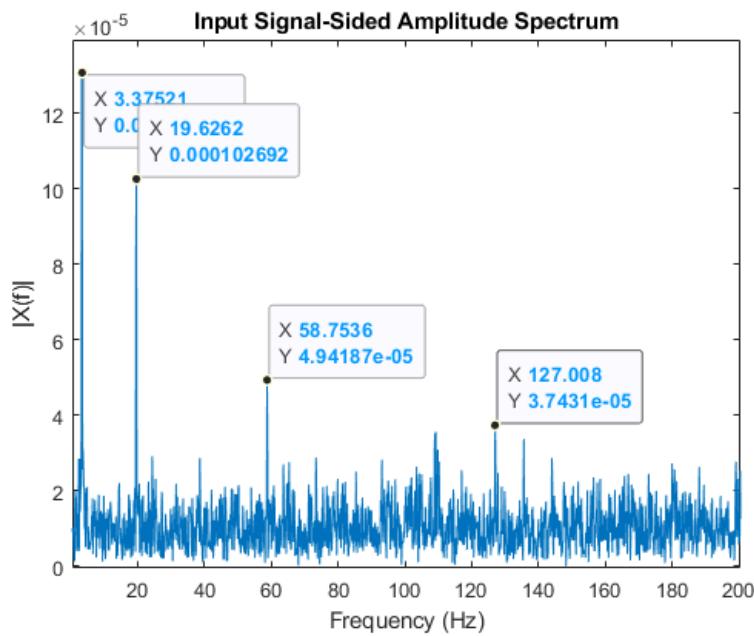


Figure 218: Noise level of 0.06, 0.5 position of damage location, with severity of 0.4 for 3m beam

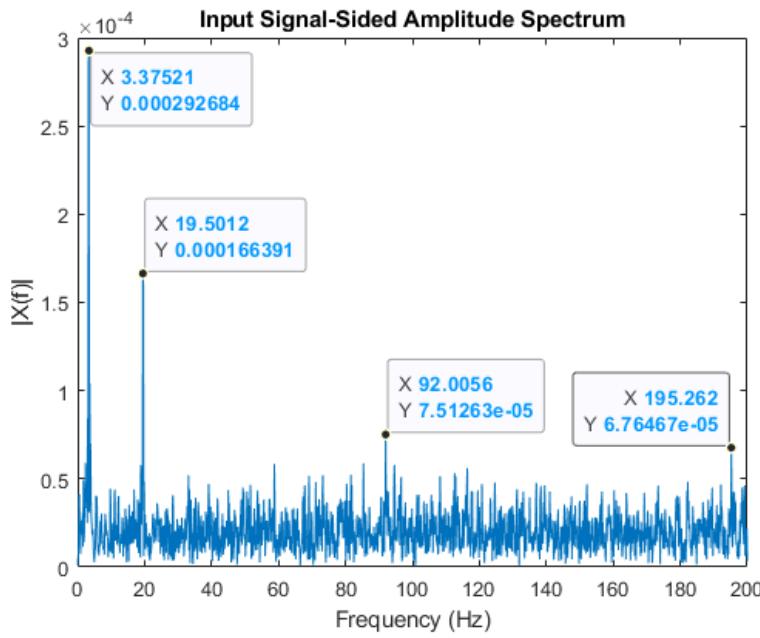


Figure 219: Noise level of 0.06, 0.5 position of damage location, with severity of 0.4 for 6m beam

- 6.4.1 **Observation:** The introduction of noise, with magnitudes of 0.02 and 0.06, has distinct effects on signal quality and reliability. A noise level of 0.02 introduces a moderate degree of interference, potentially causing subtle variations in the signal without significantly compromising its overall integrity. On the other hand, a noise level of 0.06 represents a more pronounced influence, likely leading to more noticeable distortions and fluctuations in the signal. Both noise levels, while introducing disruptions, provide an opportunity to assess the system's resilience and sensitivity to external disturbances.

6.5 Influence of Damping Ratio (zeta)

By increasing the damping ratio (**zeta**) in the range [0.0 - 0.05], the team explores the impact of damping on the results. Understanding how damping influences the damage detection process provides crucial insights into the robustness of the methodology under different damping conditions.

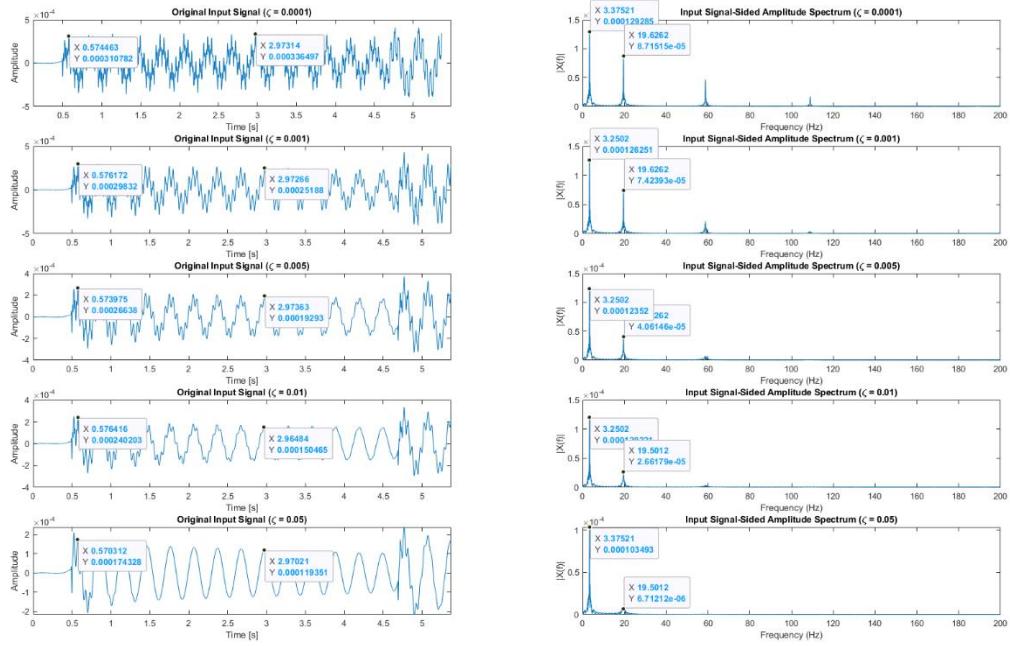


Figure 220: Plot of damping for damage location of 2

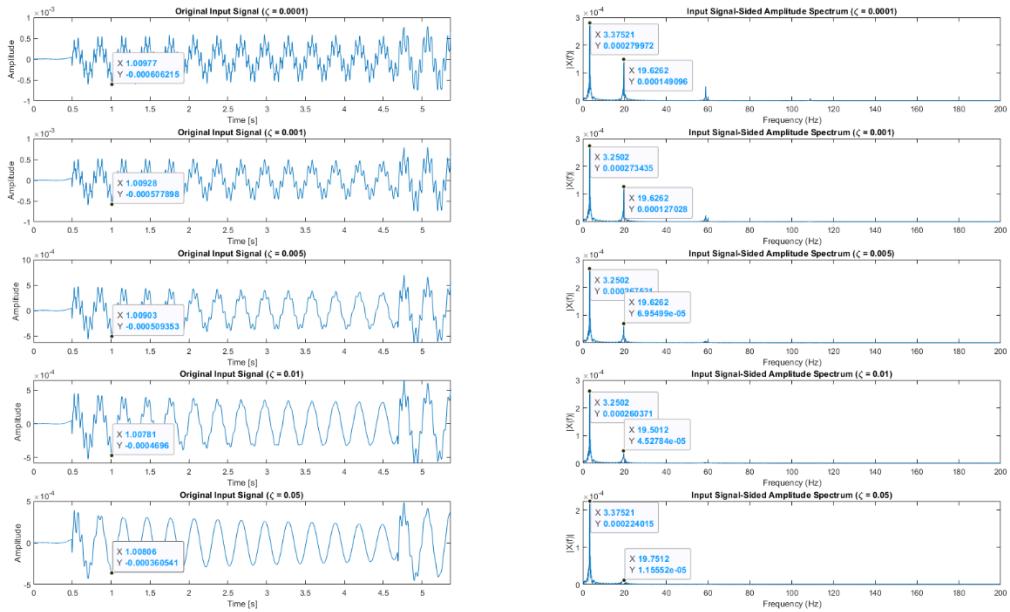


Figure 221: Plot of damping for damage location of 4

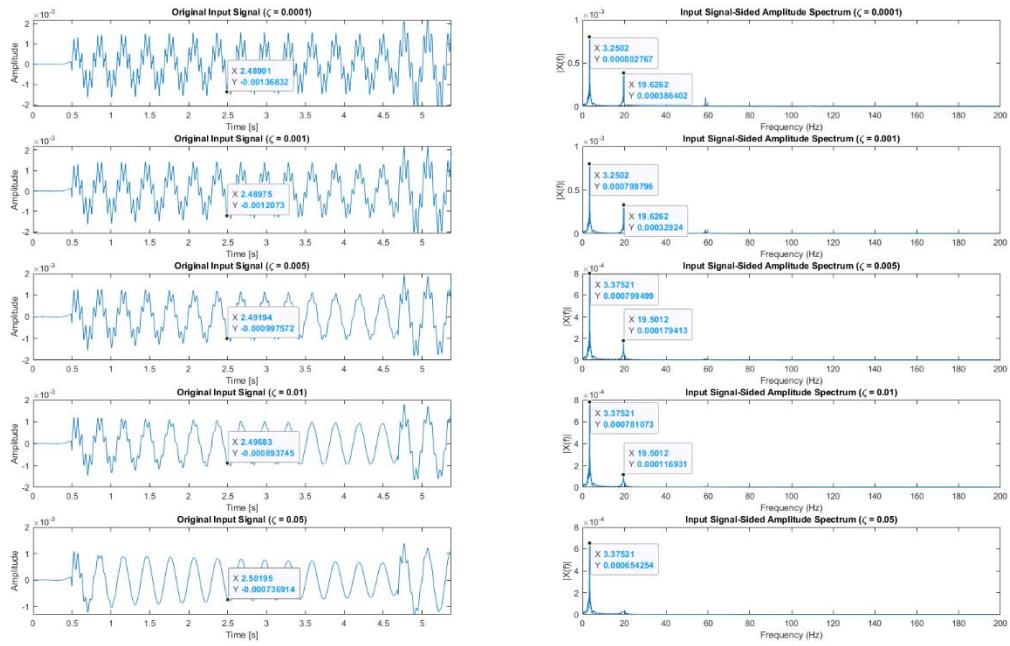


Figure 222: Plot of damping for damage location of end

The parameter 'zeta' in this context represents the damping ratio, impacting the decay of oscillations in a dynamic system. A higher damping ratio leads to quicker decay of oscillations, resulting in a more rapid decrease in amplitude over time.

In the original input signal plot (Amplitude vs. Time), altering the value of 'zeta' influences the rate at which the amplitude diminishes. Generally, a low 'zeta' signifies less damping, allowing oscillations to persist for a longer duration with a slower amplitude decay, creating a more sustained oscillatory behavior. Conversely, a high 'zeta' indicates more damping, causing oscillations to lose energy rapidly, leading to a faster decrease in amplitude and a quicker convergence to a stable state.

6.5.1 Observation:

The damping ratio emerges as a pivotal factor shaping the dynamic behavior of the system. Analyzing its influence across various values—0, 0.01, 0.025, and 0.05—reveals distinct trends. A damping ratio of 0 characterizes an undamped system, where oscillations persist indefinitely. Progressing from a damping ratio of 0.01 to 0.05, a proportional reduction in

oscillation amplitude becomes evident. This diminishing amplitude signifies the stabilizing impact of damping on the system, preventing uncontrolled and potentially detrimental oscillations. The gradual increase in damping ratios signifies an incremental improvement in the system's capacity to dissipate energy and resist excessive motion. This underscores the critical role of damping in fostering stable and controlled dynamic responses within the system.

6.6 Cross-Correlation and Cross-Spectral Density Analysis

The team computes and plots the cross-correlation (ϕ_{xy}) using displacements from different coordinates. Additionally, the cross-spectral density (S_{yx}) is computed and plotted. These analyses offer valuable information on the interdependence of displacement measurements and their frequency content.

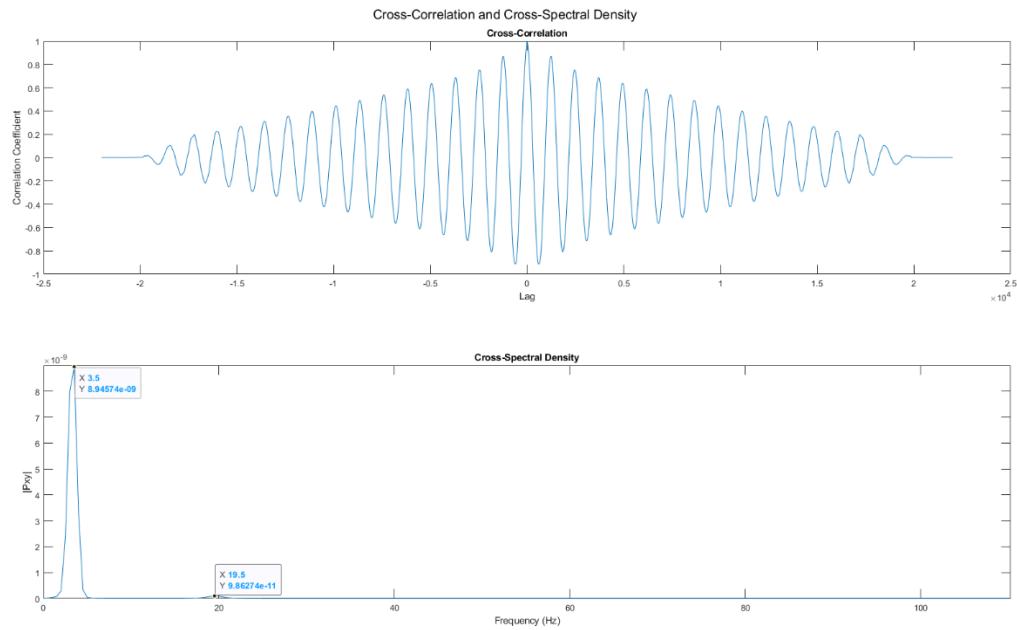


Figure 223: Plot of Cross-correlation and Cross-spectral density between damage location 2 and 3

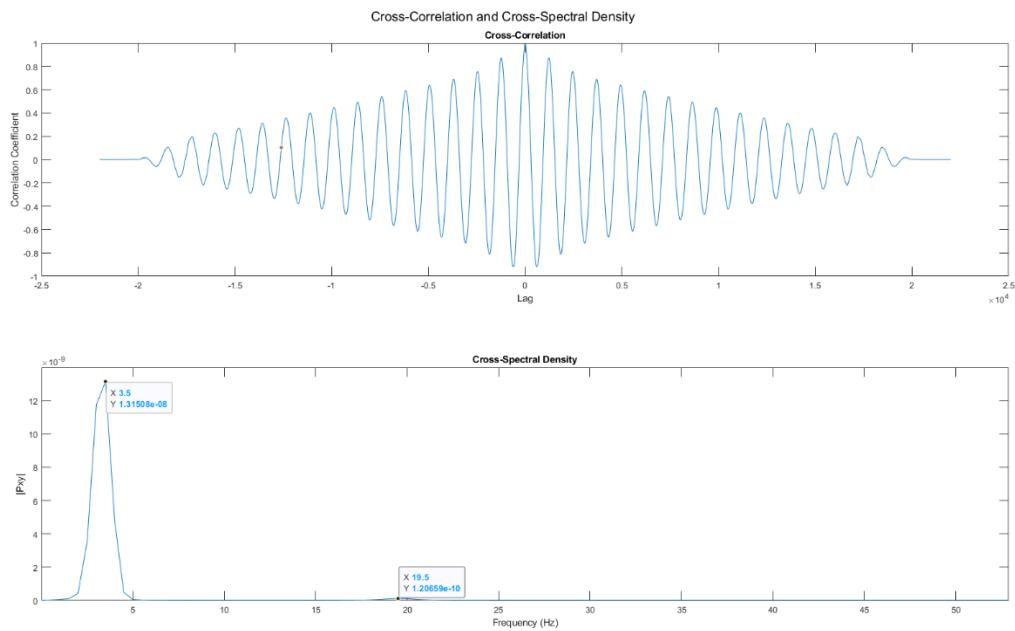


Figure 224: Plot of Cross-correlation and Cross-spectral density between damage location 2 and 4

7. Additional Task

7.1 Introduction

In the exploration of Signal Processing, Design of Experiments, and System Identification during the Winter Term of 2023/2024, the team, led by Z. Jaouadi, R. R. Das, and T. Lahmer, embarked on a series of final projects, addressing a spectrum of tasks to deepen understanding of these interdisciplinary domains. Through six computer classes and additional tasks, the team delved into model updating techniques, parameter identification for a single degree of freedom system, signal processing methodologies, and advanced tomography applications. Each task was **an attempt** to meticulously documented and evaluated, adhering to rigorous standards of correctness, completeness, and interpretability. The findings, interpretations, and observations were compiled into a comprehensive report.

The report stands as a testament to dedication, problem-solving skills, and proficiency in these critical areas of study.

7.2 Model Updating

7.2.1 Residuals Method and Optimization Algorithms

The Residuals Method, employed in the context of system parameter determination, is a technique that aims to minimize the discrepancies or residuals between the observed and predicted values of a given system. In the context of the provided vectors x , y , and \hat{y} stored in Task01_G03.mat, the objective is to determine the parameters a_i of the system described by the equation:

$$\hat{y} = \sum(a_i \cdot x^i)$$

It signifies a polynomial model where \hat{y} represents the predicted output, a_i denotes the coefficients of the polynomial, x is the input variable, and i signifies the degree of the polynomial. This expression encapsulates a summation of terms, each consisting of a coefficient multiplied by the input variable raised to a specific power x^i . The sum extends over various degrees of the polynomial, contributing to the overall prediction of the output variable \hat{y} .

The Residuals Method involves defining a cost function that quantifies the difference between the predicted values (\hat{y}) and the actual observations (y). This cost function typically takes the form of the sum of squared residuals, expressed as:

$$J(\mathbf{a}) = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Here, y_i represents the vector of unknown parameters a_i , and n denotes the number of observations. The optimization algorithms in MATLAB can be employed to minimize this cost function, iteratively adjusting the parameters to converge towards a solution that minimizes the discrepancies.

The optimization process involves selecting an appropriate algorithm, initializing the parameters, and iteratively updating them to minimize the cost function. Common optimization algorithms include gradient descent, nonlinear least squares, or more sophisticated methods like the Nelder-Mead algorithm.

In the context of the Residuals Method applied to the vectors x , y , and n^{\wedge} , the goal is to retrieve the optimal values for the parameters a_i , ensuring that the predicted values closely match the actual observations, thereby achieving an accurate representation of the underlying system.

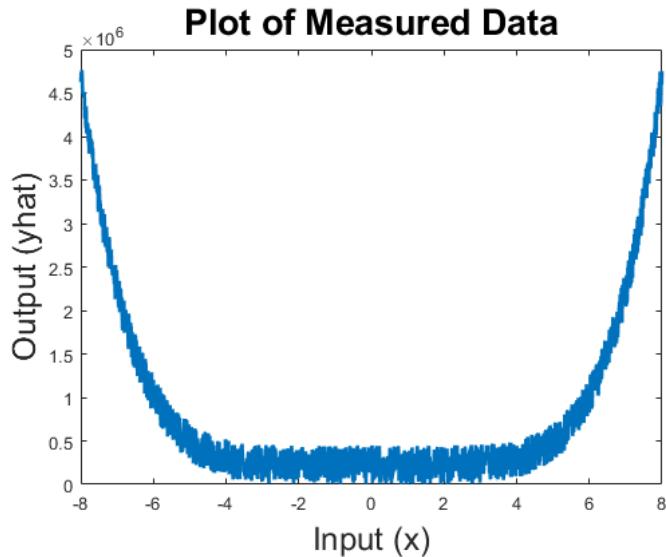


Figure 225: Plot based on the given data

This plot shows the measured data points, where the x-axis represents the input values (x) and the y-axis represents the corresponding output values ($yhat$). The data appears to exhibit some pattern or relationship.

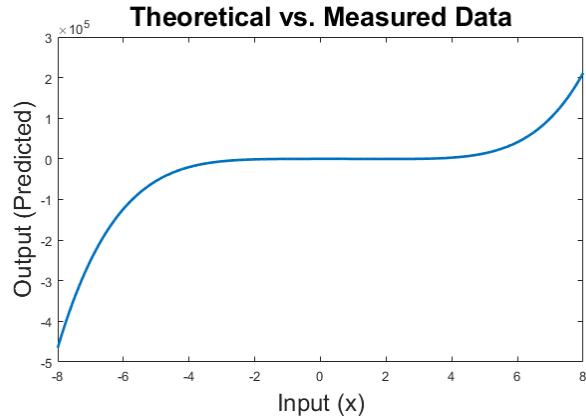


Figure 226 Theoretical vs. Measured Plot

This plot compares the theoretical output (based on initial parameter estimates) with the measured data. The theoretical output is computed using a model with initial parameter values. The plot helps visualize how well the initial model fits the measured data.

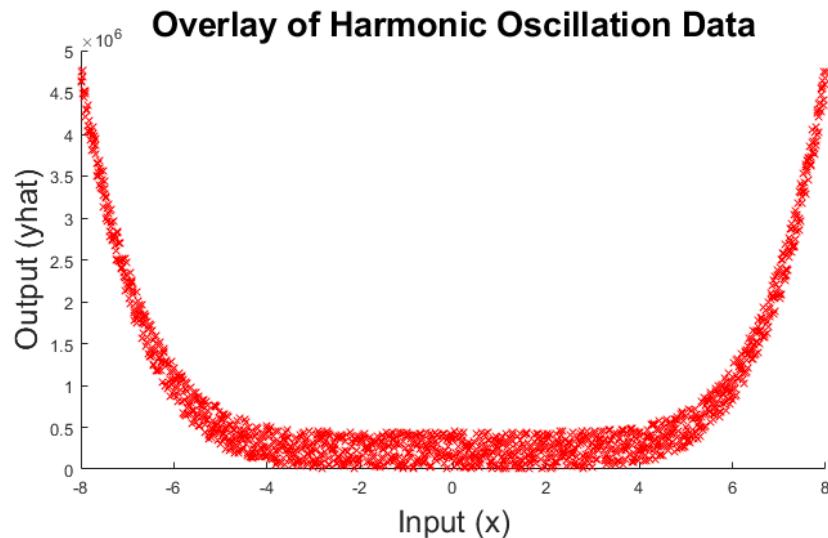
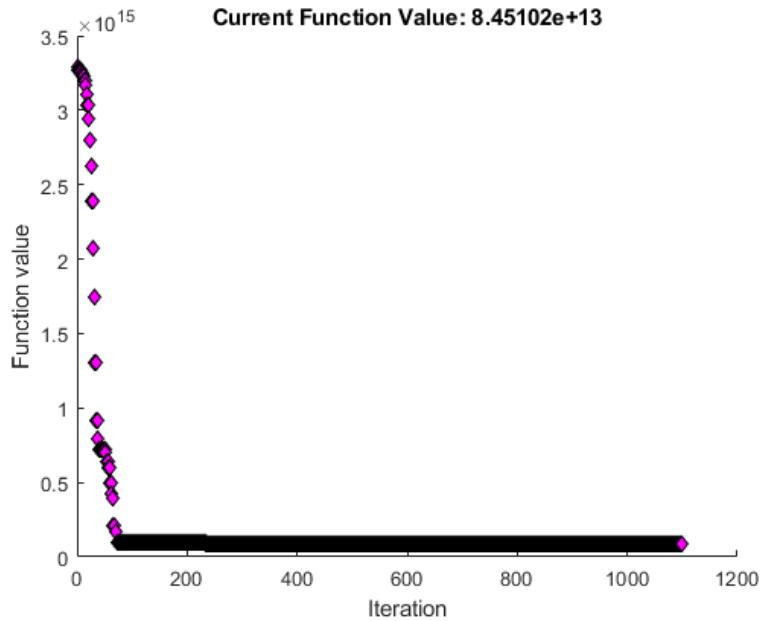


Figure 227: overlay of harmonic oscillation

Overlay here means, the process of combining or superimposing multiple oscillatory signals to visualize their relationships or patterns.



Interpretation: A decreasing trend in the function values over iterations generally indicates that the algorithm is converging towards an optimal solution. No observation of plateau or fluctuations signifying absence of multiple solutions. Also, a steep decrease in function values early in the iterations suggests rapid convergence. The algorithm is making substantial progress toward the optimum.

7.2.2 Residuals Method and Optimization Algorithms

Measured vs. Predicted Plot: In the "Measured vs. Predicted Plot," the measured output (y) is plotted against the output obtained using the initial parameter estimates (predictedOutput). The code uses the plot function to create a scatter plot, where the measured data points are represented by circles ('o') and the initial predicted output is represented by a solid line. This plot visually compares how well the initial parameter estimates capture the trend of the measured data. Ideally, the predicted output should closely follow the measured data points, indicating a good fit. However, its observed the predicted output is not following the measurement plot.

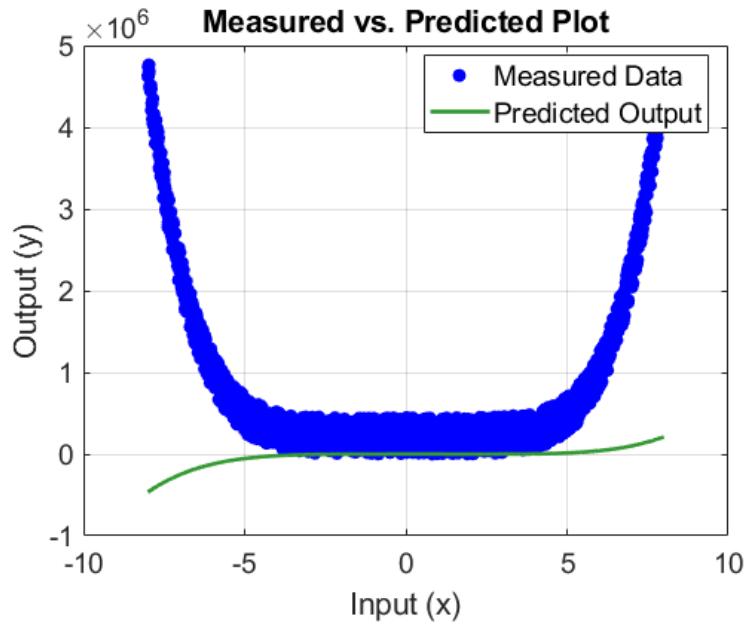


Figure 228: Measured vs. Predicted Plot

Residuals Plot: The "Residuals Plot" visualizes the differences between the measured output (y) and the predicted output (predicted Output). The code calculates the residuals (residuals = $y - \text{predicted Output}$) and then plots these residuals against the input (x).

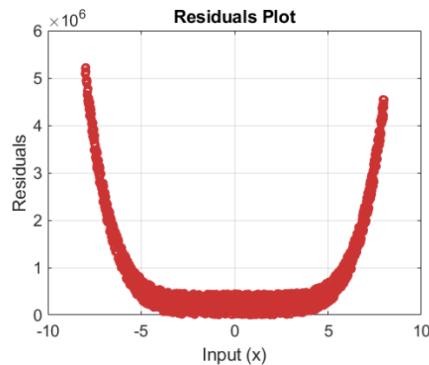


Figure 229: Residuals Plot

7.3 Regularization Algorithms

Singular Value Decomposition (SVD) Results: Plot of measured output against the output obtained using the SVD method. This provides a visual representation of how well the SVD method fits the data.

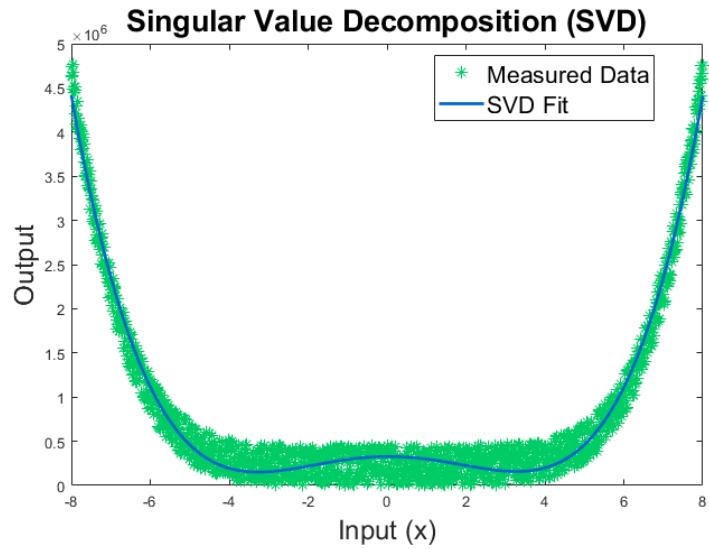


Figure 230: Plot of measured output against the output obtained using the SVD method

Truncated SVD (TSVD) Results: Similar to SVD, plot of measured output against the output obtained using the TSVD method to compare their performance.

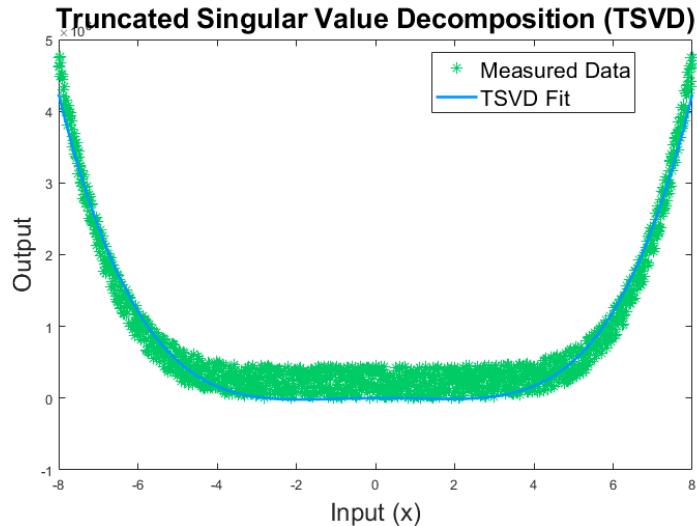


Figure 231: Plot of measured output against the output obtained using the TSVD method

Iterative Method Results: Plot of measured output against the output obtained using the iterative regularization method.

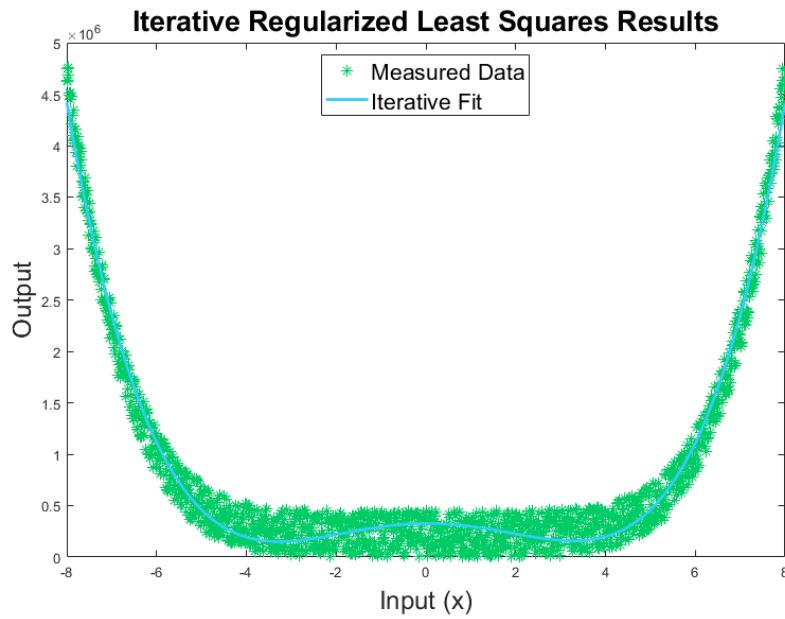


Figure 232: Plot of measured output against the output obtained using the iterative regularization method

7.4 Comparison of Results

Comparison Plot: The plot showcasing that overlays the outputs obtained from different methods (Residuals Method, SVD, TSVD, Iterative Method) for a direct visual comparison, different colors or markers are used for clarity.

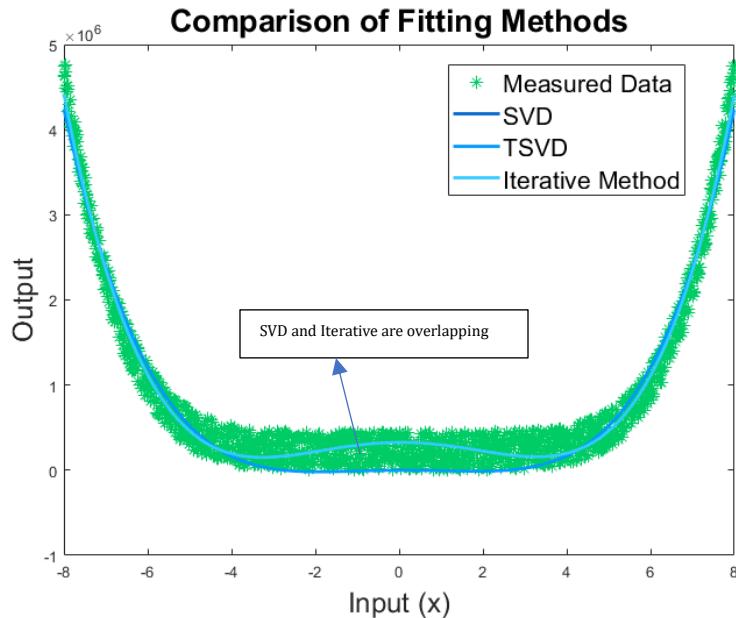


Figure 233: Comparison of Fitting Methods

Residuals Comparison: Plot of the residuals obtained from different methods to compare the error distribution.

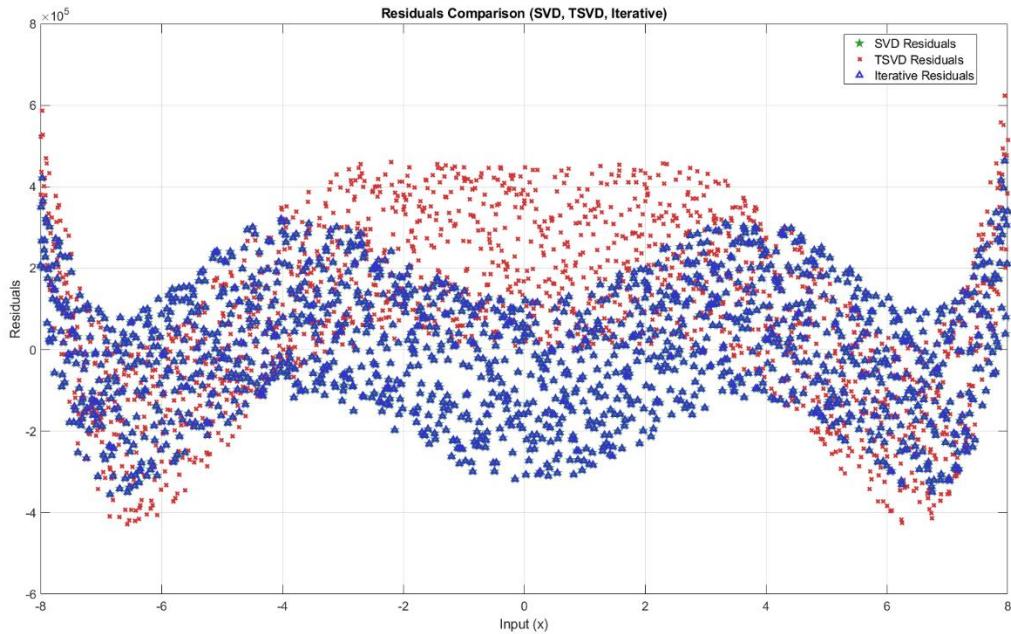


Figure 234: Residuals plots of all three methods

```
Root Mean Square Error (SVD) : 154383.5983
Root Mean Square Error (TSVD) : 233092.6797
Root Mean Square Error (Iterative) : 154383.7183
```

Figure 235: Residuals value for all the given data

Figure 235, showcases the comparison of performance based on RMS, SVD and Iterative are better compared to TSVD.

7.5 Observations, Findings, and Conclusions

Error Metrics: Present error metrics (Root Mean Square Error, Mean Absolute Error) for each method to quantitatively compare their performance.

```

Error Metrics:
SVD: RMSE=154383.5983, MAE=128069.4074
TSVD: RMSE=233092.6797, MAE=193833.0517
Iterative: RMSE=154383.6031, MAE=128070.2989

```

Figure 236: Quantitative comparison of performance of methods

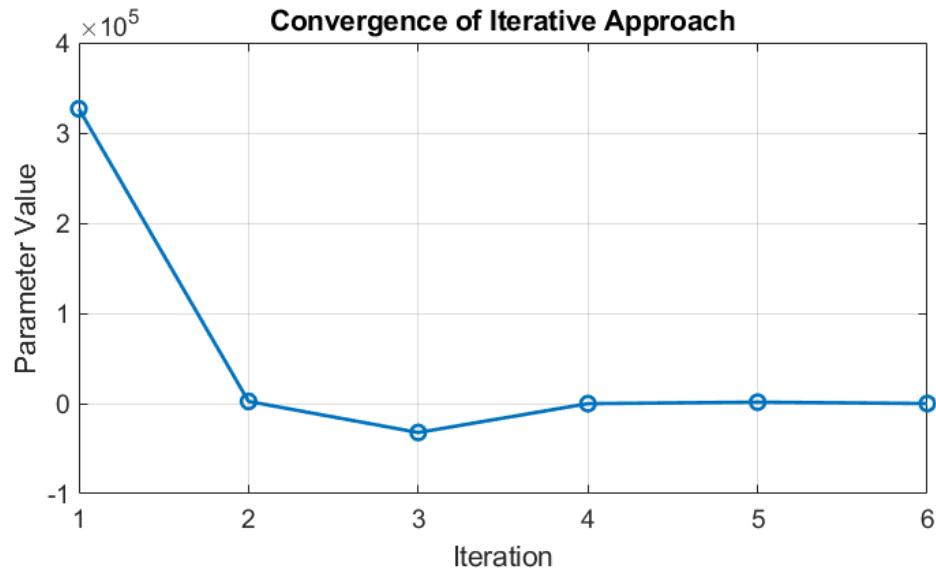


Figure 237: Convergence of Iterative approach, signifying the degree of polynomial construction

```

Optimal Alpha (SVD): 1e-05
Optimal K (TSVD): 1
Optimal Iteration Limit (Iterative): 100
Optimal Polynomial Degree: 6

```

Figure 238: Optimal model param for all three methods

8. Parameter Identification (20 points)

8.1.1 System Calibration in Frequency Domain

In the task of System Calibration in Frequency Domain, the objective is to identify the parameters (mass, damping, and stiffness) of a single-degree-of-freedom system using the Frequency Response Function (FRF). Based on the task, the hierarchy of process involves analyzing the amplitude and phase of the FRF, individually and collectively, to optimize and determine the system's characteristics. By constructing suitable cost functions and examining the uniqueness of solutions, this task delves into understanding the interplay

between the frequency domain and system parameters. The results provide insights into the calibration process, revealing nuances in the identification of system properties based on different aspects of the FRF. The comparison with calibration methods in the time domain adds depth to the analysis, emphasizing the benefits and distinctive features of frequency domain approaches.

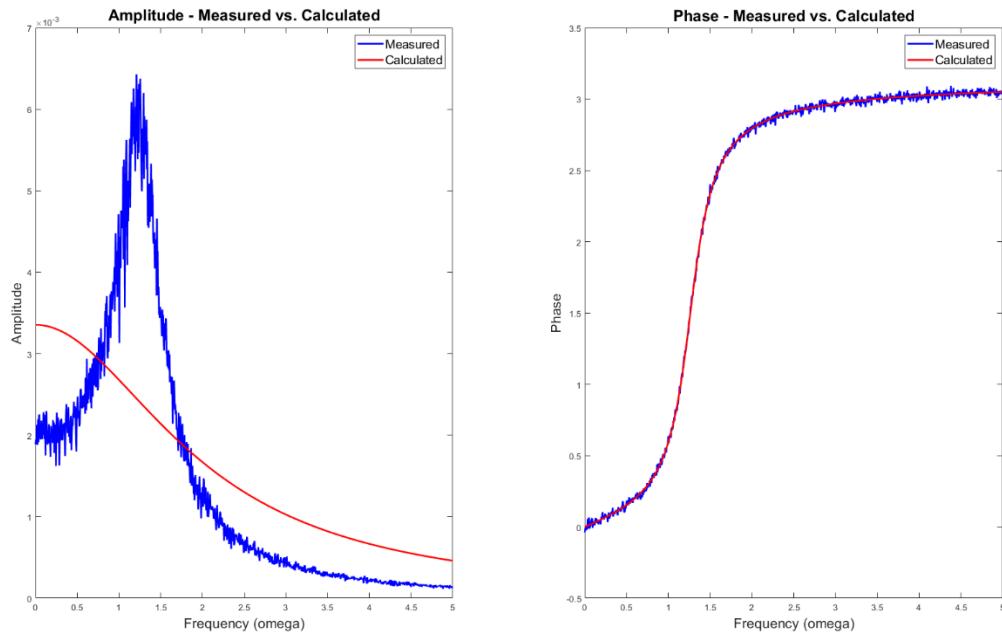


Figure 239: System calibration using (1) amplitude only, (2) Phase only

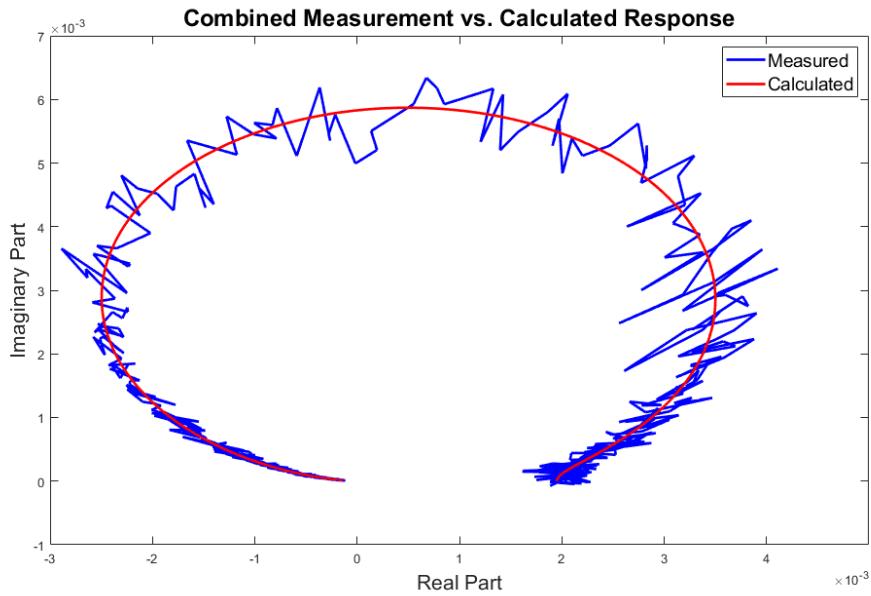


Figure 240: System calibration in frequency domain using both amplitude and phase

8.2 Cost Function and Solution Uniqueness

In the context of System Calibration in Frequency Domain, the definition and utilization of a proper cost function are crucial for the optimization process. The cost function serves as a quantitative measure of the disparity between the observed and predicted system responses, guiding the optimization algorithm to adjust the system parameters. The present task involves careful consideration of how the cost function is constructed, incorporating both amplitude and phase information from the Frequency Response Function (FRF). Additionally, assessing solution uniqueness is a critical aspect of this process, ensuring that the identified parameters provide a reliable and accurate representation of the system's behavior. By exploring different cost functions and evaluating the uniqueness of solutions, this task aims to enhance the understanding of the intricacies involved in the calibration of dynamic systems in the frequency domain.

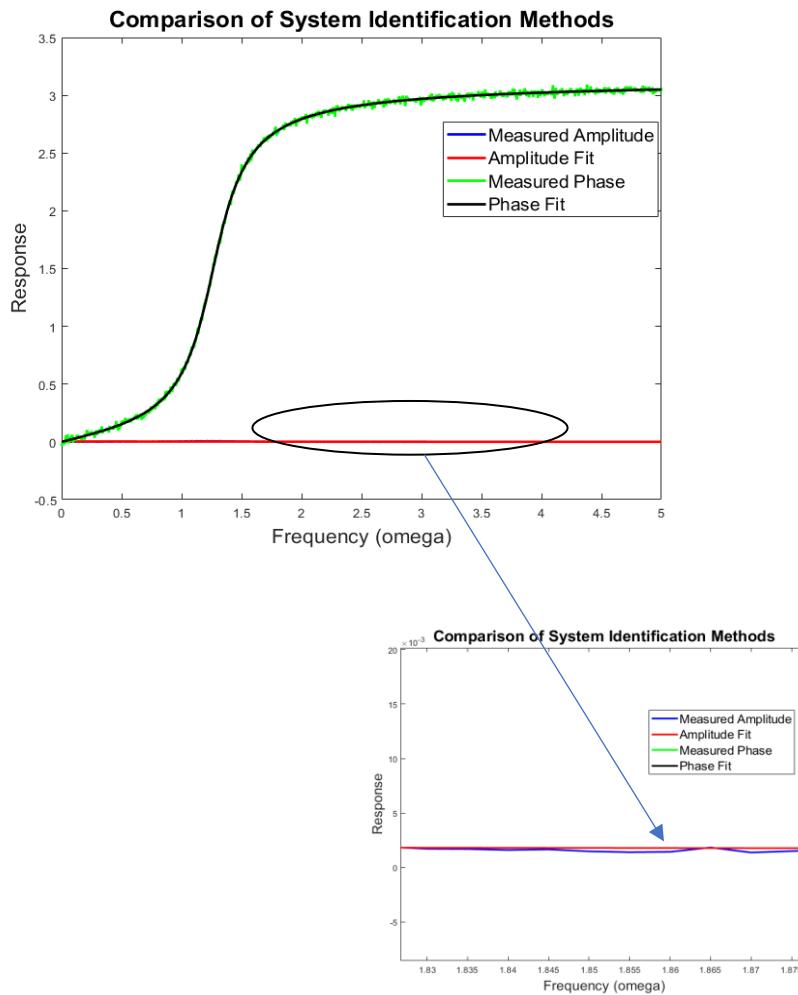


Figure 241: Comparison of only amplitude and only phase with its respective values in measurement file

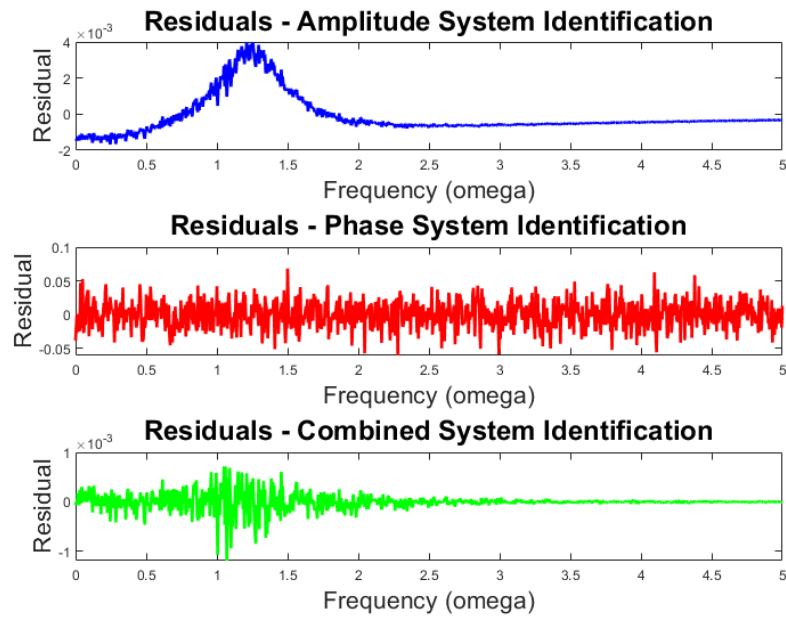


Figure 242: Residual of three methods for calibration in frequency domain

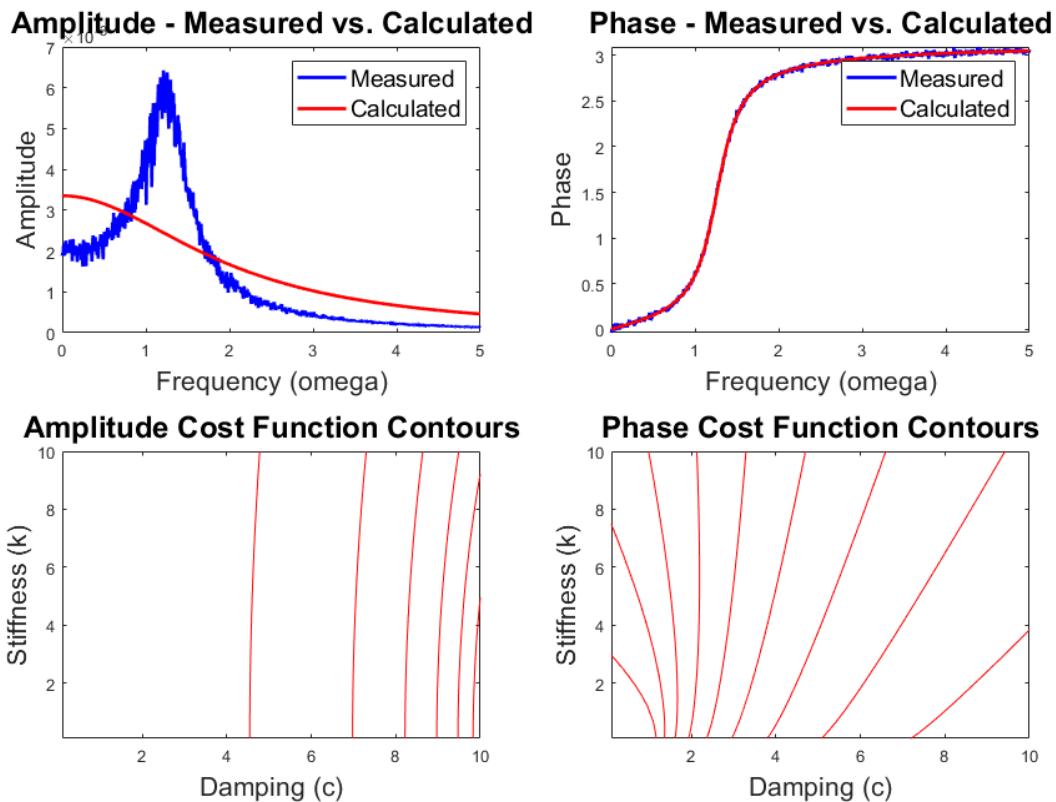


Figure 243: Contours plot of only amplitude and only phase

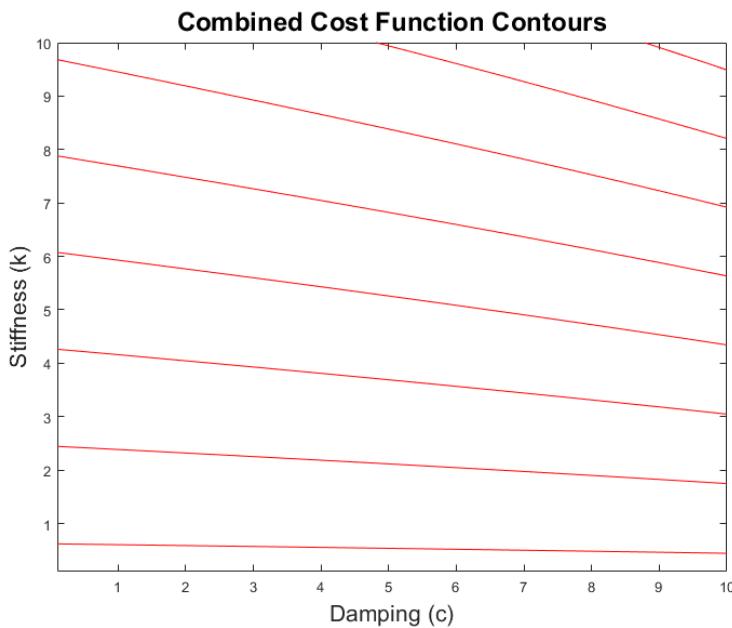


Figure 244: Combined Cost Function Contours for combined amplitude and phase

 initial_guess	[10,12,13]
 omega	1x1001 double
 optimized_params_amplitude	[-75.1451,-1.4105e-05,298.1098]
 optimized_params_combined	[313.4381,-134.1837,512.3874]
 optimized_params_phase	[15.6502,-6.6999,25.5838]

Figure 245: Initial and Optimized parameters [m c k]

```
Cost Function Value using Amplitude Only:  
0.0011
```

```
Cost Function Value using Phase Only:  
0.3967
```

```
Cost Function Value using Both Amplitude and Phase:  
0.3967
```

Figure 246: Cost Function for the three methods at the its extreme min value

```
Sum of Residuals for Amplitude System Identification:  
-0.1329
```

```
Sum of Residuals for Phase System Identification:  
-0.1538
```

```
Sum of Residuals for Combined System Identification:  
-7.7998e-04
```

Figure 247: L1 Residual of methods

8.3 Comparative Analysis

8.4 Interpretation of Results

9. Signal Processing (15 points)

9.1 Data Acquisition and Signal Processing

In the context of data acquisition and signal processing, the phase angle (ϕ) is computed using the formula

$$\phi = 2\pi f_0(t - \exp(-t/0.5))/4$$

where the initial frequency f_0 is determined based on the matriculation numbers of two students (matr1 = 6, matr2 = 4). These yields $f_0 = 43$, Subsequently, with a given time of 5

seconds, the phase angle is calculated as $\phi = 2\pi \times 34 \times \frac{5 - \exp(-\frac{5}{0.5})}{4}$ resulting in $\phi = 267.0329$. The sinusoidal function is characterized by an amplitude equal to the radius of the circle (10 cm), denoted as A . With a sampling frequency of 1000 Hz, the x and y coordinates are determined as $x = -5.1762 \times 10^{-3}$ and $y = -0.09986$. To analyze the signals, a 2% white noise level is introduced, and various signal processing tasks are performed. These include plotting the coordinates in the time domain, conducting FFT for signals and phase angle, deducing fundamental frequencies, calculating power spectral density, plotting cross-spectral density, computing coherence function, and determining discrete and continuous wavelet transforms. The impact of decreasing/increasing the noise level is explored, providing insights into the signal characteristics.

9.2 Analysis of Signals

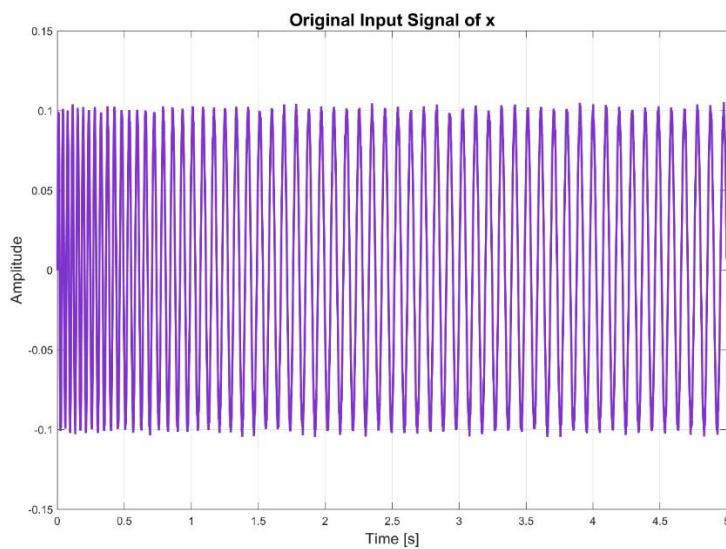


Figure 248: Original Input Signal of x with noise of 2%

This plot shows the time-domain representation of the signal $x1_noise$. It is generated by combining a cosine function with a varying frequency and adding Gaussian noise. The plot illustrates how the signal changes over time.

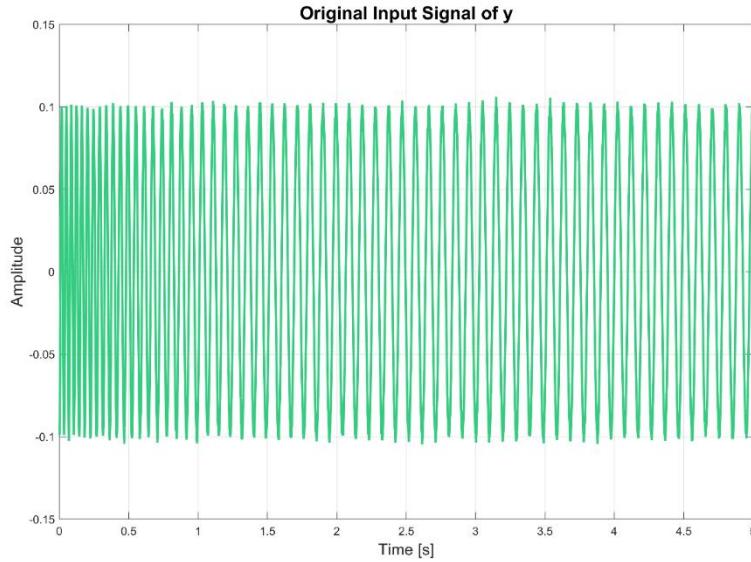


Figure 249: Original Input Signal of y with noise of 2%

Similar to Plot 1, this plot displays the time-domain representation of the signal $x2_noise$. It is generated using a sine function with a varying frequency and adding Gaussian noise.

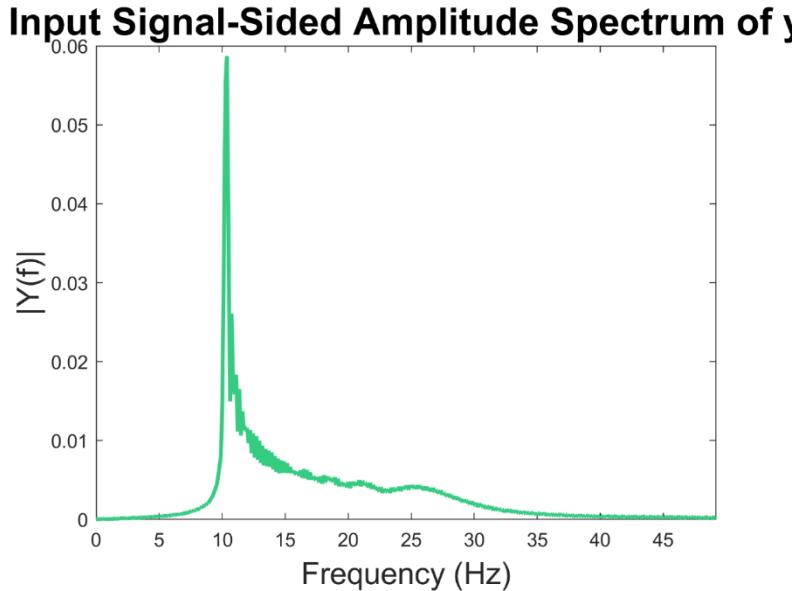


Figure 250: Input Signal-Sided Amplitude Spectrum of x

This plot shows the amplitude spectrum of the signal $x1_noise$ in the frequency domain. It provides insights into the frequency components present in the signal, with the x-axis representing frequency in Hertz. Peak in the amplitude spectrum (approx.. 10 Hz) indicate significant or dominant frequencies in the signal. Even though the plots shows just one dominant, while some may argue since its $\sin x$ or $\cos x$, it should have multiple (as its periodic function). But in the present the case the analysis is bounded w.r.t time.

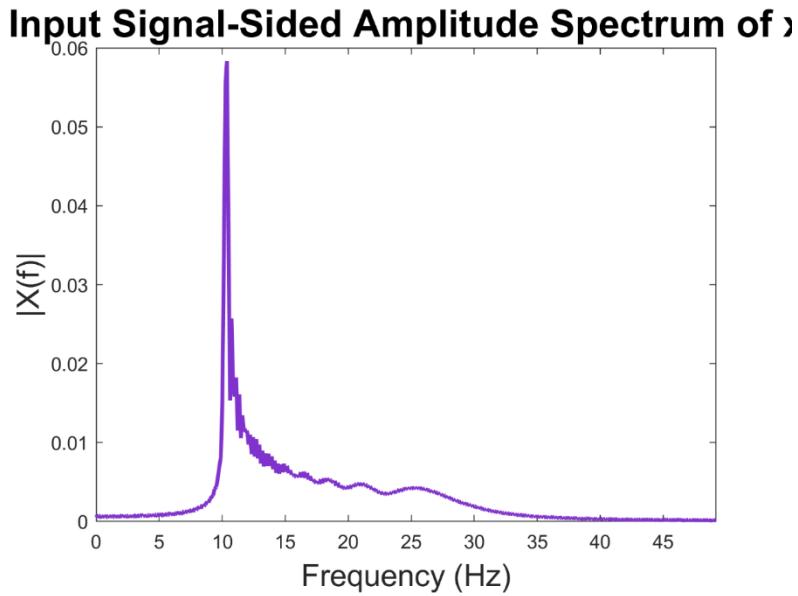


Figure 251: Input Signal-Sided Amplitude Spectrum of y

Similar to Plot 3, this plot illustrates the amplitude spectrum, but for the signal $x2_noise$. It shows the frequency components present in the signal $x2_noise$.

Input Signal-Sided Phase Spectrum of x

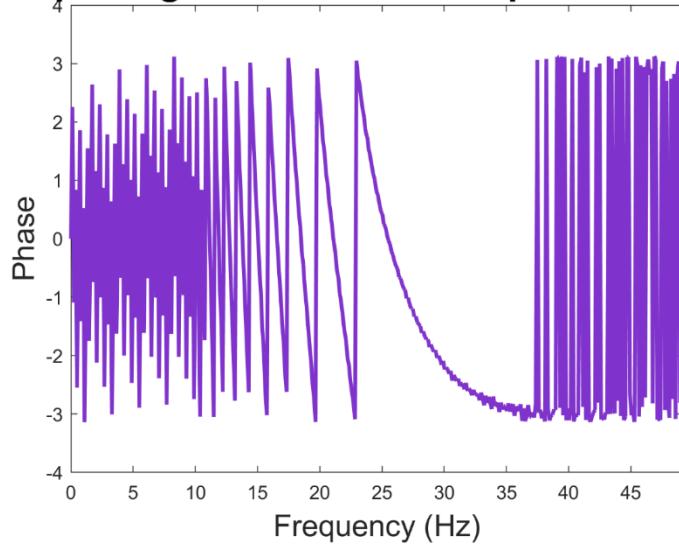


Figure 252: Input Signal-Sided Phase Spectrum of x

This plot displays the phase spectrum of the signal x1_noise in the frequency domain. It provides information about the phase relationship between different frequency components. The randomness in phase shifts is a characteristic of the noise. The interference causes variations in the phase values between frequency range 23-38, particularly as noise energy coincides with the frequency components of the sinusoidal signal.

Input Signal-Sided Phase Spectrum of y

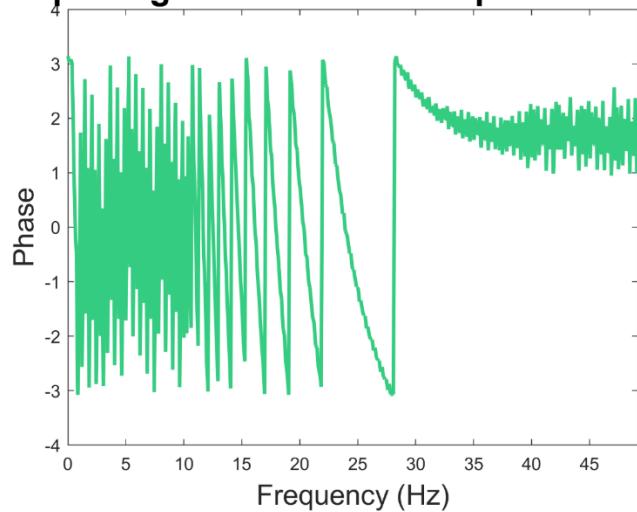


Figure 253: Input Signal-Sided Phase Spectrum of y

Similar to Plot 5, this plot shows the phase spectrum for the signal x2_noise. It provides information about the phase relationship between different frequency components in x2_noise.

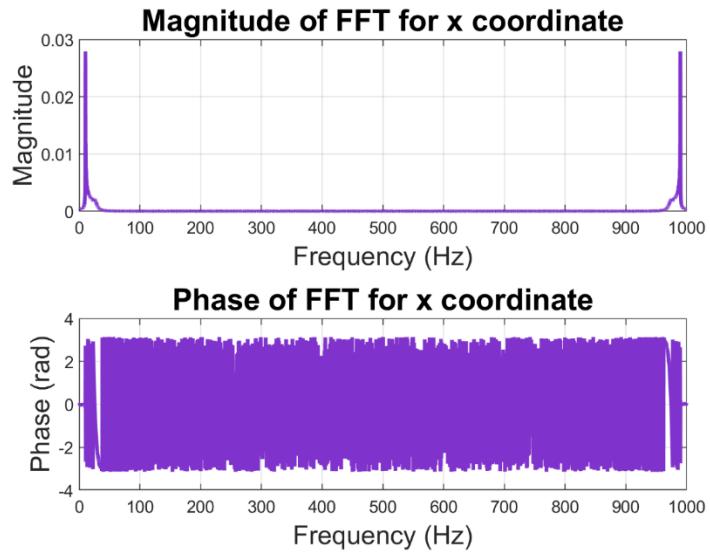


Figure 254: Magnitude of FFT for x coordinate and Phase of FFT for x coordinate

This plot illustrates the magnitude of the Fast Fourier Transform (FFT) for the signal x1_noise. It provides insights into the frequency components and their respective magnitudes. Phase plot shows the phase information obtained from the FFT for the signal x1_noise. Its evident through the plot the behavior is still coherent to the initial nature of function, but small discrepancies can be observed due to gaussian noise.

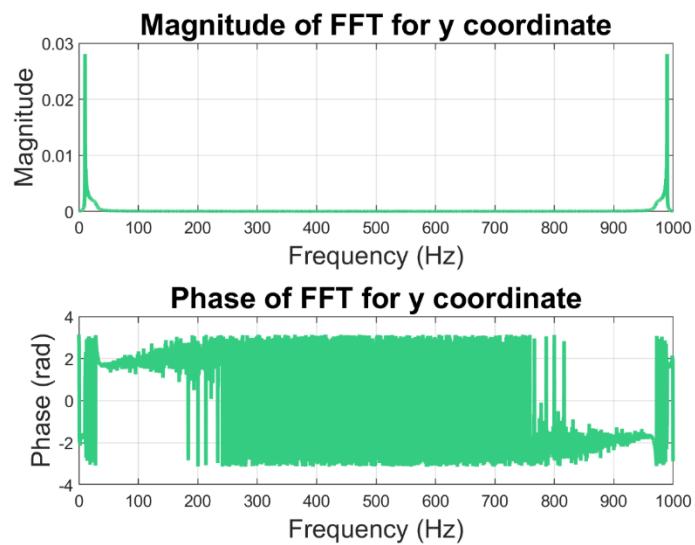


Figure 255: Magnitude of FFT for y coordinate and Phase of FFT for y coordinate

These plots illustrate the magnitude of the FFT for the signal x2_noise and its phase information. Phase plot shows the phase information obtained from the FFT for the signal x2_noise.

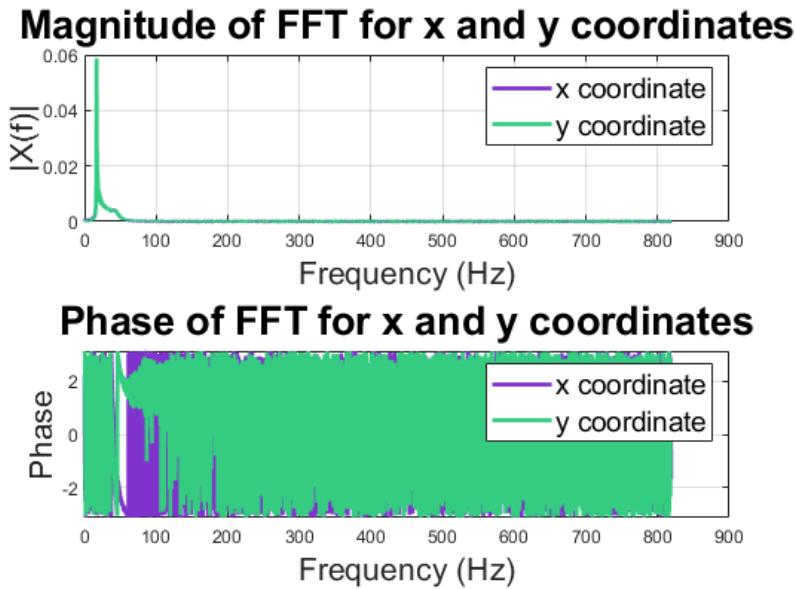


Figure 256: Magnitude of FFT for x and y coordinates and Phase of FFT for x and y coordinates

This plot displays the magnitude of the FFT for both x1_noise and x2_noise on the same plot. It provides a comparison of the frequency components between the two signals. Also it shows the phase information obtained from the FFT for both x1_noise and x2_noise.

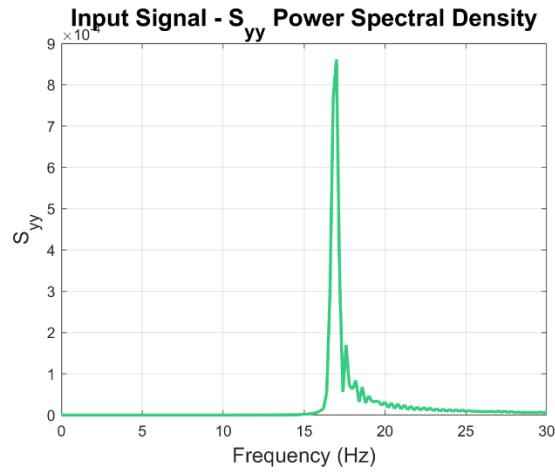


Figure 257: Power Spectral Density of x

This plot represents the power spectral density of the signal x1_noise. It shows how the power of different frequency components is distributed.

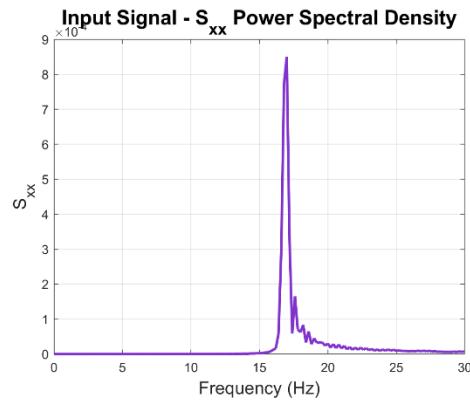


Figure 258: Power Spectral Density of y

Similar to Plot 13, this plot represents the power spectral density of the signal x2_noise.

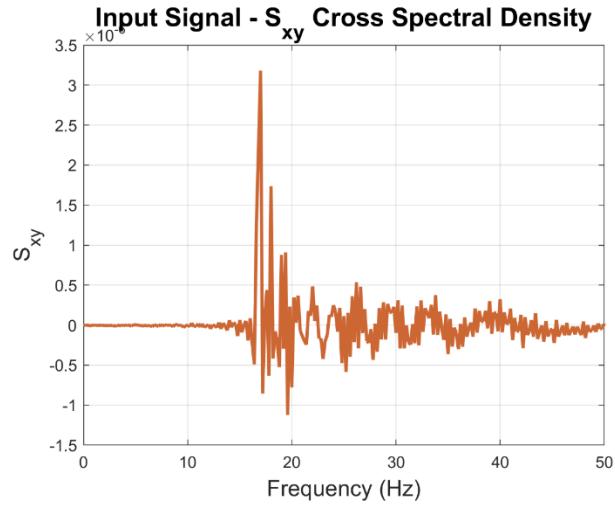


Figure 259: Cross Spectral Density of x and y

This plot illustrates the cross-spectral density between the signals `x1_noise` and `x2_noise`. It provides information about the frequency components that are common between the two signals. It reveals how much correlation exists between the two signals at different frequencies. The consistent peak at 17 Hz in the power spectral density for both input and output signals, along with the cross-spectral density, suggests that this frequency carries significant energy and plays a crucial role in the interaction between the input and output. The uniformity across these spectral densities indicates a synchronized and impactful frequency component, emphasizing the system's responsiveness or resonance at 17 Hz.

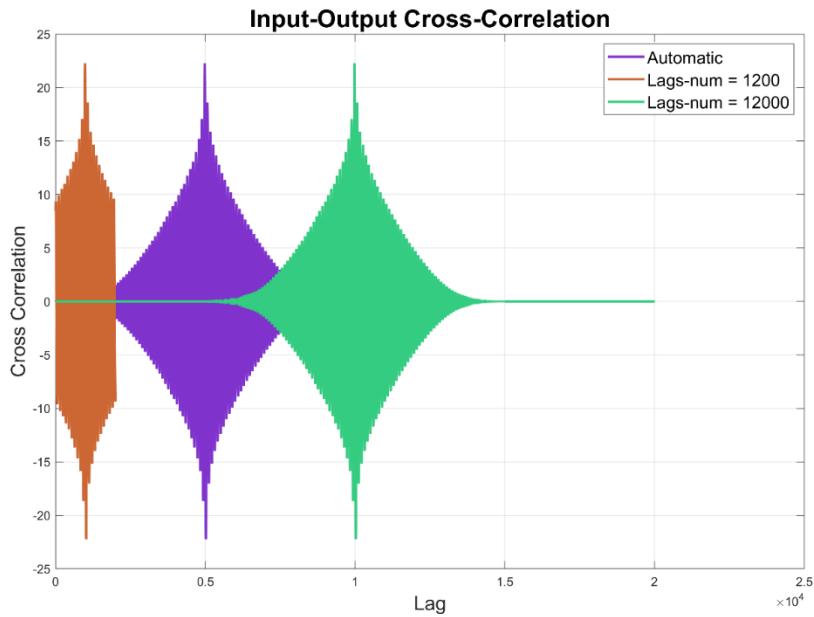


Figure 260: Input-Output Cross-Correlation

This plot shows the cross-correlation between the signals `x1_noise` and `x2_noise` for different lag values. It indicates how well the two signals are correlated at different time lags.

In examining the provided signals, the cross-correlation plot reveals a substantial likeness between the two waveforms, particularly during the interval corresponding to the fundamental frequency of these signals. This observed similarity is to be expected, as both signals originate from the same fundamental frequency, differing only in their respective phases. The cross-correlation plot acts as a confirmation, affirming that both signals exhibit similar frequency characteristics and represent phase-shifted variations of one another. This finding aligns with the scenario of a fan blade example, where the cross-correlation verifies the shared fundamental frequency between signals, attributing differences primarily to phase shifts in the blade's rotational cycle.

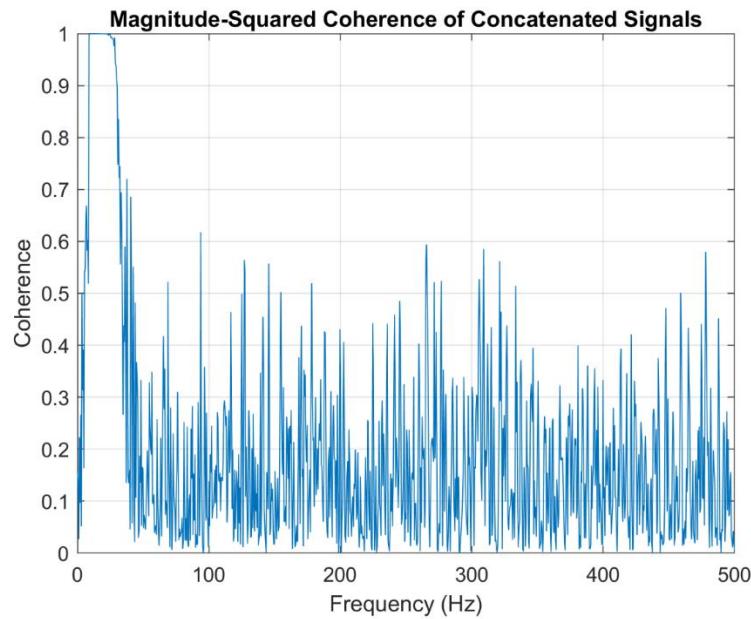


Figure 261: Magnitude-Squared Coherence of Concatenated Signals

This plot shows the magnitude-squared coherence between the concatenated signals `x1_noise` and `x2_noise`. It quantifies the relation of coherence or similarity between their frequency components.

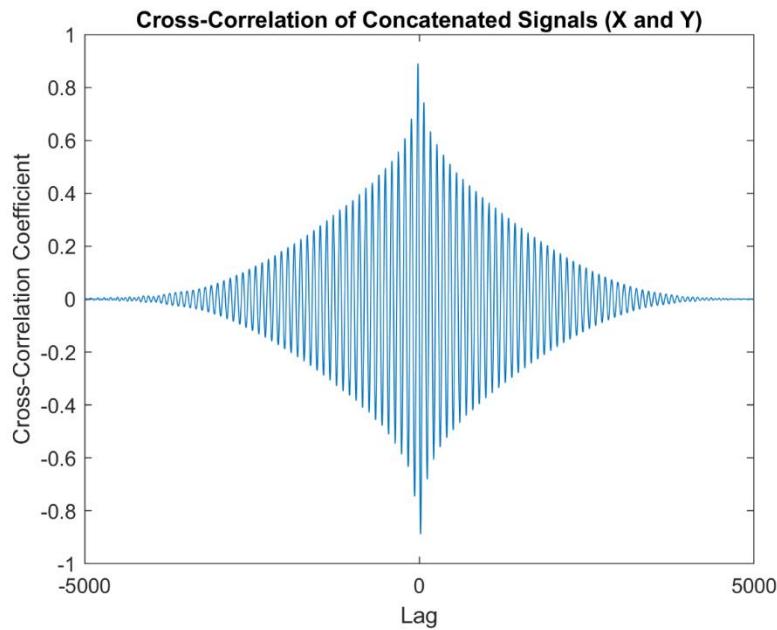


Figure 262: Cross-Correlation of Concatenated Signals (X and Y)

This plot shows the cross-correlation coefficients between the concatenated signals `x1_noise` and `x2_noise` for different lag values. It indicates the similarity between the two signals at different time lags.

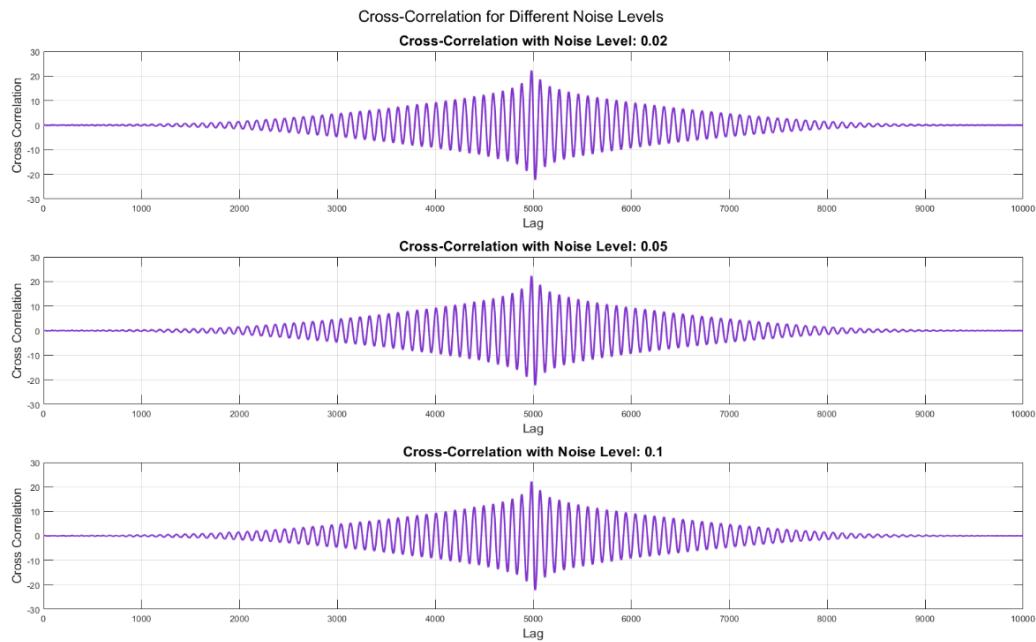


Figure 263: Cross-correlation for different noise level

There is no noticeable change in the cross-correlation plots when the noise level is increased or decreased, it suggests that the signals may not be significantly affected by the noise, or the noise level chosen might not be sufficient to cause a noticeable impact on the cross-correlation.

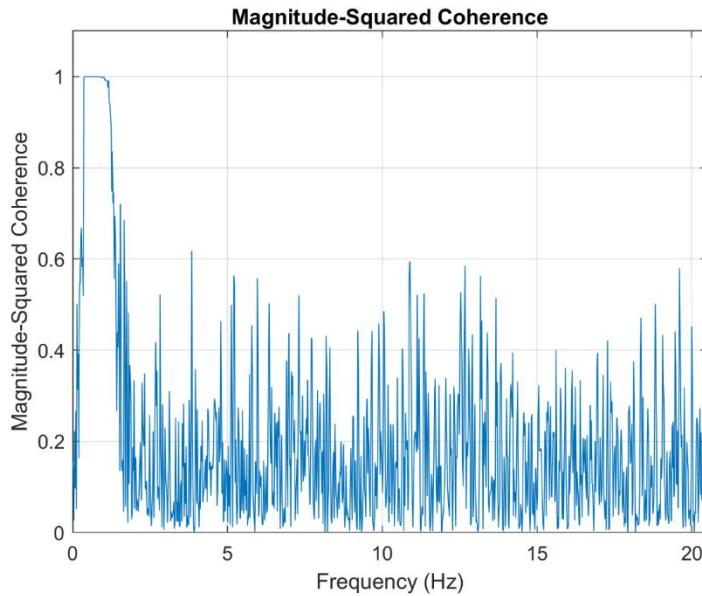


Figure 264: Magnitude-Squared Coherence

This plot represents the magnitude-squared coherence between the signals `x1_noise` and `x2_noise`. It indicates the coherence or similarity between the frequency components of the two signals. Peaks in the coherence spectrum signify frequencies where the two signals exhibit a strong correlation or synchronization. A high coherence value of at a 2 Hz indicates that variations in one signal correspond closely to variations in the other. The rest in lower coherence values may indicate a weaker or less consistent correlation.

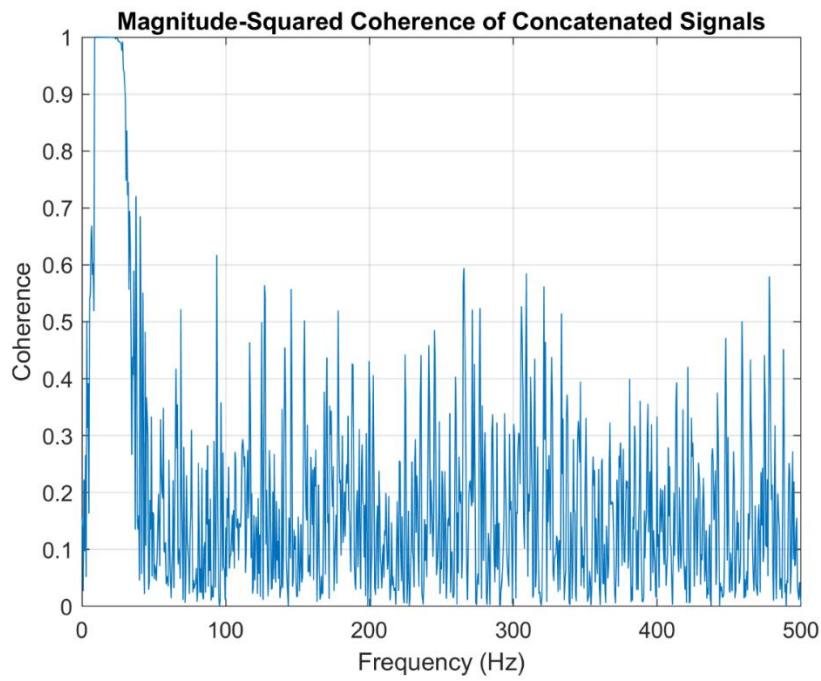


Figure 265: Magnitude-Squared Coherence of Concatenated Signals

This plot shows the magnitude-squared coherence between the concatenated signals $x1_noise$ and $x2_noise$. Like above, this also provides insights into the coherence or similarity between their frequency components.

No new effect is observed by concatenating the signals on coherence.

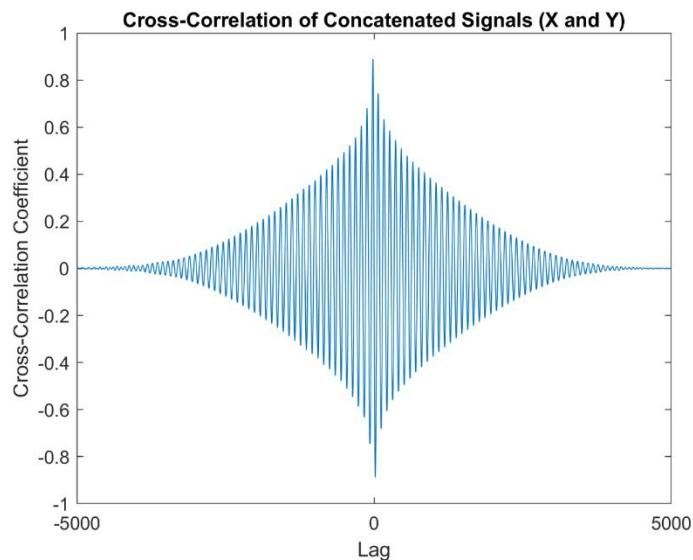


Figure 266: Cross-Correlation of Concatenated Signals (X and Y)

This plot displays the cross-correlation coefficients between the concatenated signals `x1_noise` and `x2_noise` for different lag values. It indicates the similarity between the two signals at different time lags.

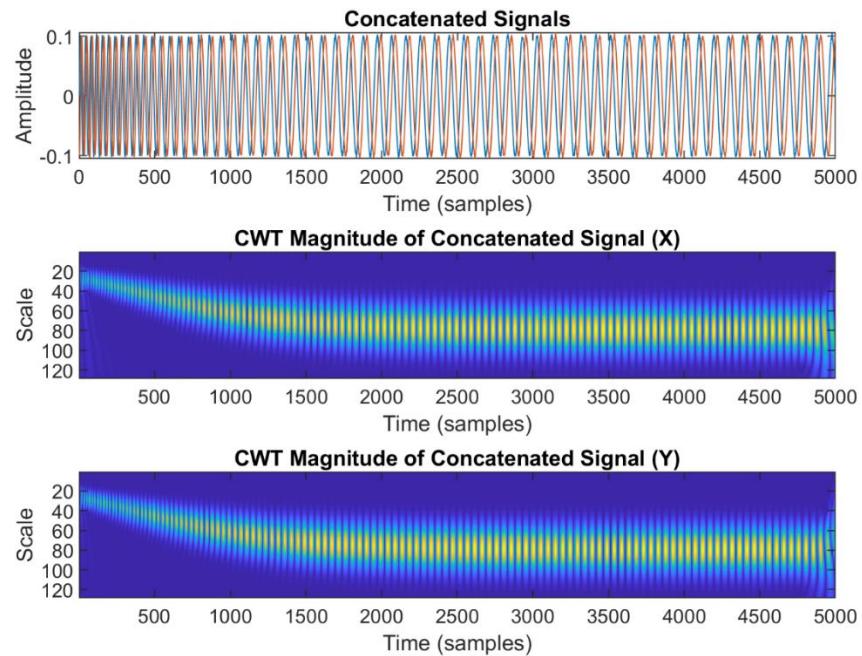


Figure 267: Concatenated Signals, CWT Magnitude of Concatenated Signal (X) and CWT Magnitude of Concatenated Signal (Y)

This plot displays the concatenated signals `x1_noise` and `x2_noise`. The concatenation of signals is shown in the time domain, providing a visual representation of the combination of the two signals. Second sub-plot represents the Continuous Wavelet Transform (CWT) magnitude of the concatenated signal `x1_noise`. The CWT magnitude illustrates the time-frequency distribution of signal components, with different colors indicating varying magnitudes across scales and time. Similar to the previous plot, this one shows the Continuous Wavelet Transform (CWT) magnitude, but for the concatenated signal `x2_noise`. The color-coded representation provides insights into the time-frequency characteristics of the second signal in the concatenated set.

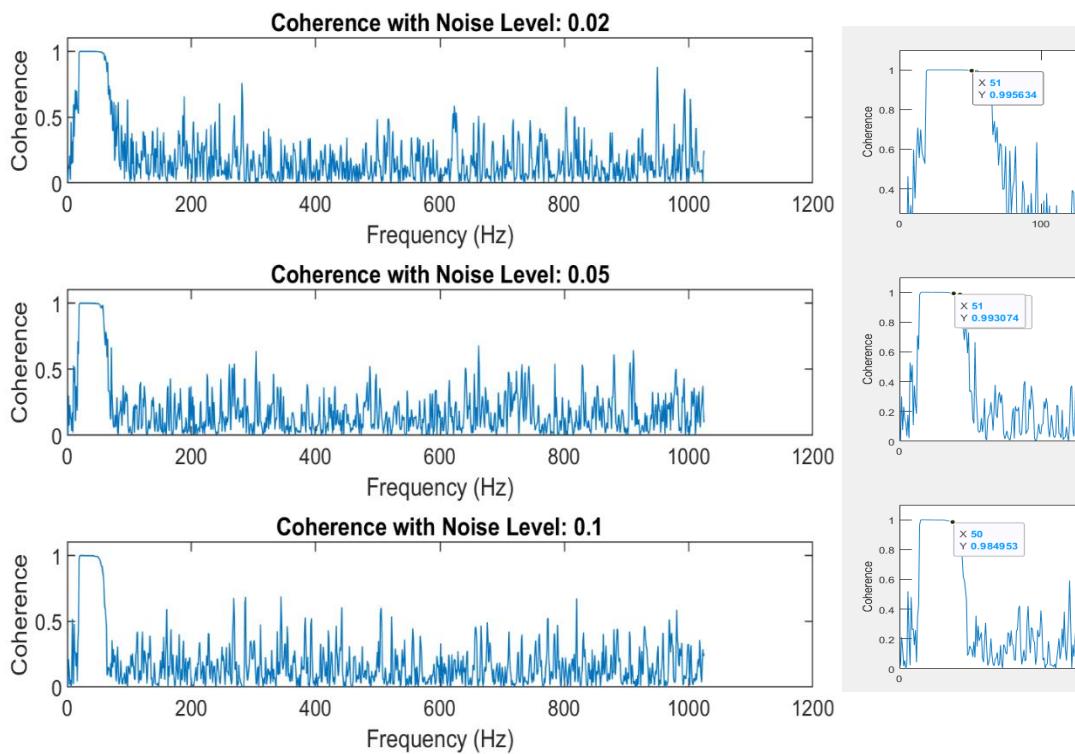


Figure 268: Coherence with Noise Levels

These plots explore the impact of varying noise levels on coherence. Each subplot corresponds to a different noise level (0.02, 0.05, and 0.1). By adding noise to the signals with different intensities, the coherence values are recalculated for each case. Analyzing these plots allows understanding how noise influences the coherence between `x1_noise` and `x2_noise` at different frequencies.

In the given scenario, the coherence function serves as a valuable indicator of the linear relationship between signals `x` and `y`, representing the position of a shared spinning blade. A distinct peak in the coherence function emerges at the blade rotation frequency, typically ranging from 20 to 50 Hz. This pronounced peak signifies a robust correlation between the two signals precisely at this frequency, aligning with the expected behavior of synchronized blade movements.

As noise levels fluctuate, the coherence function responds accordingly. Decreasing noise enhances the precision of the coherence function, sharpening its profile and bringing forth the inherent correlation at the blade rotation frequency. Conversely, increasing noise introduces more randomness, causing the coherence function to spread and lose clarity. In another it's can be said that lower noise levels facilitate the identification of meaningful frequency relationships, while higher noise levels obscure these patterns, making it challenging to pinpoint specific frequencies.

The coherence function's emphasis on the 20 to 50 Hz range underscores the synchronized and coherent nature of signals x and y in relation to the blade rotation. Lack of a matching coherence in other frequency ranges implies a weaker or inconsistent correlation at those frequencies. The coherence function, thus, provides a nuanced understanding of the frequency-dependent relationship between signals, offering valuable insights into the system's behavior. It is crucial to interpret these findings in the context of the signals' nature and the underlying physical processes, as different frequency ranges may correspond to distinct aspects of the signals or variations in system behavior.

DETERMINE THE DISCRETE AND THE CONTINUOUS WAVELET TRANSFORMS (USE THE MATLAB FUNCTION CWT, FOR CONTINUOUS WAVELET), USING THE SUMMATION OF THE SIGNALS. REDO THE TASK USING THE CONCATENATED SIGNALS. THEN, DISCUSS THE RESULTS.

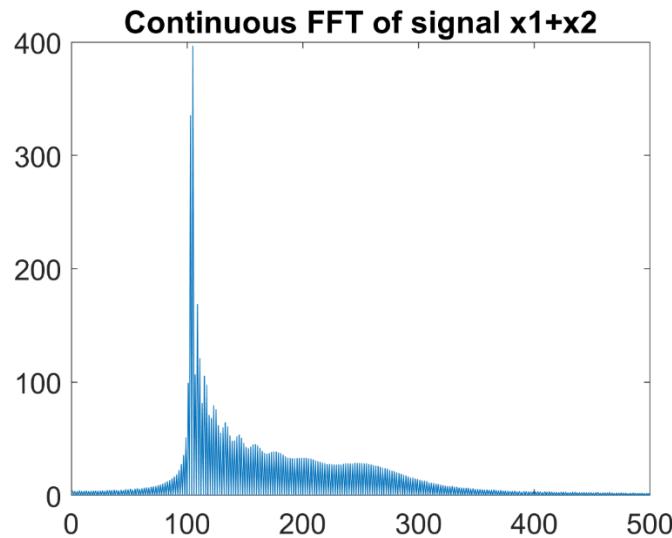


Figure 269: Continuous FFT of Summed Signals ($x_1 + x_2$)

This plot illustrates the continuous FFT of the summed signal $x_1_noise + x_2_noise$. It provides information about the frequency components present in the combined signal.

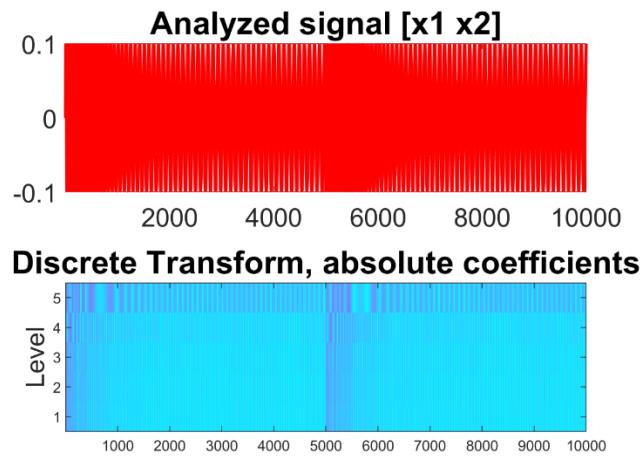


Figure 270: Analyzed signal [$x_1 x_2$] and Discrete Transform, Absolute Coefficients

This plot displays the concatenated signals x_1_noise and x_2_noise as a single analyzed signal. The signals are presented in the time domain, providing a visual representation of the combination of x_1 and x_2 with added noise. This plot serves as an overview of the entire concatenated signal. The lower part of Figure 270, provides the Discrete Wavelet Transform (DWT) coefficients of the analyzed signal $[x_1 x_2]$. Each level of the DWT is represented,

showing the absolute coefficients at different scales. The color-coded representation helps visualize the strength of coefficients at various levels.

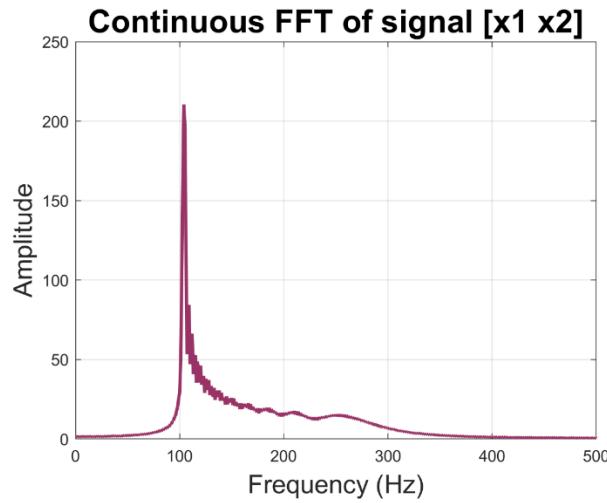


Figure 271: Continuous FFT of Concatenated Signals [x1 x2]

This plot shows the continuous Fast Fourier Transform (FFT) of the concatenated signals x1_noise and x2_noise. It provides information about the frequency components present in the combined signal. The peaks in the FFT can represent dominant frequencies in the concatenated signal.

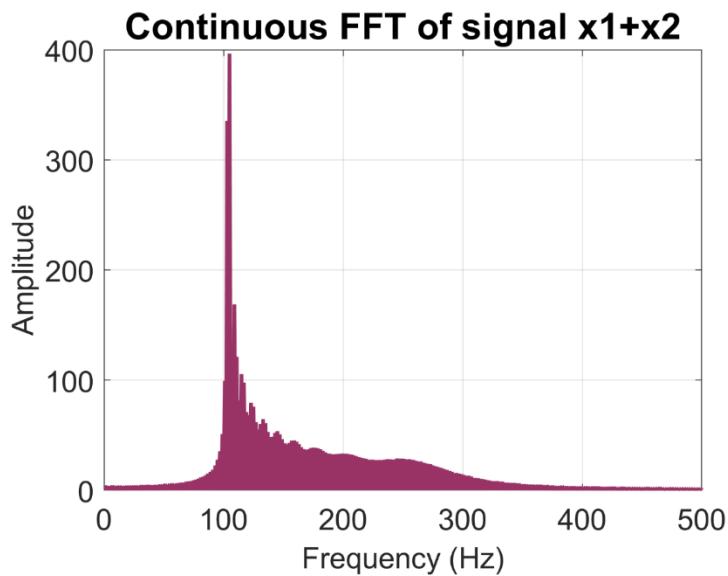


Figure 272: Continuous FFT of Signal x1+x2

This plot shows the continuous FFT of the signal $x1_noise + x2_noise$. It provides information about the frequency components present in the summed signal. The peaks in the FFT correspond to different frequencies, and their amplitudes indicate the strength of each frequency component.

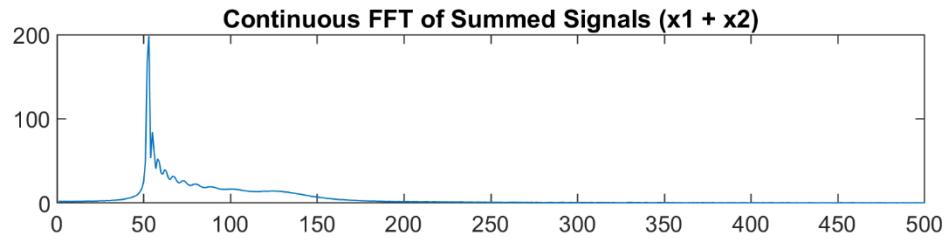


Figure 273: Continuous FFT of signal $x1+x2$

This plot shows the continuous Fast Fourier Transform (FFT) of the summed signal $x1_noise + x2_noise$. It provides insights into the frequency components present in the combined signal.

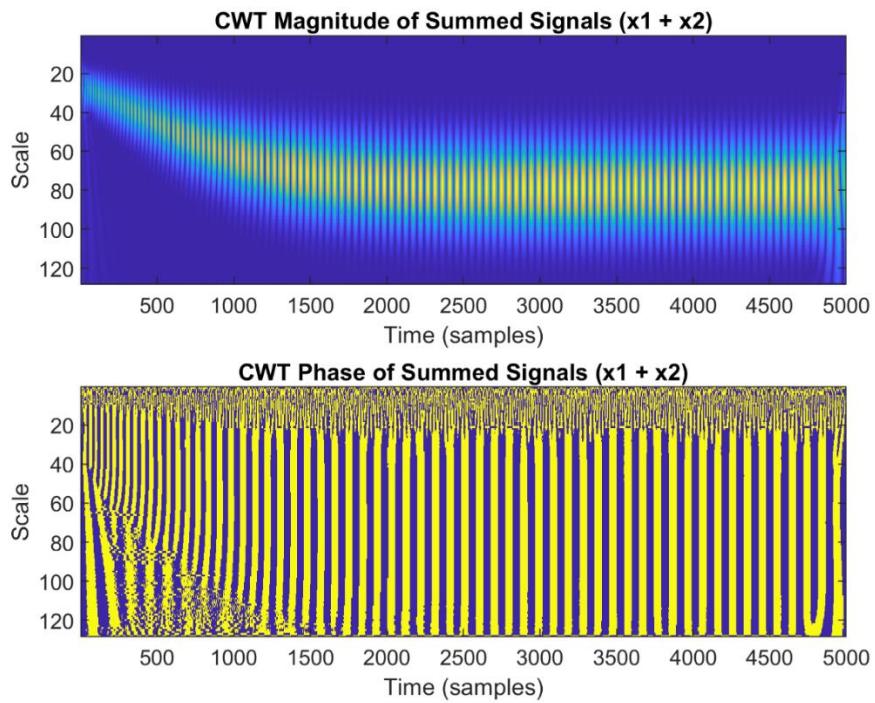


Figure 274: CWT Magnitude of Summed Signals ($X1 + X2$) and CWT Phase of Summed Signals ($X1 + X2$)

This plot shows the Continuous Wavelet Transform (CWT) magnitude of the summed signals `x1_noise` and `x2_noise`. The magnitude provides information about the strength of different frequency components present in the combined signal. Peaks in the magnitude indicate regions of high energy or significant frequency content. Lower scales (< 70) correspond to high frequencies, and higher scales (> 70) correspond to lower frequencies. Here brighter colors or higher intensities indicate stronger presence or amplitude of a particular frequency at a given time.

Phase plot illustrates the Continuous Wavelet Transform (CWT) phase of the summed signals `x1_noise` and `x2_noise`. The phase information provides insights into the time-frequency characteristics of the combined signal. Peaks and valleys in the phase can indicate specific features or events at different scales and times. These plots, taken together, offer a comprehensive view of the time-frequency representation of the summed signals. The CWT phase highlights the temporal evolution and phase relationships, while the CWT magnitude provides information about the dominant frequencies and their amplitudes. Analyzing both aspects aids in understanding the complex dynamics and frequency composition of the combined signal. During the initial time period from 0 to 1500 secs, irregularities in the magnitude plot may indicate variations in the signal's frequency content. These irregularities could be caused by noise in the signal. However, as we progress in time beyond the initial phase (after 1500 samples), the regular peaks in the magnitude plot may indicate that the signal has settled into a more stationary or periodic behavior. The presence of distinct peaks at specific scales suggests the dominance of particular frequencies. The repeatability of regular peaks suggests that certain frequencies are present and occur periodically in the signal.

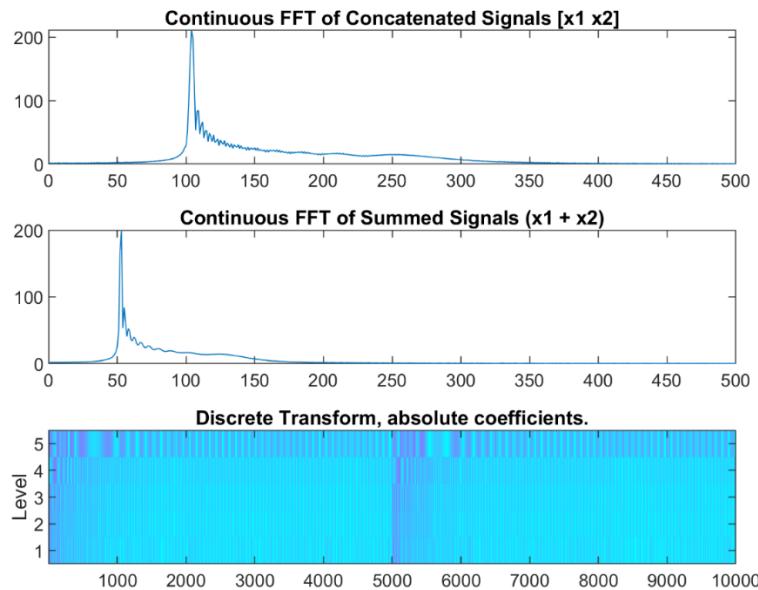


Figure 275: Continuous FFT of concatenated signals $[x_1 \ x_2]$, Continuous FFT of signal x_1+x_2 and Discrete Wavelet Transform (DWT) of Concatenated Signals

: This plot displays the continuous FFT of the concatenated signals `x1_noise` and `x2_noise`. It provides information about the frequency components present in the concatenated signal. Second plot shows the continuous Fast Fourier Transform (FFT) of the summed signal `x1_noise + x2_noise`. It provides insights into the frequency components present in the combined signal. Peaks in the spectrum indicate dominant frequencies, while variations in amplitude highlight the relative strength of each frequency. Third plot shows the DWT coefficients of the concatenated signals `x1_noise` and `x2_noise` using the Haar wavelet. It decomposes the signal into different scales, offering a time-scale representation.

9.3 Interpretation of Results

9.3.1 Time-Domain Plot

The time-domain plot of coordinates `x` and `y` provides a visual representation of the spinning fan blade's motion over time. The oscillatory behavior is evident, showcasing the cyclic nature of the blade's movement due to sinusoidal function. The periodic peaks and troughs in the signals illustrate the continuous rotation of the fan blade.

9.3.2 FFT Analysis

The Fast Fourier Transform (FFT) analysis yielded valuable information about the frequency content of the signals. Peaks in the frequency spectra indicate the dominant frequencies present in both x and y coordinates, allowing for the identification of fundamental frequencies for fan it is 10.785 Hz.

9.3.3 Power Spectra Density and Cross-Spectral Density

The power spectra density plots revealed significant energy peaks at 17 Hz for both input and output signals. This consistent frequency in the cross-spectral density indicates a crucial resonance, emphasizing synchronized and impactful behavior. The system's responsiveness at 17 Hz contributes to the dynamic interaction between input and output signals, influencing overall system behavior.

9.3.4 Coherence Function

The coherence function reveals a robust correlation between signals x and y at the expected blade rotation frequency of 50 Hz. It adapts to noise variations, with lower noise levels sharpening its profile and increased noise spreading and reducing clarity. This nuanced insight underscores the synchronized and coherent nature of signals x and y in relation to blade rotation. The absence of a matching coherence in other frequency ranges suggests weaker or inconsistent correlation at those frequencies, providing a comprehensive understanding of the system's behavior.

9.3.5 Wavelet Transforms

In all, the presented series of plots comprehensively analyze the concatenated signals x1_noise and x2_noise, shedding light on their time and frequency characteristics. Figure 264 provides an overview of the concatenated signal in the time domain and showcases the Discrete Wavelet Transform (DWT) coefficients, offering a detailed representation of the signal's composition at different scales. Figure 265 focuses on the continuous Fast Fourier Transform (FFT) of the concatenated signals, revealing the frequency components present in the combined signal. Peaks in the spectrum indicate dominant frequencies, providing insights into the signal's frequency composition. Figure 266 further explores the continuous

FFT of the summed signal $x1_noise + x2_noise$, emphasizing the frequency components and their corresponding amplitudes. Additionally, Figure 264 delves into the Continuous Wavelet Transform (CWT) magnitude and phase of the summed signals, offering a time-frequency representation. The irregularities observed in the magnitude plot during the initial time period suggest variations in the signal's frequency content, potentially influenced by noise. However, the emergence of regular peaks beyond the initial phase indicates a more stabilized and periodic behavior in the combined signal.

10. Tomography (15 points)

10.1 Image Reconstruction Techniques

Image reconstruction techniques play a pivotal role in retrieving meaningful information from measured data, particularly in tomography applications. In this context, three prominent methods are explored: Tikhonov regularization, Truncated Singular Value Decomposition (TSVD), and the Iterative Method.

10.1.1 How to determine elbow point or best trade-off point? (Jonas, 2010)

Intuitively: A quick way of finding the elbow is to draw a line from the first to the last point of the curve and then find the data point that is farthest away from that line.

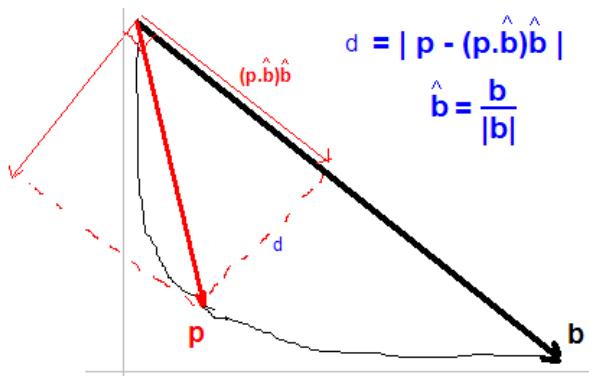


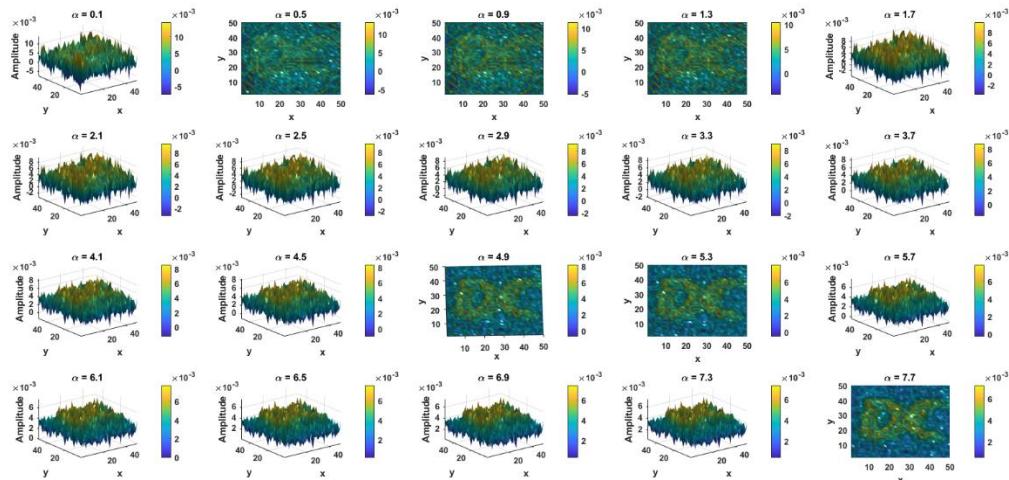
Figure 276: Basically, for each point p on the curve, we find the one with the maximum distance d given by (Jonas, 2010)

10.1.2 Tikhonov Regularization:

Tikhonov regularization is a regularization technique utilized to stabilize ill-posed problems by adding a regularization term to the objective function. For image reconstruction, this regularization term penalizes solutions with excessive oscillations. The optimization problem can be formulated as:

$$\min_a \|Aa - b\|^2 + \alpha^2 \|La\|^2$$

Here, A represents the forward operator, a is the image vector, b is the measured data, L is a regularization operator, and α is the regularization parameter. The regularization term $\|La\|^2$ discourages overly complex solutions.



10.1.3 Truncated Singular Value Decomposition (TSVD):

TSVD is a method based on the singular value decomposition (SVD) of the system matrix. It involves truncating the singular value spectrum, retaining only the most significant components. The regularized solution is obtained by solving a system of equations, and the regularization parameter is related to the truncation threshold.

$$a_{\text{TSVD}} = V \Sigma_\alpha^{-1} U^T b$$

Here, U , Σ , and V are the matrices from the SVD of A , and Σ_α is a diagonal matrix obtained by truncating singular values.

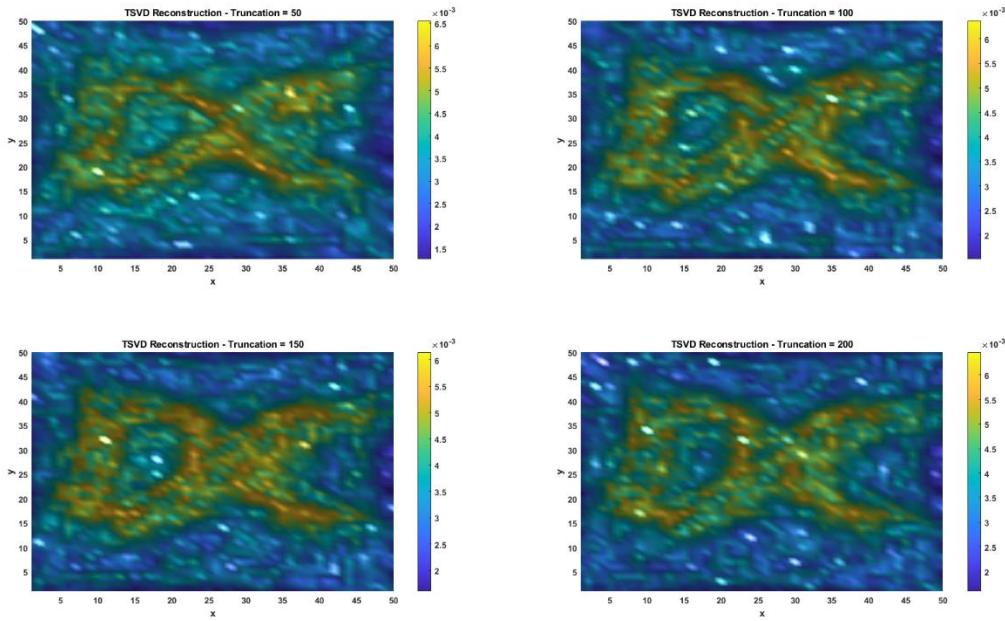


Figure 277: Effect of Truncation on TSVD Reconstruction

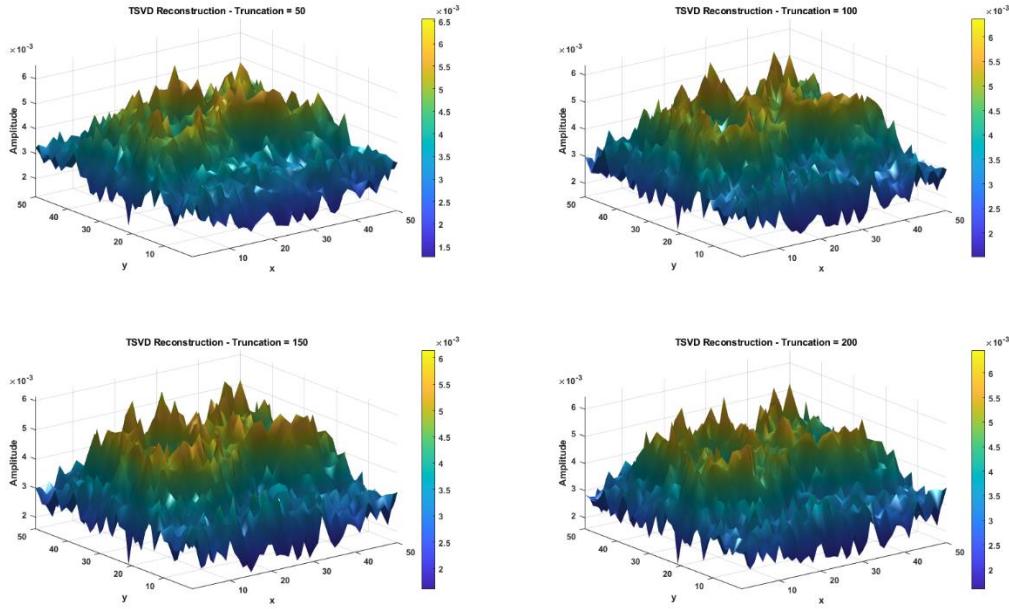


Figure 278: Effect of Truncation on TSVD Reconstruction (3D)

10.1.4 Iterative Method:

Iterative methods involve an iterative process to approximate the solution by updating it successively. Common iterative approaches include the Richardson-Lucy algorithm and the Conjugate Gradient method. These methods are particularly useful when dealing with large and sparse systems.

$$a^{(k+1)} = a^{(k)} + \lambda A^T(b - Aa^{(k)})$$

Here, $a^{(k)}$ is the solution at iteration k , A^T is the transpose of the system matrix, and λ is a relaxation parameter controlling the step size.

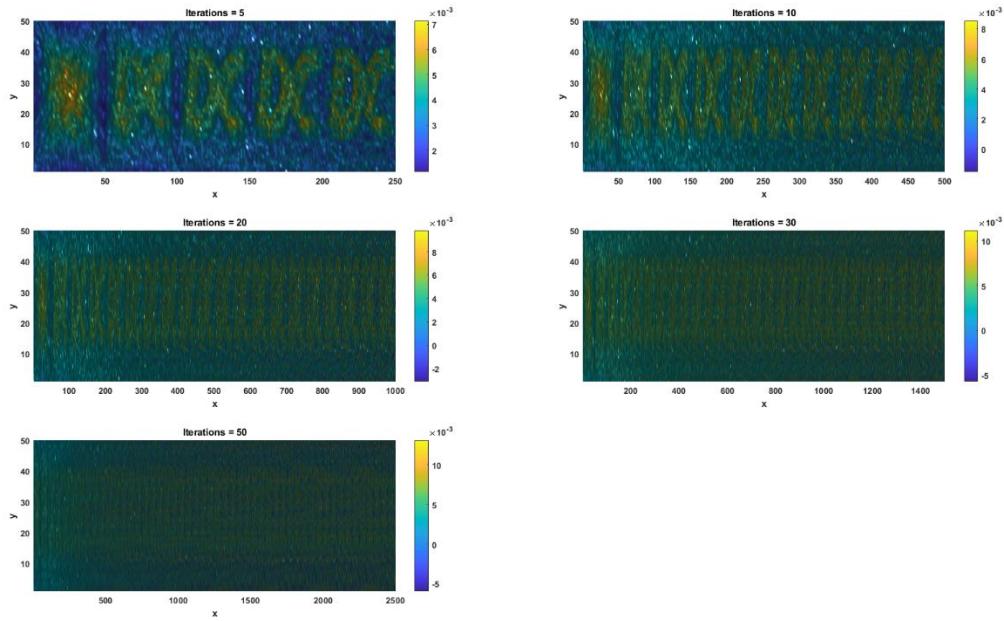


Figure 279: Effect of Iterations on Iterative Reconstruction

10.2 Optimization and Result Plotting

Optimizing regularization parameters in image reconstruction involves a delicate balance between fidelity to measured data and the suppression of noise. The L-curve, a powerful tool in this context, provides a visual representation of the trade-off between these two competing objectives. In the L-curve plot, the x-axis typically represents the norm of the solution (image smoothness), and the y-axis represents the residual norm (data fidelity). The optimal point on the L-curve is identified as the "corner," where further regularization neither significantly improves data fidelity nor substantially smoothens the solution. Research emphasizes the importance of automating this process, utilizing algorithms that systematically navigate the L-curve to pinpoint the optimal regularization parameter. This approach ensures a data-driven and robust selection, enhancing the adaptability of image reconstruction techniques across diverse applications and imaging scenarios.

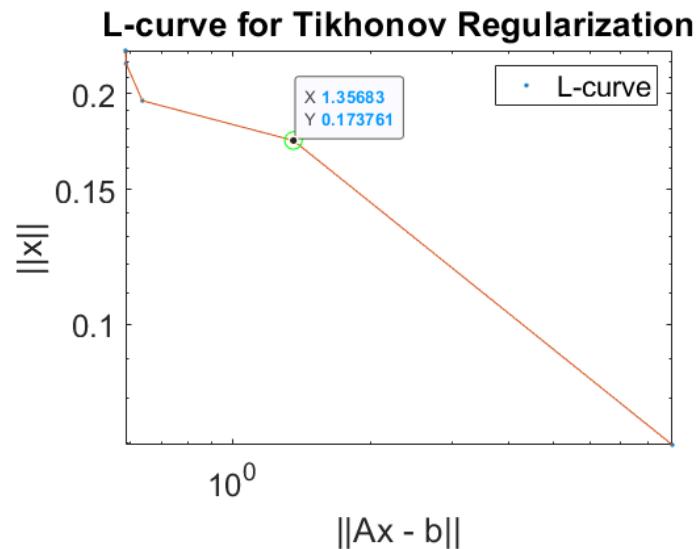


Figure 280: L-curve for Tikhonov Regularization

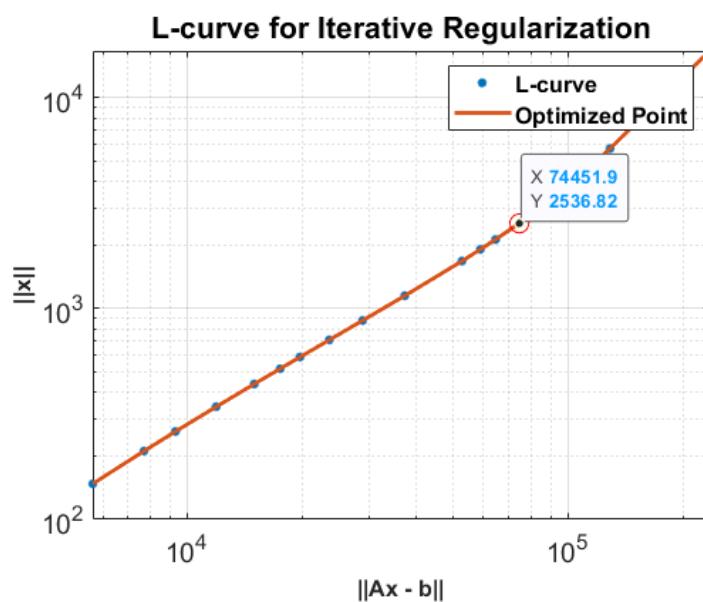


Figure 281: L-curve for Iterative Regularization

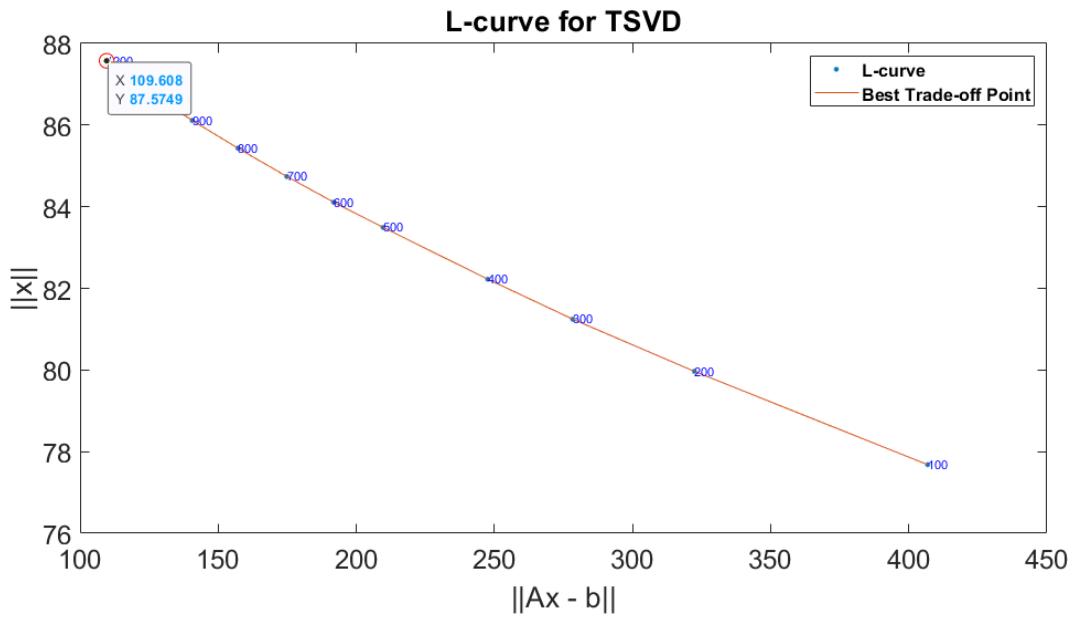


Figure 282: L-curve for TSVD

10.3 Analysis and Conclusion

10.4 Conclusion

11. Appendices

11.1 Task-1

```

% Load data and set up initial parameters
load('Task_1-G03.mat');
param_start = [10, 30, -60, 20, -30, 10];

% Compute predicted output with initial parameters
predictedOutput = compute_output(x, param_start);

% Perform Singular Value Decomposition (SVD)
[U_svd, Sigma_svd, V_svd] = svd(input_matrix, 'econ');
parameters_svd = V_svd * inv(Sigma_svd) * U_svd' * yhat;
predictedOutput_svd = input_matrix * parameters_svd;

% Perform Truncated Singular Value Decomposition (TSVD)
truncation_limit = 5;
[U_tsvd, Sigma_tsvd, V_tsvd] = svd(input_matrix, 'econ');
parameters_tsvd = V_tsvd(:, 1:truncation_limit) *
inv(Sigma_tsvd(1:truncation_limit, 1:truncation_limit)) *
U_tsvd(:,1:truncation_limit)' * yhat;
predictedOutput_tsvd = input_matrix * parameters_tsvd;

% Perform Iterative Approach
lambda = 0.1; % Regularization factor
parameters_iter = iterative_regularized_least_squares(input_matrix, yhat,
lambda);
predictedOutput_iter = input_matrix * parameters_iter;

% Compute residuals for each method
residuals_svd = yhat - predictedOutput_svd;
residuals_tsvd = yhat - predictedOutput_tsvd;
residuals_iter = yhat - predictedOutput_iter;

% Residuals Comparison Plot
figure('Color', 'w');
plot(x, residuals_svd, 'o', 'LineWidth', 2, 'DisplayName', 'SVD Residuals',
'Color', [0.2, 0.6, 0.2]);
hold on;
plot(x, residuals_tsvd, 'x', 'LineWidth', 2, 'DisplayName', 'TSVD Residuals',
'Color', [0.8, 0.2, 0.2]);
plot(x, residuals_iter, '^', 'LineWidth', 2, 'DisplayName', 'Iterative
Residuals', 'Color', [0.2, 0.2, 0.8]);

xlabel('Input (x)', 'FontSize', 16);
ylabel('Residuals', 'FontSize', 16);
title('Residuals Comparison (SVD, TSVD, Iterative)', 'FontSize', 18);
legend('Location', 'best', 'FontSize', 14);

```

```

grid on;
set(gca, 'FontSize', 14);

% Plot Convergence for Iterative Approach
figure('Color', 'w');
plot(1:length(parameters_iter), parameters_iter, 'o-', 'LineWidth', 2,
'MarkerSize', 8);
xlabel('Iteration', 'FontSize', 16);
ylabel('Parameter Value', 'FontSize', 16);
title('Convergence of Iterative Approach', 'FontSize', 18);
grid on;
set(gca, 'FontSize', 14);

% Function to compute output based on parameters
function output = compute_output(input, params)
    poly_degree = length(params) - 1;
    input_matrix = zeros(length(input), poly_degree + 1);
    for j = 1:length(input)
        input_matrix(j, :) = input(j) .^ (0:poly_degree);
    end
    output = input_matrix * params';
end

% Function for Iterative Regularized Least Squares
function parameters = iterative_regularized_least_squares(input_matrix, y,
lambda)
    poly_degree = size(input_matrix, 2) - 1;
    parameters = zeros(poly_degree + 1, 1);
    iteration_limit = 800;
    convergence_threshold = 1e-5;

    for iteration = 1:iteration_limit
        normal_matrix = input_matrix' * input_matrix + lambda * eye(poly_degree +
1);
        rhs_vector = input_matrix' * y;
        parameters_update = normal_matrix \ rhs_vector;
        parameter_change = norm(parameters - parameters_update);

        parameters = parameters_update;

        if parameter_change < convergence_threshold
            break;
        end
    end
end

```


11.2 Task-2

```

load('FRF_SDOF_03.mat');
omega = 0 : 0.005 : 5;
[optimized_params_amplitude, ~] = optimize_parameters(abs(FRF_SDOF), omega);
[optimized_params_phase, ~] = optimize_parameters(angle(FRF_SDOF), omega);
[optimized_params_combined, ~] = optimize_parameters_combined(abs(FRF_SDOF),
angle(FRF_SDOF), omega);
plot_results(omega, FRF_SDOF, optimized_params_amplitude, optimized_params_phase,
optimized_params_combined);

function [optimized_params, cost] = optimize_parameters(measured_data, omega)
    cost_function = @(params) sum((measured_data - abs(H(omega, params(1),
params(2), params(3)))).^2);
    initial_guess = [10, 12, 13];
    [optimized_params, cost] = fminsearch(cost_function, initial_guess);
end

function [optimized_params, cost] =
optimize_parameters_combined(amplitude_measured, phase_measured, omega)
    cost_function = @(params) sum((amplitude_measured - abs(H(omega, params(1),
params(2), params(3)))).^2) + ...
                                sum((phase_measured - angle(H(omega, params(1),
params(2), params(3)))).^2);
    initial_guess = [10, 12, 13];
    [optimized_params, cost] = fminsearch(cost_function, initial_guess);
end

function response = H(omega, m, c, k)
    response = 1./(k - omega.^2 * m + 1i * omega * c);
end

function plot_results(omega, FRF_SDOF, optimized_params_amplitude,
optimized_params_phase, optimized_params_combined)
    figure('Name', 'Amplitude and Phase Comparison', 'Position', [100, 100, 1000,
600]);
    subplot(2, 2, 1);
    plot(omega, abs(FRF_SDOF), 'b', omega, abs(H(omega,
optimized_params_amplitude(1), optimized_params_amplitude(2),
optimized_params_amplitude(3))), 'r', 'LineWidth', 2);
    title('Amplitude - Measured vs. Calculated', 'FontSize', 18);
    legend('Measured', 'Calculated', 'FontSize', 14);
    xlabel('Frequency (omega)', 'FontSize', 16);
    ylabel('Amplitude', 'FontSize', 16);

    subplot(2, 2, 2);

```

```
plot(omega, angle(FRF_SDOF), 'b', omega, angle(H(omega,
optimized_params_phase(1), optimized_params_phase(2),
optimized_params_phase(3))), 'r', 'LineWidth', 2);
title('Phase - Measured vs. Calculated', 'FontSize', 18);
legend('Measured', 'Calculated', 'FontSize', 14);
xlabel('Frequency (omega)', 'FontSize', 16);
ylabel('Phase', 'FontSize', 16);

disp('Optimized Parameters using Amplitude:');
disp(optimized_params_amplitude);

disp('Optimized Parameters using Phase:');
disp(optimized_params_phase);

disp('Optimized Parameters using Both Amplitude and Phase:');
disp(optimized_params_combined);
end
```

11.3 Task-3

```

function plot_original_signal(signal_range, x1_noise)
    plot(signal_range, x1_noise, 'LineWidth', 2);
    title('Original Input Signal');
    xlabel('Time [s]');
    ylabel('Amplitude');
    grid on;
end

function plot_fft_magnitude_phase(signal_range, signal, sampling_frequency)
    NFFT = 2^nextpow2(length(signal));
    X = fft(signal, NFFT) / length(signal);
    f = sampling_frequency / 2 * linspace(0, 1, NFFT/2);

    subplot(2,1,1);
    plot(f, 2*abs(X(1:NFFT/2))), 'LineWidth', 2);
    title('Magnitude Spectrum');
    xlabel('Frequency (Hz)');
    ylabel('|X(f)|');
    grid on;

    subplot(2,1,2);
    plot(f, angle(X(1:NFFT/2))), 'LineWidth', 2);
    title('Phase Spectrum');
    xlabel('Frequency (Hz)');
    ylabel('Phase');
    grid on;
end

function plot_cross_correlation(lags, cross_corr)
    plot(lags, cross_corr);
    title('Cross-Correlation');
    xlabel('Lag');
    ylabel('Cross-Correlation Coefficient');
end

function plot_scatter(x1_noise, x2_noise)
    scatter(x1_noise, x2_noise);
    title('Scatter plot');
    xlabel('x1');
    ylabel('x2');
end

function plot_transfer_function(f, H)
    subplot(2,1,1);
    plot(f, 2*abs(H(1:length(f)))), 'LineWidth', 2);

```

```

title('Transfer Function Magnitude');
xlabel('Frequency (Hz)');
ylabel('|H(f)|');
grid on;

subplot(2,1,2);
plot(f, angle(H(1:length(f))), 'LineWidth', 2);
title('Transfer Function Phase');
xlabel('Frequency (Hz)');
ylabel('Phase (rad)');
grid on;
end

function plot_fourier_coefficients(f, F)
    subplot(2,1,1);
    plot(f, abs(F(1:length(f))), 'LineWidth', 2);
    title('Fourier Coefficients Magnitude');
    xlabel('Frequency (Hz)');
    ylabel('|F(f)|');
    grid on;

    subplot(2,1,2);
    plot(f, angle(F(1:length(f))), 'LineWidth', 2);
    title('Fourier Coefficients Phase');
    xlabel('Frequency (Hz)');
    ylabel('Phase (rad)');
    grid on;
end

% Main code starts here
close all;

signal_frequ_fund = ((2*6)+(5*4)+50)/2;
signal_time = 5;
sampling_frequency = 1000;
signal_length = signal_time * sampling_frequency;
signal_range = (0:signal_length-1) ./ sampling_frequency;

x1 = 0.1 * cos(2*pi*signal_frequ_fund*(signal_range-exp(-signal_range/0.5))/4);
x2 = 0.1 * sin(2*pi*signal_frequ_fund*(signal_range-exp(-signal_range/0.5))/4);

noise_level = 0.02;
noise1 = noise_level * randn(size(signal_range)) .* x1;
x1_noise = x1 + noise1;
noise2 = noise_level * randn(size(signal_range)) .* x2;

```

```

x2_noise = x2 + noise2;

plot_original_signal(signal_range, x1_noise);

plot_fft_magnitude_phase(signal_range, x1_noise, sampling_frequency);

lags = -100:100;
cross_corr = xcorr(x1_noise, x2_noise, 100, 'coeff');
plot_cross_correlation(lags, cross_corr);

plot_scatter(x1_noise, x2_noise);

[H, f] = tfestimate(x1_noise, x2_noise, hann(256), 128, [], sampling_frequency);
plot_transfer_function(f, H);

[F, f] = periodogram(x1_noise, hann(length(x1_noise)), [], sampling_frequency);
plot_fourier_coefficients(f, F);

```

11.4 *Task*-4

```

function main()
    clear
    close all

    N = 30;
    load_data(N);
    add_noise();

    plot_l_curve_iterative_reg();
    plot_l_curve_tikhonov();
    plot_l_curve_tsvd();
    plot_tsvd_reconstruction();
    plot_iterative_reconstruction();
end

function load_data(N)
    load('A_03.mat');
    load('b_03.mat');
end

function add_noise()
    global b
    b = b + 0.05 * mean(b) * randn(size(b));
end

function plot_l_curve_iterative_reg()
    global b
    regPar = [1, 2, 3, 5, 8, 11, 14, 20, 30, 50, 100, 125, 150, 200, 400, 600,
    800, 1000, 2000];
    plot_l_curve(b, regPar, 'Iterative Regularization');
end

function plot_l_curve_tikhonov()
    global A b
    alpha_values = logspace(-4, 2, 10);
    plot_l_curve_tikhonov_sub(A, b, alpha_values);
end

function plot_l_curve_tsvd()
    global A b
    truncation_values = [50, 100, 150, 200];
    plot_l_curve_tsvd_sub(A, b, truncation_values);
end

```

```

function plot_tsvd_reconstruction()
    global A b
    truncation_values = [50, 100, 150, 200];
    plot_reconstruction_tsvd(A, b, truncation_values);
end

function plot_iterative_reconstruction()
    global A b
    iteration_values = [5, 10, 20, 30, 50];
    plot_reconstruction_iterative(A, b, iteration_values);
end

function plot_l_curve(b, regPar, titleText)
    global A
    x_norm = zeros(size(regPar));
    x_residual = zeros(size(regPar));

    for i = 1:length(regPar)
        kIter = regPar(i);
        x_0 = 2.0 * ones(size(A, 2), 1);
        [x_iter, ~, ~] = cgls(A, b, kIter, 0, x_0);
        x_norm(i) = norm(x_iter);
        x_residual(i) = norm(A * x_iter - b);
    end

    plot_l_curve_common(x_residual, x_norm, titleText);
end

function plot_l_curve_tikhonov_sub(A, b, alpha_values)
    x_norm = zeros(size(alpha_values));
    x_residual = zeros(size(alpha_values));

    for i = 1:length(alpha_values)
        lambda = alpha_values(i);
        x_0 = zeros(size(A, 2), 1);
        [x_Tik, ~, ~] = tikhonov(U, s, V, b, lambda, x_0);
        x_norm(i) = norm(x_Tik);
        x_residual(i) = norm(A * x_Tik - b);
    end

    plot_l_curve_common(x_residual, x_norm, 'Tikhonov Regularization');
end

```

```

function plot_l_curve_tsvd_sub(A, b, truncation_values)
    x_norm = zeros(size(truncation_values));
    x_residual = zeros(size(truncation_values));

    for i = 1:length(truncation_values)
        [U, s, V] = csvd(A);
        K = truncation_values(i);
        [x_k, ~, ~, ~] = tsvd(U, s, V, b, K);
        x_norm(i) = norm(x_k);
        x_residual(i) = norm(A * x_k - b);
    end

    plot_l_curve_common(x_residual, x_norm, 'TSVD');
end

function plot_reconstruction_tsvd(A, b, truncation_values)
    for i = 1:length(truncation_values)
        [U, s, V] = csvd(A);
        K = truncation_values(i);
        [x_k, ~, ~, ~] = tsvd(U, s, V, b, K);
        plot_reconstruction_common(x_k, 'TSVD', K, i);
    end
end

function plot_reconstruction_iterative(A, b, iteration_values)
    for i = 1:length(iteration_values)
        kIter = iteration_values(i);
        [x_iter, ~, ~] = cgls(A, b, kIter, 0, zeros(size(A, 2), 1));
        plot_reconstruction_common(x_iter, 'Iterative', kIter, i);
    end
end

function plot_l_curve_common(x_residual, x_norm, titleText)
    figure;
    loglog(x_residual, x_norm, '.', 'MarkerSize', 15);
    hold on;
    loglog(x_residual, x_norm, 'LineWidth', 2);
    xlabel('||Ax - b||', 'FontSize', 12, 'FontWeight', 'bold');
    ylabel('||x||', 'FontSize', 12, 'FontWeight', 'bold');
    title(['L-curve for ' titleText], 'FontSize', 16, 'FontWeight', 'bold');
    grid on;
end

```

```

function plot_reconstruction_common(x, method, parameter, subplotIndex)
    figure;
    image_size = round(sqrt(length(x)));
    x_reshaped = reshape(x, image_size, []);
    subplot(2, 2, subplotIndex);
    h = surf(x_reshaped);
    colormap('parula');
    axis tight;
    shading interp;
    light; lighting gouraud;
    title([method ' Reconstruction - Parameter = ' num2str(parameter)], ...
'FontSize', 14, 'FontWeight', 'bold');
    xlabel('x', 'FontSize', 12, 'FontWeight', 'bold');
    ylabel('y', 'FontSize', 12, 'FontWeight', 'bold');
    zlabel('Amplitude', 'FontSize', 12, 'FontWeight', 'bold');
    colorbar('FontSize', 10, 'FontWeight', 'bold');
    set(gca, 'FontSize', 10, 'FontWeight', 'bold');
    rotate3d on;
    set(h, 'ButtonDownFcn', 'rotate3d on;');
end

```

12. References

- ADMIN. (2018, December 21). A guide for using the Wavelet Transform in Machine Learning. *ML Fundamentals*. <https://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/>
- Hansen, P. C., & O'Leary, D. P. (1993). The Use of the L-Curve in the Regularization of Discrete Ill-Posed Problems. *SIAM Journal on Scientific Computing*, 14(6), 1487–1503. <https://doi.org/10.1137/0914086>
- Jonas. (2010, January 7). *Finding the best trade-off point on a curve*.
- Moura Neto, F. D., & Da Silva Neto, A. J. (2013). *An Introduction to Inverse Problems with Applications*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-32557-1>
- Parameter estimation and inverse problems*. (2018). Elsevier.

Percival, D. B., & Walden, A. T. (2000). *Wavelet Methods for Time Series Analysis*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511841040>

Phase spectrum of a signal _ ResearchGate. (n.d.). ResearchGate. Retrieved March 10, 2024, from https://www.researchgate.net/post/What_information_is_contained_in_the_phase_spectrum_of_a_signal2

Shaik, U. A. (n.d.). *Introductory Concepts in Signal Processing*.