INPUT COMMAND SHAPING USING THE VERSINE FUNCTION WITH PEAK ACCELERATION CONSTRAINT AND NUMERICAL OPTIMIZATION TO MINIMIZE RESIDUAL VIBRATION

by

Pratheek Patil

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Mechanical Engineering

School of Mechanical Engineering

West Lafayette, Indiana

May 2019

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Peter H. Meckl, Chair
School of Mechanical Engineering
Dr. George T. C. Chiu
School of Mechanical Engineering
Dr. Raymond J. Cipra
School of Mechanical Engineering

Approved by:

Dr. Jay P. Gore
Head of the Graduate Program

Dedicated to my loving parents
Kempanna Patil and Kasturi Patil for
their unsolicited support and encouragement.

ACKNOWLEDGMENTS

I would like to extend my sincere gratitude to Dr. Peter Meckl, my thesis advisor for the support, guidance, and encouragement throughout my graduate research. I would also like to thank the other members of my advisory committee, Dr. George Chiu and Dr. Raymond Cipra for their valuable input and willingness to serve in this capacity.

I would also like to extend my appreciation to Bowen Wu and Yumeng Wu who continued to provide inputs and guide me on the project. I would also like to thank my research partners Ruiwen Wei and Aparajit Venkatesh for asking me inquisitive questions during our research meetings which led me to further brainstorm and gain a deeper understanding of the project.

I would also like to extend this gratitude for the staff of E-shop for the school of Mechanical Engineering, as they have been very helpful in solving multiple hardware-related problems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Author: Patil, Pratheek. MSME
Institution: Purdue University
Degree Received: May 2019
Title: Input Command Shaping using the Versine Function with Peak Acceleration Constraint and Numerical Optimization to Minimize Residual Vibration.
Committee Chair: Dr. Peter H. Meckl, School of Mechanical Engineering

Dynamic systems and robotic manipulators designed for time-optimal point-to-point motion are adversely affected by residual vibrations introduced due to the joint flexibility inherent in the system. Over the years, multiple techniques have been employed to improve the efficiency of such systems. While some techniques focus on increasing the system damping to efficiently dissipate the residual energy at the end of the move, several techniques achieve rapid repositioning by developing cleverly shaped input profiles that aim to reduce energy around the natural frequency to avoid exciting the resonant modes altogether. In this work, a numerical framework for constructing shaped inputs using a Versine basis function with peak acceleration constraint has been developed and improvements for the existing numerical framework for the Ramped Sinusoid basis function have been made to extend the range of values of the weighting function and improve the computational time. Performance metrics to evaluate the effectiveness of the numerical framework in minimizing residual vibrations have been developed. The effects of peak input acceleration and weighting function on the residual vibration in the system have been studied. The effectiveness of the method has been tested under multiple conditions in simulations and the results were validated by performing experiments on a two-link flexible joint robotic arm. The simulation and experimental results conclusively show that the inputs developed using the constrained numerical approach result in better residual vibration performance as compared to that of an unshaped input.

CHAPTER 1. INTRODUCTION

1.1 Motivation

The need for economical and reliable means to increase productivity and manufacturing in the current industrial sector is on the rise because of recent technological advancements. Rapid repositioning and accurate trajectory tracking are two important features as robotic manipulators and computer numerical control tools have steadily taken over industrial assembly lines. Advancement in technology has also helped this transition by making systems lighter and more efficient, allowing them to be customizable to every need. However, as in any physical system, certain mechanical constraints such as joint flexibility can hinder them from performing at their optimal level.

Joint flexibility is an inherent characteristic of physical systems with flexibility being induced from gears, belt drives, actuators, and other transmission elements. Joint flexibility can induce vibrations in systems during high-speed motion that can persist even after the desired position is reached. Some systems have flexibility deliberately built into their design to enhance the degree of safety of the equipment when there is human-machine interaction. While vibrations can be used as an advantage in a few systems that assemble parts with very close tolerances, flexibility and residual vibrations are detrimental to the performance of most systems serving fast point-to-point motions. Reducing residual vibrations becomes especially crucial in systems such as surgical robots and flexible space structures to avoid risks to human lives and monetary setbacks as they operate with a very small margin for error.

High-speed point-to-point motion and accurate tracking of the desired trajectory are conflicting requirements for any underdamped systems. Figure 1.1 shows the behavior of the system when a time-optimal bang-bang input is provided to a robot. When aiming for the shortest move time, it

can be seen from the spectral plot that an unshaped input carries high energy around the resonant modes of the system. This additional energy expended in trying to reduce move time usually excites the resonant modes of the systems, causing them to vibrate and deviate from the desired trajectory. This can highly impact productivity as the induced oscillations can aggravate the residual vibrations in the system, leading to an increase in the settling time of the system at the end of the move. Accounting for the settling time between successive motions of the robotic manipulator helps us to improve the tracking but negates the original purpose of achieving shorter move time.

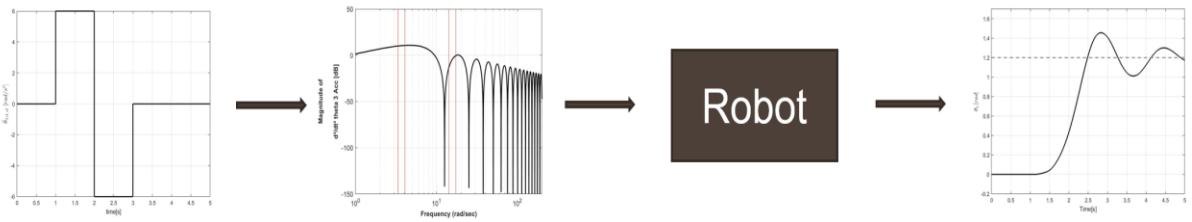

Figure 1.1. System Behaviour for a Bang-Bang input

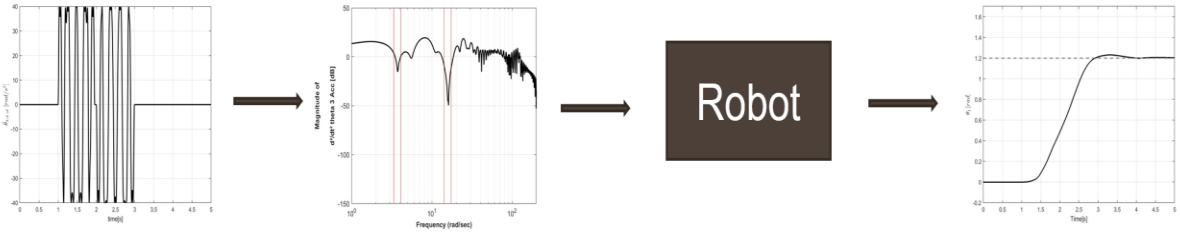

Figure 1.2. System Behaviour for a Shaped input

The general objective of this study is accurately represented in Figure 1.2. This work attempts to improve residual vibration performance of underdamped systems with joint flexibility by using

a technique known as input command shaping. This technique cleverly designs input profiles that carry less energy around the natural modes of the system to reduce the effects of resonance. The expectation is that the reduced levels of energy results in better residual vibration performance and faster settling times. The previous attempt in using command shaping with numerical optimization and peak acceleration constraint was restricted in terms of the range of parameters that it worked with and was also restricted to using only the Ramped Sinusoid as the basis function. The motivation of this work is to extend the already existing command shaping technique with Ramped sinusoid basis function over a larger range of parameter values and to introduce Segmented Versine with Numerical optimization as a second basis function to the technique as it provides control over both peak acceleration and peak velocity of the actuator.

1.2 Literature Review

Residual vibration control is a significant problem when it comes to flexible systems like robots and space structures. A lot of time, money and energy have been spent to develop remedial measures to address the problem to attain optimal performance. Many of the previous works ignore joint flexibility and only consider the rigid-body dynamics of the robots when developing control strategies. However, such simplified models are not very effective in controlling the actual robots as their joint flexibility significantly influences the system performance. Kyura and Hara [1] did some initial work to prove that the desired control performance can be better achieved by considering joint flexibility. Sweet and Good [2] derived analytical models for a drive system by considering joint flexibility and used the results to highlight that better performance can be achieved by considering joint flexibility.

Common methods for precise motion control for systems with joint flexibility can be broadly classified as techniques that modify the system dynamics and techniques that modify the inputs.

The first approach uses feedback control to force the system to respond appropriately and follow the input and by doing so it modifies the system dynamics to control the system through the vibrations. The feedback control strategies employed vary largely and range from simple PD controllers to complex robust and adaptive control methods. A good overview of the control techniques employed for flexible joint robots can be found in [3], [4], and [5]. Improvement in performance to a small degree can be observed by using standard feedback methods but the overall damping in the system could be insufficient to see any significant improvement in residual vibration control. The second approach involves techniques that modify the desired input that the system is required to follow through feedforward control methods.

The control strategies developed over the years to deal with flexible systems can be identified under two categories. The first approach works by developing models of the flexible and rigid parts of the system and using the link position in a control loop to control the final position of the end effector. But flexible elements limit the overall bandwidth and hence it becomes difficult to control the system. In the second category, the control effort acts completely on the actuators of the system and the flexible modes do not influence the control effort. For the second category, clever input trajectories that reduce residual vibrations are designed using advanced command generation techniques and they are used with a simple feedforward control that generates appropriate input commands to the actuators based on the designed trajectories.

Feedforward techniques are further categorized as inverse compensation and forward compensation methods. In inverse compensation, the inverse dynamic model of the system is used to generate the input force to the system using a pre-designed reference input [6]. Inverse compensation is susceptible to instability of the inverse models of the plants under certain conditions. To overcome this problem, advanced inverse adaptive control, where the error in the

output is utilized to modify the weights of a compensator, can be used. Lin and Hsiao [7] discuss a similar technique for system stabilization using neural network methods. The problem of instability in the plant model can also be handled by using pseudo-inverse methods as discussed by Ghosh and Paden in [8]. Better vibration performance can be achieved using these techniques but at the cost of longer move times.

The input command shaping technique that will be discussed in this work falls under the category of forward compensation methods of feedforward control. In forward compensation, the reference trajectories that produce the force to achieve residual vibration reduction are derived by constructing the force profile that needs to act on the system. Input command shaping techniques develop cleverly designed inputs that aim to avoid excitation of flexible modes. As the sharpness of the transition in the reference input increases, corresponding energy at higher frequencies that could potentially cause resonance in the flexible joints increases. The simplest of the command shaping approaches uses this as a foundation and attempts to achieve smooth transitions in the input signals. The smoothness in transitions of the input signals is dictated by the maximum frequency used to construct the inputs and it is important to always have the maximum frequency below the natural frequency of the system to avoid excitation of the natural modes [9]. As the constraint on the maximum frequency gets tighter, we achieve better residual vibration performance, but it comes at the cost of significantly longer move times.

One of the earlier open-loop approaches in command shaping is called posicast control. It worked using a simple concept of applying a series of appropriately timed step inputs at the end of the motion to cancel out the residual vibrations generated. Smith [10] in the 1950s proposed a posicast control method based on wave cancellation principle for producing a dead-beat response in an underdamped system. Unfortunately, the efficiency of the method was highly dependent on

accurate system models and precise calculations of the system damping and natural frequency of the system. Singer [11] extended this approach by making it less susceptible to modeling errors by increasing the robustness to system parameters. Singer and Seering in [12] introduced a command shaping technique that removes energy at the system resonant modes by passing a general input signal through a Finite Impulse Response (FIR) filter. Bhat and Miu [13] proved that FIR filtering is the Laplace domain equivalent of placing zeros at undesirable system poles. This technique is very useful as specifically designed FIR filters for the flexible modes of the system can be used to achieve improved vibration performance even when we don't have any control over the type of input. After successful application of FIR filter in minimizing vibrations, multiple works investigating the effectiveness of different filters and filters of different order were carried out. In [26] Mohamed and Tokhi explored flexible robot performance by applying low-pass filters and band-stop filters.

Command shaping techniques with better control of the developed input signal were also developed. Meckl in [14] and [15] proposed one such technique that constructs input commands by using harmonics of appropriately selected basis function. Inputs constructed using this technique can reduce energy at specified frequencies. To minimize residual vibrations, the specified frequencies in the command shaping technique would have to be the resonant frequencies of the system. Azad, et al. [16] gives a detailed comparison of command shaping methods and discusses the experimental results obtained. In [17], the robustness of the command shaping technique to modeling errors has been discussed and it was confirmed that this technique is effective for systems having multiple modes. Chatlatanagulchai, Beazel and Meckl [18] proved that the command shaping method can be effectively used with time-varying systems and, in [19], Pau and Lau confirmed that the technique can be used on systems with configuration-dependent

natural frequencies. Agrawal [20] introduced a new cost function and extended the approach to numerical optimization to resolve the drawbacks experienced because of the Gibbs phenomenon using the Least Absolute Deviation (LAD) method. Wu [21] modified the technique under the analytical application to impose constraints on the peak input acceleration instead of actuation time as all real systems work with actuators that saturate after a certain value and needs to be considered while developing the input. Wu [22] extended the existing numerical optimization approach to enforce constraint on peak acceleration for the Ramped Sinusoid basis function.

1.3 Overview of thesis.

In this research, the command shaping approach introduced in [15], [20], and [21] has been adopted to investigate its influence on a two-link flexible joint robotic arm. Most of the previous studies in command shaping used least square approximation in the cost function to approximate the bang-bang profile because of which the command shaping process was affected by the Gibbs phenomenon, which results in shaped profiles demanding high intermittent peak torques. Agrawal in [20] proposed a numerical optimization framework and a new cost function using Least Absolute Deviation approach to minimize the error in approximating a bang-bang profile. The new cost function with least absolute deviation formulation was successful in reducing the effects of Gibbs phenomenon in the command shaping process. Wu [21] modified the existing actuation time constraint in the optimization process with the original least square approximation technique by imposing constraints on the peak acceleration and studied the effects on actuation time. Wu [22] successfully combined the two approaches with peak acceleration constraint and LAD cost function for the Ramped Sinusoid basis function to avail the benefits of both approaches.

In this work, the numerical problem formulation in [22] with LAD cost function and peak acceleration constraints has been further optimized to decrease the computation time and extend

the range of design parameter values that can be used with the technique. This work also extends the numerical optimization process to use Segmented Versine as a basis function for generating the shaped commands as this function allows for a peak velocity constraint along with a constraint on peak acceleration. A trade-off between time optimality and vibration performance has also been investigated. Simulation results have been drawn and experiments were conducted to further verify the accuracy of the results observed from the simulation. A comparison of the simulation and experimental results of the current command shaping technique with the previous approaches and an unshaped bang-bang input has been presented in this work, and advantages and limitations of the current approach are discussed in detail.

Chapter 2 gives a brief overview of the different basis functions used in the command shaping process and an outline about the numerical optimization process and the governing equations is presented. Chapter 3 in this paper discusses the physical setup and mathematical model of the two-link flexible joint robot used in validation of the results. It is important to clearly define a performance metric to quantify vibration performance. Chapter 4 defines the performance metrics used in this work to evaluate the command shaping technique and presents simulation and experimental results by application of the numerical optimization techniques. Chapter 5 draws inferences from the comparisons made in chapter 4, summarizes the contributions of this research and provides suggestions for potential future work.

CHAPTER 2. COMMAND SHAPING WITH PEAK ACCELERATION CONSTRAINT

2.1 Background

Command shaping works on the logic that energy in the resonant frequencies of the system excites resonance and hence amplifies residual vibrations. The command shaping method cleverly designs input profiles that carry significantly lower energy around the natural frequencies of the system to alleviate the problem of residual vibration. Aspinwall [23] was one of the early researchers to propose that residual vibrations can be suppressed by modifying the frequency content of the input profile. The input shaping method proposed in [23] achieved improved vibration performance at the cost of longer move times when compared to a time-optimal bang-bang profile as a big portion of spectrum was being manipulated to carry less energy.

Meckl in [14] developed a new technique that constructs input profiles by approximating it to a bang-bang profile to have shorter move times while also avoiding energy around the resonant modes of the system. In this technique, coefficients for a pre-determined number of harmonics of selected basis function were calculated and when the harmonics are added together, the resulting input would have a trough at the natural frequencies of the system. The magnitude of spectral attenuation at the natural frequencies and the smoothness of the generated input profile depended on the total number of harmonics of the basis function being used. Meckl in [15] set up a weighted multi-objective fitness function in which the first objective function minimized the error between the generated input profile and time-optimal bang-bang profile, effectively trying to create the closest approximation of a square wave for the best move time. The second function of the multi-objective function attempts to minimize the magnitude of the Fourier transform at critical frequencies. The relative weights assigned at the start of the process to the objective functions

dictate the emphasis that each objective of the multi-objective function gets while developing the input profiles. In [15], Meckl used the bang-bang profile, Ramped sinusoid and Segmented Versine as the three forcing functions. To ensure good tracking of the input profiles generated using this technique on physical systems, a closed-loop system with a feedback controller was implemented. The plant and the feedback controller are together treated as one entity and the generated input signals are used as a reference for the feedback controller. The multi-objective function could be set to penalize a range of frequencies around the natural frequency of the system instead of just one frequency. This improved the robustness of the command shaping method against errors in parameter and natural frequency estimation. Beazel [17] further extended the command shaping method to make it applicable to multimode nonlinear systems with configuration-dependent resonance. Scheel [24] made improvements to the command shaping method for ease of implementation on the physical system. Agrawal in [20] developed a numerical optimization framework to overcome the drawbacks of the Gibbs phenomenon in the traditional approach. Wu [22] modified the numerical optimization framework to have constraints on peak acceleration based on work in [21] for Ramped sinusoid basis function. In this work, the numerical optimization framework has been extended to work with the segmented versine as a basis function. The new basis function in the numerical framework adds a second dimension of specifying the peak velocity constraint along with peak acceleration, which both represent limitations of the physical system.

2.2 Theory of Command Shaping

The two-link robotic arm used for experiments in this work can be simplified and approximated as a two-mass system as shown in Figure 2.1. The two-mass single mode system successfully captures all the important dynamic properties of the two-link robot. In the simplified two-mass model in Figure 2.1, M_1 represents the motor inertia, M_2 represents the endpoint inertia, k

represents the transmission and structural stiffness. Thus, the two-mass model can be used as a good starting point to analyze and gain understanding of the residual vibrations in the two-link robot arm.

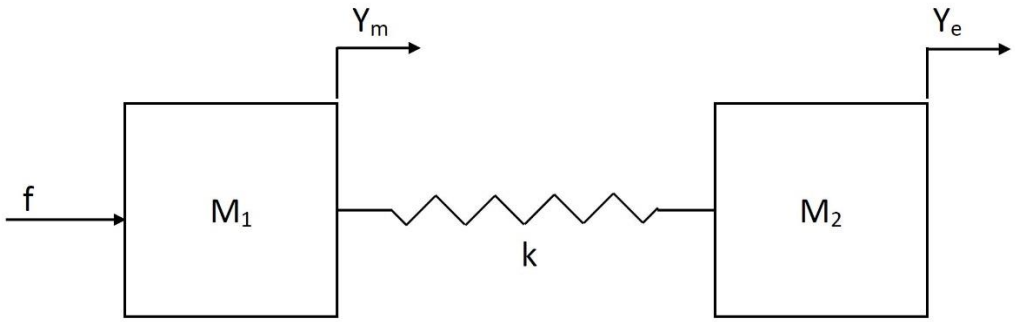

Figure 2.1. The Two-Mass System. [from 21]

Meckl [14] using a similar two-mass system formulated a relationship between the residual vibration and the spectral magnitude of the forcing function at the natural frequencies of the system

$$A^* = \omega_n T_f \left|F^*(\omega_n T_f)\right|, \tag{2.1}$$

where A^* is the dimensionless residual acceleration amplitude; ω_n is the natural frequency of the two-mass system; T_f is the actuation time; $\left|F^*(\omega_n T_f)\right|$ is the dimensionless Fourier transform of the forcing function, which is defined as

$$\left|F^*(\omega_n T_f)\right| = \frac{|F^*(\omega_n)|}{F_{max} T_f}, \tag{2.2}$$

with F_{max} as the maximum value of the forcing function.

The normalized shaped function is represented in a general form as,

$$f^*(t) = \sum_{l=1}^{L} B_l^* \Phi_l^*(t), \tag{2.3}$$

here l represents the l^{th} harmonics; L is total number of harmonics; B_l^* is the normalized coefficient of each harmonic; $\Phi_l^*(t)$ is the forcing function and the values of $f^*(t)$ range between

$$-1 \leq f^*(t) \leq 1. \tag{2.4}$$

Throughout this work, the acceleration profile after command shaping serves as a reference input to the controller. The acceleration profile is generated by multiplying the normalized shaped profile with the maximum desired acceleration ($\ddot{\theta}_{d.max}$). This input trajectory to the two-link robot is given by

$$\ddot{\theta}_{RS/V} = \ddot{\theta}_{d,max} f^*(t) \tag{2.5}$$

In $\ddot{\theta}_{RS/V}$, RS stands for ramped sinusoid and V stands for Segmented versine. Detailed information about the basis functions will be provided in section 2.4. Defining the input profile as expressed in equation (2.5) effectively constrains the peak acceleration for the input to the maximum desired acceleration.

The multi-objective fitness function defined by Meckl in [15] minimizes the error between the generated input command and a time optimal bang-bang profile while penalizing the energy at the system natural frequencies. General form of the multi-objective function is defined as

$$J = \frac{1}{T_f}\int_0^{T_f} \left[f^*_{bangbang}(t) - f^*(t)\right]^2 dt + \rho \sum_{i=1}^{22} \left(\omega_i T_f\right)^2 \left|F^*\left(\omega_i T_f\right)\right|^2 \tag{2.6}$$

where $f^*(t)$ is the shaped profile; ρ is the dimensionless relative weighting factor between the two objectives. The summation term in Equation (2.6) adds up the magnitude of the Fourier transform at the 22 selected frequencies around the system natural frequencies. The intention behind setting up the objective function in the above format was to have a big window around the natural frequencies of the system to build robustness towards modelling error. The frequencies are selected to have a $\pm$10% tolerance band around each natural frequency.

Agrawal in [20] modified the first objective in Equation (2.6) from the least square formulation to the Least Absolute Deviation (LAD) formulation and Wu in [21] modified the equation again by using the normalized shaped function to constrain the maximum input acceleration. The resulting modified multi-objective fitness function is

$$J = \frac{1}{T_f}\left[\int_0^{T_f/2}[1 - f^*(t)]\,dt + \int_{T_f/2}^{T_f}[-1 - f^*(t)\,]\,dt\right] + \rho\sum_{i=1}^{22}\left(\omega_i T_f\right)^2\left|F^*\left(\omega_i T_f\right)\right|^2 \tag{2.7}$$

To solve the objective function to minimize residual vibrations, Equation (2.6) is partially differentiated with respect to the coefficient B_l and then the result is set to 0 to find the minimum of the function. It is represented as

$$\frac{\partial J_{gen}}{\partial B_l} = 0 \tag{2.8}$$

The analytical expressions used to solve for the coefficients for both basis functions can be found in [15]. A scale factor (SF) needs to be calculated to normalize the shaped profile. The scale factor is calculated as

$$SF = \max|f(t)| \tag{2.9}$$

The coefficients (B_l) are normalized using the scale factor using the equation

$$B_l^* = \frac{1}{SF}\,B_l \tag{2.10}$$

2.3 Motivation for Numerical Optimization (Gibbs phenomenon)

A Bang-Bang profile is an input profile where the actuator switches instantaneously between peak acceleration and deceleration. Bellman et al. in [25] and LaSalle [26] proved mathematically

that a bang-bang profile is the time-optimal input for systems with controllable modes. The first term of the general multi-objective function in Equation (2.6) uses a least square approximation of a bang-bang profile from the selected basis function, but using a finite number of harmonics of a basis function to approximate an ideal bang-bang profile which jumps abruptly between peak acceleration and peak deceleration is challenging as ringing artifacts commonly known as Gibbs phenomenon start to appear in the approximation.

Gibbs phenomenon discovered by Henry Wilbraham and rediscovered by J. Willard Gibbs, is the unusual oscillatory behavior displayed at a jump discontinuity by the Fourier series of a piecewise continuously differentiable function when approximating it using a finite number of harmonics. J. W. Gibbs in 1899 pointed out that ringing artifacts were a mathematical problem, the overshoots and undershoots would always occur during reconstruction of a discontinuous function using n^{th} partial sum of Fourier series. Figure 2.2 shows the partial sum approximation of a square wave as the number of harmonics is increased.

Let f be a piecewise continuously differentiable function with a period of $L > 0$. If we consider x_0 as the point of discontinuity with a non-zero gap of a, the discontinuous gap can be mathematically represented as

$$f(x_0^+) - f(x_0^-) = a \neq 0 \tag{2.11}$$

The sum of partial Fourier series for each N with $N \geq 1$ is given by

$$\begin{aligned} S_N f(x) &= \sum_{-N \leq n \leq N} \hat{f}\,(n) e^{2\pi i n x/L} \\ &= \frac{1}{2} a_0 + \sum_{n=1}^{N} a_n \cos\left(\frac{2\pi n x}{L}\right) + b_n \sin\left(\frac{2\pi n x}{L}\right) \end{aligned} \tag{2.12}$$

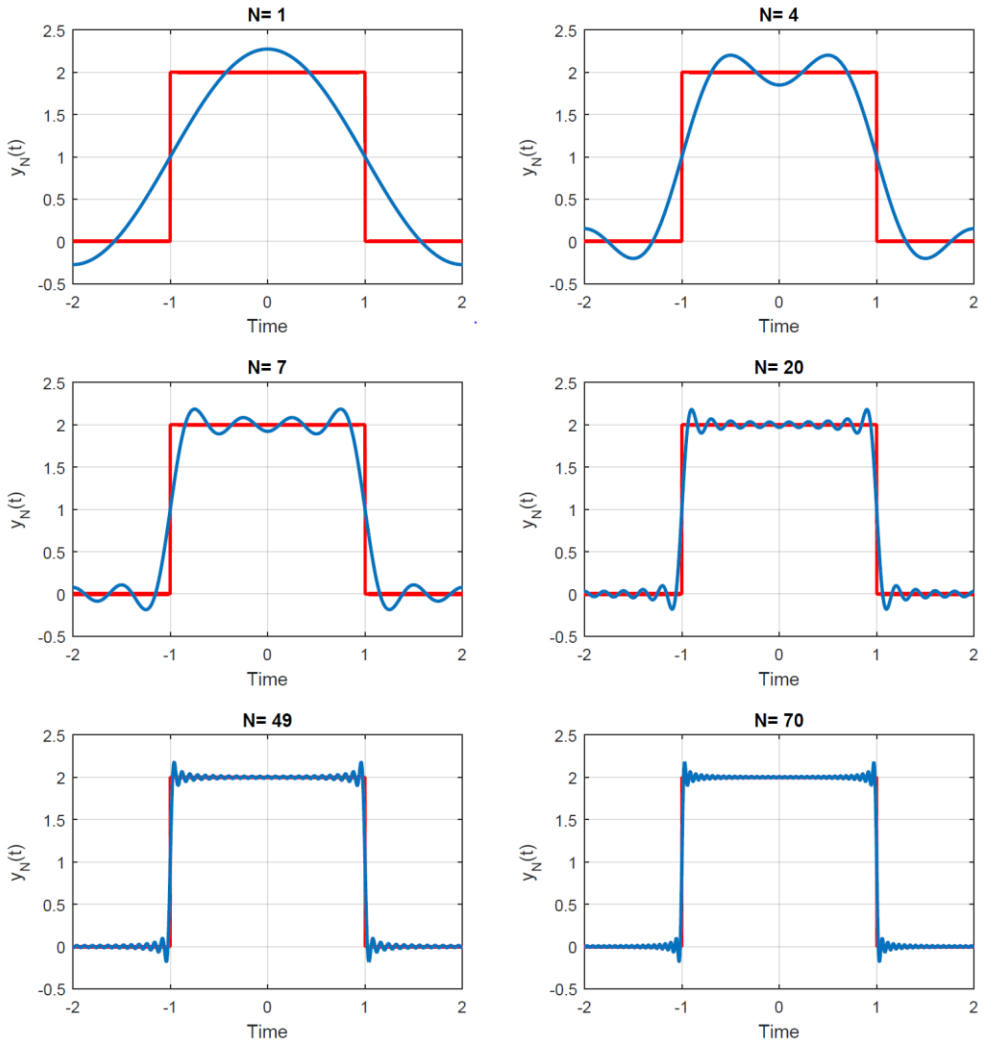

Figure 2.2. Gibbs Phenomenon [from 20]

The Fourier coefficients $\hat{f}(n)$, a_n , b_n in Equation (2.12) are defined in their general form as

$$\begin{cases} \hat{f}\,(n) := \dfrac{1}{L}\displaystyle\int_0^L f(x)\, e^{-2\pi i n x/L} dx \\ a_n := \dfrac{2}{L}\displaystyle\int_0^L f(x) \cos\left(\dfrac{2\pi \mathrm{n} x}{\mathrm{L}}\right) dx \\ b_n := \dfrac{2}{L}\displaystyle\int_0^L f(x) \sin\left(\dfrac{2\pi n x}{L}\right) dx \end{cases} \tag{2.13}$$

Evaluating the summation around the point of discontinuity using basic calculus and applying limits we get

$$\lim_{N \to \infty} S_N f\left(x_0 + \frac{L}{2N}\right) = f(x_0^+) + a.(0.0894) \tag{2.14}$$

and

$$\lim_{N \to \infty} S_N f\left(x_0 - \frac{L}{2N}\right) = f(x_0^-) + a.(0.0894) \tag{2.15}$$

As $N \to \infty$, it can be seen from Equations (2.14) and (2.15) that the height of overshoot and undershoot just before and after the point of discontinuity is equal to $a.(0.0894)$. A similar result can also be noticed from Figure 2.2, as the number of harmonics being used to approximate the square wave increases, while the width of the overshoot decreases, the height of the overshoot converges to a fixed value.

According to Weierstraas, M-test coefficients with absolute convergence will show approximations that converge uniformly and would not show any oscillatory behavior. Due to slowly decaying coefficients, functions with discontinuity will have slow convergence of Fourier series. Oscillatory behavior while approximating a square wave with a sharp discontinuity using a least square approximation indicates a non-uniform rate of decay of Fourier coefficients at higher frequencies. Oscillations at the break point can be attributed to energy carried by the higher frequency harmonics. Removing energy around the natural frequencies of the system while also pushing for time optimality using command shaping could cause more energy to be introduced to higher energy peaks which results in the overshoots and undershoots of the shaped profile. When this shaped profile is provided as an input to the system, to track the input accurately and perform the desired motion, the overshoots and the undershoots at the discontinuity would require higher intermediate torques.

Agrawal in [20] modified the multi-objective function by using the Least Absolute Deviation (LAD) approximation instead of least square approximation to reduce the error between the shaped input and the bang-bang profile. LAD, or commonly known as L_1 norm, is an optimality condition that reduces the total of absolute errors; this means each error component that contributes to the total error is weighted equally in LAD as opposed to least square approximation in which the errors are squared. The new formulation with L_1 norm cannot be solved using closed-form expressions, thus a numerical optimization framework had to be built to solve it. The updated expression used in the numerical optimization process is

$$J_{L_1} = \frac{1}{T_f}\sum_{i=1}^{N} |f^*_{bangbang}(t_i) - f^*(t_1)| + \rho \sum_{i=1}^{kM} \left(\omega_i T_f\right)^2 \left|F^*\left(\omega_i T_f\right)\right|^2, \qquad (2.16)$$

where $N = T_f/dt$, where T_f is the complete profile actuation time; dt is the discrete step size used for evaluation of numerical error points; M is the number of natural frequencies of the system and k is the number of equally distributed points around each natural frequency where the spectral energy needs to be reduced.

In Figures 2.3 and 2.4, we can see the square wave approximation plotted with penalty term set to 0 using the analytical approach and the numerical approach, respectively. It can be noticed that using the numerical approach, overshoots and undershoots at the discontinuity are significantly reduced and we get a closer approximation to the actual bang-bang profile as compared to the analytical approach. It can also be seen from the spectral plot that a uniform rate of decay at higher frequency is achieved using the numerical approach.

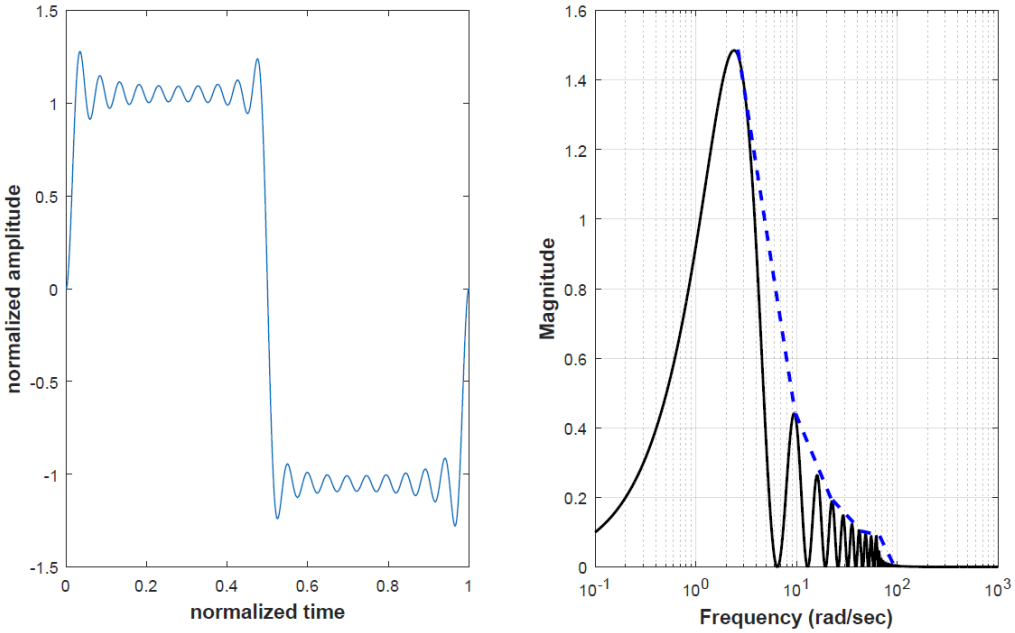

Figure 2.3. Bang-Bang approximation using Analytical Approach [from 20]

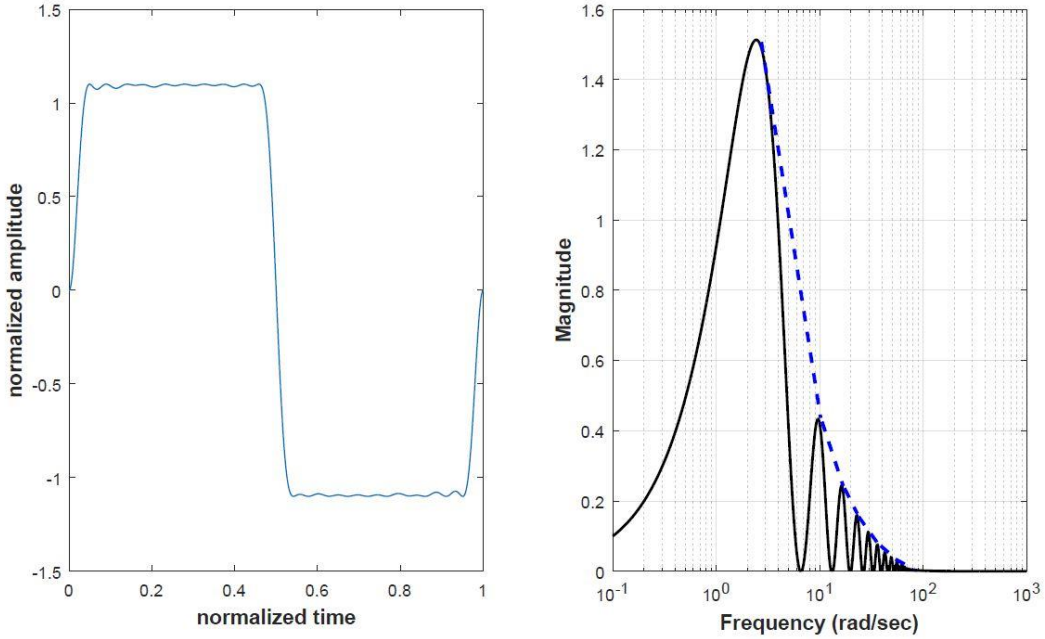

Figure 2.4. Bang-Bang approximation using Numerical approach [from 20]

The Fourier transform equation used to calculate the attenuation of the frequency content around each of the natural frequencies is given by

$$F^*(\omega) = \int_0^{T_f} F^*(t) e^{-i\omega t} dt \tag{2.17}$$

where $F^*(t)$ is the normalized forcing function given by Equation (2.3). This can be further simplified by substituting the equations of the forcing function for both the Ramped Sinusoid and Segmented Versine basis function to give the magnitude of spectral attenuation as

$$|F^*(\omega)|_{RS} = \left| \frac{2\sin(\omega T_f/2) - \omega T_f \cos(\omega T_f/2)}{(\omega T_f/2)^2} \sum_{l=1}^{L} \frac{B_l^* \alpha_l}{\alpha_l^2 - (\omega T_f)^2} \right| \tag{2.18}$$

and

$$|F^*(\omega)|_V = \left| \frac{2(2\pi)^2 \sin(\omega_i T_p/2)}{\omega_i T_p} \sum_{l=1}^{L} \frac{B_l^* l^2}{(2\pi l)^2 - (\omega_i T_p)^2} \right| \tag{2.19}$$

where T_f represents the profile actuation time; α_l is the characteristic number for each Ramped Sinusoid harmonic. For Versine, $T_f = T_p\, n_{seg}$ where T_p is the actuation time for each individually shaped segment and n_{seg} is the number of segments in the profile.

2.4 Basis Functions

2.4.1 Ramped Sinusoid

Meckl in [14] introduced the ramped sinusoid function as one of the two basis functions for command shaping method. Ramped sinusoid basis function has zero slope and magnitude at the beginning and the end of motion because of which it has the attribute to avoid abrupt discontinuity. It is given by

$$\Phi_l^*(t) = \frac{1}{\alpha_l}\left(\frac{1}{2} - \tau\right) + \frac{1}{\alpha_l^2}\sin(\alpha_l\tau) - \frac{1}{2\alpha_l}\cos(\alpha_l\tau) \tag{2.20}$$

where α_l is the characteristic number associated with each harmonic. The characteristic number for each harmonic can be calculated using

$$\alpha_l \sin(\alpha_l) + 2\cos(\alpha_l) - 2 = 0 \tag{2.21}$$

where $\alpha_l \neq n\pi$ and n is an even integer. The characteristic numbers for the first ten harmonics of the ramped sinusoid basis function are listed in Table 2.1 and the first four harmonics are plotted in Figure 2.5. The dimensionless time (τ) in Equation (2.20) is given by

$$\tau = \frac{t}{T_f} \tag{2.22}$$

T_f in the above equation stands for the profile actuation time as per the standard nomenclature in this work.

The actuation time associated with bang-bang profile to cover a distance of y_f with a maximum acceleration of a_{max} is given by

$$T_s = 2\sqrt{\frac{y_f}{a_{max}}} \tag{2.23}$$

The relationship between shaped profile actuation time (T_f) and bang-bang actuation time (T_s) for the ramped sinusoid profile is given by

$$T_f = \Gamma_{RS} T_s \tag{2.24}$$

where Γ_{RS}, which stands for the actuation time penalty, is given by

$$\Gamma_{RS} = \sqrt{\frac{3\ SF}{\sum_{l=1}^{L} B_l}} = \sqrt{\frac{3}{\sum_{l=1}^{L} B_l^*}} \tag{2.25}$$

in which the normalization factor (SF) used to normalize the shaped input in the range [-1,1] is given by Equation (2.9).

Table 2.1. Characteristic numbers for first ten harmonics of Ramped Sinusoid Basis function

$l^{th}\ Harmonic$	Value
α_1	8.9688
α_2	15.4505
α_3	21.8082
α_4	28.1324
α_5	34.4415
α_6	40.7426
α_7	47.0389
α_8	53.3321
α_9	59.6232
α_{10}	65.9128

With the constraint on the energy in a band of frequencies, for a given peak acceleration, the shaped input with the basis function does not carry as high an energy as the bang-bang profile and hence will always need a longer actuation time when compared to bang-bang profile. Therefore, Γ_{RS} will always have a value greater than 1. This relation for Γ_{RS} can be used as a nonlinear constraint for the numerical optimization setup. But this constraint on the energy at the natural frequency results in minimization of residual vibration in the system, resulting in a shorter settling

time and total move time for the system. Simulation results for the Ramped Sinusoid basis function will be further discussed in Appendix B.

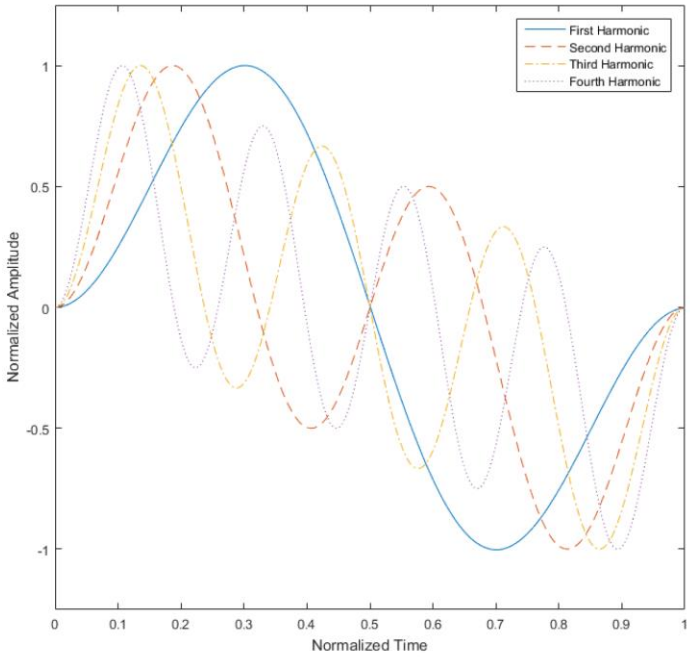

Figure 2.5. First four harmonics of Ramped Sinusoid basis function

2.4.2 Segmented Versine

Versine is the second basis function used in the command shaping process introduced by Meckl in [15]. Versine basis function splits a single profile into multiple segments by approximating a single square pulse at a time as opposed to ramped sinusoid which builds a full cycle consisting of both the acceleration and deceleration phases together. Beazel in [17] proposed dividing the command shaping method into multiple segments. Systems with configuration-dependent resonance over a large range of motion could have a large band of natural frequencies. Command shaping method, if adjusted to minimize energy at such large bands of frequencies, could result in significantly longer actuation time. Thus, the ability of the Versine to separate the inputs into multiple segments allows more flexibility while designing the input and it can be tailored to adjust

the natural frequency based on position for such systems. Combining all the shaped segments will result in an input that changes the frequency content with time to minimize energy only at the natural frequencies relevant to that segment.

When using it with actuators that have rate limits, versine basis function can be used to specify a peak velocity and a peak acceleration value while designing the shaped input, Under this condition one segment of the shaped profile can be used to drive the system to the peak velocity state, maintain the state of constant velocity and then use the second segment to decelerate to a zero velocity state resulting in a trapezoidal velocity profile. The constant velocity section in the command shaping process is designed to adjust itself so that the system accurately travels the desired distance to reach the final position. This constraint on peak velocity cannot be applied on the ramped sinusoid function as the command shaping generates both the acceleration and deceleration together.

The segmented versine basis function is given by

$$\Phi_l^*(t) = 1 - \cos(2\pi l\tau) \qquad l = 1,2,3 \dots, L, \tag{2.26}$$

where l is the harmonic of the basis function and τ is the dimensionless time given by

$$\tau = \frac{t}{T_p}, \tag{2.27}$$

where T_p is the actuation time for a single segment of the shaped input for the versine basis function. Shaped input using the versine consists of the acceleration segment, the deceleration segment and constant velocity phase depending on the peak velocity of the actuator. Thus, the total profile actuation time will be the sum of actuation times for each of the three mentioned segments. The first four harmonics of the versine function are plotted in Figure 2.6.

Since input using versine functions will at least have two segments of acceleration and deceleration, the bang-bang actuation time for each segment is calculated using

$$T_s = \sqrt{\frac{y_f}{a_{max}}} \tag{2.28}$$

where y_f and a_{max} go by the general definition used in this report. The bang-bang actuation time can also be calculated by using the peak velocity of the actuator (v_{max}) and peak acceleration of the actuator a_{max} using the relation

$$T_s = \frac{v_{max}}{a_{max}} \tag{2.29}$$

The profile actuation time (T_p) for versine can be calculated using

$$T_p = \Gamma_V T_s \tag{2.30}$$

where the time penalty factor Γ_V is defined as

$$\Gamma_V = \frac{SF}{\sum_{l=1}^{L} B_l} = \frac{1}{\sum_{l=1}^{L} B_l^*} \tag{2.31}$$

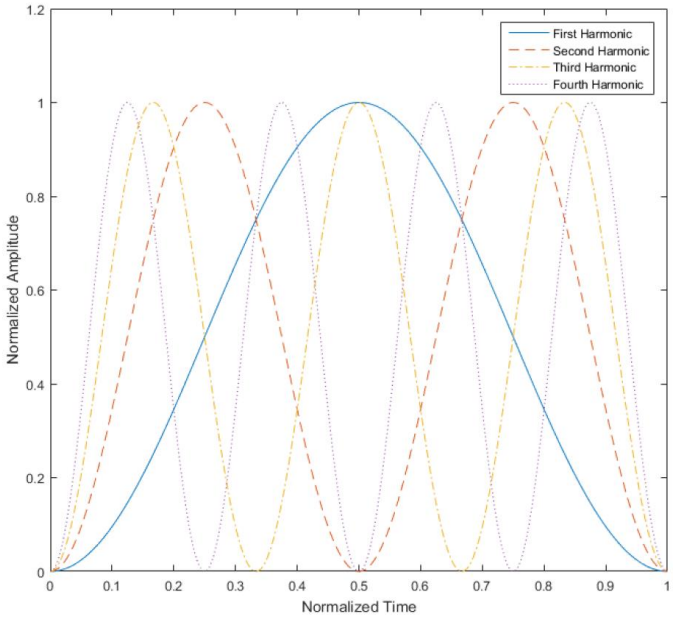

Figure 2.6. First four harmonics of Versine basis function

2.5 General outline of Numerical Optimization setup

The multi-objective fitness function defined by Agrawal in [20] is the cost function used in this work. Numerical solver `fmincon` is used to find the minimum of a function with nonlinear constraints along with Nelder mead algorithm in MATLAB program to solve for the coefficients of the basis function. The optimization is subjected to nonlinear constraints, which force the time-penalty (Γ) value to be always greater than 1 and the normalized shaped input function $f^*(t)$ values at all points to be within the range [-1,1] in the converged solution. The design variables for the numerical optimization problem that the solver can change is the coefficients of the harmonics for the basis function (B_l).

The general flow of the proposed optimization solution can be described in Figure 2.7 and the generalized problem statement for command shaping can be written as

$$\min J_{L_1}\left(B_l, T_f, \omega_n, \rho\right) \text{ such that } \begin{cases} -1 \leq \sum_{l=1}^{L} B_l^* \Phi_l^* \leq 1 \end{cases} \tag{2.32}$$

Closed-form solution cannot be obtained for the multi-objective function due to the inherent coupling between B_l and Γ, hence an iterative numerical solver is used to calculate the coefficients B_l and the corresponding time penalty (Γ). The steps involved in solving the multi-objective functions are:

1. Initialize and calculate terms that do not change with iterations such as $\omega_n, T_f, \rho, a_{max}, v_{max}$ using respective equations.
2. Make an initial guess for the 20 design variables (B_l) and find the corresponding Γ for the basis function being used.
3. Run the solver to find the minimum of the multi-objective function to calculate new set of design variables (B_l).

4. Calculate SF using Equation (2.9) and use it to normalize the coefficients and the shaped input.
5. Calculate Γ by using the appropriate equations for the basis function from the normalized coefficients.
6. Update the values of the design variables (B_l) and shaped profile actuation time T_p using the value of Γ calculated in the previous step.
7. Repeat steps 3 to 6 until two consecutive iterations of the solver converge to a value of Γ within a tolerance of 1e-3.

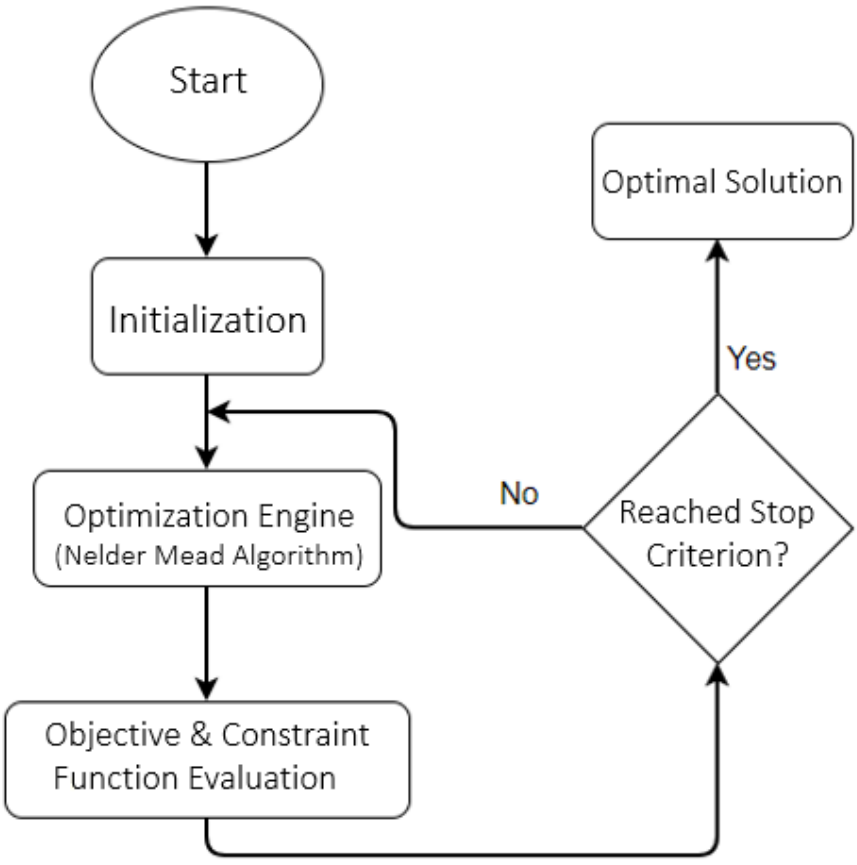

Figure 2.7. Sequence of operations for Numerical Optimization setup. [from 20]

Each iteration of the optimization process is computationally expensive and very time consuming. To accelerate the optimization process, Aitken algorithm, which takes in values of three consecutive iterations of Γ to directly give a good estimate of the Γ from the fourth iteration, is being used. The equation used by the Aitken algorithm is given by

$$\Gamma_{i+3} = \frac{\Gamma_{\mathrm{i}}\Gamma_{\mathrm{i+2}} - \Gamma_{i+2}^{2}}{\Gamma_{i+2} - 2\Gamma_{i+1} + \Gamma_{\mathrm{i}}}\ , \tag{2.33}$$

CHAPTER 3. MATHEMATICAL MODEL AND SYSTEM DESCRIPTION OF THE ROBOT

Once the shaped commands are generated using the equations and numerical framework described in chapter 2, it becomes essential that we implement the generated input for a system with joint flexibility in simulation and experiment to gain better understanding of the effectiveness of the method. In this work a two-link flexible joint robot has been used as a test bed for validation of the command shaping approach. A description of the robot setup along with various mathematical models is detailed in this chapter. Section 3.1 discusses the hardware components and the system setup of the two-link robot. Section 3.2 provides information about two different versions of the mathematical model of the robot system used in this work. These models are used in controller design, precise simulations and command shaping implementation. The first of the models is a complete Lagrangian model and the second version is a reduced model. Section 3.3 is used to provide a definition of all the system parameters along with their estimated values. Section 3.4 addresses the topic of controller design for the system. In Section 3.5, a discussion on configuration-dependent resonance is presented.

3.1 Two-Link Flexible Joint Robot

The two-link robotic arm custom-built to verify command shaping experiments is set up at Purdue University in the Ruth and Joel Spira Laboratory for Electromechanical Systems in the School of Mechanical Engineering. The robot arm, as the name suggests, is made up of two sets of links, accelerometers, encoders, brushless DC motors and transmission elements consisting of chains and sprockets. The physical setup of the two-link robotic arm is shown in Figure 3.1. The two links are referred to as link 1 and link 2 or shoulder link and elbow link, respectively, and are

designed to operate in a horizontal plane with 2-DOF. The robot was initially designed by Yegerlehner [27] to be a serial manipulator with rigid joints. Kinceler modified the robot to include joint compliance by adding flexible elements in [28]. Chatlatanagulchai in [29] programmed the FPGA and set up the controller using the LabVIEW data acquisition environment. An inertial frame of reference for the two links of the robot is provided by the base of the robot which houses the first motor and first link encoder. The end of the shoulder link has the second motor and the encoder for the second link mounted on it. The motor on the shoulder link drives the elbow link. Belt drives with gear ratios of 5, which connect the DC motors and the links, are used as transmission elements to drive the links. Torsional springs with spring coefficients ranging from 10^3 Nm/rad to 10^5 Nm/rad are added between the sprockets driven by the motors and the links to introduce joint compliance to the system. The robot system is designed such that the motors indirectly act on the links through the torsional springs and as these springs have coefficients smaller than a standard industrial manipulator, the system can be considered as a proving ground to investigate various control strategies and effectiveness of the command shaping method.

The complete setup for the two-link robot included LabVIEW version 8.5.1 with additional modules, a desktop PC with a third-generation Intel core i5 processor and 16GB RAM, and a National Instruments (NI) PXI-7831R field programmable gate array (FPGA). The FPGA with 16 analog I/O ports, 96 digital channels with Virtex-II 1M gate is easy to configure and has large data handling ability. The FPGA, which can operate at over 200kHz sampling rate, is currently configured to operate at a sampling rate of 2kHz. The FPGA is connected to two National Instruments SCB-68 Shielded I/O Connector Blocks to handle encoder signals and all other input signals, respectively.

An Electro Craft DPP242 DC motor with a torque constant of 0.118 Nm/A and a maximum torque of 2.47 Nm at 21.2 A is used to drive the shoulder link and an Inland T-3108-A DC motor with a torque constant of 0.61 Nm/A and a maximum torque of 1.35 Nm at 2A is used to drive the elbow link. An advanced motion control brushless pulse–width modulated (PWM) transconductance servo amplifier converting input voltage signals into current commands is used to drive each motor.

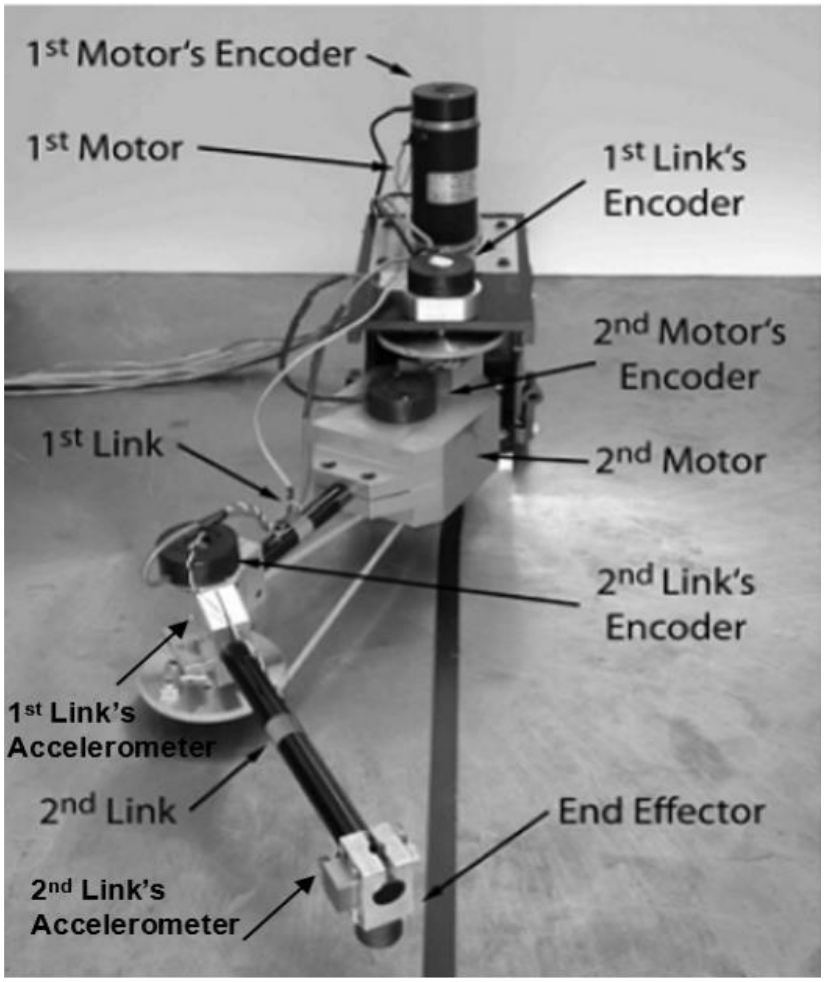

Figure 3.1. Physical Setup of the Two-link Robot [from 20].

Four incremental optical encoders from Renco Encoders are used to precisely measure the positions of the links and the motors in the robot. While model R80 encoder with a resolution of 4000 counts per revolution is used on the 1st link, a model RM21 with the same resolution is used on the second link. Finite differences of the encoder readings with a 4th-order Butterworth filter are used to calculate the velocities of the links. To measure the linear accelerations at the end of each link, two Kistler 8315A single-axis accelerometers with a range of $\pm$ 2g, sensitivity of 2V/g, frequency response between 0 to 250Hz and a resolution of 0.35 mg are mounted at the end of each link.

3.2 Mathematical Model

For the physical system shown in Figure 3.1, the schematic of the two-link robotic arm with the physical model parameters notation labelled is presented in Figure 3.2. The notations used in Figure 3.2 are considered standard in this work and are used to derive the relationship between elements. Subsection 3.2.1 provides detailed information about the complete Lagrangian model of the system developed by Nho [30]. Section 3.2.2 explains the reduced model of the system derived by Spong in [31] by making valid assumptions about viscous damping of the torsional springs and kinetic energy of the motors. Shoulder link angle θ_1 and shoulder motor angle θ_3 are defined and measured in an inertial reference frame, while angles related to the elbow link, i.e., elbow link angle θ_2 and elbow motor angle θ_4, are defined and measured relative to shoulder link position θ_1.

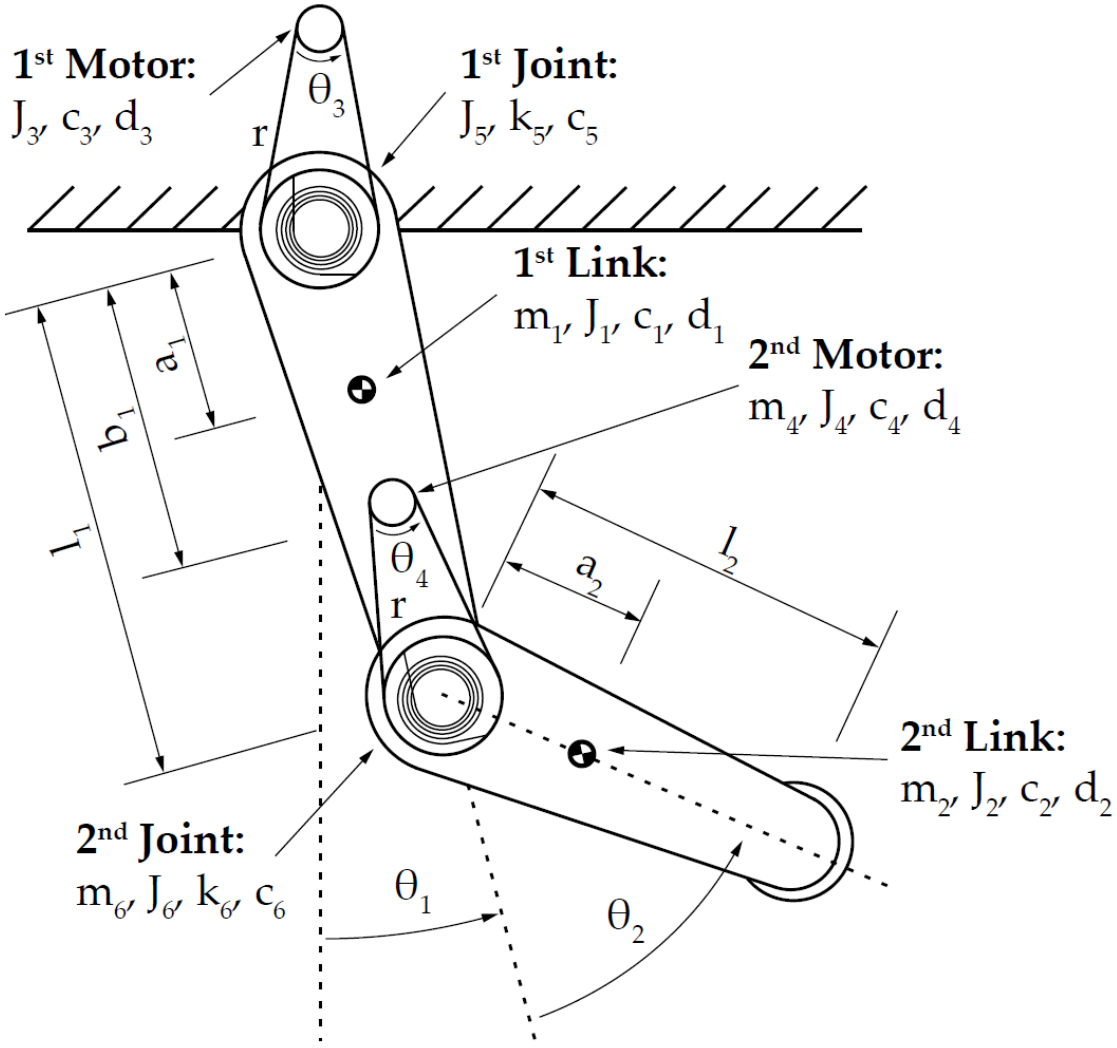

Figure 3.2. Schematic of the Two-link Robot [from 30].

3.2.1 Lagrangian model

Nho in [30] derived the Lagrangian model of the robot by considering coulomb and viscous friction in elements of the robot. The system was initially designed to operate with a payload mass in the dynamics but for the present study, the payload mass is not considered for analysis.

The Lagrangian model for the system shown in Figure 3.2 is given by,

$$\mathrm{M}(\theta)\ddot{\theta} + \mathrm{V}(\theta,\dot{\theta}) + \mathrm{C}\dot{\theta} + \mathrm{K}\theta + \mathrm{D} = \mathrm{T}, \tag{3.1}$$

where θ represents the generalized coordinate vector θ_i ; $\mathrm{M}(\theta)$ is the inertia matrix, $\mathrm{V}(\theta, \dot{\theta})$ is the vector of Coriolis and centrifugal functions; C is the viscous damping matrix; D is the Coulomb friction vector; K is the stiffness coefficient matrix, and T is the torque vector from the motors. In this work, motors are assumed to be ideal torque sources.

The inertia matrix $\mathrm{M}(\theta)$ in Equation (3.1) is calculated using

$$\mathrm{M}(\theta) = \begin{bmatrix} \mathrm{M}_1(\theta_2) & \mathrm{M}_2 \\ \mathrm{M}_2^T & M_3 \end{bmatrix}, \tag{3.2}$$

in which

$$\mathrm{M}_1(\theta_2) = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}, \tag{3.3}$$

$$\mathrm{M}_2 = \begin{bmatrix} 0 & m_{41} \\ 0 & 0 \end{bmatrix}, \tag{3.4}$$

$$\mathrm{M}_3 = \begin{bmatrix} m_{33} & 0 \\ 0 & m_{44} \end{bmatrix}. \tag{3.5}$$

The individual elements in the inertia matrix are described using the equations

$$\begin{aligned} m_{11} = {} & m_1 a_1^2 + m_2(l_1^2 + a_2^2) + m_4 b_1^2 + m_6 l_1^2 \\ & + J_1 + J_2 + J_4 + J_6 + 2 l_1 m_2 a_2 \cos(\theta_2) \, , \end{aligned} \tag{3.6}$$

$$m_{12} = m_{21} = m_2 a_2^2 + J_2 + l_1 m_2 a_2 \cos(\theta_2), \tag{3.7}$$

$$m_{14} = m_{41} = J_4 + \frac{J_6}{r}, \tag{3.8}$$

$$m_{22} = m_2 a_2^2 + J_2 , \tag{3.9}$$

$$m_{33} = J_3 + \frac{J_5}{r^2}, \tag{3.10}$$

$$m_{44} = J_4 + \frac{J_5}{r^2}, \tag{3.11}$$

where m_i indicates the lumped masses; J_i refers to the moments of inertia; l_i is the length of the corresponding link; a_i is the distance between the center of gravity of the link and its joint; b_1 represents the distance between the second motor and the first joint; the gear ratio of the transmission elements in the belt drive is represented by r.

The vector of Coriolis and Centrifugal functions $\mathrm{V}(\theta, \dot{\theta})$ is given by

$$\mathrm{V}(\theta, \dot{\theta}) = \begin{bmatrix} V_L \\ 0 \end{bmatrix} = \begin{bmatrix} -l_1 m_2 a_2 \left(2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2\right) \sin(\theta_2) \\ l_1 m_2 a_2 \dot{\theta}_1^2 \sin(\theta_2) \\ 0 \\ 0 \end{bmatrix}, \tag{3.12}$$

The viscous damping matrix C is given by

$$\mathrm{C} = \begin{bmatrix} c_1 + c_5 & 0 & -\frac{c_5}{r} & 0 \\ 0 & c_2 + c_6 & 0 & -\frac{c_6}{r} \\ -\frac{c_5}{r} & 0 & c_3 + \frac{c_5}{r^2} & 0 \\ 0 & -\frac{c_6}{r} & 0 & c_4 + \frac{c_6}{r^2} \end{bmatrix}, \tag{3.13}$$

The matrix of stiffness coefficients K is given by

$$\mathrm{K} = \begin{bmatrix} k_5 & 0 & -\frac{k_5}{r} & 0 \\ 0 & k_6 & 0 & -\frac{k_6}{r} \\ -\frac{k_5}{r} & 0 & \frac{k_5}{r^2} & 0 \\ 0 & -\frac{k_6}{r} & 0 & \frac{k_6}{r^2} \end{bmatrix}, \tag{3.14}$$

where k_i represents the torsional spring coefficients; c_i in Equation (3.13) represents the viscous friction coefficients; $\dot{\theta}_1$and $\dot{\theta}_2$ in Equation (3.12) represents the angular velocity of links 1 and 2, respectively.

The matrix of Coulomb friction D is given by,

$$\mathrm{D} = \begin{bmatrix} D_L \\ D_m \end{bmatrix} = \begin{bmatrix} d_1 \mathrm{sign}(\dot{\theta}_1) \\ d_2 \mathrm{sign}(\dot{\theta}_2) \\ d_3 \mathrm{sign}(\dot{\theta}_3) \\ d_4 \mathrm{sign}(\dot{\theta}_4) \end{bmatrix}, \tag{3.15}$$

The torque vector T is given by

$$\mathrm{T} = \begin{bmatrix} 0 \\ T_M \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ T_1 \\ T_2 \end{bmatrix}, \tag{3.16}$$

in which T_i represents the driving torque for the motors of each link, respectively.

3.2.2 Reduced Model

Designing a model-based feedback controller based on the complete Lagrangian model described in section 3.2.1 because of all its complexities is very difficult. Hence, a simplified

model of the system that preserves the important dynamic behavior of the system was needed for the ease of controller design. Spong in [31] proposed a reduced model which was recognized as a very good simplification of the complete Lagrangian model. The simplified model can be obtained from the complete model by making two important assumptions:

1. The damping coefficients of the torsional springs, c_5 and c_6, can be neglected due to their small magnitudes.
2. The kinetic energy of the motors is mainly due to their own rotation when the gear ratio r >>1 as the angular velocity of the rotor in the motor will be significantly higher when compared with the angular velocity of the link.

Based on the first assumption, the viscous damping matrix defined in Equation (3.13) can be simplified to the form

$$\mathrm{C_{red}} = \begin{bmatrix} \mathrm{C_{Link}} & 0 \\ 0 & \mathrm{C_{Motor}} \end{bmatrix} = \begin{bmatrix} c_1 & 0 & 0 & 0 \\ 0 & c_2 & 0 & 0 \\ 0 & 0 & c_3 & 0 \\ 0 & 0 & 0 & c_4 \end{bmatrix}. \tag{3.17}$$

Based on the second assumption, the inertia matrix $\mathrm{M}(\theta)$ in Equation 3.2 can be simplified to have zero non-diagonal elements in the form

$$\mathrm{M_{red}}(\theta) = \begin{bmatrix} \mathrm{M}_1(\theta_2) & 0 \\ 0 & M_3 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & 0 & 0 \\ m_{21} & m_{22} & 0 & 0 \\ 0 & 0 & m_{33} & 0 \\ 0 & 0 & 0 & m_{44} \end{bmatrix}. \tag{3.18}$$

The complete Lagrangian equation after the two assumptions can be simplified to a general equation of the form

$$\mathrm{M_{red}}(\theta)\ddot{\theta} + \mathrm{V}\left(\theta, \dot{\theta}\right) + \mathrm{C_{red}}\,\dot{\theta} + \mathrm{K}\theta = \mathrm{T}. \tag{3.19}$$

By substituting the reduced inertia matrix and reduced viscous damping matrix in Equation (3.19) and simplifying, we can see that the cross-coupling terms have been eliminated and the matrix relations for the entire system can now be separated into two separate equations for the links and motors, respectively, as

$$\mathrm{M_1}(\theta)\ddot{\theta}_{\mathrm{link}} + \mathrm{V_{link}}\left(\theta_{\mathrm{link}}, \dot{\theta}_{\mathrm{link}}\right) + \mathrm{C_{link}}\,\dot{\theta}_{\mathrm{link}} + \mathrm{K_s}\left(\theta_{\mathrm{link}} - \frac{\theta_{\mathrm{motor}}}{r}\right) = 0 \tag{3.20}$$

and

$$\mathrm{M_3}\ddot{\theta}_{\mathrm{motor}} + \mathrm{C_{motor}}\,\dot{\theta}_{\mathrm{motor}} + \mathrm{K_s}\left(\frac{\theta_{\mathrm{motor}}}{r^2} - \frac{\theta_{\mathrm{link}}}{r}\right) = \mathrm{T_{motor}}, \tag{3.21}$$

where $\mathrm{K_s}$ present in both Equations (3.20) and Equations (3.21) is

$$\mathrm{K_s} = \begin{bmatrix} k_5 & 0 \\ 0 & k_6 \end{bmatrix}, \tag{3.22}$$

which shows that the torsional springs in the joints connect the motors and the links. θ_{link} and θ_{motor} are defined as

$$\theta = \begin{bmatrix} \theta_{\mathrm{link}} \\ \theta_{\mathrm{motor}} \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix}, \tag{3.23}$$

By defining the state variable x as

$$\mathrm{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \theta_{\mathrm{Link}} \\ \theta_{\mathrm{Motor}} \\ \dot{\theta}_{\mathrm{Link}} \\ \dot{\theta}_{\mathrm{Motor}} \end{bmatrix}, \tag{3.24}$$

the reduced model Equations (3.20) and (3.21) can be expressed as a 4$^{\mathrm{th}}$-order state-space system with the following relations

$$\dot{\mathrm{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \\ -\mathrm{M}_1(\theta)^{-1}\left[\mathrm{V}_{\mathrm{link}} + \mathrm{C}_{\mathrm{link}}\dot{\theta}_{\mathrm{link}} + \mathrm{K}_{\mathrm{s}}\left(\theta_{\mathrm{link}} - \frac{\theta_{\mathrm{motor}}}{r}\right)\right] \\ -\mathrm{M}_3^{-1}\left[\mathrm{T}_{\mathrm{motor}} - \mathrm{C}_{\mathrm{motor}}\,\dot{\theta}_{\mathrm{motor}} - \mathrm{K}_{\mathrm{s}}\left(\frac{\theta_{\mathrm{motor}}}{r^2} - \frac{\theta_{\mathrm{link}}}{r}\right)\right] \end{bmatrix}. \tag{3.25}$$

3.3 System Parameters

Generating shaped commands, designing controllers for the system, developing a good simulation model of the system to predict the effectiveness of the method all hinge on having accurate system parameters that can closely represent the actual dynamics of the system. When dealing with nonlinear systems and systems with flexibility, particularly ones like the two-link robotic arm which can possibly violate the modelling assumptions, system identification to get a good estimate of the parameters becomes a vital step. The system identification of the two-link robot was first done by Nho [30]. Using a linear least-square regression approach, Nho created groups of parameters that were used to linearize the Lagrangian model. With this approach, Nho

was able to estimate all the parameters of the robot at the same time. All the experiments conducted in [30] to obtain system parameters were of the open-loop approach.

The first attempt at closed-loop estimation of the robot parameters was done by Lee in [32]. The estimated parameters with the first attempt led to unstable results in simulations, so Lee later used an approach based on a Fourier regularization algorithm that he had implemented on a NARMAX model of the robot. While this approach resulted in a better estimate of the system parameters, a torque offset remained in the simulation model.

A new system identification approach was introduced by Scheel in [33] in which the parameter estimation is split into multiple smaller experiments instead of estimating all the parameters simultaneously. This method gives the flexibility to tailor experiments specifically to a parameter set so that we can estimate the parameters with a higher degree of accuracy and reduces estimation error of one parameter affecting the other parameters. Scheel divided the system identification process into three separate experiments. The three experiments were used to identify the parameters for the two motors, Link 1, and Link 2, respectively. By using this method, the estimates of the parameters obtained were able to closely approximate the important dynamic behavior of the robot. So, this set of parameters listed in Table 3.1 has been used in the command shaping, simulation models and design of controller in this work.

Parameters p_1 , p_2 and p_3 in Table 3.1 are given by the expressions

$$p_1 = m_1 a_1^2 + m_2 l_1^2 + m_4 b_1^2 + m_6 l_1^2 + J_1 + J_4 + J_6 \tag{3.26}$$

$$p_2 = m_2 a_2^2 + J_2 \tag{3.27}$$

and

$$p_3 = l_1 m_2 a_2. \tag{3.28}$$

Table 3.1. Identified Parameters for the Two-link Robot

Parameter	Value	Parameter	Value
p_1	$0.140\ \frac{\text{kg m}^2}{\text{rad}}$	c_4	$1.497 \cdot 10^{-3}\ \frac{\text{Nms}}{\text{rad}}$
p_2	$0.0196\ \frac{\text{kg m}^2}{\text{rad}}$	c_5	$0.005\ \frac{\text{Nms}}{\text{rad}}$
p_3	$0.0234\ \frac{\text{kg m}^2}{\text{rad}}$	c_6	$8.128 \cdot 10^{-5}\ \frac{\text{Nms}}{\text{rad}}$
J_3	$4.157 \cdot 10^{-5}\ \frac{\text{kg m}^2}{\text{rad}}$	k_5	$2.848\ \frac{\text{Nm}}{\text{rad}}$
J_4	$7.543 \cdot 10^{-4}\ \frac{\text{kg m}^2}{\text{rad}}$	k_6	$2.848\ \frac{\text{Nm}}{\text{rad}}$
J_5	$0.025\ \frac{\text{kg m}^2}{\text{rad}}$	d_1	0.0199 Nm
J_6	$0.025\ \frac{\text{kg m}^2}{\text{rad}}$	d_2	0.0323 Nm
c_1	$0.04\ \frac{\text{Nms}}{\text{rad}}$	d_3	0.0053 Nm
c_2	$0.0214\ \frac{\text{Nms}}{\text{rad}}$	d_4	0.0271 Nm
c_3	$1.894 \cdot 10^{-4}\ \frac{\text{Nms}}{\text{rad}}$		

3.4 Computed Torque Controller

Advanced control strategies such as adaptive control, robust control, learning control and so on have been proposed by researchers over the years to optimize performance of flexible systems. A few of the control strategies developed are discussed in [34], and [35]. A computed torque controller is used in this work to achieve accurate trajectory tracking, good disturbance rejection and insensitivity to modelling uncertainties. In the computed torque approach, control is set up

such that the nonlinearities are cancelled out using estimates from the robot model. Using the computed torque approach, error equations in their linear form can be decoupled because of the reduced nonlinear behavior. Originally developed to be used with rigid-joint systems, it was shown in [18] that computed torque controllers can be effective in controlling robotic systems with flexible joints.

Control law partitioning is used in the computed torque approach to divide the control effort into feedforward and feedback sections. The feedforward part of the computed torque controller, also known as the model-based part, attempts to reduce the nonlinear behavior of the system by making use of the robot model, while the feedback portion, also called the servo-based part, is a simple PD controller that works to ensure asymptotic tracking of the input trajectories, disturbance rejection and compensation for modelling uncertainties.

The model-based part in the computed torque controller uses a version of the complete Lagrangian model of the robot described in Equation (3.1) in which the inertia matrix is replaced by the inertia matrix of the Spong model defined in Equation (3.18). By doing so, the cross-coupling terms are neglected, and the nonlinear behavior can be simplified for better control of the system. Let $\mathrm{C_H}$ represent the last two rows of the viscous damping matrix in Equation (3.13); $\mathrm{K_s}$ represents the simplified matrix of the torsional spring coefficients as described in Equation (3.22). The equation for the model-based part can be written as

$$\mathrm{T_{mb}} = \mathrm{C_H}\,\dot{\theta}_\mathrm{M} + \mathrm{D_M} + \mathrm{K_s}\left(\frac{\theta_\mathrm{M}}{r^2} - \frac{\theta_\mathrm{L}}{r}\right), \tag{3.29}$$

where θ_L and θ_M represent the link and motor positions, respectively. The equation for the servo-based part is written as

$$T_{sb} = M_3 \left(\ddot{\theta}_{M,d} + K_V \left(\dot{\theta}_{M,d} - \dot{\theta}_M \right) + K_p \left(\theta_{M,d} - \theta_M \right) \right), \quad (3.30)$$

where K_p and K_v stand for the diagonal matrices of proportional and derivative gains; terms with index d in Equation (3.30) represent the desired values of motor acceleration, velocity and position in the input.

The net torque input to the system is the summation of the torques from the model-based part and the servo-based part as given by

$$T_{CT} = T_{mb} + T_{sb}. \quad (3.31)$$

Define the motor tracking error as

$$e_M = \theta_{M,d} - \theta_M. \quad (3.32)$$

Equating Equation (3.31) to the model of the robot presented in Equation (3.1) gives the closed-loop motor tracking error dynamics as

$$\ddot{e}_M + K_v \dot{e}_M + K_p e_M = 0. \quad (3.33)$$

K_p and K_v are chosen such that the Hurwitz criterion is satisfied, and therefore the computed torque assures stable internal dynamics of the link subsystem and asymptotic tracking of the input. Figure 3.3 shows a schematic of the implementation of the computed torque controller on the two-link robot in the block diagram form.

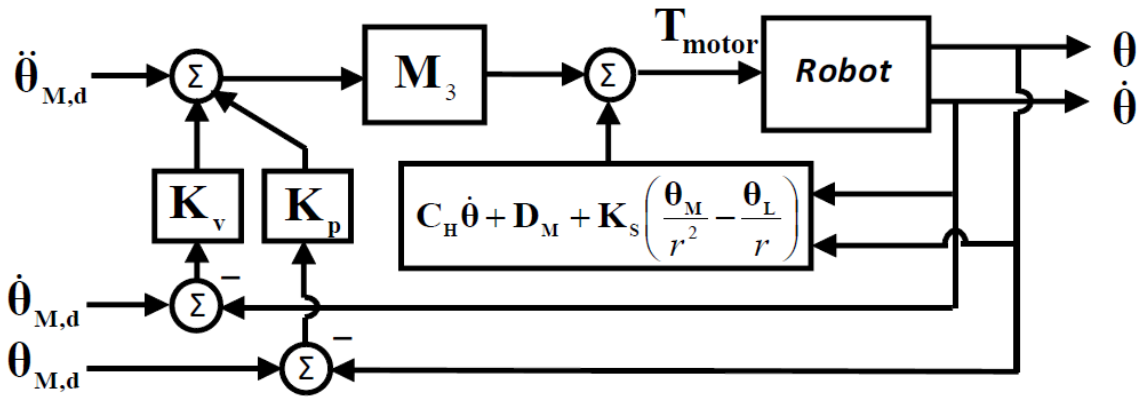

Figure 3.3. Block Diagram of the Computed Torque Controller on the Robotic arm [21]

3.5 Configuration- Dependent Resonance

The natural frequencies of the two-link robot chosen in this work vary based on the configurations of the two links, so it becomes necessary to accurately calculate the varying natural frequencies of the robot to ensure maximum effectiveness of the command shaping approach. This calculation is done by linearization of the nonlinear system around an equilibrium point by assuming the feedback controller and the robot as the complete closed-loop system. For the robot selected, as the configurations of the two links varies, the moments of inertia of the robotic manipulator change, resulting in change of the natural frequencies of the system.

Substituting for T from Equation (3.31) in the complete Lagrangian model given by Equation (3.1), the closed-loop equation of motion is calculated. The obtained equations are linearized about θ_2 using Taylor series expansions up to the first-order term for an equilibrium point where velocity and acceleration both have zero value. The linearization of the equations is done with respect to θ_2 as the matrix of inertia for the robot and hence the natural frequency of the system is a function of θ_2 in both the complete Lagrangian model and the reduced Spong model. Coulomb friction terms do not appear in the linearized equations as the derivatives of the coulomb friction terms with

respect to angular velocities are assumed to be zero. The resulting closed-loop equations after making the above-mentioned changes is given by

$$\mathrm{M_{lin}}\ddot{\theta} + \mathrm{C_{lin}}\dot{\theta} + \mathrm{K_{lin}}\theta = 0, \tag{3.34}$$

where the linearized matrix of inertia is given by

$$\mathrm{M_{lin}} = \begin{bmatrix} m_{11,lin} & m_{12,lin} & 0 & 0 \\ m_{21,lin} & m_{22} & 0 & 0 \\ 0 & 0 & m_{33} & 0 \\ 0 & 0 & 0 & m_{44} \end{bmatrix}, \tag{3.35}$$

in which $m_{11,lin}$, $m_{12,lin}$ and $m_{21,lin}$ are the linearized matrix elements given by the expressions

$$m_{11,lin} = m_1 a_1^2 + m_2(\, l_1^2 + a_2^2\,) + m_4 b_1^2 + m_6 l_1^2 + J_1 + J_2 + J_4 + J_6 + 2 l_1 m_2 a_2 \cos(\theta_{2,lin}) \tag{3.36}$$

$$m_{12,lin} = m_{21,lin} = m_2 a_2^2 + J_2 + l_1 m_2 a_2 \cos\big(\theta_{2,lin}\big). \tag{3.37}$$

The closed-loop viscous damping matrix represented by $\mathrm{C_{lin}}$ is given by

$$\mathrm{C_{lin}} = \begin{bmatrix} c_1 + c_5 & 0 & -\frac{c_5}{r} & 0 \\ 0 & c_2 + c_6 & 0 & -\frac{c_6}{r} \\ 0 & 0 & m_{33}K_{v,3} & 0 \\ 0 & 0 & 0 & m_{33}K_{v,4} \end{bmatrix} \tag{3.38}$$

The closed-loop stiffness matrix represented by $\mathrm{K_{lin}}$ is given by

$$\mathrm{K_{lin}} = \begin{bmatrix} k_5 & 0 & -\frac{k_5}{r} & 0 \\ 0 & k_6 & 0 & -\frac{k_6}{r} \\ 0 & 0 & m_{33}K_{p,3} & 0 \\ 0 & 0 & 0 & m_{44}K_{p,4} \end{bmatrix} \tag{3.39}$$

where $K_{v,i}$ and $K_{p,i}$ are the derivative and proportional gains for the PD controller in the feedback loop.

The natural frequencies of the closed-loop system with the linearized parameters can now be determined by calculating the imaginary part of the eigenvalues of the system matrix A, which is given by

$$\mathrm{A} = \begin{bmatrix} 0 & \mathrm{I} \\ -\mathrm{M_{lin}^{-1}K_{lin}} & -\mathrm{M_{lin}^{-1}C_{lin}} \end{bmatrix} \tag{3.40}$$

where I is the identity matrix and with $\mathrm{M_{lin}}$, $\mathrm{K_{lin}}$ and $\mathrm{C_{lin}}$ as defined in equations (3.35), (3.38) and (3.39), respectively.

It can be seen from Equations (3.36) and (3.37) that the linearized inertia matrix $\mathrm{M_{lin}}$ is a function of link 2 position and hence the eigenvalues of matrix A are also a function of θ_2. The natural frequencies of the system are also dependent on the closed-loop damping in the system. As the system damping increases, the resonance peak will flatten out. Meckl in [15] studied the influence of system damping on the effectiveness of the command shaping method and found out that system response is negatively impacted if the inherent damping is not considered for generating the shaped inputs or additional damping is introduced after shaped commands are

generated. Meckl in [15] proved that command shaping originally designed to minimize residual vibrations in an undamped system can be successfully used in lightly-damped systems with $\zeta < 0.3$. The two-link robot can be categorized as a lightly-damped system, and therefore it can be used as an experimental setup to study command shaping without considering system damping while generating the shaped inputs. The variation in the natural frequencies, ω_1 and ω_2, of the two-link robot as a function of link 2 position θ_2 is as shown in Figure 3.4.

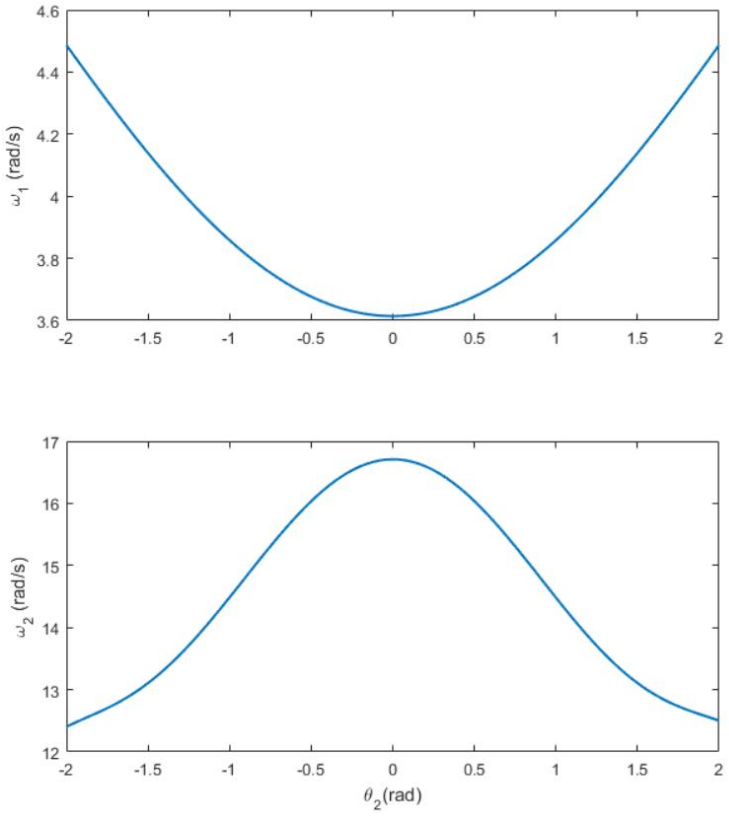

Figure 3.4. Natural frequencies ω_1 and ω_2 as a function of θ_2

CHAPTER 4. APPLICATION TO THE ROBOT

With the numerical optimization framework for command shaping and the hardware setup of the two-link robot explained in detail in Chapter 2 and Chapter 3, respectively, this chapter will focus on the validation of the command shaping method by applying it on the two-link flexible joint robot. Section 4.1 explains the kinematics of the robotic arm and derives performance metrics to quantify the effectiveness of the command shaping method based on the kinematic equations. With a standard metric established to quantify the performance, an introduction to the bang-bang profile and its experimental results are presented in section 4.2. Improvements in the simulation results for the ramped sinusoid are briefly described in section 4.3. A comparison of the simulation results for command shaping with the analytical approach and the numerical approach for the versine basis functions is presented in section 4.4 along with a discussion of the influence of the weighting factor on the command shaping process. Experimental results obtained by applying shaped inputs on the robot are presented in section 4.5 of this chapter.

4.1 Robot Kinematics and Performance Metrics

A well-defined performance metric is necessary to quantify and compare the effectiveness of the command shaping approach on the two-link flexible robot. The focus in this work being on minimizing residual vibrations, the two important features of residual vibrations to be considered are the vibration amplitude and settling time of the vibrations. Most of the literature discussed in this work so far presents methods for vibration measurement in linear single-mode systems for which closed-form analytical expressions for settling time and amplitude can be derived. The key performance features of residual vibrations in such single-mode systems can be calculated by fitting a dissipation envelope to the system response. However, when dealing with nonlinear,

multimode systems, because of the coupling and the interaction that exists between the modes, the methods used for vibration measurements in a single-mode system cannot be directly applied. Therefore, the standard metrics developed in [20] for measuring residual vibration performance for the two-link flexible-joint robot by considering the motion of the end-effector are presented in this work. Ideally, when the input profile ends, the motion of the end-effector should end. This can be used as a basis to say that, irrespective of the direction, any acceleration that persists after the input has ended is undesirable and can be used to quantify the performance of the input profile on the system. The vector diagram of the flexible joint robot is given by defining a reference frame at the base of the robot and two coordinate frames at the ends of the two links as shown in Figure 4.1.

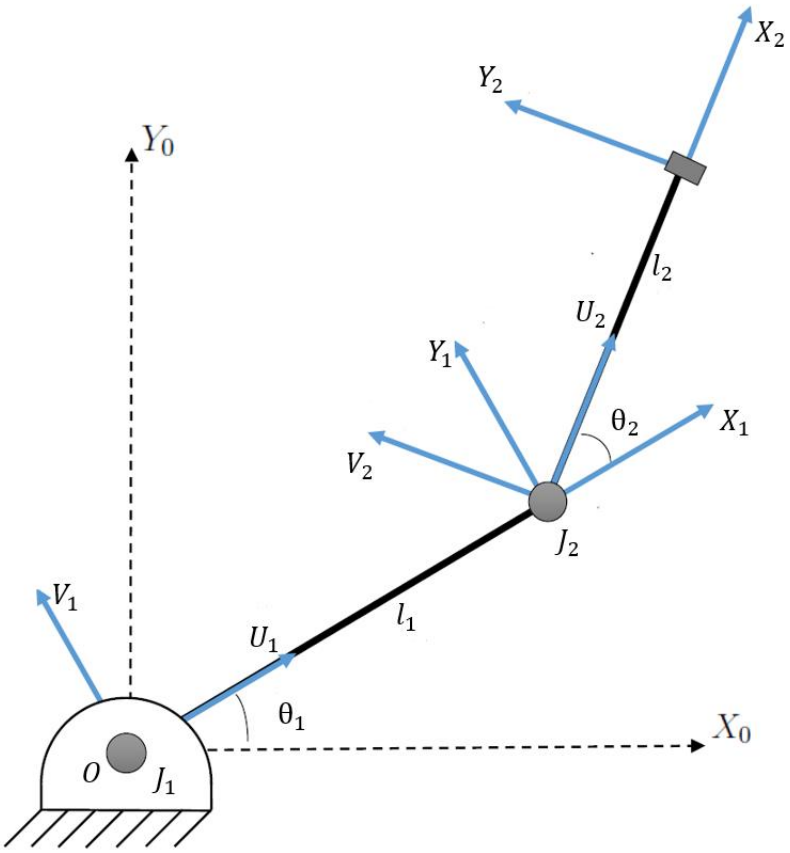

Figure 4.1. Vector Diagram notations of the Robotic Arm. [from 22]

The coordinate frame setup and the notations in Figure 4.1 can be explained as

1. The fixed reference frame at the base of the robot considered the origin is XYZ_0.
2. The coordinate frame placed at the end of link *i* is XYZ_i.
3. The coordinate frame placed at the joint *j* is UVW_j.
4. l_1 and l_2 are the lengths of links 1 and 2, respectively.
5. θ_1 represents the angle between the inertial frame of reference and the 1^{st} link.
6. θ_2 represents the angle between the 1^{st} link and 2^{nd} link.

As the two-link robot is designed to operate in the horizontal plane *X*-*Y*, the axes *Z* and *W* are perpendicular to the plane of motion and considered to be pointing out of the page. It can be seen from Figure 4.1 that the endpoints of links 1 and 2 are in different coordinate frames and hence to bring both links under the same reference frames, a transformation is needed. The rotation and translation matrices for each joint are given by

$$\phi_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.1}$$

$$T_i = \begin{bmatrix} 1 & 0 & 0 & l_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.2}$$

where ϕ_i is the rotational matrix associated with joint *i*, T_i is the translational matrix associated with link *i*.

The conversion relationship for the two-link robot that gives the end-effector coordinates about the inertial reference frame, also called homogeneous manipulator transform, is obtained by multiplying the rotational and translational elements of each link in the appropriate order and is written as

$$T_m = \phi_1 T_1 \phi_2 T_2, \tag{4.3}$$

which can be simplified by substituting the terms from Equations (4.1) and (4.2) to give

$$T_m = \begin{bmatrix} C(\theta_1)C(\theta_2) - S(\theta_1)S(\theta_2) & -S(\theta_1)C(\theta_2) - C(\theta_1)S(\theta_2) & 0 & P_x \\ S(\theta_1)C(\theta_2) + C(\theta_1)S(\theta_2) & C(\theta_1)C(\theta_2) - S(\theta_1)S(\theta)_2 & 0 & P_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.4}$$

in which the individual terms $C(\theta), S(\theta)$ and the end-effector positions P_x and P_y are given by

$$C(\theta) = \cos(\theta), \tag{4.5}$$

$$S(\theta) = \sin(\theta), \tag{4.6}$$

$$P_x = l_1 C(\theta_1) + l_2 C(\theta_1)C(\theta_2) - l_2 S(\theta_1)S(\theta_2), \tag{4.7}$$

$$P_y = l_1 S(\theta_1) + l_2 S(\theta_1)C(\theta_2) + l_2 C(\theta_1)S(\theta_2), \tag{4.8}$$

Using trigonometric relations, the manipulator-transform in Equation (4.4) and the end-effector positions P_x and P_y described above can be simplified to the form

$$T_m = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & P_x \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & P_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.9}$$

$$P_x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)\,, \tag{4.10}$$

$$P_y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2). \tag{4.11}$$

The velocity signals can be found by differentiating the manipulator-transform from Equation (4.3) with respect to time (*t*), which gives the form

$$\frac{\partial T_m}{\partial t} = \frac{\partial \phi_1}{\partial t} T_1 \phi_2 T_2 + \phi_1 T_1 \frac{\partial \phi_2}{\partial t} T_2 = \dot{\theta}_1 Q_R \phi_1 T_1 \phi_2 T_2 + \dot{\theta}_2 \phi_1 T_1 Q_R \phi_2 T_2\,, \tag{4.12}$$

where Q_R, which represents the homogeneous conversion of ϕ_i with respect to time (*t*) for pure rotational motion at a joint, is given by

$$Q_R = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.13}$$

Simplifying Equation (4.12) by substituting for all the terms gives the expressions for linear velocities of the end-effector as

$$\dot{P}_x = -l_1 \dot{\theta}_1 \sin(\theta_1) - l_2\big(\dot{\theta}_1 + \dot{\theta}_2\big)\sin(\theta_1 + \theta_2), \tag{4.14}$$

$$\dot{P}_y = l_1 \dot{\theta}_1 \cos(\theta_1) + l_2\big(\dot{\theta}_1 + \dot{\theta}_2\big)\cos(\theta_1 + \theta_2). \tag{4.15}$$

Accelerations of the end-effectors can be calculated by differentiating Equation (4.12) for velocities with respect to time, which gives the form

$$\begin{aligned} \frac{\partial^2 T_m}{\partial t^2} = {} & \ddot{\theta}_1 Q_R \phi_1 T_1 \phi_2 T_2 + \dot{\theta}_1^2 Q_R^2 \phi_1 T_1 \phi_2 T_2 + 2\dot{\theta}_1 \dot{\theta}_2 Q_R \phi_1 T_1 Q_R \phi_2 T_2 \\ & + \dot{\theta}_2^2 \phi_1 T_1 Q_R^2 \phi_2 T_2 + \ddot{\theta}_2 \phi_1 T_1 Q_R \phi_2 T_2\,. \end{aligned} \tag{4.16}$$

Simplifying Equation (4.16) gives the expressions for linear acceleration of the end-effector as

$$\ddot{P}_x = -l_1\ddot{\theta}_1 \sin(\theta_1) - l_1\dot{\theta}_1^2 \cos(\theta_1) - l_2\left(\dot{\theta}_1^2 + 2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2\right)\cos(\theta_1 + \theta_2) - l_2\left(\ddot{\theta}_1 + \ddot{\theta}_2\right)\sin(\theta_1 + \theta_2) \tag{4.17}$$

$$\ddot{P}_y = l_1\ddot{\theta}_1 \cos(\theta_1) - l_1\dot{\theta}_1^2 \sin(\theta_1) - l_2\left(\dot{\theta}_1^2 + 2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2\right)\sin(\theta_1 + \theta_2) + l_2\left(\ddot{\theta}_1 + \ddot{\theta}_2\right)\cos(\theta_1 + \theta_2) \tag{4.18}$$

The peak amplitude of the residual vibration is defined as the magnitude of the maximum acceleration of the end effector after the input command to the system has ended. The magnitude of translational acceleration can be calculated using Equations (4.17) and (4.18), and the expression is given as

$$\ddot{P}_{xy} = \sqrt{\ddot{P}_x^2 + \ddot{P}_y^2}\,, \tag{4.19}$$

The peak amplitude of residual vibration can be written in the form

$$a_{rv} = \max\left(\ddot{P}_{xy}(t)\right), \tag{4.20}$$

where *t* representing the time for residual vibration measurement can be defined as

$$T_f \leq t \leq T_{end}, \tag{4.21}$$

in which T_f is the input profile completion time and T_{end} represents the predetermined run time for the experiment.

Settling time, which is the second feature for residual vibration performance, was defined as the time required by the end-effector of the robot to settle within a certain percentage of the input amplitude and is measured after the input profile has ended. Initially, the tolerance for settling time was set to either 2% or 5% of the input acceleration, but depending on the input acceleration, this definition sometimes resulted in very high tolerance or very low tolerance for measuring settling, time making the comparisons between different profiles and different accelerations difficult.

Hence, settling time has been redefined as the time required for the end-effector acceleration of the robot to fall below a value of 0.2 m/s^2 and is measured after the input profile has ended. The tolerance value is selected based on results from experiments and simulations such that the value is not too small to be affected by noise. It is also not set too high, and it provides a good basis of measurement to rank the effectiveness of different profiles efficiently. The total move time for the robotic arm can now be defined as

$$T_{total} = T_{actuation} + T_{settling} \tag{4.22}$$

where $T_{actuation}$ is the profile actuation time. It is also represented as T_f; $T_{settling}$ is the settling time based on the standard definition in this work measured after the actuation time. From the definition of the performance metrics, it can be said that, in this work, the input most effective in reducing residual vibration will have the smallest settling time and peak amplitude of residual acceleration.

4.2 Bang-Bang Profile

The multi-objective fitness function used in developing the shaped inputs aims for time-optimality and residual vibration reduction. The objective function is set to approximate a time-optimal bang-bang profile. The inputs developed using command shaping and its performance metrics will be compared against the performance of the bang-bang profile. The acceleration profile of a bang-bang profile as shown in Figure 4.2 consists of constant acceleration and constant deceleration sections both at a given peak acceleration value. The acceleration and deceleration sections are of equal time spans, which is equal to half of the total actuation time.

The bang-bang profile plotted in Figure 4.2 has a maximum acceleration of 6 rad/s^2 and an actuation time of 2 s. A bang-bang input, unlike a shaped input, has no constraints on the energy

distribution at variable frequencies and this can be easily observed from the frequency spectrum plotted in Figure 4.3 for the bang-bang profile shown in Figure 4.2.

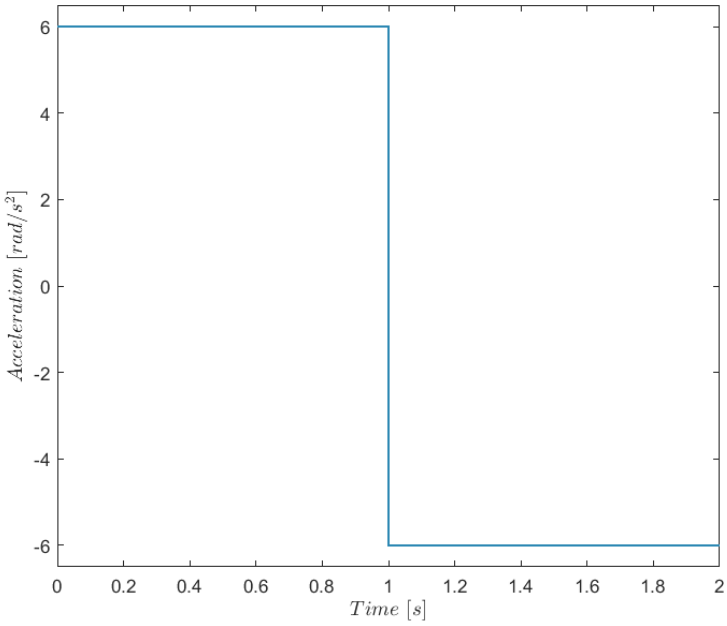

Figure 4.2. Bang-Bang Input Profile for 2s Actuation time.

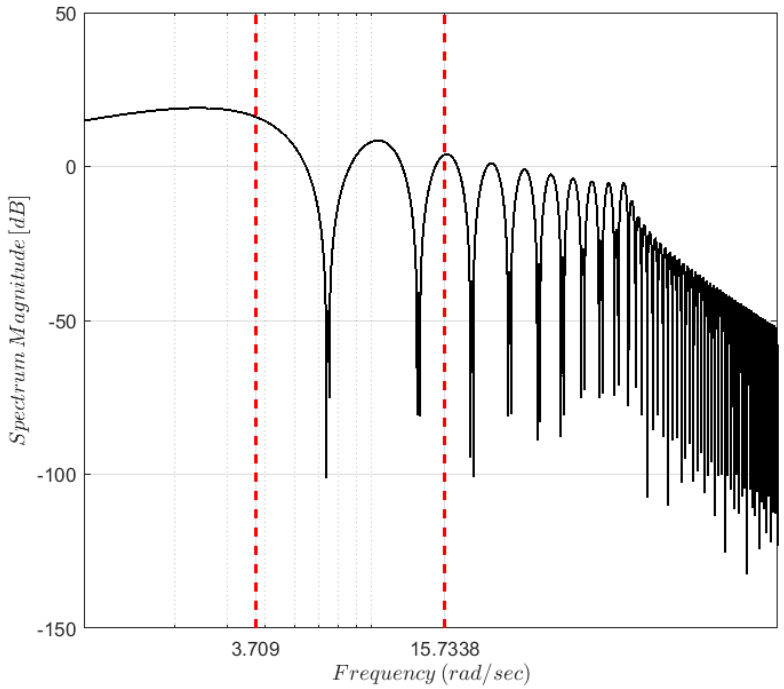

Figure 4.3. Frequency Spectrum of the Bang-Bang profile shown in Figure 4.2.

It can be seen from Figure 4.3 that the bang-bang profile carries significant energy around the resonant frequencies of the robot and hence, this input will excite resonance in the system leading to significant residual vibrations at the end of the move.

To better understand the behavior of a physical system with a bang-bang input, simulation and experiments are conducted using a bang-bang profile on the two-link flexible joint robot. The values of residual vibration amplitude and settling time obtained for this input will be used as a standard of comparison to evaluate the performance of the shaped inputs. The experimental results for the bang-bang profile are shown in Figure 4.4.

The first two sub-plots in the first row of Figure 4.4 represent the desired input acceleration profiles for both motors and the frequency spectrum for the bang-bang profile used. Row 2 of the sub-plots represents the input torque profiles for motors 1 and 2, respectively. Row 3 of the sub-plots represents the angular positions of the 1st and 2nd link, respectively, with the desired final position plotted in dashed lines. Row 4 of the sub-plots shows the actual and desired angular positions plotted on top of each other for motors 1 and 2, respectively. The dotted lines again represent the desired angular position signals. The last row of sub-plots in Figure 4.4 represents the real-time angular accelerations of links 1 and 2, respectively. From seeing the sub-plots for θ_1 and θ_2 from Figure 4.4, it can be observed that there are significant oscillations in both links after the input profile has ended. The settling time for the robotic arm is close to 2.248s which is higher than the actuation-time for the input and the peak residual acceleration for the input is 1.2376 $\mathrm{m/s^2}$.

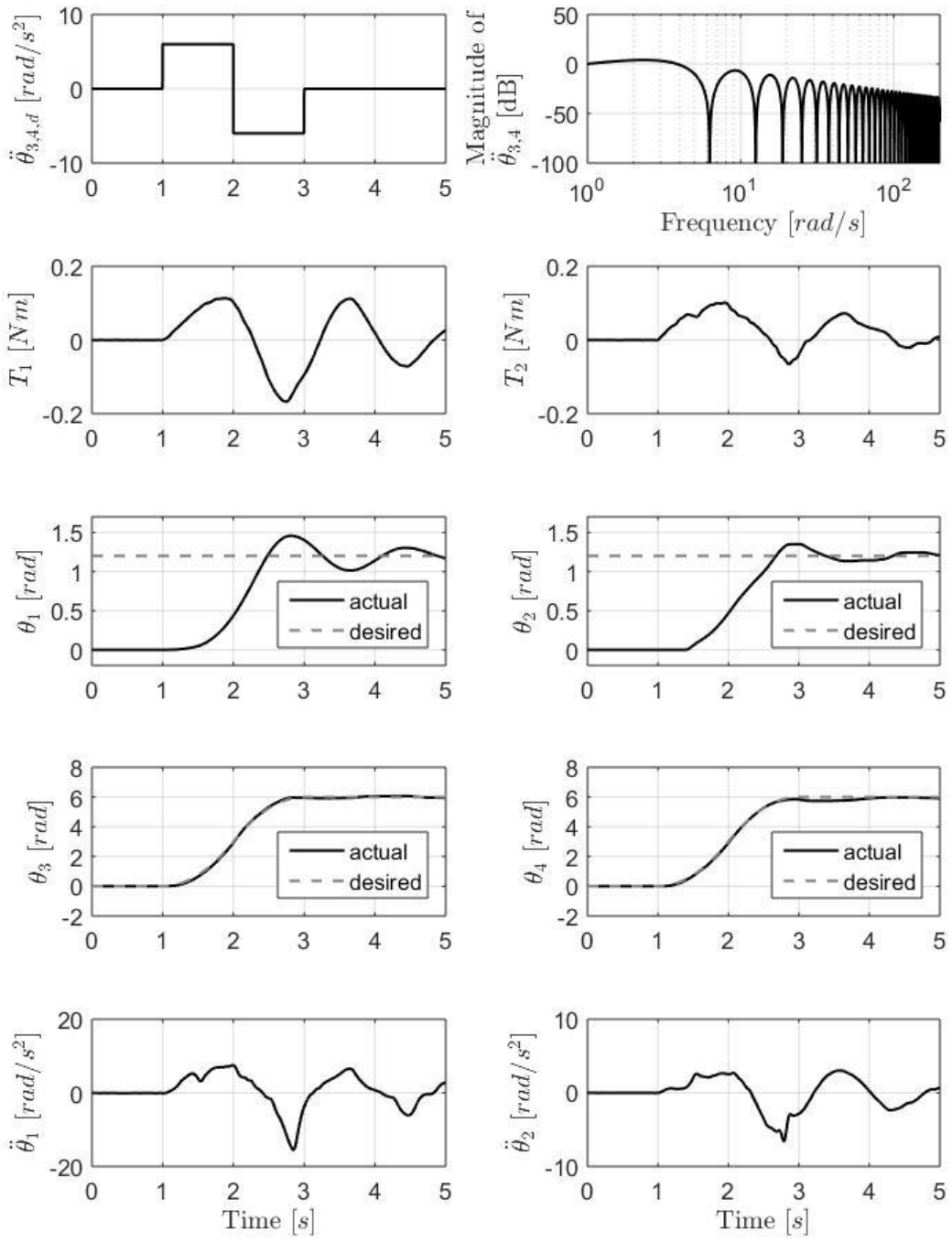

Figure 4.4. Experimental Results for Bang-Bang profile.

4.3 Ramped Sinusoid

The command shaping framework with numerical optimization and constrained peak input acceleration for the ramped sinusoid basis function was developed by Wu [22]. But this initial attempt at combining numerical optimization with constrained peak acceleration, while successful in reducing residual vibration, was severely limited by the range of design parameters such as weighting factor that it could operate with. The numerical solver setup and the results from [22] were studied and the optimization code for the ramped sinusoid was tweaked and restructured before starting work with the versine to gain an understanding of the optimization process. Input profile generation with ramped sinusoids uses two nested loops for numerical optimization and actuation time convergence to calculate the coefficients of the harmonics and the penalty factor Γ.

The original MATLAB setup worked for multiple values of peak acceleration but was limited to values of ρ below 3. The maximum spectral attenuation achieved was around -40dB. The updated version of the MATLAB code for the numerical solver with all the changes made to it can be found in Appendix A. With the changes to the solver and the MATLAB code, the input command shaping method now works for weighting factor ρ values well above 2000. Similar improvements in the energies around the natural frequencies were observed as input profiles with spectral attenuation of -100dB were generated. As ramped sinusoid is not the focus of this work, the updated numerical optimization framework was used with the ramped sinusoid basis function with a peak acceleration of 10 $\mathrm{rad/s^2}$ only. A version of the same code has been used with relevant changes made to incorporate the versine basis function. Comparison of spectral magnitude plots for the ramped sinusoid basis function with the analytical approach and the numerical approach for an acceleration of 10 $\mathrm{rad/s^2}$ has been presented in Appendix B for reference.

4.4 Segmented Versine - Simulation Analysis

Based on the model of the robot developed in Chapter 3, a simulation model of the entire two-link robot setup has been developed in SIMULINK on MATLAB. Shaped input commands for various constraints on peak input acceleration and weighting factors are developed and provided as inputs to the simulation models to study the behavior before implementing the generated commands on the physical systems. Based on the simulation results, initial observations about the effectiveness of the command shaping method on the two-link system have been drawn and these observations will be verified by conducting experiments on the physical robot setup.

Peak acceleration values of 40 $\mathrm{rad/s^2}$, 50 $\mathrm{rad/s^2}$, and 60 $\mathrm{rad/s^2}$ have been selected in this approach for both the numerical and the analytical approaches of command shaping with the segmented versine basis function.

4.4.1 Influence of Weighting Factor

The shaped commands developed hence differ in effectiveness based on the choice of a non-dimensional weighting function ρ used to define the relative significance of both objectives in the multi-objective function. Based on the requirement of the input command, emphasis on time-optimality and residual vibration performance can be adjusted with the help of ρ to develop the optimal input. Hence, it becomes necessary to understand the effect that ρ has on the shaped inputs and its performance on the system. In this work, results from the analytical approach for weighting factors up to a value of 400 are shown, while results for ρ up to 30,000 are shown for the numerical approach. Intermediate values of weighting functions have been selected such that the presented results have clear transition regions and maximum details of the system performance are obtained.

The multi-objective function is set to remove energies around two different bands of frequencies corresponding to the first and second natural frequencies of the system. The total

spectral attenuation achieved through the command shaping method, in this work, is calculated by averaging the spectral magnitude in the window selected for each resonant mode and adding the averages for both the resonant modes of the system to give a single parameter that represents the total energy carried by the input profiles at the resonant modes of the robot,

$$|F(\omega_{12})| = |F(\omega_1)| + |F(\omega_2)| \tag{4.23}$$

where $F(\omega_1)$ and $F(\omega_2)$ are the average magnitudes at the spectral window around the 1st and 2nd natural frequencies calculated using

$$|F(\omega_i)| = \frac{1}{11}\sum_{l=1}^{11} |F(\omega_l)|, \tag{4.24}$$

where $|F(\omega_l)|$ is the magnitude of the spectrum at each l^{th} point for the tolerance window selected for each mode. Figure 4.5 shows the variation of spectral magnitude at the system natural frequencies in the shaped input profile as a function of the weighting factor ρ for all values of peak acceleration under consideration.

As a general observation, it can be seen from Figure 4.5 that, with an increase in the value of ρ, the average energies at the spectral windows selected decrease for all values of peak acceleration. Command shaping using versine has been implemented over a large range of ρ, so due to the high computational time of the numerical approach, a uniform increment of the weighting factor has not been maintained. As can be seen from the markers used in Figure 4.5, shaped inputs have been analyzed with a relatively smaller increment for lower values of weighting factors to accurately capture the changes in the plot. As ρ increases to higher values, the rate of change in spectral magnitude with ρ decreases. Hence, command shaping has been implemented with larger increments around these regions. Figure 4.6 presents a relationship between the

actuation time and ρ using the numerical approach for the three values of peak acceleration under consideration.

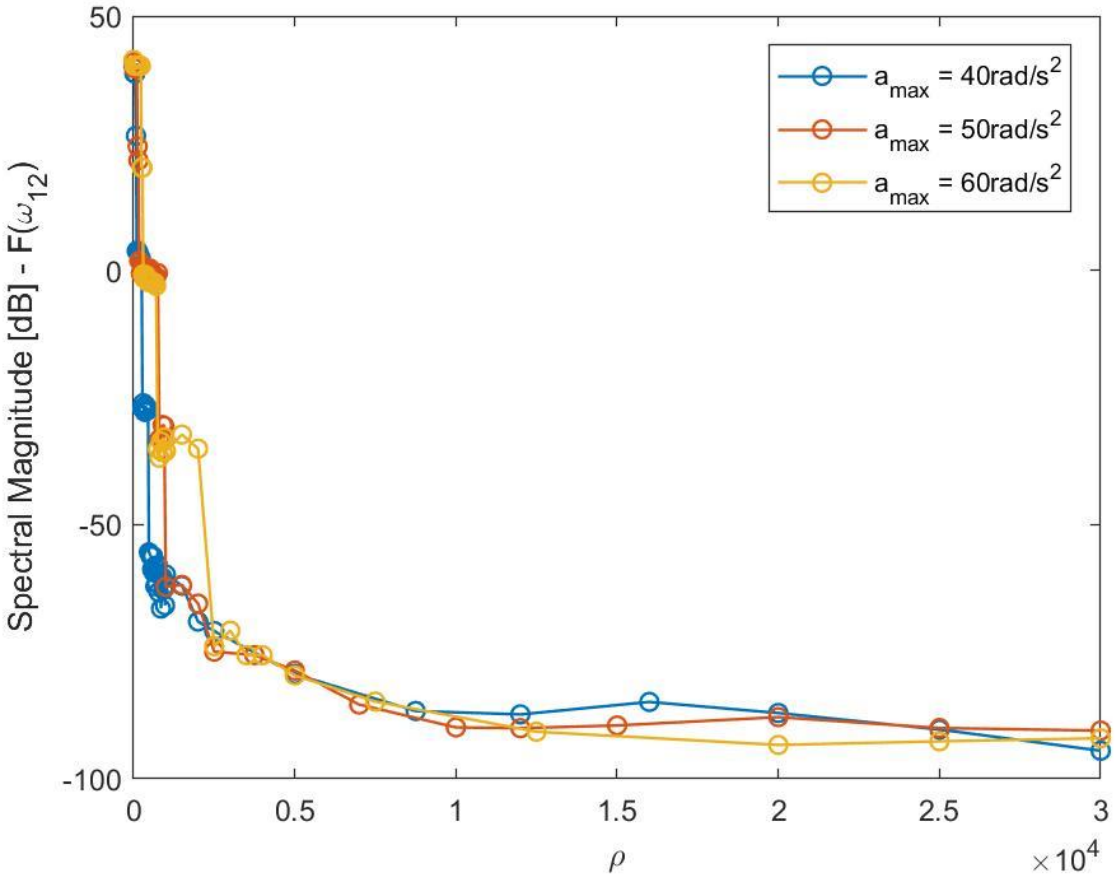

Figure 4.5. Influence of ρ on the Spectral Magnitude: Versine using Numerical Optimization.

Using the multi-objective function with a penalty factor ρ, it is natural to assume that as ρ increases, it would lead to higher attenuation of the spectral magnitude and would lead to better performance. But it can be seen from Figure 4.5 and Figure 4.6, if the saturation value of attenuation in spectral attenuation is achieved at a value of ρ, further efforts to develop inputs for higher values of ρ would result in skewed commands that give larger actuation times without any reduction in the attenuation of the spectral energies. Such commands, while demanding larger move times, would not benefit in any additional residual vibration performance. Thus, identifying

that point of diminishing return becomes essential to effectively using the command shaping approach.

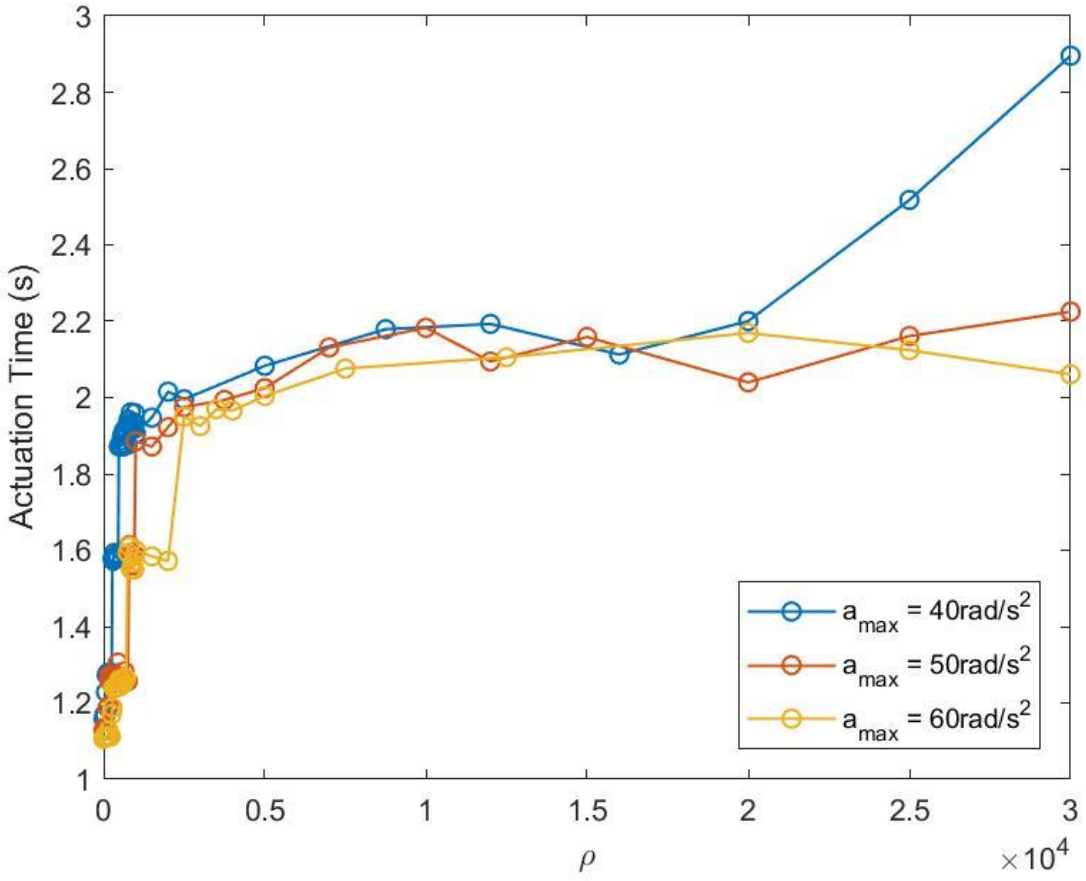

Figure 4.6. Influence of ρ on Actuation time: Versine using Numerical optimization.

From Figure 4.6, for each selected value of peak acceleration, actuation time increases with an increase in ρ, and as the peak velocity increases, actuation time decreases. But for values of ρ higher than 1500, the trends slightly deviate from this relationship because numerical optimization occasionally converges to a local minimum while using a higher penalty in the optimization method. Due to this, at higher values of ρ, values of spectral magnitude for the two modes is seen varying slightly around the maximum value, while also demanding a longer actuation time.

4.4.2 Comparison Between Approaches

An iterative numerical approach is used in this study as opposed to a closed-form analytical approach used in the previous works due to the inherent coupling of the design parameters brought about by changing the problem formulation to eliminate the Gibbs phenomenon. Comparing the results from the numerical optimization process with the analytical approach, it becomes necessary to evaluate if residual vibration performance is being sacrificed with the new formulation to prevent the appearance of the Gibbs phenomenon in the developed inputs. Figure 4.7 presents the variation in the spectral magnitude of the system natural frequencies as a function of weighting function ρ using the analytical approach for the values of peak acceleration considered.

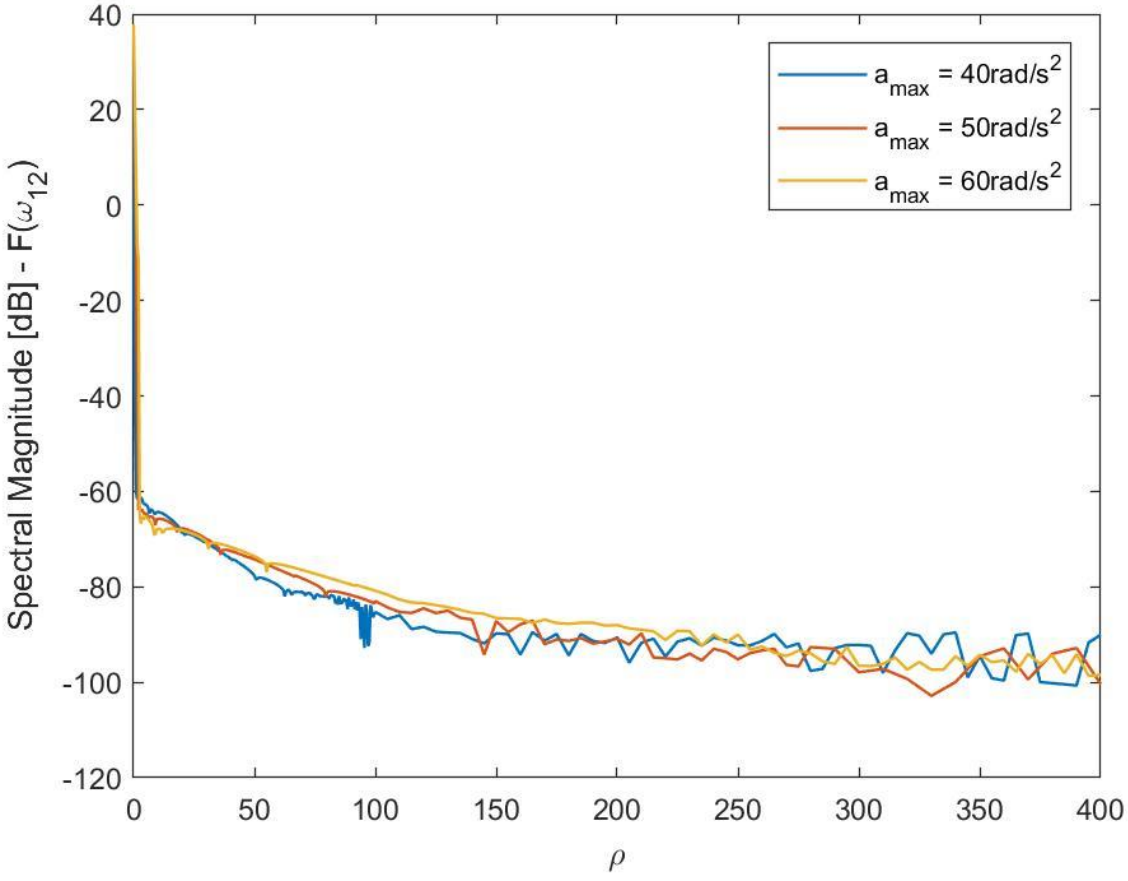

Figure 4.7. Spectral Magnitude vs ρ: Versine using closed-form analytical formulation.

From Figure 4.5 and Figure 4.7, the maximum spectral attenuation for each value of acceleration in the numerical approach and the analytical approach appear to be at similar levels. While this indicates that with the new formulation spectral attenuation is not sacrificed for the elimination of Gibbs phenomenon, it is important to notice the numerical approach and analytical approach of command shaping operate in different ranges of ρ values. Because of this reason, ρ is not an ideal metric to analyze the behavior of the shaped inputs. Based on the results obtained so far, irrespective of the value of ρ being used, the behavior of the system is dependent on the spectral magnitude at the system natural frequencies. Hence, the results presented henceforth will be expressed as a function of spectral magnitude. Figure 4.8 and Figure 4.9 present the actuation time of the input profile as a function of spectral magnitude for the numerical and the analytical approach, respectively.

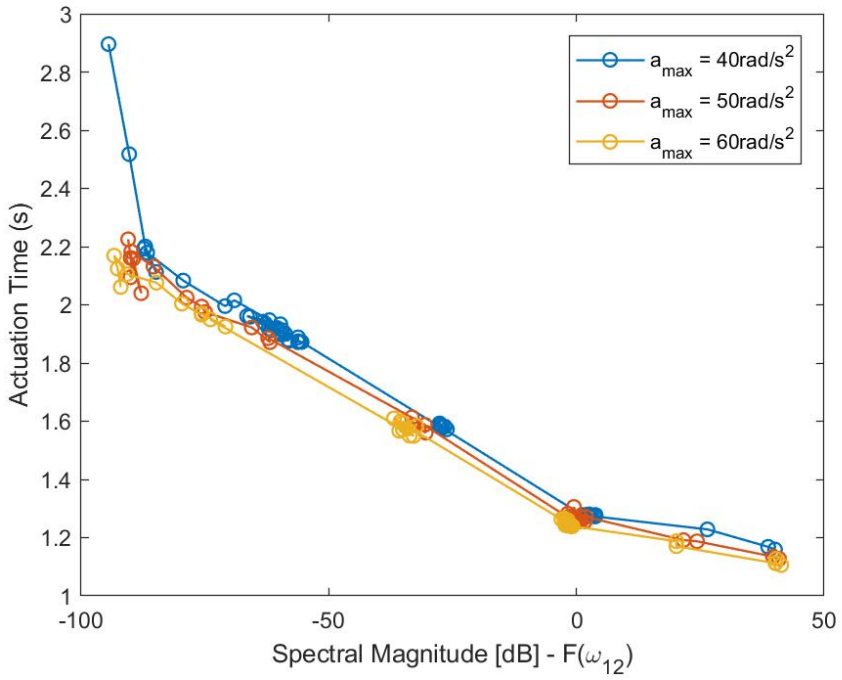

Figure 4.8. Influence of Spectral Magnitude on the actuation time: Versine using Numerical approach.

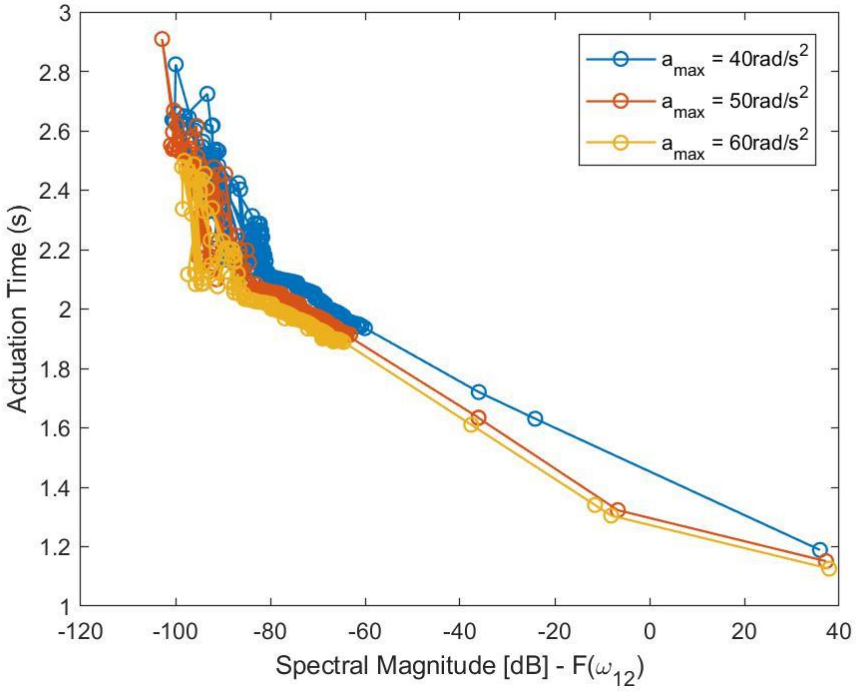

Figure 4.9. Influence of Spectral Magnitude on the actuation time: Versine using Closed-form Analytical formulation.

From the above representation of data, it can be clearly inferred that for both the numerical approach and the analytical approach, the actuation time increases as the spectral magnitude in the shaped input decreases, and actuation time decreases as the peak input acceleration increases. To compare the features of the developed input between the analytical and the numerical approach, Figures 4.10 - 4.12 plot the influence of spectral magnitude on the actuation time for each value of acceleration individually for both approaches.

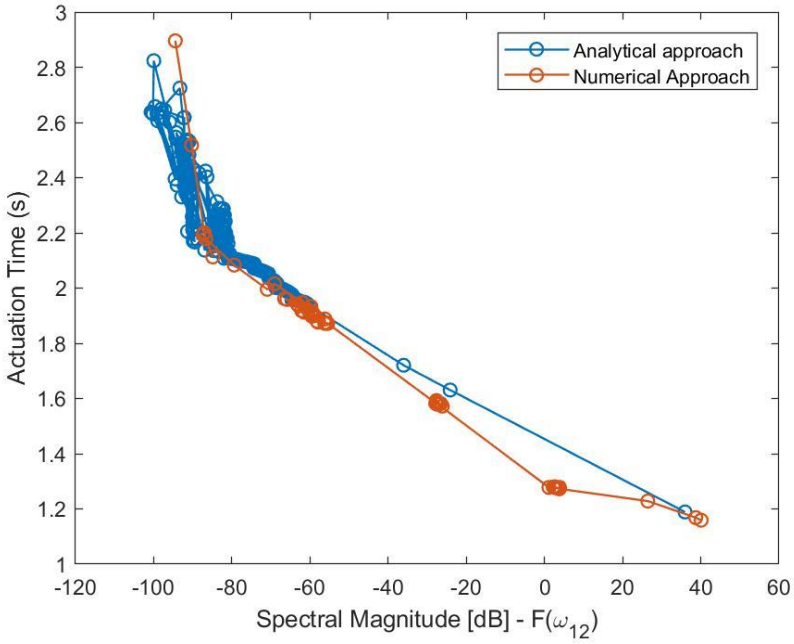

Figure 4.10. Actuation Time vs Spectral Magnitude: Versine using Numerical and Analytical Approaches with a_{max} = 40 rad/s^2.

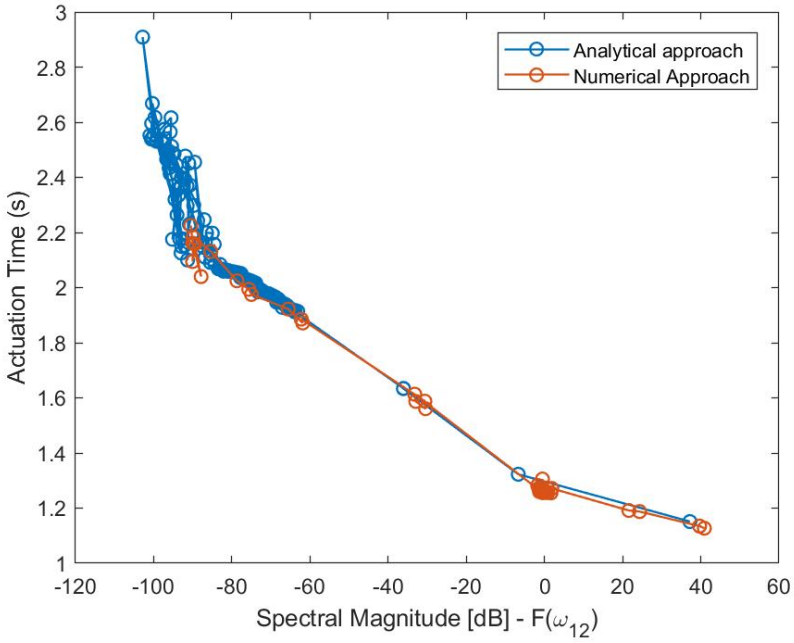

Figure 4.11. Actuation Time vs Spectral Magnitude: Versine using Numerical and Analytical Approaches with a_{max} = 50 rad/s^2.

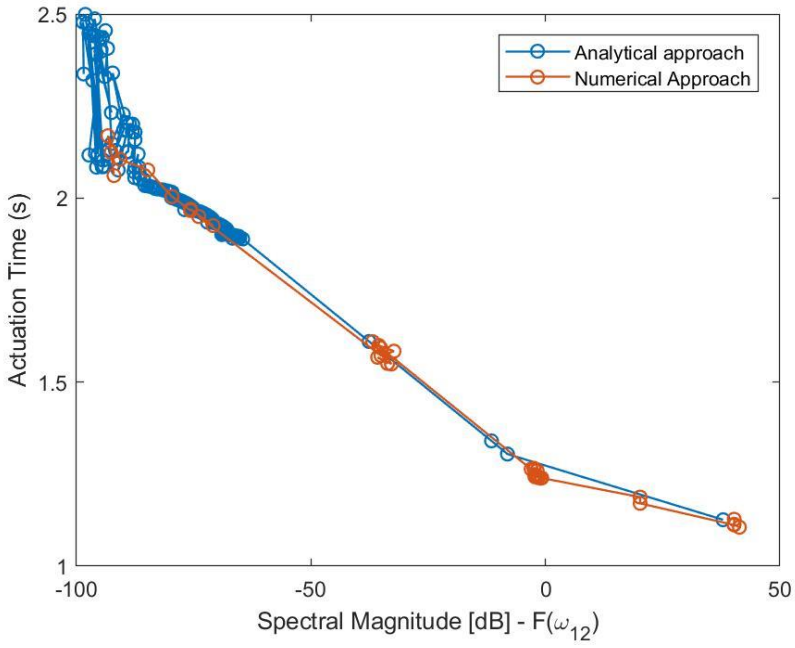

Figure 4.12. Actuation Time vs Spectral Magnitude: Versine using Numerical and Analytical Approaches with a_{max} = 60 rad/s^2.

Even though the difference in actuation time between the two approaches is small, it can be concluded from the plots that the actuation times with the numerical approach for all values of acceleration considered are in general better than the analytical approach. By eliminating Gibbs phenomenon, the overshoots and the undershoots in the shaped input profile flatten out giving a better approximation of the time-optimal bang-bang profile. Improvement in the actuation time of the shaped input using the numerical optimization can be attributed to this improvement in the approximation of the bang-bang profile.

The results discussed so far prove that inputs developed using the command shaping method can successfully constrain the energy around the selected frequencies at the cost of longer move time. Implementing these shaped inputs on a simulation model developed in Simulink will provide insight into the response of the system for the input. It is also helpful in analyzing if the shaped input results in better residual vibration performance by measuring the settling time and residual

vibration amplitude before implementing on the physical robot. For a given input, total move time is defined as the sum of actuation time and settling time. Figures 4.13- 4.15 present plots of total move time for the shaped inputs as a function of spectral magnitude with the numerical and analytical approaches for all accelerations considered individually.

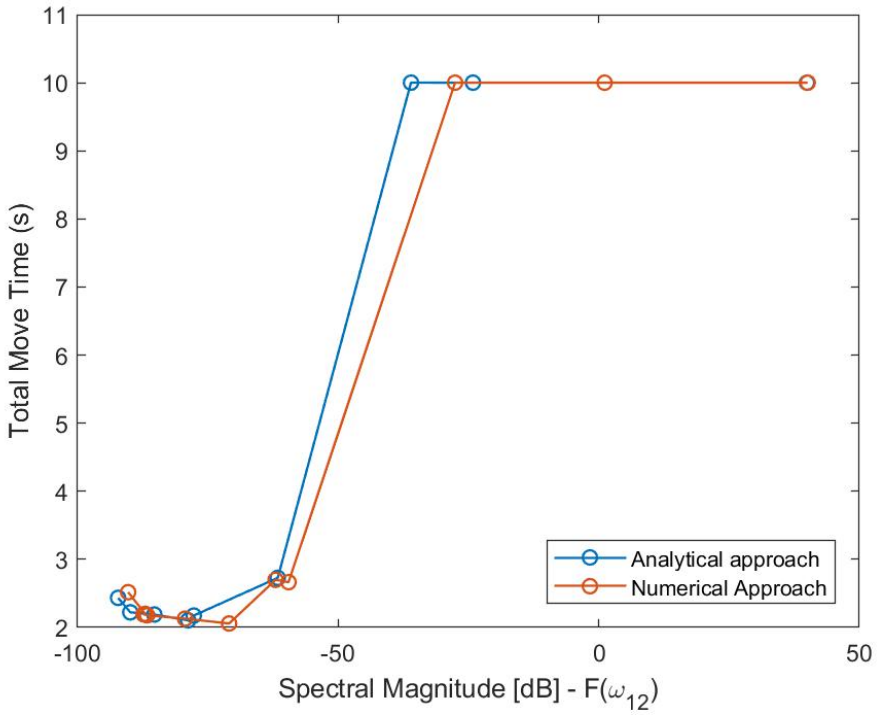

Figure 4.13. Total Move Time vs Spectral Magnitude : Versine with $a_{max} = 40\ \mathrm{rad/s^2}$.

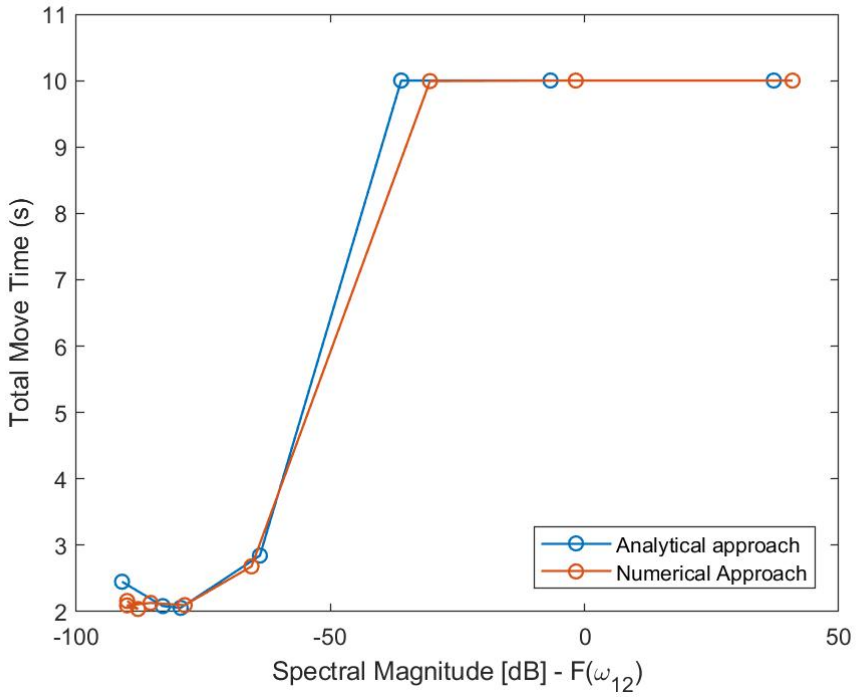

Figure 4.14. Total Move Time vs Spectral Magnitude : Versine with a_{max} = 50 rad/s^2.

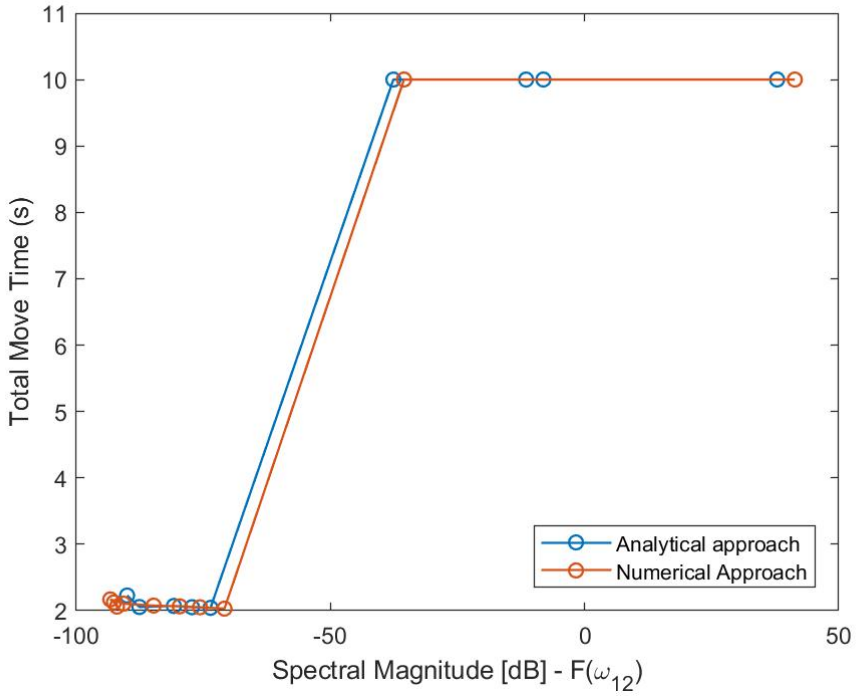

Figure 4.15. Total Move Time vs Spectral Magnitude : Versine with a_{max} = 60 rad/s^2.

From the above figures it can be observed that, irrespective of the value of peak acceleration selected, the total move time using the numerical optimization method is either the same or usually lower than the total move time using the analytical approach. This is because, for similar levels of spectral attenuation, the numerical optimized input has a shorter actuation time as compared to the input from the analytical approach. Figures 4.13 - 4.15 also show a good trend of the influence of spectral magnitude on the total move time of the two-link robot. The trade-offs between total move-time and spectral magnitude can be seen in the above plots and the point of diminishing return can be clearly identified. Total move-time of the two-link robot decreases with a decrease in spectral magnitude until a critical value of spectral magnitude when the settling time reaches zero. Further effort to reduce the energy at the natural frequencies results in inputs that carry less energy at the natural frequencies at longer move times, but these inputs do not improve the residual vibration performance of the system for the selected tolerance as settling time is already zero. Hence, efforts to develop inputs with higher spectral attenuation should be minimized after the settling time reaches zero.

4.5 Experimental Results.

The command shaping method has been proven to reduce the residual vibrations in simulations as explained in section 4.4. Simulation models always behave ideally based on the system equations used to define it, but the simulation may miss irregularities of the physical system that haven't been modeled. The response of the physical system can also be different from the simulation results because parameters that have been estimated incorrectly can cause a mismatch between the physical system and the simulation model. So, validation of the command shaping approach on the actual two-link robot becomes essential to gain a complete understanding of the command shaping method. Based on the simulation results, for each a_{max}, values of weighting

factor ρ have been selected to systematically show the improvements in residual vibration performance. Shaped inputs developed for the combinations of the design parameters selected are implemented on the two-link robot. A minimum of one ρ with settling time greater than zero, equal to zero and in the region of diminishing results is selected for each peak acceleration. Experiments with each input profile have been conducted 5 times and the results have been averaged to minimize the influence of noise. Figures 4.16- 4.18 show the actuation time and experimental values of the total move time plotted as a function of the spectral magnitude for each peak acceleration separately. Tables 4.1- 4.3 list the performance metrics for the set of input profiles with the same a_{max} for different values of ρ to quantify the results obtained from conducting the experiments on the two-link robot.

Table 4.1. Performance metrics with $a_{max} = 40\ \mathrm{rad/s^2}$.

ρ	$\lvert F(\omega_{12})\rvert\ (dB)$	$T_{actuation}(s)$	$T_{settling}(s)$	$T_{total}(s)$	$\ddot{P}_{xy}\ (m/s^2)$
700	-59.52	1.897	1.049	2.946	0.5304
1500	-62.01	1.947	0.505	2.452	0.3850
2500	-70.93	1.995	0.350	2.345	0.2804
8750	-86.66	2.179	0	2.179	0.211
20000	-89.05	2.210	0	2.210	0.1649

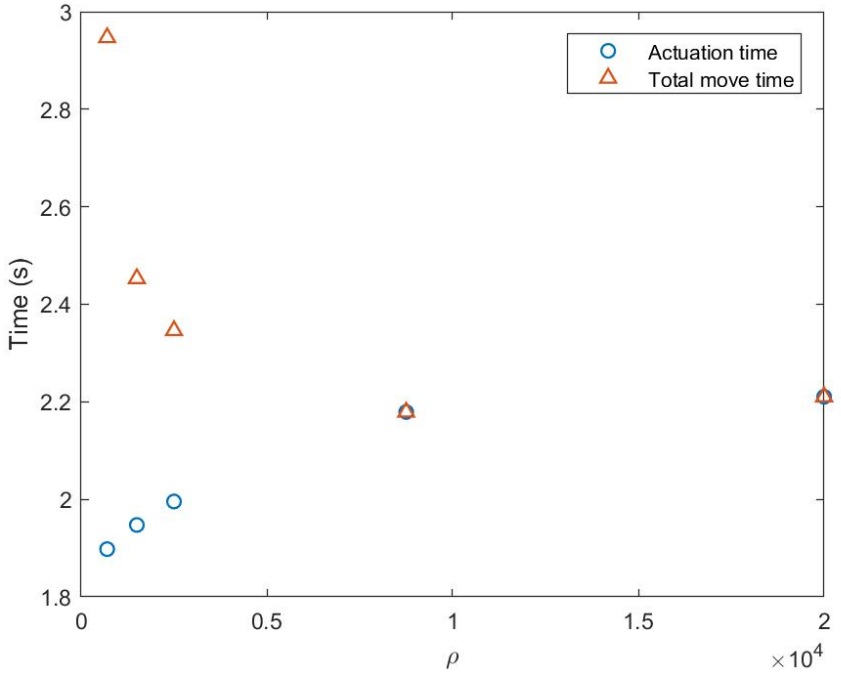

Figure 4.16. Total move time and Actuation time vs ρ: Versine with $a_{max} = 40\ rad/s^2$.

Table 4.2. Performance metrics with a_{max} = 50 rad/s^2.

ρ	$\lvert F(\omega_{12})\rvert\ (dB)$	$T_{actuation}(s)$	$T_{settling}(s)$	$T_{total}(s)$	$\ddot{P}_{xy}\ (m/s^2)$
1500	-62.01	1.894	0.778	2.672	0.5141
2500	-70.93	1.972	0.629	2.601	0.3481
5000	-79.39	2.034	0.261	2.295	0.2865
10000	-87.32	2.192	0	2.192	0.2218
30000	-90.30	2.225	0	2.225	0.1744

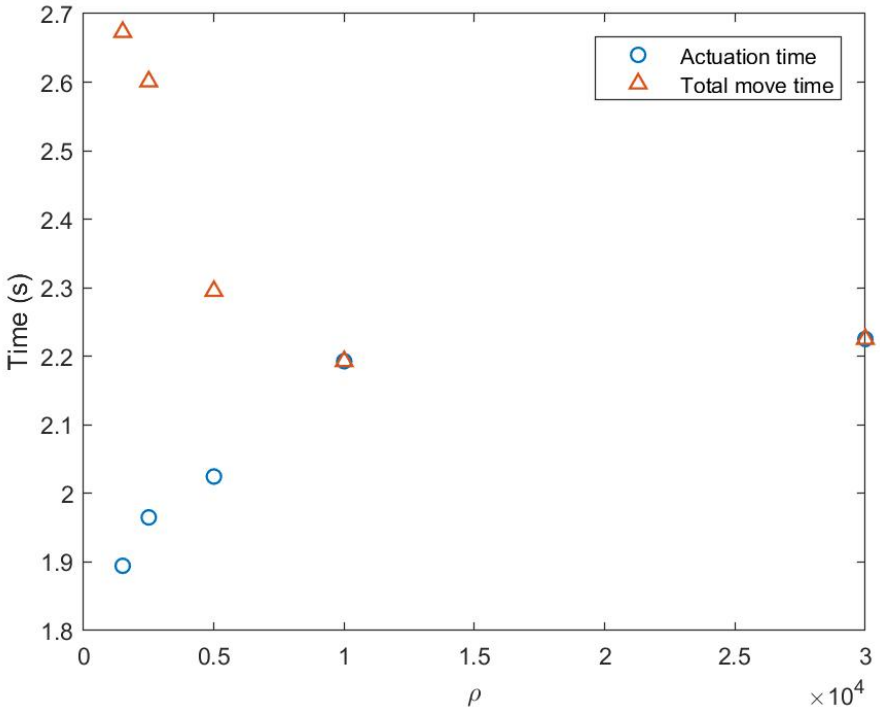

Figure 4.17. Total move time and Actuation time vs ρ: Versine with a_{max}= 50 rad/s^2.

Table 4.3. Performance metrics with a_{max} = 60 rad/s^2.

ρ	$\lvert F(\omega_{12})\rvert\ (dB)$	$T_{actuation}(s)$	$T_{settling}(s)$	$T_{total}(s)$	$\ddot{P}_{xy}\ (m/s^2)$
5000	-79.64	2.004	0.215	2.219	0.3045
7000	-84.80	2.076	0	2.076	0.2185
12500	-90.73	2.105	0	2.105	0.1856

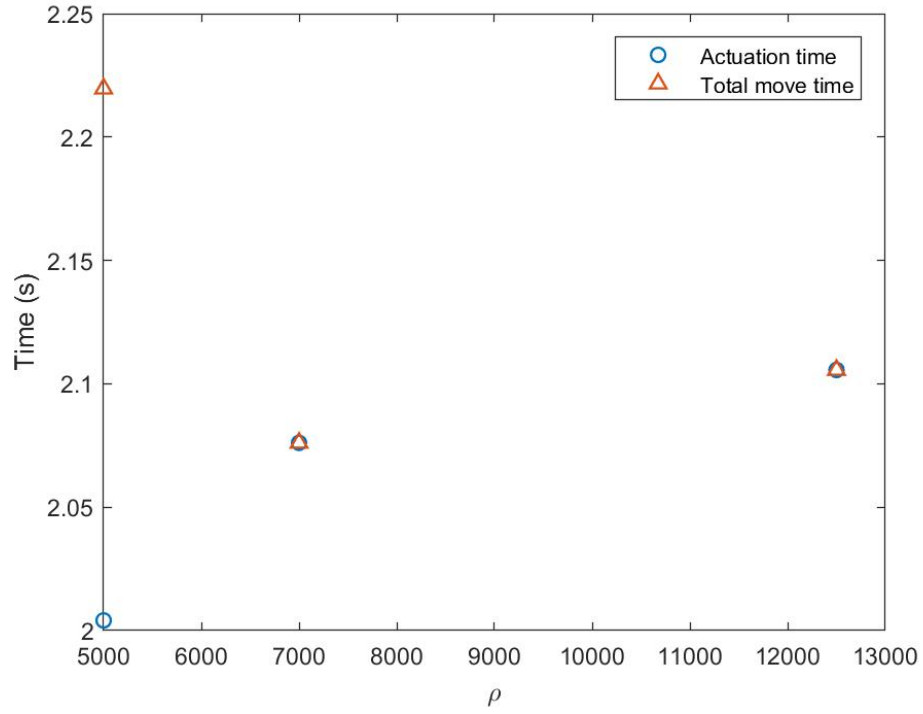

Figure 4.18. Total move time and Actuation time vs ρ : Versine with a_{max}= 60 rad/s^2.

Analysis of the experimental results for a bang-bang input of $a_{max} = 6\text{rad/s}^2$ and actuation time of 2s resulted in a settling time of 2.248s and a peak residual acceleration of 1.2376m/s^2. The results from the bang-bang profile are considered as a benchmark for comparison of the results from the shaped inputs in this work. It can be seen from Figures 4.16 - 4.18 that, with increase in value of the weighting factor ρ, while the actuation time of the input profile increases, the total move time for the two-link robot steadily decreases until the optimal value of ρ is reached and further increase in the weighting factor causes the total move time to increase. Actuation time and settling time together add to give the total move time, so the above plots can be interpreted to say that with increase in ρ, the settling time of the two-link robot decreases and eventually reaches zero. This proves the inputs developed using command shaping with energies removed around the natural frequencies are successful in reducing the residual vibrations in the system.

Tables 4.1 - 4.3 that list the performance metrics for Figures 4.16 – 4.18 show that the peak amplitude of residual vibration ($\ddot{P}_{xy}$) decreases with increase in the value of ρ as it leads to better spectral attenuation. Settling time is calculated from the experimental results based on the pre-determined tolerance of $0.2\mathrm{m/s^2}$ for residual acceleration, but peak residual acceleration is an absolute measure and it is seen from the performance results presented in the above tables that even after the settling time reaches zero, if a higher weighting factor ρ develops an input with better spectral attenuation, the peak residual acceleration decreases accordingly.

Figures 4.19 - 4.35 show the experimental responses obtained by implementation of the shaped inputs discussed and analyzed in this section. The results from the shaped inputs are evaluated against the experimental results for a bang-bang input presented in Figure 4.4. In all the experimental responses presented, the first two sub-plots in the first row represent the desired input acceleration profiles for both motors and the frequency spectrum for the input profile used. Row 2 of the sub-plots represents the input torque profiles for motors 1 and 2. Row 3 of the sub-plots represents the angular positions of the 1st and 2nd links, respectively, with the desired final position plotted in dashed lines. Row 4 of the sub-plots shows the actual and desired angular positions for motors 1 and 2, respectively, plotted on top of each other. The dotted lines again represent the desired angular position signals. The last row of sub-plots represents the real-time angular accelerations of links 1 and 2, respectively.

Figures 4.19 - 4.22 give four cases of experimental responses using the analytical approach for the versine basis function. Figures 4.19 and 4.20 give two analytical results with $a_{max} = 40\ \mathrm{rad/s^2}$ and Figures 4.21 and 4.22 present two cases with $a_{max} = 50\ \mathrm{rad/s^2}$. Figures 4.23 - 4.35 present the experimental responses using the numerical approach. This will be helpful to visualizing the differences in the shaped inputs using the numerical and the analytical approaches.

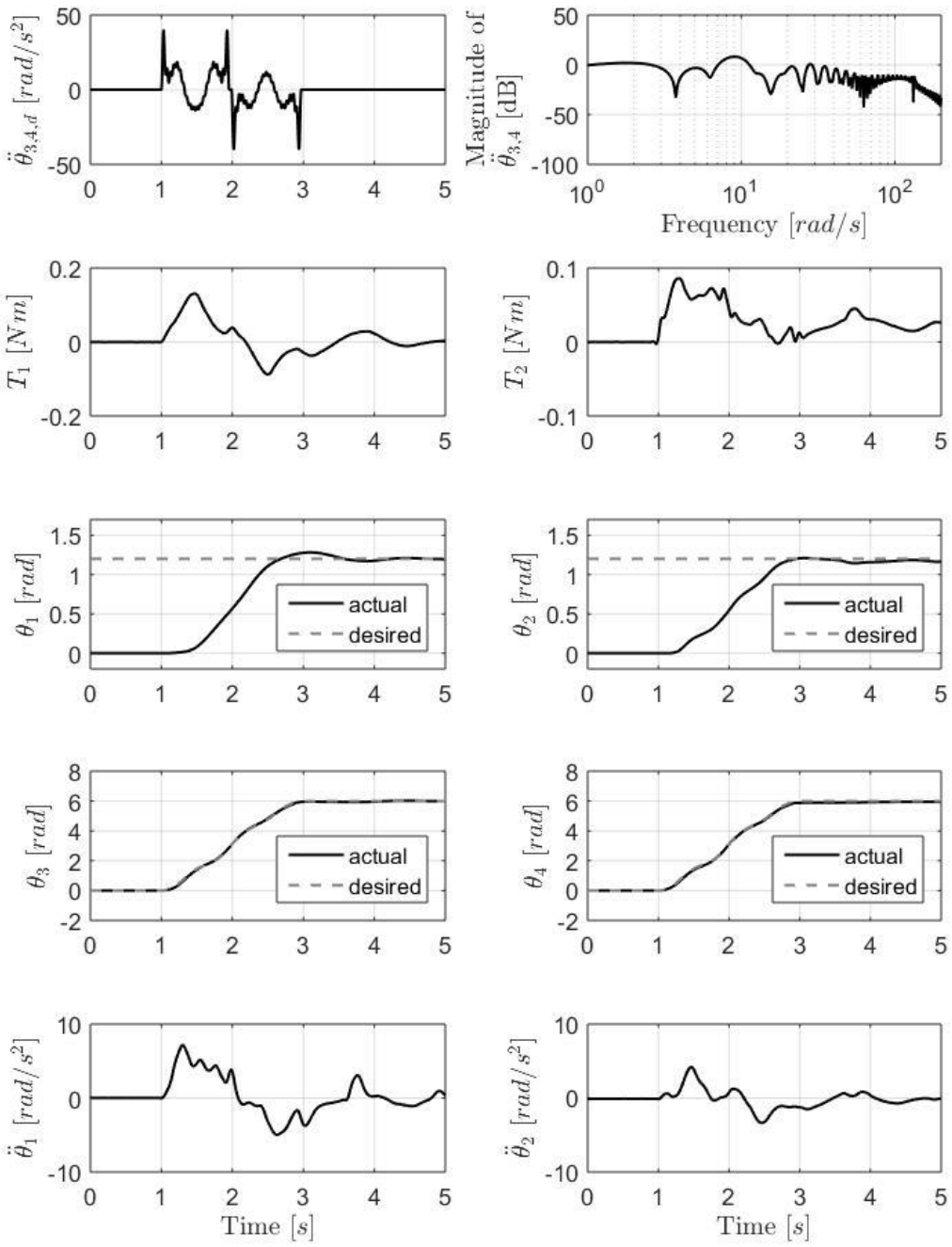

Figure 4.19. Experimental Response for Versine with Analytical Approach for $a_{max} = 40\text{rad/s}^2$ and ρ =10

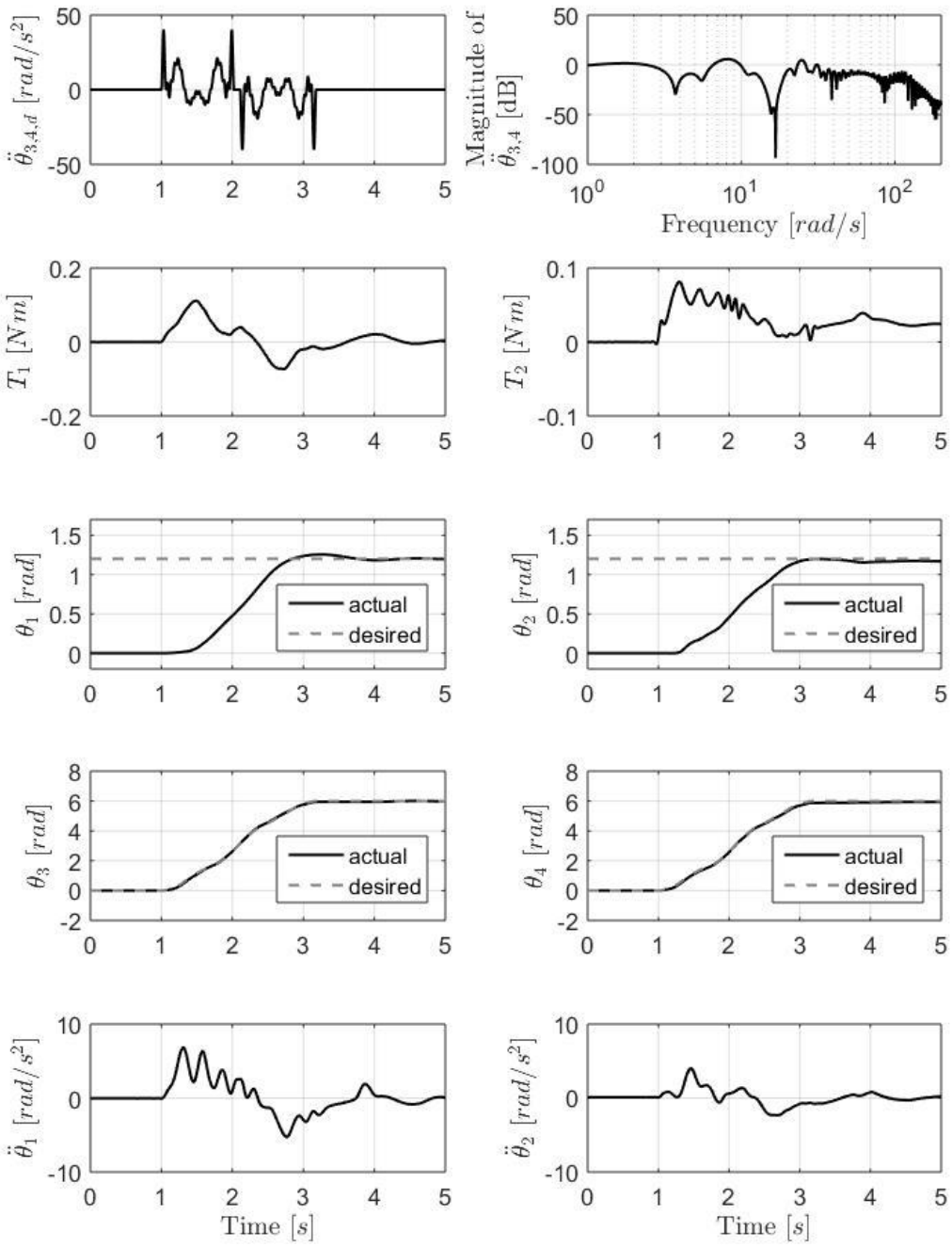

Figure 4.20. Experimental Response for Versine with Analytical Approach for a_{max}=40rad/s^2 and ρ =75

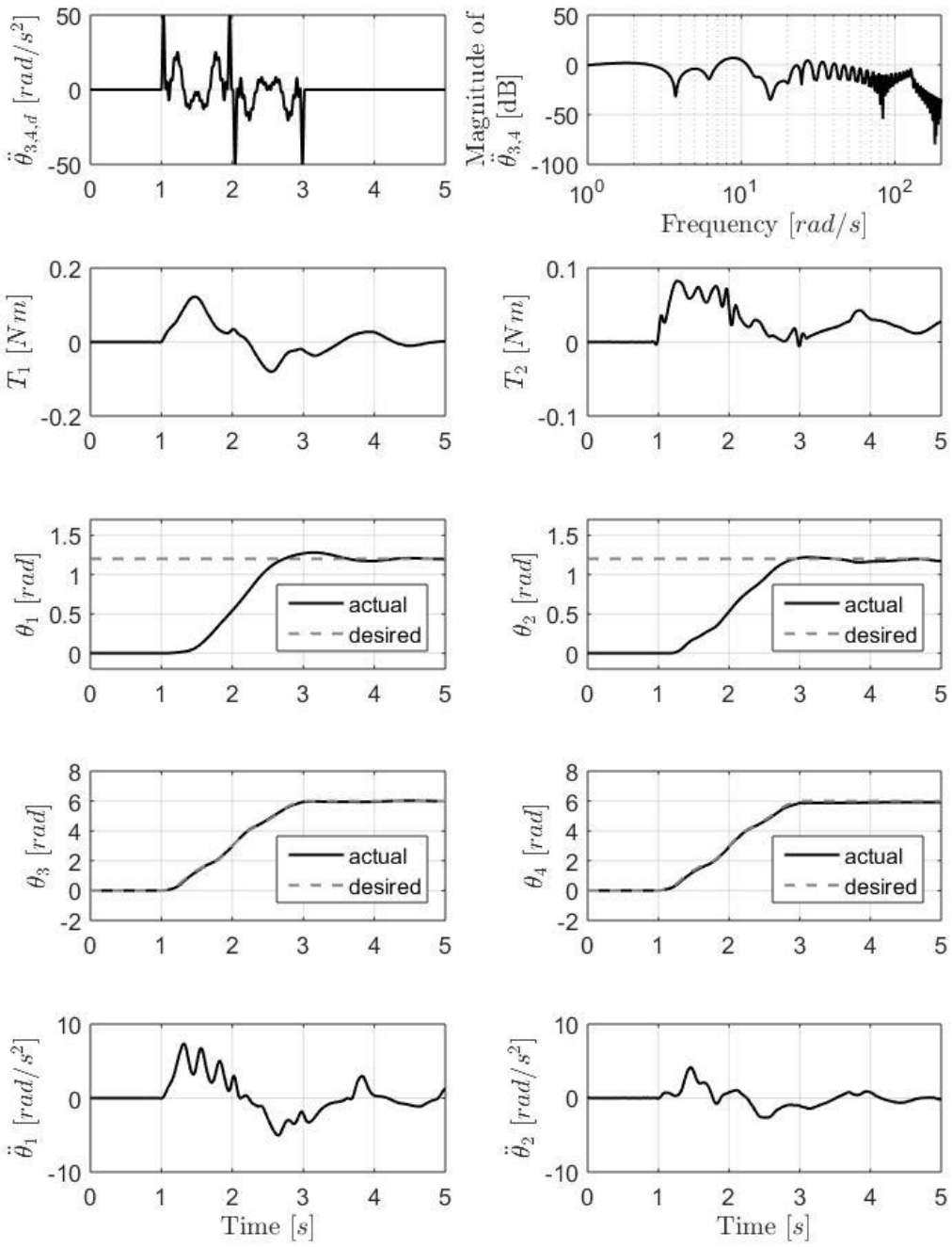

Figure 4.21. Experimental Response for Versine with Analytical Approach for a_{max}=50rad/s^2 and ρ =50

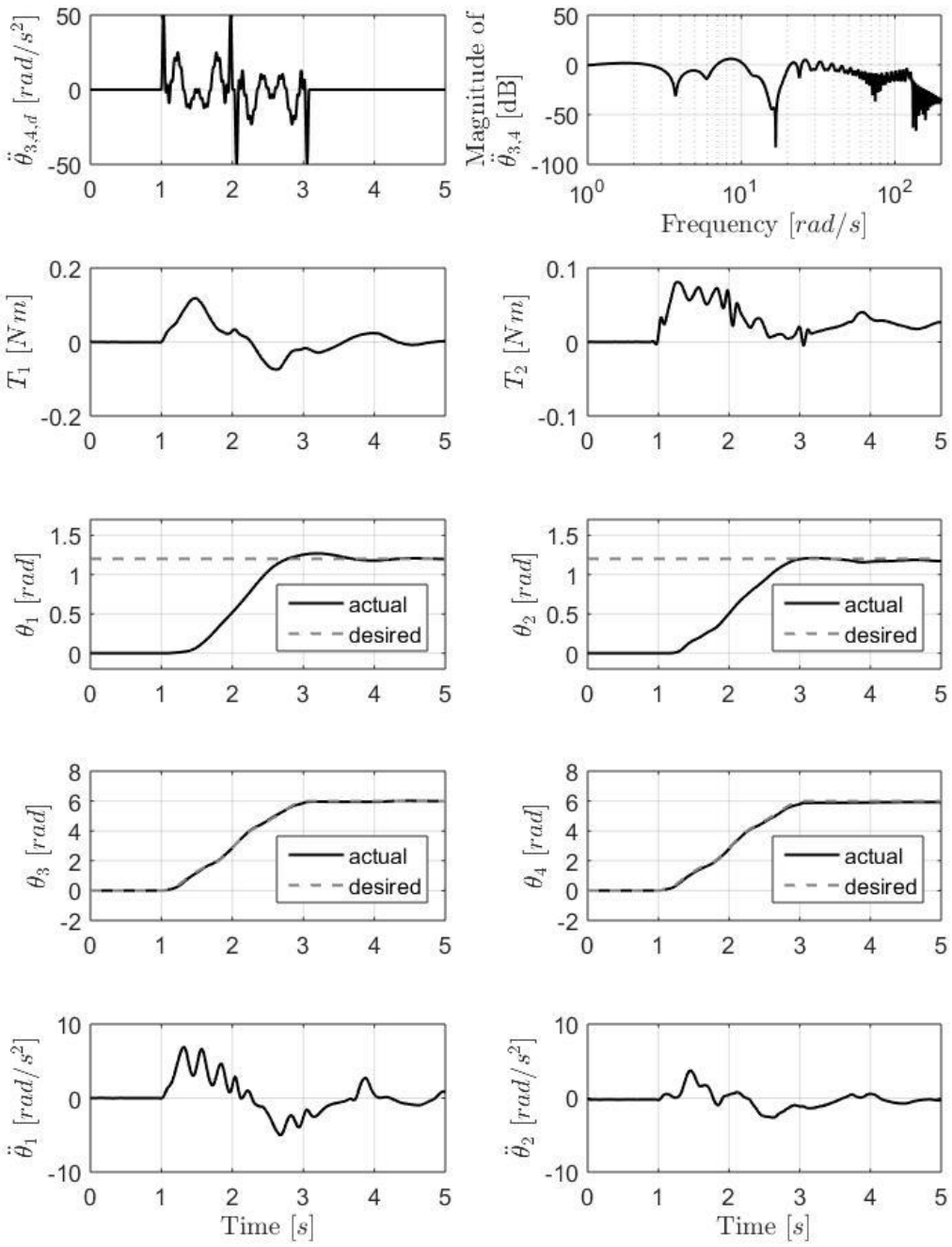

Figure 4.22. Experimental Response for Versine with Analytical Approach for a_{max}=50rad/s^2 and ρ =100

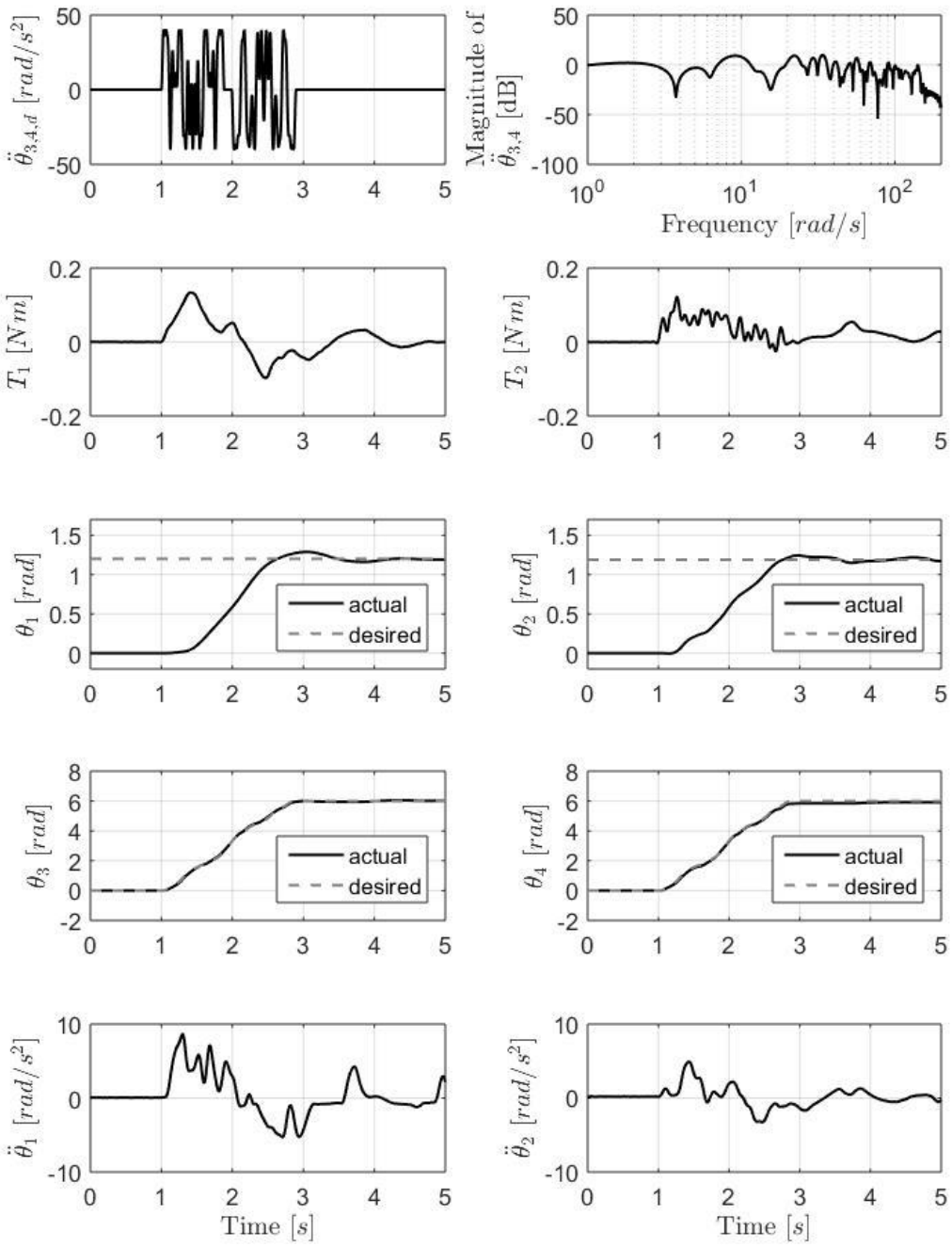

Figure 4.23. Experimental Response for Versine with Numerical Approach for a_{max}=40rad/s^2 and ρ =700

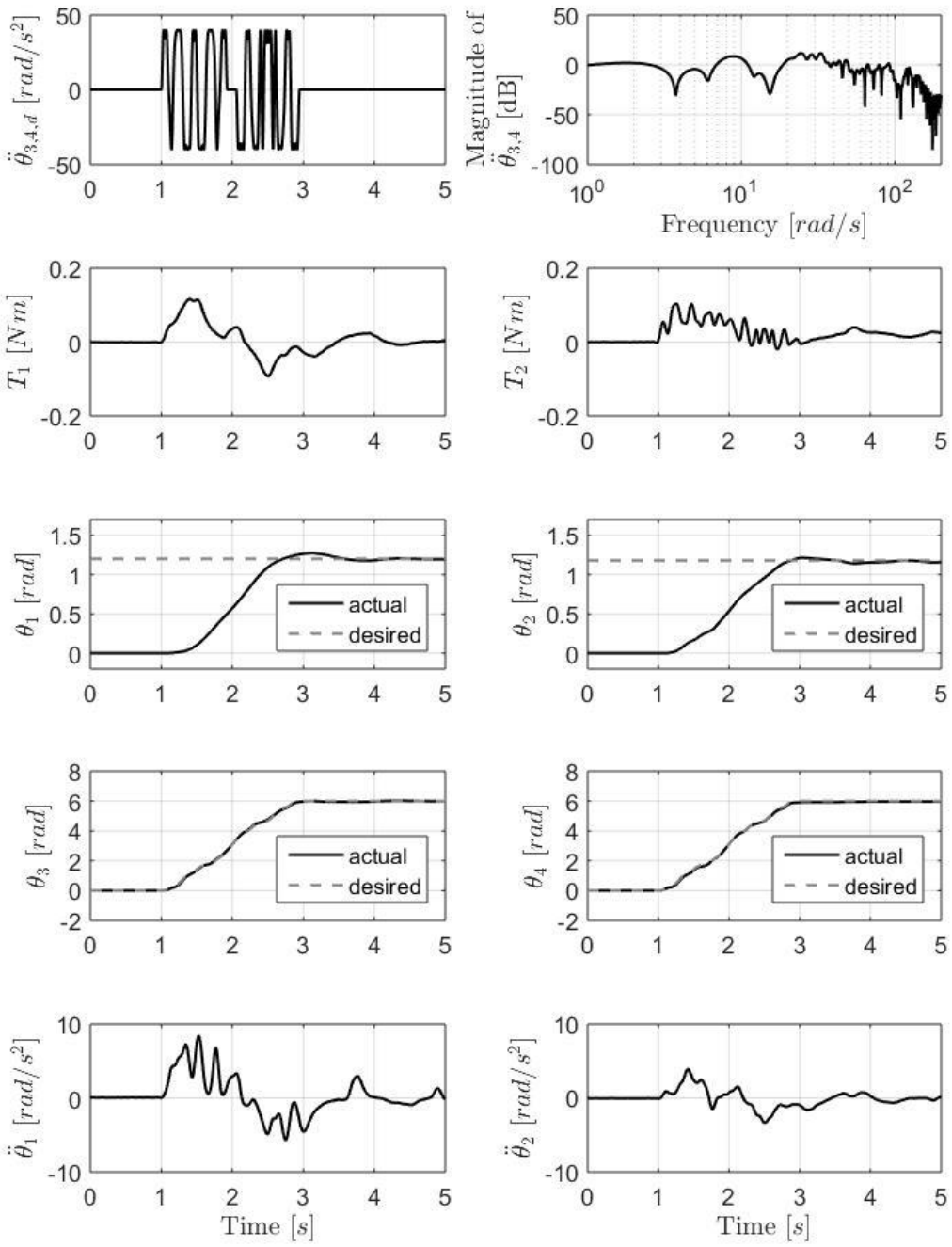

Figure 4.24. Experimental Response for Versine with Numerical Approach for a_{max}=40rad/s^2 and ρ =1500

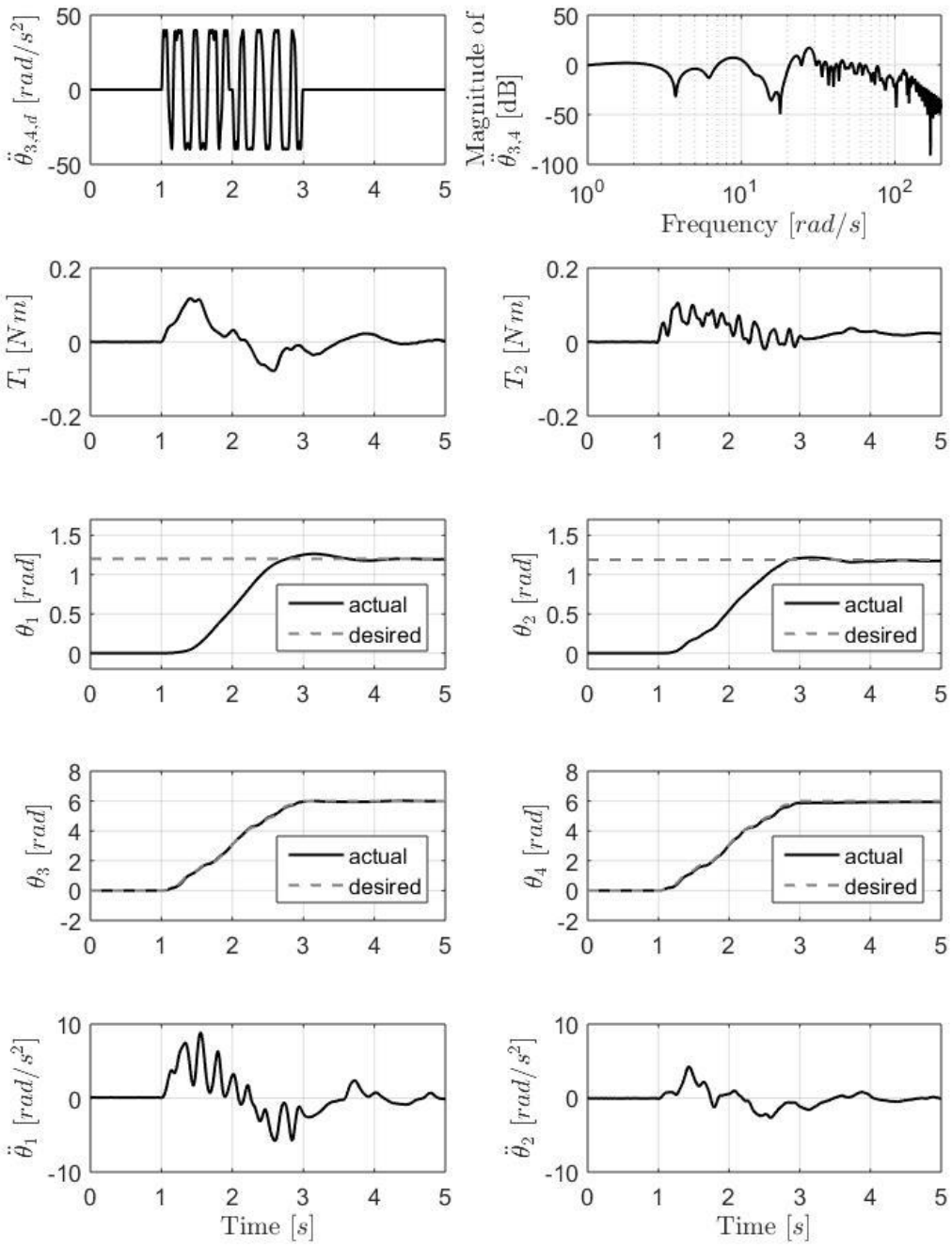

Figure 4.25. Experimental Response for Versine with Numerical Approach for a_{max}=40rad/s^2 and ρ =2500

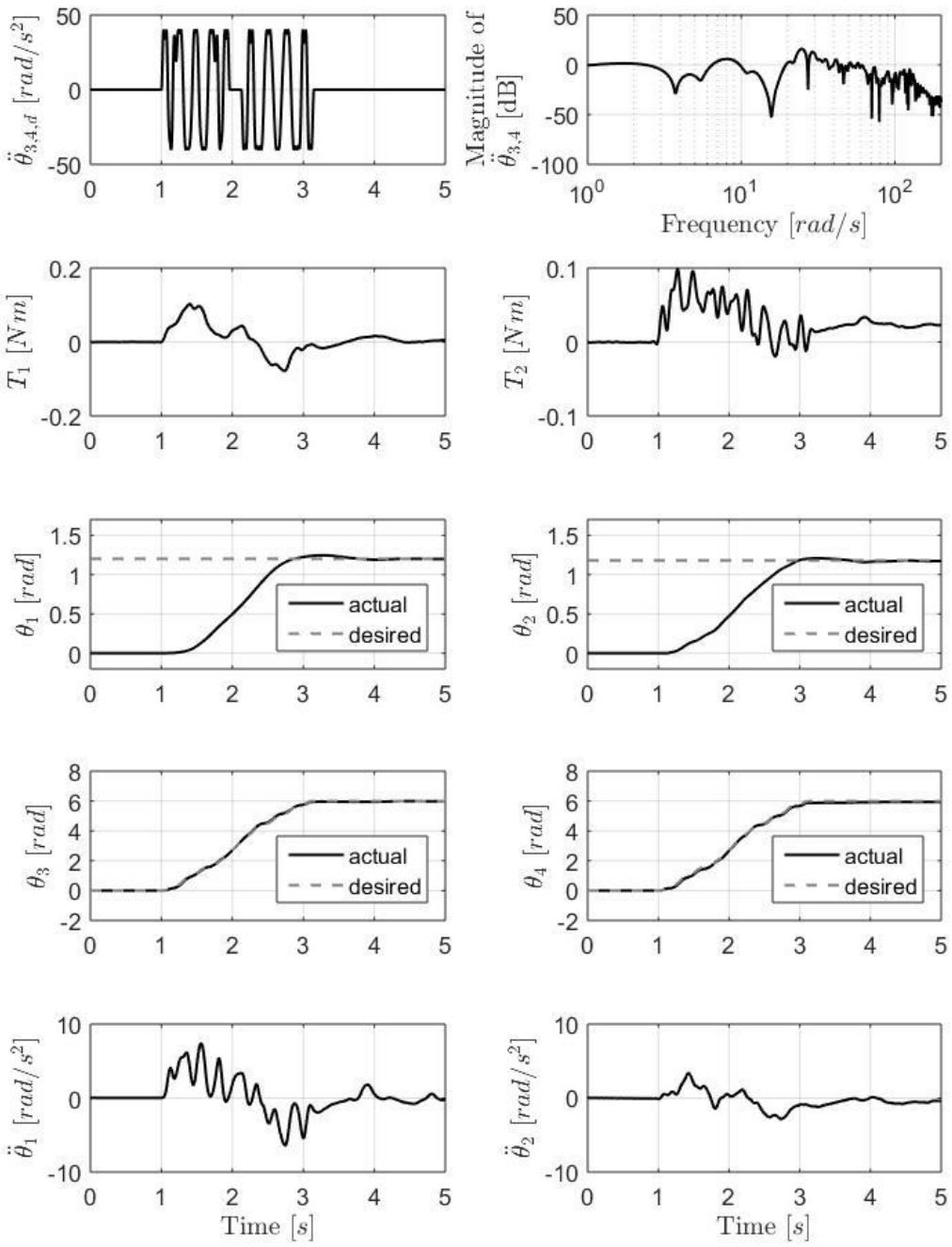

Figure 4.26. Experimental Response for Versine with Numerical Approach for a_{max}=40rad/s^2 and ρ =8750

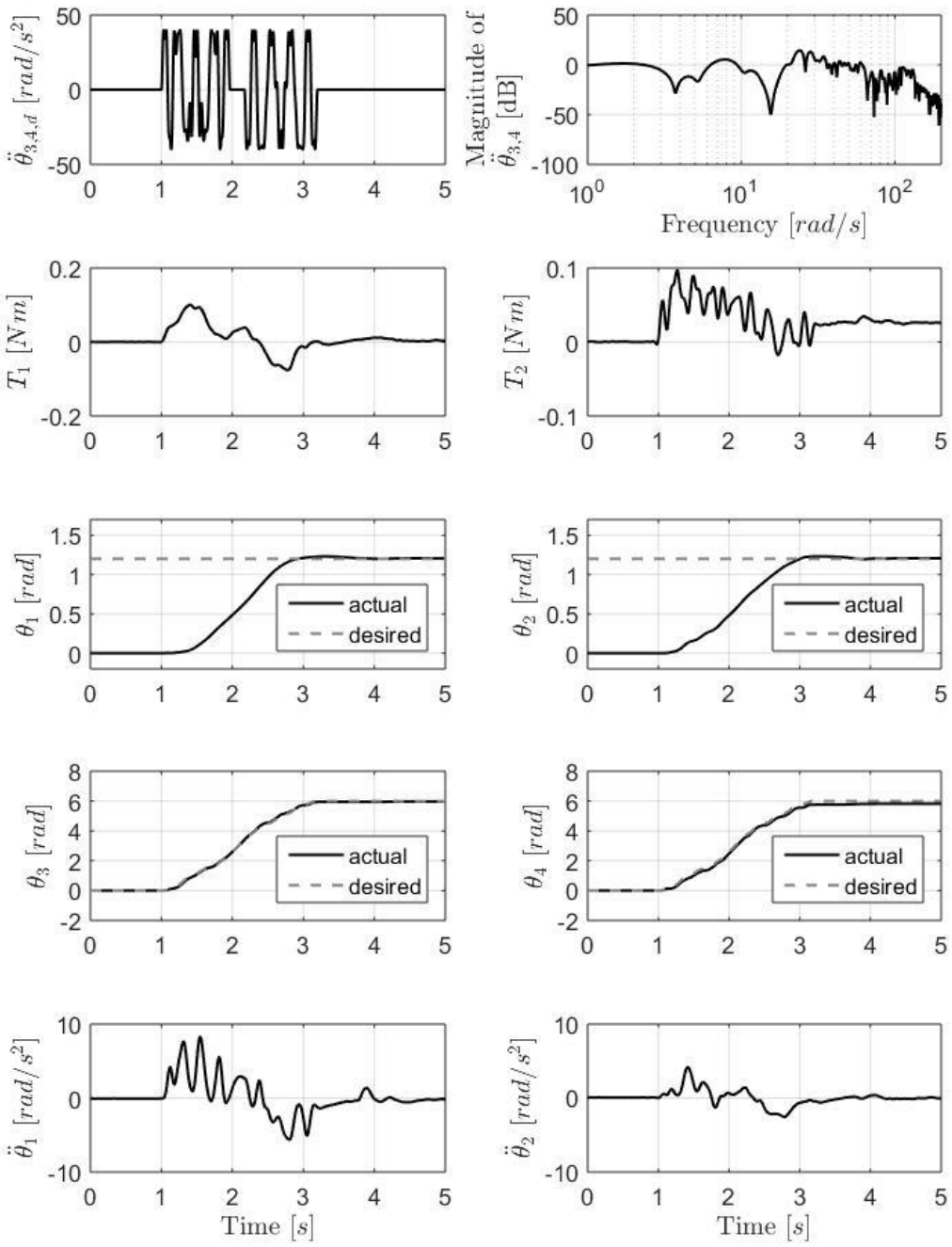

Figure 4.27. Experimental Response for Versine with Numerical Approach for a_{max}=40rad/s^2 and ρ =20,000

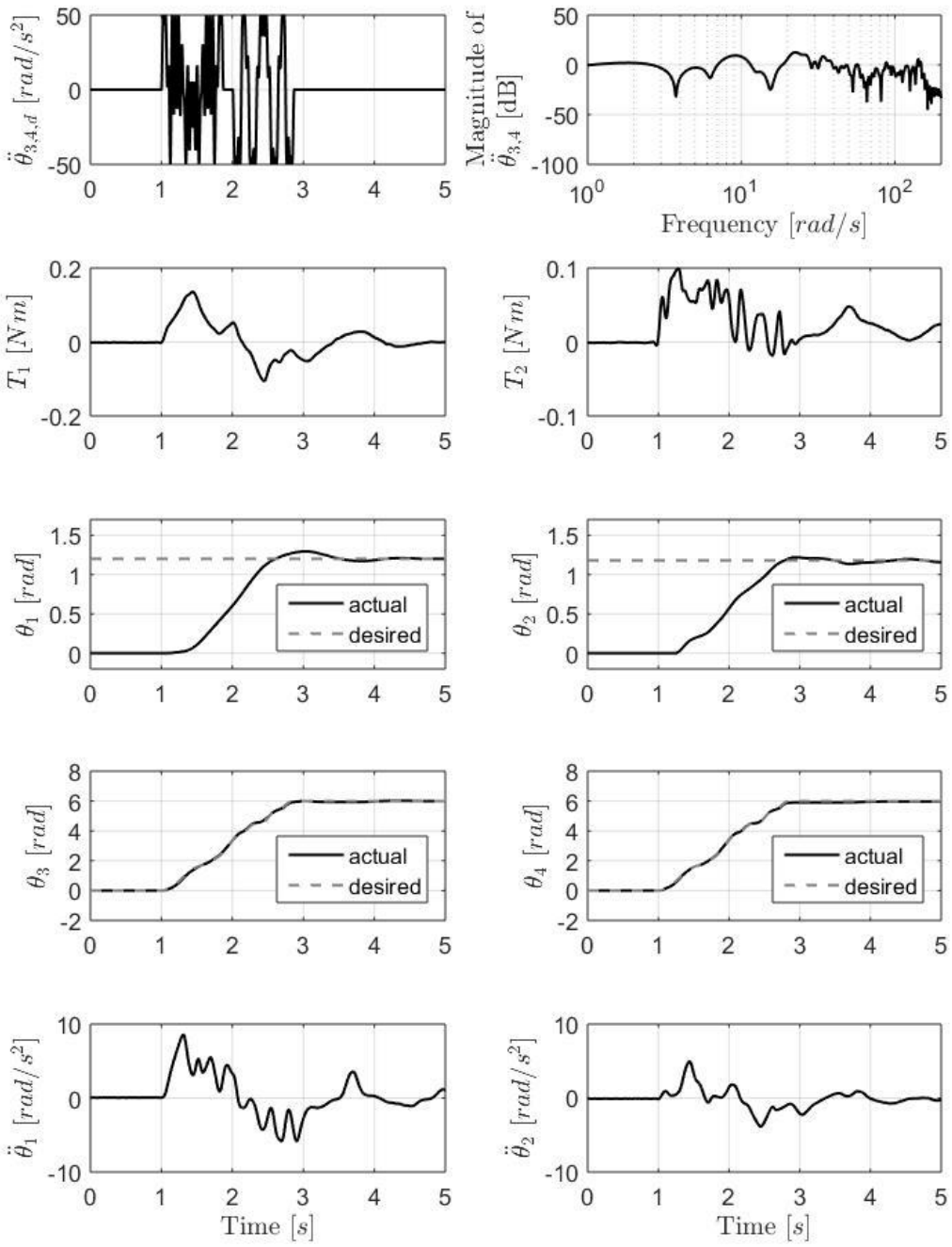

Figure 4.28. Experimental Response for Versine with Numerical Approach for a_{max}=50rad/s^2 and ρ = 1500

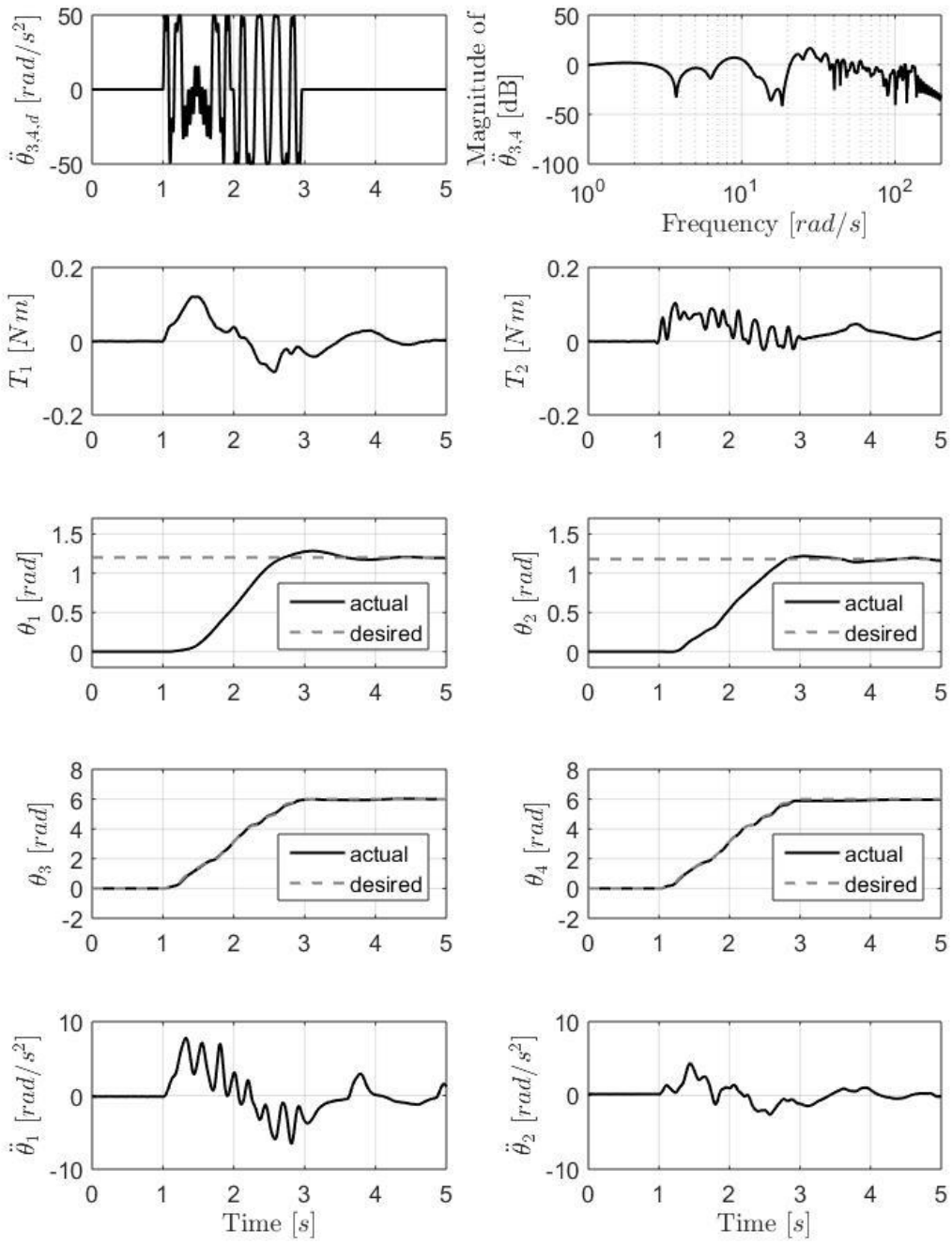

Figure 4.29. Experimental Response for Versine with Numerical Approach for a_{max}=50rad/s^2 and ρ = 2500

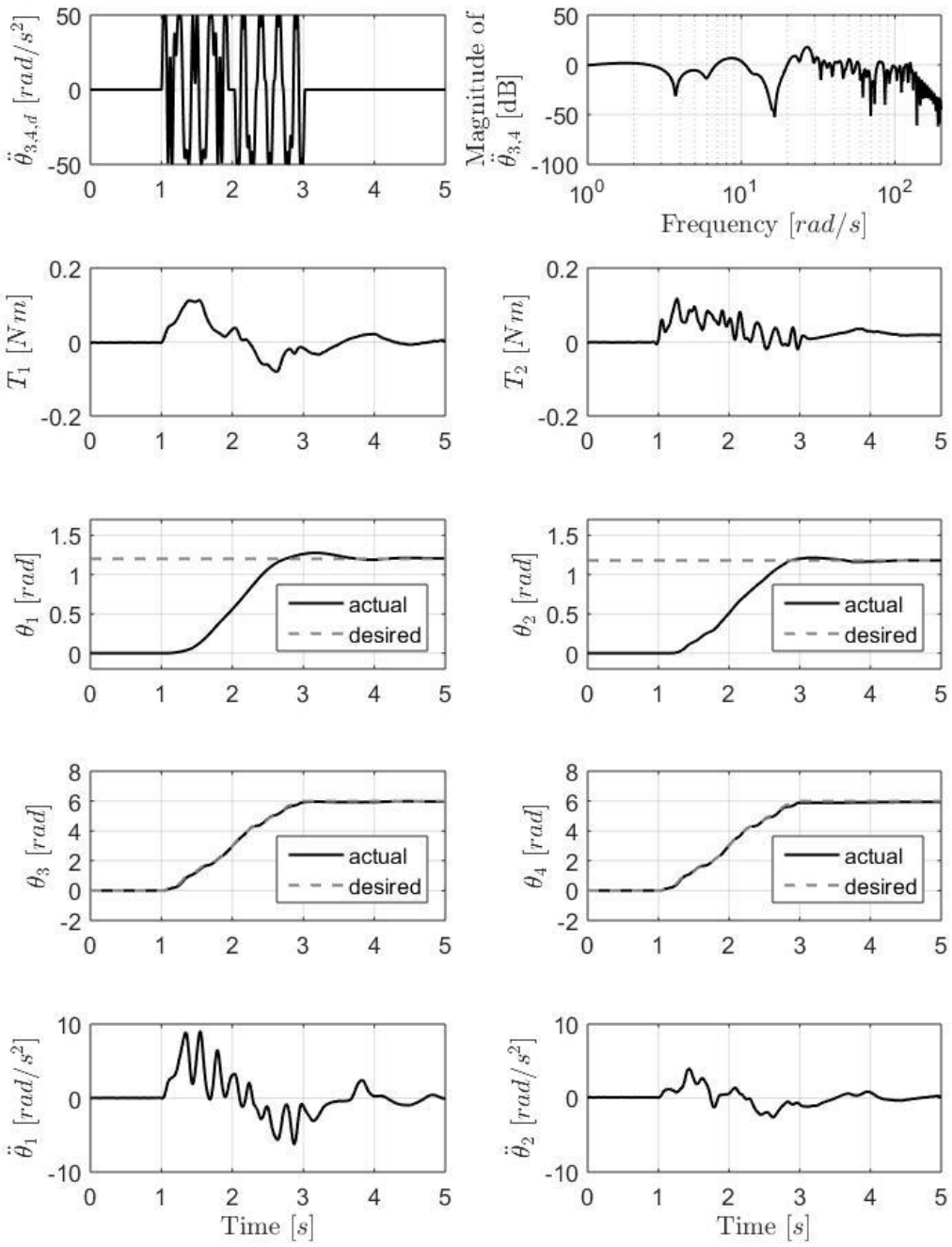

Figure 4.30. Experimental Response for Versine with Numerical Approach for a_{max}=50rad/s^2 and ρ = 5000

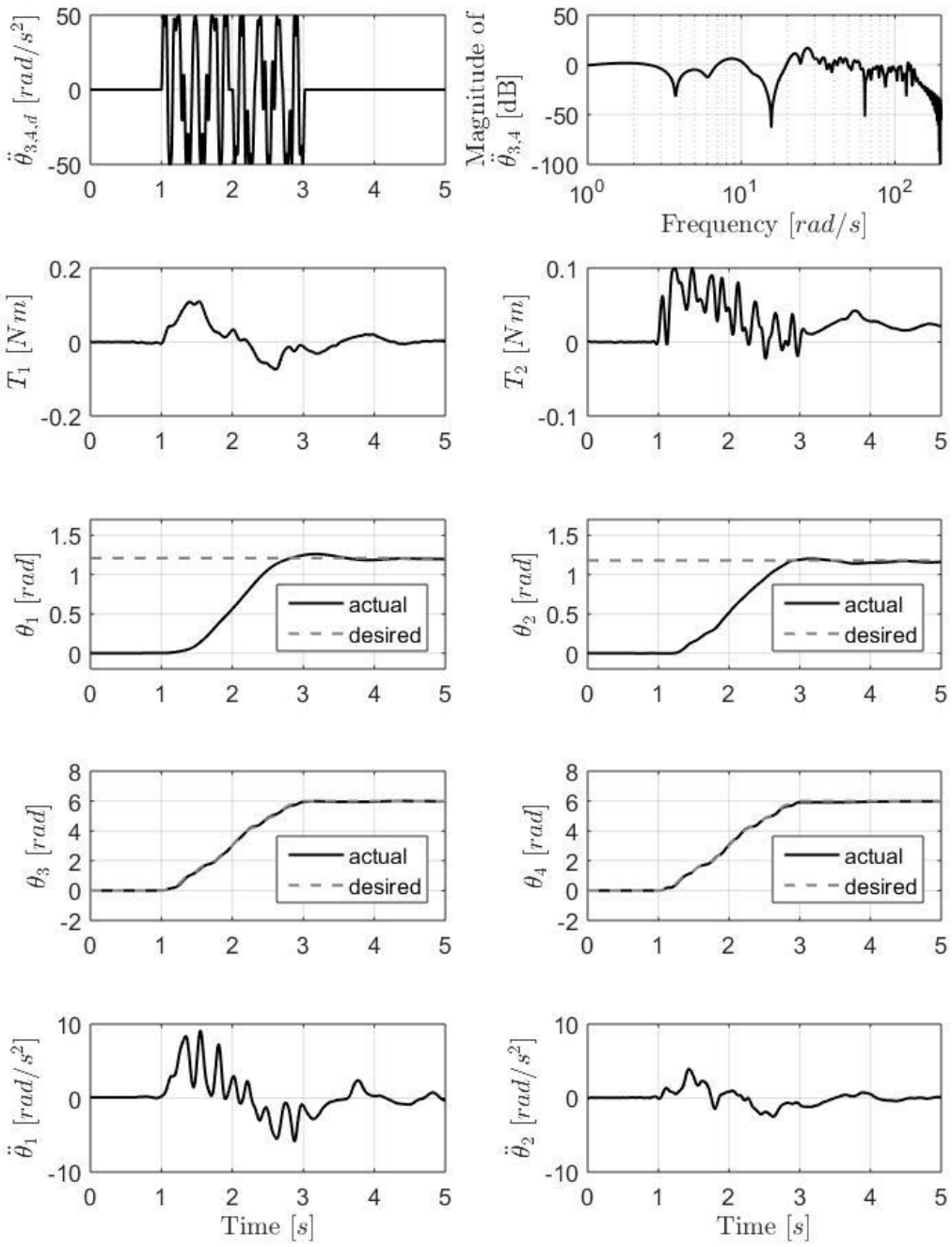

Figure 4.31. Experimental Response for Versine with Numerical Approach for a_{max}=50rad/s^2 and ρ = 10,000

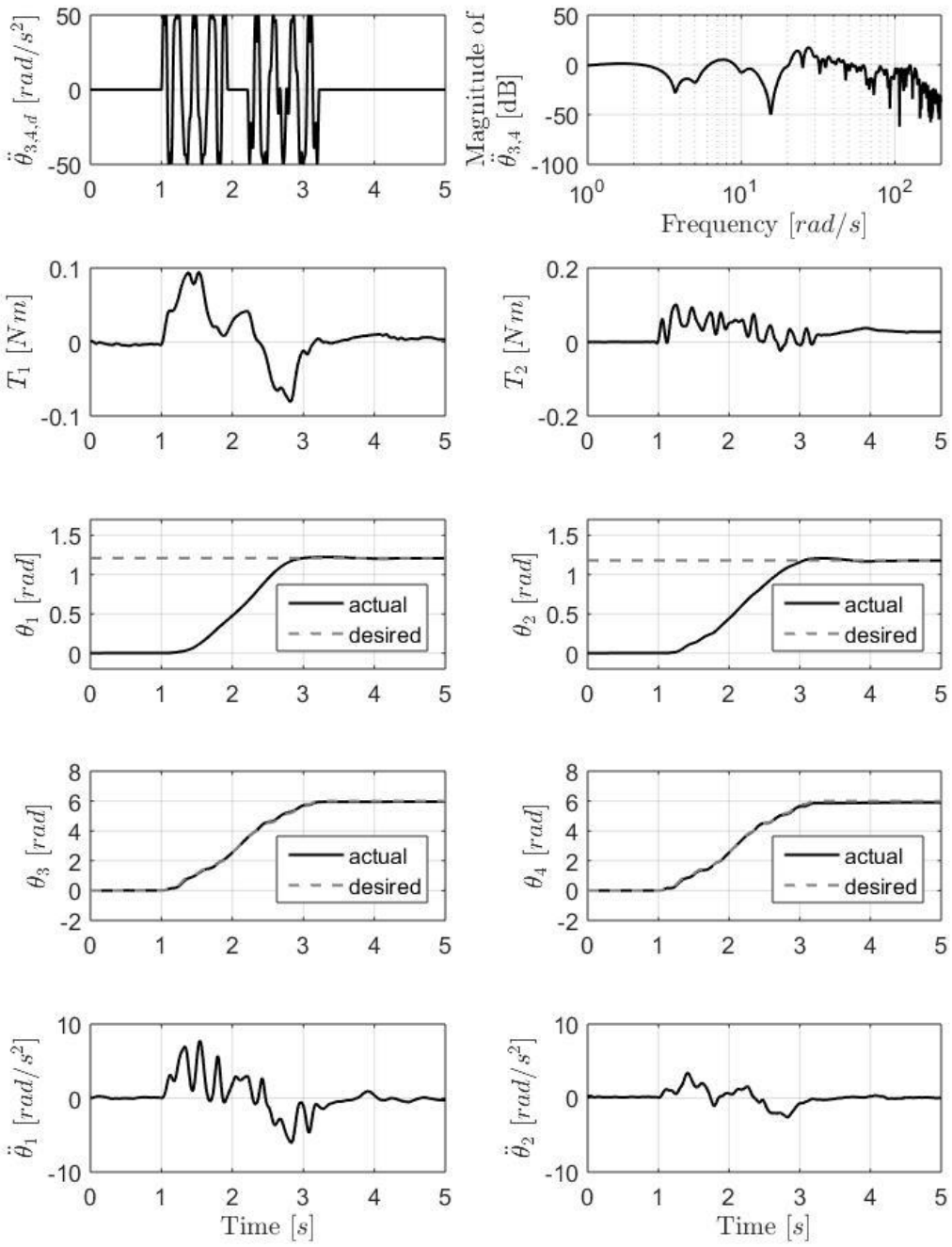

Figure 4.32. Experimental Response for Versine with Numerical Approach for a_{max}=50rad/s^2 and ρ = 30,000

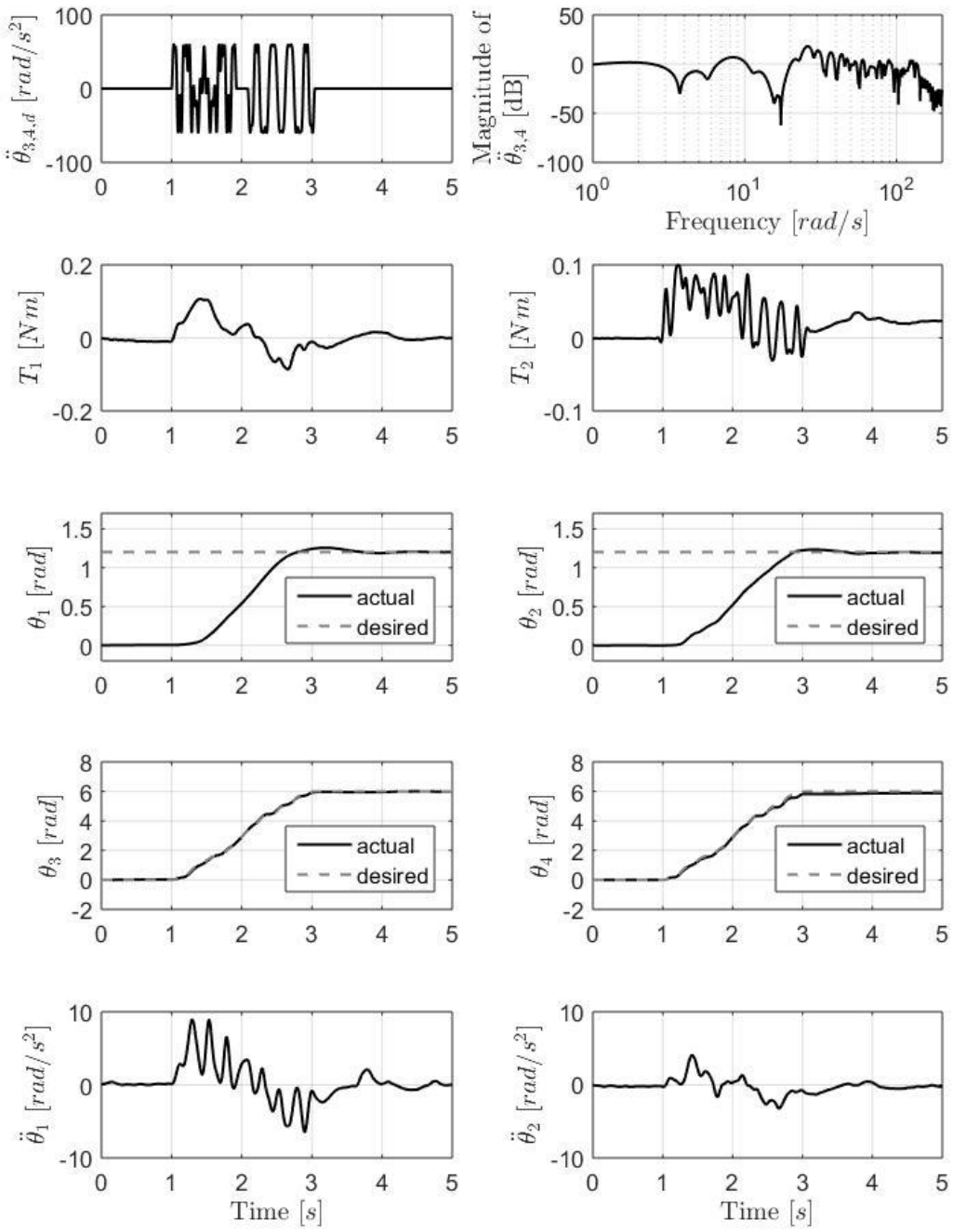

Figure 4.33. Experimental Response for Versine with Numerical Approach for a_{max}=60rad/s^2 and ρ = 5000

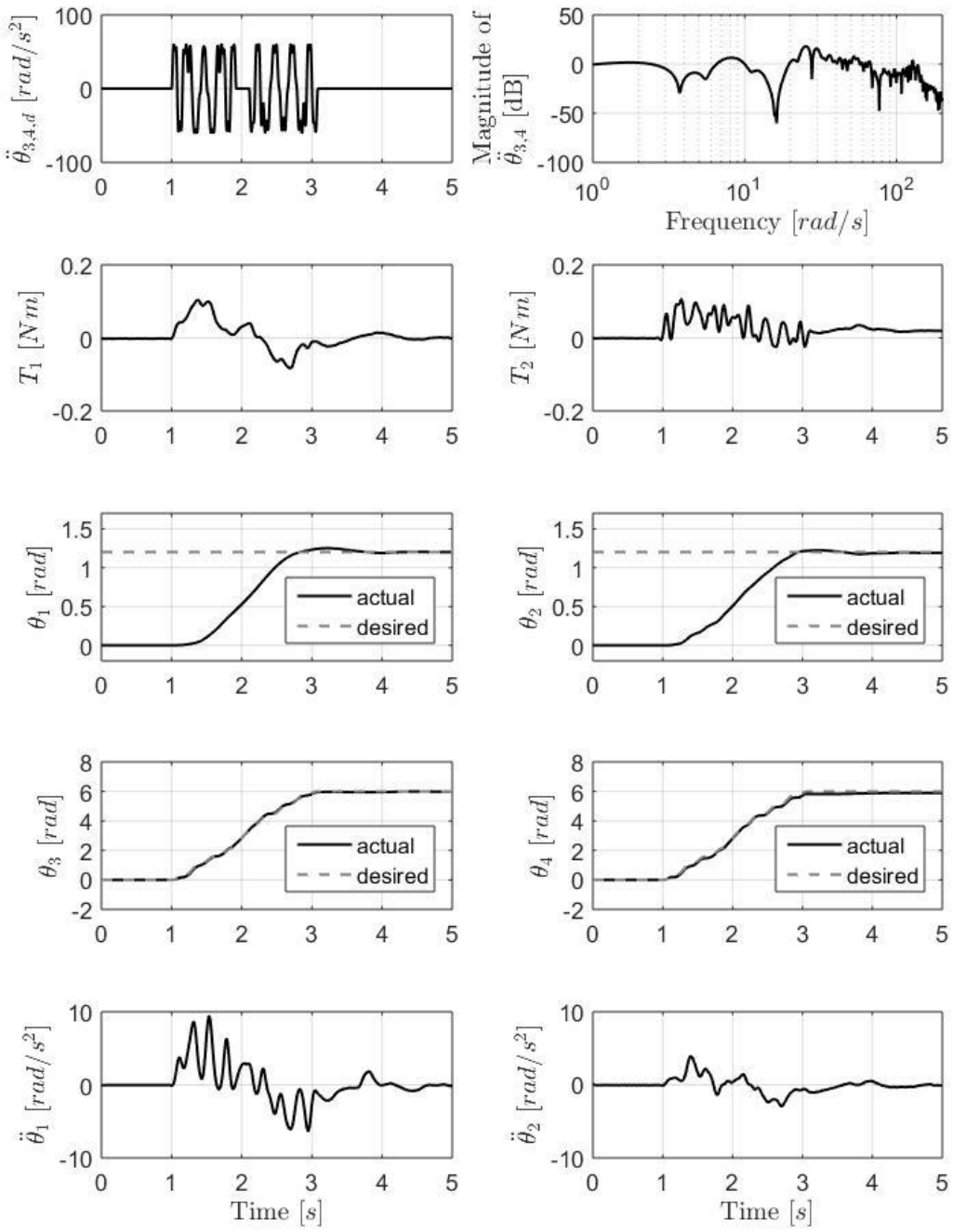

Figure 4.34. Experimental Response for Versine with Numerical Approach for a_{max}=60rad/s^2 and ρ = 7500

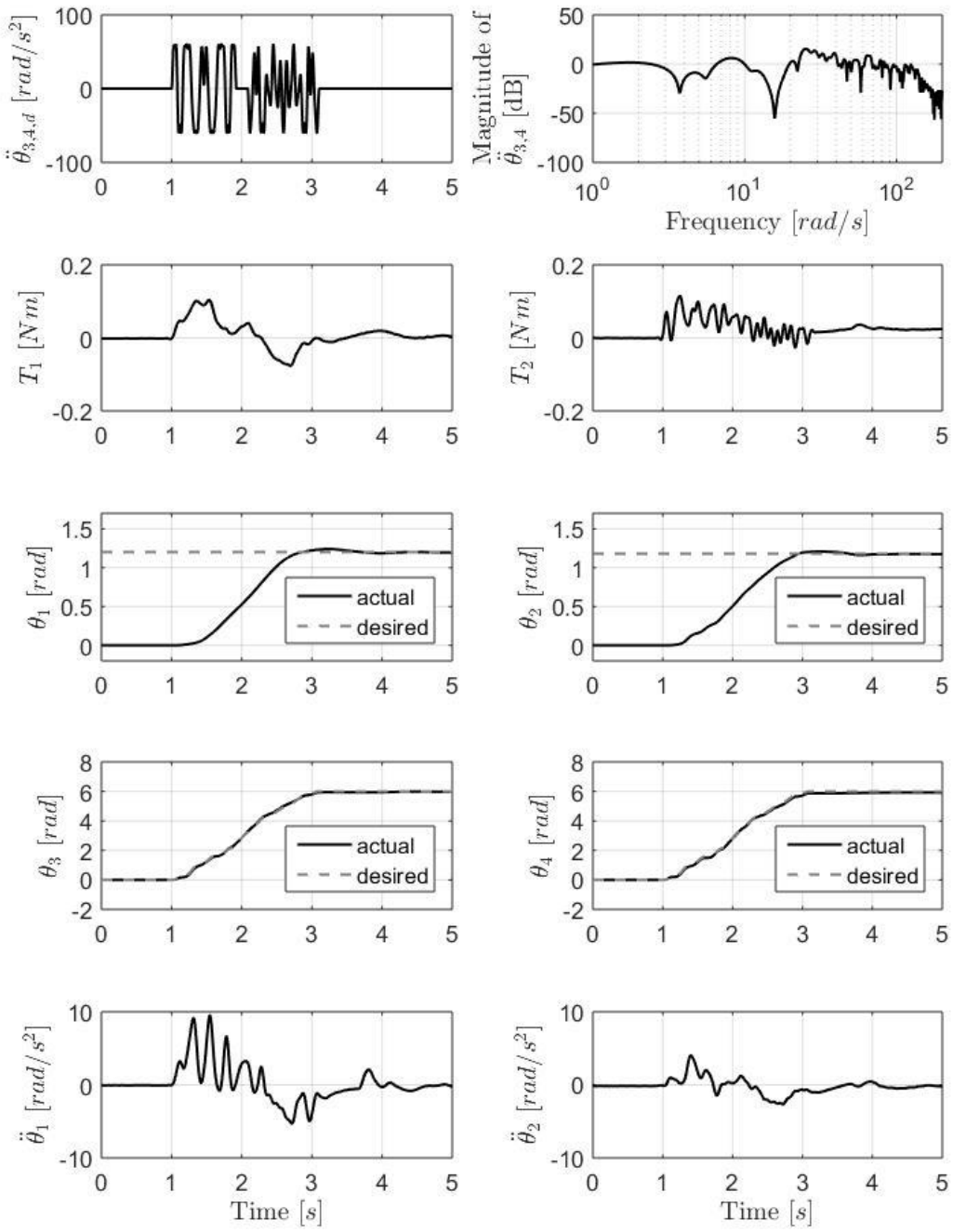

Figure 4.35. Experimental Response for Versine with Numerical Approach for a_{max}=60rad/s^2 and ρ = 12,500

The spectral plots in the experimental responses show a clear decrease in the spectral energy in the windows of natural frequencies selected with increase in weighting factor ρ. The experimental results, in line with the results from the simulation, show a decrease in the oscillation of the final positions of links 1 and 2 as the spectral attenuation in the shaped inputs increases. By relating the data presented in Tables 4.1 - 4.3 with the experimental plots in Figures 4.23 – 4.35, and comparing them, it can be observed that the peak residual acceleration of the two links continues to decrease with increase in spectral attenuation even after settling time achieves a zero value.

The input profiles under the analytical approach achieve motion at peak acceleration only for a short duration of the actuation time and complete the rest of the motion at intermediate acceleration values. In contrast, input profiles developed using the numerical optimization, while very active in switching between acceleration and deceleration, complete most of the motion under the peak acceleration condition. This can be another possible explanation why the numerically-shaped inputs have better actuation time.

Figures 4.36 – 4.38 show the generated torque signals for the shaped inputs with peak accelerations of 40 rad/s^2, 50 rad/s^2, and 60 rad/s^2, respectively. Comparing these responses, it was noticed that while the input accelerations increased by 150%, input torque to the motors does not reflect a proportional change. Specifically, for input profiles using the numerical approach which switch between peak acceleration and deceleration multiple times, it was expected that the torque profile would show similar changes when input acceleration values increase, but from the experimental results, the torques do not vary much. The torque signals to the motors are calculated in real time by the computed torque controller and warrant further study and research into the relationship between the generated torque signals and the computed torque controller.

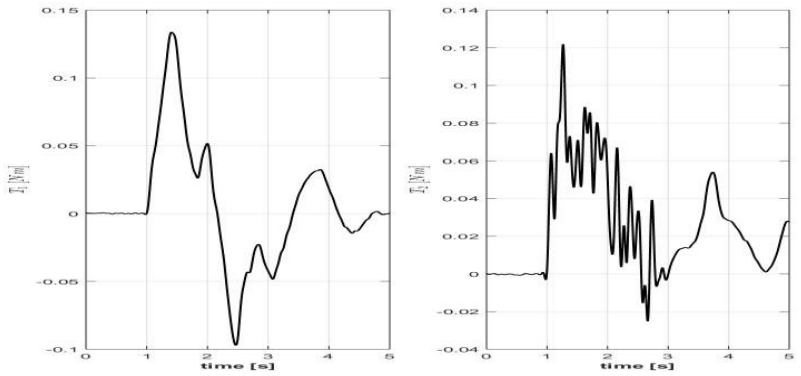

Figure 4.36. Torque signals for Shaped Input Commands: $a_{\max} = 40$ rad/s^2.

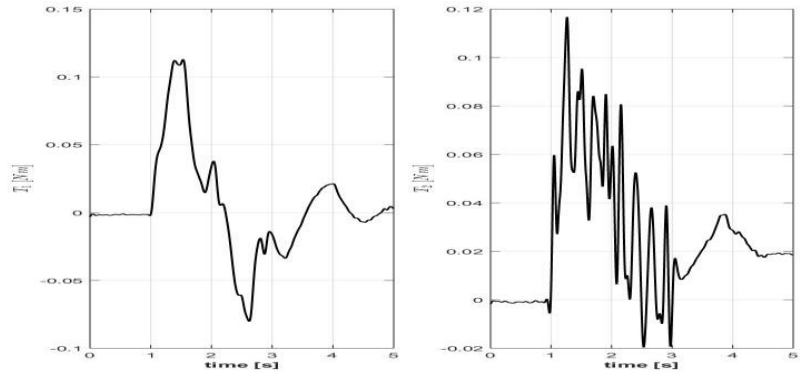

Figure 4.37. Torque signals for Shaped Input Commands: $a_{\max} = 50$ rad/s^2.

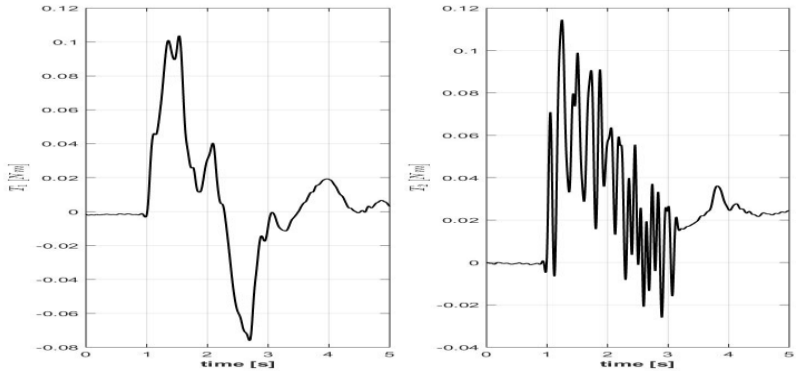

Figure 4.38. Torque signals for Shaped Input Commands: $a_{\max} = 60$ rad/s^2.

CHAPTER 5. CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary and Conclusions

The command shaping method with numerical optimization and constrained peak input acceleration has been implemented separately and studied in previous works. In this work, the command shaping technique with numerical optimization has been extended to incorporate constrained peak input acceleration in the development of the inputs using a segmented versine basis function to minimize residual vibrations. An underdamped two-link robot with flexible joints has been used as the experimental setup for validation of the command shaping approach. The command shaping approach using the original L_2 – norm formulation is susceptible to the Gibbs phenomenon while approximating discontinuous functions such as a time-optimal bang-bang profile. Numerical optimization using L_1- norm was successful in eliminating the detrimental effects of Gibbs phenomenon on the shaped inputs, resulting in better approximations. Constrained peak acceleration was proposed for the closed-form analytical formulation to account for the motor torque limitations of the physical system instead of the original fixed actuation time approach. This new approach uses input acceleration and weighting factor as the design variables and iterates for the actuation time. The merits of these approaches have been combined by developing command shaping with numerical optimization and peak input acceleration constraint to get closer approximations of the bang-bang profile with the ability to account for limitations of the motor torques.

Command shaping is tested in simulations and experiments by implementing the shaped inputs that carry reduced energy around the selected frequencies on a two-link flexible-joint robot. The analysis of the results confirmed that shaped inputs significantly improve residual vibration

performance when evaluated against a time-optimal bang-bang profile, but this is achieved at the cost of longer actuation times. The theoretical formulation indicates that the magnitude of spectral attenuation depends on the parameter ρ. Based on simulation results, the effects of the weighting factor ρ on the magnitude of spectral attenuation has been studied and comparison of the results from the numerical and analytical approaches shows similar levels of spectral attenuation being achieved in both approaches for the versine basis function. This indicates that switching from the analytical approach to the numerical approach does not affect the maximum spectral attenuation achieved for the basis function.

Input profiles of the shaped commands and the simulation results conclusively show an increase in the approximation error between the shaped inputs and bang-bang function with increasing values of ρ, leading to longer actuation times, but comparison between the numerical and the analytical approach shows that the shaped inputs developed using the numerical approach have better actuation time as compared to inputs using the analytical approach because they more closely approximate the bang-bang function. Inputs developed using the numerical approach are very active and switch multiple times between peak acceleration and peak deceleration values while inputs developed using the analytical approach are relatively settled. This needs to be further explored to clearly identify if this is a characteristic of the numerical approach.

From the simulation results, the trade-off between spectral magnitude at the resonant modes with the total move time for the shaped inputs has been analyzed and it shows, despite the increase in actuation time, the total move time decreases as the energy around the natural frequencies is minimized. This improvement in total move time can be observed only up to a value of ρ. Beyond this value of ρ, the error of fit to the bang-bang approximation starts increasing without

improvement in the spectral attenuation, resulting in input profiles that are suboptimal with respect to both objectives.

After demonstrating the effectiveness of the command shaping method in simulation, the developed inputs were applied in experiments and validated. The experimental results obtained were in line with the simulation results and showed improved total move time and settling time, as the weighting factor and the spectral attenuation increases. Experimental results also confirmed that an optimal value of ρ exists for each input such that the settling time is zero; beyond this value of ρ, the developed input leads to suboptimal performance. It can also be seen that the peak residual accelerations of the links continue to decrease with higher spectral attenuation even after the settling time of zero is reached. This indicates that the command shaping approach has scope for further developing inputs with better residual vibration performance under tighter requirements of settling time. This work proves the success of the command shaping method with numerical optimization and constrained peak input acceleration in reducing the residual vibrations in a two-link flexible-joint robot.

5.2 Contributions

The primary contribution of this work is the development of a numerical optimization framework with a relaxed actuation time and constrained peak input acceleration for the segmented versine function. The existing numerical solver on MATLAB to calculate the coefficients of the harmonics was modified for optimal performance of the ramped sinusoid basis function by making the following changes:

1. MATLAB solver `fmincon` uses coefficients B_l as design parameters and allows the user to specify the initial guess for them. The time penalty Γ is calculated based on B_l for the

first iteration. B_l and Γ are both updated at the end of each iteration and these values are used for further calculations. The original code used Γ as a design parameter, while the value of coefficients B_l was fixed for all the iterations.

2. Nonlinear constraints are defined for `fmincon` through the function `myCon` to ensure that the time penalty (Γ) at the minimum of the objective function converges to a value greater than 1.
3. The errors in the definition of the objective function and the magnitude of the forcing function at the 22 pre-determined frequencies have been corrected. The objective function `fun` has been changed such that the value of time penalty (Γ) is updated for each individual minimization iteration of `fmincon`.

The updates to the numerical solver resulted in better convergence of the multi-objective function over a larger range of design parameters. The updated framework allows greater flexibility to the user to make additional linear or nonlinear constraints to the optimization process based on requirements without changing the basic setup significantly. Improvement in the robustness to variation in design parameters and improved computation time were observed with the updated numerical solver over the previous version.

Further, the effect of flexible mode energy attenuation on the actuation time and the overall move time for the two-link robot has been studied and it was concluded that as the flexible mode energy decreases the actuation time increases but the total move time decreases. A method of selecting ρ, for a given peak input acceleration, to achieve the optimal performance of the shaped input has been presented. The results from the numerical formulation have been compared against the analytical results to show the improvements in the actuation time for the numerical approach for the same magnitude of spectral attenuation.

Apart from the contributions listed above, the study resulted in identifying simulation and experimental behavior in the two-link robot that warrant further investigation and a deeper understanding. The generated motor torque signals not being proportional to the peak accelerations of the input signals indicate that the behavior of the computed torque controller needs to be further explored. Also, the effects of parameter mismatch in simulation between the computed torque controller and the robot model and its influence on the residual vibration performance require further study.

5.3 Recommendations for Future Work

In the current study, both natural frequencies of the robot are weighted equally in the multi-objective optimization function using a single weighting factor ρ. Figure 5.1 gives a comparison of frequency attenuation at both modes as a function of weighting factor ρ. It suggests that as ρ increases, the first natural frequency carries relatively higher energy than the second frequency. It would be beneficial to develop an algorithm that weighs both natural frequencies differently and analyze the influence of weighting factors on the spectral attenuation and residual vibration performance in the two-link robot.

Numerical optimization using segmented versine requires three nested iteration loops for the numerical solver, actuation time convergence and the peak velocity calculations. In that, the numerical solver and the actuation time convergence loops would have to run multiple times depending on the number of segments in the input profile. While the new setup of the numerical solver is faster, more reliable, and can work with a higher range of design variables, it still requires a significantly higher computation time when compared against the analytical approach. Also, the numerical solver for very high values of ρ tends to converge to a local minimum at times, resulting

in input profiles with suboptimal performance. Therefore, further exploration with different numerical solvers and optimization algorithms needs to be done to reduce the computational time.

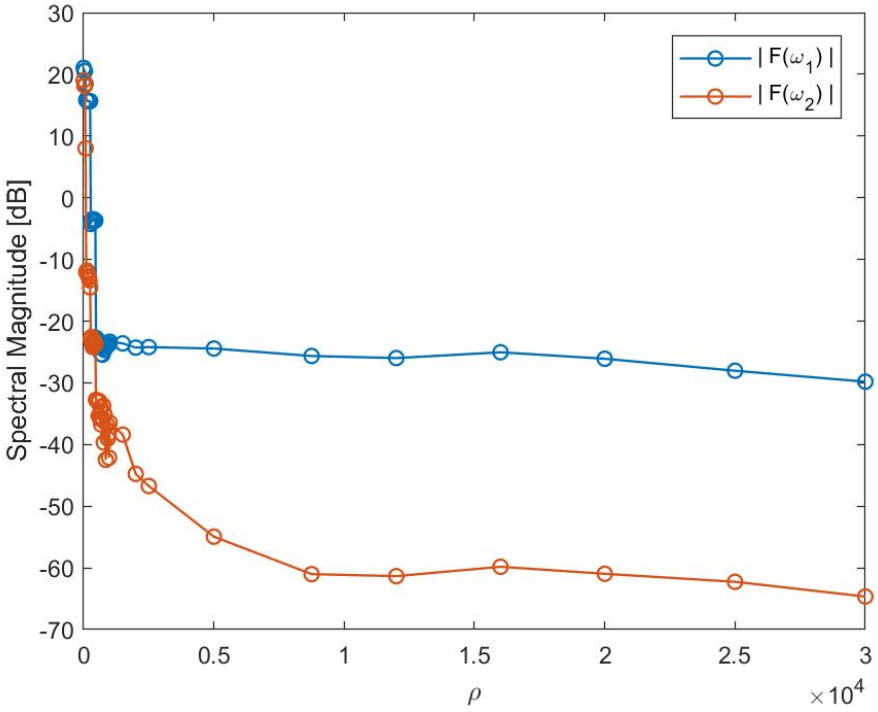

Figure 5.1. Spectral Magnitude at 1st and 2nd mode vs ρ: $a_{\max} = 40\text{rad/s}^2$.

The command shaping method used in this work only develops input profiles for motions in a cartesian plane and does not allow the user the flexibility to decide the path the links follow. Robots are generally required to have fast relocation, specifically designed trajectory tracking and ability to operate under varying payloads to be applicable in real-world situations. All of these factors affect the natural frequency of the systems and hence it becomes essential that the command shaping method is extended to incorporate features such as residual vibration reduction while also allowing path-planning flexibility.

While the simulation and experimental results discussed in this work are for a two-link robot, the command shaping concept is not restricted to 2D applications. The framework can easily be extended to develop trajectories to achieve point-to-point motion in 3D space. The changes required in the framework for the 3D motion would be to account for the effect of gravity, which was ignored in the 2D motion. Analyzing the effectiveness of the input command shaping approach on a 3D test setup will be an interesting exercise. This can be explored as an area for future work.

The natural frequencies of the system and the input trajectories are determined based on the estimated parameters of the system. These trajectories are implemented under the assumption that the parameters of the physical system completely match the parameters of the model used in the computed torque controller, but this need not always be the case. There is a 10% window built around the natural frequencies of the system to account for this, but it is possible that the parameters for the real system and the model used have a higher degree of mismatch. So, the influence of parameter mismatch on the residual vibration performance in the system needs to be further studied.

Lastly, while the peak acceleration increases by 50%, the input torque signals generated by the computed torque controller do not change proportionally with the peak acceleration of the input signals. The nonlinear nature of the computed torque controller makes it difficult to identify a relationship between the input profile and the torque signals. It is possible that this nonlinear behavior of the computed torque controller can introduce energy around the spectral attenuation regions of the input profiles or shift the points of attenuation in the generated torque signals. Hence, a deeper understanding of the computed torque controller is required to identify its influence on the shaped profile and the overall command shaping method.

LIST OF REFERENCES

[1] S. F., N. Kyura, and S. Hara. Vibration absorption control of industrial robots by acceleration feedback. *IEEE transactions on Industrial Electronics*, 30:299-305,1983.

[2] L. M. Sweet and M. C. Good. Re-Definition of the Robot Motion Control Problem: Effects of Plant Dynamics, Drive System Constraints, and User Requirements. In *Proceedings of the 23rd Conference on Decision and Control*, pages 724-732, Las Vegas, NV, 1984.

[3] P. Tomei. A Simple PD Controller for Robots with Elastic Joints. *IEEE Transactions on Automatic Control*, 36(10):1208-1213, 1991.

[4] L. Tian and A. A. Goldenberg. Robust adaptive control of flexible joint robots with joint torque feedback. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 1, pages 1229-1234 vol.1, Nagoya, Japan, May 1995.

[5] M. C. Readman. *Flexible Joint Robots*. CRC Press, Boca Raton, FL, 1994.

[6] H. Moulin and E. Bayo. On the Accuracy of End-Point Trajectory Tracking for Flexible Arms by Noncausal Inverse Dynamic Solutions. *J. Dyn. Sys., Meas., and Control*, 113:320-324, 1991.

[7] C. L. Lin and Y. H. Hsiao. Adaptive feedforward control for disturbance torque rejection in seeker stabilizing loop. *Control Systems Technology, IEEE Transactions on*, 9(1):108-121, 2001.

[8] J. Ghosh and B. Paden. Pseudo-inverse based iterative learning control for nonlinear plants with disturbances. In *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, volume 5, pages 5206-5212. IEEE, 1999.

[9] C. Lewin. Motion control gets gradually better. *Machine Design*, 66(21):90-94,1994.

[10] O. J. M. Smith. Posicast control of damped oscillatory systems. *Proceedings of the IRE*, 45(9):1249-1255, Sept 1957.

[11] N. C. Singer. *Residual vibration reduction in computer-controlled machines*. Ph.D. thesis, Massachusetts Institute of Technology, 1989.

[12] N. C. Singer and W. P. Seering. Preshaping command inputs to reduce system vibration. *ASME Journal of Dynamic Systems, Measurement, and Control*, 112:76-82, 1990.

[13] S. P. Bhat and D. K. Miu. Precise point-to-point positioning control of flexible structures. *Journal of Dynamic Systems, Measurement, and Control*, 112(4):667-674, 1990.

[14] P. H. Meckl. *Minimizing residual vibration of a linear system using appropriately shaped forcing functions*. Master's thesis, Massachusetts Institute of Technology,1984.

[15] P. H. Meckl. *Control of vibration in mechanical systems using shaped reference inputs*. Ph.D. thesis, Massachusetts Institute of Technology, 1988.

[16] A. K. M. Azad, M. H. Shaheed, Z. Mohamed, M. O. Tokhi, and H. Poerwanto. Open-Loop Control of Flexible Manipulators Using Command-Generation Techniques. In M.O. Tokhi and A. K. M. Azad, editors, *Flexible Robot Manipulators: Modelling, Simulation and, Control*, pages 207:234. Institution of Engineering and Technology, London, UK, 2008.

[17] V. M. Beazel. *Command Shaping Applied to Nonlinear Systems with Configuration-Dependent Resonance*. Ph.D. thesis, Purdue University, 2004.

[18] W. Chatlatanagulchai, V. M. Beazel, and P. H. Meckl. Command shaping applied to a flexible robot with con_guration-dependent resonance. *In American Control Conference*, 2006, pages 6-pp. IEEE, 2006.

[19] L. Y. Pao and M. A. Lau. Robust input shaper control design for parameter variations in flexible structures. *Journal of dynamic systems, measurement, and control*, 122(1):63-70, 2000.

[20] A. Agrawal. *Constrained Optimized Command Shaping for Minimizing Residual Vibration in a Flexible-Joint Robot*. Master's thesis, Purdue University, 2015.

[21] Y. Wu. *Command shaping with constrained peak input acceleration to minimize residual vibration in a flexible-joint robot*. Master's thesis, Purdue University, 2016

[22] B. Wu. *Numerical Optimized Command Shaping with Constrained Peak Input Acceleration for Minimizing Residual Vibration.* Master's thesis, Purdue University, 2017

[23] D. M. Aspinwall. Acceleration Profiles for Minimizing Residual Response. *J Dyn Syst-T ASME*, 102:3-6, 1980.

[24] A. Scheel. *Command Shaping Applied to a Two-Link Flexible-Joint Robot*. Ph.D. thesis, Purdue University, 2012.

[25] R. Bellman, I. Glicksberg, and O. Gross. *On the "bang-bang" Control Problem*. Rand Corporation, 1955.

[26] J. P. LaSalle. Time optimal control systems. *Proceedings of the National Academy of Sciences of the United States of America*, 45(4):573, 1959.

[27] J. Yegerlehner. *The Application of Artificial Neural Networks to the Control of Nonlinear System Undergoing Changes in a System Parameter*. Master's thesis, Purdue University, West Lafayette, IN, 1992.

[28] R. Kinceler and P. H. Meckl. Corrective input shaping for a exible-joint manipulator. In *American Control Conference, 1997. Proceedings of the 1997*, volume 3, pages 1335{1339. IEEE, 1997.

[29] W. Chatlatanagulchai. *Backstepping Intelligent Control Applied to a Flexible-Joint Robot Manipulator*. PhD thesis, Purdue University, 2006.

[30] H. C. Nho. *Precise motion control of flexible-joint robot manipulators with an intelligent payload estimator*. PhD thesis, Purdue University, 2004.

[31] M. W. Spong. *Modeling and control of elastic joint robots. Journal of dynamic systems, measurement, and control*, 109(4):310{318, 1987.

[32] R. S. Lee. *Optimal parameter estimation for long-term prediction in the presence of model mismatch applied to a two-link flexible joint robot.* PhD thesis, Purdue University, 2011.

[33] A. Scheel. *System Identification of Two-link Flexible Joint Robot*, 2011. http://engineering.purdue.edu/~sysIDreport

[34] A. Ghosal. *Robotics: fundamental concepts and analysis*. Oxford University Press, 2006.

[35] B. Yao. Adaptive robust control of robot manipulators-theory and experiment. *IEEE Transactions on Robotics and Automation*, 10(5):705{710, 1994.

APPENDIX A. UPDATED MATLAB SOLVER CODE

The numerical solver `fmincon` in MATLAB is used to solve for the coefficients of the harmonics of the versine function by calculating the minimum of the multi-objective function. Adding together these harmonics of the basis function gives the desired acceleration profile of the shaped input. The desired velocity and position profiles are obtained by integration for the developed acceleration profile. The MATLAB functions and subroutines used in the development of the shaped input for the versine function are presented in this section. The original scripts that read the values for the design parameters and the other variables are unchanged. The script used for calculation of the average spectral energy, plotting of the spectral magnitude and the trajectories and run the simulation for the robot are also unedited for the most part, and hence, not included in this appendix.

MATLAB script for the numerical solver and normalization of the coefficients:

```
% Filename          :  vs_coefs2_num_par.m
% Author(s)         :  Alok Agrawal
% Date created      :  2014-2015
% Date modified     :  Jan 15, 2019
% Modified by       :  Pratheek Patil
% Description       :  Computes Coefficients for Versine Forcing Function

%_____________________________________________________________________________
% vs_coefs2_num_par.m computes coefficients for versine forcing function. Rho is relative weight between spectral attenuation and faster move, L is the number of basis functions to be added together, and Tr, the rigid-body move time, is an optional parameter, only required for debugging. This uses numerical optimization code for a cost function uses a L1, or absolute difference
%_____________________________________________________________________

function [b, gam,exitflag] = vs_coefs2_num_par(wTr_2pi, Tr, rho, L, rho_option,
attbothfreq,B,gam,w)

% Compute Frequencies Surrounding Natural Frequencies

j = 0;
wTr = 0;
```

```
twopi = 2*pi;
twopi2 = twopi*twopi;
wTp1=wTr_2pi.*twopi/gam;

for ii=1:length(wTp1)
        %   Gives size of window (currently uses 5% instead of 10%)
        for jj = (0.95 : (1.05-0.95)/10 : 1.05)*wTp1(ii)
                j = j+1;
                wTr(j) = jj;   %=w*Tv for 11 different frequency;
        end % jj_loop
end % ii_loop

% Define max iterations and maximum function evaluations for the numerical solver
options= optimoptions('fmincon','MaxFunEvals',30000,'MaxIter',2000) ;

% Define the maximum and minimum limits for the design variables and the constraint function
% Function call for the numerical solver to find the minimum of the objective function.
[b,fval,exitflag]=fmincon(@fun,B,[],[],[],[],[-40*ones(1,L)],[40*ones(1,L)],@myCon,options);

function obj=fun(x)
% This is the objective function used by fmincon. It calculates the magnitude of the frequency spectra near the natural frequency (currently uses 5% window) and uses that to compute current cost using absolute difference (L1 norm). x is the design variable representing the coefficients

      N = length(wTr);          % N is the number of points selected for spectral attenuation.

      l=1:L;                    %L is the number of harmonics selected

      l2=l.*l;                  %=L^(2)

      t1 = 0:.001:1;            % Creating an array of time intervals

      F1 = ForcFunc(t1, x);     % Develop an initial profile using versine function

      SF = max(abs(F1));        % Calculate the scale factor used for normalising the coefficients

      x = x/SF;                 %Bl*    (Pre-scaling of the coefficients)

      gam =1/sum(x);            % Calculate the time penalty factor

      wTr1 = wTr*gam;
      wTr2 = wTr1.^2;

      F1 = ForcFunc(t1, x);     % Calculate profile using the normalized coefficients.
```

```
    %Calculating the sum of magnitude of the forcing function at the 22 points selected.
     for z=1:N
       fmag(z)=abs(sum(2*twopi2.*sin(wTr1(z)/2).*x(l).*l2./(twopi2.*l2-wTr2(z))));
     end

     fmag=sum(fmag.*fmag);


     % Defining the multi-objective function that the numerical solver will minimize
     % First term calculates the error between the developed function and bang-bang input
     % Second term calculates the weighted magnitude of the forcing function at 22 points
     obj = ((sum(abs(ones(1,length(t1))-F1)))/(Tr))+rho*fmag;

  end

 % Defining nonlinear constraints for the numerical optimization solver
 function [c,ceq]=myCon(x)

   t2 = 0:.001:1;
   F2 = ForcFunc(t2, x);
   SF= max(abs(F2));
   x = x/SF;

   c(1) = sum(x)-1;          % Constraint that ensures time penalty is greater than 1
   c(2) = -sum(x);           % Constraint to ensure time penalty doesn't take a negative value
   ceq = [];                 % No equality constraints
end

t = 0:0.001:1;
F = ForcFunc(t, b);          % Calculate F using coefficients calculated from solver
SF= max(abs(F));             % Calculate Scale factor
b = b/SF;                    % Use scale factor to normalize coefficients
gam = (1/sum(b))             % Gamma = 1 / sum(normalized coeffs)

end
```

The comments in the MATLAB script explain the symbols and the variables used, and the role of each function in the script. The changes made to this script from the original script used in [22] are:

1. Function `fmincon` uses the updated values of coefficients (B) each time the numerical solver is run as opposed to using a predetermined start.

2. Nonlinear constraints are defined for `fmincon` through the function `myCon` to ensure that the time penalty (Γ) at the minimum of the objective function converges to a value greater than 1.
3. The objective function `fun` has been changed such that the value of time penalty (Γ) is updated for each individual minimization iteration of `fmincon`.
4. The equations to calculate the magnitude of the forcing function at the 22 pre-determined frequencies and the multi-objective optimization function was defined incorrectly in the original script. This has been corrected.

A time penalty Γ value of 1 represents that the shaped inputs have the same actuation time as a bang-bang input. Nonlinear constraints are defined to force the time penalty to be greater than 1 as the shaped inputs are skewed compared to the bang-bang input and hence, will have a greater actuation time.

Updating the time penalty Γ with each iteration is useful as this updated value of Γ is used in the further calculation of the magnitude of the forcing function and thus, affects the results of the objective function minimizations. This correction to the code makes it more efficient and achieves faster convergence for a given set of parameters.

MATLAB Script with Aitken acceleration and the desired velocity and position profile calculations:

```
% Filename          :  versine_traj_new_acc_par.m
% Author(s)         :  Alexander Scheel
% Date created      :  February 20, 2012
% Date modified     :  Jan 15, 2019
% Modified by       :  Pratheek Patil
% Description       :  Complete version of trajectory generation with versine function
                       with the option of different approaches.

function [traj_vs_full, dtraj_vs_full, ddtraj_vs_full, testvalue, B, delta_time] =
versine_traj_new_acc_par(f, TrV, ddmax ,in0, disp, dt, L, rho, rho_option, attbothfreq,
variable_time, initial_guess)

% Create rigid-body move time for each segment.
n_segs = 2;             %Number of segments in the versine profile
Bfin = [];              % Matrix for final values of coefficients
```

```
% Compute Versine Torque Profiles
tol = 1e-3;              %Tolerance of convergence for gamma
time= [0,0];             % variable to count the number of iterations
delta = [1,1];           % variable to save error between two consecutive iterations
stop = [0, 0];
limit = 600 ;            %  Iteration limits.
flagg = [0, 0];
exitflag =[0,0];         % Variable to represent nature of convergence

gm = ones(n_segs,1);  % Variable to save final value of gamma

if variable_time         % loop to calculate for flexible actuation time condition

   for i=1:n_segs
      w = f(:,i);        % selecting natural frequencies for the particular segment
      index = 0;                        % Count for aitken acceleration algorithm
      B = 5*ones(1,L);                  % Initial guess for B to start the solver

      gms = [0, 0, initial_guess(i)];    %Loading Initial guess for gamma
      gms0 = gms;

      while delta(i) > tol

         TpV = gms(3)*TrV;              % shaped input actuation time

         for ii=1:2
            gms(ii)=gms(ii+1);
         end

         wTp_2pi = w*TpV/2/pi;

%        Function call for shaped input development using analytical approach
%         [b, gam,] = vs_coefs_par_3(wTp_2pi, TpV, rho, L, rho_option,              attbothfreq);

%        Function call for shaped input development using numerical approach
         [b, gam ,exitflag(i)] = vs_coefs2_num_par(wTp_2pi, TpV, rho, L, rho_option,
         attbothfreq, B, gms(3),w);

         B=b;                    % variable to store coefficients from solver

         gms(3) = gam;           % updating gamma

         index = index+1;        % update count for Aitken loop
```

```
%       Aitken Algorithm for estimation of gamma
        if index == 4

           % Equation for the Aitken algorithm
           gms(3) = (gms(1)*gms(3) - (gms(2))^2)/(gms(3) - 2*gms(2) + gms(1));
           delta_1 = abs(gms - gms0) < [1 1 1]*1e-5;

            if delta_1
              stop(i) = 1;
              delta(i) = abs(gms(3) - gms(2));
            break;
            end

           gms0 = gms;          % saving the updated value of gamma
           index = 1;           % reset count
        end

        % error calculation between two consecutive iterations
        delta(i) = abs(gms(3) - gms(2))

        time(i) = time(i)+1;    % increment time to count iterations

        if time(i)>limit        % No Convergence within max iteration limit
           stop(i) = 2;
             disp('Maximum Interations reached');
           break;
        end

        if (gms(3)*TrV) < 1e-3        %Odd case with time close to zero
           stop(i) = 3;
           break;
        end

     end

     gm(i) = gms(3);                  % Store final value of gamma
     Bfin = [Bfin; B];                % Store final values of coefficients
   end

else

  % Fixed time actuation
  for i=1:n_segs
     w = f(:,i);
     Tp = gms(3)*TrV;
```

```
        wTp_2pi = w*Tp/2/pi;
        [exitflag(i), b, gam] = VS_Coefs_Meckl_par(wTp_2pi, Tp, rho, L);
        gm(i) = gam;
        B = [B; b'];
    end
end

% Double integrate accelerations to obtain position & velocity
traj_vs = []; dtraj_vs = []; ddtraj_vs = [];
ddthmax = [ddmax, -ddmax];          % ddmax is the maximum allowable acceleration

B = Bfin;
Init = [in0, 0];

if (stop(1)~= 3 && stop (2) ~= 3)     % Check for true convergence
    for i=1:n_segs
        sigma = 0;      dsigma = 0;      ddsigma = 0;
        gam = gm(i);

        Tp = abs(gam*TrV);              %Actuation time of the segment
        t = 0:dt:Tp-dt;
        A = ddthmax(i);                 % Scaling factor for max acceleration

        for j=1:L                       % Double integration and adding the harmonics
            w0 = 2*pi*j/Tp;
            sigma = sigma + B(i,j)*( t.^2/2 + (cos(w0*t)-1)/w0^2 );
            dsigma = dsigma + B(i,j)*(t - sin(w0*t)/w0);
            ddsigma = ddsigma + B(i,j)*(1 - cos(w0*t));
        end

        traj_vs_0 = Init(1);
        dtraj_vs_0 = Init(2);

            if i==1             % Position, velocity and acceleration profile for 1st segment
                traj_vs_1 = traj_vs_0 + dtraj_vs_0*t + A*sigma;
                dtraj_vs_1 = dtraj_vs_0 + A*dsigma;
                ddtraj_vs_1 = A*ddsigma;
                Init = [traj_vs_1(length(traj_vs_1)), dtraj_vs_1(length(dtraj_vs_1))];
            else                % Position, velocity and acceleration profile for 2nd  segment
                traj_vs_2 = traj_vs_0 + dtraj_vs_0*t + A*sigma;
                dtraj_vs_2 = dtraj_vs_0 + A*dsigma;
                ddtraj_vs_2 = A*ddsigma;
            end
    end
```

```
   if max(max(traj_vs_2)) > disp       % Assuming initial position is 0, check for final position

      pos_stop = [3, 3]                % variable to represent overshoot in the desired position
      delta_time = 0;
      traj_vs_full = [traj_vs_1,traj_vs_2];
      dtraj_vs_full = [dtraj_vs_1,dtraj_vs_2];
      ddtraj_vs_full = [ddtraj_vs_1,ddtraj_vs_2];

   else
      delta_dis = disp - max(traj_vs_2);      % Calculate difference in desired and actual final pos
      delta_time = delta_dis / dtraj_vs_0;    %Calculate time of constant velocity section
      number = round(delta_time/dt)-1;
      delta_ddtraj_vs = zeros(1,number);
      delta_dtraj_vs(1:number) = dtraj_vs_0;
      delta_traj_vs = traj_vs_0 + dtraj_vs_0*(1:number)*dt;

      % Adding the constant velocity section to the developed profiles.
      traj_vs_3 = traj_vs_2+dtraj_vs_0*delta_time;
      traj_vs_full = [traj_vs_1,delta_traj_vs,traj_vs_3];
      dtraj_vs_full = [dtraj_vs_1,delta_dtraj_vs,dtraj_vs_2];
      ddtraj_vs_full = [ddtraj_vs_1,delta_ddtraj_vs,ddtraj_vs_2];
   end

      Tr=sum(gm)*TrV + delta_time;          % Final actuation time calculation

else              % If true convergence is not achieved, set all variables to zero
   Tr = 0;
   delta_time = 0;
   traj_vs_full = zeros(1,4000);
   dtraj_vs_full = zeros(1,4000);
   ddtraj_vs_full = zeros(1,4000);
end

 % variable to store all desired output values
 testvalue = [ddmax, rho, TrV, initial_guess, Tr, gm(1), gm(2), delta ,exitflag, stop,
              max(traj_vs_full), max(dtraj_vs_full), max(ddtraj_vs_full), time ,delta_time];

end               % End of the function
```

The two changes made to this script are:

1. The updated code uses coefficients B_l as design parameters and allows the user to specify the initial guess for them. The time penalty Γ is calculated based on B_l for the first iteration. B_l and Γ are both updated at the end of each iteration and these values are used for further calculations. The original code used Γ as a design parameter, while the value of coefficients B_l was fixed for all the iterations. This was not the optimal approach to run the solver as Γ itself was dependent on the value of the coefficients.
2. The code is made foolproof but adding in checks and conditions that set the variables to zero if convergence is not achieved at the specified iteration limits.

Along with the above-mentioned changes, multiple minor changes have been made in the subsequent scripts to build in a loop to iterate for the appropriate velocity in the case of versine function. All these changes were key in getting convergence from the numerical solver and developing shaped inputs using the versine function.

APPENDIX B. RESULTS FOR RAMPED SINUSOID INPUTS

The original code for numerical optimization worked for ρ values of up to 3 for multiple values of peak input acceleration and the maximum spectrum attenuation is -40dB as explained in [22] while the closed-form analytical approach achieved spectral attenuation of -80dB. This severely limited the residual vibration performance of the developed inputs using the numerical approach. With the changes to the MATLAB scripts explained in Appendix A, the limitations of the original numerical optimization framework with constrained peak acceleration have been fixed. To highlight the improvements to the range of parameters, a peak input acceleration of 10 $\mathrm{rad/s^2}$ is selected and the variation in the spectral energy at the selected frequencies is plotted as a function of ρ in Figure B.1. The variation of the spectral magnitude at the selected frequencies as a function of ρ for the analytical approach is shown in Figure B.2 for comparison.

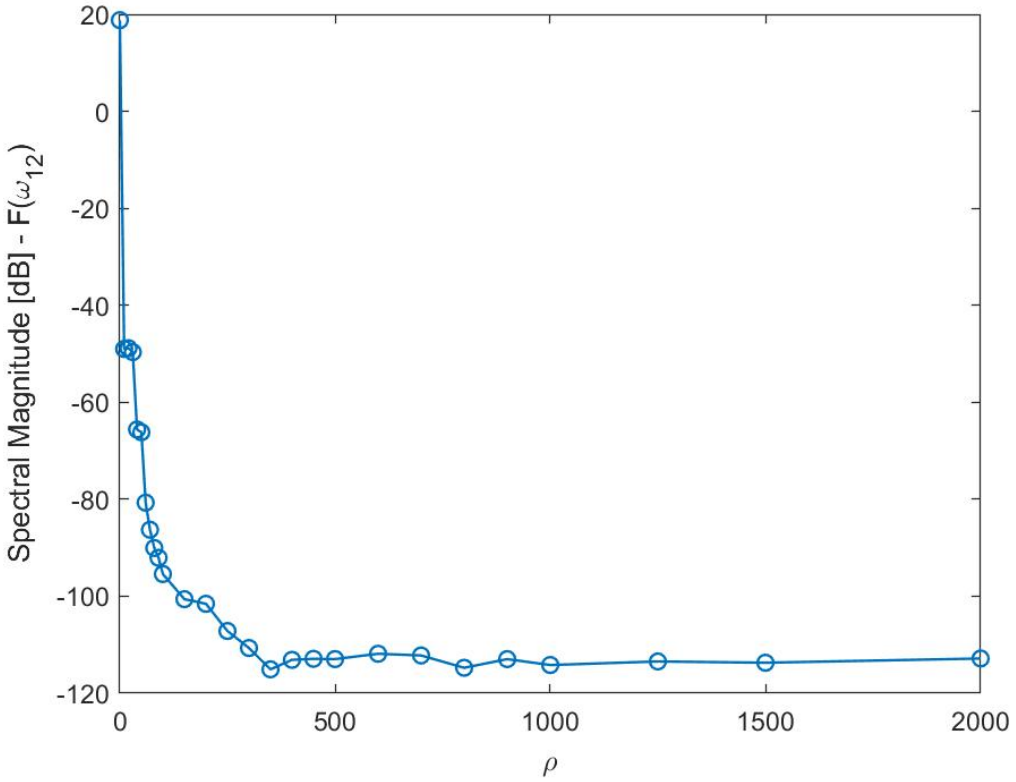

Figure B.1. Spectral Magnitude vs ρ: Numerical approach.

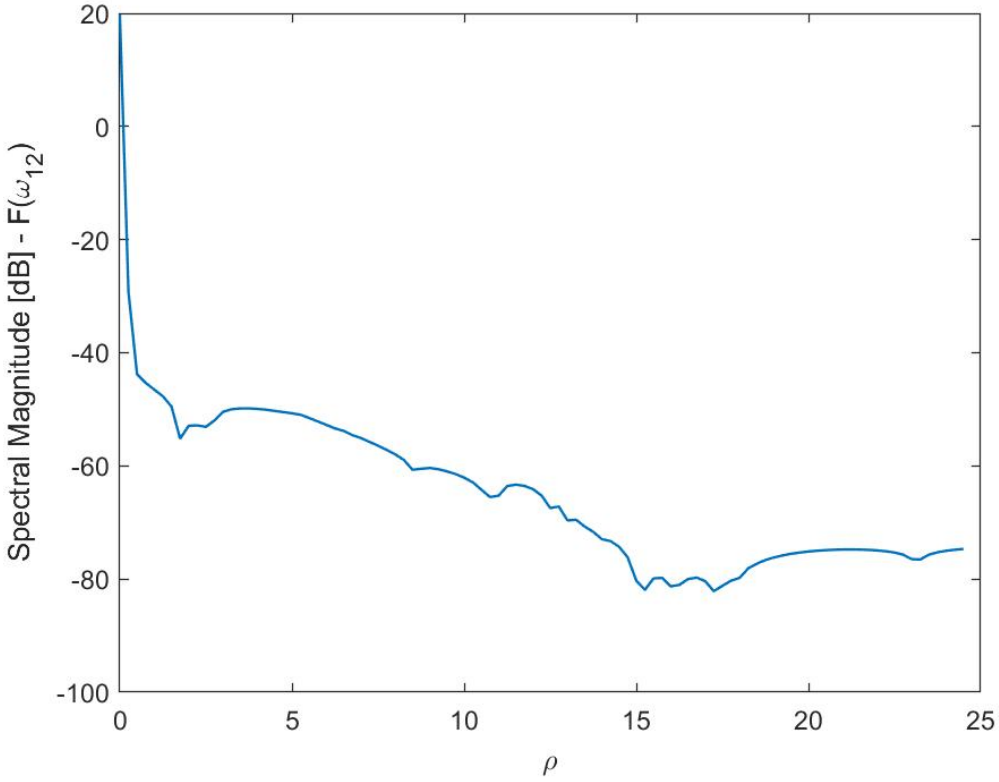

Figure B.2. Spectral Magnitude vs ρ : Closed form analytical approach.

It can be seen from Figures B.1 and B.2 that the numerical approach achieves higher spectral attenuations of close to -110 dB as compared to -80 dB by the analytical approach. The range of values for ρ has also improved and convergence was achieved for values as high as 2000. From Figure B.1 the maximum spectral attenuation is achieved for a ρ value of 350 and further reduction in spectral energy cannot be seen for inputs with higher values of ρ. Similar trends can be seen from Figure B.2, as the magnitude of spectral attenuation achieves the maximum value at ρ of 15 and higher values of ρ results in suboptimal inputs with higher spectral energies.

The effectiveness of the ramped sinusoid inputs using the numerical approach has already been investigated and explained in detail in [22]. The main goal of this exercise was to gain an understanding of the numerical optimization setup and to adjust the solver such that convergence is achieved for higher values of design parameters. Hence, further analysis of the total move time, residual vibration metrics has not been conducted for the ramped sinusoid inputs.