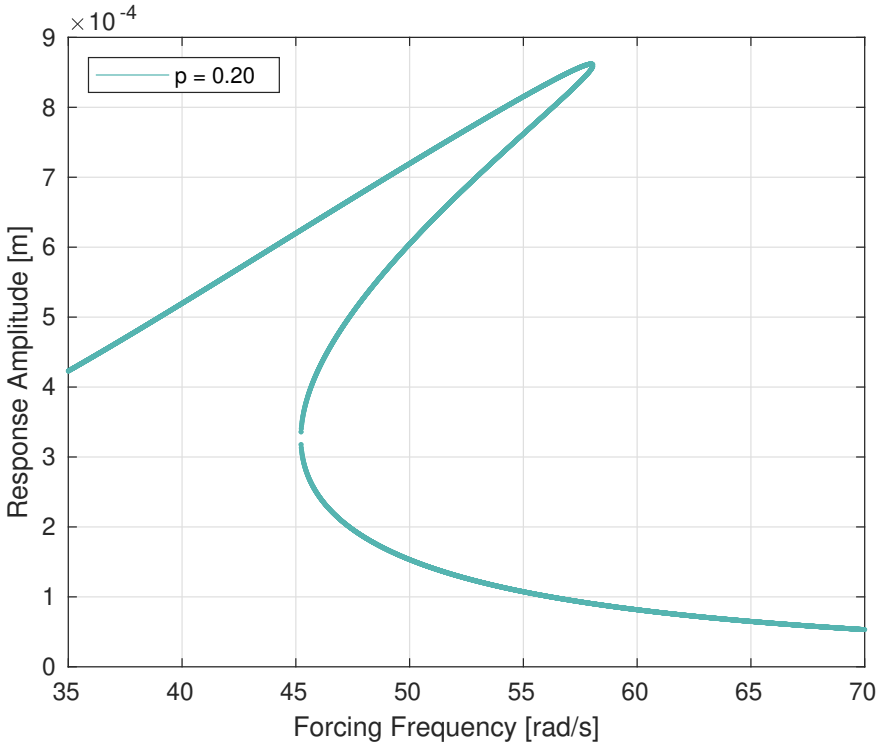


Controlled Multi-Body Dynamic Simulation for Structural Characterization

Degree project report in Systems, Control and Mechatronics

Pranav Saibhushan Ravuri
Marteinn Víðir Sigþórsson

Department of Electrical Engineering

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Degree project report 2025

Controlled Multi-Body Dynamic Simulation for Structural Characterization

Study and Analysis

Pranav Saibhushan Ravuri
Marteinn Víðir Sigþórsson

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2025

Controlled Multi-Body Dynamic Simulation for Structural Characterization
Study and Analysis
Pranav Saibhushan Ravuri
Marteinn Víðir Sigþórsson

Supervisor 1: Niclas Andersson, Volvo Car Corporation
Supervisor 2: Ingemar Andersson, Volvo Car Corporation
Examiner: Nikolce Murgovski, Electrical Engineering

Degree project report 2025
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: The plot of an analytical solution for the Frequency Response Curve that shows the nonlinear folding phenomena

Typeset in LaTeX
Gothenburg, Sweden 2025

Controlled Multi-Body Dynamic Simulation for Structural Characterization Study and Analysis

Pranav Saibhushan Ravuri
Marteinn Víðir Sigþórsson
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Virtual simulation is a popular tool used in the design of modern drivetrains, potentially offering a more efficient iterative design approach than hardware testing. For example, using feedback controls in mechanical system simulations allows testing alternative drivertrain models with different control strategies at a minimal cost. However, feedback control functionalities are seldom included in simulations for mechanical system design analysis and vice versa. Finding efficient ways that bridge the gap between the fields of structural design and feedback control design is of central interest for this project.

Phase Locked Loop (PLL) and Control Based Continuation (CBC) are feedback control architectures that recently gained popularity within the structural analysis community for characterizing nonlinear behavior. The basic analysis includes mapping of Frequency Response Curves (FRC) and the Backbone curve of the chosen dynamical system. PLL and CBC have been tested by the research community and are known to work, in both virtual and physical environments, for systems with relatively few Degrees Of Freedom (DOF). DOF can be defined as the minimum number of states required to model the system. A typical drivetrain model of today can have on the order of thousands of DOFs, depending on analysis purpose and chosen level of fidelity (higher fidelity models are typically more accurate but also use a larger number of DOFs). So, to deliver efficient feedback control of larger structural models, there is much scope for improvements among existing control architectures to satisfy industry requirements.

The thesis work aims to re-implement existing control architectures in a simple 1-DOF Duffing oscillator model. Another goal is to check whether the model-free controller used in the CBC control architecture can be replaced by a model-based controller. Model-based controllers use predicted system dynamics to generate optimal actuator signals. They provide better results, but are generally harder to implement. Finally, the process of integrating newly developed controllers into the existing drivetrain design workflow at Volvo Cars is explored.

Keywords: PLL, CBC, Control, Mechanical Oscillations, Multi-Body Dynamics

Preface

This report presents the result of our master's thesis project carried out at the Department of Electrical Engineering at Chalmers University of Technology during the spring of 2025 in cooperation with Volvo Car Corporation, Gothenburg, Sweden.

Acknowledgements

We would like to thank our supervisors, Niclas Andersson, Ingemar Andersson, Nikolce Murgovski, and our manager, Henrik Nyberg. We are grateful for their constant support and wisdom. It has been a great learning experience for us, both professionally and personally. We also thank the great scientific research community, from Galileo to Newton to Nayfeh to Abeloos, whose work forms the foundation of this thesis. Finally, we thank our family and friends for their unwavering support, which has kept us motivated and helped us in our work.

Pranav Saibhushan Ravuri, Marteinn Víðir Sigþórsson, Gothenburg, May 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis, listed in random order:

BEV	Battery Electric Vehicle
DYNO	Dynamometer
FRC	Frequency Response Curve
MBD	Multi-Body Dynamics
UUT	Unit Under Test
1-DOF	Single-Degree Of Freedom
MDOF	Multiple Degrees of Freedom
PID	Proportional-Integral-Differential
PD	Proportional-Differential
PI	Proportional-Integral
MPC	Model Predictive Control
PLL	Phase-Locked Loop
CBC	Control-Based Continuation
CBC-MPC	Control-Based Continuation Model Predictive Control
IPOPT	Interior Point OPTimizer
NLP	Non-Linear Programming
SISO	Single-Input-Single-Output
NVH	Noise, Vibration, and Harshness
RPM	Revolutions Per Minute
LMS	Least Mean Squares

Nomenclature

Below is the nomenclature of indices, parameters, and variables used throughout this thesis.

Indices

i	Index for a general iterator

Parameters and Variables

Label	Description	Units
m	Inertia	[kg] or [kg·m^2]
c	Viscous damping coefficient	[N·s/m] or [N·m·s/rad]
k	Static stiffness coefficient	[N/m] or [N·m/rad]
β	Nonlinear stiffness coefficient	[N/m^3] or [N·m/rad^3]
r	Response amplitude	[m] or [rad]
p	Excitation/Forcing amplitude	[N] or [N·m]
$a_{\mathrm{x},n}$	Sine Fourier coefficient of the n^{th} harmonic of response x	[m] or [rad]
$b_{\mathrm{x},n}$	Cosine Fourier coefficient of the n^{th} harmonic of response x	[m] or [rad]
$a_{\mathrm{u},n}$	Sine Fourier coefficient of the n^{th} harmonic of excitation u	[N] or [N·m]
$b_{\mathrm{u},n}$	Cosine Fourier coefficient of the n^{th} harmonic of excitation u	[N] or [N·m]

Variables

Label	Description	Units
x	Response displacement signal	[m] or [rad]
$x*$	Reference displacement signal for x	[m] or [rad]
u	Forcing signal	[N] or [N·m]
θ	Angular displacement	[rad]
ϕ	Response phase lag	[rad]
ω	Forcing angular frequency	[rad/s]

General Rules

- Italicized regular lower-case letter, (x): used to denote scalar variables
- Italicized bold lower-case letter, ($\boldsymbol{x}$): used to denote vector variables
- Italicized bold upper-case letter, ($\boldsymbol{A}$): used to matrix variables
- Italicized lower-case letter with a hat, ($\hat{x}$) used to denote estimated scalar variables
- Regular letter or name with concatenated parenthesis, (F(), sin(), cos()): used for functions that return scalar values
- Bold regular letter or name with concatenated parenthesis,($\mathbf{g}$()): used for functions that return vectors
- Bold zero, ($\mathbf{0}$): a vector of zeros
- A super-scripted asterisk, (x^*): used to denote the corresponding reference signal (for x in this case)
- A subscript in regular font, e.g., n in (c_{n}), : used to denote a given label of the parent variable
- A superscript regular font T: Transpose of a vector.
- General syntax for time varying variables, x: the listed variables are time varying even if they are not explicitly written as $x(t)$. Exceptions may apply in some cases

Contents

List of Figures

List of Tables

1
Introduction

The increasing popularity of Battery Electric Vehicles (BEVs) [2] over fossil-fueled vehicles is driven by a combination of economic, environmental, and technological factors. The drivetrain is an important subsystem of a BEV, consisting of components that work together, such as the rotor, stator, inverter, transmission, drive shafts, and joints, to deliver the required driving power to the wheels. These components must operate reliably, avoiding critical failures such as material fatigue, structural vibrations, or excessive noise. Extensive research is conducted during the development of the drivetrain to ensure high overall efficiency and drivability while minimizing noise and vibrations.

Virtual simulations are a popular tool frequently used in the design of drivetrains. It can be used iteratively, from early conceptual design phases through to final product verification. This allows a lot of freedom to perform virtual experiments and benchmark different control designs and mechanical structures of the drivetrain. This, in turn, has the benefit of building a good theoretical understanding of the performance of competing designs while potentially reducing total development time and cost by eliminating slow and costly hardware prototype loops.

Different kinds of simulations can be performed, depending on the drivetrain components being studied. Multi-Body Dynamic (MBD) simulation is commonly used to study interacting loads between structural parts.

1.1 Problem Description

In current development practice, feedback control functionalities are seldom included in simulations for mechanical system design analysis (or vice versa). Neglecting 3-dimensional geometry, elasticity, damping, and joint kinematics during control design can lead to inaccurate control and unstable operation. Thus, finding more robust parameter values of existing driveline control functionalities, concerning system dynamic interactions, seems very possible by introducing feedback controls into traditional mechanical simulations. Another interesting application of feedback control in structural dynamic simulations is performing virtual experimental modal analysis. Feedback control systems can be employed to precisely regulate the operating speed of a modeled test object while simultaneously applying structural excitation during simulated rig tests. The resulting collected data can then be used to dynamically characterize the virtual structure under test.

1.2 Related Work

Several recent publications within the research field of nonlinear vibration testing employs Proportional-Integral-Derivative (PID) controllers in Phase Locked Loops (PLLs) [3] and Control-Based Continuation (CBC) [1]. Abeloos conducted a comprehensive and detailed study of different feedback controllers to investigate nonlinear behavior in his Ph.D. thesis [4]. In his work, Abeloos focused on the Duffing equation, which, despite its simplicity, describes many complex nonlinear phenomena. Our research builds on this foundation by re-implementing the established feedback controllers and exploring the design of a model-based controller.

1.3 Goals and Analysis Approach

This thesis aims to integrate feedback control with possibly large-scale MBD simulations, to create a coupled system model that reflects the interaction between control and structural dynamic behavior. Feedback controls are used to identify and map complex nonlinear behaviors in dynamical systems. Backbone Curves and Frequency Response Curves (FRC) are used in the thesis to characterize nonlinear structural resonances, as these response descriptors are fundamental to the analysis of nonlinear oscillations and also well known in the structural dynamics community.

To achieve this objective, a set of structural analysis feedback controllers is researched, implemented, and evaluated with the single-degree-of-freedom (1-DOF) Duffing oscillator dynamic system. The Duffing oscillator is a simple nonlinear dynamical equation, which models rich nonlinear phenomena and is commonly studied in the research community. Matlab/Simulink [5], a popular control system design and simulation tool, is used to implement and evaluate feedback controllers for dynamical characterization of the Duffing oscillator, using scripting functionality with Matlab functions.

Volvo Cars uses Excite [6], a MBD simulation software, to design and analyze their electric drivetrains. As Excite currently does not allow for extended scripting, the controllers implemented in Simulink are used with Excite using an existing Excite–Simulink co-simulation interface.

1.4 Research questions

This thesis aims to answer the following research questions:

- Explore the simulation of a feedback-enabled controlled dynamic simulation in the Excite and Matlab software stack for the Duffing equation, and analyze the results.
- Explore how to utilize active controls to identify and track system resonance conditions.
- Design a model-based feedback controller and compare the results with the existing model-free controllers.

1.5 Sustainability

This thesis aims to develop tools for designing better driver trains that integrate well with control algorithms, increasing the lifespan of the drivetrains. Creating sustainable products requires the implementation of multi-faceted solutions that help with the Reduce-Reuse-Recycle methodology. This work tackles the first methodology, "Reduce," by increasing the lifespan of a drivetrain through better design and integration with controls.

2

Modeling Approach and Simulation Setup

This chapter describes the modeling approach and simulation environments used in this thesis. The modeling of the Duffing oscillator and the controllers was first done in Simulink. Later, the modeling of Duffing was done in Excite, and the controllers designed in Simulink are ported to work with Excite-Simulink Co-Simulation. The following subsections add more details about the setup.

2.1 Simulink

Simulink [5] is a well-established tool for control system design, widely used for time-domain simulations and automatic control code generation. This thesis work started with the implementation of feedback controllers and a single-degree-of-freedom (1-DOF) Duffing oscillator model in Simulink, leveraging its rapid modeling capabilities and robust debugging tools. The simulation results were verified against published research to confirm correctness.

2.1.1 Simulink Model Setup

The Figure 2.1 shows the Simulink model setup used in this thesis project. The Duffing oscillator is implemented in the Plant, and the prototype controllers are implemented in the Controller block. The two blocks are connected to form a closed loop. At every discrete time step, the controller receives a state x from the Plant and outputs a control signal u back to the Plant. The Plant uses this control signal along with the current state to calculate the next state using the dynamics, completing the simulation loop. This simple structure of Simulink [5] allows for prototyping of different feedback control controller designs. Three control architectures are implemented: a Phase-Locked Loop (PLL), Control-Based Continuation (CBC), and CBC with Model Predictive Control (CBC-MPC). The following chapters will add more design details.

2.2 Excite

Excite M [6] is a powerful high-fidelity modeling tool for mechanical systems, capable of solving large-scale differential-algebraic equations in the time domain. It is used to predict the nonlinear dynamic behavior, durability, and Noise, Vibration,

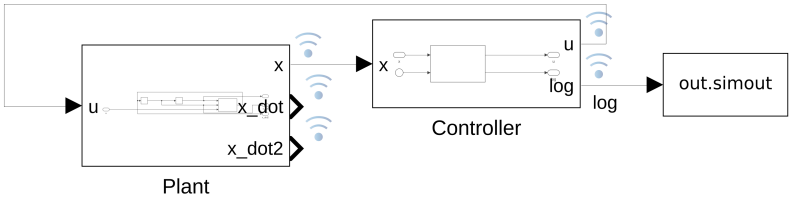


Figure 2.1: Simulink Model Block Diagram

and Harshness (NVH) characteristics of complex powertrain systems. Given that the methods developed in this project are intended for future use in iterative design analysis of state-of-the-art system models—a capability well supported by Excite—it is essential to ensure that the simulations function correctly in both Simulink and Excite environments.

After validating the control designs within the stand-alone Simulink environment, the designs are then transitioned to the Excite-Simulink Co-simulation framework for further testing. The Excite modeling approach fundamentally differs from Simulink, as Excite provides primitive mechanical components and connecting joints. This lets users construct complex mechanical systems from first principles. The underlying differential equations and algebraic constraints are automatically assembled into a unified system of equations, which is then passed to the solver. At the time of writing, Excite does not expose this unified system of equations to the user, which imposes significant constraints on controller design. Due to this limitation, only the phase-locked loop (PLL) controller was successfully implemented within the Excite environment. (Control Based Continuation) CBC was also not implemented due to a lack of time.

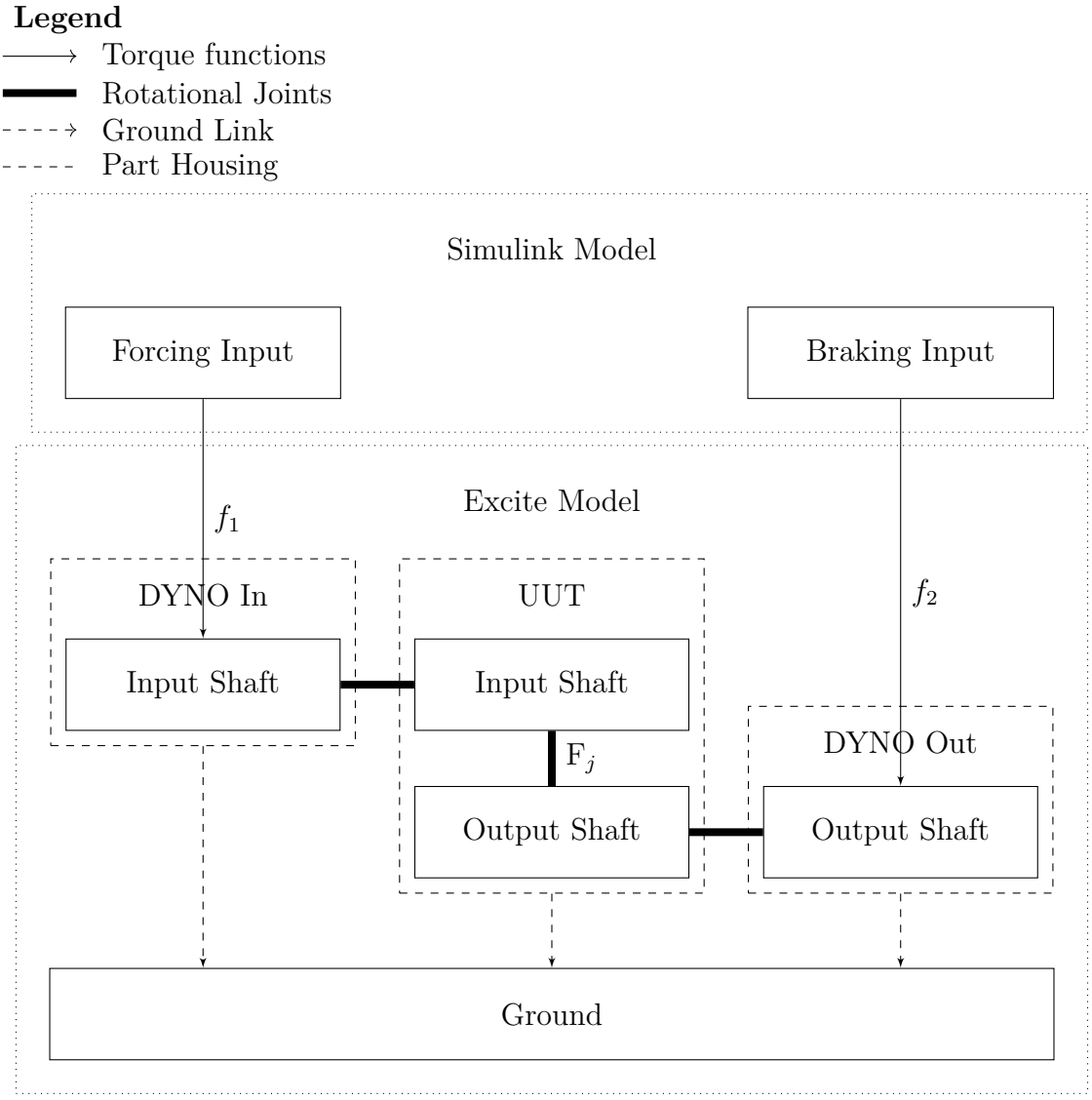
NOTE: Excite models used in this thesis project operate in rotational coordinates, natural for drivetrain applications, whereas translational ones are used in Simulink models. This leads to a general change in operational units of all parameters and variables. Hence, specific units are mentioned wherever deemed necessary.

2.2.1 Excite Model Setup

The Duffing equation is modeled within the Excite as described in Figure 2.2. This setup closely mimics a real-world test rig, featuring a Unit Under Test (UUT) positioned between the Dynamometers (DYNOs). The modular nature of the model allows the UUT to be swapped easily with alternative models in the future, facilitating the testing of various nonlinear dynamic behaviors beyond the Duffing equation.

2.2.1.1 DYNO In

This block has its rotating input shaft connected, via a stiff rotational coupling, to the Input Shaft of the UUT, and its non-rotating part connected to ground. The rotating and non-rotating parts of DYNO In are connected via an ideal 1-DOF



where,
F_j = Nonlinear joint force
f_1 = forcing torque signal from Matlab
f_2 = braking torque signal from Matlab

Figure 2.2: Excite model description

revolute joint. The joint action and reaction torques acting between the Input Shaft of UUT and DYNO In are provided by the Forcing Input block. The joint between the two input shafts of DYNO In and UUT is modeled as a Rotational Coupling Joint. The simplified equation of this joint is modeled in Excite as shown below,

$$f_M = c(\theta_2 - \theta_1) + d(\dot{\theta}_2 - \dot{\theta}_1) \tag{2.1}$$

where,

- f_M: Resulting moment (torque) in the coupling
- θ_1, θ_2: Angular positions of bodies 1 and 2
- $\dot{\theta}_1, \dot{\theta}_2$: Angular velocity of bodies 1 and 2.
- c: Torsional stiffness coefficient
- d: Damping coefficient

2.2.1.2 DYNO Out

Similar to DYNO In, this block has its rotating output shaft connected to the Output Shaft of the UUT by a stiff rotational coupling, and its non-rotating part connected to the ground. The rotating and non-rotating parts in DYNO Out are constrained with an interconnecting ideal revolute joint. For this thesis project, the DYNO Out is configured with a prescribed rotational motion of a selected fixed speed. This configuration allows Excite to calculate the required braking torque internally, instead of receiving torque values from Simulink. This choice was made to simplify the Excite model for the thesis work.

2.2.1.3 UUT

The UUT model is designed to be a block in a larger modular system assembly, to allow for fast and easy replacement of its full internal set of equations, constraints, and signals, while keeping the same controlled simulation environment. The Duffing oscillator is modeled in this thesis work, but can be replaced by much larger, interface-compatible, MBD models for future needs.

2.2.1.4 Simulink Model

The Simulink model is described in Figure 2.3. The model is straightforward: Excite provides the model states at each simulation time step, and a MATLAB function in Simulink computes the feedback-controlled torques.

2.2.1.5 Forcing Input

The Forcing Input block, implemented in a Matlab function, provides the feedback-controlled excitation torques between the Input Shaft of DYNO In and UUT. The forcing signal is defined as a function of the UUT input shaft speed, as it simply describes the intrinsic disturbance of a speed-varying electric motor. The relation is given as

$$f_1 = p \sin(c_{\text{ord}} \cdot \theta) \tag{2.2}$$

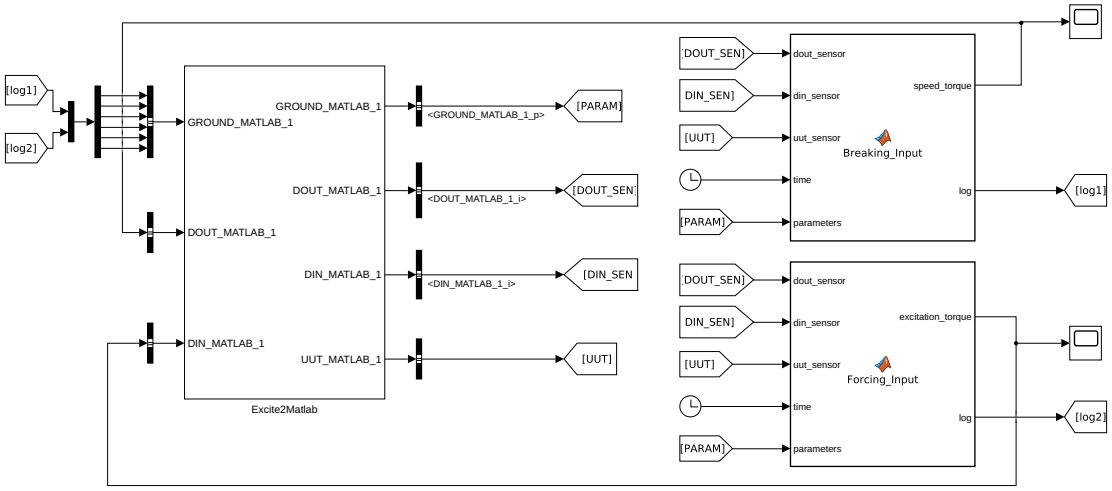


Figure 2.3: Simulink Model Block Diagram of Excite Co-Simulation

where, p is the forcing amplitude, c_{ord} is the order of angular frequency relative to the reference angular frequency (which is typically the rotor angular velocity), and θ is the DYNO In, Input shaft angle. The c_{ord} is the control output designed to push the system towards resonance based on feedback.

NOTE: The rotational speed does not affect the dynamics of the studied Duffing system, but is here used to realize a constant excitation frequency from DYNO In. This is so because of the standard setup in the Excite system model, where vibration analyses naturally use order-related excitations.

2.2.1.6 Braking Input

As mentioned earlier, this block is not utilized in the present thesis work, but is normally used to determine the action and reaction brake torques needed between the output shaft of the DYNO Out and the UUT, to actively regulate the system's operating speed.

2.2.2 Excite-Simulink Co-Simulation Setup

In the co-simulation setup, the simulation is initiated from within Excite, which subsequently triggers the execution of the Simulink model as shown in Figure 2.4. Excite

provides a pre-configured Excite Simulink data block. This block facilitates bidirectional data exchange between the Excite simulation environment and Simulink.

At each simulation time step, the Excite solver transmits the current timestamp, relevant plant states, and associated metadata to the Simulink block. The controller logic, implemented in Simulink, processes this input and generates motor control signals. These signals are then returned to the Excite solver via the same communication interface.

This closed-loop interaction enables real-time control of a high-fidelity multi-body drivetrain model using custom control algorithms developed in Simulink. The setup provides a powerful framework for evaluating control performance under realistic, dynamically rich simulation conditions.

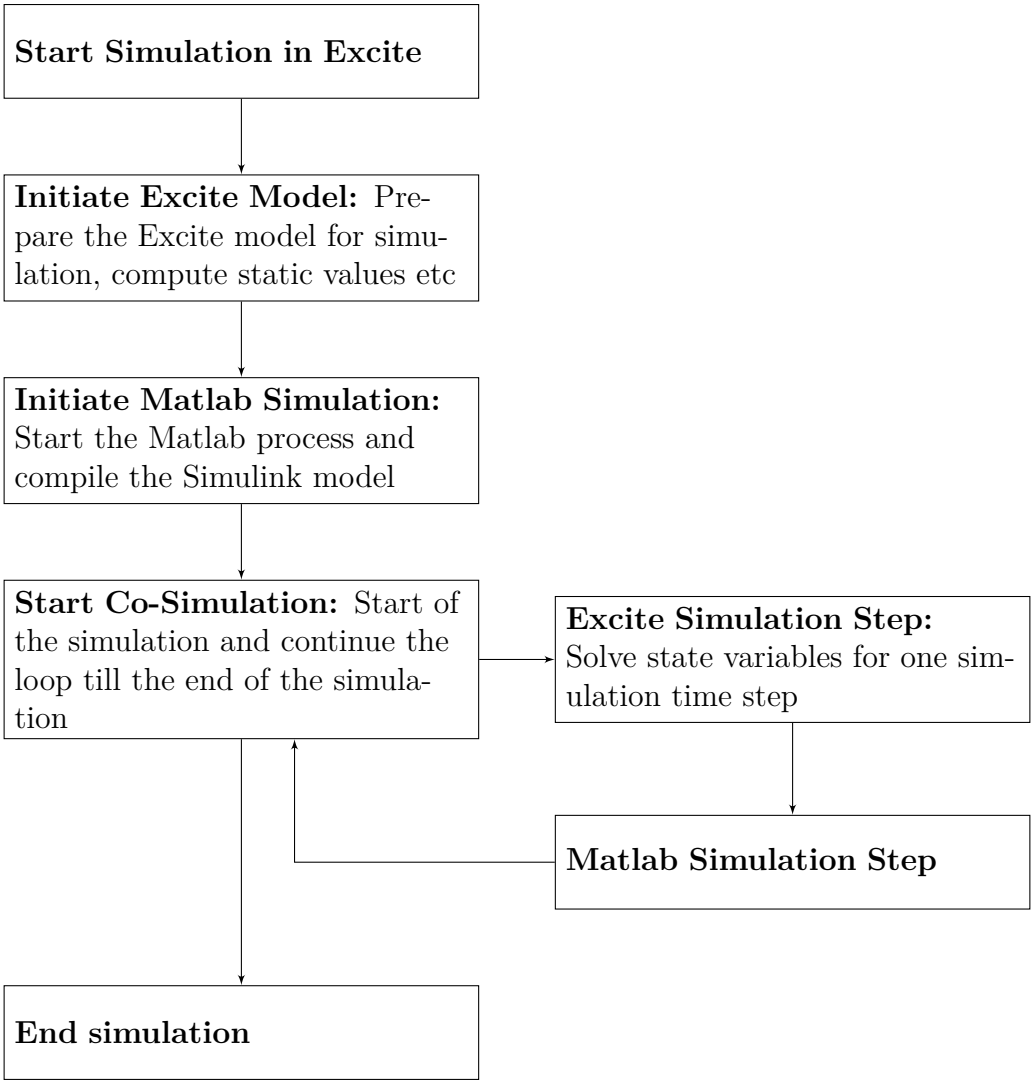


Figure 2.4: Block diagram detailing the Excite co-simulation setup

3

Theory of Mechanical Oscillations

This Chapter gives a brief introduction to linear and nonlinear mechanical oscillations, along with related phenomena.

3.1 Dynamical Equations

In statics, the focus is on the equilibrium between forces and torques while neglecting mass and inertia. In kinematics, the relationships between position, velocity, and acceleration are studied without considering the forces that cause motion. In contrast, kinetics involves solving for acceleration, velocity, and position from particle/part, mass/inertia, and acting forces/torques. Energy expressions can then be derived from these quantities. In multi-body dynamics, constraint equations are introduced to represent the relative motion between interconnected bodies. This allows for the analysis of the coupled dynamic behavior of complex mechanical systems.

Dynamical equations are a convenient way of mathematically describing complex mechanical systems that evolve. Dynamical equations can appear as non-autonomous/time-dependent or autonomous/time-independent, usually denoted by the explicit inclusion or omission of absolute time, t, respectively, as shown below:

$$\ddot{x}(t) = \mathrm{f}(x(t), \dot{x}(t), t) \quad \text{non-autonomous} \tag{3.1}$$

$$\ddot{x}(t) = \mathrm{f}(x(t), \dot{x}(t)) \quad \text{autonomous} \tag{3.2}$$

where f is a function describing the dynamics of the system (how the acceleration evolves in this case), and $x(t)$, also called the state of the system, is the quantity being used to describe the system. Notice that $x(t)$ is still time-dependent even in the autonomous case. For readability, the notation $x(t)$ will be simplified to x from this point forward.

3.2 Equations of Motion

In multi-body dynamics, second-order differential equations like Equations (3.1) and (3.2) are commonly used to model motion as described in kinetics. As described in Chapter 1 of the book by Arnold [7], based on real-world experience, second-order differential equations are usually enough to describe the mechanical systems in the

real world. Newton's equation of motion for a linear unforced system can be written as

$$m\ddot{x} + c\dot{x} + kx = 0 \tag{3.3}$$

where, m is mass, c is damping, k is stiffness.
The Equation (3.3) can be mass normalized and written as below, to make the derived analytical solutions concise.

$$\ddot{x} + c_{\mathrm{n}}\dot{x} + k_{\mathrm{n}}x = 0 \tag{3.4}$$

where, $c_{\mathrm{n}} = c/m$ and $k_{\mathrm{n}} = k/m$. The derivations in this chapter will further use the mass-normalized equation.

3.3 Duffing Type Oscillations

The Duffing Type Oscillator is given by,

$$\ddot{x} + k_{\mathrm{n}}x + \beta_{\mathrm{n}}x^3 = 0 \tag{3.5}$$

where, k_{n} is the mass normalized linear stiffness coefficient and β_{n} is the mass normalized nonlinear stiffness coefficient.

The Duffing Type Oscillator Equation (3.5) is a commonly studied dynamic equation. This oscillator includes a cubic stiffness term $\beta_{\mathrm{n}}x^3$, which introduces nonlinearity. It shows phenomena like bistability, hysteresis, amplitude-dependent frequency, and jump behaviour. This oscillator can exhibit periodic and chaotic motion depending on initial conditions. Several oscillators can be distinguished regarding the sign of the coefficients of the linear and cubic terms as described in Section 2.3 of [8].

- Hardening Duffing Oscillators (HDOs), when $k_{\mathrm{n}} > 0$, $\beta_{\mathrm{n}} > 0$
- Pure Cubic Oscillators (PCOs), when $k_{\mathrm{n}} = 0$, $\beta_{\mathrm{n}} > 0$
- Softening Duffing Oscillators (SDOs), when $k_{\mathrm{n}} > 0$, $\beta_{\mathrm{n}} < 0$
- Bistable Duffing Oscillators (BDOs), when $k_{\mathrm{n}} < 0$, $\beta_{\mathrm{n}} > 0$

We have chosen the Hardening Duffing Oscillator(HDO) as the test system for our simulations, which has also been used in previous research. In this context, as Equation (3.5) is mass normalized, we can replace the stiffness constant k_{n} with $\omega_0^2 = k_{\mathrm{n}}$, where ω_0 represents the natural (linear, undamped) resonance frequency of the system. The substitution ω_0^2 not only makes the expression more physically intuitive but also implicitly implies $k_{\mathrm{n}} > 0$, ensuring that the system retains a stable restoring force characteristic.

3.3.1 Adding Damping to the Duffing Oscillator

In many mechanical systems, damping forces, such as friction, play a crucial role in energy dissipation. For instance, in automotive braking systems, friction between

brake pads and discs slows down the vehicle by converting its kinetic energy into heat. Another less brutal damping effect is the internal material damping and the frictional forces in the mechanical joint, which, although small in absolute magnitude, quickly damp out initial vibrations. The level of modal damping determines the response amplitude of a linear oscillator. Hence, it is important to model this in the dynamic system. The amount of modal damping is highly dependent on the vibrating response mode shape, which depends on load frequencies and amplitudes, directions and distribution, temperatures, surface roughness, etc. There is no feasible approach to modeling damping from first principles, so the structural dynamic community commonly resorts to using some assumed mass and stiffness-proportional modal damping or local linear viscous damping, as in

$$\ddot{x} + c_{\mathrm{n}}\dot{x} + \omega_0^2 x + \beta_{\mathrm{n}} x^3 = 0 \tag{3.6}$$

where c_{n} is mass-normalized damping constant

3.4 Forced Duffing Oscillations

Including the Forcing term in the Duffing-type oscillator completes the equation for the Duffing Oscillator as it allows a controlled input force to be added to the system. The controlled input can be used to study the system behavior. The general non-autonomous, Equation (3.1) can be split to include a forcing function as,

$$\ddot{x}(t) = \mathrm{f}(x(t), \dot{x}(t)) + \mathrm{u}(t) \tag{3.7}$$

where, $\mathrm{u}(t)$ is the time-dependent forcing or control function. For detailed nonlinear analysis, this function is designed to drive the system towards desired dynamics.

Generally, $\mathrm{u}(t)$ can be any function; however, using a harmonic forcing function with the Duffing Type Oscillator is used in the research community as given by,

$$\ddot{x} + c_{\mathrm{n}}\dot{x} + \omega_0^2 x + \beta_{\mathrm{n}} x^3 = \underbrace{p_{\mathrm{n}} \sin(\omega t)}_{u} \tag{3.8}$$

where, $p_{\mathrm{n}} \sin(\omega t)$ is the harmonic forcing function, p_{n} is the mass normalized forcing amplitude, ω is the forcing frequency and u is the harmonic forcing or control signal. We will later prove that when ω is close to the resonant frequency, maximum response amplitude is observed.

3.4.1 Linear Case, under-damped solution

Let's consider the Linear Dynamic Equation (3.9), to derive some intermediate results to compare with nonlinear solutions.

$$\ddot{x} + c_{\mathrm{n}}\dot{x} + \omega_0^2 x = p_{\mathrm{n}} \sin(\omega t) \tag{3.9}$$

The solution for x, for the above equation, is given by the Equation (3.10) in an under-damped case $c_{\mathrm{n}}^2 < 4\omega_0^2$. The derivation is detailed in Chapter 5.1 [9]

$$x = \underbrace{\frac{p_{\mathrm{n}} \sin(\omega t - \phi)}{\left[(\omega_0^2 - \omega^2)^2 + c_{\mathrm{n}}^2 \omega^2\right]^{\frac{1}{2}}}}_{\text{forced oscillation}} + \underbrace{C e^{-\frac{1}{2} c_{\mathrm{n}} t} \sin\left[\left(\omega_0^2 - \frac{c_{\mathrm{n}}^2}{4}\right)^{\frac{1}{2}} t - \varphi\right]}_{\text{free oscillation}} \tag{3.10}$$

where ϕ is the phase lag of the response and C and φ are arbitrary constants, determined by the initial position and velocity conditions. This free oscillation term tends to zero as $t \to \infty$ due to the $e^{-\frac{1}{2} c_{\mathrm{n}} t}$ factor.

3.4.1.1 Resonance Amplitude in Linear Case

Looking at the forced oscillation term, we can see that the amplitude r of the response signal x varies with forcing frequency ω, as given by

$$r = \frac{p_{\mathrm{n}}}{\left[(\omega_0^2 - \omega^2)^2 + c_{\mathrm{n}}^2 \omega^2\right]^{\frac{1}{2}}} \tag{3.11}$$

and is maximal when the denominator of the forced oscillation term is minimum, which can be derived by setting the partial derivative to zero

$$\begin{aligned} 0 &= \frac{\partial}{\partial \omega}\left((\omega_0^2 - \omega^2)^2 + c_{\mathrm{n}}^2 \omega^2\right) \\ 0 &= 2(\omega_0^2 - \omega^2)(-2\omega) + 2c_{\mathrm{n}}^2 \omega \\ \omega_{\mathrm{res}}^2 &= \omega_0^2 - \frac{1}{2} c_{\mathrm{n}}^2 \end{aligned}$$

Hence, we can get the resonant frequency ω_{res} of the system. The corresponding forced response amplitude $r_{\max}$ of the system at resonance can be derived by substituting ω_{res} in Equation (3.11)

$$\begin{aligned} r_{\max} &= \frac{p_{\mathrm{n}}}{\left[(\omega_0^2 - \omega_{\mathrm{res}}^2)^2 + c_{\mathrm{n}}^2 \omega_{\mathrm{res}}^2\right]^{\frac{1}{2}}} \\ r_{\max} &= \frac{p_{\mathrm{n}}}{\left[(\omega_0^2 - \omega_0^2 + \frac{1}{2} c_{\mathrm{n}}^2)^2 + c_{\mathrm{n}}^2 \omega_0^2 - \frac{1}{2} c_{\mathrm{n}}^4\right]^{\frac{1}{2}}} \\ r_{\max} &= \frac{p_{\mathrm{n}}}{(c_{\mathrm{n}}^2 \omega_0^2 - \frac{1}{4} c_{\mathrm{n}}^4)^{\frac{1}{2}}} \end{aligned}$$

An observation to be made is that when the $c_{\mathrm{n}} \to 0$, $\omega \simeq \omega_0$ and $r_{\max} \to \infty$, at resonance.

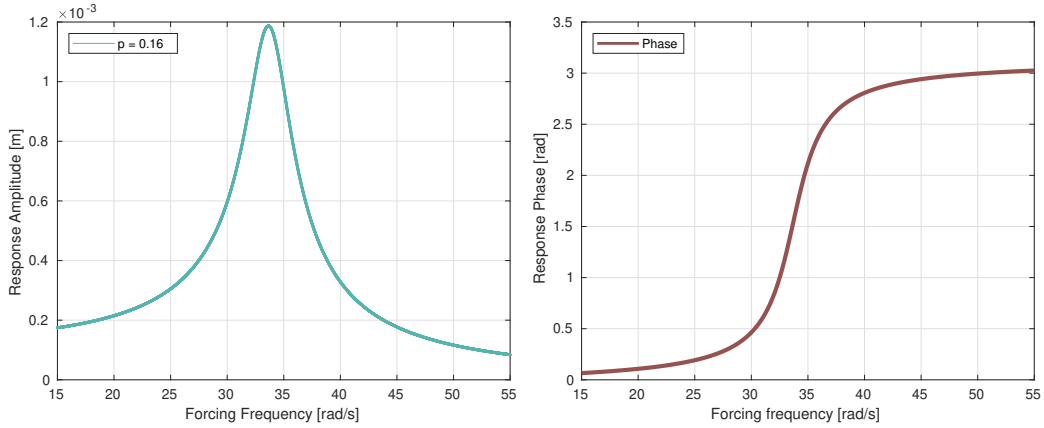
3.4.1.2 Resonance Phase Lag in Linear Case

The phase lag ϕ in Equation (3.10) denotes the phase lag between the harmonic forcing signal and the response. The equation for phase lag is given by

$$\phi = \tan^{-1}\left(\frac{c_{\mathrm{n}} \omega}{\omega_0^2 - \omega^2}\right) \tag{3.12}$$

Plotting the Equations (3.11) and (3.12), we obtain Figure 3.1. From this, it is evident that the displacement response of an undamped 1-DOF system lags the excitation force by:

- 0° below the resonance frequency,
- 90° at resonance, and
- 180° well above resonance.



(a) Response amplitude curve for different forcing frequency in linear condition

(b) Response phase lag curve for different forcing frequency in linear condition

Figure 3.1: Relation between forcing frequency ω with response amplitude r and phase ϕ.

The inclusion of damping makes the phase transition smoother, whereas reducing the damping coefficient causes the transition from 0° to 180° to be more abrupt. This phase behavior plays a critical role in defining resonance conditions and is central to techniques such as Phase-Locked Loops (PLL) and Control-Based Continuation (CBC). For example, the 90° lag at resonance is used as a phase criterion in PLL-based frequency tracking.

3.4.2 Nonlinear Case

In linear differential equations, solutions naturally divide into two additive components: the free oscillation solution, determined by initial conditions, and the forced oscillation solution, driven by external inputs and independent of initial conditions. When damping is present, the free oscillation component diminishes over time, eventually leaving only the forced response.
This separation allows for focused analysis of key characteristics such as the amplitude and phase of the steady-state response, the rate at which transient oscillations decay, and the conditions under which resonance occurs.

In nonlinear systems, this neat separation no longer holds. Nonlinear interactions between free and forced components can lead to complex behaviors [10], including self-excited oscillations, sensitivity to initial conditions, and phenomena like bifurcations (see Chapter 3 of [11]), internal resonance [12] [13], and chaos. These nonlinear

interactions result in a rich array of behaviors not present in linear systems, making their analysis both challenging and fascinating. Nonlinear normal modes [14] [15] draw parallels with the linear normal modes, formalizing the resonance conditions in nonlinear dynamics.

Rewriting the full Forced Duffing Oscillator Equation again,

$$\ddot{x} + c_{\mathrm{n}}\dot{x} + \omega_0^2 x + \beta_{\mathrm{n}} x^3 = p_{\mathrm{n}} \sin(\omega t) \tag{3.13}$$

The modern way of solving for x uses elliptical functions as described in Chapter 4 of [8]. Another approach for solving for x involves splitting of slow and fast dynamics as described in [16]. A different but relatively simpler approach used in this thesis is by using the method of multiple scales as described by Jordan in Chapters 5 and 7 of [9]. Chapter 4 of the classic Nonlinear Oscillations book by Nayfeh and Mook [17] also uses the same method. This method involves the introduction of a new variable $\tau = \omega t$, and solving it with some assumptions of light damping and small forcing amplitudes. The final equation for the approximate response amplitude is given by

$$r^2 \left\{ c_{\mathrm{n}}^2 \omega^2 + \left(\omega^2 - \omega_0^2 - \frac{3}{4} \beta_{\mathrm{n}} r_n^2 \right)^2 \right\} = p_{\mathrm{n}}^2 \tag{3.14}$$

with the response signal defined as,

$$x(t) = r \sin(\omega t + \phi) \tag{3.15}$$

where r is the amplitude and ϕ is the phase lag of the response signal.

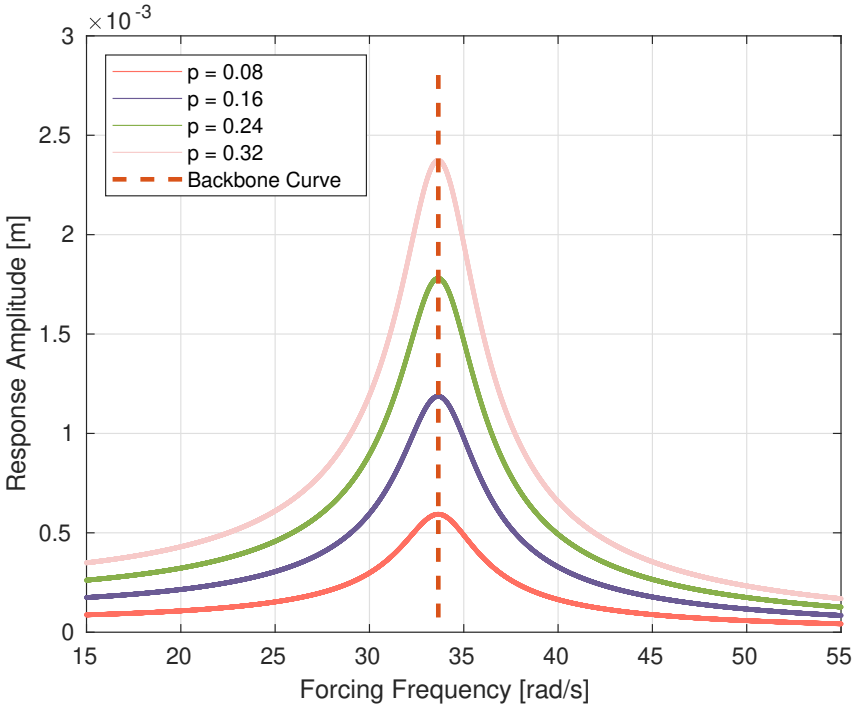
The polynomial Equation (3.14) can be solved in Matlab using the *roots* function [18] for response amplitude r for different forcing frequencies ω. The following subsections describe some nonlinear phenomena using the analytical solutions solved for the parameters listed in Table 3.1.

Table 3.1: Parameters chosen to demonstrate nonlinear phenomenon

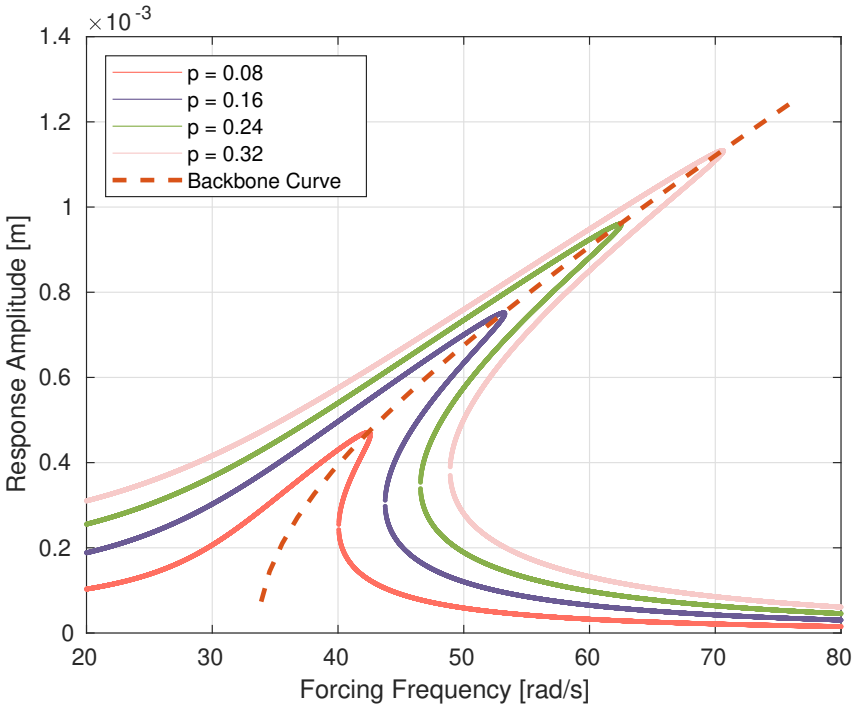
Inertia m [kg]	Damping c [N · s/m]	Stiffness k [N/m]	Nonlinear stiffness β [N/m^3]	Forcing Amp p [N]
1	4	1140	4×10^9	0.2

3.4.2.1 Frequency Response Curve (FRC)

In Figure 3.2, the forcing frequency is plotted against response amplitude for a constant forcing amplitude, p, to get a Frequency Response Curve (FRC). The FRC gives a good idea about the nonlinear response across a range of different forcing frequency values. The FRC depends on the initial conditions and previous responses of the system.



(a) Close to linear case



(b) Nonlinear case

Figure 3.2: Comparison of FRCs of closely linear cases (a) and nonlinear cases (b). Notice the folding of the backbone curve (dashed line). The solid line indicates the FRC, which is also observed to fold and shift in frequency with increasing forcing amplitude.

3.4.2.2 Backbone Curve and S-Curve

The dotted line in the Figure 3.2 shows the Backbone Curve of a close to linear case (a) and a strongly nonlinear case (b). A backbone curve is a plot of maximum response amplitude over forcing frequency for increasing values of forcing amplitude. Another way to define the backbone curve is as the plot of resonance response amplitude for a range of forcing amplitudes for different forcing frequencies. This will give a set of so-called S-curves that, taken together, describe the same nonlinear behavior as the set of FRCs does. It is important to note that the system's resonance can also be tracked using the phase as described in Section 3.4.1.2, a fact leveraged by the Phase Locked Loop (PLL) control architecture discussed later.

3.4.2.3 Folding Phenomena

The nonlinear folding phenomenon can be seen clearly in Figure 3.2 for the plotted nonlinear system. This suggests that there are multiple solutions for r for a single ω. The effect of this so-called folding phenomenon also intensifies with increasing forcing amplitude p, along with a shift in excitation frequency that corresponds to the maximum response amplitude (as shown by the backbone curve).

3.4.2.4 Jump phenomenon

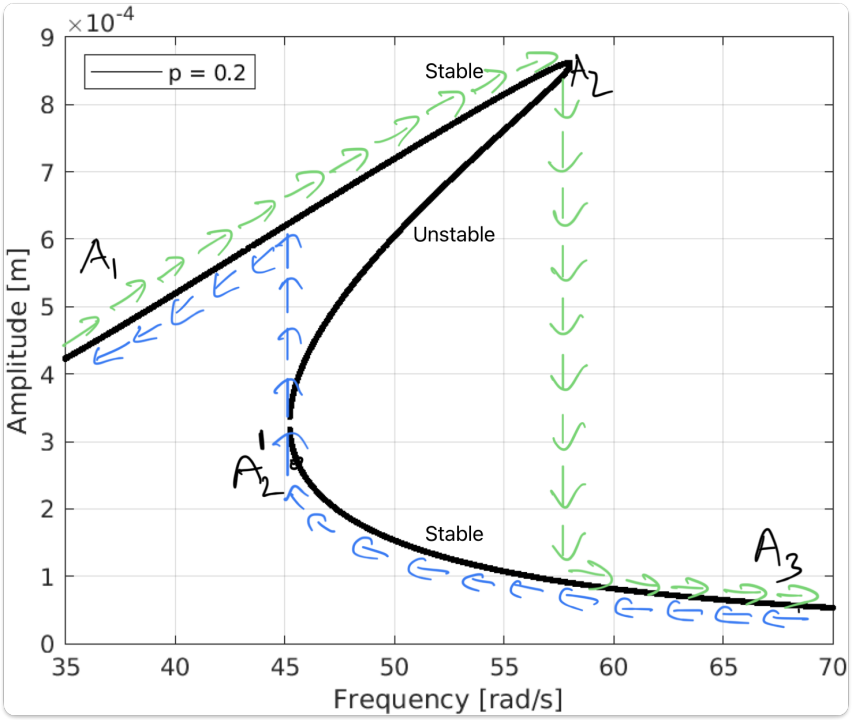


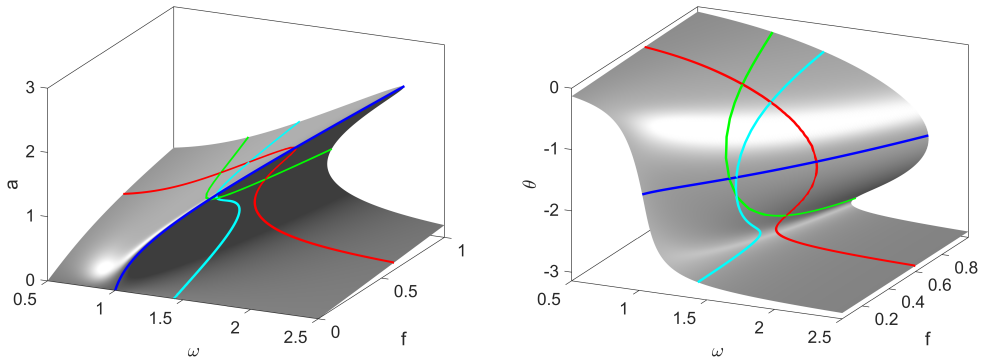
Figure 3.3: Example of jump phenomena. The green path describes the continuous speed step from left to right, and the blue path describes the corresponding speed step from right to left. The green path follows $A_1 \rightarrow A_2 \rightarrow A_2$ on the curve and the blue path follows $A_3 \rightarrow A_2' \rightarrow A_1$.

Due to the folding phenomenon, in an open-loop system as described in Equation (3.13), where r is measured over a constant increasing or decreasing of ω. The jump phenomenon is depicted in Figure 3.3. The paths followed by the response amplitude change with the direction of the changing ω (arrows indicating the direction of the changing ω). Additionally, it also depends on the initial conditions and the previous response value.

This is caused when there are multiple solutions for r. As described by Jordan in Section 7.3 of his book [9]. When there are three solutions for r in Equation (3.14), the greatest and the least values are stable, but not the intermediate one, which causes a Fold Catastrophe and leads to a jump phenomenon.

3.4.2.5 Frequency Response Manifold

By extending the FRC into a three-dimensional space—introducing an additional axis for p, we can construct the nonlinear frequency response manifold. This manifold provides a comprehensive visualization of how the system's resonant frequency evolves with varying excitation levels, capturing the intricate dynamics of nonlinear resonance. The resulting surface, often termed a forced response surface (FRS), reveals complex features, which correspond to significant resonant behaviors and bifurcations in the system's response. These characteristics are essential for understanding the underlying nonlinear dynamics and are typically computed analytically.



(a) Response amplitude nonlinear manifold

(b) Response phase lag nonlinear manifold

Figure 3.4: Image taken with permission from [1], describing the nonlinear manifold of the Duffing oscillator, linking displacement amplitude a, forcing frequency ω, and forcing amplitude f. The FRC is the red curve (left figure) with constant excitation amplitude. The S-curve is the teal curve (left figure) with constant excitation frequency. The harmonic force curve is the green curve. The backbone curve is the blue curve (left figure). The corresponding response phase lag θ manifold can be seen in the right figure (b).

3.5 Chapter Summary

To summarize, in linear differential equations, solutions naturally divide into two additive components:

- Free (homogeneous) solution: This part satisfies the homogeneous equation (i.e., with the external forcing term set to zero) and is determined by the system's initial conditions. It represents the system's natural response without external influence.
- Forced (particular) solution: This component arises from the external forcing term and is independent of initial conditions. It represents the steady-state response of the system to external inputs.

When damping is present, the free oscillation component diminishes over time, eventually leaving only the forced response. This separation allows for focused analysis of key characteristics such as:

- The amplitude and phase of the steady-state (forced) response.
- The rate at which transient (free) oscillations decay.
- The conditions under which resonance occurs, leading to amplified responses when the forcing frequency matches the system's natural frequency.

However, in nonlinear systems, this separation no longer holds. Nonlinear interactions between the free and forced components can lead to complex behaviors, including:

- Self-excited oscillations, where the system sustains oscillations without external periodic forcing.
- Sensitivity to initial conditions, where small differences in starting states can lead to vastly different behaviors.
- Complex dynamics like bifurcations and chaos, which are absent in linear systems.

Summarizing different types of curves

- FRC: plot of forcing frequency vs response amplitude with constant forcing amplitude.
- The S-curves: plot of forcing amplitude vs response amplitude with constant forcing frequency.
- The Harmonic force curve: plot of forcing frequency vs response amplitude with constant response amplitude.
- Backbone curve: plot of response amplitude at resonance vs. forcing frequency.

4

Theory of Control System

This Chapter aims to provide the theoretical background of the control system design [19]. The Figure 4.1 provides a basic structure of a closed-loop system.

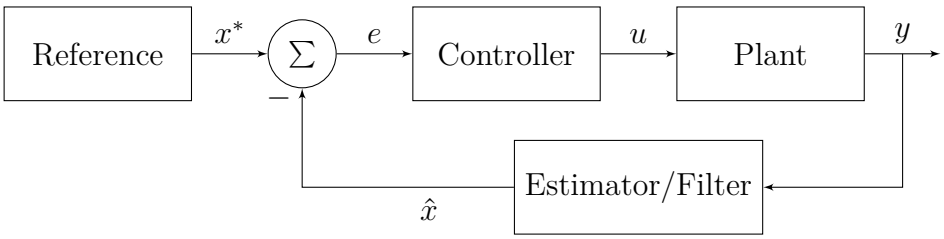


Figure 4.1: Schematic of a closed loop system

Controller

- The controller takes the error signal e and outputs a control signal u
- The controller aims to minimize the error signal $e \to 0$ as time $t \to \infty$
- As a consequence, since $e = x^* - \hat{x}$ where x^* is reference and $\hat{x}$ is the estimation of x, the controller generates a signal u such that $\hat{x} \simeq x^*$ and we can control the system by setting the reference

Plant

- The Plant takes the control signal u and outputs a measurement signal y
- The system of equations, discussed before in Chapter 3, is implemented in the plant.
- As our duffing oscillator is fully observable, $y = x$ in the simulation, we can directly observe the real state of the plant

Estimator/Filter

- The estimator is used to extract useful components from the measurements
- Some applications include using sensor fusion to combine the data from different sensors to provide a better estimate of the internal states
- We will mostly be using an adaptive filter that estimates the frequency components of the signal in real time

Reference

- The Reference is given by the desired state of the system. It can be used to calculate the error for the controller
- The reference can be static or time-dependent, as we will be discussing more in future sections of this Chapter. PLL uses a static reference, while CBC and CBC-MPC use a time-dependent reference.

4.1 Controllers

The following subsections discuss the two main controller types used in the thesis: PID and MPC.

4.1.1 PID Control Introduction

A Proportional-Integral-Derivative (PID) controller is a widely used control mechanism in industrial applications. It is designed to control Single-Input-Single-Output (SISO) systems by adjusting the control input based on error. The error is the difference between the desired setpoint with the current value. PID consists of three distinct parts: proportional, integral, and derivative. The first part creates an output that is proportional to the current error. The integral part accounts for the accumulation of past errors, and the derivative part tries to predict the future error based on the rate of change of error. Gains are used to tune their relative contribution. The PID controller can be mathematically written as:

$$u_{\text{pid}} = k_{\text{p}} e + k_{\text{i}} \int_0^t e \mathrm{d}t + k_{\text{d}} \frac{\mathrm{d}e}{\mathrm{d}t} \tag{4.1}$$

where u_{pid} is the control output of PID, e is the error, and k_{p}, k_{i}, and k_{d} are the proportional, integral, and derivative gains, respectively. The gains are tuned to give optimal control output. One of the main advantages of PID is that it does not require a model of the plant, and therefore, it can work when the dynamics of the plant are unknown. However, this controller requires manual tuning of gains, which can be time-consuming.

4.1.2 MPC Introduction

Model Predictive Control (MPC) is a control strategy that utilizes a dynamic model of the plant. At every time step, MPC solves an optimization problem to determine a sequence of optimal control actions that minimize error over a prediction horizon (N). The first control input from this optimized sequence is then applied to the plant, and the process repeats at the next time step. Unlike traditional PID controllers, which react to current errors without considering future behavior, MPC proactively anticipates future events by leveraging the plant dynamics. This predictive capability allows MPC to handle multi-variable systems with constraints more effectively and optimize performance over time.

4.1.2.1 Plant Model Transformation to State-Space

To design the MPC controller, it is nice to transform the original non-autonomous forcing Equation (3.7) to the state-space form. This is done by introducing a new global coordinate state-space vector, $\boldsymbol{x}$, that includes both the original state variable and its first time derivative

$$\boldsymbol{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \tag{4.2}$$

The final transformed form of Equation (3.7) in state-space representation, using the newly defined state vector, is:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \mathrm{f}(x, \dot{x}, t) + \mathrm{u}(t) \end{bmatrix} \tag{4.3}$$

where $\mathrm{f}(x, \dot{x}, t)$ can be the Duffing equation and $\mathrm{u}(t)$ is the time dependent forcing function. This equation can further be generalized to get the nonlinear control equation,

$$\dot{\boldsymbol{x}} = \mathrm{F}(\boldsymbol{x}) + \mathrm{G}(\boldsymbol{u}) \tag{4.4}$$

where,

- $\mathrm{F}(\boldsymbol{x})$: Represents the Plant model, which describes how the state vector $\boldsymbol{x}$ evolves without external input. It captures the natural behavior of the system based on its current state.
- $\mathrm{G}(\boldsymbol{u})$: Characterizes how the control input vector $\boldsymbol{u}$ influences the system. For the forced duffing equation, $\mathrm{G}(\boldsymbol{u}) = \begin{bmatrix} 0 \\ p\sin(\omega t) \end{bmatrix}$. Notice the first value is zero, as we do not directly control the velocity terms, but indirectly through by applying loads.

  The control vector $\boldsymbol{u}$ can include any combination of forcing amplitude, forcing frequency, or forcing phase angle, depending on the controller implementation.

Understanding the roles of $\mathrm{F}(\boldsymbol{x})$ and $\mathrm{G}(\boldsymbol{u})$ is crucial in control theory, as it allows for the design of appropriate control strategies to achieve the desired system behaviors.

4.1.2.2 Nonlinear Optimization problem

The general form of a nonlinear optimization problem can be expressed as:

$$\begin{array}{lll} \text{minimize} & \psi(\boldsymbol{v}) & \\ \text{subject to,} & \boldsymbol{g}_{\mathrm{l}} \leq \mathbf{g}_1(\boldsymbol{v}) \leq \boldsymbol{g}_{\mathrm{u}} & \text{Inequality constraints} \\ & \mathbf{g}_2(\boldsymbol{v}) = \mathbf{0} & \text{Equality constraints} \end{array}$$

here, $\psi(\boldsymbol{v})$ is the objective function to be minimized, $\boldsymbol{v}$ is a vector of optimization variables, $\mathbf{g}_1(\boldsymbol{v})$ represents the constraint function bounded by lower and upper limits $\boldsymbol{g}_{\mathrm{l}}$ and $\boldsymbol{g}_{\mathrm{u}}$, respectively, and $\mathbf{g}_2(\boldsymbol{v})$ represents the equality function. $\mathbf{0}$ is a vector of zeros.

This structure is well-established in the nonlinear optimization community, and many numerical solvers have been developed to search for the optimal values of $\boldsymbol{v}$, given the objective and the constraints. This process of solving for optimal values of $\boldsymbol{v}$ is called Non-Linear Programming (NLP).

A prominent solver used in the community is the Interior Point OPTimizer (IPOPT) [20], which employs a primal-dual interior point method and leverages first and second-order derivative information to efficiently solve large-scale NLPs.

To facilitate the computation of these derivatives, automatic differentiation tools are used, eliminating the need for manual derivative calculations. CasADi [21] is one such open-source framework that provides algorithmic differentiation capabilities and interfaces seamlessly with solvers like IPOPT. By representing mathematical expressions symbolically, CasADi can automatically generate the necessary derivative information, streamlining the formulation and solution of NLPs. In this thesis work, CasADi is utilized to define and solve the optimization problem for the MPC

4.1.2.3 Putting Everything together for MPC

The only thing that remains is to transform our dynamic equation into a nonlinear optimization problem. The two common ways of performing this transformation are through collocation or multiple shooting. We have used the multiple shooting for this thesis as explained in [22].

For simplicity, let x be a state of one dimension, with x_0 as the initial state, u be the control signal of one dimension, and N be the prediction horizon of the MPC controller. u_l and u_u are the lower and upper limit constants for the control signal. For the multiple shooting method, the vector of optimization variables $\boldsymbol{v}$ is the concatenation of states and the control signals given by,

$$\boldsymbol{v} = [\underbrace{v_{\mathrm{x},0}}_{\text{initial}}, \underbrace{v_{\mathrm{x},1}, ..., v_{\mathrm{x},N}}_{\text{state predictions}}, \underbrace{v_{\mathrm{u},1}, v_{\mathrm{u},2}, ..., v_{\mathrm{u},N}}_{\text{control predictions}}]^\mathrm{T}$$

where T is the vector transpose. With the objective function,

$$\psi(\boldsymbol{v}) = \sum_{i=1}^{N} w_\mathrm{x}(x_i - x_i^*)^2 + w_\mathrm{u}u_i^2$$

$$\text{subjected to,} \quad \boldsymbol{g}_\mathrm{l} \leq \mathbf{g}_1(\boldsymbol{v}) = \boldsymbol{v} \leq \boldsymbol{g}_\mathrm{u}$$

$$\mathbf{g}_2(\boldsymbol{v}) = \begin{bmatrix} x_1 = \mathrm{rk}_4(x_0, \mathrm{f}(x_0), \Delta t) & - & v_{\mathrm{x},0} \\ x_2 = \mathrm{rk}_4(x_1, \mathrm{f}(x_1), \Delta t) & - & v_{\mathrm{x},1} \\ \vdots & & \\ x_N = \mathrm{rk}_4(x_{N-1}, \mathrm{f}(x_{N-1}), \Delta t) & - & v_{\mathrm{x},N} \end{bmatrix} = \mathbf{0}$$

where, x_0 is the initial value, x_i is the i^{th} prediction of x, x_i^* is the reference of x in the prediction horizon i^{th}. w_x and w_u are the weight parameters for position and control errors, which can be adjusted. $\mathbf{0}$ is a vector of zeros.

Notice the function $\mathrm{rk}_4(.)$ in the equation for $\mathbf{g}_2(\boldsymbol{v})$. It is a discrete time integration function (Runge–Kutta 4). This function is used to predict the next state given the current state, time, and the plant dynamics. A good introduction to Runge-Kutta

methods is given in Chapter 4 of the book by D'Ambrosio [23].

Once this transformation is done, we can use CasADi to solve for the optimization vector $\boldsymbol{v}$ and apply the first control input to the Plant. This loop is repeated throughout the simulation.

4.2 Adaptive Notch Filter Introduction

The Adaptive Notch Filter [24] is used in this thesis to estimate the Fourier components of the response signal x and the forcing signal u in real time. The Fourier components can be used to calculate the phase lag necessary to detect resonance. As this filter is designed to dynamically adjust its parameters over time, it reduces the time to tune the filter while providing better estimates. This adaptability makes it particularly useful in applications where the target frequencies may vary over time or are not known precisely in advance.

4.2.1 Fourier Decomposition of signals

The signals x and u can be approximated as a weighted combination of sines and cosines as described below.

$$x \approx \sum_{i=0}^{\infty} a_{\mathrm{x},i} \sin(i\omega t) + b_{\mathrm{x},i} \cos(i\omega t) \tag{4.5}$$

$$u \approx \sum_{i=0}^{\infty} a_{\mathrm{u},i} \sin(i\omega t) + b_{\mathrm{u},i} \cos(i\omega t) \tag{4.6}$$

where, $a_{\mathrm{x},i}$ and $a_{\mathrm{u},i}$, $b_{\mathrm{x},i}$ and $b_{\mathrm{u},i}$ are the i^{th} harmonic Sine and Cosine Fourier coefficients of the response and excitation, respectively, ω is the chosen frequency and t is time.

To perform this operation, the complete signal in the time domain is required. However, the coefficients can be estimated in real time using the Adaptive Notch filter.

4.2.2 Design of an Adaptive Notch Filter

The adaptive filter generates an estimate, $\hat{x}$, of the signal, x, as a weighted combination of the basis vector, $\boldsymbol{q}$, with the weights updated.

$$\hat{x} = \boldsymbol{w}^{\mathrm{T}}(t)\boldsymbol{q}(t) \approx x \tag{4.7}$$

$\boldsymbol{q}$ is made out of harmonic signals and labeled as,

$$\boldsymbol{q}(t) = \begin{bmatrix} 1 \\ \sin(\omega t) \\ \cos(\omega t) \\ \sin(2\omega t) \\ \cos 2\omega t \\ \vdots \\ \sin(N_{\mathrm{H}}\omega t) \\ \cos(N_{\mathrm{H}}\omega t) \end{bmatrix} = \begin{bmatrix} q_{\mathrm{c},0} \\ q_{\mathrm{s},1} \\ q_{\mathrm{c},1} \\ q_{\mathrm{s},2} \\ q_{\mathrm{c},2} \\ \vdots \\ q_{\mathrm{s},N_{\mathrm{H}}} \\ q_{\mathrm{c},N_{\mathrm{H}}} \end{bmatrix} \tag{4.8}$$

where N_{H} is the chosen number of harmonics. Let the variables in the weight vector be labeled as

$$\boldsymbol{w}(t) = \begin{bmatrix} w_{\mathrm{c},0} \\ w_{\mathrm{s},1} \\ w_{\mathrm{c},1} \\ w_{\mathrm{s},2} \\ w_{\mathrm{c},2} \\ \vdots \\ w_{\mathrm{s},N_{\mathrm{H}}} \\ w_{\mathrm{s},N_{\mathrm{H}}} \end{bmatrix} \tag{4.9}$$

The weights are updated using the least mean squares (LMS) algorithm described in [25], resulting in Equation (4.10) with the error given by Equation (4.11).

$$\boldsymbol{w}(t_{i+1}) = \boldsymbol{w}(t_i) + \mu \boldsymbol{q}(t_i) e(t_i) \tag{4.10}$$

$$e(t_i) = x - \boldsymbol{w}^{\mathrm{T}}(t_i)\boldsymbol{q}(t_i) \tag{4.11}$$

where μ is the step size of the filter and is a tunable parameter. When the estimation error $e(t)$ is close to zero and the weights $\boldsymbol{w}$ are close to a constant vector, the weights approximate the Fourier coefficients. This is because Equation (4.7) serves as an approximation of a Fourier decomposition of x. Therefore looking at the first order components of $\hat{x}$ we get,

$$w_0 \approx b_{\mathrm{x},0} \tag{4.12}$$

$$w_{\mathrm{s},i} \approx a_{\mathrm{x},i} \tag{4.13}$$

$$w_{\mathrm{c},i} \approx b_{\mathrm{x},i} \tag{4.14}$$

for all i between 1 and N_{H}.
By representing the response x and the excitation u as fundamental harmonics, they can be written as

$$a_{\mathrm{x},1}\sin(\omega t) + b_{\mathrm{x},1}\cos(\omega t) = r\sin(\omega t + \phi_{\mathrm{x}}) \tag{4.15}$$

$$a_{\mathrm{u},1}\sin(\omega t) + b_{\mathrm{u},1}\cos(\omega t) = p\sin(\omega t + \phi_{\mathrm{u}}) \tag{4.16}$$

respectively. Then the phase lag is given by:

$$\phi_{\mathrm{x}} = \arctan\left(b_{\mathrm{x},1}, a_{\mathrm{x},1}\right) \tag{4.17}$$
$$\phi_{\mathrm{u}} = \arctan\left(b_{\mathrm{u},1}, a_{\mathrm{u},1}\right) \tag{4.18}$$
$$\phi = \phi_{\mathrm{x}} - \phi_{\mathrm{u}} \tag{4.19}$$

The amplitudes of response x and excitation u are also of interest, as they can be used for further analysis. The equations for the amplitudes can be written as:

$$r = \sqrt{a_{\mathrm{x},1}^2 + b_{\mathrm{x},1}^2} \tag{4.20}$$
$$p = \sqrt{a_{\mathrm{u},1}^2 + b_{\mathrm{u},1}^2} \tag{4.21}$$
$$\tag{4.22}$$

4.3 Control System Architectures

The control system architecture is about designing the controller, reference, and estimator blocks together for a system to produce meaningful outcomes. The subsections below discuss the closed-loop control system architectures used in the project.

4.3.1 Open loop, sinusoidal sweep

The open loop is the simplest form of control architecture shown in the Figure 4.2. An excitation input of varying frequency is given to the plant, and the measured responses are recorded and processed to generate plots for analysis. This architecture is easy to implement and provides a benchmark for the other architectures.

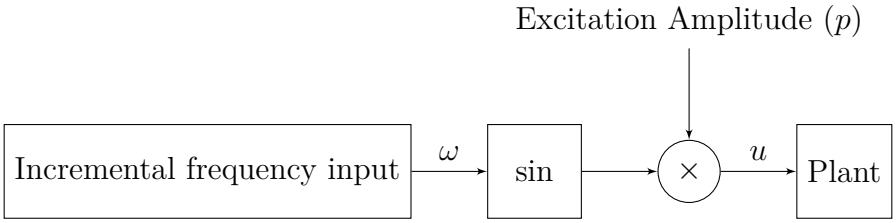


Figure 4.2: Open Loop architecture

4.3.2 PLL Introduction

The Phase Locked Loop (PLL) described in Figure 4.3 is a commonly used control architecture to perform structural analyses of nonlinear structures [3]. It takes advantage of the fact that the response amplitude, r, can be tracked on the FRC using the phase lag, ϕ, at constant forcing amplitude, p. PLL uses a classical feedback control loop, adjusting the forcing frequency, ω, to maintain a constant phase lag between the response and the forcing signal according to a reference.

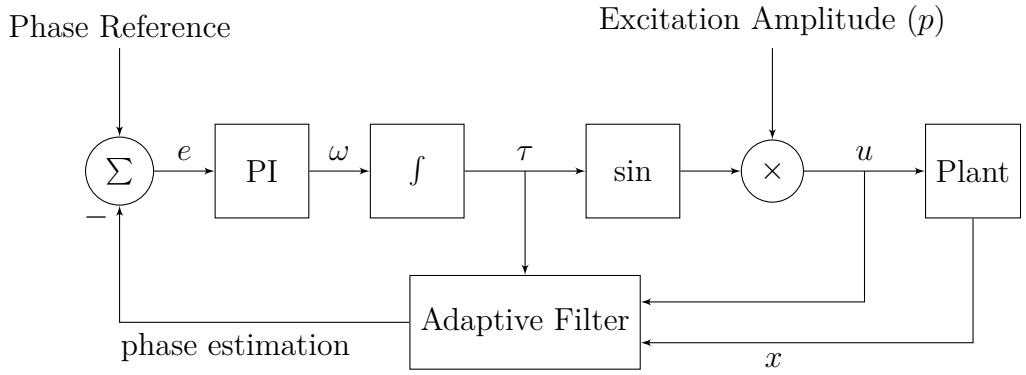


Figure 4.3: Block diagram of the architecture of PLL

$$m\ddot{x} + c\dot{x} + kx + \mathrm{f_{nl}}(x, \dot{x}) = p \sin\left(\int_0^t \omega(t)\mathrm{d}t\right) \tag{4.23}$$

where p is the forcing amplitude.
The forcing frequency ω is controlled by a simple PI controller using the error in the phase lag as:

$$\omega = \omega_0 + k_\mathrm{p}(\phi^* - \hat{\phi}) + k_\mathrm{i} \int_0^t (\phi^* - \hat{\phi})\mathrm{d}t \tag{4.24}$$

where, ω_0 is the start frequency, ϕ^* is the reference phase lag, and $\hat{\phi}$ is the estimated phase lag. As the current phase lag ϕ is not directly available, a phase lag estimation is provided by the Adaptive Notch Filter, as discussed in Section 4.2.

One drawback of PLL is its performance in multi-degree-of-freedom (MDOF) systems. When the system has multiple vibration modes that are not well-separated in frequency, the phase lag associated with each mode does not necessarily span the full range between 0 and π. This overlap or interaction between modes can compromise the ability of the PLL to accurately track a single mode [4].

4.3.2.1 FRC tracking using PLL

PLL can be used for FRC tracking by using the phase reference as an incremental variable spanning from 0 to π. The forcing amplitude stated from an initial value p_init. The PLL converges the phase lag of the system toward the current reference value, and once the response signal has stabilized, the phase reference for the PLL is incremented ($\Delta\phi^*$). This cycle repeats until the phase reference reaches π as described in Algorithm 1.

4.3.2.2 Backbone tracking using PLL

PLL is well-suited for backbone tracking because each point on a backbone curve is at resonance with the same phase lag. Thus, by setting the reference phase lag as $\frac{\pi}{2}$ (the phase lag at resonance), PLL can accurately find the point on the backbone curve for the corresponding excitation amplitude p. To track the backbone curve,

Algorithm 1 FRC tracking using PLL

1: $p \leftarrow p_{\text{init}}$
2: $\phi^* \leftarrow 0$
3: **while** $\phi^* < \pi$ **do**
4: Wait for PLL to reach steady state
5: Save response
6: $\phi^* \leftarrow \phi^* + \Delta\phi^*$
7: **end while**

PLL is initialized with an initial excitation amplitude. Once the controller has reached steady state, the excitation amplitude is incremented, and the process is repeated as described in Algorithm 2. This can be done for a certain amount of time or until a certain number of data points is gathered.

Algorithm 2 Backbone tracking using PLL

1: $p \leftarrow p_{\text{init}}$
2: $\phi^* \leftarrow \frac{\pi}{2}$
3: **while** Simulation time is not met **do**
4: Wait for PLL to reach steady state
5: Save response
6: $p \leftarrow p + \Delta p$
7: **end while**

4.3.3 CBC Introduction

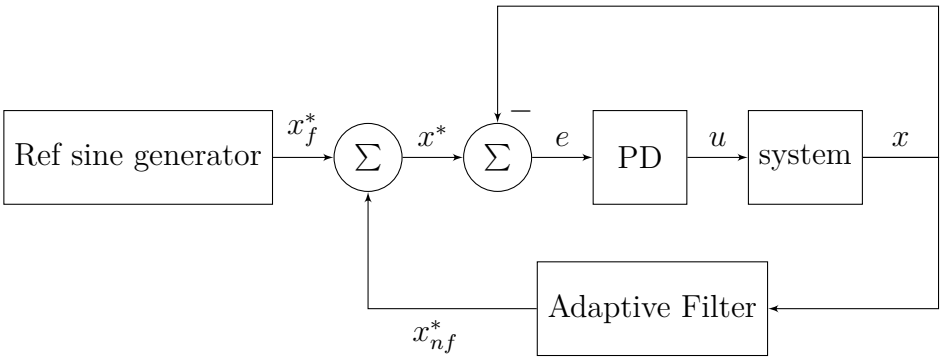


Figure 4.4: Block diagram of the architecture of CBC

The basic structure of Controlled-based continuation (CBC) architecture [1] is described in Figure 4.4. This architecture uses the time-dependent reference signal to generate the error signal for the PD controller. The novelty is in the design of the error signal, where non-fundamental response signals are canceled using the adaptive filter, to obtain the same single harmonic excitation of the system as in the open loop experiment. The control signal is defined as:

$$u = \mathrm{g}(x^* - x) = k_{\mathrm{p}}(x^* - x) + k_{\mathrm{d}}(\dot{x}^* - \dot{x}) \tag{4.25}$$

where g(*error*) is a PD function, x^* and x are the displacement reference and the measured displacement respectively. CBC can be invasive [26] as it can change the dynamics of the system through the gain values of PD, which is undesirable. This invasiveness can be detected by looking at the non-fundamental forcing signal. The stronger this signal is, the more likely the CBC is being invasive.

A nonlinear system may respond with different frequencies when subjected to mono-harmonic forcing input. Therefore, the displacement and reference displacement x and x^* can be separated and represented as follows:

$$x = x_{\mathrm{f}} + x_{\mathrm{nf}} \tag{4.26}$$

$$x^* = x^*_{\mathrm{f}} + x^*_{\mathrm{nf}} \tag{4.27}$$

where x_{f} and x^*_{f} are the fundamental components and x_{nf} and x^*_{nf} are the non-fundamental components. By feeding these signals into the controller, the control signal can be represented as:

$$u = \underbrace{\mathrm{g}(x^*_{\mathrm{f}} - x_{\mathrm{f}})}_{u_{\mathrm{f}}} + \underbrace{\mathrm{g}(x^*_{\mathrm{nf}} - x_{\mathrm{nf}})}_{u_{\mathrm{nf}}} \tag{4.28}$$

where $u_{\mathrm{f}}(t)$ and $u_{\mathrm{nf}}(t)$ are the fundamental and non-fundamental forcing control signals, respectively. The controller is said to be invasive if $u_{\mathrm{nf}}(t) \neq 0$.

The non-fundamental components can be written as:

$$x^*_{\mathrm{nf}} = b_{\mathrm{x}^*,0} + \sum_{i=2}^{N} a_{\mathrm{x}^*,i} \sin(i\omega t) + b_{\mathrm{x}^*,i} \cos(i\omega t) \tag{4.29}$$

$$x_{\mathrm{nf}} = b_{\mathrm{x},0} + \sum_{i=2}^{N} a_{\mathrm{x},i} \sin(i\omega t) + b_{\mathrm{x},i} \cos(i\omega t) \tag{4.30}$$

$$u_{\mathrm{nf}} = b_{\mathrm{u},0} + \sum_{i=2}^{N} a_{\mathrm{u},i} \sin(i\omega t) + b_{\mathrm{u},i} \cos(i\omega t) \tag{4.31}$$

By inputting the non-fundamental components of x and x^* into the Equation (4.28), the Fourier coefficients of u_{nf} are as follows:

$$b_{\mathrm{u},0} = k_{\mathrm{p}}(b_{\mathrm{x}^*,0} - b_{\mathrm{x},0}) \tag{4.32}$$

$$a_{\mathrm{u},i} = k_{\mathrm{p}}(a_{\mathrm{x}^*,i} - a_{\mathrm{x},i}) - i\omega k_{\mathrm{d}}(b_{\mathrm{x}^*,i} - b_{\mathrm{x},i}) \tag{4.33}$$

$$b_{\mathrm{u},k} = k_{\mathrm{p}}(b_{\mathrm{x}^*,n} - b_{\mathrm{x},i}) + i\omega k_{\mathrm{d}}(a_{\mathrm{x}^*,i} - a_{\mathrm{x},i}). \tag{4.34}$$

$$\tag{4.35}$$

With the system generating unknown harmonics beforehand, u_{f} and u_{nf} are non-zero, meaning that the controller is invasive. By setting the reference $x^*_{\mathrm{nf}} = x_{\mathrm{nf}}$, the Fourier coefficients of u_{nf} become zero, making the controller noninvasive.
x_{nf} is not directly measured from the system, but rather extracted using the adaptive

notch filter from Section 4.2. The output of the filter is modified to give the non-fundamental components $\hat{x}_{\mathrm{nf}}$. This can be done by reconstructing the signal using the non-fundamental weight components, as shown.

$$\hat{x}_{\mathrm{nf}} = x^*_{\mathrm{nf}} = \boldsymbol{w}^{\mathrm{T}}_{\mathrm{nf}} \boldsymbol{q}_{\mathrm{nf}} = \begin{bmatrix} w_{\mathrm{c},0} \\ w_{\mathrm{s},2} \\ w_{\mathrm{c},2} \\ \vdots \\ w_{\mathrm{s},N_{\mathrm{H}}} \\ w_{\mathrm{c},N_{\mathrm{H}}} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} q_{\mathrm{c},0} \\ q_{\mathrm{s},2} \\ q_{\mathrm{c},2} \\ \vdots \\ q_{\mathrm{s},N_{\mathrm{H}}} \\ q_{\mathrm{c},N_{\mathrm{H}}} \end{bmatrix} \tag{4.36}$$

4.3.3.1 FRC tracking using Arclength continuation CBC

To track the FRC, we employ the CBC in combination with the arc-length continuation method, as originally proposed by [27] and later modified by [28].

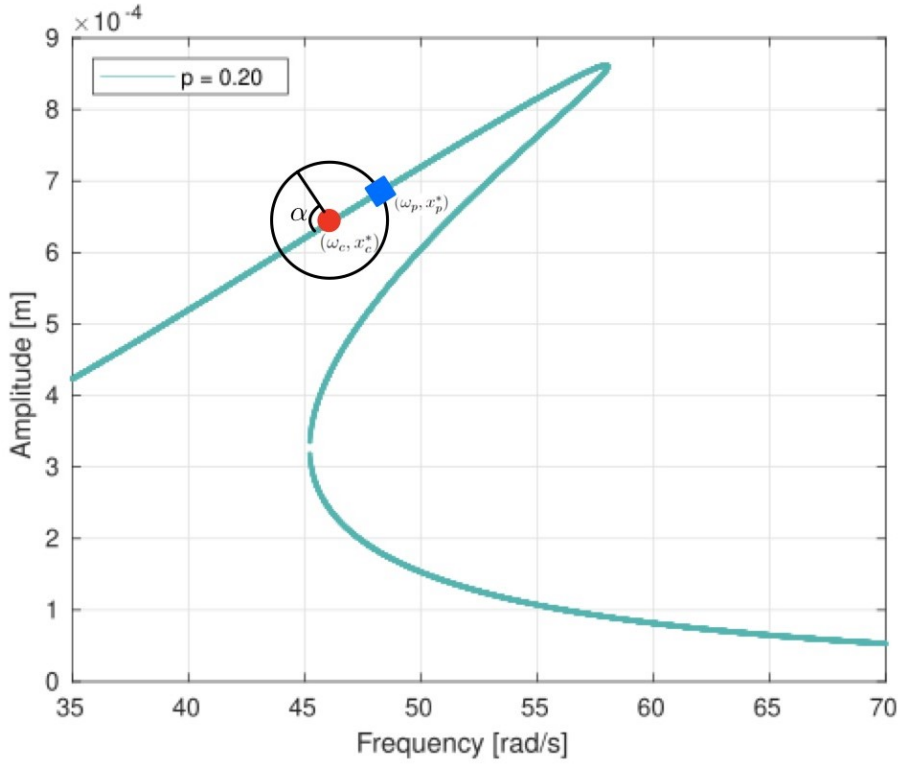


Figure 4.5: An ellipse on an FRC, showing how arc-length continuation works. The current point is indicated by a red circle, and the next point by a blue square.

An ellipse is placed on the desired FRC as shown in Figure 4.5, where the current point $(\omega_{\mathrm{c}}, x^*_{\mathrm{c}})$ is indicated by a red circle, and the next point by a blue square. The parameter α is incremented until p, the excitation amplitude, reaches its target value p^*. This can be achieved by having the adaptive filter output not only $\hat{x}_{\mathrm{nf}}$ but also excitation amplitude p. This can be done by feeding the control signal u to the adaptive filter and making p one of the outputs of the filter. Once p^* is reached (evaluated using a threshold $\mathrm{tol}_{\mathrm{p}}$), that prediction becomes the new current point,

and the ellipse is shifted; this cycle continues until the user decides to stop. The complete algorithm is shown in Algorithm 3

Algorithm 3 FRC tracking using arc-length CBC

$p^* \leftarrow p_{\text{init}}$
$\alpha \leftarrow \frac{\pi}{2}$
while Simulation time is not met **do**
    $\omega_{\text{p}} \leftarrow \omega_{\text{c}} + \Delta\omega \cos(\alpha)$
    $x^*_{\text{p}} \leftarrow x^*_{\text{c}} + \Delta x \sin(\alpha)$
    CBC action
    if $|p^* - p| > \text{tol}_{\text{p}}$ **then**
        $\omega_{\text{c}} \leftarrow \omega_{\text{p}}$
        $x^*_{\text{c}} \leftarrow x^*_{\text{p}}$
        $\alpha \leftarrow \frac{\pi}{2}$
        Save response
    else
        $\alpha \leftarrow \alpha + \alpha_{\text{step}}$
    end if
end while

One drawback of this method is that the initial center of the ellipse must be relatively accurate; otherwise, the ellipse may fail to reach a point on the FRC and get stuck in an endless loop. This issue can be avoided by having some prior knowledge of the FRC or by using another method, such as a PLL, to obtain a sufficiently good initial condition.

4.3.3.2 Backbone tracking using Arc-length continuation CBC

The same Arc-length method described in Section 4.3.3.1 can be used for backbone tracking with slight modifications. The adaptive filter is extended to also output the phase lag between displacement and forcing signal, similar to PLL. The phase lag is used to detect resonance and increment the reference forcing amplitude p^*. The complete algorithm is shown in Algorithm 4

4.3.4 CBC-MPC Introduction

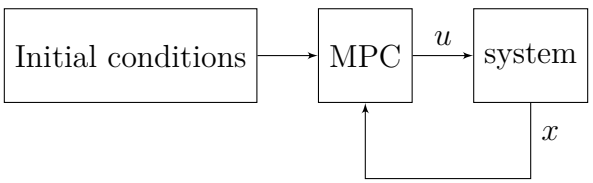


Figure 4.6: Block diagram of the architecture of CBC-MPC

The general idea of Control Based Continuation using Model Predictive Control (CBC-MPC) architecture is to replace the PD block of the CBC with an MPC.

Algorithm 4 Backbone tracking using arc-length CBC

$p^* \leftarrow p_{\text{init}}$
$\alpha \leftarrow \frac{\pi}{2}$
while Simulation time is not met **do**
    $\omega_{\text{c}} + \omega_{\text{p}} \leftarrow \Delta\omega \cos(\alpha)$
    $x_{\text{p}}^* \leftarrow x_{\text{c}}^* + \Delta x \sin(\alpha)$
    CBC action
    if $|p^* - p| > \text{tol}_{\text{p}}$ **then**
        $\omega_{\text{c}} \leftarrow \omega_{\text{p}}$
        $x_{\text{c}}^* \leftarrow x_{\text{p}}^*$
        $\alpha \leftarrow \frac{\pi}{2}$
    else
        $\alpha \leftarrow \alpha + \alpha_{\text{step}}$
    end if
    if $\left|\phi - \frac{\pi}{2}\right| > \text{tol}_{\text{phi}}$ **then**
        $p^* \leftarrow p^* + p_{\text{step}}^*$
        Save response
    end if
end while

A pure sine reference signal is generated at each time step in the MPC according to initial conditions. The adaptive filter is removed as the nonlinear optimization in the MPC takes care of it instead. The MPC is designed using the state-space equation given by,

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ -c_{\text{n}}\dot{x} - k_{\text{n}}x - \beta_{\text{n}}x^3 + u \end{bmatrix} \tag{4.37}$$

and the reference signal given by,

$$x^* = r^* \sin(\omega^* t) \tag{4.38}$$

where u is the control variable, x^*, r^*, and ω^* are the reference position, reference response amplitude, and reference response frequency, respectively.

The objective function is chosen to be

$$\psi(\boldsymbol{v}) = \sum_{i=1}^{N} w_{\text{x}}(x_i - x_i^*)^2 + w_{\text{u}}u_i^2 \tag{4.39}$$

where w_{x} and w_{u} are tunable weights.

4.3.4.1 Backbone tracking using Arc-length continuation CBC-MPC

The same arc-length algorithm described in Section 4.3.3.1 can be used for backbone tracking. This requires the estimation of response and forcing amplitudes that can be done by adding the Adaptive Notch Filter back.

4.4 Chapter Summary

A controller can be used to drive the error to zero. The controller can be model-free or model-based, where model-based controllers use the dynamics of the system to calculate the control signal. We have mainly used two controllers in this thesis.

PID Controller

- This controller generated the control signal mostly in proportion to the error signal e
- Simple to implement, but requires tuning of gains

MPC

- This controller generated the control signal based on the dynamics of the system
- A nonlinear optimizer is used to find the control signal based on the current state, system dynamics, and prediction horizon
- This controller is relatively simple to tune when compared to a PID

Adaptive Notch filter

- The filter is required to estimate different frequency components present in the system in real time.
- The tuning parameters for the filter are the step size μ and the initial weights w
- The filter can be coupled with the controllers to make control architectures that can be useful to track nonlinear behaviors

PLL Architecture

- Uses error in phase lag to determine resonance
- Works well in 1-DOF systems, but can get tricky with MDOF systems
- PLL can be used to extract backbone curves naturally by increasing the forcing amplitude
- PLL can also be used to extract FRC by adjusting the reference phase.

CBC Architecture

- This uses a reference signal to calculate error
- Care must be taken to check if the controller is invasive
- CBC is known to work well with MDOF systems
- Extracting FRCs and backbone curves can get tricky, as algorithms like arc-length continuation must be implemented

CBC-MPC Architecture

- This replaces the PD controller of CBC with an MPC
- There is no direct control on the excitation function, so determining if the controller is invasive gets tricky
- Extracting FRCs and backbone curves can get tricky, as algorithms like arc-length continuation must be implemented

5

Results of Simulink Simulations

The chapter lists all the results from the synthetic simulations in Simulink.

5.1 Chosen Dynamics

A synthetic system was selected to evaluate the various types of structural analysis controllers given in Chapter 4. The objective of these tests is to assess the performance of each controller, particularly their ability to accurately identify key characteristics of the structural system, such as the backbone curve and the frequency response curve. The 1-DOF forced duffing oscillator equation is implemented with the parameters listed in the table below:

Table 5.1: Set of parameters for the synthetic system

Inertia m [kg]	Damping c [N · s/m]	Stiffness k [N/m]	Nonlinear stiffness β [N/m^3]	Forcing Amp p [N]
1	4	1140	4×10^9	0.2

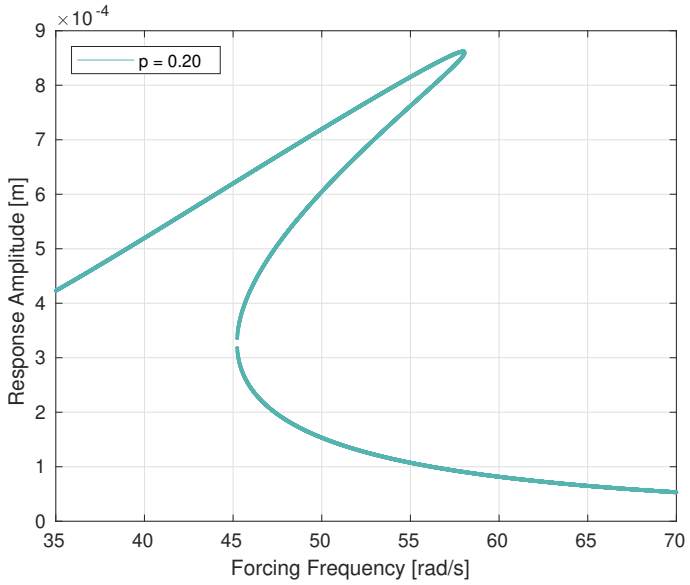


Figure 5.1: Frequency response curve of the system given in Table 5.1 with the exaction amplitude $p = 0.2$ [N]

5.2 Open Loop Simulation Results

An open-loop experiment, described in subsection 4.3.1, was conducted on the system presented in section 5.1. Both low-to-high and high-to-low frequency sweeps were performed with an excitation amplitude of $p = 0.2$ [N], with responses plotted in Figure 5.2. Evidently the controller fails to capture points on unstable branch

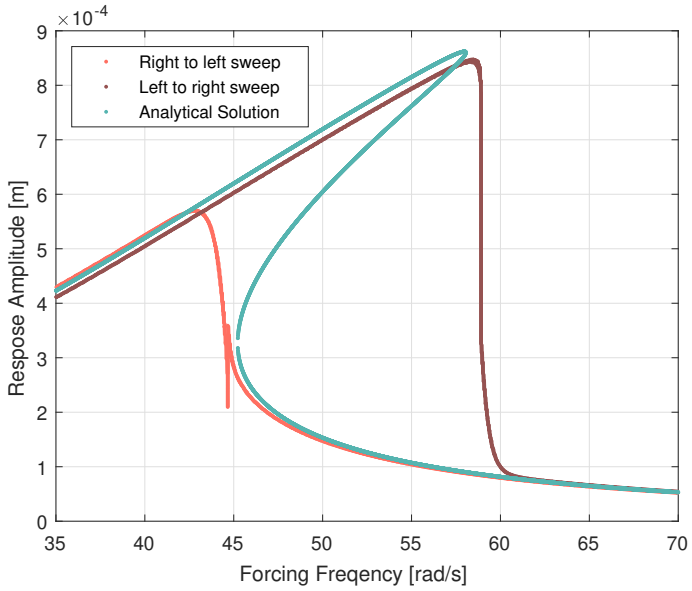
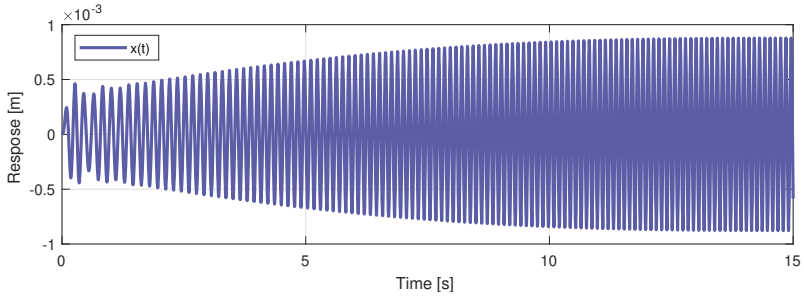


Figure 5.2: Sinusoidal sweep done on an open-loop system with excitation amplitude of $p = 0.2$ [N], clearly depicting the jump phenomenon.

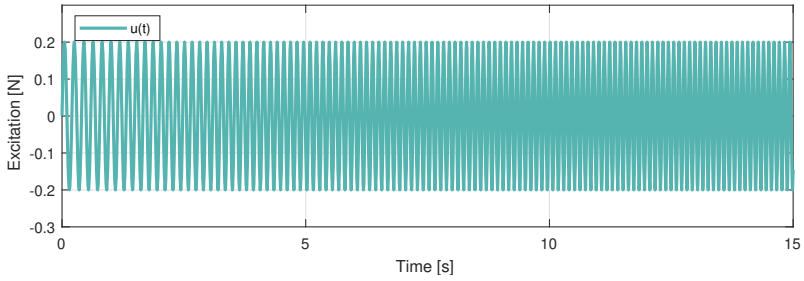
of the FRC and exhibits the jump phenomenon during both forward and backward frequency sweeps.

5.3 PLL Simulation Results

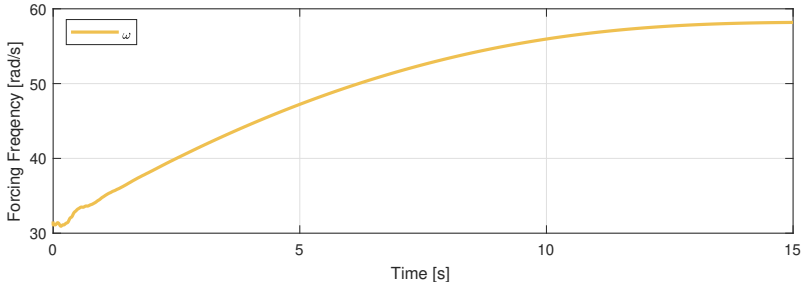
The PLL experiment was also conducted on the system given in Section 5.1 using an excitation amplitude of $p = 0.2$ [N]. Figure 5.3 shows time-responses from the numerical experiment, where the controller successfully locks onto the upper branch of the FRC, with an initial forcing frequency just below the corresponding linear system resonance frequency, and maintains this lock as the forcing frequency converges to fulfill the specified phase-lag value, in this case $\pi/2$. Initially, the controller produces inaccurate phase shift estimates due to the instability of the adaptive filter. However, once the filter stabilizes, the controller effectively minimizes the phase error.



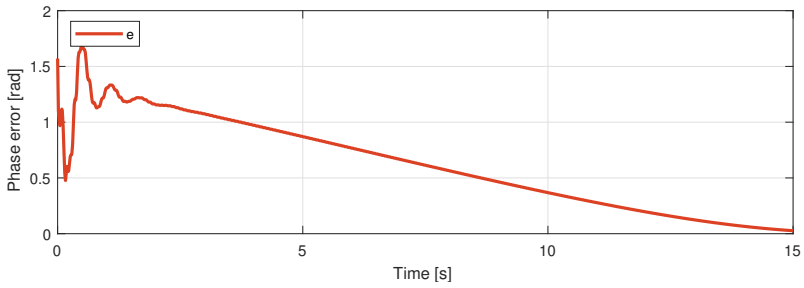
(a) Response signal x



(b) Exaction signal u



(c) Forcing Frequency



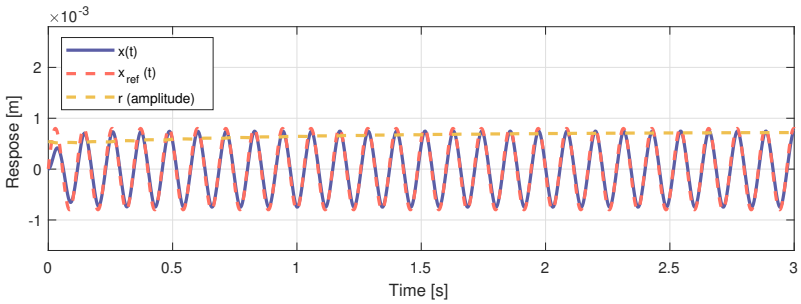
(d) Phase error

Figure 5.3: PLL Results with an excitation amplitude of $p = 0.2$ [N] with $k_\mathrm{p} = 1$ and $k_\mathrm{i} = 3$. Notice the convergence of the forcing frequency in (c); this corresponds with the resonance frequency as the error is close to zero.

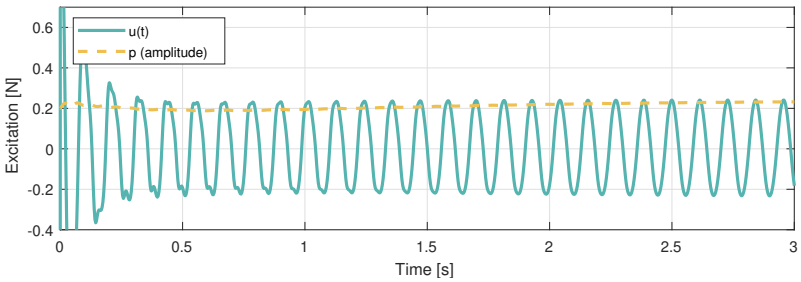
5.4 CBC Simulation Results

Figure 5.4 shows the result of the CBC experiment with forcing frequency $\omega = 50$ [rad/sec] and the reference response amplitude $r^* = 0.8 \times 10^{-3}$ [N] on the system given in Section 5.1. As observed in the figure, the CBC is able to closely follow

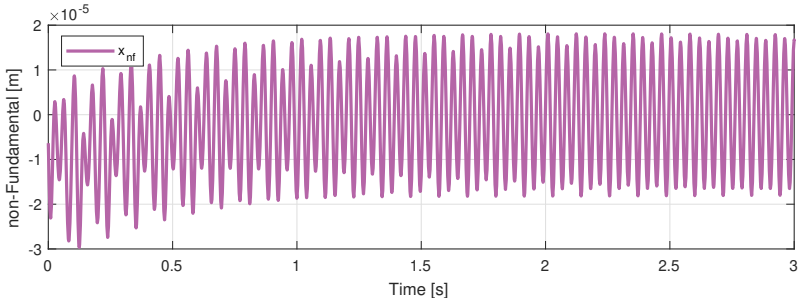
the reference signal after an initial transient period. The controller can produce the expected excitation amplitude of $p = 0.2$ [N].



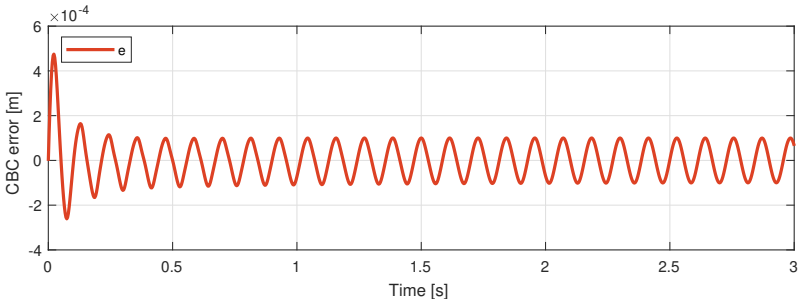
(a) Response signal x



(b) Exaction signal u



(c) Non fundamentals

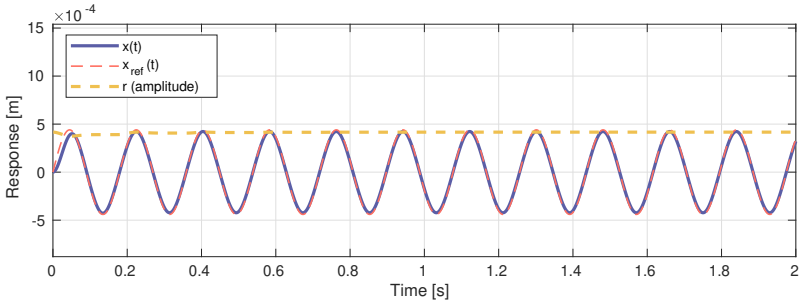


(d) CBC error

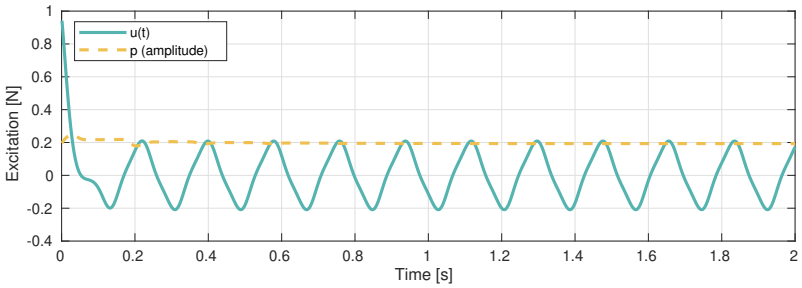
Figure 5.4: CBC Results at a $\omega = 50$ [rad/sec] and $r^* = 0.8 \times 10^{-3}$ [N] with with $k_\mathrm{p} = 20$ and $k_\mathrm{d} = 44$. The small values of the non-fundamental signal in (c) likely suggest the controller is non-invasive. The error is also small, of order 10^{-4} [m].

5.5 CBC-MPC Simulation Results

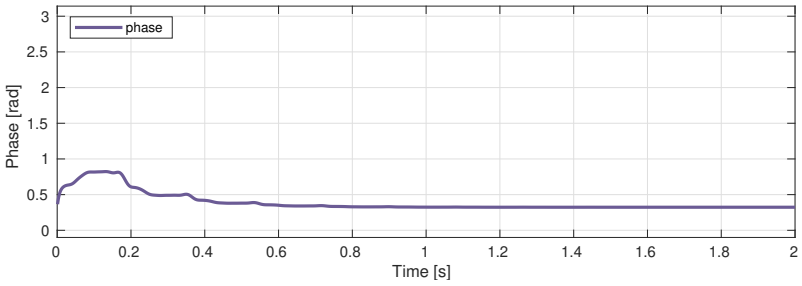
The results for this method are shown in Figure 5.5 and Figure 5.6. As observed in the figures, CBC-MPC generates large transients at the beginning of the control signal, before settling to a stationary signal. Care must be taken to allow for these large transients when setting the lower and upper limits of the optimization variables. The settling time of the response is under 0.1 [sec] for both sub-resonance and resonance conditions. The controller was run with prediction horizon $N = 100$ and with sampling time $\Delta t = 0.001$ [sec]. The average MPC loop time is 0.0263 sec. Also, notice in Figure 5.5 that the control signal generated is not a pure sine wave, hence, the benchmark experiment of Section 5.1 could not be replicated. Currently there is no direct way of controlling this signal and this could be a topic for future improvement.



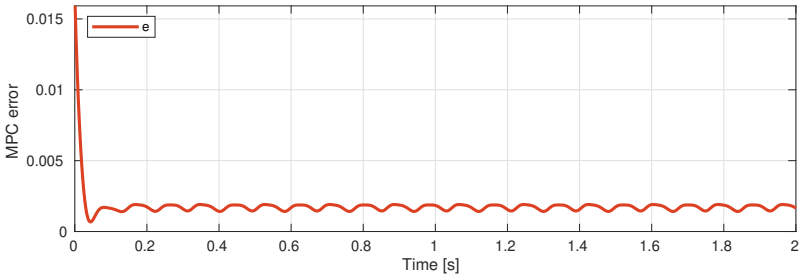
(a) Response signal x



(b) Control signal u

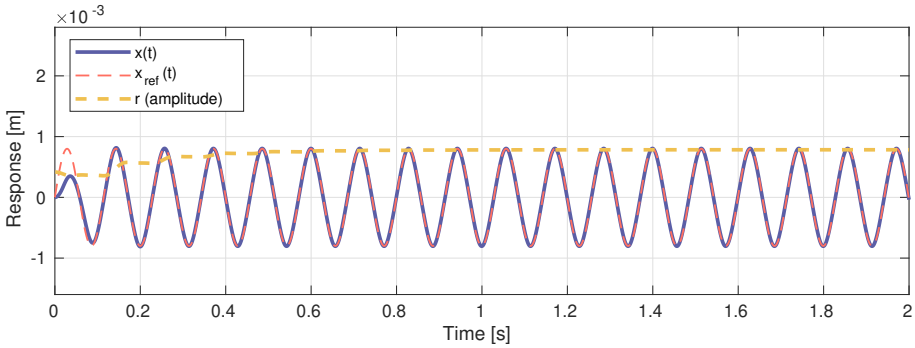


(c) Phase

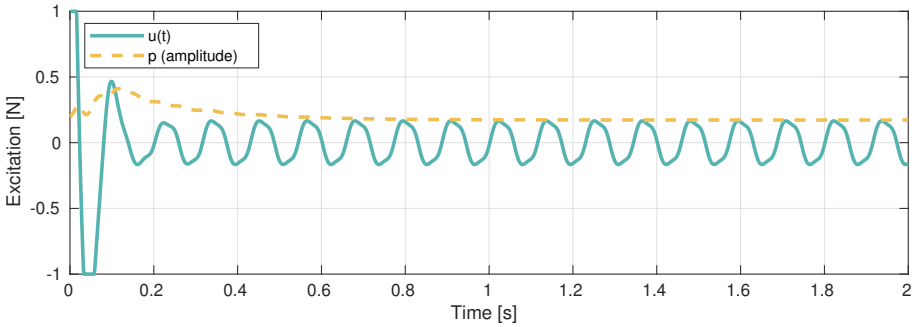


(d) MPC error

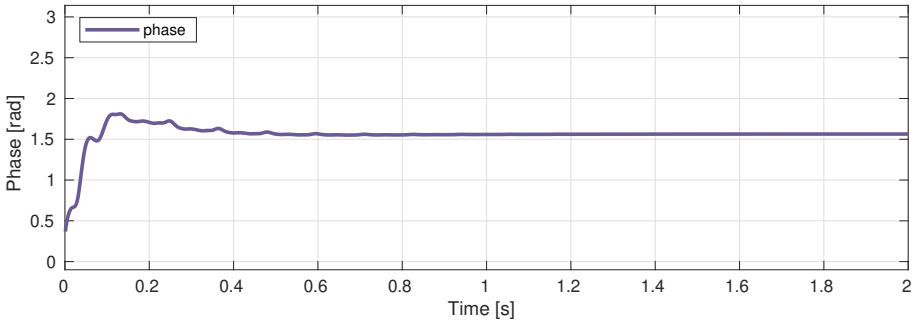
Figure 5.5: CBC-MPC Results at a sub resonance region $\omega = 35$ [rad/sec] and $x_{\text{ref}} = 0.44e - 3$ [N]. The control input is not a pure sinusoidal wave in (b), which is to be improved on in future work with a better design of the cost function in MPC. The error is minimal in (d) as the response closely follows the reference in (a).



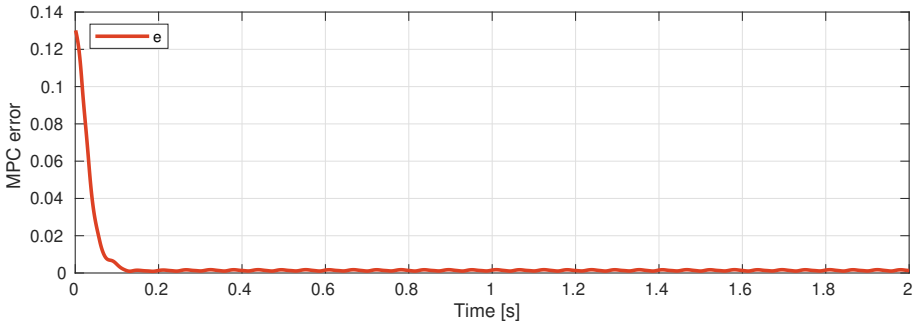
(a) Response signal x



(b) Control signal u



(c) Phase



(d) MPC error

Figure 5.6: CBC-MPC Results at a resonance region $\omega = 55$ [rad/sec] and $x_{\text{ref}} = 0.8e - 3$ [N]. The control input is not a pure sinusoidal wave in (b), which is to be improved on in future work with a better design of the cost function in MPC. The error is minimal in (d) as the response closely follows the reference in (a).

5.6 Comparison of FRC Curve Results

Figure 5.7 presents a comparison of the different frequency response curves obtained using previously discussed control methods. As shown, the PLL method struggles to capture points of the unstable branch of the FRC and is unable to identify them. However, the points it detects are close to the analytical solution. Unlike the PLL, the CBC method does not struggle to identify the unstable regions; however, the accuracy of the detected points is somewhat off compared to the analytical solution. This is most likely due to the invasive-ness of the controller, meaning that the adaptive filter is not accurately estimating the non-fundamental components, which in turn makes the control action invasive. The CBC-MPC failed to generate usable outputs for the FRC, potentially due to invasiveness or lack of tuning, and hence, was omitted from the plot.

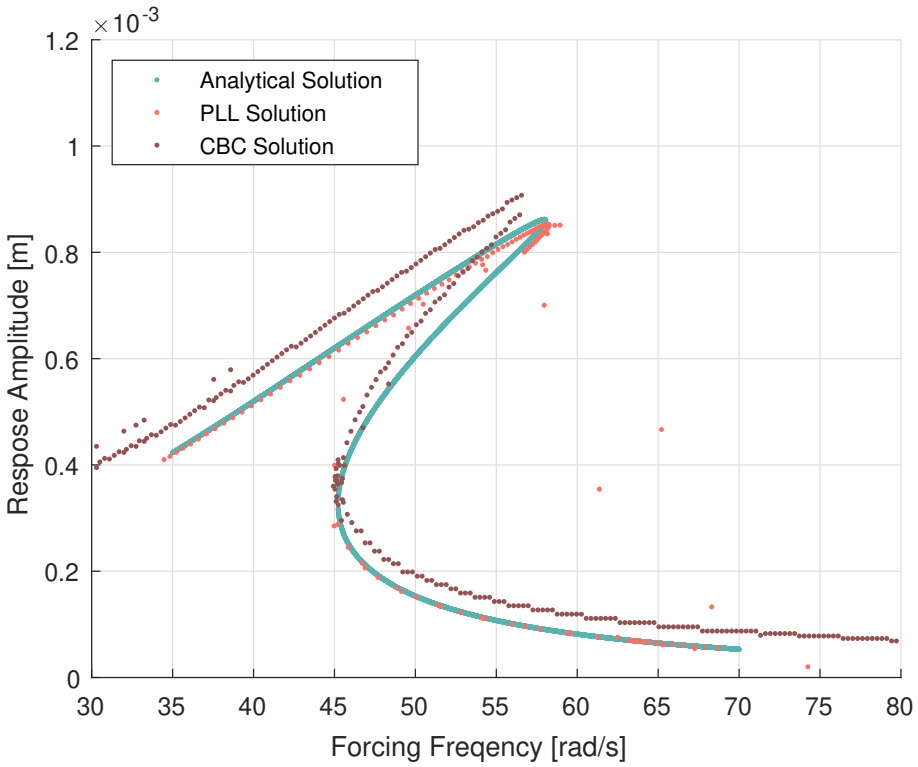


Figure 5.7: Comparison of the frequency response curves extracted from different methods. PLL closely follows the analytical results, while CBC is shifted a little bit.

5.7 Comparison of Backbone Curve Results

The Figure 5.8 shows the comparison between the different backbone curves obtained using the methods discussed before. We can see, the results extracted from the PLL are closest to the analytical solution, with CBC results slightly higher. CBC and CBC-MPC are most likely experiencing the same issue during backbone tracking as it does with FRC tracking, where the control action becomes invasive.

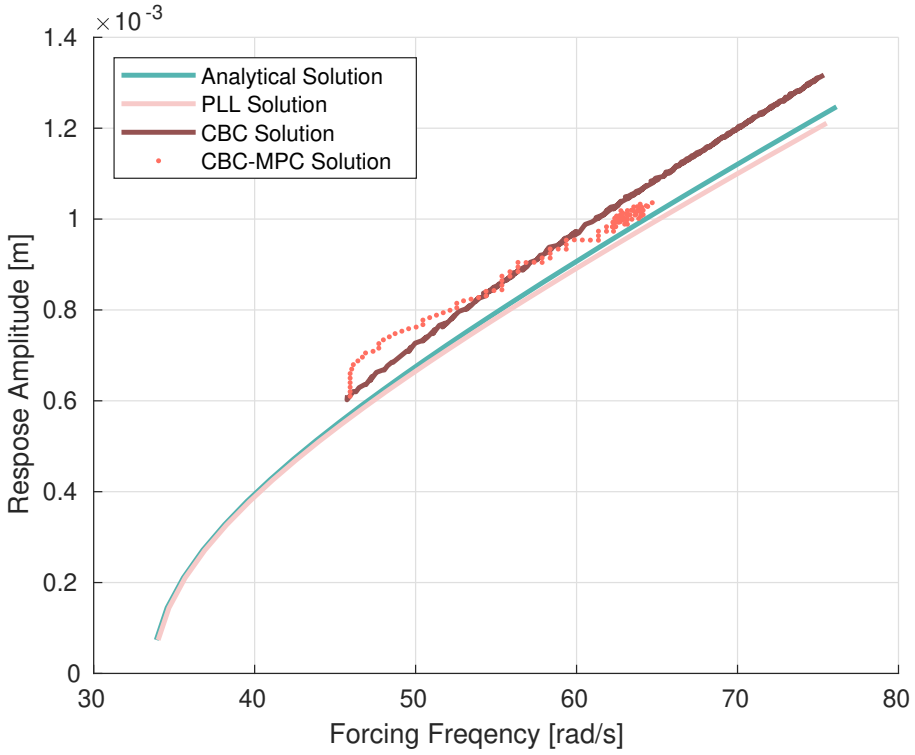


Figure 5.8: Comparison of the backbone curves extracted from different methods. While PLL, CBC follow the analytical curve closely, the CBC-MPC results in noisy results.

6

Results of Excite-Simulink Co-Simulations

The chapter aims to explain the setup of the Excite model and present the simulation results.

6.1 Excite PLL Simulation Model Description

The Excite PLL setup is described in Figure 6.1, which is slightly different from the original PLL as described in Section 4.3.2

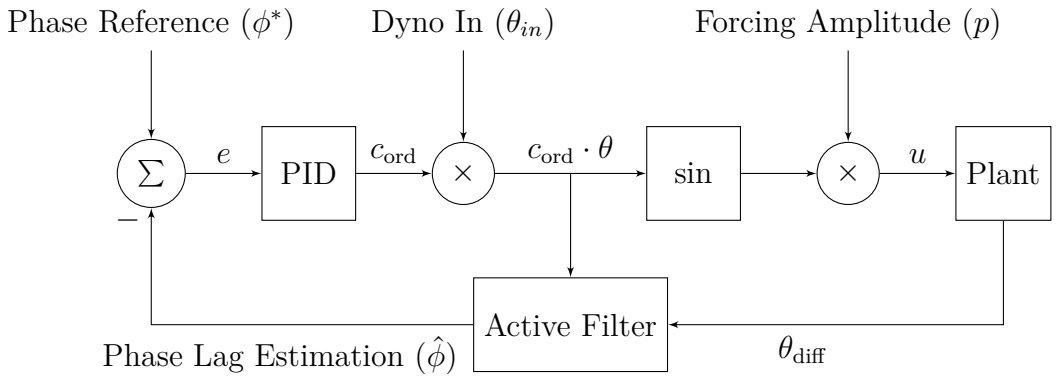


Figure 6.1: PLL setup for excite model

The summary of the changes made is listed below,

- The difference between the input and output shaft speed of the UUT θ_{diff} is used as the feedback signal.
- The controller equation is modified to output control order as described below. Also note that the new control order depends on its previous value. This change gave us better results.

$$c_{\text{ord}} = c_{\text{ord}} + k_{\text{p}}(\phi^* - \hat{\phi}) + k_{\text{i}} \int_0^t (\phi^* - \hat{\phi}) \mathrm{d}t \tag{6.1}$$

- The input for the adaptive filter is the product of DYNO In input shaft speed with the control order ($c_{\text{ord}} \cdot \theta_{\text{in}}$) directly, which removes the need for integration.
- The DYNO Out, output shaft speed is chosen to be 600 [RPM] and the DYNO In, input shaft speed can be derived as follows,

$$
\begin{aligned}
\text{gear ratio} &= 1:1 \\
\text{output shaft speed} \quad (\theta_{\text{out}}) &= 600 \quad [\text{RPM}] \\
\text{input shaft speed} \quad (\theta_{\text{in}}) &= 600 \quad [\text{RPM}] \\
\text{input shaft frequency} \quad (\theta_{\text{in}}) &= 600/60 \quad = \quad 10 \quad [\text{Hz}]
\end{aligned}
$$

6.2 Results

The Excite PLL results for the linear and nonlinear conditions are shown below. The excite model has an input static torque which causes the response signal to shift in the y-axis. This torque is added to engage the gears in the drivetrain.

6.2.1 Linear Conditions

In this case, the nonlinear coefficient is set to zero to test the co-simulation environment. This test also allows for comparing the Excite PLL setup with the analytical results. The chosen parameters for this test are listed in Table 6.1.

Table 6.1: Set of parameters for the Excite PLL experiment

Inertia m [kg · m^2]	Damping c [N · m · s/rad]	Stiffness k [N · m/rad]	Nonlinear stiffness β [N · m/rad^3]	Forcing Amp p [N · m]
1	10	1140	0	5

Proportional Gain k_{p} [1/rad]	Integral Gain k_{i} [s/rad]	Output shaft speed ω_{out} [RPM]
1×10^{-6}	1×10^{-6}	600

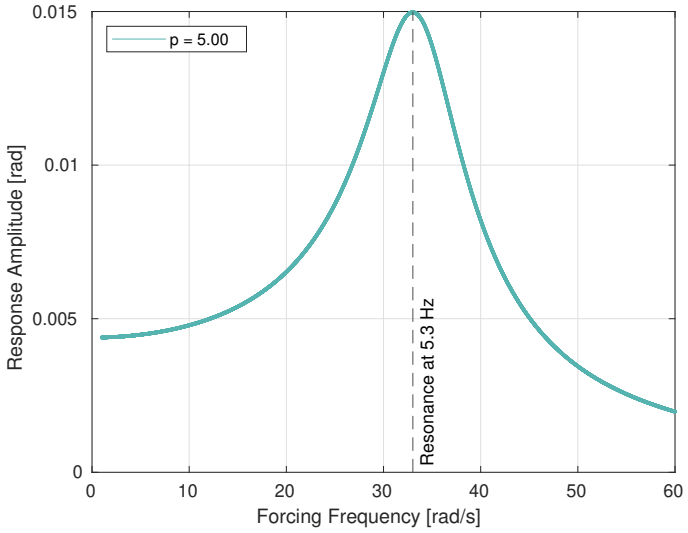
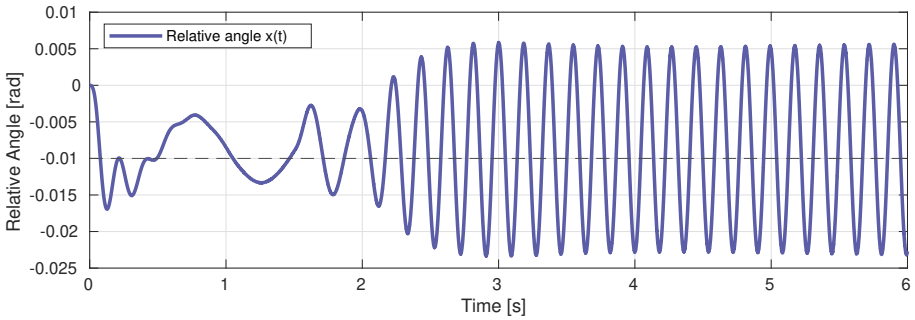


Figure 6.2: Analytical FRC curve for the linear parameters

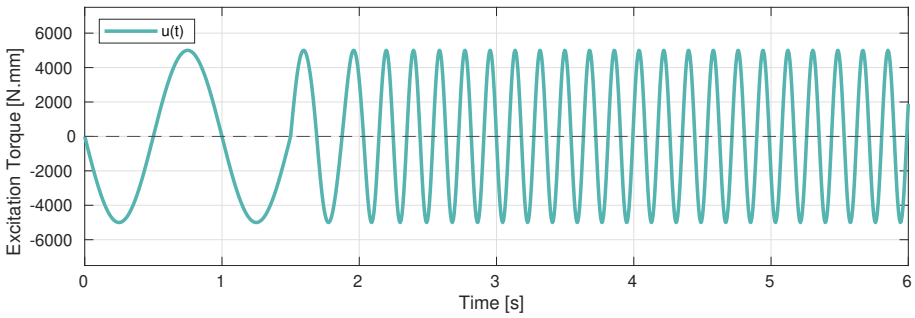
The analytical FRC results for the chosen parameters are shown in the Figure 6.2. As shown in the figure, the resonance is observed to be at 5.3 [Hz]. The control order should converge close to the value derived below,

$$\begin{aligned} \omega &= c_{\text{ord}} \cdot \theta_{\text{in}} \\ c_{\text{ord}} &= \omega/\theta_{\text{in}} \\ c_{\text{ord}} &= 5.3/10 \quad = \quad 0.53 \quad [-] \end{aligned}$$

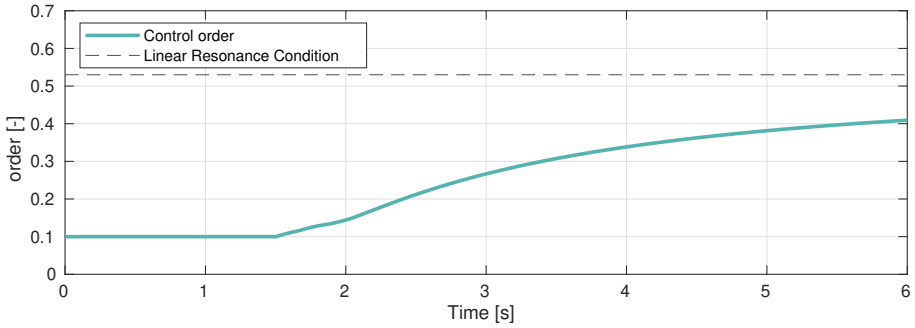
Figure 6.3 shows the results of the experiment. As seen in Figure 6.3c, the control order starts from a subharmonic excitation condition at 0.1 [-], and converges to a slightly lower value than the expected value. We have attempted to tune the system, but the results remain consistent. The controller is configured to start after 1.5 seconds to allow sufficient time for the adaptive filter to converge. This shift can be seen the the control signal Figure 6.3b at the 1.5 [sec] mark.



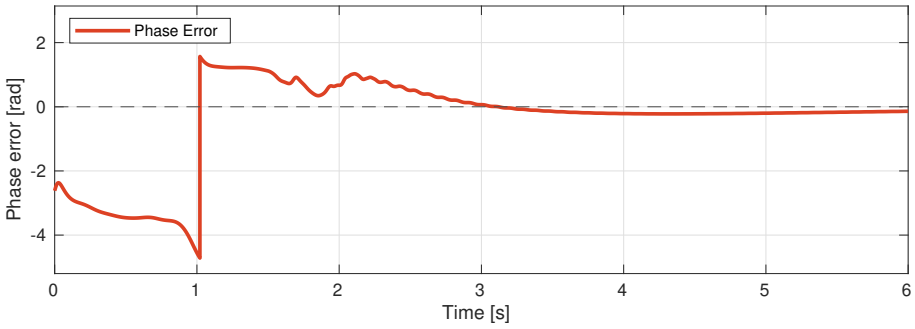
(a) Response signal x



(b) Control signal u



(c) Control order



(d) phase error

Figure 6.3: Excite PLL results for the linear case. The control order c_{ord} is seen to converge to a value close to resonance in (c). The error is also seen to reduce in (d).

6.2.2 Nonlinear Conditions

The parameters for this condition are given below,

Table 6.2: Set of parameters for the Nonlinear Excite PLL experiment

Inertia m [kg · m^2]	Damping c [N · m · s/rad]	Stiffness k [N · m/rad]	Nonlinear stiffness β [N · m/rad^3]	Forcing Amp p [N · m]
1	10	1140	4×10^6	5

Proportional Gain k_p [1/rad]	Integral Gain k_i [s/rad]	Output shaft speed ω_{out} [RPM]
1×10^{-6}	1×10^{-6}	600

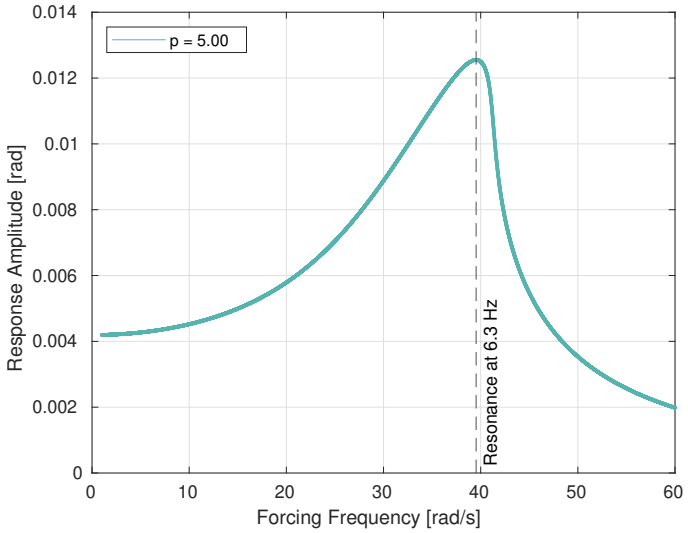
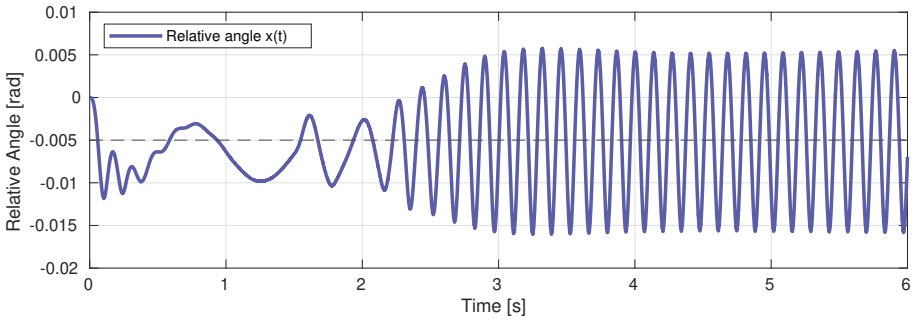


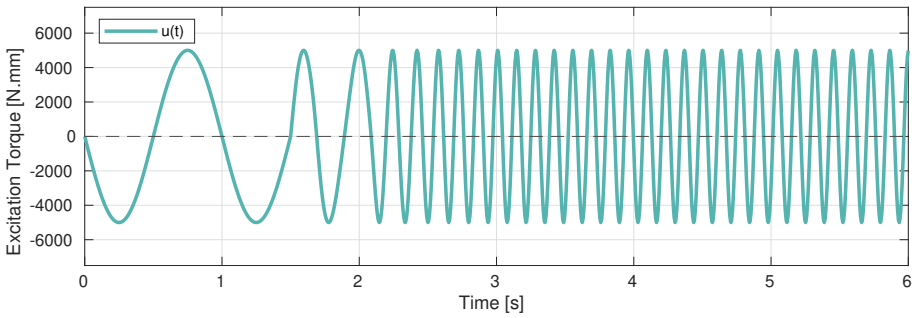
Figure 6.4: Analytical FRC for the nonlinear parameters

The analytical FRC results for the chosen parameters are shown in the Figure 6.4.

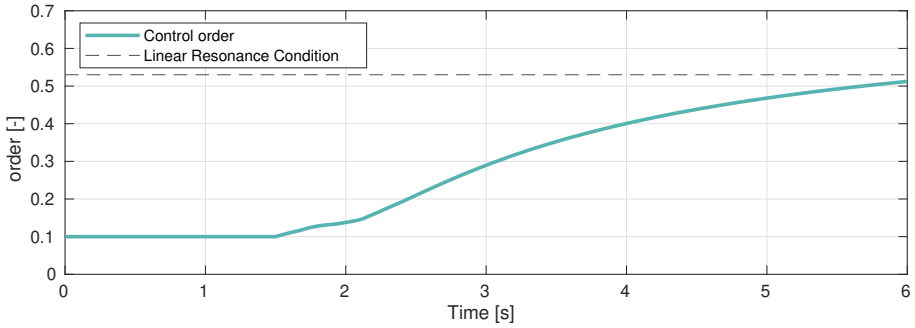
Figure 6.5 shows the results of the experiment. As seen in 6.5b, the control order starts from the same subharmonic excitation condition as before, and the PLL controller is allowed to adjust its values to drive the system towards resonance. The control order converged to a slightly lower value than expected, like before. However, this value is still higher than in the linear condition, as expected.



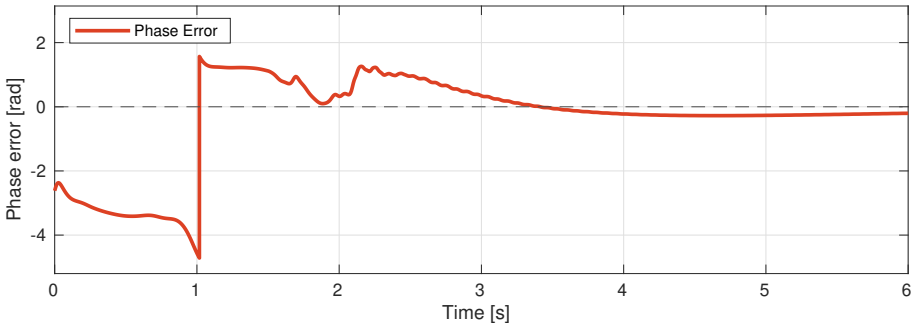
(a) Response signal x



(b) Control signal u



(c) Main order



(d) phase error

Figure 6.5: Excite PLL results for the nonlinear case. The control order c_{ord} is seen to converge to a value close to resonance in (c). The error is also seen to reduce in (d).

6.2.3 Comparison of Backbone Curve Results

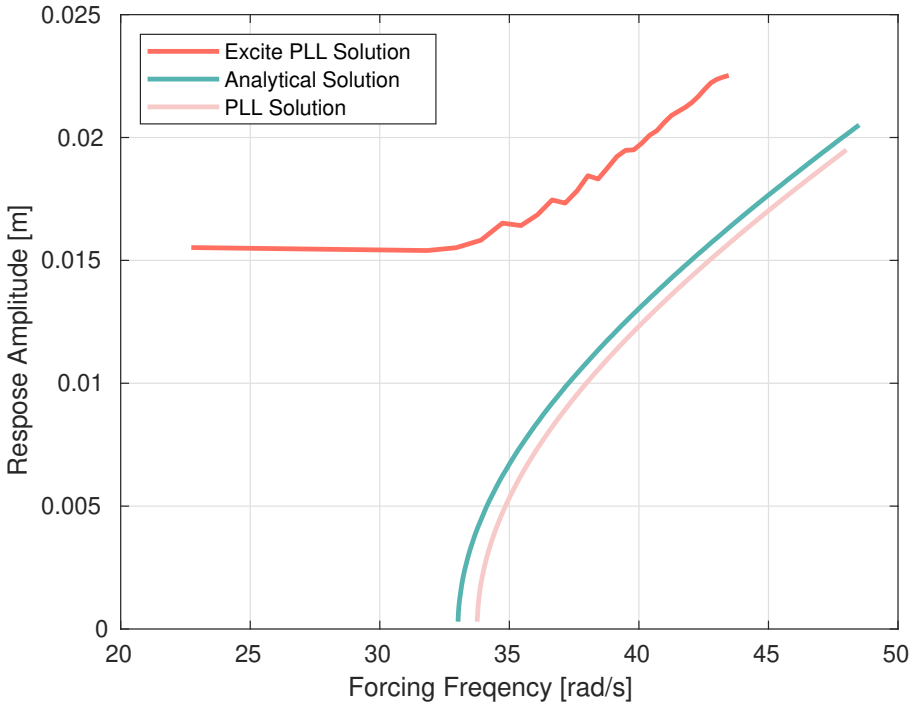


Figure 6.6: Comparison of the backbone curves from Excite, synthetic PLL, and Analytical solution. The PLL closely follows the analytical results, but the Excite-PLL implementation experiences a shift. This behavior can be due to multiple reasons, but currently unknown.

The Figure 6.6 shows the comparison between the backbone curves obtained by different methods. The Excite PLL result experiences a small offset with the analytical results. This could be explained by the fact that the Excite model has more complexity, and although it is modeled close to the analytical system, it is still a different system.

7

Model-Based Optimization Method

Another approach that was explored during this thesis project was to formulate the problem as a nonlinear optimization problem to include just the reference signal and the estimated response signal (generated from the dynamics) as optimizing variables. The requires a little bit of tweaking to the cost function, but has the potential to plot the FRC curves directly from the dynamics, saving time. So, the nonlinear optimizer is used to search for the optimal response amplitude, unlike in MPC, where the nonlinear optimizer is used to find the optimal control. The approach currently assumes that the complete model of the system is known, with the noise not being present in the system. More robust optimization methods can be explored in the future to address these deficiencies.

7.1 Method

The nonlinear optimization method discussed here requires three components: a reference signal, an objective function, and optimization variables, each of which is described in the following subsections.

7.1.1 Reference Signal

First, looking at the Duffing nonlinear equation again

$$m\ddot{x} + c\dot{x} + kx + \beta x^3 = p\sin(\omega t + \phi) \tag{7.1}$$

The reference signal can be approximated as a combination of sines and cosines, like in CBC as discussed in Section 4.3.3, like below,

$$x^* = \sum_{i=0}^{N_\mathrm{H}} a_i \cos(i\omega t) + b_i \sin(i\omega t) \tag{7.2}$$

where N_H can be a tunable parameter determining the number of harmonics chosen to be added in the reference signal. If the system is expected to be close to linear, the value of m can be small as the first-order frequency is dominant in this case; if the system is more nonlinear, the value of m can be higher.

An important point to note is that the first-order coefficients, a_1 and b_1, must be maximized. This requirement is addressed by including them in the objective function.

7.1.2 Objective Function

The structure of the objective function is similar to the one discussed for CBC-MPC in 4.39. This function is chosen to include two parts, the first part deals with the maximization of the first-order components of the reference signal. The second part consists of the error (the difference between the reference and response signal). A small constant γ is added for tuning. The final objective function is given by,

$$\psi(\boldsymbol{w}) = -(a_1^2 + b_1^2) + \sum_{i=1}^{N} \gamma(x_i - x_i^*)^2 \tag{7.3}$$

here, N is the prediction horizon and x_i is determined by the system dynamics and discrete time integration, x_i^* is the reference signal. a_1 and b_1 are the first-order coefficients of the reference signal as discussed in the previous subsection.

7.1.3 Optimization Variables

Listing down the vector of optimization variables,

$$\boldsymbol{v} = \Big[\phi, a_0, a_1, b_1, ..., a_m, b_m, \boldsymbol{x}\Big] \tag{7.4}$$

where ϕ is the phase lag from Equation (7.1), $\boldsymbol{x}$ is the flattened vector of states (positions and velocities). As can be seen in the equation, the optimization variables include the reference signal

7.2 Outer loop

To map the FRC, there needs to be an outer loop to find the optimal response amplitude for every forcing frequency ω. Currently, a simple sweep over a range of ω values is performed.

7.3 Results

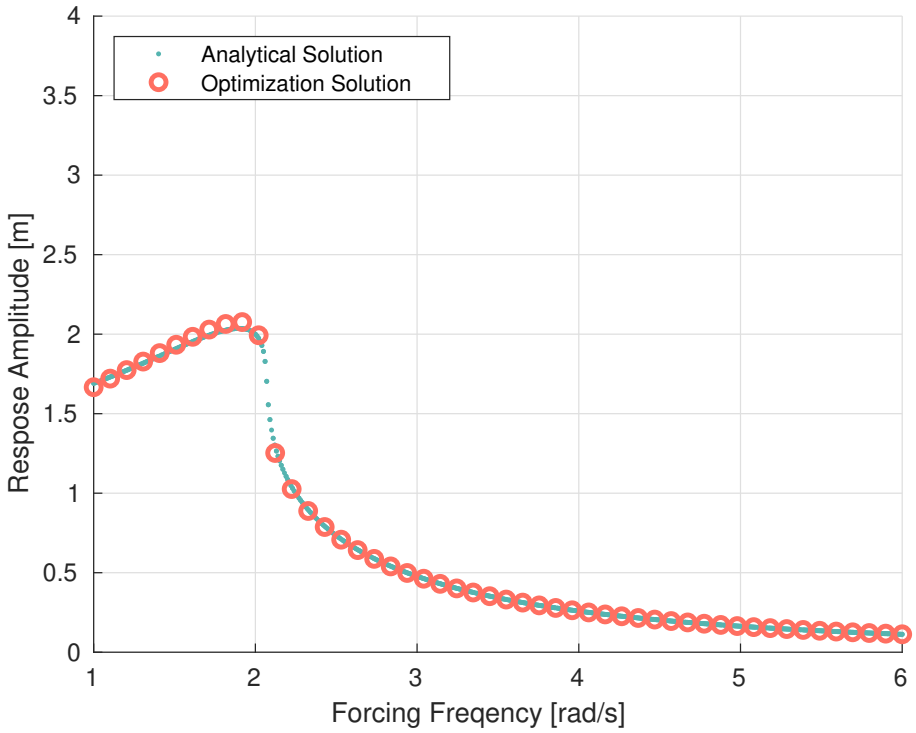
The chosen dynamics for this method are listed below

Table 7.1: Set of parameters for the Optimization method experiment

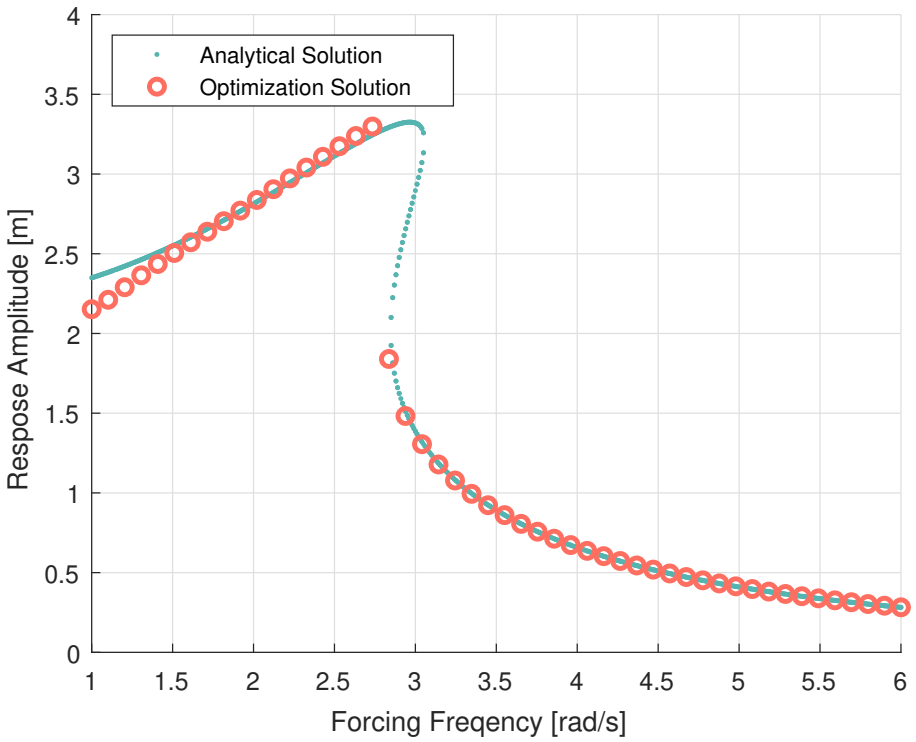
Inertia m [kg]	Damping c [N · s/m]	Stiffness k [N/m]	Nonlinear stiffness β [N/m^3]
1	1	1	1

The results are shown in the Figure 7.1. We can see that the unstable points underneath the FRC are missing as the outer loop of the optimization method is a

simple loop over a range of ω. However, ignoring that, the optimization solution and analytical solutions are really close and suggest that this method can be interesting to explore further in future research. The solutions obtained by this method are also fairly quick, usually taking just a couple of seconds.



(a) Forcing amplitude $p = 4$ [N]



(b) Forcing amplitude $p = 10$ [N]

Figure 7.1: Comparison of FRC between the analytical solution and the optimization solution. Notice the missing points in the unstable region.

8

Discussions and Conclusion

8.1 Discussion on the PLL

With our testing, the PLL controller worked with minimal tuning and gave good results consistently. However, as discussed before, this controller is tricky to implement in multi-DoF systems, and we are unsure how this controller could be scaled up to complex models. The simulations were also completed quickly, with each simulation completing within a few minutes. This made prototyping of the controls easier to iterate on. After incorporating improved initial weights as described in Appendix A, the PLL showed enhanced performance. The adaptive filter settled significantly faster, leading to quicker performance from the PLL.

8.2 Discussion on the CBC

During our testing, the PD controller used in CBC required significant tuning, as it directly influences the system dynamics. This is most likely due to inaccuracies in the estimation of the non-fundamental components by the adaptive filter, which in turn makes the controller invasive. For CBC to function as intended, the estimation of these non-fundamental components must be more accurate. Furthermore, the control architecture is highly sensitive to the initial weights of the adaptive filter and therefore requires a hot start of the weight parameters to achieve reliable performance. The CBC architecture is a promising control system framework for the analysis of structural systems, provided that the estimation of non-fundamental components is sufficiently accurate.

8.3 Discussion on the CBC-MPC

We believe further research in model-based controllers can be more effective as they require less tuning, however, the design of such controllers can get complex. The CBC-MPC controller we proposed in this thesis can be improved a lot further in terms of non-invasiveness, stability, and FRC tracking. We have tried a few variants of MPC by modifying the dynamic equation 4.37 with

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} \begin{bmatrix} \dot{x} \\ -c\dot{x} - kx - \beta x^3 + p\sin(\omega t + u) \end{bmatrix} \tag{8.1}$$

where the control parameter is just the phase of the forcing signal. This controller produced varying results, and the main limiting factor was the fact that the dynamic equation was too restrictive, as it was found out that allowing large transients during the initial time is required to control the system. This can be solved with a couple of interesting ideas, ranging from tweaking the cost function to not include the initial dynamics, allowing new control inputs in the dynamic equation, etc. The limitation of this controller also lies in the fact that it needs the dynamic equation, which is not possible to get for the Excite model at the time of writing. This severely limits the use of this controller to the matlab environment at the moment.

The optimization approach also shows promising results; however, the outer loop needs to be tweaked to loop over different values of ω and not just strictly increasing or strictly decreasing.

8.4 Discussion on nonlinear optimization method

The idea of using the nonlinear optimization solver to solve for both the control signal and the reference was novel to us, and we believe that this technique worked with reasonable success as discussed in Chapter 7. The outer loop can be improved further to find solutions for different forcing frequency values (the folding under the curve). Currently, we do not have any great ideas to achieve this. The simulation time on the other hand, was shorter than the CBC-MPC case as there are fewer computations.

8.5 Discussion on Excite Simulation

The Excite simulation results turned out to be decent, however, we faced a lot of technical challenges in implementing our system. These challenges were mostly regarding the tricky debugging process of the co-simulation environment of Excite-Simulink. However, we believe our implementations and parameters can be used by future researchers to quickly make their models work. The simulation times were also longer in the co-simulation, with simulation times reaching almost 6 hours for an 8-second simulation on a fairly decent CPU. This time was later reduced to around 40 minutes for the same 8-second simulation by tuning the solver settings.

8.6 Answers to research questions

In this thesis, we successfully implemented a feedback controller for a dynamic simulation in both Matlab and Excite. The results were promising, as demonstrated in Figure 6.6. The designed controller was able to identify several key characteristics of the system as intended. The Excite-based controller exhibited a small offset, which is expected given that the Excite model is more complex and not easily constrained to a purely 1-DOF system. Additionally, a model-based MPC controller was implemented. A comparison between the model-based and model-free controllers is shown

in Figure 5.8, where it can be seen that the model-based controller also displays a small offset compared to the others.

8.7 Final thoughts

The research on using control systems to map out nonlinear behavior was exciting, as it is an unfamiliar domain for us, coming from a control background. We hope this thesis could provide a basic introduction to this application of controls and introduce them to the latest research in a systematic manner.

Bibliography

[1] G. Raze, G. Abeloos, and G. Kerschen, "Recent advances in control-based nonlinear vibration testing," in *Proceedings of ISMA2024 and USD2024*. KU Leuven, Leuven, Belgium, 2024.

[2] F. Jung, M. Schröder, and M. Timme, "Exponential adoption of battery electric cars," 2023. [Online]. Available: https://arxiv.org/abs/2306.16152

[3] P. Hippold, M. Scheel, L. Renson, and M. Krack, "Robust and fast backbone tracking via phase-locked loops," *Mechanical Systems and Signal Processing*, vol. 220, p. 111670, Nov. 2024. [Online]. Available: http://dx.doi.org/10.1016/j.ymssp.2024.111670

[4] G. Abeloos, "Control-based methods for the identification of nonlinear structures," Ph.D. dissertation, ULiège - Université de Liège [Sciences Appliquées], Liège, Belgium, 24 November 2022.

[5] T. M. Inc., "Matlab version: 9.11.0 (r2023b)," Natick, Massachusetts, United States, 2023. [Online]. Available: https://www.mathworks.com

[6] AVL, "Avl excite m," 2023. [Online]. Available: https://www.avl.com/en-se/simulation-solutions/software-offering/simulation-tools-a-z/avl-excite-m

[7] V. Arnold, *Mathematical Methods of Classical Mechanics*, 2nd ed., ser. Graduate Texts in Mathematics. New York, NY: Springer-Verlag New York, 1989, vol. 60, originally published by Nauka, Moscow, 1974. [Online]. Available: https://doi.org/10.1007/978-1-4757-2063-1

[8] I. Kovacic, *Nonlinear Oscillations: Exact Solutions and their Approximations*. Springer Cham, 2021.

[9] D. W. Jordan and S. Peter, *Nonlinear Ordinary Differential Equations: An Introduction for Scientists and Engineers*. Oxford University Press, 2007.

[10] M. Li, S. Jain, and G. Haller, "Fast computation and characterization of forced response surfaces via spectral submanifolds and parameter continuation," *Nonlinear Dynamics*, vol. 112, no. 10, p. 7771–7797, Apr. 2024. [Online]. Available: http://dx.doi.org/10.1007/s11071-024-09482-2

[11] J. Guckenheimer and P. Holmes, *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, 7th ed., ser. Applied mathematical sciences. New York: Springer, 2002, no. 42.

[12] S. Tatzko, "Internal resonance in nonlinear structures—and what it can be used for," *PAMM*, vol. 23, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:261978351

[13] A. Hajjaj, N. Jaber, S. Ilyas, F. Alfosail, and M. Younis, "Linear and nonlinear dynamics of micro and nano-resonators: Review of recent advances," *International Journal of Non-Linear Mechanics*, vol. 119, p. 103328,

2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020746219302082

[14] M. Kwarta and M. S. Allen, "Nonlinear normal mode backbone estimation with near-resonant steady state inputs," *Mechanical Systems and Signal Processing*, vol. 162, p. 108046, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327021004362

[15] G. Kerschen, M. Peeters, J. Golinval, and A. Vakakis, "Nonlinear normal modes, part i: A useful framework for the structural dynamicist," *Mechanical Systems and Signal Processing*, vol. 23, no. 1, pp. 170–194, 2009, special Issue: Non-linear Structural Dynamics. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327008001015

[16] M. Krack, L. Panning-von Scheidt, and J. Wallaschek, "On the computation of the slow dynamics of nonlinear modes of mechanical systems," *Mechanical Systems and Signal Processing*, vol. 42, no. 1–2, p. 71–87, Jan. 2014. [Online]. Available: http://dx.doi.org/10.1016/j.ymssp.2013.08.031

[17] A. Nayfeh and D. Mook, *Forced Oscillations of Systems Having a Single Degree of Freedom*. John Wiley & Sons, Ltd, 1995, ch. 4, pp. 161–257.

[18] L. Z. Tahara, "Duffing-harmonic-balance-method," 2022. [Online]. Available: https://github.com/lucaszanov/Duffing-Harmonic-Balance-Method

[19] T. Glad and L. Ljung, *Control Theory*, 1st ed. CRC Press, 2000. [Online]. Available: https://doi.org/10.1201/9781315274737

[20] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar 2006. [Online]. Available: https://doi.org/10.1007/s10107-004-0559-y

[21] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[22] B. Ahmadi, M. W. Mehrez, W. W. Melek, and A. Khajepour, "Model predictive control for reliable path following with application to the autonomous vehicle and considering different vehicle models," in *2021 5th International Conference on Vision, Image and Signal Processing (ICVISP)*, 2021, pp. 27–32.

[23] R. D'Ambrosio, *Runge-Kutta Methods*. Cham: Springer Nature Switzerland, 2023, pp. 109–150. [Online]. Available: https://doi.org/10.1007/978-3-031-31343-1_4

[24] B. Widrow, J. Glover, J. McCool, J. Kaunitz, C. Williams, R. Hearn, J. Zeidler, J. Eugene Dong, and R. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692–1716, 1975.

[25] S. Haykin, *Adaptive Filter Theory*, 3rd ed. New Jersey: Prentice Hall, 1996.

[26] G. Abeloos, L. Renson, C. Collette, and G. Kerschen, "Stepped and swept control-based continuation using adaptive filtering," *Nonlinear Dynamics*, vol. 104, pp. 1–16, 06 2021.

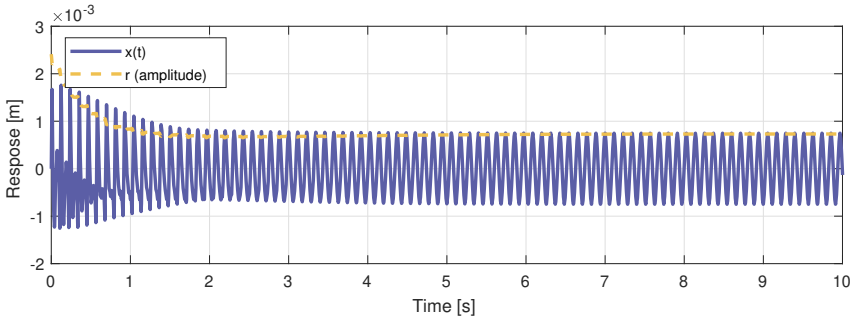
[27] E. Riks, "An incremental approach to the solution of snapping and buckling problems," *International Journal of Solids and Structures*, vol. 15, no. 7, pp. 529–551, 1979. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0020768379900817

[28] M. Crisfield, “A fast incremental/iterative solution procedure that handles “snap-through”,” *Computers & Structures*, vol. 13, no. 1, pp. 55–62, 1981. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0045794981901085

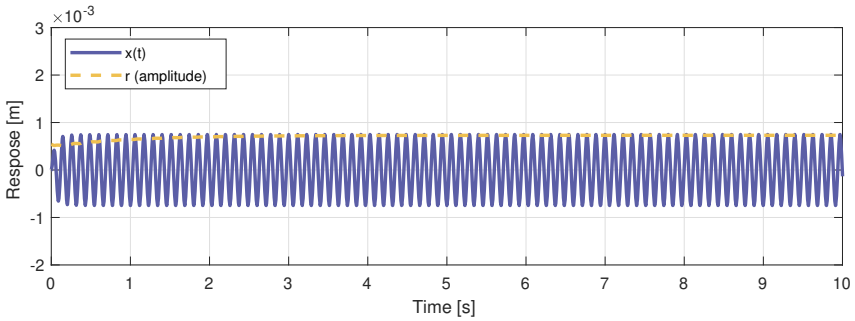
A

Appendix: Tuning of Adaptive filter

Good initial values for the adaptive filter weights are crucial, as they significantly affect the convergence speed of the filter. As shown in Figure A.1, the adaptive filter initialized with optimized weights settles faster, allowing the controller to obtain accurate estimates more quickly. This improved initialization also results in smaller deviations during the initial phase of operation, making the controller produce a more accurate control signal during the initial phase. To determine suitable initial weights, the system was first simulated using a set of guessed weights. After allowing the system to reach a steady state, the final weight values from this initial run were extracted and used as the initial weights for subsequent simulations of the same system.



(a) Response signal x and r with poor initial weights



(b) Response signal x and r with good initial weights

Figure A.1: Comparison of adaptive filter performance with good and poor initial weight estimates

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY