Delft University of Technology
Delft Institute of Applied Mathematics

Mathematical Model of Ventura's Bus Door System

Submitted in partial fulfilment
of the requirements for the degree of

Bachelor of Science

Author:
Lisette de Bruin (4856821)

Supervisor:
Dr. D.J.P. Lahaye

Abstract

After decades of urban growth, mass transport, including buses, will play a significant role in our daily life. Therefore the requirements for buses and their bus door system are increasing. Ventura Systems is a company that is specialized in bus door systems and wishes to gain knowledge on their bus door system using mathematical modeling. This bachelor thesis provides the base for the mathematical models for modeling a bus door system. Multiple models are presented and one model is analyzed in more depth.

Acknowledgements

First of all, I would like to thank my supervisor Dr. D.J.P. Lahaye, whose efforts pushed this bachelor thesis to a higher level. He helped me structure the process and gave numerous insights and feedback. The feedback was extremely helpful. I especially value the fact that he would always allow me to incorporate my ideas and would just provide some minor hints about the direction that I should take. I would also like to thank my friends and family, for their support and interest in my project. Without any obligation to do so, I could always reach out for any sort of unconditional support. This has been a great help while writing this thesis and has enhanced its quality.

Contents

List of Figures

Chapter 1

Introduction

The future sustainable growth of cities and urban areas will increasingly rely on means of mass transport. All major bus manufacturers are therefore studying new designs of mini-vans, city buses, coaches, and mono-rails. This development increases the requirements of the door systems. Bus door systems are expected to become lighter, and more responsive to passenger via sensors and electronics and more adaptable to the market (Lahaye, n.d.). In the design stage of a bus, it is important to understand its response to different maneuvers, such as bumps and emergency braking. Virtual simulations can give worthwhile information and knowledge about the behavior of these vehicles under these different conditions, allowing faster, cheaper, and more precise design. However, numerical models of passenger buses are not straightforward due to the multiple elements that interact with each other and with nonlinear responses, making it harder to predict their behavior to different requests (Teixeira, Moreira, and Tavares, 2015b). Ventura Systems is an example of a bus door manufacturer that wishes to gain such understanding.

Ventura Systems, a Dutch business, is the market leader in innovative door systems for public transport (Systems, n.d.). Met by new challenges from the market, the company is eager to engage in the mathematical modeling and numerical simulation of the products it manufactures. The company wishes to gain an understanding of factors such as weight reduction and the enlargement of glass surfaces impacts factors such as the live cycle of door systems and the production process (Lahaye, n.d.).This project aims to provide the start of the mathematical models which could be of help for Ventura to answer some of the challenges it is faced with. Eventually, an answer will be given to what insights the explained models give. Firstly, a review of past and recent developments in the dynamics of flexible multibody systems is given based on an article by Ahmed A. Shabana. Secondly, the vibration data obtained by Ventura is presented. The data is not included in the Appendix due to the size of the data set. Afterward, the first model is introduced. The one-dimensional single-degree-of-freedom mass-spring system is a simplification of a bus door in its frame. Multiple situations are considered and the corresponding equations of motions are solved. Thereafter, the parameters from these equations of motion are estimated using the previously presented real-life data. In the following chapter, the same differential equations as before are solved using a different method, namely the Laplace transform method. The single-degree-of-freedom mass-spring system is extended to a two-degree-of-freedom mass-spring system in Chapter 7.

Chapter 2

Developments in dynamics of flexible multibody systems

A multibody system shown in Figure 2.1 is a collection of bodies and interconnection elements. Multibody dynamics allows dynamic analysis of interconnected rigid and deformable components (Teixeira, Moreira, and Tavares, 2015a). Flexible multibody dynamics is a rapidly growing field with various applications in vehicle analysis, aerospace engineering, robotics, and biomechanics (Bauchau, 2011). Multibody systems take the dynamic interactions of the different components of the system. The motion of the subsystem is kinematically constrained due to different types of joints (Teixeira, Moreira, and Tavares, 2015a). In 1997, Ahmed A. Shabana wrote a paper on the basic approaches used in computer-aided kinematic and dynamic analysis of flexible mechanic systems. In this chapter, a part of Shabana's paper will be outlined and, if needed, further explained. In Section 2.1, four methods used in the computer-aided kinematic and dynamic analysis of flexible mechanical systems are explained. In the following section, two approaches that can be used for the dynamic analysis of flexible multibody systems are presented.

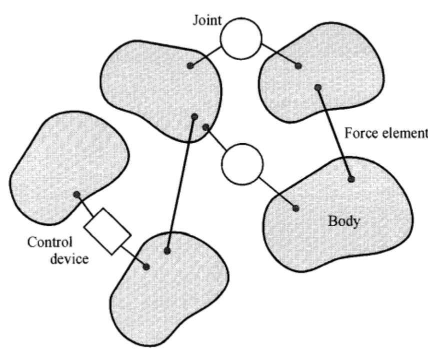


Figure 2.1: Flexible multibody system (Shabana, 1997)

2.1 Methods

For the kinematic description of the motion of deformable bodies that undergo large displacement, multiple methods are used. In this section, the most used methods are pointed out and a short explanation is given.

2.1.1 Floating frame of reference

The most widely used method in computer simulation of flexible multibody systems is the floating frame of reference formulation (FFR). In this formulation, two sets of coordinates are used. The idea is to split the overall motion of bodies that experience small deformation into one frame that describes the location and orientation of a selected body coordinate system, while the other describes the deformation of the body with respect to its coordinate system. The idea is visualized in the figure below.

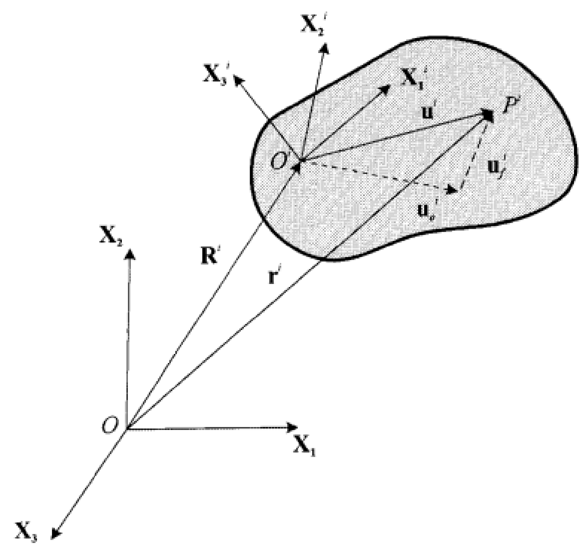


Figure 2.2: Floating frame of reference (Shabana, 1997)

Using FFR kinematics, the global position vector of an arbitrary point in body i can be describes as

$$\vec{r^i} = R^i + A^i(\vec{u}_o^i + \vec{u}_f^i) \tag{2.1}$$

A^i represents the transformation matrix that defines the orientation of the body coordinate system. The vectors $\vec{u^i}$ and $\vec{u}_f^i$) define the position of the point with respect to the FFR (Shabana, 1997). The subscript o refers to the position in the undeformed state, while subscript f refers to the generalized elastic coordination, defined in the local coordinate system (Nada et al., 2010).

2.1.2 Convected coordinate system

The incremental finite element formulations that use the convected coordinate system have been widely used to solve large rotation and deformation problems. There are two types of finite elements that are used in the static and dynamic analysis of deformable bodies, namely the isoparametric and non-isoparametric elements (Shabana, 1997). The term isoparametric is derived from the use of the same shape functions to define the element's geometric shape as are used to define the displacements within the element (Logan, 2007). On the other hand, the non-isoparametric elements use infinite small rotations as nodal coordinates. One of the most widely used computational procedures for non-isoparametric parameters elements is the incremental finite element approach. In the incremental methods, it is assumed that the element shape function can describe small rotations. A convected coordinate system that shares the large rotation of the element is chosen for each element. The incremental finite element formulations do not lead to exact modeling of the rigid body dynamics when the structures rotate as rigid bodies. To solve this, a finite element formulation for the dynamic analysis of multibody systems that consists of interconnected deformable bodies was presented. Four coordinate systems are used to define the finite element configuration (Shabana, 1997). The coordinate systems will be explained with the help of Figure 2.3.

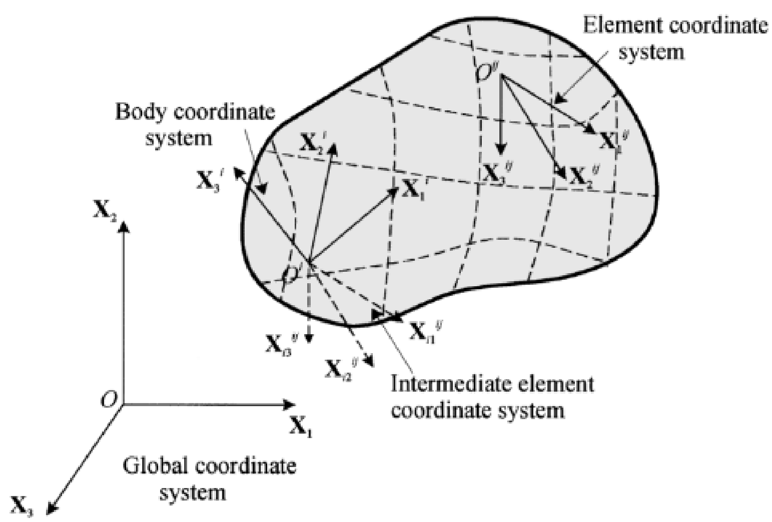


Figure 2.3: Four coordinate systems (Shabana, 1997)

In a multibody system, a deformable body is divided into more than one element. To avoid confusion, some vectors have multiple superscripts. The superscript i to the body number in the multibody system and the superscript j refers to the element number of the deformable body i. First of all, a global coordinate system is employed, which is fixed in time and serves as a unique standard for all deformable bodies (Shabana, 1997). Second of all, the system X_1^i, X_2^i, X_3^i is the floating body coordinate system, which represents the overall motion of the body (Lahaye, personal communication, 2022). This system does not have to be rigidly attached to the origin of the body. Connectivity conditions between the finite elements of this body are defined in the body coordinate system using a Boolean matrix approach (Shabana, 1997). Thirdly, there is an element coordinate system, in Figure 2.3, $X_1^{ij}, X_2^{ij}, X_3^{ij}$, that translates and rotates with the element. The origin of the coordinate system is rigidly attached to a point on the element (Shabana, 2003). Lastly, the system $X_{i1}^{ij}, X_{i2}^{ij}, X_{i3}^{ij}$ is an intermediate element coordinate system. Its origin is rigidly attached to the origin of the body X_1^i, X_2^i, X_3^i coordinate system. The axes of this system are selected in such a way that they are parallel to the axes of the element coordinate system in the undeformed initial configuration (Shabana, 1997).

2.1.3 Finite segment method

In the finite segment method (FSM), the deformable body is assumed to consist of a set of rigid bodies which are connected by springs and/or dampers as shown in Figure 2.4 (Shabana, 1997). An advantage of this method is that it only requires the use of rigid multibody dynamics formulations (Hamper et al., 2012). However, some problems remain to be solved using the FSM. For example, the selection of the number, size, and location of the rigid segments, and the representation of the inertia coupling between these multibody segments (Shabana, 1997).

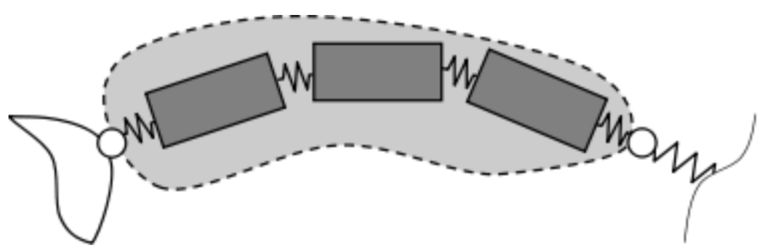


Figure 2.4: Finite segments (Eberhard, n.d.)

2.1.4 Absolute nodal coordinate formulation

As mentioned before, classical beam and plate element shape functions cannot be used to describe large rotations. Another procedure, called the absolute nodal coordinate formulation (ANCF) introduces large displacements of finite elements relative to the global reference frame without using any local frame (Dmitrochenko, 2008). It was demonstrated that this absolute nodal coordinate formulation leads to exact modeling of the rigid body inertia when the structures rotate as rigid bodies. The locations and the deformations of the material points on the finite element are defined in the global coordinate system using the element shape function and the nodal coordinates (Shabana, 1997). The model of the ANCF rope element is shown in the figure below.

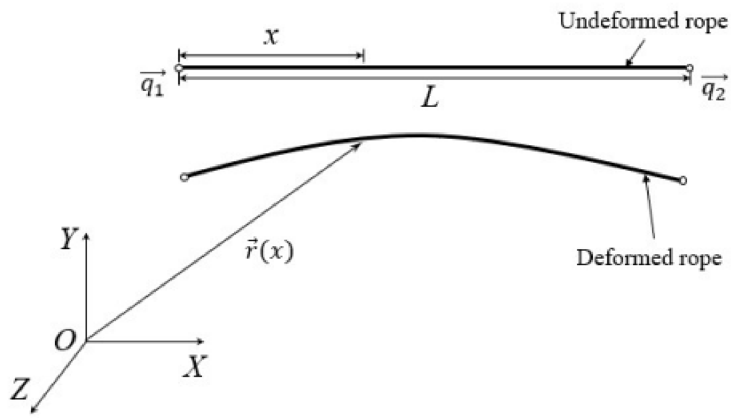


Figure 2.5: Model of ANCF rope element (Luo, Fan, and Cui, 2021)

L is the length of the rope element and x is the coordinate of the element in the length direction. $\vec{q}_1$ and $\vec{q}_2$ are the generalized coordinates of the two nodes of the rope element. $\vec{r}(x)$ is the absolute coordinate of the point on the element, whose coordinate is x. The node coordinates of the gradient default rope beam element are composed of the node position and its derivative to the axial element coordinates. For this case, the coordinates can be expressed as

$$\vec{q}_e = \begin{bmatrix} \vec{q}_1 \\ \vec{q}_2 \end{bmatrix} \tag{2.2}$$

$$= \begin{bmatrix} \vec{r}^T(x=0) \\ \vec{r'}^T(x=0) \\ \vec{r}^T(x=L) \\ \vec{r'}^T(x=L) \end{bmatrix} \tag{2.3}$$

$$= \begin{bmatrix} \vec{e}_1 \\ \vec{e}_2 \\ \vec{e}_3 \\ \vec{e}_4 \end{bmatrix} \tag{2.4}$$

as defined in (Luo, Fan, and Cui, 2021). Now, the global position vector can be expressed as

$$\vec{r} = \begin{bmatrix} s_1(x)I_3 & s_2(x)I_3 & s_3(x)I_3 & s_4(x)I_3 \end{bmatrix} \begin{bmatrix} \vec{e}_1 \\ \vec{e}_2 \\ \vec{e}_3 \\ \vec{e}_4 \end{bmatrix} \tag{2.5}$$

$$\vec{r} = Se \tag{2.6}$$

S is called the element global shape function. Using this method, beam and plate elements can be used to obtain exact modeling of the rigid body dynamics, and these elements can be considered as isoparametric elements (Shabana, 1997).

2.2 Dynamic formulations

Using different methods for kinematic description, various dynamic approaches can be used for formulating the dynamic equations of flexible multibody systems (Shabana, 1997). In this section,

2.2.1 Floating frame of reference

When using the FFR, described in Section 2.1.1, Lagrange's equation can be used to develop the dynamic equations of motion of the deformable bodies that undergo large reference displacements. The equations of motion of a deformable body can be written as follows

$$M^i\ddot{\vec{y}}^i + K^i\vec{y}^i = \vec{q}_e^i + \vec{q}_v^i + \vec{q}_c^i \tag{2.7}$$

as stated in (Shabana, 1997). The superscript i refers to the body number. M is the mass matrix, K is the stiffness matrix, and $\vec{y}$ is the vector of the system generalized coordinates. The vector $\vec{q}_e$ is the vector of externally applied forces, $\vec{q}_v$ is the vector of the Coriolis and centrifugal forces, and $\vec{q}_c$ is the vector of the constraint forces. A kinematic pair is a connection between two bodies that impose constraints on their relative movement, which is the case for flexible multibody systems (Gufler, Wehrle, and Zwölfer, 2021). When these conditions are expressed in analytical form, they are called equations of kinematic constraints (Flores, 2015). The constraint forces divert unconstrained movement to valid movement (Vaxman, 2018). The equations of motion of the multibody system can be written as

$$M\ddot{\vec{y}} + K\vec{y} = \vec{q}_e + \vec{q}_v + \vec{q}_c \tag{2.8}$$

Using this method, $\vec{y}$ can be expressed as follows

$$\vec{y} = \begin{bmatrix} \vec{y}_r \\ \vec{y}_f \end{bmatrix} \tag{2.9}$$

In the case of a rigid body displacement, the elastic coordinates $\vec{y}_f$ are equal to zero (Nada et al., 2010). Using the expression from Equation 2.9, the equations of motion can be written as

$$\begin{bmatrix} M_{rr} & M_{rf} \\ M_{fr} & M_{ff} \end{bmatrix} \begin{bmatrix} \ddot{\vec{y}}_r \\ \ddot{\vec{y}}_f \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_{ff} \end{bmatrix} \begin{bmatrix} \vec{y}_r \\ \vec{y}_f \end{bmatrix} = \begin{bmatrix} (\vec{q}_e)_r \\ (\vec{q}_e)_f \end{bmatrix} + \begin{bmatrix} (\vec{q}_v)_r \\ (\vec{q}_v)_f \end{bmatrix} + \begin{bmatrix} (\vec{q}_c)_r \\ (\vec{q}_c)_f \end{bmatrix} \tag{2.10}$$

The stiffness matrix takes a simple form and it is the same as the stiffness matrix that appears in structural mechanics.

2.2.2 Linear theory of elastodynamics

Before the FFR was introduced, there was another approach to obtaining the total motion of the deformable bodies, namely the linear theory of elastodynamics. In the linear theory of elastodynamics, the rigid body motion and elastic deformation are not solved simultaneously. The assumption is made that the elastic deformation does not have a significant effect on the rigid body displacements, thus the inertia terms in the reference equations are assumed to be independent of the elastic deformation. However, this assumption may not be valid when high-speed, lightweight mechanical systems are considered. The effect of the coupling between the elastic deformation and the gross rigid body motion can be significant (Shabana, 1997). To understand the assumptions better, the equations of motion for a deformable body i shown in Equation 2.10 are rewritten as

$$M^i_{rr}\ddot{\vec{y}}^i_r + M^i_{rf}\ddot{\vec{y}}^i_f = (\vec{q}^i_e)_r + (\vec{q}^i_v)_r \tag{2.11}$$

$$M^i_{fr}\ddot{\vec{y}}^i_r + M^i_{ff}\ddot{\vec{y}}^i_f + K^i_{ff}\ddot{\vec{y}}^i_f = (\vec{q}^i_e)_f + (\vec{q}^i_v)_f \tag{2.12}$$

Notice that the constraint forces do not appear in this equation, since the dependent coordinates are eliminated (Shabana, 2003). In linear theory of elastodynamics, the effect of the deformation on the rigid body displacement is neglected (Shabana, 1997). Thus the equation can be written as a linear system of algebraic equations as follows

$$M^i_{rr}\ddot{\vec{y}}^i_r = (\vec{q}^i_e)_r \tag{2.13}$$

$$M^i_{ff}\ddot{\vec{y}}^i_f + K^i_{ff}\ddot{\vec{y}}^i_f = (\vec{q}^i_e)_f + (\vec{q}^i_v)_f - M^i_{fr}\ddot{\vec{y}}^i_r \tag{2.14}$$

The first equation can be solved using rigid multibody computer programs. The obtained results are substituted into the second equation in order to determine the deformation of the bodies using standard finite element techniques (Shabana, 1997).

Chapter 3

Ventura's data

In this chapter, Ventura's real-life data will be presented and visualized. The data is useful to gain more insight into the simplified model, which will be introduced in the following chapter. First, Fourier analysis will be explained. Thereafter, a short explanation of the data is given and the data is visualized. The mentioned theory is brought into practice by analyzing the data. Both JULIA and PYTHON are used. The code can be found in Appendix A.

3.1 Fourier analysis

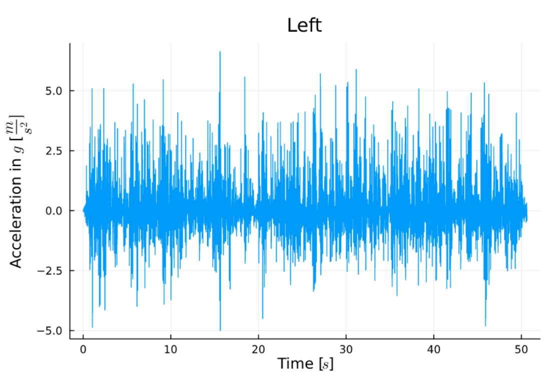
All waveforms are composed of sinusoids with different properties, including their frequency (Sindy, Zandbergen, and Zon, 2021). Until now, the solutions were given with respect to time and are represented in the time domain. However, when dealing with multiple sinusoids, all with different amplitude and frequency, the frequency domain is more compact and useful. Fourier analysis is a translation between these two mathematical worlds. The Fourier Transform transforms a function from the time domain into the frequency domain. The Fourier Transform of a function $f(t)$ is defined as

$$F(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(x) e^{i\omega t} \, dt \tag{3.1}$$

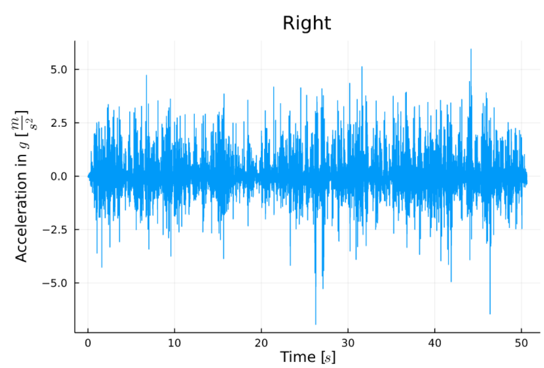
as stated in (Haberman, 2013). The Discrete Fourier Transform (DFT) is the equivalent of the continuous Fourier Transform for signals known only at instants separated by sample times (i.e. a finite sequence of data). The Fast Fourier Transform (FFT) is an algorithm for efficiently computing the DFT of discrete data samples and is used in this chapter to be able to analyze the data (Cochran et al., 1967).

3.2 Test data

The data from a test is described in Sindy's, Zandbergen's, and van Zon's report. In this test, three accelerometers were placed on a bus. Two meters were placed at the bottom center of the aluminum frame of each door. The other one was placed at the top center of the bus portal. These accelerometers measure the accelerations in g (9.80665 $\frac{m}{s^2}$) of the door in the x-, y- and z-direction. Since this research only focuses on the vibration of the door, the data measured by the accelerometer placed at the top center of the bus portal is disregarded (Lahaye, personal communication, 2022). Ventura drove the bus with the three accelerometers over an extremely bumpy road (Sindy, Zandbergen, and Zon, 2021). From a conversation with Ventura, it turns out that the acceleration in the y-direction is most useful and therefore used in this research (Lahaye, personal communication, 2022). The accelerations in the y-direction, measured by the two accelerometers placed on the door frame, are presented below.



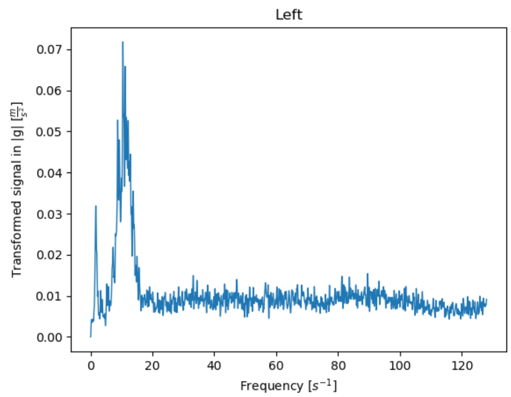
(a) Acceleration measured by the accelerometer on the left door frame



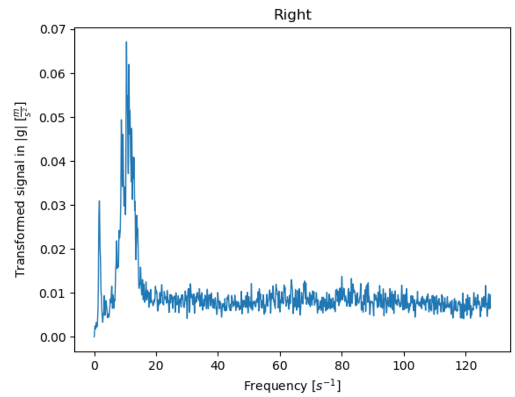
(b) Measured acceleration measured by the accelerometer on the right door frame

Figure 3.1: Acceleration plotted against time

When performing an FFT on these results, the plots contain noise. Therefore, low-pass filtering is used to denoise the data of the FFT and to get more accurate results. A low-pass Butterworth filter allows signals with a frequency lower than the cut-off frequency. All the signals with frequencies more than the cut-off frequency are enervated. Ventura advised a cutoff frequency of 10% of the Nyquist frequency (Sindy, Zandbergen, and Zon, 2021). The Nyquist frequency equals half the sampling frequency (Yamamoto, Yamamoto, and Nagahara, 2016). With help of the function `fft` in Python, the following graphs for the FFTs of the data are found.



(a) FFT of the acceleration measured by the accelerometer on the left door frame



(b) FFT of the acceleration measured by the accelerometer on the right door frame

Figure 3.2: FFTs of the acceleration data

From the graphs, the eigenfrequency of the bus door can be derived. There are clearly visible peaks at approximately 10 Hz in both graphs. From a conversation with Ventura, it can be concluded that these peaks represent the eigenfrequency of the bus. There are also two smaller peaks at approximately 2 Hz. These peaks represent the frequency of the external force exerted by the road on the bus door (Lahaye, personal communication, 2022).

Chapter 4

A single-degree-of-freedom mass-spring system

In this chapter, different situations for a one-dimensional mass-spring system are considered. In real life, the mass would represent the bus door in its frame. For every situation, the equation of motion is stated and a solution is found by plugging in a guess solution. Firstly, there are no external forces included in the system. Afterwards, a constant external force is added. Finally, a periodic external force is taken into consideration. For all three situations, both an undamped as a damped situation is analyzed. The analytically found solution and numerical solution are plotted in JULIA to make sure that the found solution is correct. The initial conditions $u_0 = 3$ and $v_0 = 0$ are assumed to be able to verify the solutions.

4.1 No external forces

A point mass m is connected to a wall by means of a spring with spring constant k. In this chapter, the assumption is made that the point mass rests on a frictionless surface, so gravity does not play a role. Since no external forces are applied, any movement of the mass will be due to the initial conditions. In this chapter, the mass can only move in the x-direction. The initial position is defined as u_0 and the initial velocity is defined as v_0. x_0 defines the equilibrium position of the mass. $u(t)$ denotes the displacement of the mass from its equilibrium position. In this chapter, this notation will be simplified to u in the equations of motion. Remember u still depends on time.
The figure below shows a diagram of a simple system with a mass and a spring.

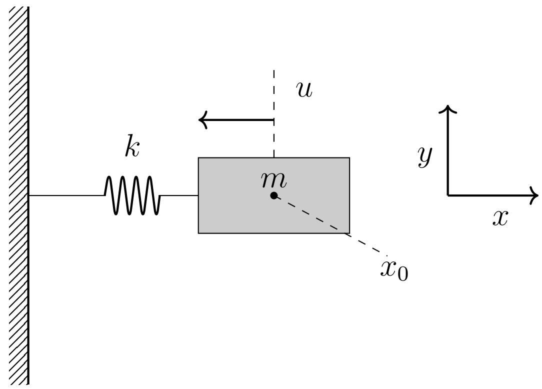


Figure 4.1: A visualization of an undamped mass-spring system without external forces

The spring force acts to restore the spring to its natural length and can be calculated using Hooke's law (Braun, 1991). Hooke's law states that the force is proportional to the extension. A spring will apply an opposing force which is proportional to the extension or compression of the spring. The equation of motion for the mass is

$$m\ddot{u} + ku = 0 \tag{4.1}$$

Since $u(t)$ is an oscillating movement, the movement can be described as $u(t) = F\cos(\omega t)$, but also as the real part of $Fe^{i\omega t} = F(\cos(\omega t) + i\sin(\omega t)$. The reason to do this, is that it is easier to work with an exponential function than with a cosine, since the algebra of exponentials is much easier than that of sines and cosines (Gottlieb and Pfeiffer, 2013).
By substituting $Fe^{i\omega_0 t}$ for u in Equation 4.1 the following equations are obtained

$$-m\omega_0^2 Fe^{i\omega t} + kFe^{i\omega_0 t} = 0 \tag{4.2}$$

$$Fe^{i\omega_0 t}(-m\omega_0^2 + k) = 0 \tag{4.3}$$

$$-m\omega_0^2 + k = 0 \tag{4.4}$$

$$\omega_0 = \sqrt{\frac{k}{m}} \vee \omega_{0'} = -\sqrt{\frac{k}{m}} \tag{4.5}$$

This frequency, ω_0, is known as the angular eigenfrequency or natural angular frequency. The natural frequency of a system, is the frequency for which a systems tends to oscillate in absence of any driving force (COMSOL, 2018). The plus or minus sign determines the direction of the rotation. For this research, there is no need to take both angular frequencies into consideration and therefore, from now on, the negative one is being neglected. From

Equation 4.5, it follows that

$$y(t) = Ae^{i\omega_0 t} \tag{4.6}$$
$$y(t) = A\cos(\omega_0 t) + Ai\sin(\omega_0 t) \tag{4.7}$$
$$y(t) = y_1(t) + iy_2(t) \tag{4.8}$$

is a complex solution for Equation 4.1, where A is an arbitrary complex-valued amplitude. According to Braun, $y_1(t)$ and $y_2(t)$ are two real-valued solutions of Equation 4.1 (Braun, 1991). Thus an expression for $u(t)$ can be obtained

$$u(t) = c_1\cos(\omega_0 t) + c_2\sin(\omega_0 t) \tag{4.9}$$

This method will be used throughout the project, to obtain a real-valued solution. By plugging in the initial conditions, it turns out that $c_1 = u_0$ and $c_2 = \frac{v_0}{w_0}$. From above figure,

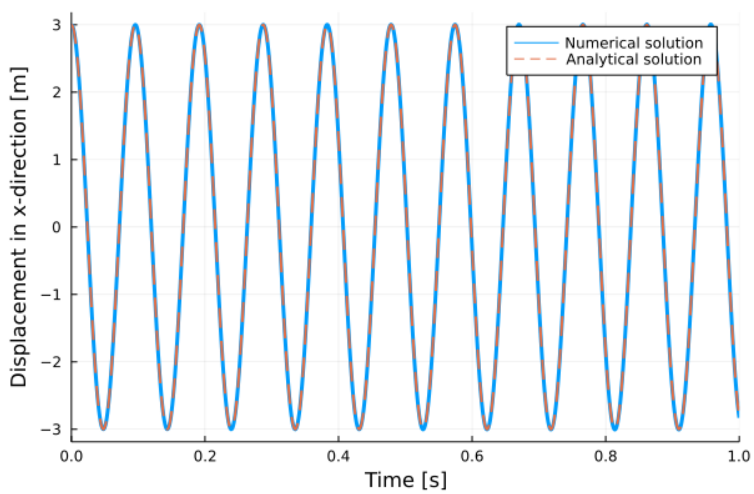


Figure 4.2: A graph of the analytical and numerical solution of the displacement of the mass for the undamped situation without external forces

it can be concluded that the obtained solution is correct.

4.1.1 Damped system without external forces

Now a damper with damping constant c is added to the system. A visualization of this system is presented is the figure below.

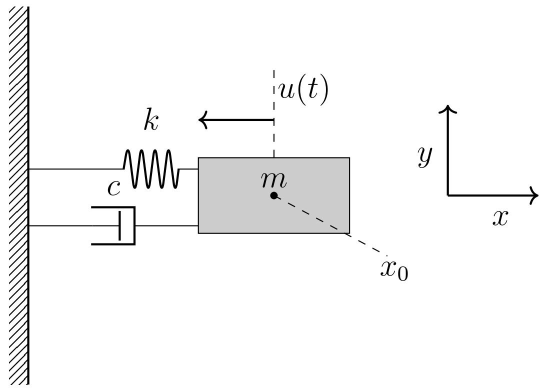


Figure 4.3: A visualization of a damped mass-spring system without external forces

The damping force always acts in the direction opposite the direction of motion, and is proportional to the velocity $\dot{u}$ (Braun, 1991). The equation of motion for this system is

$$m\ddot{u} + c\dot{u} + ku = 0 \tag{4.10}$$

For convenience, complex notation is used again. Consider $u(t)$ of the form $u(t) = Fe^{i\omega t}$ where F is a complex-valued amplitude. Remember $\omega_0 = \sqrt{\frac{k}{m}}$ and use the notation $\zeta = \frac{c}{2\sqrt{km}}$. ζ is called the damping ratio and described how rapidly the oscillations decay from one bounce to the next. Equation 4.10 can be written as

$$-mF\omega^2 e^{i\omega t} + cFi\omega e^{i\omega t} + kFe^{iwt} = 0 \tag{4.11}$$

$$Fe^{i\omega t}(-m\omega^2 + ci\omega + k) = 0 \tag{4.12}$$

$$-m\omega^2 + ci\omega + k = 0 \tag{4.13}$$

$$-\omega^2 + \frac{c}{m}i\omega + \frac{k}{m} = 0 \tag{4.14}$$

$$-\omega^2 + 2i\zeta\omega_0\omega + \omega_0^2 = 0 \tag{4.15}$$

$$\omega_n = \frac{-2i\zeta\omega_0 + \sqrt{w_0^2(-4\zeta^2 + 4)}}{-2} \tag{4.16}$$

$$\omega_n = i\zeta\omega_0 + \omega_0\sqrt{(1-\zeta^2)} \tag{4.17}$$

$$\omega_n = \omega_0(i\zeta + \sqrt{(1-\zeta^2)}) \tag{4.18}$$

The method used in Section 4.1 is used to obtain a real-valued solution. In this way, equation 4.10 can be rewritten as

$$u(t) = e^{-\omega_0\zeta t}(c_1\cos\Big(\omega_0\sqrt{(1-\zeta^2)}t\Big) + c_2\sin\Big(\omega_0\sqrt{(1-\zeta^2)}t\Big)) \tag{4.19}$$

By plugging in the initial condition, it turns out that $c_1 = u_0$ and $c_2 = \frac{v_0 + u_0 w_0 \zeta}{w_0 \sqrt{1-\zeta^2}}$. The expression $\omega_0\sqrt{1-\zeta^2}$ is called the damped natural (angular) frequency (COMSOL, 2018).

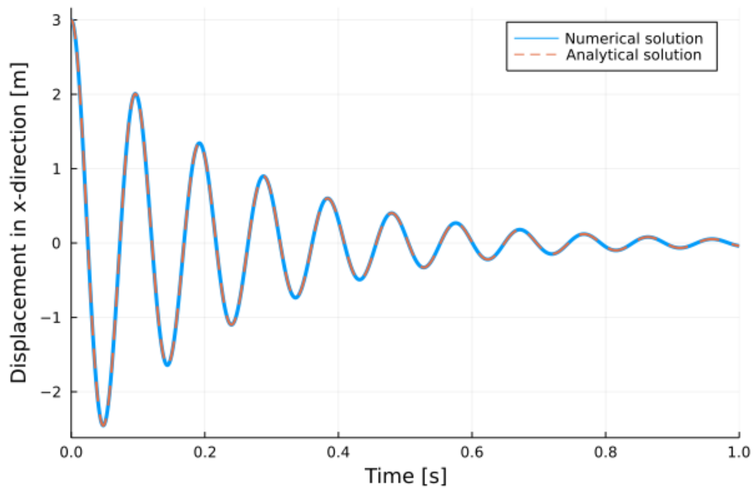


Figure 4.4: A graph of the analytical and numerical solution of the displacement of the mass for the damped situation without external forces

Figure 4.4 shows that the analytical solution coincides with the numerical solution.

4.2 Constant external force

Since the final model will represent a bus door, a realistic constant force would be a person leaning against the bus door. In this case, the external force would be a constant value. Equation 4.1 could be written as

$$m\ddot{u} + ku = F \tag{4.20}$$

where F is a constant. Every solution of Equation 4.20 is of the form $u(t) = u_{homogeneous}(t) + u_{particular}(t)$ (Braun, 1991). A homogeneous solution is already obtained in Section 4.1 and equals $u_{homogeneous}(t) = c_1 \cos(\omega_0 t) + c_2 \sin(\omega_0 t)$. For a particular solution $\hat{u}(t) = A \cdot F$, where A is some constant value, is substituted in Equation 4.20. Thus a value for A can be found

$$k(A \cdot F) = F \tag{4.21}$$

$$A = \frac{1}{k} \tag{4.22}$$

Therefore, the solution for Equation 4.20 equals

$$u(t) = c_1 \cos(\omega_0 t) + c_2 \sin(\omega_0 t) + \frac{F}{k} \tag{4.23}$$

By substituting the initial conditions it turns out that $c_1 = u_0 - \frac{F}{k}$ and $c_2 = \frac{v_0}{\omega_0}$.

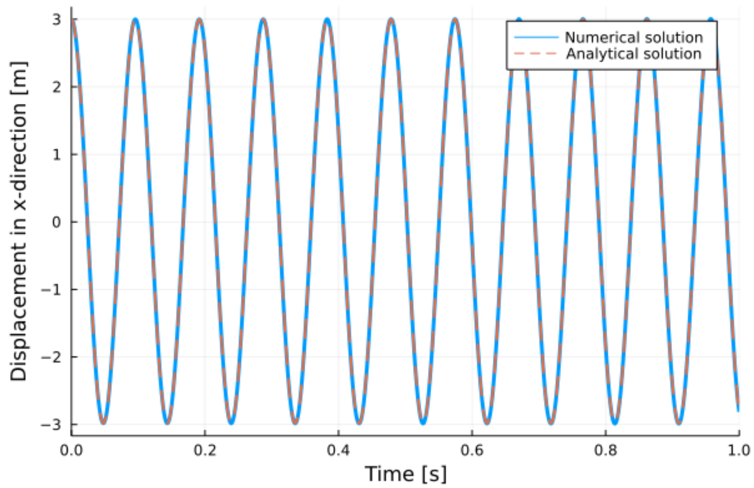


Figure 4.5: A graph of the analytical and numerical solution of the displacement of the mass for the undamped situation including a constant external force

With help of Figure 4.5, the analytical solution is verified to be correct

4.2.1 Damped system with constant external force

Now, the same damper as in Section 4.1.1 is added to this system and the equation of motion equals

$$m\ddot{u} + c\dot{u} + ku = F \tag{4.24}$$

where F represents the constant external force. In Section 4.1.1, the homogeneous solution for Equation 4.23 is already obtained. The particular solution is the same as the particular solution is Section 4.2. So the solution for Equation 4.23 equals

$$u(t) = u_{homogeneous}(t) + u_{particular}(t) \tag{4.25}$$

$$u(t) = e^{-\omega_0 \zeta t}(c_1 \cos\left(\omega_0 \sqrt{(1-\zeta^2)}t\right) + c_2 \sin\left(\omega_0 \sqrt{(1-\zeta^2)}t\right)) + \frac{F}{k} \tag{4.26}$$

Plugging in the initial conditions, it can be concluded that $c_1 = u_0 - \frac{F}{k}$ and $c_2 = \frac{v_0 + \omega_0 \zeta c_1}{w_0 \sqrt{1-\zeta^2}}$.

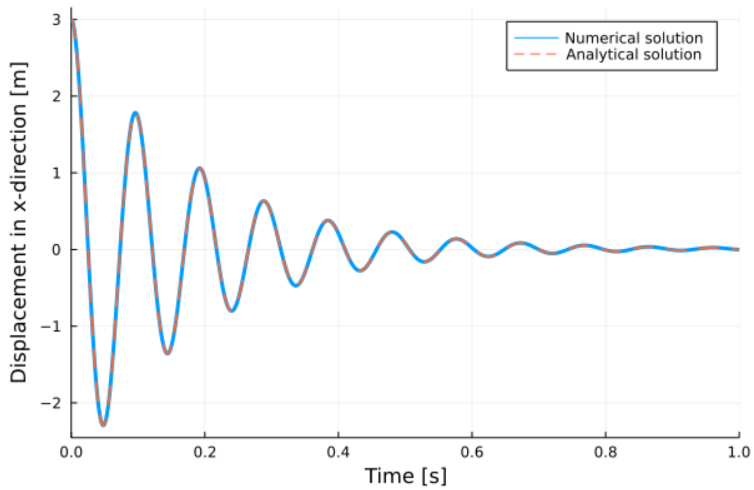


Figure 4.6: A graph of the analytical and numerical solution of the displacement of the mass for the damped situation including a constant external force

The found solution coincides with the numerical solution as showed in the figure above.

4.3 Periodic external force

When a bus is driving on the road, the bumps in the road exert a periodic force on the bus with a certain amplitude, depending on the quality of the road (Sindy, Zandbergen, and Zon, 2021). These bumps can cause movement in the support of the spring. An external force, which is moving the spring up and down, can be written as $f(t) = F\cos(\omega t)$ (Gottlieb and Pfeiffer, 2013). F is the amplitude of the force and ω the angular frequency of the force. Figure 4.7 visualizes this situation.

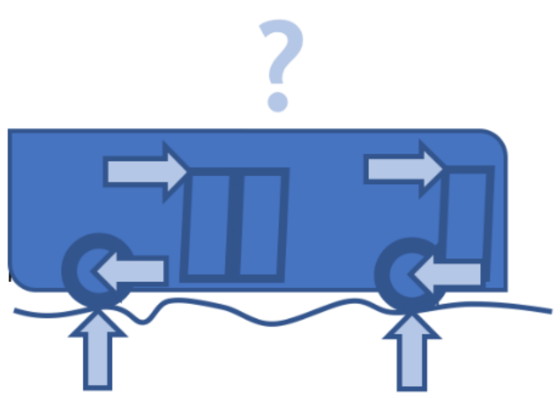


Figure 4.7: A schematic picture of a bus driving on the road (Sindy, Zandbergen, and Zon, 2021)

Adding this force to the system gives the following equation of motion

$$m\ddot{u} + ku = F\cos(\omega t) \tag{4.27}$$

Notice that $f(t)$ is equal to $\mathrm{Re}\{Fe^{i\omega t}\}$. A homogeneous solution is already obtained in Section 4.1 and equals $u_{homogeneous}(t) = c_1\cos(\omega_0 t) + c_2\sin(\omega_0 t)$. A particular solution can be found by using complex exponentials. The problem is divided into two separate problems. First, the assumption that $\omega \neq \omega_0$ is being made. Afterwards, it is assumed that $\omega = \omega_0$.

Case 1: $\omega \neq \omega_0$**.** For a trial solution $\hat{u}(t) = Be^{i\omega t}$ is taken and it is assumed that $\omega \neq \omega_0$. B denotes some complex number. By substituting $\hat{u(t)}$ in Equation 4.27, it turns out that

$$-m\omega^2 Be^{i\omega t} + kBe^{i\omega t} = Fe^{i\omega t} \tag{4.28}$$

$$Be^{i\omega t}(-m\omega^2 + k) = Fe^{i\omega t} \tag{4.29}$$

$$B = \frac{F}{-m\omega^2 + k} \tag{4.30}$$

$$B = \frac{F}{m(\omega_0^2 - \omega^2)} \tag{4.31}$$

$u_{particular}(t) = \mathrm{Re}\{\hat{u}(t)\}$, thus $u_{particular}(t) = \frac{F}{m(\omega_0^2-\omega^2)}\cos(\omega t)$. By adding the homogeneous solution and the particular solution, a solution for the non-homogeneous problem can be obtained.

$$u(t) = c_1\cos(\omega_0 t) + c_2\sin(\omega_0 t) + \frac{F}{m(\omega_0^2 - \omega^2)}\cos(\omega t) \tag{4.32}$$

By plugging in the initial conditions, it turns out that $c_1 = u_0 - \frac{F}{m(\omega_0^2-\omega^2)}$ and $c_2 = \frac{v_0}{\omega_0}$.

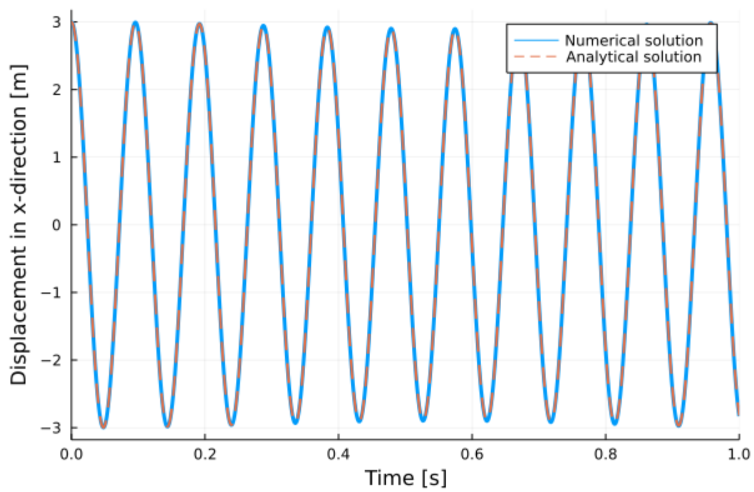


Figure 4.8: A graph of the analytical and numerical solution of the displacement of the mass for the undamped situation including a periodic external force

From Figure 4.8, it can be concluded that the obtained solution is correct.

Case 2: $\omega = \omega_0$**.** The frequency of the external force equals the natural frequency of the system if $\omega = \omega_0$. This case is called the resonance case (Braun, 1991). The solution found in Case 1 does not work if $\omega = \omega_0$, since the denominator would equal zero. To solve this problem, the trial solution is chosen differently. The usual choice is multiplied by t, thus $\hat{u}(t) = Bte^{iwt}$. B is still some complex number. Equation 4.27 is rewritten and the expressions $\hat{u}(t) = Bte^{iwt}$ and $\omega = \omega_0$is substituted, which gives

$$m\ddot{u} + ku = Fe^{i\omega_0 t} \tag{4.33}$$

$$\ddot{u} + \frac{k}{m}u = \frac{F}{m}e^{i\omega_0 t} \tag{4.34}$$

$$2i\omega_0 Be^{i\omega_0 t} - \omega_0^2 Bte^{i\omega_0 t} + \omega_0^2 Bte^{i\omega_0 t} = \frac{F}{m}e^{i\omega_0 t} \tag{4.35}$$

$$2i\omega_0 Be^{i\omega_0 t} = \frac{F}{m}e^{i\omega_0 t} \tag{4.36}$$

$$B = \frac{1}{2i\omega_0}\frac{F}{m} \tag{4.37}$$

$$B = \frac{-iF}{2\omega_0 m} \tag{4.38}$$

$u_{particular}(t) = \text{Re}\{\hat{u}(t)\}$, thus $u_{particular}(t) = \frac{F}{2\omega_0 m}t\sin(\omega_0 t)$. By adding the homogeneous solution and the particular solution, a solution for the non-homogeneous problem can be obtained.

$$u(t) = c_1 \cos(\omega_0 t) + c_2 \sin(\omega_0 t) + \frac{F}{2\omega_0 m} t \sin(\omega_0 t) \tag{4.39}$$

The same method is applied to find the value of c_1 and c_2. For this situation $c_1 = u_0$ and $c_2 = \frac{v_0}{\omega_0}$.

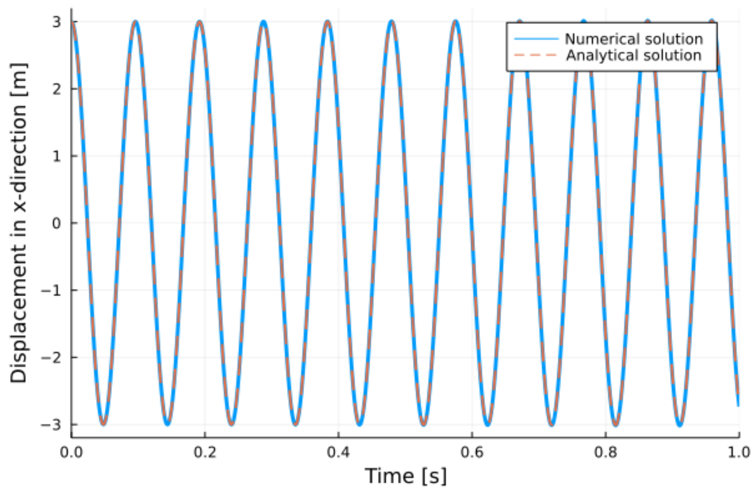


Figure 4.9: A graph of the analytical and numerical solution of the displacement of the mass for the undamped situation including a constant external force with $\omega = \omega_0$

The figure above shows that analytical solution is correct.

4.3.1 Damped system with periodic external force

A damper is added to the system and the equation of motion described in Section 4.3.

$$m\ddot{u} + c\dot{u} + ku = F\cos(\omega t) \tag{4.40}$$

The homogeneous solution for the above equation is already found in Section 4.1.1 and equals $u(t) = e^{-\omega_0 \zeta t}(c_1 \cos\Big(\omega_0 \sqrt{(1-\zeta^2)} t\Big) + c_2 \sin\Big(\omega_0 \sqrt{(1-\zeta^2)} t\Big))$. To find a particular solution for Equation 4.40, the trial solution $\hat{u}(t) = Xe^{\omega i t}$ is substituted for u in the equation of motion, where X denotes some complex number. The following expression for X is obtained

$$-m\omega^2 X e^{i\omega t} + cXi\omega e^{i\omega t} + kXe^{i\omega t} = Fe^{i\omega t} \tag{4.41}$$

$$Xe^{i\omega t}(-m\omega^2 + ci\omega + k) = Fe^{i\omega t} \tag{4.42}$$

$$X(-m\omega^2 + ci\omega + k) = F \tag{4.43}$$

$$X = \frac{F}{-m\omega^2 + ci\omega + k} \tag{4.44}$$

$$X = \frac{F}{-m\omega^2 + ci\omega + k} \cdot \frac{-m\omega^2 - ci\omega + k}{-m\omega^2 - ci\omega + k} \tag{4.45}$$

$$X = \frac{-Fm\omega^2 + Fk - Fci\omega}{m^2\omega^4 - 2mk\omega^2 + k^2 + c^2\omega^2} \tag{4.46}$$

$$X = \frac{-m\omega^2 F + Fk}{(m\omega^2 - k)^2 + c^2\omega^2} - \frac{Fc\omega}{(m\omega^2 - k)^2 + c^2\omega^2} i \tag{4.47}$$

$u_{particular}(t) = \mathrm{Re}\{\hat{u}(t)\}$, thus $u_{particular}(t) = \frac{-m\omega^2 F + Fk}{(m\omega^2 - k)^2 + c^2\omega^2}\cos(\omega t) + \frac{Fc\omega}{(m\omega^2 - k)^2 + c^2\omega^2}\sin(\omega t)$. Adding up the particular solution to the homogeneous solution, a solution for Equation 4.40 can be found.

$$\begin{aligned} u(t) = e^{-\omega_0 \zeta t}(c_1 \cos\Big(\omega_0\sqrt{(1-\zeta^2)}t\Big) + c_2 \sin\Big(\omega_0\sqrt{(1-\zeta^2)}t\Big)) \\ + \frac{-m\omega^2 F + Fk}{(m\omega^2 - k)^2 + c^2\omega^2}\cos(\omega t) + \frac{Fc\omega}{(m\omega^2 - k)^2 + c^2\omega^2}\sin(\omega t) \end{aligned} \tag{4.48}$$

From the initial conditions, it follows that $c_1 = u_0 - \frac{-m\omega^2 F + Fk}{(m\omega^2-k)^2 + c^2\omega^2}$ and $c_2 = \frac{v_0 + \zeta\omega_0 c_1 - \frac{Fc\omega^2}{(m\omega^2-k)^2+c^2\omega^2}}{\omega_0\sqrt{1-\zeta^2}}$.

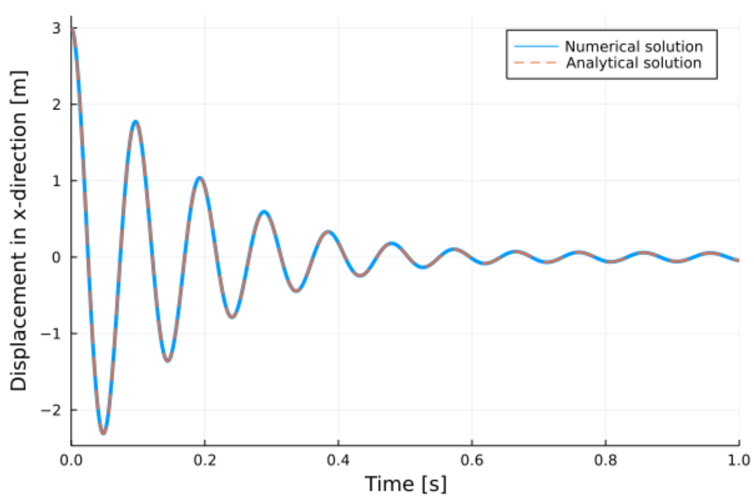


Figure 4.10: A graph of the analytical and numerical solution of the displacement of the mass for the damped situation including an periodic external force

The obtained solution is correct, since the analytical solution coincides with the numerical solution as showed in Figure 4.10.

Chapter 5

Estimating parameters

In this chapter, the spring and damping constant from the simplified model can be determined. This is done with the help of Ventura's data, presented in Chapter 3, and Python. The same code is used as in Chapter 3 and thus can be found in Appendix A. First, the spring constant is calculated using the eigenfrequency. Secondly, the damping constant is estimated using the previously presented FFTs.

5.1 Estimation of the spring constant

The bus door's eigenfrequency equals approximately 10 Hz as mentioned before. The exact eigenfrequency is calculated in Python by taking the average of the peaks' frequencies. This frequency will be denoted f_0. In the same manner, the frequency of the external force exerted on the bus door can be calculated. This frequency equals approximately 1.7 Hz. Using the formula $\omega = 2\pi f$, the angular frequency of the external force can be found. It turns out that the angular frequency of the external force equals about 10.7 Hz.
It appears that a good approximation of the door's mass equals 80 kg (Lahaye, personal communication, 2022). With the help of the eigenfrequency and the mass, an estimation for the spring constant k can be made.

$$\omega_0 = 2\pi f_0 \tag{5.1}$$

$$\sqrt{\frac{k}{m}} = 2\pi f_0 \tag{5.2}$$

$$\sqrt{\frac{k}{80}} = 2\pi \cdot f_0 \tag{5.3}$$

$$k \approx 344531.52 \quad \frac{N}{m} \tag{5.4}$$

The exact value is calculated in PYTHON and is included in the code.

5.2 Estimation of the damping constant

With help of the peak widths of the two FFTs, an estimation for the damping constant c can be made. The real-life system is damped. Previously, the conclusion is made that the external force has an angular frequency. Therefore, the real-life data will be compared with the data obtained by the model described in Section 4.1.1. The solution described in Section 4.3.1 is differentiated twice to obtain a solution for the acceleration of the mass. To be able to compare this solution to the measured data, the solution is divided by the gravity acceleration. Afterward, the FFT of the solution is plotted. Preferably, the peak of the FFT coincides with the peaks of the FFTs shown in Figure 3.2a and 3.2b. By trial and error, it can be concluded that $c = 670$ and $F = 25$ are good estimations for the damping constant and the amplitude of the external force as shown in the figure below.

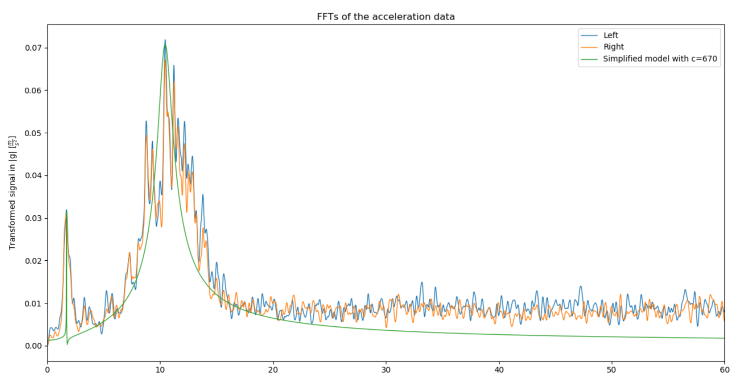


Figure 5.1: A graph of the FFTs of the acceleration measured by the accelerometers and of the simplified model

Chapter 6

Laplace transforms

Using Laplace Transforms is another way to solve differential equations. When solving the differential equation, some characteristics of the system can be determined. First, the Laplace transform is explained and some useful properties are stated. Afterward, all differential equations of the systems from Chapter 4 are solved using this method. To be able to find the solutions for some systems, systems of equations are solved with the help of Matlab. The code can be found in Appendix C.

6.1 Introduction to Laplace transforms

The Laplace transform of a function $f(t)$ is defined as follows:

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^\infty f(t)e^{-st}\,dt \tag{6.1}$$

at least for those s, such that the integral converges (Kazem, 2013). The Laplace transform is very useful for solving differential equations since the Laplace transform of $f'(t)$ is very closely related to the Laplace transform of $f(t)$ (Braun, 1991). This is shown in the lemma below.

Lemma 6.1.1.

$$\begin{aligned}
\mathcal{L}\{f'(t)\} &= \int_0^\infty f'(t)e^{-st}\,dt \\
&= \left[e^{-st}f(t)\right]_0^\infty - \int_0^\infty -se^{-st}\,dt \\
&= -f(0) + s\mathcal{L}\{f(t)\}
\end{aligned}$$

In the same manner, the Laplace transform of $f''(t)$ can be related to $f(t)$.

Lemma 6.1.2.

$$\begin{aligned}\mathcal{L}\{f''(t)\} &= -f'(0) + s\mathcal{L}\{f'(t)\} \\ &= -f'(0) + s(-f(0) + s\mathcal{L}\{f(t)\}) \\ &= -f'(0) - sf(0) + s^2\mathcal{L}\{f(t)\}\end{aligned}$$

To find the solution of a differential equation using Laplace transforms, the following steps are taken. First of all, replace each term in the differential equation with its Laplace transform. Secondly, rearrange the equation to give the transform of the solution and substitute the initial conditions. Third of all, rewrite the equation such that the equation is composed of expressions for which the Laplace inverse is known. Lastly, invert the Laplace transform to obtain the solution. The inversions of the Laplace transform can be found in Appendix B.

6.2 No external forces

Recall that the equation of motion of the undamped free motion equals

$$m\ddot{u} + ku = 0 \tag{6.2}$$

$$\ddot{u} + \frac{k}{m}u = 0 \tag{6.3}$$

Let $U(s) = \mathcal{L}\{u(t)\}$ and substitute this expression is the above equation.

$$\mathcal{L}\{u''(t)\} + \frac{k}{m}\mathcal{L}\{u(t)\} = \mathcal{L}\{0\} \tag{6.4}$$

$$s^2U(s) - su(0) - f'(0) + \frac{k}{m}U(s) = 0 \tag{6.5}$$

$$U(s) = \frac{su(0) + f'(0)}{s^2 + \frac{k}{m}} \tag{6.6}$$

$$U(s) = u(0) \cdot \frac{s}{s^2 + \frac{k}{m}} + \frac{f'(0)}{\sqrt{\frac{k}{m}}} \cdot \frac{\sqrt{\frac{k}{m}}}{s^2 + \frac{k}{m}} \tag{6.7}$$

$$u(t) = u(0)\cos\left(\sqrt{\frac{k}{m}}t\right) + \frac{f'(0)}{\sqrt{\frac{k}{m}}}\sin\left(\sqrt{\frac{k}{m}}t\right) \tag{6.8}$$

In Section 4.1 is described that $\omega_0 = \sqrt{\frac{k}{m}}$. Plugging in the initial conditions and the expression for ω_0, the following solution can be obtained

$$u(t) = u_0 \cos(\omega_0 t) + \frac{v_0}{w_0} \sin(\omega_0 t) \tag{6.9}$$

This is the same solution as the solution found in Section 4.1.

6.2.1 Damped system without external forces

Equation 4.10 is rewritten to the following equation

$$\ddot{u} + \frac{c}{m}\dot{u} + \frac{k}{m}u = 0 \tag{6.10}$$

$U(s) = \mathcal{L}\{u(t)\}$ is substituted in the above equation and the following equation is obtained

$$\mathcal{L}\{u''(t)\} + \frac{c}{m}\mathcal{L}\{u'(t)\} + \frac{k}{m}\mathcal{L}\{u(t)\} = \mathcal{L}\{0\} \tag{6.11}$$

$$s^2U(s) - su(0) - u'(0) + \frac{c}{m}(sU(s) - u(0)) + \frac{k}{m}U(s) = 0 \tag{6.12}$$

$$U(s)[s^2 + \frac{c}{m}s + \frac{k}{m}] = su(0) + u'(0) + \frac{c}{m}u(0) \tag{6.13}$$

$$U(s) = \frac{su(0) + \frac{c}{m}u(0) + u'(0)}{s^2 + \frac{c}{m}s + \frac{k}{m}} \tag{6.14}$$

To simplify the notation the initial condition $u(0)$ and $u'(0)$ are respectively notated as u_0 and v_0. Also, remember from the Sections 4.1 and 4.1.1 that $\omega_0 = \sqrt{\frac{k}{m}}$ and $\zeta = \frac{c}{2\sqrt{km}}$. Thus Equation 6.14 can be rewritten as

$$U(s) = \frac{su_0 + 2\omega_0\zeta u_0 + v_0}{s^2 + 2\omega_0\zeta s + \omega_0^2} \tag{6.15}$$

$$U(s) = \frac{u_0(s + \omega_0\zeta)}{s^2 + 2\omega_0\zeta s + \omega_0^2} + \frac{v_0 + u_0\omega_0\zeta}{s^2 + 2\omega_0\zeta s + \omega_0^2} \tag{6.16}$$

$$U(s) = \frac{u_0(s + \omega_0\zeta)}{s^2 + 2\omega_0\zeta s + \omega_0^2\zeta^2 + \omega_0^2(1 - \zeta^2)} + \frac{v_0 + u_0\omega_0\zeta}{s^2 + 2\omega_0\zeta s + \omega_0^2\zeta^2 + \omega_0^2(1 - \zeta^2)} \tag{6.17}$$

$$U(s) = \frac{u_0(s + \omega_0\zeta)}{(s + \omega_0\zeta)^2 + (\omega_0\sqrt{1 - \zeta^2})^2} + \frac{v_0 + u_0\omega_0\zeta}{\omega_0\sqrt{1 - \zeta^2}} \frac{\omega_0\sqrt{1 - \zeta^2}}{(s + \omega_0\zeta)^2 + (\omega_0\sqrt{1 - \zeta^2})^2} \tag{6.18}$$

$$u(t) = u_0 e^{-\omega_0\zeta t} \cos\Big(\omega_0\sqrt{(1 - \zeta^2)}t\Big) + \frac{v_0 + u_0\omega_0\zeta}{\omega_0\sqrt{1 - \zeta^2}} e^{-\omega_0\zeta t} \sin\Big(\omega_0\sqrt{(1 - \zeta^2)}t\Big)) \tag{6.19}$$

$$u(t) = e^{-\omega_0\zeta t}(u_0 \cos\Big(\omega_0\sqrt{(1 - \zeta^2)}t\Big) + \frac{v_0 + u_0\omega_0\zeta}{\omega_0\sqrt{1 - \zeta^2}} \sin\Big(\omega_0\sqrt{(1 - \zeta^2)}t\Big)) \tag{6.20}$$

The same solution is found in Section 4.1.1.

6.3 Constant external force

The equation of motion from Section 4.2 is rewritten as

$$\ddot{u} + \frac{k}{m}u = \frac{F}{m} \tag{6.21}$$

By taking the Laplace transform of Equation 6.21, the following equations are obtained

$$\mathcal{L}\{u''(t)\} + \frac{k}{m}\mathcal{L}\{u(t)\} = \mathcal{L}\{\frac{F}{m}\} \tag{6.22}$$

$$s^2U(s) - su(0) - u'(0) + \frac{k}{m}U(s) = \frac{F}{m}\frac{1}{s} \tag{6.23}$$

$$U(s)(s^2 + \frac{k}{m}) = \frac{F}{m}\frac{1}{s} + su(0) + u'(0) \tag{6.24}$$

$$U(s) = \frac{\frac{F}{ms}}{s^2 + \frac{k}{m}} + \frac{su(0) + u'(0)}{s^2 + \frac{k}{m}} \tag{6.25}$$

The initial conditions and ω_0 can be substituted in the above equation.

$$U(s) = \frac{\frac{F}{ms}}{s^2 + \omega_0^2} + \frac{su_0 + v_0}{s^2 + \omega_0^2} \tag{6.26}$$

$$U(s) = \frac{\frac{F}{m}}{s(s^2 + \omega_0^2)} + \frac{su_0 + v_0}{s^2 + \omega_0^2} \tag{6.27}$$

To be able to apply the Laplace inverse, fraction decomposition is used.

$$\frac{\frac{F}{m}}{s(s^2 + \omega_0^2)} = \frac{\alpha_1}{s} - \frac{\alpha_2 s + \alpha_3}{s^2 + \omega_0^2} \tag{6.28}$$

$$= \frac{\alpha_1 s^2 + \alpha_1\omega_0^2 - \alpha_2 s^2 - \alpha_3 s}{s(s^2 + \omega_0^2)} \tag{6.29}$$

The following system of equations has to be solved

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & -1 \\ \frac{k}{m} & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{6.30}$$

With the help of the theory known from linear algebra, the following values for the $alpha_i$'s are found

$$\alpha_1 = \frac{m}{k} \tag{6.31}$$
$$\alpha_2 = \frac{m}{k} \tag{6.32}$$
$$\alpha_3 = 0 \tag{6.33}$$

With this information Equation 6.27 can be written as

$$U(s) = \frac{F}{m}(\frac{\frac{m}{k}}{s} - \frac{\frac{m}{k}s}{s^2+\omega_0^2}) + \frac{su_0+v_0}{s^2+\omega_0^2} \tag{6.34}$$
$$U(s) = \frac{\frac{F}{k}}{s} - \frac{\frac{F}{k}s}{s^2+\omega_0^2} + \frac{su_0}{s^2+\omega_0^2} + \frac{v_0}{s^2+\omega_0^2} \tag{6.35}$$
$$U(s) = u_0\frac{s}{s^2+\omega_0^2} - \frac{\frac{F}{k}s}{s^2+\omega_0^2} + \frac{v_0}{\omega_0}\frac{\omega_0}{s^2+\omega_0^2} + \frac{F}{ks} \tag{6.36}$$
$$u(t) = (u_0 - \frac{F}{k})\cos(\omega_0 t) + \frac{v_0}{\omega_0}\sin(\omega_0 t) + \frac{F}{k} \tag{6.37}$$

This appears to be the same solution obtained in Section 4.2

6.3.1 Damped system with constant external force

In this section, the equation of motion for a damped spring with constant external force will be solved with the Laplace transform. The equation of motion for this situation can be written as

$$\ddot{u} + \frac{c}{m}\dot{u} + \frac{k}{m}u = \frac{F}{m} \tag{6.38}$$

Repeating the steps executed before, the following expressions are obtained

$$\mathcal{L}\{u''(t)\} + \frac{c}{m}\mathcal{L}\{u'(t)\} + \frac{k}{m}\mathcal{L}\{u(t)\} = \mathcal{L}\{\frac{F}{m}\} \tag{6.39}$$
$$s^2U(s) - su(0) - u'(0) + \frac{c}{m}(sU(s) - u(0)) + \frac{k}{m}U(s) = \frac{F}{m}\frac{1}{s} \tag{6.40}$$
$$U(s)[s^2 + \frac{c}{m}s + \frac{k}{m}] = su(0) + u'(0) + \frac{c}{m}u(0) + \frac{F}{m}\frac{1}{s} \tag{6.41}$$
$$U(s) = \frac{su(0) + \frac{c}{m}u(0) + u'(0)}{s^2 + \frac{c}{m}s + \frac{k}{m}} + \frac{\frac{F}{m}\frac{1}{s}}{s^2 + \frac{c}{m}s + \frac{k}{m}} \tag{6.42}$$

The initial conditions and ω_0 are plugged into Equation 6.42. Afterwards, partial fraction decomposition is used to obtain the solution.

$$U(s) = \frac{su_0 + \frac{c}{m}u_0 + v_0}{s^2 + \frac{c}{m}s + \omega_0^2} + \frac{\frac{F}{m}\frac{1}{s}}{s^2 + \frac{c}{m}s + \omega_0^2} \tag{6.43}$$

$$\tag{6.44}$$

$$U(s) = \frac{su_0 + \frac{c}{m}u_0 + v_0}{s^2 + \frac{c}{m}s + \omega_0^2} + \frac{\frac{F}{m}}{s(s^2 + \frac{c}{m}s + \omega_0^2)} \tag{6.45}$$

$$\frac{1}{s(s^2 + \frac{c}{m}s + \omega_0^2)} = \frac{\alpha_1}{s} - \frac{\alpha_2 s + \alpha_3}{s^2 + \frac{c}{m}s + \omega_0^2} \tag{6.46}$$

$$= \frac{\alpha_1 s^2 + \alpha_1 \frac{c}{m}s + \alpha_1\omega_0^2 - \alpha_2 s^2 - \alpha_3 s}{s(s^2 + \frac{c}{m}s + \omega_0^2)} \tag{6.47}$$

From the above expression, a matrix equation is obtained.

$$\begin{bmatrix} 1 & -1 & 0 \\ \frac{c}{m} & 0 & -1 \\ \frac{k}{m} & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{6.48}$$

The following values for α_i's are found

$$\alpha_1 = \frac{m}{k} \tag{6.49}$$

$$\alpha_2 = \frac{m}{k} \tag{6.50}$$

$$\alpha_3 = \frac{c}{k} \tag{6.51}$$

By substituting these values the Laplace transform can be simplified

$$U(s) = \frac{su_0 + \frac{c}{m}u_0 + v_0}{s^2 + \frac{c}{m}s + \omega_0^2} + \frac{F}{m}(\frac{\frac{m}{k}}{s} - \frac{\frac{m}{k}s + \frac{c}{k}}{s^2 + \frac{c}{m}s + \frac{k}{m}}) \tag{6.52}$$

$$U(s) = \frac{su_0 + \frac{c}{m}u_0 + v_0}{s^2 + \frac{c}{m}s + \omega_0^2} + \frac{\frac{F}{k}}{s} - \frac{\frac{F}{k}s + \frac{c}{m}\frac{F}{k}}{s^2 + \frac{c}{m}s + \frac{k}{m}} \tag{6.53}$$

$$U(s) = \frac{su_0 + 2\omega_0\zeta u_0 + v_0}{s^2 + \frac{c}{m}s + \omega_0^2} + \frac{\frac{F}{k}}{s} - \frac{\frac{F}{k}s + 2\omega_0\zeta\frac{F}{k}}{s^2 + \frac{c}{m}s + \frac{k}{m}} \tag{6.54}$$

$$U(s) = (u_0 - \frac{F}{k})\frac{(s + \omega_0\zeta)}{(s + \omega_0\zeta)^2 + (\omega_0\sqrt{1 - \zeta^2})^2} + \frac{\omega_0\zeta u_0 + v_0 - \omega_0\zeta\frac{F}{k}}{s^2 + \frac{c}{m}s + \frac{k}{m}} + \frac{\frac{F}{k}}{s} \tag{6.55}$$

$$U(s) = (u_0 - \frac{F}{k})\frac{(s + \omega_0\zeta)}{(s + \omega_0\zeta)^2 + (\omega_0\sqrt{1 - \zeta^2})^2} + \frac{v_0 + \omega_0\zeta(u_0 - \frac{F}{k})}{\omega_0\sqrt{1 - \zeta^2}}\frac{\omega_0\sqrt{1 - \zeta^2}}{(s + \zeta\omega_0)^2 + (\omega_0\sqrt{1 - \zeta^2})^2} + \frac{\frac{F}{k}}{s} \tag{6.56}$$

$$u(t) = e^{-\omega_0\zeta t}((u_0 - \frac{F}{k})\cos\Big(\omega_0\sqrt{(1 - \zeta^2)}t\Big) + \frac{v_0 + \omega_0\zeta(u_0 - \frac{F}{k})}{\omega_0\sqrt{1 - \zeta^2}}\sin\Big(\omega_0\sqrt{(1 - \zeta^2)}t\Big)) + \frac{1}{k}\cdot F \tag{6.57}$$

This is the same solution as found in Section 4.2.1.

6.4 Periodic external force

Recall that the equation of motion for the undamped situation including a periodic external force equals

$$m\ddot{u} + ku = F\cos(\omega t) \tag{6.58}$$

$$\ddot{u} + \frac{k}{m}u = \frac{F}{m}\cos(\omega t) \tag{6.59}$$

The Laplace transform is substituted to obtain

$$\mathcal{L}\{u''(t)\} + \frac{k}{m}\mathcal{L}\{u(t)\} = \mathcal{L}\{\frac{F}{m}\cos(\omega t)\} \tag{6.60}$$

$$s^2U(s) - su(0) - u'(0) + \frac{k}{m}U(s) = \frac{F}{m}\frac{s}{s^2 + \omega^2} \tag{6.61}$$

$$U(s)(s^2 + \frac{k}{m}) = \frac{F}{m}\frac{s}{s^2 + \omega^2} + su(0) + u'(0) \tag{6.62}$$

$$U(s) = \frac{F}{m}\frac{s}{(s^2 + \omega^2)(s^2 + \frac{k}{m})} + \frac{su(0) + u'(0)}{s^2 + \frac{k}{m}} \tag{6.63}$$

$$\tag{6.64}$$

The initial conditions and the expression for ω_0 are used to rewrite Equation 6.64 to

$$U(s) = \frac{F}{m}\frac{s}{(s^2+\omega^2)(s^2+\omega_0^2)} + \frac{su_0+v_0}{s^2+\omega_0^2} \tag{6.65}$$

$$\tag{6.66}$$

Once again, two different situations have to been discussed. Firstly, it is assumed that $\omega \neq \omega_0$. Secondly, the assumption that $\omega = \omega_0$

Case 1: $\omega \neq \omega_0$ Equation 6.66 can be written as

$$U(s) = \frac{F}{m}\frac{s}{(s^2+\omega^2)(s^2+\omega_0^2)} + \frac{su_0}{s^2+\omega_0^2} + \frac{v_0}{s^2+\omega_0^2} \tag{6.67}$$

To be able to find the Laplace inverse of this expression, partial fraction decomposition is used.

$$\frac{s}{(s^2+\omega^2)(s^2+\omega_0^2)} = \frac{s^2+\alpha_1 s+\alpha_2}{s^2+\omega^2} - \frac{s^2+\alpha_3 s+\alpha_4}{s^2+\omega_0^2} \tag{6.68}$$

$$= \frac{s^4+\omega_0^2 s^2+\alpha_1 s^3+\alpha_1\omega_0^2 s+\alpha_2 s^2+\alpha_2\omega_0^2 - s^4-\omega^2 s^2-\alpha_3 s^3-\alpha_3\omega^2 s-\alpha_4 s^2 -}{(s^2+\omega^2)(s^2+\omega_0^2)} \tag{6.69}$$

This gives us a system of equations that has to be solved.

$$\begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ \omega_0^2 & 0 & -\omega^2 & 0 \\ 0 & w_0^2 & 0 & -\omega^2 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} 0 \\ -\omega_0^2+\omega^2 \\ 1 \\ 0 \end{bmatrix} \tag{6.70}$$

With the help of Matlab, the values for the α_i's are obtained.

$$\alpha_1 = \frac{m}{-m\omega^2+k} = \frac{-1}{\omega^2-\omega_0^2} \tag{6.71}$$

$$\alpha_2 = \omega^2 \tag{6.72}$$

$$\alpha_3 = \frac{m}{-m\omega^2+k} = \frac{-1}{\omega^2-\omega_0^2} \tag{6.73}$$

$$\alpha_4 = \frac{k}{m} = \omega_0^2 \tag{6.74}$$

Thus Equation 6.67 can be rewritten as

$$U(s) = \frac{F}{m}(\frac{s^2+\alpha_1 s+\alpha_2}{s^2+\omega^2} - \frac{s^2+\alpha_3 s+\alpha_4}{s^2+\omega_0^2}) + \frac{su_0}{s^2+\omega_0^2} + \frac{v_0}{s^2+\omega_0^2} \tag{6.75}$$

$$U(s) = \frac{F}{m}(1+\frac{\alpha_1 s}{s^2+\omega^2} - 1 + \frac{\alpha_3 s}{s^2+\omega_0^2}) + \frac{su_0}{s^2+\omega_0^2} + \frac{v_0}{\omega_0}\frac{\omega_0}{s^2+\omega_0^2} \tag{6.76}$$

$$u(t) = \frac{F}{m}(\alpha_1\cos(\omega t) + \alpha_3\cos(\omega_0 t)) + u_0\cos(\omega_0 t) + \frac{v_0}{\omega_0}\sin(\omega_0 t) \tag{6.77}$$

$$u(t) = (u_0 - \frac{F}{m(\omega_0^2-\omega^2)})\cos(\omega_0 t) + \frac{v_0}{\omega_0}\sin(\omega_0 t) + \frac{F}{m(\omega_0^2-\omega^2)}\cos(\omega t) \tag{6.78}$$

Case 2: $\omega = \omega_0$ If $\omega = \omega_0$, the solution differs from the solution found in the first case. Equation 6.66 can be written as

$$U(s) = \frac{F}{2\omega_0 m}\frac{2\omega_0 s}{(s^2+\omega_0^2)^2} + \frac{su_0+v_0}{s^2+\omega_0^2} \tag{6.79}$$

$$U(s) = \frac{u_0 s}{s^2+\omega_0^2} + \frac{v_0}{\omega_0}\frac{\omega_0}{s^2+\omega_0^2} + \frac{F}{2\omega_0 m}\frac{2\omega_0 s}{(s^2+\omega_0^2)^2} \tag{6.80}$$

$$u(t) = u_0\cos(\omega_0 t) + \frac{v_0}{\omega_0}\sin(\omega_0 t) + \frac{F}{2\omega_0 m}t\sin(\omega_0 t) \tag{6.81}$$

These two solutions correspond with the solutions found in Chapter 4.

6.4.1 Damped system with periodic external force

The damping factor is added to Equation 6.59.

$$\ddot{u} + +\frac{c}{m}\dot{u} + \frac{k}{m}u = \frac{F}{m}\cos(\omega t) \tag{6.82}$$

The Laplace transform is substituted to obtain

$$\mathcal{L}\{u''(t)\} + \frac{c}{m}\mathcal{L}\{u'(t)\} + \frac{k}{m}\mathcal{L}\{u(t)\} = \mathcal{L}\{\frac{F}{m}\cos(\omega t)\} \tag{6.83}$$

$$s^2U(s) - su(0) - u'(0) + \frac{c}{m}sU(s) - \frac{c}{m}u(0) + \frac{k}{m}U(s) = \frac{F}{m}\frac{s}{s^2+\omega^2} \tag{6.84}$$

$$U(s)(s^2 + \frac{c}{m}s + \frac{k}{m}) = \frac{F}{m}\frac{s}{s^2+\omega^2} + su(0) + u'(0) + \frac{c}{m}u(0) \tag{6.85}$$

$$U(s) = \frac{F}{m}\frac{s}{(s^2+\omega^2)(s^2+\frac{c}{m}s+\frac{k}{m})} \tag{6.86}$$

$$+ \frac{su(0) + u'(0) + \frac{c}{m}u(0)}{s^2+\frac{c}{m}s+\frac{k}{m}} \tag{6.87}$$

Partial fraction decomposition is applied to divide the first fraction into two separate fractions.

$$\frac{s}{(s^2+\omega^2)(s^2+\frac{c}{m}s+\frac{k}{m})} = F_1 - F_2 \tag{6.88}$$

$$= \frac{s^2+\alpha_1 s+\alpha_2}{s^2+\omega^2} - \frac{s^2+\alpha_3 s+\alpha_4}{s^2+\frac{c}{m}s+\frac{k}{m}} \tag{6.89}$$

$$\begin{bmatrix} 1 & 0 & -1 & 0 \\ \frac{c}{m} & 1 & 0 & -1 \\ \frac{k}{m} & \frac{c}{m} & -\omega^2 & 0 \\ 0 & \frac{k}{m} & 0 & -\omega^2 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} -\frac{c}{m} \\ -\frac{k}{m}+\omega^2 \\ 1 \\ 0 \end{bmatrix} \tag{6.90}$$

With the help of Matlab, the α_i's can be found. To simplify these expressions, the constants $q = c^2\omega^2 + k^2 - 2km\omega^2 + m^2\omega^4$ is introduced.

$$\alpha_1 = \frac{-m^2\omega^2 + km}{c^2\omega^2 + k^2 - 2km\omega^2 + m^2\omega^4} = \frac{-m^2\omega^2 + km}{q} \tag{6.91}$$

$$\alpha_2 = \frac{c^2\omega^4 + cm\omega^2 + k^2\omega^2 - 2km\omega^4 + m^2\omega^6}{c^2\omega^2 + k^2 - 2km\omega^2 + m^2\omega^4} = \frac{c^2\omega^4 + cm\omega^2 + k^2\omega^2 - 2km\omega^4 + m^2\omega^6}{q} \tag{6.92}$$

$$\alpha_3 = \frac{c^3\omega^2 + ck^2 - 2ckm\omega^2 + km^2 + cm^2\omega^4 - m^3\omega^2}{m(c^2\omega^2 + k^2 - 2km\omega^2 + m^2\omega^4)} = \frac{c^3\omega^2 + ck^2 - 2ckm\omega^2 + km^2 + cm^2\omega^4 - m^3\omega^2}{mq} \tag{6.93}$$

$$\alpha_4 = \frac{k(c^2\omega^2 + cm + k^2 - 2km\omega^2 + m^2\omega^4)}{m(c^2\omega^2 + k^2 - 2km\omega^2 + m^2\omega^4)} = \frac{k(c^2\omega^2 + cm + k^2 - 2km\omega^2 + m^2\omega^4)}{mq} \tag{6.94}$$

Firstly, the values for α_1 and α_2 are plugged into the expression for F_1.

$$F_1 = \frac{s^2 + (\frac{-m^2\omega^2+km}{q})s + \frac{c^2\omega^4+cm\omega^2+k^2\omega^2-2km\omega^4+m^2\omega^6}{q}}{s^2+\omega^2} \tag{6.95}$$

$$= \frac{s^2 + (\frac{-m^2\omega^2+km}{q})s + \frac{\omega^2 q+cm\omega^2}{q}}{s^2+\omega^2} \tag{6.96}$$

$$= \frac{s^2 + (\frac{-m^2\omega^2+km}{q})s + \omega^2 + \frac{cm\omega^2}{q}}{s^2+\omega^2} \tag{6.97}$$

$$= 1 + \frac{(\frac{-m^2\omega^2+km}{q})s + \frac{cm\omega^2}{q}}{s^2+\omega^2} \tag{6.98}$$

The same is done to simplify the expression for F_2.

$$F_2 = \frac{s^2 + (\frac{c^3\omega^2+ck^2-2ckm\omega^2+km^2+cm^2\omega^4-m^3\omega^2}{mq})s + (\frac{k(c^2\omega^2+cm+k^2-2km\omega^2+m^2\omega^4)}{mq})}{s^2 + \frac{c}{m}s + \frac{k}{m}} \tag{6.99}$$

$$= \frac{s^2 + (\frac{cq+km^2-m^3\omega^2}{mq})s + (\frac{kq+kcm}{mq})}{s^2 + \frac{c}{m}s + \frac{k}{m}} \tag{6.100}$$

$$= \frac{s^2 + \frac{c}{m}s + \frac{km-m^2\omega^2}{q}s + \frac{k}{m} + \frac{kc}{q}}{s^2 + \frac{c}{m}s + \frac{k}{m}} \tag{6.101}$$

$$= 1 + \frac{\frac{km-m^2\omega^2}{q}s + \frac{kc}{q}}{s^2 + \frac{c}{m}s + \frac{k}{m}} \tag{6.102}$$

Using these expressions, the partial fraction decomposition of Equation 6.89 can be found.

$$\frac{s}{(s^2+\omega^2)(s^2+\frac{c}{m}s+\frac{k}{m})} = F_1 - F_2 \tag{6.103}$$

$$= 1 + \frac{(\frac{-m^2\omega^2+km}{q})s + \frac{cm\omega^2}{q}}{s^2+\omega^2} - (1 + \frac{\frac{km-m^2\omega^2}{q}s + \frac{kc}{q}}{s^2+\frac{c}{m}s+\frac{k}{m}}) \tag{6.104}$$

$$= \frac{(\frac{-m^2\omega^2+km}{q})s + \frac{cm\omega^2}{q}}{s^2+\omega^2} - \frac{\frac{km-m^2\omega^2}{q}s + \frac{kc}{q}}{s^2+\frac{c}{m}s+\frac{k}{m}} \tag{6.105}$$

With the above expression, the Laplace inverse of Equation 6.87 can be obtained.

$$U(s) = \frac{F}{m}\frac{s}{(s^2+\omega^2)(s^2+\frac{c}{m}s+\frac{k}{m})} + \frac{su(0)+u'(0)+\frac{c}{m}u(0)}{s^2+\frac{c}{m}s+\frac{k}{m}} \tag{6.106}$$

$$U(s) = \frac{F}{m}(\frac{(\frac{-m^2\omega^2+km}{q})s + \frac{cm\omega^2}{q}}{s^2+\omega^2} - \frac{\frac{km-m^2\omega^2}{q}s + \frac{kc}{q}}{s^2+\frac{c}{m}s+\frac{k}{m}}) + \frac{su(0)+u'(0)+\frac{c}{m}u(0)}{s^2+\frac{c}{m}s+\frac{k}{m}} \tag{6.107}$$

Another constant, $p = \frac{(-m\omega^2+k)F}{q}$ is introduced. This expression and the initial conditions are substituted.

$$U(s) = \frac{ps+\frac{c\omega^2 F}{q}}{s^2+\omega^2} - \frac{ps+\frac{kcF}{mq}}{s^2+\frac{c}{m}s+\frac{k}{m}} + \frac{su_0+v_0+\frac{c}{m}u_0}{s^2+\frac{c}{m}s+\frac{k}{m}} \tag{6.108}$$

$$\tag{6.109}$$

Recall that $\frac{c}{m}$ can be written as $2\zeta\omega_0$.

$$U(s) = \frac{ps+\frac{c\omega^2 F}{q}}{s^2+\omega^2} - \frac{ps+2\zeta\omega_0\frac{kF}{q}}{s^2+\frac{c}{m}s+\frac{k}{m}} + \frac{su_0+v_0+2\zeta\omega_0 u_0}{s^2+\frac{c}{m}s+\frac{k}{m}} \tag{6.110}$$

$$U(s) = p\frac{s}{s^2+\omega^2} + \frac{c\omega F}{q}\frac{\omega}{s^2+\omega^2} - p\frac{s+\omega_0\zeta}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})} \tag{6.111}$$

$$+ p\frac{\omega_0\zeta}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})} - \frac{2\zeta\omega_0\frac{kF}{q}}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})} + u_0\frac{(s+\omega_0\zeta)}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})} \tag{6.112}$$

$$+ \frac{v_0+\zeta\omega_0 u_0}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})}$$

$$U(s) = p\frac{s}{s^2+\omega^2} + \frac{c\omega F}{q}\frac{\omega}{s^2+\omega^2} + (u_0-p)\frac{s+\omega_0\zeta}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})}$$
$$+\frac{\zeta\omega_0\frac{(-m\omega^2+k)F}{q} - \zeta\omega_0\frac{2kF}{q}}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})} + \frac{v_0+\zeta\omega_0 u_0}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})} \tag{6.113}$$

$$U(s) = p\frac{s}{s^2+\omega^2} + \frac{c\omega F}{q}\frac{\omega}{s^2+\omega^2} + (u_0-p)\frac{s+\omega_0\zeta}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})}$$
$$+\frac{\zeta\omega_0\frac{F(-m\omega^2-k)}{q}}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})} + \frac{v_0+\zeta\omega_0 u_0}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})} \tag{6.114}$$

$$U(s) = p\frac{s}{s^2+\omega^2} + \frac{c\omega F}{q}\frac{\omega}{s^2+\omega^2} + (u_0-p)\frac{s+\omega_0\zeta}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})}$$
$$+\frac{\zeta\omega_0\frac{F(m\omega^2-k)}{q} - 2\zeta\omega_0\frac{Fm\omega^2}{q}}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})} + \frac{v_0+\zeta\omega_0 u_0}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})} \tag{6.115}$$

$$U(s) = p\frac{s}{s^2+\omega^2} + \frac{c\omega F}{q}\frac{\omega}{s^2+\omega^2} + (u_0-p)\frac{s+\omega_0\zeta}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})}$$
$$+\frac{-\zeta\omega_0 p - \frac{Fc\omega^2}{q}}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})} + \frac{v_0+\zeta\omega_0 u_0}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})} \tag{6.116}$$

$$U(s) = p\frac{s}{s^2+\omega^2} + \frac{c\omega F}{q}\frac{\omega}{s^2+\omega^2} + (u_0-p)\frac{s+\omega_0\zeta}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})}$$
$$+\frac{v_0+\zeta\omega_0(u_0-p) - \frac{Fc\omega^2}{q}}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2}} \tag{6.117}$$

$$U(s) = p\frac{s}{s^2+\omega^2} + \frac{c\omega F}{q}\frac{\omega}{s^2+\omega^2} + (u_0-p)\frac{s+\omega_0\zeta}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2})}$$
$$+\frac{v_0+\zeta\omega_0(u_0-p) - \frac{Fc\omega^2}{q}}{\omega_0\sqrt{1-\zeta^2}}\frac{\omega_0\sqrt{1-\zeta^2}}{(s+\omega_0\zeta)^2+(\omega_0\sqrt{1-\zeta^2}} \tag{6.118}$$

$$u(t) = e^{-\zeta\omega_0 t}((u_0-p)\cos\Big(\omega_0\sqrt{(1-\zeta^2)}t\Big) + \frac{v_0+\zeta\omega_0(u_0-p)-\frac{Fc\omega^2}{q}}{\omega_0\sqrt{1-\zeta^2}}\sin\Big(\omega_0\sqrt{(1-\zeta^2)}t\Big))$$
$$+p\cos(\omega t) + \frac{Fc\omega}{q}\sin(\omega t) \tag{6.119}$$

The same solution is obtained is Section 4.3.1.

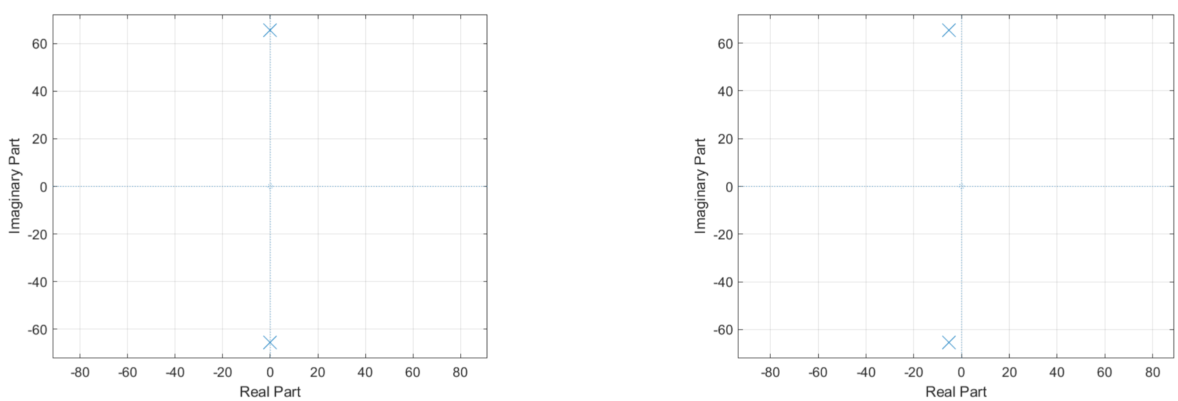
6.5 The Transfer function

Without solving the differential equation, the transfer function provides a basis for determining characteristics of the differential equation and the system. The transfer function is the ratio of output to input of a system after taking the Laplace transform. The transfer function is often called $H(s)$. It is often convenient to write $H(s) = \frac{N(s)}{D(s)}$. The z_i's such that $N(z_i) = 0$ are called the zeros of the system and the p_i's such that $D(p_i) = 0$ are called the poles of the system (Olsder et al., 2011). The transfer functions for the systems described in Chapter 4 are defined as follows:

$$H_{\text{nodamping}}(s) = \frac{1}{s^2 + \frac{k}{m}} \tag{6.120}$$

$$H_{\text{damping}}(s) = \frac{1}{s^2 + \frac{c}{m}s + \frac{k}{m}} \tag{6.121}$$

The function `pzmap` in MATLAB gives a pole-zero map of a transfer function. A pole-zero map is a graphical representation of the poles and zeros of a transfer function. The location of the poles and zeros gives information of the behavior of the system. In the figures below, the pole-zero maps for the transfer functions stated in Equation 6.120 and 6.121 are given.



(a) $H_{\text{nodamping}}(s)$ (b) $H_{\text{damping}}(s)$

Figure 6.1: Pole-zero maps of both transfer functions

Above figures show that the poles of the transfer functions lie in the complex plane. This means that the dynamic behavior of the system is periodic. As can be seen in Figure 6.1a the poles of the undamped system lie on the imaginary axis. A pole lying on the imaginary axis generates an oscillatory component with a constant amplitude determined by the initial conditions, which is the case for an undamped mass spring system as showed in Figure 4.2 (*Understanding Poles and Zeros* n.d.). Figure 6.1b shows that the poles for the damped

system lie in the left-half plane. The real parts of the poles are negative, which causes an oscillation decaying over time. By looking at Figure 4.4, it can be concluded that adding a damper indeed decreases the amplitude over time.
Figure 6.2 shows the relation between the poles of the transfer function and the eigenfrequency and damping ratio of a system.

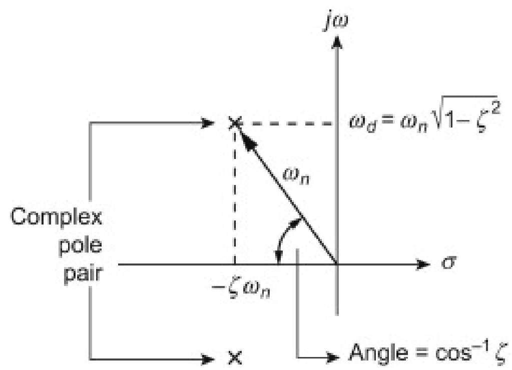


Figure 6.2: A visualization of the relation between the pole location, eigenfrequency and damping ratio (Thompson, 2014)

The eigenfrequency equals the magnitude of the complex poles (Aghajanian et al., 2014). The imaginary part of the pole is the damped eigenfrequency, while the real part of the pole sets the rate at which the oscillation decays (Thompson, 2014).

Chapter 7

Two-degree-of-freedom mass-spring system

In this chapter, the first model is extended to a two spring-coupled mass system. In real life, this model could for example represent the upper and lower part of the bus door. On the one side, the second point mass is attached to the first mass using a spring. On the other side, the point mass is connected to a wall through a third spring. Using two masses brings the model a little closer to the real-life situation. This could represent for example the upper and lower part of the bus door. The analytical solution and numerical solution are plotted in JULIA such that the obtained solutions can be verified as correct. The code is included in Appendix E.

7.1 No external forces

The first model does not include damping or external forces. Also, the two masses are restricted to only moving in the x-direction. It is still assumed that the masses slide over a frictionless table, so there is still no gravity. The vector $\vec{u}(t)$ exists of the displacement of mass 1 and the displacement of mass 2. The equilibrium positions for mass 1 and mass 2 are denoted, respectively, by x_{01} and x_{02} A visualization of this model is presented in the figure below.

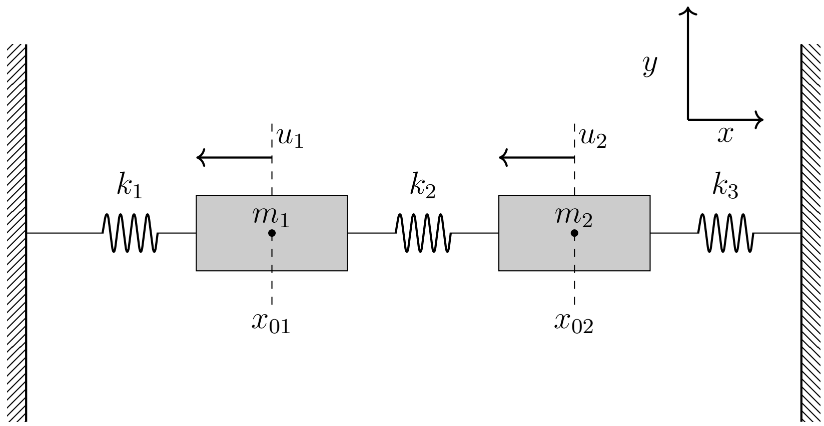


Figure 7.1: A visualization of an undamped two spring-coupled mass system without external forces

The equations of motion for this system are as follows

$$\begin{cases} m_1\ddot{u}_1 &= -(k_1+k_2)u_1 + k_2u_2 \\ m_2\ddot{u}_2 &= k_2u_1 - (k_2+k_3)u_2 \end{cases} \tag{7.1}$$

The above equation can also be written as a matrix vector system.

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{u}_1(t) \\ \ddot{u}_2(t) \end{bmatrix} + \begin{bmatrix} k_1+k_2 & -k_2 \\ -k_2 & k_2+k_3 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{7.2}$$

$$M\ddot{\vec{u}}(t) + K\vec{u}(t) = \vec{0} \tag{7.3}$$

To solve these differential equations, the same method is used as in Chapter 4. A solution is guessed of the form

$$\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t} = \vec{a}e^{i\omega t} \tag{7.4}$$

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \cdot -\omega^2 \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t} + \begin{bmatrix} k_1+k_2 & -k_2 \\ -k_2 & k_2+k_3 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{7.5}$$

$$\begin{bmatrix} -m_1\omega^2 & 0 \\ 0 & -m_2\omega^2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t} + \begin{bmatrix} k_1+k_2 & -k_2 \\ -k_2 & k_2+k_3 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{7.6}$$

$$\begin{bmatrix} -m_1\omega^2 + (k_1+k_2) & -k_2 \\ -k_2 & -m_2\omega^2 + (k_2+k_3) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{7.7}$$

$$\begin{bmatrix} -m_1\omega^2 + (k_1+k_2) & -k_2 \\ -k_2 & -m_2\omega^2 + (k_2+k_3) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{7.8}$$

If both sides of Equation 7.8 are multiplied by the inverse of the obtained matrix, the values of a_1 and a_2 can be found. This leads to the solution $\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, which is obviously a solution. The masses are just holding still. However, a nontrivial solution that actually contains motion is preferred. If both a_1 and a_2 do not equal zero, it can be concluded that the inverse of the matrix from Equation 7.8 does not exists, which implies that the determinant of the specific matrix equals zero (Nicholson, 2020). The determinant of the matrix is calculated as follows

$$\begin{vmatrix} -m_1\omega^2 + (k_1+k_2) & -k_2 \\ -k_2 & -m_2\omega^2 + (k_2+k_3) \end{vmatrix} = 0 \tag{7.9}$$

$$(-m_1\omega^2 + (k_1+k_2))(-m_2\omega^2 + (k_2+k_3)) - (-k_2 \cdot -k_2) = 0 \tag{7.10}$$

$$m_1m_2\omega^4 - (m_1(k_1+k_3) + m_2(k_1+k_2))\omega^2 + (k_1+k_2)(k_2+k_3) - k_2^2 = 0 \tag{7.11}$$

$$m_1m_2\omega^4 - (m_1(k_1+k_3) + m_2(k_1+k_2))\omega^2 + k_1k_2 + k_1k_3 + k_2k_3 = 0 \tag{7.12}$$

$$\tag{7.13}$$

With the help of the quadratic formula the two natural frequencies, $\omega_0^{(1)}$ and $\omega_0^{(2)}$ can be found.

$$\begin{aligned}\omega_0^{2(1)} = &\frac{1}{2}(\frac{(k_1+k_2)m_2+(k_2+k_3)m_1}{m_1m_2}) + \frac{1}{2}[(\frac{(k_1+k_2)m_2+(k_2+k_3)m_1}{m_1m_2})^2 \\ &- 4\sqrt{\frac{(k_1+k_2)(k_2+k_3)-k_2^2}{m_1m_2}}]\end{aligned} \tag{7.14}$$

$$\begin{aligned}\omega_0^{2(2)} = &\frac{1}{2}(\frac{(k_1+k_2)m_2+(k_2+k_3)m_1}{m_1m_2}) - \frac{1}{2}[(\frac{(k_1+k_2)m_2+(k_2+k_3)m_1}{m_1m_2})^2 \\ &- 4\sqrt{\frac{(k_1+k_2)(k_2+k_3)-k_2^2}{m_1m_2}}]\end{aligned} \tag{7.15}$$

Thus one of the requirements of the solution $\vec{u}(t)$ is found and can be written as

$$(-\omega_0^{2(1,2)}M + K)\vec{a} = 0 \tag{7.16}$$

$$M^{-1}K\vec{a} = \omega_0^{2(1,2)}\vec{a} \tag{7.17}$$

However, $M^{-1}K$ is not a symmetric matrix. For efficiency, symmetry is preferred (Turtellaub, n.d.). The inverse of the square root of matrix M is used, which is defined as follows

$$M^{-\frac{1}{2}} = \begin{bmatrix} \frac{1}{\sqrt{m_1}} & 0 \\ 0 & \frac{1}{\sqrt{m_2}} \end{bmatrix} \tag{7.18}$$

Let $\vec{b} = M^{\frac{1}{2}}\vec{a}$ to obtain

$$(-\omega_0^{2(1,2)}M + K)M^{-\frac{1}{2}}\vec{b} = 0 \tag{7.19}$$

$$(-\omega_0^{2(1,2)}MM^{-\frac{1}{2}} + KM^{-\frac{1}{2}})\vec{b} = 0 \tag{7.20}$$

$$M^{-\frac{1}{2}}(-\omega_0^{2(1,2)}MM^{-\frac{1}{2}} + KM^{-\frac{1}{2}})\vec{b} = 0 \tag{7.21}$$

$$(-\omega_0^{2(1,2)}I_2 + M^{-\frac{1}{2}}KM^{-\frac{1}{2}})\vec{b} = 0 \tag{7.22}$$

$$(-\omega_0^{2(1,2)}I_2 + \hat{K})\vec{b} = 0 \tag{7.23}$$

$$\hat{K}\vec{b} = \omega_0^{2(1,2)}\vec{b} \tag{7.24}$$

Notice that $\hat{K}$ is a symmetric matrix, which means that the matrix is orthogonally diagonalizable (Nicholson, 2020). In JULIA, the eigenvalues and eigenvectors of the matrix $\hat{K}$ are found and normalized. The normalized eigenvectors are plugged in a matrix P. P is called the model matrix of this system (Adhikari and Phani, 2007). Recall that the inverse of an orthogonal matrix equals the transposed of that matrix (Nicholson, 2020). Λ is a diagonal matrix with the corresponding eigenvalues on the diagonal. Thus $\hat{K}$ can be written as $P^T \Lambda P$. Now, transform the system by substituting $\vec{q}(t) = M^{\frac{1}{2}}\vec{u}(t)$ and $\ddot{\vec{q}}(t) = M^{\frac{1}{2}}\ddot{\vec{u}}(t)$.

$$MM^{-\frac{1}{2}}\ddot{q}(t) + KM^{-\frac{1}{2}}q(t) = 0 \tag{7.25}$$

$$M^{-\frac{1}{2}}MM^{-\frac{1}{2}}\ddot{\vec{q}}(t) + M^{-\frac{1}{2}}KM^{-\frac{1}{2}}\vec{q}(t) = \vec{0} \tag{7.26}$$

$$I_2\ddot{\vec{q}}(t) + \hat{K}\vec{q}(t) = \vec{0} \tag{7.27}$$

$$\tag{7.28}$$

Now a second transformation takes place, namely $\vec{q}(t) = P\vec{r}(t)$ and $\ddot{\vec{q}}(t) = P\ddot{\vec{r}}(t)$. This gives

$$I_2P\ddot{\vec{r}}(t) + \hat{K}P\vec{r}(t) = \vec{0} \tag{7.29}$$

$$P^T I_2 P\ddot{\vec{r}}(t) + P^T\hat{K}P\vec{r}(t) = \vec{0} \tag{7.30}$$

$$I_2\ddot{\vec{r}}(t) + \Lambda\vec{r}(t) = \vec{0} \tag{7.31}$$

$$\tag{7.32}$$

From Equation 7.32 a new system of equations can be derived, namely:

$$\ddot{\vec{r_1}}(t) + \omega_n^{2(1)}\vec{r_1}(t) = \vec{0} \tag{7.33}$$

$$\ddot{\vec{r_2}}(t) + \omega_n^{2(2)}\vec{r_2}(t) = \vec{0} \tag{7.34}$$

$$\tag{7.35}$$

Now, the multiple degree of freedom system is uncoupled into two differential equations and can treated as a collection of single degree-of-freedom systems, which are described in Section 4.1. Thus the following solution is obtained

$$\begin{bmatrix} r_1(t) \\ r_2(t) \end{bmatrix} = \begin{bmatrix} c_1\cos\left(\omega_0^{(1)}t\right) + c_2\sin\left(\omega_0^{(1)}t\right) \\ c_3\cos\left(\omega_0^{(2)}t\right) + c_4\sin\left(\omega_0^{(2)}t\right) \end{bmatrix} \tag{7.36}$$

The initial conditions are of the same form as in Section 4.1, so $\begin{bmatrix} c_1 \\ c_3 \end{bmatrix} = P^T M^{\frac{1}{2}}\vec{u}_0$ and $\begin{bmatrix} c_2 \\ c_4 \end{bmatrix} = P^T M^{\frac{1}{2}} \begin{bmatrix} \frac{v_{0,1}}{\omega_0^{(1)}} \\ \frac{v_{0,2}}{\omega_0^{(2)}} \end{bmatrix}$. However, this is a solution for $\vec{r}(t)$ and not for $\vec{u}(t)$. To obtain a solution for $\vec{u}(t)$, the solution for $\vec{r}(t)$ has to be transformed back to the original system. The used transformations are $\vec{u}(t) = M^{-\frac{1}{2}}\vec{q}(t)$ and $\vec{q}(t) = P\vec{r}(t)$. So $\vec{u}(t) = M^{-\frac{1}{2}}P\vec{r}(t)$. So the solution for Equation 7.1 is as follows

$$\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = M^{-\frac{1}{2}} P \begin{bmatrix} c_1 \cos\left(\omega_0^{(1)} t\right) + c_2 \sin\left(\omega_0^{(1)} t\right) \\ c_3 \cos\left(\omega_0^{(2)} t\right) + c_4 \sin\left(\omega_0^{(2)} t\right) \end{bmatrix} \tag{7.37}$$

where the values of the c_i's are described above. As can see in the figure below, this solution is correct.

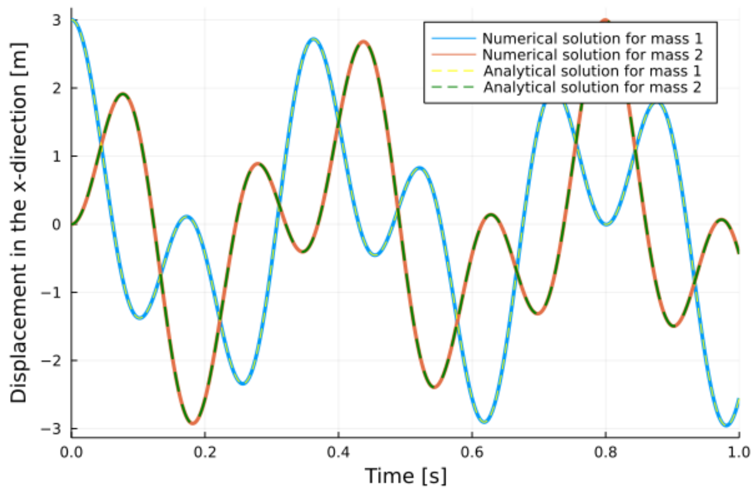


Figure 7.2: A graph of the analytical and numerical solution of the displacement of both masses for the undamped situation without external forces

7.1.1 Damped system without external forces

Three dampers are added to the system described in Section 7.1. The new situation is visualized in the figure below.

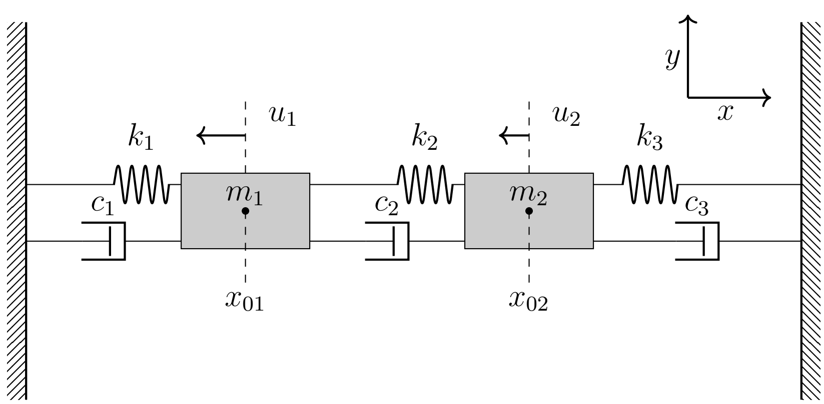


Figure 7.3: A diagram of a damped two spring-coupled mass system without external forces

Adding dampers to the system gives the following equations of motion

$$\begin{cases} m_1\ddot{u}_1 &= -(c_1+c_2)\dot{u}_1 + c_2\dot{u}_2 - (k_1+k_2)u_1 + k_2u_2 \\ m_2\ddot{u}_2 &= c_2\dot{u}_1 - (c_2+c_3)\dot{u}_2 + k_2u_1 - (k_2+k_3)u_2 \end{cases} \tag{7.38}$$

This equation can be rewritten in the following matrix vector system

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{u}_1(t) \\ \ddot{u}_2(t) \end{bmatrix} + \begin{bmatrix} c_1+c_2 & -c_2 \\ -c_2 & c_2+c_3 \end{bmatrix} \begin{bmatrix} \dot{u}_1(t) \\ \dot{u}_2(t) \end{bmatrix} + \begin{bmatrix} k_1+k_2 & -k_2 \\ -k_2 & k_2+k_3 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{7.39}$$

$$M\ddot{\vec{u}}(t) + C\dot{\vec{u}}(t) + K\vec{u}(t) = \vec{0} \tag{7.40}$$

In this section, a distinction is made between the classically damped situation and the non-classically damped situation.

Classically damped

Firstly, the assumption is made that the system is classically damped. The same transformations are used as in Section 7.1 to obtain the following equation

$$I_2\ddot{\vec{r}}(t) + P^T M^{-\frac{1}{2}} C M^{-\frac{1}{2}} P\dot{\vec{r}}(t) + \Lambda r(t) = \vec{0} \tag{7.41}$$

Since classical damping is assumed, $P^T M^{-\frac{1}{2}} C M^{-\frac{1}{2}} P$ is a diagonal matrix. Rayleigh showed that a system is classically damped if the damping matrix is a linear combination of the mass matrix and stiffness matrix (Adhikari and Phani, 2007). Thus Equation 7.41 can be written in the following form

$$\ddot{\vec{r_1}}(t) + \beta_1\dot{r_1}(t) + \omega_n^{(1)}\vec{r_1}(t) = \vec{0} \tag{7.42}$$

$$\ddot{\vec{r_2}}(t) + \beta_2\dot{r_2}(t) + \omega_n^{(2)}\vec{r_2}(t) = \vec{0} \tag{7.43}$$

$$\tag{7.44}$$

From Section 4.1.1, it is known that the solution is of the form

$$\ddot{\vec{r_1}}(t) + 2\zeta_1\omega_i\dot{r_1}(t) + \omega_n^{2(1)}\vec{r_1}(t) = \vec{0} \tag{7.45}$$

$$\ddot{\vec{r_2}}(t) + 2\zeta_2\omega_i\dot{r_2}(t) + \omega_n^{2(2)}\vec{r_2}(t) = \vec{0} \tag{7.46}$$

$$\tag{7.47}$$

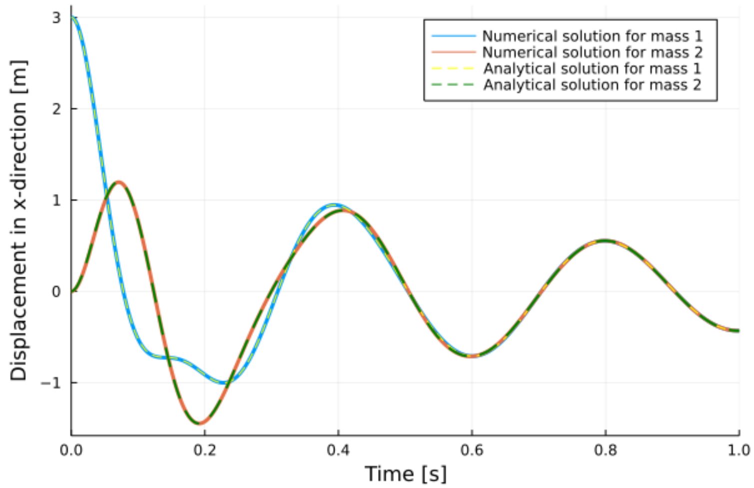


Figure 7.4: A graph of the analytical and numerical solution of the displacement of both masses for the undamped situation without external forces

where ζ_i is defined as $\frac{\beta_i}{2\omega_0^{(i)}}$. Now, the solution for the two degree-of-freedom situation can be copied from Section 4.1.1. The solution for $r(t)$ is

$$\begin{bmatrix} r_1(t) \\ r_2(t) \end{bmatrix} = \begin{bmatrix} e^{-\omega_0^{(1)}\zeta_1 t}(\alpha_1 \cos\left(\omega_0^{(1)}\sqrt{1-\zeta_1^2}t\right) + \alpha_2 \sin\left(\omega_0^{(1)}\sqrt{1-\zeta_1^2}t\right)) \\ e^{-\omega_0^{(2)}\zeta_2 t}(\alpha_3 \cos\left(\omega_0^{(2)}\sqrt{1-\zeta_2^2}t\right) + \alpha_4 \sin\left(\omega_0^{(2)}\sqrt{1-\zeta_2^2}t\right)) \end{bmatrix} \tag{7.48}$$

The values for the α_i's in the above equation are comparable to the previously found values. $\begin{bmatrix} \alpha_1 \\ \alpha_3 \end{bmatrix} = P^T M^{\frac{1}{2}} \vec{u_0}$ and $\begin{bmatrix} \alpha_2 \\ \alpha_4 \end{bmatrix} = P^T M^{\frac{1}{2}} \begin{bmatrix} \frac{v_{0,1}+u_{0,1}\omega_0^{(1)}\zeta_1}{\omega_0^{(1)}\sqrt{1-\zeta_1^2}} \\ \frac{v_{0,2}+u_{0,2}\omega_0^{(2)}\zeta_2}{\omega_0^{(2)}\sqrt{1-\zeta_2^2}} \end{bmatrix}$ To obtain a solution for the displacement of the two masses, the solution is transformed back. Thus a solution is

$$\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = M^{-\frac{1}{2}} P \begin{bmatrix} e^{-\omega_0^{(1)}\zeta_1 t}(\alpha_1 \cos\left(\omega_0^{(1)}\sqrt{1-\zeta_1^2}t\right) + \alpha_2 \sin\left(\omega_0^{(1)}\sqrt{1-\zeta_1^2}t\right)) \\ e^{-\omega_0^{(2)}\zeta_2 t}(\alpha_3 \cos\left(\omega_0^{(2)}\sqrt{1-\zeta_2^2}t\right) + \alpha_4 \sin\left(\omega_0^{(2)}\sqrt{1-\zeta_2^2}t\right)) \end{bmatrix} \tag{7.49}$$

The graph below shows that this solution is correct.

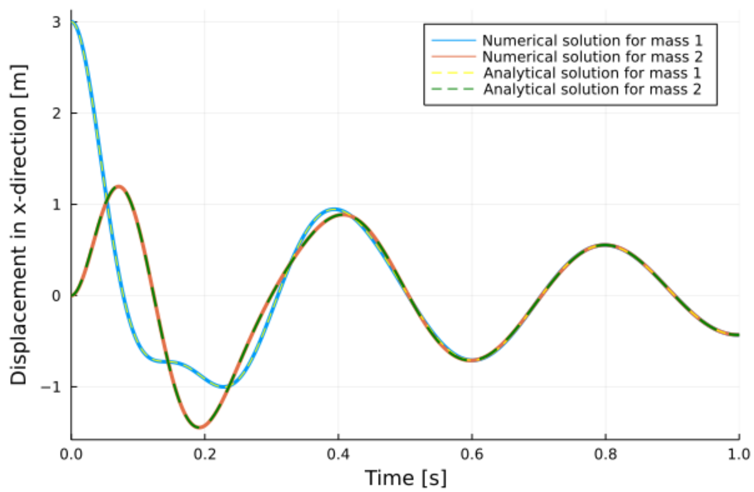


Figure 7.5: A graph of the analytical and numerical solution of the displacement of both masses for the classically damped situation without external forces

Non-classically damping

There are situations that cannot be accurately analyzed by the classical damping model. These situations are defined as non-classically damped. The modal equations of motion of non-classically damped structures are coupled by off-diagonal terms in the modal damping matrix (Xu and Igusa, 1991). In fact, experimental modal testing suggests that no physical system is strictly classically damped. In the analysis of non-classically damped systems, a common approximation is to ignore the off-diagonal elements of the modal damping matrix. This is called the "decoupling approximation". Multiple studies were done whether the approximation could be improved. However, it turned out that the one that minimizes the error bound of the decoupling approximation is the modal damping matrix with omitted off-diagonal elements (Adhikari, 2001). Suppose $D = P^T M^{-\frac{1}{2}} C M^{-\frac{1}{2}} P$, then the equations of motion equal

$$\ddot{\vec{r}}(t) + D\dot{\vec{r}}(t) + \Lambda\vec{r}(t) = \vec{0} \tag{7.50}$$

The modal damping matrix is split such that $D = D_d + D_o$. D_d is a diagonal matrix composed of the diagonal elements of D, and D_o is matrix with zero diagonal elements and whose off-diagonal elements coincide with those in D. The decoupling approximation amounts to simply neglecting D_o and thus replacing D by D_d (Morzfeld, Ajavakom, and Ma, 2009). Now, the equations of motion are decoupled and the method described in Section 7.1.1 can be used. The solutions for the decoupled equations would be close to the exact solution of the coupled equations if the non-classical damping terms are sufficiently small (Adhikari, 2001). To get an idea of the error, the numerical solution of a non-classically damped system is plotted,

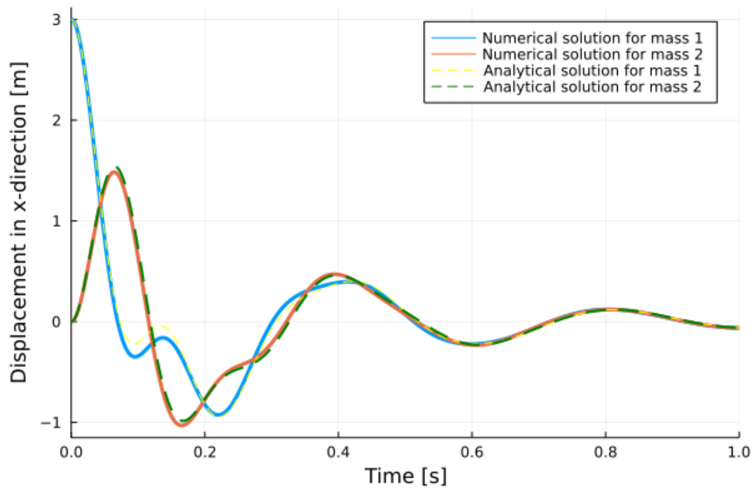


Figure 7.6: A graph of the analytical and numerical solution of the displacement of both masses for the non-classically damped situation without external forces

together with the analytical solution of the non-classically damped system using the method described above.

Chapter 8

Single point mass in two spatial dimensions mass-spring system

In the previous models, the point masses' movements were restricted to one spatial direction. This chapter describes a model that allows the point mass to move in both the x- and y-direction, which is also the case in the practical situation. First, an undamped system is considered excluding external forces. Afterwards, dampers are added to the system. Lastly, the situations including a constant external force and a periodic force are taken into consideration. All the analytically found solutions for the displacement in the x- and y-direction are plotted against each other in JULIA. The code can be found in Appendix F, including the initial conditions.

8.1 No external forces

A point mass of mass m is in the middle of two walls and connected to the walls by means of two springs with spring constants k_1 and k_2. Gravity is still not included in the system. No external forces are applied, so the movement of the mass depends on the initial position of the mass, which is denoted by $\vec{u}_0$. (x_0, y_0) represents the equilibrium position of the mass. The initial position $\vec{u}_0$ consists of an intial x- and y-postion $(u_{0,x}, u_{0,y})$. The same holds for the initial velocity, denoted $\vec{v}_0$. $\vec{u}(t)$ is the displacement of the mass from the equilibrium position and exists of the displacement in the x-direction and in the y-direction. The situation is visualized below.

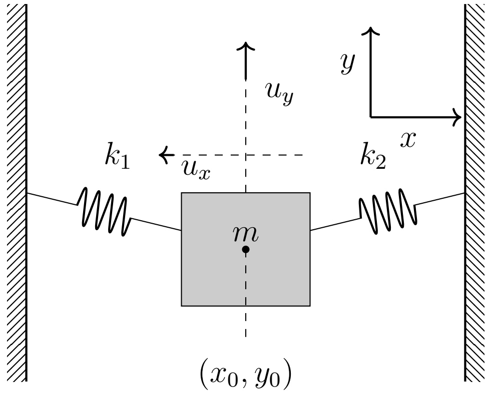


Figure 8.1: A diagram of an undamped mass-spring system without external forces moving in both the x- and y-direction

First, the spring force is divided in the force resulting from the right and left spring.

$$\begin{bmatrix} F_{spring,x} \\ F_{spring,y} \end{bmatrix} = \begin{bmatrix} F^{(1)}_{spring,x} + F^{(2)}_{spring,x} \\ F^{(1)}_{spring,y} + F^{(2)}_{spring,y} \end{bmatrix} \tag{8.1}$$

$$\begin{bmatrix} F_{spring,x} \\ F_{spring,y} \end{bmatrix} = \begin{bmatrix} -k_{1x}u_x - k_{2x}u_x \\ -k_{1y}u_y - k_{2y}u_y \end{bmatrix} \tag{8.2}$$

$$\begin{bmatrix} F_{spring,x} \\ F_{spring,y} \end{bmatrix} = \begin{bmatrix} -(k_{1x} + k_{2x}) & 0 \\ 0 & -(k_{1y} + k_{2y}) \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{8.3}$$

$$\tag{8.4}$$

Now, the equations of motion can easily be obtained.

$$m\ddot{u}_x + (k_{1x} + k_{2x})u_x = 0 \tag{8.5}$$

$$m\ddot{u}_y + (k_{1y} + k_{2y})u_y = 0 \tag{8.6}$$

The equations of motion can be written as a matrix vector system.

$$\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} \ddot{u}_x \\ \ddot{u}_y \end{bmatrix} + \begin{bmatrix} (k_{1x} + k_{2x}) & 0 \\ 0 & (k_{1y} + k_{2y}) \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = \vec{0} \tag{8.7}$$

$$M\ddot{\vec{u}} + K\vec{u} = \vec{0} \tag{8.8}$$

The same guess solution is used as in the previous chapter,$\vec{u}(t) = \vec{a}e^{i\omega t}$ and is plugged into the equations of motion, which gives the following expressions

$$\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} - \omega^2 \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t} + \begin{bmatrix} (k_{1x}+k_{2x}) & 0 \\ 0 & (k_{1y}+k_{2y}) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t} = \vec{0} \tag{8.9}$$

$$\begin{bmatrix} -m\omega^2 & 0 \\ 0 & -m\omega^2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t} + + \begin{bmatrix} (k_{1x}+k_{2x}) & 0 \\ 0 & (k_{1y}+k_{2y}) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t} = \vec{0} \tag{8.10}$$

$$\begin{bmatrix} -m\omega^2 + (k_{1x}+k_{2x}) & 0 \\ 0 & -m\omega^2 + (k_{1y}+k_{2y}) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \vec{0} \tag{8.11}$$

This system also has two eigenfrequencies, namely one for the x-direction and one for the y-direction. By setting the determinant of above matrix equal to zero, the natural frequencies of the system can be found.

$$\begin{vmatrix} -m\omega^2 + (k_{1x}+k_{2x}) & 0 \\ 0 & -m\omega^2 + (k_{1y}+k_{2y}) \end{vmatrix} = 0 \tag{8.12}$$

$$(-m\omega^2 + (k_{1x}+k_{2x}))(-m\omega^2 + (k_{1y}+k_{2y})) = 0 \tag{8.13}$$

$$\tag{8.14}$$

These equation are of exact the same form as Equation 4.4. Thus the eigenfrequencies can easily be obtained by substituting $k = k_{1x} + k_{2x}$ or $k = k_{1y} + k_{2y}$ into the eigenfrequency found in Section 4.1.

$$\omega_0^{(x)} = \sqrt{\frac{k_{1x}+k_{2x}}{m}} \tag{8.15}$$

$$\omega_0^{(y)} = \sqrt{\frac{k_{1y}+k_{2y}}{m}} \tag{8.16}$$

Thus one of the requirements of the solution is

$$(-\omega_0^{2(x,y)} M + K)\vec{a} = 0 \tag{8.17}$$

$$M^{-1}K\vec{a} = \omega_0^{2(x,y)}\vec{a} \tag{8.18}$$

Now, Equation 8.8 is multiplied by M^{-1} to obtain the following equations

$$M^{-1}M\ddot{\vec{u}} + M^{-1}K\vec{u} = \vec{0} \tag{8.19}$$

$$I_2\ddot{\vec{u}} + \Lambda\vec{u} = \vec{0} \tag{8.20}$$

where Λ is a diagonal matrix with $\omega_0^{2(x)}$ and $\omega_0^{2(y)}$ on the diagonal. This gives the following system of equations

$$\ddot{u}_x + \omega_0^{2(x)} u_x = 0 \tag{8.21}$$
$$\ddot{u}_y + \omega_0^{2(y)} u_y = 0 \tag{8.22}$$

These equations are comparable with Equation 4.1 and the same calculations can be done to obtain the solution

$$\begin{bmatrix} u_x(t) \\ u_y(t) \end{bmatrix} = \begin{bmatrix} c_1 \cos\left(\omega_0^{(x)} t\right) + c_2 \sin\left(\omega_0^{(x)} t\right) \\ c_3 \cos\left(\omega_0^{(y)} t\right) + c_4 \sin\left(\omega_0^{(y)} t\right) \end{bmatrix} \tag{8.23}$$

The values for the c_i's are also the same as described in Section 4.1, so $\begin{bmatrix} c_1 \\ c_3 \end{bmatrix} = \begin{bmatrix} u_{0,x} \\ u_{0,y} \end{bmatrix}$ and $\begin{bmatrix} c_2 \\ c_4 \end{bmatrix} = \begin{bmatrix} \frac{v_{0,x}}{\omega_0^{(x)}} \\ \frac{v_{0,y}}{\omega_0^{(y)}} \end{bmatrix}$ A plot in JULIA, which is presented below, shows us that this is the right solution for the equations of motion.

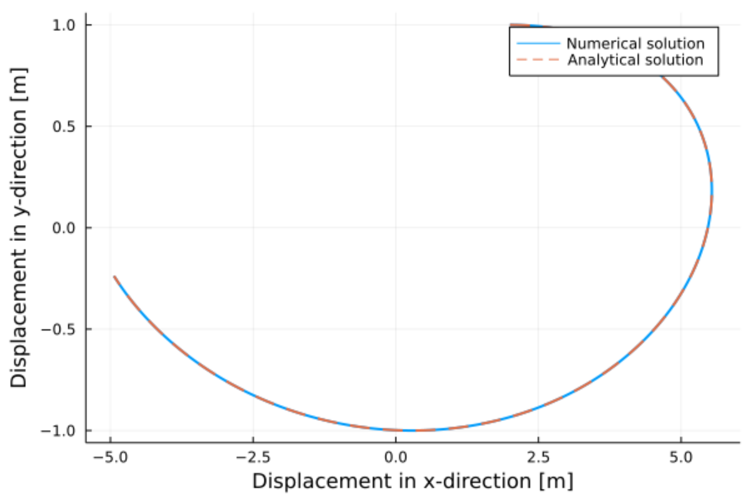


Figure 8.2: A visualization of an undamped mass-spring system without external forces moving in both x- and y-direction

8.1.1 Damped system without external forces

Two dampers, with damping constant c_1 and c_2, are added to the system described in Section ??. This gives the following situation

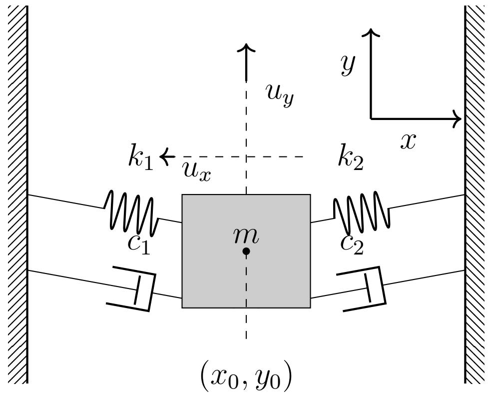


Figure 8.3: A diagram of a damped mass-spring system without external forces moving in both the x- and y-direction

Thus a damping force has to be added to the equations of motion. The damping force is also divided in the damping force resulting from the left damper and the damping force resulting from the right damper.

$$\begin{bmatrix} F_{damper,x} \\ F_{damper,y} \end{bmatrix} = \begin{bmatrix} F^{(1)}_{damper,x} + F^{(2)}_{damper,x} \\ F^{(1)}_{damper,y} + F^{(2)}_{damper,y} \end{bmatrix} \tag{8.24}$$

$$\begin{bmatrix} F_{damper,x} \\ F_{damper,y} \end{bmatrix} = \begin{bmatrix} -c_{1x}\dot{u}_x - c_{1x}\dot{u}_x \\ -c_{1y}\dot{u}_y - c_{2y}\dot{u}_y \end{bmatrix} \tag{8.25}$$

$$\begin{bmatrix} F_{damper,x} \\ F_{damper,y} \end{bmatrix} = \begin{bmatrix} -(c_{1x} + c_{2x}) & 0 \\ 0 & -(c_{1y} + c_{2y}) \end{bmatrix} \begin{bmatrix} \dot{u}_x \\ \dot{u}_y \end{bmatrix} \tag{8.26}$$

This leads to the following equations of motion

$$m\ddot{u}_x + (c_{1x} + c_{2x})\dot{u}_x + (k_{1x} + k_{2x})u_x = 0 \tag{8.27}$$

$$m\ddot{u}_y + (c_{1y} + c_{2y})\dot{u}_y + (k_{1y} + k_{2y})u_y = 0 \tag{8.28}$$

The equations of motion can be written as a matrix vector system.

$$\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} \ddot{u}_x \\ \ddot{u}_y \end{bmatrix} + \begin{bmatrix} (c_{1x} + c_{2x}) & 0 \\ 0 & (c_{1y} + c_{2y}) \end{bmatrix} \begin{bmatrix} \dot{u}_x \\ \dot{u}_y \end{bmatrix} + \begin{bmatrix} (k_{1x} + k_{2x}) & 0 \\ 0 & (k_{1y} + k_{2y}) \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = \vec{0} \tag{8.29}$$

$$M\ddot{\vec{u}} + C\dot{\vec{u}} + K\vec{u} = \vec{0} \tag{8.30}$$

The same guess solution as used before is plugged into the equations of motion, which gives the following expressions

$$\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \cdot -\omega^2 \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t} + \begin{bmatrix} (c_{1x}+c_{2x}) & 0 \\ 0 & (c_{1y}+c_{2y}) \end{bmatrix} \cdot i\omega \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t} + \begin{bmatrix} (k_{1x}+k_{2x}) & 0 \\ 0 & (k_{1y}+k_{2y}) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t} = \vec{0} \tag{8.31}$$

$$\begin{bmatrix} -m\omega^2 & 0 \\ 0 & -m\omega^2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t} + \begin{bmatrix} (c_{1x}+c_{2x})i\omega & 0 \\ 0 & (c_{1y}+c_{2y})i\omega \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t} + \begin{bmatrix} (k_{1x}+k_{2x}) & 0 \\ 0 & (k_{1y}+k_{2y}) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t} = \vec{0} \tag{8.32}$$

$$\begin{bmatrix} -m\omega^2 + (c_{1x}+c_{2x})i\omega + (k_{1x}+k_{2x}) & 0 \\ 0 & -m\omega^2 + (c_{1y}+c_{2y})i\omega + (k_{1y}+k_{2y}) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \vec{0} \tag{8.33}$$

The determinant of the above matrix is set equal to zero.

$$\begin{vmatrix} -m\omega^2 + (c_{1x}+c_{2x})i\omega + (k_{1x}+k_{2x}) & 0 \\ 0 & -m\omega^2 + (c_{1y}+c_{2y})i\omega + (k_{1y}+k_{2y}) \end{vmatrix} = 0 \tag{8.34}$$

$$(-m\omega^2 + (c_{1x}+c_{2x})i\omega + (k_{1x}+k_{2x}))(-m\omega^2 + (c_{1y}+c_{2y})i\omega + (k_{1y}+k_{2y})) = 0 \tag{8.35}$$

These equations are of the same form as Equation 4.13 and thus the eigenfrequencies can be obtained by substitution.

$$w_n^{(x)} = \omega_0^{(x)}(i\zeta_x + \sqrt{(1-\zeta_x^2)}) \tag{8.36}$$

$$w_n^{(y)} = \omega_0^{(y)}(i\zeta_y + \sqrt{(1-\zeta_y^2)}) \tag{8.37}$$

where $\zeta_x = \frac{c_{1x}+c_{2x}}{2\sqrt{(k_{1x}+k_{2x})m}}$ and $\zeta_y = \frac{c_{1y}+c_{2y}}{2\sqrt{(k_{1y}+k_{2y})m}}$. Now, Equation 8.30 is multiplied by the inverse of the mass matrix to obtain the following

$$M^{-1}M\ddot{\vec{u}} + M^{-1}C\dot{\vec{u}} + M^{-1}K\vec{u} = \vec{0} \tag{8.38}$$

$$I_2\ddot{\vec{u}} + \Theta\dot{\vec{u}} + \Lambda\vec{u} = \vec{0} \tag{8.39}$$

Λ is the same matrix as described in Section 7.1. Θ is a diagonal matrix with the values $2\zeta_x\omega_0^{(x)}$ and $2\zeta_y\omega_0^{(y)}$ on the diagonal. This gives us the following equations of motion for the x- and y-direction

$$\ddot{u}_x + 2\zeta_x\omega_0^{(x)}\vec{\dot{u}}_x + \omega_0^{2(x)}\vec{u}_x = 0 \tag{8.40}$$
$$\ddot{u}_y + 2\zeta_y\omega_0^{(y)}\vec{\dot{u}}_y + \omega_0^{2(y)}\vec{u}_y = 0 \tag{8.41}$$

These equations are comparable with Equation 4.10 and the same calculations can be done to obtain the solution

$$\begin{bmatrix} u_x(t) \\ u_y(t) \end{bmatrix} = \begin{bmatrix} e^{-\omega_0^{(x)}t}(\alpha_1 \cos\left(\omega_0^{(x)}\sqrt{1-\zeta_x^2}\right) + \alpha_2 \sin\left(\omega_0^{(x)}\sqrt{1-\zeta_x^2}\right)) \\ e^{-\omega_0^{(y)}t}(\alpha_3 \cos\left(\omega_0^{(y)}\sqrt{1-\zeta_y^2}\right) + \alpha_4 \sin\left(\omega_0^{(y)}\sqrt{1-\zeta_y^2}\right)) \end{bmatrix} \tag{8.42}$$

The α_i's are of the same form described in Section 4.1.1. Thus $\begin{bmatrix} \alpha_1 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} u_{0,x} \\ u_{0,y} \end{bmatrix}$ and $\begin{bmatrix} \alpha_2 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} \frac{v_{0,x}+u_{0,x}\omega_0^{(x)}\zeta_x}{\omega_0^{(x)}\sqrt{1-\zeta_x^2}} \\ \frac{v_{0,y}+u_{0,y}\omega_0^{(y)}\zeta_x}{\omega_0^{(y)}\sqrt{1-\zeta_y^2}} \end{bmatrix}$ A plot in JULIA, which is presented below, shows us that this is the right solution for the equations of motion.

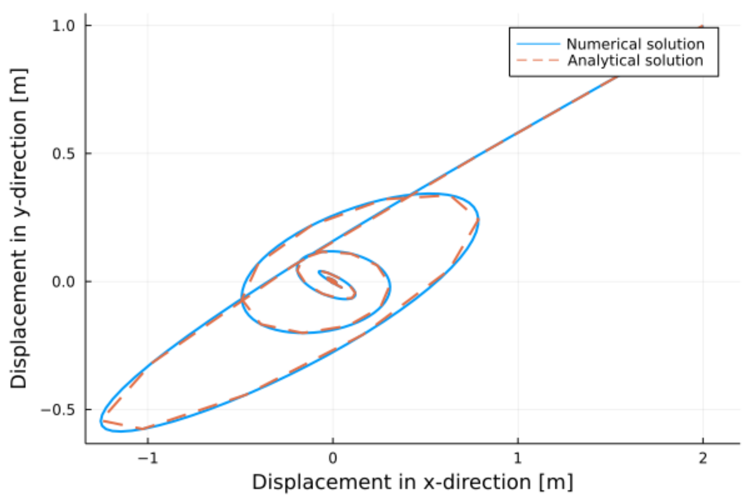


Figure 8.4: A visualization of a damped mass-spring system without external forces moving in both x- and y-direction

8.2 Constant external force in the y-direction

An advantage of this model is that an external force in one direction can be included. In this section, a constant external force, F, is included in the y-direction. This means that the equation of motion for the displacement in the x-direction does not change. However, the

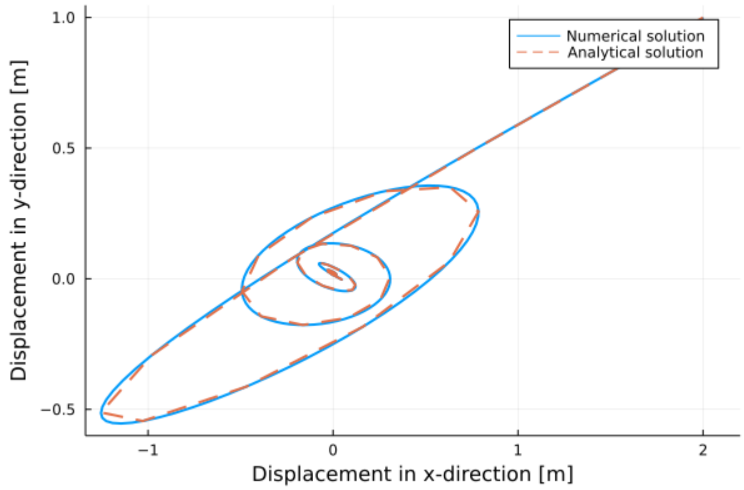


Figure 8.5: A visualization of a damped mass-spring system including constant external force moving in both x- and y-direction

external force is included in the equation of motion for the displacement in the y-direction.

$$m\ddot{u}_x + (c_{1x} + c_{2x})\dot{u}_x + (k_{1x} + k_{2x})u_x = 0 \tag{8.43}$$

$$m\ddot{u}_y + (c_{1y} + c_{2y})\dot{u}_y + (k_{1y} + k_{2y})u_y = F \tag{8.44}$$

The solution for Equation 8.43 is already found in the previous section. Also, a particular solution for Equation 8.44 has already been found in Section 4.2.1. By substituting the right values for the spring and damping constant, the following solution can be obtained.

$$\begin{bmatrix} u_x(t) \\ u_y(t) \end{bmatrix} = \begin{bmatrix} e^{-\omega_0^{(x)}t}(\alpha_1 \cos\left(\omega_0^{(x)}\sqrt{1-\zeta_x^2}\right) + \alpha_2 \sin\left(\omega_0^{(x)}\sqrt{1-\zeta_x^2}\right)) \\ e^{-\omega_0^{(y)}\zeta_y t}(\alpha_3 \cos\left(\omega_0^{(y)}\sqrt{(1-\zeta_y^2)t}\right) + \alpha_4 \sin\left(\omega_0^{(y)}\sqrt{(1-\zeta_y^2)t}\right)) + \frac{F}{k_{1y}+k_{2y}} \end{bmatrix} \tag{8.45}$$

α_1 and α_2 do have the same value as described above. For this situation, $\alpha_3 = u_{0,y} - \frac{F}{k_{1y}+k_{2y}}$ and $\alpha_4 = \frac{v_{0,y}+\omega_0^{(y)}\zeta_y\alpha_3}{w_0^{(y)}\sqrt{1-\zeta_y^2}}$. From Figure 8.5 it can be concluded that the found solution is correct.

8.3 Periodic external force in the y-direction

Now, a periodic external force in the y-direction is included. This external force has an amplitude F and angular frequency ω. Once again, this does not change the equation of motion for the displacement in the x-direction and this solution can simply be copied from before. The equation of motion for the displacement in the y-direction does differ from

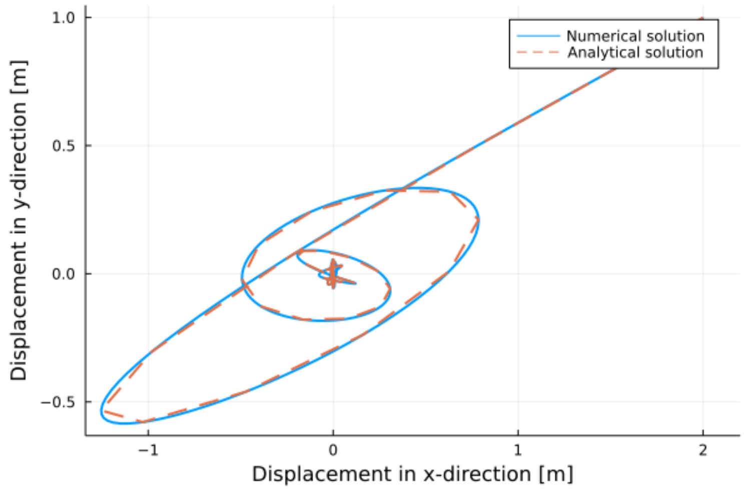


Figure 8.6: A visualization of a damped mass-spring system including periodic external force moving in both x- and y-direction

before. The equations of motion are

$$m\ddot{u}_x + (c_{1x} + c_{2x})\dot{u}_x + (k_{1x} + k_{2x})u_x = 0 \tag{8.46}$$

$$m\ddot{u}_y + (c_{1y} + c_{2y})\dot{u}_y + (k_{1y} + k_{2y})u_y = F\cos(\omega t) \tag{8.47}$$

The equation of motion for the displacement in the y-direction is of the same form as the equation of motion analyzed in Section 4.3.1. Thus the particular solution of Equation 8.47 can be found, using the previously found solution and substitution. The solutions for the equations of motion are

$$\begin{bmatrix} u_x(t) \\ \\ u_y(t) \end{bmatrix} = \begin{bmatrix} e^{-\omega_0^{(x)}t}(\alpha_1 \cos\left(\omega_0^{(x)}\sqrt{1-\zeta_x^2}\right) & +\alpha_2 \sin\left(\omega_0^{(x)}\sqrt{1-\zeta_x^2}\right)) \\ \\ e^{-\omega_0^{(y)}\zeta_y t}(\alpha_3 \cos\left(\omega_0^{(y)}\sqrt{(1-\zeta_y^2)}t\right) & +\alpha_4 \sin\left(\omega_0^{(y)}\sqrt{(1-\zeta_y^2)}t\right)) \\ +\frac{-m\omega^2 F + F(k_{1y}+k_{2y})}{(m\omega^2-(k_{1y}+k_{2y})^2+(c_{1y}+c_{2y})^2\omega^2}\cos(\omega t) & +\frac{F(c_{1y}+c_{2y})\omega}{(m\omega^2-(k_{1y}+k_{2y})^2+(c_{1y}+c_{2y})^2\omega^2}\sin(\omega t) \end{bmatrix}$$

The values of α_1 and α_2 are still the same as described as described above. However, the values of α_3 and α_4 are not the same as before. In the above equation, $\alpha_3 = u_{0,y} - \frac{-m\omega^2 F + F(k_{1y}+k_{2y})}{(m\omega^2-(k_{1y}+k_{2y}))^2+(c_{1y}+c_{2y})^2\omega^2}$ and $\alpha_4 = \frac{v_{0,y}+\zeta\omega_0\alpha_3 - \frac{F(c_{1y}+c_{2y})\omega^2}{(m\omega^2-(k_{1y}+k_{2y}))^2+(c_{1y}+c_{2y})^2\omega^2}}{\omega_0^{(y)}\sqrt{1-\zeta^2}}$. Since the analytical and numerical solution coincide, as showed in Figure 8.6, the analytical solution is correct.

Chapter 9

A two dimensional two spring-coupled masses system

In this chapter, the models described in Chapter 7 and Chapter 8 will be combined. The two spring-coupled masses from Chapter 7 have the possibility to move both in the x- and y-direction. As a result, the displacement vector $\vec{u}(t)$ consists of four functions depending time.

$$\vec{u}(t) = \begin{bmatrix} \vec{u}_1(t) \\ \vec{u}_2(t) \end{bmatrix} = \begin{bmatrix} u_{1x}(t) \\ u_{1y}(t) \\ u_{2x}(t) \\ u_{2y}(t) \end{bmatrix} \tag{9.1}$$

Since this chapter is only combining previously found solution, the undamped situation is skipped. The first situation that is taken into consideration is the damped system. The solution is verified in JULIA and can be found in Appendix

9.1 Damped without external forces

As mentioned above, the model from Chapter 7 will be extended to two spatial dimensions. The components of the displacement vector $\vec{u}$(t) are visualized in Figure 9.1. Also, the equilibrium position of the two masses are displayed.

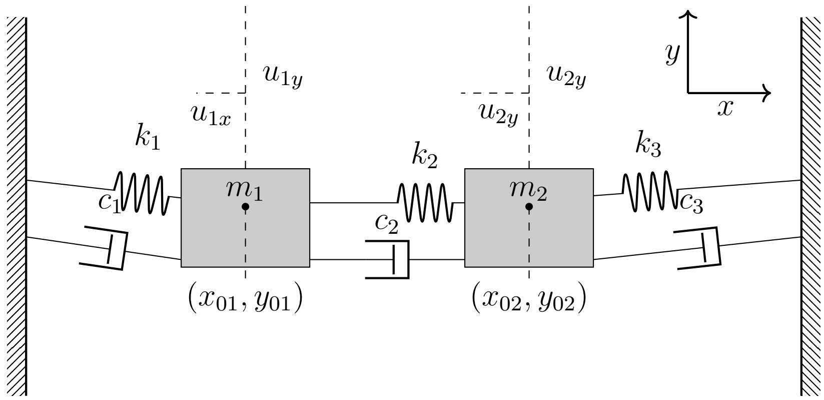


Figure 9.1: A diagram of a damped two spring-coupled mass system without external forces moving in both x- and y-direction

Combining the equations of motion from the previous chapters, the equations of motion for this situations can be obtained

$$\begin{cases} m_1\ddot{u}_{1x} &= -(c_{1x}+c_{2x})\dot{u_{1x}} + c_{2x}\dot{u_{2x}} - (k_{1x}+k_{2x})u_{1x} + k_{2x}u_{2x} \\ m_1\ddot{u}_{1y} &= -(c_{1y}+c_{2y})\dot{u_{1y}} + c_{2y}\dot{u_{2y}} - (k_{1y}+k_{2y})u_{1y} + k_{2y}u_{2y} \\ m_2\ddot{u}_{2x} &= c_{2x}\dot{u_{1x}} - (c_{2x}+c_{3x})\dot{u_{2x}} + k_{2x}u_{1x} - (k_{2x}+k_{3x})u_{2x} \\ m_2\ddot{u}_{2y} &= c_{2y}\dot{u_{1y}} - (c_{2y}+c_{3y})\dot{u_{2y}} + k_{2y}u_{1y} - (k_{2y}+k_{3y})u_{2y} \end{cases} \tag{9.2}$$

Writing the above system of equation in matrix vector form gives

$$\begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_1 & 0 \\ 0 & 0 & 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{u}_{1x} \\ \ddot{u}_{2x} \\ \ddot{u}_{1y} \\ \ddot{u}_{2y} \end{bmatrix} + \begin{bmatrix} (c_{1x}+c_{2x}) & -c_{2x} & 0 & 0 \\ -c_{2x} & (c_{2x}+c_{3x}) & 0 & 0 \\ 0 & 0 & (c_{1y}+c_{2y}) & -c_{2y} \\ 0 & 0 & -c_{2y} & (c_{2y}+c_{3y}) \end{bmatrix} \begin{bmatrix} \dot{u}_{1x} \\ \dot{u}_{2x} \\ \dot{u}_{1y} \\ \dot{u}_{2y} \end{bmatrix} \tag{9.3}$$

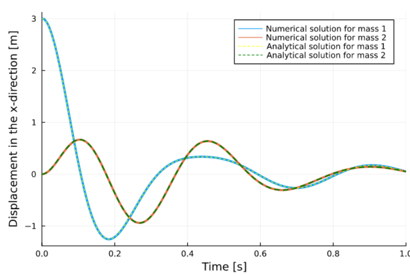
$$+ \begin{bmatrix} (k_{1x}+k_{2x}) & -k_{2x} & 0 & 0 \\ -k_{2x} & (k_{2x}+k_{3x}) & 0 & 0 \\ 0 & 0 & (k_{1y}+k_{2y}) & -k_{2y} \\ 0 & 0 & -c_{ky} & (k_{2y}+k_{3y}) \end{bmatrix} \begin{bmatrix} u_{1x} \\ u_{2x} \\ u_{1y} \\ u_{2y} \end{bmatrix} = \vec{0} \tag{9.4}$$

$$M\ddot{\vec{u}} + C\dot{\vec{u}} + K\vec{u} = \vec{0} \tag{9.5}$$

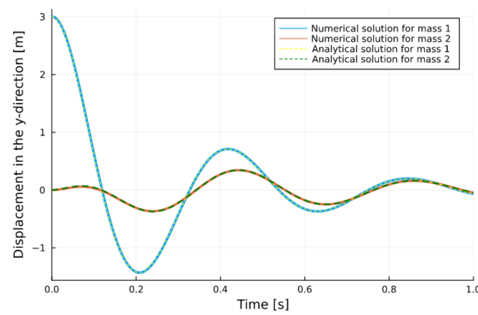
From the above equation, the same problem as in Section 7.1.1 is recognized. Thus in the same manner, this problem can be solved and a solution is

$$\begin{bmatrix} u_{1x} \\ u_{2x} \\ u_{1y} \\ u_{2y} \end{bmatrix} = M^{-\frac{1}{2}} P \begin{bmatrix} e^{-\omega_0^{(1x)}\zeta_{1x}t}\left(\alpha_{1x}\cos\left(\omega_0^{(1x)}\sqrt{1-\zeta_{1x}^2}t\right) + \alpha_{2x}\sin\left(\omega_0^{(1x)}\sqrt{1-\zeta_{1x}^2}t\right)\right) \\ e^{-\omega_0^{(2x)}\zeta_{2x}t}\left(\alpha_{3x}\cos\left(\omega_0^{(2x)}\sqrt{1-\zeta_{2x}^2}t\right) + \alpha_{4x}\sin\left(\omega_0^{(2x)}\sqrt{1-\zeta_{2x}^2}t\right)\right) \\ e^{-\omega_0^{(1y)}\zeta_{1y}t}\left(\alpha_{1y}\cos\left(\omega_0^{(1y)}\sqrt{1-\zeta_{1y}^2}t\right) + \alpha_{2y}\sin\left(\omega_0^{(1y)}\sqrt{1-\zeta_{1y}^2}t\right)\right) \\ e^{-\omega_0^{(2y)}\zeta_{2y}t}\left(\alpha_{3y}\cos\left(\omega_0^{(2y)}\sqrt{1-\zeta_{2y}^2}t\right) + \alpha_{4y}\sin\left(\omega_0^{(2y)}\sqrt{1-\zeta_{2y}^2}t\right)\right) \end{bmatrix} \tag{9.6}$$

The values of the α_i's are defined as follows $\begin{bmatrix} \alpha_{1x} \\ \alpha_{3x} \end{bmatrix} = P^T M^{\frac{1}{2}} \vec{u_{x0}}$, $\begin{bmatrix} \alpha_{1y} \\ \alpha_{3y} \end{bmatrix} = P^T M^{\frac{1}{2}} \vec{u_{y0}}$, $\begin{bmatrix} \alpha_{2x} \\ \alpha_{4x} \end{bmatrix} = P^T M^{\frac{1}{2}} \begin{bmatrix} \frac{v_{x0,1}+u_{x0,1}\omega_0^{(1x)}\zeta_{1x}}{\omega_0^{(1x)}\sqrt{1-\zeta_{1x}^2}} \\ \frac{v_{x0,2}+u_{x0,2}\omega_0^{(2x)}\zeta_{2x}}{\omega_0^{(2x)}\sqrt{1-\zeta_{2x}^2}} \end{bmatrix}$ and $\begin{bmatrix} \alpha_{2y} \\ \alpha_{4y} \end{bmatrix} = P^T M^{\frac{1}{2}} \begin{bmatrix} \frac{v_{y0,1}+u_{y0,1}\omega_0^{(1y)}\zeta_{1y}}{\omega_0^{(1y)}\sqrt{1-\zeta_{1y}^2}} \\ \frac{v_{y0,2}+u_{y0,2}\omega_0^{(2y)}\zeta_{2y}}{\omega_0^{(2y)}\sqrt{1-\zeta_{2y}^2}} \end{bmatrix}$. The equations can be verified with the help of Julia.



(a) Displacement of the masses in the x-direction



(b) Displacement of the masses in the y-direction

Figure 9.2: Both the analytical and numerical solution of the displacement of the masses in a damped system

Chapter 10

Conclusion

A single-degree-freedom mass-spring system was introduced and several situations are considered. For every situation, the equation of motion is analyzed and a solution for the displacement of the mass is found. By doing an analysis of Ventura's test data and the simplified models, several similarities can be found. Using these similarities and Fourier analysis, estimations for the spring constant and damping constant for the simplified model were made. It can be concluded that the spring constant equals approximately 344531.52 $\frac{N}{m}$ and the damping constant equals approximately $c = 670$. Afterward, the solution for the displacement of the mass was found using the Laplace transform method. There is a connection between the transfer function and the characteristics of a mass-spring system. In the following chapter, the bus door is regarded as two separate parts connected by a spring. The solutions for the displacement of the masses are found. The third model that is introduced consists of one mass connected to two walls through springs. Now, the mass can move in both x- and y-direction. Lastly, two models are combined to obtain two spring-coupled masses system moving in both x- and y-direction. This model is the closest to the real-life situation of all mentioned models. The solution for the displacements in both directions of the masses is found when no external forces are included. Unfortunately, there was no time left to include a constant and periodic external force. As a result, there is no analysis done on this model. This could be done in further research.

Bibliography

Adhikari, Sondipon (2001). "Damping models for structural vibration". PhD thesis. Citeseer.

Adhikari, Sondipon and A Srikantha Phani (2007). "Rayleigh's classical damping revisited". In: *International Conference on Civil Engineering in the New Millennium: Opportunities and Challenges*.

Aghajanian, Saeid et al. (June 2014). "Optimal Control of Steel Structures by Improved Particle Swarm". In: *International Journal of Steel Structures* 14, pp. 223–230. DOI: 10.1007/s13296-014-2003-3.

Bauchau, Olivier Andre (2011). *Flexible multibody dynamics*. Vol. 176. Springer.

Braun, M. (1991). *Differential Equations and Their Applications*. Springer.

Cochran, W.T. et al. (1967). "What is the fast Fourier transform?" In: *Proceedings of the IEEE* 55.10, pp. 1664–1674. DOI: 10.1109/PROC.1967.5957.

COMSOL (2018). *Eigenfrequency Analysis*. URL: https://www.comsol.com/multiphysics/eigenfrequency-analysis. (accessed: 25.04.2022).

Daud, Nariman Haji, Siti Nurul Huda Romli, and Nor Aishah Ahmad (2021). *LAPLACE TRANSFORM*.

Dmitrochenko, Oleg (2008). "Finite elements using absolute nodal coordinates for large-deformation flexible multibody dynamics". In: *Journal of Computational and Applied Mathematics* 215.2, pp. 368–377.

Eberhard, P. (n.d.). *Flexible Multibody Systems*. University of Stuttgart. URL: https://www.itm.uni-stuttgart.de/en/research/flexible_multibody_systems/. (accessed: 21.06..2022).

Flores, Paulo (Mar. 2015). "Kinematic Constraint Equations". In: vol. 168. ISBN: 978-3-319-16189-1. DOI: 10.1007/978-3-319-16190-7_7.

Gottlieb, M. and R. Pfeiffer (2013). *Lecture 24 Resonance*. Ed. by California Institute of Technology. URL: https://www.feynmanlectures.caltech.edu/I_23.html. (accessed: 29.04..2022).

Gufler, Veit, Erich Wehrle, and Andreas Zwölfer (2021). "A review of flexible multibody dynamics for gradient-based design optimization". In: *Multibody System Dynamics* 53.4, pp. 379–409.

Haberman, Richard (2013). *Applied partial differential equations: with Fourier series and boundary value problems*. Pearson Education.

Hamper, Martin B et al. (2012). "Use of finite element and finite segment methods in modeling rail flexibility: a comparative study". In: *Journal of computational and nonlinear dynamics* 7.4.

Kazem, Saeed (2013). "Exact solution of some linear fractional differential equations by Laplace transform". In: *International Journal of nonlinear science* 16.1, pp. 3–11.

Lahaye, Domenico (n.d.). "Master Thesis Assignment: The Computational Structural Mechanics of Driving Busses and Vibrating Doors". In: *Delft Institute of Applied Mathematics (DIAM) - TU Delft* ().

Logan, D.L. (2007). *A First Course in the Finite Element Method*. University of Wisconsin–Platteville.

Luo, Shihao, Youhua Fan, and Naigang Cui (2021). "Application of absolute nodal coordinate formulation in calculation of space elevator system". In: *Applied Sciences* 11.23, p. 11576.

Morzfeld, Matthias, N Ajavakom, and F Ma (2009). "Diagonal dominance of damping and the decoupling approximation in linear vibratory systems". In: *Journal of Sound and Vibration* 320.1-2, pp. 406–420.

Nada, AA et al. (2010). "Use of the floating frame of reference formulation in large deformation analysis: experimental and numerical validation". In: *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics* 224.1, pp. 45–58.

Nicholson, W Keith (2020). "Linear algebra with applications". In.

Olsder, G.J. et al. (2011). *Mathematical Systems Theory*. VSSD.

Shabana, Ahmed A (1997). "Flexible multibody dynamics: review of past and recent developments". In: *Multibody system dynamics* 1.2, pp. 189–222.

— (2003). *Dynamics of multibody systems*. Cambridge university press.

Sindy, F., A. Zandbergen, and M. van Zon (2021). "Mathematical Modeling of Bus Doors". Unpublished project. TU Delft.

Systems, Ventura (n.d.). *about us*.

Teixeira, Ricardo, Sérgio Moreira, and Sérgio Tavares (Dec. 2015a). "Multibody Dynamics Simulation of an Electric Bus". In: *Procedia Engineering* 114, pp. 470–477. DOI: `10.1016/j.proeng.2015.08.094`.

Teixeira, Ricardo R, Sérgio RDS Moreira, and SMO Tavares (2015b). "Multibody dynamics simulation of an electric bus". In: *Procedia Engineering* 114, pp. 470–477.

Thompson, Marc T. (2014). *Intuitive Analog Circuit Design*. Elsevier. DOI: 10.1016/C2012-0-03027-X.

Turtellaub, S (n.d.). *Modal Analysis for free vibration of undamped Multiple Degree of Freedom.* (accessed: 15.06..2022).

Understanding Poles and Zeros (n.d.). MASSACHUSETTS INSTITUTE OF TECHNOLOGY. URL: https://web.mit.edu/2.14/www/Handouts/PoleZero.pdf. (accessed: 15.06..2022).

Vaxman, A. (2018). *Lecture VI: Constraints and Controllers.*

Xu, Kangming and Takeru Igusa (1991). "Dynamic characteristics of non-classically damped structures". In: *Earthquake engineering & structural dynamics* 20.12, pp. 1127–1144.

Yamamoto, Yutaka, Kaoru Yamamoto, and Masaaki Nagahara (2016). "Tracking of signals beyond the Nyquist frequency". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, pp. 4003–4008.

Appendix A

Python Code Used in Chapter 3 and 5

```
# import numpy for elementary functions and constants
import numpy as np
# import fftpack for numerically computing the Fourier transform
from scipy.fftpack import fft, ifft
# import matplotlib for plotting
import matplotlib.pyplot as plt
import pandas as pd
from scipy.signal import butter, lfilter, freqz, find_peaks, chirp,
    peak_widths
from math import exp, cos, sin, sqrt, pi
import sympy as sym
from sympy.plotting import plot
from scipy.misc import derivative


#Obtain data from Excel
df = pd.read_csv (r'C:\Users\liset\Documents\Lisette\Year 3\BEP\FFT\dummy.
    csv')
numpy_array = df.to_numpy()
time=numpy_array[:,0]
y_values=numpy_array[:, [2, 5,8]]
bus_y=y_values[:,0]
left_y=y_values[:,1]
right_y=y_values[:,2]

#Define function for low-passfilter

def butter_lowpass(cutoff, fs, order):
    b, a = butter(order, cutoff, btype='low', analog=False)
    return b, a
```

```

def butter_lowpass_filter(data, cutoff, fs, order):
    b, a = butter_lowpass(cutoff, fs, order)
    y = lfilter(b, a, data)
    return y

#Perform FFT

Fs = 256 #sampling frequency (Hz)
T = 1/Fs; #sampling period
L = 52.836*256 #length of signal
t = np.arange(0,L)*T #time vector;
f =np.arange(0,L+2)*Fs #the frequency domain

#Left data
n=len(left_y)
Yleft=fft(left_y) #perform fft of data
P2left=abs(Yleft/L) #two-sided spectrum
P1left=P2left[:int((n/2+1))] #single-sided spectrum
P1left[1:]=2*P1left[1:]

#Right data
Yright=fft(right_y) #perform fft of data
P2right=abs(Yright/L) #two-sided spectrum
P1right=P2right[:int((n/2+1))] #single-sided spectrum
P1right[1:]=2*P1right[1:]

xf=np.arange(0,(Fs/2),Fs/n)

#Filter

cutoff=0.1 #Adviced by Ventura
order=4
yleft = butter_lowpass_filter(P1left, cutoff, Fs, order)
yright = butter_lowpass_filter(P1right, cutoff, Fs, order)


# plot the frequency content
plt.plot(xf, yleft[:int(n/2)],linewidth=1)
plt.title("Left")
plt.xlabel(r"Frequency [$s^{-1}$]")
plt.ylabel(r"Transformed signal in |g| [$\frac{m}{s^2}$]")
plt.show()
```

```
plt.plot(xf, yright[:int(n/2)],linewidth=1)
plt.title("Right")
plt.xlabel(r"Frequency [$s^{-1}$]")
plt.ylabel(r"Transformed signal in |g| [$\frac{m}{s^2}$]")
plt.show()

indexleft=np.where(yleft[:int(n/2)]==np.amax(yleft[:int(n/2)]))
eigenfreqleft=xf[indexleft]
indexright=np.where(yright[:int(n/2)]==np.amax(yright[:int(n/2)]))
eigenfreqright=xf[indexright]
eigenfreq=(eigenfreqleft+eigenfreqright)/2

#..set mass of point mass
m = 80
#..set spring constant of spring
k=m*(2*pi*eigenfreq)**2 #=344531.52293809
#..set damping constant
c= 670

idxleft = np.where(yleft[:int(n/2)]==np.max(yleft[:int(n/2)][(xf>0)&(xf<5)
    ]))[0][0]
freqroadleft=xf[idxleft]
idxright = np.where(yright[:int(n/2)]==np.max(yright[:int(n/2)][(xf>0)&(xf
    <5)]))[0][0]
freqroadright=xf[idxright]
freqroad=(freqroadleft+freqroadright)/2

w0=sqrt(k/m)
zeta=c/(2*sqrt(k*m))
wd=w0*sqrt(1-zeta**2)
w= 2*pi*freqroad #10.730746216426404
F= 25
#..set initial position and velocity
u0 = 0.035
v0 = 0.0
#..set time begin and end forward
q=(m*w**2-k)**2+c**2*w**2
p=(-m*w**2*F+F*k)/q
b=F*c*w/q
c1=u0-p
c2=(v0+zeta*w0*c1-w*b)/wd

tspan=np.linspace(0,10,100)
```

```

#Define functions for position, velocity and acceleration
def u(t):
    return np.exp(-zeta*w0*t)*c1*np.cos(wd*t)+np.exp(-zeta*w0*t)*c2*np.sin
   (wd*t)+p*np.cos(w*t)+b*np.sin(w*t)

def v(t):
    return -w0*zeta*np.exp(-zeta*w0*t)*u(t)+np.exp(-zeta*w0*t)*wd*(-c1*np.
   sin(wd*t)+c2*np.cos(wd*t))-p*w*np.sin(w*t)+w*b*np.cos(w*t)

def a(t):
    return -(k/m)*u(t)-(c/m)*v(t)

def a_in_g(t):
    return a(t)/9.80665

#Verify a(t)

def u_ver(t):
    return exp(-zeta*w0*t)*c1*cos(wd*t)+exp(-zeta*w0*t)*c2*sin(wd*t)+p*cos
   (w*t)+b*sin(w*t)

# calculating its derivative
def v_ver(t):
    return derivative(u, t)

def a_ver(t):
    return derivative(v,t)

def a_ver_in_g(t):
    return a_ver(t)/9.80665

plt.plot(tspan,a_in_g(tspan))
plt.plot(tspan,a_in_g(tspan))
plt.ylim(-0.1,0.1)
plt.show()


#Perform FFT on acceleration
a_t=a_in_g(t)
na=len(a_t)
Ya=fft(a_t)
P2a=abs(Ya/L) #two-sided spectrum
```

```
P1a=P2a[:int((na/2+1))] #single-sided spectrum
P1a[1:]=2*P1a[1:]
xfa=np.arange(0,(Fs/2),Fs/na)

plt.plot(xfa,P1a[:int(na/2+1)])
plt.title("FFT performed on acceleration simple model")
plt.xlim(0,20)
plt.xlabel("Frequency (in Hz)")
plt.ylabel("Transformed signal")
plt.show()

indexmodel=np.where(P1a[:int(n/2)]==np.amax(P1a[:int(n/2)]))
eigenfreqmodel=xfa[indexmodel]

#Estimate c
plt.plot(xf, yleft[:int(n/2)],linewidth=1,label="Left")
plt.plot(xf, yright[:int(n/2)],linewidth=1,label="Right")
plt.plot(xfa,P1a[:int(na/2+1)],linewidth=1,label="Simplified model with c
    =670")
plt.xlim(0,60)
plt.title("FFTs of the acceleration data")
plt.xlabel(r"Frequency [$s^{-1}$]")
plt.ylabel(r"Transformed signal in |g| [$\frac{m}{s^2}$]")
plt.legend()
plt.show()
```

Listing A.1: Code used to visualize Ventura's data and to perform FFTs

Appendix B

Laplace Transform Table

LAPLACE TRANSFORM TABLE

No.	$f(t)$	$F(s)$	No.	$f(t)$	$F(s)$
1.	a	$\frac{a}{s}$	13.	$e^{-at}\sin\omega t$	$\frac{\omega}{(s+a)^2+\omega^2}$
2.	at	$\frac{a}{s^2}$	14.	$e^{-at}\cos\omega t$	$\frac{s+a}{(s+a)^2+\omega^2}$
3.	t^n	$\frac{n!}{s^{n+1}}$	15.	$\sinh\omega t$	$\frac{\omega}{s^2-\omega^2}$
4.	e^{at}	$\frac{1}{s-a}$	16.	$\cosh\omega t$	$\frac{s}{s^2-\omega^2}$
5.	e^{-at}	$\frac{1}{s+a}$	17.	$e^{at}\sinh\omega t$	$\frac{\omega}{(s-a)^2-\omega^2}$
6.	te^{-at}	$\frac{1}{(s+a)^2}$	18.	$e^{-at}\sinh\omega t$	$\frac{\omega}{(s+a)^2-\omega^2}$
7.	$t^n\cdot e^{at}$, $n=1,2,3..$	$\frac{n!}{(s-a)^{n+1}}$	19.	$e^{-at}\cosh\omega t$	$\frac{s+a}{(s+a)^2-\omega^2}$
8.	$t^n\cdot f(t)$	$(-1)^n\frac{d^n}{ds^n}[F(s)]$	20.	$f_1(t)+f_2(t)$	$F_1(s)+F_2(s)$
9.	$\sin\omega t$	$\frac{\omega}{s^2+\omega^2}$	21.	$\int_0^t f(u)du$	$\frac{F(s)}{s}$
10.	$\cos\omega t$	$\frac{s}{s^2+\omega^2}$	22.	$f(t-a)\,u(t-a)$	$e^{-as}F(s)$
11.	$t\sin\omega t$	$\frac{2\omega s}{(s^2+\omega^2)^2}$	23.	*First derivative:* $\frac{dy}{dt}, y'(t)$	$sY(s)-y(0)$
12.	$t\cos\omega t$	$\frac{s^2-\omega^2}{(s^2+\omega^2)^2}$	24.	*Second derivative* $\frac{d^2y}{dt^2}, y''(t)$	$s^2Y(s)-sy(0)$ $-y'(0)$

Figure B.1: Laplace transform table (Daud, Romli, and Ahmad, 2021)

Appendix C

Matlab Code Used in Chapter 6

```
syms w c m k

%Finding the alpha's for the undamped system with external force in the
%x-direction
A_external_x=[1 -1 0;
    0 0 -1;
    k/m 0 0];
b_external_x=[0;
    0;
    1];
alpha_external_x=linsolve(A_external_x,b_external_x);

%Finding the alpha's for the damped system with external force in the
%x-direction
A_damped_external_x=[1 -1 0;
    c/m 0 -1;
    k/m 0 0];
b_damped_external_x=[0;
    0;
    1];
alpha_damped_external_x=linsolve(A_damped_external_x,b_damped_external_x);

%Finding the alpha's for the undamped system with external force in the
%y-direction
A_external_y=[1 0 -1 0;
    0 1 0 -1;
    k/m 0 -w^2 0;
    0 k/m 0 -w^2];
b_external_y=[0;
    (-k/m +w^2);
```

```
    1;
    0];
alpha_external_y=linsolve(A_external_y,b_external_y);

%Finding the alpha's for the damped system with external force in the
%y-direction
A_damped_external_y=[1 0 -1 0;
    c/m 1 0 -1;
    k/m c/m -w^2 0;
    0 k/m 0 -w^2];
b_damped_external_y=[-c/m;
    (-k/m +w^2);
    1;
    0];
alpha_damped_external_y=linsolve(A_damped_external_y,b_damped_external_y);
```

Listing C.1: Code to solve the systems of equation to be able to apply the Laplace inverse

```
m=80;
k=344531.52293809;
c=870;

%No damping
H=tf(1,[1 0 (k/m)]);
[p,z] = pzmap(H);
figure(1);
fig1=zplane(z,p)
hm = findobj(gca, 'Type', 'Line');          % Handle To 'Line' Objects
hm(2).MarkerSize = 14;                      %    Zero    Marker
grid on
saveas(fig1,'pzmapnodamping.png','png')

%Damping
H=tf(1,[1 (c/m) (k/m)]);
[p,z] = pzmap(H);
figure(2);
fig2=zplane(z,p)
hm = findobj(gca, 'Type', 'Line');          % Handle To 'Line' Objects
hm(2).MarkerSize = 14;                      %    Zero    Marker
grid on
saveas(fig2,'pzmapdamping.png','png')
```

Listing C.2: Code to plot the pole-zero map of the transfer functions

Appendix D

Julia Code Used in Chapter 4

```
using LinearAlgebra
using DifferentialEquations
using SparseArrays
using Plots
using Calculus


#..set mass of point mass
m = 80
#..set spring constant of spring
k = 344531.52293809
w0=sqrt(k/m)

#..define the right-hand side of the ordinary differential equation of the equation of motion
function mass_system!(du,u,p,t)
    # solve m \ddot{u} + k u = 0 or \ddot{u} = -(k/m) u
    ddu = -(k/m)*u
end

#..set initial position and velocity
u0 = 3.0
v0 = 0.0
#..set time begin and end forward
tspan = (0.0,1)

#..solution found analytically
u(t)=u0*cos(sqrt(k/m)*t)+(v0/w0)*sin(sqrt(k/m)*t)

#..define ODE problem to be solved
```

```
prob = SecondOrderODEProblem(mass_system!,v0,u0,tspan)

#..solve ODE problem
sol = solve(prob)

#..plot solution of velocity and position as function of time

plot(sol,vars=2,linewidth=3,label="Numerical solution",yaxis="Displacement
    (in m)",xaxis="Time (in s)")
plot!(u,tspan[1],tspan[2],vars=2,ls=:dash,linewidth=2,label="Analytical
    solution")
savefig("noexternalforcesnodamping")
```

Listing D.1: Code to plot both the analytical and numerical solution for the undamped system without external forces

```
#Damped free motion

#..set mass of point mass
m = 80
#..set spring constant of spring
k = 344531.52293809
#..set damping constant
c= 870

w0=sqrt(k/m)
zeta=c/(2*sqrt(k*m))
wd=w0*sqrt(1-zeta^2)


#..define the right-hand side of the ordinary differential equation of the
    equation of motion
function mass_system!(du,u,p,t)
    # solve \ddot{u} = -(k/m) u - (c/m) \dot u + f(t)
    ddu = -(k/m)*u - (c/m)*du
end

#..set initial position and velocity
u0 = 3.0
v0 = 0.0
#..set time begin and end forward
tspan = (0.0,1.0)
c1=u0
c2=(v0+u0*w0*zeta)/wd
```

```
u(t)=exp(-zeta*w0*t)*c1*cos(wd*t)+exp(-zeta*w0*t)*c2*sin(wd*t)

#..define ODE problem to be solved
prob = SecondOrderODEProblem(mass_system!,v0,u0,tspan)

#..solve ODE problem
sol = solve(prob)

#..plot solution of velocity and position as function of time
#How greater the damping the less the two solutions correspond?

plot(sol,vars=2,linewidth=3,label="Numerical solution")
plot!(u,tspan[1],tspan[2],vars=2,ls=:dash,linewidth=2,label="Analytical
   solution")
savefig("noexternalforcesdamping")
```

Listing D.2: Code to plot both the analytical and numerical solution for the damped system without external forces

```
#Undamped external force in x-direction

#..set mass of point mass
m = 80
#..set spring constant of spring
k = 344531.52293809


w0=sqrt(k/m)
F=3000

#..set imposed acceleration on the door
function f(t)
    return F
end

#..define the right-hand side of the ordinary differential equation of the
    equation of motion
function mass_system!(du,u,p,t)
    # solve \ddot{u} = -(k/m) u - (c/m) \dot u + f(t)
    ddu = -(k/m)*u + F/m
end

#..set initial position and velocity
```

```
u0 = 3.0
v0 = 0.0
c1=u0-(F/k)
c2=v0/w0

#..set time begin and end forward
tspan = (0.0,1.0)

u(t)= c1*cos(w0*t)+c2*sin(w0*t)+(F/k)


#..define ODE problem to be solved
prob = SecondOrderODEProblem(mass_system!,v0,u0,tspan)

#..solve ODE problem
sol = solve(prob)

#..plot the source term
tvec = Vector(0.:0.001:1.)
fvec = f.(tvec)
#p1 = plot(tvec,fvec,label="External force")

#..plot solution of velocity and position as function of time
plot(sol,vars=2,linewidth=3,label="Numerical solution",yaxis="Displacement
    (in m)",xaxis="Time (in s)")
plot!(u,tspan[1],tspan[2],vars=2,ls=:dash,linewidth=2,label="Analytical
   solution")
savefig("exxnodamping")
```

Listing D.3: Code to plot both the analytical and numerical solution for the undamped system including an external force in the x-direction

```
#Damped with external force in x-direction

#..set mass of point mass
m = 80
#..set spring constant of spring
k = 344531.52293809
#..set damping constant
c=870


w0=sqrt(k/m)
zeta=c/(2*sqrt(k*m))
```

```
wd=w0*sqrt(1-zeta^2)
F=3000

function f(t)
    return F
end

#..define the right-hand side of the ordinary differential equation of the
     equation of motion
function mass_system!(du,u,p,t)
    # solve \ddot{u} = -(k/m) u - (c/m) \dot u + f(t)
    ddu = -(k/m)*u - (c/m)*du +f(t)/m
end

#..set initial position and velocity
u0 = 3.0
v0 = 0.0
#..set time begin and end forward
tspan = (0.0,1.0)
c1=u0-(F/k)
c2=(v0+w0*zeta*c1)/wd

u(t)=exp(-zeta*w0*t)*c1*cos(wd*t)+exp(-zeta*w0*t)*c2*sin(wd*t)+(F/k)

#..define ODE problem to be solved
prob = SecondOrderODEProblem(mass_system!,v0,u0,tspan)

#..solve ODE problem
sol = solve(prob)

#..plot the source term
tvec = Vector(0.:0.01:10.)
fvec = f.(tvec)
#p1 = plot(tvec,fvec,label="External force",yaxis="Force (in N)",xaxis="
    Time (in s)")

#..plot solution of velocity and position as function of time

plot(sol,vars=2,linewidth=3,label="Numerical solution",yaxis="Displacement
     (in m)",xaxis="Time (in s)")
plot!(u,tspan[1],tspan[2],vars=2,ls=:dash,linewidth=2,label="Analytical
    solution")
```

```
savefig("exxdamped")
```

Listing D.4: Code to plot both the analytical and numerical solution for the damped system including an external force in the x-direction

```
#Undamped external force in y-direction


#..set mass of point mass
m = 80
#..set spring constant of spring
k = 344531.52293809
w0=sqrt(k/m)
w=60
F=3000
#..set imposed acceleration on the door
function f(t)
    return F*cos(w*t)
end

#..define the right-hand side of the ordinary differential equation of the
    equation of motion
function mass_system!(du,u,p,t)
    # solve \ddot{u} = -(k/m) u - (c/m) \dot u + f(t)
    ddu = -(k/m)*u + f(t)/m
end

#..set initial position and velocity
u0 = 3.0
v0 = 0.0
c1=u0-(F/(m*(w0^2-w^2)))
c2=v0/w0

#..set time begin and end forward
tspan = (0.0,1.0)

u(t)= c1*cos(w0*t)+c2*sin(w0*t)+(F/(m*(w0^2-w^2)))*cos(w*t)


#..define ODE problem to be solved
prob = SecondOrderODEProblem(mass_system!,v0,u0,tspan)

#..solve ODE problem
sol = solve(prob)
```

```

#..plot the source term
tvec = Vector(0.:0.001:1.)
fvec = f.(tvec)
#p1 = plot(tvec,fvec,label="External force")

#..plot solution of velocity and position as function of time
plot(sol,vars=2,linewidth=3,label="Numerical solution")
plot!(u,tspan[1],tspan[2],vars=2,ls=:dash,linewidth=2,label="Analytical
    solution")
savefig("exynodamping")
```

Listing D.5: Code to plot both the analytical and numerical solution for the undamped system including an external force in the y-direction

```
#Undamped external force in y-direction as w=w0

#..set mass of point mass
m = 80
#..set spring constant of spring
k = 344531.52293809
w0=sqrt(k/m)
w=w0
F=3000



#..set imposed acceleration on the door
function f(t)
    return F*cos(w*t)
end

#..define the right-hand side of the ordinary differential equation of the
     equation of motion
function mass_system!(du,u,p,t)
    # solve \ddot{u} = -(k/m) u - (c/m) \dot u + f(t)
    ddu = -(k/m)*u + f(t)/m
end

#..set initial position and velocity
u0 = 3.0
v0 = 0.0
c1=u0
c2=v0/w0
```

```

#..set time begin and end forward
tspan = (0.0,1.0)

u(t)= c1*cos(w0*t)+c2*sin(w0*t)+(F/(2*w*m))*t*sin(w0*t)


#..define ODE problem to be solved
prob = SecondOrderODEProblem(mass_system!,v0,u0,tspan)

#..solve ODE problem
sol = solve(prob)

#..plot the source term
tvec = Vector(0.:0.01:1.)
fvec = f.(tvec)
#p1 = plot(tvec,fvec,label="External force",yaxis="Force (in N)",xaxis="
    Time (in s)")

#..plot solution of velocity and position as function of time
plot(sol,vars=2,linewidth=3,label="Numerical solution",yaxis="Displacement
     (in m)",xaxis="Time (in s)")
plot!(u,tspan[1],tspan[2],vars=2,ls=:dash,linewidth=2,label="Analytical
    solution")
savefig("exynodampingw0eqw")
```

Listing D.6: Code to plot both the analytical and numerical solution for the undamped system including an external force in the y-direction with $\omega = \omega_0$

```
#Damped motion with external force in the y-direction


#..set mass of point mass
m = 80
#..set spring constant of spring
k = 344531.52293809
#..set damping constant
c = 870

w0=sqrt(k/m)
zeta=c/(2*sqrt(k*m))
wd=w0*sqrt(1-zeta^2)
w=60
F=3000
```

```

function f(t)
    return F*cos(w*t)
end

#..define the right-hand side of the ordinary differential equation of the
    equation of motion
function mass_system!(du,u,p,t)
    # solve \ddot{u} = -(k/m) u - (c/m) \dot u + f(t)
    ddu = -(k/m)*u - (c/m)*du +f(t)/m
end

#..set initial position and velocity
u0 = 3.0
v0 = 0.0
#..set time begin and end forward
tspan = (0.0,1.0)
q=(m*w^2-k)^2+c^2*w^2
p=(-m*w^2*F+F*k)/q
b=F*c*w/q
c1=u0-p
c2=(v0+zeta*w0*c1-w*b)/wd

u(t)=exp(-w0*zeta*t)*c1*cos(wd*t)+exp(-w0*zeta*t)*c2*sin(wd*t)+p*cos(w*t)+
   b*sin(w*t)

#..define ODE problem to be solved
prob = SecondOrderODEProblem(mass_system!,v0,u0,tspan)

#..solve ODE problem
sol = solve(prob)

#..plot the source term
tvec = Vector(0.:0.001:1.)
fvec = f.(tvec)
#p1 = plot(tvec,fvec,label="External force",yaxis="Force (in N)",xaxis="
   Time (in s)")

#..plot solution of velocity and position as function of time

plot(sol,vars=2,linewidth=3,label="Numerical solution",yaxis="Displacement
    (in m)",xaxis="Time (in s)")
plot!(u,tspan[1],tspan[2],vars=2,ls=:dash,linewidth=2,label="Analytical
```

```
    solution")
savefig("exydamping")
```

Listing D.7: Code to plot both the analytical and numerical solution for the damped system including an external force in the y-direction

Appendix E

Julia Code Used in Chapter 7

```
using LinearAlgebra
using DifferentialEquations
using SparseArrays
using Plots
using Calculus

#Undamped no external force

#Define the spring constants
k1 = 10000
k2 = 20000
k3 = 10000

#Define the masses
m1=40
m2=40

#Define the needed matrices
K  = [(k1+k2) -k2
    -k2 (k2+k3)]
M=[m1 0
    0 m2]
Ktau=M^(-.5)*K*M^(-.5)

#Find eigenvectors and eigenvalues (they are already normalized?)
eigenvectors=eigvecs(Ktau)
wn_1,wn_2=eigvals(Ktau)


#Create two orthogonal eigenvectors and create matrix P
```

```
eigenvector1=eigenvectors[:,1]
eigenvector2=[-eigenvector1[2]
    eigenvector1[1]]
P=[eigenvector1 eigenvector2]



function mass_system!(du,u,p,t)
    # solve m \ddot{u} + k u = 0 or \ddot{u} = -(k/m) u
    ddu = (inv(M)*-K)*u
end


#..set time begin and end forward
tspan = (0.0,1.0)
#Define initial conditions
u0 = [3
    0]
v0= [1
    0]

r0=transpose(P)*M^(0.5)*u0
r_dot_0=transpose(P)*M^(0.5)*v0
c1=r0[1]
c2=r_dot_0[1]/sqrt(wn_1)
c3=r0[2]
c4=r_dot_0[2]/sqrt(wn_2)
r(t)=[c1*cos(sqrt(wn_1)*t)+c2*sin(sqrt(wn_1)*t)
c3*cos(sqrt(wn_2)*t)+c4*sin(sqrt(wn_2)*t)]

u(t)=M^(-0.5)*P*r(t)
u1(t)=u(t)[1]
u2(t)=u(t)[2]

#..define ODE problem to be solved
prob = SecondOrderODEProblem(mass_system!,v0,u0,tspan)


#..solve ODE problem
sol = solve(prob)

#..plot solution of velocity and position as function of time
plot(sol,vars=[3,4],linewidth=3,label=["Numerical solution for mass 1" "
```

```
    Numerical solution for mass 2"],yaxis="Displacement in the x-direction
    [m]",xaxis="Time [s]")
plot!(u1,tspan[1],tspan[2],vars=2,ls=:dash,color=:yellow,label="Analytical
     solution for mass 1")
plot!(u2,tspan[1],tspan[2],vars=2,ls=:dash,linewidth=2,color=:green,label=
    "Analytical solution for mass 2")
savefig("2DOFnoexternalforcesnodamping")
```

Listing E.1: Code to plot both the analytical and numerical solution for the undamped system without external forces

```
#Damped no external force

#Define the spring constants
k1 = 10000
k2 = 20000
k3 = 10000
c1 = 100
c2 = 300
c3 = 100

#Define the masses
m1=40
m2=40
#Define the needed matrices
K  = [(k1+k2) -k2
    -k2 (k2+k3)]
C = [(c1+c2) -c2
    -c2 (c2+c3)]
M=[m1 0
    0 m2]
Ktau=M^(-.5)*K*M^(-.5)
Ctau=M^(-.5)*C*M^(-.5)

#Find eigenvectors and eigenvalues (they are already normalized?)
eigenvectors=eigvecs(Ktau)
ev1,ev2=eigvals(Ktau)


#Create two orthogonal eigenvectors and create matrix P
eigenvector1=eigenvectors[:,1]
eigenvector2=[-eigenvector1[2]
    eigenvector1[1]]
P=[eigenvector1 eigenvector2]
```

```

K_new=transpose(P)*Ktau*P
C_new=transpose(P)*Ctau*P
dampingratio1=C_new[1]/(2*sqrt(ev1))
dampingratio2=C_new[4]/(2*sqrt(ev2))


wd1=sqrt(ev1)*sqrt(1-dampingratio1^2)
wd2=sqrt(ev2)*sqrt(1-dampingratio2^2)

function mass_system!(du,u,p,t)
    # solve m \ddot{u} + k u = 0 or \ddot{u} = -(k/m) u
    ddu = inv(M)*-C*du-(inv(M)*K)*u
end


#..set time begin and end forward
tspan = (0.0,1.0)
#Define initial conditions
u0 = [3
    0]
v0= [1
    0]

r0=transpose(P)*M^(0.5)*u0
r_dot_0=transpose(P)*M^(0.5)*v0
c1=r0[1]
c2=(r_dot_0[1]+r0[1]*sqrt(ev1)*dampingratio1)/sqrt(ev1)
c3=r0[2]
c4=(r_dot_0[2]+r0[2]*sqrt(ev2)*dampingratio2)/sqrt(ev2)
print(ev1)
print(ev2)
r(t)=[exp(-dampingratio1*sqrt(ev1)*t)*(c1*cos(wd1*t)+c2*sin(wd1*t))
exp(-dampingratio2*sqrt(ev2)*t)*(c3*cos(wd2*t)+c4*sin(wd2*t))]

u(t)=M^(-0.5)*P*r(t)
u1(t)=u(t)[1]
u2(t)=u(t)[2]

#..define ODE problem to be solved
prob = SecondOrderODEProblem(mass_system!,v0,u0,tspan)


```

```
#..solve ODE problem
sol = solve(prob)

#..plot solution of velocity and position as function of time
plot(sol,vars=[3,4],linewidth=3,label=["Numerical solution for mass 1" "
    Numerical solution for mass 2"],yaxis="Displacement in x-direction [m]"
    ,xaxis="Time [s]")
plot!(u1,tspan[1],tspan[2],vars=2,ls=:dash,color=:yellow,label="Analytical
     solution for mass 1")
plot!(u2,tspan[1],tspan[2],vars=2,ls=:dash,linewidth=2,color=:green,label=
    "Analytical solution for mass 2")
savefig("2DOFnoexternalforcesclasdamped")
```

Listing E.2: Code to plot both the analytical and numerical solution for the classically damped system without external forces

```
#Damped no external force

#Define the spring constants
k1 = 10000
k2 = 30000
k3 = 10000
c1 = 200
c2 = 230
c3 = 300

#Define the masses
m1=40
m2=40
#Define the needed matrices
K  = [(k1+k2) -k2
     -k2 (k2+k3)]
C = [(c1+c2) -c2
     -c2 (c2+c3)]
M=[m1 0
     0 m2]
Ktau=M^(-.5)*K*M^(-.5)
Ctau=M^(-.5)*C*M^(-.5)

#Find eigenvectors and eigenvalues (they are already normalized?)
eigenvectors=eigvecs(Ktau)
ev1,ev2=eigvals(Ktau)
```

```
#Create two orthogonal eigenvectors and create matrix P
eigenvector1=eigenvectors[:,1]
eigenvector2=[-eigenvector1[2]
    eigenvector1[1]]
P=[eigenvector1 eigenvector2]

K_new=transpose(P)*Ktau*P
C_new=transpose(P)*Ctau*P
dampingratio1=C_new[1]/(2*sqrt(ev1))
dampingratio2=C_new[4]/(2*sqrt(ev2))

print(C_new)
wd1=sqrt(ev1)*sqrt(1-dampingratio1^2)
wd2=sqrt(ev2)*sqrt(1-dampingratio2^2)

function mass_system!(du,u,p,t)
    # solve m \ddot{u} + k u = 0 or \ddot{u} = -(k/m) u
    ddu = inv(M)*-C*du-(inv(M)*K)*u
end


#..set time begin and end forward
tspan = (0.0,1.0)
#Define initial conditions
u0 = [3
    0]
v0= [1
    0]

r0=transpose(P)*M^(0.5)*u0
r_dot_0=transpose(P)*M^(0.5)*v0
c1=r0[1]
c2=(r_dot_0[1]+r0[1]*sqrt(ev1)*dampingratio1)/sqrt(ev1)
c3=r0[2]
c4=(r_dot_0[2]+r0[2]*sqrt(ev2)*dampingratio2)/sqrt(ev2)
print(ev1)
print(ev2)
r(t)=[exp(-dampingratio1*sqrt(ev1)*t)*(c1*cos(wd1*t)+c2*sin(wd1*t))
exp(-dampingratio2*sqrt(ev2)*t)*(c3*cos(wd2*t)+c4*sin(wd2*t))]

u(t)=M^(-0.5)*P*r(t)
u1(t)=u(t)[1]
u2(t)=u(t)[2]
```

```

#..define ODE problem to be solved
prob = SecondOrderODEProblem(mass_system!,v0,u0,tspan)


#..solve ODE problem
sol = solve(prob)

#..plot solution of velocity and position as function of time
plot(sol,vars=[3,4],linewidth=3,label=["Numerical solution for mass 1" "
    Numerical solution for mass 2"],yaxis="Displacement in x-direction [m]"
    ,xaxis="Time [s]")
plot!(u1,tspan[1],tspan[2],vars=2,ls=:dash,color=:yellow,label="Analytical
     solution for mass 1")
plot!(u2,tspan[1],tspan[2],vars=2,ls=:dash,linewidth=2,color=:green,label=
    "Analytical solution for mass 2")
savefig("2DOFnoexternalforcesnonclasdamped")
```

Listing E.3: Code to plot both the analytical and numerical solution for the non-classically damped system without external forces

Appendix F

Julia Code Used in Chapter 8

```julia
#Undamped free motion

m = 80
#..set spring constant of spring
k1x = 1.0
k1y = 3.0
k2x = 2.0
k2y = 1.0

K= [k1x+k2x 0
    0 k2x+k2y]
M=[m 0
    0 m]
wx=sqrt((k1x+k2x)/m)
wy=sqrt((k1y+k2y)/m)

#..set imposed acceleration on the door
function f(t)
    #return exp(-(t-2)^2/0.01)
    #return t>=2
    return 0
end

#..define the right-hand side of the ordinary differential equation of the equation of motion
function mass_system2!(ddu,du,u,p,t)
    # solve \ddot{u} = -(k/m) u - (c/m) \dot u + f(t)
    ddu[1] = -((k1x+k2x)/m)*u[1] + f(t)
    ddu[2] = -((k1y+k2y)/m)*u[2] + f(t)
end
```

```julia

#..set initial position and velocity
u0 = [2.0, 1.0]
v0 = [1.0, 0.0]
#..set time begin and end forward
tspan = (0.0,20.0)

#..define ODE problem to be solved
prob = SecondOrderODEProblem(mass_system2!,v0,u0,tspan)

#..solve ODE problem
sol = solve(prob)

#..plot the source term
tvec = Vector(0.:0.01:10.)
fvec = f.(tvec)

#Define analytical solution
u(t)=[u0[1]*cos(wx*t)+(v0[1]/wx)*sin(wx*t)
    u0[2]*cos(wy*t)+(v0[2]/wy)*sin(wy*t)]
u1(t)=u(t)[1]
u2(t)=u(t)[2]


p1 = plot(sol,vars=(3,4),linewidth=2,label="Numerical solution")
xlabel!("Displacement in x-direction [m]")
ylabel!("Displacement in y-direction [m]")

#..plot solution of velocity and position as function of time
plot(p1)
plot!(u1,u2,tspan[1],tspan[2],ls=:dash,linewidth=2,label="Analytical solution")
savefig("2dimnoexternalforcesnodamping")
```

Listing F.1: Code to plot both the analytical and numerical solution for the undamped system without external forces

```julia
m = 80
#..set spring constant of spring
k1x = 8000
k1y = 9000
k2x = 5000
k2y = 1000
c1x = 200
```

```
c2x = 100
c1y = 200
c2y = 100

#..set damping constant
wx=sqrt((k1x+k2x)/m)
wy=sqrt((k1y+k2y)/m)
zetax=(c1x+c2x)/(2*sqrt((k1x+k2x)*m))
zetay=(c1y+c2y)/(2*sqrt((k1y+k2y)*m))
wdx=wx*sqrt(1-zetax^2)
wdy=wy*sqrt(1-zetay^2)

#..set imposed acceleration on the door
function f(t)
    #return exp(-(t-2)^2/0.01)
    #return t>=2
    return 0
end

#..define the right-hand side of the ordinary differential equation of the
    equation of motion
function mass_system2!(ddu,du,u,p,t)
    # solve \ddot{u} = -(k/m) u - (c/m) \dot u + f(t)
    ddu[1] = -((c1x+c2x)/m)*du[1]-((k1x+k2x)/m)*u[1] + f(t)
    ddu[2] = -((c1y+c2y)/m)*du[2]-((k1y+k2y)/m)*u[2] + f(t)
end

#..set initial position and velocity
u0 = [2.0, 1.0]
v0 = [1.0, 0.0]
#..set time begin and end forward
tspan = (0.0,10.0)

#..define ODE problem to be solved
prob = SecondOrderODEProblem(mass_system2!,v0,u0,tspan)

#..solve ODE problem
sol = solve(prob)

#..plot the source term
tvec = Vector(0.:0.01:10.)
fvec = f.(tvec)
c1=u0[1]
```

```
c3=u0[2]
c2=(v0[1]+u0[1]*wx*zetax)/wdx
c4=(v0[2]+u0[2]*wy*zetay)/wdy


#Define analytical solution
u(t)=[exp(-wx*zetax*t)*(c1*cos(wdx*t)+c2*sin(wdx*t))
exp(-wy*zetay*t)*(c3*cos(wdy*t)+c4*sin(wdy*t))]
u1(t)=u(t)[1]
u2(t)=u(t)[2]

p1 = plot(sol,vars=(3,4),linewidth=2,label="Numerical solution")
xlabel!("Displacement in x-direction [m]")
ylabel!("Displacement in y-direction [m]")

#..plot solution of velocity and position as function of time
plot(p1)
plot!(u1,u2,tspan[1],tspan[2],ls=:dash,linewidth=2,label="Analytical
    solution")
savefig("2dimnoexternalforcesdamping")
```

Listing F.2: Code to plot both the analytical and numerical solution for the damped system without external forces

```
m = 80
#..set spring constant of spring
k1x = 8000
k1y = 9000
k2x = 5000
k2y = 1000
c1x = 200
c2x = 100
c1y = 200
c2y = 100

#..external force
F=200

#..set damping constant
wx=sqrt((k1x+k2x)/m)
wy=sqrt((k1y+k2y)/m)
zetax=(c1x+c2x)/(2*sqrt((k1x+k2x)*m))
zetay=(c1y+c2y)/(2*sqrt((k1y+k2y)*m))
wdx=wx*sqrt(1-zetax^2)
```

```
wdy=wy*sqrt(1-zetay^2)

#..set imposed acceleration on the door
function f(t)
    #return exp(-(t-2)^2/0.01)
    #return t>=2
    return 0
end

#..define the right-hand side of the ordinary differential equation of the
     equation of motion
function mass_system2!(ddu,du,u,p,t)
    # solve \ddot{u} = -(k/m) u - (c/m) \dot u + f(t)
    ddu[1] = -((c1x+c2x)/m)*du[1]-((k1x+k2x)/m)*u[1]
    ddu[2] = -((c1y+c2y)/m)*du[2]-((k1y+k2y)/m)*u[2] + F/m
end

#..set initial position and velocity
u0 = [2.0, 1.0]
v0 = [1.0, 0.0]
#..set time begin and end forward
tspan = (0.0,10.0)

#..define ODE problem to be solved
prob = SecondOrderODEProblem(mass_system2!,v0,u0,tspan)

#..solve ODE problem
sol = solve(prob)

#..plot the source term
tvec = Vector(0.:0.01:10.)
fvec = f.(tvec)
c1=u0[1]
c3=u0[2]-F/(k1y+k2y)
c2=(v0[1]+u0[1]*wx*zetax)/wdx
c4=(v0[2]+u0[2]*wy*zetay*c3)/wdy


#Define analytical solution
u(t)=[exp(-wx*zetax*t)*(c1*cos(wdx*t)+c2*sin(wdx*t))
exp(-wy*zetay*t)*(c3*cos(wdy*t)+c4*sin(wdy*t))+F/(k1y+k2y)]
u1(t)=u(t)[1]
u2(t)=u(t)[2]
```

```

p1 = plot(sol,vars=(3,4),linewidth=2,label="Numerical solution")
xlabel!("Displacement in x-direction [m]")
ylabel!("Displacement in y-direction [m]")

#..plot solution of velocity and position as function of time
plot(p1)
plot!(u1,u2,tspan[1],tspan[2],ls=:dash,linewidth=2,label="Analytical
    solution")
savefig("2dimexxdamping")
```

Listing F.3: Code to plot both the analytical and numerical solution for the damped system including constant external force

```
m = 80
#..set spring constant of spring
k1x = 8000
k1y = 9000
k2x = 5000
k2y = 1000
c1x = 200
c2x = 100
c1y = 200
c2y = 100

#..external force
F=200
w=10

#..set damping constant
wx=sqrt((k1x+k2x)/m)
wy=sqrt((k1y+k2y)/m)
zetax=(c1x+c2x)/(2*sqrt((k1x+k2x)*m))
zetay=(c1y+c2y)/(2*sqrt((k1y+k2y)*m))
wdx=wx*sqrt(1-zetax^2)
wdy=wy*sqrt(1-zetay^2)

#..set imposed acceleration on the door
function f(t)
    #return exp(-(t-2)^2/0.01)
    #return t>=2
    return 0
end

```

```julia
#..define the right-hand side of the ordinary differential equation of the equation of motion
function mass_system2!(ddu,du,u,p,t)
    # solve \ddot{u} = -(k/m) u - (c/m) \dot u + f(t)
    ddu[1] = -((c1x+c2x)/m)*du[1]-((k1x+k2x)/m)*u[1]
    ddu[2] = -((c1y+c2y)/m)*du[2]-((k1y+k2y)/m)*u[2] + F*cos(w*t)/m
end

#..set initial position and velocity
u0 = [2.0, 1.0]
v0 = [1.0, 0.0]
#..set time begin and end forward
tspan = (0.0,10.0)

#..define ODE problem to be solved
prob = SecondOrderODEProblem(mass_system2!,v0,u0,tspan)

#..solve ODE problem
sol = solve(prob)

p=(-m*w^2*F+F*(k1y+k2y))
q=(m*w^2-(k1y+k2y))^2+(c1y+c2y)^2*w^2
#..plot the source term
tvec = Vector(0.:0.01:10.)
fvec = f.(tvec)
c1=u0[1]
c3=u0[2]-p/q
c2=(v0[1]+u0[1]*wx*zetax)/wdx
c4=(v0[2]+u0[2]*wy*zetay*c3-(F*(c1y+c2y)*w^2)/q)/wdy


#Define analytical solution
u(t)=[exp(-wx*zetax*t)*(c1*cos(wdx*t)+c2*sin(wdx*t))
exp(-wy*zetay*t)*(c3*cos(wdy*t)+c4*sin(wdy*t))+p/q*cos(w*t)+(F*(c1y+c2y)*w)/q*sin(w*t)]
u1(t)=u(t)[1]
u2(t)=u(t)[2]

p1 = plot(sol,vars=(3,4),linewidth=2,label="Numerical solution")
xlabel!("Displacement in x-direction [m]")
ylabel!("Displacement in y-direction [m]")

#..plot solution of velocity and position as function of time
```

```
plot(p1)
plot!(u1,u2,tspan[1],tspan[2],ls=:dash,linewidth=2,label="Analytical
    solution")
savefig("2dimexydamping")
```

Listing F.4: Code to plot both the analytical and numerical solution for the damped system including periodic external force

Appendix G

Julia Code Used in Chapter 8

```
#Damped no external force

#Define the spring constants
k1x = 8000
k1y = 9000
k2x = 5000
k2y = 1000
k3x = 8000
k3y = 9000
c1x = 200
c2x = 100
c1y = 200
c2y = 100
c3x=200
c3y=200

#Define the masses
m1=40
m2=40
#Define the needed matrices
Kx  = [(k1x+k2x) -k2x
    -k2x (k2x+k3x)]
Cx = [(c1x+c2x) -c2x
    -c2x (c2x+c3x)]
Ky  = [(k1y+k2y) -k2y
    -k2y (k2y+k3y)]
Cy = [(c1y+c2y) -c2y
    -c2y (c2y+c3y)]
M=[m1 0
    0 m2]
```

```
Ktaux=M^(-.5)*Kx*M^(-.5)
Ctaux=M^(-.5)*Cx*M^(-.5)
Ktauy=M^(-.5)*Ky*M^(-.5)
Ctauy=M^(-.5)*Cy*M^(-.5)

#Find eigenvectors and eigenvalues (they are already normalized?)
eigenvectorsx=eigvecs(Ktaux)
ev1x,ev2x=eigvals(Ktaux)

eigenvectorsy=eigvecs(Ktauy)
ev1y,ev2y=eigvals(Ktauy)

#Create two orthogonal eigenvectors and create matrix P
eigenvector1x=eigenvectorsx[:,1]
eigenvector2x=[-eigenvector1x[2]
    eigenvector1x[1]]
Px=[eigenvector1x eigenvector2x]

eigenvector1y=eigenvectorsy[:,1]
eigenvector2y=[-eigenvector1y[2]
    eigenvector1y[1]]
Py=[eigenvector1y eigenvector2y]

K_newx=transpose(Px)*Ktaux*Px
C_newx=transpose(Px)*Ctaux*Px
K_newy=transpose(Py)*Ktauy*Py
C_newy=transpose(Py)*Ctauy*Py



dampingratio1x=C_newx[1]/(2*sqrt(ev1x))
dampingratio2x=C_newx[4]/(2*sqrt(ev2x))
dampingratio1y=C_newy[1]/(2*sqrt(ev1y))
dampingratio2y=C_newy[4]/(2*sqrt(ev2y))


wd1x=sqrt(ev1x)*sqrt(1-dampingratio1x^2)
wd2x=sqrt(ev2x)*sqrt(1-dampingratio2x^2)
wd1y=sqrt(ev1y)*sqrt(1-dampingratio1y^2)
wd2y=sqrt(ev2y)*sqrt(1-dampingratio2y^2)

function mass_systemx!(du,u,p,t)
    # solve m \ddot{u} + k u = 0 or \ddot{u} = -(k/m) u
```

```
    ddu = inv(M)*-Cx*du-(inv(M)*Kx)*u
end

function mass_systemy!(du,u,p,t)
    # solve m \ddot{u} + k u = 0 or \ddot{u} = -(k/m) u
    ddu = inv(M)*-Cy*du-(inv(M)*Ky)*u
end

#..set time begin and end forward
tspan = (0.0,1.0)
#Define initial conditions
u0x = [3
    0]
v0x= [1
    0]

u0y = [3
    0]
v0y= [1
    0]


r0x=transpose(Px)*M^(0.5)*u0x
r_dot_0x=transpose(Px)*M^(0.5)*v0x
r0y=transpose(Py)*M^(0.5)*u0y
r_dot_0y=transpose(Py)*M^(0.5)*v0y

c1x=r0x[1]
c2x=(r_dot_0x[1]+r0x[1]*sqrt(ev1x)*dampingratio1x)/sqrt(ev1x)
c1y=r0y[1]
c2y=(r_dot_0y[1]+r0y[1]*sqrt(ev1y)*dampingratio1y)/sqrt(ev1y)

c3x=r0x[2]
c4x=(r_dot_0x[2]+r0x[2]*sqrt(ev2x)*dampingratio2x)/sqrt(ev2x)
c3y=r0y[2]
c4y=(r_dot_0y[2]+r0y[2]*sqrt(ev2y)*dampingratio2y)/sqrt(ev2y)

rx(t)=[exp(-dampingratio1x*sqrt(ev1x)*t)*(c1x*cos(wd1x*t)+c2x*sin(wd1x*t))
exp(-dampingratio2x*sqrt(ev2x)*t)*(c3x*cos(wd2x*t)+c4x*sin(wd2x*t))]
ry(t)=[exp(-dampingratio1y*sqrt(ev1y)*t)*(c1y*cos(wd1y*t)+c2y*sin(wd1y*t))
exp(-dampingratio2y*sqrt(ev2y)*t)*(c3y*cos(wd2y*t)+c4y*sin(wd2y*t))]

ux(t)=M^(-0.5)*Px*rx(t)
```

```
uy(t)=M^(-0.5)*Py*ry(t)

u1x(t)=ux(t)[1]
u2x(t)=ux(t)[2]
u1y(t)=uy(t)[1]
u2y(t)=uy(t)[2]

#..define ODE problem to be solved
probx = SecondOrderODEProblem(mass_systemx!,v0x,u0x,tspan)
proby = SecondOrderODEProblem(mass_systemy!,v0y,u0y,tspan)


#..solve ODE problem
solx = solve(probx)
soly = solve(proby)

#..plot solution of velocity and position as function of time
#plot(solx,vars=[3,4],linewidth=3,label=["Numerical solution for mass 1" "Numerical solution for mass 2"],yaxis="Displacement in the x-direction [m]",xaxis="Time [s]")
#plot!(u1x,tspan[1],tspan[2],vars=2,ls=:dash,color=:yellow,label="Analytical solution for mass 1")
#plot!(u2x,tspan[1],tspan[2],vars=2,ls=:dash,linewidth=2,color=:green,label="Analytical solution for mass 2")

plot(soly,vars=[3,4],linewidth=3,label=["Numerical solution for mass 1" "Numerical solution for mass 2"],yaxis="Displacement in the y-direction [m]",xaxis="Time [s]")
plot!(u1y,tspan[1],tspan[2],vars=2,ls=:dash,color=:yellow,label="Analytical solution for mass 1")
plot!(u2y,tspan[1],tspan[2],vars=2,ls=:dash,linewidth=2,color=:green,label="Analytical solution for mass 2")

#plot(sol,vars=[3,4],linewidth=3,label=["Numerical solution for mass 1" "Numerical solution for mass 2"],yaxis="Displacement (in m)",xaxis="Time (in s)")
#plot!(u1,tspan[1],tspan[2],vars=2,ls=:dash,color=:yellow,label="Analytical solution for mass 1")
#plot!(u2,tspan[1],tspan[2],vars=2,ls=:dash,linewidth=2,color=:green,label="Analytical solution for mass 2")
```

Listing G.1: Code to plot both the analytical and numerical solution for the damped system