ISBN 978-82-471-2774-2 (printed ver.)
ISBN 978-82-471-2776-6 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2011:119

Magne Bratland

Modal Analysis of Active Flexible Multibody Systems in a Finite Element Environment

Magne Bratland

Doctoral Thesis

NTNU
Norwegian University of
Science and Technology
Thesis for the degree of
philosophiae doctor
Faculty of Engineering Science and Technology
Department of Engineering Design and Materials

NTNU
Norwegian University of
Science and Technology

Doctoral theses at NTNU, 2011:119

NTNU

NTNU
Norwegian University of
Science and Technology

Magne Bratland

Modal Analysis of Active Flexible Multibody Systems in a Finite Element Environment

Thesis for the degree of philosophiae doctor

Trondheim, February 2011

Norwegian University of
Science and Technology
Faculty of Engineering Science and Technology
Department of Engineering Design and Materials

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of philosophiae doctor

Faculty of Engineering Science and Technology
Department of Engineering Design and Materials

ISBN 978-82-471-2774-2 (printed ver.)
ISBN 978-82-471-2776-6 (electronic ver.)
ISSN 1503-8181

Doctoral Theses at NTNU, 2011:119

Printed by Tapir Uttrykk

NTNU – Trondheim
Norwegian University of
Science and Technology

DOCTORAL THESIS

Modal Analysis of Active Flexible Multibody Systems in a Finite Element Environment

by

Magne Bratland

Supervisor: Terje Rølvåg

Co-supervisors: Bjørn Haugen
Ole Ivar Sivertsen
Knut Morten Okstad

Faculty of Engineering Science and Technology

Department of Engineering Design and Materials

Trondheim, 2011

Preface

The present doctoral thesis is submitted to the Norwegian University of Science and Technology (NTNU) for the degree of Philosophiae Doctor (Ph.D.). The work has been carried out at the Department of Engineering Design and Materials (IPM) under the supervision of Professor Terje Rølvåg (NTNU), Associate Professor Bjørn Haugen (NTNU), Professor Ole Ivar Sivertsen (NTNU) and Dr. Ing. Knut Morten Okstad (Fedem Technology AS). The research has been funded by the Research Council of Norway and the Lean Product Development (LPD) project.

Acknowledgements

The idea for me to embark on a Ph.D. study was sparked during the final year of my master's degree by Professor Terje Rølvåg and Professor Sjur Dagestad. My first thanks goes to Professor Rølvåg, who has been the main supervisor of my Ph.D. project, for believing in me and, by arranging my scholarship, giving me the opportunity to work with such an interesting topic. Professor Rølvåg has involved himself in this Ph.D. project, not only professionally but also personally, by offering me his friendship and engaging me in lots of interesting and often off-topic conversations and activities.

Associate Professor Bjørn Haugen entered my Ph.D. project after six months, and has been my main source of professional and technical guidance. Associate Professor Haugen has been a patient supervisor, taking time to answer many of my questions, both the intelligent and not so intelligent ones, and thoroughly teaching me in regard to relevant special fields.

Professor Rølvåg and Associate Professor Haugen, together with Professor Ole Ivar Sivertsen and Dr. Ing. Knut Morten Okstad, have been my supervisory team, a team which has been of tremendous help during my Ph.D. project. I would like to give a special thanks to Professor Sivertsen for his help and support. An additional thanks goes to Professor Kristian Tønder, who has been of great assistance during my Ph.D. project.

To Professor Sjur Dagestad, I would like to offer my thanks for encouraging me to pursue a doctoral degree, and for bringing me the words of wisdom relevant to so many aspects of life: "You can decide if you're going to make it, or you can decide if you're not going to make it; and in both cases, you will be right."

I would like to thank my many colleagues at IPM, particularly Fredrik Widerøe and Martin Gudem, with whom I have shared an office with for two years, Anna Smirnova, John Harald Grave, Jan Magnus Granheim Farstad and Ali Cetin, for many fruitful discussions and enjoyable social activities in an otherwise labor-intensive schedule.

A special thanks goes to my family: my mother, father and sister for enabling, encouraging and challenging me to pursue my dreams and providing me with support, and always being there for me; and to my dear friends, all of you, for your support and providing me with enjoyable times, and in general, doing what friends do best.

Finally, a very special thanks to Lise for loving me, supporting me, encouraging me, enduring with me, being my best friend and sharing your life with me on a daily basis.

Trondheim, February 2011

Magne Bratland

Motivation

The main motivation behind this Ph.D. project can be traced back to Professor Rølvåg's work with the European Space Agency (ESA) and various projects with contractors in the military industry. From 1996 to 1999, ESA, Matra Marconi Space and Fedem Technology AS worked together in a project called the "SAR (Synthetic Aperture Radar) Deployment Modelling Project" [1]. The purpose of the project was to develop a software tool capable of simulating the deployment and latching of space mechanisms. From about the years 2000 to 2003, Professor Rølvåg was working on designing and simulating the IDG (Integrated Director Group) radar, which was to be installed on the new frigates of the Royal Norwegian Navy. Due to the intended use of the radar, very strict requirements were set to the resonance frequencies of the radar. During the design process of the radar, over 2 000 simulations were performed on virtual models of the radar in FEDEM. The radar is an active mechanism, meaning two separate models of the radar had to be created, one for time domain simulations and one for modal analyses. This is because the dynamical effects caused by the active elements are not included in modal analyses in FEDEM, or to the best of the author's knowledge, any other flexible multibody systems simulation software. Instead, they have to be included in the virtual model as passive elements when performing modal analyses. This meant that for each design suggestion, two models and simulations of the suggested design had to be performed in order to see if the radar design met the initial requirements. This inefficiency gave life to the idea of a simulation tool which could include system properties of both active and passive elements in virtual models. This would eliminate the need to create two separate system models for time domain simulations and modal analyses, and hence reduce the modeling and simulation work load by almost 50 %.

Abstract

When performing modal analyses of active flexible multibody systems, both controller effects and flexible body dynamics should be included in a multidisciplinary system model. Control system software, such as MATLAB and Simulink, usually supports both controller design and control system simulation, in which the mechanical system can be modeled with rigid bodies, lumped masses, inertias, springs, dampers or analytical equations. This will cause the flexible body dynamics to be predicted by very simplified models. In active flexible multibody dynamics software systems, such as FEDEM, feedback type controllers will typically calculate loads applied to the mechanical model based on feedback measurements of the system. This approach works well in a time domain analysis when the controller drives the mechanical model with applied loads based on the given controller algorithms. However, a major problem occurs in modal analyses of the closed-loop system. In a free vibration analysis, all loads are set to zero, which decouples the controller and mechanical model. As a result, the mechanical model becomes singular in all controlled degrees of freedom. A common approach by mechanical engineers when performing modal analyses of active flexible multibody systems is to introduce additional boundary conditions for the system degrees of freedom affected by controllers. This causes the flexibility in the different joints of the mechanism to be omitted since the joints are made rigid at relevant positions. Another common, though inaccurate, solution to this problem is to represent the controller effects by virtual springs, dampers and inertias in the mechanical model. Nonetheless, this approach is only applicable for simple control systems in which the mechanical engineer knows how to transform the controller into an equivalent mechanical model. Additionally, when designing and optimizing active flexible multibody systems, the engineer also has to update two system models simultaneously: one for the modal analysis and one for the time domain dynamic simulation.

This thesis presents a method for performing modal analyses of active flexible multibody systems in a finite element environment based on the generalized eigenvalue problem. The mechanical equivalent properties of position, velocity or acceleration feedback proportional-integral-derivative (PID) controllers are derived, and it is shown how these properties can be included into a system model appropriate for modal analysis. Controllers containing non-collocated sensors and actuators are also covered. Since the controller parameters may not be explicitly defined for the engineer working in a finite element environment, a method for deriving the controller gains for PID controllers using perturbations is also presented. Two versions of the modal analysis method are presented: one simplified n-dimensional undamped version and one complete $3n$ first-order version which include all system properties. The derived theory is verified throughout this work through examples. The presented method is of relevance to mechatronic products involving vibrational issues in disciplines such as robotics, aerospace and aviation, military, etc.

List of Appended Papers

This thesis is based on the following appended papers, which are referred to by roman numerals in the text.

I Bratland, M. and Rølvåg, T., "Modal Analysis of Lumped Flexible Active Systems (Part 1)," presented at the SIMS 2008: The 48th Scandinavian Conference on Simulation and Modeling, Oslo, Norway, 2008.

II Bratland, M., Haugen, B., and Rølvåg, T., "Modal analysis of active flexible multibody systems," *Computers and Structures*, vol. 89, pp. 750-761, 2011.

III Bratland, M., Haugen, B., and Rølvåg, T., "Modal Analysis of Active Flexible Multibody Systems Containing PID Controllers with Non-Collocated Sensors and Actuators," *Computers and Structures*, 2011, (submitted).

IV Bratland, M., Haugen, B., and Rølvåg, T., "A Method for Controller Parameter Estimation Based on Perturbations," *Multibody System Dynamics*, 2011, (submitted).

List of Abbreviations

DOF	Degree of Freedom
FE	Finite Element
FEA	Finite Element Analysis
FFT	Fast Fourier Transform
MDOF	Multiple Degrees of Freedom
MIMO	Multiple-Input Multiple-Output
PD	Proportional-Derivative
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
SDOF	Single Degree of Freedom
SISO	Single-Input Single-Output

Table of Contents

1. Introduction

This thesis presents a method for performing modal analyses of active flexible multibody systems in a finite element (FE) environment based on the generalized eigenvalue problem. In this chapter, current approaches for modal analysis of such systems are presented, and some of the shortcomings of these approaches are highlighted. The aim of this Ph.D. project is presented, along with some questions regarding this issue which may be familiar to mechanical engineers who have been working with active flexible multibody systems.

1.1. The Problem

The modal parameters, i.e. natural frequencies, mode shapes and damping ratios, of a system are of special interest to engineers working with the dynamics of structures and mechanisms. One of the classic textbook examples of failure to accurately predict such parameters is that of the Tacoma Narrows bridge, a suspension bridge which was opened to traffic on 1 July 1940 and collapsed on 7 November of that same year due to wind induced vibrations [2, 3].

The dynamic performance of active flexible multibody systems is strongly dependent on an optimal interaction between the mechanical components and the controllers. To illustrate some of the challenges for accurately predicting the modal parameters of active flexible multibody systems, a satellite tracking antenna has been chosen as an example. A virtual model of the topical satellite tracking antenna is depicted in Figure 1.

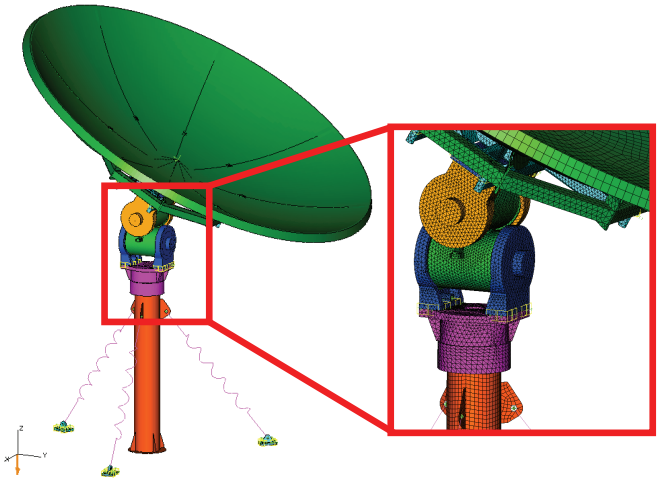


Figure 1: Satellite tracking antenna.

The intended usage of the antenna in Figure 1 is to track earth orbiting satellites from an earthbound position. The antenna mechanism has two main joints for manipulating the pointing direction of the antenna dish: rotations about the global *x*- and *y*-axis (elevations). Each of these are handled by a motor and controlled by an angular position feedback PID controller. Due to the

intended use of the antenna, its pointing accuracy is a subject of great importance. If the antenna starts to vibrate during operational use, its pointing accuracy may be greatly reduced, and its usefulness diminished. Identifying the modal parameters of the mechanism thus becomes a key engineering task. This task, often referred to as *modal analysis*, usually follow either one of two routes: solving the eigenvalue problem or analyzing the time response of the system when subjected to disturbances. The former method may be referred to as *analytical* or *theoretical modal analysis*, while the latter may be referred to as *experimental modal analysis*.

A great source of insight into experimental modal analysis is a paper written by Avitabile [4] and a book written by Ewins [5]. Basically, experimental modal analysis consists of analyzing the responses of a system when exited or otherwise brought into motion. Time domain responses can in themselves be used to derive natural or resonance frequencies and damping ratios, but one of the core aiding tools in experimental modal analysis is the Fourier transform, or more specifically the discrete Fourier transform, and its most widely used algorithm, the fast Fourier transform (FFT) [6]. One major issue with the experimental modal analysis approach is the challenge in deriving the mode shapes of the system, especially for a system containing a large number of degrees of freedom (DOFs).

Analytical or theoretical modal analysis, often referred to simply as modal analysis, has its origin in the eigenvalue problem and the free vibrations of a system. For an active flexible multibody system such as the satellite tracking antenna, the challenge is to include all relevant system properties in a multidisciplinary system model. One obstacle for this is that the basic formulation and solvers for control systems and mechanical systems are different. Control systems are often modeled as first-order equation systems (state-space formulation), e.g. [7-9], while mechanical systems are usually modeled as second-order symmetrical equation systems, e.g. [3, 10-12]. Solving the eigenvalue problem for the satellite tracking antenna may thus be approached from two different discipline strategies: control system engineering or mechanical engineering.

Control system software, such as MATLAB and Simulink[1], usually support both controller design and control system simulation. The mechanical systems can be modeled with rigid bodies, lumped masses, inertias, springs and dampers or analytical equations. Using this approach, the closed-loop eigenvalues and eigenvectors can be predicted as shown in e.g. [13-15]. From a mechanical engineers point of view, the flexible body dynamics are by this approach predicted by very simplified models. This may work well if the dynamics due to flexible bodies can be neglected. If not, control and observation spillover can cause a reduction in dynamic performance and may lead to system instability. Unmodeled flexible body dynamics also make modal analysis and controller synthesis unreliable.

Due to the complexity of the mechanical components, both in form and function, it may be practical to handle a system such as the satellite tracking antenna through an FE approach. Effective time domain dynamic simulations of flexible multibody systems in an FE environment have been described by for instance Sivertsen [12] and Géradin and Cardona [16]. FE software,

[1] MATLAB and Simulink by The MathWorks, Inc.

such as FEDEM[2], can also be interfaced with control system software for dynamic time simulation analyses of active systems [12]. Feedback type controllers will typically calculate loads applied to the FE structure based on feedback measurements of the system. For that reason, the controller is comparable to a "black box" or unknown function, as seen from the mechanical engineer's point of view. This approach works well in time domain analyses when the controller drives the FE model with applied loads based on the given controller algorithms. Even so, a major problem occurs in modal analyses of the closed-loop system. In free vibration analyses, all loads are set to zero, which decouples the controller and mechanical model. As a result, the antenna mechanism becomes singular in both elevation DOFs.

Most FE software systems today support modal analysis of flexible mechanical systems but do not include effects from controllers. In addition, mechanical engineers may experience limited knowledge about controllers, actuators and sensors, and are therefore susceptible to doing critical mistakes. Probably the most common approach by mechanical engineers when performing modal analyses of systems such as the satellite tracking antenna is to introduce additional boundary conditions for the system DOFs affected by controllers, thereby omitting the flexibility in the different joints of the mechanism by making the joints rigid at relevant positions. The greatest flaw in this approach is that the eigenfrequencies and eigenmodes for an active mechanism are not the same as for a purely mechanical system, as shown in for example [8, 10, 18]. Several other sources exist that support this statement, either directly or indirectly, e.g. [19-25].

Another common, though inaccurate, solution to this problem is to represent the controller effects by virtual springs, dampers and inertias in the mechanical model [18]. For example, Sharon *et al.* [19] have stated that: "If an ideal actuator and corresponding ideal sensor are acting on the same point (collocated control) in a purely inertial system, then: 1. Negative position feedback is equivalent to a spring action. 2. Negative velocity feedback is equivalent to a damping action. 3. Negative force feedback is equivalent to decreasing inertia. 4. Positive force feedback is equivalent to increasing inertia." Bernzen [21] has demonstrated that actuators can be controlled to act like virtual passive mechanical spring-damper elements using a velocity feedback PI controller. Ryu *et al*. [23] mentioned that a position feedback PD controller is physically equivalent to a virtual spring and damper whose reference position is moving at a desired velocity. Nonetheless, this approach is only applicable for simple control systems in which the mechanical engineer knows how to transform the controller into an equivalent mechanical model. The engineer also has to update two system models: one for modal analysis and one for time domain dynamic simulation.

1.2. Aim of Study

Due to the reasons listed in the previous section, several industrial companies, one of which being the Kongsberg Group, have expressed a desire for a method which can be used to accurately predict the modal parameters of active flexible multibody systems. The overall aim of

[2] FEDEM (Finite Element in Dynamics of Elastic Mechanisms) simulation software is a multibody dynamics package distributed by Fedem Technology AS. It is based on the finite element method and uses model reduction techniques to effectively perform nonlinear time domain dynamic simulations of active flexible multibody systems [12, 17].

this work has been to allow engineers working in an FE environment to be able to accurately predict modal parameters of such systems, and the work has been aimed at covering the controller types most commonly used in the industry. This work has therefore been limited to single or multiple degrees of freedom flexible multibody systems with position, velocity or acceleration feedback single or multiple input and output controllers of type proportional-integral-derivative (PID), the most common type of controllers in use today [26, 27]. The sensors and actuators to the controller can either be collocated or non-collocated, and the controller can contain both continuous and discontinuous elements. The presented method is intended to be implemented into an FE software system, preferably FEDEM.

The main findings which will be presented in this work is how various controllers affect the dynamic performance of the active system, and how the controller properties may be included in a system model appropriate for modal analysis. Two modal analysis versions will be proposed: one simplified version based on the second-order n-dimensional differential equations of the system, and one complete version which transforms the second-order n-dimensional differential equations into a first-order $3n$-dimensional form. In addition, a method for deriving the controller parameters of unknown PID-type controllers will be presented. The method is based on perturbations and is able to handle both continuous and discrete time independent and time dependent controller elements. With this knowledge at hand, FE-based eigenvalue solvers may be upgraded with relatively reasonable effort to not only include the flexibility of elastic systems but also the active elements from controllers. This allows for more accurate modal analyses of active and flexible system, in addition to being and efficiency improvement for engineer working with such systems by eliminating the need for two synchronized system models.

This thesis is divided into six main sections, including this introductory chapter, in addition to the four appended papers. In Chapter 2, the basic theory comprising this work is explained. How to solve eigenvalue problems and theory into common control engineering practice are covered. A major focal point in this work has been on how to express the properties of various controllers in time domain, thus making the controller compatible with standard mechanical second-order differential equations. This theory is explained in more details in the appended papers, but is for convenience also included in Chapter 2 of this thesis, although in a somewhat more summarizing form. The focus for this thesis has been more on the initial or preliminary work to be carried out prior to the actual modal analysis, i.e. pre-processing, and not so much on actually solving the problem (processing) or interpreting the results from the modal analysis (post-processing). Chapter 3 is a brief summary of the appended papers and lists some of the most important findings in these. In Chapter 4, a brief discussion regarding the proposed methods for solving the eigenvalue problem for active flexible multibody systems is presented, along with answers to questions raised in Section 1.3. Chapters 5 and 6 are a conclusion to this work and suggestions for future work. Paper I is intended as being an initial investigation into the field of modal analysis of active flexible multibody systems. The main objective of Paper II is to derive a method for performing modal analyses of active flexible multibody systems based on second-order differential equations. An n-dimensional eigenvalue problem version is presented for systems without damping and controllers not containing position integral gains. In Paper III, the theory derived in Paper II is built on and further expanded, focusing on active systems with non-collocated sensors and actuators and PID controllers containing position feedback integral gains.

A 3*n* state-space version for the eigenvalue problem is presented. A method for deriving the controller parameters of unknown PID-type controllers is presented in Paper IV.

1.3. Practical Engineering Questions

Engineering is all about solving practical problems. What kind of questions could arise from the issue presented in Section 1.1.? The main question is if there are any eigenfrequencies within the bandwidth of the controlled system that may cause vibrational issues. Below are listed some examples of questions which are frequently asked by mechanical engineers regarding this issue:

1. A proportional-derivative (PD) controller and a hydraulic actuator are driving a suspension system. In what way does the effective system mass, stiffness and damping, and hence the eigenfrequencies of the system, become affected? What if the controller use both position, velocity and force feedback? Can fixed boundary conditions be applied to the driving DOFs to remove the singularities occurring when the forces are set to zero? If not, how can the controller be represented by equivalent mechanical properties supported by the FE software?

2. One might know how to solve the problem raised by Question 1, however, the controller may also contain discrete elements like hysteresis, logical switches, dead zones, time delays and limit elements. How do these types of controller elements affect the closed-loop eigenfrequencies, and can they be represented by mechanical properties in the FE software?

3. It may be decided to use a PID controller to minimize position and velocity deviations on a machining centre. Does the integral part of the controller affect the stiffness or damping of the mechanical system, and does the derivate effect introduce any artificial inertia to the mechanical system?

4. It may be desired to optimize a rotating machinery, and the clutch and actuator have a limited and nonlinear torque capacity. How can one model the boundary conditions when they are dependent on the reaction torque? When the applied torque exceeds the clutch and actuator capacity the machinery is suddenly free to rotate!

The topic regarding these questions will be treated in this work, and each of these questions will be answered in the Section 4.2.

2. Theory

2.1. Mechanical Systems

The core basis for dynamic mechanical systems is the second-order ordinary differential equations of motion, which for a system containing multiple degrees of freedom (MDOF) may be written as:

$$\mathbf{M}\ddot{\mathbf{r}}(t) + \mathbf{C}\dot{\mathbf{r}}(t) + \mathbf{K}\mathbf{r}(t) = \mathbf{F}(t) \tag{1}$$

where **M**, **C** and **K** are the $n \times n$ system mass, damping and stiffness matrices, respectively, and where n is the number of DOFs. $\mathbf{F}(t)$ is the $n \times 1$ vector of driving or input forces acting on the various masses, and $\mathbf{r}(t)$ is the $n \times 1$ vector of positions for the masses with respect to time. $\dot{\mathbf{r}}(t)$ and $\ddot{\mathbf{r}}(t)$ are the first and second time derivatives of $\mathbf{r}(t)$. Typically, **M**, **C** and **K** will be symmetrical matrices.

In general, the modal parameters of a system described by Equation (1) may be derived by solving the generalized eigenvalue problem:

$$\mathbf{A\Phi} = \mathbf{B\Phi\Lambda} \tag{2}$$

where **A** and **B** are both an $n \times n$ matrix, $\mathbf{\Lambda}$ is an $n \times n$ diagonal matrix of the generalized eigenvalues and $\mathbf{\Phi}$ is an $n \times n$ full matrix whose columns are the corresponding eigenvectors. The eigenvalues correspond to the roots of the characteristic polynomial equation. If the mechanical MDOF system is undamped, the generalized eigenvalue problem can be given as:

$$\mathbf{K\Phi} = \mathbf{M\Phi\Lambda} \tag{3}$$

Both the eigenvalues and eigenvectors of Equation (3) will typically be real.

2.2. State-Space Methods

For a damped MDOF mechanical system, Equation (3) cannot be used to accurately derive the modal parameters of the system, since the damping matrix **C** introduces coupling between the derivatives in Equation (1). The exception being proportional or Rayleigh damping [10], which will not be covered here. In general, the eigenvalues and eigenvectors of the damped MDOF system will be complex [28]. The imaginary part of the complex eigenvalues is the frequency of oscillation, and the real part is the constant in the exponent of the oscillation amplitude envelope. For the complex eigenvectors, the amplitude at each DOF can be viewed as having both a magnitude and a phase angle [5]. In the introduction of [28], Adhikari gives a brief and good description of the issue of non-proportionally damped systems. He states that the solution procedures of such systems will primarily follow two routes: the state-space method and approximate methods in n-space.

The basic principle behind the state-space method is to express the second-order differential equations in a first-order (state-space) form. There exist several ways of doing this, but two main categories will be covered here. The first is a method often encountered in control engineering practice. The equation for free vibrations of Equation (1) may be rewritten as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \tag{4}$$

where

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \end{bmatrix} \quad , \quad \dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{r}} \\ \ddot{\mathbf{r}} \end{bmatrix} \quad , \quad \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \tag{5}$$

The eigenvalues and eigenvectors of the system can now be calculated by solving the standard eigenvalue problem for the system matrix **A**:

$$\mathbf{A}\boldsymbol{\Phi} = \boldsymbol{\Phi}\boldsymbol{\Lambda} \tag{6}$$

The solution of Equation (6) will yield 2*n* complex eigenvalues and eigenvectors. For systems which are less-than-critically damped, the complex eigenvalues occur as complex conjugate pairs with corresponding complex conjugate modal columns. The eigenvectors will occur in accordance with the state vector **x**, i.e.:

$$\boldsymbol{\varphi}_i = \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{bmatrix} \tag{7}$$

One drawback of this method is the matrix inversion which has to take place on matrix **M**. Matrix inversion is both computationally expensive and a source for errors, and are usually avoided whenever possible.

Foss [29] has described another method for expressing the second-order differential equations on a state-space form. Keeping in line with the format used in this work, the method is here reproduced on the form most closely resembling that which is presented by Ewins [5]. The equation for free vibrations of Equation (1) may be rewritten as:

$$\mathbf{B}\dot{\mathbf{x}} + \mathbf{A}\mathbf{x} = \mathbf{0} \tag{8}$$

where

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \end{bmatrix} \quad , \quad \dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{r}} \\ \ddot{\mathbf{r}} \end{bmatrix} \quad , \quad \mathbf{A} = \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & -\mathbf{M} \end{bmatrix} \quad , \quad \mathbf{B} = \begin{bmatrix} \mathbf{C} & \mathbf{M} \\ \mathbf{M} & \mathbf{0} \end{bmatrix} \tag{9}$$

The eigenvalues and eigenvectors of the system can now be calculated by solving the generalized eigenvalue problem for the system matrix **A** and **B** using Equation (2). As for the system described by Equation (4), the solution of the generalized eigenvalue problem for Equation (8)

will yield $2n$ complex eigenvalues and eigenvectors, which will occur in complex conjugate pairs. The variant of Foss' method used by FEDEM is:

$$\mathbf{A}\mathbf{x} - \mathbf{B}\dot{\mathbf{x}} = \mathbf{0} \tag{10}$$

where:

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \end{bmatrix} \quad , \quad \dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{r}} \\ \ddot{\mathbf{r}} \end{bmatrix} \quad , \quad \mathbf{A} = \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{-M} \end{bmatrix} \quad , \quad \mathbf{B} = \begin{bmatrix} \mathbf{-C} & \mathbf{-M} \\ \mathbf{-M} & \mathbf{0} \end{bmatrix} \tag{11}$$

The methodology derived through this Ph.D. project and presented in this thesis is mainly based on this FEDEM formulation of the state-space method.

As stated by Adhikari [28], one concern with the state-space method is that although the state-space method is exact in nature, it requires significant numerical effort for obtaining the eigensolutions as the size of the problem doubles. How to efficiently solve these systems is treated in Section 2.5.

2.3. Control Systems

If a mechanical system is affected by controllers of some kind, the system is said to be *active*. Its counterpart, a *passive* system, is a mechanical system without any form of active controllers. In this work, the term *control system* is used to refer to the complete active system (physical process, sensor, actuator and controller), whereas the word *controller* is used to refer to the part of the control system containing the control algorithms. Only feedback controllers will be dealt with in this work, so all control system terminology used here refers implicitly to feedback controllers.

If the control system has only one sensor measuring a system variable and only one actuator executing its commands, the system is said to be single-input single-output (SISO). A control system containing several sensors and actuators is said to be multiple-input multiple-output (MIMO). A control system is usually described by using a block diagram. An example of such a diagram is given in Figure 2. The figure shows a block diagram used for describing a SISO feedback control system.

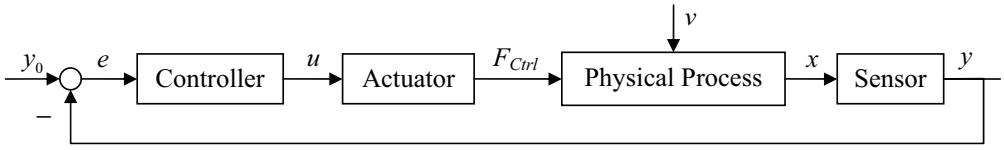


Figure 2: Block diagram of a SISO feedback control system.

In Figure 2, y_0 is the reference variable, y is the measured variable and e is the difference between y_0 and y. u is the controller output and F_{Ctrl} is a force from the controller exerted by an

actuator. x is the state variable from the physical process, either position r, velocity $\dot{r}$ or acceleration $\ddot{r}$, while v is the disturbance on the physical process.

It is a customary control engineering practice to use Laplace transformation on both the mechanical system and the controllers, and then treat the system in the s-plane and on state-space form [9]. A brief summary of customary control engineering practice is given below. A general state-space representation of a linear and time-invariant system (without disturbances) with p inputs, q outputs and n DOFs is [30]:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{aligned} \tag{12}$$

where $\mathbf{x}(t)$ is the $2n \times 1$ state vector, $\mathbf{A}$ is the $2n \times 2n$ state matrix, $\mathbf{B}$ is the $2n \times p$ input matrix and $\mathbf{u}$ is the $p \times 1$ input vector. $\mathbf{B}$ is hence an array which maps the physical locations of the input forces to the internal variables of realization. $\mathbf{y}(t)$ is the $q \times 1$ output vector, $\mathbf{C}$ is the $q \times 2n$ output matrix and $\mathbf{D}$ is the $q \times p$ matrix corresponding to direct input/output feedforward or feedthrough. $\mathbf{y}(t)$ is physical sensor measurements such as displacement, velocity or acceleration and $\mathbf{C}$ is an array which constructs these physical quantities from the internal variables $\mathbf{x}(t)$. The input control force can be written as:

$$\mathbf{u}(t) = \mathbf{K}\mathbf{y}(t) \tag{13}$$

where $\mathbf{K}$ is the $p \times q$ feedback gain matrix. Inserting Equation (13) into Equation (12) and performing some intermediate calculations yields:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \left(\mathbf{A} + \mathbf{B}\mathbf{K}\left(\mathbf{I} - \mathbf{D}\mathbf{K}\right)^{-1}\mathbf{C} \right)\mathbf{x}(t) \\ \mathbf{y}(t) &= \left(\mathbf{I} - \mathbf{D}\mathbf{K}\right)^{-1}\mathbf{C}\mathbf{x}(t) \end{aligned} \tag{14}$$

For the free vibration of the system, $\mathbf{D}$ can be set to zero, which gives:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \left(\mathbf{A} + \mathbf{B}\mathbf{K}\mathbf{C}\right)\mathbf{x}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \end{aligned} \tag{15}$$

The closed-loop eigenvalue problem can hence be formulated as:

$$\left(\mathbf{A} + \mathbf{B}\mathbf{K}\mathbf{C}\right)\boldsymbol{\Phi} = \boldsymbol{\Phi}\boldsymbol{\Lambda} \tag{16}$$

Even though Equation (16) can solve the eigenvalue problem for active systems, all of the matrices in Equation (16) are given in state-space form, making them difficult to combine with FE-based software systems. In this work, one of predominant motivations has been to keep both systems in time domain and in line with the second-order differential equations in order to make

the derived theory more accessible to mechanical engineers and more compatible with FE-based simulation software for active flexible multibody systems.

As stated in the introductory chapter, the most common type of controllers in use today is the PID controller. For a feedback PID controller, the controller output u is given by:

$$u_{PID}(t) = K_p e(t) + K_i \int e(t)\,dt + K_d \frac{d}{dt} e(t) \tag{17}$$

where K_p is the proportional gain, K_d is the derivative gain and K_i is the integral gain from the controller. Equation (17) is sometimes referred to as the ideal form of the PID controller. The form of the PID controller most often encountered in the industry is the standard form:

$$u_{PID}(t) = K_p \left(e(t) + \frac{1}{T_i} \int e(t)\,dt + T_d \frac{d}{dt} e(t) \right) \tag{18}$$

where T_i is the integral time and T_d is the derivative time. T_i and T_d are related to the parameters of the ideal form in Equation (17) through:

$$K_i = \frac{K_p}{T_i} \Leftrightarrow T_i = \frac{K_p}{K_i} \quad , \quad K_d = K_p T_d \Leftrightarrow T_d = \frac{K_d}{K_p} \tag{19}$$

In this work, the ideal form of the PID controller given by Equation (17) will be used.

Since F_{Ctrl} is a force from the controller which acts on the physical process, the equation of motion for a single degree of freedom (SDOF) mechanical system combined with a feedback PID controller can be written as:

$$m\ddot{r}(t) + c\dot{r}(t) + kr(t) = F(t) + F_{Ctrl}(t) \tag{20}$$

This way of representing the controller is in accordance with equations found in [15] and Papers I and II. If for simplicity the actuator is said to have an exchange ratio of 1:1, Equation (20) can be written as:

$$m\ddot{r}(t) + c\dot{r}(t) + kr(t) = F(t) + u_{PID}(t) \tag{21}$$

Using the fact that e is the difference between y_0 and y, Equation (17) can be split into a feedforward or feedthrough part:

$$u_{PID_{Feedforward}}(t) = K_p\, y_0(t) + K_i \int y_0(t)\,dt + K_d \frac{d}{dt} y_0(t) \tag{22}$$

and a feedback part:

$$u_{PID_{Feedback}}(t) = K_p\, y(t) + K_i \int y(t)\, dt + K_d \frac{d}{dt} y(t) \tag{23}$$

If for simplicity the sensor is said to have an exchange ratio of 1:1, and depending on what state variable the sensor measures, i.e. position $(x(t) = r(t))$, velocity $(x(t) = \dot{r}(t))$ or acceleration $(x(t) = \ddot{r}(t))$, Equation (21) can be written as:

$$m\ddot{r}(t) + c\dot{r}(t) + kr(t) + u_{PID_{Feedback}}(x,t) = F(t) + u_{PID_{Feedforward}}(x_0,t) \tag{24}$$

In Equation (24), the part of the controller output u associated with the feedforward part can be viewed as an external driving force since x_0 is a predefined value or function of an external reference, and not a function of the system's variables. Due to the fact that x is a function of one of the system variables, the part of the controller output u associated with the feedback part can be viewed as an internal force similar to inertia, damping and spring forces. The free vibration of the active system is thus given by the following equation:

$$m\ddot{r}(t) + c\dot{r}(t) + kr(t) + K_p\, x(t) + K_i \int x(t)\, dt + K_d \frac{d}{dt} x(t) = 0 \tag{25}$$

where $x(t)$ is either a function of position $r(t)$, velocity $\dot{r}(t)$ or acceleration $\ddot{r}(t)$.

From Equation (25) it can be seen that if $x(t)$ is a function of position $r(t)$, K_p will affect the stiffness of the system, whereas K_d and K_i will affect the damping of the system. Similarly, if $x(t)$ is a function of velocity $\dot{r}(t)$, K_p will affect the damping of the system, K_d the inertia and K_i the stiffness of the system, and if $x(t)$ is a function of acceleration $\ddot{r}(t)$, K_p will affect the inertia and K_i the damping of the system. In contrast, K_d is proportional to the derivative of the acceleration with respect to time, $\dddot{r}$, often referred to as jerk or jolt. This pattern is summarized in the following figure which can be found in Paper II:

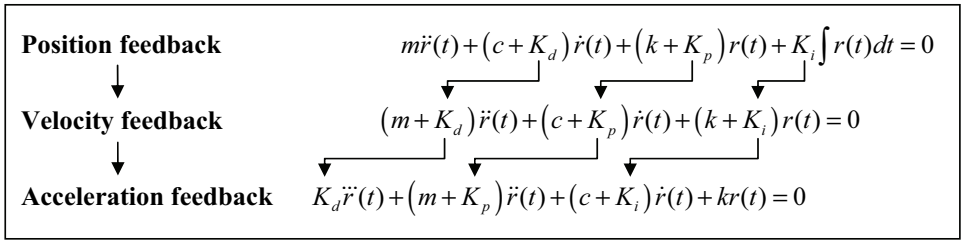


Figure 3: Pattern for the addition of controller gains into the system equation based on type of sensor input.

The position of the proportional gain K_p in Figure 3 is also in agreement with the statement made by Sharon *et al.* [19]: "If an ideal actuator and corresponding ideal sensor are acting on the same point (collocated control) in a purely inertial system, then: 1. Negative position feedback is equivalent to a spring action. 2. Negative velocity feedback is equivalent to a damping action. 3. Negative force feedback is equivalent to decreasing inertia. 4. Positive force feedback is equivalent to increasing inertia."

Similarly as for Equations (24) and (25), for an active MDOF system containing MIMO PID controllers, the equation of motion can be written as:

$$\mathbf{M}\ddot{\mathbf{r}}(t)+\mathbf{C}\dot{\mathbf{r}}(t)+\mathbf{K}\mathbf{r}(t)+\mathbf{u}_{PID_{Feedback}}(\mathbf{x},t)=\mathbf{F}(t)+\mathbf{u}_{PID_{Feedforward}}(\mathbf{x}_0,t) \tag{26}$$

and the equation of motion for the free vibration as:

$$\mathbf{M}\ddot{\mathbf{r}}(t)+\mathbf{C}\dot{\mathbf{r}}(t)+\mathbf{K}\mathbf{r}(t)+\mathbf{K}_p\,\mathbf{x}(t)+\mathbf{K}_i\int\mathbf{x}(t)\,dt+\mathbf{K}_d\frac{d}{dt}\mathbf{x}(t)=\mathbf{0} \tag{27}$$

where $\mathbf{x}(t)$ is a vector of various state variables in accordance with what types of state variables the various sensors are measuring.

In Paper II, the gradients for a MDOF MIMO system is written as:

$$dF_{Ctrl_i}=\frac{\partial F_{Ctrl_i}}{\partial u_j}\frac{\partial u_j}{\partial y_k}\frac{\partial y_k}{\partial x_l}dx_l=G_{Act_{ij}}G_{Ctrl_{jk}}G_{Sens_{kl}}dx_l \tag{28}$$

or, on matrix form as:

$$d\mathbf{F}_{Ctrl}=\frac{\partial\mathbf{F}_{Ctrl}}{\partial\mathbf{u}}\frac{\partial\mathbf{u}}{\partial\mathbf{y}}\frac{\partial\mathbf{y}}{\partial\mathbf{x}}d\mathbf{x}=\mathbf{G}_{Act}\mathbf{G}_{Ctrl}\mathbf{G}_{Sens}\,d\mathbf{x} \tag{29}$$

Where $\mathbf{G}_{Act}$, $\mathbf{G}_{Ctrl}$ and $\mathbf{G}_{Sens}$ are the actuator gradient, controller gradient and sensor gradient matrices, respectively.

The actuator gradients describe the relationship between the controller forces $\mathbf{F}_{Ctrl}$ exerted by the actuator and the output signals $\mathbf{u}$ from the controller. It should here be clarified that in control engineering, inputs to the physical process (i.e outputs from the controller) is entitled inputs (u), whereas outputs from the physical process (i.e. inputs to the controller) is called outputs (y). In this work however, the inputs to the controller (y) has been labeled inputs, while the outputs from the controller (u) has been labeled outputs. The gradients of the controller force F_{Ctrl_i} with respect to controller output u_j can be written as:

$$d\mathbf{F}_{Ctrl}=\mathbf{G}_{Act}d\mathbf{u}\quad\text{or}\quad dF_{Ctrl_i}=\frac{\partial F_{Ctrl_i}}{\partial u_j}du_j=G_{Act_{ij}}du_j \tag{30}$$

Matrix $\mathbf{G}_{Act}$ has the dimensions $n_{F_{Ctrl}}\times n_u$ where $n_{F_{Ctrl}}$ is the number of controller forces and n_u is the number of controller outputs.

The controller gradients describe the relationship between the input variables $\mathbf{y}$ and output variables $\mathbf{u}$ both to and from the controller, respectively; that is, the various controller gains. The gradients of the controller output u_i with respect to the controller input y_j can be written as:

$$d\mathbf{u} = \mathbf{G}_{Ctrl} d\mathbf{y} \quad \text{or} \quad du_i = \frac{\partial u_i}{\partial y_j} dy_j = G_{Ctrl_{ij}} dy_j \tag{31}$$

Matrix $\mathbf{G}_{Ctrl}$ has the dimensions $n_u \times n_y$ where n_u is the number of controller outputs and n_y is the number of controller inputs.

The sensor gradients describe the relationship between the controller input variables **y** and the system state variables $\mathbf{r}$, $\dot{\mathbf{r}}$ and $\ddot{\mathbf{r}}$ represented by the vector **x**. **x** is given as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \\ \ddot{\mathbf{r}} \end{bmatrix} \tag{32}$$

Vector **x** has the dimensions $3n_r \times 1$ where n_r is the number of all system DOFs. Each sensor is limited to measure only one state variable in only one single system DOF or between two system DOFs. The gradients of the controller input y_i with respect to system DOF and state variable x_j can be written as:

$$d\mathbf{y} = \mathbf{G}_{Sens} d\mathbf{x} \quad \text{or} \quad dy_i = \frac{\partial y_i}{\partial x_j} dx_j = G_{Sens_{ij}} dx_j \tag{33}$$

Matrix $\mathbf{G}_{Sens}$ has the dimensions $n_y \times 3n_r$ where n_y is the number of controller inputs and n_r is the number of all system DOFs.

The matrix product **G** of the gradient matrices $\mathbf{G}_{Act}$, $\mathbf{G}_{Ctrl}$ and $\mathbf{G}_{Sens}$ has the dimensions $n_{F_{Ctrl}} \times 3n_r$. If **G** is to be used with **M**, **C** and **K**, it should be of the same dimensions, i.e. $n_r \times n_r$. This can be done by pre-multiplying **G** with the topology matrix relating each controller force F_{Ctrl_i} with its respective system DOFs and then splitting the new $n_r \times 3n_r$ matrix product into 3 $n_r \times n_r$ matrices, $\mathbf{G}_{Pos}$, $\mathbf{G}_{Vel}$ and $\mathbf{G}_{Acc}$, one for each state variable $\mathbf{r}$, $\dot{\mathbf{r}}$ and $\ddot{\mathbf{r}}$. These new matrices $\mathbf{G}_{Pos}$, $\mathbf{G}_{Vel}$ and $\mathbf{G}_{Acc}$ can then be added to their respective system matrix, yielding the following equation system for the free vibration of a controlled mechanism:

$$\left(\mathbf{M} + \mathbf{G}_{Acc}\right)\ddot{\mathbf{r}}(t) + \left(\mathbf{C} + \mathbf{G}_{Vel}\right)\dot{\mathbf{r}}(t) + \left(\mathbf{K} + \mathbf{G}_{Pos}\right)\mathbf{r}(t) = \mathbf{0} \tag{34}$$

or, more conveniently:

$$\mathbf{M}_{eff}\ddot{\mathbf{r}} + \mathbf{C}_{eff}\dot{\mathbf{r}} + \mathbf{K}_{eff}\mathbf{r} = \mathbf{0} \tag{35}$$

where $\mathbf{M}_{eff}$ is the effective mass matrix of the system, while $\mathbf{C}_{eff}$ is the effective damping matrix and $\mathbf{K}_{eff}$ the effective stiffness matrix of the system. The dimensions of all the matrices are $n \times n$ and the vectors $n \times 1$.

As stated in Paper III, Equation (34) is valid for all PID controllers only containing controller elements proportional to position, velocity or acceleration. Examples of controllers not covered by Equation (34) are: position feedback controllers containing integral gains or acceleration feedback controllers containing derivative gains, though the latter variant will not be covered in this work. Based on Equations (27) and (34), the equation of motion for the free vibration of an active MDOF system containing a position feedback PID controller can be written as:

$$\left(\mathbf{M}+\mathbf{G}_{Acc}\right)\ddot{\mathbf{r}}(t)+\left(\mathbf{C}+\mathbf{G}_{Vel}\right)\dot{\mathbf{r}}(t)+\left(\mathbf{K}+\mathbf{G}_{Pos}\right)\mathbf{r}(t)+\mathbf{G}_{SSEE}\int\mathbf{r}(t)dt=\mathbf{0} \tag{36}$$

where $\mathbf{G}_{SSEE}$ is the controller gradient steady-state error elimination matrix and $\int\mathbf{r}\,dt$ is the position time integral vector of the system. Similarly as for Equation (34), Equation (36) may be more conveniently written as:

$$\mathbf{M}_{eff}\ddot{\mathbf{r}}+\mathbf{C}_{eff}\dot{\mathbf{r}}+\mathbf{K}_{eff}\mathbf{r}+\mathbf{Q}_{eff}\int\mathbf{r}\,dt=\mathbf{0} \tag{37}$$

where $\mathbf{Q}_{eff}$ is the effective steady-state error elimination matrix of the system. As in Equations (34) and (35), the dimensions of all the matrices in both Equations (36) and (37) are $n\times n$ and the vectors $n\times 1$. For a system with only collocated sensors and actuators, the controller gradient matrices $\mathbf{G}_{Acc}$, $\mathbf{G}_{Vel}$, $\mathbf{G}_{Pos}$ and $\mathbf{G}_{SSEE}$ will all be diagonal matrices. For a system with one or more non-collocated sensors and actuators, the $\mathbf{G}_{Acc}$, $\mathbf{G}_{Vel}$, $\mathbf{G}_{Pos}$ and $\mathbf{G}_{SSEE}$ matrices will be unsymmetrical. Systems containing non-collocated sensors and actuators are more thoroughly treated in Paper III.

As shown in Paper II, in order to derive the modal parameters of an undamped system described by Equation (34), Equation (3) may be written as:

$$\mathbf{K}_{eff}\boldsymbol{\Phi}=\mathbf{M}_{eff}\boldsymbol{\Phi}\boldsymbol{\Lambda} \tag{38}$$

For a system containing damping, the generalized eigenvalue problem given by Equation (2) can be solved inserted for $\mathbf{A}$ and $\mathbf{B}$ in accordance with the system outlined by Equation (11). The basis for the $2n$ state-space generalized eigenvalue problem is then:

$$\mathbf{x}=\begin{bmatrix}\mathbf{r}\\ \dot{\mathbf{r}}\end{bmatrix}\quad,\quad\dot{\mathbf{x}}=\begin{bmatrix}\dot{\mathbf{r}}\\ \ddot{\mathbf{r}}\end{bmatrix}\quad,\quad\mathbf{A}=\begin{bmatrix}\mathbf{K}_{eff} & \mathbf{0}\\ \mathbf{0} & -\mathbf{M}_{eff}\end{bmatrix}\quad,\quad\mathbf{B}=\begin{bmatrix}-\mathbf{C}_{eff} & -\mathbf{M}_{eff}\\ -\mathbf{M}_{eff} & \mathbf{0}\end{bmatrix} \tag{39}$$

which inserted in Equation (2) yields:

$$\begin{bmatrix}\mathbf{K}_{eff} & \mathbf{0}\\ \mathbf{0} & -\mathbf{M}_{eff}\end{bmatrix}\boldsymbol{\Phi}=\begin{bmatrix}-\mathbf{C}_{eff} & -\mathbf{M}_{eff}\\ -\mathbf{M}_{eff} & \mathbf{0}\end{bmatrix}\boldsymbol{\Phi}\boldsymbol{\Lambda} \tag{40}$$

However, as described in Paper III, if one or more of the controllers contain position feedback integral gains, Equation (37) has to be used. It is then not sufficient to only use a $2n$ state-space formulation for the generalized eigenvalue problem, since one of the state variables will not be

included. A proposed remedy to this problem is to expand the $2n$ state-space formulation into a $3n$ state-space formulation for the generalized eigenvalue problem, following a similar pattern as Equations (11) and (40). The state vectors $\mathbf{x}$ and $\dot{\mathbf{x}}$ and the matrices $\mathbf{A}$ and $\mathbf{B}$ can be given as:

$$\mathbf{x} = \begin{bmatrix} \int \mathbf{r}\, dt \\ \mathbf{r} \\ \dot{\mathbf{r}} \end{bmatrix}, \quad \dot{\mathbf{x}} = \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \\ \ddot{\mathbf{r}} \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{Q}_{eff} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{eff} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{eff} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -\mathbf{K}_{eff} & -\mathbf{C}_{eff} & -\mathbf{M}_{eff} \\ \mathbf{K}_{eff} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{eff} & \mathbf{0} \end{bmatrix} \tag{41}$$

where the dimensions of $\mathbf{x}$ and $\dot{\mathbf{x}}$ are $3n \times 1$, and those of $\mathbf{A}$ and $\mathbf{B}$ are both $3n \times 3n$. The generalized eigenvalue problem may then be solved as:

$$\begin{bmatrix} \mathbf{Q}_{eff} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{eff} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{eff} \end{bmatrix} \boldsymbol{\Phi} = \begin{bmatrix} -\mathbf{K}_{eff} & -\mathbf{C}_{eff} & -\mathbf{M}_{eff} \\ \mathbf{K}_{eff} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{eff} & \mathbf{0} \end{bmatrix} \boldsymbol{\Phi}\boldsymbol{\Lambda} \tag{42}$$

2.4. Identifying Unknown Controller Parameters

One remark about the method presented in the previous section is that the controller properties have to be known explicitly prior to the modal analysis. To the best of the author's knowledge, no commercial software system for simulation of active mechanisms fully integrates flexible multibody dynamics and control system simulation, since the underlying equations for control systems and mechanical systems are expressed in different forms. In flexible multibody dynamics software systems, such as FEDEM, feedback type controllers will typically calculate loads applied to the mechanism based on feedback measurements of the system. Additionally, some flexible multibody dynamics software systems also have the option of importing or communicating with the controller model as an external process, for instance through a dynamic link library (dll) or Simulink. For these reasons, the controller is comparable to a "black box" or unknown function, as seen from the mechanical part of the software system. In order to overcome this issue, a method for identifying the controller parameters may be applied. One such method is presented in Paper IV, and is a method for estimating controller parameters for systems containing either higher-order integral gains, higher-order derivative gains or a combination of proportional, integral and derivative gains, i.e. PID controller. Some of the theory in Paper IV is reproduced here with focus on controller parameter estimations for PID controllers.

A potential method for parameter estimation, as described more completely in Paper IV, is to introduce perturbations into the system. This approach is not to be confused with the perturbation method described in [3], which can be used to solve nonlinear differential equations in which the solution is in the form of a power series. Perturbations in this context are incremental changes in a system variable. The basis of this technique can be found in, for instance, the principle of virtual work [3, 31, 32], the displacement method/direct stiffness method [33], system identification/parameter estimation [34, 35] and optimization theory [36]. For all the various

fields listed above, the concept remains the same: apply changes in one variable, measure reactions from other variables and then process the results in order to derive the desired system parameters.

By applying incremental changes, perturbations, to the input (y) of the controller, small changes in the output (u) from the controller can be registered. These changes will be in accordance with the internal control routine of the controller. The parameters of the controller can thus be estimated based on predetermined changes in the controller input and registered changes from the controller output.

A perturbation, as described in the previous paragraph, is illustrated in Figure 4:

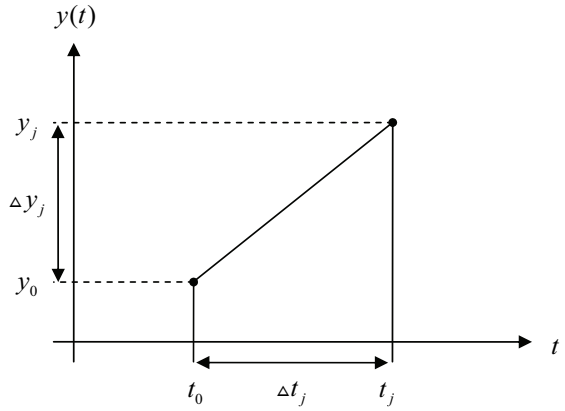


Figure 4: Perturbation j of t and y.

In Figure 4, the variables time t and controller input y are perturbed by the values Δy_j and Δt_j during perturbation j. y_0 and t_0 are the initial values for y and t, respectively, at the present time step. From Figure 4, the following relationships can be derived:

$$\Delta y_j = y_j - y_0 \Rightarrow y_j = y_0 + \Delta y_j \tag{43}$$

$$\Delta t_j = t_j - t_0 \Rightarrow t_j = t_0 + \Delta t_j \tag{44}$$

Since the controller output u is a function of y and t, the following equation can be given:

$$\begin{aligned} \Delta u_j &= u_j - u_0 \\ &= u_j(y_j, t_j) - u(y_0, t_0) \\ &= u_j(y_0 + \Delta y_j, t_0 + \Delta t_j) - u(y_0, t_0) \end{aligned} \tag{45}$$

The values Δy_j and Δt_j can be chosen arbitrarily, but it can be practical to express Δy_j as a function of Δt_j. The linear equation for $y_j(t)$ for perturbation j can then be written as:

$$y_j(t) = y_0 + \frac{\triangle y_j}{\triangle t_j} t \tag{46}$$

2.4.1. *Proportional gain*

For a feedback type controller containing only a controller output u proportional to the input variable y, its feedback gain equation can be written as:

$$u_P(t) = K_p\, y(t) \tag{47}$$

where K_p is the proportional gain. Equation (47) can be written on a general differential form as:

$$du = \frac{\partial u}{\partial y} dy \quad \text{or} \quad du = K_p\, dy \tag{48}$$

or, in discrete differential form as:

$$\triangle u = \frac{\partial u}{\partial y} \triangle y \quad \text{or} \quad \triangle u = K_p \triangle y \tag{49}$$

K_p can thus be calculated by solving the following equation:

$$K_p = \left(\triangle y\right)^{-1} \triangle u \tag{50}$$

2.4.2. *Integral gain*

For a feedback type controller containing only a controller output u proportional to the time integral of the input variable y, its feedback gain equation can be written as:

$$u_I(t) = K_i \int y(t)\, dt \tag{51}$$

where K_i is the integral gain. Equation (51) can be written in discrete differential form as:

$$\triangle u = \frac{\partial u}{\partial \int y\, dt} \triangle \int y\, dt \quad \text{or} \quad \triangle u = K_i \triangle \int y\, dt \tag{52}$$

and K_i can be calculated by solving the equation:

$$K_i = \left(\triangle \int y_j\, dt\right)^{-1} \triangle u_j \tag{53}$$

In order to estimate K_i using the perturbation technique described in the previous sections, $\triangle\int y_j\,dt$ needs to be discretized. In Figure 4, $\triangle\int y_j\,dt$ is the area under the linear curve. If $y_j(t)$ is given as in Equation (46), $\triangle\int y_j\,dt$ can be made as a function of $\triangle y_j$ and $\triangle t_j$ by:

$$\triangle\int y_j\,dt = \int_0^{\triangle t_j} y_j\,dt = \left[y_0 t + \frac{1}{2}\frac{\triangle y_j}{\triangle t_j} t^2 \right]_0^{\triangle t_j} = \left(y_0 + \frac{1}{2}\triangle y_j \right)\triangle t_j \tag{54}$$

Inserting Equation (54) into Equation (53) yields:

$$K_i = \left(\left(y_0 + \frac{1}{2}\triangle y_j \right)\triangle t_j \right)^{-1} \triangle u_j \tag{55}$$

2.4.3. *Derivative gain*

For a feedback type controller containing only a controller output u proportional to the time derivative of the input variable y, its feedback gain equation can be written as:

$$u_D(t) = K_d \frac{d}{dt} y(t) = K_d\, \dot{y}(t) \tag{56}$$

where K_d is the derivative gain. Equation (56) can be written in discrete differential form as:

$$\triangle u = \frac{\partial u}{\partial \dot{y}} \triangle \dot{y} \quad \text{or} \quad \triangle u = K_d \triangle \dot{y} \tag{57}$$

K_d can be calculated by solving the equation:

$$K_d = \left(\triangle \dot{y}_j \right)^{-1} \triangle u_j \tag{58}$$

In order to estimate K_d using the perturbation technique described in the previous sections, $\triangle \dot{y}_j$ has to be discretized. The derivative in Figure 4 can be given as:

$$\dot{y}_j = \frac{\triangle y_j}{\triangle t_j} \tag{59}$$

Using Equation (43) as a basis, $\triangle \dot{y}_j$ can be given as:

$$\triangle \dot{y}_j = \dot{y}_j - \dot{y}_0 = \frac{\triangle y_j}{\triangle t_j} - \frac{\triangle y_0}{\triangle t_0} \tag{60}$$

However, $\dot{y}_0$ does not exist in Figure 4. In order to have both $\triangle \dot{y}_j$ and $\triangle \dot{y}_0$, two perturbation steps have to be performed. An example of a two-step perturbation is illustrated in Figure 5.

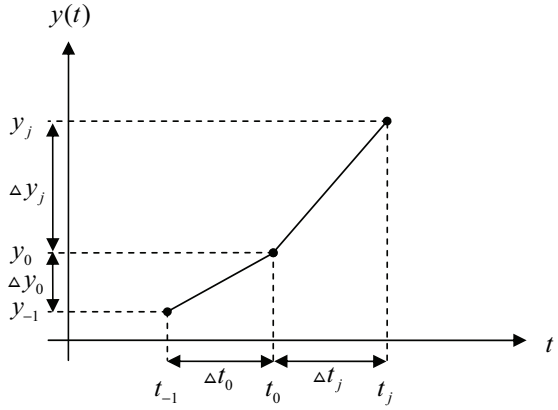


Figure 5: Two-step perturbation.

As for the one-step perturbation illustrated in Figure 4, the values $\triangle y_0$, $\triangle y_j$, $\triangle t_0$ and $\triangle t_j$ can be chosen arbitrarily, but it can be practical to express $\triangle y_j$ as a function of $\triangle t_j$, while $\triangle t_0$ can be given as $\triangle t_0 = \triangle t_j$ and $\triangle y_0 = 0$. Equation (60) can then be simplified to:

$$\triangle \dot{y}_j = \dot{y}_j - 0 = \frac{\triangle y_j}{\triangle t_j} \tag{61}$$

Inserting Equation (61) into Equation (58) yields:

$$K_d = \left(\frac{\triangle y_j}{\triangle t_j} \right)^{-1} \triangle u_j \tag{62}$$

2.4.4. *PID controller*

For a PID controller, the feedback controller output is given by Equation (23). In a discrete differential form, this can be written as:

$$\triangle u = \frac{\partial u}{\partial y} \triangle y + \frac{\partial u}{\partial \int y\, dt} \triangle \int y\, dt + \frac{\partial u}{\partial \dot{y}} \triangle \dot{y} \quad \text{or} \quad \triangle u = K_p \triangle y + K_i \triangle \int y\, dt + K_d \triangle \dot{y} \tag{63}$$

As demonstrated in Equation (63), a PID controller is a compound controller, consisting of both a proportional gain, an integral gain and a derivative gain. In order to estimate all three gains K_p, K_i and K_d using the perturbation technique, three perturbations need to be performed. And since the controller contains a derivative gain, a two-step perturbation algorithm needs to be used, as explained in Section 2.4.3. This gives the following set of equations:

$$\begin{aligned} \triangle u_1 &= K_p \triangle y_1 + K_i \triangle \int y_1 \, dt + K_d \triangle \dot{y}_1 \\ \triangle u_2 &= K_p \triangle y_2 + K_i \triangle \int y_2 \, dt + K_d \triangle \dot{y}_2 \\ \triangle u_3 &= K_p \triangle y_3 + K_i \triangle \int y_3 \, dt + K_d \triangle \dot{y}_3 \end{aligned} \tag{64}$$

which can be written in matrix form as:

$$\begin{bmatrix} \triangle u_1 \\ \triangle u_2 \\ \triangle u_3 \end{bmatrix} = \begin{bmatrix} \triangle y_1 & \triangle \int y_1 \, dt & \triangle \dot{y}_1 \\ \triangle y_2 & \triangle \int y_2 \, dt & \triangle \dot{y}_2 \\ \triangle y_3 & \triangle \int y_3 \, dt & \triangle \dot{y}_3 \end{bmatrix} \begin{bmatrix} K_p \\ K_i \\ K_d \end{bmatrix} \tag{65}$$

To derive the controller properties K_p, K_i and K_d, one can solve the following matrix system by the use of matrix inversion:

$$\begin{bmatrix} K_p \\ K_i \\ K_d \end{bmatrix} = \begin{bmatrix} \triangle y_1 & \triangle \int y_1 \, dt & \triangle \dot{y}_1 \\ \triangle y_2 & \triangle \int y_2 \, dt & \triangle \dot{y}_2 \\ \triangle y_3 & \triangle \int y_3 \, dt & \triangle \dot{y}_3 \end{bmatrix}^{-1} \begin{bmatrix} \triangle u_1 \\ \triangle u_2 \\ \triangle u_3 \end{bmatrix} \tag{66}$$

Inserting Equation (55) and Equation (61) into Equation (66) yields:

$$\begin{bmatrix} K_p \\ K_i \\ K_d \end{bmatrix} = \begin{bmatrix} \triangle y_1 & \left(y_0 + \frac{1}{2} \triangle y_1 \right) \triangle t_1 & \frac{\triangle y_1}{\triangle t_1} \\ \triangle y_2 & \left(y_0 + \frac{1}{2} \triangle y_2 \right) \triangle t_2 & \frac{\triangle y_2}{\triangle t_2} \\ \triangle y_3 & \left(y_0 + \frac{1}{2} \triangle y_3 \right) \triangle t_3 & \frac{\triangle y_3}{\triangle t_3} \end{bmatrix}^{-1} \begin{bmatrix} \triangle u_1 \\ \triangle u_2 \\ \triangle u_3 \end{bmatrix} \tag{67}$$

To avoid singularities when performing the matrix inversion in Equation (67), the determinant of the invertible matrix should be nonzero. This requirement is met for $\triangle y_1 \neq \triangle y_2 \neq \triangle y_3$ and $\triangle t_1 \neq \triangle t_2 \neq \triangle t_3$. Typically, $\triangle t_j$ and $\triangle y_j$ can be given as:

$$\triangle t_1 = \delta \cdot \triangle t_{sim} \quad , \quad \triangle y_1 = \triangle t_1 \quad , \quad \triangle t_j = j \cdot \triangle t_1 \quad , \quad \triangle y_j = j \cdot \triangle y_1 \tag{68}$$

where Δt_{sim} is the simulation time increment. δ is a small positive scalar called the relative perturbation step size [36]. A possible default value of δ, as used by the authors in Paper IV, is 0.1.

2.5. Eigensolver Methods

A description of some eigensolver methods may be helpful and is thus included in this following section, with focus on common methods found in commercial software systems. Because of various cooperative policies, the underlying methods in commercial software codes are often hard to identify and document.

Solving the eigenvalue problem $\mathbf{Ax} = \lambda\mathbf{Bx}$ is equivalent to calculating the roots of the characteristic polynomial equations. Since there is no general algebraic solution for the roots of polynomial equations of degree greater than four, it follows that all solution methods are necessarily iterative in nature. The different methods for solving the eigenvalue problem may be divided into three categories, based on the properties they utilize: polynomial iteration/ determinant methods, transformation methods and vector iteration methods [37]. The least applicable of these methods for large systems are the polynomial iteration/determinant methods; consequently, these methods will not be covered in this work. The basic property used by the transformation methods is [31]:

$$\mathbf{\Phi}^T\mathbf{A}\mathbf{\Phi} = \mathbf{\Lambda} \quad , \quad \mathbf{\Phi}^T\mathbf{B}\mathbf{\Phi} = \mathbf{I} \tag{69}$$

while for vector iteration methods, it is [31]:

$$\mathbf{A}\boldsymbol{\varphi}_i = \lambda_i\mathbf{B}\boldsymbol{\varphi}_i \tag{70}$$

where λ_i is an eigenvalue and $\boldsymbol{\varphi}_i$ its corresponding eigenvector.

The transformation methods rely on transformations of the matrix under investigation into another matrix of same eigenvalues. Typical transformation methods are Givens tridiagonalization [37], Householders tridiagonalization [31, 37], and the QR [3, 31, 37-42] and QZ algorithms [40, 41]. The QR algorithm may be applied to solve the standard eigenvalue problem $\mathbf{Cx} = \lambda\mathbf{x}$ [43], while the QZ algorithm is an extension of the QR algorithm and may be used to solve the generalized eigenvalue problem $\mathbf{Ax} = \lambda\mathbf{Bx}$ [40]. For the QZ algorithm, **A** and **B** are typically transformed into upper triangular matrices from which the eigenvalues may be calculated from the diagonal elements. The QZ algorithm is implemented in LAPACK[3], and may be used to solve the generalized non-symmetric eigenproblem. Several commercial software systems, such as MATLAB and FEDEM, use LAPACK. One of the LAPACK subroutines for computing the generalized eigenvalues for a pair of *n*-by-*n* real non-symmetric matrices, and optionally, the left and/or right generalized eigenvectors is the DGGEVX [44] routine. FEDEM

[3] LAPACK – Linear Algebra PACKage, a software package provided by Univ. of Tennessee; Univ. of California, Berkeley; Univ. of Colorado Denver and NAG Ltd. Version 3.3.0.

R5.0 has the option of using this routine to solve the generalized eigenvalue problem. On the issue: "What algorithm does the EIG function use to compute eigenvalues?", the MathWorks replied in 2009: "As of MATLAB 6.0 (R12) the EIG function makes use of several LAPACK functions to compute the eigenvalues and eigenvectors, depending on the structure of the matrix." [45]. The function `eig(A,B,'qz')` in MATLAB uses the QZ algorithm [46].

The vector iteration methods are based upon iterations of arbitrary starting vectors in order to derive the systems eigenvalues and eigenvectors. One of the core algorithms in this group of methods is power iteration [37] or forward iteration [31]. Starting with the standard eigenvalue problem $\mathbf{Cx} = \lambda\mathbf{x}$ and an arbitrary starting vector $\mathbf{x}_0$; by repeatedly multiplying this vector with matrix $\mathbf{C}$, a vector is obtained which is more and more parallel with the dominant eigenvector, i.e. the eigenvector corresponding to the largest eigenvalue [37]. For most engineering applications, the lowest rather than the highest eigenpair is of most interest. Using the abovementioned procedure on $\mathbf{C}^{-1}\mathbf{x} = \lambda^{-1}\mathbf{x}$ instead of $\mathbf{Cx} = \lambda\mathbf{x}$ yields the dominant eigenpair of $\mathbf{C}^{-1}$, which is the lowest or least dominant eigenpair of $\mathbf{C}$. This procedure is called inverse vector iteration. By introducing a shifting factor to these methods, other eigenpairs than the highest or lowest can be acquired. According to Bell [37], the most popular version of simultaneous inverse vector iteration methods is the subspace iteration method, developed and named by Bathe [31]. This method is generally applied to the generalized eigenvalue problem, and seeks to determine all the m lowest eigenpairs through an inverse iteration scheme that operates on p trial vectors, or a p-dimensional subspace, where $p \geq m$. In 1950, Lanczos [47] proposed a method for transforming a given $n \times n$ matrix into a similar tridiagonal matrix in n iteration steps. However, as already recognized by Lanczos, the tridiagonalization procedure has a major shortcoming in that the constructed vectors, which in theory should be orthogonal, are, as a result of round-off errors, not orthogonal in practice. A remedy is to use Gram-Schmidt orthogonalization, but such an approach is also sensitive to round-off errors and renders the process ineffective when a complete matrix is to be tridiagonalized. If, on the other hand, the objective is to calculate only a few eigenvectors and corresponding eigenvectors of the problem $\mathbf{Ax} = \lambda\mathbf{Bx}$, an iteration based on the Lanczos transformation can be very efficient [31]. The technique can be used to solve certain large, sparse, symmetric eigenvalue problems [41], and is now developed into perhaps the most efficient and robust technique for finding a limited number of eigenvalues/eigenvectors of large symmetric eigenvalue problems [37]. The symmetric Lanczos method can also be extended to unsymmetric matrices using the Arnoldi method or the unsymmetric Lanczos method [41]. The unsymmetric Lanczos method is generally advised against due to instability [41, 48, 49], while the Arnoldi method is suffering from some storage issues [49, 50]. FEDEM R5.0 has the option of using the Lanczos method to solve the symmetric generalized eigenvalue problem.

For real eigenvalue problems, MSC.Nastran[4] [51] offers methods in two categories: the reduction (tridiagonal) method, and the iterative (Lanczos) method. The reduction methods are: Givens method, Householder method, Modified Givens method and Modified Householder method. For these methods, MSC.Nastran employs the QR transformation. Although the methods are mathematically interchangeable, the Lanczos method is recommended in MSC.Nastran for the solution of large buckling and normal modes problems, while the reduction methods are in

[4] MSC.Nastran by MSC Software.

MSC.Nastran useful for small normal modes problems in analysis of structural components. For complex eigenvalue problem, MSC.Nastran offers QZ and complex Lanczos methods.

From the Abaqus[5] Version 6.7 Documentation [52] it is said that Abaqus provides eigenvalue extraction procedures for both symmetric and complex eigenproblems. For symmetrized eigenproblems Abaqus/Standard offers two approaches: Lanczos and subspace iteration methods. For complex eigenproblems the subspace projection method is used. Both the subspace iteration and the Lanczos methods use the Householder and QR algorithm for the reduced eigenproblem, and that the Lanczos solver with the traditional architecture is the default eigenvalue extraction method because it has the most general capabilities.

In practical engineering applications, the system matrices are commonly large and real, and only a few of the system's eigensolutions are of interest. For undamped mechanical problems, the system matrices will also be symmetrical, and as long as the sensors and corresponding actuators of the controller are collocated, the system matrices will remain symmetrical. The eigensolutions of these systems can be obtained by solving the eigenvalue problem given by Equation (38), and the solvers based on the Lanczos method may be a preferred choice. The Lanczos method gives faster convergence than subspace iteration [49, 53] and, as stated by Fischer [54], the direct application of QR type algorithms is limited to systems of several hundred DOFs and are thus not useful to solve large engineering problems with many thousand DOFs where actually only a limited set of the lowest modes is required. If damping is introduced into the system, Equation (2) is typically required to solve the eigenvalue problem using state-space methods, meaning that the eigenproblem matrices **A** and **B** may be unsymmetrical, unless the state-space form of Equation (9) or Equation (11) is used. In addition, as described in Section 2.3. and Paper III, non-collocated sensors and actuators will typically yield unsymmetrical system matrices. A potent solver method for such systems is the Arnoldi method, which over the years has been attempted improved in order to overcome some of its storage issues [49, 55, 56].

[5] Abaqus FEA by Dassault Systèmes

3. Summary of Appended Papers

3.1. Paper I: Modal Analysis of Lumped Flexible Active Systems (Part 1)

This paper was presented at the SIMS 2008: The 48th Scandinavian Conference on Simulation and Modeling in Oslo, Norway in 2008. It served the purpose of being an initial investigation into the field of modal analysis of active flexible multibody systems. The mechanical system was limited to be an SDOF mass-spring-damper system, and the controller was limited to be of type position feedback PD controller. This paper highlights some of the problems regarding modal analysis of active systems, particularly the one about external forces being set to zero for free vibration analyses. An active system containing no mechanical components other than a mass is shown to possess a natural frequency, which cannot be predicted by classical methods. But by investigating the active system, it is shown how the controller forces affecting the internal dynamics of the system, based on the equation of motion, can be isolated and included in a modal analysis. Equations for the damped and undamped natural frequencies of active systems based on classical equations are presented and verified through examples. The main results from this paper are shown in the following paragraph.

The equation of motion for an active SDOF system containing a position feedback PD controller is shown to be:

$$m\ddot{r}(t)+\left(c+K_d\right)\dot{r}(t)+\left(k+K_p\right)r(t)=F(t)+K_d\dot{r}_{ref}(t)+K_p r_{ref}(t) \tag{71}$$

which yields the equation of motion for the free vibration as:

$$m\ddot{r}(t)+\left(c+K_d\right)\dot{r}(t)+\left(k+K_p\right)r(t)=0 \tag{72}$$

Equation (72) yields the following equations for the undamped and damped natural frequency and the damping ratio as, respectively:

$$\omega_n=\sqrt{\frac{\left(k+K_p\right)}{m}} \tag{73}$$

$$\omega_d=\sqrt{\frac{\left(k+K_p\right)}{m}}\cdot\sqrt{1-\zeta^2} \tag{74}$$

$$\zeta=\frac{\left(c+K_d\right)}{\sqrt{4m\left(k+K_p\right)}} \tag{75}$$

3.2. Paper II: Modal analysis of active flexible multibody systems

This paper has been published in the journal Computers and Structures. The main objective of this paper was to derive a method for performing modal analyses of active flexible multibody systems based on second-order differential equations. The paper was limited to dealing with the undamped eigenfrequencies of active MDOF systems containing SISO or MIMO PID controllers, and builds on findings in Paper I. It reveals how the active elements can be broken up into three separate parts (sensor, actuator and controller), and shows how the equations of motion for active systems are affected by the sensor input (position, velocity or acceleration) to the controllers. Further, an equation for the generalized undamped eigenvalue problem for active systems is presented and verified through two examples. The examples also serve the purpose of highlighting some of the shortcomings of current methods: FE-based methods do not include the controller properties and analytical methods do not include flexible body dynamics. The main results from this paper are exhibited in the following paragraph.

The equation of motion for the free vibration of an active SDOF system containing a position, velocity or acceleration feedback PID controller is shown to be, respectively:

$$m\ddot{r}(t)+\left(c+K_d\right)\dot{r}(t)+\left(k+K_p\right)r(t)+K_i\int r(t)\,dt=0 \tag{76}$$

$$\left(m+K_d\right)\ddot{r}(t)+\left(c+K_p\right)\dot{r}(t)+\left(k+K_i\right)r(t)=0 \tag{77}$$

$$K_d\dddot{r}(t)+\left(m+K_p\right)\ddot{r}(t)+\left(c+K_i\right)\dot{r}(t)+kr(t)=0 \tag{78}$$

The equation for the gradient matrices of a MIMO controller can be written in matrix form as:

$$d\mathbf{F}_{Ctrl}=\frac{\partial\mathbf{F}_{Ctrl}}{\partial\mathbf{u}}\frac{\partial\mathbf{u}}{\partial\mathbf{y}}\frac{\partial\mathbf{y}}{\partial\mathbf{x}}d\mathbf{x}=\mathbf{G}_{Act}\mathbf{G}_{Ctrl}\mathbf{G}_{Sens}\,d\mathbf{x} \tag{79}$$

where $\mathbf{G}_{Act}$, $\mathbf{G}_{Ctrl}$ and $\mathbf{G}_{Sens}$ are the actuator, controller and sensor gradient matrices, respectively. In Equation (79), matrix $\mathbf{G}_{Act}$ has the dimensions $n_{F_{Ctrl}}\times n_u$, where $n_{F_{Ctrl}}$ is the number of controller forces and n_u is the number of controller outputs; matrix $\mathbf{G}_{Ctrl}$ has the dimensions $n_u\times n_y$, where n_u is the number of controller outputs and n_y is the number of controller inputs and matrix $\mathbf{G}_{Sens}$ has the dimensions $n_y\times 3n_r$, where n_y is the number of controller inputs and n_r is the number of all system DOFs. Vector $\mathbf{x}$ is on the form $\mathbf{x}=\left[\mathbf{r}\ \dot{\mathbf{r}}\ \ddot{\mathbf{r}}\right]^T$ and thus has the dimensions $3n_r\times 1$, where n_r is the number of all system DOFs. Transforming the gradient matrices $\mathbf{G}_{Act}$, $\mathbf{G}_{Ctrl}$ and $\mathbf{G}_{Sens}$ into system gradient matrices of dimension $n\times n$, $\mathbf{G}_{Acc}$, $\mathbf{G}_{Vel}$ and $\mathbf{G}_{Pos}$, the equation of motion for the free vibration of an active MDOF system containing MIMO controllers can be given as:

$$\left(\mathbf{M}+\mathbf{G}_{Acc}\right)\ddot{\mathbf{r}}(t)+\left(\mathbf{C}+\mathbf{G}_{Vel}\right)\dot{\mathbf{r}}(t)+\left(\mathbf{K}+\mathbf{G}_{Pos}\right)\mathbf{r}(t)=\mathbf{0} \tag{80}$$

where the dimensions of all the matrices are $n \times n$ and the vectors $n \times 1$. Equation (80) yields the following equation for the generalized undamped eigenvalue problem:

$$\left(\mathbf{K} + \mathbf{G}_{Pos}\right)\mathbf{\Phi} = \left(\mathbf{M} + \mathbf{G}_{Acc}\right)\mathbf{\Phi}\mathbf{\Lambda} \tag{81}$$

3.3. Paper III: Modal Analysis of Active Flexible Multibody Systems Containing PID Controllers with Non-Collocated Sensors and Actuators

This paper is submitted for publication in the journal Computers and Structures. The main objective of this paper was to build on and further expand the theory derived in Paper II, focusing on active systems with non-collocated sensors and actuators and PID controllers containing position feedback integral gains. The paper presents equations of motion for the above-mentioned systems and gives equations for performing modal analysis of such systems, which are verified through examples. The main results from this paper are shown in the following paragraph.

The equation of motion for the free vibration of an MDOF system containing a position feedback PID controller can be written as:

$$\left(\mathbf{M} + \mathbf{G}_{Acc}\right)\ddot{\mathbf{r}}(t) + \left(\mathbf{C} + \mathbf{G}_{Vel}\right)\dot{\mathbf{r}}(t) + \left(\mathbf{K} + \mathbf{G}_{Pos}\right)\mathbf{r}(t) + \mathbf{G}_{SSEE}\int \mathbf{r}(t)dt = \mathbf{0} \tag{82}$$

where the matrix $\mathbf{G}_{SSEE}$ is the gradient matrix of the steady-state error elimination, which corresponds to the position feedback integral gain. For an active SISO system with n system DOFs in which a sensor is placed on DOF i and an actuator is acting on DOF j, the controller gradient matrices will contain non-zero elements on position ij. If the controller is of type position feedback PID, the gradient matrices will be: $\mathbf{G}_{Acc_{ij}} = 0$, $\mathbf{G}_{Vel_{ij}} = K_d$, $\mathbf{G}_{Pos_{ij}} = K_p$ and $\mathbf{G}_{SSEE_{ij}} = K_i$. If the controller is of type velocity feedback PID, the gradient matrices will be: $\mathbf{G}_{Acc_{ij}} = K_d$, $\mathbf{G}_{Vel_{ij}} = K_p$, $\mathbf{G}_{Pos_{ij}} = K_i$ and $\mathbf{G}_{SSEE_{ij}} = 0$. Additionally, Equation (82) can be rewritten as:

$$\mathbf{M}_{eff}\ddot{\mathbf{r}} + \mathbf{C}_{eff}\dot{\mathbf{r}} + \mathbf{K}_{eff}\mathbf{r} + \mathbf{Q}_{eff}\int \mathbf{r}\,dt = \mathbf{0} \tag{83}$$

where $\mathbf{M}_{eff}$ is the effective mass matrix of the system, while $\mathbf{C}_{eff}$ is the effective damping matrix, $\mathbf{K}_{eff}$ the effective stiffness matrix and $\mathbf{Q}_{eff}$ the effective steady-state error elimination matrix of the system. The dimensions of all the matrices are still $n \times n$ and the vectors $n \times 1$. Equation (83) can be transformed into a first-order form as:

$$\mathbf{A}\mathbf{x} - \mathbf{B}\dot{\mathbf{x}} = \mathbf{0} \tag{84}$$

where:

$$\mathbf{x}=\begin{bmatrix}\int \mathbf{r}\,dt\\ \mathbf{r}\\ \dot{\mathbf{r}}\end{bmatrix},\quad \dot{\mathbf{x}}=\begin{bmatrix}\mathbf{r}\\ \dot{\mathbf{r}}\\ \ddot{\mathbf{r}}\end{bmatrix},\quad \mathbf{A}=\begin{bmatrix}\mathbf{Q}_{eff} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{K}_{eff} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{eff}\end{bmatrix},\quad \mathbf{B}=\begin{bmatrix}\mathbf{-K}_{eff} & \mathbf{-C}_{eff} & \mathbf{-M}_{eff}\\ \mathbf{K}_{eff} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{M}_{eff} & \mathbf{0}\end{bmatrix} \tag{85}$$

The dimensions of $\mathbf{x}$ and $\dot{\mathbf{x}}$ are $3n\times 1$ and those of $\mathbf{A}$ and $\mathbf{B}$ are both $3n\times 3n$. The generalized eigenvalue problem for the active flexible multibody system can thus be solved as given by Equation (2).

3.4. Paper IV: A Method for Controller Parameter Estimation Based on Perturbations

This paper is submitted for publication in the journal Multibody System Dynamics. The main objective of this paper was to derive a method for estimating the unknown controller gains K_p, K_i and K_d in the controller gradient matrix $\mathbf{G}_{Ctrl}$, shown in Equation (79), of a SISO or MIMO PID controller. Since, as stated in the introduction, the controller is comparable to a "black box" or unknown function as seen from the mechanical engineer's point of view, the parameters K_p, K_i and K_d are not necessarily explicitly known a priori for the engineer performing modal analyses. This paper presents a method which uses incremental changes (perturbations) in the controller inputs to derive the controller gains, and the method is verified through examples. The main results from this paper are shown in the following paragraph.

The feedback controller output u of a PID controller can be given in a discrete differential form as a function of the controller input y as:

$$\triangle u=\frac{\partial u}{\partial y}\triangle y+\frac{\partial u}{\partial \int y\,dt}\triangle\int y\,dt+\frac{\partial u}{\partial \dot{y}}\triangle\dot{y}\quad\text{or}\quad \triangle u=K_p\triangle y+K_i\triangle\int y\,dt+K_d\triangle\dot{y} \tag{86}$$

Hence, for a PID controller, the gains K_p, K_i and K_d can be derived from solving the following matrix equation:

$$\begin{bmatrix}K_p\\ K_i\\ K_d\end{bmatrix}=\begin{bmatrix}\triangle y_1 & \triangle\int y_1\,dt & \triangle\dot{y}_1\\ \triangle y_2 & \triangle\int y_2\,dt & \triangle\dot{y}_2\\ \triangle y_3 & \triangle\int y_3\,dt & \triangle\dot{y}_3\end{bmatrix}^{-1}\begin{bmatrix}\triangle u_1\\ \triangle u_2\\ \triangle u_3\end{bmatrix} \tag{87}$$

where $\triangle y_j$ and $\triangle u_j$ are incremental changes in controller input and output, respectively, for perturbation j. Discretization of $\triangle\int y_j\,dt$ and $\triangle\dot{y}_j$ yields Equation (87) as:

$$\begin{bmatrix} K_p \\ K_i \\ K_d \end{bmatrix} = \begin{bmatrix} \Delta y_1 & \left(y_0 + \frac{1}{2}\Delta y_1 \right)\Delta t_1 & \frac{\Delta y_1}{\Delta t_1} \\ \Delta y_2 & \left(y_0 + \frac{1}{2}\Delta y_2 \right)\Delta t_2 & \frac{\Delta y_2}{\Delta t_2} \\ \Delta y_3 & \left(y_0 + \frac{1}{2}\Delta y_3 \right)\Delta t_3 & \frac{\Delta y_3}{\Delta t_3} \end{bmatrix}^{-1} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \end{bmatrix} \tag{88}$$

where Δt_j is the time increment for perturbation j.

4. Discussions

4.1. Discussion of the *n*-Dimensional and 3*n* State-Space Versions

Presented by Equation (38) and in Paper II is an *n*-dimensional modal analysis version which can be used to solve systems containing marginal, negligible or no damping or steady-state error elimination. Undercritically damped natural frequencies do not differ significantly from undamped natural frequencies, the reason for this can be seen in the following equation [10]:

$$\omega_d = \omega_n \sqrt{1-\zeta^2} \tag{89}$$

where ω_n and ω_d are the undamped and damped natural frequencies, respectively, and ζ is the damping ratio. Using the *n*-dimensional modal analysis version has several advantages over the more comprehensive 3*n* state-space version. FE-based multibody system simulation software, such as FEDEM, is usually based on the second-order mechanical dynamics equations. By keeping both the modal analysis and the time-domain dynamic equations in the *n*-space, the method may experience a higher degree of compatibility and be more easily implemented into such a software system, since, as stated by Alvin and Park [57], the equations determined in a state-space form are difficult to transform into well-known second-order equations.

A typical method or algorithm for solving the full eigenvalue problem in FE software systems is the QR or QZ algorithm; the QZ algorithm being a generalization the QR algorithm. The QR algorithm is of order n^3 [37]. For the proposed 3*n* state-space version, this would mean an $(3n)^3/n^3$ increase in computation time for systems of large *n*, which means that solving the eigenvalue problem using the proposed 3*n* state-space version will be up to 27 times more expensive with respect to computational time than an *n*-dimensional version. In addition, as stated in Section 2.1., both eigenvalues and eigenvectors of the *n*-dimensional version will be real, significantly reducing the complexity of the solutions. However, the main drawback of the *n*-dimensional version is accuracy. The 3*n* state-space version includes both damping and steady-state error elimination effects, complementing the *n*-dimensional version on the accuracy issue. In addition, one positive side effect by the 3*n* state-space version is that both positive and negative damping can be calculated, making this version a potential tool for capturing system instability issues.

One note about implementing these eigenvalue problem versions into software systems should be made. Most mechanical systems will only contain symmetrical matrices. As long as the sensors and actuators of the controllers are collocated, the system matrices will still be symmetric. However, if the sensors and actuators are non-collocated, the affected system matrices will become unsymmetrical, meaning that only unsymmetrical eigensolvers should be used, regardless of whether the *n*-dimensional or 3*n* state-space versions are used. This is an issue solely dependent on the eigensolver capacities and not on eigenproblem setup versions.

4.2. Returning to the Practical Engineering Questions

In Section 1.3., several relevant engineering questions regarding the topic addressed in this thesis were asked. With the knowledge presented in this work at hand, these questions can now be answered. Below are the questions repeated, and their answers listed in successive order.

Question 1: A proportional-derivative (PD) controller and a hydraulic actuator are driving a suspension system. In what way does the effective system mass, stiffness and damping, and hence the eigenfrequencies of the system, become affected? What if the controller use both position, velocity and force feedback? Can fixed boundary conditions be applied to the driving DOFs to remove the singularities occurring when the forces are set to zero? If not, how can the controller be represented by equivalent mechanical properties supported by the FE software?

Answer: Depending on the type of sensor input, the system mass, stiffness and damping will be affected as illustrated in Figure 3. If fixed boundary conditions are applied to the driving DOFs, the singularities which will occur due to forces being set to zero for free vibrations will be omitted. However, the predicted modal parameters may be incorrect since fixing the boundary conditions introduces additional erroneous constraints on the system. The controller may be represented by equivalent mechanical properties as outlined by Equations (27) and (34), and its modal parameters derived by Equations (38) or (39).

Question 2: One might know how to solve the problem raised by Question 1, however, the controller may also contain discrete elements like hysteresis, logical switches, dead zones, time delays and limit elements. How do these types of controller elements affect the closed-loop eigenfrequencies, and can they be represented by mechanical properties in the FE software?

Answer: As shown explicitly in Paper IV through examples, discrete elements may be treated as other discontinuous elements. The perturbation technique presented in details in Paper IV has the capabilities to calculate the effective controller gains at any instant in the dynamic time-domain motion simulation. Each time a modal analysis is performed, the instantaneous controller gains are derived. These gains may now be added to the system matrices in accordance with Equations (27), (34) and (36).

Question 3: It may be decided to use a PID controller to minimize position and velocity deviations on a machining centre. Does the integral part of the controller affect the stiffness or damping of the mechanical system, and does the derivate effect introduce any artificial inertia to the mechanical system?

Answer: Depending on the type of sensor input, the integral part of the controller might affect either stiffness or damping. If the sensor input is of type position, then the integral part will negatively affect the damping of the system. A too high integral gain for such a controller may lead to system instability and ultimately system failure. If the sensor input is of type velocity, then the integral part will affect the stiffness, whereas if the sensor input is of type acceleration or force, the integral part will affect the damping of the system. The derivative effect may

introduce artificial inertia to the system if the sensor input is of type velocity. If the sensor input is of type acceleration or force, the damping is equivalent to the time derivative of acceleration, also known as jerk or jolt. This has, however, not been covered in any extent in this work.

Question 4: It may be desired to optimize a rotating machinery, and the clutch and actuator have a limited and nonlinear torque capacity. How can one model the boundary conditions when they are dependent on the reaction torque? When the applied torque exceeds the clutch and actuator capacity the machinery is suddenly free to rotate!

Answer: By modeling the torque capacity as a limited controller, the coupling between the clutch and actuator may be kept intact as long as the reaction torque does not exceed the torque capacity. If or when the reaction torque exceeds the torque capacity, the coupling will then be broken and the rotational DOF will be free. Similarly as for the discrete elements asked for in Question 2, the perturbation technique can be utilized in order to detect changes in discontinuous elements. Additional boundary conditions may be applied as long as the reaction torque is within the limits of the torque capacity, and removed at the instant the torque capacity is exceeded.

5. Conclusions

In this thesis, a method for performing modal analyses of active flexible multibody systems based on the generalized eigenvalue problem has been presented. The overall aim of this work has been to make engineers working in a finite element environment able to accurately predict modal parameters of such systems. This work is aimed at covering the controller types most commonly used in the industry. Therefore, this work has been limited to single or multiple degrees of freedom flexible multibody systems with position, velocity or acceleration feedback single or multiple input and output PID controllers. The sensors and actuators to the controller can be either collocated or non-collocated, and the controller can contain both continuous and discontinuous elements. The presented method is intended to be implemented into a finite element software system.

Two modal analysis versions have been proposed: one simplified and one complete. For the simplified version, the second-order n-dimensional differential equations form the basis for the method. The controller properties are incorporated into their respective system matrices, and the generalized eigenvalue problem is solved using the system's mass and stiffness matrices. Damping, both passive and active, and steady-state error elimination are therefore not included in this version. This version derives the active system's undamped eigenfrequencies and their corresponding real mode shapes. For the complete version, the second-order n-dimensional differential equations are transformed into a first-order $3n$-dimensional form. As for the simplified version, the controller properties are added to their respective system matrices, but the generalized eigenvalue problem is solved using all of the system matrices. Hence, this version derives the active system's complex eigenvalues and their corresponding complex mode shapes. Both versions of the presented method can handle collocation and/or non-collocation of sensors and actuators.

Since the controller parameters may not be explicitly defined for the engineer working in a finite element environment, a method for deriving the controller gains for PID controllers using perturbations has been presented. By using the proposed perturbation technique, the controller properties can be estimated and added to their respective system matrices, making the perturbation technique a useful supplement for performing modal analyses of active flexible multibody systems in a finite element environment. It should be noted that the proposed method, particularly the perturbation technique, assumes a separation of the controller and the mechanical components, as shown in Figure 2. In control system synthesis, it is a common practice to include simplified mechanical properties such as inertias and spring properties into the controller model. Combining such controller models with a finite element model of the flexible multibody system will cause the properties of the active system to become severely altered in comparison to the actual physical or intended product, since some system properties will be counted for twice.

So, what has been achieved by this Ph.D. project? Looking back at the Motivation section on page III, this project has its origin in a simulation inefficiency and inaccuracy encountered by Professor Terje Rølvåg during some of his previous work. The desire was to have an available tool that could eliminate the need for working with two separate system models for time domain simulations and modal analyses of active flexible multibody systems. Even though this tool is not yet completely implemented, this work represents a vital step on the road to providing a

countermeasure to this simulation inefficiency and inaccuracy. In this thesis, a method which eliminates the need for two separate system models for time domain simulations and modal analyses of active flexible multibody systems has been presented. Now, both passive and active system properties can be included, not only when performing time domain simulations, but also when performing modal analyses. By using the presented method, the work load required to perform simulations of active flexible multibody systems should be reduced by almost 50 %. In addition, the presented method should also yield more accurate eigenfrequency results than traditional methods, since both flexible multibody dynamics and controller effects are accounted for. Some of the results from this Ph.D. project have been highlighted through the practical engineering questions given in Sections 1.3. and 4.2. With this new knowledge at hand, these have now been answered.

6. Future Work

To move this project further, a handful of options are available. Even though the presented method is intended to be implemented into a finite element software system, this has not yet been fully accomplished. Implementation may reveal possible problems or shortcomings of the presented method that have not been considered by the author. In addition, the method and its algorithms have not been considered to be optimized with respect to computational effort. Optimizing the algorithms may greatly enhance the presented method with respect to for instance implementation, computational efficiency and ease of use.

Deriving the modal parameters of the active system using the complete modal analysis version may yield complex eigenvectors. For the $2n$ state-space method presented by Foss [29], the information obtained by the $2n$ eigenvector matrix is explained. For the proposed $3n$ state-space method, no investigation has been made regarding the $3n$ eigenvector matrix. Nevertheless, it is likely that since one of the roots s in the cubic characteristic equation of Equation (76) is always real, its corresponding eigenvalues and eigenvectors can be ruled out of the equation. For the $2n$ state-space method, there is a correlation between the eigenvectors and the state vector $\mathbf{x}$ in Equation (10), as explained by Equation (7). This is also most likely the case for the $3n$ state-space method.

As stated in both the introduction and conclusion of this thesis, the work conducted herein has been limited to PID controllers, which are the most common type of controllers in use today. One possible extension of this project is to expand the theory to include other types of controllers.

It would also be very interesting to see the method presented in this work applied in the design process of an actual product. Issues which could be clarified by performing such a task are for instance the actual increase in accuracy of the simulation results, how great the work load reduction due to the presented method will be and where the area of validity of the presented method lies.

References

[1] Rølvåg, T., "ESA study - SAR modelling and simulation results," in *8th European Space Mechanisms and Tribology Symposium*, Toulouse, France, 1999.

[2] Billah, K. Y. and Scanlan, R. H., "Resonance, Tacoma Narrows bridge failure, and undergraduate physics textbooks," *American Journal of Physics,* vol. 59, pp. 118-24, 1991.

[3] Thomson, W. T. and Dahleh, M. D., *Theory of Vibration with Applications*, 5th ed. Upper Saddle River, NJ, USA: Prentice Hall, Inc., 1998.

[4] Avitabile, P., "Experimental modal analysis a simple non-mathematical presentation," *S V Sound and Vibration,* vol. 35, pp. 20-31, 2001.

[5] Ewins, D. J., *Modal testing: theory, practice and application.* Baldock, England: Research Studies Press, 2000.

[6] Cooley, J. W. and Tukey, J. W., "An algorithm for the machine calculation of complex fourier series," *Mathematics of Computation,* vol. 19, pp. 297-301, 1965.

[7] Preumont, A., *Vibration Control of Active Structures: An Introduction*, 2nd ed. Dordrecht, The Netherlands: Kluwer Academic Publishers, 2002.

[8] Inman, D. J., *Vibration with control*. Chichester, England: John Wiley & Sons Ltd., 2006.

[9] Balchen, J. G., Andresen, T., and Foss, B. A., *Reguleringsteknikk*, 5th ed. Trondheim, Norway: Department of Engineering Cybernetics, Norwegian University of Science and Technology, 2003. (In Norwegian).

[10] Palm, W. J., *Mechanical Vibration*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2007.

[11] Cook, R. D., Malkus, D. S., Plesha, M. E., and Witt, R. J., *Concepts and applications of finite element analysis*, 4th ed. New York, NY, USA: John Wiley & Sons. Inc., 2002.

[12] Sivertsen, O. I., *Virtual Testing of Mechanical Systems - Theories and Techniques*. Lisse, The Netherlands: Swets & Zeitlinger B.V., 2001.

[13] Meirovitch, L., *Dynamics and control of structures*. New York, NY, USA: John Wiley & Sons. Inc., 1990.

[14] Rastgaar, M. A., Ahmadian, M., and Southward, S. C., "Orthogonal eigenstructure control with non-collocated actuators and sensors," *Journal of Vibration and Control,* vol. 15, pp. 1019-1047, 2009.

[15] Alkhatib, R. and Golnaraghi, M. F., "Active structural vibration control: a review," *Shock and Vibration Digest,* vol. 35, pp. 367-83, 2003.

[16] Géradin, M. and Cardona, A., *Flexible Multibody Dynamics: A Finite Element Approach*. Chichester, England: John Wiley & Sons, Ltd., 2001.

[17] Sivertsen, O. I. and Waloen, A. O., "Non-Linear Finite Element Formulations for Dynamic Analysis of Mechanisms with Elastic Components," Washington, DC, USA, 1982, p. 7.

[18] Bratland, M. and Rølvåg, T., "Modal Analysis of Lumped Flexible Active Systems (Part 1)," presented at the SIMS 2008: The 48th Scandinavian Conference on Simulation and Modeling, Oslo, Norway, 2008.

[19] Sharon, A., Hogan, N., and Hardt, D. E., "Controller design in the physical domain," *Journal of the Franklin Institute,* vol. 328, pp. 697-721, 1991.

[20] Baz, A., "Active control of periodic structures," *Transactions of the ASME. Journal of Vibration and Acoustics,* vol. 123, pp. 472-9, 2001.

[21] Bernzen, W., "Active vibration control of flexible robots using virtual spring-damper systems," *Journal of Intelligent and Robotic Systems: Theory and Applications,* vol. 24, pp. 69-88, 1999.

[22] Ohnishi, K., Shibata, M., and Murakami, T., "Motion control for advanced mechatronics," *IEEE/ASME Transactions on Mechatronics,* vol. 1, pp. 56-67, 1996.

[23] Ryu, J.-H., Kwon, D.-S., and Hannaford, B., "Stability guaranteed control: Time domain passivity approach," *IEEE Transactions on Control Systems Technology,* vol. 12, pp. 860-868, 2004.

[24] Tzou, H. S. and Tseng, C. I., "Distributed Piezoelectric Sensor Actuator Design for Dynamic Measurement Control of Distributed Parameter-Systems - a Piezoelectric Finite-Element Approach," *Journal of Sound and Vibration,* vol. 138, pp. 17-34, April 8, 1990.

[25] Lim, Y. H., Gopinathan, S. V., Varadan, V. V., and Varadan, V. K., "Finite element simulation of smart structures using an optimal output feedback controller for vibration and noise control," *Smart Materials & Structures,* vol. 8, pp. 324-337, June 1999.

[26] Astrom, K. J. and Hagglund, T., "The future of PID control," *Control Engineering Practice,* vol. 9, pp. 1163-1175, 2001.

[27] Astrom, K. J. and Hagglund, T., "Revisiting the Ziegler-Nichols step response method for PID control," *Journal of Process Control,* vol. 14, pp. 635-650, 2004.

[28] Adhikari, S., "Rates of change of eigenvalues and eigenvectors in damped dynamic system," *Aiaa Journal,* vol. 37, pp. 1452-1458, 1999.
[29] Foss, K. A., "Coordinates which uncouple equations of motion of damped linear dynamic systems," in *ASME Meeting A-86, Dec 1-6 1957*, New York, NY, USA, 1957, p. 4.
[30] Alvin, K. F., Robertson, A. N., Reich, G. W., and Park, K. C., "Structural system identification: From reality to models," *Computers and Structures,* vol. 81, pp. 1149-1176, 2003.
[31] Bathe, K.-J., *Finite Element Procedures*. Englewood Cliffs, NJ, USA: Prentice Hall, 1996.
[32] Park, K. C., Felippa, C. A., and Ohayon, R., "The d'Alembert-Lagrange principal equations and applications to floating flexible systems," *International Journal for Numerical Methods in Engineering,* vol. 77, pp. 1072-1099, 2009.
[33] Felippa, C. A., "A historical outline of matrix structural analysis: A play in three acts," *Computers and Structures,* vol. 79, pp. 1313-1324, 2001.
[34] Astrom, K. J. and Eykhoff, P., "System identification-a survey," *Automatica,* vol. 7, pp. 123-62, 1971.
[35] Perreault, E. J., Kirsch, R. F., and Acosta, A. M., "Multiple-input, multiple-output system identification for characterization of limb stiffness dynamics," *Biological Cybernetics,* vol. 80, pp. 327-37, 1999.
[36] Trier, S. D., "Design Optimization of Flexible Multibody Systems," Doctoral Thesis, Department of Machine Design and Materials Technology, Norwegian University of Science and Technology, Trondheim, Norway, 2001.
[37] Bell, K., *Eigensolvers for structural problems: some algorithms for symmetric eigenvalue problems and their merits*. Delft, The Netherlands: Delft University Press, 1998.
[38] Francis, J. G. F., "The QR Transformation A Unitary Analogue to the LR Transformation—Part 1," *The Computer Journal,* vol. 4, pp. 265-271, January 1, 1961 1961.
[39] Francis, J. G. F., "The QR Transformation—Part 2," *The Computer Journal,* vol. 4, pp. 332-345, January 1, 1962 1962.
[40] Moler, C. B. and Stewart, G. W., "An algorithm for generalized matrix eigenvalue problems," *SIAM Journal on Numerical Analysis,* vol. 10, pp. 241-56, 1973.
[41] Golub, G. H. and Van Loan, C. F., *Matrix computations*, 2nd ed. Baltimore, MD, USA: Johns Hopkins University Press, 1989.
[42] Kreyszig, E., *Advanced Engineering Mathematics*, 8th ed. New York, NY, USA: John Wiley & Sons, Inc., 1999.
[43] Kaufman, L., "Some thoughts on the QZ algorithm for solving the generalized eigenvalue problem," *ACM Transactions on Mathematical Software,* vol. 3, pp. 65-75, 1977.
[44] LAPACK, "LAPACK: SRC/dggevx.f Source File," 23 Feb. 2011; http://www.netlib.org/lapack/explore-html-3.3.0-2011-01-25/dggevx_8f_source.html
[45] The MathWorks Inc., "What algorithm does the EIG function use to compute eigenvalues?," 23 Feb. 2011; http://www.mathworks.com/support/solutions/en/data/1-16J7K/index.html?product=ML&solution=1-16J7K
[46] The MathWorks Inc., "Eigenvalues and eigenvectors - MATLAB," 28 Sept. 2010; http://www.mathworks.com/help/techdoc/ref/eig.html
[47] Lanczos, C., "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," *Journal of Research of the National Bureau of Standards,* vol. 45, pp. 255-282, 1950.
[48] Saad, Y., "Variations on Arnoldi's method for computing eigen elements of large unsymmetric matrices," *Linear Algebra and Its Applications,* vol. 34, pp. 269-95, 1980.
[49] Morgan, R. B., "On restarting the Arnoldi method for large nonsymmetric eigenvalue problems," *Mathematics of Computation,* vol. 65, pp. 1213-1230, 1996.
[50] Saad, Y., "Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems," *Mathematics of Computation,* vol. 42, pp. 567-88, 1984.
[51] Komzsik, L. and MSC.Software. (2001). *MSC.Nastran 2001 - Numerical Methods User's Guide*. Available: http://simcompanion.mscsoftware.com/resources/sites/MSC/content/meta/DOCUMENTATION/9000/DOC9186/~secure/numerical.pdf?token=uxdTHzt4T2RqP7XRk4C/Nrv7e9PJ20DNws9qUdUqVs3Cn9iibRfHmd7azamnPe3HLiyzFHQcFRVTitjlHvsXlx+XWMk9Axr9Dx/Pi+XflxkMyOa7eUFHr0q7Br69uQ+GlDydXXNzOalHzG2XfWEo4g==
[52] Dassault Systèmes. (2007). *Abaqus Version 6.7 Documentation*.
[53] Nour-Omid, B., Parlett, B. N., and Taylor, R. L., "LANCZOS VERSUS SUBSPACE ITERATION FOR SOLUTION OF EIGENVALUE PROBLEMS," *International Journal for Numerical Methods in Engineering,* vol. 19, pp. 859-871, 1983.

[54] Fischer, P., "Eigensolution of nonclassically damped structures by complex subspace iteration," *Computer Methods in Applied Mechanics and Engineering,* vol. 189, pp. 149-166, 2000.
[55] Najafi, H. S. and Ghazvini, H., "A modification on minimum restarting method in the Arnoldi algorithm for computing the eigenvalues of a nonsymmetric matrix," *Applied Mathematics and Computation (New York),* vol. 181, pp. 1455-1461, 2006.
[56] Dookhitram, K., Boojhawon, R., and Bhuruth, M., "A new method for accelerating Arnoldi algorithms for large scale eigenproblems," *Mathematics and Computers in Simulation,* vol. 80, pp. 387-401, 2009.
[57] Alvin, K. F. and Park, K. C., "Second-order structural identification procedure via state-space-based system identification," *Aiaa Journal,* vol. 32, pp. 397-406, 1994.

PAPER I

Bratland, M. and Rølvåg, T., "Modal Analysis of Lumped Flexible Active Systems (Part 1)," presented at the SIMS 2008: The 48th Scandinavian Conference on Simulation and Modeling, Oslo, Norway, 2008.

MODAL ANALYSIS OF LUMPED FLEXIBLE ACTIVE SYSTEMS (PART 1)

Magne Bratland, Terje Rølvåg

Norwegian University of Science and Technology,
Faculty of Engineering Science and Technology,
Trondheim, Norway

magne.bratland@ntnu.no (Magne Bratland)

Abstract

Simulation and prediction of eigen-frequencies and resonance problems for flexible structures is an important task in disciplines such as robotics and aerospace engineering. However, little effort seems to have been put into the problem dealing with modal analysis of mechatronic systems containing coupled flexible structures and control systems. When, for instance, designing a satellite tracking radar, it is crucial to be able to predict resonance in the radar system during normal working conditions. Resonance may lead to loss of satellite tracking accuracy and long term fatigue problems.

This paper addresses the theory of solving the eigenvalue problem for a simple one-degree-of-freedom system coupled with a single position feedback PD-controller. To test the theory, the nonlinear multi-disciplinary simulation software FEDEM has been used. This paper is planned as the first in a series of papers addressing modal analysis of active flexible multibody systems.

Keywords:
Modal analysis, eigenvalue problem, flexible multibody system, PD-controller.

1 Introduction

To optimize performance and reduce development costs of mechatronic products, it is very important to use virtual testing. During the later years, mechanical products have become increasingly complex and mechanical functionality has gradually been replaced by cheaper and smarter control (active) systems. Typical examples are active / adaptive car suspensions, cranes, robots, machining centers, airplanes and satellites.

Mechatronic systems are traditionally designed and tested in separate software systems since the underlying mathematics used to solve the subsystems are different. Control systems are often modeled as 1st order equation systems (state-space-formulation), while mechanical systems usually are modeled as 2nd order symmetrical equation systems. These subsystems are therefore traditionally solved decoupled by different equation solvers. This approach has several disadvantages:

- The subsystems become sub-optimized because the couplings between them are limited. Control systems are often modeled as lumped springs and dampers in the mechanical subsystem and mechanical components are simplified as lumped masses, inertias and amplifiers in the control subsystem. The couplings between them are established through iterations and interchanges of force and response variables. The performance of the combined mechatronic system can thus not be simulated and optimized with a satisfactory accuracy and efficiency.

- The mechanical system and the control system are mutually affected by each other. Changes in either of the systems will cause alterations in the other. This means that the two mathematical models must be updated separately, which is both time consuming and demands coordination and handling of different software versions between engineers from different departments.

- A decoupled model representation does not support calculations of eigenfrequencies and mode shapes (modal analysis), which give

engineers vital information about the overall performance of a mechatronic system.

A literature survey performed by the authors indicated that little effort has been put into the problem dealing with modal analysis of mechatronic systems containing flexible structures, like robots, cranes, suspension and aerospace systems. However, the topic has been discussed in some papers and reports. In [1] it is shown that when combining passive mechanical springs and active piezoelectric springs, the total stiffness of the system is a sum of the stiffness from each of the springs, as can be expected based on basic theory of dynamics. In [2] it is shown that actuators can be controlled to act like virtual passive mechanical spring-damper elements using a velocity feedback PI-controller. In [3] it is shown that in contact motion force control, both the gain from a controller and the stiffness of the structure influences the natural frequency of the system. In [4] it is mentioned that a position feedback PD-controller is physically equivalent to a virtual spring and damper whose reference position is moving with a desired velocity.

This paper focuses on eigenfrequency analysis for a mechanical system with one degree of freedom, combined with a position feedback PD-controller. First, a basic description of the PD-controller is given. Next, different variants of the one-degree-of-freedom system combining the PD-controller and the mechanical system are described, and an equation for the eigenfrequency of these systems is given. Finally, results derived from the eigenfrequency equation for a total of six different scenarios are compared to experimental tests performed in the nonlinear multi-disciplinary simulation software FEDEM [5].

2 The PD-controller

Fig. 1 shows a simple block diagram used for describing a single-input single-output (SISO) feed-back control system:

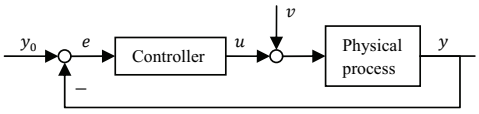


Fig. 1 Block diagram for a single-input single output (SISO) feedback control system.

The whole idea behind a control system is to manipulate a physical process to behave in a certain desired way. y_0 is the reference value for a parameter in the physical process and represents how this parameter should behave; y is the measured value for the same parameter and represents how this parameter actually is behaving. In a steady-state process, the aim is to keep the process as stable as possible, suppressing the disturbances v acting on the process as effectively as possible.

As shown in the block diagram, the difference e between the reference value y_0 and a measured value y is given by:

$$e = y_0 - y \tag{1}$$

e is somehow manipulated in the controller, and out comes a controller value u. This controller value is added with the disturbance v on the physical process. When combined, u and v make up the input value for the parameter in the physical process. The output value from the physical process is the measured value y, which is then compared to the reference value y_0, and the loop repeats itself.

The function of the controller is to manipulate the physical process so that it behaves in the most satisfactory way. One common approach to achieving this goal is to construct the controller with a combination of a proportional part (P), an integral part (I) and a derivative part (D). Based on the controller's incoming value $e(t)$, the outgoing controller value $u(t)$ for each of the three parts is, respectively:

$$u_P(t) = K_p e(t) \tag{2}$$

$$u_I(t) = K_i \int_0^t e(\tau)d\tau \tag{3}$$

$$u_D(t) = K_d \frac{de(t)}{dt} \tag{4}$$

where K_p, K_i and K_d are the proportional, integral and derivative gains, respectively.

If combined, the different parts give the following equation for the outgoing controller value $u(t)$:

$$u_{PID}(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{5}$$

If the integral gain K_i and derivative gain K_d are given by:

$$K_i = \frac{K_p}{T_i} \tag{6}$$

$$K_d = K_p T_d \tag{7}$$

then Eq. (5) can be written as:

$$u_{PID}(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau)d\tau + T_d \frac{de(t)}{dt} \right) \tag{8}$$

The outgoing controller value $u(t)$ for a PD-controller, which only has a proportional and a derivative part, is:

$$u_{PD}(t) = K_p e(t) + K_d \frac{de(t)}{dt} \qquad (9)$$

3 System with one degree of freedom

Fig. 2 illustrates three linear systems with one degree of freedom: a passive, an active and a coupled system. The passive system in Fig. 2 (a) is a mass-spring-damper system. The active system in Fig. 2 (b) contains only a mass and a controller. The coupled system in Fig. 2 (c) is a combination of the active and the passive system.

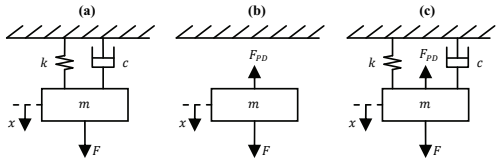


Fig. 2 Systems with one degree of freedom; (a) passive system, (b) active system, (c) coupled system.

All of the systems shown in Fig. 2 have the same degree of freedom: the position x of the mass m. Since these systems only have one degree of freedom, they will only have one eigenfrequency with one corresponding mode shape (oscillation).

3.1 Passive system

Fig. 3 illustrates a passive system with one degree of freedom. This is a system consisting of mechanical parts only.

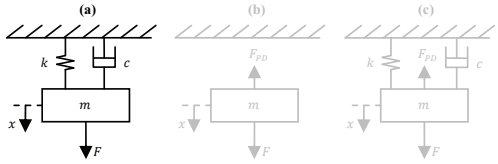


Fig. 3 Passive system with one degree of freedom.

The dynamic equation of motion for the passive system is:

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = F(t) \qquad (10)$$

where m is the mass in kg, c is the damping coefficient in Ns/m, k is the spring stiffness in N/m and F is an external force in N acting on the system. x, $\dot{x}$ and $\ddot{x}$ are the position, velocity and acceleration of the mass, respectively.

$$\dot{x}(t) = \frac{dx(t)}{dt}, \ddot{x}(t) = \frac{d^2x(t)}{dt^2} \qquad (11)$$

When calculating the eigenvalue of the system, all the external forces F are set to zero. The dynamic equation for the system thus becomes:

$$m\ddot{x} + c\dot{x} + kx = 0 \qquad (12)$$

The eigenfrequency of an undamped system ($c = 0$) is then given by:

$$\omega_e = \sqrt{\frac{k}{m}} \qquad (13)$$

If the system is damped, the eigenfrequency is given by [6]:

$$\omega_{e,d} = \omega_e\sqrt{1-\zeta^2} = \sqrt{\frac{k}{m}} \cdot \sqrt{1-\zeta^2} \qquad (14)$$

where ζ is the damping ratio ($\zeta = \frac{c}{c_c} = \frac{c}{2\sqrt{km}}$).

3.2 Active system

Fig. 4 illustrates an active system with one degree of freedom. This system is almost identical to the passive system shown in Fig. 3. However, as a crucial difference, the spring and damper have been replaced by a control system, in this case a standard PD-controller.

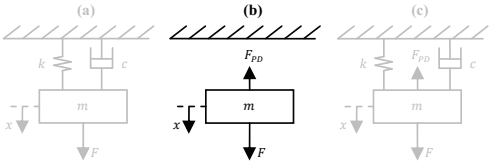


Fig. 4 Active system with one degree of freedom.

The dynamic equation of motion for the active system is:

$$m\ddot{x}(t) + 0\dot{x}(t) + 0x(t) = F(t) + F_{PD}(t) \qquad (15)$$

where F_{PD} is the force from the PD-controller acting on the mass m.

When calculating the eigenvalues of the system, the dynamic equation has now become:

$$m\ddot{x} + 0\dot{x} + 0x = 0 \qquad (16)$$

This implies that, from a classical mechanical point of view, the system does not have any eigenvalues since the elastic part of the dynamic equation is equal to zero. Thus the undamped eigenfrequency should be:

$$\omega_e = \sqrt{\frac{k}{m}} = \sqrt{\frac{0}{m}} = 0 \qquad (17)$$

However, when observing the active system, it is obvious that the system does have an eigenfrequency. A simple active system, in accordance to Fig. 4, was created and dynamically simulated in FEDEM, with no initial equilibrium iterations. The following figure illustrates the position of the mass m in the active system when under influence of a constant force F (equaling gravity). The mass was set to 1 kg and, for simplicity, the force was set to 10 N.

For the PD-controller, the proportional gain K_p was set to 100, the derivative gain K_d to 0 and the reference value y_0 representing the desired position of the mass to 0.

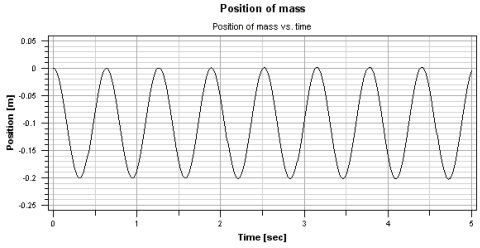


Fig. 5 Position of the mass in the active system with $K_p = 100$ and $K_d = 0$.

As Fig. 5 shows, the system clearly oscillates, even though the reference value y_0 is 0. This implies that the system does possess an eigenfrequency.

3.3 Coupled system

If the passive system in Fig. 3 is combined with the active system in Fig. 4 they form a coupled system, as shown in Fig. 6:

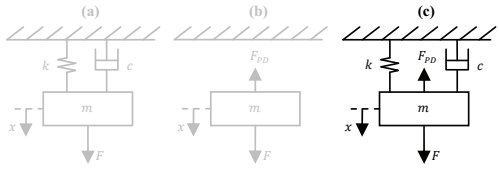


Fig. 6 Coupled system with one degree of freedom.

The dynamic equation of motion for the coupled system is:

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = F(t) + F_{PD}(t) \qquad (18)$$

Fig. 7 shows a block diagram of the coupled system in Fig. 6:

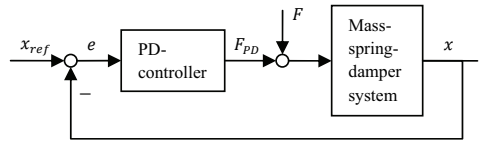


Fig. 7 Block diagram for a mass-spring-damper system and a PD-controller.

In the block diagram, x_{ref} is the reference position for the mass m. e is the difference between the reference position x_{ref} and the actual position x:

$$e = x_{ref} - x \qquad (19)$$

The PD-controller consists of a proportional part K_p and a derivative part $K_d \frac{d}{dt}$, as shown in Eq. (9). The outgoing value from the PD-controller is a force F_{PD}, which is added together with the force F acting on the mass object, and forms the basis for the incoming value to the mass-spring-damper system. The mass-spring-damper system is described by Eq. (18); its outgoing value is the position x of the mass m.

All of the forces acting on the system can be gathered into an equation containing the internal forces of the mechanical system on the left side and the external and controller forces on the right side:

$$F_m + F_c + F_k = F + F_P + F_D \qquad (20)$$

where F_m is the inertia force, F_c is the damping force, F_k is the elastic force, F is the external force, F_P is the force from the proportional part of the controller and F_D is the force from the derivative part of the controller. Eq. (20) can be rewritten to:

$$m\ddot{x} + c\dot{x} + kx = F + K_p e + K_d \dot{e} \qquad (21)$$

where $\dot{e}$ follows the same notation as given in Eq. (11). If Eq. (19) is inserted into Eq. (21), it can now be written as:

$$m\ddot{x} + c\dot{x} + kx = F + K_p(x_{ref} - x) + K_d(\dot{x}_{ref} - \dot{x}) \qquad (22)$$

which is equal to:

$$m\ddot{x} + c\dot{x} + kx + K_p x + K_d \dot{x} = F + K_p x_{ref} + K_d \dot{x}_{ref} \qquad (23)$$

or:

$$m\ddot{x} + (c + K_d)\dot{x} + (k + K_p)x = F + K_p x_{ref} + K_d \dot{x}_{ref} \qquad (24)$$

When calculating the eigenvalue of the system, the external force F and the reference position x_{ref} is set to zero, giving the following equation:

$$m\ddot{x} + (c + K_d)\dot{x} + \left(k + K_p\right)x = 0 \qquad (25)$$

The undamped and damped eigenfrequencies for the coupled system now becomes, respectively:

$$\omega_e = \sqrt{\frac{\left(k + K_p\right)}{m}} \qquad (26)$$

$$\omega_{e,d} = \sqrt{\frac{\left(k + K_p\right)}{m}} \cdot \sqrt{1 - \zeta^2} \qquad (27)$$

where the damping ratio ζ now has become:

$$\zeta = \frac{(c + K_d)}{2\left(\sqrt{\left(k + K_p\right)m}\right)} \qquad (28)$$

Eq. (26) and Eq. (28) correspond with formulas for effective natural frequency and effective damping ratio given in [6].

3.4 Experimental results

To verify the theoretical results derived above, a simple test was created. The objective of the test was to see how the mass-spring-damper system and the PD-controller actually acted on the eigenfrequency of the coupled system. The testing environment was created in the nonlinear multidisciplinary simulation software FEDEM, rather than using actual physical equipment. A total of 6 different testing scenarios were created:

Undamped

I) Only a spring connecting the mass to the ground.
II) Only a P-controller connecting the mass to the ground.
III) A spring and a P-controller connecting the mass to the ground.

Damped

IV) A spring-damper system and P-controller connecting the mass to the ground.
V) A spring and a PD-controller connecting the mass to the ground.
VI) A spring-damper system and a PD-controller connecting the mass to the ground.

An FE-model consisting of the coupled system shown in Fig. 6 was created in FEDEM. The external force F was set to 10 N and the reference position x_{ref} to 0. The mass m was set to 1 kg, the spring stiffness k to 100 N/m, the damping coefficient c to 7.2 Ns/m, the proportional gain K_p to 44 N/m and the derivative gain K_d to 4.8 Ns/m. k and K_p, and also c and K_d, have deliberately been given different values, such that differences are easier to distinguish. With respect to Eq. (26), Eq. (27) and Eq. (28), this should give the eigenfrequencies listed in Tab. 1 (calculations are shown in the appendix).

Fig. 8 and Fig. 9 show one graph from each of the six scenarios. The graphs picture the velocity of the mass m versus the time. The reason for using the velocity of the mass rather than its position is that it makes it easier to see the period of the oscillation. One period is then where the velocity is zero for the second time. Since FEDEM uses a numerical algorithm to solve the dynamic equation of motion, the results are only accurate to a certain number of decimals. To balance between accuracy of results and simulation running time, the time increment in the simulations was set to 0.0005 seconds.

4 Discussion

The results presented in Fig. 8 and Fig. 9 show that the eigenfrequencies from the simulations corresponds perfectly to the pre-calculated eigenfrequencies for the six different scenarios. As these results imply, the proportional gain K_p and derivative gain K_d from the PD-controller influences the stiffness and damping properties of the mechanical system, respectively. So, when performing an eigenvalue analysis for a mechanism coupled with a PD-controller, the proportional and derivative gain from the PD-controller should somehow be added to the stiffness and damping properties of the system. One way of doing this is to add a virtual spring with spring stiffness k_P corresponding to K_p and a virtual damper with damping coefficient c_D corresponding to K_d to the mechanical model. Another approach is to establish and solve the eigenvalues using a set of coupled equations representing the mechatronic system (mechanical and control system). This approach will be developed and reported in later papers, with a mission to solve eigenfrequencies and mode shapes for active flexible systems.

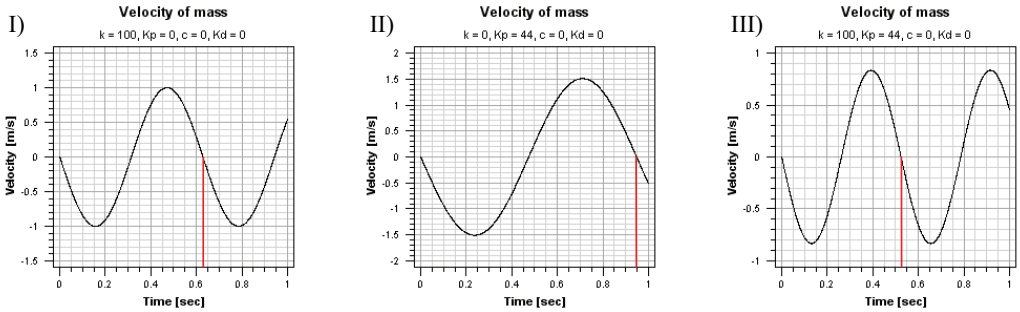
5 Conclusion

In this paper, a brief study of the eigenfrequencies of an active system containing a mass-spring-damper system and a position feedback PD-controller has been conducted. Theory for modal analysis of such a system has been derived and presented. The theory has been verified by experiments conducted in the nonlinear multidisciplinary simulation software FEDEM, showing that the derived theory concur with the experimental results.

Tab. 1 Properties, damping ratio and eigenfrequency for the six different scenarios, based on analytical calculations (which are shown in the appendix).

Scenario	k [N/m]	K_p [N/m]	c [Ns/m]	K_d [Ns/m]	Damping ratio	Eigenfrequency
Undamped						
I)	100	0	0	0	$\zeta = 0$	$\omega_e = 1.59$ Hz
II)	0	44	0	0	$\zeta = 0$	$\omega_e = 1.06$ Hz
III)	100	44	0	0	$\zeta = 0$	$\omega_e = 1.91$ Hz
Damped						
IV)	100	44	7.2	0	$\zeta = 0.3$	$\omega_{e,d} = 1.82$ Hz
V)	100	44	0	4.8	$\zeta = 0.2$	$\omega_{e,d} = 1.87$ Hz
VI)	100	44	7.2	4.8	$\zeta = 0.5$	$\omega_{e,d} = 1.65$ Hz

Undamped



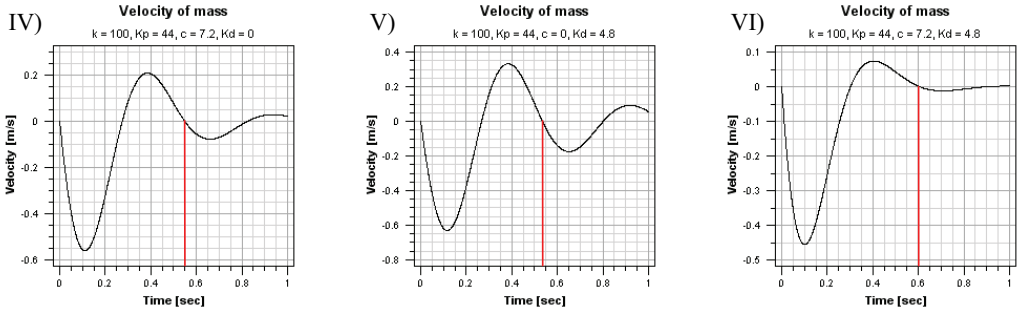
$t = 0.6285$ sec
$\Rightarrow \omega_e = 1.59$ Hz

$t = 0.9475$ sec
$\Rightarrow \omega_e = 1.06$ Hz

$t = 0.5240$ sec
$\Rightarrow \omega_e = 1.91$ Hz

Fig. 8 Results from FEDEM for the three undamped scenarios.

Damped



$t = 0.5490$ sec
$\Rightarrow \omega_{e,d} = 1.82$ Hz

$t = 0.5345$ sec
$\Rightarrow \omega_{e,d} = 1.87$ Hz

$t = 0.6045$ sec
$\Rightarrow \omega_{e,d} = 1.65$ Hz

Fig. 9 Results from FEDEM for the three damped scenarios.

6 Acknowledgements

The authors would like to acknowledge the financial support from The Research Council of Norway and the LPD project.

7 References

[1] A. Baz, "Active control of periodic structures," *Transactions of the ASME. Journal of Vibration and Acoustics,* vol. 123, pp. 472-9, 2001.

[2] W. Bernzen, "Active vibration control of flexible robots using virtual spring-damper systems," *Journal of Intelligent and Robotic Systems: Theory and Applications,* vol. 24, pp. 69-88, 1999.

[3] K. Ohnishi, M. Shibata, and T. Murakami, "Motion control for advanced mechatronics," *IEEE/ASME Transactions on Mechatronics,* vol. 1, pp. 56-67, 1996.

[4] J.-H. Ryu, D.-S. Kwon, and B. Hannaford, "Stability guaranteed control: Time domain passivity approach," *IEEE Transactions on Control Systems Technology,* vol. 12, pp. 860-868, 2004.

[5] O. I. Sivertsen and A. O. Waloen, "Non-Linear Finite Element Formulations for Dynamic Analysis of Mechanisms with Elastic Components," Washington, DC, USA, 1982, p. 7.

[6] W. J. Palm, *Mechanical Vibration*. Hoboken, N.J., USA: John Wiley & Sons, Inc., 2007.

8 Appendix

Below are shown the calculations for the eigenfrequencies in the six different experiment scenarios. The formulas used are: Eq. (26) for the undamped eigen-frequencies, Eq. (27) for the damped eigenfrequencies and Eq. (28) for the damping ratios.

Undamped

I) $\omega_e = \sqrt{\frac{(k+K_p)}{m}} = \sqrt{\frac{(100+0)}{1}} =$
$10\ \text{rad/sec} = 1.59\ \text{Hz}$

II) $\omega_e = \sqrt{\frac{(k+K_p)}{m}} = \sqrt{\frac{(0+44)}{1}} =$
$6.63\ \text{rad/sec} = 1.06\ \text{Hz}$

III) $\omega_e = \sqrt{\frac{(k+K_p)}{m}} = \sqrt{\frac{(100+44)}{1}} =$
$12\ \text{rad/sec} = 1.91\ \text{Hz}$

Damped

IV) $\zeta = \frac{(c+K_d)}{2\left(\sqrt{(k+K_p)m}\right)} = \frac{(7.2+0)}{2\left(\sqrt{(100+44)1}\right)} = 0.3$
$\Rightarrow \omega_{e,d} = \sqrt{\frac{(k+K_p)}{m}} \cdot \sqrt{1-\zeta^2} =$
$\sqrt{\frac{(100+44)}{1}} \cdot \sqrt{1-0.3^2} =$
$11.44\ \text{rad/sec} = 1.82\ \text{Hz}$

V) $\zeta = \frac{(c+K_d)}{2\left(\sqrt{(k+K_p)m}\right)} = \frac{(0+4.8)}{2\left(\sqrt{(100+44)1}\right)} = 0.2$
$\Rightarrow \omega_{e,d} = \sqrt{\frac{(k+K_p)}{m}} \cdot \sqrt{1-\zeta^2} =$
$\sqrt{\frac{(100+44)}{1}} \cdot \sqrt{1-0.2^2} =$
$11.76\ \text{rad/sec} = 1.87\ \text{Hz}$

VI) $\zeta = \frac{(c+K_d)}{2\left(\sqrt{(k+K_p)m}\right)} = \frac{(7.2+4.8)}{2\left(\sqrt{(100+44)1}\right)} = 0.5$
$\Rightarrow \omega_{e,d} = \sqrt{\frac{(k+K_p)}{m}} \cdot \sqrt{1-\zeta^2} =$
$\sqrt{\frac{(100+44)}{1}} \cdot \sqrt{1-0.5^2} =$
$10.39\ \text{rad/sec} = 1.65\ \text{Hz}$

PAPER II

Bratland, M., Haugen, B., and Rølvåg, T., "Modal analysis of active flexible multibody systems," Computers and Structures, vol. 89, pp. 750-761, 2011.

Computers and Structures 89 (2011) 750–761

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

Modal analysis of active flexible multibody systems

Magne Bratland *, Bjørn Haugen, Terje Rølvåg

Department of Engineering Design and Materials, Norwegian University of Science and Technology, Richard Birkelands veg 2B, N-7491 Trondheim, Norway

ARTICLE INFO

Article history:
Received 14 May 2010
Accepted 14 February 2011

Keywords:
Modal analysis
Closed-loop eigenvalue problem
Flexible multibody system
PID controller

ABSTRACT

When performing modal analyses of active flexible multibody systems, both controller effects and flexible body dynamics should be included in a multidisciplinary system model. This paper deals with the theory of solving the closed-loop eigenvalue problem for active flexible multibody systems with multiple-input multiple-output proportional-integral-derivative (PID) type feedback controllers and multiple degrees of freedom finite element models. Modal analyses are performed on both a simple and complex active flexible multibody system in order to illustrate the difference between current modal analysis method for such systems and the proposed theory derived in this paper.

1. Introduction

Modal analysis and dynamic simulation of active mechanisms are a multidisciplinary challenge. The dynamic performance of such products is strongly dependent on an optimal interaction between the controllers and mechanical components. An important tool in the optimization of these products is modal analysis, which predicts natural frequencies, mode shapes and damping ratios, often referred to as modal parameters, for the active system.

The challenge is to include all properties in a multidisciplinary system model that is appropriate for modal analysis. One obstacle is that the basic formulations and solvers for control systems and mechanical systems are different. Control systems are often modeled as 1st order equation systems (state-space-formulation) ($\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$, $\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$), see for example [1–3], while mechanical systems are usually modeled as 2nd order symmetrical equation systems ($\mathbf{M}\ddot{\mathbf{r}} + \mathbf{C}\dot{\mathbf{r}} + \mathbf{K}\mathbf{r} = \mathbf{F}$), see for example [4–7]. In addition, for mechanisms involving large displacements and other nonlinearities, the modal parameters are time dependent due to time varying mass, damping and stiffness matrices. To the best of the authors' knowledge, a closed-loop modal analysis is not implemented in any software package today. However, modal analysis of active mechanisms can be approached from two different discipline strategies.

Control system software, such as MATLAB and Simulink, usually support both controller design and control system simulation. The mechanical systems can be modeled with rigid bodies, lumped masses, inertias, springs and dampers or analytical equations. Using this approach, the closed-loop eigenvalues and eigenvectors can be predicted as shown in for example [8–10]. From a mechanical engineers point of view, the flexible body dynamics are by this approach predicted by very simplified models. This may work well if the dynamics due to flexible bodies can be neglected. If not, control and observation spillover can cause a reduction in dynamic performance and may lead to system instability. Unmodeled flexible body dynamics also make modal analysis and controller synthesis unreliable. Thus, modal analysis should be performed in finite element (FE) based software systems.

The finite element method (FEM) is one of the best tools for performing modal analysis of flexible structures due to its multidisciplinary modeling capabilities and ease of use. FE software can also be interfaced with control system software for dynamic time simulation analyses of active systems [7]. Feedback type controllers will typically calculate loads applied to the FE structure based on feedback measurements of the system. For that reason, the controller is comparable to a "black box" or unknown function, as seen from the mechanical engineer's point of view. This approach works well in time domain analyses when the controller drives the FE model with applied loads based on the given controller algorithms. Even so, a major problem occurs in modal analyses of the closed-loop system. In free vibration analyses, all loads are set to zero, which decouples the controller and mechanical model. As a result, the FE model becomes singular in all controlled FE degrees of freedom (DOFs).

Probably the most common approach by mechanical engineers when performing modal analyses of active mechanisms is to introduce additional boundary conditions for the system DOFs affected by controllers, thereby omitting the flexibility in the different joints of the mechanism by making the joints rigid at relevant positions. The greatest flaw in this approach is that the eigenfrequencies and eigenmodes for an active mechanism are not the same

* Corresponding author.
E-mail address: magne.bratland@ntnu.no (M. Bratland).

0045-7949/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compstruc.2011.02.010

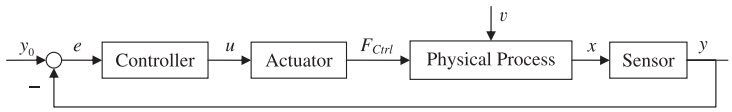


Fig. 1. Block diagram for a SISO feedback control system.

as for a purely mechanical system, as shown in for example [2,5,11]. Several other sources exist that support this statement, either directly or indirectly, see for example [12–18].

Another common, though inaccurate, solution to this problem is to represent the controller effects by virtual springs, dampers and inertias in the mechanical model [11]. For example, Sharon et al. [12] have stated that: "If an ideal actuator and corresponding ideal sensor are acting on the same point (collocated control) in a purely inertial system, then: (1) Negative position feedback is equivalent to a spring action. (2) Negative velocity feedback is equivalent to a damping action. (3) Negative force feedback is equivalent to decreasing inertia. (4) Positive force feedback is equivalent to increasing inertia." Bernzen [14] has demonstrated that actuators can be controlled to act like virtual passive mechanical spring-damper elements using a velocity feedback PI controller. Ryu et al. [16] mentioned that a position feedback PD controller is physically equivalent to a virtual spring and damper whose reference position is moving at a desired velocity. Nonetheless, this approach is only applicable for simple control systems in which the mechanical engineer knows how to transform the controller into an equivalent mechanical model. The engineer also has to update two system models: one for modal analysis and one for time domain dynamic simulation.

This paper reveals how a controller influences mechanical system properties based on the type of controller and the type of feedback. The results indicate how a finite element model of an active mechanism can be modified by equivalent mechanical properties that represent the controller. The objective is to improve the accuracy of closed-loop modal analyses of flexible mechanisms driven by controllers. First, a basic description of the interaction between a single-input single-output (SISO) controller and a single degree of freedom (SDOF) mechanical system is given. Next, the interaction theory is expanded to include multiple-input multiple-output (MIMO) controllers and multiple degrees of freedom (MDOF) mechanical systems, putting an emphasis on the gradients between actuator exerted forces and sensor readings. The controllers are limited to be of type proportional-integral-derivative (PID), which are the most common type of controllers in use today [19,20]. Finally, modal analyses are performed on both a two-joint mechanism and a satellite tracking antenna, illustrating the limitations by the current methods and how the derived theory can be used to improve modal analyses. In both cases, the results of the modal analyses are compared to eigenfrequency estimations of the systems derived using the fast Fourier transform (FFT) algorithm [21] on time series of various system responses, a method that is frequently used in experimental modal analysis, see for example [22–25]. This latter method is here used only as a reference, and is in itself an area of ongoing research. For the sake of convenience, all simulations and experiments in this paper are performed on virtual models in FEDEM[1]; however, the theory derived in this paper is not dependent on any particular software system.

[1] FEDEM (Finite Element in Dynamics of Elastic Mechanisms) simulation software is a multibody dynamics package distributed by Fedem Technology AS. It is based on the finite element method and uses model reduction techniques to effectively perform nonlinear time domain dynamic simulations of active flexible multibody systems [7,26].

2. Interaction between mechanical systems and controllers

2.1. Single degree of freedom (SDOF) mechanical system with single-input single-output (SISO) controller

Fig. 1 shows a simple block diagram used for describing a SISO feedback control system.

In Fig. 1, y_0 is the reference variable, y is the measured variable and e is the difference between y_0 and y. u is the controller output and F_{Ctrl} is a force from the controller exerted by an actuator. x is the state variable from the physical process, either position r, velocity $\dot{r}$ or acceleration $\ddot{r}$, while v is the disturbance on the physical process.

One view of the control system is to isolate the control elements from the physical process. The control elements then principally contain three parts: a sensor, an actuator and a controller containing the various controller elements, as shown in Fig. 2.

The sensor and actuator are the interfaces to the mechanical system; the sensor is the input to the controller and the actuator is the output from the controller. Usually, the sensor measures state variables, that is, position, velocity or acceleration, in the mechanism's relevant DOFs. The actuator can be simplified to a set of forces acting on the mechanism. The equation of motion for an SDOF mechanical system with a SISO controller can then be given as:

$$m\ddot{r}(t) + c\dot{r}(t) + kr(t) = F_{App}(t) + F_{Ctrl}(t) \tag{1}$$

where m is the mass, c is the damping and k is the stiffness. r is the displacement of the mass m with respect to time; $\dot{r}$ and $\ddot{r}$ are the 1st and 2nd time derivatives of r, respectively, that is, velocity and acceleration of the mass m. F_{App} is the applied mechanical force and F_{Ctrl} is the force from the controller. This is in accordance with equations found in [10].

For a feedback PID-type controller, the controller output u is given by:

$$u_{PID}(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{d}{dt} e(t) \tag{2}$$

where K_p is the proportional gain, K_d is the derivative gain and K_i is the integral gain from the controller. It is implied that the parameters m, c, k, K_p, K_i and K_d are positive. As shown in Fig. 2, the effects by the control elements on the mechanical system can be given as:

$$\frac{\partial F_{Ctrl}}{\partial x} = \frac{\partial F_{Ctrl}}{\partial u}\frac{\partial u}{\partial y}\frac{\partial y}{\partial x} \quad \text{or} \quad dF_{Ctrl} = G_{Act}G_{Ctrl}G_{Sens}\,dx \tag{3}$$

where G_{Act} is the actuator gradient, G_{Ctrl} is the controller gradient and G_{Sens} is the sensor gradient.

Combining Eqs. (1) and (3) yields an equation of motion for the free vibration of a SDOF mechanical system with a SISO controller as:

$$m\ddot{r}(t) + c\dot{r}(t) + kr(t) + G_{Act}G_{Ctrl}G_{Sens}x(t) = 0 \tag{4}$$

For a position feedback controller, $x(t) = r(t)$. Combining Eqs. (2) and (4), and setting G_{Act} and G_{Sens} to 1, yields an equation of motion for the free vibration of a SDOF mechanical system with a position feedback PID controller as:

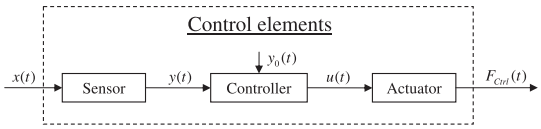


Fig. 2. Control elements.

$$m\ddot{r}(t) + (c + K_d)\dot{r}(t) + (k + K_p)r(t) + K_i \int r(t)\,dt = 0 \tag{5}$$

Assuming a solution of the form $r(t) = e^{st}$ for Eq. (5) gives a characteristic equation:

$$ms^2 + (c + K_d)s + (k + K_p) + K_i s^{-1} = 0 \tag{6}$$

For a velocity feedback controller, $x(t) = \dot{r}(t)$, thus yielding the equation of motion for the free vibration and the characteristic equation for a SDOF mechanical system with a velocity feedback PID controller as Eqs. (7) and (8), respectively:

$$(m + K_d)\ddot{r}(t) + (c + K_p)\dot{r}(t) + (k + K_i)r(t) = 0 \tag{7}$$

$$(m + K_d)s^2 + (c + K_p)s + (k + K_i) = 0 \tag{8}$$

Similarly, for an acceleration feedback controller, $x(t) = \ddot{r}(t)$, the equation of motion for the free vibration and characteristic equation for an SDOF mechanical system with an acceleration feedback PID controller are given by Eqs. (9) and (10), respectively:

$$K_d \dddot{r}(t) + (m + K_p)\ddot{r}(t) + (c + K_i)\dot{r}(t) + kr(t) = 0 \tag{9}$$

$$K_d s^3 + (m + K_p)s^2 + (c + K_i)s + k = 0 \tag{10}$$

Solving Eqs. (6), (8) or Eq. (10) with respect to s gives s as either a real or complex number. As shown in [5], for a purely mechanical system, s can be written as:

$$s = -\zeta\omega_n \pm i\omega_n\sqrt{1 - \zeta^2} = -\zeta\omega_n \pm i\omega_d \tag{11}$$

where ω_n is the undamped natural frequency, ω_d is the damped natural frequency and ζ is the damping ratio. The imaginary part is the frequency of oscillation, and the real part is the constant in the exponent of the oscillation amplitude envelope. Thus, if the controller gains K_p, K_i and K_d are known, they can be represented by equivalent mechanical properties in the system equation, thereby taking their effect on the modal parameters of the system into consideration when solving the eigenvalue problem for the active system. However, their relation to the sensor input has to be taken into consideration. As shown by Eqs. (6), (8), and (10), if the sensor measures position, K_p affects the stiffness of the system, whereas K_d and K_i affect the damping of the system. If the sensor measures velocity, K_p affects the damping of the system, K_d affects the inertia and K_i the stiffness of the system. If the sensor measures acceleration, K_p should affect the inertia and K_i the damping of the system. In contrast, K_d would be proportional to the derivative of the acceleration with respect to time, $\dot{r}$, often referred to as jerk or jolt. This pattern is illustrated in Fig. 3 and summarized in Tables 1–3.

If the real part of the complex number s is positive, the oscillation is growing, which means that the system is unstable. This can occur for both the position and acceleration feedback PID controllers because of the K_i and K_d terms, respectively.

2.2. Multiple degrees of freedom (MDOF) mechanical system with multiple-input multiple-output (MIMO) controller

As for the SDOF SISO system gradients in Eq. (3), the gradients for a MDOF MIMO system can be written as:

$$dF_{Ctrl_i} = \frac{\partial F_{Ctrl_i}}{\partial u_j}\frac{\partial u_j}{\partial y_k}\frac{\partial y_k}{\partial x_l}\,dx_l = G_{Act_{ij}}G_{Ctrl_{jk}}G_{Sens_{kl}}\,dx_l \tag{12}$$

or, on matrix form as:

$$d\mathbf{F}_{Ctrl} = \frac{\partial \mathbf{F}_{Ctrl}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\,d\mathbf{x} = \mathbf{G}_{Act}\mathbf{G}_{Ctrl}\mathbf{G}_{Sens}\,d\mathbf{x} \tag{13}$$

Eq. (4) describes the equation of motion for the free vibration of a SDOF mechanical system with a SISO controller. Similarly, the equation for the free vibration of a mechatronic system with n degrees of freedoms can be written as:

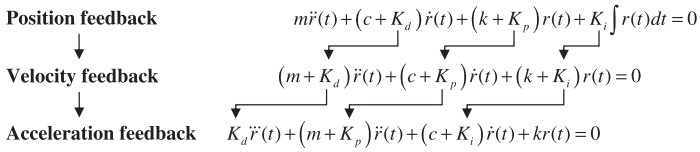
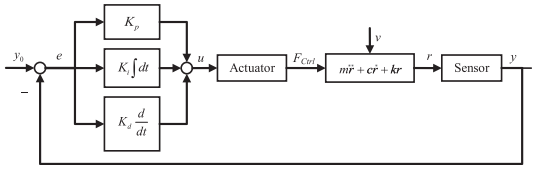


Fig. 3. Pattern for the addition of controller gains into the system equation based on type of sensor input.

Table 1
Control system with position feedback PID controller.

Position feedback

Block diagram:



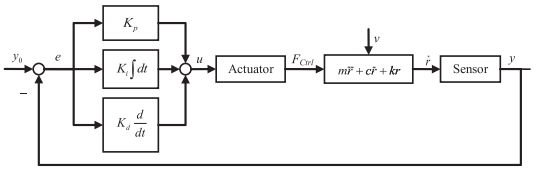
Equations:
$m\ddot{r}(t) + (c + K_d)\dot{r}(t) + (k + K_p)r(t) + K_i\int r(t)dt = 0$
$ms^2 + (c + K_d)s + (k + K_p) + K_i s^{-1} = 0$

Controller gain contribution on mechanical system
An increase in K_p will increase the system stiffness
An increase in K_i will decrease the system damping
An increase in K_d will increase the system damping

Table 2
Control system with velocity feedback PID controller.

Velocity feedback

Block diagram:



Equations:

$(m + K_d)\ddot{r}(t) + (c + K_p)\dot{r}(t) + (k + K_i)r(t) = 0$

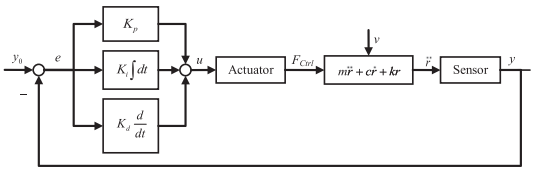
$(m + K_d)s^2 + (c + K_p)s + (k + K_i) = 0$

Controller gain contribution on mechanical system
An increase in K_p will increase the system damping
An increase in K_i will increase the system stiffness
An increase in K_d will increase the system mass

Table 3
Control system with acceleration feedback PID controller.

Acceleration feedback

Block diagram:



Equations:

$K_d \dddot{r}(t) + (m + K_p)\ddot{r}(t) + (c + K_i)\dot{r}(t) + kr(t) = 0$

$K_d s^3 + (m + K_p)s^2 + (c + K_i)s + k = 0$

Controller gain contribution on mechanical system
An increase in K_p will increase the system mass
An increase in K_i will increase the system damping
An increase in K_d will decrease the system damping

$$\mathbf{M}\ddot{\mathbf{r}}(t) + \mathbf{C}\dot{\mathbf{r}}(t) + \mathbf{K}\mathbf{r}(t) + \mathbf{G}_{Act}\mathbf{G}_{Ctrl}\mathbf{G}_{Sens}\mathbf{x}(t) = \mathbf{0} \tag{14}$$

where $\mathbf{M}$ is the $n \times n$ mass matrix, $\mathbf{C}$ is the $n \times n$ damping matrix, $\mathbf{K}$ is the $n \times n$ stiffness matrix and $\mathbf{r}$, $\dot{\mathbf{r}}$ and $\ddot{\mathbf{r}}$ are the $n \times 1$ position, velocity and acceleration vectors, respectively. $\mathbf{x}$ is a vector of the system state variables, that is, position, velocity and acceleration. $\mathbf{G}_{Act}$, $\mathbf{G}_{Ctrl}$ and $\mathbf{G}_{Sens}$ are the actuator gradient, controller gradient and sensor gradient matrices, respectively.

2.2.1. Actuator gradient matrix $\mathbf{G}_{Act}$

The actuator gradients describe the relationship between the controller forces $\mathbf{F}_{Ctrl}$ exerted by the actuator and the output signals $\mathbf{u}$ from the controller. The gradients of the controller force F_{Ctrl_i} with respect to controller output u_j can be written as:

$$d\mathbf{F}_{Ctrl} = \mathbf{G}_{Act}\, d\mathbf{u} \quad \text{or} \quad dF_{Ctrl_i} = \frac{\partial F_{Ctrl_i}}{\partial u_j}\, du_j = G_{Act_{ij}}\, du_j \tag{15}$$

Matrix $\mathbf{G}_{Act}$ has the dimensions $n_{F_{Ctrl}} \times n_u$ where $n_{F_{Ctrl}}$ is the number of controller forces and n_u is the number of controller outputs.

2.2.2. Controller gradient matrix $\mathbf{G}_{Ctrl}$

The controller gradients describe the relationship between the input variables $\mathbf{y}$ and output variables $\mathbf{u}$ both to and from the controller, respectively; that is, the various controller gains. The gradients of the controller output u_i with respect to the controller input y_j can be written as:

$$d\mathbf{u} = \mathbf{G}_{Ctrl}\, d\mathbf{y} \quad \text{or} \quad du_i = \frac{\partial u_i}{\partial y_j}\, dy_j = G_{Ctrl_{ij}}\, dy_j \tag{16}$$

Matrix $\mathbf{G}_{Ctrl}$ has the dimensions $n_u \times n_y$ where n_u is the number of controller outputs and n_y is the number of controller inputs.

2.2.3. Sensor gradient matrix $\mathbf{G}_{Sens}$

The sensor gradients describe the relationship between the controller input variables $\mathbf{y}$ and the system state variables $\mathbf{r}$, $\dot{\mathbf{r}}$ and $\ddot{\mathbf{r}}$ represented by the vector $\mathbf{x}$. $\mathbf{x}$ is given as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \\ \ddot{\mathbf{r}} \end{bmatrix} \tag{17}$$

Vector $\mathbf{x}$ has the dimensions $3n_r \times 1$ where n_r is the number of all system DOFs. Each sensor is limited to measure only one state variable in only one single system DOF or between two system DOFs. The gradients of the controller input y_i with respect to system DOF and state variable x_j can be written as:

$$d\mathbf{y} = \mathbf{G}_{Sens}\, d\mathbf{x} \quad \text{or} \quad dy_i = \frac{\partial y_i}{\partial x_j}\, dx_j = G_{Sens_{ij}}\, dx_j \tag{18}$$

Matrix $\mathbf{G}_{Sens}$ has the dimensions $n_y \times 3n_r$ where n_y is the number of controller inputs and n_r is the number of all system DOFs.

The matrix product $\mathbf{G}$ of the gradient matrices $\mathbf{G}_{Act}$, $\mathbf{G}_{Ctrl}$ and $\mathbf{G}_{Sens}$ has the dimensions $n_{F_{Ctrl}} \times 3n_r$. If $\mathbf{G}$ is to be used with $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$, it should be of the same dimensions, that is, $n_r \times n_r$. This can be done by pre-multiplying $\mathbf{G}$ with the topology matrix relating each controller force F_{Ctrl_i} with its respective system DOFs and then splitting the new $n_r \times 3n_r$ matrix product into 3 $n_r \times n_r$ matrices, $\mathbf{G}_{Pos}$, $\mathbf{G}_{Vel}$ and $\mathbf{G}_{Acc}$, one for each state variable $\mathbf{r}$, $\dot{\mathbf{r}}$ and $\ddot{\mathbf{r}}$. These new matrices $\mathbf{G}_{Pos}$, $\mathbf{G}_{Vel}$ and $\mathbf{G}_{Acc}$ can then be added to their respective system matrix, yielding the following equation system for the free vibration of a controlled mechanism:

$$(\mathbf{M} + \mathbf{G}_{Acc})\ddot{\mathbf{r}}(t) + (\mathbf{C} + \mathbf{G}_{Vel})\dot{\mathbf{r}}(t) + (\mathbf{K} + \mathbf{G}_{Pos})\mathbf{r}(t) = \mathbf{0} \tag{19}$$

Usually, **M**, **C** and **K** are symmetrical matrices. If the control system's sensors and actuators are collocated, meaning that they are sharing the same system DOFs, $\mathbf{G}_{Pos}$, $\mathbf{G}_{Vel}$ and $\mathbf{G}_{Acc}$ will also be symmetrical (diagonal). If the sensors and actuators are non-collocated, $\mathbf{G}_{Pos}$, $\mathbf{G}_{Vel}$ and $\mathbf{G}_{Acc}$ will be unsymmetrical. For symmetrical matrices, standard FE solvers can be used, while for unsymmetrical matrices, unsymmetrical matrix solvers need to be used. This work is limited to dealing with the former situation, that is, collocated sensors and actuators.

For lightly damped systems, the damped and undamped natural frequencies are approximately the same. This also holds true for systems with a relatively slow steady state error elimination, that is, a relatively low effect by the integral gain K_i for position feedback controllers. With this type of case, it is sufficient to solve the eigenvalue problem for the undamped active system. If the system is heavily damped, whether due to mechanical or controller parameters, the actual and undamped eigenfrequencies are not the same, as shown in Eq. (11). In such a case, the eigenvalue problem for the damped active system should be solved. If the steady state error elimination of the system is relatively quick, which means a relatively high effect by the integral gain K_i for position feedback controllers, the system can become unstable. This would be similar to experiencing an inverted or negative damping since the constant in the exponent of the oscillation amplitude envelope would be positive, thus yielding a growing oscillation. As for heavily damped systems, the actual and undamped eigenfrequencies in highly unstable systems are not the same. In such a case, the eigenvalue problem for the unstable active system should be solved. This work is limited to the former situation, that is, lightly damped systems with relatively slow steady state error elimination.

The generalized eigenvalue problem for the symmetrical undamped case can then be given as:

$$(\mathbf{K} + \mathbf{G}_{Pos})\mathbf{\Phi} = (\mathbf{M} + \mathbf{G}_{Acc})\mathbf{\Phi}\mathbf{\Lambda} \tag{20}$$

where $\mathbf{\Lambda}$ is a diagonal matrix of the generalized eigenvalues and $\mathbf{\Phi}$ is a full matrix whose columns are the corresponding eigenvectors.

3. Numerical examples

To verify the theory derived in the previous paragraphs and illustrate the difference in accuracy of the various modal analysis approaches mentioned in the introduction, two different examples were chosen: one simple and one complex. The simple example consisted of a two-joint mechanism, and the complex example was a satellite tracking antenna. The intention behind the former example was to have a simple and verifiable model for verification of the derived theory, as well as to highlight some of the limitations in the different modal analysis approaches mentioned in the introduction. The objective of the latter example was to further illustrate some of the limitations in the various modal analysis approaches. Since the former example is fairly simple, its utilitarian value may be lost, so the latter example was chosen to compensate for this. The antenna in the latter example was intended as an illustrative example only; therefore, it is deliberately not an exact replication of a real physical product. Because of this, the parameter values used for the antenna are not of importance in this context.

3.1. Modal analysis of a two-joint mechanism

Fig. 4 shows a schematic illustration of a two-joint mechanism. The two-joint mechanism consisted of two highly flexible beams and two rotational joints, each governed by an angular position feedback PD controller. The two joint DOFs were angular rotations about the global y-axis, and were named θ_1 and θ_2, respectively. The beams were created by using aluminum properties with a density of $\rho = 2794\ \text{kg/m}^3$ and a Young's modulus of $E = 65.7 \times 10^9$ Pa. The dimensions of the beams were 18 mm in width, 0.5 mm in thickness and 150 mm in length. An FE model was created for each beam using 34 4-node quadrilateral 2D elements. The entire mechanism was assembled in FEDEM, as shown in Fig. 5.

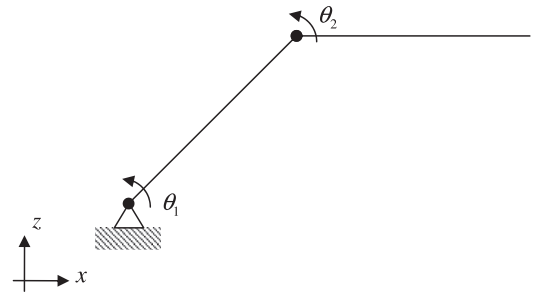


Fig. 4. Two-joint mechanism.

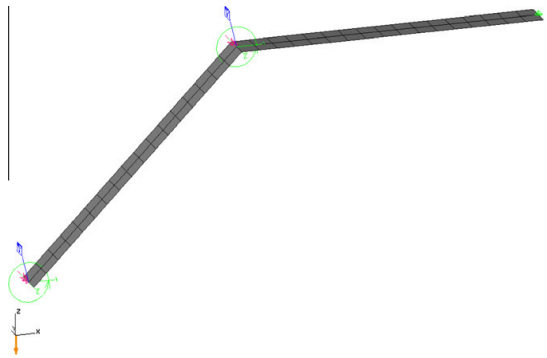


Fig. 5. FE model of the two-joint mechanism.

Since a position feedback PD controller is physically equivalent to a virtual spring and damper whose reference position is moving with a desired velocity, as mentioned in [16], the controllers were represented for simplicity by virtual springs and dampers in the mechanism joints. The values for each virtual spring and damper in the joint DOFs were, respectively, $\theta_1 : k_{\theta_1} = 0.1$ Nm/rad, $c_{\theta_1} = 0.001$ Nms/rad; $\theta_2 : k_{\theta_2} = 0.1$ Nm/rad, $c_{\theta_2} = 0$ Nms/rad.

To perform modal analyses of the mechanism, three different approaches were used:

1. Rigid body mechanism with controller effects: Controller effects were included, but the bodies of the mechanism were assumed to be rigid, leaving only the two joint DOFs θ_1 and θ_2 remaining (typical control theory approach).
2. Flexible body mechanism with rigid joint constraints: The flexibility of the bodies of the mechanism were included, but the controllers were replaced by rigid boundary conditions, meaning that $\Delta\theta_1$ and $\Delta\theta_2$ are set to zero, leaving only structural DOFs represented by FE models of the various bodies remaining (typical finite element analysis (FEA) practice).
3. Flexible body mechanism with controller effects: Both flexibility of the mechanism bodies and the controller effects were included, thereby ensuring that all system DOFs were kept intact (new multidisciplinary approach). Eq. (20) formed the basis for this modal analysis approach.

Because of the rigid body simplification made by the control theory approach, only the two eigenfrequencies involving angular rotations in the joint DOFs θ_1 and θ_2 can be retrieved for this approach; consequently, all work in this paper involving the two-joint mechanism is limited to these two eigenfrequencies and their corresponding mode shapes.

Since all simulations and modal analyses of the two-joint mechanism were performed in FEDEM, which is a flexible multibody simulation software, the flexible bodies had to be emulated as rigid for the control theory approach. This was achieved by scaling the stiffness of the structural parts by a factor of 100, thus emulating the structures as rigid.

To verify the accuracy of the modal analyses, a time simulation response of the mechanism when suddenly subjected to gravity was analyzed using the fast Fourier transform (FFT) algorithm. Initial modal analyses of the mechanism revealed that both eigenmodes of interest were observable at θ_2. As a result, the time response of the angular rotations about θ_2 was used. The time simulation ran for 10 s with a time increment of 0.001 s, giving a frequency sampling rate f_s of 1000 Hz and a frequency resolution of approximately 0.1 Hz. No windowing functions were used for the FFT, and thoughts regarding this choice are made in Section 3.1.1. A plot of the FFT is shown in Fig. 6.

A comparison of the estimated eigenfrequencies derived by the different modal analysis approaches and the estimated eigenfrequencies derived from the FFT are shown in Table 4. The mode shapes of interest for the various approaches are sketched in Figs. 7–9.

As can be seen in Table 4, there is a difference in the results from the three modal analysis approaches. All eigenfrequencies derived by the different modal analysis approaches are higher than those estimated by the FFT. The modal analysis method which yields eigenfrequency estimations closest to the result of the FFT is the multidisciplinary approach, while the FEA approach yields the least concurring results. For the 1st eigenfrequency, ω_1, the difference between the FFT and the multidisciplinary approach is 0.06 Hz, the difference between the FFT and the control theory approach is 0.69 Hz and the difference between the FFT and the FEA approach is 2.02 Hz. For the 2nd eigenfrequency, ω_2, the difference between the FFT and the multidisciplinary approach is 0.50 Hz, the difference between the FFT and the control theory approach is 3.86 Hz and the difference between the FFT and the FEA approach is 9.72 Hz.

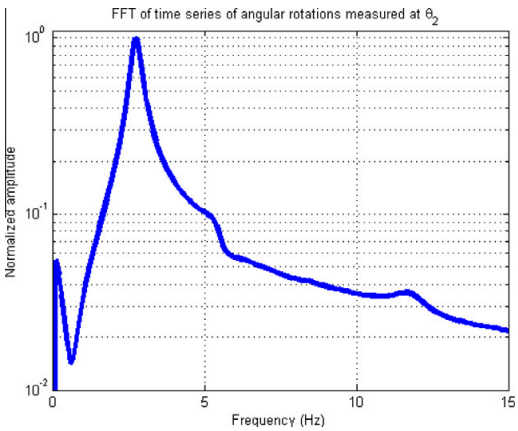


Fig. 6. FFT plot of time series of angular rotations about θ_2.

Table 4
Comparison of estimated eigenfrequencies from the FFT and different modal analysis approaches.

	Reference	Multidisciplinary approach	FEA approach	Control theory approach
	FFT of angular rotations at θ_2	Flexible body mechanism with controller effects	Flexible body mechanism with rigid joint constraints	Rigid body mechanism with controller effects
ω_1	2.7 Hz	2.76 Hz	4.72 Hz	3.39 Hz
ω_2	11.6 Hz	12.10 Hz	21.32 Hz	15.46 Hz

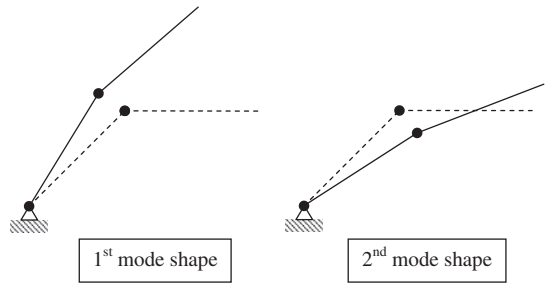


Fig. 7. 1st and 2nd mode shapes of the mechanism derived using the control theory approach (rigid bodies, flexible joints).

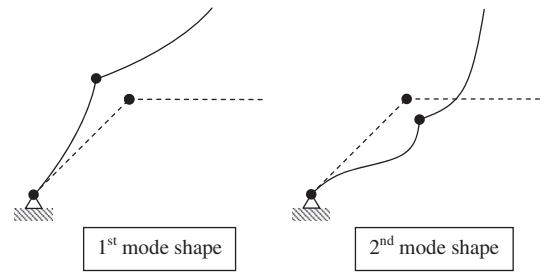


Fig. 8. 1st and 2nd mode shapes of the mechanism derived using the FEA approach (flexible bodies, rigid joints).

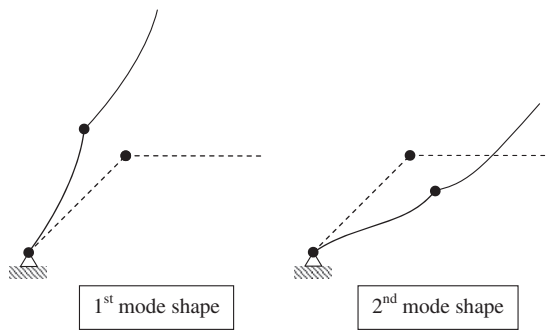


Fig. 9. 1st and 2nd mode shapes of the mechanism using the multidisciplinary approach (flexible bodies, flexible joints).

Fig. 7 is an illustration of the mode shapes derived using the control theory approach. These modes have a purely rigid body

motion, meaning no deformation of the beams. In the 1st mode shape, the joints deflect in phase, while in the 2nd mode shape, the joints deflect in anti-phase.

Fig. 8 is an illustration of the mode shapes derived using the FEA approach. These modes have a purely flexible body deformation, meaning no motion in the joints. In the 1st mode shape, both beams deflect similarly; the deformation has a U-shaped form. In the 2nd mode shape, the inner beam has an S-shape deformation, whereas the outer beam has a U-shaped deformation, although somewhat distorted.

Fig. 9 is an illustration of the mode shapes derived using the multidisciplinary approach. These mode shapes are comprised of a combination of the rigid body motions from Fig. 7 and the flexible body deformations from Fig. 8. In the 1st mode shape, the joints move in phase while the beams deform in the form of a U-shape, which further adds to the deflection of each beam. In the 2nd mode shape, the joints move in anti-phase. The inner beam has an S-shape deformation and the outer beam has a distorted U-shape deformation. In order to quantify the contributions from the rigid body motions and flexible body deformations on the mode shapes, the tip deflection in the 1st mode shape was analyzed. The total vertical displacement of the tip is the most extreme displacement in the 1st mode shape, and is a result of both joint and beam deformation. Multiplying the joint deformations $\Delta\theta_1$ and $\Delta\theta_1$ with their respective horizontal distance to the tip should give a vertical displacement of 12.88. The total vertical displacement is 20.43, yielding a ratio between joint motion and total deflection of 0.63, which means that approximately 63% of the contribution in the 1st mode shape is from the joint deflection, while approximately 37% is from the flexible body deformation.

3.1.1. Discussion of the results from the analysis of the two-joint mechanism

The results in Table 4 demonstrate a significant difference in the accuracy of the three modal analysis approaches. The modal analysis method which yields eigenfrequency estimations closest to the result of the FFT is the multidisciplinary approach. The reason for this can be found by observing the mode shapes derived using the various methods as illustrated in Figs. 7–9; the mode shapes contain significant contributions from both rigid body motions and flexible body deformations. The multidisciplinary approach includes both rigid body motions and flexible body deformations, whereas the control theory approach only includes rigid body motions and the FEA approach only includes flexible body deformations.

In this example, both the control theory approach and the FEA approach yield an overly stiff mechanism, resulting in eigenfrequencies that are too high. One reason for this can be that in this example, the joint and beam stiffness can be viewed as two springs in series. Two springs in series will always have a combined total stiffness which is less than the stiffness of the individual springs. For both the control theory and FEA approaches, either the beam or the joint stiffness, respectively, will be emulated as infinitely stiff. Yet, the stiffer the beams become, the more accurate the control theory approach will become, and similarly, the stiffer the joints become, the more accurate the FEA approach will be.

Furthermore, by simplifying the mechanism with rigid bodies for the control theory approach, all structural DOFs are excluded, which results in only two derived eigenfrequencies. Too much information about the modal parameters of the mechanism is lost in the simplification of flexible to rigid structural bodies. For the FEA approach, in which the controllers are replaced by rigid joint constraints, nearly all eigenfrequencies and mode shapes can be retrieved due to the fact that the number of structural DOFs is almost equal to the number of DOFs for the complete system. In this case, however, the eigenfrequencies and mode shapes of interest are incorrect since the joint DOFs, with its flexibility and controller effects, are not included.

A technique for quantifying the comparison between mode shapes is the Modal Assurance Criterion (MAC) [23], which can be viewed as a squared, linear regression correlation coefficient [27]. However, as pointed out in both [23,27], much care has to be taken when using the MAC, one of which being the difficulty in comparing translational and rotational DOFs due to the difference in units [23]. This is also an issue when deriving mode shapes using experimental modal analysis techniques. As can be seen in Figs. 7–9, both translational and rotational DOFs are essential in describing the mode shapes of the two-joint mechanism, thus making it difficult to derive the mode shape of the mechanism using experimental modal analysis, and complicating the task of deriving sensible results using the MAC. Excluding either type of DOFs may only serve the purpose of confirming a pre-judged result.

It is worth mentioning that because no windowing functions were used on the FFT, the accuracy of the eigenfrequency estimation by the FFT could be reduced. As mentioned in [22], in order for the Fourier transform process to produce a proper representation of the time domain sampled data in the frequency domain, the sampled data must consist of a complete representation of the data for all time or contain a periodic repetition of the measured data. When this is not the case, the error, which is known as leakage, could cause a serious distortion of the data in the frequency domain. In order to avoid this, weighting functions called windows can be used to better satisfy the periodicity requirement of the FFT. Nevertheless, even though windows can greatly reduce the leakage effect, they do cause some distortion in the data themselves and should be avoided whenever possible. For the time simulation of the two-joint mechanism in this example, which can be viewed as a form of impact test, only marginal oscillations were recordable after about 6 s of simulation time. By allowing the simulation to run for a total of 10 s, the sampled data should satisfy the requirements of the Fourier transform process, thereby eliminating the need to use windowing functions.

The basis for the FFT was the time response for a damped mechanism; in order to perform the modal analyses, the generalized eigenvalue problem, as given by Eq. (20), was solved for an undamped mechanism. This could lead to some variation in the results since, as seen in Eq. (11), undamped and damped eigenfrequencies are different. This could also contribute to explaining why the FFT in this example yields results that are consistently lower than those of the various modal analysis approaches; the damped eigenfrequencies are lower than the undamped eigenfrequencies since $\omega_d = \omega_n\sqrt{1-\zeta^2}$.

A comment about the control theory modal analysis approach should also be made. As previously mentioned, all simulations and modal analyses of the two-joint mechanism were performed in FEDEM, and to emulate rigid bodies the structural stiffness of the different flexible bodies was scaled by a factor of 100. However, this does not yield entirely rigid bodies, which means that some flexible body effects could unintentionally be included. Still, by looking at the 3rd estimated eigenfrequency for this approach, which was 143.05 Hz (compared to 14.31 Hz from the multidisciplinary approach), this should indicate that the structural parts of the mechanism are quite rigid, meaning this method should be a good approximation for a rigid body mechanism.

3.2. Modal analysis of a satellite tracking antenna

Fig. 10 shows a FEDEM model of a satellite tracking antenna. The antenna basically consists of five structural parts: (1) a pedestal, (2) a dish, (3) a combined *x*- and *y*-axis rotation housing, (4) a pair of brackets attaching the *xy* rotation housing to the pedestal

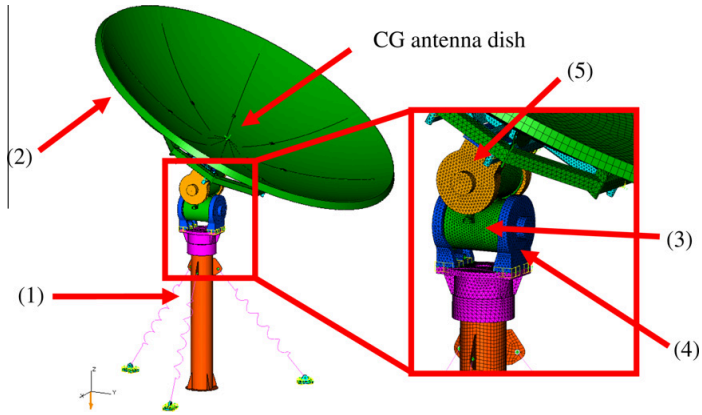


Fig. 10. FEDEM model of a satellite tracking antenna. The structural parts of the antenna are: (1) the pedestal, (2) the antenna dish, (3) the combined *xy* rotation housing, (4) the brackets attaching the *xy* rotation housing to the pedestal and (5) the brackets attaching the dish to the *xy* rotation housing.

and (5) another pair of brackets attaching the dish to the *xy* rotation housing. Each structural part in the mechanism was modeled as an FE model, therefore keeping the elasticity of the different structures intact. Due to FEDEM's reduction techniques [7], the virtual model of the mechanism was reduced from approximately 950,000 DOFs to approximately 850 DOFs. There were two joints in the mechanism: rotations about the global *x*- and *y*-axis (elevations). Each of these were handled by a motor and controlled by an angular position feedback PID controller. However, for this case, the controllers were deliberately not fully optimized. The sensors and actuators for each respective rotational DOF were all collocated.

To verify the accuracy of the modal analyses of the antenna mechanism, a time simulation response of the antenna when suddenly subjected to gravity was analyzed using the FFT algorithm. The time responses of the angular rotations in the antenna dish's center of gravity (CG) in the *x*-, *y*- and *z*-direction were used. The simulation ran for 10 s with time increment of 0.001 s, yielding a frequency sampling rate f_s of 1000 Hz and a frequency resolution of approximately 0.1 Hz. A decaying exponential windowing function, e^{-t}, was used for the FFTs. The FFT plots of the time series are shown in Figs. 11–13, and the estimated eigenvalues are presented in Table 5.

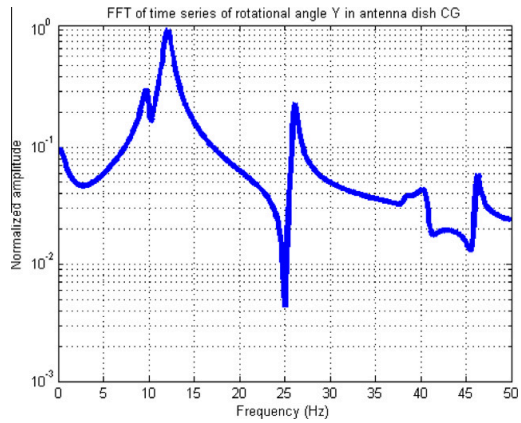


Fig. 12. FFT plot of time series of angular rotations in the antenna dish's CG in *y*-direction.

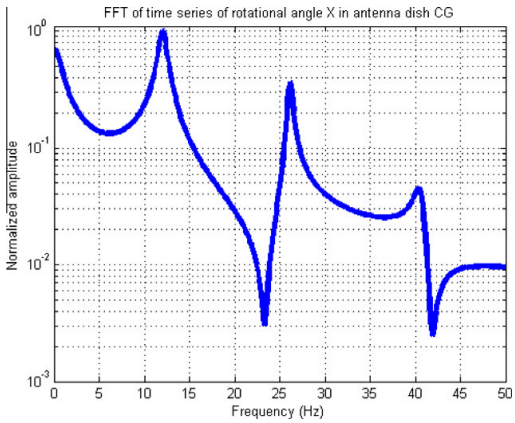


Fig. 11. FFT plot of time series of angular rotations in the antenna dish's CG in *x*-direction.

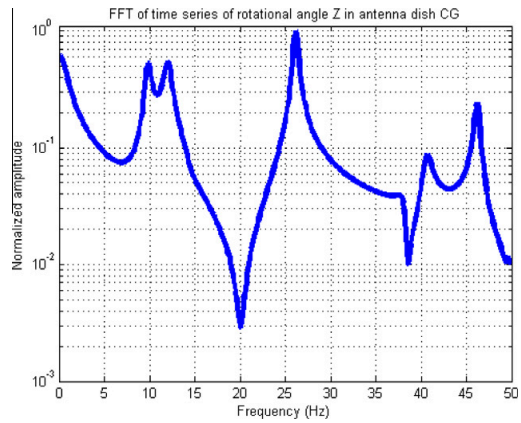


Fig. 13. FFT plot of time series of angular rotations in the antenna dish's CG in *z*-direction.

Table 5
Comparison of estimated eigenfrequencies from the FFTs and the different modal analysis approaches.

	References			Multidisciplinary approach	FEA approach	Control theory approach
	FFT of angular rotations about X (Hz)	FFT of angular rotations about Y (Hz)	FFT of angular rotations about Z (Hz)	Flexible body mechanism with controller effects (Hz)	Flexible body mechanism with rigid joint constraints (Hz)	Rigid body mechanism with controller effects (Hz)
ω_1	–	9.7	9.8	9.26	16.05	10.91
ω_2	12.0	12.1	12.0	11.16	16.60	13.91
ω_3	26.1	26.2	26.1	26.41	27.55	–
ω_4	40.4	40.2	40.7	41.64	45.91	–
ω_5	–	46.3	46.2	47.24	49.46	–

As in the previous section, three different approaches were used to perform modal analyses of the antenna: a control theory approach (rigid body mechanism with controller effects), an FEA approach (flexible body mechanism with rigid joint constraints) and a multidisciplinary approach (flexible body mechanism with controller effects). To derive the eigenfrequencies for the antenna using the control theory approach, analytical calculations based on the antenna's rigid body motions were used. The only rigid body motions of the antenna were rotations in the two joints of the mechanism: rotations about the global x- and y-axis (elevations). The following data were used to derive the eigenfrequencies using the control theory approach: the mass of the antenna dish was approximately 275 kg, while the distance from the antenna dish's CG to the x- and y-axis revolute joints were 0.69 m and 0.88 m, respectively. This should yield a moment of inertia for joint rotations about the x-axis of $J_x = 131\ \text{kgm}^2$, and for joint rotations about the y-axis of $J_y = 213\ \text{kgm}^2$. Both the x- and y-axis revolute joint stiffness were 1,000,000 Nm/rad. A comparison of the estimated eigenfrequencies derived by the different modal analysis approaches and the estimated eigenfrequencies derived from the FFTs are shown in Table 5.

As can be seen in Figs. 11–13 and Table 5, the FFTs yield approximately identical eigenfrequency estimations. Both of the FFTs for the angular rotations about the y-axis (Fig. 12) and z-axis (Fig. 13) in the dish's CG yield values for all five eigenfrequencies. The FFT for the angular rotations about the x-axis (Fig. 11) in the dish's CG only yields values for the 2nd, 3rd and 4th eigenfrequencies. The multidisciplinary approach yields values closest to the FFTs for all eigenfrequencies. The FEA approach yields less confirming values for all eigenfrequencies than the multidisciplinary approach. The control theory approach only yields values for the first two eigenfrequencies. The most extreme differences between the FFTs and the multidisciplinary, FEA and control theory approaches for the five eigenfrequencies are listed in Table 6.

As can be seen in Tables 5 and 6, for the first two eigenfrequencies, the multidisciplinary approach yields values less than the FFTs, though for all other eigenfrequencies, the multidisciplinary approach yields values greater than the FFTs. Both the FEA and control theory approach yield values consistently greater than the FFTs. As previously mentioned, the control theory approach yields values only for the first two eigenfrequencies.

Figs. 14–18 show the first five mode shapes of the antenna. The 1st mode shape has mainly rotations about the y-joint and some swaying of the pedestal in the global x-direction. The 2nd mode shape primarily has mainly rotations about the x-joint and some swaying of the pedestal in the global y-direction. The 3rd mode shape has both flexing of the antenna dish, rotations about the x-joint and swaying of the pedestal in the global y-direction. The 4th mode shape has both flexing of the antenna dish, rotations about the x-joint and large swaying of the pedestal in the global x-direction. The 5th mode shape has large swaying of the pedestal in the global x-direction and rotations about the y-joint.

3.2.1. Discussion of the results from the analysis of the satellite tracking antenna

As can be seen in Figs. 11–13 and Table 5, there seems to be a close correlation between the estimated eigenfrequencies derived by the various FFTs, indicating that the FFTs should represent a satisfactory control sample. The reason why all the FFTs do not yield values for all eigenfrequencies can be found when looking at the mode shapes that correspond to the different eigenfrequencies. Only the 2nd, 3rd and 4th mode shapes contain any significant rotations about the x-axis of the antenna dish's CG. Since the FFTs for angular rotations about the y- and z-axis of the antenna dish's CG yield results for all eigenfrequencies, this indicates that all mode shapes have at least some angular rotations about these axes. In order to better satisfy the requirements of the Fourier transform process, a decaying exponential windowing function was used on the time domain sampled data of the angular rotations about the x-, y- and z-axis of the antenna dish's CG prior to the Fourier transformation. As stated in Section 3.1.1, even though windows can greatly reduce the leakage effect, they do cause some distortion in the data. However, for this example, none of the time domain sampled data satisfied the periodicity requirement of the Fourier transform process, thus causing serious errors in the FFTs such as false and suppressed frequency peaks and noise. But when comparing the windowed and unwindowed FFTs, only slight differences in the true frequency peaks were discovered, primarily on the y-axis angular rotations; the greatest of these were 0.2 Hz.

Additionally, the FFTs were used on time series for a mechanism controlled by a PID controller; to perform the modal analyses, the generalized eigenvalue problem, as given by Eq. (20), was solved

Table 6
Most extreme differences between the FFTs and the different modal analysis approaches. The +/− signs indicate higher and lower eigenfrequency values, respectively, compared to the FFTs.

	Reference values from FFTs High/low FFT values (Hz)	Multidisciplinary approach Flexible body mechanism with controller effects (Hz)	FEA approach Flexible body mechanism with rigid joint constraints (Hz)	Control theory approach Rigid body mechanism with controller effects (Hz)
ω_1	9.8/9.7	−0.54	+6.35	+1.21
ω_2	12.1/12.0	−0.94	+4.60	+1.91
ω_3	26.1	+0.31	+1.45	–
ω_4	40.2	+1.44	+5.71	–
ω_5	46.2	+1.04	+3.26	–

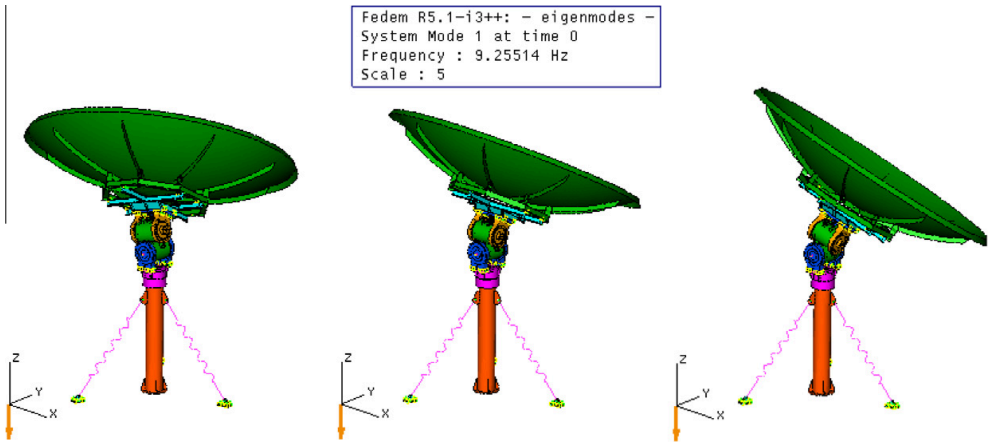


Fig. 14. The 1st mode shape. The middle figure is the undeformed shape. The left and right figures are the two extremity deformational shapes. This mode shape mainly has rotations about the *y*-joint and some swaying of the pedestal in the global *x*-direction.

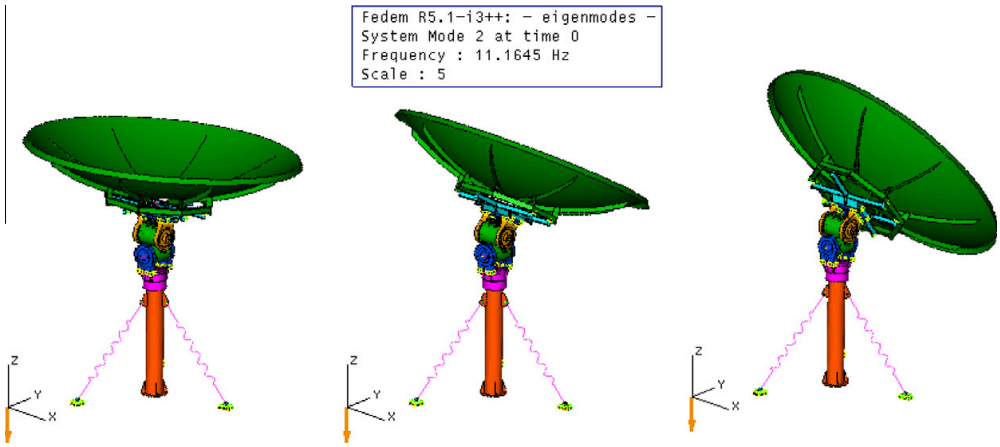


Fig. 15. The 2nd mode shape. This mode shape mainly has rotations about the *x*-joint and some swaying of the pedestal in the global *y*-direction.

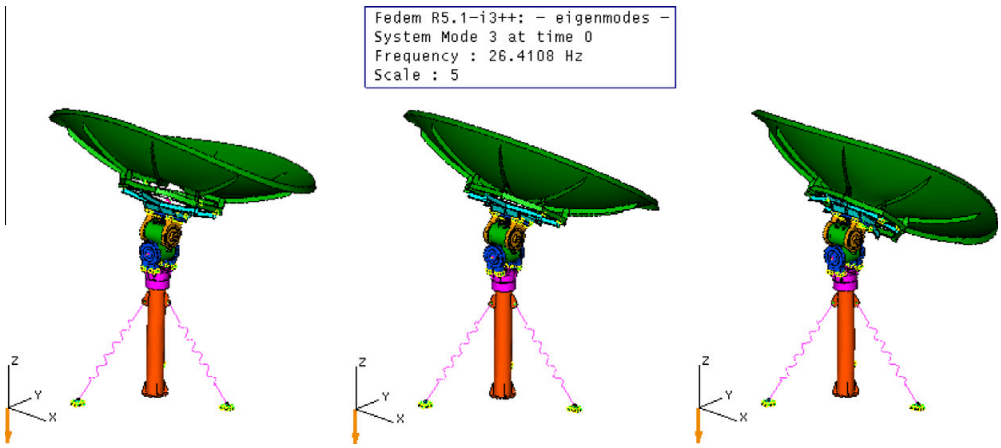


Fig. 16. The 3rd mode shape. This mode shape has both flexing of the antenna dish, rotations about the *x*-joint and swaying of the pedestal in the global *y*-direction.

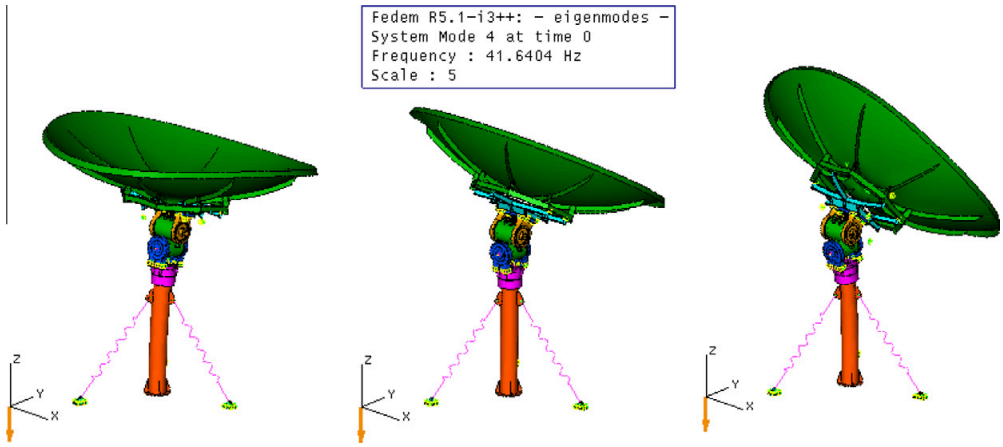


Fig. 17. The 4th mode shape. This mode shape has both flexing of the antenna dish, rotations about the *x*-joint and large swaying of the pedestal in the global *x*-direction.

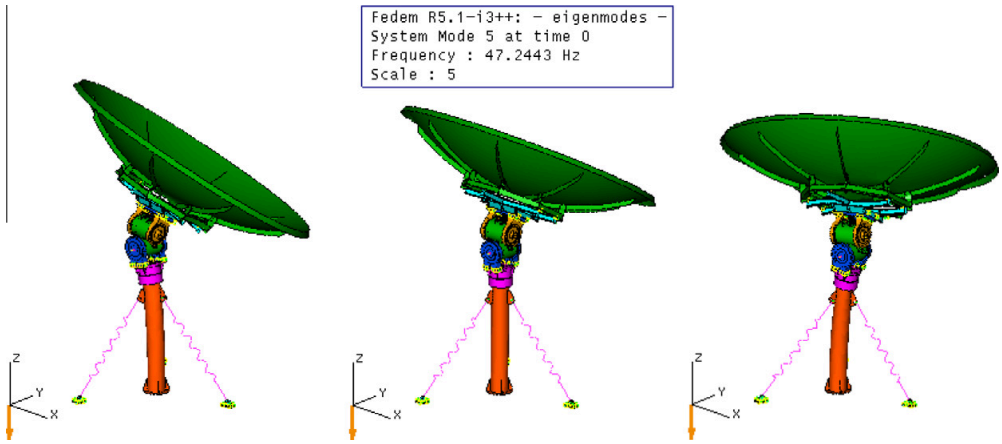


Fig. 18. The 5th mode shape. This mode shape has large swaying of the pedestal in the global *x*-direction and rotations about the *y*-joint.

for undamped mechanisms. This could lead to some variation in the results since, as mentioned in Table 1, both K_i and K_d affect the damping of the system, and as seen in Eq. (11), undamped and damped eigenvalues are different.

As can be seen in Tables 5 and 6, there is a significant difference in the results from the three modal analysis approaches. As for the two-joint mechanism, the modal analysis method which yields eigenfrequency estimations closest to the results of the FFTs is the multidisciplinary approach. As for the two-joint mechanism, both the control theory approach and the FEA approach consistently estimate eigenfrequencies that are too high. Again, much of the reason for this can be found by observing the mode shapes. All mode shapes have a significant contribution from flexible body dynamics, while only the first two mode shapes have any significant rigid body motion. For this reason, the control theory approach only yields two eigenfrequencies: rotations about the *x*- and *y*-joint. These mode shapes are also derived using both the multidisciplinary approach and the FEA approach. Still, the FEA approach does not include the stiffness properties of the joints since these are restricted by rigid boundary conditions for this approach. This results in the yielding of a much stiffer mechanism for these eigenfrequencies in comparison to the multidisciplinary approach, which includes both joint stiffness and flexibility of the various structures. Since these two mode shapes have significant contributions from both joint and structural stiffness, this would explain why the control theory approach and the FEA approach yield higher eigenfrequencies than the multidisciplinary approach and the FFTs. For the other mode shapes, the FEA approach yields consistently higher eigenfrequencies than both the multidisciplinary approach and the FFTs. Again, the reason for this could be that the FEA makes the mechanism too stiff by making the joints rigid. For the control theory approach, these eigenfrequencies cannot be discovered because the flexibility of the structure is the dominant factor in these mode shapes. As a result, all stiffness contributors should be included in the modal analysis in order to make the analysis more accurate.

The virtual simulations in FEDEM were performed using the Newmark-β method with Newton–Raphson iterations [7] for time integration of the mechanisms, and the Lobatto IIIC algorithm [7] for time integration of the controllers. The use of different

algorithms for time integration of the mechanisms and controllers could be a source of error; a slight difference in frequency response functions was discovered when two identical models, one active and one passive, were simulated in FEDEM. For the eigenvalue analyses, the Lanczos algorithm [28] was used. The FFTs were calculated in both FEDEM and MATLAB, and yielded similar results.

One final comment should be made about the computational work load caused by the multidisciplinary approach compared to that of the FEA approach. For active systems, whether they be structures or mechanisms, the number of structural DOFs is usually far greater than the number of DOFs governed by controllers. The controllers are usually limited to affecting only a few system DOFs. Thus, solving the eigenvalue problem for the complete system, meaning that both controller effects and flexible body dynamics are included, should not take a significantly greater amount of time to solve than that of regular FEA eigenvalue problem solutions, in which only structural DOFs are included.

4. Conclusion

In this paper, a method for modal analysis of active flexible multibody systems has been derived. Including both controller effects and flexible body dynamics when performing modal analyses of active flexible multibody systems seems to yield more accurate eigenfrequency estimations. The new multidisciplinary modal analysis approach combines the best from the traditional separate control and structural design disciplines to provide accurate eigenfrequency and mode shape calculations.

By implementing this approach in a multidisciplinary software system (FEDEM), finite element analysts will be able to include the controller effects by augmenting the mass, stiffness and damping properties of the mechanical system. The passive mechanical system will then represent the active system during modal analyses since all singularities will be replaced by mechanical equivalents of the control system. Traditional finite element solvers can then be used to derive the eigenfrequencies and eigenvectors of the closed-loop system. This approach can therefore easily be implemented into a finite element code. However, it is important to bear in mind that the proposed multidisciplinary modal analysis approach has not yet been fully implemented into a commercial software system, and is therefore considered to be undergoing further research and improvement.

This paper has been limited to dealing with an undamped modal analysis of active flexible multibody systems in which only controller effects equivalent to mechanical stiffness and mass properties are included.

Acknowledgements

The authors would like to acknowledge the guidance and assistance of Professor Ole Ivar Sivertsen and Professor Kristian Tønder at the Norwegian University of Science and Technology (NTNU). The authors would also like to acknowledge the financial support from the Research Council of Norway and the other partners in the Lean Product Development (LPD) Project.

References

[1] Preumont A. Vibration control of active structures: an introduction. 2nd ed. Dordrecht, The Netherlands: Kluwer Academic Publishers; 2002.
[2] Inman DJ. Vibration with control. Chichester, England: John Wiley & Sons Ltd.; 2006.
[3] Balchen JG, Andresen T, Foss BA. Reguleringsteknikk. 5th ed. Trondheim, Norway: Department of Engineering Cybernetics, Norwegian University of Science and Technology; 2003 [In Norwegian].
[4] Thomson WT, Dahleh MD. Theory of vibration with applications. 5th ed. Upper Saddle River, NJ, USA: Prentice Hall, Inc.; 1998.
[5] Palm WJ. Mechanical vibration. Hoboken, NJ, USA: John Wiley & Sons, Inc.; 2007.
[6] Cook RD, Malkus DS, Plesha ME, Witt RJ. Concepts and applications of finite element analysis. 4th ed. New York, NY, USA: John Wiley & Sons. Inc.; 2002.
[7] Sivertsen OI. Virtual testing of mechanical systems – theories and techniques. Lisse, The Netherlands: Swets & Zeitlinger B.V.; 2001.
[8] Meirovitch L. Dynamics and control of structures. New York, NY, USA: John Wiley & Sons. Inc.; 1990.
[9] Rastgaar MA, Ahmadian M, Southward SC. Orthogonal eigenstructure control with non-collocated actuators and sensors. J Vib Contr 2009;15:1019–47.
[10] Alkhatib R, Golnaraghi MF. Active structural vibration control: a review. Shock Vib Dig. 2003;35:367–83.
[11] Bratland M, Rølvåg T. Modal analysis of lumped flexible active systems (part 1). In: Proceedings of SIMS 2008: The 48th Scandinavian conference on simulation and modeling. Oslo, Norway; 2008.
[12] Sharon A, Hogan N, Hardt DE. Controller design in the physical domain. J Franklin Inst. 1991;328:697–721.
[13] Baz A. Active control of periodic structures. Trans ASME J Vib Acoust 2001;123:472–9.
[14] Bernzen W. Active vibration control of flexible robots using virtual spring-damper systems. J Intell Robot Syst, Theory Appl. 1999;24:69–88.
[15] Ohnishi K, Shibata M, Murakami T. Motion control for advanced mechatronics. IEEE/ASME Trans Mechatron 1996;1:56–67.
[16] Ryu J-H, Kwon D-S, Hannaford B. Stability guaranteed control: time domain passivity approach. IEEE Trans Contr Syst Technol 2004;12:860–8.
[17] Tzou HS, Tseng CI. Distributed piezoelectric sensor actuator design for dynamic measurement control of distributed parameter-systems – a piezoelectric finite-element approach. J Sound Vib 1990;138:17–34.
[18] Lim YH, Gopinathan SV, Varadan VV, Varadan VK. Finite element simulation of smart structures using an optimal output feedback controller for vibration and noise control. Smart Mater Struct 1999;8:324–37.
[19] Astrom KJ, Hagglund T. The future of PID control. Control Eng Pract 2001;9:1163–75.
[20] Astrom KJ, Hagglund T. Revisiting the Ziegler–Nichols step response method for PID control. J Process Contr 2004;14:635–50.
[21] Cooley JW, Tukey JW. An algorithm for the machine calculation of complex Fourier series. Math Comput 1965;19:297–301.
[22] Avitabile P. Experimental modal analysis a simple non-mathematical presentation. Sound Vib 2001;35:20–31.
[23] Ewins DJ. Modal testing: theory, practice and application. Baldock, England: Research Studies Press; 2000.
[24] Schwarz BJ, Richardson MH. Experimental modal analysis. In: Proceedings of CSI Reliability Week. Orlando, Florida, USA; 1999.
[25] Ramsey KA. Experimental modal analysis, structural modifications and FEM analysis on a desktop computer. Sound Vib 1983;17:19–27.
[26] Sivertsen OI, Waloen AO. Non-linear finite element formulations for dynamic analysis of mechanisms with elastic components. Washington, New York, USA: ASME; 1982. p. 7.
[27] Allemang RJ. The modal assurance criterion – twenty years of use and abuse. Sound Vib 2003;37:14–23.
[28] Lanczos C. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. J Res Natl Bur Stand 1950;45:255–82.

PAPER III

Bratland, M., Haugen, B., and Rølvåg, T., "Modal Analysis of Active Flexible Multibody Systems Containing PID Controllers with Non-Collocated Sensors and Actuators," *Computers and Structures*, 2011, (submitted).

Modal Analysis of Active Flexible Multibody Systems Containing PID Controllers with Non-Collocated Sensors and Actuators

Magne Bratland, Bjørn Haugen, Terje Rølvåg

Department of Engineering Design and Materials, Norwegian University of Science and Technology, Richard Birkelands veg 2B, N-7491 Trondheim, Norway

E-mail address for corresponding author: magne.bratland@ntnu.no

Abstract
A method for performing modal analysis of undamped active flexible multibody systems with collocated sensors and actuators in a finite element environment was recently developed by the authors. In this paper, the theory is further expanded to include systems with non-collocated sensors and actuators, damping and steady-state error elimination. The closed-loop eigenvalue problem for active flexible multibody systems with multiple-input multiple-output proportional-integral-derivative (PID) feedback type controllers and multiple degrees of freedom finite element models is solved.

Keywords:
modal analysis, closed-loop eigenvalue problem, flexible multibody system, position feedback PID controller, non-collocated sensors and actuators.

1. Introduction

Modal analysis and dynamic simulation of active flexible multibody systems - from now on referred to as active mechanisms - are a multidisciplinary challenge. The dynamic performance of such products is strongly dependent on an optimal interaction between the controllers and the mechanical components. An important tool in the optimization of such products is modal analysis, which predicts modal parameters, i.e. natural frequencies, mode shapes and damping ratios, for the active system. Due to the complexity of the mechanical components, both in form and function, it may practical to handle such systems through a finite element (FE) approach. Effective time domain dynamic simulations of multibody systems in an FE environment have been described by for instance Géradin and Cardona [1] and Sivertsen [2].

The authors have shown in [3] how PID-type controllers with collocated sensors and actuators affect mechanical systems based on the various types of sensor input (position, velocity or acceleration), and how to add the controller gains and mechanical properties into the second-order system matrices. However, the theory for performing modal analyses of such systems was only derived for the undamped case without steady-state error elimination, i.e. no damping or position feedback integral gain. Damping can occur due to both the mechanical system and controllers, e.g. [3-9], while position feedback integral gain can cause the system to become unstable [3]. As demonstrated in [4], damping and system instability are basically determined by

the same factor: the constant in the exponent of the oscillation amplitude envelope; a negative constant yields a decaying oscillation, i.e. damping, whereas a positive constant yields a growing oscillation, and thus instability. The eigenfrequencies of such systems are not the same as for a stable undamped system, e.g. [4, 10]. In addition to altering the eigenfrequencies, damping also causes the eigenvectors to become complex [11].

As stated in [12], the solution procedures of non-proportionally damped systems mainly follow two routes: the state-space method and approximate methods in "*n*-space". The state-space method is exact in nature but requires significant numerical effort for obtaining the eigensolutions, as the size of the problem doubles from *n* to 2*n*. Most of the *n*-space methods either seek an optimal decoupling of the equations of motion or simply neglect the off-diagonal terms of the modal damping matrix. Following such methodologies will still yield only real mode shapes. The accuracy of these methods, other than the light damping assumption, depends upon various factors such as frequency separation between modes, driving frequency, etc [12]. In order to solve the damped eigenvalue problem for a system with *n* degrees of freedom (DOFs), the second-order differential equations can be reformulated as a first-order 2*n*-dimensional matrix equation system, as shown for instance in [13].

How to solve the eigenvalue problem for a closed-loop system on state-space form has been shown by e.g. Rastgaar *et al.* [14]. However, their theory is only valid for systems containing controller gains proportional to position and velocity, i.e. equivalent to stiffness and damping, respectively. As stated by Astrom and Hagglund [15, 16], PID controllers are the most common type of controllers in use today, and as shown by the authors in [3], depending on the type of sensor inputs, PID controllers can contain gains proportional to position, velocity, acceleration or the time integral of position, i.e. equivalent to stiffness, damping, mass and steady-state error elimination, respectively. Thus, modal analysis methods for active mechanisms should also take into consideration the effects caused by the various types of PID controllers.

This paper addresses the theory for solving the eigenvalue problem for active mechanisms that contain both damping and steady-state error elimination and controllers with non-collocated sensors and actuators. The objective of this work is to help engineers working in an FE environment be able to accurately predict eigenfrequencies and mode shapes of active mechanisms containing any type of PID controllers, with the exception being controllers containing acceleration feedback derivative gains. The controllers can be of type single-input single-output (SISO) or multiple-input multiple-output (MIMO), and the sensors and actuators for the controllers can be either collocated or non-collocated. The theory derived in this work is intended to be implemented in an FE software system, but for the sake of validation, all eigenvalue problems in this work are solved in MATLAB[1] using the `eig()` routine [17]. All time domain simulations are performed in FEDEM[2].

[1] MATLAB by The MathWorks, Inc., version R2010a.
[2] FEDEM (Finite Element in Dynamics of Elastic Mechanisms) simulation software is a multibody dynamics package distributed by Fedem Technology AS. It is based on the finite element method and uses model reduction techniques to effectively perform nonlinear time domain dynamic simulations of active flexible multibody systems [2, 18], version R5.0.

2. Non-Collocated Sensors and Actuators

It has been demonstrated by the authors in [3] that for a single degree of freedom (SDOF) mechanical system with a PID feedback type controller, the equations of motion for the free vibrations based on sensor readings can be written as:

Position feedback PID

$$m\ddot{r}(t)+\left(c+K_d\right)\dot{r}(t)+\left(k+K_p\right)r(t)+K_i\int r(t)dt=0 \tag{1}$$

Velocity feedback PID

$$\left(m+K_d\right)\ddot{r}(t)+\left(c+K_p\right)\dot{r}(t)+\left(k+K_i\right)r(t)=0 \tag{2}$$

Acceleration feedback PID

$$K_d\dddot{r}(t)+\left(m+K_p\right)\ddot{r}(t)+\left(c+K_i\right)\dot{r}(t)+kr(t)=0 \tag{3}$$

Further, it was also shown that for a multiple degrees of freedom (MDOF) mechanical system with a feedback type controller, the equation of motion for the free vibration can be written as:

$$\left(\mathbf{M}+\mathbf{G}_{Acc}\right)\ddot{\mathbf{r}}(t)+\left(\mathbf{C}+\mathbf{G}_{Vel}\right)\dot{\mathbf{r}}(t)+\left(\mathbf{K}+\mathbf{G}_{Pos}\right)\mathbf{r}(t)=\mathbf{0} \tag{4}$$

where **M** is the mass matrix, **C** is the damping matrix and **K** is the stiffness matrix of the mechanical system. $\mathbf{G}_{Acc}$, $\mathbf{G}_{Vel}$ and $\mathbf{G}_{Pos}$ are the controller gradient acceleration, velocity and position matrices, respectively, corresponding to the equivalent mass, damping and stiffness matrices from the controller. $\mathbf{r}$ is the position vector, $\dot{\mathbf{r}}$ is the velocity vector and $\ddot{\mathbf{r}}$ is the acceleration vector of the system. The dimensions of all the matrices are $n\times n$ and the vectors $n\times 1$, where n is the number of DOFs.

Equation (4) is valid for all PID controllers only containing controller elements proportional to position, velocity or acceleration. Examples of controllers not covered by Equation (4) are: position feedback controllers containing integral gains or acceleration feedback controllers containing derivative gains, though the latter variant will not be covered in this work. Based on Equations (1) and (4), the equation of motion for the free vibration of an active MDOF system containing a position feedback PID controller can be written as:

$$\left(\mathbf{M}+\mathbf{G}_{Acc}\right)\ddot{\mathbf{r}}(t)+\left(\mathbf{C}+\mathbf{G}_{Vel}\right)\dot{\mathbf{r}}(t)+\left(\mathbf{K}+\mathbf{G}_{Pos}\right)\mathbf{r}(t)+\mathbf{G}_{SSEE}\int\mathbf{r}(t)dt=\mathbf{0} \tag{5}$$

where $\mathbf{G}_{SSEE}$ is the controller gradient steady-state error elimination matrix and $\int\mathbf{r}\,dt$ is the position time integral vector of the system. As in Equation (4), the dimensions of all the matrices are $n\times n$ and the vectors $n\times 1$.

For a system with only collocated sensors and actuators, the controller gradient matrices $\mathbf{G}_{Acc}$, $\mathbf{G}_{Vel}$, $\mathbf{G}_{Pos}$ and $\mathbf{G}_{SSEE}$ will all be diagonal matrices. For a system with one or more non-collocated sensors and actuators, the $\mathbf{G}_{Acc}$, $\mathbf{G}_{Vel}$, $\mathbf{G}_{Pos}$ and $\mathbf{G}_{SSEE}$ matrices will be unsymmetrical.

Figure 1 illustrates an active MDOF system in which the sensor and actuator are non-collocated.

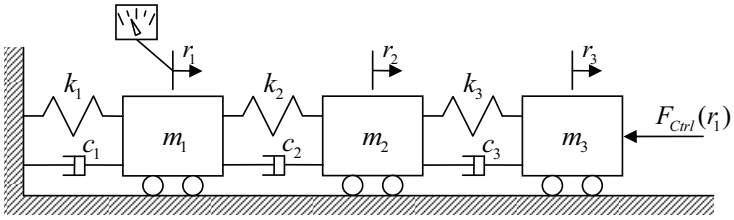


Figure 1: Active MDOF system with non-collocated sensor and actuator.

The mass-spring-damper system in Figure 1 consists of three masses (m_1, m_2 and m_3) in series connected by springs (k_1, k_2 and k_3) and dampers (c_1, c_2 and c_3). Each mass has one DOF: translation in the horizontal plane, named r_1, r_2 and r_3, respectively. Expressing the position vector as: $\mathbf{r} = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}^T$, and the velocity and acceleration vectors in a similar manner, the mass, damping and stiffness matrix of the system in Figure 1 can be written as:

$$\mathbf{M} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} c_1 + c_2 & -c_2 & 0 \\ -c_2 & c_2 + c_3 & -c_3 \\ 0 & -c_3 & c_3 \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix} \tag{6}$$

In Figure 1, a sensor is placed on r_1, while the actuator exerts a force on m_3, i.e. directly affecting r_3. If the controller of the system in Figure 1 is a velocity feedback PID controller, all the matrices $\mathbf{G}_{Acc}$, $\mathbf{G}_{Vel}$ and $\mathbf{G}_{Pos}$ will be non-zero, as shown in [3]. Figure 2 illustrates a block diagram for the velocity feedback PID controller used in the system in Figure 1.

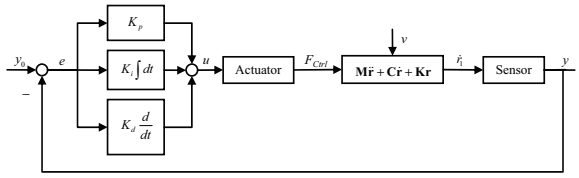


Figure 2: Velocity feedback PID controller.

Based on Equations (2) and (5), this would yield the following gradient matrices $\mathbf{G}_{Acc}$, $\mathbf{G}_{Vel}$, $\mathbf{G}_{Pos}$ and $\mathbf{G}_{SSEE}$:

$$\mathbf{G}_{Acc} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ K_d & 0 & 0 \end{bmatrix}, \quad \mathbf{G}_{Vel} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ K_p & 0 & 0 \end{bmatrix}, \quad \mathbf{G}_{Pos} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ K_i & 0 & 0 \end{bmatrix}, \quad \mathbf{G}_{SSEE} = \mathbf{0} \tag{7}$$

The position of the controller gains in their respective matrices in Equation (7) are determined by the DOFs of the sensor and actuator placements. Since the sensor is placed on r_1, the controller gains will be placed in column number 1, and since the actuator affects r_3, the controller gains will be placed in row number 3.

Combining Equations (6) and (7) yields the following matrix equation system for the free vibration of the active system in Figure 1 containing a velocity feedback PID controller as:

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ K_d & 0 & m_3 \end{bmatrix}\begin{bmatrix} \ddot{r}_1 \\ \ddot{r}_2 \\ \ddot{r}_3 \end{bmatrix}+\begin{bmatrix} c_1+c_2 & -c_2 & 0 \\ -c_2 & c_2+c_3 & -c_3 \\ K_p & -c_3 & c_3 \end{bmatrix}\begin{bmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \dot{r}_3 \end{bmatrix}+\begin{bmatrix} k_1+k_2 & -k_2 & 0 \\ -k_2 & k_2+k_3 & -k_3 \\ K_i & -k_3 & k_3 \end{bmatrix}\begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}=\mathbf{0} \tag{8}$$

If the controller for the system in Figure 1 is a position feedback PID controller, $\mathbf{G}_{Acc}$ will be zero, while all of the matrices $\mathbf{G}_{Vel}$, $\mathbf{G}_{Pos}$ and $\mathbf{G}_{SSEE}$ will be non-zero. Based on Equations (1) and (5), this would yield the following gradient matrices $\mathbf{G}_{Acc}$, $\mathbf{G}_{Vel}$, $\mathbf{G}_{Pos}$ and $\mathbf{G}_{SSEE}$:

$$\mathbf{G}_{Acc}=\mathbf{0} \quad , \quad \mathbf{G}_{Vel}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ K_d & 0 & 0 \end{bmatrix} \quad , \quad \mathbf{G}_{Pos}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ K_p & 0 & 0 \end{bmatrix} \quad , \quad \mathbf{G}_{SSEE}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ K_i & 0 & 0 \end{bmatrix} \tag{9}$$

Combining Equations (6) and (9) yields the following matrix equation system for the free vibration of the active system in Figure 1 containing a position feedback PID controller as:

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}\begin{bmatrix} \ddot{r}_1 \\ \ddot{r}_2 \\ \ddot{r}_3 \end{bmatrix}+\begin{bmatrix} c_1+c_2 & -c_2 & 0 \\ -c_2 & c_2+c_3 & -c_3 \\ K_d & -c_3 & c_3 \end{bmatrix}\begin{bmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \dot{r}_3 \end{bmatrix}$$
$$+\begin{bmatrix} k_1+k_2 & -k_2 & 0 \\ -k_2 & k_2+k_3 & -k_3 \\ K_p & -k_3 & k_3 \end{bmatrix}\begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}+\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ K_i & 0 & 0 \end{bmatrix}\begin{bmatrix} \int r_1 dt \\ \int r_2 dt \\ \int r_3 dt \end{bmatrix}=\mathbf{0} \tag{10}$$

3. Eigenvalue Analysis of System Containing Damping and Steady-State Error Elimination

The equation of motion for the free vibration of an SDOF mechanical system with a position feedback PID controller is given by Equation (1). For convenience, Equation (1) can be rewritten as:

$$m_{eff}\ddot{r}+c_{eff}\dot{r}+k_{eff}r+q_{eff}\int r\,dt=0 \tag{11}$$

where m_{eff} is the effective mass of the system, while c_{eff} is the effective damping, k_{eff} is the effective stiffness and q_{eff} is the effective steady-state error elimination of the system. Assuming a solution for Equation (11) with the form $r(t) = e^{st}$ gives a characteristic equation:

$$m_{eff}s^2 + c_{eff}s + k_{eff} + q_{eff}s^{-1} = 0 \tag{12}$$

which is equal to:

$$m_{eff}s^3 + c_{eff}s^2 + k_{eff}s + q_{eff} = 0 \tag{13}$$

Equation (13) is a cubic equation and has a solution for the roots s as either three real and unequal roots, three real roots in which at least two are equal, or one real root and a pair of complex conjugate roots [19]. If the roots of Equation (13) are a pair of complex conjugate roots, the imaginary part of s is the frequency of oscillation and the real part is the constant in the exponent of the oscillation amplitude envelope. A schematic overview of different instances of the roots s is shown in Figure 3.

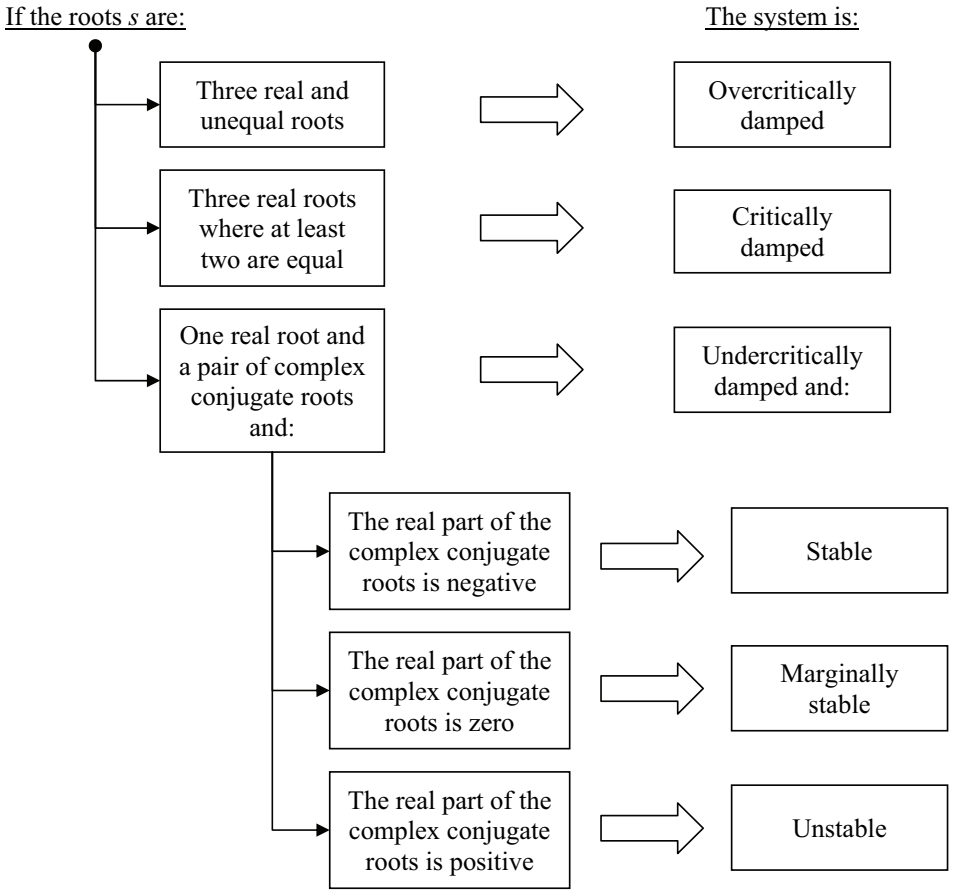


Figure 3: Schematic illustration of the various instances of the roots s.

If the real part of the complex conjugate roots is positive, the oscillation is growing, and the system is therefore unstable. The borderline case for stability is when the real part of s is zero. When this occurs, s is equal to ω_n. For this to occur, $c_{eff}s$ and $q_{eff}s^{-1}$ have to be zero. This yields:

$$c_{eff}s - q_{eff}s^{-1} = 0 \;\Rightarrow\; q_{eff} = c_{eff}s^2 \tag{14}$$

and since for this special case, $s = \omega_n = \sqrt{k_{eff} / m_{eff}}$, the stability borderline value for q_{eff} can thus be derived as:

$$q_{eff_s} = c_{eff}\frac{k_{eff}}{m_{eff}} \tag{15}$$

In order to derive the roots of Equation (11), one possibility is to transform it into a first-order or state-space form. Based on [13], one possible way of writing Equation (11) in state-space form is:

$$\mathbf{A}\mathbf{x} - \mathbf{B}\dot{\mathbf{x}} = \mathbf{0} \tag{16}$$

where:

$$\mathbf{x} = \begin{bmatrix} \int r\,dt \\ r \\ \dot{r} \end{bmatrix}, \quad \dot{\mathbf{x}} = \begin{bmatrix} r \\ \dot{r} \\ \ddot{r} \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} q_{eff} & 0 & 0 \\ 0 & k_{eff} & 0 \\ 0 & 0 & m_{eff} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -k_{eff} & -c_{eff} & -m_{eff} \\ k_{eff} & 0 & 0 \\ 0 & m_{eff} & 0 \end{bmatrix} \tag{17}$$

In general, the dimensions of $\mathbf{x}$ and $\dot{\mathbf{x}}$ are $3n \times 1$, whereas the dimensions of $\mathbf{A}$ and $\mathbf{B}$ are both $3n \times 3n$, where n is the number of DOFs. In order to obtain the roots of Equation (16), the generalized eigenvalue problem, as given for instance in [20], can be solved. With the matrices in Equation (17) inserted, it may be written as:

$$\mathbf{A}\boldsymbol{\Phi} = \mathbf{B}\boldsymbol{\Phi}\boldsymbol{\Lambda} \tag{18}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix of the generalized eigenvalues and $\boldsymbol{\Phi}$ is a full matrix whose columns are the corresponding eigenvectors. The diagonal elements of the eigenvalue matrix $\boldsymbol{\Lambda}$ correspond to the roots s of Equation (13).

Based on Equation (11), the second-order differential equation of the free vibration for an MDOF system can be written as:

$$\mathbf{M}_{eff}\ddot{\mathbf{r}} + \mathbf{C}_{eff}\dot{\mathbf{r}} + \mathbf{K}_{eff}\mathbf{r} + \mathbf{Q}_{eff}\int \mathbf{r}\,dt = \mathbf{0} \tag{19}$$

where $\mathbf{M}_{eff}$ is the effective mass matrix of the system, while $\mathbf{C}_{eff}$ is the effective damping matrix, $\mathbf{K}_{eff}$ the effective stiffness matrix and $\mathbf{Q}_{eff}$ the effective steady-state error elimination matrix of the system. The dimensions of all the matrices are $n \times n$ and the vectors $n \times 1$. If the system in Equation (19) is written in state-space form as in Equation (16), its state-space matrices would be:

$$\mathbf{x} = \begin{bmatrix} \int \mathbf{r}\, dt \\ \mathbf{r} \\ \dot{\mathbf{r}} \end{bmatrix} \quad , \quad \dot{\mathbf{x}} = \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \\ \ddot{\mathbf{r}} \end{bmatrix} \quad , \quad \mathbf{A} = \begin{bmatrix} \mathbf{Q}_{eff} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{eff} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{eff} \end{bmatrix} \quad , \quad \mathbf{B} = \begin{bmatrix} \mathbf{-K}_{eff} & \mathbf{-C}_{eff} & \mathbf{-M}_{eff} \\ \mathbf{K}_{eff} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{eff} & \mathbf{0} \end{bmatrix} \tag{20}$$

where the dimensions of $\mathbf{x}$ and $\dot{\mathbf{x}}$ are $3n \times 1$, and those of $\mathbf{A}$ and $\mathbf{B}$ are both $3n \times 3n$.

4. Numerical Examples

In this chapter, three examples for assessing the validity of the theory derived in Chapters 2 and 3 are presented. The first example focuses on verifying the pattern for adding the controller gains into the system matrices for systems containing non-collocated sensors and actuators. The second example aims at verifying the proposed properties of a system containing a term proportional to the time integral of the position, i.e. steady-state error elimination. In the last example, the objective is to verify the derived theory for a system by combining the theory derived in Chapters 2 and 3. This is of importance when considering the implementation of the derived theory into an FE software system.

4.1. Numerical Examples for Non-Collocated Sensors and Actuators

In order to test the theory derived in Chapter 2, some numerical examples were made based on the active system in Figure 1. For each version of the system, two separate methods for deriving the natural frequencies of the system were used: (*a*) modal analysis by solving the eigenvalue problem, and (*b*) frequency analysis of the time responses using the fast Fourier transform (FFT) algorithm. By comparing the results from the two methods (*a*) and (*b*), the degree of validity of the derived theory should be revealed, meaning that if both methods yield concurring results, the validity of the method is supported. The mechanical properties of the system were set to: $m_1 = m_2 = m_3 = 1$ kg, $c_1 = c_2 = c_3 = 0$ Ns/m and $k_1 = k_2 = k_3 = 100$ N/m, while the controller was set to be a velocity feedback PID controller with controller gains $K_p = 0$, $K_i = 50$ and $K_d = 0.5$. Damping, both passive and active, were deliberately not included in this example in order to better see all frequency peaks in the FFT plots. To derive the natural frequencies using the FFT, the mass m_3 was given an initial deflection and the time domain response of the mechanism was recorded. This simulation was carried out in FEDEM, with a total simulation time of 10 seconds and a simulation time increment of 0.001 seconds, yielding a frequency sampling rate f_s of 1 000 Hz and a frequency resolution of approximately 0.1 Hz. The result from the FFT is shown in Figure 4.

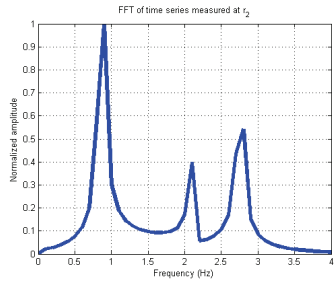


Figure 4: FFT plot of time series of displacements measured at r_2.

As can be seen in Figure 4, three frequency peaks are present, appearing at $\omega_1 = 0.9$ Hz, $\omega_2 = 2.1$ Hz and $\omega_3 = 2.8$ Hz.

Inserting the given values for the active system into Equation (8) gives the following equation:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.5 & 0 & 1 \end{bmatrix} \begin{bmatrix} \ddot{r}_1 \\ \ddot{r}_2 \\ \ddot{r}_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \dot{r}_3 \end{bmatrix} + \begin{bmatrix} 200 & -100 & 0 \\ -100 & 200 & -100 \\ 50 & -100 & 100 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \mathbf{0} \qquad (21)$$

which, by solving the eigenvalue problem, yields the eigenfrequencies $\omega_1 = 0.8613$ Hz, $\omega_2 = 2.0795$ Hz and $\omega_3 = 2.7566$ Hz. A comparison of the results from the FFT and the modal analysis is shown in Table 1.

Table 1: Comparison of eigenfrequencies derived using FFT and modal analysis.

	FFT	Modal analysis
ω_1	0.9 Hz	0.8613 Hz
ω_2	2.1 Hz	2.0795 Hz
ω_3	2.8 Hz	2.7566 Hz

As can be seen in Table 1, the two methods yield concurrent eigenfrequency estimates for the system, thereby indicating a validity of the derived theory. To better distinguish between the results, the eigenfrequencies derived by the modal analysis are given with four decimals, while the FFT is only given with one since the FFT only has a frequency resolution of 0.1 Hz.

In order to further support the validity of the derived theory, two more tests were performed on the given system, although with a slight alteration: K_d was changed to 0.3 and 0.7. The FFT plots for these tests are shown in Figure 5 and Figure 6, and the results from the tests are shown in Table 2 and Table 3, respectively. For the modal analysis of these tests, the relevant values for K_d were altered in Equation (21), and the eigenvalue problem was solved. The results from the modal analyses are also shown in Table 2 and Table 3.

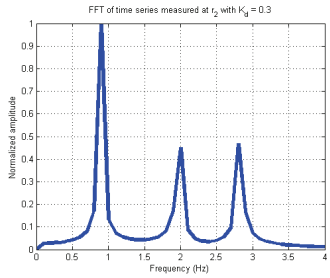


Figure 5: FFT plot of time series of displacements measured at r_2 for system with $K_d = 0.3$.

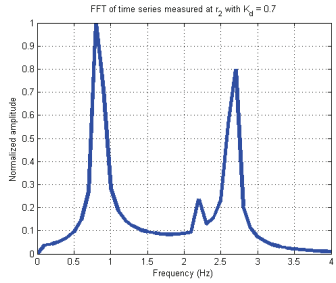


Figure 6: FFT plot of time series of displacements measured at r_2 for system with $K_d = 0.7$.

Table 2: Comparison of eigenfrequencies derived using FFT and modal analysis for system with $K_d = 0.3$.

	FFT	Modal analysis
ω_1	0.9 Hz	0.8852 Hz
ω_2	2.0 Hz	1.9739 Hz
ω_3	2.8 Hz	2.8259 Hz

Table 3: Comparison of eigenfrequencies derived using FFT and modal analysis for system with $K_d = 0.7$.

	FFT	Modal analysis
ω_1	0.8 Hz	0.8400 Hz
ω_2	2.2 Hz	2.2093 Hz
ω_3	2.7 Hz	2.6606 Hz

As seen in both Table 2 and Table 3, the FFT and the eigenvalue problem yield concurrent results, which further support a validity of the derived theory. As before, the difference between the results is probably due to the number of given decimals.

4.2. Numerical Examples for Active System Containing Damping and Steady-State Error Elimination

In order to test the theory derived in Chapter 3, four experiments involving an active SDOF system were conducted. Figure 7 depicts a sketch of the experiment setup.

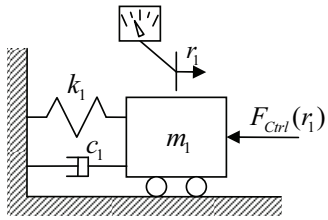


Figure 7: Active SDOF system.

The system in Figure 7 consists of a mass m_1 connected to a wall via a spring k_1 and a damper c_1. There is one DOF in the system: translation in the horizontal direction (r_1). An active force F_{Ctrl} is acting on the mass m_1; the active force is governed by a position feedback PID controller whose reference is the position r_1 of the mass. The parameters of the system were given as: $m_1 = 1$ kg, $c_1 = 8$ Ns/m, $k_1 = 12$ N/m, $K_p = 4$, $K_i = q$ and $K_d = 2$. Based on Equation (1), the effective mass m, damping c and stiffness k of the system are: $m = 1$ kg, $c = 10$ Ns/m and $k = 16$ N/m, yielding an undamped natural frequency of:

$$\omega_n = \sqrt{\frac{k}{m}} = \sqrt{\frac{16}{1}} = 4 \text{ rad/sec} = 0.6366 \text{ Hz} \tag{22}$$

and a value for the critical damping of:

$$c_c = \sqrt{4mk} = \sqrt{4 \cdot 1 \cdot 16} = 8 \text{ Ns/m} \tag{23}$$

A comparison of the values for c and c_c reveals that, in principle at least, the system is an overdamped one, i.e. no oscillation should occur. However, increasing K_i should make the system start to oscillate. Using Equation (15), the value for K_i that should give a constant oscillation is:

$$q_s = c\frac{k}{m} = 10 \cdot \frac{16}{1} = 160 \text{ N/ms} \tag{24}$$

Four different experiments were performed on this system by changing the value for the integral gain K_i from 0 to 80, 160 and 240, which should yield the following systems: overcritically damped, undercritically damped, undamped/marginally stable and unstable, respectively. The objectives of the experiments were to illustrate how K_i affects the system and to derive the eigenfrequencies and damping ratios for each case using the method outlined in Chapter 3. Additionally, in order to verify the theory, one time simulation of the free vibration of the system was performed for each case. The simulations were carried out in FEDEM; the system was set into motion by giving the mass an initial displacement. Each simulation had a total simulation time of 5 seconds, with a time increment of 0.0005 seconds.

4.2.1. $K_i = 0$: *Overcritically Damped System*

The time response for the simulation with $K_i = 0$ is shown in Figure 8.

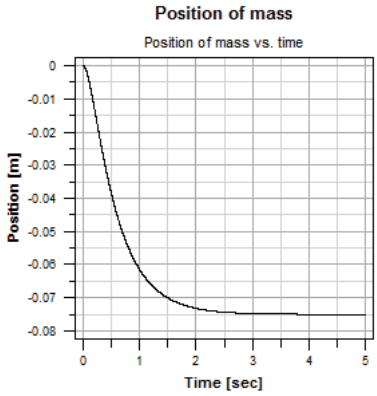


Figure 8: Time response of system with $K_i = 0$.

As seen from Figure 8, the system is not oscillating. Solving Equation (18) in MATLAB using the `eig()` routine yields a diagonal matrix **D** of generalized eigenvalues and a full matrix **V**, whose columns are the corresponding eigenvectors. The diagonal elements in **D** correspond to the roots s of Equation (13) when using the system shown in Equation (17). Hence, solving Equation (13) with respect to s, or Equation (18) inserted for **A** and **B** in accordance with Equation (17), yields the eigenvalues as three real and unequal values, thereby indicating a system without oscillations.

4.2.2. $K_i = 80$: *Undercritically Damped System*

The time response for the simulation with $K_i = 80$ is shown in Figure 9.

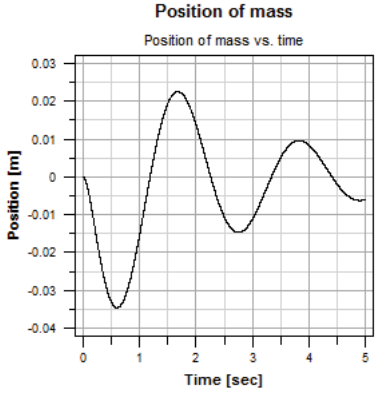


Figure 9: Time response of system with $K_i = 80$.

As seen from Figure 9, the system is oscillating with decreasing amplitudes. Peaks 1 and 2 occur at time t_1 = 1.6770 sec and t_2 = 3.8280 sec with amplitudes x_1 = 0.0225356 m and x_2 = 0.0095886 m, respectively. This yields a time period of:

$$\tau = t_2 - t_1 = 3.8280 - 1.6770 = 2.1510 \text{ sec} \tag{25}$$

and an eigenfrequency of:

$$\omega = \frac{1}{\tau} = \frac{1}{2.1510} = 0.4649 \text{ Hz} \tag{26}$$

The damping ratio ζ of an oscillation can be derived from [10]:

$$\delta = \ln\left(\frac{x_1}{x_2}\right) = \ln e^{\zeta\omega_n\tau} \Rightarrow \zeta = \frac{\delta}{\omega_n\tau} = \frac{\ln\left(\frac{x_1}{x_2}\right)}{\omega_n\tau} \tag{27}$$

Inserting for x_1 and x_2 yields the following damping ratio:

$$\zeta = \frac{\ln\left(\frac{x_1}{x_2}\right)}{\omega_n\tau} = \frac{\ln\left(\frac{0.0225356}{0.0095886}\right)}{4 \cdot 2.1510} = 0.0993 \tag{28}$$

As in Section 4.2.1, solving Equation (13) with respect to *s*, or Equation (18) inserted for **A** and **B** in accordance with Equation (17), yields the eigenvalues as one real value and a pair of complex conjugate values. The complex conjugate values are:

$$s = -0.3970 \pm 2.9210i \tag{29}$$

which gives the eigenfrequency:

$$\omega = \frac{2.9210}{2\pi} = 0.4649 \text{ Hz} \tag{30}$$

and the damping ratio:

$$\zeta = -\frac{-0.3970}{\omega_n} = -\frac{-0.3970}{4} = 0.0993 \tag{31}$$

The results from this experiment are summarized in Table 4.

Table 4: Results from experiments on system with $q = 80$.

	ω [Hz]	ζ
Time simulation	0.4649	0.0993
Eigenvalue problem	0.4649	0.0993

As seen from the results presented in Table 4, the time simulation and solution of the eigenvalue problem yield identical results.

4.2.3. *$K_i = 160$: Marginally Stable System*

The time response for the simulation with $K_i = 160$ is shown in Figure 10.

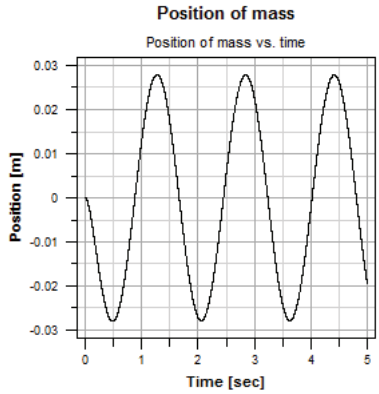


Figure 10: Time response of system with $K_i = 160$.

As seen from Figure 10, the system is oscillating with a constant amplitude. Peaks 1 and 2 occur at time $t_1 = 1.2735$ sec and $t_2 = 2.8440$ sec with amplitudes $x_1 = 0.0278458$ m and $x_2 = 0.0278311$ m, respectively. This yields a time period of:

$$\tau = t_2 - t_1 = 2.8440 - 1.2735 = 1.5705 \text{ sec} \tag{32}$$

and an eigenfrequency of :

$$\omega = \frac{1}{\tau} = \frac{1}{1.5705} = 0.6367 \text{ Hz} \tag{33}$$

Inserting for x_1 and x_2 yields the following damping ratio:

$$\zeta = \frac{\ln\left(\frac{x_1}{x_2}\right)}{\omega_n \tau} = \frac{\ln\left(\frac{0.0278458}{0.0278311}\right)}{4 \cdot 1.5705} = 0.0001 \tag{34}$$

Solving Equation (13) with respect to s, or Equation (18) inserted for **A** and **B** in accordance with Equation (17), yields the eigenvalues as one real value and a pair of complex conjugate values. The complex conjugate values are:

$$s = 0 \pm 4i \tag{35}$$

which gives the eigenfrequency:

$$\omega = \frac{4}{2\pi} = 0.6366 \text{ Hz} \tag{36}$$

and the damping ratio:

$$\zeta = -\frac{0}{\omega_n} = -\frac{0}{4} = 0 \tag{37}$$

The results from this experiment are summarized in Table 5.

Table 5: Results from experiments on system with q = 160.

	ω [Hz]	ζ
Time simulation	0.6367	0.0001
Eigenvalue problem	0.6366	0

As seen from the results presented in Table 5, the time simulation and solution of the eigenvalue problem yield almost identical results. The difference in the results between the time simulation and eigenvalue problem is probably due to the limited numerical accuracy of the time simulation.

4.2.4. *K_i = 240: Unstable System*

The time response for the simulation with K_i = 240 is shown in Figure 11.

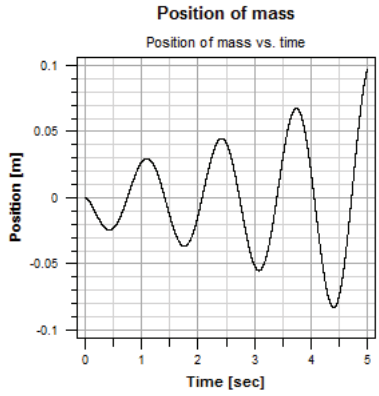


Figure 11: Time response of system with K_i = 240.

As seen from Figure 11, the system is oscillating with increasing amplitudes. Peaks 1 and 2 occur at time $t_1 = 1.0935$ sec and $t_2 = 2.4180$ sec with amplitudes $x_1 = 0.0297421$ m and $x_2 = 0.0448529$ m, respectively. This yields a time period of:

$$\tau = t_2 - t_1 = 2.4180 - 1.0935 = 1.3245 \text{ sec} \tag{38}$$

and an eigenfrequency of :

$$\omega = \frac{1}{\tau} = \frac{1}{1.3245} = 0.7550 \text{ Hz} \tag{39}$$

Inserted for x_1 and x_2 yields the following damping ratio:

$$\zeta = \frac{\ln\left(\frac{x_1}{x_2}\right)}{\omega_n \tau} = \frac{\ln\left(\frac{0.0297421}{0.0448529}\right)}{4 \cdot 1.3245} = -0.0775 \tag{40}$$

Solving Equation (13) with respect to s, or Equation (18) inserted for **A** and **B** in accordance with Equation (17), yields the eigenvalues as one real value and a pair of complex conjugate values. The complex conjugate values are:

$$s = 0.3105 \pm 4.7434i \tag{41}$$

which gives the eigenfrequency:

$$\omega = \frac{4.7434}{2\pi} = 0.7549 \text{ Hz} \tag{42}$$

and the damping ratio:

$$\zeta = -\frac{0.3105}{\omega_n} = -\frac{0.3105}{4} = -0.0776 \tag{43}$$

The results from this experiment are summarized in Table 6.

Table 6: Results from experiments on system with $q = 240$.

	ω [Hz]	ζ
Time simulation	0.7550	-0.0775
Eigenvalue problem	0.7549	-0.0776

As can be seen from the results presented in Table 6, the time simulation and solution of the eigenvalue problem yield almost identical results. Again, the difference in the results between the time simulation and eigenvalue problem is probably due to the limited numerical accuracy of the time simulation.

4.2.5. *Varying q from 0 to $2q_s$*

When solving Equation (13) with respect to s or Equation (18) inserted for **A** and **B** in accordance with Equation (17), the effects of varying q can be seen over a greater range of values. The eigenfrequency ω and damping ratio ζ for $0 \le q \le 2q_s = 0 \ldots 320$ are shown in Figure 12.

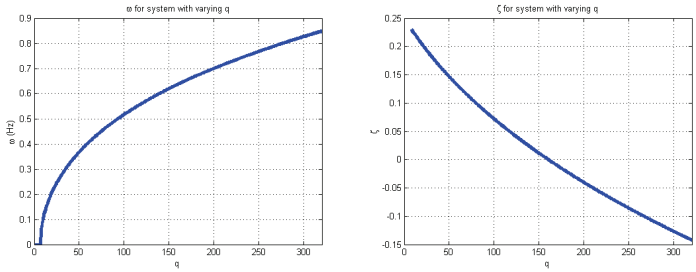


Figure 12: ω and ζ for $0 \le q \le 2q_s = 0 \ldots 320$.

As seen in Figure 12, the eigenfrequency ω goes from 0 Hz to 0.8481 Hz while the damping ratio ζ goes from 0.2301 to -0.1427. For $q < 8$, $\omega = 0$ Hz, indicating a system with a non-oscillatory motion, i.e. an overcritically damped system. For this reason, ζ does not have any values for $q < 8$. For $q > 160$, $\zeta < 0$, which indicates a system with "negative damping", i.e. a growing oscillation and thus an unstable system.

4.3. Active MDOF System with Position Feedback PID Controller and Non-Collocated Sensor and Actuator

To further test the theory derived in Chapters 2 and 3, the system shown in Figure 13 was used. To illustrate the effects of the controller, two versions of the system in Figure 13 were made: one with and one without the controller, i.e. a passive and an active system, respectively.

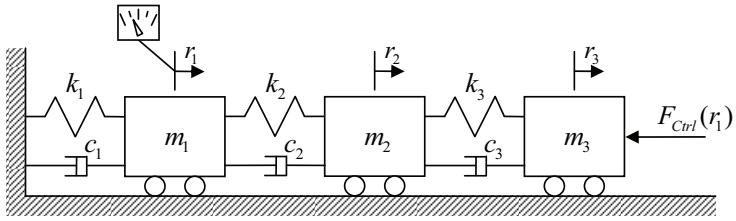


Figure 13: Active MDOF system with position feedback PID controller with non-collocated sensor and actuator.

The system in Figure 13 is comprised of three masses (m_1, m_2 and m_3) in series connected by springs (k_1, k_2 and k_3) and dampers (c_1, c_2 and c_3). Each mass has one DOF: translation in the horizontal plane, named r_1, r_2 and r_3, respectively. The system is controlled by a position feedback PID controller with a sensor measuring position of r_1, whereas an actuator is affecting mass m_3 as a force F_{Ctrl}. The mechanical properties of the system were set to:

$m_1 = m_2 = m_3 = 1$ kg, $c_1 = c_2 = c_3 = 1$ Ns/m and $k_1 = k_2 = k_3 = 100$ N/m, and the controller gains to: $K_p = 80$, $K_i = 20$ and $K_d = 0.5$. Arranging these system properties in matrix form based on Equation (5), and expressing the position vector as: $\mathbf{r} = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}^T$ and the velocity and acceleration vectors in a similar manner, the matrices of the mechanical system were:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad \mathbf{C} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \quad , \quad \mathbf{K} = \begin{bmatrix} 200 & -100 & 0 \\ -100 & 200 & -100 \\ 0 & -100 & 100 \end{bmatrix} \tag{44}$$

while the matrices of the controller gradients were:

$$\mathbf{G}_{Acc} = \mathbf{0} \quad , \quad \mathbf{G}_{Vel} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.5 & 0 & 0 \end{bmatrix} \quad , \quad \mathbf{G}_{Pos} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 80 & 0 & 0 \end{bmatrix} \quad , \quad \mathbf{G}_{SSEE} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 20 & 0 & 0 \end{bmatrix} \tag{45}$$

For the passive version of the system, only the mechanical properties of the system were included in the eigenvalue problem. This yields the following eigenfrequencies: $\omega_1 = 0.7081$ Hz, $\omega_2 = 1.9808$ Hz and $\omega_3 = 2.8562$ Hz.

To derive the eigenfrequencies of the active system in Figure 13 with the given properties, the generalized eigenvalue problem shown by Equation (18) was solved. Based on Equation (20), the **A** and **B** matrices for Equation (18) were:

$$\mathbf{A} = \left[\begin{array}{ccc:ccc:ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 200 & -100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -100 & 200 & -100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 80 & -100 & 100 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \tag{46}$$

$$\mathbf{B} = \left[\begin{array}{ccc|ccc|ccc} -200 & 100 & 0 & -2 & 1 & 0 & -1 & 0 & 0 \\ 100 & -200 & 100 & 1 & -2 & 1 & 0 & -1 & 0 \\ -80 & 100 & -100 & -0.5 & 1 & -1 & 0 & 0 & -1 \\ \hline 200 & -100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -100 & 200 & -100 & 0 & 0 & 0 & 0 & 0 & 0 \\ 80 & -100 & 100 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right] \quad (47)$$

Solving Equation (18) with Equation (46) and Equation (47) inserted yields the following eigenfrequencies: $\omega_1 = 1.0783$ Hz, $\omega_2 = 1.7088$ Hz and $\omega_3 = 2.9158$ Hz.

To verify the eigenfrequencies derived for both versions of the system in Figure 13, two time-simulations of the system were performed in FEDEM, one for the passive and one for the active system. To initiate the simulations, the mass m_3 was given an initial deflection and the time domain response of the mechanism was recorded. The time simulation ran for 10 seconds with a time increment of 0.001 seconds, giving a frequency sampling rate f_s of 1 000 Hz and a frequency resolution of approximately 0.1 Hz. The time domain results from the simulations for position r_2 were transformed into frequency domain results by using the FFT algorithm. The FFT results for the passive and active systems are shown in Figure 14 and Figure 15, respectively.

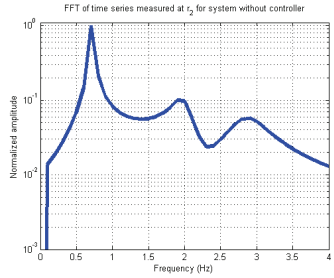


Figure 14: FFT plot of time series of displacements measured at r_2 for system without controller.

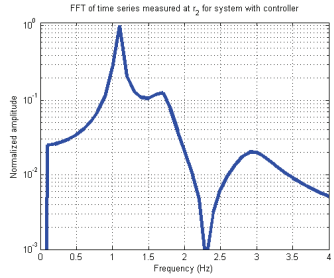


Figure 15: FFT plot of time series of displacements measured at r_2 for system with controller.

In both Figure 14 and Figure 15, three frequency peaks are present, appearing at $\omega_1 = 0.7$ Hz, $\omega_2 = 2.0$ Hz and $\omega_3 = 2.9$ Hz, and $\omega_1 = 1.1$ Hz, $\omega_2 = 1.7$ Hz and $\omega_3 = 2.9$ Hz, respectively. The results from the simulations and the modal analyses are shown in Table 7.

Table 7: Results from FFT and modal analysis for the passive and active version of the system in Figure 13.

	Without controller		With controller	
	FFT	Modal analysis	FFT	Modal analysis
ω_1	0.7 Hz	0.7081 Hz	1.1 Hz	1.0783 Hz
ω_2	2.0 Hz	1.9808 Hz	1.7 Hz	1.7088 Hz
ω_3	2.9 Hz	2.8562 Hz	2.9 Hz	2.9158 Hz

As seen from the results in Table 7, there is a close correlation between the FFT and the modal analysis for both versions of the system in Figure 13, thus indicating a validity of the modal analyses. Also, changing the system from passive to active greatly alters the eigenfrequencies of the system.

5. Discussion

In Section 4.1, a total of three cases for an SDOF system containing non-collocated sensors and actuators were examined. As seen from the results presented in that section, there is a close correlation between the derived theory and the results from the tests performed, indicating a validity of the derived theory for non-collocated sensors and actuators.

In Section 4.2, four cases of an SDOF system containing a position feedback PID controller were examined, putting an emphasis on the effects on the eigenfrequency and damping ratio caused by the steady-state error elimination. The results demonstrate that the steady-state error elimination can make a non-oscillatory system become not only oscillatory, but also highly unstable. Both eigenfrequency and damping ratio were shown to be affected by the steady-state error elimination term, indicating the importance of including these terms when performing modal analysis of systems containing such effects. Moreover, the presented method for deriving eigenvalues and eigenvectors for such systems using a $3n$ state-space formulation appears to be valid since all results derived using the state-space method concur well with the results derived using time simulations.

In Section 4.3, one example utilizing the theory derived in both Chapters 2 and 3 was given. The system in the example contained a position feedback PID controller with a non-collocated sensor and actuator. Two tests were run on the system, one with and one without the controller being activated. The results show that the proposed eigenvalue solution method yields concurring results compared with results derived using Fourier transforms of the time simulations. They also reveal the numerical difference in the derived eigenvalues for the passive and active system, once again highlighting the importance of including all properties in the system model when performing modal analysis of active systems.

One major concern about the proposed 3*n* state-space method is the threefold increase in dimensions of the eigenvalue problem. A typical method or algorithm for solving the full eigenvalue problem in FE software systems is the QR [10, 20-26] or QZ algorithm [23, 24]; the QZ algorithm being a generalization the QR algorithm [23]. The QR algorithm is of order n^3 [20]. For the proposed 3*n* state-space method, this would mean an $(3n)^3/n^3$ increase in computation time for systems of large *n*, which means that solving the eigenvalue problem using the proposed 3*n* state-space method will be up to 27 times more expensive with respect to computational time than an *n*-space method.

It is worth mentioning that the proposed 3*n* state-space method given by Equation (20) has not been considered as being optimized with respect to computational efficiency. There may also be other and more computational cost effective and/or well-conditioned ways of expressing that equation. The main effort in this work has been directed at deriving a functional expression, not an optimal one.

The proposed 3*n* state-space method is intended as a complement to the *n*-space method proposed by the authors in [3]. The proposed *n*-space method takes its basis in the same methods as the presented 3*n* state-space method, i.e. Equations (4) and (5), respectively, and is able to handle controller properties equivalent to mechanical mass and stiffness, but can be expanded to also include proportional/Rayleigh damping. The *n*-space method should be easier to implement in an FE software system and is more computationally effective than the presented 3*n* state-space method; however, it does not handle non-proportional damping or steady-state error elimination. For situations in which solving speed is more important than solution accuracy, or the effects by damping and steady-state error elimination are negligible with respect to the modal parameters, the *n*-space method can be used. Yet, if solution accuracy is an issue, or a stability analysis is desired, the presented 3*n* state-space method may be used. By offering these complementary methods for deriving the modal parameters of active flexible multibody systems, the ability to perform modal analyses of such systems for engineers working in an FE environment may be greatly improved.

6. Conclusion

In this work, a method for solving the eigenvalue problem for active multiple degrees of freedom systems containing position feedback PID controllers and non-collocated sensors and actuators has been derived and verified through numerical examples. The derived theory is intended to be implemented in a finite element software system, providing a powerful and accurate tool for engineers working in a finite element environment when performing modal analysis of active flexible multibody systems.

Acknowledgements

The authors would like to acknowledge the guidance and assistance of Professor Ole Ivar Sivertsen and Professor Kristian Tønder at the Norwegian University of Science and Technology

(NTNU). The authors would also like to acknowledge the financial support from the Research Council of Norway and the other partners in the Lean Product Development (LPD) Project.

References

[1] Géradin M, Cardona A. Flexible Multibody Dynamics: A Finite Element Approach. Chichester, England: John Wiley & Sons, Ltd.; 2001.
[2] Sivertsen OI. Virtual Testing of Mechanical Systems - Theories and Techniques. Lisse, The Netherlands: Swets & Zeitlinger B.V.; 2001.
[3] Bratland M, Haugen B, Rølvåg T. Modal analysis of active flexible multibody systems. Comput Struct. 2011;89:750-61.
[4] Palm WJ. Mechanical Vibration. Hoboken, NJ, USA: John Wiley & Sons, Inc.; 2007.
[5] Bratland M, Rølvåg T. Modal Analysis of Lumped Flexible Active Systems (Part 1). In: Proceedings of SIMS 2008: The 48th Scandinavian Conference on Simulation and Modeling. Oslo, Norway; 2008.
[6] Sharon A, Hogan N, Hardt DE. Controller design in the physical domain. J Franklin Inst. 1991;328:697-721.
[7] Bernzen W. Active vibration control of flexible robots using virtual spring-damper systems. J Intell Robot Syst, Theory Appl. 1999;24:69-88.
[8] Ryu J-H, Kwon D-S, Hannaford B. Stability guaranteed control: Time domain passivity approach. IEEE Trans Control Syst Technol. 2004;12:860-8.
[9] Tisseur F, Meerbergen K. The quadratic eigenvalue problem. SIAM Rev. 2001;43:235-86.
[10] Thomson WT, Dahleh MD. Theory of Vibration with Applications. 5th ed. Upper Saddle River, NJ, USA: Prentice Hall, Inc.; 1998.
[11] Ewins DJ. Modal testing: theory, practice and application. Baldock, England: Research Studies Press; 2000.
[12] Adhikari S. Rates of change of eigenvalues and eigenvectors in damped dynamic system. AIAA J. 1999;37:1452-8.
[13] Foss KA. Coordinates which uncouple equations of motion of damped linear dynamic systems. ASME Meeting A-86, Dec 1-6 1957. New York, NY, USA: American Society of Mechanical Engineers (ASME); 1957. p. 4.
[14] Rastgaar MA, Ahmadian M, Southward SC. Orthogonal eigenstructure control with non-collocated actuators and sensors. J Vib Control. 2009;15:1019-47.
[15] Astrom KJ, Hagglund T. The future of PID control. Control Eng Pract. 2001;9:1163-75.
[16] Astrom KJ, Hagglund T. Revisiting the Ziegler-Nichols step response method for PID control. J Process Control. 2004;14:635-50.
[17] The MathWorks, Inc. Eigenvalues and eigenvectors - MATLAB. 28 Sept. 2010. http://www.mathworks.com/help/techdoc/ref/eig.html.
[18] Sivertsen OI, Waloen AO. Non-Linear Finite Element Formulations for Dynamic Analysis of Mechanisms with Elastic Components. Washington, DC, USA: ASME, New York, NY, USA; 1982. p. 7.
[19] Burnside WS, Panton AW. The theory of equations: with an introduction to the theory of binary algebraic forms. London1886.
[20] Bell K. Eigensolvers for structural problems: some algorithms for symmetric eigenvalue problems and their merits. Delft, The Netherlands: Delft University Press; 1998.
[21] Francis JGF. The QR Transformation A Unitary Analogue to the LR Transformation—Part 1. The Computer Journal. 1961;4:265-71.
[22] Francis JGF. The QR Transformation—Part 2. The Computer Journal. 1962;4:332-45.
[23] Moler CB, Stewart GW. An algorithm for generalized matrix eigenvalue problems. SIAM Journal on Numerical Analysis. 1973;10:241-56.
[24] Golub GH, Van Loan CF. Matrix computations. 2nd ed. Baltimore, MD, USA: Johns Hopkins University Press; 1989.
[25] Kreyszig E. Advanced Engineering Mathematics. 8th ed. New York, NY, USA: John Wiley & Sons, Inc.; 1999.
[26] Bathe K-J. Finite Element Procedures. Englewood Cliffs, NJ, USA: Prentice Hall; 1996.

PAPER IV

Bratland, M., Haugen, B., and Rølvåg, T., "A Method for Controller Parameter Estimation Based on Perturbations," *Multibody System Dynamics*, 2011, (submitted).

A Method for Controller Parameter Estimation Based on Perturbations

Magne Bratland, Bjørn Haugen, Terje Rølvåg

Department of Engineering Design and Materials, Norwegian University of Science and Technology, Richard Birkelands veg 2B, N-7491 Trondheim, Norway

E-mail address for corresponding author: magne.bratland@ntnu.no

Abstract

Simulation and prediction of eigenfrequencies and mode shapes for active flexible multibody systems is an important task in disciplines such as robotics and aerospace engineering. A challenge is to accurately include both controller effects and flexible body dynamics in a multidisciplinary system model appropriate for modal analysis. A method for performing modal analyses of such systems in a finite element environment was recently developed by the authors. On issue is, however, that for engineers working in a finite element environment, the controller properties are not always explicitly available prior to modal analyses. The authors encountered this problem when working with the design of a particular offshore windmill. The controller for the windmill was delivered in the form of a dynamic link library (dll) from a third party provider, and when performing virtual testing of the windmill design, it was of great importance to use the "real" controller in the form of the provided dll, rather than re-model it in for instance Simulink or EASY5.

This paper presents a method for estimating the controller parameters of PID-type controllers when solving the closed-loop eigenvalue problem for active flexible multibody systems in a finite element environment. The method is based on applying incremental changes, perturbations, to relevant system variables while recording reactions from other system variables. In this work, the theory of the method is derived, and the method is tested through several numerical examples.

Keywords:
Modal analysis, Finite element method, Control system, Parameter estimation, Perturbation.

1. Introduction

Modal analysis and dynamic simulation of active flexible multibody systems - from now on referred to as active mechanisms - are a multidisciplinary challenge. The dynamic performance of such products is strongly dependent on an optimal interaction between the controllers and the mechanical components. An important tool in the optimization of such products is modal analysis, which predicts modal parameters, i.e. natural frequencies, mode shapes and damping ratios, for the active system. Due to the complexity of the mechanical components, both in form

and in function, it may practical to handle such systems through a finite element (FE) approach. Effective time domain dynamic simulations of multibody systems in an FE environment have been described by for instance Géradin and Cardona [1] and Sivertsen [2].

The authors have recently developed a method for performing modal analyses of active mechanisms in an FE environment [3]. In that work, the equations for the control system are expressed in second-order form, rather than in first-order or state-space form, which is typical practice in control system disciplines, see for instance [4-6]. One of the advantages of this approach is an increased compatibility with the mechanical equations, which are typically expressed in second-order form, e.g. [2,7-10], since equations determined in state-space form are difficult to transform into second-order structural dynamics equations [11]. The generalized eigenvalue problem will be of size *n*, where *n* is the number of degrees of freedom (DOFs), and traditional FE eigenvalue problem solvers can be utilized.

One remark about that method is that the controller properties have to be known explicitly prior to the modal analysis. To the best of the authors' knowledge, no commercial software system for simulation of active mechanisms fully integrates flexible multibody dynamics and control system simulation, since the equations for control systems are typically expressed in first-order form $(\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u},\ \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u})$ while they are for mechanical systems typically expressed in second-order form $(\mathbf{M}\ddot{\mathbf{r}} + \mathbf{C}\dot{\mathbf{r}} + \mathbf{K}\mathbf{r} = \mathbf{F})$. Control system software, such as MATLAB and Simulink[1], usually support both controller design and control system simulation where the mechanical system can be modeled with rigid bodies, lumped masses, inertias, springs, dampers or analytical equations. This will cause the flexible body dynamics to be predicted by very simplified models. In flexible multibody dynamics software systems, such as FEDEM[2], feedback type controllers will typically calculate loads applied to the mechanism based on feedback measurements of the system [2]. Additionally, some flexible multibody dynamics software systems also have the option of importing or communicating with the controller model as an external process, for instance through a dynamic link library (dll) or Simulink. For these reasons, the controller is comparable to a "black box" or unknown function, as seen from the mechanical part of the software system. This approach works well in a time domain analysis when the controller drives the mechanism with applied loads based on the given controller algorithms, however, a major problem occurs in modal analyses of the closed-loop system. In free vibration analysis, all loads are set to zero, which decouples the controller and mechanical model. As a result, the mechanism becomes singular in all controlled DOFs.

In order to overcome this issue, methods for identifying the controller parameters may be applied. This paper is focused on presenting a method for estimating controller parameters for systems containing either higher-order integral gains, higher-order derivative gains or a combination of proportional, integral and derivative gains, the latter often being referred to as a proportional-integral-derivative (PID) controller, the most common type of controllers in use today [13,14]. However, the method presented in this work is not limited to apply to such

[1] MATLAB and Simulink by The MathWorks, Inc.
[2] FEDEM (Finite Element in Dynamics of Elastic Mechanisms) simulation software is a multibody dynamics package distributed by Fedem Technology AS. It is based on the finite element method and uses model reduction techniques to effectively perform nonlinear time domain dynamic simulations of active flexible multibody systems [2,12].

controllers only. It may also be applied to any system containing properties corresponding to the ones listed above. An example can be the identification of the properties of a mechanical system equal to mass, damping and stiffness based on for instance position, velocity or acceleration parameters only. The objective of this work is to derive a method which can be used when performing modal analyses of active mechanisms, using FE based software systems. The software systems used in this paper are MATLAB and Simulink[3] and FEDEM[4].

2. Interaction between Mechanism and Controller

The equation of motion for a single degree of freedom (SDOF) mechanical system with a single-input single-output (SISO) feedback controller can be written as [3]:

$$m\,\ddot{r}(t) + c\,\dot{r}(t) + k\,r(t) = F_{App}(t) + F_{Ctrl}(t) \tag{1}$$

where m is the mass, c is the damping and k is the stiffness. r is the displacement of the mass m with respect to time; $\dot{r}$ and $\ddot{r}$ are the first and second time derivatives of r, i.e. velocity and acceleration of the mass m. F_{App} is the applied mechanical force and F_{Ctrl} is the force from the controller. This is in accordance with equations found in [15].

Figure 1 shows a simple block diagram used for describing a SISO feedback control system.

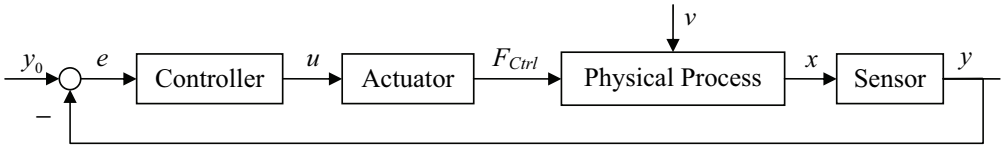


Figure 1: Block diagram for a SISO feedback control system.

In Figure 1, y_0 is the reference variable, y is the measured variable and e is the difference between y_0 and y. u is the controller output and F_{Ctrl} is a force from the controller exerted by an actuator. x is the state variable from the physical process (i.e. position r, velocity $\dot{r}$ or acceleration $\ddot{r}$), and v is the disturbance on the physical process. Only feedback controllers will be dealt with in this work, hence all control system terminology used here refers implicitly to feedback controllers.

For a feedback PID-type controller, the controller output u is given by:

$$u_{PID}(t) = K_p\,e(t) + K_i \int e(t)\,dt + K_d \frac{d}{dt} e(t) \tag{2}$$

where K_p is the proportional gain, K_d is the derivative gain and K_i is the integral gain from the controller.

[3] MATLAB and Simulink version R2010a.
[4] FEDEM version R5.0.

Since e is the difference between y_0 and y, the controller output can be split into a feedforward or feedthrough part governed by y_0 and a feedback part governed by y, as shown in [16]. The feedforward part can be interpreted as an applied force whose parameters are not affected by the system itself, and will not affect the internal dynamics of the system. Therefore, it is not of particular interest in this context. The only part which does affect the internal dynamics of the system is the feedback part. Thus, Equation (2) can more conveniently be written as:

$$u_{PID_{Feedback}}(t) = K_p\, y(t) + K_i \int y(t)\, dt + K_d \frac{d}{dt} y(t) \tag{3}$$

One view of the control system is to isolate the control elements from the physical process. The control elements then principally contain three parts: a sensor, an actuator and a controller that contains the various controller elements, as shown in Figure 2.

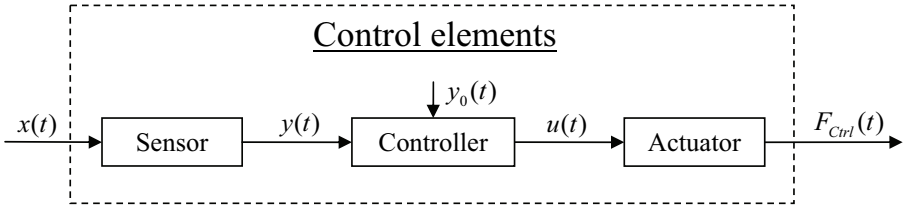


Figure 2: Control elements.

As shown in Figure 2, the effects by the control elements on the mechanical system can be given as:

$$\frac{\partial F_{Ctrl}}{\partial x} = \frac{\partial F_{Ctrl}}{\partial u}\frac{\partial u}{\partial y}\frac{\partial y}{\partial x} \quad \text{or} \quad dF_{Ctrl} = G_{Act}\, G_{Ctrl}\, G_{Sens}\, dx \tag{4}$$

where G_{Act} is the actuator gradient, G_{Ctrl} is the controller gradient and G_{Sens} is the sensor gradient. Similarly, the gradients for a multiple-input multiple-output (MIMO) system can be written as:

$$dF_{Ctrl_i} = \frac{\partial F_{Ctrl_i}}{\partial u_j}\frac{\partial u_j}{\partial y_k}\frac{\partial y_k}{\partial x_l} dx_l = G_{Act_{ij}}\, G_{Ctrl_{jk}}\, G_{Sens_{kl}}\, dx_l \tag{5}$$

or, in matrix form, as:

$$d\mathbf{F}_{Ctrl} = \frac{\partial \mathbf{F}_{Ctrl}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{x}} d\mathbf{x} = \mathbf{G}_{Act}\, \mathbf{G}_{Ctrl}\, \mathbf{G}_{Sens}\; d\mathbf{x} \tag{6}$$

3. Estimation of Controller Parameters

One of the main motivations behind this paper is to be able to perform accurate modal analyses of active mechanisms using FE based software systems. To be able to do so, the various controller gains, and hence the controller's equivalent mechanical properties, must be known. However, these values are not always explicitly available for the person performing the modal analysis since mechanical engineers and control engineers usually operate in different software systems. Consequently, a method for estimating the values of interest should be derived.

A potential method for parameter estimation is to introduce perturbations into the system. This approach is not to be confused with the perturbation method described in [7], which can be used to solve nonlinear differential equations in which the solution is in the form of a power series. Perturbations in this context are incremental changes in a system variable. The basis of this technique can be found in, for instance, the principle of virtual work [7,10,17], the displacement method/direct stiffness method [18], system identification/parameter estimation [19,20] and optimization theory [21]. For all the various fields listed above, the concept remains the same: apply changes in one variable, measure reactions from other variables and then process the results in order to derive the desired system parameters.

In this paper, perturbations will be used on the decoupled controller in order to estimate the desired controller parameters. The controller is treated as a "black box" or an unknown function. By applying incremental changes, perturbations, to the input of the controller, small changes in the output from the controller can be registered. These changes will be in accordance with the internal control routine of the controller. The parameters of the controller can thus be estimated based on predetermined changes in the controller input and registered changes from the controller output. One important feature of the proposed technique is a save-and-restore capability of the system variables. After perturbing, all system variables are reset to their pre-perturbation state in order to not affect any other simulations.

3.1. The Perturbation Technique

A perturbation, as described in the previous section, is illustrated in Figure 3.

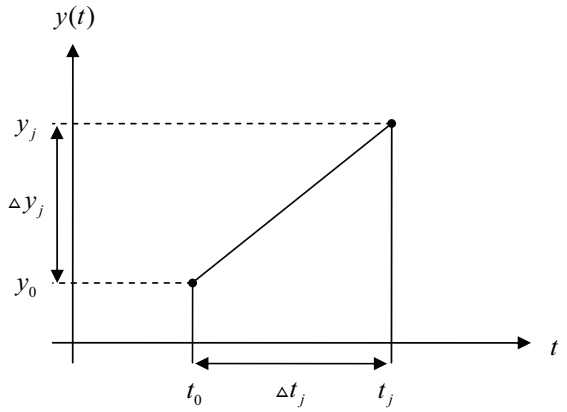


Figure 3: Perturbation *j* of *t* and *y*.

In Figure 3, the variables time *t* and controller input *y* are perturbed by the values Δy_j and Δt_j during perturbation *j*. y_0 and t_0 are the initial values for *y* and *t*, respectively, at the present time step. From Figure 3, the following relationships can be derived:

$$\Delta y_j = y_j - y_0 \Rightarrow y_j = y_0 + \Delta y_j \tag{7}$$

$$\Delta t_j = t_j - t_0 \Rightarrow t_j = t_0 + \Delta t_j \tag{8}$$

Since the controller output *u* is a function of *y* and *t*, the following equation can be given:

$$\begin{aligned} \Delta u_j &= u_j - u_0 \\ &= u_j(y_j, t_j) - u(y_0, t_0) \\ &= u_j(y_0 + \Delta y_j, t_0 + \Delta t_j) - u(y_0, t_0) \end{aligned} \tag{9}$$

The values Δy_j and Δt_j can be chosen arbitrarily, but it can be practical to express Δy_j as a function of Δt_j. The linear equation for $y_j(t)$ for perturbation *j* can then be written as:

$$y_j(t) = y_0 + \frac{\Delta y_j}{\Delta t_j} t \tag{10}$$

3.2. Estimation of Controller Parameters for Controllers Containing Proportional Gain

For a feedback type controller containing only a controller output *u* proportional to the input variable *y*, its feedback gain equation can be written as:

$$u_P(t) = K_p\, y(t) \tag{11}$$

where K_p is the proportional gain. Equation (11) can be written on a general differential form as:

$$du = \frac{\partial u}{\partial y} dy \quad \text{or} \quad du = K_p\, dy \tag{12}$$

Since the perturbation technique is meant to be used with computers, it is more suitable to treat Equation (12) numerically rather than analytically. In discrete differential form, Equation (12) can be written as:

$$\triangle u = \frac{\partial u}{\partial y} \triangle y \quad \text{or} \quad \triangle u = K_p \triangle y \tag{13}$$

K_p can thus be calculated by solving the following equation:

$$K_p = \left(\triangle y\right)^{-1} \triangle u \tag{14}$$

A perturbation algorithm for estimating K_p can be broken into six steps. Since the controller parameters may not be constant with time, the perturbation algorithm should be performed each time an eigenvalue analysis is to be performed. The steps in the perturbation algorithm are:

1) Obtain the initial values y_0 and u_0 for the controller.
2) Establish $\triangle y_j$ and $\triangle t_j$. For simplicity, $\triangle y_j$ can be given as $\triangle y_j = \triangle t_j$, making it sufficient to establish $\triangle t_j$.
3) Calculate y_j and t_j in accordance with Equation (7) and Equation (8).
4) Iterate the controller with these new values for the input y_j and time t_j, and record the reaction from the controller u_j due to the change in the input.
5) Calculate $\triangle u_j$ based on u_0 and u_j in accordance with Equation (9).
6) Use Equation (14) to estimate K_p.

3.3. Estimation of Controller Parameters for Controllers Containing Integral Gain

3.3.1. Single Integration

For a feedback type controller containing only a controller output u proportional to the time integral of the input variable y, its feedback gain equation can be written as:

$$u_I(t) = K_i \int y(t)\, dt \tag{15}$$

where K_i is the integral gain. Equation (15) can be written in discrete differential form as:

$$\triangle u = \frac{\partial u}{\partial \int y\,dt} \triangle \int y\,dt \quad \text{or} \quad \triangle u = K_i \triangle \int y\,dt \tag{16}$$

and K_i can be calculated by solving the equation:

$$K_i = \left(\triangle \int y_j\,dt \right)^{-1} \triangle u_j \tag{17}$$

In order to estimate K_i using the perturbation technique described in the previous sections, $\triangle \int y_j\,dt$ needs to be discretized. In Figure 3, $\triangle \int y_j\,dt$ is the area under the linear curve. If $y_j(t)$ is given as in Equation (10), $\triangle \int y_j\,dt$ can be made as a function of $\triangle y_j$ and $\triangle t_j$ by:

$$\triangle \int y_j\,dt = \int_0^{\triangle t_j} y_j\,dt = \left[y_0 t + \frac{1}{2} \frac{\triangle y_j}{\triangle t_j} t^2 \right]_0^{\triangle t_j} = \left(y_0 + \frac{1}{2} \triangle y_j \right) \triangle t_j \tag{18}$$

Inserting Equation (18) into Equation (17) yields:

$$K_i = \left(\left(y_0 + \frac{1}{2} \triangle y_j \right) \triangle t_j \right)^{-1} \triangle u_j \tag{19}$$

3.3.2. Double Integration

Like the single integration presented in Section 3.3.1, the feedback gain equation for a feedback type controller containing only a controller output u proportional to the double time integral of the input variable y can be written as:

$$u_{II}(t) = K_{ii} \iint y(t)\,dtdt \tag{20}$$

where K_{ii} is the double integral gain. Equation (20) can be written in discrete differential form as:

$$\triangle u = \frac{\partial u}{\partial \iint y\,dtdt} \triangle \iint y\,dtdt \quad \text{or} \quad \triangle u = K_{ii} \triangle \iint y\,dtdt \tag{21}$$

Based on antidifferentiation, the double integral $\triangle \iint y_j\,dtdt$ can be derived in discrete form as:

$$\triangle\iint y_j\,dtdt = \int_0^{\triangle t_j}\int_0^{t} y_j\,dtdt = \left[\frac{1}{2}y_0\,t^2 + \frac{1}{6}\frac{\triangle y_j}{\triangle t_j}t^3\right]_0^{\triangle t_j} = \left(\frac{1}{2}y_0 + \frac{1}{6}\triangle y_j\right)\triangle {t_j}^2 \tag{22}$$

K_{ii} can thus be estimated by solving:

$$K_{ii} = \left(\left(\frac{1}{2}y_0 + \frac{1}{6}\triangle y_j\right)\triangle {t_j}^2\right)^{-1}\triangle u_j \tag{23}$$

3.3.3. Triple Integration

Like the single and double integral presented in the previous sections, the feedback gain equation for a feedback type controller containing only a controller output u proportional to the triple time integral of the input variable y can be written as:

$$u_{III}(t) = K_{iii}\iiint y(t)\,dtdtdt \tag{24}$$

where K_{iii} is the triple integral gain. Following the same procedure as for the double integral in the previous section, the triple integral $\triangle\iiint y_j\,dtdtdt$ can be derived in discrete form as:

$$\triangle\iiint y_j\,dtdtdt = \int_0^{\triangle t_j}\int_0^{t}\int_0^{t} y_j\,dtdtdt = \left[\frac{1}{6}y_0\,t^3 + \frac{1}{24}\frac{\triangle y_j}{\triangle t_j}t^4\right]_0^{\triangle t_j} = \left(\frac{1}{6}y_0 + \frac{1}{24}\triangle y_j\right)\triangle {t_j}^3 \tag{25}$$

and K_{iii} can be estimated by solving:

$$K_{iii} = \left(\left(\frac{1}{6}y_0 + \frac{1}{24}\triangle y_j\right)\triangle {t_j}^3\right)^{-1}\triangle u_j \tag{26}$$

3.4. Estimation of Controller Parameters for Controllers Containing Derivative Gain

3.4.1. Single Derivation

For a feedback type controller containing only a controller output u proportional to the time derivative of the input variable y, its feedback gain equation can be written as:

$$u_D(t) = K_d\frac{d}{dt}y(t) = K_d\,\dot{y}(t) \tag{27}$$

where K_d is the derivative gain. Equation (27) can be written in discrete differential form as:

$$\triangle u = \frac{\partial u}{\partial \dot{y}} \triangle \dot{y} \quad \text{or} \quad \triangle u = K_d \triangle \dot{y} \tag{28}$$

K_d can be calculated by solving the equation:

$$K_d = \left(\triangle \dot{y}_j \right)^{-1} \triangle u_j \tag{29}$$

In order to estimate K_d using the perturbation technique described in the previous sections, $\triangle \dot{y}_j$ has to be discretized. The derivative in Figure 3 can be given as:

$$\dot{y}_j = \frac{\triangle y_j}{\triangle t_j} \tag{30}$$

Using Equation (7) as a basis, $\triangle \dot{y}_j$ can be given as:

$$\triangle \dot{y}_j = \dot{y}_j - \dot{y}_0 = \frac{\triangle y_j}{\triangle t_j} - \frac{\triangle y_0}{\triangle t_0} \tag{31}$$

However, $\dot{y}_0$ does not exist in Figure 3. In order to have both $\triangle \dot{y}_j$ and $\triangle \dot{y}_0$, two perturbation steps have to be performed. An example of a two-step perturbation is illustrated in Figure 4.

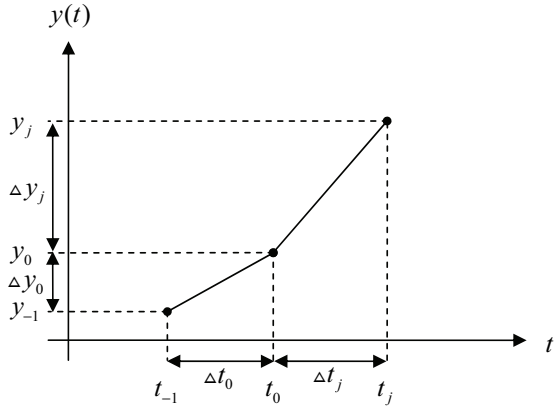


Figure 4: Two-step perturbation.

As for the one-step perturbation illustrated in Figure 3, the values $\triangle y_0$, $\triangle y_j$, $\triangle t_0$ and $\triangle t_j$ can be chosen arbitrarily, but it can be practical to express $\triangle y_j$ as a function of $\triangle t_j$, while $\triangle t_0$ can be given as $\triangle t_0 = \triangle t_j$ and $\triangle y_0 = 0$. Equation (31) can then be simplified to:

$$\Delta \dot{y}_j = \dot{y}_j - 0 = \frac{\Delta y_j}{\Delta t_j} \tag{32}$$

Inserting Equation (32) into Equation (29) yields:

$$K_d = \left(\frac{\Delta y_j}{\Delta t_j} \right)^{-1} \Delta u_j \tag{33}$$

3.4.2. Double Derivation

For a feedback type controller containing only a controller output *u* proportional to the double time derivative of the input variable *y*, its feedback gain equation can be written as:

$$u_{DD}(t) = K_{dd} \frac{d^2}{dt^2} y(t) = K_{dd}\, \ddot{y}(t) \tag{34}$$

where K_{dd} is the double derivative gain. Equation (34) can be written in discrete differential form as:

$$\Delta u = \frac{\partial u}{\partial \ddot{y}} \Delta \ddot{y} \quad \text{or} \quad \Delta u = K_{dd}\, \Delta \ddot{y} \tag{35}$$

Similarly to the discretization of $\Delta \dot{y}_j$ in Section 3.4.1, $\Delta \ddot{y}_j$ can be written as:

$$\Delta \ddot{y}_j = \ddot{y}_j - \ddot{y}_0 = \frac{\Delta \dot{y}_j}{\Delta t_j} - \frac{\Delta \dot{y}_0}{\Delta t_0} \tag{36}$$

In order to derive $\Delta \ddot{y}$, three perturbation steps have to be performed. An example of a three-step perturbation is given in Figure 5.

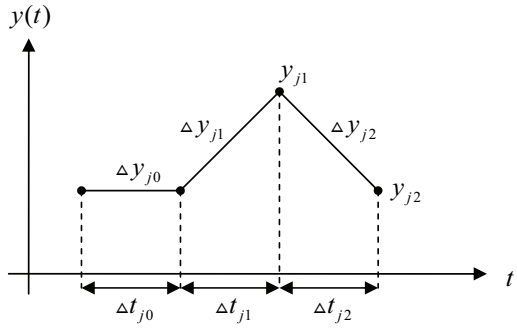


Figure 5: Three-step perturbation.

In Figure 5, $\triangle y_{j0}$ is the first perturbation step, $\triangle y_{j1}$ is the second perturbation step and $\triangle y_{j2}$ is the third perturbation step in perturbation *j*. In the figure, the following parameters are given: $\triangle t_{j0} = \triangle t_{j1} = \triangle t_{j2} = \triangle t_j$, $\triangle y_{j0} = 0$, $\triangle y_{j1} = \triangle t_j$ and $\triangle y_{j2} = -\triangle y_{j1}$. Using this three-step perturbation series with the parameters as shown in Figure 5, the variables $\triangle y_j$, $\triangle \dot{y}_j$ and $\triangle \ddot{y}_j$ can be given as:

$$\triangle y_j = \triangle y_{j2} \tag{37}$$

$$\triangle \dot{y}_j = \triangle \dot{y}_{j2} = \frac{\triangle y_{j2} - \triangle y_{j1}}{\triangle t_j} \tag{38}$$

$$\triangle \ddot{y}_j = \frac{\triangle \dot{y}_{j2} - \triangle \dot{y}_{j1}}{\triangle t_j} = \frac{\dfrac{\triangle y_{j2} - \triangle y_{j1}}{\triangle t_j} - \dfrac{\triangle y_{j1} - \triangle y_{j0}}{\triangle t_j}}{\triangle t_j} = \frac{\triangle y_{j2} - 2\triangle y_{j1} + \triangle y_{j0}}{\triangle t_j^{\,2}} \tag{39}$$

Since $\triangle y_{j0} = 0$, Equation (39) can be reduced to:

$$\triangle \ddot{y}_j = \frac{\triangle y_{j2} - 2\triangle y_{j1}}{\triangle t_j^{\,2}} \tag{40}$$

K_{dd} can thus be calculated by solving:

$$K_{dd} = \left(\frac{\triangle y_{j2} - 2\triangle y_{j1}}{\triangle t_j^{\,2}} \right)^{-1} \triangle u_j \tag{41}$$

3.5. Estimation of Controller Parameters for Controllers Containing Combinations of Proportional, Integral and Derivative Gains

For a PID controller, the feedback controller output is given by Equation (3). In a discrete differential form, this can be written as:

$$\triangle u = \frac{\partial u}{\partial y} \triangle y + \frac{\partial u}{\partial \int y\,dt} \triangle \int y\,dt + \frac{\partial u}{\partial \dot{y}} \triangle \dot{y} \quad \text{or} \quad \triangle u = K_p \triangle y + K_i \triangle \int y\,dt + K_d \triangle \dot{y} \tag{42}$$

As demonstrated in Equation (42), a PID controller is a compound controller, consisting of both a proportional gain, an integral gain and a derivative gain. In order to estimate all three gains K_p, K_i and K_d using the perturbation technique, three perturbations need to be performed. And since the controller contains a derivative gain, a two-step perturbation algorithm needs to be used, as explained in Section 3.4.1. This gives the following set of equations:

$$\begin{aligned}
\triangle u_1 &= K_p \triangle y_1 + K_i \triangle \int y_1\, dt + K_d \triangle \dot{y}_1 \\
\triangle u_2 &= K_p \triangle y_2 + K_i \triangle \int y_2\, dt + K_d \triangle \dot{y}_2 \\
\triangle u_3 &= K_p \triangle y_3 + K_i \triangle \int y_3\, dt + K_d \triangle \dot{y}_3
\end{aligned} \tag{43}$$

which can be written in matrix form as:

$$\begin{bmatrix} \triangle u_1 \\ \triangle u_2 \\ \triangle u_3 \end{bmatrix} = \begin{bmatrix} \triangle y_1 & \triangle \int y_1\, dt & \triangle \dot{y}_1 \\ \triangle y_2 & \triangle \int y_2\, dt & \triangle \dot{y}_2 \\ \triangle y_3 & \triangle \int y_3\, dt & \triangle \dot{y}_3 \end{bmatrix} \begin{bmatrix} K_p \\ K_i \\ K_d \end{bmatrix} \tag{44}$$

To derive the controller properties K_p, K_i and K_d, one can solve the following matrix system by the use of matrix inversion:

$$\begin{bmatrix} K_p \\ K_i \\ K_d \end{bmatrix} = \begin{bmatrix} \triangle y_1 & \triangle \int y_1\, dt & \triangle \dot{y}_1 \\ \triangle y_2 & \triangle \int y_2\, dt & \triangle \dot{y}_2 \\ \triangle y_3 & \triangle \int y_3\, dt & \triangle \dot{y}_3 \end{bmatrix}^{-1} \begin{bmatrix} \triangle u_1 \\ \triangle u_2 \\ \triangle u_3 \end{bmatrix} \tag{45}$$

Inserting Equation (19) and Equation (32) into Equation (45) yields:

$$\begin{bmatrix} K_p \\ K_i \\ K_d \end{bmatrix} = \begin{bmatrix} \triangle y_1 & \left(y_0 + \frac{1}{2} \triangle y_1 \right) \triangle t_1 & \frac{\triangle y_1}{\triangle t_1} \\ \triangle y_2 & \left(y_0 + \frac{1}{2} \triangle y_2 \right) \triangle t_2 & \frac{\triangle y_2}{\triangle t_2} \\ \triangle y_3 & \left(y_0 + \frac{1}{2} \triangle y_3 \right) \triangle t_3 & \frac{\triangle y_3}{\triangle t_3} \end{bmatrix}^{-1} \begin{bmatrix} \triangle u_1 \\ \triangle u_2 \\ \triangle u_3 \end{bmatrix} \tag{46}$$

To avoid singularities when performing the matrix inversion in Equation (46), the determinant of the invertible matrix should be nonzero. This requirement is met for $\triangle y_1 \neq \triangle y_2 \neq \triangle y_3$ and $\triangle t_1 \neq \triangle t_2 \neq \triangle t_3$. Typically, $\triangle t_j$ and $\triangle y_j$ can be given as:

$$\triangle t_1 = \delta \cdot \triangle t_{sim} \quad , \quad \triangle y_1 = \triangle t_1 \quad , \quad \triangle t_j = j \cdot \triangle t_1 \quad , \quad \triangle y_j = j \cdot \triangle y_1 \tag{47}$$

where $\triangle t_{sim}$ is the simulation time increment. δ is a small positive scalar called the relative perturbation step size [21]. A possible default value of δ, as used by the authors in this work, is 0.1.

A perturbation algorithm for estimating K_p, K_i and K_d can be broken down into eight steps. Since the controller parameters may not be constant with time, as mentioned in Section 3.2, the

perturbation algorithm should be performed each time an eigenvalue analysis is to be performed. The steps in the perturbation algorithm are:

1) Do one initial perturbation on the controller with $\triangle y = 0$ and $\triangle t \neq 0$. This is to ensure $\dot{y}_0 = 0$.
2) Obtain the initial values y_0 and u_0 for the controller.
3) Establish $\triangle y_j$ and $\triangle t_j$. For a PID controller, $j = 1 \ldots 3$, but if Equation (47) is used, it is sufficient to establish $\triangle t_1$.
4) Calculate $\triangle \int y_j \, dt$ and $\triangle \dot{y}_j$. These are given by Equation (18) and Equation (32).
5) Calculate y_j and t_j in accordance with Equation (7) and Equation (8).
6) Iterate the controller with these new values for the input y_j and time t_j, and record the reaction from the controller u_j due to the change in the input.
7) Calculate $\triangle u_j$ based on u_0 and u_j in accordance with Equation (9).
8) Use the matrix system in Equation (46) to estimate K_p, K_i and K_d.

3.6. Partitioning of Perturbation Steps

When testing the perturbation technique presented in Sections 3.1 to 3.5, the authors experienced a problem with some controller simulation algorithms. In some cases, the controllers were suffering from what seemed as an erroneous time step dependent delay from when a change in the input resulted in a change in the outputs. In order to extract the gradients from such a system, it may be necessary to perform a perturbation using multiple time steps, that is, introducing incremental steps in between each perturbation. Such a multistep perturbation is illustrated in Figure 6.

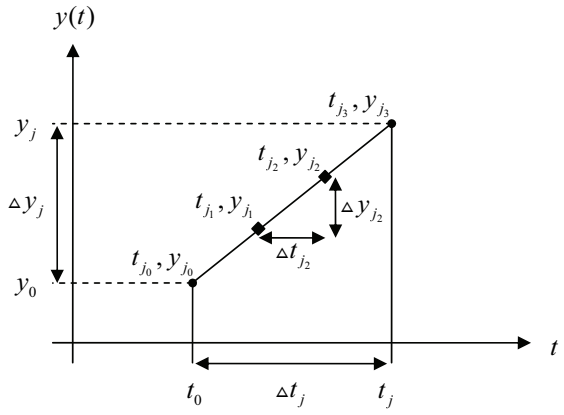


Figure 6: Partition of perturbation.

As illustrated in Figure 6, if the perturbation is divided into *n* equal sections, each perturbation step $\triangle t_j$ and $\triangle y_j$ would be comprised of *n* incremental sub-steps $\triangle t_{j_i}$ and $\triangle y_{j_i}$, where $i = 1 \ldots n$. Note that the subscripts in this section are not to be confused with the subscripting presented in Section 3.4.2. For each t_{j_i} and y_{j_i}, the system would be iterated. For the perturbation technique itself, only t_{j_n} and y_{j_n} would be used. The incremental sub-steps would then be of size:

$$\triangle t_{j_i} = \frac{1}{n} \triangle t_j \tag{48}$$

4. Testing of the Perturbation Technique

In order to verify the theory and methods derived in Chapter 3, some numerical tests were performed on three different controllers. The first controller was one containing a higher-order integral gain. The second controller contained a higher-order derivative gain, while the last controller was a compound controller containing combinations of proportional, integral and derivative gains (PID-type controller). For the first two cases, the objective was to test the derived theory by comparing it against a commercial software system, represented here by Simulink. Since the perturbation technique is meant to be implemented in FEDEM, which is a FE bases software system, the final case was focused on the accuracy of parameter estimation for variants of PID controllers.

4.1. Testing of the Perturbation Technique for Parameter Estimation of Controllers Containing Integral Gain

A comparison was made between the perturbation technique and Simulink for a system containing single, double and triple integration. For the perturbation technique, Equation (18), Equation (22) and Equation (25) were used to calculate $\triangle \int y\,dt$, $\triangle \iint y\,dtdt$ and $\triangle \iiint y\,dtdtdt$, respectively. In Simulink, this system was created by using three integration blocks in series, as shown in Figure 7.

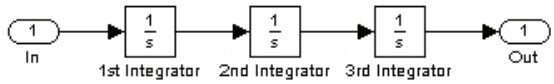


Figure 7: System of triple integration modeled in Simulink using the Continuous Integrator block.

For the triple integration system, the parameters $\triangle t_j$, $\triangle y_j$ and y_0 were given as $\triangle t_1 = 0.1$ sec, $\triangle y_1 = \triangle t_1$ and $y_0 = 0$. In Simulink, the ode4 (Runge-Kutta) solver was used. The simulation start time was set to 0.0 and the simulation stop time to 0.1, with a fixed-step size of 0.1. A comparison between the perturbation technique and the simulation results from Simulink is shown in Table 1.

Table 1: Comparison of the perturbation technique against results from the simulation in Simulink for a system with triple integration.

	Perturbation	Simulink
$\triangle t_1$	0.1	0.1
$\triangle y_1$	0.1	0.1
$\triangle \int y_1\, dt$	0.005	0.005
$\triangle \iint y_1\, dtdt$	1.6667×10^{-4}	1.6667×10^{-4}
$\triangle \iiint y_1\, dtdtdt$	4.1667×10^{-6}	4.1667×10^{-6}

As can be seen in Table 1, the perturbation technique and the Simulink simulation are identical for $\triangle \int y_1\, dt$, $\triangle \iint y_1\, dtdt$ and $\triangle \iiint y_1\, dtdtdt$.

4.2. Testing of the Perturbation Technique for Parameter Estimation of Controllers Containing Derivative Gain

A comparison was made between the perturbation technique and Simulink for a system containing single and double derivation. For the perturbation technique, Equation (38) and Equation (39) were used for $\triangle \dot{y}$ and $\triangle \ddot{y}$, respectively. In Simulink, this system was created by using two derivation blocks in series, as shown in Figure 8.

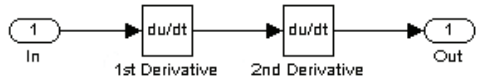


Figure 8: System of double derivation modeled in Simulink using the Continuous Derivative block.

For the double derivation system, the parameters of interest used in the simulations were $\triangle t_{j0} = \triangle t_{j1} = \triangle t_{j2} = \triangle t_j$, $\triangle y_{j0} = 0$, $\triangle y_{j1} = \triangle t_j$ and $\triangle y_{j2} = -\triangle y_{j1}$, with the value for $\triangle t_j$ given as $\triangle t_1 = 0.1$ sec. In Simulink, the ode4 (Runge-Kutta) solver was used. The simulation start time was set to 0.0 and the simulation stop time to 0.2, with a fixed-step size of 0.1. A comparison between the perturbation technique and the simulation results from Simulink is shown in Table 2.

Table 2: Comparison of the perturbation technique against results from the simulation in Simulink for a system with double derivation.

	Perturbation	Simulink
$\triangle t_1$	0.1	0.1
$\triangle y_1$	-0.1	-0.1
$\triangle \dot{y}_1$	-2	-2.0
$\triangle \ddot{y}_1$	-30	-30.0

As can be seen in Table 2, the perturbation technique and the Simulink simulation are identical for both $\triangle \dot{y}_1$ and $\triangle \ddot{y}_1$.

4.3. Testing of the Perturbation Technique for Parameter Estimation of PID Controllers

To verify the theory and method derived in Section 3.5, some basic tests were performed using the perturbation technique. The objective of the tests was to verify whether the perturbation technique could be used to estimate the controller parameters for any PID-type controller during any time step of a nonlinear dynamic time domain simulation. Since the perturbation technique is intended to be implemented in FEDEM, it is vital to test and verify the method in this software system. Additionally, one initial test of the perturbation technique for PID controllers was performed in Simulink. One possible setup for a PID controller in Simulink is shown in Figure 9.

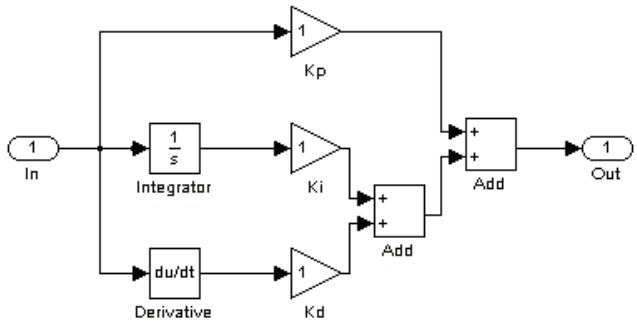


Figure 9: PID controller modeled in Simulink.

For the PID system in Simulink, the following controller parameters were used: $K_p = 1$, $K_i = 1$ and $K_d = 1$. These are the values which are to be treated as the unknown parameters of the PID controller, although in order to verify the perturbation technique, the values are in this example known a priori. For the perturbation of the PID controller based on Figure 4, the following parameters were used: $\triangle t_j$, $\triangle y_j$, $\triangle t_0$, $\triangle y_0$, t_0 and y_0 were given as $\triangle t_1 = 0.1$ sec, $\triangle t_j = j \cdot \triangle t_1$, $\triangle y_j = \triangle t_j$, $\triangle t_0 = \triangle t_1$ and $\triangle y_0 = t_0 = y_0 = 0$. The ode4 (Runge-Kutta) solver was used, and the simulation start time was set to 0.0. Three perturbations were performed ($j = 1 \ldots 3$), and for each perturbation j, the fixed-step size was set to $\triangle t_j$ and the simulation stop time to $2 \cdot \triangle t_j$. Based on Equation (46), the results from the simulation in Simulink are shown in Equation (49).

$$\begin{bmatrix} K_p \\ K_i \\ K_d \end{bmatrix} = \begin{bmatrix} 0.1000 & 0.0050 & 1.0000 \\ 0.2000 & 0.0200 & 1.0000 \\ 0.3000 & 0.0450 & 1.0000 \end{bmatrix}^{-1} \begin{bmatrix} 1.1050 \\ 1.2200 \\ 1.3450 \end{bmatrix} = \begin{bmatrix} 1.0000 \\ 1.0000 \\ 1.0000 \end{bmatrix} \tag{49}$$

As revealed in Equation (49), $K_p = 1$, $K_i = 1$ and $K_d = 1$. These estimated values for K_p, K_i and K_d are identical to the actual values for the controller gains of the PID controller, indicating a validity of the perturbation technique for such controllers.

Since the initial simulation results from Simulink indicate that the perturbation technique is valid for PID controllers, three additional tests were performed using the perturbation technique in FEDEM. As previously mentioned, the objective of these tests was to establish whether the perturbation technique could be used to estimate the controller parameters for any PID-type controller during any time step of a nonlinear dynamic time domain simulation. The setup for the tests is shown in Figure 10.

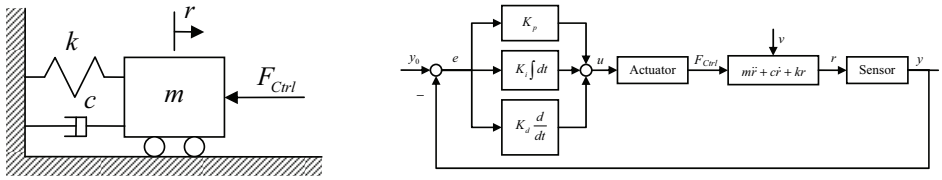


Figure 10: SDOF Mass-spring-damper system with position feedback PID controller.

The setup in Figure 10 consists of a SDOF system with mass *m*, damping *c* and stiffness *k*. There is only one DOF: position *r* of the mass. *r* is the input for the controller, which is of type PID. Since the perturbation technique is used to estimate the parameters of the controller, the parameters of the mass-spring-damper system are not of relevance in this context. The numerical values for the controller gains were arbitrarily chosen, but were deliberately given different values in order to more easily distinguish between them. The values for $\triangle t_j$ and $\triangle y_j$ were given by Equation (47), with $\delta = 0.1$. The simulation time increment $\triangle t_{sim}$ for all tests in this section was set to 0.01 seconds, i.e. $\triangle t_1 = 0.001$.

Three different tests were performed on the active mass-spring-damper system shown in Figure 10. The first test was to insure that the perturbation technique worked for controllers of any possible combination of P, I and D. The next test was to insure that the perturbation technique worked at any time step of the dynamic time domain simulation and not only at the start-up of the simulation. The last test was to insure that the perturbation technique would also work for discontinuous systems.

4.3.1. Various Controller Combinations of P, I and D

The perturbation technique should be able to yield correct estimations of the controller parameters for any PID-type controller for the possible combinations of P, I and D, including the trivial system without any controller present. To verify this, the active system in Figure 10 was set in static equilibrium, and the perturbation technique was performed on a total of eight different types of PID controllers: PID, PI, PD, ID, P, I, D and zero-gain controller. The basic values for the different gains were set to $K_p = 100$, $K_i = 20$ and $K_d = 6$, in addition to zero when not included. The results from the different perturbations are shown in Table 3. The numerical values of the estimations are given here with 12 decimals. With regard to the intended usage of the perturbation technique, correct values up to the second decimal should be more than sufficient, but in order to depict the accuracy of the method and show when the estimated values deviate from the correct ones, 12 decimals were used.

Table 3: Estimated controller parameters for different combinations of P, I and D controllers.

	K_p	K_i	K_d
PID	100.000000000005	19.999999999069	6.000000000000
PI	100.000000000000	19.999999999884	0.000000000000
PD	100.000000000000	0.000000000000	6.000000000000
ID	0.000000000000	20.000000000000	6.000000000000
P	100.000000000000	0.000000000000	0.000000000000
I	0.000000000000	20.000000000000	0.000000000000
D	0.000000000000	0.000000000000	6.000000000000
None	0.000000000000	0.000000000000	0.000000000000

The results presented in Table 3 demonstrate that the perturbation technique yields approximately correct controller parameter estimations for any PID-type controller. All but the PI and PID controller yield correct values up to the 12^{th} decimal. For the PI and PID controller, the integral gain K_i is only correct up to the 9^{th} decimal, and the proportional gain K_p for the PID controller is incorrect on the 12^{th} decimal.

4.3.2. Perturbations on PID Controller during Time Simulation with Sinusoidal Input Signal

The perturbation technique should be able to yield correct estimations of the controller parameters for any value of the initial values y_0 and u_0 for the controller, i.e. yield correct estimations of the controller parameters at any time step of the dynamic time domain simulation. To verify this, the position r of the mass m in the active system in Figure 10 was given a prescribed sinusoidal motion of 1Hz, and the perturbation technique was performed on a PID controller at various time steps. The simulation ran for one second with a time increment of 0.01 seconds. The perturbation technique was performed both at start-up and at time intervals of 0.1 seconds, giving a total of 11 different perturbations. The gains to the PID controller were set to $K_p = 100$, $K_i = 20$ and $K_d = 6$. The input signal to the controller is shown in Figure 11. The results from the simulation are shown in Table 4.

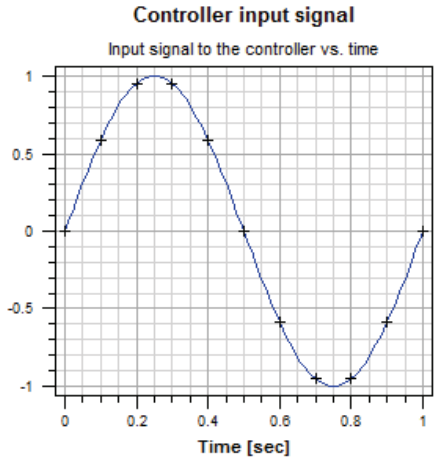


Figure 11: Input signal to the controller. The crosses mark each time the perturbation sequence is performed.

Table 4: Estimated controller parameters at different time steps for sinusoidal input signal.

Time	K_p	K_i	K_d
0.0	100.000000000005	19.9999999990687	6.00000000000000
0.1	99.999999986962	20.0000000223517	6.00000000000005
0.2	100.000000018626	19.9999999795109	5.99999999999996
0.3	99.999999991618	20.0000000083819	6.00000000000003
0.4	99.999999997206	20.0000000037253	6.00000000000003
0.5	100.000000000000	20.0000000018626	6.00000000000000
0.6	99.999999994412	19.9999999916181	5.99999999999999
0.7	100.000000019558	20.0000000204891	6.00000000000003
0.8	99.999999994412	19.9999999944121	6.00000000000002
0.9	100.000000000000	20.0000000009313	5.99999999999999
1.0	100.000000000005	19.9999999990687	6.00000000000000

As can be seen in Table 4, the perturbation technique yields correct results up to the 8th decimal at any time step for any of the controller parameters for this simulation. Both the K_p and K_i estimations are accurate up to about the 8th decimal, while the K_d estimations are accurate up to the 13th decimal. For the proportional gain, K_p, the perturbation technique yields the greatest errors at time 0.2 and 0.7 seconds, both being approximately 2.0×10^{-8}. For the integral gain, K_i, the perturbation technique yields the greatest errors at time 0.1, 0.2 and 0.7 seconds, the difference being approximately 2.5×10^{-8} for 0.1 seconds and approximately $\mp2.0\times10^{-8}$ for 0.2 and 0.7 seconds, respectively. For the derivative gain, K_d, the perturbation technique yields the greatest errors at time 0.1 and 0.2 seconds, being about 5×10^{-14} and -4×10^{-14}, respectively.

4.3.3. Perturbations on PID Controller during Time Simulation with Discontinuous Sinusoidal Input Signal

To further test the capabilities of the perturbation technique to yield correct estimations of the controller parameters for any value of the initial values y_0 and u_0 for the controller, the system

described in Section 4.3.2 was used with a discontinuous input signal. The sinusoidal signal was given a switch-to-zero-value at ±0.7, meaning it would go to zero whenever the absolute value of the input signal became larger than 0.7. As shown in Figure 12, this should occur at time 0.2, 0.3, 0.7 and 0.8 seconds. All other parameters were the same as they were in Section 4.3.2. The results from the simulation are shown in Table 5.

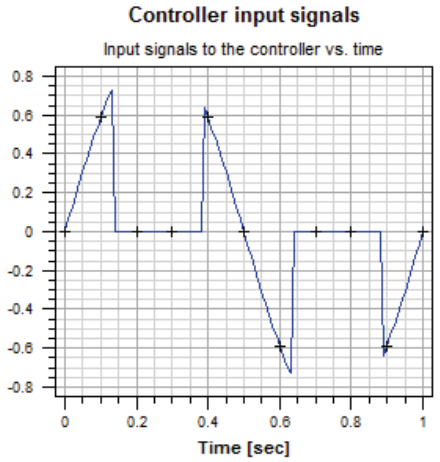


Figure 12: Input signals to the controller. The crosses mark each time the perturbation sequence is performed. The input signal is a discontinuous sinusoidal signal with switch-to-zero-value at ± 0.7.

Table 5: Estimated controller parameters at different time steps for discontinuous sinusoidal input signal.

Time	K_p	K_i	K_d
0.0	100.00000000000400	19.99999999813740	6.00000000000000
0.1	100.00000011222400	19.99999981001020	6.00000000000060
0.2	-0.00000000044409	0.00000000000000	0.00000000000222
0.3	-0.00000082446987	0.00000086508578	0.00000000000040
0.4	99.99999936856320	20.00000107102100	6.00000000000230
0.5	99.99999999535250	20.00000215135510	6.00000000000253
0.6	99.99999972432850	19.99999952781950	5.99999999999841
0.7	-0.00000004406638	-0.00000004640732	-0.00000000000005
0.8	-0.00000004406638	-0.00000004640732	-0.00000000000005
0.9	100.00000010058300	20.00000017136340	5.99999999999991
1.0	100.00000000017100	19.99999991711230	5.99999999999981

As can be seen in Table 5, the perturbation technique yields correct results up to the 6th decimal at any time step for any of the controller parameters for this simulation. Both the K_p and K_i estimations are accurate up to about the 6th decimal, while the K_d estimations are accurate up to the 12th decimal. For the K_p estimation, the greatest error is encountered at 0.3 seconds, being approximately -8×10^{-7}. For the K_i estimation, the greatest error is encountered at 0.5 seconds, being approximately 2×10^{-6}. For the K_d estimation, the greatest errors are encountered at 0.2, 0.4 and 0.5 seconds, all being approximately 2.5×10^{-12}.

5. Discussion

The main motivation behind this work is to make engineers working in an FE environment able to perform accurate modal analyses of active mechanisms. Today, FEDEM has the capability of performing accurate time domain simulations by using the controllers to drive the FE model with applied loads based on the given controller algorithms. These controller algorithms can be created either in FEDEM's Control Editor or in an external software system, such as for instance Simulink. Therefore, the controller algorithms are not required to be known for the engineer working in the FE environment. As stated in the introduction, a major obstacle for modal analysis of the closed-loop system is that in free vibration analysis all loads are set to zero, thereby resulting in a decoupling of the controller and mechanical model. By identifying the controller parameters, the controller's mechanically equivalent properties can be added to the FE model for the modal analyses, thus including both controller and mechanical properties and hence improving the accuracy of the modal analysis. The method derived in this work is intended to be implemented in FEDEM, though as presented here, it is not dependent on any particular software system.

The results presented in Sections 4.1 and 4.2 show that the perturbation technique yields correct estimations for systems containing single, double and triple integration, as well as single and double derivation compared to results derived using Simulink. This indicates validity of the perturbation technique for such systems. Controllers containing either triple integration or double derivation are not a very common type of controllers; however, in this work they do serve the purpose of testing the robustness of the perturbation technique, which only strengthens the validity of the derived method for its intended use.

The validity of the perturbation technique for PID controllers was briefly tested in Simulink. The results from that initial test demonstrated that the perturbation technique is able to correctly estimate such controllers in Simulink. More thorough tests of the technique were conducted in FEDEM, and as can be seen from the results presented in Section 4.3, the perturbation technique yields estimations for controller parameters with a highly satisfactory accuracy for any PID-type controller during any time step of a nonlinear dynamic time domain simulation in FEDEM. The greatest estimation error in any of the tests still yielded correct results up to the sixth decimal. This should be more than sufficient since a requirement for satisfactory accuracy should be correct results up to the second decimal with regard to the intended usage of the perturbation technique. Hence, for PID-type controllers, the derived method should be able to provide accurate estimations of the controller parameters.

Still, one note about the perturbation technique should be made. There has to be a correlation between the state variables of the perturbed system and those used in the perturbation technique. For instance, it can be tempting to believe that the perturbation technique is able to accurately predict the effective mass, stiffness and damping values for an active system containing a position feedback PID controller by perturbing the active system and deriving the system parameters only with respect to mass, stiffness and damping, and not including the integral gain from the controller. By ignoring some of the state variables of the perturbed system, critical and vital system information can either be lost or estimated to incorrect values, rendering the technique virtually useless for its intended use. However, when used properly, the technique has

the potential of being of great assistance in identifying the unknown parameters of a controller, such as when performing modal analysis of active mechanisms in an FE environment.

6. Conclusion

In this paper, a method for controller parameter estimation by the use of perturbations has been presented. The theory for perturbation of systems containing single, double or triple integral functions, single or double derivative functions or a combination of proportional, integral and derivative functions has been derived and tested using commercial software systems. The results from the tests reveal that the derived theory works well for all the mentioned controller variants.

If used properly, the presented technique, with its capabilities of accurate controller parameter estimation, has the potential of being a powerful tool for engineers who are conducting modal analysis of active flexible multibody systems in a finite element environment.

Acknowledgements

The authors would like to acknowledge the guidance and assistance of Professor Ole Ivar Sivertsen and Professor Kristian Tønder at the Norwegian University of Science and Technology (NTNU). The authors would also like to acknowledge the financial support from the Research Council of Norway and the other partners in the Lean Product Development (LPD) Project.

References

1. Géradin, M., Cardona, A.: Flexible Multibody Dynamics: A Finite Element Approach. John Wiley & Sons, Ltd., Chichester, England (2001)
2. Sivertsen, O.I.: Virtual Testing of Mechanical Systems - Theories and Techniques. Advances in Engineering 4. Swets & Zeitlinger B.V., Lisse, The Netherlands (2001)
3. Bratland, M., Haugen, B., Rølvåg, T.: Modal analysis of active flexible multibody systems. Comput Struct **89**(9-10), 750-761 (2011).
4. Preumont, A.: Vibration Control of Active Structures: An Introduction, 2nd ed. Kluwer Academic Publishers, Dordrecht, The Netherlands (2002)
5. Inman, D.J.: Vibration with control. John Wiley & Sons Ltd., Chichester, England (2006)
6. Balchen, J.G., Andresen, T., Foss, B.A.: Reguleringsteknikk, 5th ed. Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway (2003. (In Norwegian).)
7. Thomson, W.T., Dahleh, M.D.: Theory of Vibration with Applications, 5th ed. Prentice Hall, Inc., Upper Saddle River, NJ, USA (1998)
8. Palm, W.J.: Mechanical Vibration. John Wiley & Sons, Inc., Hoboken, NJ, USA (2007)
9. Cook, R.D., Malkus, D.S., Plesha, M.E., Witt, R.J.: Concepts and applications of finite element analysis, 4th ed. John Wiley & Sons. Inc., New York, NY, USA (2002)
10. Bathe, K.-J.: Finite Element Procedures. Prentice Hall, Englewood Cliffs, NJ, USA (1996)
11. Alvin, K.F., Park, K.C.: Second-order structural identification procedure via state-space-based system identification. AIAA J **32**, 397-406 (1994).
12. Sivertsen, O.I., Waloen, A.O.: Non-Linear Finite Element Formulations for Dynamic Analysis of Mechanisms with Elastic Components. In, Washington, DC, USA 1982. American Society of Mechanical Engineers, p. 7. ASME, New York, NY, USA
13. Astrom, K.J., Hagglund, T.: The future of PID control. Control Eng Pract **9**, 1163-1175 (2001).

14. Astrom, K.J., Hagglund, T.: Revisiting the Ziegler-Nichols step response method for PID control. J Process Control **14**, 635-650 (2004).
15. Alkhatib, R., Golnaraghi, M.F.: Active structural vibration control: a review. Shock Vib Dig **35**, 367-383 (2003).
16. Bratland, M., Rølvåg, T.: Modal Analysis of Lumped Flexible Active Systems (Part 1). Paper presented at the SIMS 2008: The 48th Scandinavian Conference on Simulation and Modeling, Oslo, Norway, 2008-10-07 - 2008-10-08
17. Park, K.C., Felippa, C.A., Ohayon, R.: The d'Alembert-Lagrange principal equations and applications to floating flexible systems. Int J Numer Meth Eng **77**, 1072-1099 (2009).
18. Felippa, C.A.: A historical outline of matrix structural analysis: A play in three acts. Comput Struct **79**, 1313-1324 (2001).
19. Astrom, K.J., Eykhoff, P.: System identification-a survey. Automatica **7**, 123-162 (1971).
20. Perreault, E.J., Kirsch, R.F., Acosta, A.M.: Multiple-input, multiple-output system identification for characterization of limb stiffness dynamics. Biol. Cybern. **80**, 327-337 (1999).
21. Trier, S.D.: Design Optimization of Flexible Multibody Systems. Doctoral Thesis, Norwegian University of Science and Technology (2001)