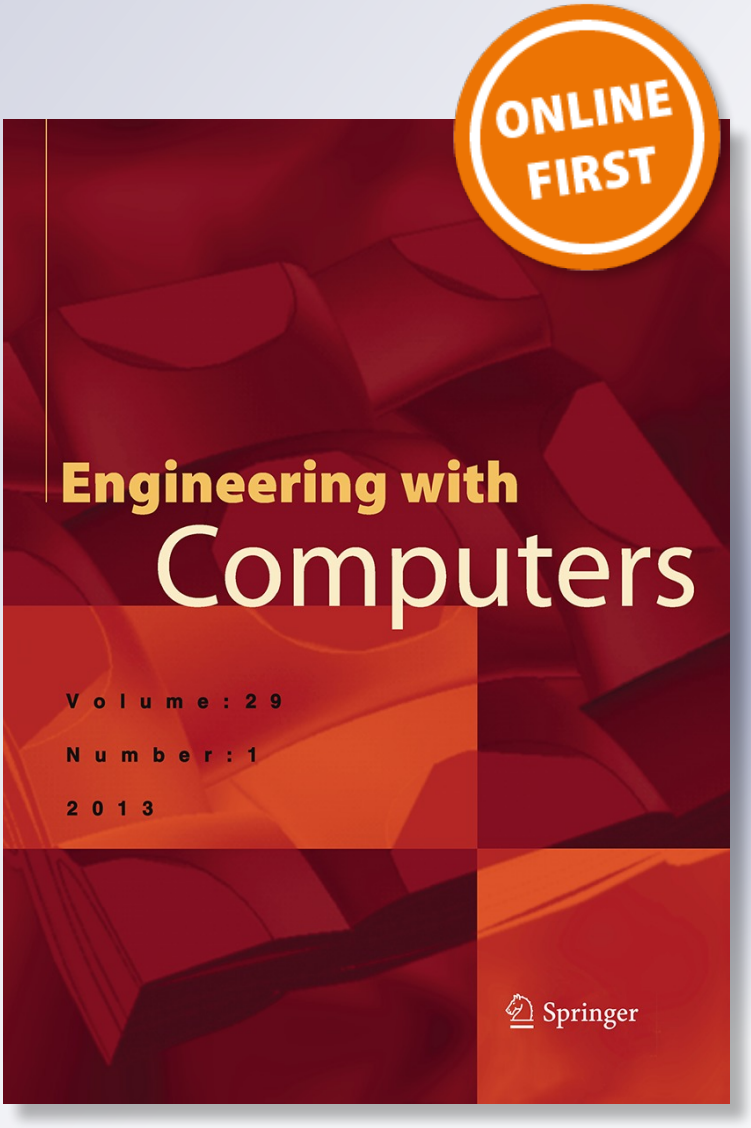
A method for controller parameter estimation based on perturbations

Magne Bratland, Bjørn Haugen & Terje Rølvåg

Engineering with Computers
An International Journal for Simulation-Based Engineering

ISSN 0177-0667

Engineering with Computers
DOI 10.1007/s00366-013-0312-3



Springer

Engineering with Computers
DOI 10.1007/s00366-013-0312-3

ORIGINAL ARTICLE

A method for controller parameter estimation based on perturbations

Magne Bratland · Bjørn Haugen · Terje Rølvåg

Received: 24 August 2011 / Accepted: 3 January 2013

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.

M. Bratland (✉) · B. Haugen · T. Rølvåg
Department of Engineering Design and Materials,
Norwegian University of Science and Technology,
Richard Birkelands veg 2B, 7491 Trondheim, Norway
e-mail: magne.bratland@ntnu.no

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 be 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.

Published online: 31 January 2013

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 ($\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$, $\mathbf{y} = \mathbf{Cx} + \mathbf{Du}$) while they are for mechanical systems typically expressed in second-order form ($\mathbf{M}\ddot{\mathbf{r}} + \mathbf{C}\dot{\mathbf{r}} + \mathbf{Kr} = \mathbf{F}$). 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]. In addition, 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 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]:

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

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.

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 (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:

$$u_{\mathrm{PID}}(t) = K_{\mathrm{p}}\,e(t) + K_{\mathrm{i}}\int e(t)\,\mathrm{d}t + K_{\mathrm{d}}\,\frac{\mathrm{d}}{\mathrm{d}t}e(t) \tag{2}$$

where K_{p} is the proportional gain, K_{d} is the derivative gain and K_{i} is the integral gain from the controller.

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, Eq. (2) can more conveniently be written as:

[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].

[3] MATLAB and Simulink version R2010a.

[4] FEDEM version R5.0.

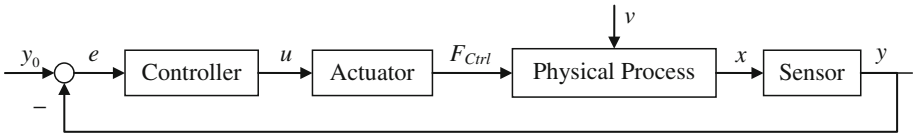


Fig. 1 Block diagram for a SISO feedback control system

$$u_{\mathrm{PID}_{\mathrm{Feedback}}}(t) = K_{\mathrm{p}}\, y(t) + K_{\mathrm{i}} \int y(t)\,\mathrm{d}t + K_{\mathrm{d}}\,\frac{\mathrm{d}}{\mathrm{d}t} y(t) \tag{3}$$

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 Fig. 2.

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

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

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:

$$\mathrm{d}F_{\mathrm{Ctrl}_i} = \frac{\partial F_{\mathrm{Ctrl}_i}}{\partial u_j}\,\frac{\partial u_j}{\partial y_k}\,\frac{\partial y_k}{\partial x_l}\,\mathrm{d}x_l = G_{\mathrm{Act}_{ij}}\, G_{\mathrm{Ctrl}_{jk}}\, G_{\mathrm{Sens}_{kl}}\,\mathrm{d}x_l \tag{5}$$

or, in matrix form, as:

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

Hence, as explained in [3], the equation of motion for the free vibration of a multiple degree of freedom (MDOF) mechanical system with a MIMO feedback controller can thus be written as:

$$\mathbf{M}\ddot{\mathbf{r}}(t) + \mathbf{C}\dot{\mathbf{r}}(t) + \mathbf{K}\mathbf{r}(t) + \mathbf{G}_{\mathrm{Act}}\mathbf{G}_{\mathrm{Ctrl}}\mathbf{G}_{\mathrm{Sens}}\,\mathbf{x}(t) = \mathbf{0} \tag{7}$$

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

The actuator gradient $\mathbf{G}_{\mathrm{Act}}$ describes the relationship between the controller forces $\mathbf{F}_{\mathrm{Ctrl}}$ exerted by the actuator and the output signals $\mathbf{u}$ from the controller, and has dimensions $n_{F_{\mathrm{Ctrl}}} \times n_u$ where $n_{F_{\mathrm{Ctrl}}}$ is the number of controller forces and n_u is the number of controller outputs. The controller gradient $\mathbf{G}_{\mathrm{Ctrl}}$ describes 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. Matrix $\mathbf{G}_{\mathrm{Ctrl}}$ 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 gradient $\mathbf{G}_{\mathrm{Sens}}$ describes 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 $\mathbf{x}$, and has 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. $\mathbf{x}$ is given as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \\ \ddot{\mathbf{r}} \end{bmatrix} \tag{8}$$

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 matrix product $\mathbf{G}$ of the gradient matrices $\mathbf{G}_{\mathrm{Act}}$, $\mathbf{G}_{\mathrm{Ctrl}}$ and $\mathbf{G}_{\mathrm{Sens}}$ has dimensions $n_{F_{\mathrm{Ctrl}}} \times 3n_r$. If $\mathbf{G}$ is pre-multiplied with the topology matrix relating each controller force F_{Ctrl_i} with its respective system DOFs, and then split into 3 $n_r \times n_r$ matrices, $\mathbf{G}_{\mathrm{Pos}}$, $\mathbf{G}_{\mathrm{Vel}}$ and $\mathbf{G}_{\mathrm{Acc}}$, one for each state variable $\mathbf{r}$, $\dot{\mathbf{r}}$ and $\ddot{\mathbf{r}}$, the matrices $\mathbf{G}_{\mathrm{Pos}}$, $\mathbf{G}_{\mathrm{Vel}}$ and $\mathbf{G}_{\mathrm{Acc}}$ can be added to their respective system matrix yielding the following equation system for the free vibration of a controlled mechanism [3]:

$$(\mathbf{M} + \mathbf{G}_{\mathrm{Acc}})\ddot{\mathbf{r}}(t) + (\mathbf{C} + \mathbf{G}_{\mathrm{Vel}})\dot{\mathbf{r}}(t) + (\mathbf{K} + \mathbf{G}_{\mathrm{Pos}})\mathbf{r}(t) = \mathbf{0} \tag{9}$$

As shown in Eqs. (7) and (9), there is an uncoupling between the mechanical system and the control elements, hence, the properties of the mechanical system are not of relevance when concerned with identifying the controller parameters.

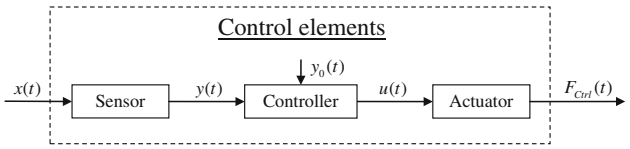


Fig. 2 Control elements

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 to derive the desired system parameters.

In this paper, perturbations will be used on the decoupled controller 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 to not affect any other simulations.

3.1 The perturbation technique

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

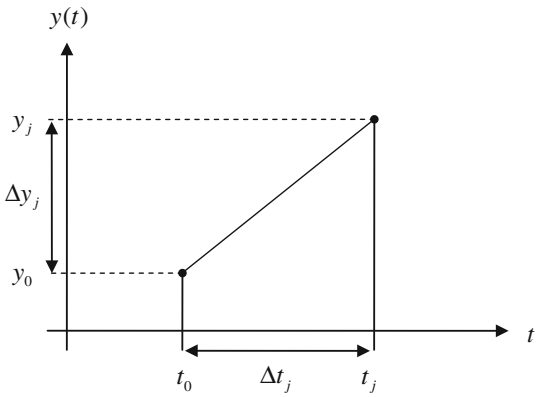


Fig. 3 Perturbation j of t and y

In Fig. 3, the variables time t and controller input y are perturbed by the values $\triangle y_j$ and $\triangle t_j$ during perturbation j. y_0 and t_0 are the initial values for y and t, respectively, at the present time step. From Fig. 3, the following relationships can be derived:

$$\triangle y_j = y_j - y_0 \Rightarrow y_j = y_0 + \triangle y_j \tag{10}$$

$$\triangle t_j = t_j - t_0 \Rightarrow t_j = t_0 + \triangle t_j \tag{11}$$

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

$$\begin{aligned} \triangle u_j &= u_j - u_0 \\ &= u_j(y_j, t_j) - u(y_0, t_0) \\ &= u_j(y_0 + \triangle y_j, t_0 + \triangle t_j) - u(y_0, t_0) \end{aligned} \tag{12}$$

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

$$y_j(t) = y_0 + \frac{\triangle y_j}{\triangle t_j}\, t \tag{13}$$

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:

$$u_{\mathrm{P}}(t) = K_{\mathrm{p}}\, y(t) \tag{14}$$

where K_{p} is the proportional gain. Equation (14) can be written on a general differential form as:

$$\mathrm{d}u = \frac{\partial u}{\partial y}\,\mathrm{d}y \quad \text{or} \quad \mathrm{d}u = K_{\mathrm{p}}\,\mathrm{d}y \tag{15}$$

Since the perturbation technique is meant to be used with computers, it is more suitable to treat Eq. (15) numerically rather than analytically. In discrete differential form, Eq. (15) can be written as:

$$\triangle u = \frac{\partial u}{\partial y}\,\triangle y \quad \text{or} \quad \triangle u = K_{\mathrm{p}} \triangle y \tag{16}$$

K_{p} can thus be calculated by solving the following equation:

$$K_{\mathrm{p}} = (\triangle y)^{-1}\,\triangle u \tag{17}$$

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 Eqs. (10) and (11).
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 Eq. (12).
6. Use Eq. (17) 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:

$$u_I(t) = K_\mathrm{i} \int y(t)\,\mathrm{d}t \tag{18}$$

where K_i is the integral gain. Equation (18) can be written in discrete differential form as:

$$\triangle u = \frac{\partial u}{\partial \int y\,\mathrm{d}t} \triangle \int y\,\mathrm{d}t \quad \text{or} \quad \triangle u = K_\mathrm{i} \triangle \int y\,\mathrm{d}t \tag{19}$$

and K_i can be calculated by solving the equation:

$$K_\mathrm{i} = \left(\triangle \int y_j\,\mathrm{d}t \right)^{-1} \triangle u_j \tag{20}$$

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 Fig. 3, $\triangle \int y_j\,dt$ is the area under the linear curve. If $y_j(t)$ is given as in Eq. (13), $\triangle \int y_j\,dt$ can be made as a function of $\triangle y_j$ and $\triangle t_j$ by:

$$\begin{aligned} \triangle \int y_j\,\mathrm{d}t = \int_0^{\triangle t_j} y_j\,\mathrm{d}t &= \left[y_0 t + \frac{1}{2}\frac{\triangle y_j}{\triangle t_j} t^2 \right]_0^{\triangle t_j} \\ &= \left(y_0 + \frac{1}{2}\triangle y_j \right) \triangle t_j \end{aligned} \tag{21}$$

Inserting Eq. (21) into Eq. (20) yields:

$$K_\mathrm{i} = \left(\left(y_0 + \frac{1}{2}\triangle y_j \right) \triangle t_j \right)^{-1} \triangle u_j \tag{22}$$

3.3.2 Double integration

Like the single integration presented in Sect. 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:

$$u_\mathrm{II}(t) = K_\mathrm{ii} \iint y(t)\,\mathrm{d}t\,\mathrm{d}t \tag{23}$$

where K_ii is the double integral gain. Equation (23) can be written in discrete differential form as:

$$\triangle u = \frac{\partial u}{\partial \iint y\,\mathrm{d}t\,\mathrm{d}t} \triangle \iint y\,\mathrm{d}t\,\mathrm{d}t \text{ or } \triangle u = K_\mathrm{ii} \triangle \iint y\,\mathrm{d}t\,\mathrm{d}t \tag{24}$$

Based on antidifferentiation, the double integral $\triangle \iint y_j\,\mathrm{d}t\,\mathrm{d}t$ can be derived in discrete form as:

$$\begin{aligned} \triangle \iint y_j\,\mathrm{d}t\,\mathrm{d}t = \int_0^{\triangle t_j} \int_0^{t} y_j\,\mathrm{d}t\,\mathrm{d}t &= \left[\frac{1}{2} y_0 t^2 + \frac{1}{6}\frac{\triangle y_j}{\triangle t_j} t^3 \right]_0^{\triangle t_j} \\ &= \left(\frac{1}{2} y_0 + \frac{1}{6}\triangle y_j \right) \triangle t_j^2 \end{aligned} \tag{25}$$

K_ii can thus be estimated by solving:

$$K_\mathrm{ii} = \left(\left(\frac{1}{2} y_0 + \frac{1}{6}\triangle y_j \right) \triangle t_j^2 \right)^{-1} \triangle u_j \tag{26}$$

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:

$$u_\mathrm{III}(t) = K_\mathrm{iii} \iiint y(t)\,\mathrm{d}t\,\mathrm{d}t\,\mathrm{d}t \tag{27}$$

where K_iii is the triple integral gain. Following the same procedure as for the double integral in the previous section, the triple integral $\triangle \iiint y_j\,\mathrm{d}t\,\mathrm{d}t\,\mathrm{d}t$ can be derived in discrete form as:

$$\begin{aligned} \triangle \iiint y_j\,\mathrm{d}t\,\mathrm{d}t\,\mathrm{d}t &= \int_0^{\triangle t_j} \int_0^{t} \int_0^{t} y_j\,\mathrm{d}t\,\mathrm{d}t\,\mathrm{d}t \\ &= \left[\frac{1}{6} y_0 t^3 + \frac{1}{24}\frac{\triangle y_j}{\triangle t_j} t^4 \right]_0^{\triangle t_j} \\ &= \left(\frac{1}{6} y_0 + \frac{1}{24}\triangle y_j \right) \triangle t_j^3 \end{aligned} \tag{28}$$

and K_iii can be estimated by solving:

$$K_\mathrm{iii} = \left(\left(\frac{1}{6} y_0 + \frac{1}{24}\triangle y_j \right) \triangle t_j^3 \right)^{-1} \triangle u_j \tag{29}$$

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:

$$u_{\mathrm{D}}(t) = K_{\mathrm{d}} \frac{\mathrm{d}}{\mathrm{d}t} y(t) = K_{\mathrm{d}}\, \dot{y}(t) \tag{30}$$

where K_{d} is the derivative gain. Equation (30) can be written in discrete differential form as:

$$\triangle u = \frac{\partial u}{\partial \dot{y}}\, \triangle \dot{y} \quad \text{or} \quad \triangle u = K_{\mathrm{d}}\, \triangle \dot{y} \tag{31}$$

K_{d} can be calculated by solving the equation:

$$K_{\mathrm{d}} = \left(\triangle \dot{y}_j\right)^{-1} \triangle u_j \tag{32}$$

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 Fig. 3 can be given as:

$$\dot{y}_j = \frac{\triangle y_j}{\triangle t_j} \tag{33}$$

Using Eq. (10) as a basis, $\triangle \dot{y}_j$ can be given as:

$$\triangle \dot{y}_j = \dot{y}_j - \dot{y}_0 = \frac{\triangle y_j}{\triangle t_j} - \frac{\triangle y_0}{\triangle t_0} \tag{34}$$

However, $\dot{y}_0$ does not exist in Fig. 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 Fig. 4.

As for the one-step perturbation illustrated in Fig. 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 (34) can then be simplified to:

$$\triangle \dot{y}_j = \dot{y}_j - 0 = \frac{\triangle y_j}{\triangle t_j} \tag{35}$$

Inserting Eq. (35) into Eq. (32) yields:

$$K_{\mathrm{d}} = \left(\frac{\triangle y_j}{\triangle t_j}\right)^{-1} \triangle u_j \tag{36}$$

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:

$$u_{\mathrm{DD}}(t) = K_{\mathrm{dd}} \frac{\mathrm{d}^2}{\mathrm{d}t^2} y(t) = K_{\mathrm{dd}}\, \ddot{y}(t) \tag{37}$$

where K_{dd} is the double derivative gain. Equation (37) can be written in discrete differential form as:

$$\triangle u = \frac{\partial u}{\partial \ddot{y}}\, \triangle \ddot{y} \quad \text{or} \quad \triangle u = K_{\mathrm{dd}}\, \triangle \ddot{y} \tag{38}$$

Similarly to the discretization of $\triangle \dot{y}_j$ in Sect. 3.4.1, $\triangle \ddot{y}_j$ can be written as:

$$\triangle \ddot{y}_j = \ddot{y}_j - \ddot{y}_0 = \frac{\triangle \dot{y}_j}{\triangle t_j} - \frac{\triangle \dot{y}_0}{\triangle t_0} \tag{39}$$

In order to derive $\triangle \ddot{y}$, three perturbation steps have to be performed. An example of a three-step perturbation is given in Fig. 5.

In Fig. 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 Fig. 5, the variables $\triangle y_j$, $\triangle \dot{y}_j$ and $\triangle \ddot{y}_j$ can be given as:

$$\triangle y_j = \triangle y_{j2} \tag{40}$$

$$\triangle \dot{y}_j = \triangle \dot{y}_{j2} = \frac{\triangle y_{j2} - \triangle y_{j1}}{\triangle t_j} \tag{41}$$

$$\triangle \ddot{y}_j = \frac{\triangle \dot{y}_{j2} - \triangle \dot{y}_{j1}}{\triangle t_j} = \frac{\frac{\triangle y_{j2} - \triangle y_{j1}}{\triangle t_j} - \frac{\triangle y_{j1} - \triangle y_{j0}}{\triangle t_j}}{\triangle t_j} = \frac{\triangle y_{j2} - 2\,\triangle y_{j1} + \triangle y_{j0}}{\triangle t_j^2} \tag{42}$$

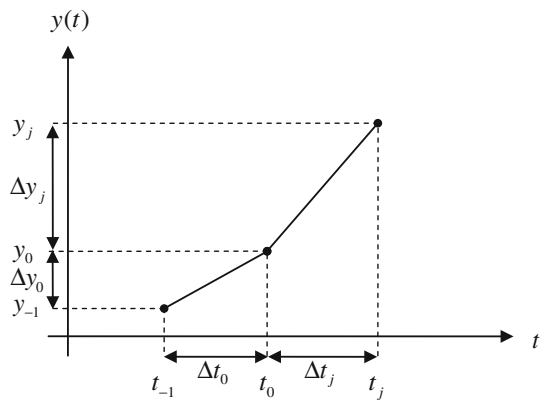


Fig. 4 Two-step perturbation

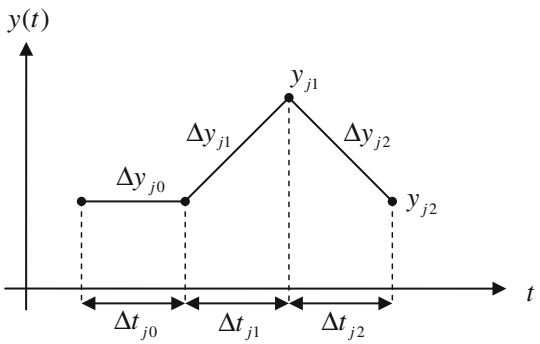


Fig. 5 Three-step perturbation

Since $\triangle y_{j0} = 0$, Eq. (42) can be reduced to:

$$\triangle \ddot{y}_j = \frac{\triangle y_{j2} - 2\,\triangle y_{j1}}{\triangle t_j^2} \tag{43}$$

K_{dd} can thus be calculated by solving:

$$K_{\mathrm{dd}} = \left(\frac{\triangle y_{j2} - 2\,\triangle y_{j1}}{\triangle t_j^2}\right)^{-1} \triangle u_j \tag{44}$$

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 Eq. (3). In a discrete differential form, this can be written as:

$$\begin{aligned}\triangle u &= \frac{\partial u}{\partial y}\,\triangle y + \frac{\partial u}{\partial \int y\,\mathrm{d}t}\,\triangle \int y\,\mathrm{d}t + \frac{\partial u}{\partial \dot{y}}\,\triangle \dot{y} \text{ or } \triangle u \\ &= K_{\mathrm{p}}\,\triangle y + K_{\mathrm{i}}\,\triangle \int y\,\mathrm{d}t + K_{\mathrm{d}}\,\triangle \dot{y}\end{aligned} \tag{45}$$

As demonstrated in Eq. (45), 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 Sect. 3.4.1. This gives the following set of equations:

$$\begin{aligned}\triangle u_1 &= K_{\mathrm{p}}\,\triangle y_1 + K_{\mathrm{i}}\,\triangle \int y_1\,\mathrm{d}t + K_{\mathrm{d}}\,\triangle \dot{y}_1 \\ \triangle u_2 &= K_{\mathrm{p}}\,\triangle y_2 + K_{\mathrm{i}}\,\triangle \int y_2\,\mathrm{d}t + K_{\mathrm{d}}\,\triangle \dot{y}_2 \\ \triangle u_3 &= K_{\mathrm{p}}\,\triangle y_3 + K_{\mathrm{i}}\,\triangle \int y_3\,\mathrm{d}t + K_{\mathrm{d}}\,\triangle \dot{y}_3\end{aligned} \tag{46}$$

which can be written in matrix form as:

$$\begin{bmatrix}\triangle u_1 \\ \triangle u_2 \\ \triangle u_3\end{bmatrix} = \begin{bmatrix}\triangle y_1 & \triangle \int y_1\,\mathrm{d}t & \triangle \dot{y}_1 \\ \triangle y_2 & \triangle \int y_2\,\mathrm{d}t & \triangle \dot{y}_2 \\ \triangle y_3 & \triangle \int y_3\,\mathrm{d}t & \triangle \dot{y}_3\end{bmatrix} \begin{bmatrix}K_{\mathrm{p}} \\ K_{\mathrm{i}} \\ K_{\mathrm{d}}\end{bmatrix} \tag{47}$$

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:

$$\begin{bmatrix}K_{\mathrm{p}} \\ K_{\mathrm{i}} \\ K_{\mathrm{d}}\end{bmatrix} = \begin{bmatrix}\triangle y_1 & \triangle \int y_1\,\mathrm{d}t & \triangle \dot{y}_1 \\ \triangle y_2 & \triangle \int y_2\,\mathrm{d}t & \triangle \dot{y}_2 \\ \triangle y_3 & \triangle \int y_3\,\mathrm{d}t & \triangle \dot{y}_3\end{bmatrix}^{-1} \begin{bmatrix}\triangle u_1 \\ \triangle u_2 \\ \triangle u_3\end{bmatrix} \tag{48}$$

Inserting Eqs. (21) and (35) into Eq. (48) yields:

$$\begin{bmatrix}K_{\mathrm{p}} \\ K_{\mathrm{i}} \\ K_{\mathrm{d}}\end{bmatrix} = \begin{bmatrix}\triangle y_1 & \left(y_0 + \frac{1}{2}\triangle y_1\right)\triangle t_1 & \frac{\triangle y_1}{\triangle t_1} \\ \triangle y_2 & \left(y_0 + \frac{1}{2}\triangle y_2\right)\triangle t_2 & \frac{\triangle y_2}{\triangle t_2} \\ \triangle y_3 & \left(y_0 + \frac{1}{2}\triangle y_3\right)\triangle t_3 & \frac{\triangle y_3}{\triangle t_3}\end{bmatrix}^{-1} \begin{bmatrix}\triangle u_1 \\ \triangle u_2 \\ \triangle u_3\end{bmatrix} \tag{49}$$

To avoid singularities when performing the matrix inversion in Eq. (49), 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:

$$\begin{aligned}&\triangle t_1 = \delta \cdot \triangle t_{\mathrm{sim}}, \quad \triangle y_1 = \triangle t_1, \quad \triangle t_j = j \cdot \triangle t_1, \\ &\triangle y_j = j \cdot \triangle y_1\end{aligned} \tag{50}$$

where $\triangle t_{\mathrm{sim}}$ 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 Sect. 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, $j = 1\ldots3$, but if Eq. (50) is used, it is sufficient to establish $\triangle t_1$.
4. Calculate $\triangle \int y_j\,\mathrm{d}t$ and $\triangle \dot{y}_j$. These are given by Eqs. (21) and (35).
5. Calculate y_j and t_j in accordance with Eqs. (10) and (11).
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 Eq. (12).
8. Use the matrix system in Eq. (49) to estimate K_{p}, K_{i} and K_{d}.

3.6 Partitioning of perturbation steps

When testing the perturbation technique presented in Sects. 3.1–3.5, 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 Fig. 6.

As illustrated in Fig. 6, if the perturbation is divided into n equal sections, each perturbation step $\triangle t_j$ and $\triangle y_j$ would be composed 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

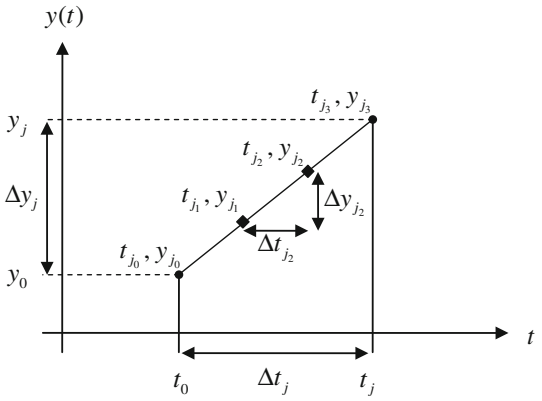


Fig. 6 Partition of perturbation

not to be confused with the subscripting presented in Sect. 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:

$$\triangle t_{j_i} = \frac{1}{n}\ \triangle t_j \tag{51}$$

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, Eqs. (21), (25) and (28) were used to calculate $\triangle \int y\,\mathrm{d}t$, $\triangle \iint y\,\mathrm{d}t\,\mathrm{d}t$ and $\triangle \iiint y\,\mathrm{d}t\,\mathrm{d}t\,\mathrm{d}t$, respectively. In Simulink, this system was created using three integration blocks in series, as shown in Fig. 7.

For the triple integration system, the parameters $\triangle t_j$, $\triangle y_j$ and y_0 were given as $\triangle t_1 = 0.1$ s, $\triangle y_1 = \triangle t_1$ 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.

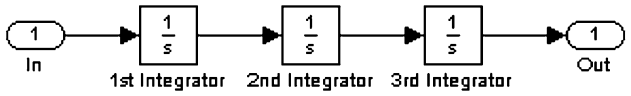


Fig. 7 System of triple integration modeled in Simulink using the Continuous Integrator block

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
$\triangle \int y_1\,\mathrm{d}t$	0.005	0.005
$\triangle \iint y_1\,\mathrm{d}t\,\mathrm{d}t$	1.6667×10^{-4}	1.6667×10^{-4}
$\triangle \iiint y_1\,\mathrm{d}t\,\mathrm{d}t\,\mathrm{d}t$	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 $\triangle \int y_1\,\mathrm{d}t$, $\triangle \iint y_1\,\mathrm{d}t\,\mathrm{d}t$ and $\triangle \iiint y_1\,\mathrm{d}t\,\mathrm{d}t\,\mathrm{d}t$.

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, Eqs. (41) and (42) were used for $\triangle \dot{y}$ and $\triangle \ddot{y}$, respectively. In Simulink, this system was created using two derivation blocks in series, as shown in Fig. 8.

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 $\triangle t_1 = 0.1$ s. 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.

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 Sect. 3.5, some basic tests were performed using the perturbation technique. The objective of the tests was to verify whether the

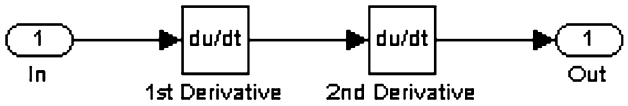


Fig. 8 System of double derivation modeled in Simulink using the Continuous Derivative block

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

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. In addition, 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 Fig. 9.

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 to verify the perturbation technique, the values in this example are known a priori. For the perturbation of the PID controller based on Fig. 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 $\triangle t_1 = 0.1$ s, $\triangle t_j = j \cdot \triangle t_1$, $\triangle y_j = \triangle t_j$, $\triangle t_0 = \triangle t_1$ 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 Eq. (49), the results from the simulation in Simulink are shown in Eq. (52).

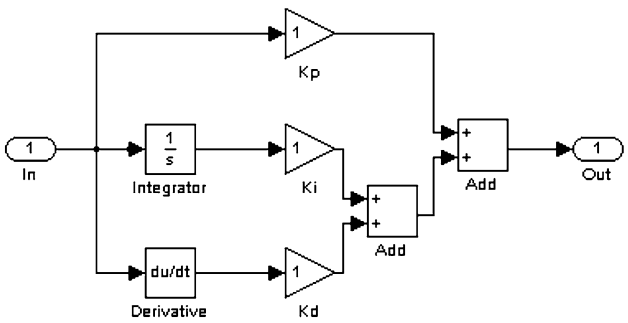


Fig. 9 PID controller modeled in Simulink

$$\begin{bmatrix} K_p \\ K_i \\ K_d \end{bmatrix} = \begin{bmatrix} 0.1000 & 0.0050 & 1.0000 \\ 0.2000 & 0.0200 & 1.0000 \\ 0.3000 & 0.0450 & 1.0000 \end{bmatrix}^{-1} \begin{bmatrix} 1.1050 \\ 1.2200 \\ 1.3450 \end{bmatrix} = \begin{bmatrix} 1.0000 \\ 1.0000 \\ 1.0000 \end{bmatrix} \quad (52)$$

As revealed in Eq. (52), $K_p = 1$, $K_i = 1$ 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 Fig. 10.

The setup in Fig. 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. As stated in Sect. 2, the properties of the mechanical system are not of relevance when concerned with identifying the controller parameters, hence, neither the parameters value nor the number of DOFs of the mass-spring-damper system are of relevance in this context. The numerical values for the controller gains were arbitrarily chosen, but were deliberately given different values to more easily distinguish between them. The values for $\triangle t_j$ and $\triangle y_j$ were given by Eq. (50), with $\delta = 0.1$. The simulation time increment $\triangle t_{sim}$ for all tests in this section was set to 0.01 s, i.e. $\triangle t_1 = 0.001$.

Three different tests were performed on the active mass-spring-damper system shown in Fig. 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 Fig. 10 was set in static

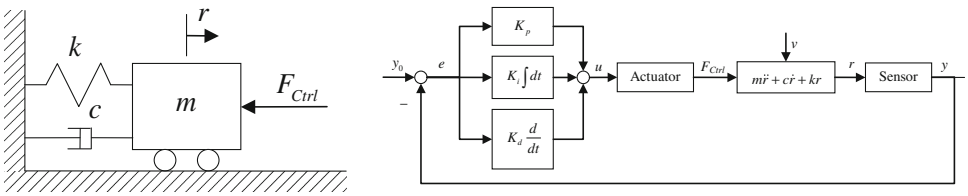


Fig. 10 SDOF Mass-spring-damper system with position feedback PID controller

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

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 to depict the accuracy of the method and show when the estimated values deviate from the correct ones, 12 decimals were used.

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 12th decimal. For the PI and PID controller, the integral gain K_i is only correct up to the 9th decimal, and the proportional gain K_p for the PID controller is incorrect on the 12th 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 Fig. 10 was given a prescribed sinusoidal motion of 1 Hz, and the perturbation technique was performed on a PID controller at

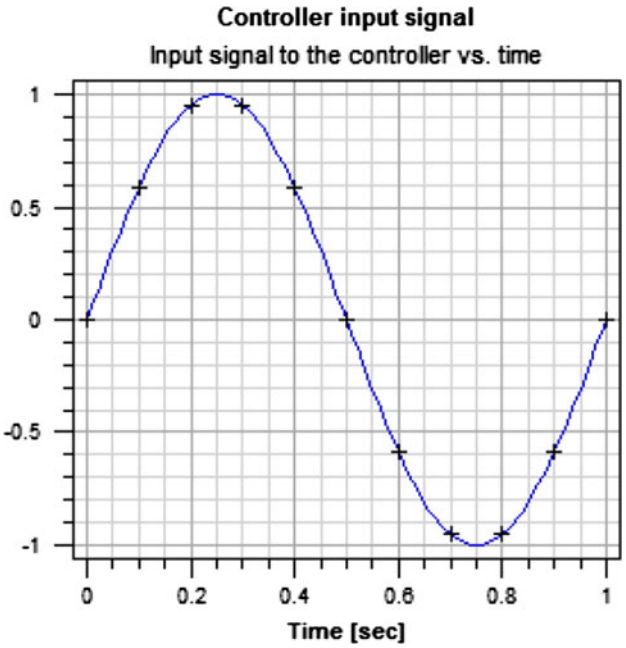


Fig. 11 Input signal to the controller. The crosses mark each time the perturbation sequence is performed

various time steps. The simulation ran for one second with a time increment of 0.01 s. The perturbation technique was performed both at start-up and at time intervals of 0.1 s, 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 Fig. 11. The results from the simulation are shown in Table 4.

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 s, 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 s, the difference being approximately 2.5×10^{-8} for 0.1 s and approximately $\mp 2.0 \times 10^{-8}$ for 0.2 and 0.7 s, respectively. For the derivative gain, K_d, the perturbation technique yields the greatest errors at time 0.1 and 0.2 s, being about 5×10^{-14} and -4×10^{-14}, respectively.

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

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 Sect. 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 Fig. 12, this should occur at time 0.2, 0.3, 0.7 and 0.8 s. All other parameters were the same as they were in Sect. 4.3.2. The results from the simulation are shown in Table 5.

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 s, being approximately -8×10^{-7}. For the K_i estimation, the greatest error is encountered at 0.5 s, being approximately 2×10^{-6}. For the K_d estimation, the greatest errors are encountered at 0.2, 0.4 and 0.5 s, all being approximately 2.5×10^{-12}.

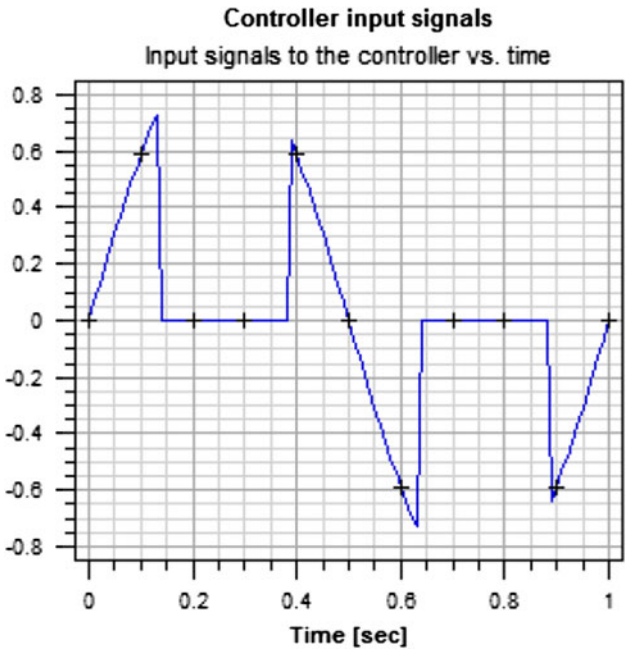


Fig. 12 Input signals to the controller. The crosses mark each time the perturbation sequence is performed. The input signal is a discontinuous sinusoidal signal with switch-to-zero-value at ± 0.7

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

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 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 Sects. 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 Sect. 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 resulting 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.

Acknowledgments 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 (2001) Flexible multibody dynamics: a finite element approach. Wiley, Chichester
2. Sivertsen OI (2001) Virtual testing of mechanical systems—theories and techniques. Advances in Engineering 4. Swets and Zeitlinger B.V., Lisse, The Netherlands
3. Bratland M, Haugen B, Rølvåg T (2011) Modal analysis of active flexible multibody systems. Comput Struct 89(9–10):750–761
4. Preumont A (2002) Vibration control of active structures: an introduction, 2nd edn. Kluwer Academic Publishers, Dordrecht
5. Inman DJ (2006) Vibration with control. Wiley, Chichester
6. Balchen JG, Andresen T, Foss BA (2003) Reguleringsteknikk, 5th ed. Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway (in Norwegian)
7. Thomson WT, Dahleh MD (1998) Theory of Vibration with Applications, 5th edn. Prentice Hall, Inc., Upper Saddle River
8. Palm WJ (2007) Mechanical Vibration. John Wiley & Sons, Inc., Hoboken
9. Cook RD, Malkus DS, Plesha ME, Witt RJ (2002) Concepts and applications of finite element analysis, 4th edn. Wiley, New York
10. Bathe K-J (1996) Finite Element Procedures. Prentice Hall, Englewood Cliffs
11. Alvin KF, Park KC (1994) Second-order structural identification procedure via state-space-based system identification. AIAA J 32:397–406
12. Sivertsen OI, Waloen AO (1982) Non-linear finite element formulations for dynamic analysis of mechanisms with elastic components. In: Washington, DC, USA. American Society of Mechanical Engineers, ASME, New York, NY, USA, p 7
13. Astrom KJ, Hagglund T (2001) The future of PID control. Control Eng Pract 9:1163–1175
14. Astrom KJ, Hagglund T (2004) Revisiting the Ziegler-Nichols step response method for PID control. J Process Control 14:635–650

15. Alkhatib R, Golnaraghi MF (2003) Active structural vibration control: a review. Shock Vib Dig 35:367–383
16. Bratland M, Rølvåg T (2008) Modal analysis of lumped flexible active systems (Part 1). In: Paper presented at the SIMS 2008: the 48th Scandinavian Conference on Simulation and Modeling, Oslo, Norway, 07 October 2008–08 October 2008
17. Park KC, Felippa CA, Ohayon R (2009) The d'Alembert-Lagrange principal equations and applications to floating flexible systems. Int J Numer Meth Eng 77:1072–1099
18. Felippa CA (2001) A historical outline of matrix structural analysis: a play in three acts. Comput Struct 79:1313–1324
19. Astrom KJ, Eykhoff P (1971) System identification-a survey. Automatica 7:123–162
20. Perreault EJ, Kirsch RF, Acosta AM (1999) Multiple-input, multiple-output system identification for characterization of limb stiffness dynamics. Biol Cybern 80:327–337
21. Trier SD (2001) Design optimization of flexible multibody systems. Doctoral Thesis, Norwegian University of Science and Technology