Modeling Flexible Bodies in SimMechanics

Victor Chudnovsky, Arnav Mukherjee,
Jeff Wendlandt, and Dallas Kennedy

May 11, 2006

Abstract

SimMechanics does not offer native support for flexible-body modeling. Nevertheless, the extensibility of the underlying MATLAB® and Simulink® environment allows users to model and encapsulate flexible-body models. This paper discusses two approaches to this problem. The first is the lumped-parameter method, in which a flexible body is discretized into a collection of rigid bodies connected by springs. The second is the finite-element analysis (FEA) method, which incorporates vibration analysis done using third-party FEA applications to model the relative motion of points of interest on a flexible body. We present examples and discuss the merits of each approach. To download these examples, please visit `http://tinyurl.com/onz3m`.

Contents

1 Introduction

Mechanical models often play an important role in control systems simulation. SimMechanics makes it easy to create a representation of a physical mechanism within a Simulink®model, but SimMechanics simulates only rigid-body dynamics: none of the parts represented by Body blocks are assumed to change their shape or mass distribution. In real applications, however, there is often a need to model flexible dynamics. The extensibility of MATLAB®, Simulink, and SimMechanics makes it possible for users to create their own flexible-body models and libraries.

In this paper, we explore two distinct approaches to modeling flexible bodies in SimMechanics. The lumped-parameter method, best suited for modeling beam-like geometries, discretizes the flexible body into a series of constituent elements, each of which contains a spring-damper that models the flexibility. The finite-element analysis (FEA) approach, in contrast, incorporates the output of third-party FEA programs to actuate the relative motion of the bodies that are connected to the flexible part under consideration.

2 The Lumped-Parameter Method

For most engineering purposes, a real flexible body is a continuous medium. The lumped-parameter method approximates a flexible body as a set of rigid bodies coupled with springs and dampers and, in SimMechanics, can be implemented by a chain of alternating bodies and joints. The springs and dampers act either on the bodies or the joints. The spring stiffness coefficients and damping coefficients are functions of the material properties and the geometry of the flexible member under consideration. Lumped-parameter methods are best suited to models with linear geometries, such as beams, in which each fundamental flexible element is coupled to two others in a simple chain.[1] Although this method can be extended, bodies with more complicated geometry are easier to model with other approaches.

2.1 Theory

The lumped-parameter method discretizes the beam of length L into n identical generalized beam elements (GBEs), each[2] of length $l = L/n$ and mass $m = M/n$. Each GBE is a body–joint–body combination, with the joint primitives chosen to reflect the flexible degrees of freedom being modeled. The material properties determine the spring stiffness and damping that are applied to the joint. The

[1]Note that the flexibility being modeled need not be one-dimensional; for example, the model could include both bending and torsion.

[2]By making the GBEs identical, we are assuming that the flexibility properties of the beam are uniform along its length. This assumption can be relaxed by letting the material properties vary from one beam element to the next. Moreover, assigning the same length to all the GBEs is a modeling decision. We can vary GBE length if we need, for example, a more detailed approximation to the shape of a beam portion.

beam, shown in Figure 1, consists of adjacent GBEs welded together. We now develop a simple theory for modeling the general flexibilty of this beam by assuming that the stiffness of each GBE is *local*, or caused by its own deflection.[3] We then specialize the theory to the case of pure bending.

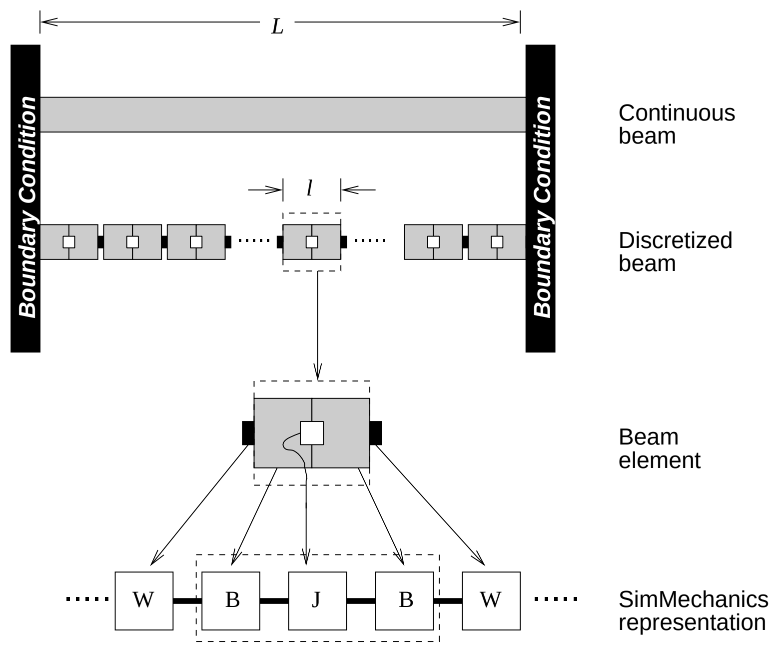

Figure 1. Welds (W), bodies (B), and joints (J) comprising the lumped-parameter discretization of a beam.

2.1.1 General Flexible Beam

Figure 2 shows an individual GBE together with the forces and moments acting on it. The GBE consists of two bodies, each of length $l/2$ and mass $m/2$. The joint acts along the neutral axis[4] of the GBE. In the following analysis, one end of the GBE is fixed, and X and F are generalized coordinates and forces, respectively, at the other end. Let x be the parameterization of the degrees of freedom of the joint. Then

$$X = g(x)$$

[3] A more detailed approach would derive the discretization from the fourth-order differential equation describing the beam deflection.

[4] The neutral axis is defined as the axis that does not undergo compression or elongation.

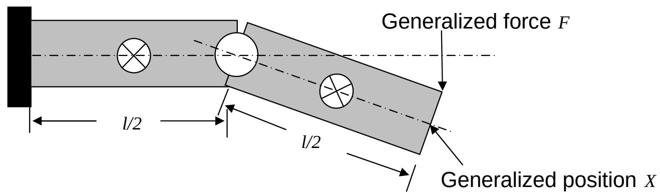

Figure 2. Individual generalized beam element (GBE).

$$dX = J(x)dx, \tag{1}$$

where $J(x)$ is the Jacobian

$$J = \frac{\partial g}{\partial x}.$$

The generalized force F at the tip can be expressed with a generalized *stiffness matrix* K:

$$F = KdX. \tag{2}$$

The generalized force f at the joint can be expressed as

$$f = kdx, \tag{3}$$

where k is the equivalent spring constant at the joint of a GBE and dx is the infinitesimal generalized relative displacement between the two bodies across the joint.

The infinitesimal work $F^T dX$ done at the tip of the GBE by the force F is equivalent to the work $f^T dx$ done at the GBE joint by the equivalent force f instead. Thus,

$$\begin{aligned} f^T dx &= F^T dX \\ &= F^T J(x) dx \end{aligned}$$

for any dx, from which we obtain the generalized force on the joint

$$f = J^T F. \tag{4}$$

Substituting equations (2) and (1) yields

$$f = J^T KJdx, \tag{5}$$

so that

$$k = J^T KJ. \tag{6}$$

Equation (5) is the general stiffness expression of a GBE with multiple degrees of freedom (DOFs).

2.1.2 Beam Undergoing Pure Bending

Consider now the case in which the beam undergoes pure bending only (no shear). We expect this case to be best modeled by having just a revolute primitive in each GBE, so that $F = [P, T]$ and $X = [S, \Theta]$, where P, T, S, and Θ are the force, moment, vertical deflection, and slope associated at the free end of the GBE, respectively. Because we are only using a single revolute joint in our model, the generalized force on the joint is the torque $f = \tau$, and the generalized displacement of the joint is the revolute angle $x = \theta$.

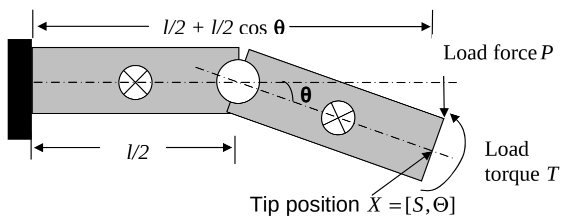

Figure 3. Sample beam element.

The end displacement X is a combination of a linear displacement and a rotational displacement,

$$X = g(x) = \begin{bmatrix} (l/2)\sin\theta \\ \theta \end{bmatrix},$$

so that, for small values of θ,

$$J \approx \begin{bmatrix} l/2 \\ 1 \end{bmatrix}.$$

Equation (6) then becomes

$$k = \begin{bmatrix} l/2 & 1 \end{bmatrix} K \begin{bmatrix} l/2 \\ 1 \end{bmatrix}. \tag{7}$$

A well-known result from beam theory yields the following relationship between the deflection, slope, force, and moment:

$$\begin{bmatrix} dy \\ \theta_e \end{bmatrix} = \left(\frac{l}{EI_{zz}}\right) \begin{bmatrix} l^2/3 & l/2 \\ l/2 & 1 \end{bmatrix} \begin{bmatrix} P \\ T \end{bmatrix}$$

or

$$K = \frac{12EI_{zz}}{l^3} \begin{bmatrix} 1 & -l/2 \\ -l/2 & l^2/3 \end{bmatrix}, \tag{8}$$

where E is the Young's modulus of the material and $I_{zz} = \int_A y^2 dA$ is the *area moment of inertia*. This yields the rotational stiffness or effective torsional spring constant at the joint:

$$k = \frac{EI_{zz}}{l}. \tag{9}$$

The rotational joint in the nth GBE executes damped oscillations according to the normalized moment equation

$$\ddot{\theta}_n + 2\xi\omega_0\dot{\theta}_n + \omega_0^2\theta_n \quad = \quad \text{external moments}$$

where $\omega_0^2 = k/I$ and I is the moment of inertia. The damping coefficient $2\xi\omega_0$ is a quasi-empirical value that accounts for energy lost to visco-elastic effects.

In SimMechanics, the relative angle θ_n and joint velocity $\dot{\theta}_n$ are measured at the joint using a Joint Sensor block. Then k is multiplied by the angle, and a material damping coefficient $2\xi\omega_0$ is multiplied by the angular velocity. The two resulting moments are added and applied back to the joint using a joint actuator.

2.2 Implementing the Lumped-Parameter Method

The lumped-parameter method follows these steps:

1. Divide the beam into discrete elements.
2. Determine the DOFs of the elements and model them with the appropriate joint primitives.
3. Apply a spring-damper to each joint to encapsulate the flexibility parameters.
4. Use flexible-body theory to determine the effective spring constants from the geometry, material properties, and boundary conditions.
5. Apply damping as necessary to each DOF.
6. Construct the beam by welding together a chain of GBEs so configured.

3 The FEA Method

A different approach to modeling flexible bodies allows finite-element analysis (FEA) software applications to discretize the bodies and obtain their *frequency* (or *modal*) *response.* The FEA results can then be incorporated into a SimMechanics model by superimposing the flexible-body deflection on the rigid-body motion.

This approach consists of modeling the part as a rigid body in SimMechanics by using the Body block, making sure to insert a coordinate system at each point where we want to measure or actuate a flexible-body deflection. At each of these coordinate systems, we connect another Body though a Joint that has primitives corresponding to the deflection studied in the FEA model. These primitives are motion-actuated by a "black box" subsystem whose output is the deflection calculated using the FEA data. The input to this black box, in turn, is the load on the body, either the load force from a connected body or an external condition. As explained below, the black box turns out to be a simple state-space model. Figure 4 shows the implementation of the FEA method in SimMechanics.

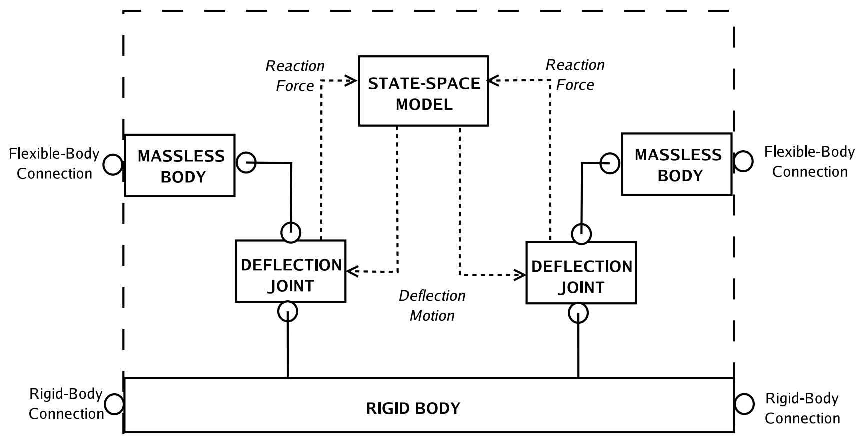

Figure 4. Representation of the FEA method.

3.1 Theory

FEA programs mesh a flexible body into a set of nodes that carry one or more degrees of freedom. For linear FEA, the n_s-component column vector s of these degrees of freedom has the dynamics of a series of forced, damped, coupled harmonic oscillators:

$$M\ddot{s} + C\dot{s} + Ks = Fu. \tag{10}$$

In this equation, M is the mass matrix, C is the damping matrix, and K is the stiffness matrix. The n_u-component vector u represents the nontrivial force inputs to the system, while the $n_s \times n_u$ matrix F maps these nontrivial inputs to the corresponding degrees of freedom. Typically, each entry of F is either zero or one, though scale factors may be introduced (for example, to convert entries of u provided in a different system of units). In a dynamic simulation, the output of interest is often just the position response $y = Ts$ of some subset of the nodes, where T is an $n_y \times n_s$ matrix that, just like F, typically has entries that are ones or zeros. Since the Joint Actuator used in our SimMechanics flexible body model requires consistent values for the position, velocity, and acceleration signals, we are instead interested in the output vector

$$y = \begin{bmatrix} Ts \\ T\dot{s} \\ T\ddot{s} \end{bmatrix}. \tag{11}$$

When performing a frequency analysis on a flexible part, FEA programs typically provide the mode shapes and frequencies in terms of the solutions to the free ($F = 0$), undamped ($C = 0$) coupled oscillators

$$M\ddot{s} + Ks = 0. \tag{12}$$

This differential equation can be solved by postulating the ansatz $s(t) = \phi e^{i\omega t}$, which, upon substitution into (12), leads to the generalized eigenvalue problem

$$K\phi = \omega^2 M\phi. \tag{13}$$

There are n_s eigenvectors ϕ_i that solve this equation with corresponding eigenvalues ω_i. These eigenvectors can be chosen to be M-orthonormal:[5]

$$\phi_i M \phi_j = \delta_{ij}. \tag{14}$$

Typical modeling applications focus on only a few modes, usually those with low frequencies.[6] We define an $n_s \times n_m$ matrix $\Phi = [\phi_1 \ldots \phi_{n_m}]$ whose columns are the n_m eigenvectors corresponding to the modes we wish to incorporate into our SimMechanics model, and we define an $n_s \times (n_s - n_m)$ matrix $\bar{\Phi}$ whose columns are all the other modes. Since the $\{\phi_i\}$ span the n_s-dimensional space $\mathbb{R}^{n_s}$, we can expand

$$s(t) = \Phi\eta(t) + \bar{\Phi}\bar{\eta}(t). \tag{15}$$

This is simply a coordinate transformation between the $(n_m, n_s - n_m)$-tuple of *modal coordinates* $(\eta, \bar{\eta})$ and the n_s-tuple s. Since the modes remain uncoupled in what follows, and we are interested only in the contributions from the n_m modes corresponding to the vector η, we omit the second term in the above equation without any loss of generality:

$$s(t) = \Phi\eta(t). \tag{16}$$

Note, too, that the M-orthogonality relation (14) can be rewritten as

$$\Phi^T M \Phi = I_{n_m \times n_m}. \tag{17}$$

To simulate the behavior of a flexible body when subjected to external forces, we rewrite (10) in terms of the modal coordinates η by substituting (16), multiplying by Φ^T on the left, and applying (17) to obtain

$$\ddot{\eta} + 2\Gamma\dot{\eta} + \Omega\eta = \Sigma u, \tag{18}$$

where the *stiffness matrix* Ω is diagonal by construction,

$$\Omega = \Phi^T K \Phi = \begin{bmatrix} \omega_1 & & \\ & \ddots & \\ & & \omega_{n_m} \end{bmatrix},$$

[5]Eigenvectors with distinct eigenvalues are orthogonal; degenerate eigenvectors with the same eigenvalue can be chosen orthogonal; the M-norm of all the eigenvectors is arbitrary and can be chosen to be unity.

[6]High-frequency modes are less accurate because they are tainted by discretization artifacts. The wavelengths of those modes are comparable to the mesh spacing.

the $n_m \times n_m$ *damping matrix* Γ is given by

$$2\Gamma = \Phi^T C \Phi,$$

and the $n_m \times n_u$ *forcing matrix* is given by

$$\Sigma = \Phi^T F.$$

Similarly, (11) becomes

$$y = \begin{bmatrix} \Theta\eta \\ \Theta\dot{\eta} \\ \Theta\ddot{\eta} \end{bmatrix}, \tag{19}$$

where Θ is an $n_y \times n_m$ matrix,

$$\Theta = T\Phi.$$

The entries of Ω are the frequencies calculated by the FEA program and the columns of Φ are the corresponding mode shapes, while the matrices F and T are specified by the user to indicate which degrees of freedom are to be actuated and sensed, respectively. To calculate the damping matrix Γ, however, we can use any one of several approximation schemes. A simple choice is *proportional damping*, in which each normal mode η_i has its own damping constant $\gamma_i = \xi_i \omega_i$, and the damping does not introduce coupling among the modes:

$$\Gamma = \begin{bmatrix} \xi_1\omega_1 & & \\ & \ddots & \\ & & \xi_{n_m}\omega_{n_m} \end{bmatrix}.$$

The special case $\xi_1 = \ldots = \xi_{n_m}$ is called *uniform damping*.

The harmonic oscillators can now be cast in the standard state-space form

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

by identifying the state vector x with the normal nodes

$$x = \begin{bmatrix} \eta \\ \dot{\eta} \end{bmatrix}. \tag{20}$$

From (18) we read off the entries of A and B,

$$A = \begin{bmatrix} 0 & I \\ -\Omega & -2\Gamma \end{bmatrix} \tag{21}$$

$$B = \begin{bmatrix} 0 \\ \Sigma \end{bmatrix}, \tag{22}$$

while from (19), we can read off the entries of C and D:

$$C = \begin{bmatrix} \Theta & 0 \\ 0 & \Theta \\ -\Theta\Omega & -2\Theta\Gamma \end{bmatrix} \tag{23}$$

$$D = \begin{bmatrix} 0 \\ 0 \\ \Theta\Sigma \end{bmatrix} \tag{24}$$

The linear time invariant (LTI) system representing this state-space model can be created using the `ss` command in the Control System Toolbox to create a continuous-time state-space object that can be used in MATLAB® and inserted into a Simulink®model using the LTI System block. Alternatively, this state-space model can be used directly in Simulink®with the State-Space block from the main Simulink®library.

The output of the state-space system drives the motion of the adjacent bodies relative to the unflexed (rigid-body base) position of the flexible body. A convenient modeling technique is to place joints with the appropriate degrees of freedom (reflecting the flexing being modeled) at the corresponding coordinate systems of the rigid body base. These joints are driven by the output of the state-space model, and are connected on the other side to massless Body blocks. This entire construct can then be encapsulated into a Simulink subsytem that represents the complete flexible-body model and has coordinate system ports to which additional SimMechanics joints can be connected. This flexible-body subsystem is represented by the dashed line in Figure 4 and becomes a modular component of a larger mechanical simulation.

3.2 Algebraic Loops

The state-space formulation introduces algebraic loops as the solver attempts to find a consistent solution for the acceleration of the adjoining bodies whose motion is actuated by the flexible-body dynamics. This actuation depends on the measured reaction force, which is itself a function of the acceleration. There are two ways to break the algebraic loop by using transfer functions:

1. Filter each component of the output y as we have formulated it above through a transfer function of the form

$$H(s) = \frac{K}{s+K}.$$

2. Filter only the position components of y through three separate transfer functions to obtain consistent positions, velocities, and accelerations that can be used in the Joint Actuator:

$$p = \frac{K^3}{(s+K)^3} y$$

$$
\begin{aligned}
v &= \frac{K^3 s}{(s+K)^3} y \\
a &= \frac{K^3 s^2}{(s+K)^3} y.
\end{aligned}
$$

With this choice, we can simplify the state-space system by not outputting velocities or accelerations.

Both formulations are low-pass filters with poles at $s = -K$ and require the somewhat arbitrary selection of a cutoff constant K. The second formulation illustrates a general approach for getting consistent position, velocity, and acceleration signals given only a position input.

3.3 Implementing the FEA Method

The FEA method follows these steps:

1. Identify the FEA degrees of freedom and corresponding SimMechanics Coordinate Systems for the load forces.

2. Identify the FEA degrees of freedom and corresponding SimMechanics Coordinate Systems for the deflections.

3. Perform the FEA analysis and extract the modes.

4. Construct the state-space object.

5. Feed the correct forces into the state-space model. These are either externally-imposed forces or reaction forces from neighboring bodies.

6. Use the output of the state-space object to motion-actuate the appropriate joint primitives connected to the deflection coordinate systems. The state-space output can be filtered through a transfer function to break algebraic loops.

Note that when modeling more than one load force on a single flexible body, all the forces must be inputs to the *same* state-space model. Likewise, when the deflection of more than one coordinate system on a single flexible body is significant for the SimMechanics model, the motion of each of those coordinate systems must be obtained from the *same* state-space model. In these situations, Goto/From block pairs can prevent connection lines from cluttering the diagram. Figure 4 illustrates the SimMechanics connections required.

This approach captures the flexible-body dynamics in the state-space model and uses the results to actuate the motion of one or more massless blocks, to which other SimMechanics blocks can be connected as usual. This method is quite general, as long as the deformations are sufficiently small to remain in the linear regime, and can be used to easily model a complex system with one or more flexible parts.

Unlike the lumped-parameter case, in which the mass of the overall part can be distributed among each of the GBEs, the FEA approach does not automatically take into account the effect of the flexible body's weight in causing its deformation. To include this effect, we must model the gravitational force of the distributed mass as further inputs to the state-space system. If the flexible body is light enough for its self-deflection to be acceptably small, this effect can be neglected.

4 Example: Aluminum Cantilever

To illustrate both of these implementations, consider the aluminum cantilever shown in Figure 5, which has length $L = 50\,\mathrm{cm}$ in the x-direction, height $2a = 20\,\mathrm{mm}$ in the y-direction, and depth $2b = 50\,\mathrm{mm}$ in the z-direction, welded to the wall at the origin and free to bend only in the xy plane. We take the beam to be an aluminum alloy with the following material properties:

$$\begin{aligned} E &= 6.9 \times 10^{10}\,\mathrm{N/m^2} \quad \text{(Young's modulus)} \\ \rho &= 2700\,\mathrm{kg/m^3} \quad \text{(density)}. \end{aligned}$$

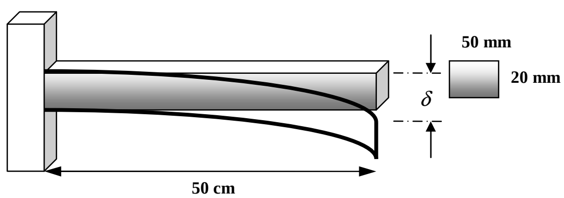

Figure 5. Cantilever example.

We capture the beam specifications in the file `cantilever_aluminum.m`, and the material properties in the file `material_aluminum1060.m`. From these fundamental parameters, we use the file `derive_cantilever_properties.m` to compute the derived parameters, such as the total mass of the beam, $M = 1.35\,\mathrm{kg}$.

4.1 Lumped-Parameter Model

The library file `flex_element_lib.mdl` contains an implementation of the pure-bending generalized beam element discussed earlier. We use $n = 10$ of these GBEs to create the beam in our example. GBEs. The parameters of each GBE appear below:

$$
\begin{aligned}
l &= 5\,\mathrm{cm} \\
m &= 135\,\mathrm{g} \\
k &= 4.6 \times 10^4\,\mathrm{N \cdot m/rad} \\
\xi &= 0.075.
\end{aligned}
$$

Note that the value of the damping coefficient ξ was specified arbitrarily and does not follow from the specified material properties.

4.2 FEA Model

We performed the FEA analysis in COSMOSWorks 2006 SP2.0 using the default mesh parameters (global size 7.94 mm, tolerance 0.40 mm). We modeled the material as a 1060 Aluminum Alloy, which corresponds to the material properties cited earlier. COSMOSWorks reports the five lowest vibration frequencies shown in the table below:

Mode	Frequency(Hz)
1	69.8
2	433.6
3	1198.7
4	2305.8
5	2698.8

For the purposes of this example, we are interested only in the lowest-frequency mode and we choose the same damping coefficient $\xi_1 = 0.075$ as in the lumped-parameter case. To process the FEA data, we use a script `cosmos_cantilever.m` that:

1. Calls an M-function `cosmos2m.m` to read the CosmosWorks data
2. Chooses the appropriate modes to be simulated and degrees of freedom to be actuated and sensed
3. Calls another M-function `fea_to_statespace.m` to construct the state-space matrices in (21)-(24)
4. Creates a MATLAB® continuous-time state-space object `ss_cantilever_pva` that we can include in the SimMechanics model.

We can check our expected output by running

```
step( ss_cantilever_pva )
```

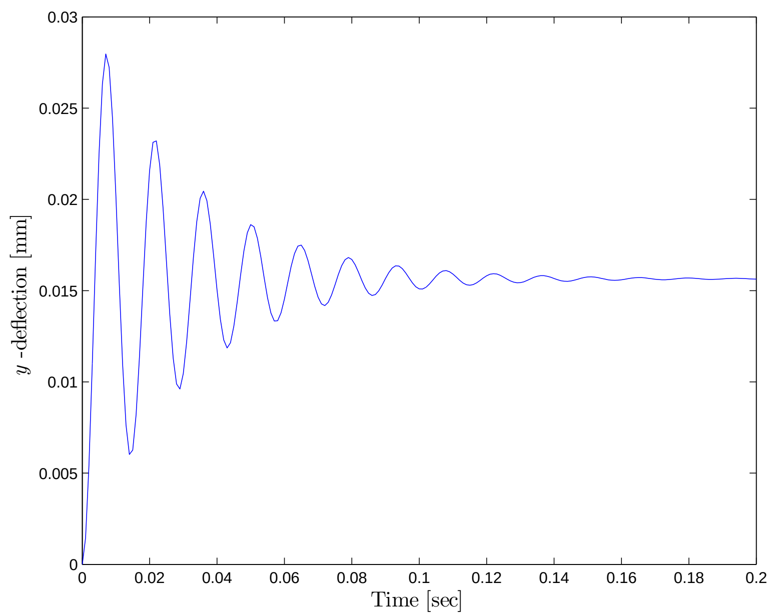

Figure 6. Step response in the state-space model.

The output shows the expected oscillation frequency, while the steady-state response to the unit step is a deflection of $\delta^{(\mathrm{FEA})} = -1.57 \times 10^{-2}\,\mathrm{mm}$.

4.3 Constant Load

We wish to understand how the rod reacts to a step load $P = -196.2\hat{y}\,\mathrm{N}$ at the tip (which corresponds to the weight of a 20 kg mass) between times $t_1 = 0.05$sec and $t_2 = 0.3$ sec.

Figure 7. Step load at the tip.

We note first that standard beam theory predicts a static deflection (neglecting the weight of the beam itself)

$$\delta^{(\mathrm{th})} = \frac{(Fu)L^3}{4Eba^3} = -3.56\,\mathrm{mm},$$

while a COSMOSWorks static analysis yields

$$\delta^{(\mathrm{FEA})} = -3.17\,\mathrm{mm}$$

at this mesh size. The step response at which we looked earlier (which is linear in Fu) predicts a deflection of $\delta^{(\mathrm{step})} = -3.07\,\mathrm{mm}$

4.3.1 Lumped-Parameter Model

The SimMechanics model `lump_cantilever_load.mdl` uses the lumped-parameter approach for the constant-load scenario. The flexible beam is welded to Ground on one side; the other end is actuated with our step force and measured with motion sensors. Since we neglect the weight of the beam in this example, the GBE mass is set to zero. The output is show in Figure 8.

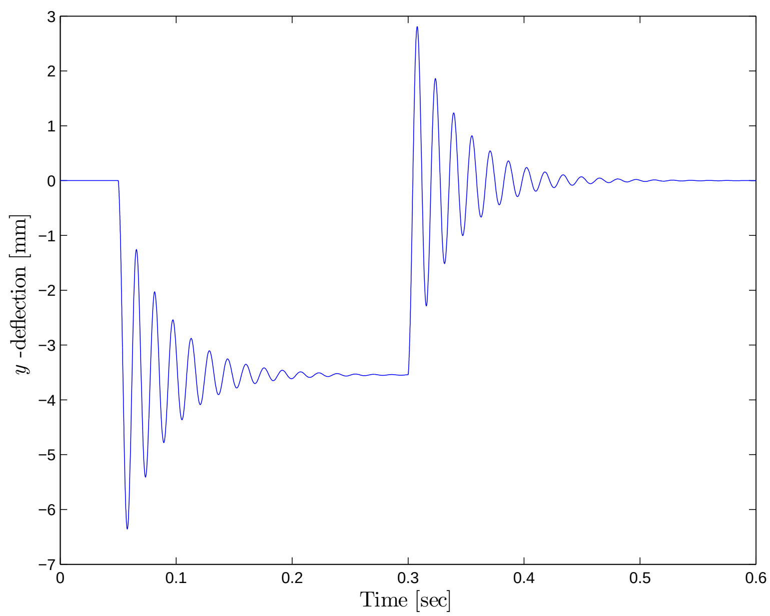

Figure 8. Lumped-parameter step response.

When the nonzero load is applied, the steady-state deflection and vibration frequency are, respectively,

$$\begin{aligned} \delta^{(\mathrm{GBE})} &= -3.54\,\mathrm{mm} \\ f^{(\mathrm{GBE})} &= 63.75\,\mathrm{Hz}. \end{aligned}$$

4.3.2 FEA Model

The SimMechanics model `fea_cantilever_load.mdl` represents the FEA approach. The Cantilever block is just a (rigid) body welded to the wall, but the deflection of the tip is implemented by means of a prismatic joint that is motion-actuated by the output of the state-space system. Note that the massless body is needed because SimMechanics does not allow dangling joints; the fact that it is massless does not cause a simulation singularity because we are completely specifying its motion rather than allowing it to have dynamics. Note, too, that we have placed the state-space model and the massless body inside the Cantilever Deflection subsystem, which encapsulates the flexible-body dynamics within a single block that can be used modularly at the top level of the SimMechanics model.

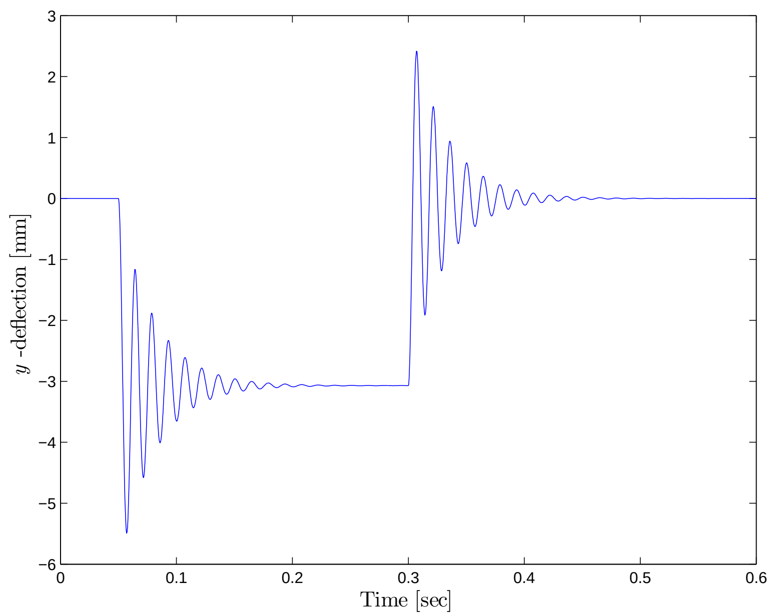

Figure 9. FEA step response.

As expected, the cantilever exhibits oscillations of frequency $f^{(\mathrm{FEA})} = 69.65\,\mathrm{Hz}$ as it damps down to the steady-state deflection corresponding to first the 196.2 N and then the zero loads. The steady-state deflection under the nonzero load is $\delta^{(\mathrm{FEA})} = -3.07\,\mathrm{mm}$.

4.4 Body Load

When simulating the behavior of mechanical systems, one is usually interested in situations where the load on the flexible part is caused by other bodies rather than by an idealized signal. Let us then remove the load signal and weld a load of mass $m_l = 20\,\mathrm{kg}$ at the free tip of the cantilever.

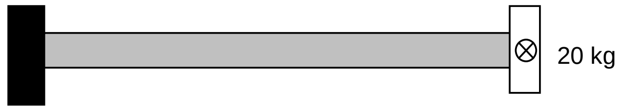

Figure 10. Mass load at the tip.

4.4.1 Lumped-Parameter Model

In the lumped-parameter implementation `lump_cantilever_body.mdl`, we weld the mass load to the last GBE at the free tip of the beam. Because we again neglect the weight of the beam in this example, the GBE mass is set to zero. Figure 11 illustrates the motion of this load body.

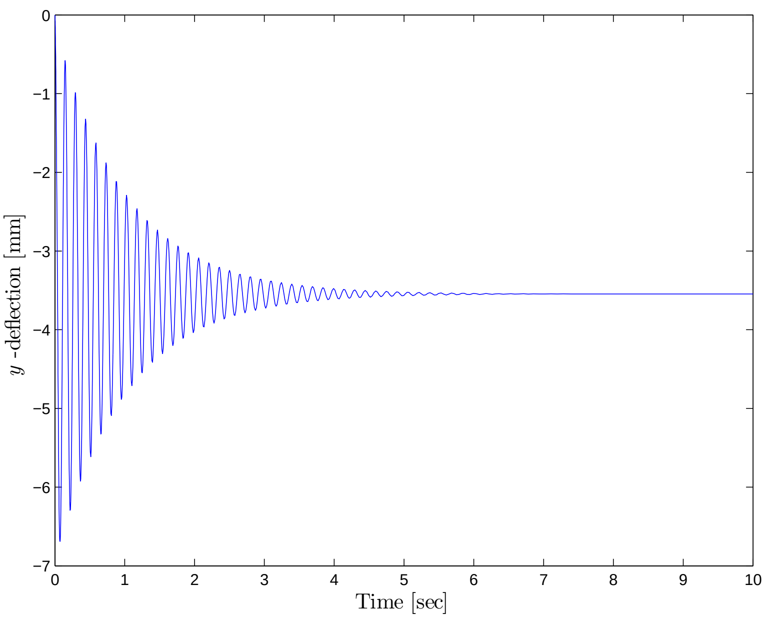

Figure 11. Lumped-parameter mass load response.

The steady-state deflection is $\delta^{(\mathrm{GBE})} = -3.55\,\mathrm{mm}$ and the oscillation frequency is $f^{(\mathrm{GBE})} = 6.82\,\mathrm{Hz}$. The frequency shift is expected, as we explain in section 4.4.3.

4.4.2 FEA Model

In the FEA implementation, we weld the new massive load to the massless body representing the real motion of the attachment coordinate frame. The massless body now exerts a reaction force on the cantilever, which causes it to deflect. The model `fea_cantilever_body.mdl` illustrates this construction. We modified the Cantilever Deflection subsystem to feed the reaction force on the cantilever as the input to the state-space calculation. This model corresponds to placing a load on the undeflected cantilever at $t = 0$; the output shows the cantilever oscillating with a frequency of about $f^{(\mathrm{FEA})} = 8.91\,\mathrm{Hz}$. Running the model for a sufficiently long time achieves a steady-state deflection $\delta^{(\mathrm{FEA})} = -3.07\,\mathrm{mm}$.

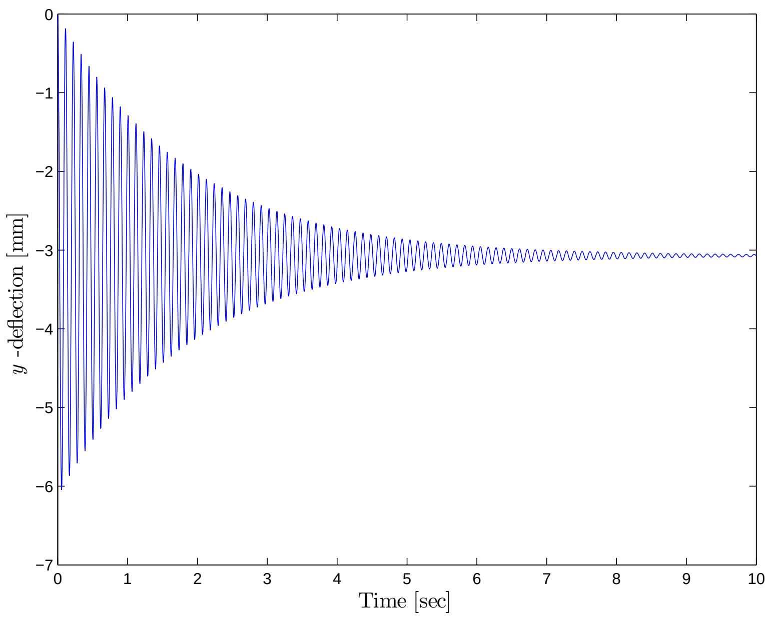

Figure 12. FEA mass load response.

4.4.3 Shift in the Oscillation Frequency

Why does the frequency of oscillation using either of these modeling approaches differ from the fundamental 69.8 Hz observed in the previous example with the constant 196.2 N load? The difference arises from the fact that the load now has dynamics of its own to which the cantilever couples. Consider the free body diagram of the load body: according to Newton's law,

$$m\ddot{x}_l = \Sigma F = F_{0l} + F_{lRc},$$

where, in this example, F_{0l} is the weight of the load body and F_{lRc} is the reaction force from the cantilever on the load. The applied force on the cantilever is the cantilever's action-reaction partner,

$$F_{cRl} = -F_{lRc} = F_{0l} - m\ddot{x}_l,$$

which highlights that the dynamics $m_l \ddot{x}_l$ of the load enter into the right-hand side of the state-space equations (22) and (24).

Indeed, because the cantilever acts as a spring with some intrinsic equivalent mass m_s and some spring constant k_s on the massive load, the equations of motion for the whole system can be written (neglecting damping for simplicity) as

$$(m_s + m_l)\,\ddot{x} = -k_s x + F_{0l},$$

or, equivalently

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m_s & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m_s \end{bmatrix} (F_{ol}/m_s + m_l \ddot{x}/m_s)\,. \qquad (25)$$

The second equation is just the state-space equation we derived earlier. The former makes it clear that the frequency of oscillation with the additional load is given by

$$f_m^2 = \left(\frac{1}{2\pi}\right)^2 \frac{k}{m_s + m_l} = f_0^2 \left(\frac{1}{1 + m_l/m_s}\right).$$

To check that our formulation is consistent, we can see how the frequency varies depending with the applied load m_l in Figure 13. On a sample cantilever with no damping ($\xi = 0$), the squared periods measured on the FEA model lie on the line

$$T_m^2 = f_m^{-2} = f_0^{-2} + m\Delta.$$

A fit yields $f_0 = 69.8\,\mathrm{Hz}$, the no-load oscillation frequency derived by the FEA program.

Finally, recall that nonzero damping will further shift the resonant frequency according to $\omega = \omega_0\sqrt{1 - \xi^2}$.

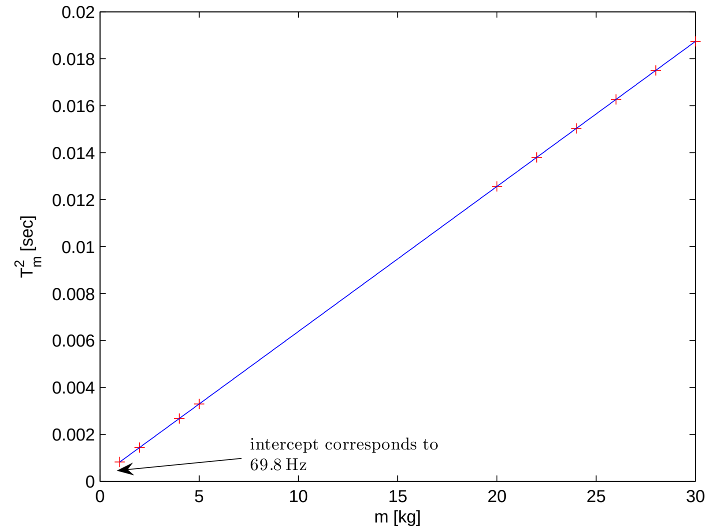

Figure 13. Load mass dependence of (undamped) oscillation frequency.

4.5 Body Load Coupled to Rigid-Body Modes

Finally, consider the situation shown in Figure 14, in which the cantilever not only deflects due to the load, but also moves on its own; in other words, it has a "rigid-body mode." We retain the load at the tip, but replace the weld at the wall with a revolute joint and a Joint Spring Damper with spring constant $k = 1 \times 10^4\,\mathrm{N \cdot m/rad}$ and damping constant $\beta = 0$, which yield a rigid-body oscillation frequency for the (unloaded) cantilever of approximately 8.6 Hz.

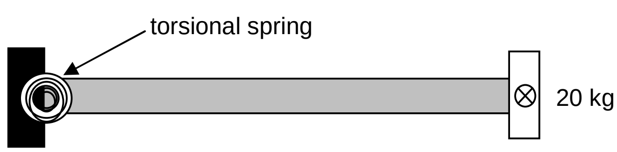

Figure 14. Spring mounting plus tip load.

4.5.1 Lumped-Parameter Model

The model `lump_cantilever_body_spring.mdl` uses the lumped-parameter approach in this situation. This model is just like the one in the body-load scenario, except for the replacement of the wall Weld with a revolute joint actuated with a Joint Spring Damper. Note that we no longer ignore the mass of the beam in this example. Figure 15 shows the displacement of the load body. The steady-state deflection of the system is $\delta^{(\mathrm{GBE})} = -3.63\,\mathrm{mm}$, while the oscillation frequency is $f^{(\mathrm{GBE})} = 6.79\,\mathrm{Hz}$.

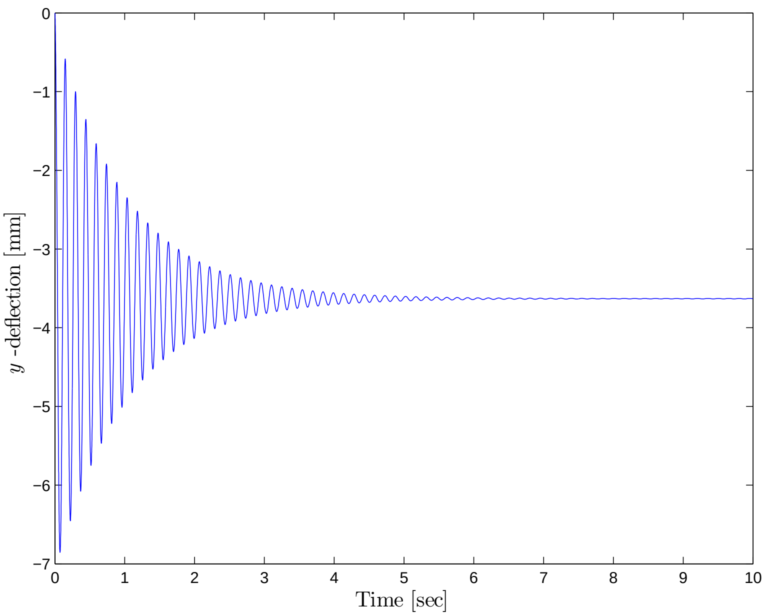

Figure 15. Response of lumped-parameter model on a spring.

4.5.2 FEA Model

The model `fea_cantilever_body_spring.mdl` implements the FEA approach. As shown in Figure 16, the position measurement of the load body is the superposition of the resulting two normal modes of this coupled oscillator system. The mass of the beam in this example is carried entirely by the rigid SimMechanics body, which causes the deflection of the wall spring; we still assume the beam weight does not contribute to flexing. Note, too, that as soon as the cantilever rotates away from the horizontal, the reaction force is no longer purely in the y direction; however, taking the y-component of the reaction force on the cantilever in the *local* coordinate system assures that we always use the component of the force that we modeled in the state-space system. With this approach, the steady-state deflection of the system is $\delta^{(\mathrm{FEA})} = -3.20\,\mathrm{mm}$, while the oscillation frequency is $f^{(\mathrm{FEA})} = 8.76\,\mathrm{Hz}$.

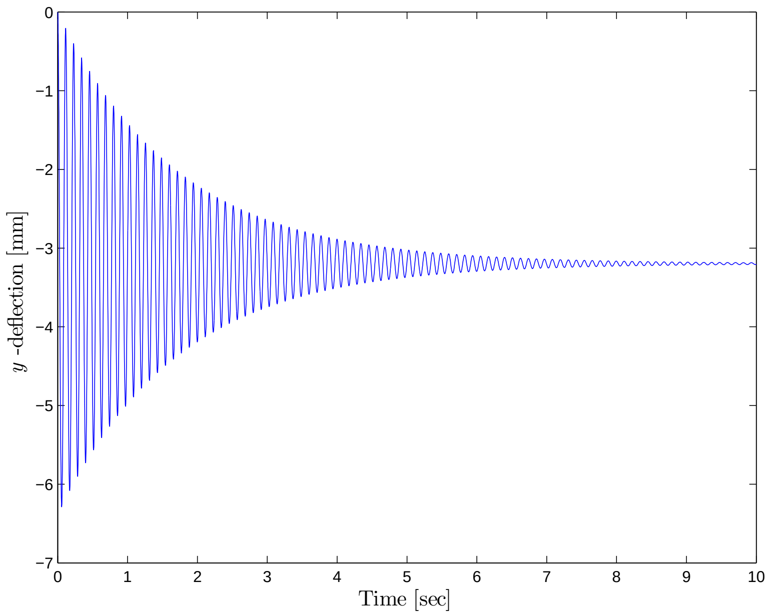

Figure 16. Response of the FEA model on a spring.

5 Conclusion

The results in these examples show a slight frequency discrepancy between the two approaches, which is probably due to the simple discretization scheme that we chose for the lumped-parameter method. A more careful application would begin the numerical analysis from the fourth-order partial differential equation describing the dynamics of the flexible beam and discretize the beam accordingly. Nevertheless, the lumped-parameter approach is easily constructed from fundamental building blocks once the proper material parameters are computed and can yield useful insights into the flexible deformations of a model. The FEA approach, which is more easily applied to arbitrary geometries, relies on detailed flexibility analysis carried out externally and is, within the linear regime, as valid as that analysis and simulation precision permit.

As seen in the examples, both approaches can be made more manageable by using several modeling tools afforded by Simulink®and SimMechanics:

- Masked subsystems can encapsulate the complexity of details into a black box that we can use modularly without having to worry about the detailed implementation at the top level. For example, a GBE made into a mask can be welded to other similarly masked GBEs, and an FEA flexible body can be used as a building block for a larger model.

- Made into library blocks, these masked subsystems facilitate reuse and modification. For example, a GBE from the library can be instantiated multiple times in a model to construct the beam; a change in the parametrization of the GBE made in the library gets propagated automatically to each instance.[7] Similarly, a library block containing an FEA-approach implementation of a flexible body can be instantiated multiple times, and its parameters need only be changed in the library
- The data necessary to parametrize the flexible body models, such as material properties and geometry, can be organized into MATLAB® data structures. The model and M-file examples cited in this paper illustrate this technique.
- Body blocks enable the use of "Adjoining" coordinate systems. This choice makes the library blocks easily reusable by referring to their neighbors' coordinate systems rather than to absolute frames. The top-level system can thus be constructed from modular components.

References

[Bathe] Klaus-Jürgen Bathe, *Finite Element Procedures.* Prentice-Hall, NJ, 1996.

[Hatch] Michael R. Hatch, *Vibration Simulation using Matlab and ANSYS.* Chapman & Hall/CRC, FL, 2001.

[Reddy] J.N. Reddy, *An Introduction to the Finite Element Method.* McGraw-Hill, 1993.

[Crandall] Stephen Crandall, D. Karnopp, E. Kurtz, Jr., D. Pridmore-Brown, *Dynamics of Mechanical and Electromechanical Systems.* McGraw-Hill, 1982.

[7] Assuming the library link from the block in the model to the library remains unbroken.

List of Figures