Lehrstuhl für
Regelungstechnik

Technical Reports on Automatic Control
Vol. TRAC-4, Nov. 2009
Available online at: www.rt.mw.tum.de

TECHNISCHE
UNIVERSITÄT
MÜNCHEN

Generating a Parametric Finite Element Model of a 3D Cantilever Timoshenko Beam Using Matlab

Heiko Panzer, Jörg Hubele, Rudy Eid and Boris Lohmann
Institute of Automatic Control, Technische Universität München
Boltzmannstr. 15, D-85748 Garching, Germany

Abstract

In this note, a fast and easy way to produce a Finite Element model of a 3D cantilever Timoshenko beam in state space is presented, for instance for benchmarking purposes of parametric Model Order Reduction. The model is generated by a single MATLAB file and provides the possibility to modify several parameters like the beam's geometry, material properties and the number of degrees of freedom.

Keywords: FEM; Parametric Model Order Reduction; Timoshenko Beam

1 Introduction

Many techniques for the mathematical modeling of dynamical systems, e. g. for simulating, optimizing or controlling purposes, employ linear systems of differential equations. The desire for increasing accuracy of such models, that may for instance originate from a Finite Element method, and the often occurring effect of slow convergence lead to a rapidly growing number of describing equations that eventually can no more be solved and used efficiently. Model Order Reduction (MOR) techniques are designed to answer this problem by replacing the large scale original model by a considerably smaller one, preserving characteristic properties of the former and approximating its transfer behavior as accurately as possible [2]. This task turns out to be even more complex when the original model depends on parameters like geometry or material properties that are to be preserved in the reduced model. This problem is known as parametric Model Order Reduction (pMOR).

In [1, 3], several models of dynamical systems from different fields of engineering have been collected and made available for benchmarking purposes. Nevertheless, due to the high interest in pMOR and the various reduction methods, there is currently a need for parametric models of different complexities with one or several parameters.

In the following, an easy way to construct such a model ”manually” without the need of professional software besides Matlab is presented. Section 2 revisits the basic elastic theory for slender beams and allocates all quantities needed for the calculation of a local finite Timoshenko beam element. Section 3 shows how to assemble the global mass and stiffness matrices and a ready-to-run Matlab implementation is given in the Appendix.

Corresponding author: Tel: +49 89 289 15592, Fax: +49 89 289 15653, email: panzer@tum.de

2 Elastic Beam Theory and the Local Element Matrices

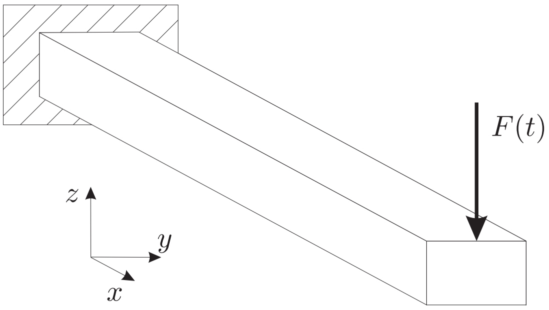

Figure 1: The cantilever beam

Consider a simply supported cantilever beam that is loaded by a dynamic force $F(t)$ applied vertically at its tip, as shown in Fig. 1.
Let the length of the beam be L, its cross section a rectangle of height h and thickness t, thus with area $A = h \cdot t$. We assume it is made of an isotropic, linear elastic material (e. g. steel), with density ρ, Young's modulus E and Poisson's ratio ν. Its mass is accordingly given by $m = L \cdot h \cdot t \cdot \rho$ and its shear modulus by $G = \frac{E}{2(\nu+1)}$.
Now, we separate the beam into N finite elements, each of which has length $l = \frac{L}{N}$ and mass $m_e = \frac{m}{N}$. The global numbering is carried out according to Fig. 2, thus the i-th Finite Element connects the $(i-1)^{\text{st}}$ and the i-th node. [5]

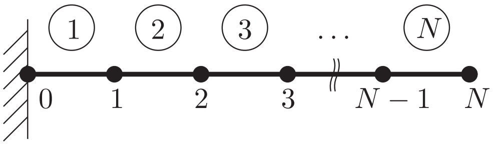

Figure 2: Global numbering of elements and nodes

For the 3D case, every node has six degrees of freedom (dof): Three translational displacements u, v, w and three rotational degrees of freedom α, β, γ (w. r. t. x-, y- and z-axis, respectively) as seen in Fig. 2.

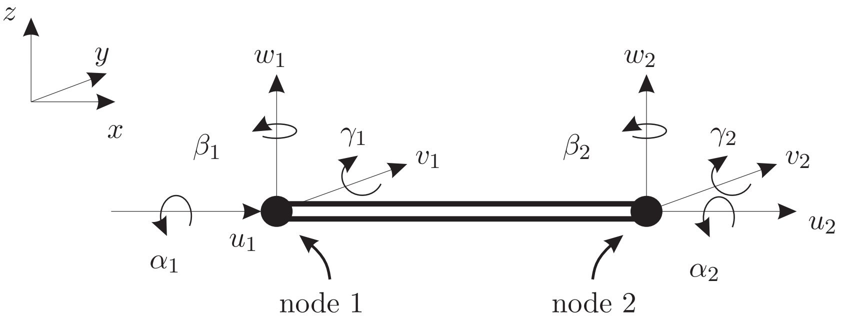

Figure 3: Local degrees of freedom

Recall that the element's second moments of area about y and z and its polar moments of inertia about x are given by

$$I_y = \frac{t\,h^3}{12}, \qquad I_z = \frac{t^3\,h}{12}, \qquad I_p = \frac{h^2 + t^2}{12}\frac{m}{N}.$$

For the element's torsion constant I_t, only an approximation can be given, as no exact analytical equations exist for non-circular cross-sections. In case of an almost square cross section, i. e. $h \approx t$, it is given by [4]

$$I_t = 0.141\,t\,h^3, \quad \text{if } h < t, \qquad \text{or} \qquad I_t = 0.141\,t^3\,h, \quad \text{if } h > t.$$

For simplification, define

$$P_y = \frac{12\,E\,I_z}{G\,A_{S,y}\,l^2}, \qquad P_z = \frac{12\,E\,I_y}{G\,A_{S,z}\,l^2},$$

where the effective area of shear for a rectangular cross-section is given by $A_{S,y} = A_{S,z} = \frac{5}{6}\,t\,h$. Note that setting $P_y = P_z = 0$ leads to an Euler Bernoulli beam model as the influence of shear stiffness is neglected.

We are now ready to compute the local mass and stiffness matrices, $\mathbf{M}^e_{11}$ and $\mathbf{K}^e_{11}$, of the finite Timoshenko beam element as presented in [6]. Remember that each element links twelve local dofs $\{u_1, v_1, w_1, \alpha_1, \beta_1, \gamma_1, u_2, v_2, w_2, \alpha_2, \beta_2, \gamma_2\}$, six belonging to each node. Therefore, we partition $\mathbf{M}^e_{11}$ and $\mathbf{K}^e_{11}$ into

$$\mathbf{M}^e = \begin{bmatrix} \mathbf{M}^e_{11} & \mathbf{M}^e_{12} \\ \mathbf{M}^e_{21} & \mathbf{M}^e_{22} \end{bmatrix}, \quad \mathbf{K}^e = \begin{bmatrix} \mathbf{K}^e_{11} & \mathbf{K}^e_{12} \\ \mathbf{K}^e_{21} & \mathbf{K}^e_{22} \end{bmatrix}, \tag{1}$$

where $(\cdot)^e_{11}$ and $(\cdot)^e_{22}$ describe the interdependency of the dofs of the first and second node, respectively, and $(\cdot)^e_{12} = (\cdot)^e_{21}$ the interconnection between both of them [5]. $\mathbf{M}^e_{22}$ and $\mathbf{K}^e_{22}$ are equal to $\mathbf{M}^e_{11}$ and $\mathbf{K}^e_{11}$, respectively, except for the sign of the off-diagonal entries.

$$\mathbf{M}^e_{11} = m_e \begin{bmatrix} \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{13}{35} + \frac{6\,I_z}{5\,Al^2} & 0 & 0 & 0 & \frac{11\,l}{210} + \frac{I_z}{10\,Al} \\ 0 & 0 & \frac{13}{35} + \frac{6\,I_y}{5\,Al^2} & 0 & -\frac{11\,l}{210} - \frac{I_y}{10\,Al} & 0 \\ 0 & 0 & 0 & \frac{I_p}{3\,A} & 0 & 0 \\ 0 & 0 & -\frac{11\,l}{210} - \frac{I_y}{10\,Al} & 0 & \frac{l^2}{105} + \frac{2\,I_y}{15\,A} & 0 \\ 0 & \frac{11\,l}{210} + \frac{I_z}{10\,Al} & 0 & 0 & 0 & \frac{l^2}{105} + \frac{2\,I_z}{15\,A} \end{bmatrix} \tag{2}$$

$$\mathbf{M}^e_{21} = {\mathbf{M}^e_{12}}^T = m_e \begin{bmatrix} \frac{1}{6} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{9}{70} - \frac{6\,I_z}{5\,Al^2} & 0 & 0 & 0 & +\frac{13\,l}{420} - \frac{I_z}{10\,Al} \\ 0 & 0 & \frac{9}{70} - \frac{6\,I_y}{5\,Al^2} & 0 & -\frac{13\,l}{420} + \frac{I_y}{10\,Al} & 0 \\ 0 & 0 & 0 & \frac{I_p}{6\,A} & 0 & 0 \\ 0 & 0 & \frac{13\,l}{420} - \frac{I_y}{10\,Al} & 0 & -\frac{l^2}{140} - \frac{2\,I_y}{30\,A} & 0 \\ 0 & -\frac{13\,l}{420} + \frac{I_z}{10\,Al} & 0 & 0 & 0 & -\frac{l^2}{140} - \frac{2\,I_z}{30\,A} \end{bmatrix} \tag{3}$$

$$\mathbf{K}_{11}^e = \frac{1}{l^3}\begin{bmatrix} Al^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \dfrac{12EI_z}{1+P_y} & 0 & 0 & 0 & \dfrac{6EI_zl}{1+P_y} \\ 0 & 0 & \dfrac{12EI_y}{1+P_z} & 0 & -\dfrac{6EI_yl}{1+P_z} & 0 \\ 0 & 0 & 0 & GI_tl^2 & 0 & 0 \\ 0 & 0 & -\dfrac{6EI_yl}{1+P_z} & 0 & \dfrac{EI_yl^2(4+P_z)}{1+P_z} & 0 \\ 0 & \dfrac{6EI_zl}{1+P_y} & 0 & 0 & 0 & \dfrac{EI_zl^2(4+P_y)}{1+P_y} \end{bmatrix} \tag{4}$$

$$\mathbf{K}_{21}^e = \mathbf{K}_{12}^{e\,T} = \frac{1}{l^3}\begin{bmatrix} -EAl^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\dfrac{12EI_z}{1+P_y} & 0 & 0 & 0 & -\dfrac{6EI_zl}{1+P_y} \\ 0 & 0 & -\dfrac{12EI_y}{1+P_z} & 0 & \dfrac{6EI_yl}{1+P_z} & 0 \\ 0 & 0 & 0 & -GI_tl^2 & 0 & 0 \\ 0 & 0 & -\dfrac{6EI_yl}{1+P_z} & 0 & \dfrac{EI_yl^2(2-P_z)}{1+P_z} & 0 \\ 0 & \dfrac{6EI_zl}{1+P_y} & 0 & 0 & 0 & \dfrac{EI_zl^2(2-P_y)}{1+P_y} \end{bmatrix} \tag{5}$$

3 Assembly of Global Matrices and Insertion of Boundary Conditions

The next step is the assembly of the global mass and stiffness matrices $\mathbf{M}$ and $\mathbf{K}$.
Due to the simple geometry of the beam, no coordinate transformations are needed. In addition, the particularly easy allocation from local to global dofs according to the numbering shown in Fig. 2 allows us to immediately state the assembled matrices $\mathbf{M}^A$ and $\mathbf{K}^A$ to be

$$\mathbf{M}^A = \begin{bmatrix} \mathbf{M}_{11}^e & \mathbf{M}_{12}^e & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{M}_{21}^e & \mathbf{M}_{22}^e + \mathbf{M}_{11}^e & \mathbf{M}_{12}^e & & \\ \mathbf{0} & \mathbf{M}_{21}^e & \mathbf{M}_{11}^e + \mathbf{M}_{22}^e & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \\ & & & \mathbf{M}_{22}^e + \mathbf{M}_{11}^e & \mathbf{M}_{12}^e \\ \mathbf{0} & \cdots & & \mathbf{M}_{21}^e & \mathbf{M}_{22}^e \end{bmatrix}, \tag{6}$$

$$\mathbf{K}^A = \begin{bmatrix} \mathbf{K}_{11}^e & \mathbf{K}_{12}^e & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{K}_{21}^e & \mathbf{K}_{22}^e + \mathbf{K}_{11}^e & \mathbf{K}_{12}^e & & \\ \mathbf{0} & \mathbf{K}_{21}^e & \mathbf{K}_{11}^e + \mathbf{K}_{22}^e & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \\ & & & \mathbf{K}_{22}^e + \mathbf{K}_{11}^e & \mathbf{K}_{12}^e \\ \mathbf{0} & \cdots & & \mathbf{K}_{21}^e & \mathbf{K}_{22}^e \end{bmatrix}. \tag{7}$$

The cantilever beam is supported at its left end where neither deflection nor slope may occur. Accordingly, we demand all degrees of freedom of the very left node to be 0 (Dirichlet Boundary Condition). This complies to removing the first six rows and columns of $\mathbf{M}^A$ and $\mathbf{K}^A$ to get our final mass and stiffness matrices $\mathbf{M}$ and $\mathbf{K}$ [5].

The remaining degrees of freedom can be collected to the global state vector

$$\mathbf{z} = \left[u^{(1)}v^{(1)}w^{(1)}\alpha^{(1)}\beta^{(1)}\gamma^{(1)}\ u^{(2)}v^{(2)}w^{(2)}\alpha^{(2)}\beta^{(2)}\gamma^{(2)}\ \ldots\ u^{(N)}v^{(N)}w^{(N)}\alpha^{(N)}\beta^{(N)}\gamma^{(N)}\right]^T.$$

The load $F(t)$, that acts on the tip of the beam along the negative z-axis, affects $w^{(N)}$, which is the fourth from last state. We choose this displacement to be also the output. Accordingly, the global load vector $\hat{\mathbf{b}}$ and the output vector $\hat{\mathbf{c}}$ are given by

$$-\hat{\mathbf{b}}^T = \hat{\mathbf{c}} = \begin{bmatrix}\mathbf{0}^T, & \ldots & , & \mathbf{0}^T, & 0 & 0 & 1 & 0 & 0 & 0\end{bmatrix}.$$

Note that in static Finite Elements, the discrete application of forces leads to convergence problems. For dynamic considerations however, this is a valid simplification. Note further that due to the decoupling between transversal bending deformations along y and z, longitudinal displacement and torsion, only part of the states will be excited for the given load vector $\hat{\mathbf{b}}$. Naturally, it is possible to augment the input and output vectors with further columns and rows, respectively, in order to obtain a MIMO model.
Finally, we assume the dampening matrix $\mathbf{D}$ to be given by (Rayleigh damping)

$$\mathbf{D} = d_1\mathbf{M} + d_2\mathbf{K}. \tag{8}$$

This leads us to the second-order system

$$\begin{aligned} \mathbf{M}\,\ddot{\mathbf{z}}(t) + \mathbf{D}\,\dot{\mathbf{z}}(t) + \mathbf{K}\,\mathbf{z}(t) &= \hat{\mathbf{b}}\,F(t), \qquad (9)\\ y(t) &= \hat{\mathbf{c}}^T\,\mathbf{z}(t), \qquad (10)\end{aligned}$$

which can be rewritten as a state space model

$$\left\{\begin{aligned} \overbrace{\begin{bmatrix}\mathbf{F} & \mathbf{0}\\ \mathbf{0} & \mathbf{M}\end{bmatrix}}^{\mathbf{E}} \overbrace{\begin{bmatrix}\dot{\mathbf{z}}\\ \ddot{\mathbf{z}}\end{bmatrix}}^{\dot{\mathbf{x}}}(t) &= \overbrace{\begin{bmatrix}\mathbf{0} & \mathbf{F}\\ -\mathbf{K} & -\mathbf{D}\end{bmatrix}}^{\mathbf{A}} \overbrace{\begin{bmatrix}\mathbf{z}\\ \dot{\mathbf{z}}\end{bmatrix}}^{\mathbf{x}}(t) + \overbrace{\begin{bmatrix}\mathbf{0}\\ \hat{\mathbf{b}}\end{bmatrix}}^{\mathbf{b}} F(t)\\ y(t) &= \underbrace{\begin{bmatrix}\hat{\mathbf{c}}^T & \mathbf{0}^T\end{bmatrix}}_{\mathbf{c}^T} \begin{bmatrix}\mathbf{z}\\ \dot{\mathbf{z}}\end{bmatrix}(t), \end{aligned}\right. \tag{11}$$

where $\mathbf{F} \in \mathbb{R}^{6N\times 6N}$ is an arbitrary nonsingular matrix [8].

Remarks for use in MOR:

- For the aim of stability preservation during the MOR process, it can be advisable to choose $\mathbf{F} = \mathbf{K}$ in (11). This makes $\mathbf{A}$ contractive and $\mathbf{E}$ positive definite, which, according to [7], guarantees a stable reduced system for any one-sided projection method like one-sided Arnoldi.
- $V = \mathbf{x}^T\mathbf{Q}\mathbf{x}$ with $\mathbf{Q} = \mathbf{E}^{-1}$ is then a Lyapunov function of system (11).
- Besides, changing the output matrix to $\mathbf{c}^T = \mathbf{b}^T\mathbf{Q}$, (11) is a Port Hamiltonian system where $\mathbf{J} = \begin{bmatrix}\mathbf{0} & \mathbf{K}\\ -\mathbf{K} & \mathbf{0}\end{bmatrix}$ and $\mathbf{R} = \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}\end{bmatrix}$.

4 Conclusions

We have presented a thorough way to produce state space models of a 3D cantilever Timoshenko beam that can be easily implemented. A ready-to-run Matlab code, that can be adapted straightforward to match the user's needs, is given in the Appendix. The function can be modified to have any of the quantities L, t, d, ρ, E, ν, $\ldots$, as well as N, the number of elements, as parameters.

For more MATLAB codes and further information, please visit our website: www.rt.mw.tum.de.

References

[1] *Oberwolfach Model Reduction Benchmark Collection.* available online at http://www.imtek.uni-freiburg.de/simulation/benchmark/.

[2] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems.* SIAM, Philadelphia, 2005.

[3] Y. Chahlaoui and P. Van Dooren. A collection of Benchmark Examples for Model Reduction of Linear Time Invariant Dynamical Systems. Slicot working note, 2002. Available at ftp://wgs.esat.kuleuven.ac.be/pub/WGS/ REPORTS/SLWN2002-2.ps.Z.

[4] K.-H. Grote and J. Feldhusen. *Dubbel, Taschenbuch für den Maschinenbau.* Springer, 21. Auflage, 2004.

[5] Y. W. Kwon and H. Bang. *The Finite Element Method Using MATLAB.* Crc Pr Inc., 2nd Edition, 2000.

[6] J. S. Przemieniecki. *Theory of Matrix Structural Analysis.* McGraw-Hill Book Company, 1968.

[7] R. Castañé-Selga, R. Eid, and B. Lohmann. A generalized theory for stability preservation in model order reduction of large scale systems. *(In preparation).*

[8] B. Salimbahrami. *Structure Preserving Order Reduction of Large Scale Second Order Models.* PhD thesis, Institute of Automatic Control, Technical University of Munich, 2005.

Appendix A: Implementation in MATLAB

```
function sys = fem_beam(L)
% Generates a FEM model of a 3D cantilever Timoshenko beam
% Replace arbitrarily many of the preset parameters by inputs to the file!
%
% Please cite this model as:
% H. Panzer, J. Hubele et al.:
% Generating a Parametric Finite Element Model
%   of a 3D Cantilever Timoshenko Beam Using Matlab
% Technical reports on Automatic Control, vol. TRAC−4,
% Institute of Automatic Control, Technische Universität München, 2009
%
% Usage example: sys = fem_beam(1); % for beam of length L=1m
%
```

```
% Authors:      Jörg Hubele (joerg.hubele@mytum.de)
%               Heiko Panzer (panzer@tum.de)
% Last Change:  20 Nov 2009
%
% Visit our webpage for more code downloads: www.rt.mw.tum.de

% ========== parameters ==========

% (input) L:          total beam length (x) [m]
N = 20;             % number of elements [1]
t = .01;            % t: beam thickness (y) [m]
h = .01;            % h: beam height (z) [m]
rho = 7850;         % density of steel [kg/m^3]
m = L*t*h*rho;      % m: total beam mass [kg]
E = 210e9;          % E: Young's modulus of steel [N/m^2]
nu = 3/10;          % nu: Poisson's ratio
d1 = 8e-6;  d2 = 8; % d1, d2: dampening ratio (D = d1*K + d2*M)


% ========== physical quantities ==========

G = E/2/(1+nu);                         % G: Shear modulus [N/m^2]
l = L/N;                                % l: beam element length
A = t*h ;                               % beam area [m^2]
ASy = 5/6*A ;   ASz = 5/6*A ;           % effective area of shear
Iy = 1/12*h^3*t ;   Iz = 1/12*t^3*h;    % second moments of area [m^4]
Ip = 1/12*t*h*(h^2+t^2);                % polar moment of inertia [m^4]
It = min([h t])^3*max([h t])/7;         % torsion constant [m^4]
Py = 12*E*Iz/(G*ASy*l^2) ;  Pz = 12*E*Iy/(G*ASz*l^2) ;  % Phi

% ========== element mass and stiffness matrices ==========

% element mass matrix

M11 = zeros(6,6);
M11(1,1) = 1/3;
M11(2,2) = 13/35 + 6*Iz/(5*A*l^2);
M11(3,3) = 13/35 + 6*Iy/(5*A*l^2);
M11(4,4) = Ip/(3*A);
M11(5,5) = l^2/105 + 2*Iy/(15*A);
M11(6,6) = l^2/105 + 2*Iz/(15*A);
M11(6,2) = 11*l/210 + Iz/(10*A*l);
M11(2,6) = M11(6,2) ;
M11(5,3) = -11*l/210 - Iy/(10*A*l);
M11(3,5) = M11(5,3) ;

M22 = -M11 + 2*diag(diag(M11));

M21 = zeros(6,6);
M21(1,1) = 1/6;
M21(2,2) = 9/70 - 6*Iz/(5*A*l^2);
M21(3,3) = 9/70 - 6*Iy/(5*A*l^2);
M21(4,4) = Ip/(6*A);
M21(5,5) = -l^2/140 - Iy/(30*A);
M21(6,6) = -l^2/140 - Iz/(30*A);
M21(6,2) = -13*l/420 + Iz/(10*A*l);
M21(2,6) = -M21(6,2);
M21(5,3) = 13*l/420 - Iy/(10*A*l);
M21(3,5) = -M21(5,3);

Me = m/N*[M11, M21'; M21, M22];
```

```
% element stiffness matrix

K11 = zeros(6,6);
K11(1,1) = E*A/l ;
K11(2,2) = 12*E*Iz/(l^3*(1+Py)) ;
K11(3,3) = 12*E*Iy/(l^3*(1+Pz)) ;
K11(4,4) = G*It/l ;
K11(5,5) = (4+Pz)*E*Iy/(l*(1+Pz)) ;
K11(6,6) = (4+Py)*E*Iz/(l*(1+Py)) ;
K11(2,6) = 6*E*Iz/(l^2*(1+Py)) ;
K11(6,2) = K11(2,6) ;
K11(3,5) = -6*E*Iy/(l^2*(1+Pz)) ;
K11(5,3) = K11(3,5) ;

K22 = -K11 + 2*diag(diag(K11));

K21 = K11 - 2*diag(diag(K11));
K21(5,5) = (2-Pz)*E*Iy/(l*(1+Pz)) ;
K21(6,6) = (2-Py)*E*Iz/(l*(1+Py)) ;
K21(2,6) = -K21(6,2);
K21(3,5) = -K21(5,3);

Ke = [K11, K21'; K21, K22];

% ========== assembly ==========

% global mass and stiffness matrices: N*(6+1) dofs
M = zeros(6*(N+1), 6*(N+1));
K = zeros(6*(N+1), 6*(N+1));

% assembly
for i = 1:N
    a=1+6*(i-1):6*(i+1);
    K(a,a) = K(a,a) + Ke;
    M(a,a) = M(a,a) + Me;
end

% boundary condition
K(1:6,:)=[]; K(:,1:6)=[];
M(1:6,:)=[]; M(:,1:6)=[];

% dampening matrix
D = d1*K + d2*M ;

% global load and output vectors
B=zeros(6*N,1);    C=zeros(1,6*N);
B(N*6-3,1)=-1;        % input:  beam tip, -z
C(1,N*6-3)=1;         % output: beam tip, +z


% ========== second order --> first order ==========

F = eye(6*N);
E = [F zeros(6*N,6*N); zeros(6*N,6*N) M];
A = [zeros(6*N,6*N) F; -K -D] ;
B = [zeros(6*N,size(B,2)); B];
C = [C, zeros(size(C,1), 6*N)];

sys = dss(A,B,C,0,E);

end
```