NASS - Finite Element Models with Simscape

Dehaeze Thomas

February 7, 2023

Contents

In this document, Finite Element Models (FEM) of parts of the Nano-Hexapod are developed and integrated into Simscape for dynamical analysis.

It is divided in the following sections:

- Section 1: A super-element of the Amplified Piezoelectric Actuator APA300ML used for the NASS is exported using Ansys and imported in Simscape. The static and dynamical properties of the APA300ML are then estimated using the Simscape model.

- Section 2: A first geometry of a Flexible joint is modelled and its characteristics are identified from the Stiffness matrix as well as from the Simscape model.

- Section 3: An optimized flexible joint is developed for the Nano-Hexapod and is then imported in a Simscape model.

- Section 4: A super element of a complete strut is studied.

1 APA300ML

In this section, the Amplified Piezoelectric Actuator APA300ML (doc) is modeled using a Finite Element Software. Then a *super element* is exported and imported in Simscape where its dynamic is studied.

A 3D view of the Amplified Piezoelectric Actuator (APA300ML) is shown in Figure 1.1. The remote point used are also shown in this figure.

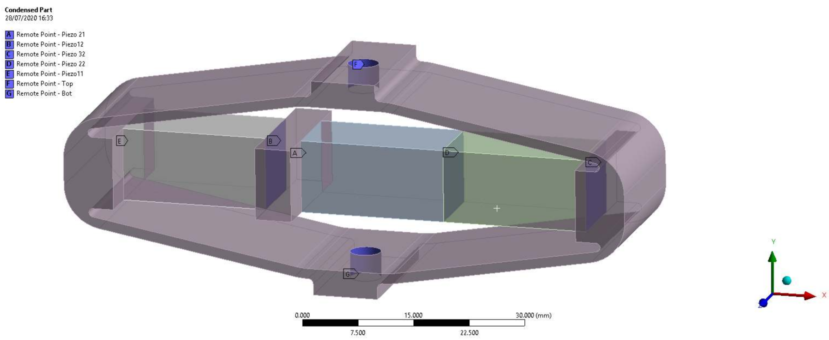

Figure 1.1: Ansys FEM of the APA300ML

1.1 Import Mass Matrix, Stiffness Matrix, and Interface Nodes Coordinates

We first extract the stiffness and mass matrices.

```matlab
K = readmatrix('APA300ML_mat_K.CSV');
M = readmatrix('APA300ML_mat_M.CSV');
```

Table 1.1: First 10x10 elements of the Stiffness matrix

200000000	30000	-20000	-70	300000	40	10000000	10000	-6000	30
30000	30000000	2000	-200000	60	-10	4000	2000000	-500	9000
-20000	2000	7000000	-10	-30	10	6000	900	-500000	3
-70	-200000	-10	1000	-0.1	08	-20	-9000	3	-30
300000	60	-30	-0.1	900	0.1	30000	20	-10	06
40	-10	10	08	0.1	10000	20	9	-5	03
10000000	4000	6000	-20	30000	20	200000000	10000	9000	50
10000	2000000	900	-9000	20	9	10000	30000000	-500	200000
-6000	-500	-500000	3	-10	-5	9000	-500	7000000	-2
30	9000	3	-30	06	03	50	200000	-2	1000

Then, we extract the coordinates of the interface nodes.

Table 1.2: First 10x10 elements of the Mass matrix

0.01	-2e-06	1e-06	6e-09	5e-05	-5e-09	-0.0005	-7e-07	6e-07	-3e-09
-2e-06	0.01	8e-07	-2e-05	-8e-09	2e-09	-9e-07	-0.0002	1e-08	-9e-07
1e-06	8e-07	0.009	5e-10	1e-09	-1e-09	-5e-07	3e-08	6e-05	1e-10
6e-09	-2e-05	5e-10	3e-07	2e-11	-3e-12	3e-09	9e-07	-4e-10	3e-09
5e-05	-8e-09	1e-09	2e-11	6e-07	-4e-11	-1e-06	-2e-09	1e-09	-8e-12
-5e-09	2e-09	-1e-09	-3e-12	-4e-11	1e-07	-2e-09	-1e-09	-4e-10	-5e-12
-0.0005	-9e-07	-5e-07	3e-09	-1e-06	-2e-09	0.01	1e-07	-3e-07	-2e-08
-7e-07	-0.0002	3e-08	9e-07	-2e-09	-1e-09	1e-07	0.01	-4e-07	2e-05
6e-07	1e-08	6e-05	-4e-10	1e-09	-4e-10	-3e-07	-4e-07	0.009	-2e-10
-3e-09	-9e-07	1e-10	3e-09	-8e-12	-5e-12	-2e-08	2e-05	-2e-10	3e-07

Matlab
```
[int_xyz, int_i, n_xyz, n_i, nodes] = extractNodes('APA300ML_out_nodes_3D.txt');
```

Table 1.3: Coordinates of the interface nodes

Node i	Node Number	x [m]	y [m]	z [m]
1.0	697783	0.0	0.0	-0.015
2.0	697784	0.0	0.0	0.015
3.0	697785	-0.0325	0.0	0.0
4.0	697786	-0.0125	0.0	0.0
5.0	697787	-0.0075	0.0	0.0
6.0	697788	0.0125	0.0	0.0
7.0	697789	0.0325	0.0	0.0

Table 1.4: Some extracted parameters of the FEM

Total number of Nodes	7
Number of interface Nodes	7
Number of Modes	120
Size of M and K matrices	162

Using `K`, `M` and `int_xyz`, we can now use the `Reduced Order Flexible Solid` simscape block.

1.2 Piezoelectric parameters

In order to make the conversion from applied voltage to generated force or from the strain to the generated voltage, we need to defined some parameters corresponding to the piezoelectric material:

Matlab
```
d33 = 600e-12; % Strain constant [m/V]
n   = 80;      % Number of layers per stack
eT  = 1.6e-8;  % Permittivity under constant stress [F/m]
sD  = 1e-11;   % Compliance under constant electric displacement [m2/N]
ka  = 235e6;   % Stack stiffness [N/m]
C   = 5e-6;    % Stack capactiance [F]
```

PZT-4

```matlab
d33 = 300e-12; % Strain constant [m/V]
n   = 80;      % Number of layers per stack
eT  = 5.3e-9;  % Permittivity under constant stress [F/m]
sD  = 1e-11;   % Compliance under constant electric displacement [m2/N]
ka  = 235e6;   % Stack stiffness [N/m]
C   = 5e-6;    % Stack capactiance [F]
```

The ratio of the developed force to applied voltage is:

$$F_a = g_a V_a, \quad g_a = d_{33} n k_a \tag{1.1}$$

where:

- F_a: developed force in [N]
- n: number of layers of the actuator stack
- d_{33}: strain constant in [m/V]
- k_a: actuator stack stiffness in [N/m]
- V_a: applied voltage in [V]

If we take the numerical values, we obtain:

```matlab
d33*n*ka % [N/V]
```

```
5.64
```

From [1] (page 123), the relation between relative displacement of the sensor stack and generated voltage is:

$$V_s = \frac{d_{33}}{\epsilon^T s^D n} \Delta h \tag{1.2}$$

where:

- V_s: measured voltage in [V]
- d_{33}: strain constant in [m/V]
- ϵ^T: permittivity under constant stress in [F/m]
- s^D: elastic compliance under constant electric displacement in [m^2/N]
- n: number of layers of the sensor stack
- Δh: relative displacement in [m]

If we take the numerical values, we obtain:

```
Matlab
1e-6*d33/(eT*sD*n) % [V/um]
```

```
Results
23.438
```

1.3 Simscape Model

The flexible element is imported using the `Reduced Order Flexible Solid` simscape block.

Let's say we use two stacks as a force sensor and one stack as an actuator:

- A `Relative Motion Sensor` block is added between the nodes A and C
- An `Internal Force` block is added between the remote points E and B

The interface nodes are shown in Figure 1.1.

One mass is fixed at one end of the piezo-electric stack actuator (remove point F), the other end is fixed to the world frame (remote point G).

1.4 Identification of the APA Characteristics

1.4.1 Stiffness

The transfer function from vertical external force to the relative vertical displacement is identified.

The inverse of its DC gain is the axial stiffness of the APA:

```
Matlab
1e-6/dcgain(G) % [N/um]
```

```
Results
1.753
```

The specified stiffness in the datasheet is $k = 1.8\,[N/\mu m]$.

1.4.2 Resonance Frequency

The resonance frequency is specified to be between 650Hz and 840Hz. This is also the case for the FEM model (Figure 1.2).

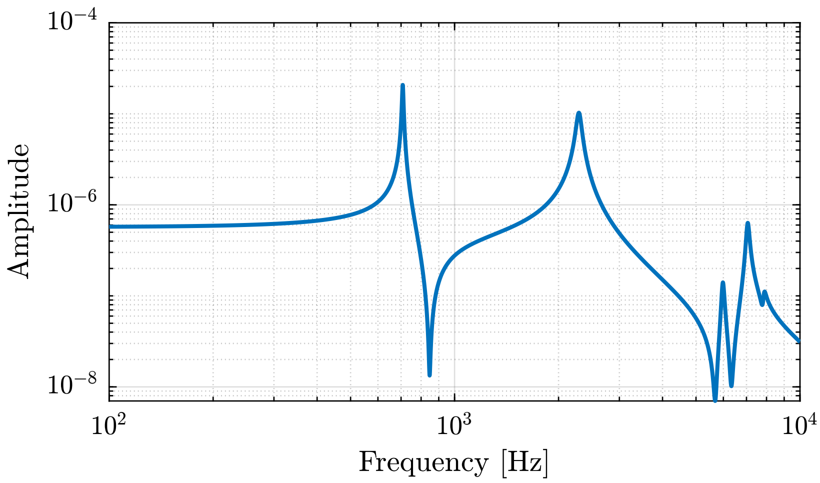

Figure 1.2: First resonance is around 800Hz

1.4.3 Amplification factor

The amplification factor is the ratio of the vertical displacement to the stack displacement.

The ratio of the two displacement is computed from the FEM model.

Matlab
```
abs(dcgain(G(1,1))./dcgain(G(2,1)))
```

Results
```
5.0749
```

This is actually correct and approximately corresponds to the ratio of the piezo height and length:

Matlab
```
75/15
```

Results
```
5
```

1.4.4 Stroke

Estimation of the actuator stroke:

$$\Delta H = An\Delta L$$

with:

- ΔH Axial Stroke of the APA

- A Amplification factor (5 for the APA300ML)
- n Number of stack used
- ΔL Stroke of the stack (0.1% of its length)

Matlab
```
1e6 * 5 * 3 * 20e-3 * 0.1e-2
```

Results
```
300
```

This is exactly the specified stroke in the data-sheet.

1.4.5 Stroke BIS

Identified the stroke form the transfer function from V to z

1.5 Identification of the Dynamics from actuator to replace displacement

We first set the mass to be approximately zero. The dynamics is identified from the applied force to the measured relative displacement. The same dynamics is identified for a payload mass of 10Kg.

Matlab
```
m = 10;
```

The root locus corresponding to Direct Velocity Feedback with a mass of 10kg is shown in Figure 1.4.

1.6 Identification of the Dynamics from actuator to force sensor

Let's use 2 stacks as a force sensor and 1 stack as force actuator.

The transfer function from actuator voltage to sensor voltage is identified and shown in Figure 1.5.

For root locus corresponding to IFF is shown in Figure 1.6.

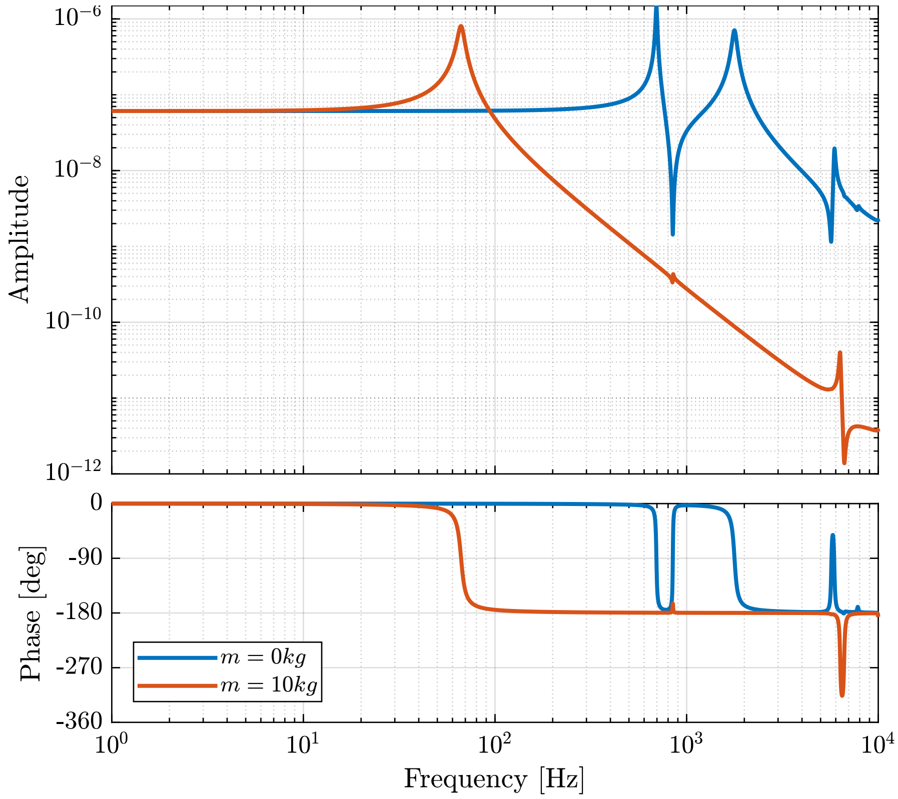

Figure 1.3: Transfer function from forces applied by the stack to the axial displacement of the APA

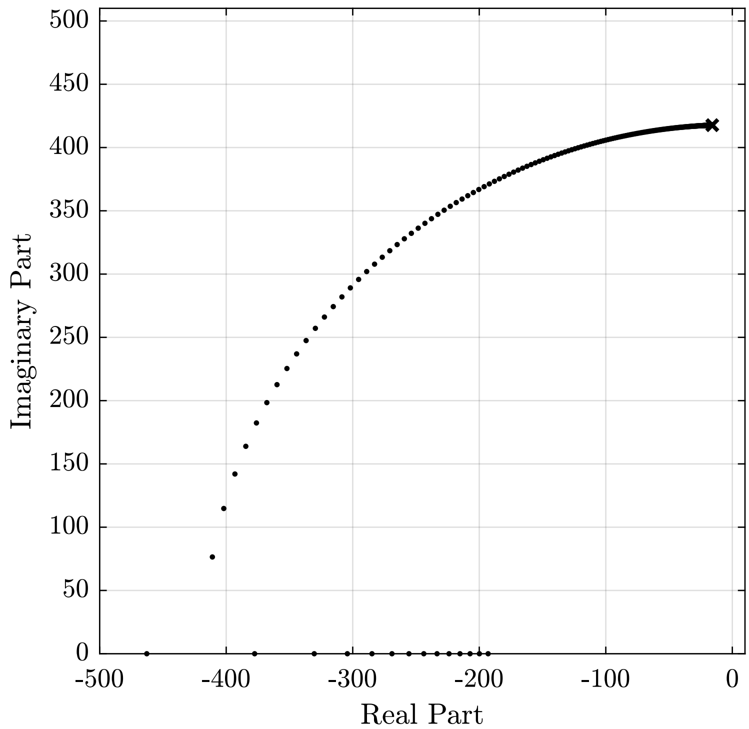

Figure 1.4: Root Locus for Direct Velocity Feedback

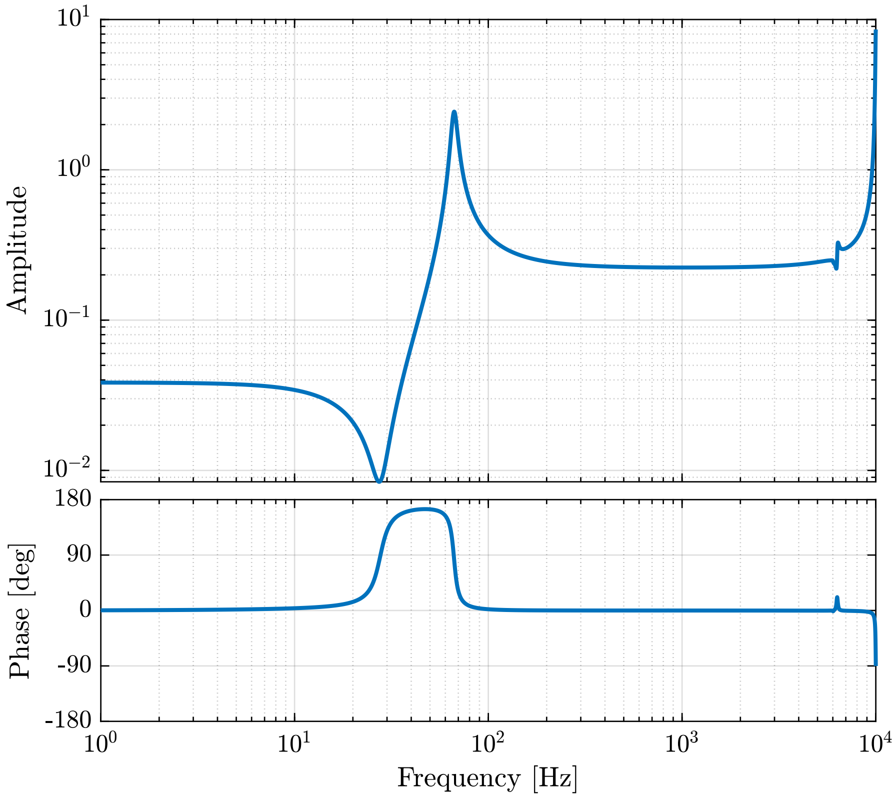

Figure 1.5: Transfer function from actuator to force sensor

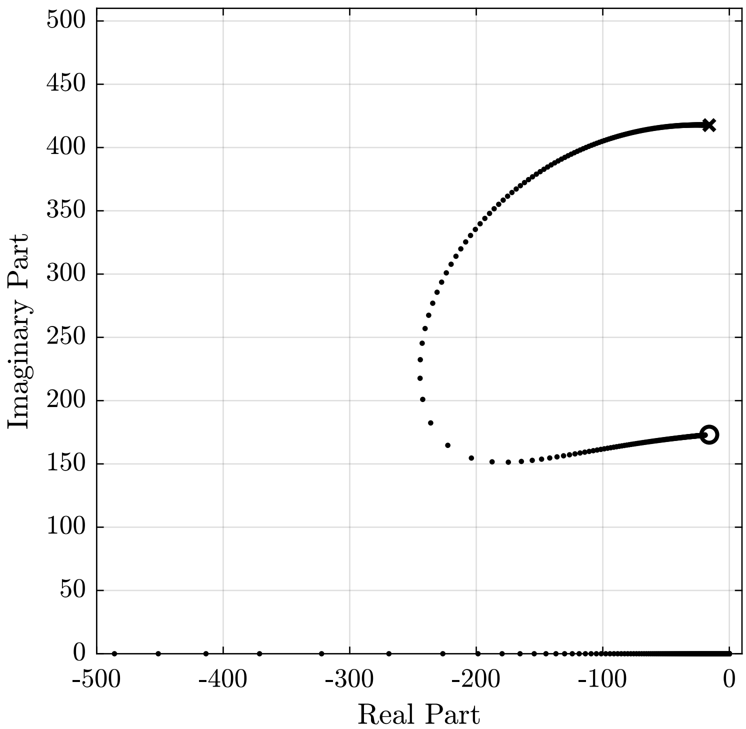

Figure 1.6: Root Locus for IFF

1.7 Identification for a simpler model

The goal in this section is to identify the parameters of a simple APA model from the FEM. This can be useful is a lower order model is to be used for simulations.

The presented model is based on [2].

The model represents the Amplified Piezo Actuator (APA) from Cedrat-Technologies (Figure 1.7). The parameters are shown in the table below.

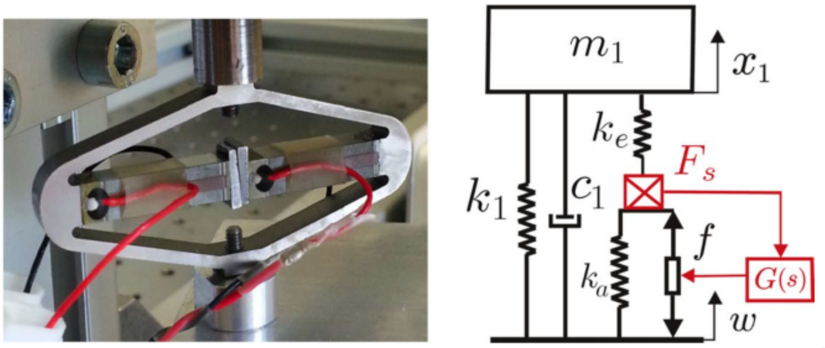

Figure 1.7: Picture of an APA100M from Cedrat Technologies. Simplified model of a one DoF payload mounted on such isolator

Table 1.5: Parameters used for the model of the APA 100M

	Meaning
k_e	Stiffness used to adjust the pole of the isolator
k_1	Stiffness of the metallic suspension when the stack is removed
k_a	Stiffness of the actuator
c_1	Added viscous damping

The goal is to determine k_e, k_a and k_1 so that the simplified model fits the FEM model.

$$\alpha = \frac{x_1}{f}(\omega = 0) = \frac{\frac{k_e}{k_e + k_a}}{k_1 + \frac{k_e k_a}{k_e + k_a}}$$

$$\beta = \frac{x_1}{F}(\omega = 0) = \frac{1}{k_1 + \frac{k_e k_a}{k_e + k_a}}$$

If we can fix k_a, we can determine k_e and k_1 with:

$$k_e = \frac{k_a}{\frac{\beta}{\alpha} - 1}$$

$$k_1 = \frac{1}{\beta} - \frac{k_e k_a}{k_e + k_a}$$

From the identified dynamics, compute α and β

Matlab
```
alpha = abs(dcgain(G('y', 'Fa')));
beta  = abs(dcgain(G('y', 'Fd')));
```

k_a is estimated using the following formula:

Matlab
```
ka = 0.8/abs(dcgain(G('y', 'Fa')));
```

The factor can be adjusted to better match the curves.

Then k_e and k_1 are computed.

Matlab
```
ke = ka/(beta/alpha - 1);
k1 = 1/beta - ke*ka/(ke + ka);
```

Table 1.6: Obtained stiffnesses of the APA 100M

	Value [N/um]
ka	40.5
ke	1.5
k1	0.4

The damping in the system is adjusted to match the FEM model if necessary.

Matlab
```
c1 = 1e2;
```

The analytical model of the simpler system is defined below:

Matlab
```
Ga = 1/(m*s^2 + k1 + c1*s + ke*ka/(ke + ka)) * ...
     [ 1 ,                k1 + c1*s + ke*ka/(ke + ka)  , ke/(ke + ka) ;
      -ke*ka/(ke + ka), ke*ka/(ke + ka)*m*s^2 ,       -ke/(ke + ka)*(m*s^2 + c1*s + k1)];

Ga.InputName = {'Fd', 'w', 'Fa'};
Ga.OutputName = {'y', 'Fs'};
```

And the DC gain is adjusted for the force sensor:

Matlab
```
F_gain = dcgain(G('Fs', 'Fd'))/dcgain(Ga('Fs', 'Fd'));
```

The dynamics of the FEM model and the simpler model are compared in Figure 1.8.

The simplified model has also been implemented in Simscape.

The dynamics of the Simscape simplified model is identified and compared with the FEM one in Figure 1.9.

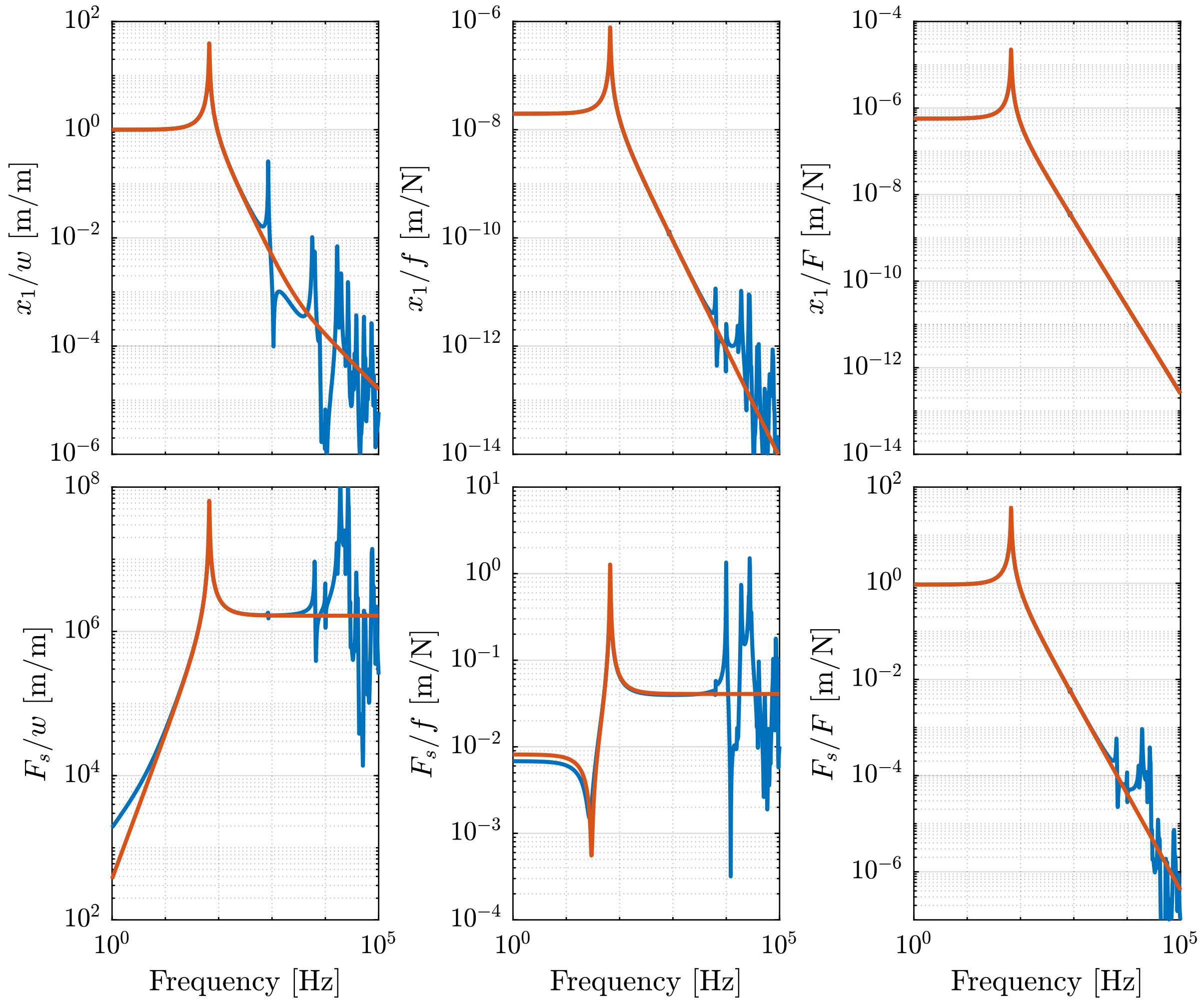

Figure 1.8: Comparison of the Dynamics between the FEM model and the simplified one

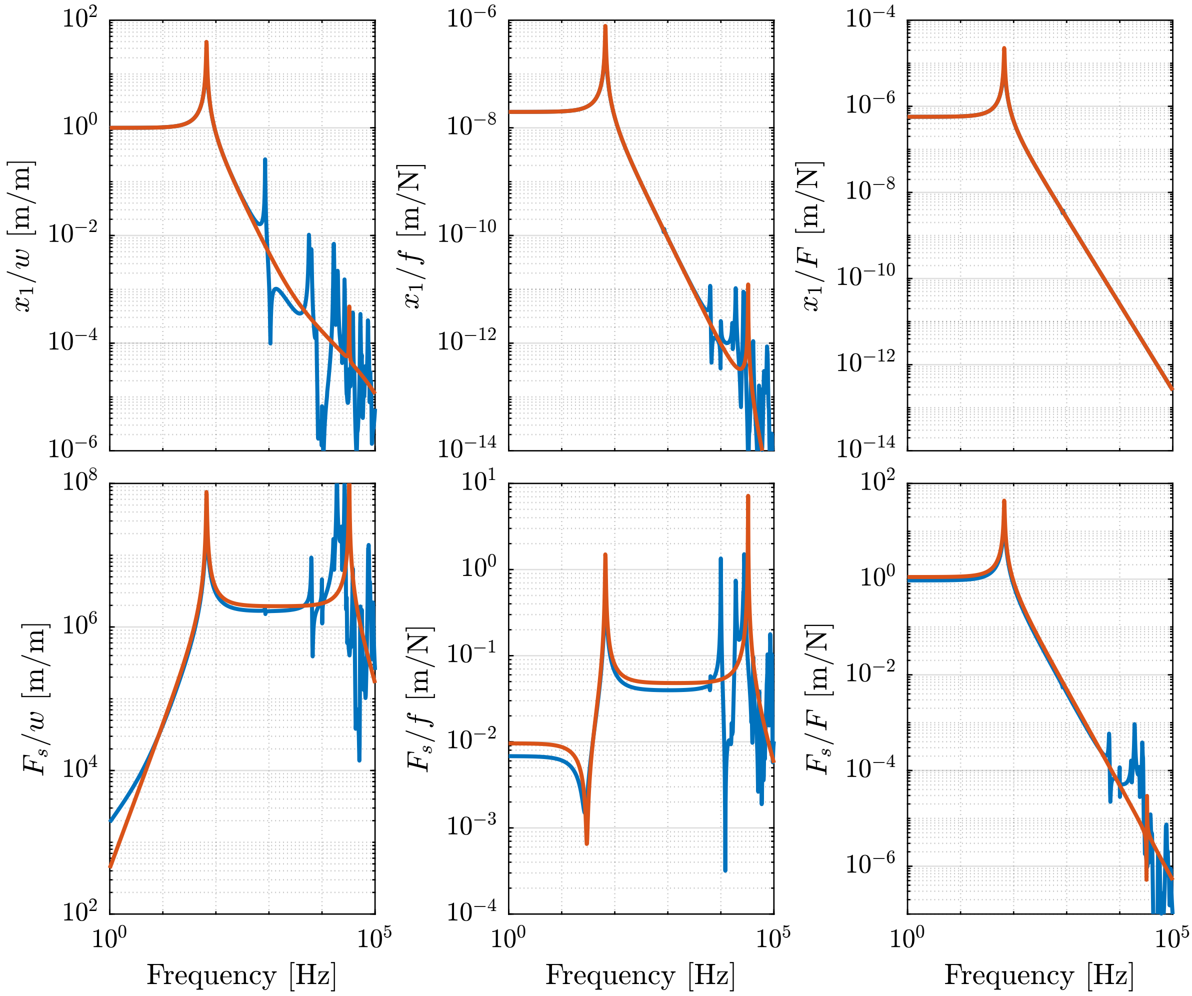

Figure 1.9: Comparison of the Dynamics between the FEM model and the simplified simscape model

1.8 Integral Force Feedback

In this section, Integral Force Feedback control architecture is applied on the APA300ML.

First, the plant (dynamics from voltage actuator to voltage sensor is identified). The payload mass is set to 10kg.

```
m = 10;
```

The obtained dynamics is shown in Figure 1.10.

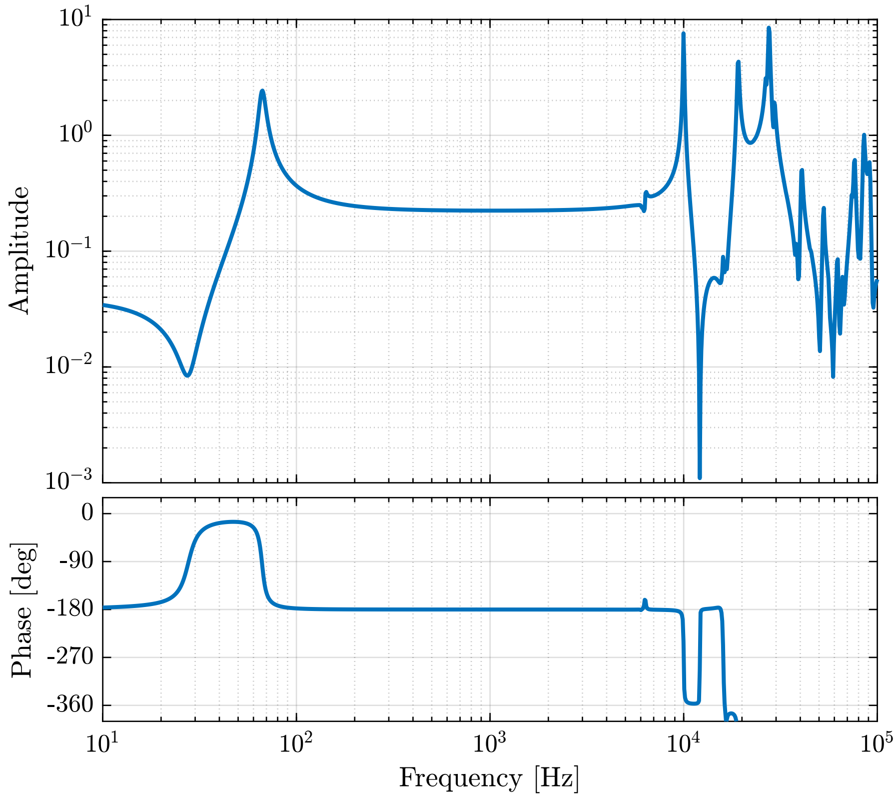

Figure 1.10: IFF Plant

The controller is defined below and the loop gain is shown in Figure 1.11.

```
Kiff = -1e3/s;
```

Now the closed-loop system is identified again and compare with the open loop system in Figure 1.12.

It is the expected behavior as shown in the Figure 1.13 (from [2]).

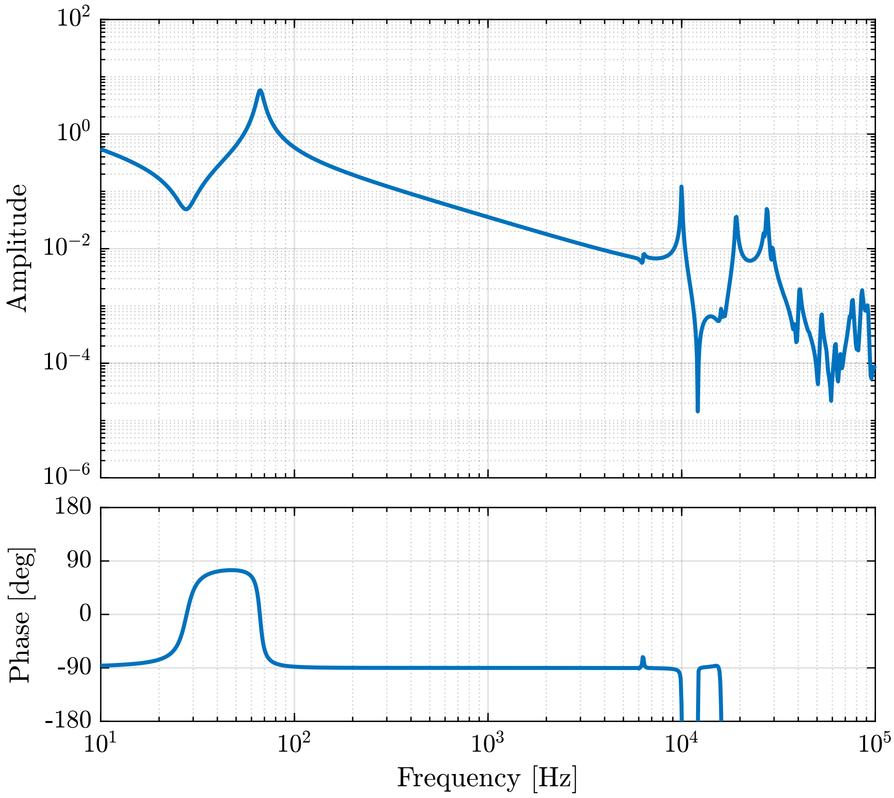

Figure 1.11: IFF Loop Gain

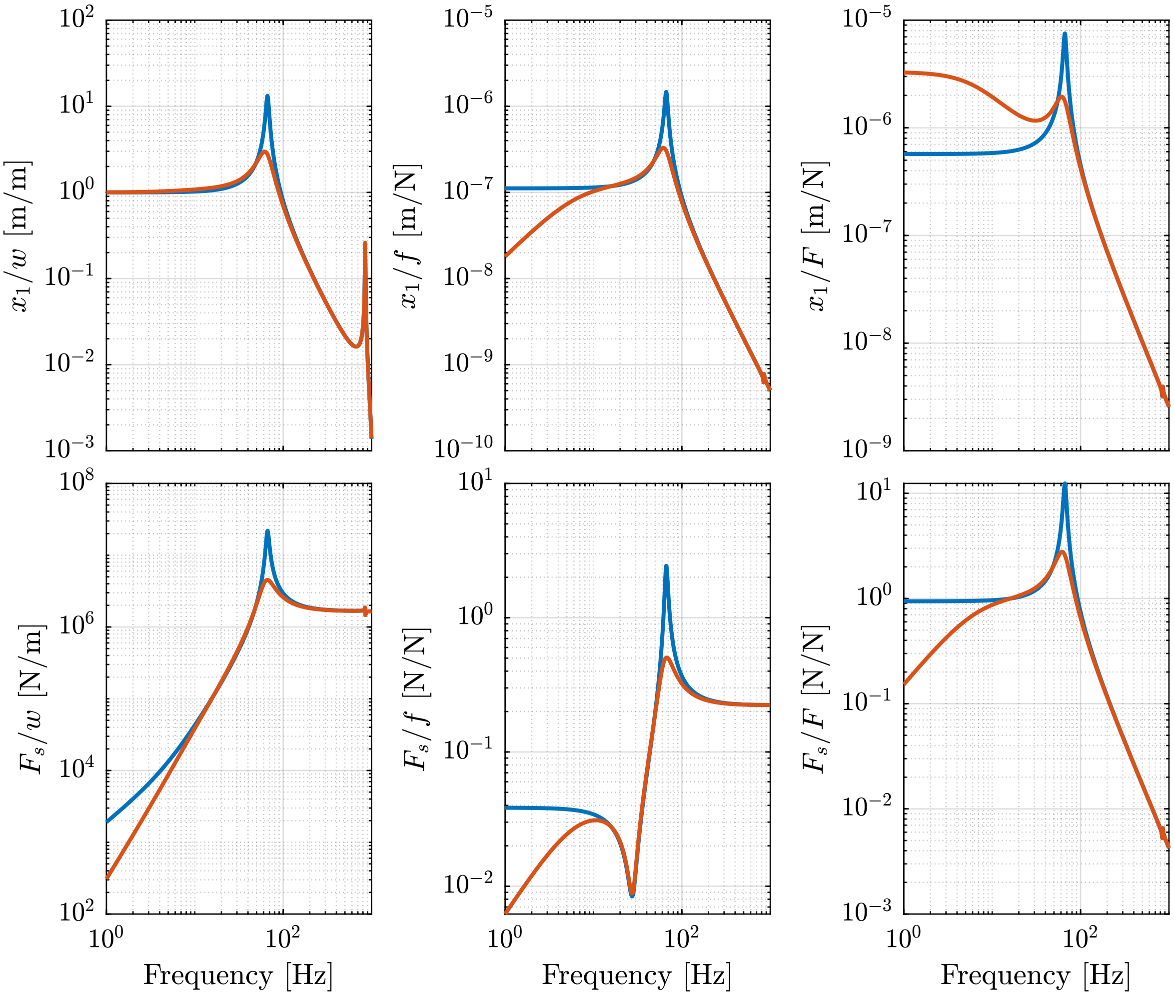

Figure 1.12: OL and CL transfer functions

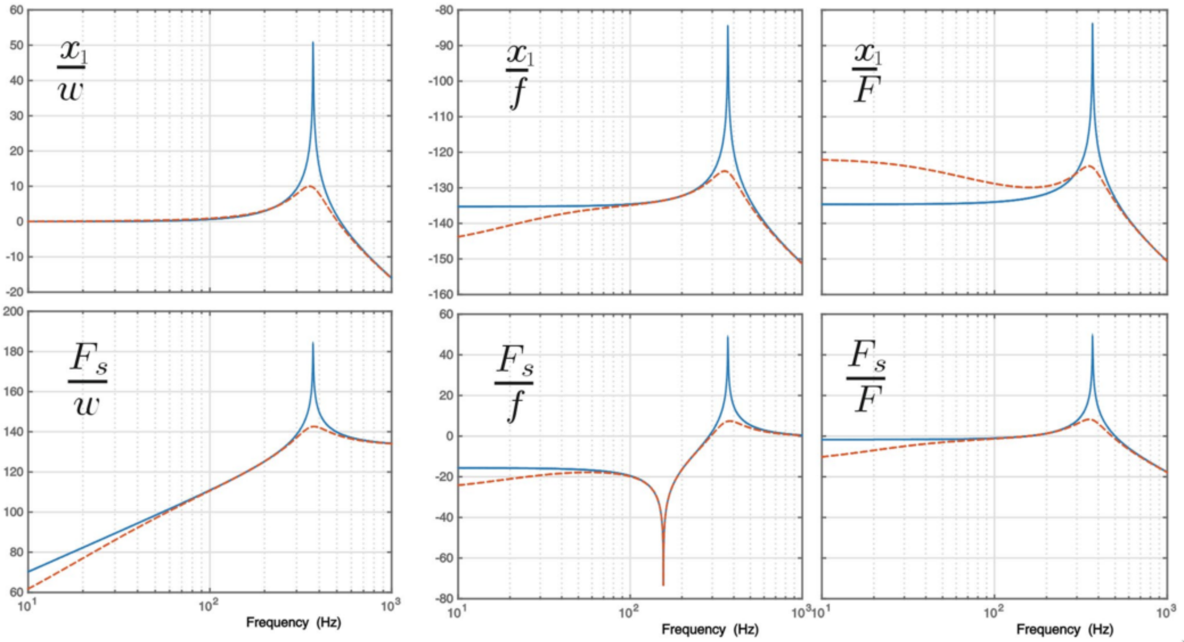

Figure 1.13: Results obtained in [2]

2 First Flexible Joint Geometry

The studied flexor is shown in Figure 2.1.

The stiffness and mass matrices representing the dynamics of the flexor are exported from a FEM. It is then imported into Simscape.

A simplified model of the flexor is then developped.

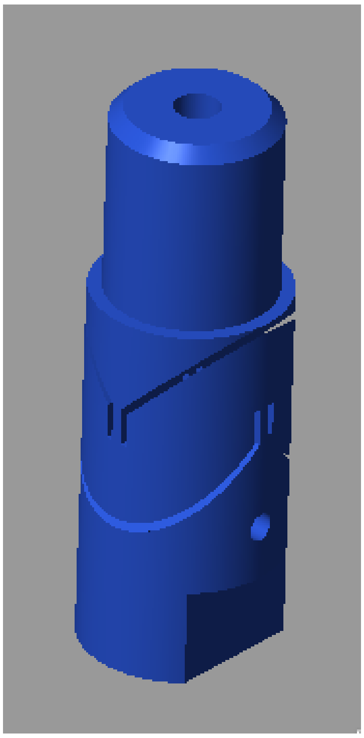

Figure 2.1: Flexor studied

2.1 Import Mass Matrix, Stiffness Matrix, and Interface Nodes Coordinates

We first extract the stiffness and mass matrices.

Matlab
```
K = extractMatrix('mat_K_6modes_2MDoF.matrix');
M = extractMatrix('mat_M_6modes_2MDoF.matrix');
```

Table 2.1: First 10x10 elements of the Stiffness matrix

11200000	195	2220	-0.719	-265	1.59	-11200000	-213	-2220	0.147
148	195	11400000	1290	-148	-0.188	2.41	-212	-11400000	-1290
2220	1290	119000000	1.31	1.49	1.79	-2220	-1290	-119000000	-1.31
-0.719	-148	1.31	33	000488	-000977	0.141	148	-1.31	-33
-265	-0.188	1.49	000488	33	00293	266	0.154	-1.49	00026
1.59	2.41	1.79	-000977	00293	236	-1.32	-2.55	-1.79	000379
-11200000	-212	-2220	0.141	266	-1.32	11400000	24600	1640	120
-213	-11400000	-1290	148	0.154	-2.55	24600	11400000	1290	-72
-2220	-1290	-119000000	-1.31	-1.49	-1.79	1640	1290	119000000	1.32
0.147	148	-1.31	-33	00026	000379	120	-72	1.32	34.7

Table 2.2: First 10x10 elements of the Mass matrix

0.02	1e-09	-4e-08	-1e-10	0.0002	-3e-11	0.004	5e-08	7e-08	1e-10
1e-09	0.02	-3e-07	-0.0002	-1e-10	-2e-09	2e-08	0.004	3e-07	1e-05
-4e-08	-3e-07	0.02	7e-10	-2e-09	1e-09	3e-07	7e-08	0.003	1e-09
-1e-10	-0.0002	7e-10	4e-06	-1e-12	-6e-13	2e-10	-7e-06	-8e-10	-1e-09
0.0002	-1e-10	-2e-09	-1e-12	3e-06	2e-13	9e-06	4e-11	2e-09	-3e-13
-3e-11	-2e-09	1e-09	-6e-13	2e-13	4e-07	8e-11	9e-10	-1e-09	2e-12
0.004	2e-08	3e-07	2e-10	9e-06	8e-11	0.02	-7e-08	-3e-07	-2e-10
5e-08	0.004	7e-08	-7e-06	4e-11	9e-10	-7e-08	0.01	-4e-08	0.0002
7e-08	3e-07	0.003	-8e-10	2e-09	-1e-09	-3e-07	-4e-08	0.02	-1e-09
1e-10	1e-05	1e-09	-1e-09	-3e-13	2e-12	-2e-10	0.0002	-1e-09	2e-06

Table 2.3: Some extracted parameters of the FEM

Total number of Nodes	2
Number of interface Nodes	2
Number of Modes	6
Size of M and K matrices	18

Then, we extract the coordinates of the interface nodes.

```matlab
[int_xyz, int_i, n_xyz, n_i, nodes] = extractNodes('out_nodes_3D.txt');
```

Table 2.4: Coordinates of the interface nodes

Node i	Node Number	x [m]	y [m]	z [m]
1.0	181278.0	0.0	0.0	0.0
2.0	181279.0	0.0	0.0	-0.0

Using `K`, `M` and `int_xyz`, we can use the `Reduced Order Flexible Solid` simscape block.

2.2 Identification of the parameters using Simscape and looking at the Stiffness Matrix

The flexor is now imported into Simscape and its parameters are estimated using an identification.

The dynamics is identified from the applied force/torque to the measured displacement/rotation of the flexor. And we find the same parameters as the one estimated from the Stiffness matrix.

Table 2.5: Comparison of identified and FEM stiffnesses

Characteristic	**Value**	**Identification**
Axial Stiffness Dz [N/um]	119	119
Bending Stiffness Rx [Nm/rad]	33	33
Bending Stiffness Ry [Nm/rad]	33	33
Torsion Stiffness Rz [Nm/rad]	236	236

2.3 Simpler Model

Let's now model the flexible joint with a "perfect" Bushing joint as shown in Figure 2.2.

The parameters of the Bushing joint (stiffnesses) are estimated from the Stiffness matrix that was computed from the FEM.

```matlab
Kx = K(1,1); % [N/m]
Ky = K(2,2); % [N/m]
Kz = K(3,3); % [N/m]
Krx = K(4,4); % [Nm/rad]
Kry = K(5,5); % [Nm/rad]
Krz =  K(6,6); % [Nm/rad]
```

The dynamics from the applied force/torque to the measured displacement/rotation of the flexor is identified again for this simpler model. The two obtained dynamics are compared in Figure

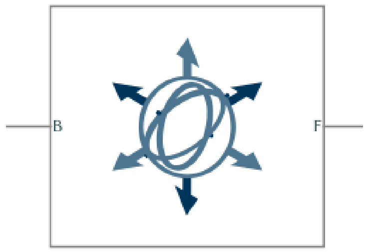

Figure 2.2: Bushing Joint used to model the flexible joint

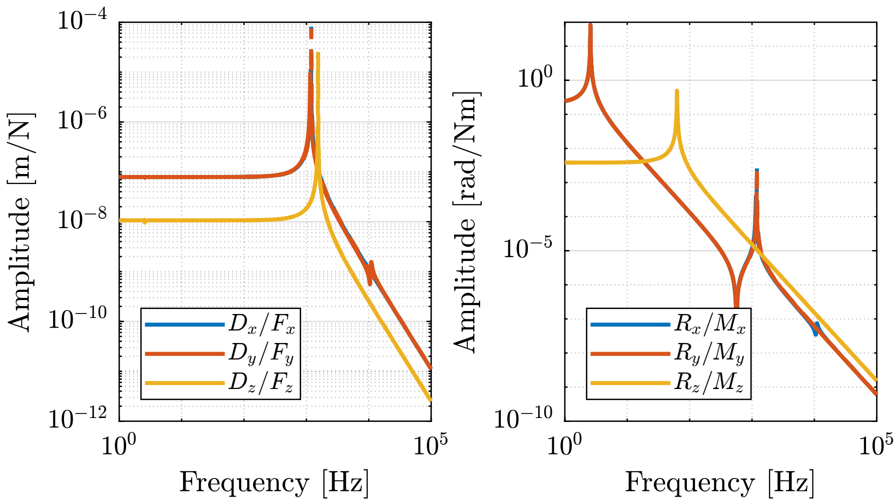

Figure 2.3: Comparison of the Joint compliance between the FEM model and the simpler model

3 Optimized Flexible Joint

The joint geometry has been optimized using Ansys to have lower bending stiffness while keeping a large axial stiffness.

The obtained geometry is shown in Figure 3.1.

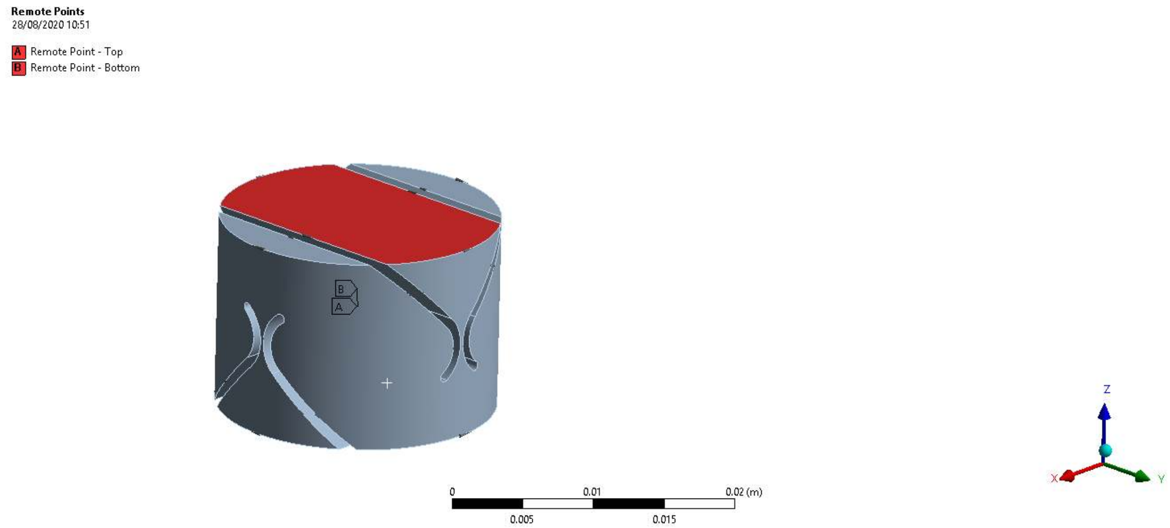

Figure 3.1: Flexor studied

3.1 Import Mass Matrix, Stiffness Matrix, and Interface Nodes Coordinates

We first extract the stiffness and mass matrices.

```matlab
K = readmatrix('flex025_mat_K.CSV');
M = readmatrix('flex025_mat_M.CSV');
```

Then, we extract the coordinates of the interface nodes.

```matlab
[int_xyz, int_i, n_xyz, n_i, nodes] = extractNodes('flex025_out_nodes_3D.txt');
```

Using `K`, `M` and `int_xyz`, we can use the `Reduced Order Flexible Solid` simscape block.

Table 3.1: First 10x10 elements of the Stiffness matrix

12700000	-18.5	-26.8	00162	-4.63	64	-12700000	18.3	26.7	00234
-18.5	12700000	-499	-132	00414	-0.495	18.4	-12700000	499	132
-26.8	-499	94000000	-470	00771	-0.855	26.8	498	-94000000	470
00162	-132	-470	4.83	2.61e-07	000123	-00163	132	470	-4.83
-4.63	00414	00771	2.61e-07	4.83	4.43e-05	4.63	-00413	-00772	-4.3e-07
64	-0.495	-0.855	000123	4.43e-05	260	-64	0.495	0.855	-000124
-12700000	18.4	26.8	-00163	4.63	-64	12700000	-18.2	-26.7	-00234
18.3	-12700000	498	132	-00413	0.495	-18.2	12700000	-498	-132
26.7	499	-94000000	470	-00772	0.855	-26.7	-498	94000000	-470
00234	132	470	-4.83	-4.3e-07	-000124	-00234	-132	-470	4.83

Table 3.2: First 10x10 elements of the Mass matrix

0.006	8e-09	-2e-08	-1e-10	3e-05	3e-08	0.003	-3e-09	9e-09	2e-12
8e-09	0.02	1e-07	-3e-05	1e-11	6e-10	1e-08	0.003	-5e-08	3e-09
-2e-08	1e-07	0.01	-6e-08	-6e-11	-8e-12	-1e-07	1e-08	0.003	-1e-08
-1e-10	-3e-05	-6e-08	1e-06	7e-14	6e-13	1e-10	1e-06	-1e-08	3e-10
3e-05	1e-11	-6e-11	7e-14	2e-07	1e-10	3e-08	-7e-12	6e-11	-6e-16
3e-08	6e-10	-8e-12	6e-13	1e-10	5e-07	1e-08	-5e-10	-1e-11	1e-13
0.003	1e-08	-1e-07	1e-10	3e-08	1e-08	0.02	-2e-08	1e-07	-4e-12
-3e-09	0.003	1e-08	1e-06	-7e-12	-5e-10	-2e-08	0.006	-8e-08	3e-05
9e-09	-5e-08	0.003	-1e-08	6e-11	-1e-11	1e-07	-8e-08	0.01	-6e-08
2e-12	3e-09	-1e-08	3e-10	-6e-16	1e-13	-4e-12	3e-05	-6e-08	2e-07

Table 3.3: Coordinates of the interface nodes for Flexible Joint

Total number of Nodes	2
Number of interface Nodes	2
Number of Modes	6
Size of M and K matrices	18

Table 3.4: Coordinates of the interface nodes

Node i	Node Number	x [m]	y [m]	z [m]
1.0	528875.0	0.0	0.0	0.0
2.0	528876.0	0.0	0.0	-0.0

3.2 Identification of the parameters using Simscape

The flexor is now imported into Simscape and its parameters are estimated using an identification.

The dynamics is identified from the applied force/torque to the measured displacement/rotation of the flexor. And we find the same parameters as the one estimated from the Stiffness matrix.

Table 3.5: Comparison of identified and FEM stiffnesses

Caracteristic	Value	Identification
Axial Stiffness Dz [N/um]	94.0	93.9
Bending Stiffness Rx [Nm/rad]	4.8	4.8
Bending Stiffness Ry [Nm/rad]	4.8	4.8
Torsion Stiffness Rz [Nm/rad]	260.2	260.2

3.3 Simpler Model

Let's now model the flexible joint with a "perfect" Bushing joint as shown in Figure 2.2.

The parameters of the Bushing joint (stiffnesses) are estimated from the Stiffness matrix that was computed from the FEM.

Matlab
```
Kx = K(1,1); % [N/m]
Ky = K(2,2); % [N/m]
Kz = K(3,3); % [N/m]
Krx = K(4,4); % [Nm/rad]
Kry = K(5,5); % [Nm/rad]
Krz =  K(6,6); % [Nm/rad]
```

The dynamics from the applied force/torque to the measured displacement/rotation of the flexor is identified again for this simpler model. The two obtained dynamics are compared in Figure

3.4 Comparison with a stiffer Flexible Joint

The stiffness matrix with the flexible joint with a "hinge" size of 0.50mm is loaded.

Matlab
```
K_050 = readmatrix('flex050_mat_K.CSV');
```

Its parameters are compared with the Flexible Joint with a size of 0.25mm in the table below.

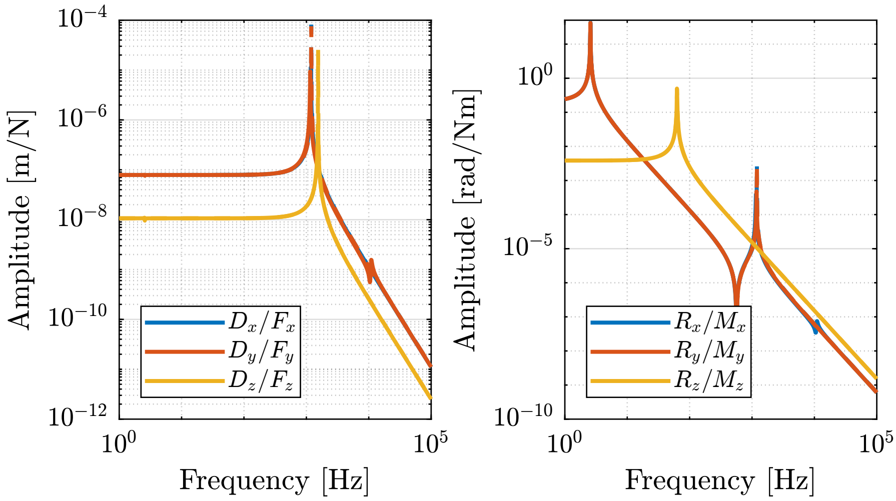

Figure 3.2: Comparison of the Joint compliance between the FEM model and the simpler model

Table 3.6: Comparison of flexible joint stiffnesses for 0.25 and 0.5mm

Caracteristic	**0.25 mm**	**0.50 mm**
Axial Stiffness Dz [N/um]	94.0	124.7
Shear Stiffness [N/um]	12.7	25.8
Bending Stiffness Rx [Nm/rad]	4.8	26.0
Bending Stiffness Ry [Nm/rad]	4.8	26.0
Torsion Stiffness Rz [Nm/rad]	260.2	538.0

4 Complete Strut with Encoder

4.1 Introduction

Now, the full nano-hexapod strut is modelled using Ansys.

The 3D as well as the interface nodes are shown in Figure 4.1.

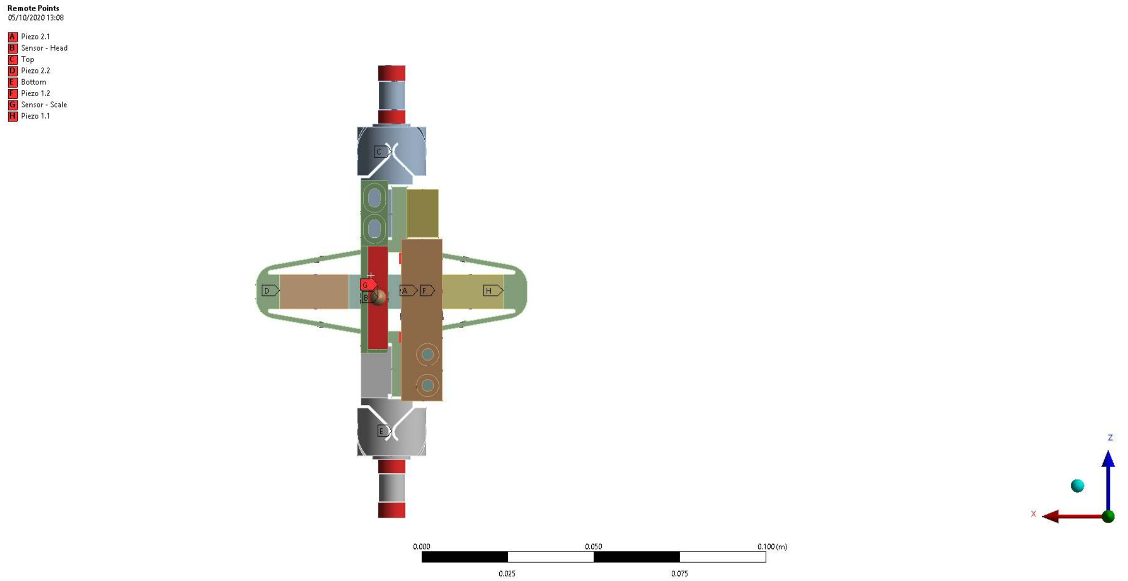

Figure 4.1: Interface points

A side view is shown in Figure 4.2.

The flexible joints used have a 0.25mm width size.

4.2 Import Mass Matrix, Stiffness Matrix, and Interface Nodes Coordinates

We first extract the stiffness and mass matrices.

```matlab
K = readmatrix('strut_encoder_mat_K.CSV');
M = readmatrix('strut_encoder_mat_M.CSV');
```

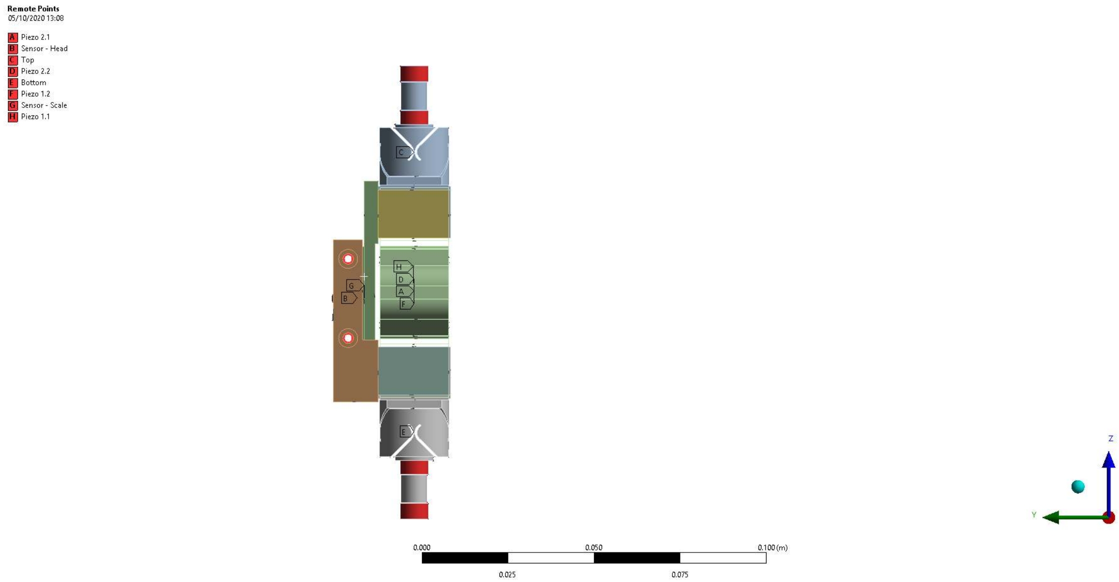

Figure 4.2: Interface points - Side view

Table 4.1: First 10x10 elements of the Stiffness matrix

2000000	1000000	-3000000	-400	300	200	-30	2000	-10000	0.3
1000000	4000000	-8000000	-900	400	-50	-6000	10000	-20000	3
-3000000	-8000000	20000000	2000	-900	200	-10000	20000	-300000	7
-400	-900	2000	5	-0.1	05	1	-3	6	-0007
300	400	-900	-0.1	5	04	-0.1	0.5	-3	0001
200	-50	200	05	04	300	4	-01	-1	3e-05
-30	-6000	-10000	1	-0.1	4	3000000	-1000000	-2000000	-300
2000	10000	20000	-3	0.5	-01	-1000000	6000000	7000000	1000
-10000	-20000	-300000	6	-3	-1	-2000000	7000000	20000000	2000
0.3	3	7	-0007	0001	3e-05	-300	1000	2000	5

Table 4.2: First 10x10 elements of the Mass matrix

0.04	-0.005	0.007	2e-06	0.0001	-5e-07	-1e-05	-9e-07	8e-05	-5e-10
-0.005	0.03	0.02	-0.0001	1e-06	-3e-07	3e-05	-0.0001	8e-05	-3e-08
0.007	0.02	0.08	-6e-06	-5e-06	-7e-07	4e-05	-0.0001	0.0005	-3e-08
2e-06	-0.0001	-6e-06	2e-06	-4e-10	2e-11	-8e-09	3e-08	-2e-08	6e-12
0.0001	1e-06	-5e-06	-4e-10	3e-06	2e-10	-3e-09	3e-09	-7e-09	6e-13
-5e-07	-3e-07	-7e-07	2e-11	2e-10	5e-07	-2e-08	5e-09	-5e-09	1e-12
-1e-05	3e-05	4e-05	-8e-09	-3e-09	-2e-08	0.04	0.004	0.003	1e-06
-9e-07	-0.0001	-0.0001	3e-08	3e-09	5e-09	0.004	0.02	-0.02	0.0001
8e-05	8e-05	0.0005	-2e-08	-7e-09	-5e-09	0.003	-0.02	0.08	-5e-06
-5e-10	-3e-08	-3e-08	6e-12	6e-13	1e-12	1e-06	0.0001	-5e-06	2e-06

Then, we extract the coordinates of the interface nodes.

```matlab
[int_xyz, int_i, n_xyz, n_i, nodes] = extractNodes('strut_encoder_out_nodes_3D.txt');
```

Table 4.3: Some extracted parameters of the FEM

Total number of Nodes	8
Number of interface Nodes	8
Number of Modes	6
Size of M and K matrices	54

Table 4.4: Coordinates of the interface nodes

Node i	Node Number	x [m]	y [m]	z [m]
1.0	504411.0	0.0	0.0	0.0405
2.0	504412.0	0.0	0.0	-0.0405
3.0	504413.0	-0.0325	0.0	0.0
4.0	504414.0	-0.0125	0.0	0.0
5.0	504415.0	-0.0075	0.0	0.0
6.0	504416.0	0.0325	0.0	0.0
7.0	504417.0	0.004	0.0145	-0.00175
8.0	504418.0	0.004	0.0166	-0.00175

Using `K`, `M` and `int_xyz`, we can use the `Reduced Order Flexible Solid` simscape block.

4.3 Piezoelectric parameters

Parameters for the APA300ML:

```matlab
d33 = 300e-12; % Strain constant [m/V]
n   = 80;      % Number of layers per stack
eT  = 1.6e-8;  % Permittivity under constant stress [F/m]
sD  = 1e-11;   % Compliance under constant electric displacement [m2/N]
ka  = 235e6;   % Stack stiffness [N/m]
C   = 5e-6;    % Stack capactiance [F]
```

```matlab
na = 2; % Number of stacks used as actuator
ns = 1; % Number of stacks used as force sensor
```

4.4 Identification of the Dynamics

The dynamics is identified from the applied force to the measured relative displacement. The same dynamics is identified for a payload mass of 10Kg.

```
m = 10;
```

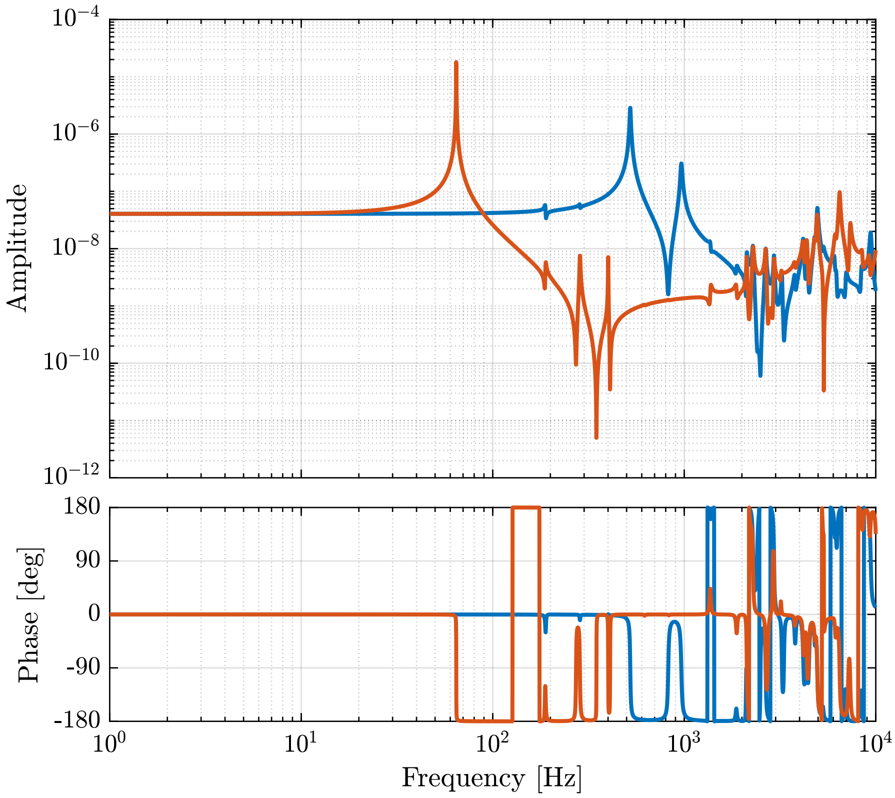

Figure 4.3: Dynamics from the force actuator to the measured motion by the encoder

Bibliography

[1] A. J. Fleming and K. K. Leang, *Design, Modeling and Control of Nanopositioning Systems* (Advances in Industrial Control). Springer International Publishing, 2014, nil (cit. on p. 6).

[2] A. Souleille, T. Lampert, V. Lafarga, *et al.*, "A concept of active mount for space applications," *CEAS Space Journal*, vol. 10, no. 2, pp. 157–165, 2018 (cit. on pp. 12, 16, 19).