Decoupling strategies for MIMO system

Done by:
Haidar Lakkis
Thomas Dehaeze

mhlakkis@uliege.be

Referencing

Most of the work presented in this seminar was done with the help of documentations and models done by Thomas Dehaeze.

Here is a link to this documentation:

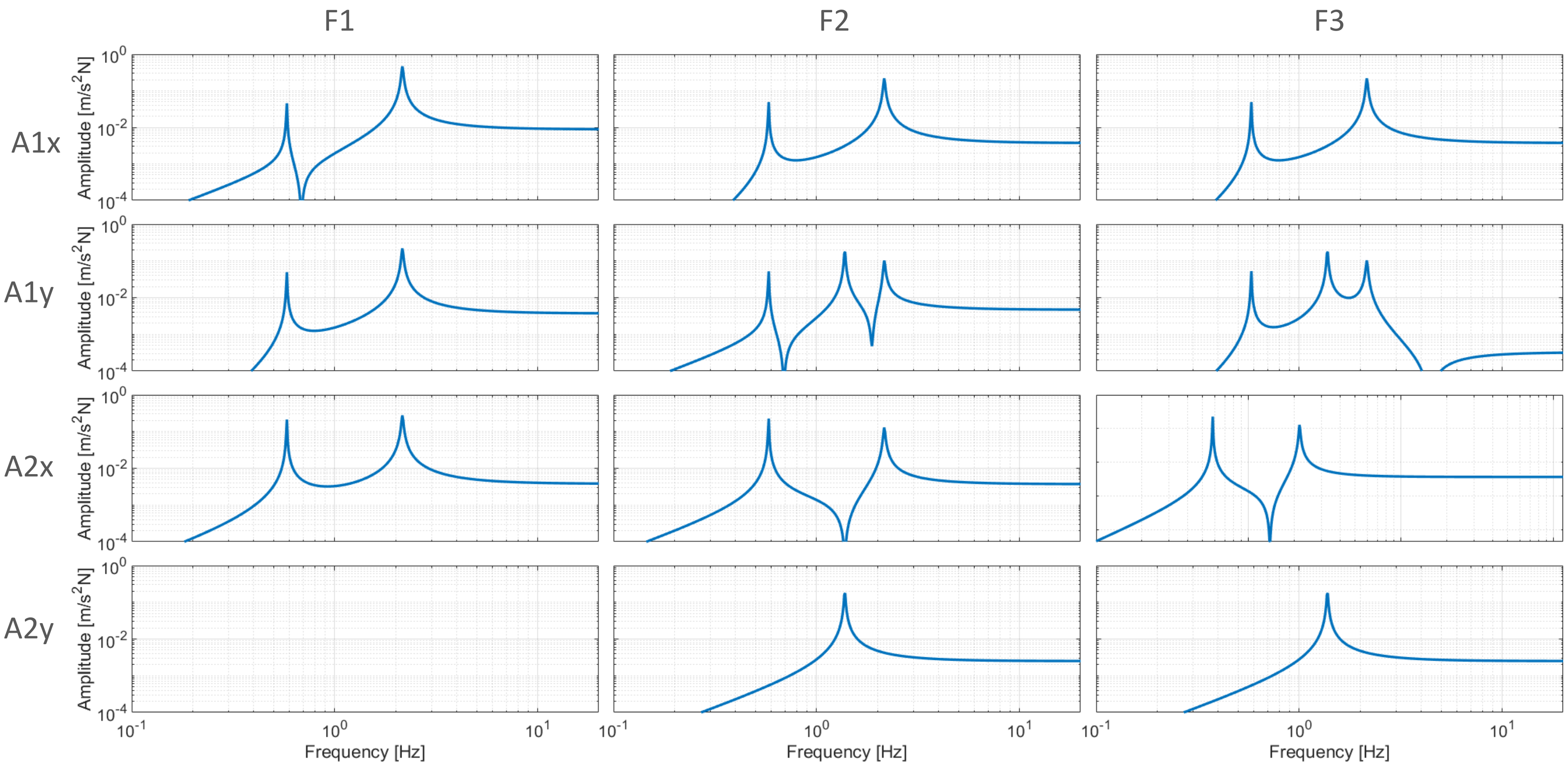
- For Documentation: https://research.tdehaeze.xyz/svd-control/
- Matlab codes and simscape model: https://git.tdehaeze.xyz/tdehaeze/svd-control

Why decouple ?

Why decouple ?

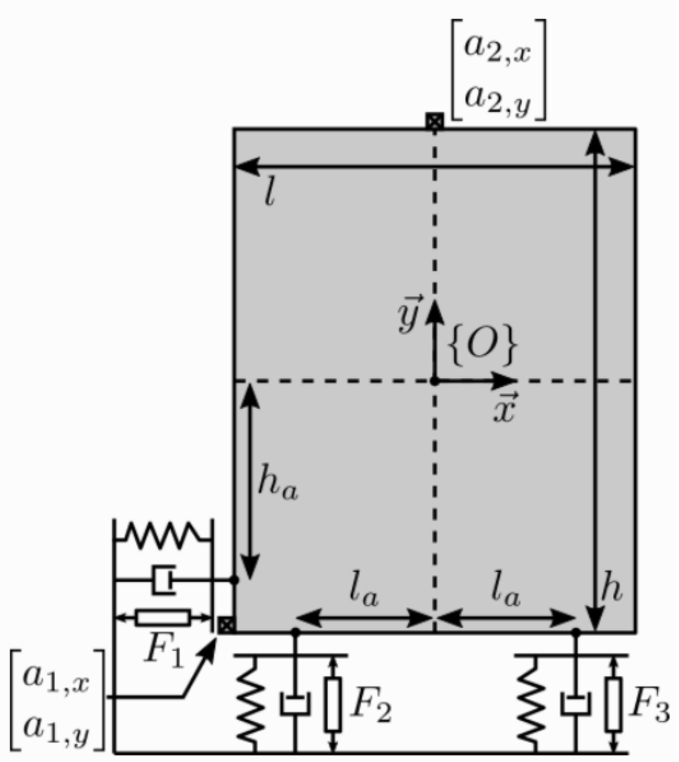
Open Loop transfer matrix of a coupled system:

Open Loop Transfer Functions from different Actuators to different sensors



Considered Plant:

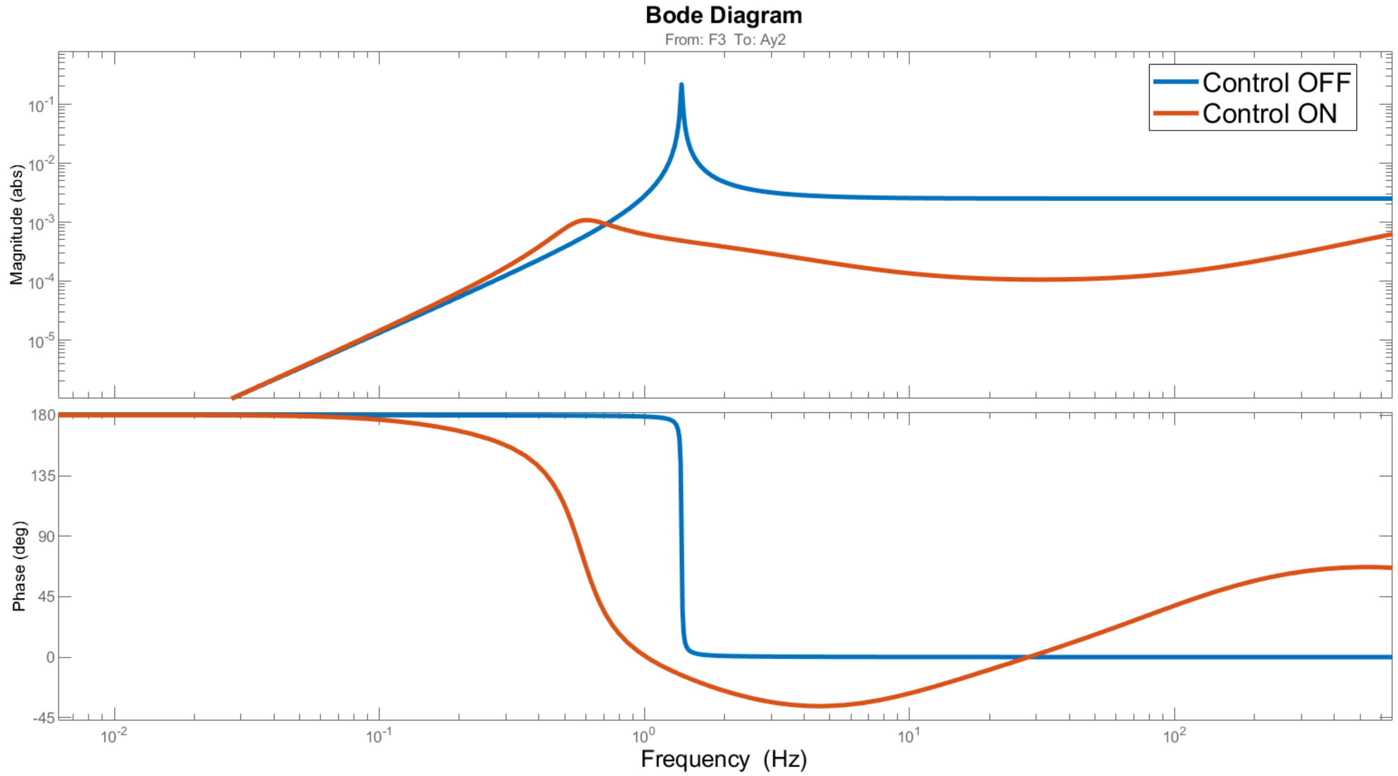
- 3 Actuators
- 4 sensors(accelerometers)



Model of Gravimeter

Why decouple ?

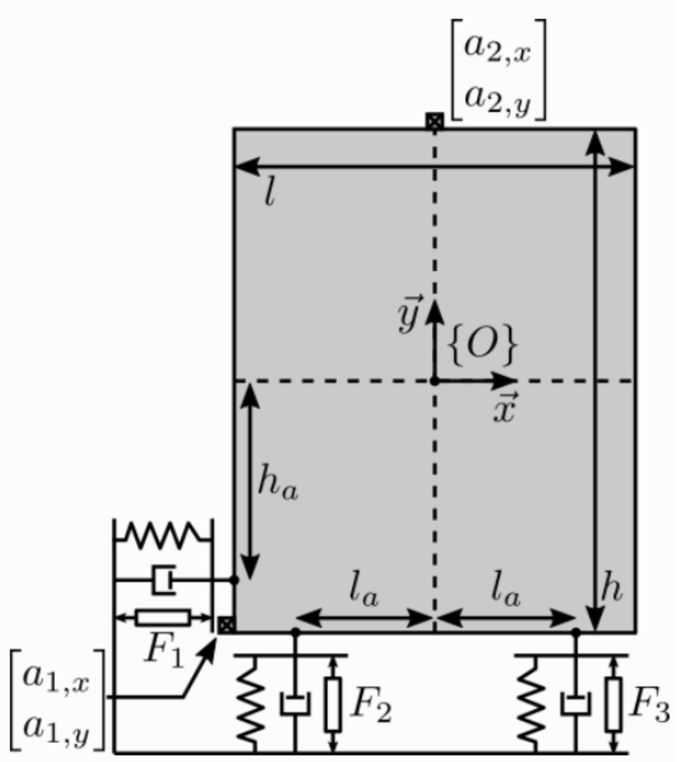
Transfer Function from Actuator F3 to sensor A2y



Treat the transfer function as a SISO TF and actively controlling it, good controller with good stability margins

Considered Plant:

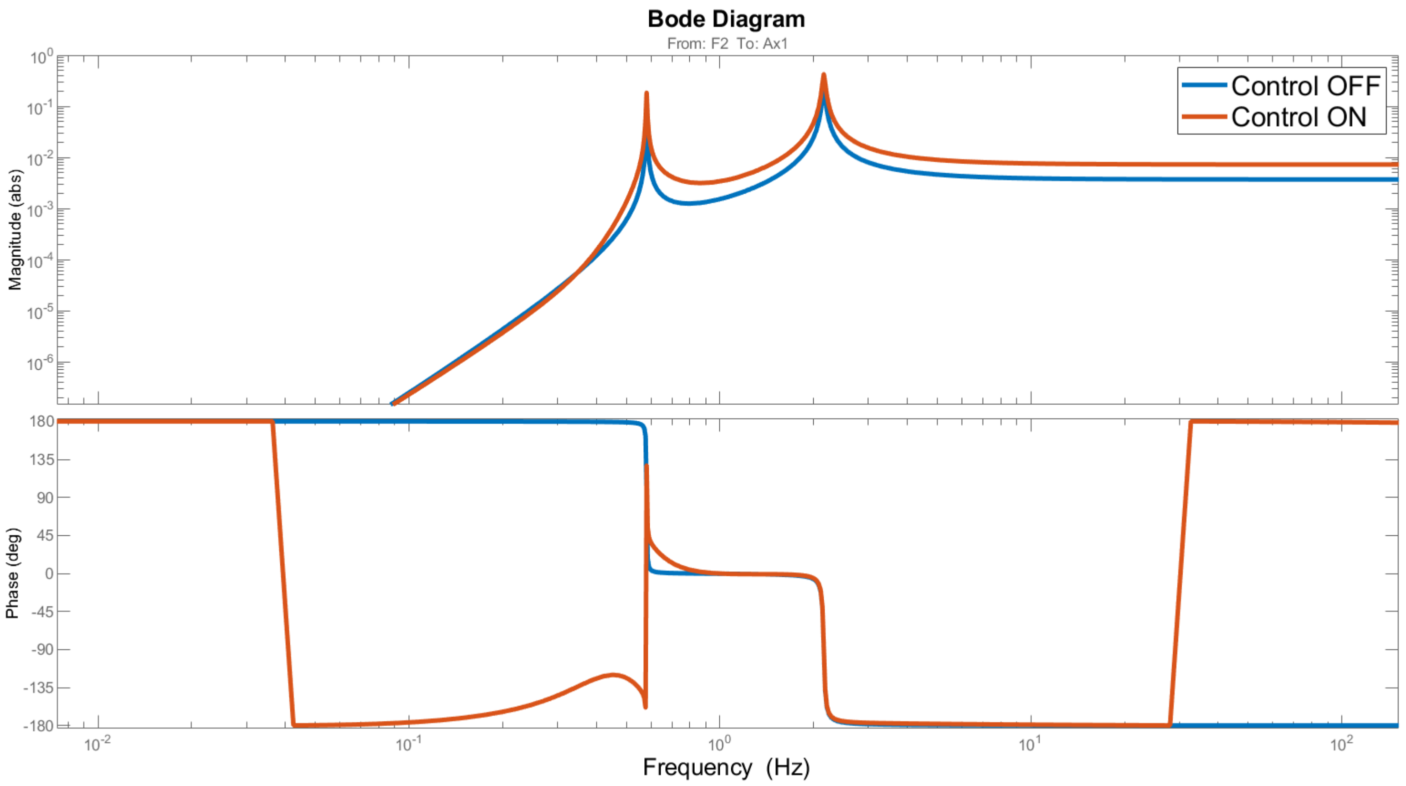
- 3 Actuators
- 4 sensors(accelerometers)



Model of Gravimeter

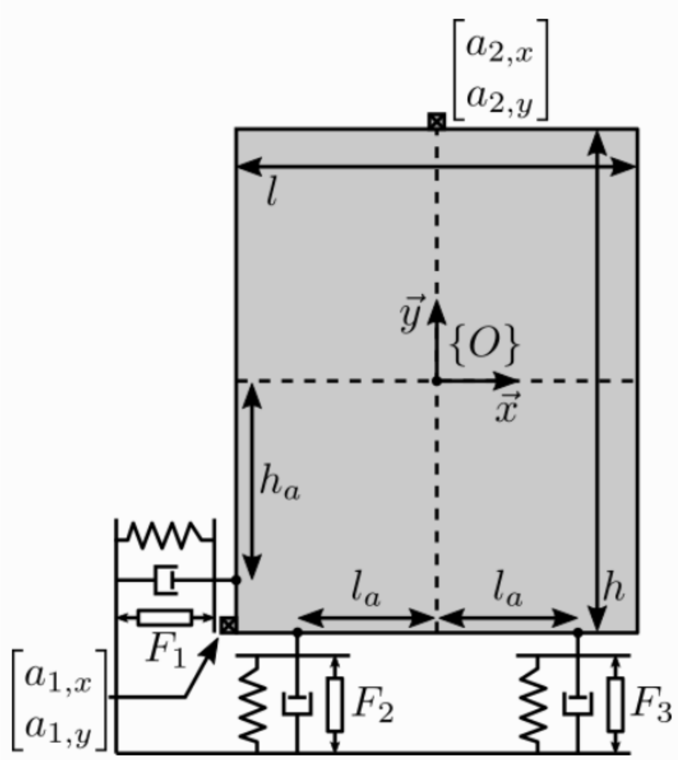
Why decouple ?

Transfer Function from Actuator F2 to sensor A1x



Considered Plant:

- 3 Actuators
- 4 sensors(accelerometers)

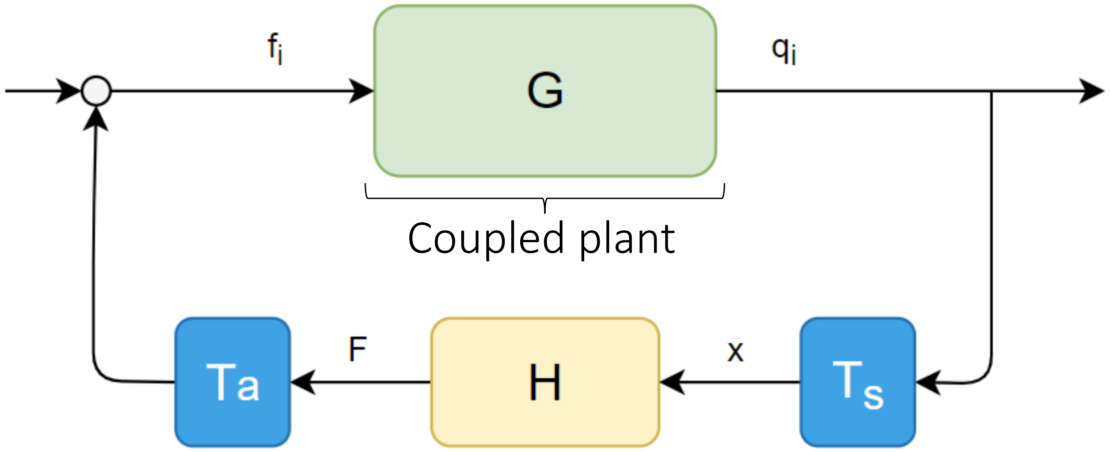


Model of Gravimeter

Other TF in the MIMO system is degraded after closing the controller loop → Coupling → SISO approaches not easily applicable in coupled systems → Risk of Instability !!

Why decouple ?

Centralized Control:



Ts: sensor transformation matrix

Ta: actuator transformation matrix

$$F = \begin{pmatrix} F_1 \\ \vdots \\ F_n \end{pmatrix} \qquad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

Actuation Vector in decoupled frame

Output Vector in decoupled frame

- SISO control approaches could be applied.
- Easy to control independently the degrees of freedom.
- Certain knowledge of the plant is required (kind of knowledge depends on decoupling strategy).

$F = H * x$ Where:

$$H_i = \begin{pmatrix} H_1 & & & & & \\ & \ddots & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & \ddots & \\ & & & & & H_n \end{pmatrix}$$

Control matrix

Table of contents

Singular Value Decomposition

II. Singular Value Decomposition

$$A = U\Sigma V^T$$

Where :
- A is an mxn rectangular matrix
- U is an mxm orthogonal matrix, $U^T U = I$.
- V is an nxn orthogonal matrix , $V^T V = I$
- Σ is an mxn pseudo-diagonal matrix where first r elements on the diagonal are the singular values of A, which we denote as $\Sigma_{ii} = \sigma_i$ of $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, and all other elements of Σ equal to zero

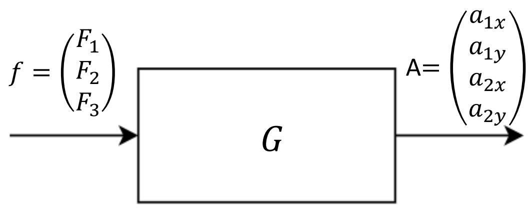
$$\underbrace{\begin{pmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} U_{11} & \cdots & U_{1m} \\ \vdots & \ddots & \vdots \\ U_{m1} & \cdots & U_{mm} \end{pmatrix}}_{U} \underbrace{\left(\begin{array}{ccc|ccc} \sigma_1 & \cdots & 0 & 0 & \cdots & 0_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_r & 0 & \cdots & 0 \\ \hline 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0_{m1} & \cdots & 0 & 0 & \cdots & 0_{mn} \end{array}\right)}_{\Sigma} \underbrace{\begin{pmatrix} V_{11} & \cdots & V_{1n} \\ \vdots & \ddots & \vdots \\ V_{n1} & \cdots & V_{nn} \end{pmatrix}^T}_{V^T}$$

Coupled Plant | Transform matrix | Decoupled Plant | Transform matrix

Interesting to check: https://www.youtube.com/watch?v=nbBvuuNVfco

II. Singular Value Decomposition

According to the SVD formula, the decoupled plant should be : $\underbrace{G_{svd}}_{\Sigma} = U^{-1}\underbrace{G(s)}_{A}V^{-T}$



Decentralized Scheme

However, since G is frequency dependent, U and V can’t decouple the system over all frequency bandwidth.

⇩

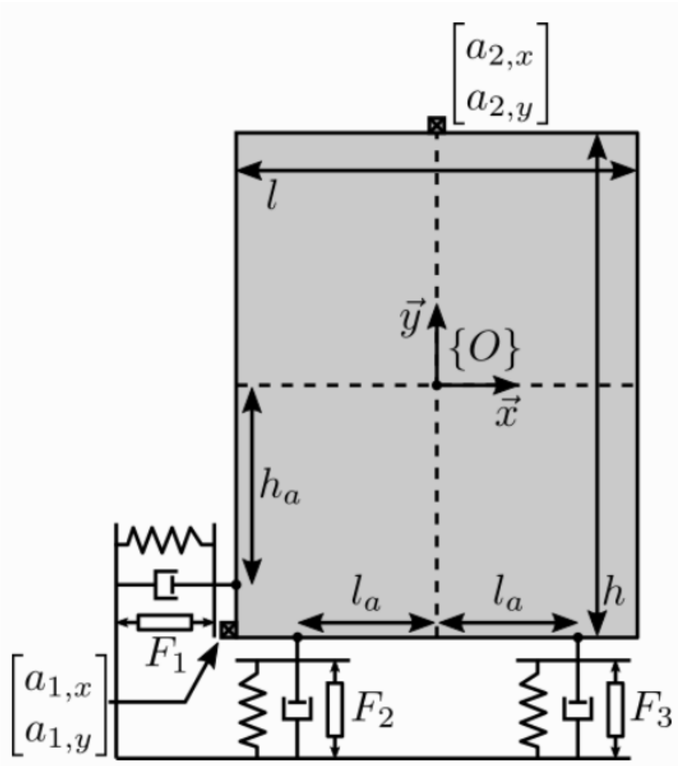
Choose a frequency for decoupling (preferably crossover frequency)

⇩

Get real approximation of the transfer matrix G at this frequency

⇩

Get U and V matrices that decouple plant G at the chosen frequency neighborhood



Model of Gravimeter

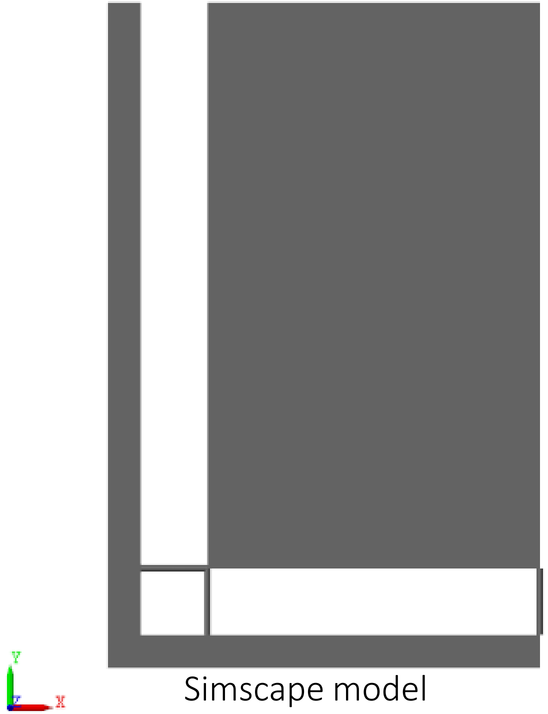
Plant Modelling using Matlab and Simscape:

Parameter definition :

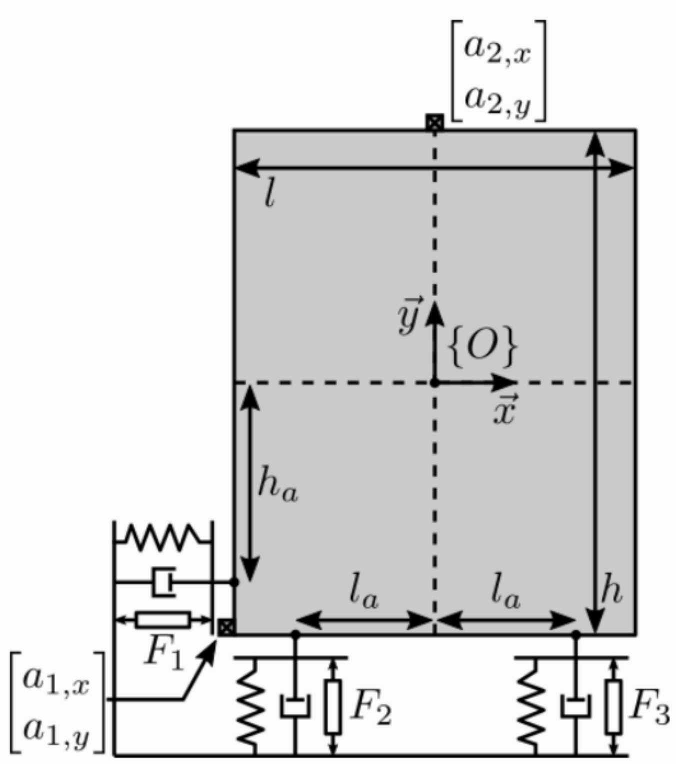
```matlab
l  = 1.0; % Length of the mass [m]
h  = 1.7; % Height of the mass [m]

la = l/2; % Position of Act. [m]
ha = h/2; % Position of Act. [m]

m = 400; % Mass [kg]
I = 115; % Inertia [kg m^2]

k = 15e3; % Actuator Stiffness [N/m]
c = 2e1; % Actuator Damping [N/(m/s)]

deq = 0.2; % Length of the actuators [m]

g = 0; % Gravity [m/s2]
```

System identification :

```matlab
%% Name of the Simulink File
mdl = 'gravimeter';

%% Input/Output definition
clear io; io_i = 1;
io(io_i) = linio([mdl, '/F1'], 1, 'openinput');  io_i = io_i + 1;
io(io_i) = linio([mdl, '/F2'], 1, 'openinput');  io_i = io_i + 1;
io(io_i) = linio([mdl, '/F3'], 1, 'openinput');  io_i = io_i + 1;
io(io_i) = linio([mdl, '/Acc_side'], 1, 'openoutput'); io_i = io_i + 1;
io(io_i) = linio([mdl, '/Acc_side'], 2, 'openoutput'); io_i = io_i + 1;
io(io_i) = linio([mdl, '/Acc_top'], 1, 'openoutput'); io_i = io_i + 1;
io(io_i) = linio([mdl, '/Acc_top'], 2, 'openoutput'); io_i = io_i + 1;

G = linearize(mdl, io);
G.InputName  = {'F1', 'F2', 'F3'};
G.OutputName = {'Ax1', 'Ay1', 'Ax2', 'Ay2'};
```



Simscape model



Model of Gravimeter

II. Singular Value Decomposition

SVD decoupling using Matlab :

Evaluating transfer matrix values at frequency of 10Hz:

Matlab
```
wc = 2*pi*10; % Decoupling frequency [rad/s]

H1 = evalfr(G, j*wc);
```

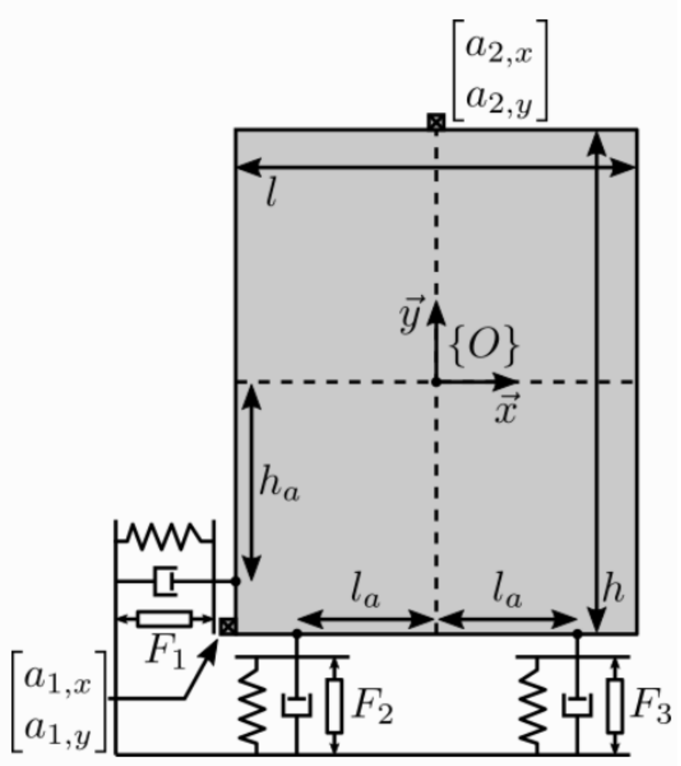
Real approximation of the computed transfer matrix at 10Hz:

Matlab
```
D = pinv(real(H1'*H1));
H1 = pinv(D*real(H1'*diag(exp(j*angle(diag(H1*D*H1.'))/2))));
```

SVD decomposition performed using the following matlab command:

Matlab
```
[U,S,V] = svd(H1);
```

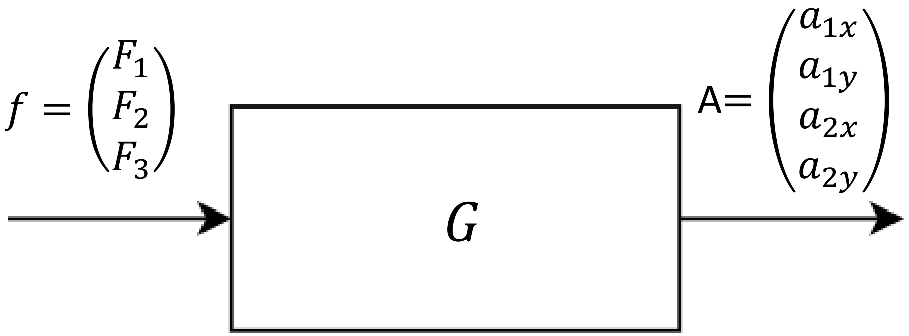
$$U = \begin{pmatrix} -0{,}78 & 0{,}26 & -0{,}53 & 0{,}2 \\ 0{,}4 & 0{,}61 & -0{,}04 & -0{,}68 \\ 0{,}48 & -0{,}14 & -0{,}85 & 0{,}2 \\ 0{,}03 & 0{,}73 & 0{,}06 & 0{,}68 \end{pmatrix} \qquad V = \begin{pmatrix} -0{,}79 & 0{,}11 & -0{,}6 \\ 0{,}51 & 0{,}67 & -0{,}54 \\ -0{,}35 & 0{,}73 & 0{,}59 \end{pmatrix}$$



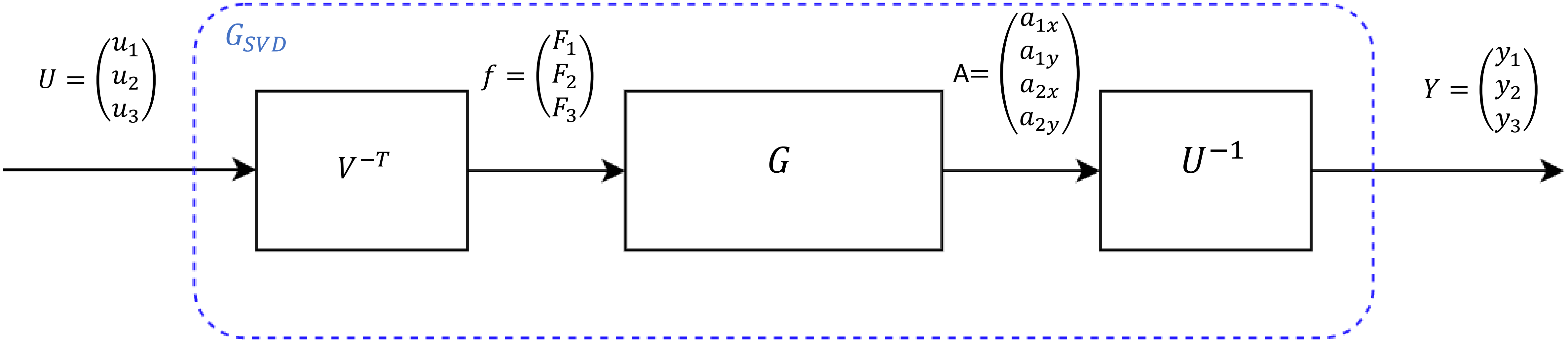
Model of Gravimeter

II. Singular Value Decomposition

SVD centralized vs decentralized control schemes:



Decentralized Scheme



Centralized Scheme

II. Singular Value Decomposition

SVD decoupling using Matlab:

Decoupled Plant can be obtained by:

Matlab
```
Gsvd = inv(U)*G*inv(V');
```

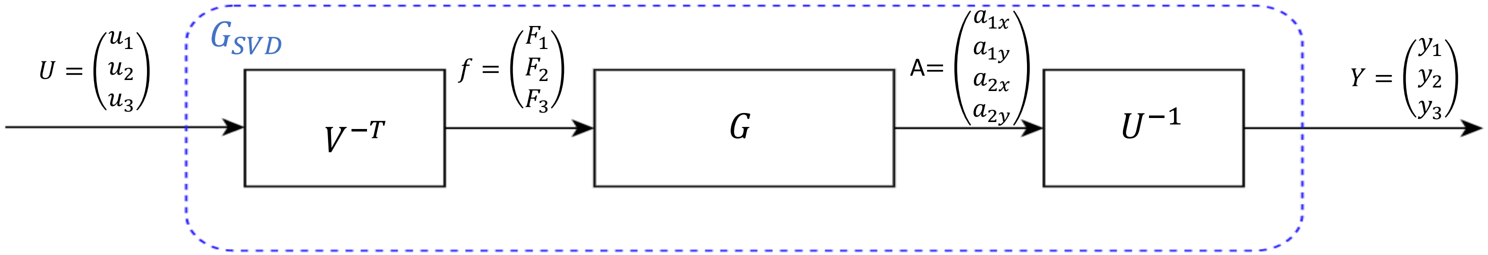
Discarding 4th output (correspond to null singular value)

Matlab
```
Gsvd = Gsvd(1:3, 1:3);
```

Decoupling could be bad in frequencies not falling in the neighborhood of the decoupling frequency

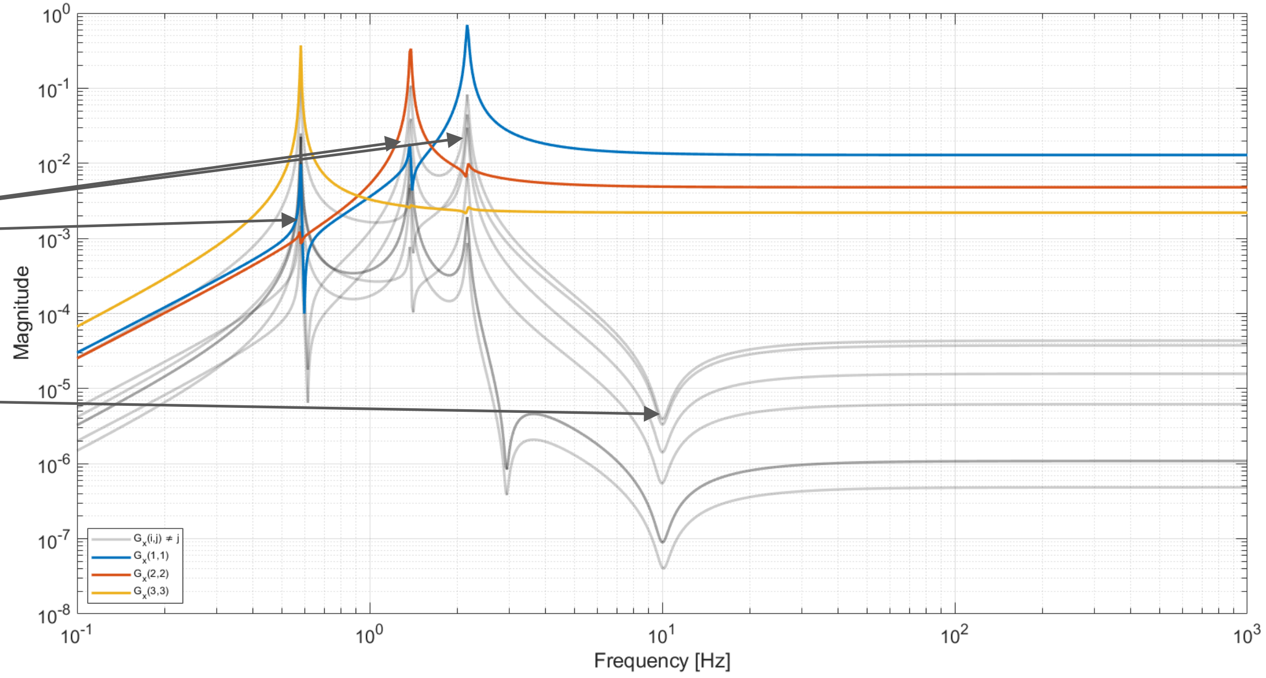
Best Decoupling Performance obtained at the frequency where the SVD decomposition was applied

Very effective decoupling on a specific frequency, but not necessarily over all the bandwidth



Centralized Scheme

Open Loop Transfer Functions of the decoupled plant using SVD



Jacobian decoupling

III. Jacobian decoupling

Analytical calculation of the Jacobians:

In cartesian coordinates:

$$M\ddot{x} + Kx = F$$

Since at COM:

$$M = diag(m, m, I_\theta)$$

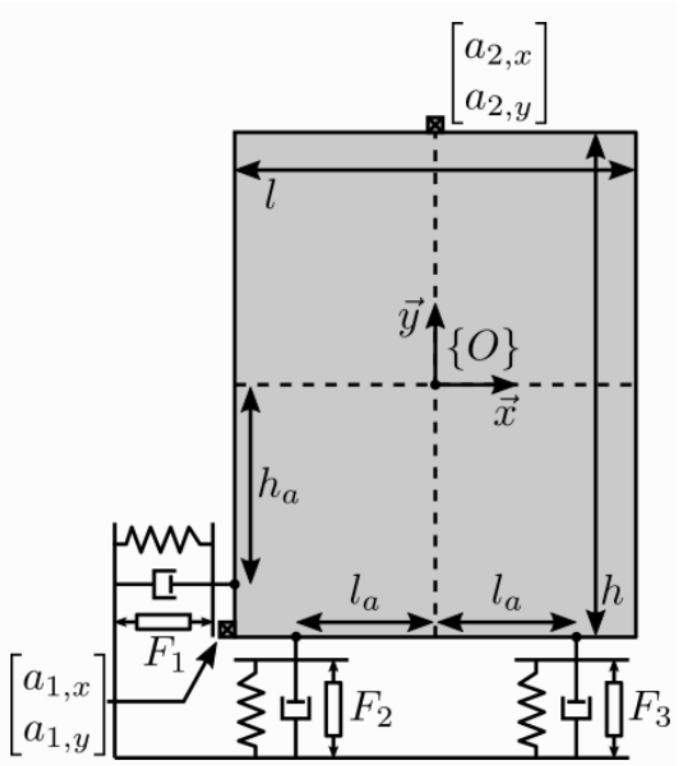
Consider J as Jacobian matrix from cartesian coordinates to the coordinates of the actuators/sensors:

$$q = Jx$$

$$F = Bf \qquad \text{Where: } f = \begin{pmatrix} F1 \\ F2 \\ F3 \end{pmatrix} \text{ and } F = \begin{pmatrix} F_x \\ F_y \\ M_\theta \end{pmatrix}$$

According to principle of virtual work:

$$F^T \delta x = f^T \delta q = f^T J \delta x \quad \Rightarrow \quad F = J^T f \quad \Rightarrow \quad B = J^T$$



Model of Gravimeter

III. Jacobian decoupling

Analytical calculation of the Jacobians:

Actuator Jacobian:

From rigid body dynamics:

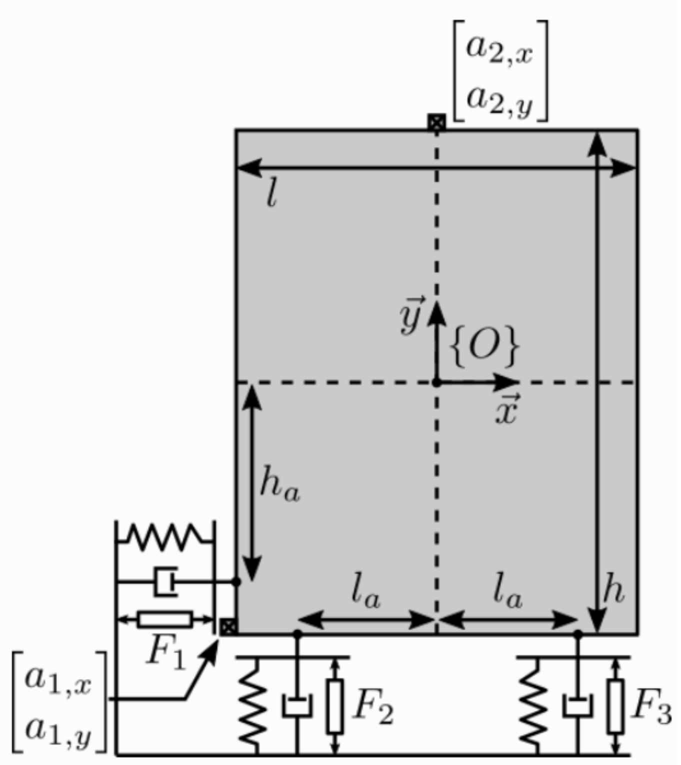
$$x_e = x_{CM} + R(\theta)\, x_{e0}$$

Relative displacement:

$$q = x_e - x_{e0}$$

Where :

- x_e : position of an element in the Cartesian frame in the deformed configuration.
- x_{CM} : is the position of the center of mass in the Cartesian frame.
- x_{eo} : is the position of the considered element in the reference configuration.



Model of Gravimeter

Relative displacements at actuator locations can be calculated as follows:

$$R(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

$$q_{F1} = (1\ 0)\left[\begin{pmatrix} x \\ y \end{pmatrix} + R(\theta)\begin{pmatrix} -\frac{l}{2} \\ -h_a \end{pmatrix} - \begin{pmatrix} -\frac{l}{2} \\ -h_a \end{pmatrix}\right] = \mathrm{x} + (1-\cos(\theta))\frac{l}{2} - \sin(\theta)h_a$$

$$q_{F2} = (0\ 1)\left[\begin{pmatrix} x \\ y \end{pmatrix} + R(\theta)\begin{pmatrix} -l_a \\ -\frac{h}{2} \end{pmatrix} - \begin{pmatrix} -l_a \\ -\frac{h}{2} \end{pmatrix}\right] = \mathrm{y} + l_a(\sin(\theta)) + \frac{h}{2}(1-\cos(\theta))$$

III. Jacobian decoupling

Analytical calculation of the Jacobians:

Actuator Jacobian:

$$q_{F3} = (0\ 1)\left[\begin{pmatrix} x \\ y \end{pmatrix} + R(\theta)\begin{pmatrix} l_a \\ -\frac{h}{2} \end{pmatrix} - \begin{pmatrix} l_a \\ -\frac{h}{2} \end{pmatrix}\right] = \mathrm{y} - l_a(\sin(\theta)) + \frac{h}{2}(1 - \cos(\theta))$$

$$J_{act} = \begin{bmatrix} \frac{\partial q_{F1}}{\partial x} & \frac{\partial q_{F1}}{\partial y} & \frac{\partial q_{F1}}{\partial \theta} \\ \frac{\partial q_{F2}}{\partial x} & \frac{\partial q_{F2}}{\partial y} & \frac{\partial q_{F2}}{\partial \theta} \\ \frac{\partial q_{F3}}{\partial x} & \frac{\partial q_{F3}}{\partial y} & \frac{\partial q_{F3}}{\partial \theta} \end{bmatrix}_{(x,y,\theta)=(0,0,0)} = \begin{bmatrix} 1 & 0 & -h_a \\ 0 & 1 & l_a \\ 0 & 1 & -l_a \end{bmatrix}$$

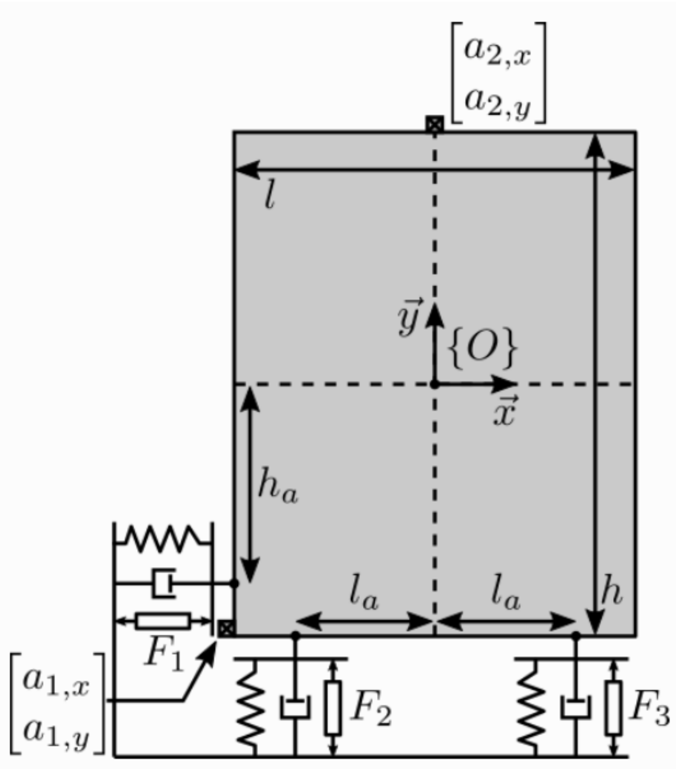
Equation used to move from actuation in coupled frame to actuation in the cartesian frame centered at COM:

$$F = J_{act}^T * f \quad \Rightarrow \quad f = (J_{act})^{-T} * F$$

Where :

$F = \begin{pmatrix} F_x \\ F_y \\ M_\theta \end{pmatrix}$ Forces in Cartesian coordinates

$f = \begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix}$ Forces in actuators coordinates



Model of Gravimeter

III. Jacobian decoupling

Analytical calculation of the Jacobians:

Sensor Jacobian:

Relative displacements at sensor locations can be calculated as follows:

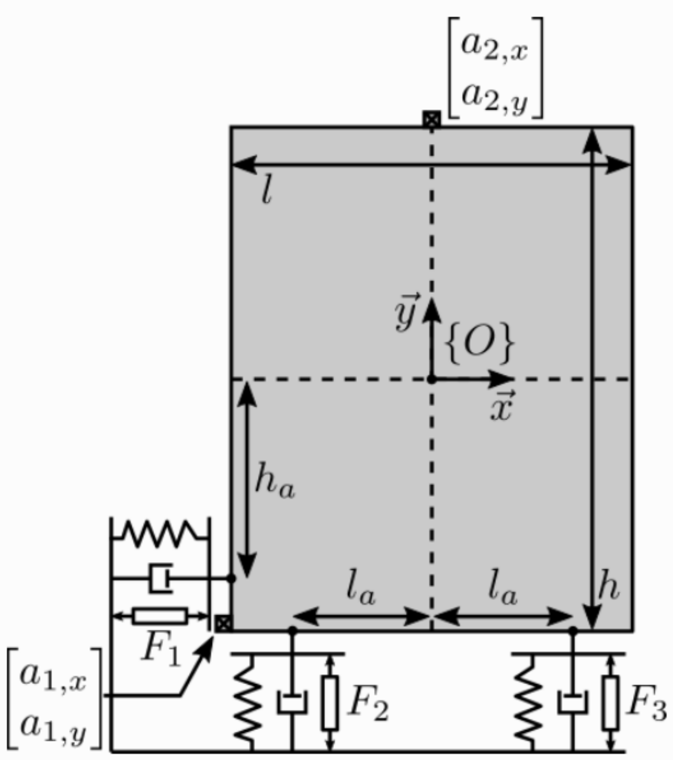
$$R(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

$$q_{a1x} = (1\ 0)\left[\begin{pmatrix} x \\ y \end{pmatrix} + R(\theta)\begin{pmatrix} -\frac{l}{2} \\ -\frac{h}{2} \end{pmatrix} - \begin{pmatrix} -\frac{l}{2} \\ -\frac{h}{2} \end{pmatrix}\right] = \mathrm{x} - \frac{h}{2}(\sin(\theta)) + \frac{l}{2}(1 - \cos(\theta))$$

$$q_{a1y} = (0\ 1)\left[\begin{pmatrix} x \\ y \end{pmatrix} + R(\theta)\begin{pmatrix} -\frac{l}{2} \\ -\frac{h}{2} \end{pmatrix} - \begin{pmatrix} -\frac{l}{2} \\ -\frac{h}{2} \end{pmatrix}\right] = \mathrm{y} + \frac{l}{2}(\sin(\theta)) + \frac{h}{2}(1 - \cos(\theta))$$

$$q_{a2x} = (1\ 0)\left[\begin{pmatrix} x \\ y \end{pmatrix} + R(\theta)\begin{pmatrix} 0 \\ \frac{h}{2} \end{pmatrix} - \begin{pmatrix} 0 \\ \frac{h}{2} \end{pmatrix}\right] = \mathrm{x} + \frac{h}{2} sin(\theta)$$

$$q_{a2y} = (0\ 1)\left[\begin{pmatrix} x \\ y \end{pmatrix} + R(\theta)\begin{pmatrix} 0 \\ \frac{h}{2} \end{pmatrix} - \begin{pmatrix} 0 \\ \frac{h}{2} \end{pmatrix}\right] = \mathrm{y} + \frac{h}{2}(-1 + \cos(\theta))$$



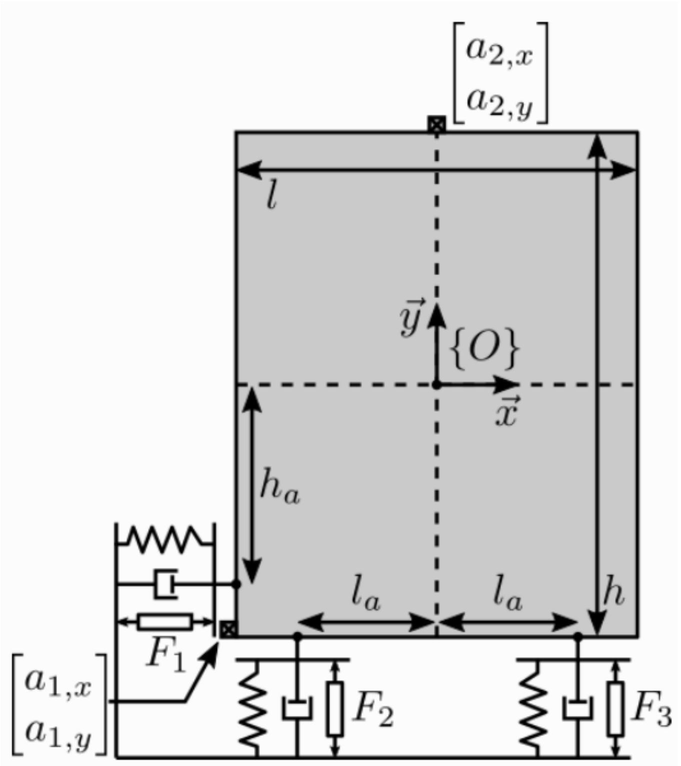
Model of Gravimeter

III. Jacobian decoupling

Analytical calculation of the Jacobians:

Sensor Jacobian:

$$J_{sens} = \begin{bmatrix} \frac{\partial q_{a1x}}{\partial x} & \frac{\partial q_{a1x}}{\partial y} & \frac{\partial q_{a1x}}{\partial \theta} \\ \frac{\partial q_{a1y}}{\partial x} & \frac{\partial q_{a1y}}{\partial y} & \frac{\partial q_{a1y}}{\partial \theta} \\ \frac{\partial q_{a2x}}{\partial x} & \frac{\partial q_{a2x}}{\partial y} & \frac{\partial q_{a2x}}{\partial \theta} \\ \frac{\partial q_{a2y}}{\partial \theta} & \frac{\partial q_{a2y}}{\partial \theta} & \frac{\partial q_{a2y}}{\partial \theta} \end{bmatrix}_{(x,y,\theta)=(0,0,0)} = \begin{bmatrix} 1 & 0 & -\frac{h}{2} \\ 0 & 1 & \frac{l}{2} \\ 1 & 0 & \frac{h}{2} \\ 0 & 1 & 0 \end{bmatrix}$$



Model of Gravimeter

Equations used to move from sensing accelerations in decoupled frame to sensing accelerations in cartesian frame.

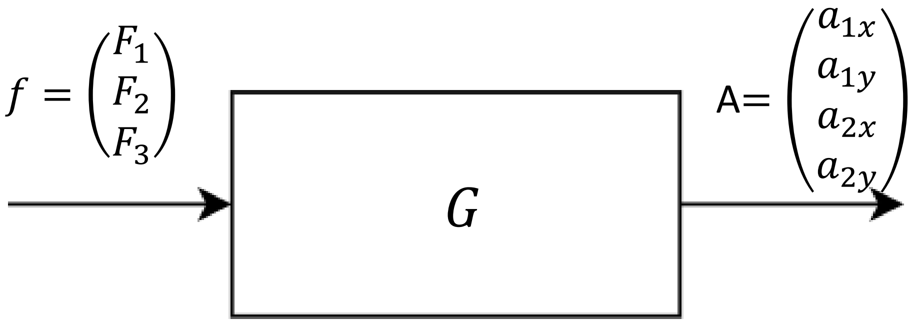
$$A = J_{sens} * A_x \quad \Rightarrow \quad A_x = (J_{sens})^{-1} * A$$

Where :

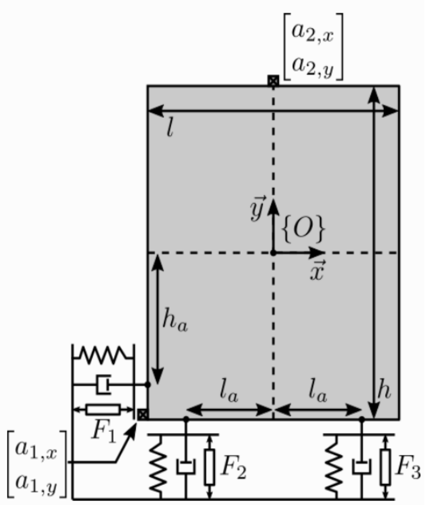
$A = \begin{pmatrix} a_{1x} \\ a_{1y} \\ a_{2x} \\ a_{2y} \end{pmatrix}$ Accelerations measured by the accelerometers

$A_x = \begin{pmatrix} a_x \\ a_y \\ a_\theta \end{pmatrix}$ Accelerations measured accelerations in cartesian coordinates

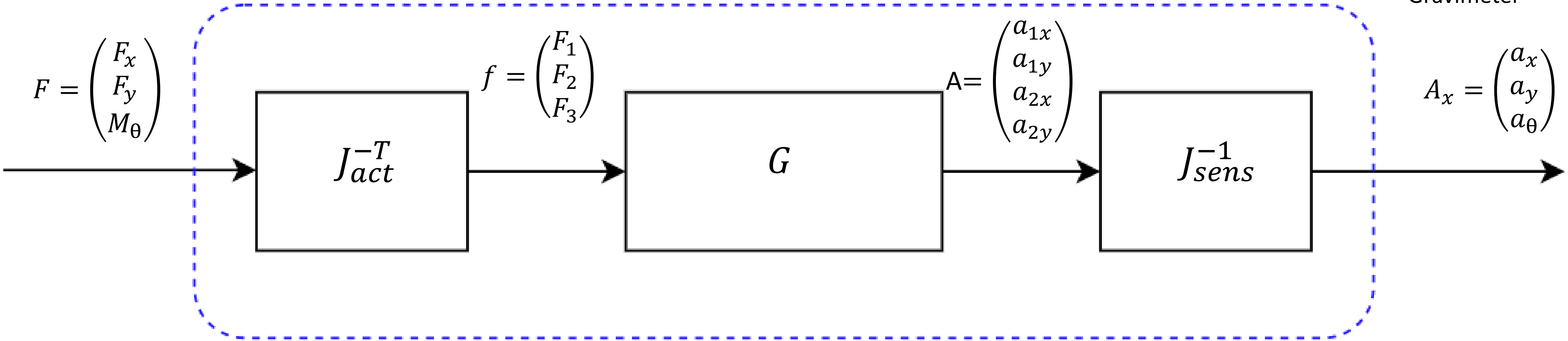
Jacobian centralized vs decentralized Control schemes:



Decentralized Scheme



Model of Gravimeter



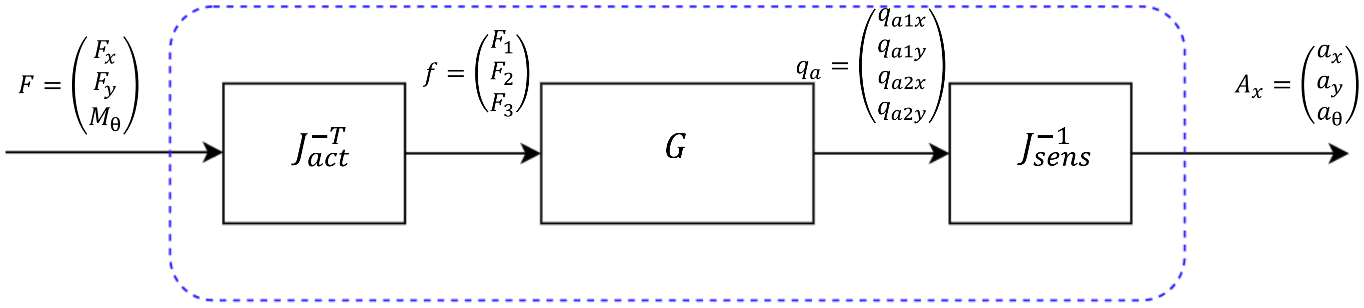
Centralized Scheme

III. Jacobian decoupling

Jacobian decoupling using Matlab:

Actuator and sensor Jacobian definition :

Matlab
```
Ja = [1 0 -h/2
      0 1  1/2
      1 0  h/2
      0 1  0];

Jt = [1 0 -ha
      0 1  la
      0 1 -la];
```



Centralized Scheme

Decoupled Plant definition:

Matlab
```
Gx = pinv(Ja)*G*pinv(Jt');
Gx.InputName  = {'Fx', 'Fy', 'Mz'};
Gx.OutputName  = {'Dx', 'Dy', 'Rz'};
```

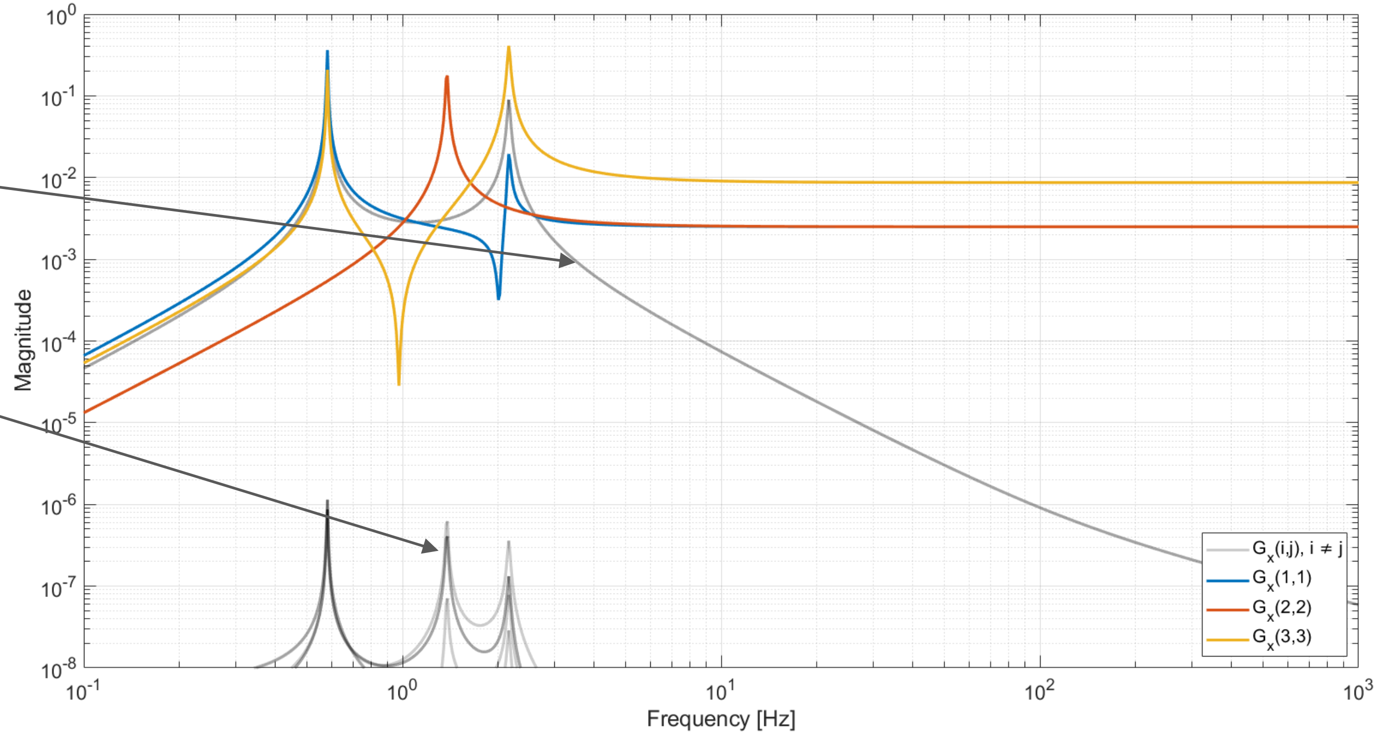
Two equal off diagonal elements G(1,3) and G(3,1) corresponding to coupling between translation in X direction and rotation around Z

Other off diagonal elements very small compared to diagonal elements

Negligible impact on diagonal elements and hence they are considered as decoupled directions

Problem: Coupling at low frequency ?

Open Loop Transfer Functions of the decoupled plant at COM



III. Jacobian decoupling

Question: Why decoupling the Jacobian at the COM lead to better decoupling at high frequency?

Roughly speaking :

$$\frac{X}{F} = [Ms^2 + K]^{-1}$$

At high frequency:

$$\frac{X}{F} \approx [Ms^2]^{-1}$$

and since mass matrix is diagonal at COM

Good decoupling at high frequency

At low frequency:

$$\frac{X}{F} \approx [K]^{-1}$$

and since K matrix is not diagonal at COM in the given system

Bad decoupling at low frequency

Problem: Coupling at low frequency ?

Idea: Maybe try having a diagonal stiffness matrix

HOW ???

III. Jacobian decoupling

Jacobian decoupling at center of stiffness (COK):

COK is the geometrical point corresponding to obtaining diagonal stiffness matrix $\mathcal{K}$:

$$\mathcal{K} = \begin{bmatrix} k_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & k_n \end{bmatrix} \qquad K_{\{K\}} = J_{\{K\}}^T \mathcal{K} J_{\{K\}}$$

COK of this model

Note: For higher degrees of freedom additional conditions could apply.

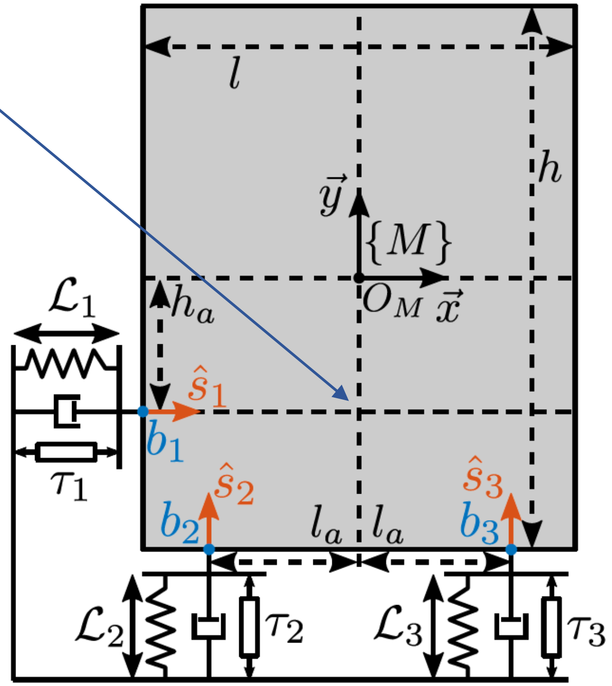
Conditions for the existence of COK for a planar system:

$$k_i \hat{s}_i \hat{s}_i^T = diag\ matrix$$

$$k_i \hat{s}_i (b_{i,x} \hat{s}_{i,y} - b_{i,y} \hat{s}_{i,x}) = 0$$

With :

- $\hat{s}_i$:unit vector corresponding to the struts
- k_i: stiffness of the struts
- b_i :location of joints on the platform



Collocated Model

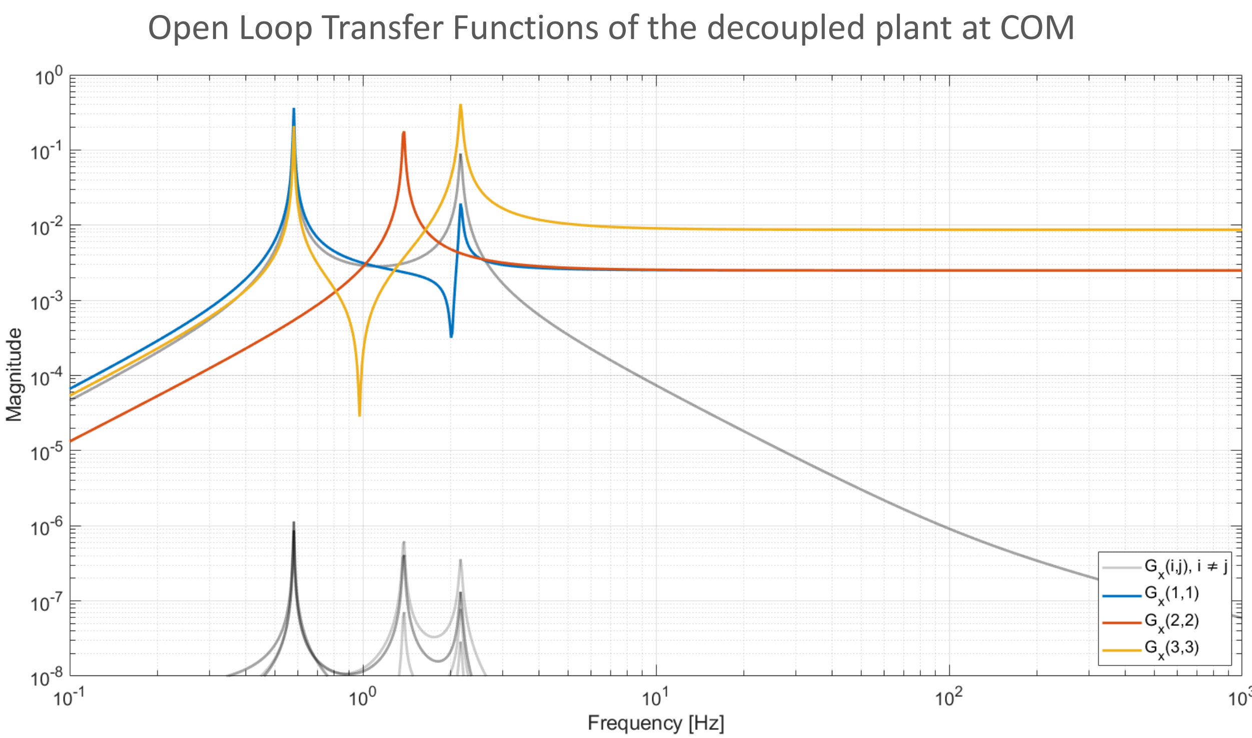
At the end the distance between the COM an COK could be calculated :

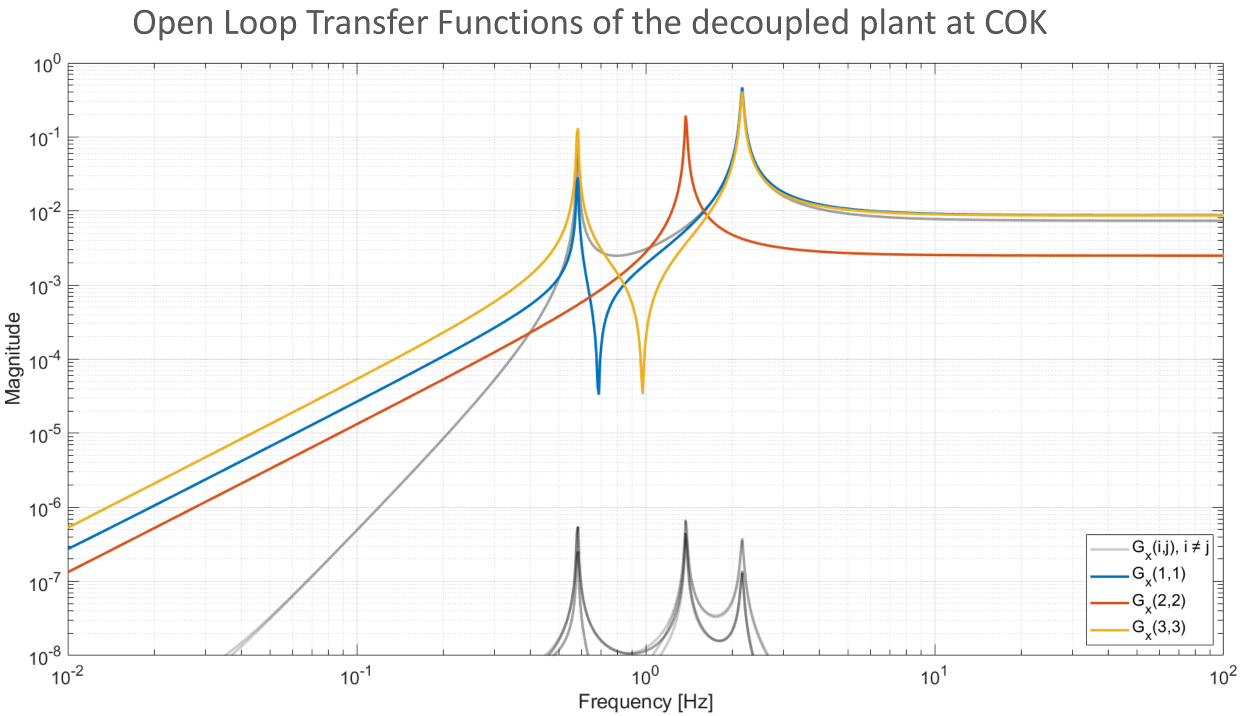
$$^{M}O_K = \begin{bmatrix} k_i \hat{s}_{i,y} \hat{s}_i & -k_i \hat{s}_{i,x} \hat{s}_i \end{bmatrix}^{-1} k_i (^{M}b_{i,x} \hat{s}_{i,y} - {}^{M}b_{i,y} \hat{s}_{i,x}) \hat{s}_i$$

⇨ Calculate the Jacobians considering COK as new reference

III. Jacobian decoupling

Jacobian decoupling at COK and COM:





Good decoupling at high frequency achieved

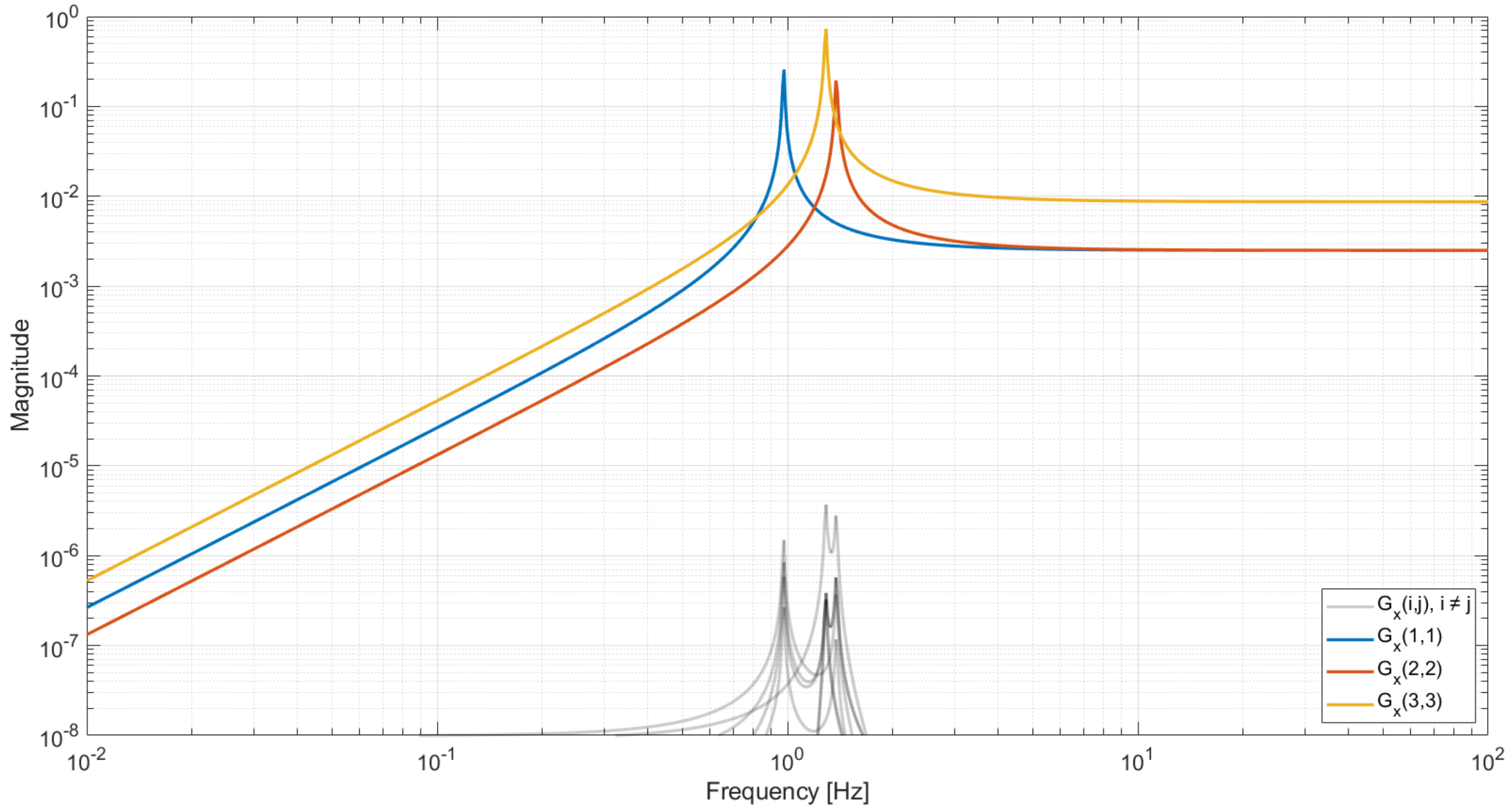
Good decoupling at low frequency achieved

Question: What could be done to obtain full decoupling over all bandwidth using one Jacobian ?

Jacobian decoupling at collocated COM and COK:

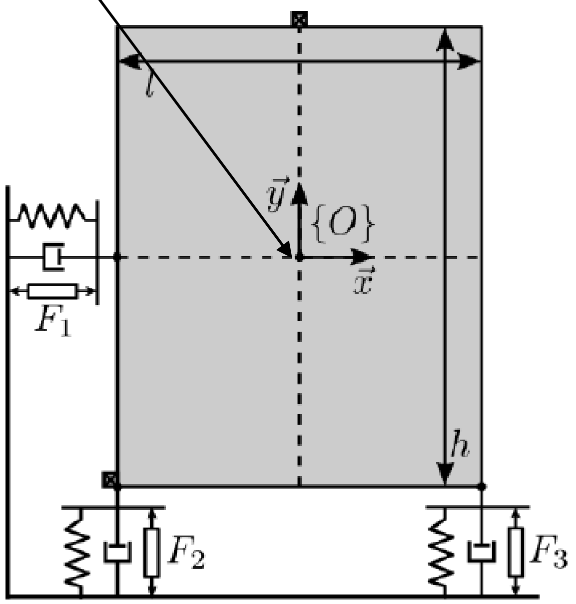
Answer: Place stiffnesses in the system in such a way that COK and COM are collocated

Open Loop Transfer Functions of the decoupled plant at collocated COK and COM



Very good decoupling over all the bandwidth

Collocated COM and COK



Plant with collocated COM and COK

Modal decoupling

IV. Modal decoupling

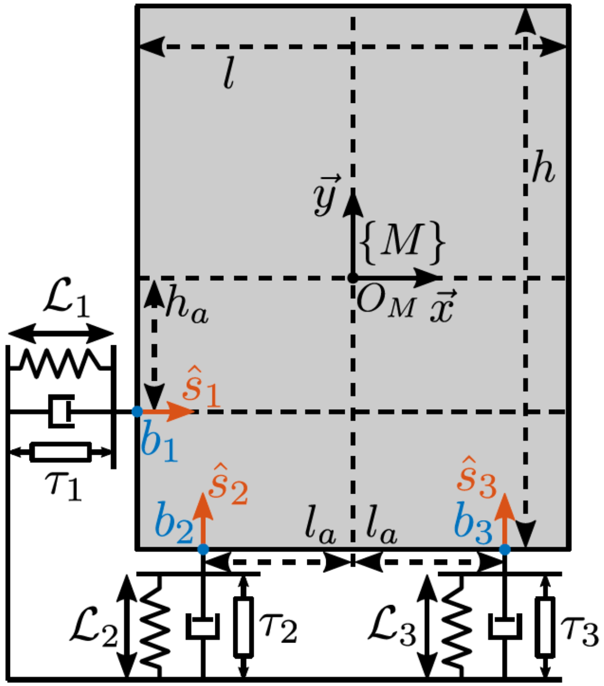
Simplified Simscape model :

New parameter definition :

Matlab

```
%% System parameters
l  = 1.0; % Length of the mass [m]
h  = 2*1.7; % Height of the mass [m]

la = l/2; % Position of Act. [m]
ha = h/2; % Position of Act. [m]

m = 400; % Mass [kg]
I = 115; % Inertia [kg m^2]

%% Actuator Damping [N/(m/s)]
c1 = 2e1;
c2 = 2e1;
c3 = 2e1;

%% Actuator Stiffness [N/m]
k1 = 15e3;
k2 = 15e3;
k3 = 15e3;

%% Unit vectors of the actuators
s1 = [1;0];
s2 = [0;1];
s3 = [0;1];

%%  Location of the joints
Mb1 = [-l/2;-ha];
Mb2 = [-la; -h/2];
Mb3 = [ la; -h/2];

%% Jacobian matrix
J = [s1', Mb1(1)*s1(2)-Mb1(2)*s1(1);
     s2', Mb2(1)*s2(2)-Mb2(2)*s2(1);
     s3', Mb3(1)*s3(2)-Mb3(2)*s3(1)];

%% Stiffnesss and Damping matrices of the struts
Kr = diag([k1,k2,k3]);
Cr = diag([c1,c2,c3]);
```

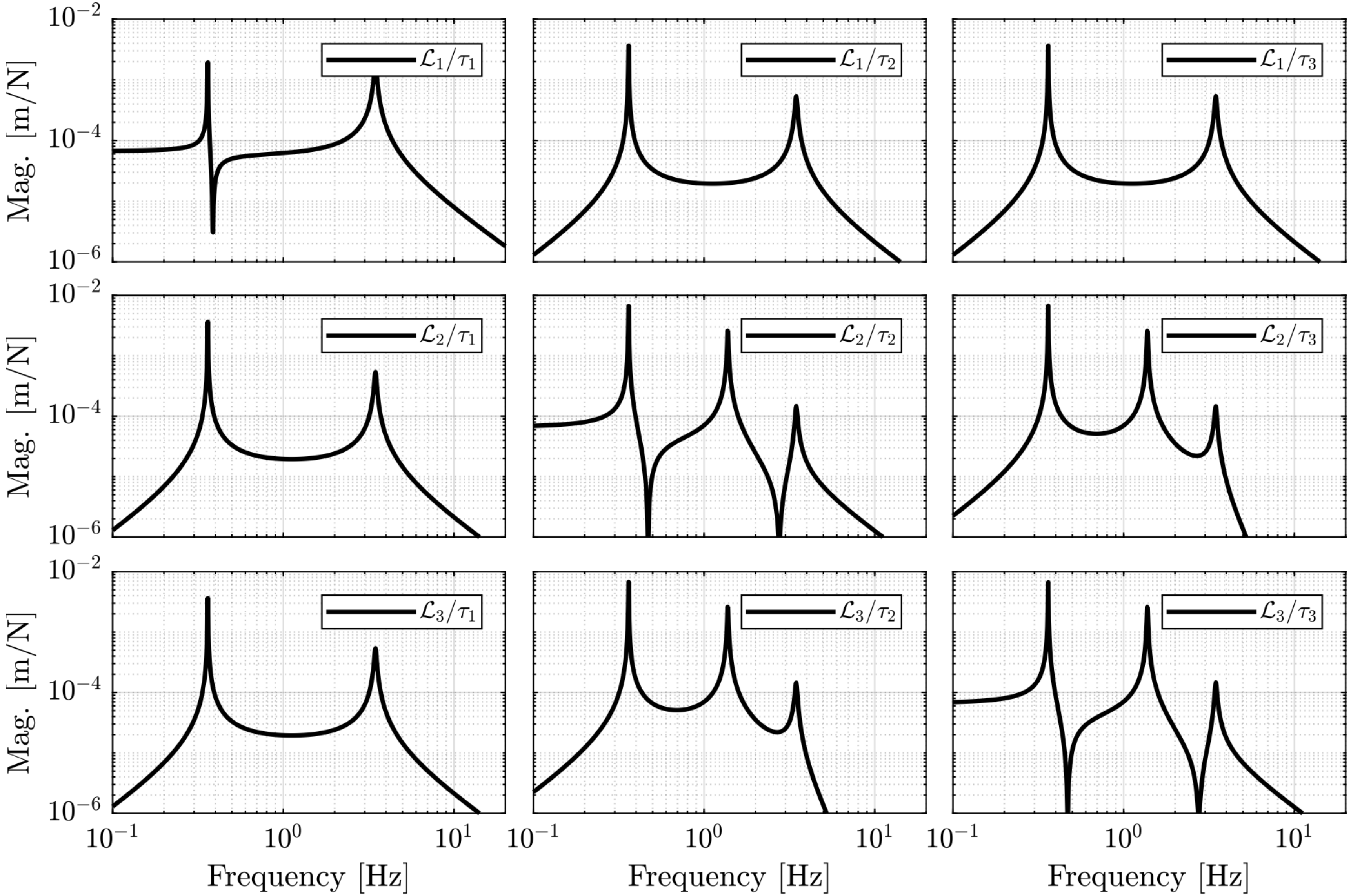
New simplified plant: 3 actuators with 3 collocated displacement sensors



Collocated Model

IV. Modal decoupling

Open Loop Transfer Functions from different Actuators to different sensors



Too many Off diagonal elements

Highly Coupled system

IV. Modal decoupling

Analytical development of modal decomposition:

Modal decoupling depends on the equations of motion:

$$M\ddot{x} + C\dot{x} + Kx = F$$

Measurement output combination of the motion variable x:

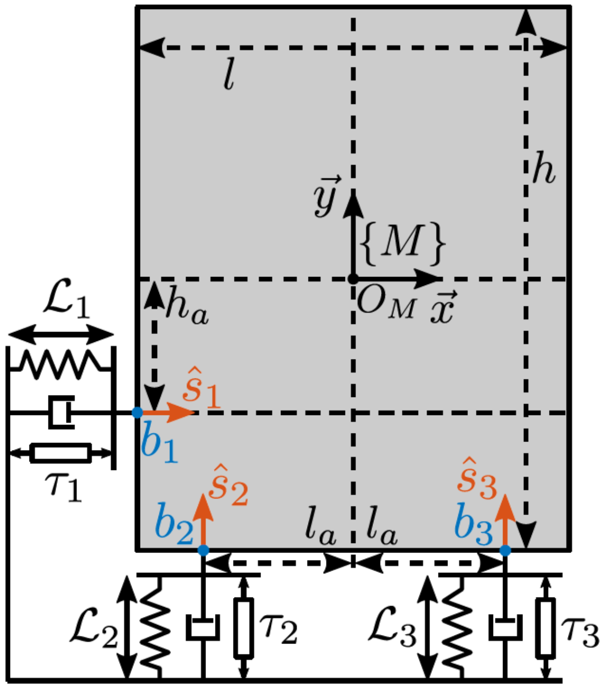
$$y = C_{ox}x + C_{ov}\dot{x}$$

Then apply change of variables :

$$x = \Phi x_m$$

With :

- x_m modal amplitudes
- Φ a matrix whose columns are the mode shapes of the system



Collocated Model

Map actuator forces as follows using COM jacobian:

$$F = J^T\tau$$

IV. Modal decoupling

New equation of motion become:

$$M\Phi\ddot{x}_m + C\Phi\dot{x}_m + K\Phi x_m = J^T\tau$$

And new form of measured output becomes:

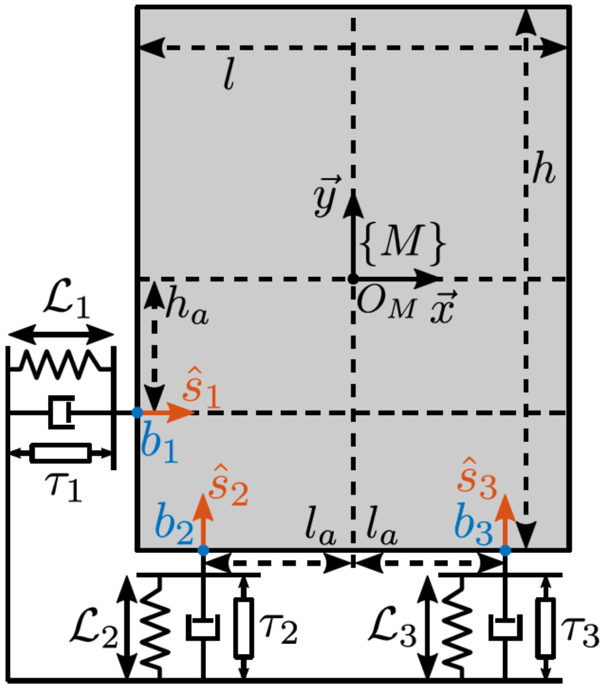
$$y = C_{ox}\Phi x_m + C_{ov}\Phi\dot{x}_m$$

After multiplying both sides with Φ^T:

$$\Phi^T M\Phi\ddot{x}_m + \Phi^T C\Phi\dot{x}_m + \Phi^T K\Phi x_m = \Phi^T J^T\tau$$

We denote :

- $M_{modal} = \Phi^T M\Phi = diag(\mu_i)$ as modal mass matrix
- $C_{modal} = \Phi^T C\Phi = diag(2\xi_i\mu_i w_i)$ (classical damping)
- $K_{modal} = \Phi^T K\Phi = diag(\mu_i w_i^2)$ (modal stiffness matrix)



Collocated Model

IV. Modal decoupling

Substituting again in EOM yields :

$$\ddot{x}_m + 2\Xi\Omega\dot{x}_m + \Omega^2 x_m = \mu^{-1}\Phi^T J^T \tau$$

With :

- $\mu = diag(\mu_i)$
- $\Omega = diag(w_i)$
- $\Xi = diag(\xi_i)$

Modal input matrix:

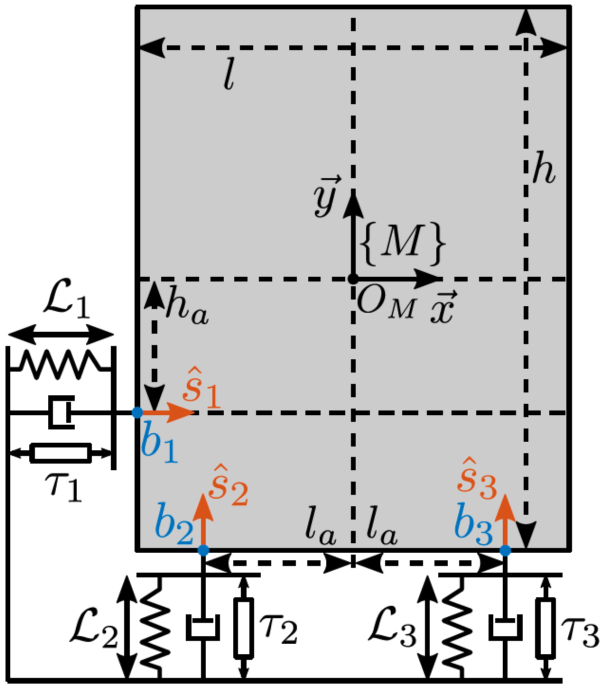
$$B_m = \mu^{-1}\Phi^T J^T$$

Modal output matrices:

$$C_m = C_{ox}\Phi + sC_{ov}\Phi$$

Modal input :

$$\tau_m = B_m\tau$$



Collocated Model

IV. Modal decoupling

The final transfer function from τ_m to x_m can be shown:

$$\frac{x_m}{\tau_m} = (I_n s^2 + 2\Xi\,\Omega s + \Omega^2)^{-1}$$

⇩

This form correspond to a diagonal transfer matrix

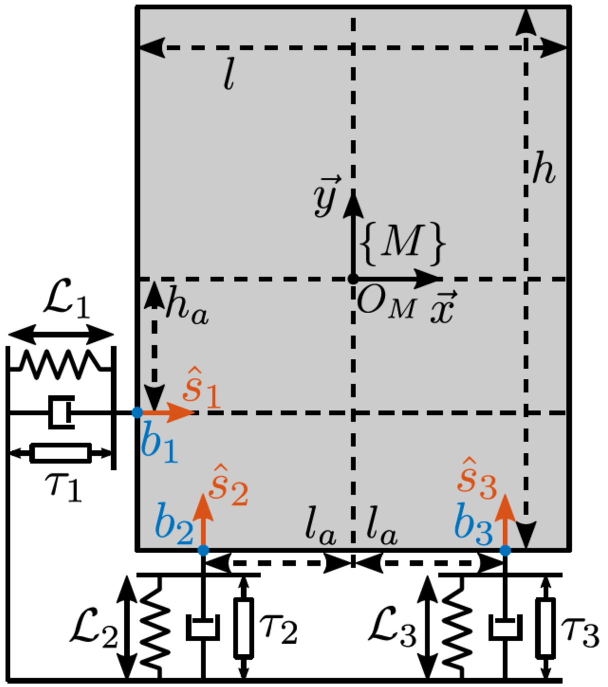
⇩

Dynamics of the system are decoupled from τ_m to x_m

Going back to y and τ, they can be expressed using modal variables :

$$\frac{y}{\tau} = (C_{ox}\Phi + sC_{ov}\Phi)(I_n s^2 + 2\Xi\,\Omega s + \Omega^2)^{-1}(\mu^{-1}\Phi^T J^T)$$

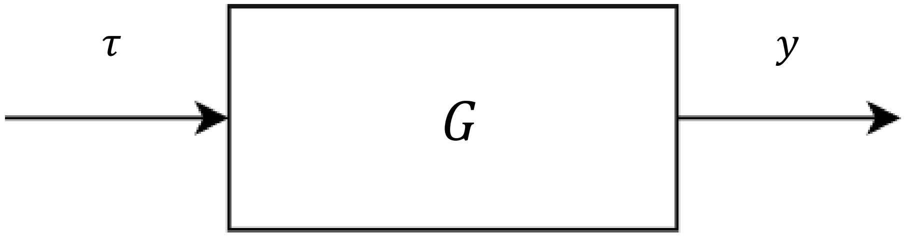
C_m — *diagonal matrix* — B_m



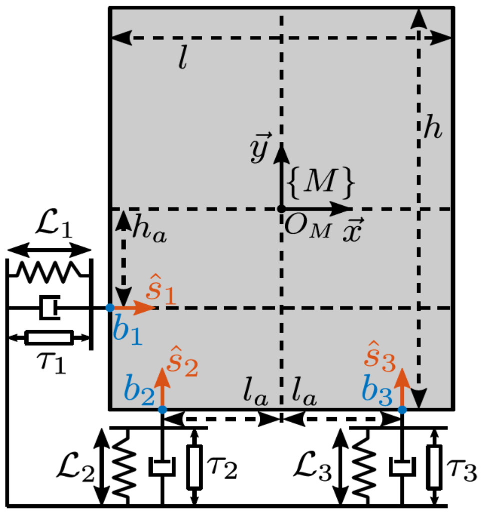
Collocated Model

IV. Modal decoupling

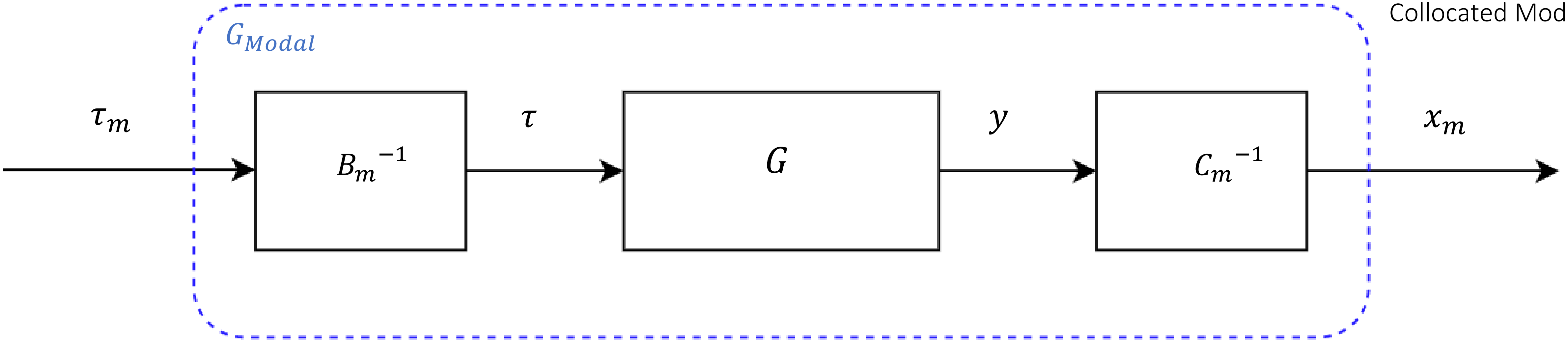
Centralized vs decentralized Control schemes:



Decentralized Scheme



Collocated Model



Centralized Scheme

To get C_m and B_m we need to compute C_{ox} , C_{ov}, Φ, μ and J:

$$x = \begin{bmatrix} x \\ y \\ R_z \end{bmatrix}$$

get Φ from K and M matrices

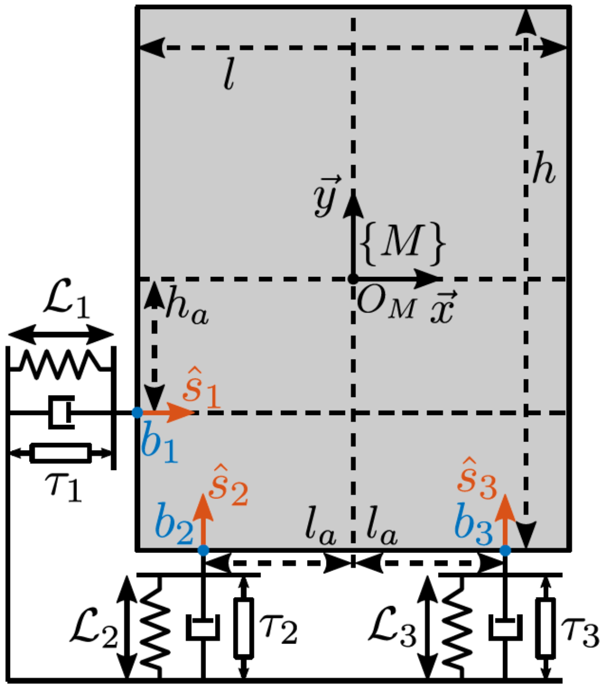
J is Jacobian

$$y = \mathcal{L} = Jx$$

And $y = C_{ox}x + C_{ov}\dot{x}$

$$C_{ox} = J$$
$$C_{ov} = 0$$

Calculated similar to what seen before



Collocated Model

Matlab
```
%% Modal Decomposition
[V,D] = eig(M\K);

%% Modal Mass Matrix
mu = V'*M*V;

%% Modal output matrix
Cm = J*V;

%% Modal input matrix
Bm = inv(mu)*V'*J';
```

IV. Modal decoupling

Modal decomposition using Matlab:

$$B_m = \begin{pmatrix} -0.0004 & -0.0007 & -0.0007 \\ -0.051 & 0.0041 & -0.0041 \\ 0 & 0.0025 & 0.0025 \end{pmatrix}$$

$$C_m = \begin{pmatrix} -0.1 & -1.8 & 0 \\ -0.2 & 0{,}5 & 1 \\ 0.2 & -0.5 & 1 \end{pmatrix}$$

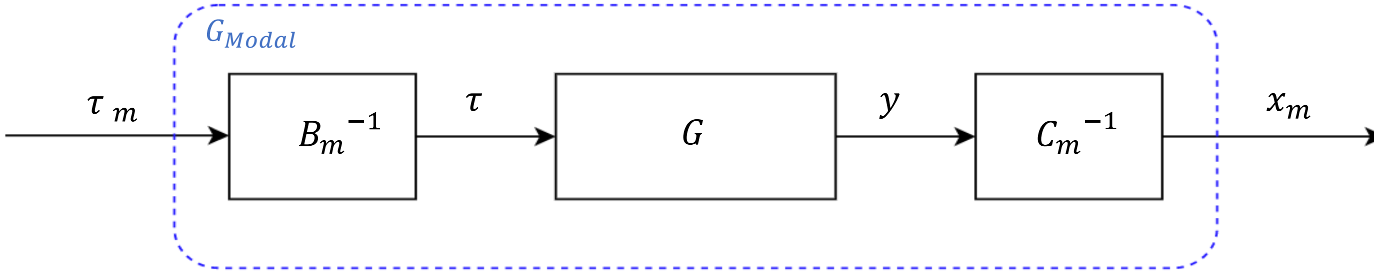
Modal decomposition performed using the following matlab command:

Matlab
```
Gm = inv(Cm)*G*inv(Bm);
```

Plant perfectly decoupled over the whole bandwidth

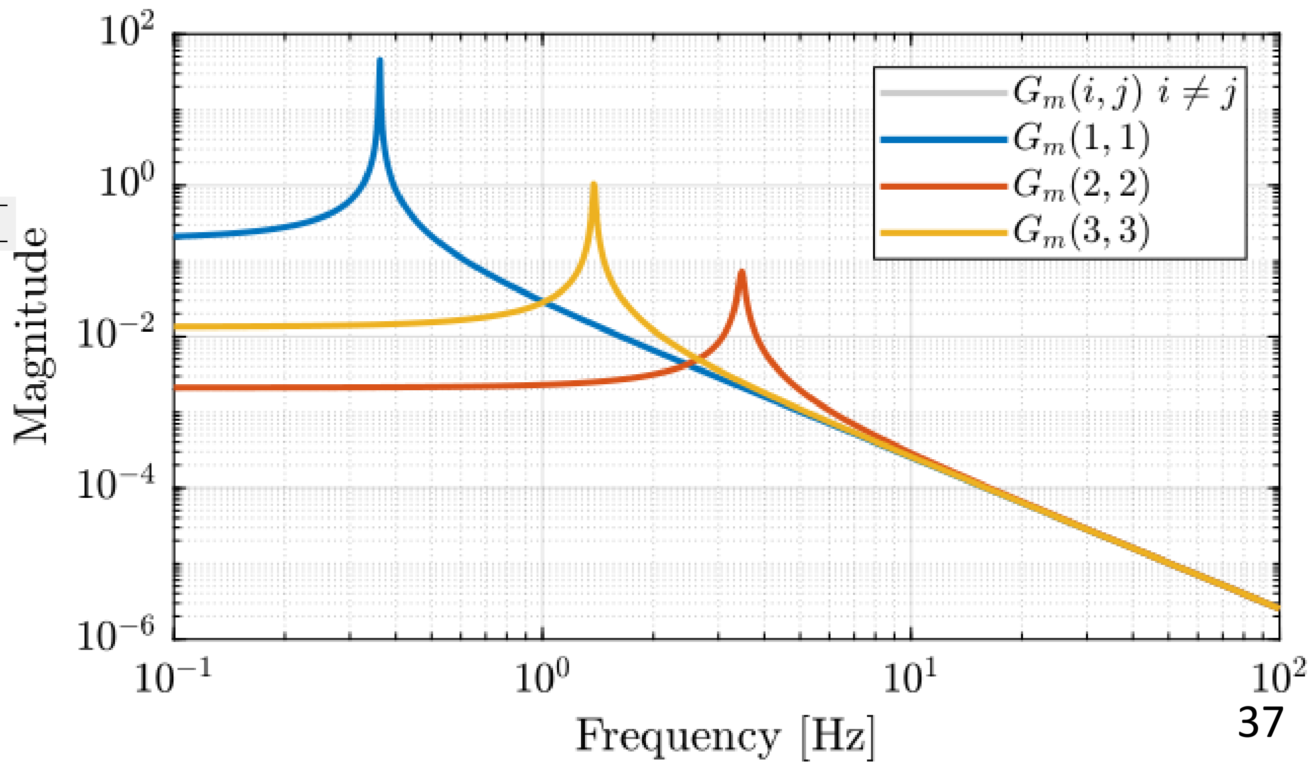
⇩

It is possible to use SISO control approaches to actively control all decoupled modes.



Centralized Scheme

Open Loop Transfer Functions of the decoupled plant using modal decomposition



Comparison and Conclusions

V. Comparison and Conclusions

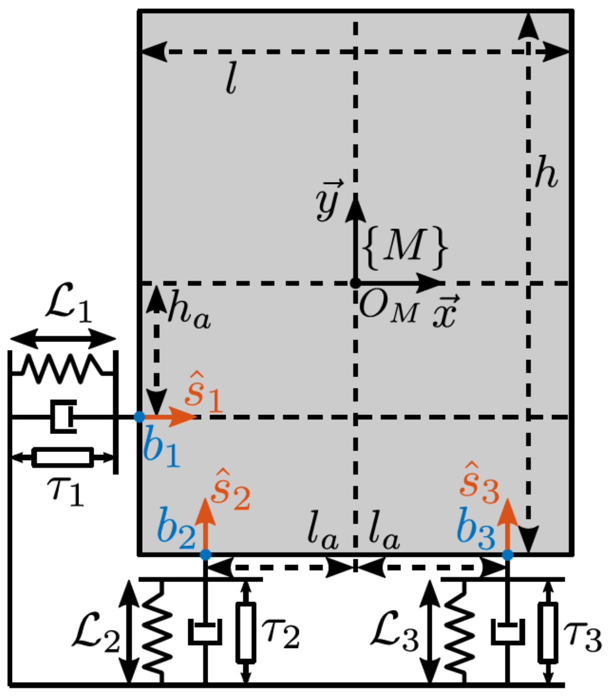
Compare all presented decoupling strategies:

- SVD decomposition
- Jacobian decoupling at COM
- Jacobian decoupling at COK
- Modal decomposition

Do that using a simplified collocated system.

Apply SVD and Jacobian decoupling at COM for this model and compare them with results obtained by model decomposition and Jacobian decoupling at COM

New simplified plant: 3 actuators with 3 collocated displacement sensors

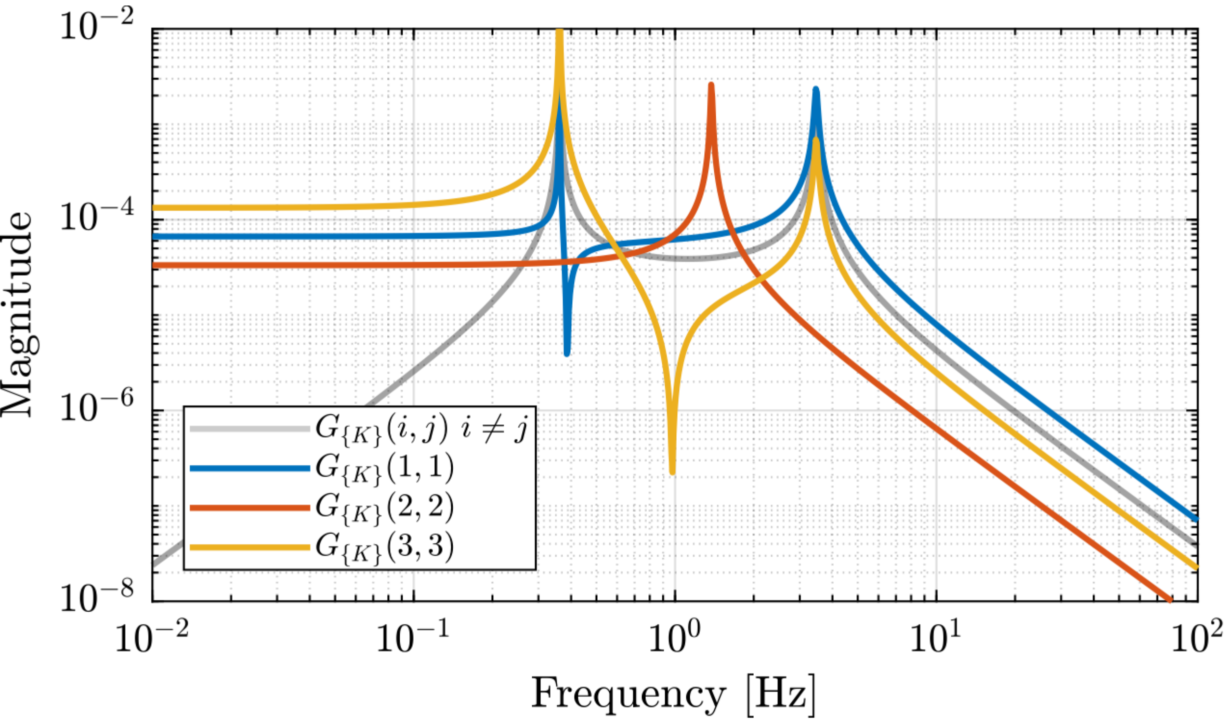


Collocated Model

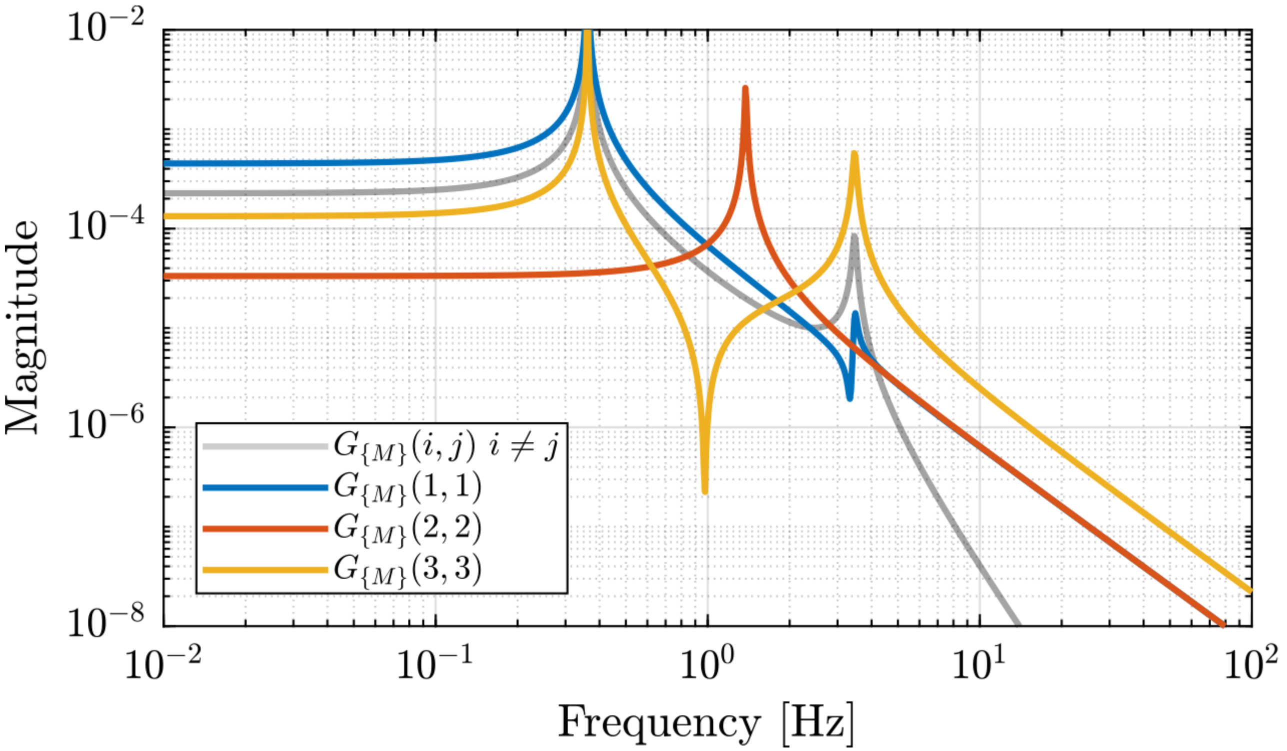
V. Comparison and Conclusions

Jacobian decoupling of the collocated system:

Open Loop Transfer Functions of the decoupled plant using Jacobian at COK



Open Loop Transfer Functions of the decoupled plant using Jacobian at COM



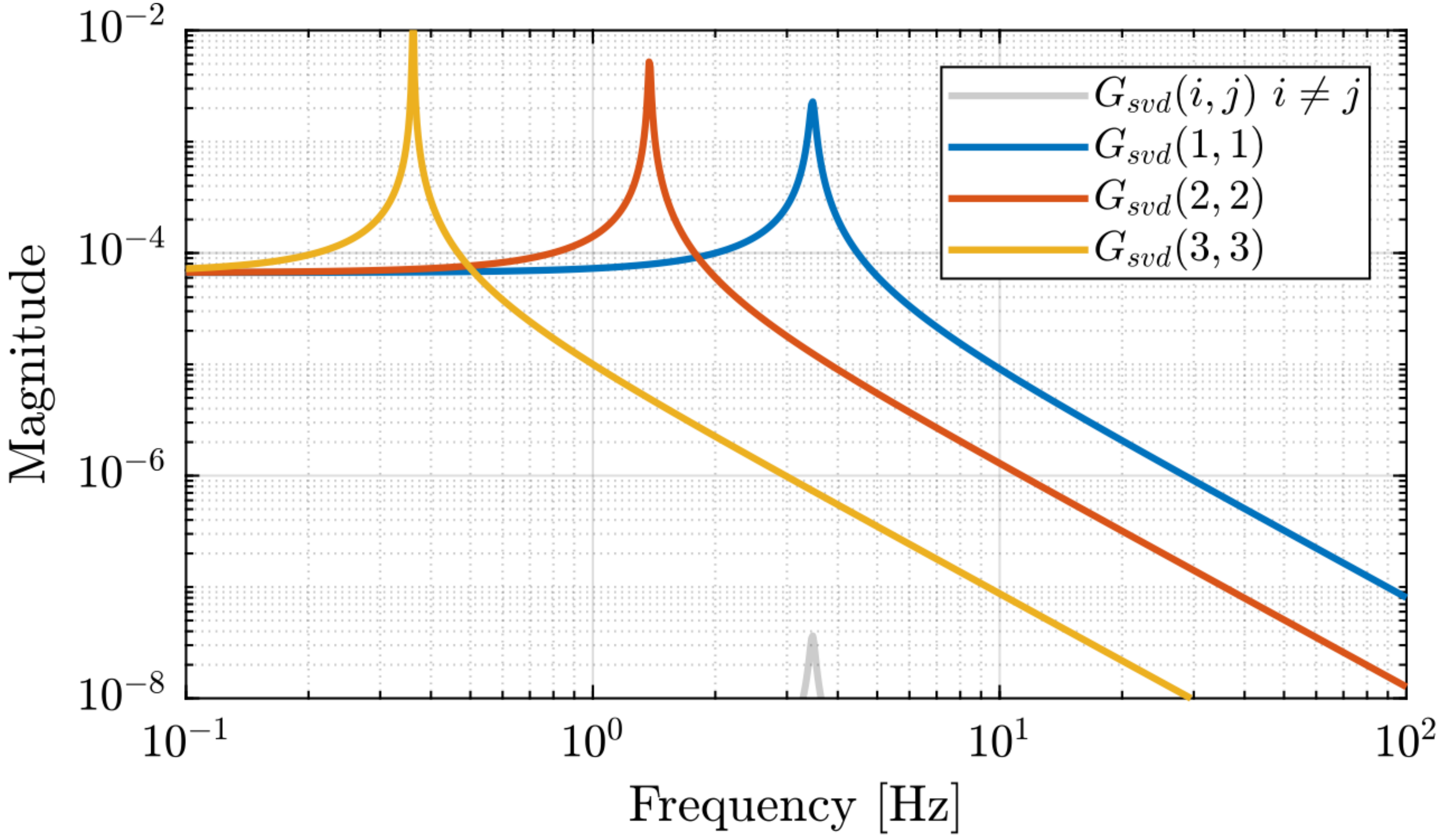
Same compromise as shown before:

- Decouple at COK for good decoupling at low frequencies
- Decouple at COM for good decoupling at high frequencies

V. Comparison and Conclusions

SVD decoupling of the collocated system:

Open Loop Transfer Functions of the decoupled plant using SVD



Decoupling frequency = 10Hz

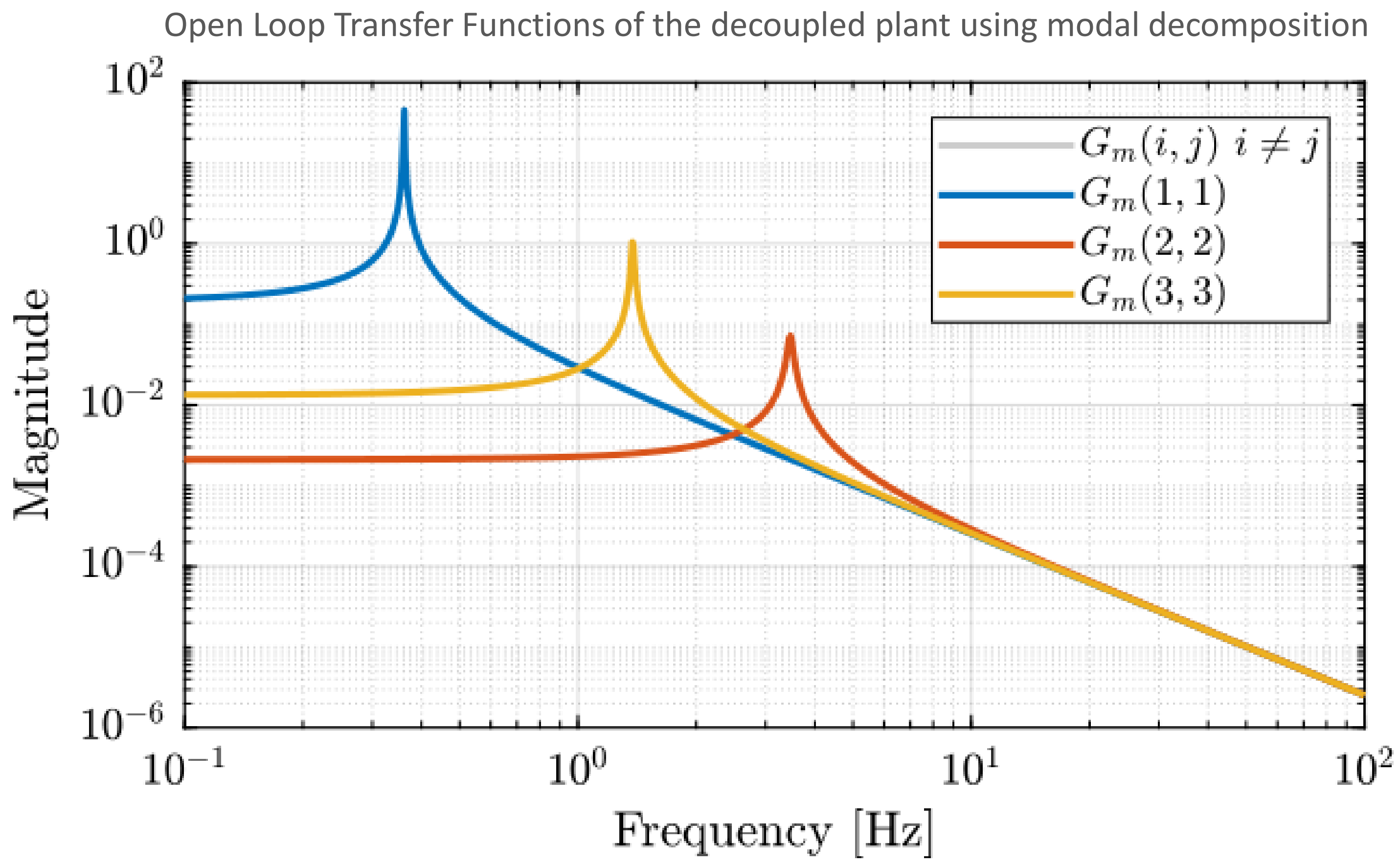
Perfect decoupling over all the frequency bandwidth

SVD gave much better decoupling when having collocated actuators and sensors

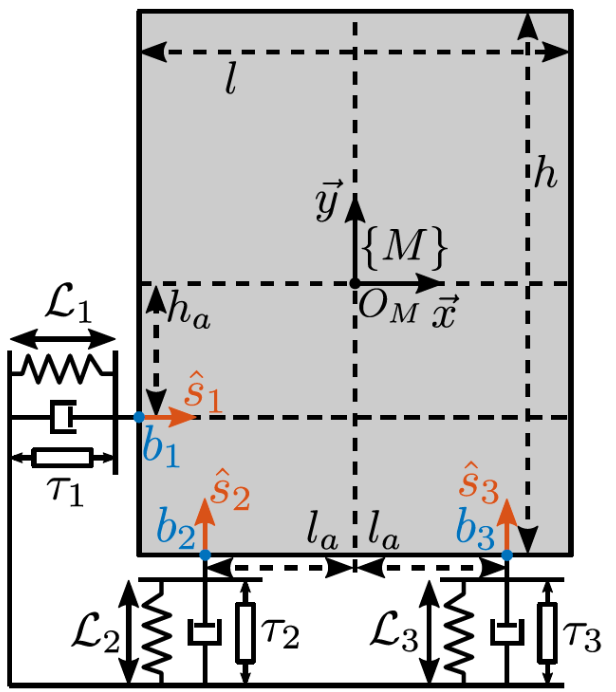
Better performance then Jacobians decoupling for this system

V. Comparison and Conclusions

Modal decoupling of the collocated system:



New simplified plant: 3 actuators with 3 collocated displacement sensors



Collocated Model

Perfect decoupling over all bandwidth

Have similar performance to SVD and much better decoupling compared to Jacobians

V. Comparison and Conclusions

Comparison between Three different strategies:

	Jacobians	Modal Decomposition	SVD
Philosophy	Topology Driven	Physics Driven	Data Driven
Requirements	Known geometry	Known equations of motion	Identified FRF
Decoupling Matrices	Decoupling using J obtained from geometry	Decoupling using Φ obtained from modal decomposition	Decoupling using U and V obtained from SVD

V. Comparison and Conclusions

Comparison between Three different strategies:

	Jacobians	Modal Decomposition	SVD
Decoupled Plant	$G_{\{O\}} = J_{\{O\}}^{-1} G J_{\{O\}}^{-T}$	$G_m = C_m^{-1} G B_m^{-1}$	$G_{SVD} = U^{-1} G(s) V^{-T}$
Physical Interpretation	Forces/Torques to Displacement/Rotation in chosen frame	Inputs to excite individual modes	Directions of max to min controllability/observability
Decoupling Properties	Decoupling at low or high frequency depending on the chosen frame	Good decoupling at all frequencies	Good decoupling near the chosen frequency

V. Comparison and Conclusions

Comparison between Three different strategies:

	Jacobians	Modal Decomposition	SVD
Pros	• Physical inputs / outputs • Good decoupling at High frequency (diagonal mass matrix if Jacobian taken at the COM) • Good decoupling at Low frequency (if Jacobian taken at specific point)	• Target specific modes • 2nd order diagonal plant	• Good Decoupling near the crossover • Very General

V. Comparison and Conclusions

Comparison between Three different strategies:

	Jacobians	Modal Decomposition	SVD
Cons	• Coupling between force/rotation may be high at low frequency (non diagonal terms in K) • If good decoupling at all frequencies => requires specific mechanical architecture	• Need analytical equations	• Loose the physical meaning of inputs/outputs • Decoupling depends on the real approximation validity

Thank you