An Efficient Approximation of Optimal Damping in Mechanical Systems

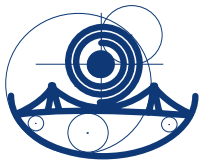
Ninoslav Truhar

University J. J. Strossmayer of Osijek
Department of Mathematics
Trg Ljudevita Gaja 6
31000 Osijek, Croatia
http://www.mathos.unios.hr

ntruhar@mathos.hr

Coauthors: **Ivica Nakić and Maja Petrač**

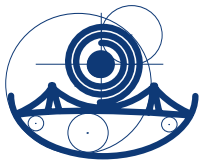
[Workshop on control of dynamical systems, Dubrovnik, Croatia] 15.6.2021



The talk contains results from

N. Truhar, and I. Nakić; *Damping Optimization Of Mechanical Systems;* work in progress

M. Petrač, N. Truhar and I. Nakić; *Damping optimization of linear vibrational systems with a singular mass matrix;* submitted



The talk contains results from

N. Truhar, and I. Nakić; *Damping Optimization Of Mechanical Systems;* work in progress

M. Petrač, N. Truhar and I. Nakić; *Damping optimization of linear vibrational systems with a singular mass matrix;* submitted

Consider a damped linear vibrational system

$$M\ddot{x} + D\dot{x} + Kx = 0,$$
$$x(0) = x_0, \quad and \quad \dot{x}(0) = \dot{x}_0$$

where $M, D, K \in \mathbb{R}^{n\times n}$ (mass, damping, stiffness), M and $K > 0$ (positive definite).

$D = C_u + C$, $C_u > 0$ internal damping, $C \geq 0$ external (viscous) damping.

A very important question arises in consideration of such systems: For the given mass (M) and stiffness (K) determine the best (optimal) damping which will insure an optimal evanescence.

Consider a damped linear vibrational system

$$M\ddot{x} + D\dot{x} + Kx = 0,$$
$$x(0) = x_0, \quad and \quad \dot{x}(0) = \dot{x}_0$$

where $M, D, K \in \mathbb{R}^{n\times n}$ (mass, damping, stiffness), M and $K > 0$ (positive definite).

$D = C_u + C$, $C_u > 0$ internal damping, $C \geq 0$ external (viscous) damping.

A very important question arises in consideration of such systems: For the given mass (M) and stiffness (K) determine the best (optimal) damping which will insure an optimal evanescence.

Consider a damped linear vibrational system

$$M\ddot{x} + D\dot{x} + Kx = 0,$$
$$x(0) = x_0, \quad and \quad \dot{x}(0) = \dot{x}_0$$

where $M, D, K \in \mathbb{R}^{n\times n}$ (mass, damping, stiffness), M and $K > 0$ (positive definite).

$D = C_u + C$, $C_u > 0$ internal damping, $C \geq 0$ external (viscous) damping.

A very important question arises in consideration of such systems:
For the given mass (M) and stiffness (K) determine the best (optimal) damping which will insure an optimal evanescence.

Consider a damped linear vibrational system

$$M\ddot{x} + D\dot{x} + Kx = 0,$$
$$x(0) = x_0, \quad and \quad \dot{x}(0) = \dot{x}_0$$

where $M, D, K \in \mathbb{R}^{n\times n}$ (mass, damping, stiffness), M and $K > 0$ (positive definite).

$D = C_u + C$, $C_u > 0$ internal damping, $C \geq 0$ external (viscous) damping.

A very important question arises in consideration of such systems:
For the given mass (M) and stiffness (K) determine the best (optimal) damping which will insure an optimal evanescence.

Consider a damped linear vibrational system

$$M\ddot{x} + D\dot{x} + Kx = 0,$$
$$x(0) = x_0, \quad and \quad \dot{x}(0) = \dot{x}_0$$

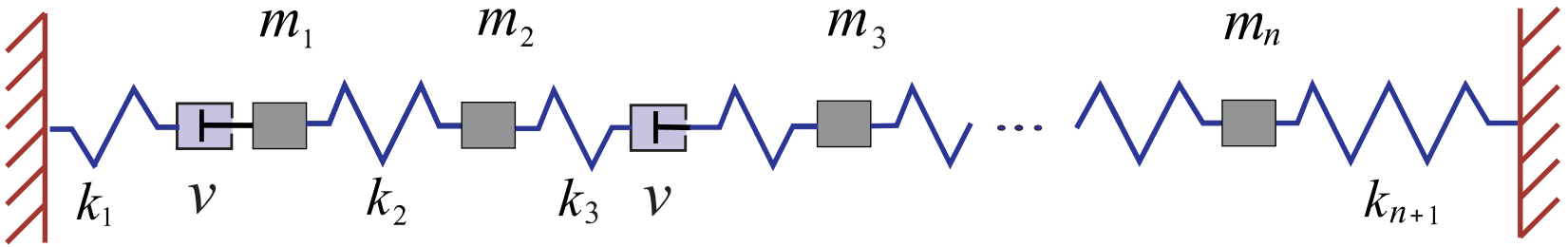
where $M, D, K \in \mathbb{R}^{n\times n}$ (mass, damping, stiffness), M and $K > 0$ (positive definite).

$D = C_u + C$, $C_u > 0$ internal damping, $C \geq 0$ external (viscous) damping.

A very important question arises in consideration of such systems: For the given mass (M) and stiffness (K) determine the best (optimal) damping which will insure an optimal evanescence.

Example I

Consider *n-mass oscillator or oscillator ladder*



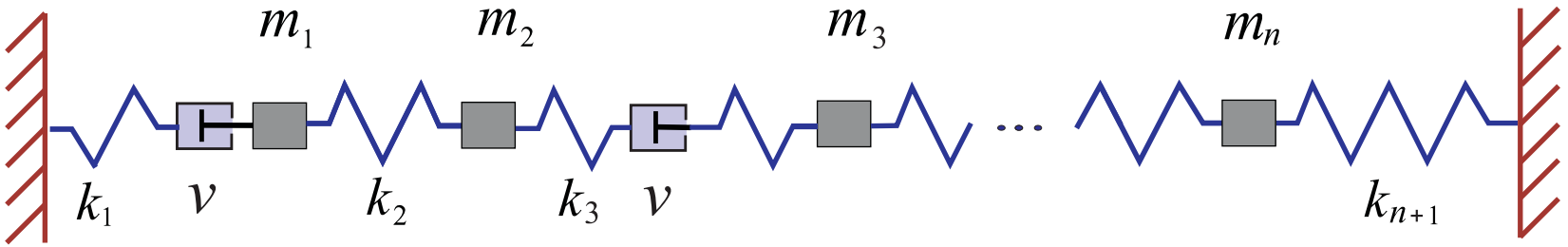
$$M = \mathrm{diag}(m_1, m_2, \ldots, m_n), \quad D \equiv C_u + C = C_u + v_1 e_1 e_1^T + v_3 e_3 e_3^T.$$

$$K = \begin{pmatrix} k_1 + k_2 & -k_2 & & & \\ -k_2 & k_2 + k_3 & -k_3 & & \\ & \ddots & \ddots & \ddots & \\ & & -k_{n-1} & k_{n-1} + k_n & -k_n \\ & & & -k_n & k_n + k_{n+1} \end{pmatrix},$$

$m_i > 0$ - masses, v_i - viscosity applied on the i-th mass,
$k_i > 0$ - stiffnesses

Example I

Consider *n-mass oscillator or oscillator ladder*



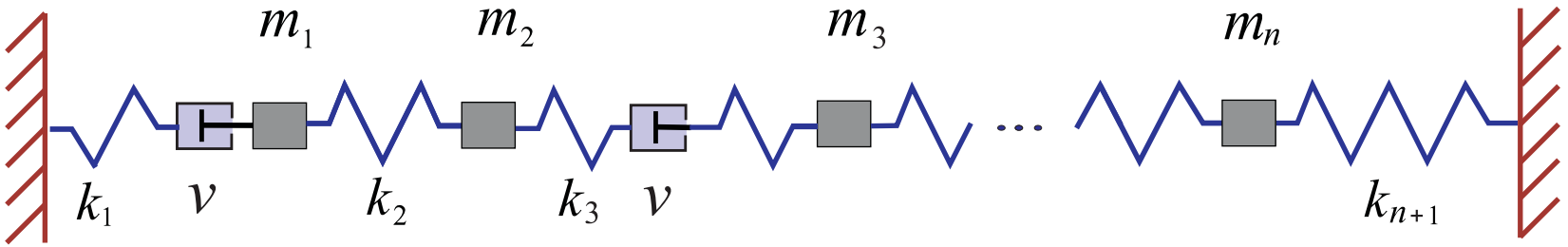
$$M = \mathrm{diag}(m_1, m_2, \ldots, m_n), \quad D \equiv C_u + C = C_u + v_1 e_1 e_1^T + v_3 e_3 e_3^T.$$

$$K = \begin{pmatrix} k_1 + k_2 & -k_2 & & & \\ -k_2 & k_2 + k_3 & -k_3 & & \\ & \ddots & \ddots & \ddots & \\ & & -k_{n-1} & k_{n-1} + k_n & -k_n \\ & & & -k_n & k_n + k_{n+1} \end{pmatrix},$$

$m_i > 0$ - masses, v_i - viscosity applied on the i-th mass,
$k_i > 0$ - stiffnesses

Example I

Consider *n-mass oscillator or oscillator ladder*



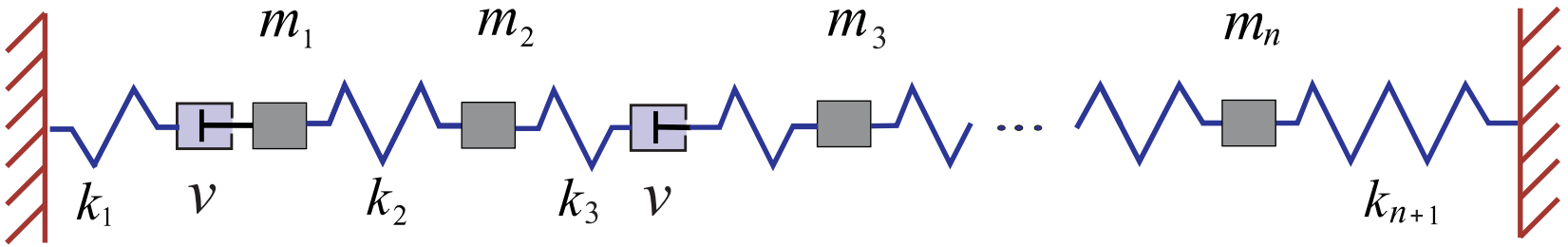
$$M = \mathrm{diag}(m_1, m_2, \ldots, m_n), \quad D \equiv C_u + C = C_u + v_1 e_1 e_1^T + v_3 e_3 e_3^T.$$

$$K = \begin{pmatrix} k_1 + k_2 & -k_2 & & & \\ -k_2 & k_2 + k_3 & -k_3 & & \\ & \ddots & \ddots & \ddots & \\ & & -k_{n-1} & k_{n-1} + k_n & -k_n \\ & & & -k_n & k_n + k_{n+1} \end{pmatrix},$$

$m_i > 0$ - masses, v_i - viscosity applied on the i-th mass,
$k_i > 0$ - stiffnesses

Example I

Consider *n-mass oscillator or oscillator ladder*



$$M = \mathrm{diag}(m_1, m_2, \ldots, m_n), \quad D \equiv C_u + C = C_u + v_1 e_1 e_1^T + v_3 e_3 e_3^T.$$

$$K = \begin{pmatrix} k_1 + k_2 & -k_2 & & & \\ -k_2 & k_2 + k_3 & -k_3 & & \\ & \ddots & \ddots & \ddots & \\ & & -k_{n-1} & k_{n-1} + k_n & -k_n \\ & & & -k_n & k_n + k_{n+1} \end{pmatrix},$$

$m_i > 0$ - masses, v_i - viscosity applied on the i-th mass,
$k_i > 0$ - stiffnesses

For this optimization problem one needs optimization criteria

- Spectral abscissa criterion

$$\max_k \operatorname{Re}(\lambda_k) \quad \rightarrow \quad min$$

where λ_k are the complex eigenvalues of $\left(\lambda^2 M + \lambda D + K\right) x = 0$

- we will use the following criterion:

$$\operatorname{tr} X \quad \rightarrow \quad min$$

where X is the solution

$$AX + XA^T = -Z.$$

A is $2n \times 2n$ matrix obtained from M, D, K; Z has structure.

For this optimization problem one needs optimization criteria

- Spectral abscissa criterion

$$\max_k \operatorname{Re}(\lambda_k) \quad \rightarrow \quad min$$

where λ_k are the complex eigenvalues of $\left(\lambda^2 M + \lambda D + K\right) x = 0$

- we will use the following criterion:

$$\operatorname{tr} X \quad \rightarrow \quad min$$

where X is the solution

$$AX + XA^T = -Z.$$

A is $2n \times 2n$ matrix obtained from M, D, K; Z has structure.

For this optimization problem one needs optimization criteria

- Spectral abscissa criterion

$$\max_k \operatorname{Re}(\lambda_k) \quad \rightarrow \quad min$$

where λ_k are the complex eigenvalues of $\left(\lambda^2 M + \lambda D + K\right) x = 0$

- we will use the following criterion:

$$\operatorname{tr} X \quad \rightarrow \quad min$$

where X is the solution

$$AX + XA^T = -Z.$$

A is $2n \times 2n$ matrix obtained from M, D, K; Z has structure.

Efficient overall algorithm for dampers' optimization is still in want !

For the moment we assume: given dampers' positions !

We optimize dampers viscosities $v_1, \ldots, v_r$

$$A(v)X(v) + X(v)A(v)^T = -Z \qquad v = (v_1, \ldots, v_r)$$

Z depends on dominant frequencies.

Standard approach:

- Pick (define) dampers position
- Solve Lyapunov equation (Bartels-Stewart or Hammerling (Schur) algorithm)
- using obtained orthogonal basis to calculate Gradient and Hessian
- Newton minimization process

K. Brabender 1996, flops $= n_{iter} 30 \left(1 + \frac{3r+r^2}{2}\right) m^3 + \mathcal{O}(m^3)$, $m = 2n$.

Efficient overall algorithm for dampers' optimization is still in want !

For the moment we assume: given dampers' positions !

We optimize dampers viscosities $v_1, \ldots, v_r$

$$A(v)X(v) + X(v)A(v)^T = -Z \qquad v = (v_1, \ldots, v_r)$$

Z depends on dominant frequencies.

Standard approach:

- Pick (define) dampers position
- Solve Lyapunov equation (Bartels-Stewart or Hammerling (Schur) algorithm)
- using obtained orthogonal basis to calculate Gradient and Hessian
- Newton minimization process

K. Brabender 1996, flops $= n_{iter} 30 \left(1 + \frac{3r+r^2}{2}\right) m^3 + \mathcal{O}(m^3)$, $m = 2n$.

Efficient overall algorithm for dampers' optimization is still in want !

For the moment we assume: given dampers' positions !

We optimize dampers viscosities $v_1, \ldots, v_r$

$$A(v)X(v) + X(v)A(v)^T = -Z \qquad v = (v_1, \ldots, v_r)$$

Z depends on dominant frequencies.

Standard approach:

- Pick (define) dampers position
- Solve Lyapunov equation (Bartels-Stewart or Hammerling (Schur) algorithm)
- using obtained orthogonal basis to calculate Gradient and Hessian
- Newton minimization process

K. Brabender 1996, flops $= n_{iter} 30 \left(1 + \frac{3r+r^2}{2}\right) m^3 + \mathcal{O}(m^3)$, $m = 2n$.

Efficient overall algorithm for dampers' optimization is still in want !

For the moment we assume: given dampers' positions !

We optimize dampers viscosities $v_1, \ldots, v_r$

$$A(v)X(v) + X(v)A(v)^T = -Z \qquad v = (v_1, \ldots, v_r)$$

Z depends on dominant frequencies.

Standard approach:

- Pick (define) dampers position
- Solve Lyapunov equation (Bartels-Stewart or Hammerling (Schur) algorithm)
- using obtained orthogonal basis to calculate Gradient and Hessian
- Newton minimization process

K. Brabender 1996, flops $= n_{iter} 30 \left(1 + \frac{3r+r^2}{2}\right) m^3 + \mathcal{O}(m^3)$, $m = 2n$.

Efficient overall algorithm for dampers' optimization is still in want !

For the moment we assume: given dampers' positions !

We optimize dampers viscosities $v_1, \ldots, v_r$

$$A(v)X(v) + X(v)A(v)^T = -Z \qquad v = (v_1, \ldots, v_r)$$

Z depends on dominant frequencies.

Standard approach:

- Pick (define) dampers position
- Solve Lyapunov equation (Bartels-Stewart or Hammerling (Schur) algorithm)
- using obtained orthogonal basis to calculate Gradient and Hessian
- Newton minimization process

K. Brabender 1996, $\text{flops} = n_{iter} 30 \left(1 + \frac{3r+r^2}{2}\right) m^3 + \mathcal{O}(m^3)$, $m = 2n$.

Efficient overall algorithm for dampers' optimization is still in want !

For the moment we assume: given dampers' positions !

We optimize dampers viscosities $v_1, \ldots, v_r$

$$A(v)X(v) + X(v)A(v)^T = -Z \qquad v = (v_1, \ldots, v_r)$$

Z depends on dominant frequencies.

Standard approach:

- Pick (define) dampers position
- Solve Lyapunov equation (Bartels-Stewart or Hammerling (Schur) algorithm)
- using obtained orthogonal basis to calculate Gradient and Hessian
- Newton minimization process

K. Brabender 1996, $\text{flops} = n_{iter} 30 \left(1 + \frac{3r+r^2}{2}\right) m^3 + \mathcal{O}(m^3)$, $m = 2n$.

Efficient overall algorithm for dampers' optimization is still in want !

For the moment we assume: given dampers' positions !

We optimize dampers viscosities $v_1, \ldots, v_r$

$$A(v)X(v) + X(v)A(v)^T = -Z \qquad v = (v_1, \ldots, v_r)$$

Z depends on dominant frequencies.

Standard approach:

- Pick (define) dampers position
- Solve Lyapunov equation (Bartels-Stewart or Hammerling (Schur) algorithm)
- using obtained orthogonal basis to calculate Gradient and Hessian
- Newton minimization process

K. Brabender 1996, $\text{flops} = n_{iter} 30 \left(1 + \frac{3r+r^2}{2}\right) m^3 + \mathcal{O}(m^3)$, $m = 2n$.

Efficient overall algorithm for dampers' optimization is still in want !

For the moment we assume: given dampers' positions !

We optimize dampers viscosities $v_1, \ldots, v_r$

$$A(v)X(v) + X(v)A(v)^T = -Z \qquad v = (v_1, \ldots, v_r)$$

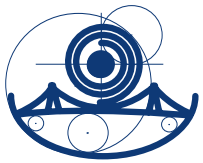
Z depends on dominant frequencies.

Standard approach:

- Pick (define) dampers position
- Solve Lyapunov equation (Bartels-Stewart or Hammerling (Schur) algorithm)
- using obtained orthogonal basis to calculate Gradient and Hessian
- Newton minimization process

K. Brabender 1996, flops $= n_{iter} 30 \left(1 + \frac{3r+r^2}{2}\right) m^3 + \mathcal{O}(m^3)$, $m = 2n$.

Main drawback of direct methods



All direct methods for trace calculate whole solution - to expensive.

For Example:

$$n = 600, \qquad r = 2, \qquad \text{2 similar dampers, fixed positions}$$

with a proper starting point v_0; viscosity optimization

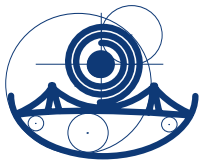
$$\text{Hammarling algorithm} \sim 3 \cdot 10^{12} \quad \text{flops} = 3 \quad \text{teraflops.}$$
$$\sim 20 \text{ min (2 GHz, 2 GB)}$$

Positions' optimization:

discrete optimization over 180000 positions **6.8 years**

It does not look efficient at all!

Main drawback of direct methods



All direct methods for trace calculate whole solution - to expensive.
For Example:

$$n = 600, \qquad r = 2, \qquad \text{2 similar dampers, fixed positions}$$

with a proper starting point v_0; viscosity optimization

Hammarling algorithm $\sim 3 \cdot 10^{12}$ flops $= 3$ teraflops.

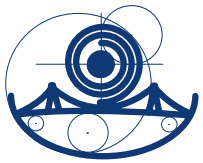
~ 20 min (2 GHz, 2 GB)

Positions' optimization:

discrete optimization over 180000 positions **6.8 years**

It does not look efficient at all!

Main drawback of direct methods



All direct methods for trace calculate whole solution - to expensive.
For Example:

$$n = 600, \qquad r = 2, \qquad \text{2 similar dampers, fixed positions}$$

with a proper starting point v_0; viscosity optimization

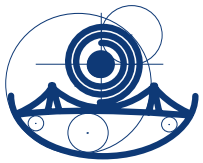
$$\text{Hammarling algorithm} \sim 3 \cdot 10^{12} \quad \text{flops} = 3 \quad \text{teraflops.}$$
$$\sim 20 \text{ min (2 GHz, 2 GB)}$$

Positions' optimization:

discrete optimization over 180000 positions **6.8 years**

It does not look efficient at all!

Main drawback of direct methods



All direct methods for trace calculate whole solution - to expensive.
For Example:

$$n = 600, \qquad r = 2, \qquad \text{2 similar dampers, fixed positions}$$

with a proper starting point v_0; viscosity optimization

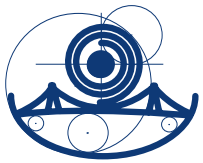
$$\text{Hammarling algorithm} \sim 3 \cdot 10^{12} \quad \text{flops} = 3 \quad \text{teraflops.}$$
$$\sim 20 \text{ min (2 GHz, 2 GB)}$$

Positions' optimization:

discrete optimization over 180000 positions **6.8 years**

It does not look efficient at all!

Main drawback of direct methods



All direct methods for trace calculate whole solution - to expensive.
For Example:

$$n = 600, \qquad r = 2, \qquad \text{2 similar dampers, fixed positions}$$

with a proper starting point v_0; viscosity optimization

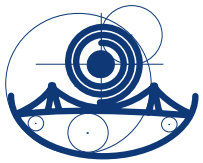
$$\text{Hammarling algorithm} \sim 3 \cdot 10^{12} \quad \text{flops} = 3 \quad \text{teraflops.}$$
$$\sim 20 \text{ min (2 GHz, 2 GB)}$$

Positions' optimization:

discrete optimization over 180000 positions **6.8 years**

It does not look efficient at all!

Main drawback of direct methods



All direct methods for trace calculate whole solution - to expensive.
For Example:

$$n = 600, \qquad r = 2, \qquad \text{2 similar dampers, fixed positions}$$

with a proper starting point v_0; viscosity optimization

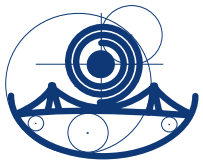
$$\text{Hammarling algorithm} \sim 3 \cdot 10^{12} \quad \text{flops} = 3 \quad \text{teraflops.}$$
$$\sim 20 \text{ min (2 GHz, 2 GB)}$$

Positions' optimization:

discrete optimization over 180000 positions 6.8 years

It does not look efficient at all!

Main drawback of direct methods



All direct methods for trace calculate whole solution - to expensive.
For Example:

$$n = 600, \qquad r = 2, \qquad \text{2 similar dampers, fixed positions}$$

with a proper starting point v_0; viscosity optimization

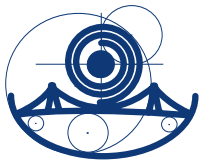
$$\text{Hammarling algorithm} \sim 3 \cdot 10^{12} \quad \text{flops} = 3 \quad \text{teraflops.}$$
$$\sim 20 \text{ min (2 GHz, 2 GB)}$$

Positions' optimization:

discrete optimization over 180000 positions **6.8 years**

It does not look efficient at all!

Experience



We have notices:

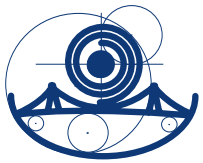
- "Bad positions" can not be significantly improved by viscosities
- For detecting "bad positions" (better a good), we do not need 16 or even 8 exact digits.

♢ We will try to find a small number of "good positions".

Idea:

♢ Find the optimal damping (optimal viscosity) for each position, for 1 damper; with just 1 dominant frequency.

Experience



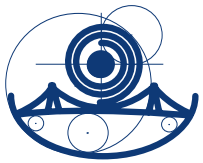
We have notices:

- "Bad positions" can not be significantly improved by viscosities
- For detecting "bad positions" (better a good), we do not need 16 or even 8 exact digits.
- ♢ We will try to find a small number of "good positions".

Idea:

- ♢ Find the optimal damping (optimal viscosity) for each position, for 1 damper; with just 1 dominant frequency.

Experience



We have notices:

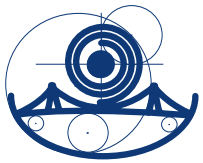
- “Bad positions” can not be significantly improved by viscosities
- For detecting “bad positions” (better a good), we do not need 16 or even 8 exact digits.

♢ We will try to find a small number of “good positions”.

Idea:

♢ Find the optimal damping (optimal viscosity) for each position, for 1 damper; with just 1 dominant frequency.

Experience



We have notices:

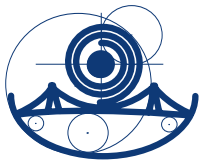
- "Bad positions" can not be significantly improved by viscosities
- For detecting "bad positions" (better a good), we do not need 16 or even 8 exact digits.

♢ We will try to find a small number of "good positions".

Idea:

♢ Find the optimal damping (optimal viscosity) for each position, for 1 damper; with just 1 dominant frequency.

Experience



We have notices:

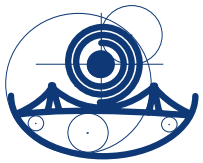
- "Bad positions" can not be significantly improved by viscosities
- For detecting "bad positions" (better a good), we do not need 16 or even 8 exact digits.

♢ We will try to find a small number of "good positions".

Idea:

♢ Find the optimal damping (optimal viscosity) for each position, for 1 damper; with just 1 dominant frequency.

Experience



We have notices:

- "Bad positions" can not be significantly improved by viscosities
- For detecting "bad positions" (better a good), we do not need 16 or even 8 exact digits.

♢ We will try to find a small number of "good positions".

Idea:

♢ Find the optimal damping (optimal viscosity) for each position, for 1 damper; with just 1 dominant frequency.

Idea.

Let $\Phi^T K \Phi = \Omega^2$, $\Phi^T M \Phi = I$, for $i = 1, \ldots, n$, compute minimal traces $tr_{opt} = \mathrm{tr}(X_k(i))$, where

$$A(i)X_k(i) + X_k(i)A(i)^T = Z_k\,, Z_k = \widehat{Z}_k \oplus \widehat{Z}_k\,, (\widehat{Z}_k)_{jk} = \delta_{j,k}\,,$$

$k = 1, \ldots, s$.

$$A(i) = A_0 - v_i \begin{bmatrix} 0 & 0 \\ 0 & c_i c_i^T \end{bmatrix}, A_0 = \begin{bmatrix} 0 & \Omega \\ -\Omega & -\alpha\Omega \end{bmatrix}, \quad c_i = \Phi^T e_i\,,$$

$i = 1, \ldots, n$.
For each $k = 1, \ldots, s$ we get function $g_k : \{1, \ldots, n\} \to \mathbb{R}$,

$$g_k(i) = \mathrm{tr}(X_k(i))\,,$$

The possible best positions are local minimums of $g_k(i)$ for $i = 1, \ldots, n$.

Idea.

Let $\Phi^T K \Phi = \Omega^2$, $\Phi^T M \Phi = I$, for $i = 1, \ldots, n$, compute minimal traces $tr_{opt} = \mathrm{tr}(X_k(i))$, where

$$A(i)X_k(i) + X_k(i)A(i)^T = Z_k\,, Z_k = \widehat{Z}_k \oplus \widehat{Z}_k\,, (\widehat{Z}_k)_{jk} = \delta_{j,k}\,,$$

$k = 1, \ldots, s$.

$$A(i) = A_0 - v_i \begin{bmatrix} 0 & 0 \\ 0 & c_i c_i^T \end{bmatrix}, A_0 = \begin{bmatrix} 0 & \Omega \\ -\Omega & -\alpha\Omega \end{bmatrix}, \quad c_i = \Phi^T e_i\,,$$

$i = 1, \ldots, n$.
For each $k = 1, \ldots, s$ we get function $g_k : \{1, \ldots, n\} \to \mathbb{R}$,

$$g_k(i) = \mathrm{tr}(X_k(i))\,,$$

The possible best positions are local minimums of $g_k(i)$ for $i = 1, \ldots, n$.

Idea.

Let $\Phi^T K \Phi = \Omega^2$, $\Phi^T M \Phi = I$, for $i = 1, \ldots, n$, compute minimal traces $tr_{opt} = \mathrm{tr}(X_k(i))$, where

$$A(i)X_k(i) + X_k(i)A(i)^T = Z_k\,, Z_k = \widehat{Z}_k \oplus \widehat{Z}_k\,, (\widehat{Z}_k)_{jk} = \delta_{j,k}\,,$$

$k = 1, \ldots, s.$

$$A(i) = A_0 - v_i \begin{bmatrix} 0 & 0 \\ 0 & c_i c_i^T \end{bmatrix}, A_0 = \begin{bmatrix} 0 & \Omega \\ -\Omega & -\alpha\Omega \end{bmatrix}, \quad c_i = \Phi^T e_i\,,$$

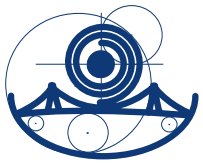
$i = 1, \ldots, n\,.$

For each $k = 1, \ldots, s$ we get function $g_k : \{1, \ldots, n\} \to \mathbb{R}$,

$$g_k(i) = \mathrm{tr}(X_k(i))\,,$$

The possible best positions are local minimums of $g_k(i)$ for $i = 1, \ldots, n$.

Idea.



Let $\Phi^T K \Phi = \Omega^2$, $\Phi^T M \Phi = I$, for $i = 1, \ldots, n$, compute minimal traces $tr_{opt} = \mathrm{tr}(X_k(i))$, where

$$A(i)X_k(i) + X_k(i)A(i)^T = Z_k \,, Z_k = \widehat{Z}_k \oplus \widehat{Z}_k \,, (\widehat{Z}_k)_{jk} = \delta_{j,k} \,,$$

$k = 1, \ldots, s$.

$$A(i) = A_0 - v_i \begin{bmatrix} 0 & 0 \\ 0 & c_i c_i^T \end{bmatrix} , A_0 = \begin{bmatrix} 0 & \Omega \\ -\Omega & -\alpha\Omega \end{bmatrix} , \quad c_i = \Phi^T e_i \,,$$

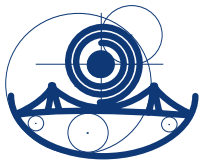
$i = 1, \ldots, n$.

For each $k = 1, \ldots, s$ we get function $g_k : \{1, \ldots, n\} \to \mathbb{R}$,

$$g_k(i) = \mathrm{tr}(X_k(i)) \,,$$

The possible best positions are local minimums of $g_k(i)$ for $i = 1, \ldots, n$.

Idea.



Let $\Phi^T K \Phi = \Omega^2$, $\Phi^T M \Phi = I$, for $i = 1, \ldots, n$, compute minimal traces $tr_{opt} = \mathrm{tr}(X_k(i))$, where

$$A(i)X_k(i) + X_k(i)A(i)^T = Z_k \,, Z_k = \widehat{Z}_k \oplus \widehat{Z}_k \,, (\widehat{Z}_k)_{jk} = \delta_{j,k} \,,$$

$k = 1, \ldots, s$.

$$A(i) = A_0 - v_i \begin{bmatrix} 0 & 0 \\ 0 & c_i c_i^T \end{bmatrix} , A_0 = \begin{bmatrix} 0 & \Omega \\ -\Omega & -\alpha\Omega \end{bmatrix} , \quad c_i = \Phi^T e_i \,,$$

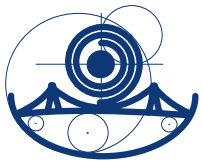
$i = 1, \ldots, n$.

For each $k = 1, \ldots, s$ we get function $g_k : \{1, \ldots, n\} \to \mathbb{R}$,

$$g_k(i) = \mathrm{tr}(X_k(i)) \,,$$

The possible best positions are local minimums of $g_k(i)$ for $i = 1, \ldots, n$.

Idea.



Let $\Phi^T K \Phi = \Omega^2$, $\Phi^T M \Phi = I$, for $i = 1, \ldots, n$, compute minimal traces $tr_{opt} = \mathrm{tr}(X_k(i))$, where

$$A(i)X_k(i) + X_k(i)A(i)^T = Z_k \,, Z_k = \widehat{Z}_k \oplus \widehat{Z}_k \,, (\widehat{Z}_k)_{jk} = \delta_{j,k} \,,$$

$k = 1, \ldots, s$.

$$A(i) = A_0 - v_i \begin{bmatrix} 0 & 0 \\ 0 & c_i c_i^T \end{bmatrix} , A_0 = \begin{bmatrix} 0 & \Omega \\ -\Omega & -\alpha\Omega \end{bmatrix} , \quad c_i = \Phi^T e_i \,,$$

$i = 1, \ldots, n$.

For each $k = 1, \ldots, s$ we get function $g_k : \{1, \ldots, n\} \to \mathbb{R}$,

$$g_k(i) = \mathrm{tr}(X_k(i)) \,,$$

The possible best positions are local minimums of $g_k(i)$ for $i = 1, \ldots, n$.

Idea.

Let $\Phi^T K \Phi = \Omega^2$, $\Phi^T M \Phi = I$, for $i = 1, \ldots, n$, compute minimal traces $tr_{opt} = \mathrm{tr}(X_k(i))$, where

$$A(i)X_k(i) + X_k(i)A(i)^T = Z_k \,, Z_k = \widehat{Z}_k \oplus \widehat{Z}_k \,, (\widehat{Z}_k)_{jk} = \delta_{j,k} \,,$$

$k = 1, \ldots, s$.

$$A(i) = A_0 - v_i \begin{bmatrix} 0 & 0 \\ 0 & c_i c_i^T \end{bmatrix} , A_0 = \begin{bmatrix} 0 & \Omega \\ -\Omega & -\alpha\Omega \end{bmatrix} , \quad c_i = \Phi^T e_i \,,$$

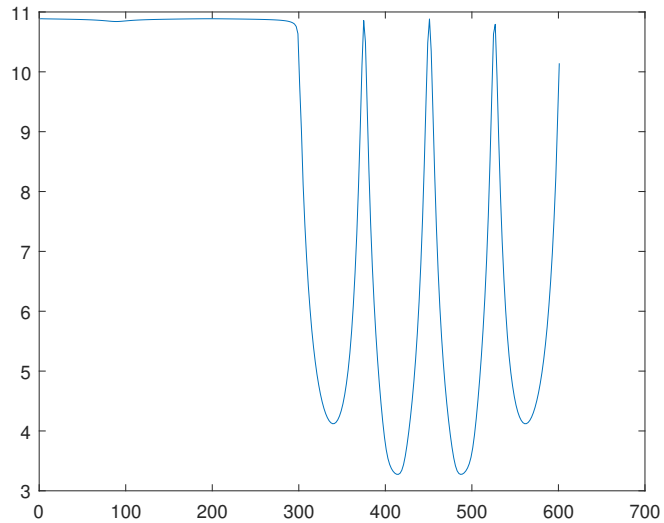
$i = 1, \ldots, n$.
For each $k = 1, \ldots, s$ we get function $g_k : \{1, \ldots, n\} \to \mathbb{R}$,

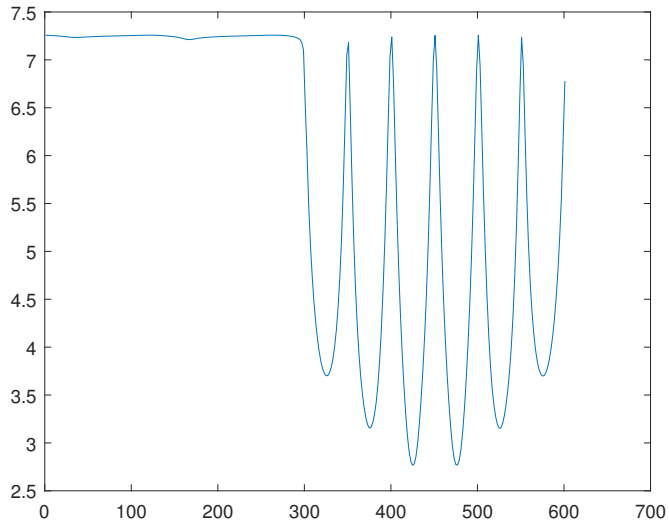
$$g_k(i) = \mathrm{tr}(X_k(i)) \,,$$

The possible best positions are local minimums of $g_k(i)$ for $i = 1, \ldots, n$.

Idea.

Consider system on figure, $n = 601$





Left figure is for the 5-th frequency ($s = 5$).

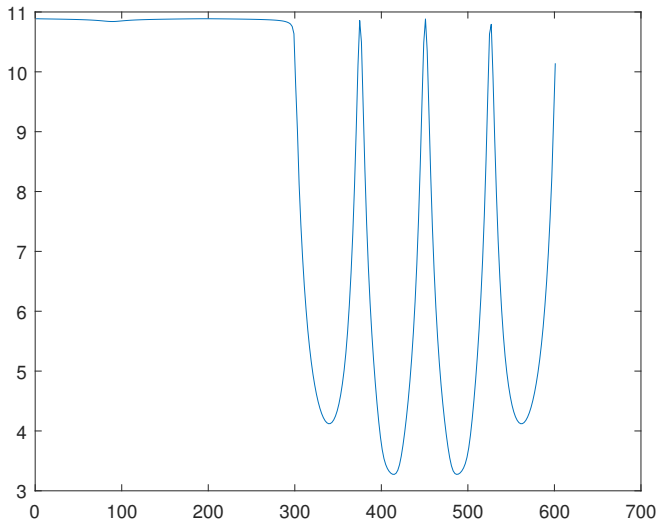
Right figure is for the 8-th frequency ($s = 8$).

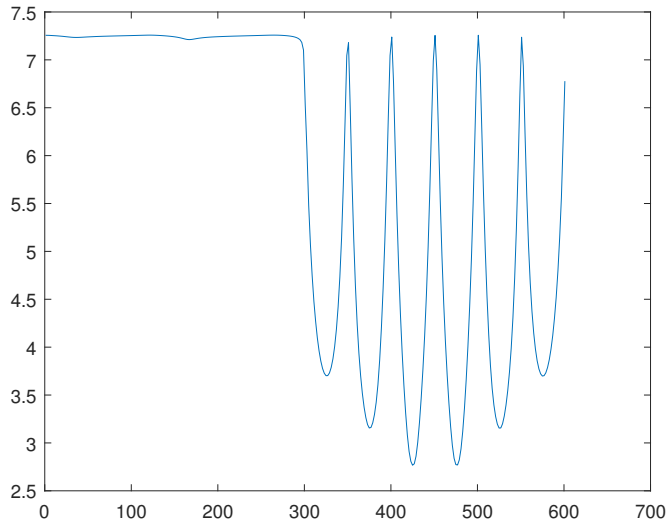
The candidate: $339(0.3688)$, $415(0.4285)$, $487(0.4324)$, $561(0.3698)$ (for $s = 5$)

The candidate: $326(0.3618)$, $375(0.4180)$, $425(0.4326)$, $477(0.4333)$, $526(0.4498)$, $575(0.3411)$ (for $s = 8$)

Idea.

Consider system on figure, $n = 601$





Left figure is for the 5-th frequency ($s = 5$).

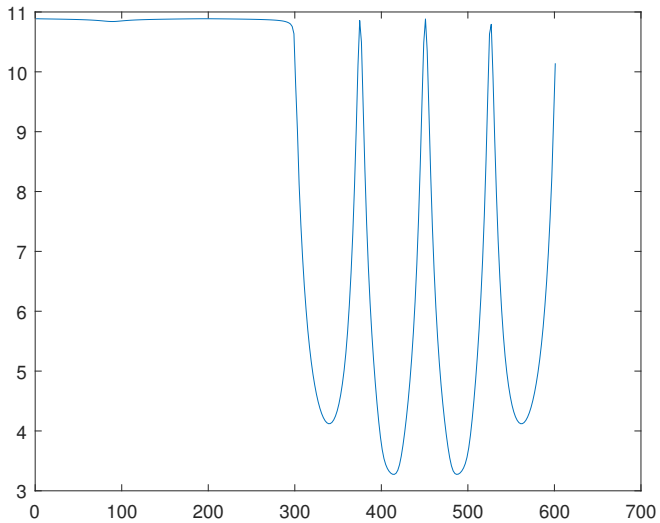
Right figure is for the 8-th frequency ($s = 8$).

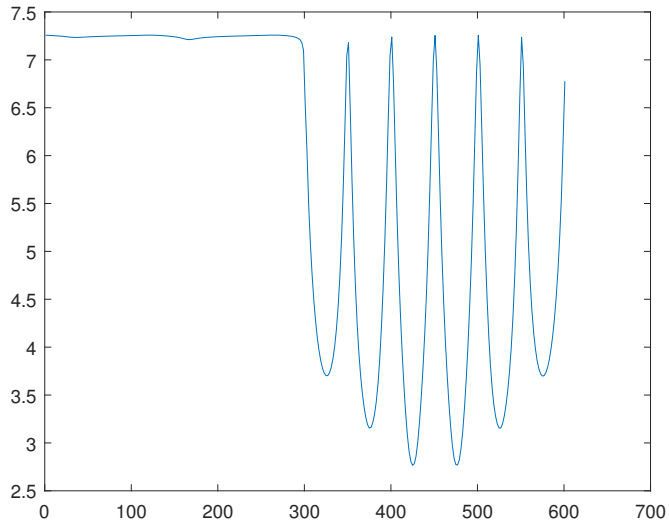
The candidate: $339(0.3688)$, $415(0.4285)$, $487(0.4324)$, $561(0.3698)$ (for $s = 5$)

The candidate: $326(0.3618)$, $375(0.4180)$, $425(0.4326)$, $477(0.4333)$, $526(0.4498)$, $575(0.3411)$ (for $s = 8$)

Idea.

Consider system on figure, $n = 601$





Left figure is for the 5-th frequency ($s = 5$).

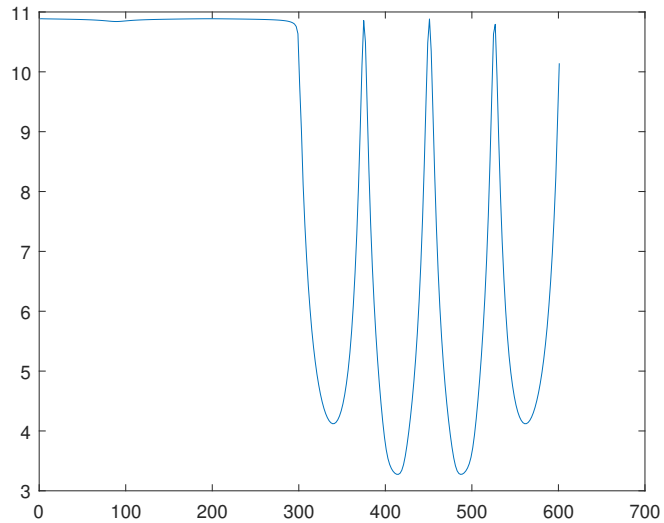
Right figure is for the 8-th frequency ($s = 8$).

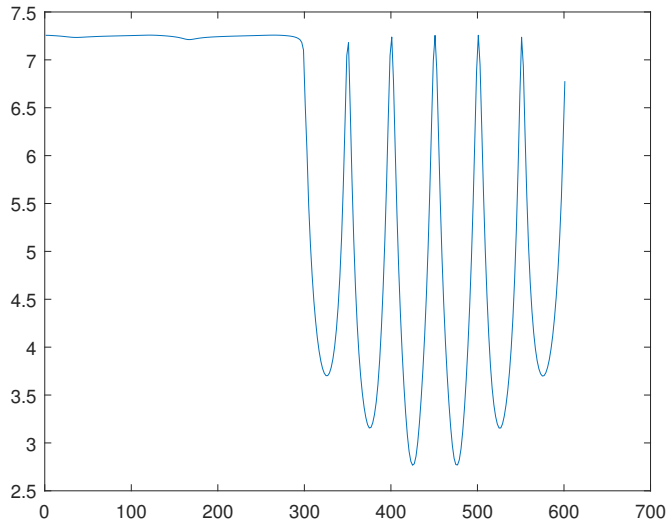
The candidate: $339(0.3688)$, $415(0.4285)$, $487(0.4324)$, $561(0.3698)$ (for $s = 5$)

The candidate: $326(0.3618)$, $375(0.4180)$, $425(0.4326)$, $477(0.4333)$, $526(0.4498)$, $575(0.3411)$ (for $s = 8$)

Idea.

Consider system on figure, $n = 601$





Left figure is for the 5-th frequency ($s = 5$).

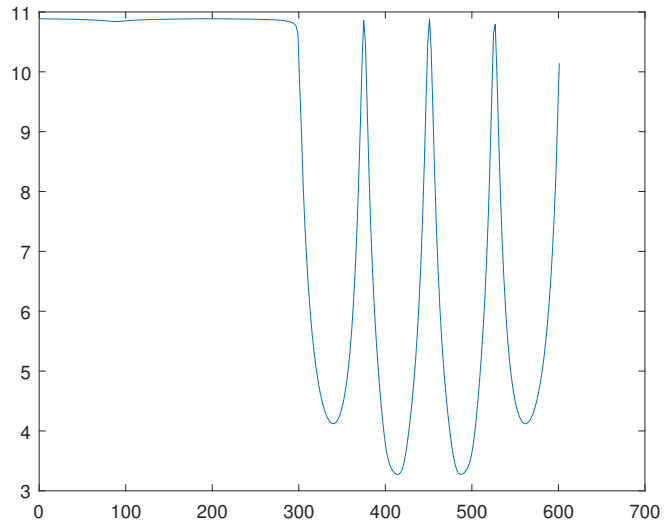
Right figure is for the 8-th frequency ($s = 8$).

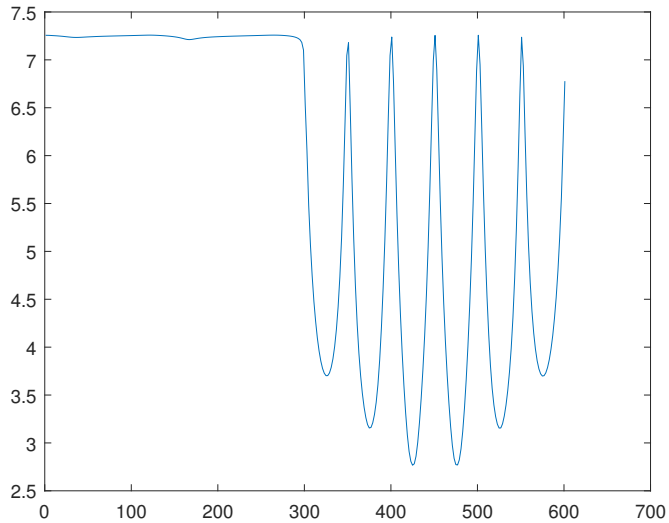
The candidate: $339(0.3688)$, $415(0.4285)$, $487(0.4324)$, $561(0.3698)$ (for $s = 5$)

The candidate: $326(0.3618)$, $375(0.4180)$, $425(0.4326)$, $477(0.4333)$, $526(0.4498)$, $575(0.3411)$ (for $s = 8$)

Idea.

Consider system on figure, $n = 601$





Left figure is for the 5-th frequency ($s = 5$).

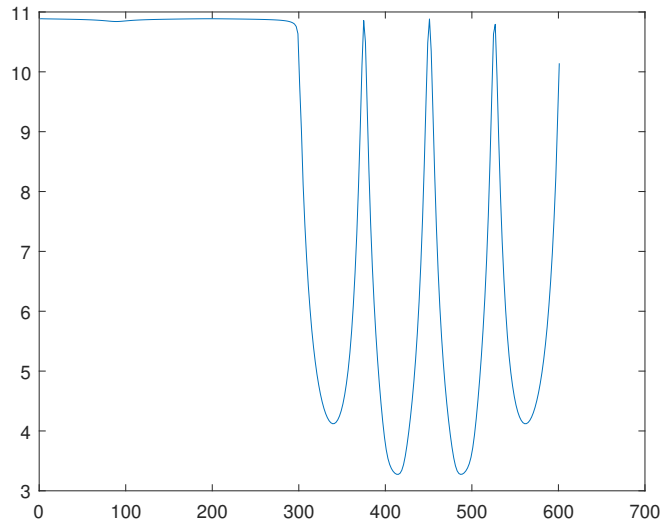
Right figure is for the 8-th frequency ($s = 8$).

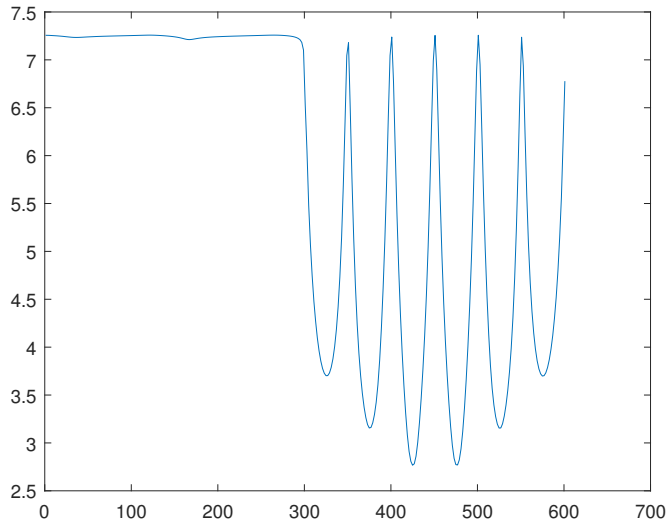
The candidate: $339(0.3688)$, $415(0.4285)$, $487(0.4324)$, $561(0.3698)$ (for $s = 5$)

The candidate: $326(0.3618)$, $375(0.4180)$, $425(0.4326)$, $477(0.4333)$, $526(0.4498)$, $575(0.3411)$ (for $s = 8$)

Idea.

Consider system on figure, $n = 601$





Left figure is for the 5-th frequency ($s = 5$).

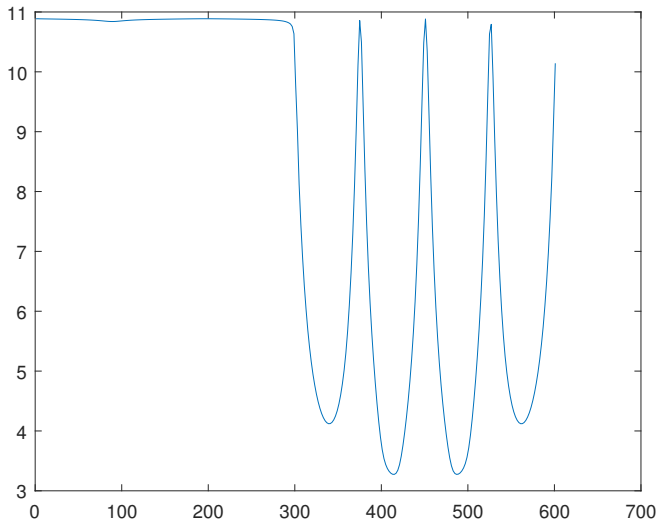
Right figure is for the 8-th frequency ($s = 8$).

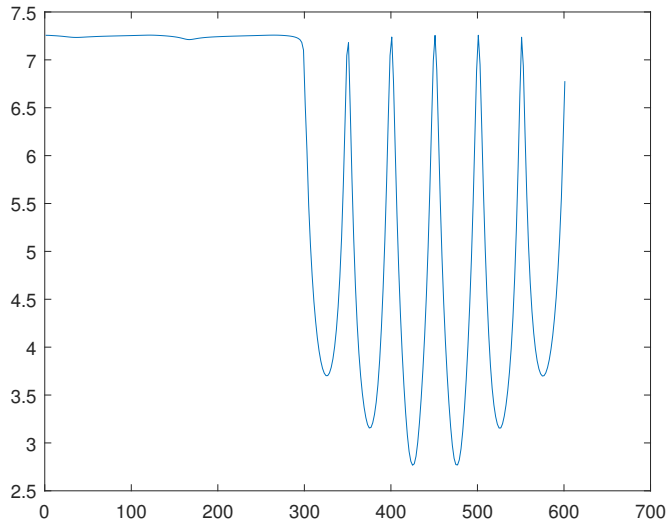
The candidate: $339(0.3688)$, $415(0.4285)$, $487(0.4324)$, $561(0.3698)$ (for $s = 5$)

The candidate: $326(0.3618)$, $375(0.4180)$, $425(0.4326)$, $477(0.4333)$, $526(0.4498)$, $575(0.3411)$ (for $s = 8$)

Idea.

Consider system on figure, $n = 601$





Left figure is for the 5-th frequency ($s = 5$).

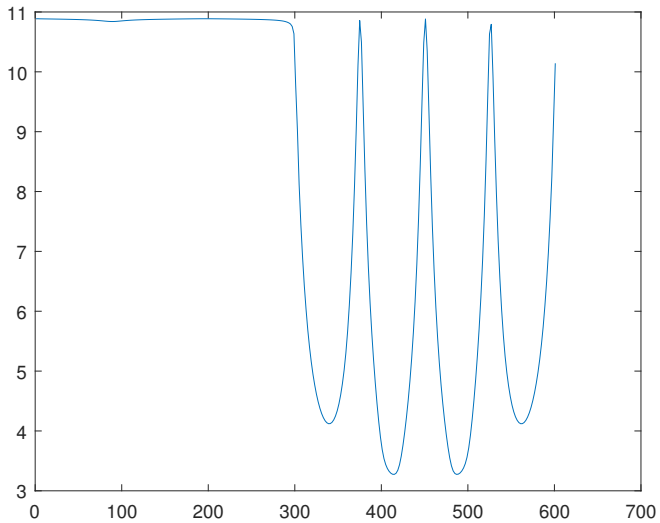
Right figure is for the 8-th frequency ($s = 8$).

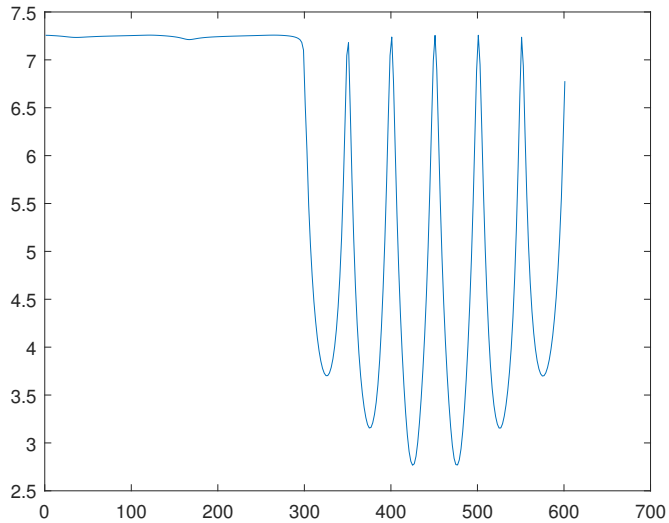
The candidate: $339(0.3688)$, $415(0.4285)$, $487(0.4324)$, $561(0.3698)$ (for $s = 5$)

The candidate: $326(0.3618)$, $375(0.4180)$, $425(0.4326)$, $477(0.4333)$, $526(0.4498)$, $575(0.3411)$ (for $s = 8$)

Idea.

Consider system on figure, $n = 601$



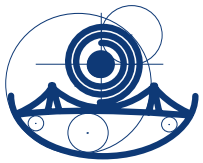


Left figure is for the 5-th frequency ($s = 5$).

Right figure is for the 8-th frequency ($s = 8$).

The candidate: $339(0.3688)$, $415(0.4285)$, $487(0.4324)$, $561(0.3698)$ (for $s = 5$)

The candidate: $326(0.3618)$, $375(0.4180)$, $425(0.4326)$, $477(0.4333)$, $526(0.4498)$, $575(0.3411)$ (for $s = 8$)



Significant improvement:

For $s = 1, \ldots, 40$ instead 180000 ($n = 601$) possibilities, we get 9000 ($n_0 = 176$)!

Still, it is very demanding job! 9000 2D optimizations.

We need: a efficient solver (faster then n^2) and more efficient optimization

Case without internal damping: $C_u = 0$, there exist formula: Veselić (1990):

$$Tr(ZX) = const + \frac{a}{v} + bv, \quad a, b > 0\,.$$

total cost $\mathcal{O}(n^2)$, optimal v for free, $v_{opt} = \sqrt{a/b}$.

Unfortunately, for $C_u \neq 0$, the above formula is very unstable.

Idea: derive a similar formula (even complicated) for $C_u \neq 0$!

We failed! It still open?

But, we have an asymptotic analysis, which allows a rational approximation!

Significant improvement:

For $s = 1, \ldots, 40$ instead 180000 ($n = 601$) possibilities, we get 9000 ($n_0 = 176$)!

Still, it is very demanding job! 9000 2D optimizations.

We need: a efficient solver (faster then n^2) and more efficient optimization

Case without internal damping: $C_u = 0$, there exist formula: Veselić (1990):

$$Tr(ZX) = const + \frac{a}{v} + bv, \quad a, b > 0\,.$$

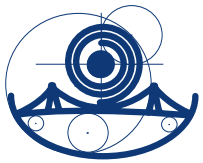
total cost $\mathcal{O}(n^2)$, optimal v for free, $v_{opt} = \sqrt{a/b}$.

Unfortunately, for $C_u \neq 0$, the above formula is very unstable.

Idea: derive a similar formula (even complicated) for $C_u \neq 0$!

We failed! It still open?

But, we have an asymptotic analysis, which allows a rational approximation!



Significant improvement:

For $s = 1, \ldots, 40$ instead 180000 ($n = 601$) possibilities, we get 9000 ($n_0 = 176$)!

Still, it is very demanding job! 9000 2D optimizations.

We need: a efficient solver (faster then n^2) and more efficient optimization

Case without internal damping: $C_u = 0$, there exist formula: Veselić (1990):

$$Tr(ZX) = const + \frac{a}{v} + bv, \quad a, b > 0\,.$$

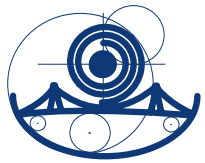
total cost $\mathcal{O}(n^2)$, optimal v for free, $v_{opt} = \sqrt{a/b}$.

Unfortunately, for $C_u \neq 0$, the above formula is very unstable.

Idea: derive a similar formula (even complicated) for $C_u \neq 0$!

We failed! It still open?

But, we have an asymptotic analysis, which allows a rational approximation!



Significant improvement:

For $s = 1, \ldots, 40$ instead 180000 ($n = 601$) possibilities, we get 9000 ($n_0 = 176$)!

Still, it is very demanding job! 9000 2D optimizations.

We need: a efficient solver (faster then n^2) and more efficient optimization

Case without internal damping: $C_u = 0$, there exist formula: Veselić (1990):

$$Tr(ZX) = const + \frac{a}{v} + bv, \quad a, b > 0\,.$$

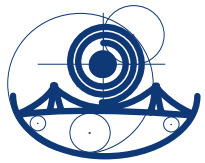
total cost $\mathcal{O}(n^2)$, optimal v for free, $v_{opt} = \sqrt{a/b}$.

Unfortunately, for $C_u \neq 0$, the above formula is very unstable.

Idea: derive a similar formula (even complicated) for $C_u \neq 0$!

We failed! It still open?

But, we have an asymptotic analysis, which allows a rational approximation!



Significant improvement:

For $s = 1, \ldots, 40$ instead 180000 ($n = 601$) possibilities, we get 9000 ($n_0 = 176$)!

Still, it is very demanding job! 9000 2D optimizations.

We need: a efficient solver (faster then n^2) and more efficient optimization

Case without internal damping: $C_u = 0$, there exist formula: Veselić (1990):

$$Tr(ZX) = const + \frac{a}{v} + bv, \quad a, b > 0\,.$$

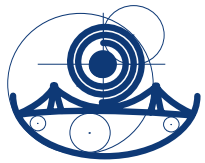
total cost $\mathcal{O}(n^2)$, optimal v for free, $v_{opt} = \sqrt{a/b}$.

Unfortunately, for $C_u \neq 0$, the above formula is very unstable.

Idea: derive a similar formula (even complicated) for $C_u \neq 0$!

We failed! It still open?

But, we have an asymptotic analysis, which allows a rational approximation!



Significant improvement:

For $s = 1, \ldots, 40$ instead 180000 ($n = 601$) possibilities, we get 9000 ($n_0 = 176$)!

Still, it is very demanding job! 9000 2D optimizations.

We need: a efficient solver (faster then n^2) and more efficient optimization

Case without internal damping: $C_u = 0$, there exist formula: Veselić (1990):

$$Tr(ZX) = const + \frac{a}{v} + bv, \quad a, b > 0\,.$$

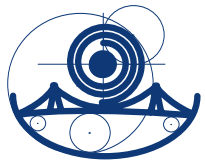
total cost $\mathcal{O}(n^2)$, optimal v for free, $v_{opt} = \sqrt{a/b}$.

Unfortunately, for $C_u \neq 0$, the above formula is very unstable.

Idea: derive a similar formula (even complicated) for $C_u \neq 0$!

We failed! It still open?

But, we have an asymptotic analysis, which allows a rational approximation!



Significant improvement:

For $s = 1, \ldots, 40$ instead 180000 ($n = 601$) possibilities, we get 9000 ($n_0 = 176$)!

Still, it is very demanding job! 9000 2D optimizations.

We need: a efficient solver (faster then n^2) and more efficient optimization

Case without internal damping: $C_u = 0$, there exist formula: Veselić (1990):

$$Tr(ZX) = const + \frac{a}{v} + bv, \quad a, b > 0\,.$$

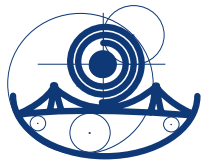
total cost $\mathcal{O}(n^2)$, optimal v for free, $v_{opt} = \sqrt{a/b}$.

Unfortunately, for $C_u \neq 0$, the above formula is very unstable.

Idea: derive a similar formula (even complicated) for $C_u \neq 0$!

We failed! It still open?

But, we have an asymptotic analysis, which allows a rational approximation!



Significant improvement:

For $s = 1, \ldots, 40$ instead 180000 ($n = 601$) possibilities, we get 9000 ($n_0 = 176$)!

Still, it is very demanding job! 9000 2D optimizations.

We need: a efficient solver (faster then n^2) and more efficient optimization

Case without internal damping: $C_u = 0$, there exist formula: Veselić (1990):

$$Tr(ZX) = const + \frac{a}{v} + bv, \quad a, b > 0\,.$$

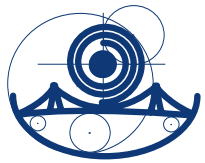
total cost $\mathcal{O}(n^2)$, optimal v for free, $v_{opt} = \sqrt{a/b}$.

Unfortunately, for $C_u \neq 0$, the above formula is very unstable.

Idea: derive a similar formula (even complicated) for $C_u \neq 0$!

We failed! It still open?

But, we have an asymptotic analysis, which allows a rational approximation!



Significant improvement:

For $s = 1, \ldots, 40$ instead 180000 ($n = 601$) possibilities, we get 9000 ($n_0 = 176$)!

Still, it is very demanding job! 9000 2D optimizations.

We need: a efficient solver (faster then n^2) and more efficient optimization

Case without internal damping: $C_u = 0$, there exist formula: Veselić (1990):

$$Tr(ZX) = const + \frac{a}{v} + bv, \quad a, b > 0\,.$$

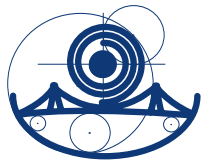
total cost $\mathcal{O}(n^2)$, optimal v for free, $v_{opt} = \sqrt{a/b}$.

Unfortunately, for $C_u \neq 0$, the above formula is very unstable.

Idea: derive a similar formula (even complicated) for $C_u \neq 0$!

We failed! It still open?

But, we have an asymptotic analysis, which allows a rational approximation!



Significant improvement:

For $s = 1, \ldots, 40$ instead 180000 ($n = 601$) possibilities, we get 9000 ($n_0 = 176$)!

Still, it is very demanding job! 9000 2D optimizations.

We need: a efficient solver (faster then n^2) and more efficient optimization

Case without internal damping: $C_u = 0$, there exist formula: Veselić (1990):

$$Tr(ZX) = const + \frac{a}{v} + bv, \quad a, b > 0\,.$$

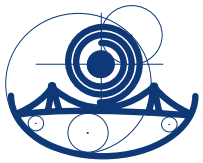
total cost $\mathcal{O}(n^2)$, optimal v for free, $v_{opt} = \sqrt{a/b}$.

Unfortunately, for $C_u \neq 0$, the above formula is very unstable.

Idea: derive a similar formula (even complicated) for $C_u \neq 0$!

We failed! It still open?

But, we have an asymptotic analysis, which allows a rational approximation!



Idea: avoid 9000 2D optimizations!

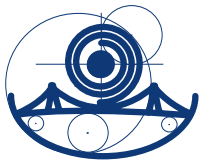
Idea: use 9000 obtained 1D optimal viscosities!

How to perform 1D optimization?

Asymptotics:

Thus, for $\mathrm{tr}(X(v))$ we have

$$\mathrm{tr}(X(v)) \approx \frac{av^2 + bv + c \cdot \mathrm{tr}(X(0))}{v + c}$$



Idea: avoid 9000 2D optimizations!

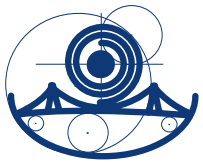
Idea: use 9000 obtained 1D optimal viscosities!

How to perform 1D optimization?

Asymptotics:

Thus, for $\mathrm{tr}(X(v))$ we have

$$\mathrm{tr}(X(v)) \approx \frac{av^2 + bv + c \cdot \mathrm{tr}(X(0))}{v + c}$$



Idea: avoid 9000 2D optimizations!

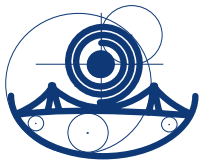
Idea: use 9000 obtained 1D optimal viscosities!

How to perform 1D optimization?

Asymptotics:

Thus, for $\mathrm{tr}(X(v))$ we have

$$\mathrm{tr}(X(v)) \approx \frac{av^2 + bv + c \cdot \mathrm{tr}(X(0))}{v + c}$$



Idea: avoid 9000 2D optimizations!

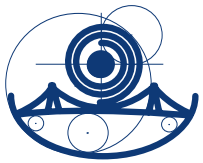
Idea: use 9000 obtained 1D optimal viscosities!

How to perform 1D optimization?

Asymptotics:

Thus, for $\mathrm{tr}(X(v))$ we have

$$\mathrm{tr}(X(v)) \approx \frac{av^2 + bv + c \cdot \mathrm{tr}(X(0))}{v + c}$$



Idea: avoid 9000 2D optimizations!

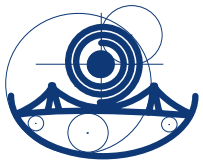
Idea: use 9000 obtained 1D optimal viscosities!

How to perform 1D optimization?

Asymptotics:

Thus, for $\mathrm{tr}(X(v))$ we have

$$\mathrm{tr}(X(v)) \approx \frac{av^2 + bv + c \cdot \mathrm{tr}(X(0))}{v + c}$$



Idea: avoid 9000 2D optimizations!

Idea: use 9000 obtained 1D optimal viscosities!

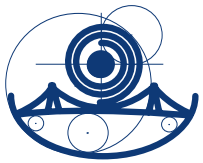
How to perform 1D optimization?

Asymptotics:

Thus, for $\mathrm{tr}(X(v))$ we have

$$\mathrm{tr}(X(v)) \approx \frac{av^2 + bv + c \cdot \mathrm{tr}(X(0))}{v + c}$$

Optimization process

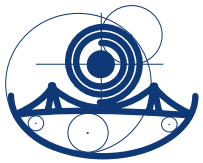


Now we can apply the optimization process (similar as parabolic minimization):
define rational functions

$$g(v) = \frac{av^2 + bv + c \cdot \mathrm{tr}(X(0))}{v + c},$$

a, b and c are determined by a simple interpolation through v_1, v_2, v_3.
The zero of $g'(v_4) = 0$ is the first approximation for the optimal v.
Replace $\{v_1, v_2, v_3\}$ by the new minimum v_4 and repeat the process until the selected tolerance level is attained.
We noticed: that 5-10 steps are enough for tolerance $\approx 10^{-4}$.
Still we need to solve $5 - 10$ Lyapunov equations, with a low tolerance $(\approx 10^{-4})$.

Optimization process



Now we can apply the optimization process (similar as parabolic minimization):

define rational functions

$$g(v) = \frac{av^2 + bv + c \cdot \mathrm{tr}(X(0))}{v + c},$$

a, b and c are determined by a simple interpolation through v_1, v_2, v_3.

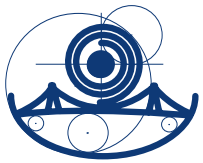
The zero of $g'(v_4) = 0$ is the first approximation for the optimal v.

Replace $\{v_1, v_2, v_3\}$ by the new minimum v_4 and repeat the process until the selected tolerance level is attained.

We noticed: that 5-10 steps are enough for tolerance $\approx 10^{-4}$.

Still we need to solve $5 - 10$ Lyapunov equations, with a low tolerance ($\approx 10^{-4}$).

Optimization process

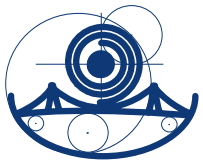


Now we can apply the optimization process (similar as parabolic minimization):
define rational functions

$$g(v) = \frac{av^2 + bv + c \cdot \mathrm{tr}(X(0))}{v + c},$$

a, b and c are determined by a simple interpolation through v_1, v_2, v_3.
The zero of $g'(v_4) = 0$ is the first approximation for the optimal v.
Replace $\{v_1, v_2, v_3\}$ by the new minimum v_4 and repeat the process until the selected tolerance level is attained.
We noticed: that 5-10 steps are enough for tolerance $\approx 10^{-4}$.
Still we need to solve $5 - 10$ Lyapunov equations, with a low tolerance $(\approx 10^{-4})$.

Optimization process

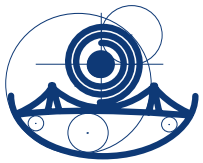


Now we can apply the optimization process (similar as parabolic minimization):
define rational functions

$$g(v) = \frac{av^2 + bv + c \cdot \mathrm{tr}(X(0))}{v + c},$$

a, b and c are determined by a simple interpolation through v_1, v_2, v_3.
The zero of $g'(v_4) = 0$ is the first approximation for the optimal v.
Replace $\{v_1, v_2, v_3\}$ by the new minimum v_4 and repeat the process until the selected tolerance level is attained.
We noticed: that 5-10 steps are enough for tolerance $\approx 10^{-4}$.
Still we need to solve $5 - 10$ Lyapunov equations, with a low tolerance $(\approx 10^{-4})$.

Optimization process

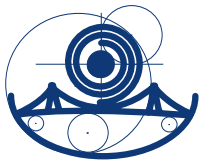


Now we can apply the optimization process (similar as parabolic minimization):
define rational functions

$$g(v) = \frac{av^2 + bv + c \cdot \mathrm{tr}(X(0))}{v + c},$$

a, b and c are determined by a simple interpolation through v_1, v_2, v_3.
The zero of $g'(v_4) = 0$ is the first approximation for the optimal v.
Replace $\{v_1, v_2, v_3\}$ by the new minimum v_4 and repeat the process until the selected tolerance level is attained.
We noticed: that 5-10 steps are enough for tolerance $\approx 10^{-4}$.
Still we need to solve $5 - 10$ Lyapunov equations, with a low tolerance $(\approx 10^{-4})$.

Optimization process

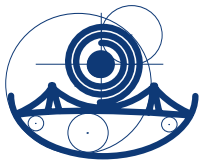


Now we can apply the optimization process (similar as parabolic minimization):
define rational functions

$$g(v) = \frac{av^2 + bv + c \cdot \mathrm{tr}(X(0))}{v + c},$$

a, b and c are determined by a simple interpolation through v_1, v_2, v_3.
The zero of $g'(v_4) = 0$ is the first approximation for the optimal v.
Replace $\{v_1, v_2, v_3\}$ by the new minimum v_4 and repeat the process until the selected tolerance level is attained.
We noticed: that 5-10 steps are enough for tolerance $\approx 10^{-4}$.
Still we need to solve $5 - 10$ Lyapunov equations, with a low tolerance $(\approx 10^{-4})$.

Optimization process

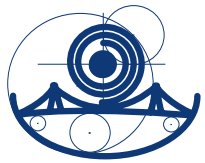


Now we can apply the optimization process (similar as parabolic minimization):
define rational functions

$$g(v) = \frac{av^2 + bv + c \cdot \mathrm{tr}(X(0))}{v + c},$$

a, b and c are determined by a simple interpolation through v_1, v_2, v_3.
The zero of $g'(v_4) = 0$ is the first approximation for the optimal v.
Replace $\{v_1, v_2, v_3\}$ by the new minimum v_4 and repeat the process until the selected tolerance level is attained.
We noticed: that 5-10 steps are enough for tolerance $\approx 10^{-4}$.
Still we need to solve $5 - 10$ Lyapunov equations, with a low tolerance $(\approx 10^{-4})$.

Optimization process

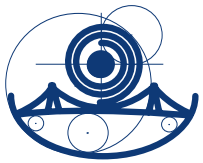


Now we can apply the optimization process (similar as parabolic minimization):
define rational functions

$$g(v) = \frac{av^2 + bv + c \cdot \mathrm{tr}(X(0))}{v + c},$$

a, b and c are determined by a simple interpolation through v_1, v_2, v_3.
The zero of $g'(v_4) = 0$ is the first approximation for the optimal v.
Replace $\{v_1, v_2, v_3\}$ by the new minimum v_4 and repeat the process until the selected tolerance level is attained.
We noticed: that 5-10 steps are enough for tolerance $\approx 10^{-4}$.
Still we need to solve $5 - 10$ Lyapunov equations, with a low tolerance ($\approx 10^{-4}$).

Error bound



For efficient optimization, we need an apriority error bound.

Unfortunately, up to date we did not find usefull apriority error bound (for any existing “fast solvers”)

Thus, we derive a new error bound.

Let

$$A^T X + X A = -Z,$$

be the Lyapunov equation, with

$$A = \begin{bmatrix} 0 & A_{12} \\ -A_{12} & -\Delta \end{bmatrix}, \quad Z = \begin{bmatrix} Z_1 & 0 \\ 0 & Z_2 \end{bmatrix},$$

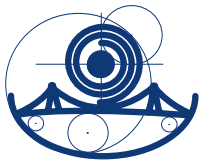
with A_{12} and $\Delta > 0$.

Let

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \text{ and } Y = \begin{bmatrix} Y_{11} & X_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix},$$

be solution and its approximation (obtained by some appropriate solver), resp.

Error bound



For efficient optimization, we need an apriority error bound.

Unfortunately, up to date we did not find usefull apriority error bound (for any existing “fast solvers”)

Thus, we derive a new error bound.

Let

$$A^T X + X A = -Z,$$

be the Lyapunov equation, with

$$A = \begin{bmatrix} 0 & A_{12} \\ -A_{12} & -\Delta \end{bmatrix}, \quad Z = \begin{bmatrix} Z_1 & 0 \\ 0 & Z_2 \end{bmatrix},$$

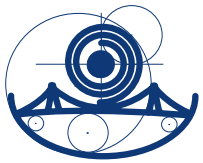
with A_{12} and $\Delta > 0$.

Let

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \text{ and } Y = \begin{bmatrix} Y_{11} & X_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix},$$

be solution and its approximation (obtained by some appropriate solver), resp.

Error bound



For efficient optimization, we need an apriority error bound.

Unfortunately, up to date we did not find usefull apriority error bound (for any existing “fast solvers”)

Thus, we derive a new error bound.

Let

$$A^T X + X A = -Z,$$

be the Lyapunov equation, with

$$A = \begin{bmatrix} 0 & A_{12} \\ -A_{12} & -\Delta \end{bmatrix}, \quad Z = \begin{bmatrix} Z_1 & 0 \\ 0 & Z_2 \end{bmatrix},$$

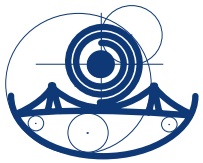
with A_{12} and $\Delta > 0$.

Let

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \text{ and } Y = \begin{bmatrix} Y_{11} & X_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix},$$

be solution and its approximation (obtained by some appropriate solver), resp.

Error bound



For efficient optimization, we need an apriority error bound.
Unfortunately, up to date we did not find usefull apriority error bound (for any existing “fast solvers”)
Thus, we derive a new error bound.
Let

$$A^T X + X A = -Z,$$

be the Lyapunov equation, with

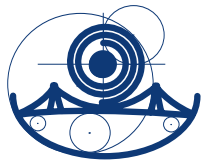
$$A = \begin{bmatrix} 0 & A_{12} \\ -A_{12} & -\Delta \end{bmatrix}, \quad Z = \begin{bmatrix} Z_1 & 0 \\ 0 & Z_2 \end{bmatrix},$$

with A_{12} and $\Delta > 0$.
Let

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \text{ and } Y = \begin{bmatrix} Y_{11} & X_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix},$$

be solution and its approximation (obtained by some appropriate solver), resp.

Error bound



For efficient optimization, we need an apriority error bound.
Unfortunately, up to date we did not find usefull apriority error bound (for any existing “fast solvers”)
Thus, we derive a new error bound.
Let

$$A^T X + X A = -Z,$$

be the Lyapunov equation, with

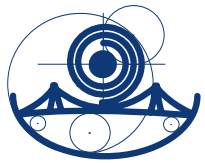
$$A = \begin{bmatrix} 0 & A_{12} \\ -A_{12} & -\Delta \end{bmatrix}, \quad Z = \begin{bmatrix} Z_1 & 0 \\ 0 & Z_2 \end{bmatrix},$$

with A_{12} and $\Delta > 0$.
Let

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \text{ and } Y = \begin{bmatrix} Y_{11} & X_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix},$$

be solution and its approximation (obtained by some appropriate solver), resp.

Error bound



For efficient optimization, we need an apriority error bound.
Unfortunately, up to date we did not find usefull apriority error bound (for any existing “fast solvers”)
Thus, we derive a new error bound.
Let

$$A^T X + X A = -Z,$$

be the Lyapunov equation, with

$$A = \begin{bmatrix} 0 & A_{12} \\ -A_{12} & -\Delta \end{bmatrix}, \quad Z = \begin{bmatrix} Z_1 & 0 \\ 0 & Z_2 \end{bmatrix},$$

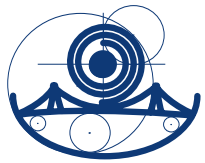
with A_{12} and $\Delta > 0$.

Let

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \text{ and } Y = \begin{bmatrix} Y_{11} & X_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix},$$

be solution and its approximation (obtained by some appropriate solver), resp.

Error bound



For efficient optimization, we need an apriority error bound.
Unfortunately, up to date we did not find usefull apriority error bound (for any existing “fast solvers”)
Thus, we derive a new error bound.
Let

$$A^T X + X A = -Z,$$

be the Lyapunov equation, with

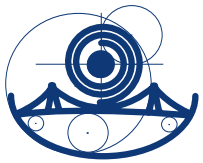
$$A = \begin{bmatrix} 0 & A_{12} \\ -A_{12} & -\Delta \end{bmatrix}, \quad Z = \begin{bmatrix} Z_1 & 0 \\ 0 & Z_2 \end{bmatrix},$$

with A_{12} and $\Delta > 0$.
Let

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \text{ and } Y = \begin{bmatrix} Y_{11} & X_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix},$$

be solution and its approximation (obtained by some appropriate solver), resp.

Error bound



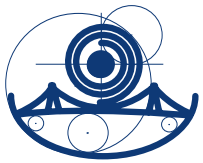
Let $E \doteq A^T Y + Y A + Z$, be residual, written

$$E = \begin{bmatrix} 0 & E_2 \\ E_2^T & E_3 \end{bmatrix} .$$

If A_{12} is diagonal, then the following (new) bound holds

$$|\mathrm{tr}(X - Y)| \lessapprox |\mathrm{tr}(\Delta^{-1} E_3)| + |\mathrm{tr}(E_2 A_{12}^{-1})| \doteq Bnd\,.$$

Error bound



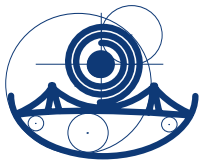
Let $E \doteq A^T Y + Y A + Z$, be residual, written

$$E = \begin{bmatrix} 0 & E_2 \\ E_2^T & E_3 \end{bmatrix}.$$

If A_{12} is diagonal, then the following (new) bound holds

$$|\mathrm{tr}(X - Y)| \lessapprox |\mathrm{tr}(\Delta^{-1} E_3)| + |\mathrm{tr}(E_2 A_{12}^{-1})| \doteq Bnd\,.$$

Error bound



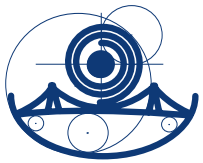
Let $E \doteq A^T Y + Y A + Z$, be residual, written

$$E = \begin{bmatrix} 0 & E_2 \\ E_2^T & E_3 \end{bmatrix} .$$

If A_{12} is diagonal, then the following (new) bound holds

$$|\mathrm{tr}(X - Y)| \lessapprox |\mathrm{tr}(\Delta^{-1} E_3)| + |\mathrm{tr}(E_2 A_{12}^{-1})| \doteq Bnd\,.$$

Error bound



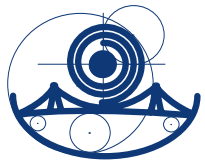
Let $E \doteq A^T Y + Y A + Z$, be residual, written

$$E = \begin{bmatrix} 0 & E_2 \\ E_2^T & E_3 \end{bmatrix} .$$

If A_{12} is diagonal, then the following (new) bound holds

$$|\mathrm{tr}(X - Y)| \lessapprox |\mathrm{tr}(\Delta^{-1} E_3)| + |\mathrm{tr}(E_2 A_{12}^{-1})| \doteq Bnd\,.$$

New algorithm for damping optimization



For illustration let $r = 2$, i.e. number of positions is 2

Input:

Ω – undamped frequencies

(square root of eigenvalues of the pair M and K);

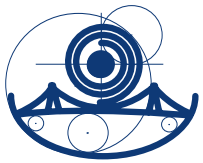
s – the number of dominant undamped frequencies ($2s = \mathrm{rank} Z$);

D_{in} – internal damping, default $D_{in} = \alpha\Omega$, $\alpha \in \{0.01, 0.05\}$;

$maxiter, tol$;

Output: quadruple (s_1, s_2, v_1, v_2) as an approximate minimizer of the trace function.

New algorithm for damping optimization



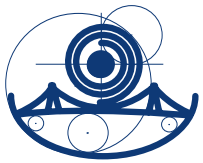
For illustration let $r = 2$, i.e. number of positions is 2

Input:

Ω – undamped frequencies
(square root of eigenvalues of the pair M and K);
s – the number of dominant undamped frequencies ($2s = \mathrm{rank} Z$);
D_{in} – internal damping, default $D_{in} = \alpha\Omega$, $\alpha \in \{0.01, 0.05\}$;
$maxiter, tol$;

Output: quadruple (s_1, s_2, v_1, v_2) as an approximate minimizer of the trace function.

New algorithm for damping optimization



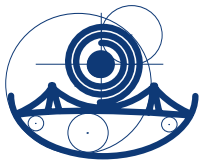
For illustration let $r = 2$, i.e. number of positions is 2

Input:

Ω – undamped frequencies
(square root of eigenvalues of the pair M and K);
s – the number of dominant undamped frequencies ($2s = \mathrm{rank} Z$);
D_{in} – internal damping, default $D_{in} = \alpha\Omega$, $\alpha \in \{0.01, 0.05\}$;
$maxiter, tol$;

Output: quadruple (s_1, s_2, v_1, v_2) as an approximate minimizer of the trace function.

New algorithm for damping optimization



For illustration let $r = 2$, i.e. number of positions is 2

Input:

Ω – undamped frequencies

(square root of eigenvalues of the pair M and K);

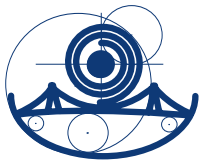
s – the number of dominant undamped frequencies ($2s = \mathrm{rank} Z$);

D_{in} – internal damping, default $D_{in} = \alpha\Omega$, $\alpha \in \{0.01, 0.05\}$;

$maxiter, tol$;

Output: quadruple (s_1, s_2, v_1, v_2) as an approximate minimizer of the trace function.

New algorithm for damping optimization



For illustration let $r = 2$, i.e. number of positions is 2

Input:

Ω – undamped frequencies

(square root of eigenvalues of the pair M and K);

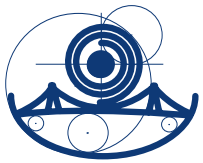
s – the number of dominant undamped frequencies ($2s = \mathrm{rank} Z$);

D_{in} – internal damping, default $D_{in} = \alpha\Omega$, $\alpha \in \{0.01, 0.05\}$;

$maxiter, tol$;

Output: quadruple (s_1, s_2, v_1, v_2) as an approximate minimizer of the trace function.

New algorithm for damping optimization



For illustration let $r = 2$, i.e. number of positions is 2

Input:

Ω – undamped frequencies

(square root of eigenvalues of the pair M and K);

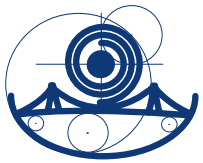
s – the number of dominant undamped frequencies ($2s = \mathrm{rank} Z$);

D_{in} – internal damping, default $D_{in} = \alpha\Omega$, $\alpha \in \{0.01, 0.05\}$;

$maxiter, tol$;

Output: quadruple (s_1, s_2, v_1, v_2) as an approximate minimizer of the trace function.

New algorithm for damping optimization



For illustration let $r = 2$, i.e. number of positions is 2

Input:

Ω – undamped frequencies

(square root of eigenvalues of the pair M and K);

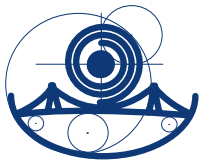
s – the number of dominant undamped frequencies ($2s = \mathrm{rank} Z$);

D_{in} – internal damping, default $D_{in} = \alpha\Omega$, $\alpha \in \{0.01, 0.05\}$;

$maxiter$, tol;

Output: quadruple (s_1, s_2, v_1, v_2) as an approximate minimizer of the trace function.

New algorithm for damping optimization



For illustration let $r = 2$, i.e. number of positions is 2

Input:

Ω – undamped frequencies

(square root of eigenvalues of the pair M and K);

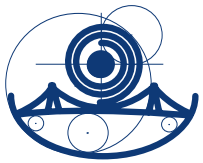
s – the number of dominant undamped frequencies ($2s = \mathrm{rank} Z$);

D_{in} – internal damping, default $D_{in} = \alpha\Omega$, $\alpha \in \{0.01, 0.05\}$;

$maxiter, tol$;

Output: quadruple (s_1, s_2, v_1, v_2) as an approximate minimizer of the trace function.

New algorithm for damping optimization



Phase I: From T_0 local minimums
form a set of quadruples (possible minimums) $(s_1(t), s_2(t), v_1(t), v_2(t))$, $t = 1, \ldots T_0 \cdot (T_0 - 1)/2$, usually (for $r = 2$) $T_0 \leq 0.3n$.

Phase II:
for each quadruple $(s_1(t), s_2(t), v_1(t), v_2(t))$ calculate corresponding trace of the solution of Lyapunov equation

$$A_p(s, v)Y(t) + Y(t)A_p(s, v)^T = -Z_p$$

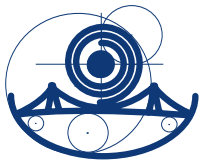
where

$$A_p(s, v) = A_0 - v_1(t) \cdot c_{s_1(t)} c_{s_1(t)}^T - v_2(t) \cdot c_{s_2(t)} c_{s_2(t)}^T ,$$

Phase II: Output

$$(s_{\min}, v_{\min}) \doteq (s_1, s_2, v_1, v_2)_{\min} = \underset{1 \leq t \leq \widehat{T}_0}{\operatorname{argmin}} \operatorname{tr}(Y(t)) .$$

New algorithm for damping optimization



Phase I: From T_0 local minimums

form a set of quadruples (possible minimums) $(s_1(t), s_2(t), v_1(t), v_2(t))$, $t = 1, \ldots T_0 \cdot (T_0 - 1)/2$, usually (for $r = 2$) $T_0 \leq 0.3n$.

Phase II:

for each quadruple $(s_1(t), s_2(t), v_1(t), v_2(t))$ calculate corresponding trace of the solution of Lyapunov equation

$$A_p(s, v)Y(t) + Y(t)A_p(s, v)^T = -Z_p$$

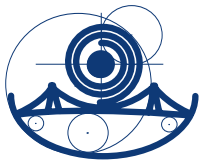
where

$$A_p(s, v) = A_0 - v_1(t) \cdot c_{s_1(t)} c_{s_1(t)}^T - v_2(t) \cdot c_{s_2(t)} c_{s_2(t)}^T ,$$

Phase II: Output

$$(s_{\min}, v_{\min}) \doteq (s_1, s_2, v_1, v_2)_{\min} = \underset{1 \leq t \leq \widehat{T}_0}{\operatorname{argmin}} \operatorname{tr}(Y(t)) .$$

New algorithm for damping optimization



Phase I: From T_0 local minimums

form a set of quadruples (possible minimums) $(s_1(t), s_2(t), v_1(t), v_2(t))$, $t = 1, \ldots T_0 \cdot (T_0 - 1)/2$, usually (for $r = 2$) $T_0 \leq 0.3n$.

Phase II:

for each quadruple $(s_1(t), s_2(t), v_1(t), v_2(t))$ calculate corresponding trace of the solution of Lyapunov equation

$$A_p(s, v)Y(t) + Y(t)A_p(s, v)^T = -Z_p$$

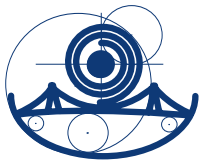
where

$$A_p(s, v) = A_0 - v_1(t) \cdot c_{s_1(t)} c_{s_1(t)}^T - v_2(t) \cdot c_{s_2(t)} c_{s_2(t)}^T ,$$

Phase II: Output

$$(s_{\min}, v_{\min}) \doteq (s_1, s_2, v_1, v_2)_{\min} = \underset{1 \leq t \leq \widehat{T}_0}{\operatorname{argmin}} \operatorname{tr}(Y(t)) .$$

New algorithm for damping optimization



Phase I: From T_0 local minimums
form a set of quadruples (possible minimums) $(s_1(t), s_2(t), v_1(t), v_2(t))$, $t = 1, \ldots T_0 \cdot (T_0 - 1)/2$, usually (for $r = 2$) $T_0 \leq 0.3n$.

Phase II:
for each quadruple $(s_1(t), s_2(t), v_1(t), v_2(t))$ calculate corresponding trace of the solution of Lyapunov equation

$$A_p(s, v)Y(t) + Y(t)A_p(s, v)^T = -Z_p$$

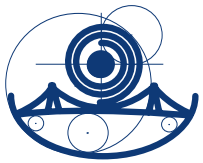
where

$$A_p(s, v) = A_0 - v_1(t) \cdot c_{s_1(t)} c_{s_1(t)}^T - v_2(t) \cdot c_{s_2(t)} c_{s_2(t)}^T ,$$

Phase II: Output

$$(s_{\min}, v_{\min}) \doteq (s_1, s_2, v_1, v_2)_{\min} = \underset{1 \leq t \leq \widehat{T}_0}{\operatorname{argmin}} \operatorname{tr}(Y(t)) .$$

New algorithm for damping optimization



Phase I: From T_0 local minimums
form a set of quadruples (possible minimums) $(s_1(t), s_2(t), v_1(t), v_2(t))$, $t = 1, \ldots T_0 \cdot (T_0 - 1)/2$, usually (for $r = 2$) $T_0 \leq 0.3n$.

Phase II:
for each quadruple $(s_1(t), s_2(t), v_1(t), v_2(t))$ calculate corresponding trace of the solution of Lyapunov equation

$$A_p(s, v)Y(t) + Y(t)A_p(s, v)^T = -Z_p$$

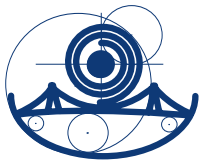
where

$$A_p(s, v) = A_0 - v_1(t) \cdot c_{s_1(t)} c_{s_1(t)}^T - v_2(t) \cdot c_{s_2(t)} c_{s_2(t)}^T ,$$

Phase II: Output

$$(s_{\min}, v_{\min}) \doteq (s_1, s_2, v_1, v_2)_{\min} = \underset{1 \leq t \leq \widehat{T}_0}{\operatorname{argmin}} \operatorname{tr}(Y(t)) .$$

New algorithm for damping optimization



Phase I: From T_0 local minimums
form a set of quadruples (possible minimums) $(s_1(t), s_2(t), v_1(t), v_2(t))$, $t = 1, \ldots T_0 \cdot (T_0 - 1)/2$, usually (for $r = 2$) $T_0 \leq 0.3n$.

Phase II:
for each quadruple $(s_1(t), s_2(t), v_1(t), v_2(t))$ calculate corresponding trace of the solution of Lyapunov equation

$$A_p(s, v)Y(t) + Y(t)A_p(s, v)^T = -Z_p$$

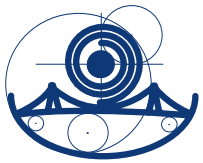
where

$$A_p(s, v) = A_0 - v_1(t) \cdot c_{s_1(t)} c_{s_1(t)}^T - v_2(t) \cdot c_{s_2(t)} c_{s_2(t)}^T ,$$

Phase II: Output

$$(s_{\min}, v_{\min}) \doteq (s_1, s_2, v_1, v_2)_{\min} = \underset{1 \leq t \leq \widehat{T}_0}{\operatorname{argmin}} \operatorname{tr}(Y(t)) .$$

New algorithm for damping optimization



Phase I: From T_0 local minimums
form a set of quadruples (possible minimums) $(s_1(t), s_2(t), v_1(t), v_2(t))$, $t = 1, \ldots T_0 \cdot (T_0 - 1)/2$, usually (for $r = 2$) $T_0 \leq 0.3n$.

Phase II:
for each quadruple $(s_1(t), s_2(t), v_1(t), v_2(t))$ calculate corresponding trace of the solution of Lyapunov equation

$$A_p(s, v)Y(t) + Y(t)A_p(s, v)^T = -Z_p$$

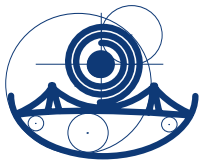
where

$$A_p(s, v) = A_0 - v_1(t) \cdot c_{s_1(t)} c_{s_1(t)}^T - v_2(t) \cdot c_{s_2(t)} c_{s_2(t)}^T \,,$$

Phase II: Output

$$(s_{\min}, v_{\min}) \doteq (s_1, s_2, v_1, v_2)_{\min} = \underset{1 \leq t \leq \widehat{T}_0}{\operatorname{argmin}} \operatorname{tr}(Y(t)) \,.$$

New algorithm for damping optimization



Phase I: From T_0 local minimums
form a set of quadruples (possible minimums) $(s_1(t), s_2(t), v_1(t), v_2(t))$, $t = 1, \ldots T_0 \cdot (T_0 - 1)/2$, usually (for $r = 2$) $T_0 \leq 0.3n$.

Phase II:
for each quadruple $(s_1(t), s_2(t), v_1(t), v_2(t))$ calculate corresponding trace of the solution of Lyapunov equation

$$A_p(s, v)Y(t) + Y(t)A_p(s, v)^T = -Z_p$$

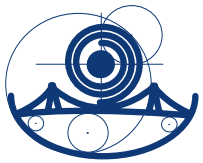
where

$$A_p(s, v) = A_0 - v_1(t) \cdot c_{s_1(t)} c_{s_1(t)}^T - v_2(t) \cdot c_{s_2(t)} c_{s_2(t)}^T \,,$$

Phase II: Output

$$(s_{\min}, v_{\min}) \doteq (s_1, s_2, v_1, v_2)_{\min} = \underset{1 \leq t \leq \widehat{T}_0}{\operatorname{argmin}} \operatorname{tr}(Y(t)) \,.$$

New algorithm for damping optimization



Phase I: From T_0 local minimums
form a set of quadruples (possible minimums) $(s_1(t), s_2(t), v_1(t), v_2(t))$, $t = 1, \ldots T_0 \cdot (T_0 - 1)/2$, usually (for $r = 2$) $T_0 \leq 0.3n$.

Phase II:
for each quadruple $(s_1(t), s_2(t), v_1(t), v_2(t))$ calculate corresponding trace of the solution of Lyapunov equation

$$A_p(s, v)Y(t) + Y(t)A_p(s, v)^T = -Z_p$$

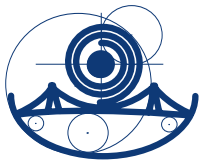
where

$$A_p(s, v) = A_0 - v_1(t) \cdot c_{s_1(t)} c_{s_1(t)}^T - v_2(t) \cdot c_{s_2(t)} c_{s_2(t)}^T \,,$$

Phase II: Output

$$(s_{\min}, v_{\min}) \doteq (s_1, s_2, v_1, v_2)_{\min} = \underset{1 \leq t \leq \widehat{T}_0}{\operatorname{argmin}} \operatorname{tr}(Y(t)) \,.$$

Numerical examples



Case I Random

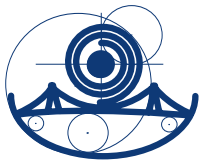
$M_0 = \mathrm{randn}(n), K_0 = \mathrm{randn}(n); M = M_0 M_0^*, K = \sqrt{K_0 K_0^*},$
$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5};\ \ \Phi^* M \Phi = I, \Phi^* M \Phi = \Omega^2,$
"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 50$

Our approach: 33 potential positions, ($n\times$ 1D opt. $+528$ Lyap. sol.) $\Rightarrow$
opt. pos. $(16, 31)$; opt. visc. $(18.5, 25.4)$.
after 13 sec. opt. trace 651.82

vs. standard approach 50 potential positions, ($1225\times$ 2D opt. each
$\approx 40\times$ Lyap. sol.); opt. pos. $(16, 31)$ and visc. $(22.63, 29.59)$
after 600 sec. opt trace 645.957

Numerical examples



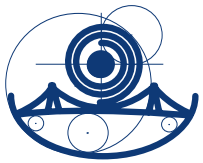
Case I Random

$M_0 = \mathrm{randn}(n), K_0 = \mathrm{randn}(n); M = M_0 M_0^*, K = \sqrt{K_0 K_0^*},$
$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5};\ \ \Phi^* M \Phi = I, \Phi^* M \Phi = \Omega^2,$
"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 50$
Our approach: 33 potential positions, ($n\times$ 1D opt. $+528$ Lyap. sol.) $\Rightarrow$
opt. pos. $(16, 31)$; opt. visc. $(18.5, 25.4)$.
after 13 sec. opt. trace 651.82

vs. standard approach 50 potential positions, ($1225\times$ 2D opt. each
$\approx 40\times$ Lyap. sol.); opt. pos. $(16, 31)$ and visc. $(22.63, 29.59)$
after 600 sec. opt trace 645.957

Numerical examples



Case I Random

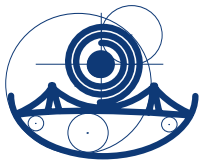
$M_0 = \mathrm{randn}(n)$, $K_0 = \mathrm{randn}(n)$; $M = M_0 M_0^*$, $K = \sqrt{K_0 K_0^*}$,
$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5}$; $\Phi^* M \Phi = I$, $\Phi^* M \Phi = \Omega^2$,
"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 50$

Our approach: 33 potential positions, ($n\times$ 1D opt. $+528$ Lyap. sol.) $\Rightarrow$
opt. pos. $(16, 31)$; opt. visc. $(18.5, 25.4)$.
after 13 sec. opt. trace 651.82

vs. standard approach 50 potential positions, ($1225\times$ 2D opt. each
$\approx 40\times$ Lyap. sol.); opt. pos. $(16, 31)$ and visc. $(22.63, 29.59)$
after 600 sec. opt trace 645.957

Numerical examples



Case I Random

$M_0 = \mathrm{randn}(n)$, $K_0 = \mathrm{randn}(n)$; $M = M_0 M_0^*$, $K = \sqrt{K_0 K_0^*}$,

$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5}$; $\Phi^* M \Phi = I$, $\Phi^* M \Phi = \Omega^2$,

"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 50$

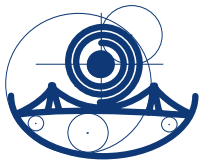
Our approach: 33 potential positions, ($n\times$ 1D opt. $+528$ Lyap. sol.) $\Rightarrow$ opt. pos. $(16, 31)$; opt. visc. $(18.5, 25.4)$.

after 13 sec. opt. trace 651.82

vs. standard approach 50 potential positions, ($1225\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); opt. pos. $(16, 31)$ and visc. $(22.63, 29.59)$

after 600 sec. opt trace 645.957

Numerical examples



Case I Random

$M_0 = \mathrm{randn}(n)$, $K_0 = \mathrm{randn}(n)$; $M = M_0 M_0^*$, $K = \sqrt{K_0 K_0^*}$,

$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5}$; $\Phi^* M \Phi = I$, $\Phi^* M \Phi = \Omega^2$,

"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 50$

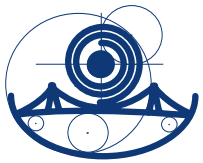
Our approach: 33 potential positions, ($n\times$ 1D opt. $+528$ Lyap. sol.) $\Rightarrow$ opt. pos. $(16, 31)$; opt. visc. $(18.5, 25.4)$.

after 13 sec. opt. trace 651.82

vs. standard approach 50 potential positions, ($1225\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); opt. pos. $(16, 31)$ and visc. $(22.63, 29.59)$

after 600 sec. opt trace 645.957

Numerical examples



Case I Random

$M_0 = \mathrm{randn}(n)$, $K_0 = \mathrm{randn}(n)$; $M = M_0 M_0^*$, $K = \sqrt{K_0 K_0^*}$,

$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5}$; $\Phi^* M \Phi = I$, $\Phi^* M \Phi = \Omega^2$,

"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 50$

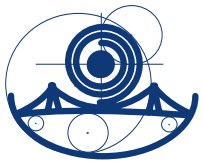
Our approach: 33 potential positions, ($n\times$ 1D opt. $+528$ Lyap. sol.) $\Rightarrow$ opt. pos. $(16, 31)$; opt. visc. $(18.5, 25.4)$.

after 13 sec. opt. trace 651.82

vs. standard approach 50 potential positions, ($1225\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); opt. pos. $(16, 31)$ and visc. $(22.63, 29.59)$

after 600 sec. opt trace 645.957

Numerical examples



Case I Random

$M_0 = \mathrm{randn}(n)$, $K_0 = \mathrm{randn}(n)$; $M = M_0 M_0^*$, $K = \sqrt{K_0 K_0^*}$,

$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5}$; $\Phi^* M \Phi = I$, $\Phi^* M \Phi = \Omega^2$,

"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 50$

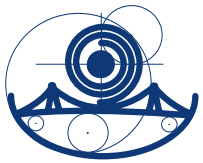
Our approach: 33 potential positions, ($n\times$ 1D opt. $+528$ Lyap. sol.) $\Rightarrow$ opt. pos. $(16, 31)$; opt. visc. $(18.5, 25.4)$.

after 13 sec. opt. trace 651.82

vs. standard approach 50 potential positions, ($1225\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); opt. pos. $(16, 31)$ and visc. $(22.63, 29.59)$

after 600 sec. opt trace 645.957

Numerical examples



Case I Random

$M_0 = \mathrm{randn}(n)$, $K_0 = \mathrm{randn}(n)$; $M = M_0 M_0^*$, $K = \sqrt{K_0 K_0^*}$,

$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5}$; $\Phi^* M \Phi = I$, $\Phi^* M \Phi = \Omega^2$,

"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 50$

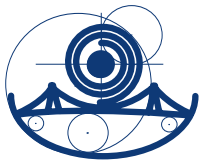
Our approach: 33 potential positions, ($n\times$ 1D opt. $+528$ Lyap. sol.) $\Rightarrow$ opt. pos. $(16, 31)$; opt. visc. $(18.5, 25.4)$.

after 13 sec. opt. trace 651.82

vs. standard approach 50 potential positions, ($1225\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); opt. pos. $(16, 31)$ and visc. $(22.63, 29.59)$

after 600 sec. opt trace 645.957

Numerical examples



Case I Random

$M_0 = \mathrm{randn}(n)$, $K_0 = \mathrm{randn}(n)$; $M = M_0 M_0^*$, $K = \sqrt{K_0 K_0^*}$,

$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5}$; $\Phi^* M \Phi = I$, $\Phi^* M \Phi = \Omega^2$,

"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 50$

Our approach: 33 potential positions, ($n\times$ 1D opt. $+528$ Lyap. sol.) $\Rightarrow$ opt. pos. $(16, 31)$; opt. visc. $(18.5, 25.4)$.

after 13 sec. opt. trace 651.82

vs. standard approach 50 potential positions, ($1225\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); opt. pos. $(16, 31)$ and visc. $(22.63, 29.59)$

after 600 sec. opt trace 645.957

Numerical examples

Case I Random

$M_0 = \mathrm{randn}(n)$, $K_0 = \mathrm{randn}(n)$; $M = M_0 M_0^*$, $K = \sqrt{K_0 K_0^*}$,

$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5}$; $\Phi^* M \Phi = I$, $\Phi^* M \Phi = \Omega^2$,

"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 100$

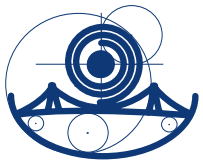
Our approach: 59 potential positions, ($n\times$ 1D opt. $+1770$ Lyap. sol.) $\Rightarrow$ opt. pos. $(14, 98)$; opt. visc. $(30.5631, 35.7604)$.

after 157 sec. opt. trace 1428.2

vs. standard approach 100 potential positions, ($4950\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); opt. pos. $(14, 98)$ and visc. $(29.9, 52.14)$

after 4.5 hours opt trace 1251.5

Numerical examples



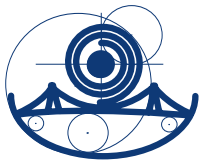
Case I Random

$M_0 = \mathrm{randn}(n), K_0 = \mathrm{randn}(n); M = M_0 M_0^*, K = \sqrt{K_0 K_0^*},$
$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5};\ \Phi^* M \Phi = I, \Phi^* M \Phi = \Omega^2,$
"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 100$
Our approach: 59 potential positions, ($n\times$ 1D opt. $+1770$ Lyap. sol.) $\Rightarrow$
opt. pos. $(14, 98)$; opt. visc. $(30.5631, 35.7604)$.
after 157 sec. opt. trace 1428.2

vs. standard approach 100 potential positions, ($4950\times$ 2D opt. each
$\approx 40\times$ Lyap. sol.); opt. pos. $(14, 98)$ and visc. $(29.9, 52.14)$
after 4.5 hours opt trace 1251.5

Numerical examples



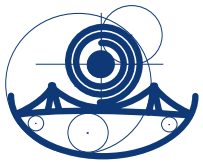
Case I Random

$M_0 = \mathrm{randn}(n)$, $K_0 = \mathrm{randn}(n)$; $M = M_0 M_0^*$, $K = \sqrt{K_0 K_0^*}$,
$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5}$; $\Phi^* M \Phi = I$, $\Phi^* M \Phi = \Omega^2$,
"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 100$
Our approach: 59 potential positions, ($n\times$ 1D opt. $+1770$ Lyap. sol.) $\Rightarrow$
opt. pos. $(14, 98)$; opt. visc. $(30.5631, 35.7604)$.
after 157 sec. opt. trace 1428.2

vs. standard approach 100 potential positions, ($4950\times$ 2D opt. each
$\approx 40\times$ Lyap. sol.); opt. pos. $(14, 98)$ and visc. $(29.9, 52.14)$
after 4.5 hours opt trace 1251.5

Numerical examples



Case I Random

$M_0 = \mathrm{randn}(n)$, $K_0 = \mathrm{randn}(n)$; $M = M_0 M_0^*$, $K = \sqrt{K_0 K_0^*}$,

$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5}$; $\Phi^* M \Phi = I$, $\Phi^* M \Phi = \Omega^2$,

"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 100$

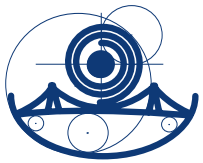
Our approach: 59 potential positions, ($n\times$ 1D opt. $+1770$ Lyap. sol.) $\Rightarrow$ opt. pos. $(14, 98)$; opt. visc. $(30.5631, 35.7604)$.

after 157 sec. opt. trace 1428.2

vs. standard approach 100 potential positions, ($4950\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); opt. pos. $(14, 98)$ and visc. $(29.9, 52.14)$

after 4.5 hours opt trace 1251.5

Numerical examples



Case I Random

$M_0 = \mathrm{randn}(n)$, $K_0 = \mathrm{randn}(n)$; $M = M_0 M_0^*$, $K = \sqrt{K_0 K_0^*}$,

$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5}$; $\Phi^* M \Phi = I$, $\Phi^* M \Phi = \Omega^2$,

"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 100$

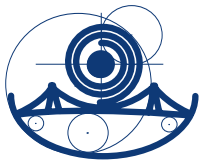
Our approach: 59 potential positions, ($n\times$ 1D opt. $+1770$ Lyap. sol.) $\Rightarrow$ opt. pos. $(14, 98)$; opt. visc. $(30.5631, 35.7604)$.

after 157 sec. opt. trace 1428.2

vs. standard approach 100 potential positions, ($4950\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); opt. pos. $(14, 98)$ and visc. $(29.9, 52.14)$

after 4.5 hours opt trace 1251.5

Numerical examples



Case I Random

$M_0 = \mathrm{randn}(n)$, $K_0 = \mathrm{randn}(n)$; $M = M_0 M_0^*$, $K = \sqrt{K_0 K_0^*}$,

$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5}$; $\Phi^* M \Phi = I$, $\Phi^* M \Phi = \Omega^2$,

"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 100$

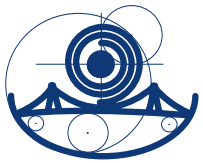
Our approach: 59 potential positions, ($n\times$ 1D opt. $+1770$ Lyap. sol.) $\Rightarrow$ opt. pos. $(14, 98)$; opt. visc. $(30.5631, 35.7604)$.

after 157 sec. opt. trace 1428.2

vs. standard approach 100 potential positions, ($4950\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); opt. pos. $(14, 98)$ and visc. $(29.9, 52.14)$

after 4.5 hours opt trace 1251.5

Numerical examples



Case I Random

$M_0 = \mathrm{randn}(n)$, $K_0 = \mathrm{randn}(n)$; $M = M_0 M_0^*$, $K = \sqrt{K_0 K_0^*}$,

$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5}$; $\Phi^* M \Phi = I$, $\Phi^* M \Phi = \Omega^2$,

"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 100$

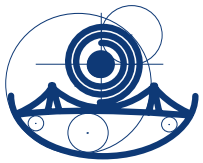
Our approach: 59 potential positions, ($n\times$ 1D opt. $+1770$ Lyap. sol.) $\Rightarrow$ opt. pos. $(14, 98)$; opt. visc. $(30.5631, 35.7604)$.

after 157 sec. opt. trace 1428.2

vs. standard approach 100 potential positions, ($4950\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); opt. pos. $(14, 98)$ and visc. $(29.9, 52.14)$

after 4.5 hours opt trace 1251.5

Numerical examples



Case I Random

$M_0 = \mathrm{randn}(n)$, $K_0 = \mathrm{randn}(n)$; $M = M_0 M_0^*$, $K = \sqrt{K_0 K_0^*}$,

$Z = I_5 \oplus 0_{n-5} \oplus I_5 \oplus 0_{n-5}$; $\Phi^* M \Phi = I$, $\Phi^* M \Phi = \Omega^2$,

"dampers' positions" $(s_1, s_2) \Rightarrow \Phi(s_1, s_2, :)$

For $n = 100$

Our approach: 59 potential positions, ($n\times$ 1D opt. $+1770$ Lyap. sol.) $\Rightarrow$ opt. pos. $(14, 98)$; opt. visc. $(30.5631, 35.7604)$.

after 157 sec. opt. trace 1428.2

vs. standard approach 100 potential positions, ($4950\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); opt. pos. $(14, 98)$ and visc. $(29.9, 52.14)$

after 4.5 hours opt trace 1251.5

Numerical examples

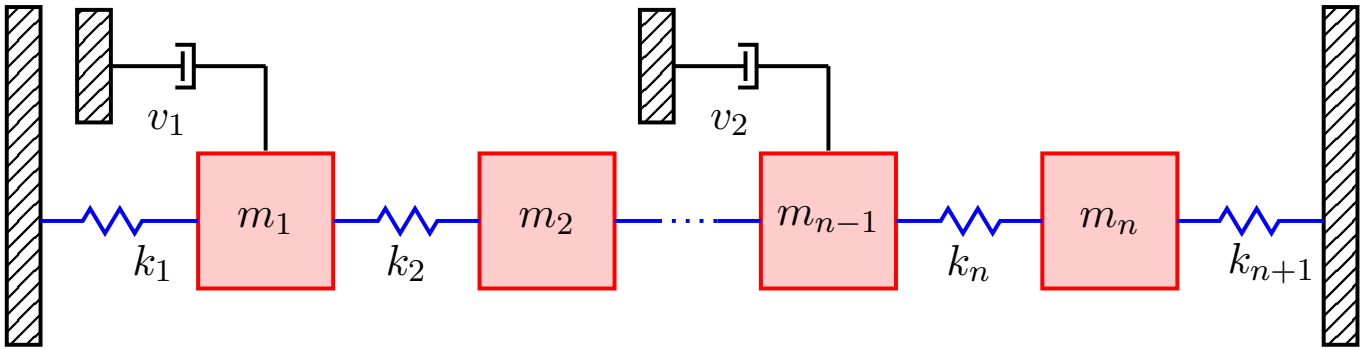


Figure: n mass oscillator

Case II $n = 1000$

$m_i = 100(i-1)+2, k_i = 1, i = 1, \ldots, n,$

$Z = I_{100} \oplus 0_{900}, \ \oplus I_{100} \oplus 0_{900}, \quad M = \mathrm{diag}(m_i)/(\sum m_i).$

Our approach: 45 potential positions (with 292 different viscosities) , ($n\times$ 1D opt. $+42486$ Lyap. sol.) $\Rightarrow$ opt. pos. $(292, 830)$; opt. visc. $(0.0469, 0.0744)$.

Using projection with $p = 200$, we need approx. 2.5 hours

vs. standard approach 1000 potential positions, ($499500\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); No Way !!! approx. 5550 hours

Numerical examples

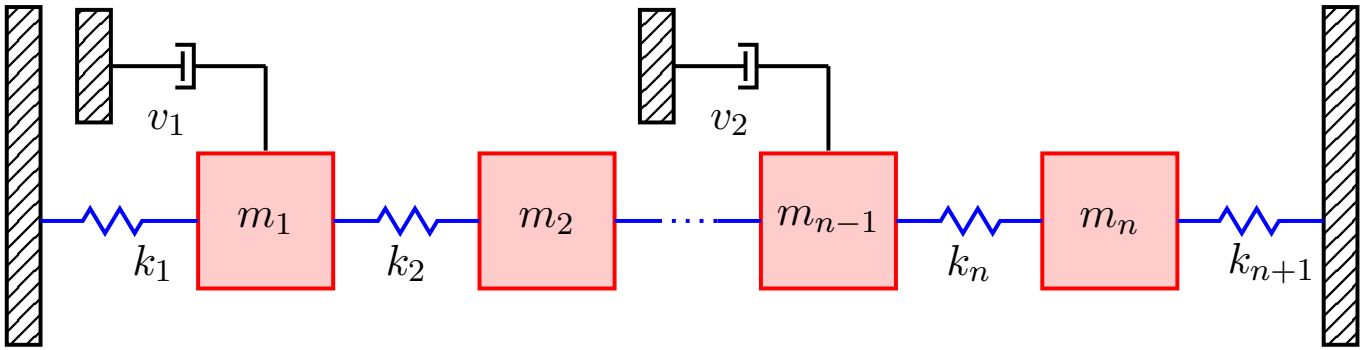


Figure: n mass oscillator

Case II $n = 1000$

$m_i = 100(i-1)+2, k_i = 1, i = 1, \ldots, n,$

$Z = I_{100} \oplus 0_{900}, \ \oplus I_{100} \oplus 0_{900}, \quad M = \mathrm{diag}(m_i)/(\sum m_i).$

Our approach: 45 potential positions (with 292 different viscosities) , ($n\times$ 1D opt. $+42486$ Lyap. sol.) $\Rightarrow$ opt. pos. $(292, 830)$; opt. visc. $(0.0469, 0.0744)$.

Using projection with $p = 200$, we need approx. 2.5 hours

vs. standard approach 1000 potential positions, ($499500\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); No Way !!! approx. 5550 hours

Numerical examples

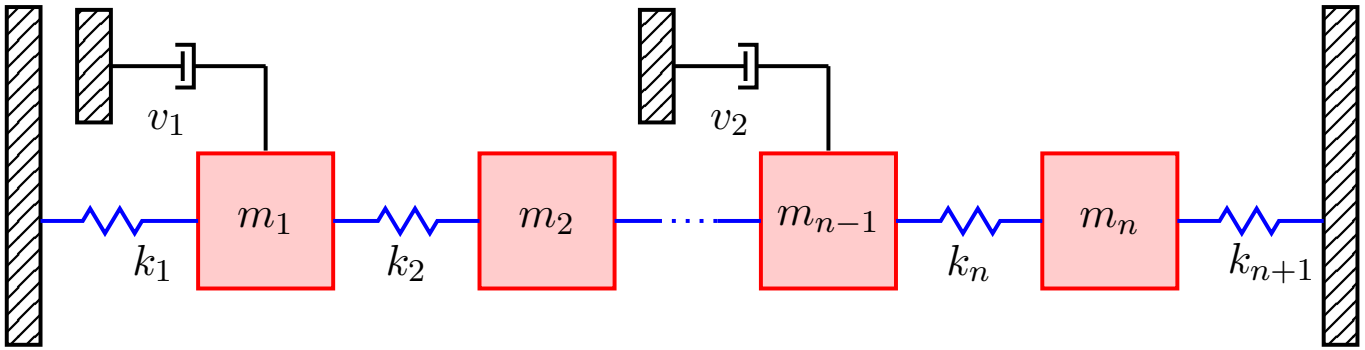


Figure: n mass oscillator

Case II $n = 1000$

$m_i = 100(i-1)+2, k_i = 1, i = 1, \ldots, n,$

$Z = I_{100} \oplus 0_{900}, \ \oplus I_{100} \oplus 0_{900}, \quad M = \mathrm{diag}(m_i)/(\sum m_i).$

Our approach: 45 potential positions (with 292 different viscosities) , ($n\times$ 1D opt. $+42486$ Lyap. sol.) $\Rightarrow$ opt. pos. $(292, 830)$; opt. visc. $(0.0469, 0.0744)$.

Using projection with $p = 200$, we need approx. 2.5 hours

vs. standard approach 1000 potential positions, ($499500\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); No Way !!! approx. 5550 hours

Numerical examples

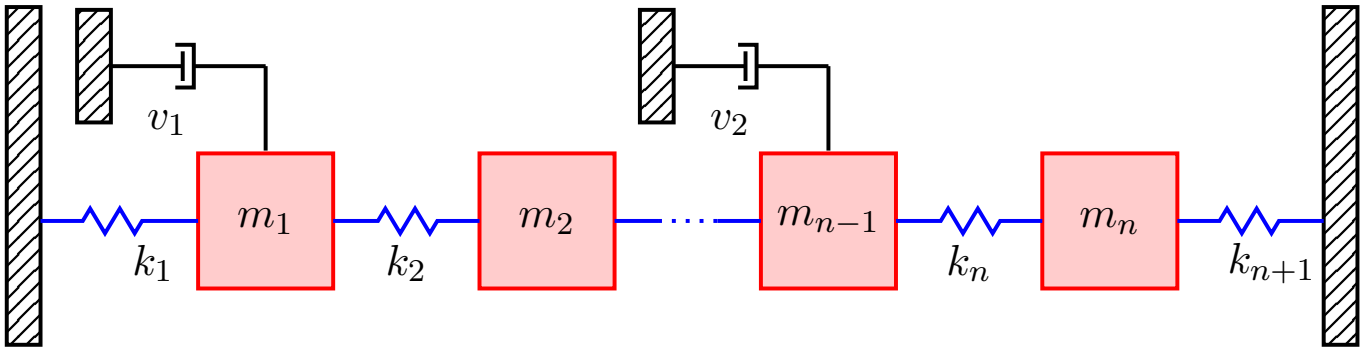


Figure: n mass oscillator

Case II $n = 1000$

$m_i = 100(i-1)+2, k_i = 1, i = 1, \ldots, n,$

$Z = I_{100} \oplus 0_{900}, \ \oplus I_{100} \oplus 0_{900}, \quad M = \mathrm{diag}(m_i)/(\sum m_i).$

Our approach: 45 potential positions (with 292 different viscosities) , ($n\times$ 1D opt. $+42486$ Lyap. sol.) $\Rightarrow$ opt. pos. $(292, 830)$; opt. visc. $(0.0469, 0.0744)$.

Using projection with $p = 200$, we need approx. 2.5 hours

vs. standard approach 1000 potential positions, ($499500\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); No Way !!! approx. 5550 hours

Numerical examples

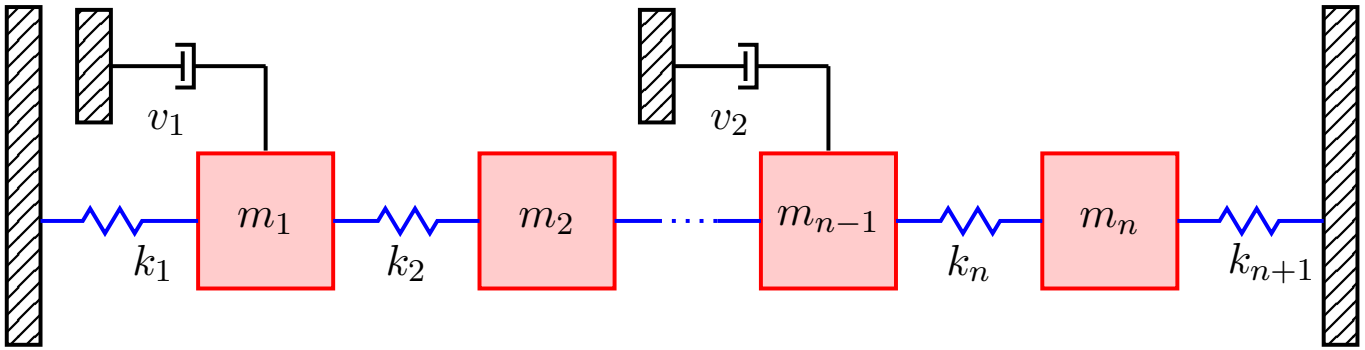


Figure: n mass oscillator

Case II $n = 1000$

$m_i = 100(i-1)+2,\ k_i = 1,\ i = 1,\dots,n,$

$Z = I_{100} \oplus 0_{900},\ \oplus I_{100} \oplus 0_{900},\quad M = \mathrm{diag}(m_i)/(\sum m_i).$

Our approach: 45 potential positions (with 292 different viscosities) , ($n\times$ 1D opt. $+42486$ Lyap. sol.) $\Rightarrow$ opt. pos. $(292, 830)$; opt. visc. $(0.0469, 0.0744)$.

Using projection with $p = 200$, we need approx. 2.5 hours

vs. standard approach 1000 potential positions, ($499500\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); No Way !!! approx. 5550 hours

Numerical examples

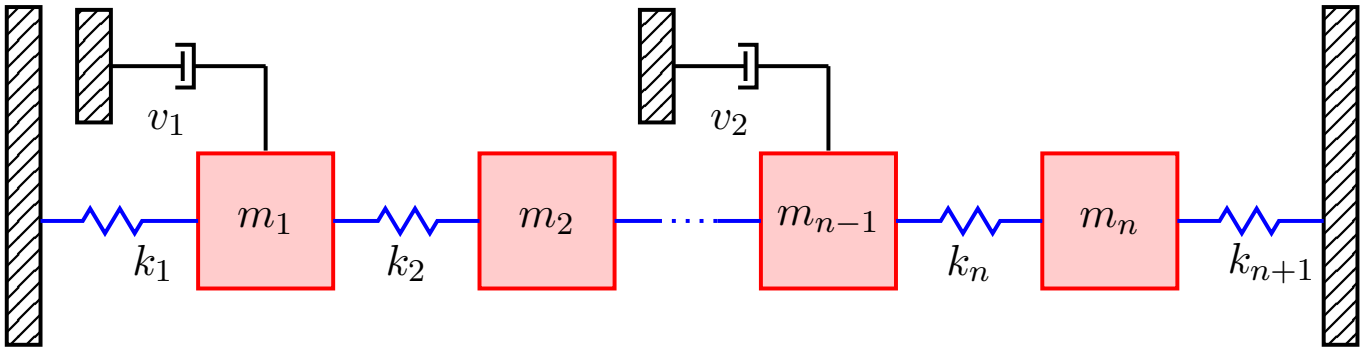


Figure: n mass oscillator

Case II $n = 1000$

$m_i = 100(i-1)+2,\ k_i = 1,\ i = 1,\ldots,n,$

$Z = I_{100} \oplus 0_{900},\ \oplus I_{100} \oplus 0_{900},\quad M = \mathrm{diag}(m_i)/(\sum m_i).$

Our approach: 45 potential positions (with 292 different viscosities) , ($n\times$ 1D opt. $+42486$ Lyap. sol.) $\Rightarrow$ opt. pos. $(292, 830)$; opt. visc. $(0.0469, 0.0744)$.

Using projection with $p = 200$, we need approx. 2.5 hours

vs. standard approach 1000 potential positions, ($499500\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); No Way !!! approx. 5550 hours

Numerical examples

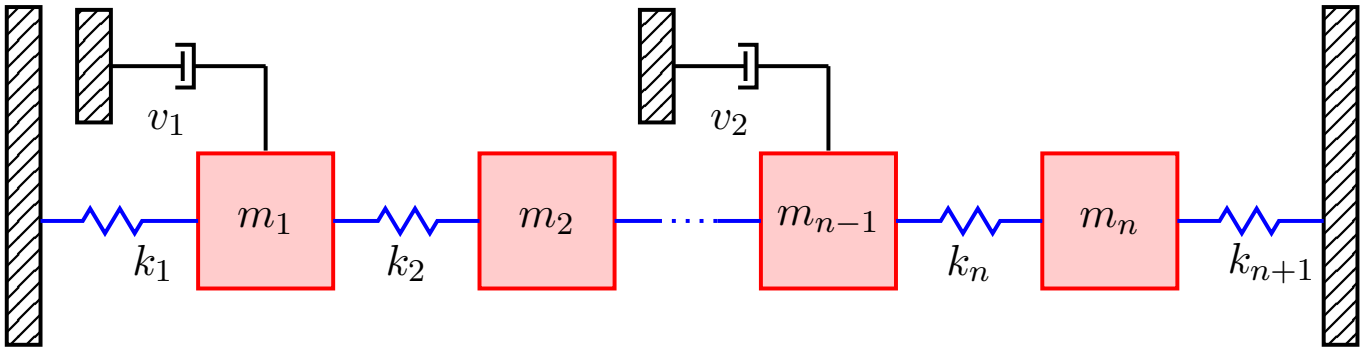


Figure: n mass oscillator

Case II $n = 1000$

$m_i = 100(i-1)+2,\ k_i = 1,\ i = 1,\ldots,n,$

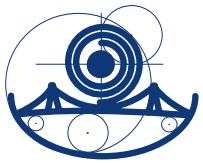
$Z = I_{100} \oplus 0_{900},\ \oplus I_{100} \oplus 0_{900},\quad M = \mathrm{diag}(m_i)/(\sum m_i).$

Our approach: 45 potential positions (with 292 different viscosities) , ($n\times$ 1D opt. $+42486$ Lyap. sol.) $\Rightarrow$ opt. pos. $(292, 830)$; opt. visc. $(0.0469, 0.0744)$.

Using projection with $p = 200$, we need approx. 2.5 hours

vs. standard approach 1000 potential positions, ($499500\times$ 2D opt. each $\approx 40\times$ Lyap. sol.); No Way !!! approx. 5550 hours

Singular mass matrix



We still consider a system:

$$M\ddot{x} + C\dot{x} + Kx = 0\,, \quad x(0) = x_0, \quad \dot{x}(0) = \dot{x}_0\,,$$

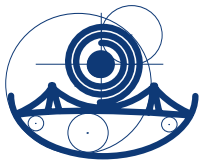
where $M \geq 0, C \geq 0$ and $K > 0$.

Simult. diagonal.

$$\Phi^T M \Phi = \begin{bmatrix} 0_{n_0} & \\ & I_m \end{bmatrix}, \qquad \Phi^T K \Phi = \begin{bmatrix} \Omega_0^2 & \\ & \Omega_+^2 \end{bmatrix},$$

Let $n_0 \doteq \dim(\mathcal{N}(M)) \geq 1$.

Singular mass matrix



We still consider a system:

$$M\ddot{x} + C\dot{x} + Kx = 0\,, \quad x(0) = x_0, \quad \dot{x}(0) = \dot{x}_0\,,$$

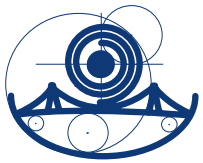
where $M \geq 0, C \geq 0$ and $K > 0$.

Simult. diagonal.

$$\Phi^T M \Phi = \begin{bmatrix} 0_{n_0} & \\ & I_m \end{bmatrix}, \qquad \Phi^T K \Phi = \begin{bmatrix} \Omega_0^2 & \\ & \Omega_+^2 \end{bmatrix},$$

Let $n_0 \doteq \dim(\mathcal{N}(M)) \geq 1$.

Singular mass matrix



We still consider a system:

$$M\ddot{x} + C\dot{x} + Kx = 0\,, \quad x(0) = x_0, \quad \dot{x}(0) = \dot{x}_0\,,$$

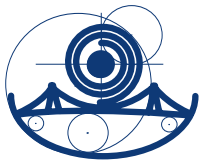
where $M \geq 0$, $C \geq 0$ and $K > 0$.

Simult. diagonal.

$$\Phi^T M \Phi = \begin{bmatrix} 0_{n_0} & \\ & I_m \end{bmatrix}, \qquad \Phi^T K \Phi = \begin{bmatrix} \Omega_0^2 & \\ & \Omega_+^2 \end{bmatrix},$$

Let $n_0 \doteq \dim(\mathcal{N}(M)) \geq 1$.

Singular mass matrix



We still consider a system:

$$M\ddot{x} + C\dot{x} + Kx = 0\,, \quad x(0) = x_0, \quad \dot{x}(0) = \dot{x}_0\,,$$

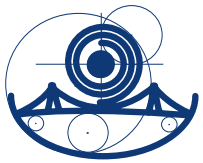
where $M \geq 0$, $C \geq 0$ and $K > 0$.

Simult. diagonal.

$$\Phi^T M \Phi = \begin{bmatrix} 0_{n_0} & \\ & I_m \end{bmatrix}, \qquad \Phi^T K \Phi = \begin{bmatrix} \Omega_0^2 & \\ & \Omega_+^2 \end{bmatrix},$$

Let $n_0 \doteq \dim(\mathcal{N}(M)) \geq 1$.

Singular mass matrix



We still consider a system:

$$M\ddot{x} + C\dot{x} + Kx = 0\,, \quad x(0) = x_0, \quad \dot{x}(0) = \dot{x}_0\,,$$

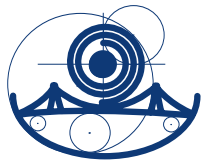
where $M \geq 0, C \geq 0$ and $K > 0$.

Simult. diagonal.

$$\Phi^T M \Phi = \begin{bmatrix} 0_{n_0} & \\ & I_m \end{bmatrix}\,, \qquad \Phi^T K \Phi = \begin{bmatrix} \Omega_0^2 & \\ & \Omega_+^2 \end{bmatrix}\,,$$

Let $n_0 \doteq \dim(\mathcal{N}(M)) \geq 1$.

Singular mass matrix



Now, the damping

$$D = \Phi^T \left(C_{in} + v \cdot C_{ex}\right) \Phi = \begin{bmatrix} D_{11} & D_{12} \\ D_{12}^T & D_{22} \end{bmatrix},$$
$$D_{11} \in \mathbb{R}^{n_0, n_0}, \qquad D_{22} \neq 0\,.$$

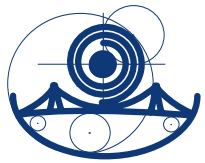
All this implies

$$D_{11}\dot{x}_1 + D_{12}\dot{x}_2 + \Omega_0^2 x_0 = 0,$$
$$\ddot{x}_2 + D_{12}^T\dot{x}_1 + D_{22}\dot{x}_2 + \Omega_+^2 x_+ = 0\,,$$

where $x = [x_0, x_+]$, x_0 and x_+ are of dimension n_0 and m, respectively.
By substitution

$$z_1 = \Omega_0 x_0\,, \qquad z_2 = \Omega_+ x_+\,, \qquad z_3 = \dot{x}_2\,,$$

Singular mass matrix



Now, the damping

$$D = \Phi^T \left(C_{in} + v \cdot C_{ex}\right) \Phi = \begin{bmatrix} D_{11} & D_{12} \\ D_{12}^T & D_{22} \end{bmatrix},$$
$$D_{11} \in \mathbb{R}^{n_0, n_0}, \qquad D_{22} \neq 0\,.$$

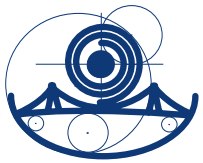
All this implies

$$D_{11}\dot{x}_1 + D_{12}\dot{x}_2 + \Omega_0^2 x_0 = 0,$$
$$\ddot{x}_2 + D_{12}^T\dot{x}_1 + D_{22}\dot{x}_2 + \Omega_+^2 x_+ = 0\,,$$

where $x = [x_0, x_+]$, x_0 and x_+ are of dimension n_0 and m, respectively.
By substitution

$$z_1 = \Omega_0 x_0\,, \qquad z_2 = \Omega_+ x_+\,, \qquad z_3 = \dot{x}_2\,,$$

Singular mass matrix



Now, the damping

$$D = \Phi^T \left(C_{in} + v \cdot C_{ex}\right) \Phi = \begin{bmatrix} D_{11} & D_{12} \\ D_{12}^T & D_{22} \end{bmatrix},$$
$$D_{11} \in \mathbb{R}^{n_0, n_0}, \qquad D_{22} \neq 0\,.$$

All this implies

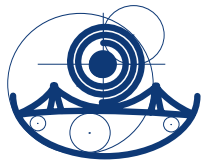
$$D_{11}\dot{x}_1 + D_{12}\dot{x}_2 + \Omega_0^2 x_0 = 0,$$
$$\ddot{x}_2 + D_{12}^T\dot{x}_1 + D_{22}\dot{x}_2 + \Omega_+^2 x_+ = 0\,,$$

where $x = [x_0, x_+]$, x_0 and x_+ are of dimension n_0 and m, respectively.

By substitution

$$z_1 = \Omega_0 x_0\,, \qquad z_2 = \Omega_+ x_+\,, \qquad z_3 = \dot{x}_2\,,$$

Singular mass matrix



Now, the damping

$$D = \Phi^T \left(C_{in} + v \cdot C_{ex}\right) \Phi = \begin{bmatrix} D_{11} & D_{12} \\ D_{12}^T & D_{22} \end{bmatrix},$$
$$D_{11} \in \mathbb{R}^{n_0, n_0}, \qquad D_{22} \neq 0\,.$$

All this implies

$$D_{11}\dot{x}_1 + D_{12}\dot{x}_2 + \Omega_0^2 x_0 = 0,$$
$$\ddot{x}_2 + D_{12}^T\dot{x}_1 + D_{22}\dot{x}_2 + \Omega_+^2 x_+ = 0\,,$$

where $x = [x_0, x_+]$, x_0 and x_+ are of dimension n_0 and m, respectively.
By substitution

$$z_1 = \Omega_0 x_0\,, \qquad z_2 = \Omega_+ x_+\,, \qquad z_3 = \dot{x}_2\,,$$

Singular mass matrix

The ODE system

$$\dot{z}_1 = -\Omega_0 D_{11}^{-1} D_{12} z_3 - \Omega_0 D_{11}^{-1} \Omega_0 z_1$$
$$\dot{z}_2 = \Omega_+ z_3$$
$$\dot{z}_3 = D_{12}^T D_{11}^{-1} \Omega_0 z_1 - \Omega_+ z_2 + \left(D_{12}^T D_{22}^{-1} D_{12} - D_{22} \right) z_3 \,,$$

which yields to the following linearization:

$$\dot{z} = Az,$$
$$A = \begin{bmatrix} -\Omega_0 D_{11}^{-1} \Omega_0 & 0_{n_0,m} & -\Omega_+ D_{11}^{-1} D_{12} \\ 0_{m,n_0} & 0_{m,m} & \Omega_+ \\ D_{12}^T D_{11}^{-1} \Omega_+ & -\Omega_+ & D_{12}^T D_{11}^{-1} D_{12} - D_{22} \end{bmatrix} .$$

If $D_{12} = 0$, then the linearization

$$A_+ = \begin{bmatrix} 0 & \Omega_+ \\ -\Omega_+ & -D_{22} \end{bmatrix} .$$

The standard linearization used I. Nakić 2002, K. Brabender 1998, Müeller and Gürgöze 1991, Benner, Tomljanović Truhar 2010, 2011.

Singular mass matrix

The ODE system

$$\dot{z}_1 = -\Omega_0 D_{11}^{-1} D_{12} z_3 - \Omega_0 D_{11}^{-1} \Omega_0 z_1$$
$$\dot{z}_2 = \Omega_+ z_3$$
$$\dot{z}_3 = D_{12}^T D_{11}^{-1} \Omega_0 z_1 - \Omega_+ z_2 + \left(D_{12}^T D_{22}^{-1} D_{12} - D_{22} \right) z_3 \,,$$

which yields to the following linearization:

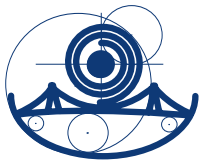
$$\dot{z} = Az,$$
$$A = \begin{bmatrix} -\Omega_0 D_{11}^{-1} \Omega_0 & 0_{n_0,m} & -\Omega_+ D_{11}^{-1} D_{12} \\ 0_{m,n_0} & 0_{m,m} & \Omega_+ \\ D_{12}^T D_{11}^{-1} \Omega_+ & -\Omega_+ & D_{12}^T D_{11}^{-1} D_{12} - D_{22} \end{bmatrix} .$$

If $D_{12} = 0$, then the linearization

$$A_+ = \begin{bmatrix} 0 & \Omega_+ \\ -\Omega_+ & -D_{22} \end{bmatrix} .$$

The standard linearization used I. Nakić 2002, K. Brabender 1998, Müeller and Gürgöze 1991, Benner, Tomljanović Truhar 2010, 2011.

Singular mass matrix



The ODE system

$$\dot{z}_1 = -\Omega_0 D_{11}^{-1} D_{12} z_3 - \Omega_0 D_{11}^{-1} \Omega_0 z_1$$
$$\dot{z}_2 = \Omega_+ z_3$$
$$\dot{z}_3 = D_{12}^T D_{11}^{-1} \Omega_0 z_1 - \Omega_+ z_2 + \left(D_{12}^T D_{22}^{-1} D_{12} - D_{22} \right) z_3 \,,$$

which yields to the following linearization:

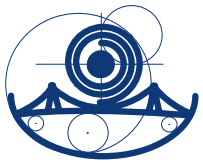
$$\dot{z} = Az,$$
$$A = \begin{bmatrix} -\Omega_0 D_{11}^{-1} \Omega_0 & 0_{n_0,m} & -\Omega_+ D_{11}^{-1} D_{12} \\ 0_{m,n_0} & 0_{m,m} & \Omega_+ \\ D_{12}^T D_{11}^{-1} \Omega_+ & -\Omega_+ & D_{12}^T D_{11}^{-1} D_{12} - D_{22} \end{bmatrix} .$$

If $D_{12} = 0$, then the linearization

$$A_+ = \begin{bmatrix} 0 & \Omega_+ \\ -\Omega_+ & -D_{22} \end{bmatrix} .$$

The standard linearization used I. Nakić 2002, K. Brabender 1998, Müeller and Gürgöze 1991, Benner, Tomljanović Truhar 2010, 2011.

Singular mass matrix



The ODE system

$$\dot{z}_1 = -\Omega_0 D_{11}^{-1} D_{12} z_3 - \Omega_0 D_{11}^{-1} \Omega_0 z_1$$
$$\dot{z}_2 = \Omega_+ z_3$$
$$\dot{z}_3 = D_{12}^T D_{11}^{-1} \Omega_0 z_1 - \Omega_+ z_2 + \left(D_{12}^T D_{22}^{-1} D_{12} - D_{22} \right) z_3 \,,$$

which yields to the following linearization:

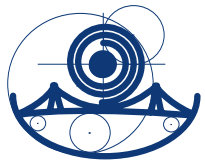
$$\dot{z} = Az,$$
$$A = \begin{bmatrix} -\Omega_0 D_{11}^{-1} \Omega_0 & 0_{n_0,m} & -\Omega_+ D_{11}^{-1} D_{12} \\ 0_{m,n_0} & 0_{m,m} & \Omega_+ \\ D_{12}^T D_{11}^{-1} \Omega_+ & -\Omega_+ & D_{12}^T D_{11}^{-1} D_{12} - D_{22} \end{bmatrix} .$$

If $D_{12} = 0$, then the linearization

$$A_+ = \begin{bmatrix} 0 & \Omega_+ \\ -\Omega_+ & -D_{22} \end{bmatrix} .$$

The standard linearization used I. Nakić 2002, K. Brabender 1998, Müeller and Gürgöze 1991, Benner, Tomljanović Truhar 2010, 2011.

Singular mass matrix



The ODE system

$$\begin{aligned}
\dot{z}_1 &= -\Omega_0 D_{11}^{-1} D_{12} z_3 - \Omega_0 D_{11}^{-1} \Omega_0 z_1 \\
\dot{z}_2 &= \Omega_+ z_3 \\
\dot{z}_3 &= D_{12}^T D_{11}^{-1} \Omega_0 z_1 - \Omega_+ z_2 + \left(D_{12}^T D_{22}^{-1} D_{12} - D_{22} \right) z_3 \,,
\end{aligned}$$

which yields to the following linearization:

$$\dot{z} = Az,$$

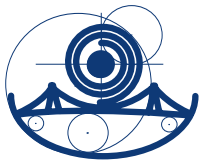
$$A = \begin{bmatrix} -\Omega_0 D_{11}^{-1} \Omega_0 & 0_{n_0,m} & -\Omega_+ D_{11}^{-1} D_{12} \\ 0_{m,n_0} & 0_{m,m} & \Omega_+ \\ D_{12}^T D_{11}^{-1} \Omega_+ & -\Omega_+ & D_{12}^T D_{11}^{-1} D_{12} - D_{22} \end{bmatrix} .$$

If $D_{12} = 0$, then the linearization

$$A_+ = \begin{bmatrix} 0 & \Omega_+ \\ -\Omega_+ & -D_{22} \end{bmatrix} .$$

The standard linearization used I. Nakić 2002, K. Brabender 1998, Müeller and Gürgöze 1991, Benner, Tomljanović Truhar 2010, 2011.

Singular mass matrix – damping optimization



Here usually, $C_{in} = 0$ ($\alpha = 0$ or $\alpha \ll 1$). We optimize only v in $D_{ex} \doteq v\Phi^T C_{ex}\Phi$

$$\mathrm{tr}(ZX) = \min\,,$$

For the $n_0 \doteq \dim(\mathcal{N}(M)) = 1$ and $\alpha = 0$ (no internal damping) we have

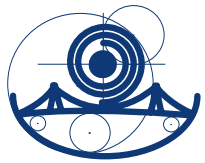
$$f(v) = \mathrm{tr}(X) = 2 \cdot \sum_i^{n-1} (X_{11})_{ii} + \sum_i^{n-1} \frac{\omega_n}{d_2} \frac{\eta_i}{\omega_i} (d_1)_i + v \frac{d_2^2}{2\omega_n^2}(2n-1)\,.$$

Thus, we approximate the trace function with

$$f(v) = \frac{a_0}{v} + b_0 v + c_0,$$

where the constants a_0, b_0 and c_0 are obtained from $\mathrm{tr}(X)(v_i)$, $i = 1, 2, 3$.

Singular mass matrix – damping optimization



Here usually, $C_{in} = 0$ ($\alpha = 0$ or $\alpha \ll 1$). We optimize only v in $D_{ex} \doteq v\Phi^T C_{ex}\Phi$

$$\mathrm{tr}(ZX) = \min\,,$$

For the $n_0 \doteq \dim(\mathcal{N}(M)) = 1$ and $\alpha = 0$ (no internal damping) we have

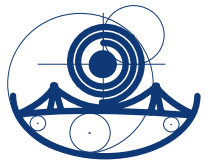
$$f(v) = \mathrm{tr}(X) = 2 \cdot \sum_i^{n-1} (X_{11})_{ii} + \sum_i^{n-1} \frac{\omega_n}{d_2}\frac{\eta_i}{\omega_i}\,(d_1)_i + v\,\frac{d_2^2}{2\omega_n^2}(2n-1)\,.$$

Thus, we approximate the trace function with

$$f(v) = \frac{a_0}{v} + b_0 v + c_0,$$

where the constants a_0, b_0 and c_0 are obtained from $\mathrm{tr}(X)(v_i)$, $i = 1, 2, 3$.

Singular mass matrix – damping optimization



Here usually, $C_{in} = 0$ ($\alpha = 0$ or $\alpha \ll 1$). We optimize only v in $D_{ex} \doteq v\Phi^T C_{ex}\Phi$

$$\mathrm{tr}(ZX) = \min\,,$$

For the $n_0 \doteq \dim(\mathcal{N}(M)) = 1$ and $\alpha = 0$ (no internal damping) we have

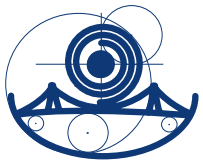
$$f(v) = \mathrm{tr}(X) = 2 \cdot \sum_i^{n-1} (X_{11})_{ii} + \sum_i^{n-1} \frac{\omega_n}{d_2}\frac{\eta_i}{\omega_i}\,(d_1)_i + v\,\frac{d_2^2}{2\omega_n^2}(2n-1)\,.$$

Thus, we approximate the trace function with

$$f(v) = \frac{a_0}{v} + b_0 v + c_0,$$

where the constants a_0, b_0 and c_0 are obtained from $\mathrm{tr}(X)(v_i)$, $i = 1, 2, 3$.

Singular mass matrix – damping optimization



Here usually, $C_{in} = 0$ ($\alpha = 0$ or $\alpha \ll 1$). We optimize only v in $D_{ex} \doteq v\Phi^T C_{ex}\Phi$

$$\mathrm{tr}(ZX) = \min\,,$$

For the $n_0 \doteq \dim(\mathcal{N}(M)) = 1$ and $\alpha = 0$ (no internal damping) we have

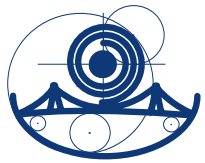
$$f(v) = \mathrm{tr}(X) = 2 \cdot \sum_i^{n-1} (X_{11})_{ii} + \sum_i^{n-1} \frac{\omega_n}{d_2}\frac{\eta_i}{\omega_i}\,(d_1)_i + v\,\frac{d_2^2}{2\omega_n^2}(2n-1)\,.$$

Thus, we approximate the trace function with

$$f(v) = \frac{a_0}{v} + b_0 v + c_0,$$

where the constants a_0, b_0 and c_0 are obtained from $\mathrm{tr}(X)(v_i)$, $i = 1, 2, 3$.

Singular mass matrix – damping optimization



Here usually, $C_{in} = 0$ ($\alpha = 0$ or $\alpha \ll 1$). We optimize only v in $D_{ex} \doteq v\Phi^T C_{ex}\Phi$

$$\mathrm{tr}(ZX) = \min\,,$$

For the $n_0 \doteq \dim(\mathcal{N}(M)) = 1$ and $\alpha = 0$ (no internal damping) we have

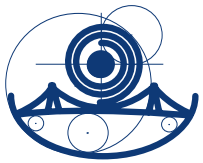
$$f(v) = \mathrm{tr}(X) = 2 \cdot \sum_i^{n-1} (X_{11})_{ii} + \sum_i^{n-1} \frac{\omega_n}{d_2}\frac{\eta_i}{\omega_i}\,(d_1)_i + v\,\frac{d_2^2}{2\omega_n^2}(2n-1)\,.$$

Thus, we approximate the trace function with

$$f(v) = \frac{a_0}{v} + b_0 v + c_0,$$

where the constants a_0, b_0 and c_0 are obtained from $\mathrm{tr}(X)(v_i)$, $i = 1, 2, 3$.

Summary



We have shown:

- **the novel approach for damping optimization for mechanical systems**

It is based on the new:

position search, base on one frequency damping optimization

viscosity optimization using new approximation for the trace function

error bound for the approximate solution of Lyapunov equation

- **singular mass case**

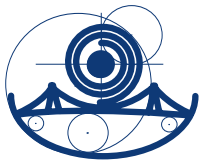
we presented new:

linearization

formula for the trace of solution of Lyapunov equation 1D case

approximation function for the trace for 2D or 3D case

Summary



We have shown:

- **the novel approach for damping optimization for mechanical systems**

It is based on the new:

position search, base on one frequency damping optimization

viscosity optimization using new approximation for the trace function

error bound for the approximate solution of Lyapunov equation

- **singular mass case**

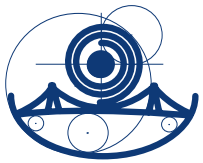
we presented new:

linearization

formula for the trace of solution of Lyapunov equation 1D case

approximation function for the trace for 2D or 3D case

Summary



We have shown:

- **the novel approach for damping optimization for mechanical systems**

It is based on the new:

position search, base on one frequency damping optimization

viscosity optimization using new approximation for the trace function

error bound for the approximate solution of Lyapunov equation

- **singular mass case**

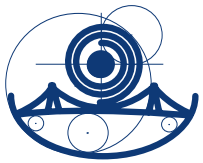
we presented new:

linearization

formula for the trace of solution of Lyapunov equation 1D case

approximation function for the trace for 2D or 3D case

Summary



We have shown:

- **the novel approach for damping optimization for mechanical systems**

It is based on the new:

position search, base on one frequency damping optimization

viscosity optimization using new approximation for the trace function

error bound for the approximate solution of Lyapunov equation

- **singular mass case**

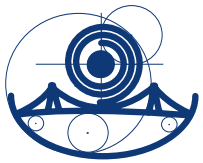
we presented new:

linearization

formula for the trace of solution of Lyapunov equation 1D case

approximation function for the trace for 2D or 3D case

Summary



We have shown:

- **the novel approach for damping optimization for mechanical systems**

It is based on the new:

position search, base on one frequency damping optimization

viscosity optimization using new approximation for the trace function

error bound for the approximate solution of Lyapunov equation

- **singular mass case**

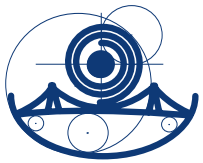
we presented new:

linearization

formula for the trace of solution of Lyapunov equation 1D case

approximation function for the trace for 2D or 3D case

Summary



We have shown:

- **the novel approach for damping optimization for mechanical systems**

It is based on the new:

position search, base on one frequency damping optimization

viscosity optimization using new approximation for the trace function

error bound for the approximate solution of Lyapunov equation

- **singular mass case**

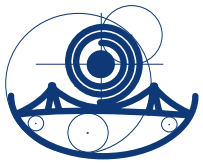
we presented new:

linearization

formula for the trace of solution of Lyapunov equation 1D case

approximation function for the trace for 2D or 3D case

Summary



We have shown:

- **the novel approach for damping optimization for mechanical systems**

It is based on the new:

position search, base on one frequency damping optimization

viscosity optimization using new approximation for the trace function

error bound for the approximate solution of Lyapunov equation

- **singular mass case**

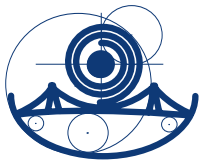
we presented new:

linearization

formula for the trace of solution of Lyapunov equation 1D case

approximation function for the trace for 2D or 3D case

Summary



We have shown:

- **the novel approach for damping optimization for mechanical systems**

It is based on the new:

position search, base on one frequency damping optimization

viscosity optimization using new approximation for the trace function

error bound for the approximate solution of Lyapunov equation

- **singular mass case**

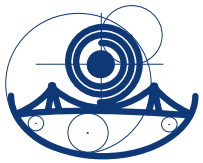
we presented new:

linearization

formula for the trace of solution of Lyapunov equation 1D case

approximation function for the trace for 2D or 3D case

Summary



We have shown:

- **the novel approach for damping optimization for mechanical systems**

It is based on the new:

position search, base on one frequency damping optimization

viscosity optimization using new approximation for the trace function

error bound for the approximate solution of Lyapunov equation

- **singular mass case**

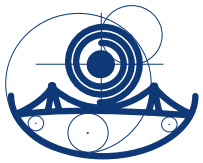
we presented new:

linearization

formula for the trace of solution of Lyapunov equation 1D case

approximation function for the trace for 2D or 3D case

Summary



We have shown:

- **the novel approach for damping optimization for mechanical systems**

It is based on the new:

position search, base on one frequency damping optimization

viscosity optimization using new approximation for the trace function

error bound for the approximate solution of Lyapunov equation

- **singular mass case**

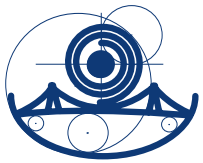
we presented new:

linearization

formula for the trace of solution of Lyapunov equation 1D case

approximation function for the trace for 2D or 3D case

Summary



We have shown:

- **the novel approach for damping optimization for mechanical systems**

It is based on the new:

position search, base on one frequency damping optimization

viscosity optimization using new approximation for the trace function

error bound for the approximate solution of Lyapunov equation

- **singular mass case**

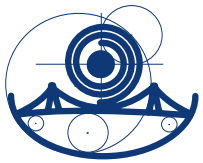
we presented new:

linearization

formula for the trace of solution of Lyapunov equation 1D case

approximation function for the trace for 2D or 3D case

Summary



We have shown:

- **the novel approach for damping optimization for mechanical systems**

It is based on the new:

position search, base on one frequency damping optimization

viscosity optimization using new approximation for the trace function

error bound for the approximate solution of Lyapunov equation

- **singular mass case**

we presented new:

linearization

formula for the trace of solution of Lyapunov equation 1D case

approximation function for the trace for 2D or 3D case

Thank you for your attention!

Thank you for your attention!