Fixed-Structure Discrete-Time $\mathcal{H}_\infty$ Controller Synthesis with HIFOO

Andrey Popov[1] Herbert Werner[1] Marc Millstone[2]

[1] Institute of Control Systems
Hamburg University of Technology
Germany

[2] Courant Institute of Mathematical Sciences
New York University
USA

16.12.2010
49th IEEE Conference on Decision and Control

Contents

Contents

Motivation - 1: Simple Controllers

Fixed-structure controllers are

+ computationally efficient
+ economic/energy efficient
+ easy to implement and verify

But:

Motivation - 1: Simple Controllers

Fixed-structure controllers are

+ computationally efficient
+ economic/energy efficient
+ easy to implement and verify

But:

– generally a non-convex problem
– convex reformulation only for SISO systems (central polynomial approach) [Henrion, 2005], [Khatibi & Karimi, 2010]

$\sim$ variety of methods proposed for continuous-time problems

Motivation - 2: Discrete-Time Synthesis

+ digital controllers
+ black-box identification
+ high performance requirements/large sampling times T

– Performance is lost by continuous-time synthesis - shown in a moment

Motivation - 2: Discrete-Time Synthesis

+ digital controllers
+ black-box identification
+ high performance requirements/large sampling times T
− Performance is lost by continuous-time synthesis - shown in a moment

Often: bilinear transformation + continuous-time synthesis

$$z = \frac{2+Ts}{2-Ts}, \qquad \text{synthesis} \qquad s = \frac{2}{T}\frac{z-1}{z+1}$$

Motivation - 2: Discrete-Time Synthesis

+ digital controllers
+ black-box identification
+ high performance requirements/large sampling times T
– Performance is lost by continuous-time synthesis - shown in a moment

Often: bilinear transformation + continuous-time synthesis

$$z = \frac{2+Ts}{2-Ts}, \qquad \text{synthesis} \qquad s = \frac{2}{T}\frac{z-1}{z+1}$$

Problems

– frequency warping
– performance loss by constrained sampling rate

Motivation - 3: Gradient-Based Techniques

- Gradient-based fixed-structure $\mathcal{H}_\infty$ synthesis [Burke et.al., 2006], [Apkarian & Noll, 2006]
- Offer
  - fast convergence
  - local optimum
  - good results [Gumussoy et.al., 2008]
- HIFOO is open source

Problem Formulation

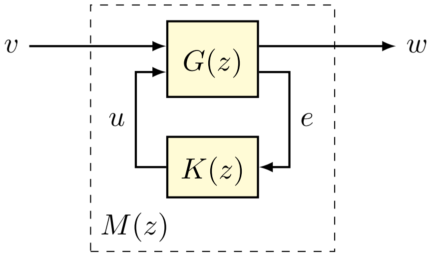
Given a generalized plant $G(z)$
find a controller $K(z) \in \mathbf{K}(z)$
such that

$$\|M(z)\|_\infty$$

is minimized,

where

$$M(z) = \mathcal{F}_L\left(G(z),\, K(z)\right)$$



Contents

HIFOO Algorithm

1. Initialization - random controllers

2. Stabilization

$$\text{find } K(z) \in \mathbf{K}(z) \text{ s.t. } M(z) = \mathcal{F}_L(G(z), K(z)) \quad \text{is stable}$$

3. Closed-loop $\mathcal{H}_\infty$ norm minimization

$$\underset{K(z) \in \mathbf{K}(z)}{\text{minimize}} \|\mathcal{F}_L(G(z), K(z))\|_\infty$$

HIFOO Algorithm

1. Initialization - random controllers

2. Stabilization

find $K(z) \in \mathbf{K}(z)$ s.t. $M(z) = \mathcal{F}_L(G(z), K(z))$ is stable

3. Closed-loop $\mathcal{H}_\infty$ norm minimization

$$\underset{K(z) \in \mathbf{K}(z)}{\text{minimize}} \|\mathcal{F}_L(G(z), K(z))\|_\infty$$

HIFOO Algorithm

1. Initialization - random controllers

2. Stabilization

   find $K(z) \in \mathbf{K}(z)$ s.t. $M(z) = \mathcal{F}_L(G(z),\, K(z))$ is stable

3. Closed-loop $\mathcal{H}_\infty$ norm minimization

$$\underset{K(z)\in\mathbf{K}(z)}{\text{minimize}} \|\mathcal{F}_L(G(z),\, K(z))\|_\infty$$

Stabilization

Continuous-time

$$\underset{K(z)}{\text{minimize}}\ \alpha(A)$$

$$\text{until } \alpha(A) < 0$$

α - spectral abscissa
A - closed-loop system matrix

Stabilization

Continuous-time

$$\underset{K(z)}{\text{minimize}}\ \alpha(A)$$
$$\text{until } \alpha(A) < 0$$

α - spectral abscissa
A - closed-loop system matrix

Gradient-steps

1. Given $K(z)$, compute spec(A) and find λ_k at which α is attained
2. Compute gradient of α w.r.t. A
3. Compute gradient of A w.r.t. $K(z)$
4. Update $K(z)$

Stabilization

Continuous-time

$$\underset{K(z)}{\text{minimize}}\ \alpha(A)$$

$$\text{until } \alpha(A) < 0$$

α - spectral abscissa

A - closed-loop system matrix

Discrete-time

$$\underset{K(z)}{\text{minimize}}\ \rho(A)$$

$$\text{until } \rho(A) < 1$$

ρ - spectral radius

Gradient-steps

1. Given $K(z)$, compute spec(A) and find λ_k at which α is attained
2. Compute gradient of α w.r.t. A
3. Compute gradient of A w.r.t. $K(z)$
4. Update $K(z)$

Stabilization

Continuous-time

$$\underset{K(z)}{\text{minimize}}\ \alpha(A)$$

$$\text{until } \alpha(A) < 0$$

α - spectral abscissa

A - closed-loop system matrix

Discrete-time

$$\underset{K(z)}{\text{minimize}}\ \rho(A)$$

$$\text{until } \rho(A) < 1$$

ρ - spectral radius

Gradient-steps

1. Given $K(z)$, compute spec(A) and find λ_k at which ρ is attained
2. Compute gradient of ρ w.r.t. A
3. Compute gradient of A w.r.t. $K(z)$
4. Update $K(z)$

Gradient Computation

Let $A(t) = A_0 + Pt,\ A \in \mathbb{R}^{n\times n}$ and small t.

Let $\lambda_k \in \mathrm{spec}(A)$ has algebraic multiplicity one and eigenvectors y (left) and x (right).

$$\frac{d\lambda_k}{dt} = \frac{y^\star P x}{y^\star x} \qquad \text{[Horn \& Johnson, 1985]}$$

Let $\lambda_k = \mathcal{R} + \mathrm{j}\mathcal{I};\quad \ell_k = \begin{bmatrix}\mathcal{R} & \mathcal{I}\end{bmatrix}$.

$$\rho = |\lambda_k| = \sqrt{\mathcal{R}^2 + \mathcal{I}^2} = |\ell_k|$$

$$\frac{\partial \rho}{\partial \ell_k} = \begin{bmatrix}\frac{\mathcal{R}}{\sqrt{\mathcal{R}^2+\mathcal{I}^2}} & \frac{\mathcal{I}}{\sqrt{\mathcal{R}^2+\mathcal{I}^2}}\end{bmatrix} = \frac{\ell_k}{|\lambda_k|} \qquad \Leftrightarrow \qquad \frac{\partial \rho}{\partial \lambda_k} = \frac{\lambda_k}{|\lambda_k|}$$

Hence

$$\frac{d\rho(A)}{dt} = \mathrm{Re}\left\{\frac{\bar{\lambda}_k}{|\lambda_k|}\frac{y^\star P x}{y^\star x}\right\}$$

Gradient Computation

Let $A(t) = A_0 + Pt,\ A \in \mathbb{R}^{n \times n}$ and small t.
Let $\lambda_k \in \text{spec}(A)$ has algebraic multiplicity one and eigenvectors y (left) and x (right).

$$\frac{d\lambda_k}{dt} = \frac{y^\star P x}{y^\star x}$$ [Horn & Johnson, 1985]

Let $\lambda_k = \mathcal{R} + \mathrm{j}\mathcal{I};\quad \ell_k = \begin{bmatrix}\mathcal{R} & \mathcal{I}\end{bmatrix}$.

$$\rho = |\lambda_k| = \sqrt{\mathcal{R}^2 + \mathcal{I}^2} = |\ell_k|$$

$$\frac{\partial \rho}{\partial \ell_k} = \begin{bmatrix}\frac{\mathcal{R}}{\sqrt{\mathcal{R}^2+\mathcal{I}^2}} & \frac{\mathcal{I}}{\sqrt{\mathcal{R}^2+\mathcal{I}^2}}\end{bmatrix} = \frac{\ell_k}{|\lambda_k|} \quad \Leftrightarrow \quad \frac{\partial \rho}{\partial \lambda_k} = \frac{\lambda_k}{|\lambda_k|}$$

Hence

$$\frac{d\rho(A)}{dt} = \mathrm{Re}\left\{\frac{\bar{\lambda}_k}{|\lambda_k|}\frac{y^\star P x}{y^\star x}\right\}$$

Gradient Computation

Let $A(t) = A_0 + Pt,\ \ A \in \mathbb{R}^{n\times n}$ and small t.
Let $\lambda_k \in \text{spec}(A)$ has algebraic multiplicity one and eigenvectors y (left) and x (right).

$$\frac{d\lambda_k}{dt} = \frac{y^\star P x}{y^\star x}$$ [Horn & Johnson, 1985]

Let $\lambda_k = \mathcal{R} + \mathrm{j}\mathcal{I};\quad \ell_k = \begin{bmatrix}\mathcal{R} & \mathcal{I}\end{bmatrix}$.

$$\rho = |\lambda_k| = \sqrt{\mathcal{R}^2 + \mathcal{I}^2} = |\ell_k|$$

$$\frac{\partial \rho}{\partial \ell_k} = \begin{bmatrix}\frac{\mathcal{R}}{\sqrt{\mathcal{R}^2+\mathcal{I}^2}} & \frac{\mathcal{I}}{\sqrt{\mathcal{R}^2+\mathcal{I}^2}}\end{bmatrix} = \frac{\ell_k}{|\lambda_k|} \qquad \Leftrightarrow \qquad \frac{\partial \rho}{\partial \lambda_k} = \frac{\lambda_k}{|\lambda_k|}$$

Hence

$$\frac{d\rho(A)}{dt} = \mathrm{Re}\left\{\frac{\bar{\lambda}_k}{|\lambda_k|}\frac{y^\star P x}{y^\star x}\right\}$$

Gradient Computation

Let $A(t) = A_0 + Pt, \;\; A \in \mathbb{R}^{n\times n}$ and small t.
Let $\lambda_k \in \text{spec}(A)$ has algebraic multiplicity one and eigenvectors y (left) and x (right).

$$\frac{d\lambda_k}{dt} = \frac{y^\star P x}{y^\star x} \qquad \text{[Horn \& Johnson, 1985]}$$

Let $\lambda_k = \mathcal{R} + \mathrm{j}\mathcal{I}; \quad \ell_k = \begin{bmatrix}\mathcal{R} & \mathcal{I}\end{bmatrix}$.

$$\rho = |\lambda_k| = \sqrt{\mathcal{R}^2 + \mathcal{I}^2} = |\ell_k|$$

$$\frac{\partial\rho}{\partial\ell_k} = \left[\frac{\mathcal{R}}{\sqrt{\mathcal{R}^2+\mathcal{I}^2}} \quad \frac{\mathcal{I}}{\sqrt{\mathcal{R}^2+\mathcal{I}^2}}\right] = \frac{\ell_k}{|\lambda_k|} \qquad \Leftrightarrow \qquad \frac{\partial\rho}{\partial\lambda_k} = \frac{\lambda_k}{|\lambda_k|}$$

Hence

$$\frac{d\rho(A)}{dt} = \mathrm{Re}\left\{\frac{\bar{\lambda}_k}{|\lambda_k|}\frac{y^\star P x}{y^\star x}\right\}$$

Gradient Computation

Let $A(t) = A_0 + Pt, \;\; A \in \mathbb{R}^{n\times n}$ and small t.
Let $\lambda_k \in$ spec(A) has algebraic multiplicity one and eigenvectors y (left) and x (right).

$$\frac{d\lambda_k}{dt} = \frac{y^\star P x}{y^\star x} \qquad \text{[Horn \& Johnson, 1985]}$$

Let $\lambda_k = \mathcal{R} + \mathrm{j}\mathcal{I}; \quad \ell_k = \begin{bmatrix}\mathcal{R} & \mathcal{I}\end{bmatrix}$.

$$\rho = |\lambda_k| = \sqrt{\mathcal{R}^2 + \mathcal{I}^2} = |\ell_k|$$

$$\frac{\partial \rho}{\partial \ell_k} = \left[\frac{\mathcal{R}}{\sqrt{\mathcal{R}^2+\mathcal{I}^2}} \quad \frac{\mathcal{I}}{\sqrt{\mathcal{R}^2+\mathcal{I}^2}}\right] = \frac{\ell_k}{|\lambda_k|} \qquad \Leftrightarrow \qquad \frac{\partial \rho}{\partial \lambda_k} = \frac{\lambda_k}{|\lambda_k|}$$

Hence

$$\frac{d\rho(A)}{dt} = \mathrm{Re}\left\{\frac{\bar{\lambda}_k}{|\lambda_k|}\frac{y^\star P x}{y^\star x}\right\}$$

Gradient Computation

Let $A(t) = A_0 + Pt, \;\; A \in \mathbb{R}^{n\times n}$ and small t.
Let $\lambda_k \in$ spec(A) has algebraic multiplicity one and eigenvectors y (left) and x (right).

$$\frac{d\lambda_k}{dt} = \frac{y^\star P x}{y^\star x} \qquad \text{[Horn \& Johnson, 1985]}$$

Let $\lambda_k = \mathcal{R} + \mathrm{j}\mathcal{I}; \quad \ell_k = \begin{bmatrix}\mathcal{R} & \mathcal{I}\end{bmatrix}$.

$$\rho = |\lambda_k| = \sqrt{\mathcal{R}^2 + \mathcal{I}^2} = |\ell_k|$$

Theorem

For a matrix A with $\rho(A) = |\lambda_k|$ holds

$$\nabla_A \rho = \mathrm{Re}\left\{ \frac{\lambda_k}{|\lambda_k|} \frac{y x^\star}{y^\star x} \right\}$$

Gradient Steps

Stabilization

1. Compute spec(A) and find λ_k at which ρ is attained
2. Compute gradient of ρ w.r.t. A √
3. Compute gradient of A w.r.t. $K(z)$
4. Update $K(z)$

Gradient Steps

Stabilization

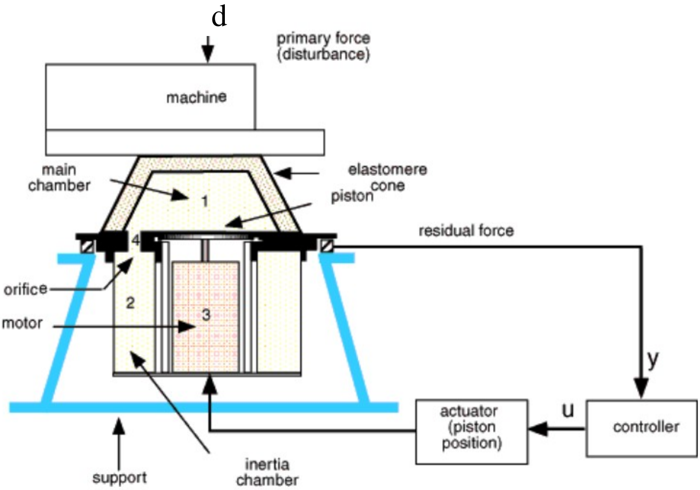
1. Compute spec(A) and find λ_k at which ρ is attained
2. Compute gradient of ρ w.r.t. A ✓
3. Compute gradient of A w.r.t. $K(z)$
4. Update $K(z)$

$\mathcal{H}_\infty$ **norm minimization**

1. Compute $f = \|M(z)\|_\infty = \bar{\sigma}(M_x)$
   where $M_x = C\left(Ie^{j\omega_x} - A\right)^{-1}B + D$
2. Compute the gradient of f w.r.t. M_x
3. Compute the gradient of M_x w.r.t. $K(z)$
4. Update $K(z)$

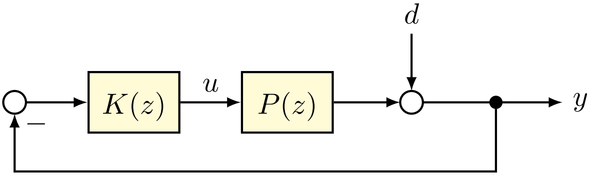
Contents

Active Suspension Benchmark System

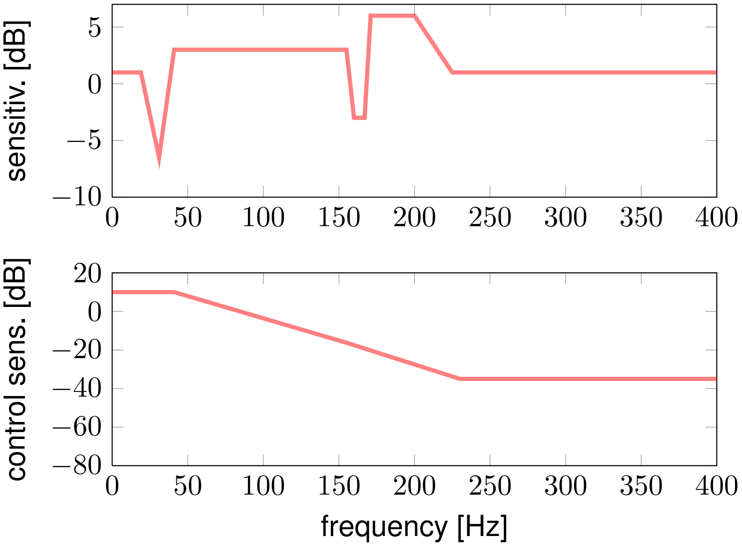


[Landau et.al. 2003]

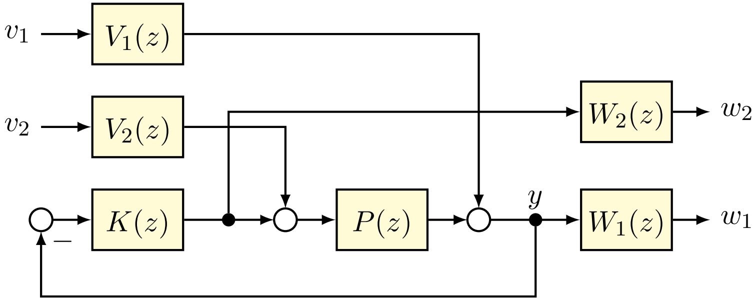
Active Suspension Benchmark System



Design Requirements

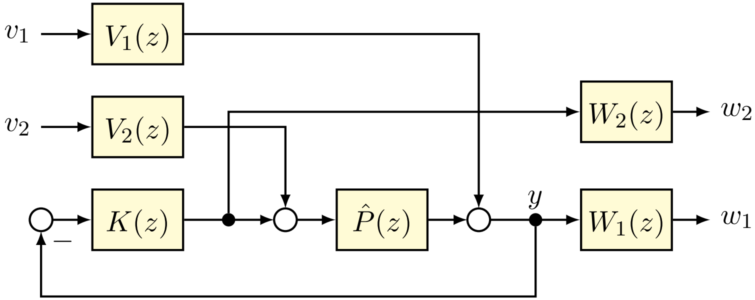


Active Suspension System - 4-Block Design



[Hol et. al. 2003]

Active Suspension System - 4-Block Design



[Hol et. al. 2003]

Cont. gain of zero at $0.5F_s$ $\Rightarrow$ $\hat{P}(z) = P(z)\frac{z+1}{z}$

Comparison

Controller design approach	k	γ	Computation time	Computation load
Full order	27	2.476	11.9 s	1
Balanced reduction	5	3.405	12.8 s	1.07
Curved line search	5	2.506	4h23m45s	1329.00
Cone complement.	5	2.630	14m33s	73.36
Hybrid Evol.-Algebr.	5	2.589	2m42s	13.61
Hybrid Evol.-Algebr.	2	2.596	1m25s	7.16
HIFOO - continuous	5	2.611	22.3 s	11.69
	2	2.473	41.4 s	21.68
	1	2.611	20.6 s	10.82
HIFOO - discrete	5	2.466	1m33s	51.87
	2	2.470	28.2 s	14.75
	1	2.611	7.7 s	4.17

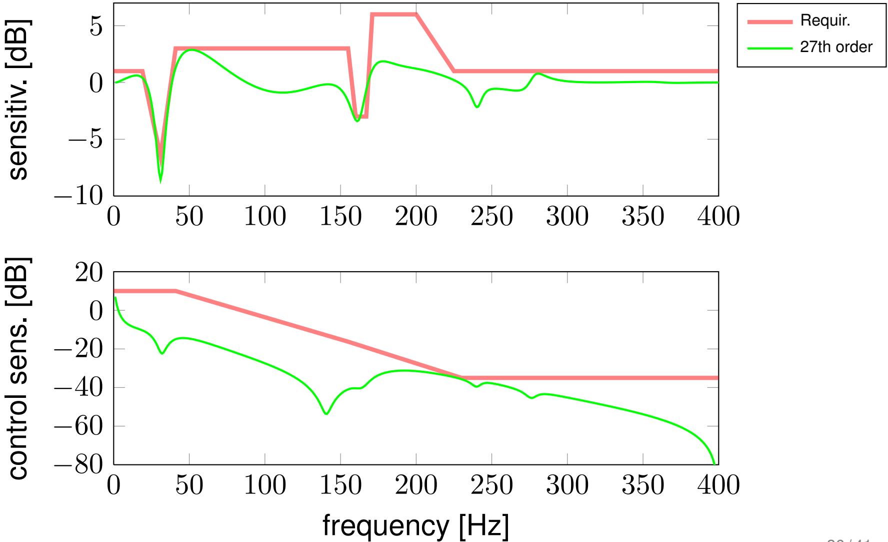
Comparison

Controller design approach	k	γ	Computation time	Computation load
Full order	27	2.476	11.9 s	1
Balanced reduction	5	3.405	12.8 s	1.07
Curved line search	5	2.506	4h23m45s	1329.00
Cone complement.	5	2.630	14m33s	73.36
Hybrid Evol.-Algebr.	5	2.589	2m42s	13.61
Hybrid Evol.-Algebr.	2	2.596	1m25s	7.16
HIFOO - continuous	5	2.611	22.3 s	11.69
	2	2.473	41.4 s	21.68
	1	2.611	20.6 s	10.82
HIFOO - discrete	5	2.466	1m33s	51.87
	2	2.470	28.2 s	14.75
	1	2.611	7.7 s	4.17

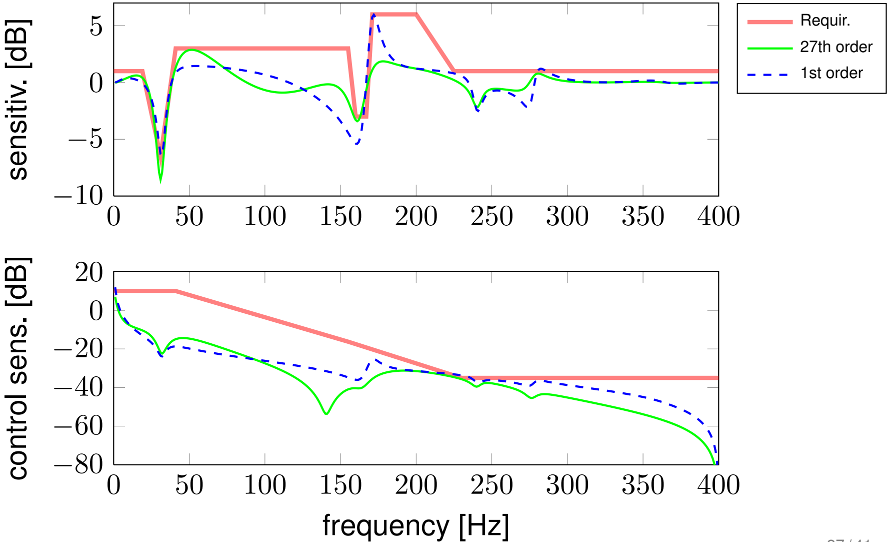
Comparison

Controller design approach	k	γ	Computation time	Computation load
Full order	27	2.476	11.9 s	1
Balanced reduction	5	3.405	12.8 s	1.07
Curved line search	5	2.506	4h23m45s	1329.00
Cone complement.	5	2.630	14m33s	73.36
Hybrid Evol.-Algebr.	5	2.589	2m42s	13.61
Hybrid Evol.-Algebr.	2	2.596	1m25s	7.16
HIFOO - continuous	5	2.611	22.3 s	11.69
	2	2.473	41.4 s	21.68
	1	2.611	20.6 s	10.82
HIFOO - discrete	5	2.466	1m33s	51.87
	2	2.470	28.2 s	14.75
	1	2.611	7.7 s	4.17

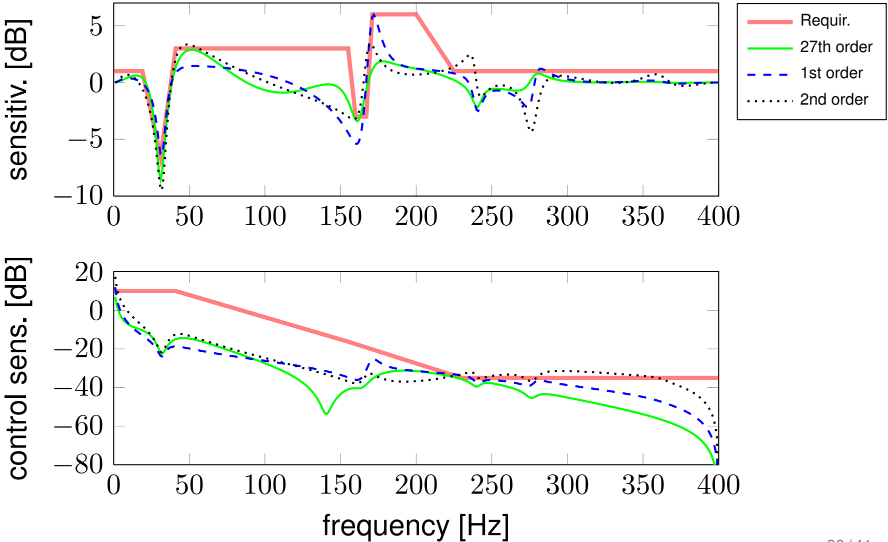
Performance



Performance



Performance



Contents

Conclusions

- Discrete-time fixed-structure $\mathcal{H}_\infty$ synthesis
- Allows direct structural restrictions
- Better results than previous techniques

Software

HIFOO+d

- `http://bitbucket.org/andrey.popov/hifoo-d`
- Discrete-time
- Replicated/repeated controller blocks:

$$K(z) \in \begin{bmatrix} \mathbf{K}_1(z) & & \mathbf{K}_1(z) \\ & \mathbf{K}_2(z) & \\ & & \mathbf{K}_2(z) \end{bmatrix}$$

  - redundant elements
  - symmetric systems
  - multi-agent systems [Popov & Werner, 2010]
  - μ-synthesis [Apkarian, 2010]

- features, not available in MATLAB 2010b or HIFOO 3.0
- GPL