NTNU

Innovation and Creativity

$\mathcal{H}_\infty$ Reduced Order Control

Numerical Implementability

Michael R. P. Ragazzon, Arnfinn A. Eielsen, J. Tommy Gravdahl

Department of Engineering Cybernetics, Norwegian University of Science and Technology

IFAC WC, Aug 27, 2014

Introduction

This presentation will discuss how model reduction affects stability and computational complexity of controllers.

Introduction

This presentation will discuss how model reduction affects stability and computational complexity of controllers.

— Consider nanopositioning as an example application.

1

Introduction

This presentation will discuss how model reduction affects stability and computational complexity of controllers.

- Consider nanopositioning as an example application.
- Challenging high resonant peaks.

1

Introduction

This presentation will discuss how model reduction affects stability and computational complexity of controllers.

- — Consider nanopositioning as an example application.
- — Challenging high resonant peaks.
- — For good performance, we employ higher-order model-matching based techniques, such as $\mathcal{H}_\infty$ control.

Introduction

This presentation will discuss how model reduction affects stability and computational complexity of controllers.

- Consider nanopositioning as an example application.
- Challenging high resonant peaks.
- For good performance, we employ higher-order model-matching based techniques, such as $\mathcal{H}_\infty$ control.
- Numerical solvers require sufficiently small step-size for stability.

NTNU
Innovation and Creativity

1

Introduction

This presentation will discuss how model reduction affects stability and computational complexity of controllers.

- Consider nanopositioning as an example application.
- Challenging high resonant peaks.
- For good performance, we employ higher-order model-matching based techniques, such as $\mathcal{H}_\infty$ control.
- Numerical solvers require sufficiently small step-size for stability.

Challenge

Controllers of high order can be challenging to implement in real-time due to computational complexity.

NTNU
Innovation and Creativity

Outline

Robust Controller Design

We design a robustly stable $\mathcal{H}_\infty$ controller which is used later for model reduction.
Procedure:

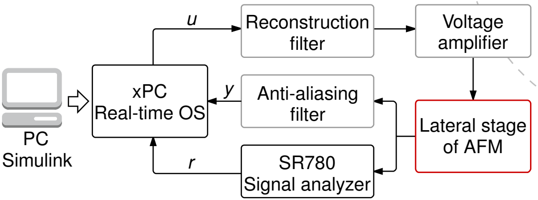
— System identification

— $\mathcal{H}_\infty$ controller design

— Robust stability

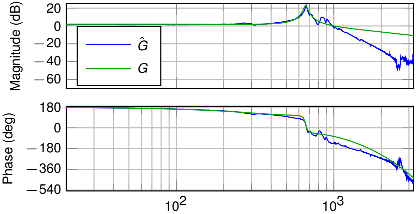
NTNU
Innovation and Creativity

Experimental Setup

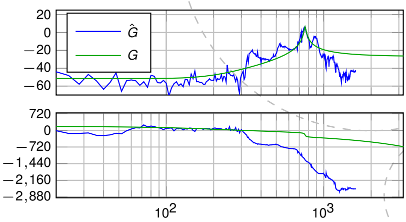


NTNU
Innovation and Creativity

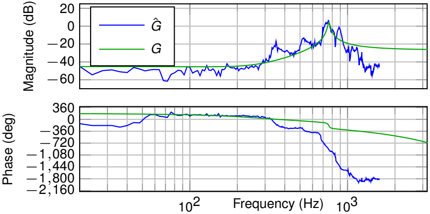
Frequency Response and Model Fit



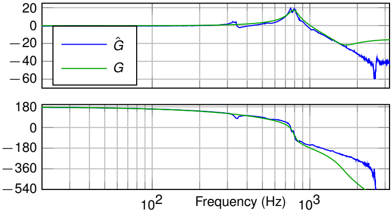
(a) G_{xx}



(b) G_{xy}



(c) G_{yx}



(d) G_{yy}

Controller Design

We have employed $\mathcal{H}_\infty$ mixed sensitivity control: Find K which satisfies

$$\min_K N(K) = \left\| \begin{matrix} W_1 S \\ W_2 T \\ W_3 KS \end{matrix} \right\|_\infty \qquad (1)$$

where

— W_1, W_2, and W_3 are user-defined frequency-dependent weightings.

— S, T is the closed-loop sensitivity and complementary sensitivity function respectively.

NTNU
Innovation and Creativity

Controller Design

We have employed $\mathcal{H}_\infty$ mixed sensitivity control: Find K which satisfies

$$\min_K N(K) = \left\| \begin{matrix} W_1 S \\ W_2 T \\ W_3 KS \end{matrix} \right\|_\infty \tag{1}$$

where

- W_1, W_2, and W_3 are user-defined frequency-dependent weightings.
- S, T is the closed-loop sensitivity and complementary sensitivity function respectively.

Resulting controller properties after specifying weights and solving (1).

	ω_{BS} [Hz]	ω_{BT} [Hz]	$\|WT\|_\infty$	Model order
PID	58.0	93.1	1.073	4th-order
$\mathcal{H}_\infty$	69.8	98.6	0.6717	18th-order

NTNU
Innovation and Creativity

Outline

Control Order Reduction

— Removing states from the controller model with minimal impact on the qualitative response.

— Purpose: To reduce the computational complexity of the controller.

— A common method is balanced residualization.

NTNU
Innovation and Creativity

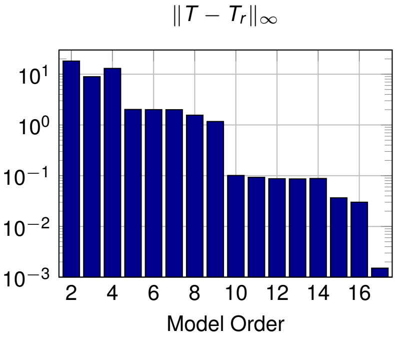
Control Order Reduction

- Removing states from the controller model with minimal impact on the qualitative response.
- Purpose: To reduce the computational complexity of the controller.
- A common method is balanced residualization.

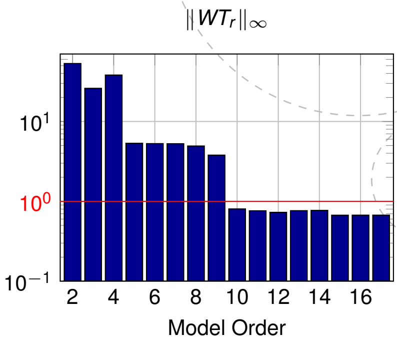
Next, we apply this method to our $\mathcal{H}_\infty$ controller to various reduced orders.

NTNU
Innovation and Creativity

Control Order Reduction Results



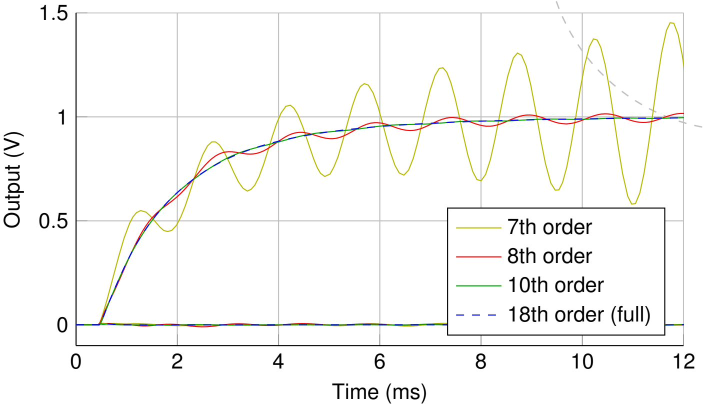
(a) Closed-loop error.
.



(b) Robustness.
Must be < 1 for robust stability.

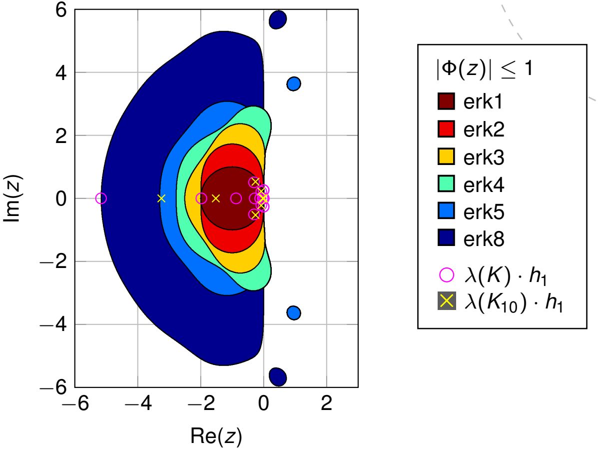
NTNU
Innovation and Creativity

Control Order Reduction Results



Simulated step-response

Control Order Reduction Results



Where $\Phi(z)$ is the stability function of the given ERK method,

$$\dot{y} = \lambda y \quad \Rightarrow \quad y_{n+1} = \Phi(h\lambda) y_n.$$

Control Order Reduction Results

Model Reduction and Numerical Stability

Not only does model reduction reduce the computational complexity of the controller, it may also allow for a larger step-size.

NTNU
Innovation and Creativity

Maximum Step-Size

Maximum step-size for a given explicit Runge-Kutta (ERK) method.

Controller Order	h_{max} [µs]					
	erk1	erk2	erk3	erk4	erk5	erk8
7 (unst.)	32.18	71.33	95.74	100.8	127.2	203.6
8	58.37	58.37	73.34	81.29	96.51	150.8
9	37.54	37.54	47.17	52.28	62.07	96.99
10	33.61	33.61	42.22	46.8	55.56	86.82
11	39.70	39.7	49.87	55.28	65.63	102.5
12	24.96	32.66	41.03	45.48	54.00	84.37
13	2.821	21.55	27.07	30.01	35.63	55.67
14	2.818	19.11	24.01	26.62	31.60	49.38
15	2.843	21.73	27.30	30.26	35.92	56.13
16	2.948	22.64	28.44	31.53	37.43	58.48
17	2.918	21.18	26.61	29.49	35.01	54.71
18 (full)	2.910	21.14	26.56	29.45	34.96	54.62
PID	80.00	80.00	100.5	111.4	132.3	206.7

Outline

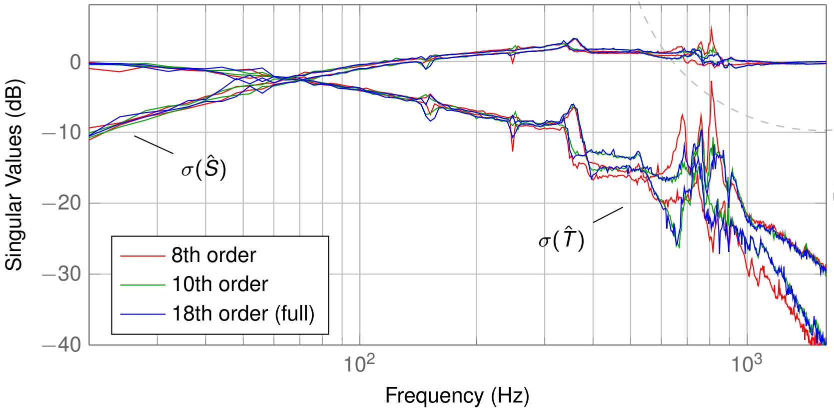
Experimental Results

Experiments were performed for two reasons:

1. To compare the reduced order controllers in terms of performance.
2. To record the average task execution time (TET).
   - A measurement of the computational complexity of the controller.
   - A lower limit on the chosen solver step-size.

NTNU
Innovation and Creativity

Closed-Loop Performance



Singular values of closed-loop frequency response, $\sigma(\hat{S})$ and $\sigma(\hat{T})$.

Computational Complexity

Average task execution time (TET). Step-size $h = 40\,\mu s$. Dash (-) unstable, not tested.

Controller Order	Average TET [µs]				
	erk1	erk2	erk3	erk5	erk8
≤ 7	-	-	-	-	-
8	-	10.25	10.43	11.10	13.64
9	-	-	10.55	11.43	14.26
10	-	-	10.71	11.64	14.99
11	-	-	10.89	11.91	15.79
12	-	-	11.07	12.36	16.79
13	-	-	-	-	17.68
14	-	-	-	-	18.82
15	-	-	-	-	19.61
16	-	-	-	-	20.91
17	-	-	-	-	22.61
18	-	-	-	-	20.11
PID	-	9.95	9.98	10.13	11.12

Computational Complexity

Average task execution time (TET). Step-size $h = 40\,\mu s$. Dash (-) unstable, not tested.

Controller Order	Average TET [µs]				
	erk1	erk2	erk3	erk5	erk8
≤7	-	-	-	-	-
8	-	10.25	10.43	11.10	13.64
9	-	-	10.55	11.43	14.26
10	-	-	10.71	11.64	14.99
11	-	-	10.89	11.91	15.79
12	-	-	11.07	12.36	16.79
13	-	-	-	-	17.68
14	-	-	-	-	18.82
15	-	-	-	-	19.61
16	-	-	-	-	20.91
17	-	-	-	-	22.61
18	-	-	-	-	20.11
PID	-	9.95	9.98	10.13	11.12

Computational Complexity

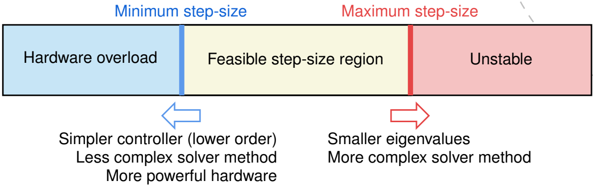
Average task execution time (TET). Step-size $h = 40\,\mu s$. Dash (-) unstable, not tested.

Controller Order	Average TET [µs]				
	erk1	erk2	erk3	erk5	erk8
≤7	-	-	-	-	-
8	-	10.25	10.43	11.10	13.64
9	-	-	10.55	11.43	14.26
10	-	-	10.71	11.64	14.99
11	-	-	10.89	11.91	15.79
12	-	-	11.07	12.36	16.79
13	-	-	-	-	17.68
14	-	-	-	-	18.82
15	-	-	-	-	19.61
16	-	-	-	-	20.91
17	-	-	-	-	22.61
18	-	-	-	-	20.11
PID	-	9.95	9.98	10.13	11.12

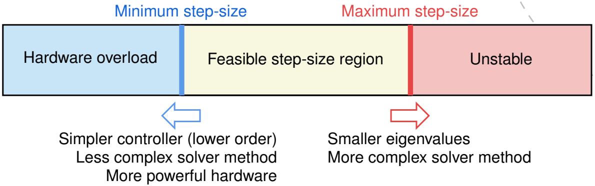
Outline

NTNU
Innovation and Creativity

Conclusion

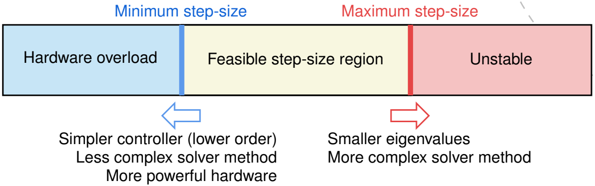

Minimum step-size
Maximum step-size
Hardware overload
Feasible step-size region
Unstable
Simpler controller (lower order)
Less complex solver method
More powerful hardware
Smaller eigenvalues
More complex solver method

Conclusion



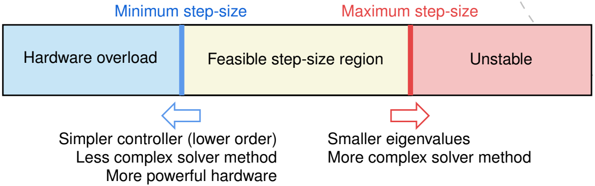
— Model reduction can reduce the size of the eigenvalues.

Conclusion



- Model reduction can reduce the size of the eigenvalues.
- Increased solver order increases the stability region, but also the execution time.

Conclusion



— Model reduction can reduce the size of the eigenvalues.
— Increased solver order increases the stability region, but also the execution time.
— Ultimately this is a tradeoff between moving an upper and lower limit for the step-size.

Conclusion

Suggested implementation procedure:

1. Design a controller and perform model reduction.
2. Find the maximum step-size providing stability for one or more chosen ERK methods.
3. An initial solver choice can be taken as the one which provides the best maximum step-size to computational complexity ratio. A rough estimate of the computational complexity can be found by simulations.
4. Run the controller at slightly below the maximum step-size, and reduce the step-size until the hardware is unable to perform the required calculations in time.
5. If no feasible step-size was found, one can reduce additional states or try another ERK method.

NTNU
Innovation and Creativity

That's all!
Any questions?

NTNU
Innovation and Creativity