Stabilization of Galerkin Reduced Order Models (ROMs) for LTI Systems Using Controllers

Irina Kalashnikova[1], Bart G. van Bloemen Waanders[1], Srinivasan Arunajatesan[2], Matthew F. Barone[2]

[1] Numerical Analysis & Applications Department, Sandia National Laboratories, Albuquerque, NM, U.S.A.

[2] Aerosciences Department, Sandia National Laboratories, Albuquerque NM, U.S.A.

SIAM Conference on Control and Its Applications (CT13)

San Diego, California

July 8 - 10, 2013

Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin company, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

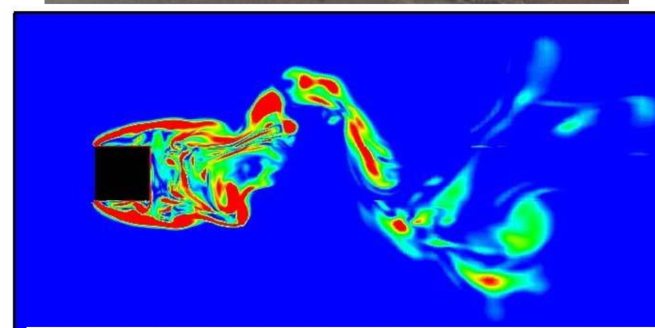
Why Develop a Reduced Order Model (ROM)?

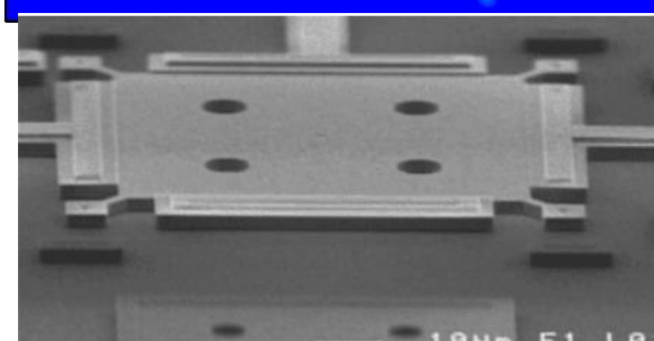
High-fidelity modeling is often too expensive for use in a design or analysis setting!

A **Reduced Order Model (ROM)** is a surrogate model that aims to capture the essential dynamics of a full numerical model but with far fewer dofs.

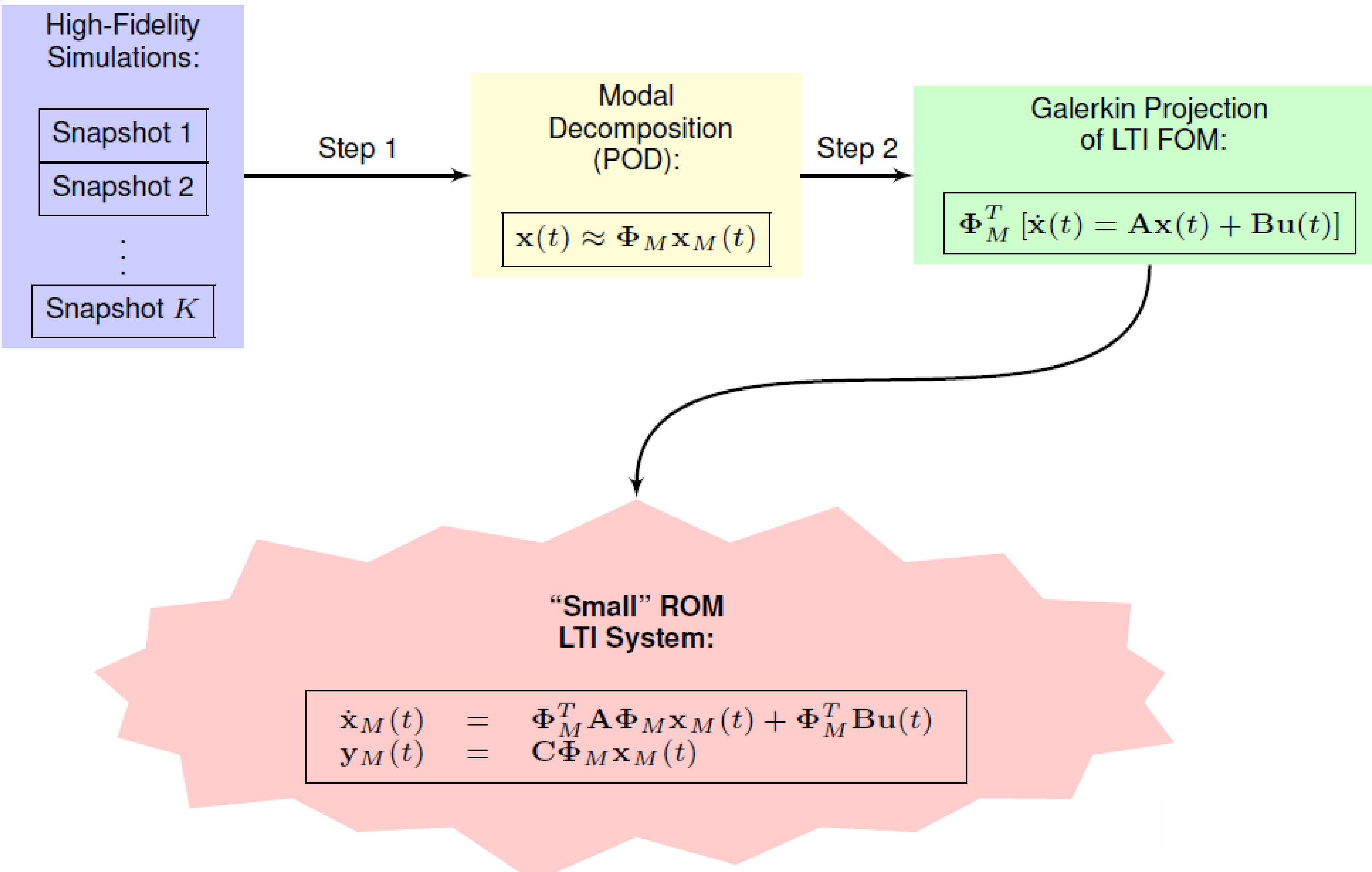
Example Applications of ROMs:

- Aeroelastic flutter analysis.
- System modeling for active flow control.
- Design/analysis of microelectromechanical devices and systems (MEMS) under electric actuation.





Proper Orthogonal Decomposition (POD)/Galerkin Method to Model Reduction



Step 1: Proper Orthogonal Decomposition (POD) Algorithm (a.k.a., "Method of Snapshots")

1. Collect K snapshots of a solution vector $\{\boldsymbol{x}^k : k = 1, \ldots, K\} \in \mathbb{R}^N$. Place these snapshots into the columns of a matrix $\boldsymbol{X}$ defined by

$$\boldsymbol{X} = (\boldsymbol{x}^1, \ldots, \boldsymbol{x}^K) \in \mathbb{R}^{NxK}$$

2. Select an inner product to build the reduced basis in, e.g., the L^2 inner product.

3. Compute the SVD of the matrix

$$\boldsymbol{Y} \equiv \frac{1}{K}\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{U\Sigma U}^T$$

4. A reduced POD basis of size $M << N$ is given by

$$\boldsymbol{\Phi}_M = (\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_M) = \boldsymbol{XU}(:, 1{:}M)$$

5. Orthonormalize and return the reduced basis.

N = # of dofs in high-fidelity simulation
K = # of snapshots
M = # of dofs in ROM
$(M << N, M << K)$

Truncated POD basis $\{\boldsymbol{\phi}_k : k = 1, \ldots, M\}$ describes more energy (on average) of the snapshot set than any other linear basis of the same dimension M.

Step 2: Galerkin Projection

LTI Full Order Model (FOM)

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$$
$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t)$$

$\sim O(N)$ dofs

1. Approximate solution as a linear combination of the POD modes obtained in Step 1:

$$\boldsymbol{x}(t) \approx \sum_{i=1}^{M} x_{M,i}(t)\boldsymbol{\phi}_i = \boldsymbol{\Phi}_M \boldsymbol{x}_M(t)$$

ROM solution

2. Project FOM system onto the modes $\boldsymbol{\phi}_i$ in some inner product, e.g., the L^2 inner product to obtain…

LTI Reduced Order Model (ROM)

$$\dot{\boldsymbol{x}}_M(t) = \boldsymbol{\Phi}_M{}^T\boldsymbol{A}\boldsymbol{\Phi}_M\boldsymbol{x}_M(t) + \boldsymbol{\Phi}_M{}^T\boldsymbol{B}\boldsymbol{u}(t)$$
$$\boldsymbol{y}_M(t) = \boldsymbol{C}\boldsymbol{\Phi}_M\boldsymbol{x}_M(t)$$

$\sim O(M)$ dofs

Sandia National Laboratories

Stability Issues of POD/Galerkin ROMs

LTI Full Order Model (FOM)	LTI Reduced Order Model (ROM)
$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$ $\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t)$	$\dot{\boldsymbol{x}}_M(t) = \boldsymbol{A}_M\boldsymbol{x}_M(t) + \boldsymbol{B}_M\boldsymbol{u}(t)$ $\boldsymbol{y}_M(t) = \boldsymbol{C}_M\boldsymbol{x}_M(t)$

- ROM LTI system matrices given by:

$$\boldsymbol{A}_M = \boldsymbol{\Phi}_M{}^T\boldsymbol{A}\boldsymbol{\Phi}_M, \quad \boldsymbol{B}_M = \boldsymbol{\Phi}_M{}^T\boldsymbol{B}, \quad \boldsymbol{C}_M = \boldsymbol{C}\boldsymbol{\Phi}_M$$

Problem: $\boldsymbol{A}$ stable $\nRightarrow$ $\boldsymbol{A}_M$ stable!

- There is no *a priori* stability guarantee for POD/Galerkin ROMs.
- Stability of a ROM is commonly evaluated *a posteriori* – **RISKY!**
- ROM instability of POD/Galerkin ROMs is a **real** problem in some applications (e.g., compressible flow).

Stability Preserving Model Reduction Approaches: Literature Review

Approaches for building stability-preserving POD/Galerkin ROMs found in the literature fall into **two categories:**

1. ROMs which derive ***a priori*** a stability-preserving model reduction framework (usually specific to an equation set).

Potentially intrusive implemetation

- ROMs based on projection in special 'energy-based' (not L^2) inner products, e.g., Rowley *et al*. (2004), Barone & Kalashnikova *et al.* (2009), Serre *et al*. (2012).

2. ROMs which stabilize an unstable ROM through an ***a posteriori*** post-processing stabilization step applied to the algebraic ROM system.

Inconsistencies between ROM and FOM physics

- Petrov-Galerkin ROMs that solve an optimization problem for the test basis given a trial POD basis, e.g., Amsallem *et al.* (2012), Bond *et al.* (2008).
- ROMs with increased numerical stability due to inclusion of 'stabilizing' terms (e.g., LES terms) in the ROM equations, e.g., Wang, Akhtar, Borggaard, Iliescu (2012)

New ROM Stabilization Approach: ROM Stabilization via Pole Placement

New ROM Stabilization Idea
(falling under second category of ROM stabilization methods):
Make ROM system matrix $\boldsymbol{A}_M$ stable through **pole (eigenvalue) placement**

LTI FOM: $\boldsymbol{A}$ stable	**LTI ROM:** $\boldsymbol{A}_M$ unstable
$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$ $\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t)$	$\dot{\boldsymbol{x}}_M(t) = \boldsymbol{A}_M\boldsymbol{x}_M(t) + \boldsymbol{B}_M\boldsymbol{u}(t)$ $\boldsymbol{y}_M(t) = \boldsymbol{C}_M\boldsymbol{x}_M(t)$

- Pick a control matrix $\boldsymbol{B}_C$.
- Modify LTI ROM by adding to it the control $\boldsymbol{B}_C\boldsymbol{u}_C(t)$:

$$\dot{\boldsymbol{x}}_M(t) = \boldsymbol{A}_M\boldsymbol{x}_M(t) + \boldsymbol{B}_M\boldsymbol{u}(t) + \boldsymbol{B}_C\boldsymbol{u}_C(t)$$
$$\boldsymbol{y}_M(t) = \boldsymbol{C}_M\boldsymbol{x}_M(t)$$

- Assume linear control of the form $\boldsymbol{u}_C(t) = -\boldsymbol{K}_C\boldsymbol{x}_M(t)$:

$$\dot{\boldsymbol{x}}_M(t) = (\boldsymbol{A}_M - \boldsymbol{B}_C\boldsymbol{K}_C)\boldsymbol{x}_M(t) + \boldsymbol{B}_M\boldsymbol{u}(t)$$
$$\boldsymbol{y}_M(t) = \boldsymbol{C}_M\boldsymbol{x}_M(t)$$

- Compute the feedback matrix $\boldsymbol{K}_C$ such that $\widetilde{\boldsymbol{A}}_M \equiv \boldsymbol{A}_M - \boldsymbol{B}_C\boldsymbol{K}_C$ has desired (stable) poles.

Open Questions:

- How to pick $\boldsymbol{B}_C$?
- What eigenvalues should $\widetilde{\boldsymbol{A}}_M$ have?
- Does solution $\boldsymbol{K}_C$ to pole placement problem exist?
- How does stabilization affect ROM accuracy?

Algorithm #1: ROM Stabilization via Pole Placement

- Ensures the solution $\boldsymbol{K}_C$ to the pole placement problem exists.

Theorem (from Aström *et al.* 2009)**:** *If the pair (*$\boldsymbol{A}_M$*,*$\boldsymbol{B}_C$*) is controllable, there exists a feedback* $\boldsymbol{u}_C(t) = -\boldsymbol{K}_C\boldsymbol{x}_M(t)$ *such that the eigenvalues of* $\widetilde{\boldsymbol{A}}_M \equiv \boldsymbol{A}_M - \boldsymbol{B}_C\boldsymbol{K}_C$ *can be arbitrarily assigned.*

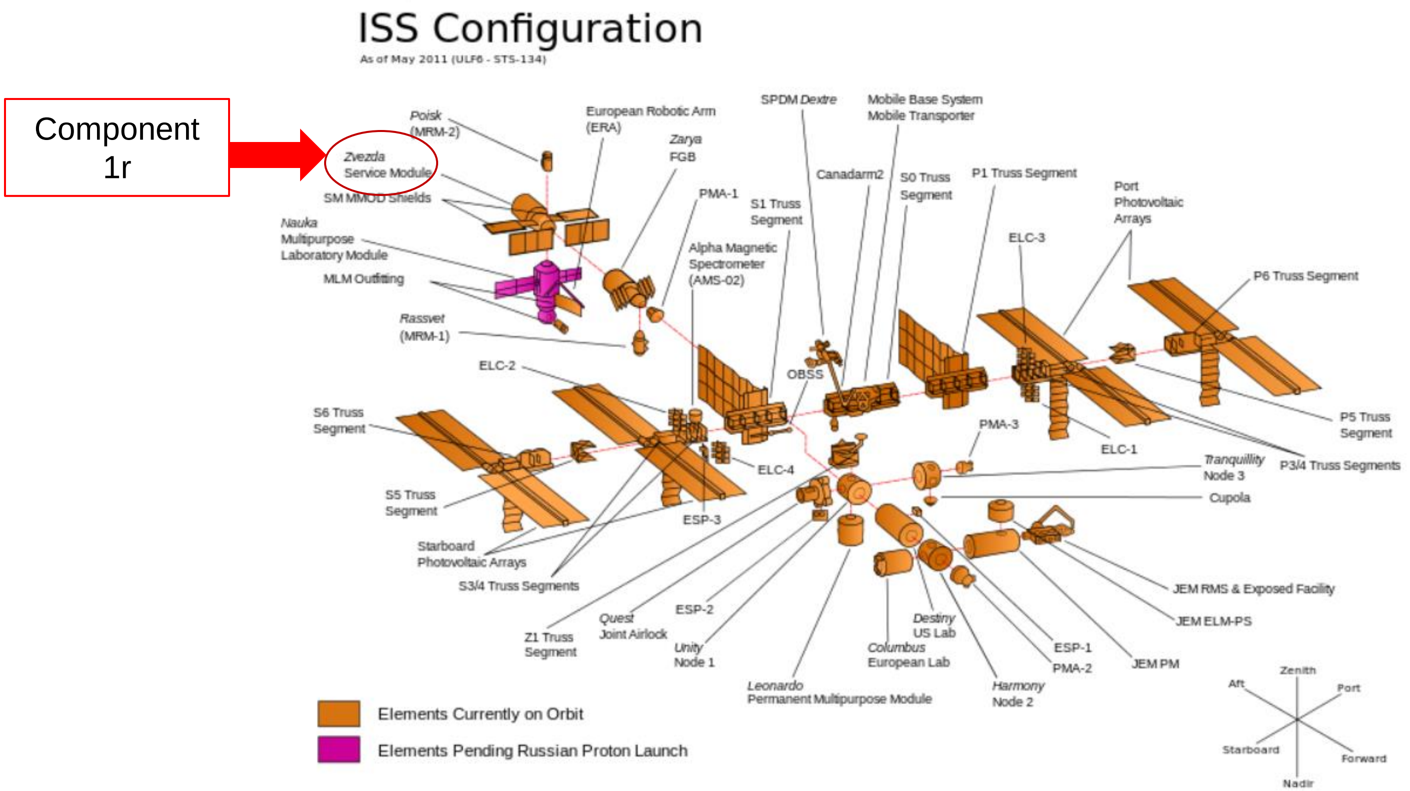
Algorithm #1 Outline

- Given $\boldsymbol{B}_C$, use Kalman decomposition to isolate controllable and observable part of $\boldsymbol{A}_M$ and $\boldsymbol{B}_C$, call them ${\boldsymbol{A}_M}^{co} = \boldsymbol{U}\boldsymbol{A}_M\boldsymbol{U}^T$, ${\boldsymbol{B}_C}^{co} = \boldsymbol{U}\boldsymbol{B}_C$.
- Compute eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_{N^{co}}$ of ${\boldsymbol{A}_M}^{co}$.
- Reassign unstable eigenvalues of ${\boldsymbol{A}_M}^{co}$, e.g., set

$$\lambda_i = \min\{Re(\lambda_i), -Re(\lambda_i)\} + i \cdot Im(\lambda_i)$$

- Compute $\boldsymbol{K}_C$ such that ${\boldsymbol{A}_M}^{co} - \boldsymbol{K}_C\, {\boldsymbol{B}_C}^{co}$ has these eigenvalues using pole placement algorithms from control theory.
- Set $\boldsymbol{A}_M = \boldsymbol{U}^T({\boldsymbol{A}_M}^{co} - \boldsymbol{K}_C\, {\boldsymbol{B}_C}^{co})\boldsymbol{U}$.

Numerical Results: International Space Station (ISS) Benchmark Stabilized via Algorithm #1

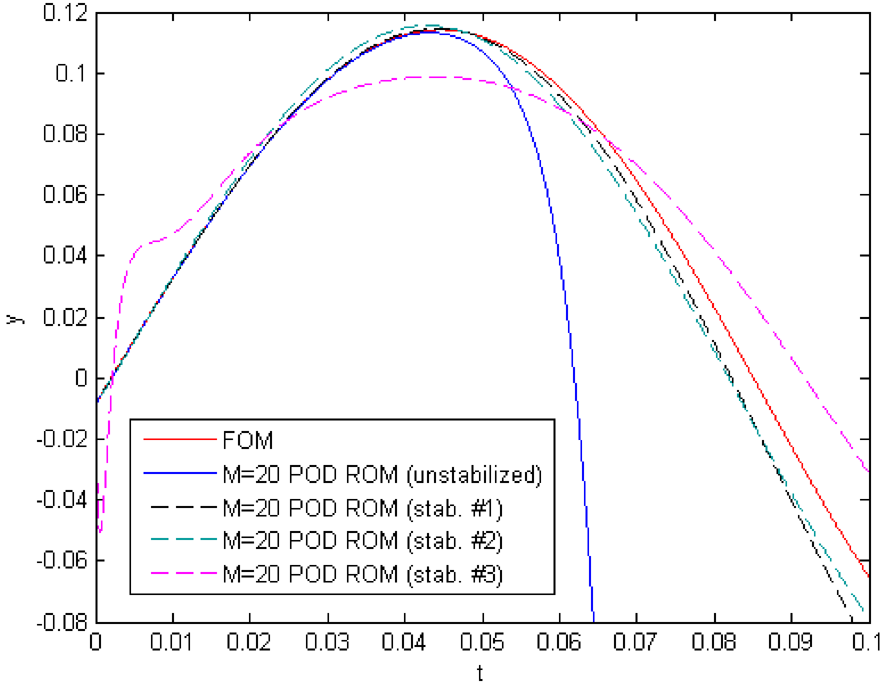


- FOM: structural model of component 1r (Russian service module) of the International Space Station (ISS).
- $\boldsymbol{A}, \boldsymbol{C}$ matrices defining FOM downloaded from NICONET model reduction benchmark repository.
- No inputs (unforced), 1 output; FOM is stable

Sandia National Laboratories

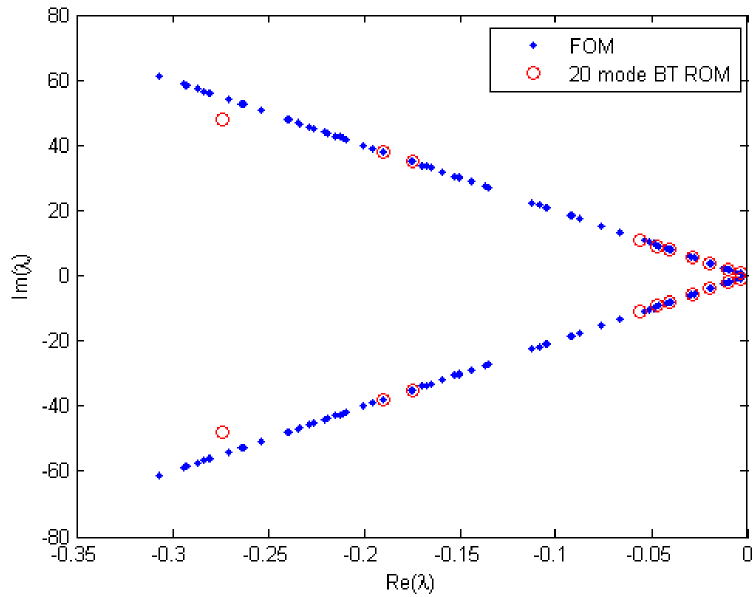
Numerical Results: ISS Benchmark Stabilized via Algorithm #1 (continued)

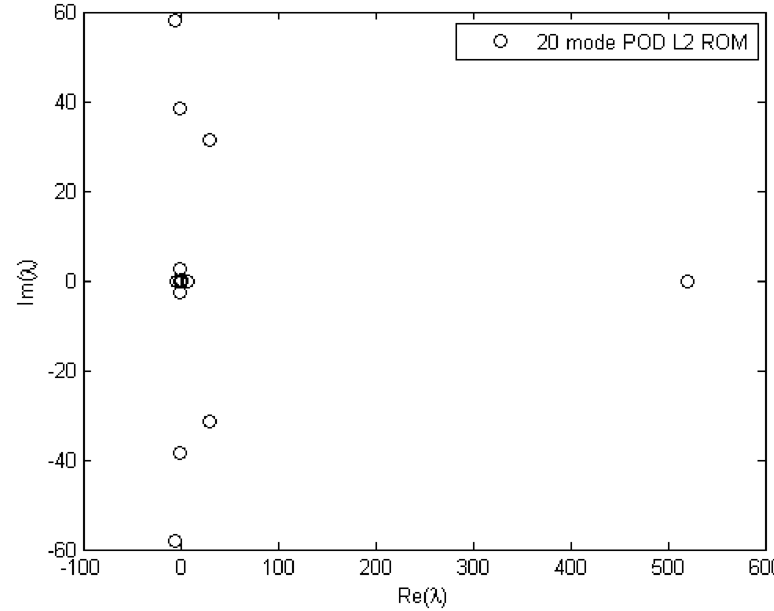
- $M = 20$ POD/Galerkin ROM constructed using $K = 2000$ snapshots collected until time $t = 0.1$, with $\boldsymbol{x}(0) = \mathbf{1}_N$.
- $M = 20$ POD/Galerkin ROM has 4 unstable eigenvalues.
- $\boldsymbol{B}_C = \mathbf{1}_M$ (stabilization matrix selected to be vector of all 1s).



ROM	$\frac{\sqrt{\sum_{k=1}^{K} \lVert \boldsymbol{y}^k - \boldsymbol{y}_M{}^k \rVert_2{}^2}}{\sqrt{\sum_{k=1}^{K} \lVert \boldsymbol{y}_k \rVert_2{}^2}}$
Unstabilized POD	1737.8
Stabilized POD #1 $Re(\lambda_i{}^u) \leftarrow -0.1 Re(\lambda_i{}^u)$	0.1151
Stabilized POD #2 $Re(\lambda_i{}^u) \leftarrow -Re(\lambda_i{}^u)$	0.1116
Stabilized POD #3 $Re(\lambda_i{}^u) \leftarrow -10 Re(\lambda_i{}^u)$	0.2267

Selecting ROM Eigenvalues (Poles)





- If ROM is stable, ROM eigenvalues will lie on same manifold as FOM eigenvalues.
- In fact, if $M = N$ and ROM is stable, $\boldsymbol{A}_M \sim \boldsymbol{A}$, so eig($\boldsymbol{A}_M$)=eig($\boldsymbol{A}$).
- **However:** if ROM is unstable, ROM eigenvalues can be chaotic.

Could try to place ROM poles to match FOM poles…
…**but** this does not ensure **accuracy** of ROM!

Algorithm #2: ROM Stabilization Optimization Problem

- Ensures stabilized ROM solution deviates minimally from FOM solution.

Key Idea: An exact solution to the ROM LTI system can be derived using the matrix exponential.

- Recall that the ROM LTI system is given by:

$$\begin{aligned} \dot{\boldsymbol{x}}_M(t) &= \boldsymbol{A}_M \boldsymbol{x}_M(t) + \boldsymbol{B}_M \boldsymbol{u}(t) \\ \boldsymbol{y}_M(t) &= \boldsymbol{C}_M \boldsymbol{x}_M(t) \end{aligned}$$

- The solution to the ROM LTI system is:

$$\boldsymbol{x}_M(t) = \exp(t\boldsymbol{A}_M)\, \boldsymbol{x}_M(0) + \int_0^t \exp\{(t-\tau)\, \boldsymbol{A}_M\} \boldsymbol{B}_M u(\tau) d\tau$$

$$\Rightarrow \boldsymbol{y}_M(t) = \boldsymbol{C}_M \left[\exp(t\boldsymbol{A}_M)\, \boldsymbol{x}_M(0) + \int_0^t \exp\{(t-\tau)\, \boldsymbol{A}_M\} \boldsymbol{B}_M u(\tau) d\tau \right]$$

Algorithm #2: ROM Stabilization Optimization Problem (continued)

ROM Stabilization Optimization Problem
(Constrained Nonlinear Least Squares):

$$\min_{\lambda_i^u} \sum_{k=1}^{K} ||\boldsymbol{y}^k - \boldsymbol{y}_M{}^k||_2{}^2$$
$$s.t.\ Re(\lambda_i^u) < 0$$

- $\lambda_i^u =$ unstable eigenvalues of original ROM matrix $\boldsymbol{A}_M$.
- $\boldsymbol{y}^k = \boldsymbol{y}(t_k) =$ snapshot output at t_k
- $\boldsymbol{y}_M{}^k = \boldsymbol{C}_M\left[\exp(t_k\boldsymbol{A}_M)\,\boldsymbol{x}_M(0) + \int_0^{t_k}\exp\{(t_k-\tau)\,\boldsymbol{A}_M\}\boldsymbol{B}_M u(\tau)d\tau\right] =$ ROM output at t_k.
- ROM stabilization optimization problem is small: $< O(M)$.
- ROM stabilization optimization problem can be solved by standard optimization algorithms, e.g., interior point method.
  - We use `fmincon` function in MATLAB's optimization toolbox.
  - We implement ROM stabilization optimization problem in ***characteristic variables*** $\boldsymbol{z}_M(t) = \boldsymbol{S}_M{}^{-1}\boldsymbol{x}_M(t)$ where $\boldsymbol{A}_M = \boldsymbol{S}_M\boldsymbol{D}_M\boldsymbol{S}_M{}^{-1}$.

Algorithm #2 Is Equivalent to Specific Instance of Algorithm #1 (Pole Placement Problem)

- Let $\widetilde{\boldsymbol{A}}_M$ be the new ROM matrix following stabilization by Algorithm #2:

$$\boldsymbol{A}_M \leftarrow \widetilde{\boldsymbol{A}}_M = \boldsymbol{S}_M \widetilde{\boldsymbol{D}}_M \boldsymbol{S}_M{}^{-1},\ \widetilde{\boldsymbol{D}}_M = \boldsymbol{D}_M \text{ stabilized by Algorithm #2}$$

Can show: Algorithm #2 is equivalent to Algorithm #1 (Pole Placement Problem) for a specific $\boldsymbol{B}_C$ and $\boldsymbol{K}_C$.

- One can show that $\widetilde{\boldsymbol{A}}_M = \boldsymbol{A}_M - \boldsymbol{B}_C \boldsymbol{K}_C$ where (WLOG, if 2 eigenvalues were stabilized):

$$\boldsymbol{B}_C = \boldsymbol{S}_M(:,1:2)$$

$$\boldsymbol{K}_C = \begin{pmatrix} \lambda_1 - \tilde{\lambda}_1 & 0 & 0 \\ 0 & \lambda_2 - \tilde{\lambda}_2 & 0 \end{pmatrix}$$

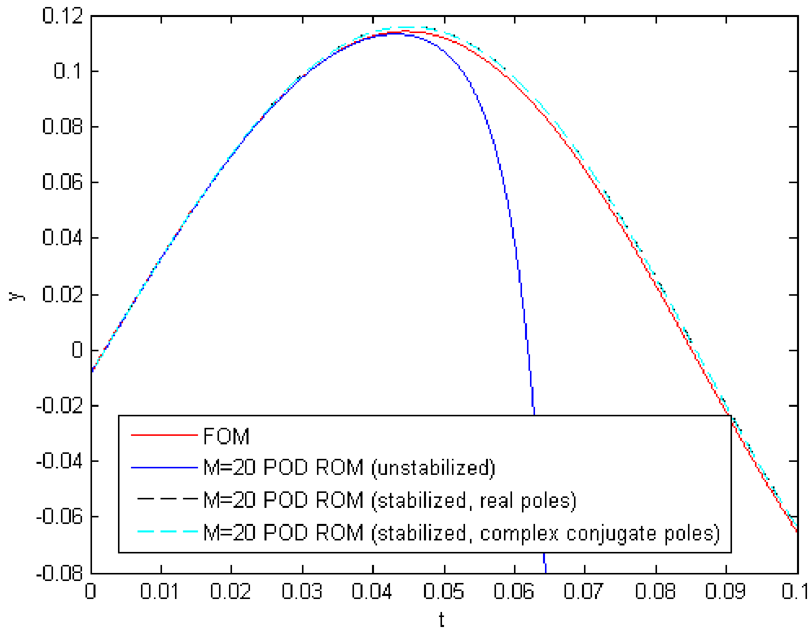
- Here:
  - λ_i denotes the original (unstable) eigenvalue of $\boldsymbol{A}_M$
  - $\tilde{\lambda}_i$ denotes the new (stable) eigenvalue of $\widetilde{\boldsymbol{A}}_M$ computed by Algorithm #2.
  - $\boldsymbol{S}_M$ denotes matrix of eigenvectors of $\boldsymbol{A}_M$.

Numerical Results: ISS Benchmark Stabilized via Algorithm #2

- $M = 20$ POD/Galerkin ROM has 4 unstable eigenvalues: 2 real, 2 complex
  - Two options for ROM stabilization optimization problem:

    Option 1: Solve for $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \epsilon \mathbb{R}$ s.t. the constraint $\lambda_1, \lambda_2, \lambda_3, \lambda_4 < 0$.

    Option 2: Solve for $\lambda_1 + \lambda_2 i$, $\lambda_1 - \lambda_2 i$, λ_3, λ_4 s.t. the constraint $\lambda_1, \lambda_3, \lambda_4 < 0$
- Initial guess for `fmincon` interior point method: $\lambda_1 = \lambda_2 = \lambda_3 = \lambda = -1$.



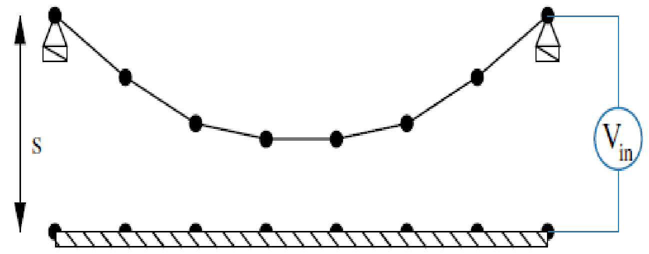
ROM	$\dfrac{\sqrt{\sum_{k=1}^{K} \lVert \boldsymbol{y}^k - \boldsymbol{y}_M{}^k \rVert_2{}^2}}{\sqrt{\sum_{k=1}^{K} \lVert \boldsymbol{y}_k \rVert_2{}^2}}$
Unstabilized POD	1737.8
Optimization Stabilized POD (Real Poles)	0.0259
Optimization Stabilized POD (Complex-Conjugate Poles)	0.0252

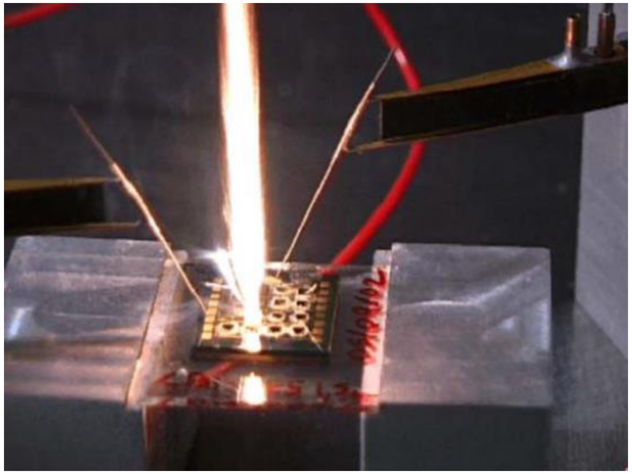
Numerical Results: Electrostatically Actuated Beam Benchmark Stabilized via Algorithm #2

- FOM = 1D model of electrostatically actuated beam.
- Application of model: microelectromechanical systems (MEMS) devices such as electromechanical radio frequency (RF) filters.
- 1 input corresponding to periodic on/off switching, 1 output, initial condition $\boldsymbol{x}(0) = \mathbf{0}_N$.
- Second order linear semi-discrete system of the form:

$$\begin{aligned} \boldsymbol{M}\ddot{\boldsymbol{x}}(t) + \boldsymbol{E}\dot{\boldsymbol{x}}(t) + \boldsymbol{K}\boldsymbol{x}(t) &= \boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) \end{aligned}$$

- Matrices $\boldsymbol{M}$, $\boldsymbol{E}$, $\boldsymbol{K}$, $\boldsymbol{B}$, $\boldsymbol{C}$ specifying the problem downloaded from the Oberwolfach model reduction repository.
- 2^{nd} order linear system re-written as 1^{st} order LTI system for purpose of analysis/model reduction.





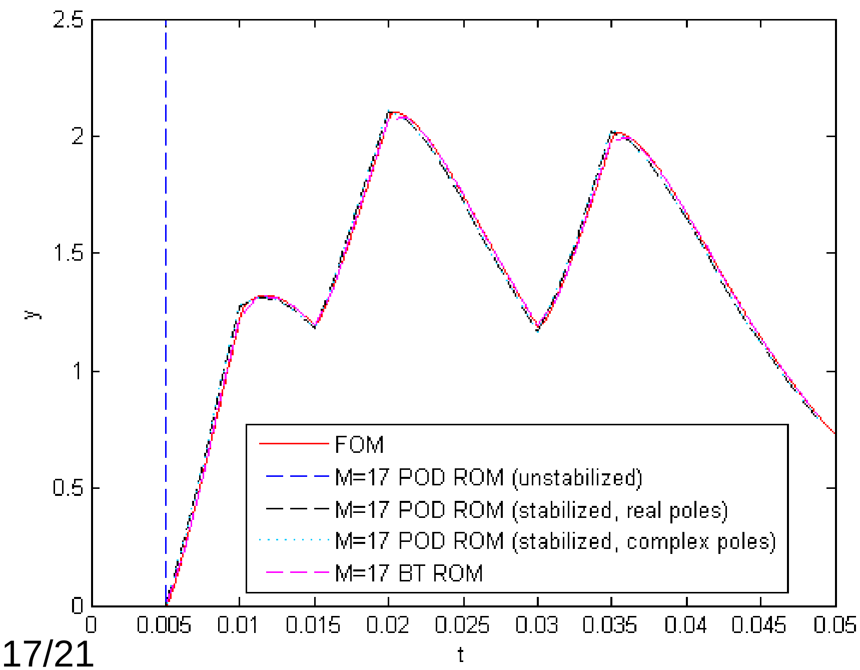
- FOM is stable.

Numerical Results: Electrostatically Actuated Beam Benchmark Stabilized via Algorithm #2 (continued)

- $M = 17$ POD/Galerkin ROM constructed from $K = 1000$ snapshots up to time $t = 0.05$.
- $M = 17$ POD/Galerkin ROM has 4 unstable eigenvalues (all real).
  - Two options for ROM stabilization optimization problem:

    Option 1: Solve for $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \epsilon \mathbb{R}$ s.t. the constraint $\lambda_1, \lambda_2, \lambda_3, \lambda_4 < 0$.

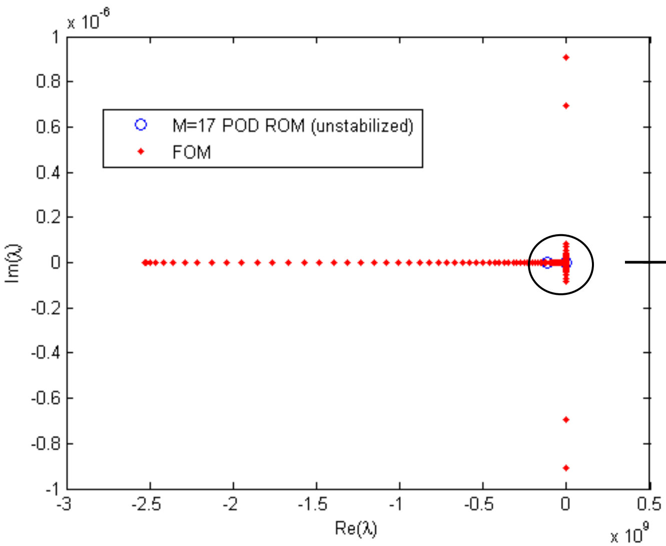
    Option 2: Solve for $\lambda_1 + \lambda_2 i$, $\lambda_1 - \lambda_2 i$, $\lambda_3 + \lambda_4 i$, $\lambda_3 - \lambda_4 i$ s.t. the constraint $\lambda_1, \lambda_3 < 0$.
- Initial guess for `fmincon` interior point method: $\lambda_1 = \lambda_2 = \lambda_3 = \lambda = -1$.

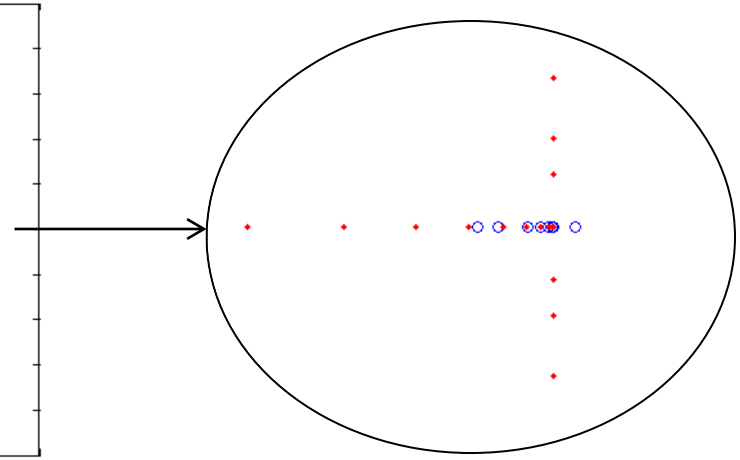


ROM	$\frac{\sqrt{\sum_{k=1}^{K} \|\|\boldsymbol{y}^k - \boldsymbol{y}_M{}^k\|\|_2{}^2}}{\sqrt{\sum_{k=1}^{K} \|\|\boldsymbol{y}_k\|\|_2{}^2}}$
Unstabilized POD	NaN
Optimization Stabilized POD (Real Poles)	0.0194
Optimization Stabilized POD (Complex-Conjugate Poles)	0.0205
Balanced Truncation	$1.370e-6$

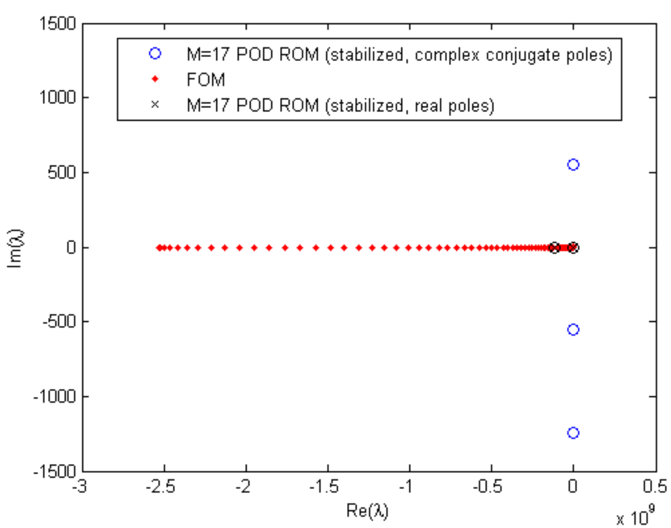
Sandia National Laboratories

Numerical Results: Electrostatically Actuated Beam Benchmark Stabilized via Algorithm #2 (continued: stabilized poles)





Unstable Eigenvalues
$\lambda_6 = 16{,}053$
$\lambda_{12} = 48.985$
$\lambda_{14} = 12.650$
$\lambda_{17} = 0.05202$



Stabilized Eigenvalues (Real)	Stabilized Eigenvalues (Complex Conjugates)
$\lambda_6 = -7{,}043{,}505$	$\lambda_6 = -106{,}976 + 551.77i$
$\lambda_{12} = -35.364$	$\lambda_{12} = -106{,}976 - 551.77$
$\lambda_{14} = -153{,}033$	$\lambda_{14} = -2954.1 - 1244.7i$
$\lambda_{17} = -99{,}175$	$\lambda_{17} = -2954.1 + 1244.7i$

Summary & Future Work

- A ***new*** ROM stabilization approach that modifies *a posteriori* an unstable ROM LTI system by changing the system's unstable eigenvalues is proposed.
- It is demonstrated how an unstable ROM's eigenvalues may be placed using ideas from control theory, i.e., pole placement assuming a linear control (Algorithm #1).
- It is unclear how stabilization via Algorithm #1 will affect the accuracy of the ROM.
- To address this, a constrained nonlinear least squares optimization problem for the ROM eigenvalues is formulated (Algorithm #2).
- It is shown that Algorithm #2 is equivalent to Algorithm #1 for a specific linear control.
- Performance of the proposed algorithms is evaluated on two benchmarks.

Ongoing/Future work

- Journal article on the ideas presented in this talk is in preparation.
- Performance/robustness studies of algorithms for solving ROM optimization problem (e.g., sensitivity to initial guess, etc.).
- Studies of predictive capabilities of stabilized ROMs.
- Extensions to nonlinear problems.

Acknowledgements

- This work was funded by Laboratories' Directed Research and Development (LDRD) Program at Sandia National Laboratories.

- Special thanks to
  - Prof. Lou Cattafesta (Florida State University)
  - Prof. Karen Willcox (MIT)

  for useful discussions that led to some of the ideas presented here.

Thank You! Questions?

ikalash@sandia.gov

http://www.sandia.gov/~ikalash

Sandia National Laboratories

References

- M.F. Barone, I. Kalashnikova, D.J. Segalman, H. Thornquist. Stable Galerkin reduced order models for linearized compressible flow. *J. Comput. Phys.* **288*:*** 1932-1946, 2009.

- C.W. Rowley, T. Colonius, R.M. Murray. Model reduction for compressible flows using POD and Galerkin projection. *Physica D.* **189**: 115-129, 2004.

- G. Serre, P. Lafon, X. Gloerfelt, C. Bailly. Reliable reduced-order models for time-dependent linearized Euler equations. *J. Comput. Phys.* **231**(15): 5176-5194, 2012.

- Z. Wang, I. Akhtar, J. Borggaard, T. Iliescu. Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison. *Comput. Methods Appl. Mech. Engrg.* **237-240**:10-26, 2012.

- B. Bond, L. Daniel, Guaranteed stable projection-based model reduction for indefinite and unstable linear systems, In: *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, 728–735, 2008.

- K.J. Aström, R.M. Murray. *Feedback systems: an introduction for scientists and engineers.* Princeton University Press, 2008.

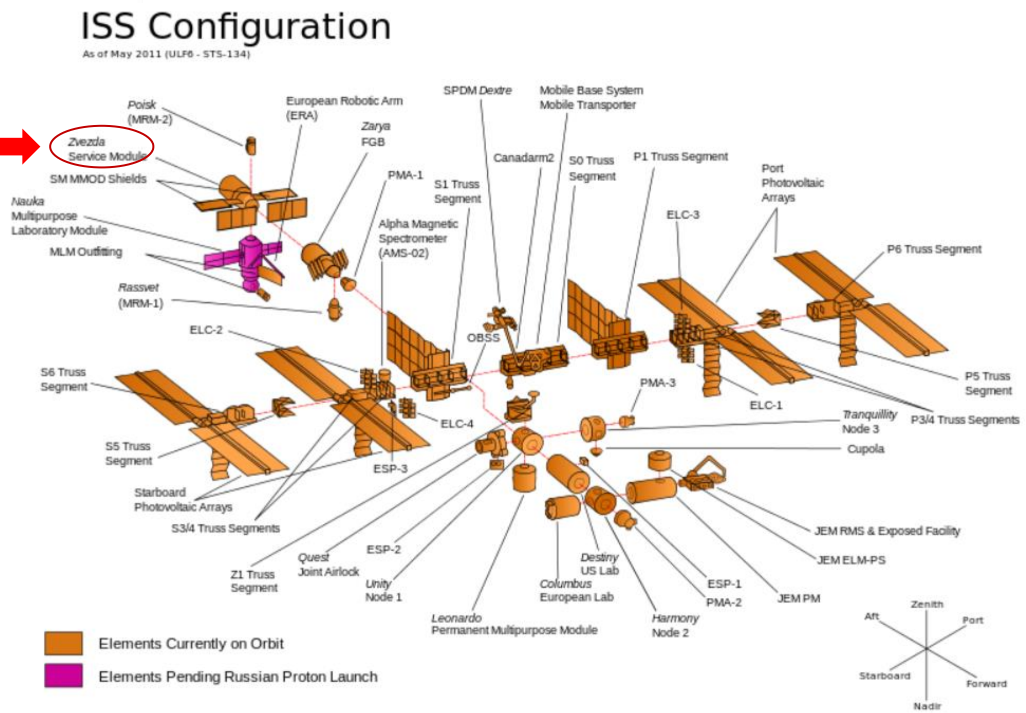
- NICONET model reduction benchmark repository: www.icm.tu-bs.de/NICONET/benchmodred.html.

- Oberwolfach model reduction benchmark repository: http://simulation.uni-freiburg.de/downloads/benchmark

Sandia National Laboratories

Appendix: ISS Benchmark Details

Compone
nt 1r



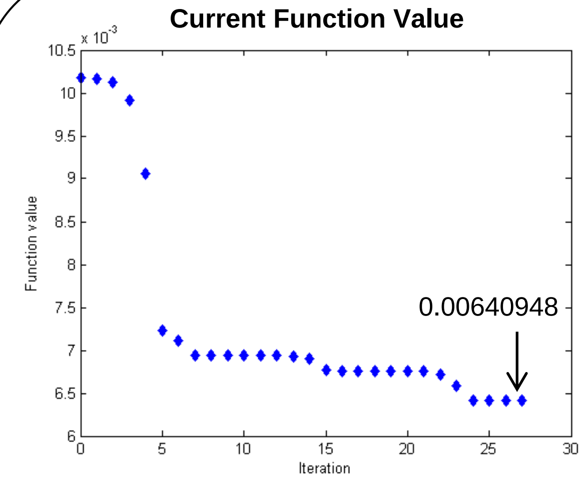
- Original benchmark: $N = 270$, 3 inputs, 3 outputs
- Physics: 2nd order semi-discrete system for flexible structure

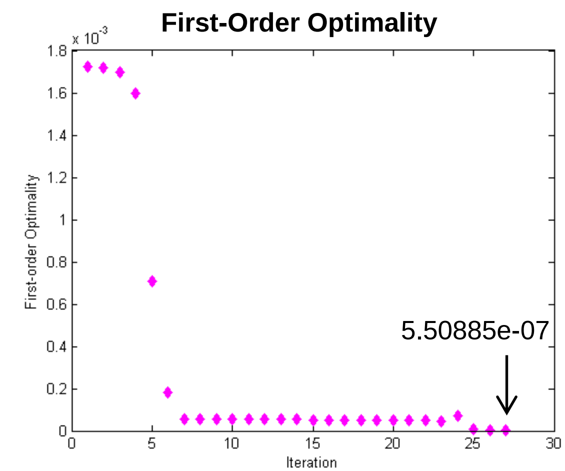
$$\boldsymbol{M}\ddot{\boldsymbol{x}}(t) + \boldsymbol{E}\dot{\boldsymbol{x}}(t) + \boldsymbol{K}\boldsymbol{x}(t) = \boldsymbol{B}\boldsymbol{u}(t)$$
$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t)$$

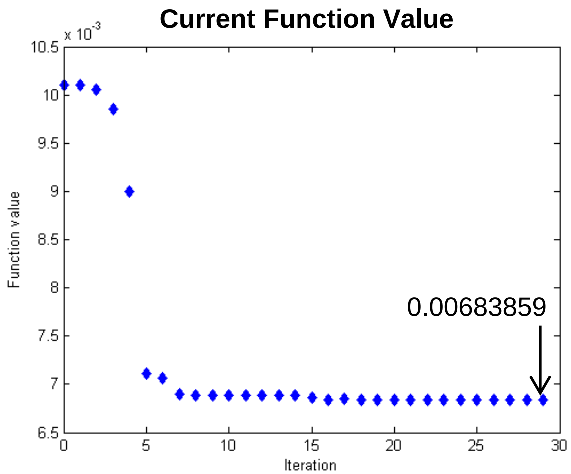
- All 3 inputs and 3 outputs are the same.
- Inputs = outputs = velocities at all discretization points (?)

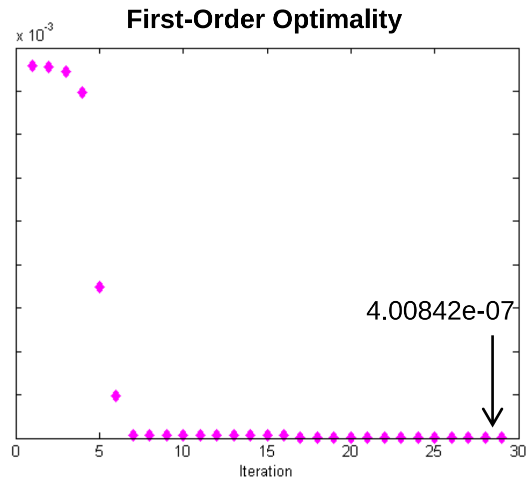
Appendix: ISS Benchmark Stabilized via Algorithm #2 (`fmincon` performance)

	Real Poles	Complex-Conjugate Poles
# upper bound constraints	4	3
# iterations	29	27
# function evaluations	150	142
max constraint violation	0	0



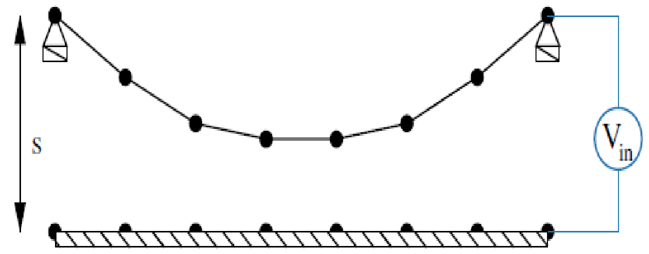






Appendix: 1D Beam Benchmark Details

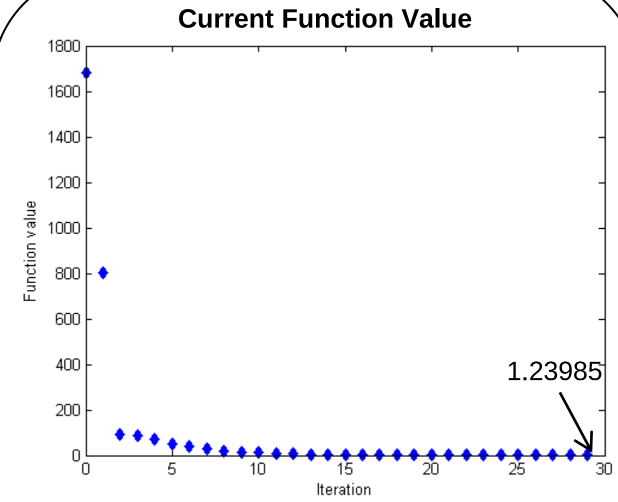
- $N = 100$, 1 input, 1 output.
- Dofs: flexural displacement and flexural rotation.
- Beam is supported on both sides (simply supported), and a node in the middle is loaded.

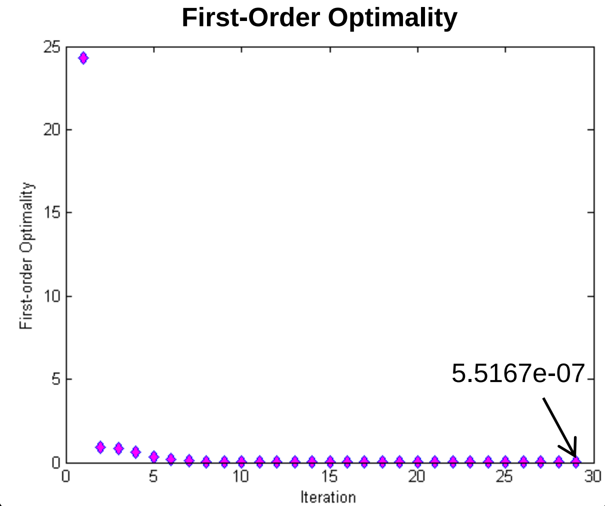


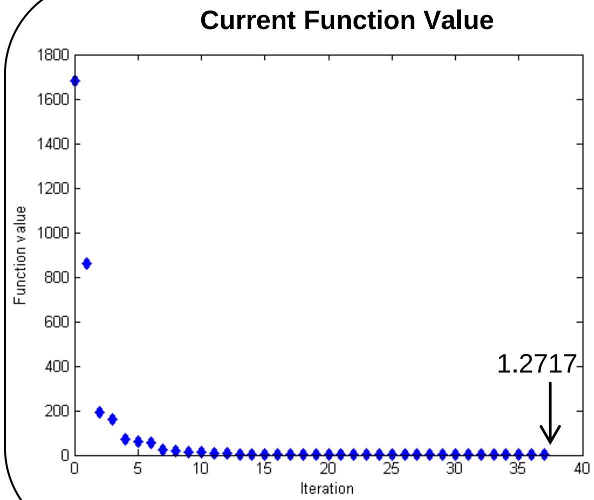
- Damping matrix is linear combination of mass and stiffness matrices:
$$\boldsymbol{E} = \alpha\boldsymbol{K} + \beta\boldsymbol{M}$$
- **B** (input) is an $n \times 1$ matrix with the only nonzero entry at the flexural dof of the middle node.
- **C** (output) is a $1 \times n$ matrix with the only nonzero entry at the flexural dof of the middle node.
- A typical input to this system is a step response; periodic on/off switching is also possible.

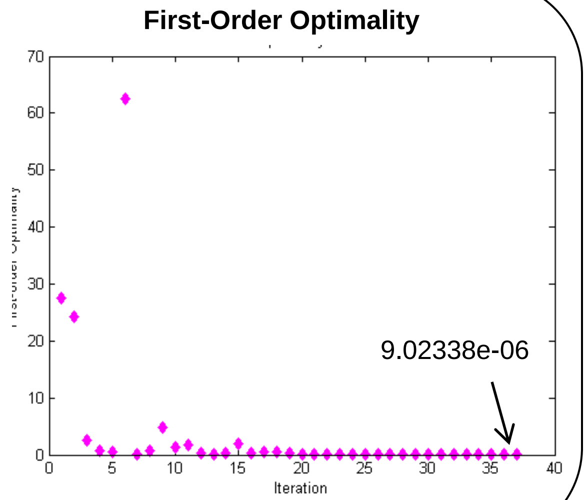
Appendix: Electrostatically Actuated Beam Benchmark Stabilized via Algorithm #2 `fmincon` performance)

	Real Poles	Complex-Conjugate Poles
# upper bound constraints	4	2
# iterations	37	29
# function evaluations	190	150
max constraint violation	0	0









Sandia National Laboratories