Introduction to Simulation - Lecture 24

Model-Order Reduction

Jacob White

Thanks to Luca Daniel, Guillaum Lassauxx, Jing Li, Mark Reichelt, Deepak Ramaswamy, Michal Rewienski, Mary Tolikas, Karen Veroy and Karen Willcox

MOR Outline

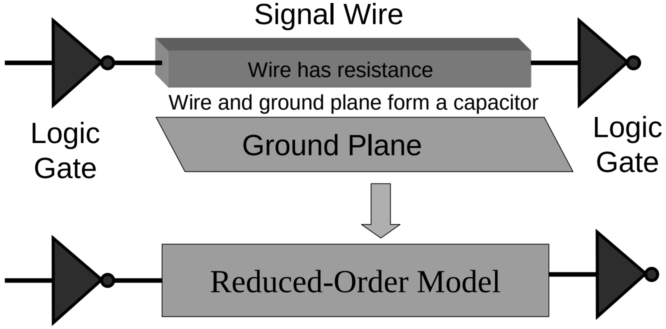
- Need For Model Reduction
  - Circuits, MEMS, Jet Engines
- Steady-State Case (linear and nonlinear)
- Dynamic Linear Case
  - Eigenmodes and Rational Functions
  - Projection Framework, Krylov, TBR
- Nonlinear Case
  - Projection Framework

Today's Outline

- Need For Model Reduction
  - Circuits, MEMS, Optics, Jet Engines
- Simple Example Problem
  - Heat Conducting bar example
- Steady-State Case (linear and nonlinear)
- Dynamic Linear Case
  - Truncating Eigenmodes
  - Rational Function Fitting

Application Example

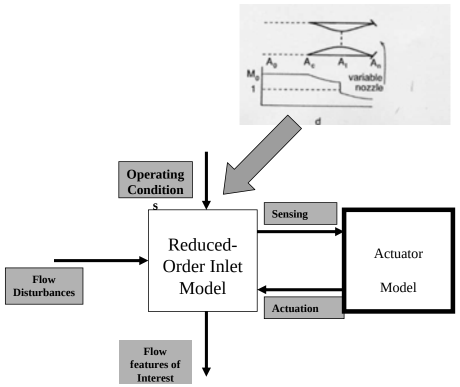
Signal Wire in an Integrated Circuit



- Assess wiring impact on IC performance
- Wire Model must preserve terminal behavior

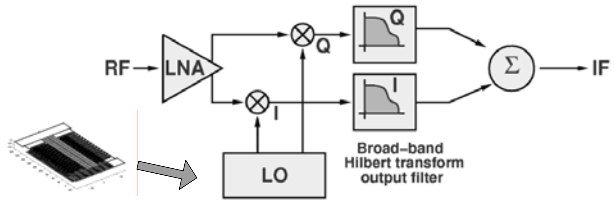
Application Example
Jet Engine Design

- Generate Low-order models directly from Navier-Stokes Equation based physical simulators.
- Reduced model must preserve instabilities.



SMA-HPC ©2003 MIT

Application Examples
Micromechanical Resonators in a Wireless transceiver

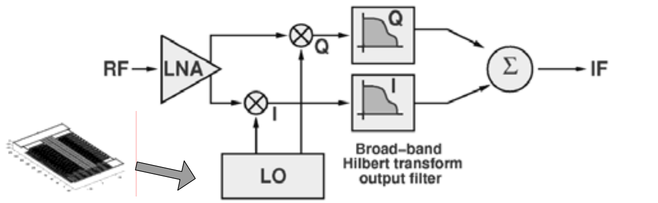


RF Front end with micromachined resonators for the oscillator

- What is system performance (noise, distortion, etc).
- Will poly-substrate separation (changes Q) matter?
- How tight must manufacturing tolerances be?

SMA-HPC ©2003 MIT

Application Examples
Micromechanical Resonators in a Wireless transceiver (cont)



Need to simulate ENTIRE system with dynamically accurate macromodels for all the components

SMA-HPC ©2003 MIT

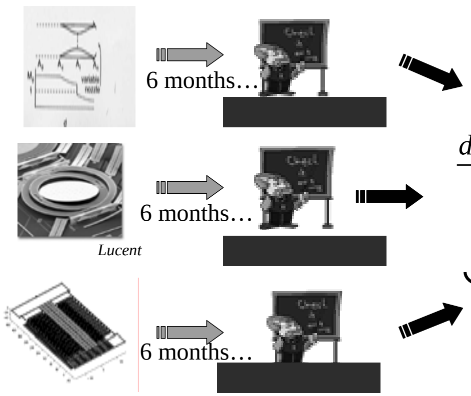
Application Examples
Common features

- Devices have Well-defined Inputs and Outputs
  - Signal Transmission on a Wire
    - Left and right end Voltages and Currents
  - Jet Engine Nozzle Design
    - Nozzle in and out flow, Nozzle size
  - Microresonator
    - Comb finger voltages and currents
- Dynamics is important
  - Internal state must be somehow represented

SMA-HPC ©2003 MIT

Application Examples

Traditional Approach to Generating Models



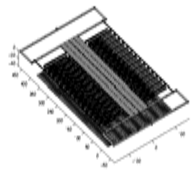
$$\frac{dx_r(t)}{dt} = F(x_r(t)) + b_r u(t)$$

$$y(t) = c_r^T x_r(t)$$

Model for the System Simulator

The Numerical Macromodeling or Model Reduction Paradigm

Generate a Reduced-Order Model Directly from 3-D Geometry and Physics



Automatic

$$\frac{dx_r(t)}{dt} = F(x_r(t)) + b_r u(t)$$

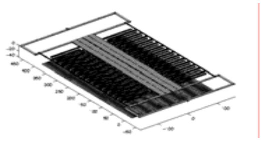
$$y(t) = c_r^T x_r(t)$$

Complicated Geometry, Coupled Electrostatics, Fluids, Elastics

Cheap to evaluate model which captures input (u)/output(y) behavior

The Numerical Macromodeling or Model Reduction Paradigm

What's Needed



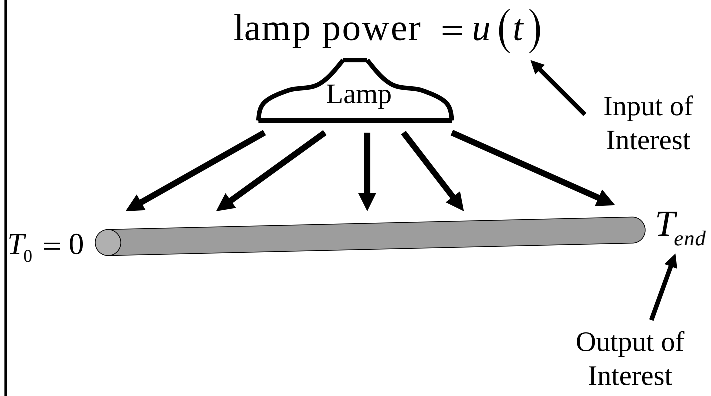
$$\frac{dx_r(t)}{dt} = F(x_r(t)) + b_r u(t)$$

$$y(t) = c_r^T x_r(t)$$

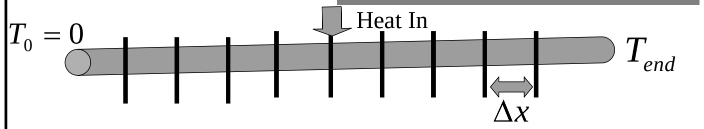
- Fast Solvers for complicated 3-D geometries
  - (Fast enough to solve ENTIRE devices)
  - for fluids, electrostatics, mechanics, …
- Approaches for coupled domain analysis
  - Multilevel-Newton methods
- Automatic extraction of reduced order models

Demonstration Example

Heat Conducting Bar



Demonstration Example — Heat Conducting Bar
Basic Equations



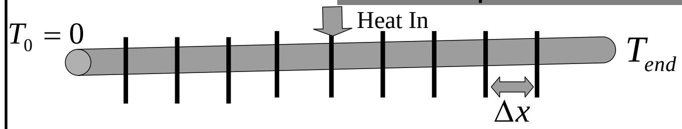
- Temperature Differential Equation

$$\underbrace{\gamma}_{\substack{specific\\heat}} \frac{\partial T(x,t)}{\partial t} - \underbrace{\kappa}_{\substack{thermal\\conductivity}} \frac{\partial^2 T(x,t)}{\partial x^2} = h(x)\underbrace{u(t)}_{\substack{scalar\\input}}$$

- Spatial Discretization (except at end)

$$\gamma \frac{d\hat{T}_i}{dt} - \frac{\kappa}{(\Delta x)^2}\left(\hat{T}_{i+1} - 2\hat{T}_i + \hat{T}_{i-1}\right) = h(x_i)u(t)$$

Demonstration Example — Heat Conducting Bar
Input-Output Discrete Equations



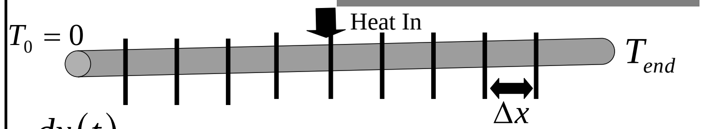
$$\gamma \frac{d\hat{T}_i}{dt} - \frac{\kappa}{(\Delta x)^2}\left(\hat{T}_{i+1} - 2\hat{T}_i + \hat{T}_{i-1}\right) = h(x_i)u(t) \quad i \in [1, \ldots, N-1]$$

$$\gamma \frac{d\hat{T}_i}{dt} - \frac{\kappa}{(\Delta x)^2}\left(\hat{T}_N - \hat{T}_{N-1}\right) = h(x_N)u(t)$$

$$T_{end} = \hat{T}_N$$

SMA-HPC ©2003 MIT

Demonstration Example — Heat Conducting Bar
State-Space Description



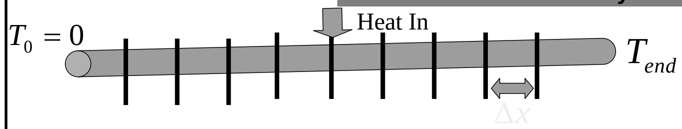
$$\frac{dx(t)}{dt} = \underbrace{A}_{NxN} x(t) + \underbrace{b}_{Nx1}\underbrace{u(t)}_{\substack{scalar\\input}} \qquad \underbrace{y(t)}_{\substack{scalar\\output}} = \underbrace{c}_{Nx1}{}^T x(t)$$

<u>Given the right scaling</u>

$$A = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 2 & -1 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix} \quad b = \begin{bmatrix} h(x_1) \\ h(x_2) \\ \vdots \\ \vdots \\ h(x_N) \end{bmatrix} \quad c = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

SMA-HPC ©2003 MIT

Demonstration Example — Heat Conducting Bar
Temperature Dependent Thermal Conductivity



- Temperature Differential Equation

$$\underbrace{\gamma}_{\substack{specific\\heat}} \frac{\partial T(x,t)}{\partial t} - \underbrace{\kappa\left(T(x,t)\right)}_{\substack{thermal\\conductivity}} \frac{\partial^2 T(x,t)}{\partial x^2} = h(x)$$

- Simple Spatial Discretization (not at ends)

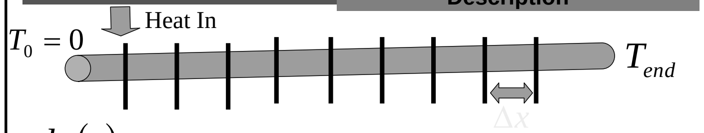
$$\gamma \frac{d\hat{T}_i}{dt} - \frac{\kappa(\hat{T}_i)}{(\Delta x)^2}\left(\hat{T}_{i+1} - 2\hat{T}_i + \hat{T}_{i-1}\right) = 0$$

SMA-HPC ©2003 MIT

Demonstration Example

Heat Conducting Bar

Nonlinear State-Space Description



$$\frac{dx(t)}{dt} = F\left(x(t)\right) + \underbrace{b}_{Nx1}\underbrace{u(t)}_{\substack{scalar\\input}} \qquad \underbrace{y(t)}_{\substack{scalar\\output}} = \underbrace{c}_{Nx1}^{T} x(t)$$

$$F(x) = \begin{bmatrix} 2\kappa(x_1) & -\kappa(x_1) & 0 & \cdots & 0 \\ -\kappa(x_2) & 2\kappa(x_2) & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 2\kappa(x_{N-1}) & -\kappa(x_{N-1}) \\ 0 & \cdots & 0 & -\kappa(x_N) & \kappa(x_N) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_N \end{bmatrix} \qquad b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \qquad c = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

No Dynamics (Steady-State) Case

Linear example

- Original System - Single Input/Output

$$0 = \underbrace{A}_{NxN} x + \underbrace{b}_{Nx1}\underbrace{u}_{\substack{scalar\\input}} \qquad \underbrace{y}_{\substack{scalar\\output}} = \underbrace{c}_{Nx1}^{T} x$$

- Reduced System

$$y = -\underbrace{c^T A^{-1} b}_{1x1} u$$

- Satisfies Reduced Model Criteria
  - Cheap to evaluate
  - Exactly reproduces I/O Behavior

No Dynamics Case

Nonlinear Example

- Original System - Single Input/Output

$$0 = F(x) + \underbrace{b}_{Nx1}\underbrace{u}_{\substack{scalar\\input}} \qquad \underbrace{y}_{\substack{scalar\\output}} = \underbrace{c}_{Nx1}^{T} x$$

- Reduced System

$$y = g(u)$$

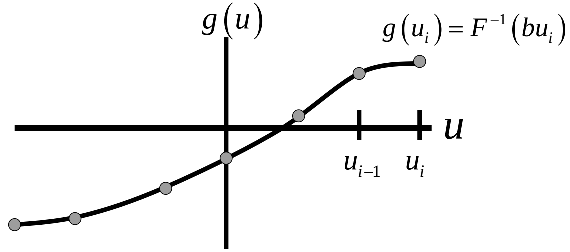
- Is "g(u)" a reduced-order model?
  - Depends how we represent g!

No Dynamics Case

Nonlinear Example

Representation of Reduced Model

- Could use an interpolated table of data



- Table is a reduced order model
  - Cheap to evaluate
  - Accurate if enough points used

No Dynamics Case — Model Construction Time

- Linear Case, one solve, one inner product
  - Solve $Ax = b \Rightarrow x = A^{-1}b$
  - Form $c^T x = c^T A^{-1} b$
- Nonlinear Case (if an interpolated table is used)
  - Solve
    $F(x_i) = bu_i \Rightarrow x_i = F^{-1}(bu_i) \quad for\ i = 1, \ldots \#samples$
  - Form $c^T x_i = c^T F^{-1}(bu_i) \quad for\ i = 1, \ldots \#samples$
- Nonlinear Reduction adds a representation problem to model reduction

Dynamic Linear case — State-Space Description

- Original Dynamical System - Single Input/Output

$$\frac{dx(t)}{dt} = \underbrace{A}_{NxN} x(t) + \underbrace{b}_{Nx1} \underbrace{u(t)}_{\substack{scalar\\input}} \qquad \underbrace{y(t)}_{\substack{scalar\\output}} = \underbrace{c}_{Nx1}{}^T x(t)$$

- Reduced Dynamical System

$$\frac{dx_r(t)}{dt} = \underbrace{A_r}_{qxq} x(t) + \underbrace{b_r}_{qx1} \underbrace{u(t)}_{\substack{scalar\\input}} \qquad \underbrace{y_r(t)}_{\substack{scalar\\output}} = \underbrace{c_r}_{qx1}{}^T x_r(t)$$

- q << N, but input/output behavior preserved

Reminder about Eigenanalysis

Consider an ODE: $\frac{dx(t)}{dt} = Ax(t) + bu(t), \quad x(0) = 0$

Eigendecomposition: $A = \underbrace{\begin{bmatrix} \vdots & \vdots & \vdots \\ E_1 & E_2 & E_N \\ \vdots & \vdots & \vdots \end{bmatrix}}_{E} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_N \end{bmatrix} \begin{bmatrix} \vdots & \vdots & \vdots \\ E_1 & E_2 & E_N \\ \vdots & \vdots & \vdots \end{bmatrix}^{-1}$

Change of variables: $Ew(t) = x(t) \Leftrightarrow w(t) = E^{-1}x(t)$

Substituting: $\frac{dEw(t)}{dt} = AEw(t) + bu(t), \quad Ew(0) = 0$

Multiply by E^{-1}: $\frac{dw(t)}{dt} = E^{-1}AEw(t) + E^{-1}bu(t)$

Reminder about Eigenanalysis Cont.

Decoupled Equations

$$\frac{d}{dt}\begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_N \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} + \underbrace{\begin{bmatrix} (E^{-1}b)_1 \\ \vdots \\ (E^{-1}b)_N \end{bmatrix}}_{\tilde{b}} u(t)$$

Output Equation

$$y(t) = c^T x(t) = c^T Ew(t) = \underbrace{(E^T c)}_{\tilde{c}}{}^T w(t)$$

Reminder about Eigenanalysis Cont.

Solving Decoupled Equations

$$w_i(t) = \int_0^t e^{\lambda_i (t-\tau)} \tilde{b}_i \, u(\tau) \, d\tau$$

Assuming Zero Initial Conditions

Output Equation

$$y(t) = \sum_{i=1}^{N} \tilde{c}_i \, w_i(t)$$

Dynamic Linear Case — Reduced models via mode truncation

$$\begin{bmatrix} \dot{w}_1 \\ \vdots \\ \dot{w}_q \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_q \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_q \end{bmatrix} + \begin{bmatrix} \tilde{b}_1 \\ \vdots \\ \tilde{b}_q \end{bmatrix} u(t)$$

Output Equation

$$y(t) = \sum_{i=1}^{q} \tilde{c}_i \, w_i(t)$$

Dynamic Linear Case — Reduced models via mode Truncation

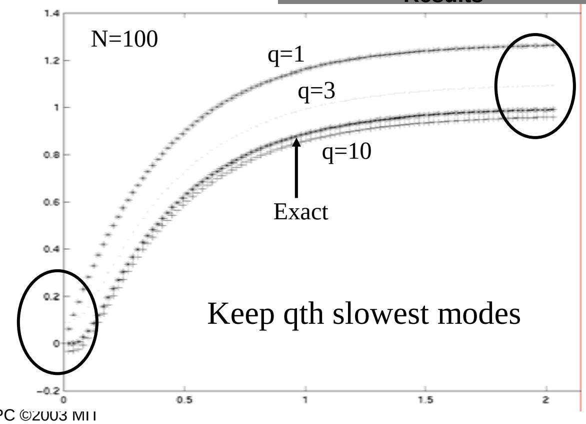
Why?

- Certain modes are not affected by the input
  - $\tilde{b}_{k+1}, \ldots, \tilde{b}_N$ are all small
- Certain modes do not affect the output
  - $\tilde{c}_{k+1}, \ldots, \tilde{c}_N$ are all small
- Keep least negative evals (slowest modes)
  - Look at response to a constant input

$$w_i(t) = \int_0^t e^{\lambda_i (t-\tau)} \tilde{b}_i \, u \, d\tau = \underbrace{\frac{1}{\lambda_i}\left(\tilde{b}_i u - \tilde{b}_i u e^{\lambda_i t}\right)}_{\text{Small if } |\lambda_i| \text{ large}}$$

Dynamic Linear Case — Reduced models via mode truncation

Heat Conducting bar Results



An Aside on Transfer Functions – Laplace Transform

Consider an ODE: $\frac{dx(t)}{dt} = Ax(t) + bu(t)$

Bilateral Laplace Transform: $X(s) = \int_{-\infty}^{\infty} x(t) e^{-st} dt$

Key Transform Property: $sX(s) = \int_{-\infty}^{\infty} \frac{dx(t)}{dt} e^{-st} dt$

Rewrite the ODE in transformed variables

$$sX(s) = AX(s) + bU(s) \quad Y(s) = c^T X(s)$$

$$\Rightarrow Y(s) = \underbrace{c^T (sI - A)^{-1} b}_{H(s)} U(s)$$

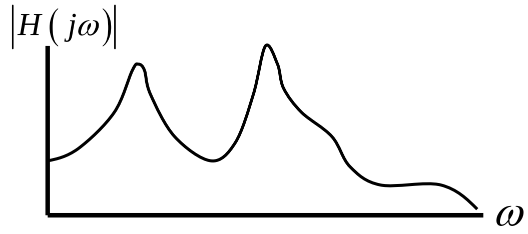
← Transfer Function

An Aside on Transfer Functions – Meaning of H(s)

For Stable Systems, H(j*w*) is the frequency response

If $u(t) = e^{j\omega t}$ ←Sinusoid

then $y(t) = H(j\omega) e^{j\omega t}$ Sinusoid with shifted phase and amplitude



An Aside on Transfer Functions – EigenAnalysis

Transfer Function

$$H(s) = c^T (sI - A)^{-1} b$$

Apply Eigendecomposition

$$H(s) = c^T E (sI - \lambda)^{-1} E^{-1} b$$

$$= \tilde{c}^T \begin{bmatrix} \frac{1}{s-\lambda_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{s-\lambda_N} \end{bmatrix} \tilde{b} \Rightarrow H(s) = \sum_{i=1}^{N} \frac{\tilde{c}_i \tilde{b}_i}{s - \lambda_i}$$

Dynamic Linear Case — Rational Transfer Function Representation

Original System Transfer Function

$$H(s) = \frac{\tilde{c}_1 \tilde{b}_1}{(s-\lambda_1)} + \cdots + \frac{\tilde{c}_N \tilde{b}_N}{(s-\lambda_N)} = \underbrace{\frac{b_0 + b_1 s + \cdots + b_{N-1} s^{N-1}}{1 + a_1 s + \cdots + a_N s^N}}_{\text{Rational Function}}$$

Reduced Model Transfer Function

$$H_r(s) = \underbrace{\frac{b_0^r + b_1^r s + \cdots + b_{q-1}^r s^{q-1}}{1 + a_1^r s + \cdots + a_q^r s^q}}_{\text{Lower Order Rational Function}}$$

Model Reduction = Find a low order rational function matching H(s)

Dynamic Linear Case — Rational Transfer Function Representation

Degrees of Freedom

Reduced Model Dynamical System

$$\frac{dx_r(t)}{dt} = \underbrace{A_r}_{qxq} x(t) + \underbrace{b_r}_{qx1} \underbrace{u(t)}_{\substack{scalar\\input}} \qquad \underbrace{y_r(t)}_{\substack{scalar\\output}} = \underbrace{c_r}_{qx1}{}^T x_r(t)$$

$2q+q^2$ coefficients

Reduced Model Transfer Function

$$H_r(s) = \frac{b_0^r + b_1^r s + \cdots + b_{q-1}^r s^{q-1}}{1 + a_1^r s + \cdots + a_q^r s^q}$$

$2q$ coefficients

Dynamic Linear Case — Rational Transfer Function Representation

Variable Changes Do not change transfer functions

Reduced Model Transfer Function

$$\frac{dx_r(t)}{dt} = A_r x(t) + b_r u(t) \qquad y_r(t) = c_r^T x_r(t)$$

$$\Rightarrow H(s) = c_r^T (sI - A_r)^{-1} b_r$$

Similarity (x = Sw) Transformed Transfer Function

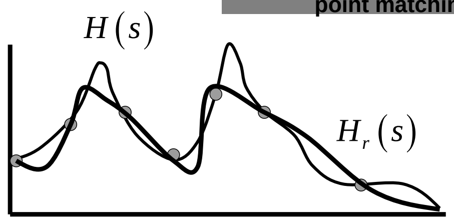
$$\frac{dw_r(t)}{dt} = S^{-1} A_r S w(t) + S^{-1} b_r u(t) \qquad y_r(t) = c_r^T S w_r(t)$$

$$\Rightarrow H(s) = c_r^T S \left(sI - S^{-1} A_r S\right)^{-1} S^{-1} b_r = c_r^T (sI - A_r)^{-1} b_r$$

Many Dynamical Systems have the same transfer function!!

Dynamic Linear Case — Rational Transfer Function Representation

Rational Function Fitting by point matching



- Can match 2q points
- cross multiplying generates a linear system

  For i = 1 to 2q

$$\left(1 + a_1^r s_i + \cdots + a_q^r s_i^{\,q}\right) H(s_i) - \left(b_0^r + b_1^r s + \cdots + b_{q-1}^r s^{q-1}\right) = 0$$

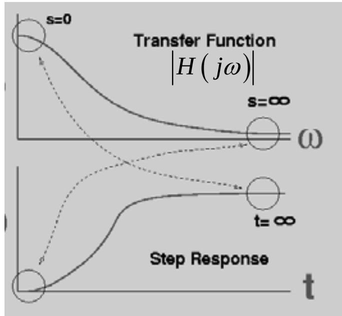
Dynamic Linear Case — Rational Transfer Function Representation

Point Matching Matrix can be ill-conditioned

$$\begin{bmatrix} s_1 H(s_1) & s_1^2 H(s_1) & \cdots & -s_1^{q-1} \\ \vdots & \vdots & \cdots & -s_2^{q-1} \\ \vdots & \vdots & \cdots & \vdots \\ s_{2q} H(s_{2q}) & s_{2q}^2 H(s_{2q}) & \cdots & -s_{2q}^{q-1} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ b_{q-1} \end{bmatrix} = \begin{bmatrix} H(s_1) \\ H(s_2) \\ \vdots \\ H(s_{2q}) \end{bmatrix}$$

- Columns contain progressively higher powers of the test frequencies
- Must orthogonalize columns during construction

Dynamic Linear Case
Rational Transfer Function Representation
Importance of Fitting at low frequency



Correct Steady State behavior requires accurate match at low frequencies

Dynamic Linear Case
Rational Transfer Function Representation
Taylor Series Expansion and Moments

Original System Transfer Function Moments

$$H(s) = c^T(sI - A)^{-1}b = -c^T \underbrace{(I - sA^{-1})^{-1}}_{\text{Taylor Expand with respect to s}} A^{-1}b$$

$$H(s) = -c^T(I - sA^{-1})^{-1}A^{-1}b = \sum_{k=0}^{\infty} c^T A^{-(k+1)} b s^k$$

$$H(s) = \underbrace{c^T A^{-1} b}_{m_0} + \underbrace{c^T A^{-2} b}_{m_1} s + \underbrace{c^T A^{-3} b}_{m_2} s^2 + \cdots = \sum_{k=0}^{\infty} m_k s^k$$

Moments

Dynamic Linear Case
Rational Transfer Function Representation
Moment Matching for accurate low frequency behavior

Reduced Model Matches Original Systems Moments

$$H_r(s) = \frac{b_0^r + b_1^r s + \cdots + b_{q-1}^r s^{q-1}}{1 + a_1^r s + \cdots + a_q^r s^q} = m_0 + m_1 s + \cdots + m_{2q-1} s + \cdots$$

Cross-Multiplying and Matching Terms

$$\begin{bmatrix} m_0 & m_1 & \cdots & m_{k-1} \\ m_1 & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & m_{2q-3} \\ m_{k-1} & \cdots & m_{2q-3} & m_{2q-2} \end{bmatrix} \begin{bmatrix} a_q \\ a_{q-1} \\ \vdots \\ a_1 \end{bmatrix} = \begin{bmatrix} m_q \\ m_{q+1} \\ \vdots \\ m_{2q-1} \end{bmatrix}$$

Dynamic Linear Case
Rational Transfer Function Representation
Explicit Moment Matching Problem

System of equations extremely ill-conditioned

$$\begin{bmatrix} m_0 & m_1 & \cdots & m_{k-1} \\ m_1 & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & m_{2q-3} \\ m_{k-1} & \cdots & m_{2q-3} & m_{2q-2} \end{bmatrix} \begin{bmatrix} a_q \\ a_{q-1} \\ \vdots \\ a_1 \end{bmatrix} = \begin{bmatrix} m_q \\ m_{q+1} \\ \vdots \\ m_{2q-1} \end{bmatrix}$$

$$m_i = c^T A^{-i} b \approx \lambda_{A_{\max}} m_{i-1}$$

Columns become linearly dependent for large q!

Dynamic Linear Case

Rational Transfer Function Representation

Problems with explicit fitting methods

- Linear Systems for fitting ill-conditioned
  - Need specialized algorithms which avoid explicit fitting matrix construction
- Rational function must be converted to state-space
  - Needed by most simulation tools
  - Requires root finding procedure, very sensitive to parameter variation

Summary

- Need For Model Reduction
  - Circuits, MEMS, Optics, Jet Engines
- Simple Example Problem
  - Heat Conducting bar example
- Steady-State Case (linear and nonlinear)
- Dynamic Linear Case
  - Truncating Eigenmodes
    - Loss correct steady state values
    - Select modes to delete
  - Rational Function Fitting
    - Generates ill-conditioned matrices