Computing Transfer Function Dominant Poles of Large Second-Order Systems

Joost Rommes

Mathematical Institute
Utrecht University
rommes@math.uu.nl
http://www.math.uu.nl/people/rommes

joint work with Nelson Martins
March 22, 2007

1. Introduction
2. Transfer functions and dominant poles
3. Quadratic Dominant Pole Algorithm
4. Results
5. Conclusions

Introduction

- Large-scale second-order dynamical systems arise in
  - electrical circuit simulation
  - structural engineering
  - acoustics
- Transfer function is used for
  - simulation
  - stability analysis
  - controller design
- Relatively few transfer function poles of practical importance
- Which poles, and how to compute?

Second-order dynamical systems

Second-order SISO dynamical system

$$\begin{cases} M\ddot{\mathbf{x}}(t) + C\dot{\mathbf{x}}(t) + K\mathbf{x}(t) & = & \mathbf{b}u(t) \\ y(t) & = & \mathbf{c}^*\mathbf{x}(t) + du(t), \end{cases}$$

where

$$\begin{array}{rcl} u(t), y(t), d & \in & \mathbb{R}, \text{ input, output, direct i/o} \\ \mathbf{x}(t), \mathbf{b}, \mathbf{c} & \in & \mathbb{R}^n, \text{ state, input-to-, -to-output,} \\ M & \in & \mathbb{R}^{n\times n} \text{ mass matrix,} \\ C & \in & \mathbb{R}^{n\times n} \text{ damping matrix,} \\ K & \in & \mathbb{R}^{n\times n} \text{ stiffness matrix.} \end{array}$$

Transfer function

Second-order SISO dynamical system ($d = 0$):

$$\begin{cases} M\ddot{\mathbf{x}}(t) + C\dot{\mathbf{x}}(t) + K\mathbf{x}(t) & = & \mathbf{b}u(t), \\ y(t) & = & \mathbf{c}^*\mathbf{x}(t). \end{cases}$$

Second-order transfer function:

$$H(s) = \mathbf{c}^*(s^2M + sC + K)^{-1}\mathbf{b}.$$

- Poles are $\lambda \in \mathbb{C}$ for which

$$\det(\lambda^2 M + \lambda C + K) = 0$$

First-order transfer function $H(s) = \mathbf{c}^*(sE - A)^{-1}\mathbf{b}$

Can be expressed as

$$H(s) = \sum_{i=1}^{n} \frac{R_i}{s - \lambda_i},$$

where residues R_i are

$$R_i = (\mathbf{c}^*\mathbf{x}_i)(\mathbf{y}_i^*\mathbf{b}),$$

and $(\lambda_i, \mathbf{x}_i, \mathbf{y}_i)$ are eigentriplets $(i = 1, \ldots, n)$

$$\begin{array}{rcll} A\mathbf{x}_i & = & \lambda_i E\mathbf{x}_i, & \text{right eigenpairs} \\ \mathbf{y}_i^* A & = & \lambda_i \mathbf{y}_i^* E, & \text{left eigenpairs} \\ \mathbf{y}_i^* E\mathbf{x}_i & = & 1, & \text{normalization} \\ \mathbf{y}_j^* E\mathbf{x}_i & = & 0 \ (i \neq j), & E\text{-orthogonality} \end{array}$$

Dominant poles of first-order transfer functions

$$H(s) = \sum_{i=1}^{n} \frac{R_i}{s - \lambda_i} \quad \text{with} \quad R_i = (\mathbf{c}^* \mathbf{x}_i)(\mathbf{y}_i^* \mathbf{b})$$

- Pole λ_i *dominant* if $\frac{|R_i|}{|\mathrm{Re}(\lambda_i)|}$ large
- Dominant poles cause peaks in Bode-plot $(\omega, |H(i\omega)|)$
- Effective transfer function behavior:

$$H_k(s) = \sum_{i=1}^{k} \frac{R_i}{s - \lambda_i},$$

where $k \ll n$ and (λ_i, R_i) ordered by decreasing dominance

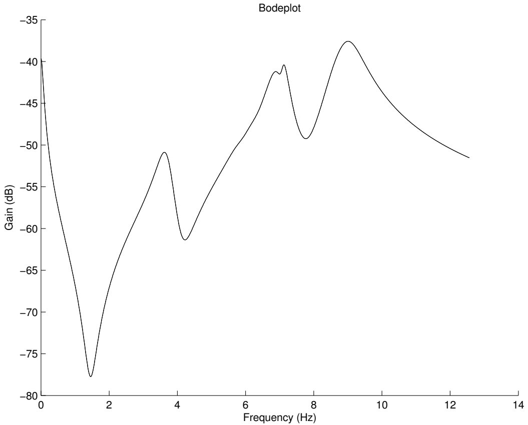

Figure: Bode plot $(\omega, |H(i\omega)|)$. Pole λ_j *dominant* if $\frac{|R_j|}{|\mathrm{Re}(\lambda_j)|}$ large.

2nd-order transfer function $H(s) = \mathbf{c}^*(s^2M + sC + K)^{-1}\mathbf{b}$

Can be expressed as

$$H(s) = \sum_{i=1}^{2n} \frac{R_i}{s - \lambda_i},$$

where residues R_i are

$$R_i = (\mathbf{c}^*\mathbf{x}_i)(\mathbf{y}_i^*\mathbf{b})\lambda_i,$$

and $(\lambda_i, \mathbf{x}_i, \mathbf{y}_i)$ eigentriplets of QEP $(i = 1, \ldots, 2n)$

$$\begin{array}{rcll} (\lambda_i^2 M + \lambda_i C + K)\mathbf{x}_i & = & 0, & \text{right eigenpairs} \\ \mathbf{y}_i^*(\lambda_i^2 M + \lambda_i C + K) & = & 0, & \text{left eigenparis} \\ -\mathbf{y}_i^* K\mathbf{x}_i + \lambda_i^2 \mathbf{y}_i^* M\mathbf{x}_i & = & 1, & \text{normalization} \end{array}$$

Dominant poles of second-order transfer functions

$$H(s) = \sum_{i=1}^{2n} \frac{R_i}{s-\lambda_i} \quad \text{with} \quad R_i = (\mathbf{c}^*\mathbf{x}_i)(\mathbf{y}_i^*\mathbf{b})\lambda_i$$

- Pole λ_i *dominant* if $\frac{|R_i|}{|\mathrm{Re}(\lambda_i)|}$ large
- Dominant poles cause peaks in Bode-plot $(\omega, |H(i\omega)|)$
- Effective transfer function behavior:

$$H_k(s) = \sum_{i=1}^{k} \frac{R_i}{s-\lambda_i},$$

where $k \ll 2n$ and (λ_i, R_i) ordered by decreasing dominance

Quadratic Dominant Pole Algorithm (QDPA)

QDPA [R., Martins (2007)] computes dominant poles of

$$H(s) = \mathbf{c}^*(s^2 M + sC + K)^{-1}\mathbf{b}$$

1. Newton scheme
2. Subspace acceleration and selection
3. Deflation

Quadratic Dominant Pole Algorithm

Dominant pole λ of $H(s) = \mathbf{c}^*(s^2M + sC + K)^{-1}\mathbf{b}$:

$$\lim_{s \to \lambda} \frac{1}{H(s)} = 0$$

Apply Newton to $1/H(s)$:

$$\begin{aligned} s_{k+1} &= s_k + \frac{1}{H(s_k)} \frac{H^2(s_k)}{H'(s_k)} \\ &= s_k - \frac{\mathbf{c}^*(s_k^2M + s_kC + K)^{-1}\mathbf{b}}{\mathbf{c}^*(s_k^2M + s_kC + K)^{-1}(2s_kM + C)(s_k^2M + s_kC + K)^{-1}\mathbf{b}} \\ &= s_k - \frac{\mathbf{c}^*\mathbf{v}}{\mathbf{w}^*(2s_kM + C)\mathbf{v}}, \end{aligned}$$

where $\mathbf{v} = (s_k^2M + s_kC + K)^{-1}\mathbf{b}$ and $\mathbf{w} = (s_k^2M + s_kC + K)^{-*}\mathbf{c}$.

Quadratic Dominant Pole Algorithm [R., Martins (2007)]

Initial pole estimate s_1, tolerance $\epsilon \ll 1$

for $k = 1, 2, \dots$ **do**

  Solve $\mathbf{v}_k \in \mathbb{C}^n$ from $(s_k^2 M + s_k C + K)\mathbf{v}_k = \mathbf{b}$

  Solve $\mathbf{w}_k \in \mathbb{C}^n$ from $(s_k^2 M + s_k C + K)^* \mathbf{w}_k = \mathbf{c}$

  Compute the new pole estimate

$$s_{k+1} = s_k - \frac{\mathbf{c}^* \mathbf{v}_k}{\mathbf{w}_k^* (2 s_k M + C)\mathbf{v}_k}$$

  The pole $\lambda = s_{k+1}$ with $\mathbf{x} = \mathbf{v}_k$ and $\mathbf{y} = \mathbf{w}_k$ has converged if

$$\|(s_{k+1}^2 M + s_{k+1} C + K)\mathbf{v}_k\|_2 < \epsilon$$

end for

Subspace acceleration and selection

- Keep approximations $\mathbf{v}_k$ and $\mathbf{w}_k$ in search spaces V and W
- Petrov-Galerkin leads to projected QEP

$$\begin{aligned}(\theta^2\widetilde{M} + \theta\widetilde{C} + \widetilde{K})\tilde{\mathbf{x}} &= 0,\\ \tilde{\mathbf{y}}^*(\theta^2\widetilde{M} + \theta\widetilde{C} + \widetilde{K}) &= 0,\end{aligned}$$

where

$$\widetilde{M} = W^*MV, \quad \widetilde{C} = W^*CV, \text{ and } \widetilde{K} = W^*KV \in \mathbb{C}^{k\times k}.$$

- Gives $2k$ approximations $(\theta_i, \hat{\mathbf{x}}_i = V\tilde{\mathbf{x}}_i, \hat{\mathbf{y}}_i = W\tilde{\mathbf{y}}_i)$ in iteration k
- Select approximation with largest residue:

$$s_{k+1} = \text{argmax}_i \left| \frac{(\mathbf{c}^*\hat{\mathbf{x}}_i)(\hat{\mathbf{y}}_i^*\mathbf{b})\theta_i}{\text{Re}(\theta_i)} \right| \quad (\text{with } \|\hat{\mathbf{x}}_i\| = \|\hat{\mathbf{y}}_i\|_2 = 1)$$

Deflation for $H(s) = \mathbf{c}^*(sE - A)^{-1}\mathbf{b}$

- Triplet $(\lambda, \mathbf{x}, \mathbf{y})$: $A\mathbf{x} = \lambda\mathbf{x}$ and $\mathbf{y}^*A = \lambda\mathbf{y}^*E$
- New search spaces: $V \perp E^*\mathbf{y}$ and $W \perp E\mathbf{x}$
- Deflate via (every iteration)

$$\begin{aligned} \mathbf{v}_k &\leftarrow (I - \mathbf{x}\mathbf{y}^*E)\mathbf{v}_k \\ \mathbf{w}_k &\leftarrow (I - \mathbf{y}\mathbf{x}^*E^*)\mathbf{w}_k \end{aligned}$$

- More efficient: deflate only once

$$\begin{aligned} \mathbf{b}_d \leftarrow (I - E\mathbf{x}\mathbf{y}^*)\mathbf{b} &\Rightarrow \mathbf{v}_k = (s_kE - A)^{-1}\mathbf{b}_d \perp E^*\mathbf{y} \\ \mathbf{c}_d \leftarrow (I - E^*\mathbf{y}\mathbf{x}^*)\mathbf{c} &\Rightarrow \mathbf{w}_k = (s_kE - A)^{-*}\mathbf{c}_d \perp E\mathbf{x} \end{aligned}$$

- Note that $\mathbf{y}^*\mathbf{b}_d = \mathbf{c}_d^*\mathbf{x} = 0$

Deflation for $H(s) = \mathbf{c}^*(s^2M + sC + K)^{-1}\mathbf{b}$

- Eigenvectors can be linearly dependent: no direct deflation
- Consider linearization of second-order system:

$$\begin{cases} E\dot{\mathbf{x}}(t) & = & A\mathbf{x}(t) + \mathbf{b}_l u(t) \\ y(t) & = & \mathbf{c}_l^*\mathbf{x}(t), \end{cases}$$

where (if K nonsingular)

$$A = \begin{bmatrix} 0 & -K \\ -K & -C \end{bmatrix}, \quad E = \begin{bmatrix} -K & 0 \\ 0 & M \end{bmatrix}, \quad \mathbf{b}_l = \begin{bmatrix} 0 \\ \mathbf{b} \end{bmatrix}, \quad \mathbf{c}_l = \begin{bmatrix} \mathbf{c} \\ 0 \end{bmatrix}.$$

- $$(\lambda^2 M + \lambda C + K)\mathbf{x} = 0 \quad \Longleftrightarrow \quad A\begin{bmatrix} \mathbf{x} \\ \lambda\mathbf{x} \end{bmatrix} = \lambda E \begin{bmatrix} \mathbf{x} \\ \lambda\mathbf{x} \end{bmatrix}$$
- Idea [Meerbergen, SISC (2001)]: deflate via linearization

Deflation via linearization 1/2

- Triplets $(\lambda, \mathbf{x}, \mathbf{y})$ and $(\lambda, \phi = [\mathbf{x}^T, \lambda \mathbf{x}^T]^T, \psi = [\mathbf{y}^T, \bar{\lambda}\mathbf{y}^T]^T)$
- Deflate via

$$\mathbf{b}_d = \begin{bmatrix} \mathbf{b}_d^1 \\ \mathbf{b}_d^2 \end{bmatrix} = (I - E\phi\psi^*) \begin{bmatrix} 0 \\ \mathbf{b} \end{bmatrix}, \text{ and } \mathbf{c}_d = \begin{bmatrix} \mathbf{c}_d^1 \\ \mathbf{c}_d^2 \end{bmatrix} = (I - E^*\psi\phi^*) \begin{bmatrix} \mathbf{c} \\ 0 \end{bmatrix}$$

- Solve expansion vectors $\mathbf{v}_k$ and $\mathbf{w}_k$ from

$$\left(s_k \begin{bmatrix} -K & 0 \\ 0 & M \end{bmatrix} - \begin{bmatrix} 0 & -K \\ -K & -C \end{bmatrix} \right) \begin{bmatrix} \mathbf{v}_k \\ \mathbf{e} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_d^1 \\ \mathbf{b}_d^2 \end{bmatrix}$$

and

$$\left(s_k \begin{bmatrix} -K & 0 \\ 0 & M \end{bmatrix} - \begin{bmatrix} 0 & -K \\ -K & -C \end{bmatrix} \right)^* \begin{bmatrix} \mathbf{w}_k \\ \mathbf{f} \end{bmatrix} = \begin{bmatrix} \mathbf{c}_d^1 \\ \mathbf{c}_d^2 \end{bmatrix}$$

Deflation via linearization 2/2

Solve $\mathbf{v}_k$ from

$$\left(s_k \begin{bmatrix} -K & 0 \\ 0 & M \end{bmatrix} - \begin{bmatrix} 0 & -K \\ -K & -C \end{bmatrix} \right) \begin{bmatrix} \mathbf{v}_k \\ \mathbf{e} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_d^1 \\ \mathbf{b}_d^2 \end{bmatrix}$$

- With $\mathbf{e} = (s_k^2 M + s_k C + K)^{-1}(s_k \mathbf{b}_d^2 + \mathbf{b}_d^1)$ follows
$$\mathbf{v}_k = (-K^{-1}\mathbf{b}_d^1 + \mathbf{e})/s_k$$
- Similarly $\mathbf{w}_k = (-K^{-*}\mathbf{c}_d^1 + \mathbf{f})/\bar{s}_k$
- Solve projected QEP
- Select largest approximate linearized residue:

$$s_{k+1} = \operatorname{argmax}_i \left| \frac{(\mathbf{c}_d^* \begin{bmatrix} \hat{\mathbf{x}}_i \\ \theta_i \hat{\mathbf{x}}_i \end{bmatrix})(\begin{bmatrix} \hat{\mathbf{y}}_i^* & \theta_i \hat{\mathbf{y}}_i^* \end{bmatrix} \mathbf{b}_d)}{\mathrm{Re}(\theta_i)} \right|$$

Butterfly gyro (Oberwolfach) ($n = 17361$)

- Modal approximation constructed by using left and right eigenspaces Y and X:

$$M_k = Y^*MX, \; C_k = Y^*CX, \; K_k = Y^*KX, \; \mathbf{b}_k = Y^*\mathbf{b}, \; \mathbf{c}_k = X^*\mathbf{c}$$

- Krylov reduced order model uses $Y = X$ with

$$\text{colspan}(X) = \mathcal{K}^k(K^{-1}M, K^{-1}\mathbf{b})$$

- Krylov model is cheaper: (20 s, 40 iters) vs. (1345 s, 233 iters)
- Modal approximation preserves poles

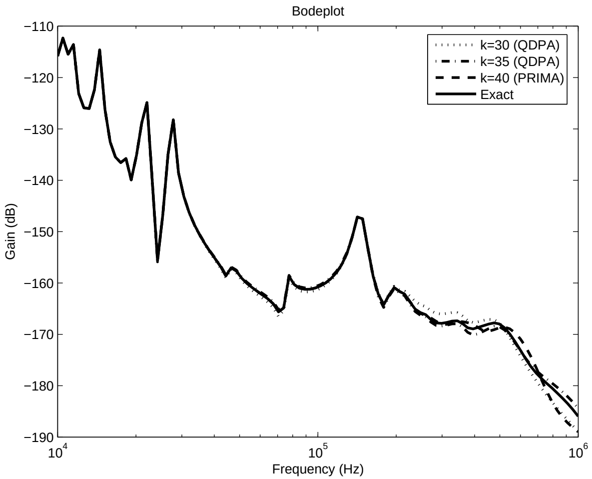

Figure: Butterfly gyro ($n = 17361$). Exact response (solid), 40th order Krylov model (dash), 35th (dash-dot) and 30th (dot) order modal approximations.

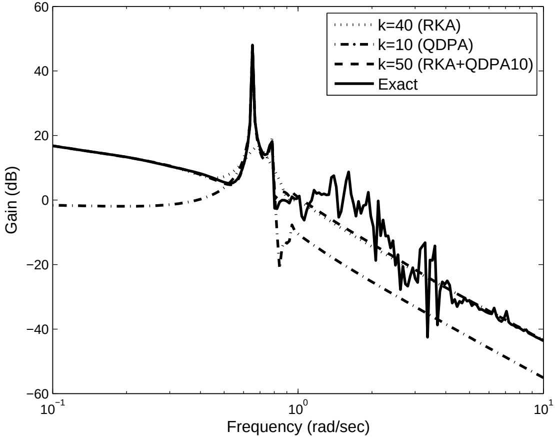

Figure: Breathing sphere ($n = 17611$). Exact transfer function (solid), 40th order SOAR RKA model (dot), 10th (dash-dot) order modal equivalent, and 50th order hybrid RKA+QDPA (dash).

Breathing sphere ([Lampe, Voss (2006)]) ($n = 17611$)

- Two-sided rational SOAR [Bai and Su, SISC (2005)] model ($k = 40$, shifts 0.1, 0.5, 1, 5) misses peaks
- Small QDPA model ($k = 10$) matches some peaks, misses global response
- 500 s (SOAR, 80 iters) vs. 2800 s (QDPA, 108 iters)
- Hybrid: $Y = [Y_{\mathrm{QDPA}}, Y_{\mathrm{SOAR}}]$ and $X = [X_{\mathrm{QDPA}}, X_{\mathrm{SOAR}}]$
- Larger SOAR models: no improvement
- More poles with QDPA: expensive
- Use imag parts of poles as shifts for SOAR!
- Shifts $\sigma_1 = 0.65i$, $\sigma_2 = 0.78i$, $\sigma_3 = 0.93i$, and $\sigma_4 = 0.1$
- Two-sided rational SOAR, 10-dimensional bases

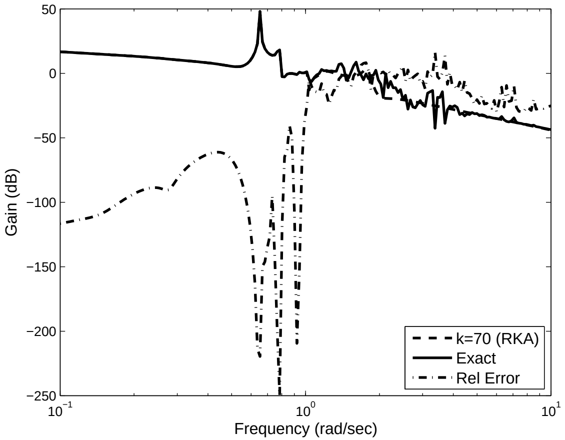

Figure: Breathing sphere ($n = 17611$). Exact transfer function (solid), 70th order SOAR RKA model (dash) using interpolation points based on dominant poles, and relative error (dash-dot).

Concluding remarks

- QDPA for computation of dominant poles
- Subspace acceleration and efficient deflation
- Applications in model order reduction:
  - Construction of modal approximations
  - Interpolation points for rational Krylov
  - Preservation of poles

Generalizations:

- Higher-order systems
- MIMO systems
- Computation of dominant zeros z: $H(z) = 0$

For preprints and more info, see
http://www.math.uu.nl/people/rommes