QZ Based Algorithm for System Pole, Transmission Zero and Residue Derivatives

Bradley T. Burchett, Ph.D., P.E.
Associate Professor
Rose-Hulman Institute of Technology

Overview

- Introduction
- Review of QZ
- Batch Calculation of the Residues
- Numerical Results
- Gradient based OOF using Sylvester
- Open issues

Introduction

- Objective: Numerically robust algorithm to compute derivatives of poles, zeros and residues
- Output will be used in Sylvester expansion solution of quadratic cost function for optimal output feedback
- Facilitates feedback design in large systems.

QZ review

- QZ historically used to solve generalized eigenvalue problem (GEP)

$$\mathbf{A}\mathbf{w} = \lambda \mathbf{B}\mathbf{w}$$

- In turn used to solve Ricatti Equations, compute transmission zeros, etc.

Steps of QZ

- **A** is reduced to upper Hessenberg form while **B** is reduced to upper triangular form,
- **A** is reduced to quasi-triangular form while the triangular form of **B** is maintained
- the quasi-triangular matrix is reduced to triangular form and the eigenvalues are extracted
- the eigenvectors are obtained from the triangular matrices and transformed back into the original coordinate system

Building blocks of QZ

- QZ refers to the left and right unitary matrices Q and Z such that QAZ is quasi-triangular and QBZ is upper triangular
- Q and Z need not be explicitly formed, rather, Householder reflectors and Givens rotations are applied to submatrices of A and B.

Householder / Givens

- Householder finds:

$$\mathbf{H} = \mathbf{I} - \mathbf{u}\mathbf{u}^H$$

- Such that

$$\mathbf{H}\mathbf{a} = \nu\mathbf{e}_1$$

- Givens rotation:

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix} \cdot = \nu\mathbf{e}_1$$

Overall QZ

- Problem is sub-divided to reduce computational burden

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ 0 & \mathbf{A}_{22} & \mathbf{A}_{23} \\ 0 & 0 & \mathbf{A}_{33} \end{bmatrix} \begin{matrix} p \\ n-p-q \\ q \end{matrix}$$
$$\qquad \begin{matrix} p & n-p-q & q \end{matrix}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \mathbf{B}_{13} \\ 0 & \mathbf{B}_{22} & \mathbf{B}_{23} \\ 0 & 0 & \mathbf{B}_{33} \end{bmatrix} \begin{matrix} p \\ n-p-q \\ q \end{matrix}$$
$$\qquad \begin{matrix} p & n-p-q & q \end{matrix}$$

- Rotations are applied to pencil and its derivatives

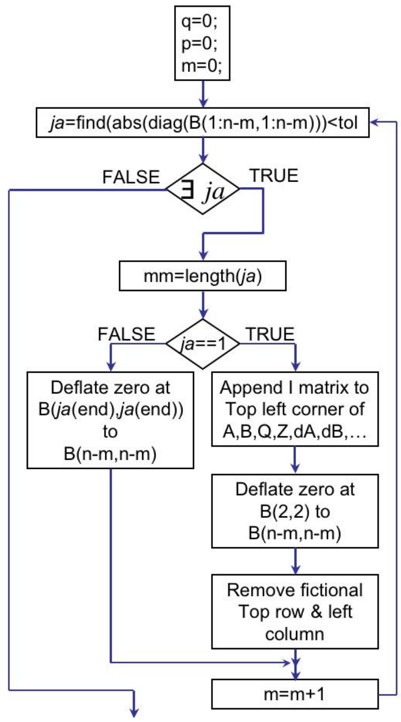
$$
\begin{aligned}
\frac{\partial \mathbf{A}}{\partial \psi} &= \mathtt{diag}(0_p, d\mathbf{Q}, 0_q)^T * \mathbf{A} * \mathtt{diag}(\mathbf{I}_p, \mathbf{Z}, \mathbf{I}_q)^T \\
&+ \mathtt{diag}(\mathbf{I}_p, \mathbf{Q}, \mathbf{I}_q)^T * d\mathbf{A} * \mathtt{diag}(\mathbf{I}_p, \mathbf{Z}, \mathbf{I}_q) \\
&+ \mathtt{diag}(\mathbf{I}_p, \mathbf{Q}, \mathbf{I}_q)^T * \mathbf{A} * \mathtt{diag}(0_p, d\mathbf{Z}, 0_q)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \mathbf{B}}{\partial \psi} &= \mathtt{diag}(0_p, d\mathbf{Q}, 0_q)^T * \mathbf{B} * \mathtt{diag}(\mathbf{I}_p, \mathbf{Z}, \mathbf{I}_q)^T \\
&+ \mathtt{diag}(\mathbf{I}_p, \mathbf{Q}, \mathbf{I}_q)^T * d\mathbf{B} * \mathtt{diag}(\mathbf{I}_p, \mathbf{Z}, \mathbf{I}_q) \\
&+ \mathtt{diag}(\mathbf{I}_p, \mathbf{Q}, \mathbf{I}_q)^T * \mathbf{B} * \mathtt{diag}(0_p, d\mathbf{Z}, 0_q)
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{A} &= \mathtt{diag}(\mathbf{I}_p, \mathbf{Q}, \mathbf{I}_q)^T * \mathbf{A} * \mathtt{diag}(\mathbf{I}_p, \mathbf{Z}, \mathbf{I}_q) \\
\mathbf{B} &= \mathtt{diag}(\mathbf{I}_p, \mathbf{Q}, \mathbf{I}_q)^T * \mathbf{B} * \mathtt{diag}(\mathbf{I}_p, \mathbf{Z}, \mathbf{I}_q)
\end{aligned}
$$

Deflation

- zeros on the diagonal of **B** must deflated out the bottom of the pencil to preserve the correct derivative information.
- When **B** initially has two or more zeros on the diagonal the deflation logic becomes somewhat more complicated. as shown below.



Extracting the Eigenvalues

- Real eigs given by: $\lambda_i = a_{i,i}/b_{i,i}$
- Derivs by: $\frac{\partial \lambda}{\partial \psi} = \frac{1}{b_{i,i}}\frac{\partial a_{i,i}}{\partial \psi} - \frac{\partial b_{i,i}}{\partial \psi}\frac{a_{i,i}}{b_{i,i}^2}$
- Complex eigs from the algorithm:

$$
\begin{aligned}
\mu &= a_{1,1}/b_{1,1} \\
\frac{\partial \mu}{\partial \psi} &= \frac{1}{b_{1,1}}\frac{\partial a_{1,1}}{\partial \psi} - \frac{\partial b_{1,1}}{\partial \psi}\frac{a_{1,1}}{b_{1,1}^2} \\
a_{1,2}^{\dagger} &= a_{1,2} - \mu b_{1,2} \\
\frac{\partial a_{1,2}^{\dagger}}{\partial \psi} &= \frac{\partial a_{1,2}}{\partial \psi} - \frac{\partial \mu}{\partial r} b_{1,2} - \mu \frac{\partial b_{1,2}}{\partial \psi}
\end{aligned}
$$

Complex eigs algorithm continued

$$a^\dagger_{2,2} = a_{2,2} - \mu b_{2,2}$$

$$\frac{\partial a^\dagger_{2,2}}{\partial \psi} = \frac{\partial a_{2,2}}{\partial \psi} - \frac{\partial \mu}{\partial r} b_{2,2} - \mu \frac{\partial b_{2,2}}{\partial \psi}$$

$$p = \frac{1}{2}\left(\frac{a^\dagger_{2,2}}{b_{2,2}} - \frac{b_{1,2}a_{2,1}}{b_{1,1}b_{2,2}}\right)$$

$$\frac{\partial p}{\partial \psi} = \frac{1}{2b_{2,2}}\left(\frac{\partial a^\dagger_{2,2}}{\partial \psi} - \frac{a^\dagger_{2,2}}{b_{2,2}}\frac{\partial b_{2,2}}{\partial \psi} - \frac{\partial b_{1,2}}{\partial \psi}\frac{a_{2,1}}{b_{1,1}} - \frac{\partial a_{2,1}}{\partial \psi}\frac{b_{1,2}}{b_{1,1}} + \frac{b_{1,2}a_{2,1}}{b_{1,1}^2}\frac{\partial b_{1,1}}{\partial \psi} + \frac{b_{1,2}a_{2,1}}{b_{1,1}}\frac{\partial b_{2,2}}{\partial \psi}\right)$$

$$q = \frac{a_{2,1}a^\dagger_{1,2}}{b_{1,1}b_{2,2}}$$

$$\frac{\partial q}{\partial \psi} = \frac{1}{b_{1,1}b_{2,2}}\left(\frac{\partial a_{2,1}}{\partial \psi}a^\dagger_{1,2} + a_{2,1}\frac{\partial a^\dagger_{1,2}}{\partial \psi} - \frac{a_{2,1}a^\dagger_{1,2}}{b_{1,1}}\frac{\partial b_{1,1}}{\partial \psi} - \frac{a_{2,1}a^\dagger_{1,2}}{b_{2,2}}\frac{\partial b_{2,2}}{\partial \psi}\right)$$

and finally

$$r = p^2 + q$$

$$\frac{\partial r}{\partial \psi} = 2p\frac{\partial p}{\partial \psi} + \frac{\partial q}{\partial \psi}$$

$$\lambda = \mu + p + \texttt{sign}(p)\sqrt{r}$$

$$\frac{\partial \lambda}{\partial \psi} = \frac{\partial \mu}{\partial \psi} + \frac{\partial p}{\partial \psi} + \texttt{sign}(p)\frac{1}{2}(r)^{-1/2}\frac{\partial r}{\partial \psi}$$

LTI System nomenclature

- State Space

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{ax} + \mathbf{bu} \\ \mathbf{y} &= \mathbf{cx} + \mathbf{du}\end{aligned}$$

- Transfer Function

$$F(s) = \mathbf{c}(s\mathbf{I} - \mathbf{a})^{-1}\mathbf{b} + \mathbf{d}$$

$$F(s) = \frac{B(s)}{A(s)}$$

Batch Calculation of the Residues (distinct poles)

- Partial fraction expansion:

$$F(s) = \frac{B(s)}{A(s)}$$

$$\frac{B(s)}{A(s)} = \frac{K\,(s+z_1)\,(s+z_2)\cdots(s+z_m)}{(s+p_1)\,(s+p_2)\cdots(s+p_n)}, \texttt{ for } m < n$$

$$\frac{B(s)}{A(s)} = \frac{r_1}{(s+p_1)} + \frac{r_2}{(s+p_2)} + \cdots + \frac{r_n}{(s+p_n)}$$

- Can be solved by finding common denominator, adding, and equating powers of *s.*

- If we write the simultaneous set of equations in matrix form we get:

- **Ξr=R**, where:

$$\Xi_{i,j} = \sum_{\substack{k,l=\binom{n-1}{i-1} \\ k,l\neq j}} (-p_k)(-p_l)\cdots$$

```
Xi(i,j) = sum(prod(combnk(-ps([1:j-1,j+1:n]),i-1),2)).
```

- and $\mathbf{R} = [\eta_1 \ \eta_2 \ \cdots \ \eta_n]^T$

$$B(s) = \eta_1 s^{n-1} + \eta_2 s^{n-2} + \cdots + \eta_n$$

- that is:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \sum_{k=2}^{n} -p_k & \sum_{\substack{k=1 \\ k\neq 2}}^{n} -p_k & \cdots & \sum_{k=1}^{n-1} -p_k \\ \sum_{\substack{k,l=\binom{n-1}{2} \\ k,l\neq 1}} (-p_k)(-p_l) & \sum_{\substack{k,l=\binom{n-1}{2} \\ k,l\neq 2}} (-p_k)(-p_l) & \cdots & \sum_{\substack{k,l=\binom{n-1}{2} \\ k,l\neq n}} (-p_k)(-p_l) \\ \vdots & \vdots & & \vdots \\ \prod_{k=2}^{n} (-p_k) & \prod_{\substack{k=1 \\ k\neq 2}}^{n} (-p_k) & \cdots & \prod_{k=1}^{n-1} (-p_k) \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{Bmatrix} = \begin{Bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta n \end{Bmatrix}$$

Batch Calculation of the Residues (repeated poles)

$$
\begin{aligned}
\frac{B(s)}{A(s)} &= \frac{r_1}{(s+p_1)^{m_k}} + \frac{r_2}{(s+p_1)^{m_k-1}} + \cdots + \frac{r_{m_k}}{(s+p_1)} \\
&+ \cdots + \frac{r_{m_k+1}}{(s+p_2)} + \cdots \frac{r_n}{(s+p_n)}
\end{aligned}
$$

- Repeating the algebraic process, we discover that the corresponding column of the **Ξ** matrix can be built from the bottom up pretending that the system is lacking $m_k - j + 1$ occurrences of the repeated pole. The top $m_k - j$ of said column will then be filled in with zeros.

Columns of Ξ corresponding to repeated pole:

$$
\left[\begin{array}{ccccc}
0 & 0 & \cdots & 1 & \\
\vdots & \vdots & \cdots & \sum_{k=1}^{n-1}(-p_k) - (m_k-1)\,p_1 & \\
0 & 1 & \cdots & \sum_{\substack{k,l=\binom{n-1}{2} \\ k,l\neq n}} (-p_k)(-p_l) & \\
1 & \sum_{\substack{k=1 \\ k\neq 2}}^{n} -p_k - p_1 & \cdots & \sum_{\substack{k,l,m=\binom{n-1}{3} \\ k,l\neq n}} (-p_k)(-p_l)(-p_m) & \\
\sum_{k=2}^{n} -p_k & \sum_{\substack{k,l=\binom{n-m_k+1}{2} \\ k,l\neq 2}} (-p_k)(-p_l) & \cdots & \vdots & \cdots \\
\sum_{\substack{k,l=\binom{n-1}{2} \\ k,l\neq 1}} (-p_k)(-p_l) & \sum_{\substack{k,l,m=\binom{n-m_k+1}{3} \\ k,l\neq 1}} (-p_k)(-p_l)(-p_m) & \cdots & \vdots & \\
\vdots & \vdots & & \vdots & \\
\prod_{k=2}^{n}(-p_k) & p_1 \prod_{k=2}^{n}(-p_k) & \cdots & p_1^{m_k-1}\prod_{k=2}^{n}(-p_k) &
\end{array}\right.
$$

Derivatives of the Residues

- The batch equation is differentiated yielding: $\frac{\partial \mathbf{r}}{\partial \mathbf{K}} = \frac{\partial \boldsymbol{\Xi}^{-1}}{\partial \mathbf{K}}\mathbf{R} + \boldsymbol{\Xi}^{-1}\frac{\partial \mathbf{R}}{\partial \mathbf{K}}$

$$\frac{\partial \boldsymbol{\Xi}^{-1}}{\partial \mathbf{K}} = -\boldsymbol{\Xi}^{-1}\frac{\partial \boldsymbol{\Xi}}{\partial \mathbf{K}}\boldsymbol{\Xi}^{-1}.$$

$$\frac{\partial \mathbf{r}}{\partial \mathbf{K}} = -\boldsymbol{\Xi}^{-1}\frac{\partial \boldsymbol{\Xi}}{\partial \mathbf{K}}\boldsymbol{\Xi}^{-1}\mathbf{R} + \boldsymbol{\Xi}^{-1}\frac{\partial \mathbf{R}}{\partial \mathbf{K}}$$

$$\frac{\partial \boldsymbol{\Xi}_{i,j}}{\partial \mathbf{K}} = \sum_{\substack{m=1\\ m\neq j}}^{n}\left(-\frac{\partial p_m}{\partial \mathbf{K}}\right)\prod_{\substack{s=\binom{n-2}{i-2}\\ s\neq j}}(-p_s)$$

- Unpacking the derivative of the $\boldsymbol{\Xi}$ matrix, we get:

$$\begin{bmatrix}
0 & 0 & \cdots & 0 \\
\sum_{i=2}^{n} -\frac{\partial p_i}{\partial \mathbf{K}} & \sum_{\substack{i=1\\ i\neq 2}}^{n} -\frac{\partial p_i}{\partial \mathbf{K}} & \cdots & \sum_{i=1}^{n-1} -\frac{\partial p_i}{\partial \mathbf{K}} \\
\vdots & \vdots & & \vdots \\
\sum_{m=2}^{n}\left(-\frac{\partial p_m}{\partial \mathbf{K}}\right)\prod_{\substack{s=\binom{n-2}{i-2}\\ s\neq 1}}(-p_s) & \sum_{\substack{m=1\\ m\neq 2}}^{n}\left(-\frac{\partial p_m}{\partial \mathbf{K}}\right)\prod_{\substack{s=\binom{n-2}{i-2}\\ s\neq 2}}(-p_s) & \cdots & \sum_{m=1}^{n-1}\left(-\frac{\partial p_m}{\partial \mathbf{K}}\right)\prod_{\substack{s=\binom{n-2}{i-2}\\ s\neq n}}(-p_s) \\
\vdots & \vdots & & \vdots \\
\sum_{m=2}^{n}\left(-\frac{\partial p_m}{\partial \mathbf{K}}\right)\prod_{\substack{s=2\\ s\neq m}}(-p_s) & \sum_{\substack{m=1\\ m\neq 2}}^{n}\left(-\frac{\partial p_m}{\partial \mathbf{K}}\right)\prod_{\substack{s=1\\ s\neq 2\\ s\neq m}}^{n}(-p_s) & \cdots & \sum_{m=1}^{n-1}\left(-\frac{\partial p_m}{\partial \mathbf{K}}\right)\prod_{\substack{s=1\\ s\neq m}}^{n-1}(-p_s)
\end{bmatrix}$$

- System numerator polynomial is found from:

$$B(s) = \phi\left(\mathbf{a} - \mathbf{b}\mathbf{c}\right) + \left(\mathbf{d} - 1\right) A(s)$$

- Thus derivatives of the numerator coefficients from the pencil

$$\left[\frac{\partial \mathbf{a}}{\partial \mathbf{K}} - \frac{\partial \mathbf{b}}{\partial \mathbf{K}}\mathbf{c} - \mathbf{b}\frac{\partial \mathbf{c}}{\partial \mathbf{K}},\ 0\right]$$

$$\frac{\partial A(s)}{\partial \mathbf{K}} = \sum_j \frac{\partial A(s)}{\partial p_j(\mathbf{K})}.$$

$$\frac{\partial A(s)}{\partial \mathbf{K}} = \sum_j^n \left(-\frac{\partial p_j}{\partial \mathbf{K}}\right)\left[\sum_{m=0}^{n-1} \prod_{\substack{i=\binom{n-1}{m} \\ i \neq j}} (-p_i) s^{n-1-m}\right]$$

```
sum(prod(combnk(-ps([1:j-1,j+1:n]),m),2))*(-dps(j))
```

Numerical Examples

$$\mathbf{a} = \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 1 & -1 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \end{bmatrix} \qquad \frac{\partial \mathbf{a}}{\partial \psi} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Table 1. Pole derivative results

real(Pole)	imag(pole)	real(Derivative)	real(FD)	imag(Derivative)	imag(FD)
-3.38389	0	-0.617431	-0.617461	0	0
-2.19994	0	0.666242	0.666232	0	0
-0.624778	∓1.34337	0.755544	0.755531	∓0.160447	∓0.160355
-0.0833092	±0.487702	1.22005	1.22008	±0.046311	± 0.0461307

Transmission Zero Calculation

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{a}\mathbf{x} + \mathbf{b}\mathbf{u} \\ \mathbf{y} &= \mathbf{c}\mathbf{x} + \mathbf{d}\mathbf{u} \end{aligned}$$

$$\mathbf{A} = \left[\begin{array}{c|c} \mathbf{a} & \mathbf{b} \\ \hline \mathbf{c} & \mathbf{d} \end{array}\right]$$

$$\mathbf{B} = \left[\begin{array}{c|c} \mathbf{I} & 0 \\ \hline 0 & 0 \end{array}\right]$$

$$\mathbf{A} = \left[\begin{array}{c|c} \mathbf{a} & \mathbf{b} \\ \hline \mathbf{c} & d \end{array}\right] = \left[\begin{array}{cccccc|c} -2 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & -2 & 1 & 0 & 1 & -1 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right]$$

$$\frac{\partial \mathbf{A}}{\partial \mathbf{K}} = \left[\begin{array}{c|c} \frac{\partial \mathbf{a}}{\partial \mathbf{K}} & \frac{\partial \mathbf{b}}{\partial \mathbf{K}} \\ \hline \frac{\partial \mathbf{c}}{\partial \mathbf{K}} & \frac{\partial d}{\partial \mathbf{K}} \end{array}\right] = \left[\begin{array}{cccccc|c} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right]$$

Table 2. Transmission zero results

Zero	Actual	Derivative	Finite Difference
-0.9999999999999992	-1	0.5000000000000018	0.4999993752363707
-0.9999999999999999	-1	0.5000000000000002	0.5000000413701855

Residue Derivatives

- For the system shown previously

$$\mathbf{A} = \left[\begin{array}{c|c} \mathbf{a} & \mathbf{b} \\ \hline \mathbf{c} & d \end{array}\right] = \left[\begin{array}{cccccc|c} -2 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & -2 & 1 & 0 & 1 & -1 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right]$$

$$\frac{B(s)}{A(s)} = \frac{4s^2 + 8s + 4}{s^6 + 7s^5 + 18s^4 + 26s^3 + 24s^2 + 8s + 4}$$

$$\begin{aligned} \frac{B(s)}{A(s)} &= \frac{r_1}{s + 3.384} + \frac{r_2}{s + 2.200} + \frac{r_3}{s + 0.625 + 1.343i} \\ &+ \frac{r_3^\dagger}{s + 0.625 - 1.343i} + \frac{r_4}{s + 0.0833 - 0.488i} + \frac{r_4^\dagger}{s + 0.0833 + 0.488i} \end{aligned}$$

$r_1 = -0.183,\ r_2 = 0.241,\ r_3 = -0.130 - 0.197i$ and $r_4 = 0.102 - 0.299i$

$$\frac{\partial \mathbf{R}}{\partial \mathbf{K}} = \left[\begin{array}{ccc} 20 & 36 & 16 \end{array}\right]^T$$

$$\left\{\begin{array}{c} \partial r_1/\partial \mathbf{K} \\ \partial r_2/\partial \mathbf{K} \\ \partial r_3/\partial \mathbf{K} \\ \partial r_4/\partial \mathbf{K} \end{array}\right\} = \left\{\begin{array}{c} -0.5328 \\ 0.7136 \\ -0.4067 - 0.9127i \\ 0.3164 - 1.5115i \end{array}\right\}$$

Sylvester Based Algorithm for OOF:

- Sylvester's expansion

residue (num polynomial)

$$e^{\mathbf{A}t} = \sum_{k=1}^{\sigma} \sum_{l=0}^{m_k-1} t^l e^{\lambda_k t} \frac{1}{l!} (\mathbf{A} - \lambda_k \mathbf{I})^l \prod_{\substack{i=1 \\ i \neq k}}^{\sigma} (\mathbf{A} - \lambda_i \mathbf{I})^{m_i} \, n_k(\mathbf{A})$$

eigenvalue

- Is substituted for the system dynamics in the quadratic cost function

$$\tilde{J} = \int_0^{\infty} (\mathbf{x}^T \tilde{\mathbf{Q}} \mathbf{x} + \mathbf{u}^T \tilde{\mathbf{R}} \mathbf{u}) \; dt$$

- I.E.

$$\tilde{J} = \int_0^{\infty} \sum_{a=1}^{\sigma} \sum_{p=0}^{m_p-1} n_a(\mathbf{A}^H) \prod_{\substack{i=\sigma \\ i \neq a}}^{1} (\mathbf{A}^H - \lambda_i^H \mathbf{I})^{m_i} \frac{1}{p!} (\mathbf{A}^H - \lambda_a^H \mathbf{I})^p \, t^p e^{\lambda_a^H t} \mathbf{Q}$$

$$\bullet \sum_{b=1}^{\sigma} \sum_{q=o}^{m_k-1} t^q e^{\lambda_b t} \frac{1}{q!} (\mathbf{A} - \lambda_b \mathbf{I})^q \prod_{\substack{j=1 \\ j \neq b}}^{\sigma} (\mathbf{A} - \lambda_j \mathbf{I})^{m_j} \, n_b(\mathbf{A}) \, dt$$

- Which is re-written

$$\tilde{J} = \int_0^{\infty} \sum_{a=1}^{\sigma} \sum_{p=1}^{m_a} \mathbf{E}_{ap}^H \frac{1}{(p-1)!} \left(\mathbf{A}^H - \lambda_a^H \mathbf{I}\right)^{p-1} t^{p-1} e^{\lambda_a^H t} \mathbf{Q}$$

$$\bullet \sum_{b=1}^{\sigma} \sum_{q=1}^{n_b} t^{(q-1)} e^{\lambda_b t} \frac{1}{(q-1)!} (\mathbf{A} - \lambda_b \mathbf{I})^{(q-1)} \, \mathbf{E}_{bq} \; dt$$

- Which is then integrated closed-form

$$\tilde{J} = \sum_{a=1}^{\sigma}\sum_{b=1}^{\sigma}\sum_{p=1}^{m_a}\sum_{q=1}^{n_b} \frac{(-1)^{p+q-1}(p+q-2)!}{(\lambda_a^{\dagger}+\lambda_b)^{p+q-1}(p-1)!(q-1)!} \mathbf{E}_{ap}^{H}\left(\mathbf{A}^{H}-\lambda_a^{H}\mathbf{I}\right)^{(p-1)}\mathbf{Q}\left(\mathbf{A}-\lambda_b\mathbf{I}\right)^{(q-1)}\mathbf{E}_{bq}$$

- Where $\mathtt{E}(:,:,\mathtt{m}) = \mathtt{polyvalm}(\Phi, \mathbf{A}) * (\mathtt{n_k} * \mathbf{I})$

$$\mathbf{E}_k = \frac{\Phi(\mathbf{A})n_k(\mathbf{A})}{(\lambda-\lambda_k)^{m_k}}$$

m_k occurences of λ are deconvolved from char poly

- and the *nk* terms come from the PFE

$$\frac{1}{\Phi(\lambda)} = \frac{n_k}{(\lambda-\lambda_k)^{m_k}} + \frac{n_{k+1}}{(\lambda-\lambda_{k+1})^{m_{k+1}}} + \ldots$$

Characteristic polynomial of **A**

Gradient of the cost function

- quotient rule

$$\frac{\partial \tilde{J}}{\partial \mathbf{p}_n} = \frac{1}{\mathbf{G}^2}\left(\mathbf{G}\frac{\partial \mathbf{F}}{\partial \mathbf{p}_n} - \mathbf{F}\frac{\partial \mathbf{G}}{\partial \mathbf{p}_n}\right)$$

$$\mathbf{F} = \mathbf{E}_{ap}^{H}(\mathbf{A}^{H}-\lambda_a^{H}\mathbf{I})^{p-1}\mathbf{Q}(\mathbf{A}-\lambda_b\mathbf{I})^{q-1}\mathbf{E}_{bq}$$

$$\mathbf{G} = (\lambda_a^{\dagger}+\lambda_b)^{p+q-1}$$

$$\begin{aligned}\frac{\partial \mathbf{F}}{\partial \mathbf{p}_n} &= \frac{\partial \mathbf{E}_{ap}^{H}}{\partial \mathbf{p}_n}(\mathbf{A}^{H}-\lambda_a^{H}\mathbf{I})^{p-1}\mathbf{Q}(\mathbf{A}-\lambda_b\mathbf{I})^{q-1}\mathbf{E}_{bq} \\ &+ \mathbf{E}_{ap}^{H}\frac{\partial[(\mathbf{A}^{H}-\lambda_a^{H}\mathbf{I})^{p-1}]}{\partial \mathbf{p}_n}\mathbf{Q}(\mathbf{A}-\lambda_b\mathbf{I})^{q-1}\mathbf{E}_{bq} \\ &+ \ldots\end{aligned}$$

$$
\begin{aligned}
\frac{\partial \mathbf{E}_{ap}}{\partial \mathbf{p}_n} &= \left[\Phi_1 \frac{\partial \mathbf{A}}{\partial \mathbf{p}_n} + \Phi_2 \left(\mathbf{A}\frac{\partial \mathbf{A}}{\partial \mathbf{p}_n} + \frac{\partial \mathbf{A}}{\partial \mathbf{p}_n}\mathbf{A}\right) + \cdots \right.\\
&+ \left. \Phi_n \left(\frac{\partial \mathbf{A}}{\partial \mathbf{p}_n}\mathbf{A}^{n-1} + \mathbf{A}\frac{\partial \mathbf{A}}{\partial \mathbf{p}_n}\mathbf{A}^{n-2} + \cdots + \mathbf{A}^{n-1}\frac{\partial \mathbf{A}}{\partial \mathbf{p}_n}\right) + \sum_{m=0}^{\sigma} \frac{\partial \Phi}{\partial \mathbf{p}_n}\mathbf{A}^m\right]
\end{aligned}
$$

- $n_k(\mathbf{A})/(\lambda - \lambda_k)^{m_k} + \frac{\Phi(\mathbf{A})}{(\lambda - \lambda_k)^{m_k}} \frac{\partial n_k(\mathbf{A})}{\partial \mathbf{p}_n}$

- Derivative of matrix polynomial leads to first / last algorithm:

```
% first term:
F = Z;
L = dA;
Z = Z + (F + L)*Psi(p-1);

% subsequent terms:
for q = p-2:-1:1
    F = A*(F + L);
    L = L*A;
    Z = Z + (F + L)*Psi(q);

end
```

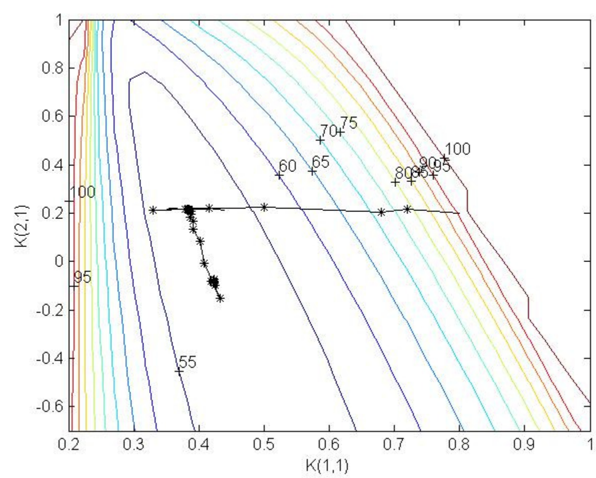
- Map eigenvalue derivatives to derivatives of char poly coeffs:

$$
\frac{\partial \Phi_{m-k}}{\partial \mathbf{p}_n} = \sum_{m=1}^{\sigma} \frac{\partial \lambda_m}{\partial \mathbf{p}_n} \prod_{\substack{s=\binom{\sigma}{k-1} \\ s \neq m}} \lambda_s .
$$

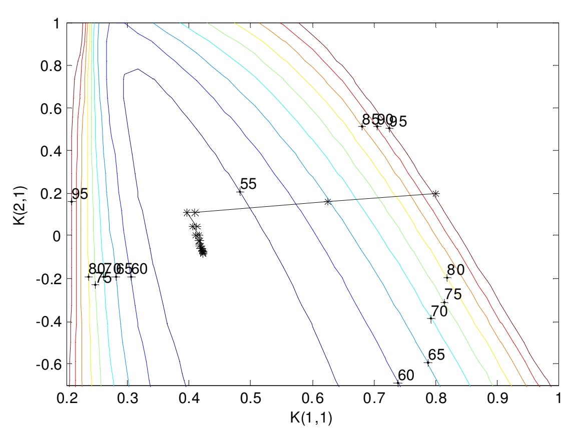
$$
\frac{\partial[(\mathbf{A}^H - \lambda_a^H \mathbf{I})^{p-1}]}{\partial \mathbf{p}_n} = (p-1)(\mathbf{A}^H - \lambda_a^H \mathbf{I})^{p-2} \left[\frac{\partial \mathbf{A}^H}{\partial \mathbf{p}_n} - \frac{\partial \lambda_a^H}{\partial \mathbf{p}_n}\mathbf{I}\right]
$$

- Is just one term in the polyderm expansion.

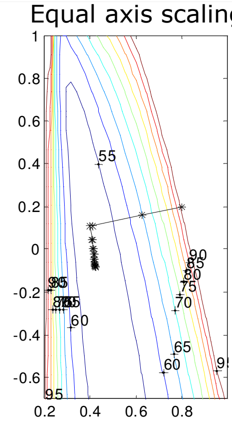
Result: Non-gradient search



Results: Gradient search



Equal axis scaling:



Open Issues

- Is the differentiated QZ stable?
- Operations counts
- Actual repeated eigenvalue cases that are not defective / overly contrived?