Solution of Algebraic Riccati Equations Using the Sum of Roots

○ M. Kanno[1] K. Yokoyama[2] H. Anai[3] S. Hara[4]

[1]Niigata University
[2]Rikkyo University
[3]Fujitsu Laboratories Ltd
[4]The University of Tokyo

International Symposium on Symbolic and Algebraic Computation 2009

Korea Institute for Advanced Study (KIAS), Seoul, Korea

30 JULY 2009

What We Want to Do....

Gröbner Basis — *Algebraic method*

Computational Algebraic Geometry

Sum of Roots

Polynomial Spectral Factorization

Important Mathematical Tool for
Signal Processing and Control

Extend the result to solve *Algebraic Riccati Equations!*

Outline of the Talk

1. Polynomial Spectral Factorization via the Sum of Roots
2. Algebraic Riccati Equations
3. Sum of Roots Approach to Algebraic Riccati Equations
4. Concluding Remarks

Outline of the Talk

1. Polynomial Spectral Factorization via the Sum of Roots
2. Algebraic Riccati Equations
3. Sum of Roots Approach to Algebraic Riccati Equations
4. Concluding Remarks

Polynomial Spectral Factorization

Problem Formulation

- **Given** : an even polynomial in λ (polynomial in λ^2)

$$f(\lambda) = \lambda^{2n} + a_{2n-2}\lambda^{2n-2} + a_{2n-4}\lambda^{2n-4} + \cdots + a_0$$

- **Assumption** : $f(\lambda)$ has no roots on the imaginary axis
- Roots of $f(\lambda)$: symmetrical w.r.t. the real & imaginary axes and the origin
- **Task** : Find a polynomial $g(\lambda)$ such that $\boxed{f(\lambda) = (-1)^n g(\lambda) g(-\lambda)}$ and $g(\lambda)$ has roots in the open *left half plane* only.
- $g(\lambda)$: monic without loss of generality

$$g(\lambda) = \lambda^n + \underbrace{\boldsymbol{\sigma}}_{-\sum_{i=1}^{n}\alpha_i}\lambda^{n-1} + b_{n-2}\lambda^{n-2} + \cdots + b_0$$

- $\boldsymbol{\sigma}$: Sum of Roots — α_i : Roots of $f(\lambda)$ in the open *left half plane*

Solution Approach Using the Sum of Roots

$$g(\lambda) = \lambda^n + \boldsymbol{\sigma}\lambda^{n-1} + b_{n-2}\lambda^{n-2} + \cdots + b_0$$

- $\boxed{f(\lambda) = (-1)^n g(\lambda) g(-\lambda)}$

  $\Longrightarrow$ a *system of algebraic equations* in $\boldsymbol{\sigma}$, b_j
- *Reduced Gröbner basis*
- *Efficient basis-conversion technique* $\Longrightarrow$ Shape basis obtained

Shape Basis

$$\begin{cases} S_f(\boldsymbol{\sigma}) = 0 & \text{— characteristic polynomial of } \sigma,\ 2^n\text{-th order} \\ b_{n-2} = h_{n-2}(\boldsymbol{\sigma}) & \text{— true } \boldsymbol{\sigma} = \text{largest real root of } S_f \\ b_{n-3} = h_{n-3}(\boldsymbol{\sigma}) & \\ \vdots & \\ b_0 = h_0(\boldsymbol{\sigma}) & \end{cases}$$

h_i: polynomial of order strictly less than 2^n

Outline of the Talk

Algebraic Riccati Equations

Problem Formulation

A quadratic matrix equation for an unknown $n \times n$ matrix X of the form

$$A^{\mathrm{T}}X + XA + XRX + Q = 0$$

- A, Q, R : given $n \times n$ matrices
- Q, R : symmetric

- Positive semi-definite X ($X = X^{\mathrm{T}} \geq 0$) to be found

Assumptions

- Q positive semi-definite, R negative semi-definite
- (A, R) stabilizable, (Q, A) detectable

$\Longrightarrow$ *Guarantees no roots on the imaginary axis and the existence of positive semi-definite X*

Target

Naïve Approach

- Elements of $X = (x_{i,j})$: variables
- $A^{\mathrm{T}}X + XA + XRX + Q = 0$
  $\Longrightarrow$ A set of algebraic equations in $x_{i,j}$
- Employ general Gröbner basis computation algorithms

Drawbacks

- Cannot make use of structural properties
- # of variables : $\frac{1}{2}n(n+1)$
- Difficult to characterize the positive semi-definite solution (?)

Aim

- Characterize the solution X by means of the SoR

Standard (Numerical) Solution Approach

$$A^{\mathrm{T}}X + XA + XRX + Q = 0$$

1. Construct the *Hamiltonian matrix*

$$H := \begin{bmatrix} A & R \\ -Q & -A^{\mathrm{T}} \end{bmatrix} \qquad (2n \times 2n \text{ matrix})$$

2. Compute n eigenvectors v_i
   associates with the eigenvalues of H in the left half plane.
3. $\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \qquad X_1, X_2 : n \times n$ matrices
4. $X = X_2 X_1^{-1} (\geq 0)$

Outline of the Talk

1. Polynomial Spectral Factorization via the Sum of Roots
2. Algebraic Riccati Equations
3. Sum of Roots Approach to Algebraic Riccati Equations
4. Concluding Remarks

Proposed Approach (1)

Input

- $A, Q, R \in \mathbb{Q}[\mathbf{q}]^{n\times n}$
- $\mathbf{q} = (q_1, \ldots, q_m)$: vector of parameters

Output

- X : rational functions of parameters $\mathbf{q}$ and the SoR σ
- $S_f(\sigma)$: Characteristic polynomial relating parameters $\mathbf{q}$ and the SoR σ

! Closed-form expression for X in terms of *parameters* — *not sought / impossible*

Proposed Approach (2)

Consists of 4 Steps....

1. Compute the characteristic polynomial $f(\lambda)$ of H. Carry out polynomial spectral factorization for $f(\lambda) = (-1)^n g(\lambda) g(-\lambda)$.
   — *Relate parameters and the SoR*
2. Express the eigenvector $v(\lambda')$ as polynomials in the eigenvalue λ'
3. Express $\begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$ as a product of two matrices: $\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{bmatrix} V$
   $\tilde{X}_1, \tilde{X}_2$: matrices in the coefficients of $g(\lambda)$ — *SoR*
   V : matrices in eigenvalues λ_i
   — *Separate σ, b_j and λ_i*
4. $X = \tilde{X}_2 \tilde{X}_1^{-1}$

Step by Step (1)

$$A = \begin{bmatrix} -2 & -3 \\ 1 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} 10 & 14 \\ 14 & 20 \end{bmatrix}, \quad R = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

1. $$H = \begin{bmatrix} A & R \\ -Q & -A^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} -2 & -3 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ -10 & -14 & 2 & -1 \\ -14 & -20 & 3 & 0 \end{bmatrix}$$

   Characteristic polynomial $f(\lambda) = \det(\lambda I_4 - H) = \lambda^4 - 28\lambda^2 + 35$

   Polynomial spectral factorization for $f(\lambda)$:

   $$g(\lambda) = \lambda^2 + \sigma\lambda + b_0$$

   $$\boxed{f(\lambda) = (-1)^n g(\lambda) g(-\lambda)} \quad \Longrightarrow \quad \begin{cases} \sigma^4 - 56\sigma^2 + 644 = 0 \\ b_0 = \frac{1}{2}\sigma^2 - 14 \end{cases}$$

Step by Step (2)

2. Get an expression for the eigenvector.
   Compute the quotient: $f(\lambda) = (\lambda - \lambda')q(\lambda) + r$

$$f(H) = (H - \lambda' I_{2n})q(H) = 0 \implies H \underbrace{q(H)u}_{v(\lambda')} = \lambda' \underbrace{q(H)u}_{v(\lambda')}$$

for suitable chosen u

$$q(\lambda) = \lambda^3 + \lambda'\lambda^2 + (\lambda'^2 - 28)\lambda + \lambda'^3 - 28\lambda'$$

$$u = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \implies v(\lambda') = q(H)u = \begin{bmatrix} 4\lambda' - 20 \\ -\lambda'^2 + 15 \\ -\lambda'^2 + 12\lambda' - 5 \\ \lambda'^3 - 11\lambda' - 20 \end{bmatrix}$$

! *The quotient to be computed only once!*

Step by Step (2)

2. Get an expression for the eigenvector.
   Compute the quotient: $f(\lambda) = (\lambda - \lambda')q(\lambda) + r$

$$f(H) = (H - \lambda' I_{2n})q(H) = 0 \implies H \underbrace{q(H)u}_{v(\lambda')} = \lambda' \underbrace{q(H)u}_{v(\lambda')}$$

for suitable chosen u

$$q(\lambda) = \lambda^3 + \lambda'\lambda^2 + (\lambda'^2 - 28)\lambda + \lambda'^3 - 28\lambda'$$

$$u = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \implies v(\lambda') = q(H)u = \begin{bmatrix} 4\lambda' - 20 \\ -\lambda'^2 + 15 \\ -\lambda'^2 + 12\lambda' - 5 \\ \lambda'^3 - 11\lambda' - 20 \end{bmatrix}$$

! *The quotient to be computed only once!*

Step by Step (3)

3. $$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} v(\lambda_1) & v(\lambda_2) & \cdots & v(\lambda_n) \end{bmatrix}$$

— λ_i : eigenvalues in the left half plane

In fact, it suffices to consider $\boxed{v(\lambda_i) \mod g(\lambda_i)}$

$$v(\lambda_i) \mod g(\lambda_i) = \left[\begin{array}{c} 4\lambda_i - 20 \\ \sigma\lambda_i + b_0 + 15 \\ \hline (\sigma + 12)\lambda_i + b_0 - 5 \\ (\sigma^2 - b_0 - 11)\lambda_i + \sigma b_0 - 20 \end{array}\right]$$

$$= \left[\begin{array}{cc} -20 & 4 \\ b_0 + 15 & \sigma \\ \hline b_0 - 5 & \sigma + 12 \\ \sigma b_0 - 20 & \sigma^2 - b_0 - 11 \end{array}\right] \begin{bmatrix} 1 \\ \lambda_i \end{bmatrix} = \left[\begin{array}{c} \tilde{X}_1 \\ \hline \tilde{X}_2 \end{array}\right] \begin{bmatrix} 1 \\ \lambda_i \end{bmatrix}$$

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ \lambda_1 & \lambda_2 \end{bmatrix} = \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{bmatrix} V$$ — *Separated*

Step by Step (4)

4. $X = X_2 X_1^{-1} = \tilde{X}_2 V V^{-1} \tilde{X}_1^{-1} = \tilde{X}_2 \tilde{X}_1^{-1}$ — rational functions in σ, b_i

$$X = -\frac{1}{20\sigma + 4b_0 + 60} \times \begin{bmatrix} 20\sigma + 12b_0 + 180 & 20\sigma + 4b_0 + 220 \\ 15\sigma^2 - b_0{}^2 + 20\sigma - 26b_0 - 165 & 20\sigma^2 + 4\sigma b_0 - 20b_0 - 300 \end{bmatrix}$$

$$= -\frac{1}{\sigma^2 + 10\sigma + 2} \begin{bmatrix} 3\sigma^2 + 10\sigma + 6 & \sigma^2 + 10\sigma + 82 \\ \sigma^2 + 10\sigma + 82 & \sigma^3 + 5\sigma^2 - 28\sigma - 10 \end{bmatrix}$$

What Is the Advantage?

What We Get....

- Algebraic relationship between *parameters* and the *SoR* σ:

$$\sigma^8 - 4(q_1^2 + 2)\sigma^6 + 2(3q_1^4 + 4q_1^2 + 8)\sigma^4 \\ - 4(q_1^6 - 2q_1^4 + 256q_1^2q_2^2 + 8q_1^2)\sigma^2 + q_1^4(q_1 - 2)^2(q_1 + 2)^2 = 0$$

- Expression (rational function) for the *Optimal Cost* in terms of *parameters* and the *SoR*

  E.g., $\mathcal{H}_2$ Optimal Cost — $\mathrm{tr}\left\{B^{\mathrm{T}}XB\right\}$

What We Can Do....

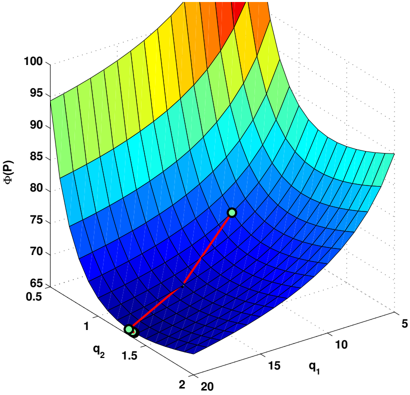
Since we get an *exact* expression, we can....

- Take the derivative of σ with respect to parameters: $\dfrac{\partial \sigma}{\partial q_i}$
- Take the derivative of the *Optimal Cost* with respect to parameters

- Optimization of both the *Plant* and the *Controller*:

$$\inf_{P\in\mathcal{P},\, K\in\mathcal{K}} J(P,\, K) \quad \Longrightarrow \quad \inf_{P\in\mathcal{P}} \inf_{K\in\mathcal{K}} J(P,\, K)$$

- The '$\inf_{K\in\mathcal{K}}$' part — *SoR Approach*
- Newton's method applicable for the '$\inf_{P\in\mathcal{P}}$' part



Outline of the Talk

Concluding Remarks

In This Talk....

- Solution approach to AREs by means of the SoR proposed
  - Polynomial spectral factorization by the SoR

In the Future....

- Solid and efficient implementation
  - Further structural properties exploited
- $\mathcal{H}_\infty$ control
- Discrete-time version

Thank you very much for your attention!