Verified Computation of Hermitian (Symmetric) Solutions to Continuous-Time Algebraic Riccati Matrix Equations

Behnam Hashemi

Department of Mathematics
Shiraz University of Technology

SCAN 2012, Novosibirsk
Outline

1 Motivation
- The Riccati Equation and Some Basic Tools
- Our Main Problem
- Previous Works

2 Our Results/Contribution
- Main Results
- Algorithms
- Numerical Results
Outline

1. Motivation
   - The Riccati Equation and Some Basic Tools
     - Our Main Problem
     - Previous Works

2. Our Results/Contribution
   - Main Results
   - Algorithms
   - Numerical Results
The matrix equation

\[ R(X) := A^* X + XA - XSX + Q = 0, \]  

is called the continuous-time algebraic Riccati equation (CARE), where

\[ A \in \mathbb{C}^{n \times n}, \]
\[ S = S^* \in \mathbb{C}^{n \times n}, \]
\[ Q = Q^* \in \mathbb{C}^{n \times n}, \]

are given and \( X \in \mathbb{C}^{n \times n} \) is the unknown solution.
The matrix $A - SX$ is called the closed loop matrix associated with the CARE (1).
Several applications require a Hermitian positive semidefinite stabilizing solution of the CARE (1).

A Hermitian solution $X$ of (1) is a **stabilizing solution** if the closed loop matrix $A - SX$ is stable, i.e., the spectrum of $A - SX$ lies in the closed left half-plane.
**Important Formula**

**vec-of-three-factors:** $\text{vec}(ABC) = (C^T \otimes A)\text{vec}(B)$.

$\otimes$: Kronecker product of matrices

$\text{vec}$: stacks columns of a matrix into a long vector
notation for simplicity: "lowercase := vec(uppercase)"

\[ b := \text{vec}(B). \]
so we write: \[ \text{vec}(ABC) = (C^T \otimes A) \, b \]
The Fréchet derivative of $R$ at $X$ in the direction $H$ is

$$R'(X) \cdot H = H(A - SX) + (A - SX)^* H,$$

which means that

$$r'(x) = I \otimes (A - SX)^* + (A - SX)^T \otimes I \in \mathbb{C}^{n^2 \times n^2}.$$
Outline

1. Motivation
   - The Riccati Equation and Some Basic Tools
   - Our Main Problem
   - Previous Works

2. Our Results/Contribution
   - Main Results
   - Algorithms
   - Numerical Results
Develop an efficient technique based on interval arithmetic which provides \textit{guaranteed error bounds} for solutions of the continuous-time algebraic Riccati equation (1)
Outline

1. **Motivation**
   - The Riccati Equation and Some Basic Tools
   - Our Main Problem
   - Previous Works

2. **Our Results/Contribution**
   - Main Results
   - Algorithms
   - Numerical Results
The Fréchet derivative of $R$ at $X$ is used to derive an interval Sylvester matrix equation of the form

$$CX + XD = F,$$

Transform the interval Sylvester equation into the large interval linear system

$$(I \otimes C + D^T \otimes I)x = f$$

with $x := \text{vec}(X)$ and $f := \text{vec}(F)$ and solve it.
The Fréchet derivative of $R$ at $X$ is used to derive an interval Sylvester matrix equation of the form $CX + XD = F$,.

Transform the interval Sylvester equation into the large interval linear system $(I \otimes C + D^T \otimes I)x = f$ with $x := \text{vec}(X)$ and $f := \text{vec}(F)$ and solve it.
The Fréchet derivative of $R$ at $X$ is used to derive an interval Sylvester matrix equation of the form $\mathbf{C}X + XD = F$,

Transform the interval Sylvester equation into the large interval linear system $(I \otimes \mathbf{C} + \mathbf{D}^T \otimes I)x = f$ with $x := \text{vec}(X)$ and $f := \text{vec}(F)$ and solve it.
The number of arithmetic operations needed to implement this interval Newton technique is roughly $O(n^6)$! because the coefficient matrix of the resulting interval linear system is $n^2 \times n^2$!
Main Issue: Computational Complexity

The number of arithmetic operations needed to implement this interval Newton technique is roughly $O(n^6)$! because the coefficient matrix of the resulting interval linear system is $n^2 \times n^2$!
Classical Krawczyk approach

\[ k(\ddot{x}, x) := \dot{x} - R \cdot r(\ddot{x}) + (I_{n^2} - R \cdot r'(x)) (x - \ddot{x}), \]

where

\[ r : \mathbb{C}^{n^2} \rightarrow \mathbb{C}^{n^2}, \quad x \mapsto r(\ddot{x}) := \text{vec}(R(\dot{X})), \]

\[ r'(x) = \left( I \otimes (A - SX)^* + (A - SX)^T \otimes I \right) \in \mathbb{C}^{n^2 \times n^2}. \]
The Riccati Equation and Some Basic Tools

Our Main Problem

Previous Works

Classical Krawczyk approach

\[ k(\dot{x}, x) := \dot{x} - R \cdot r(\dot{x}) + (I_{n^2} - R \cdot r'(x)) (x - \dot{x}), \]

where

\[ r : \mathbb{C}^{n^2} \to \mathbb{C}^{n^2}, \quad x \mapsto r(\dot{x}) := \text{vec}(R(\dot{X})), \]

\[ r'(x) = \left( I \otimes (A - SX)^* + (A - SX)^T \otimes I \right) \in \mathbb{C}^{n^2 \times n^2}. \]
Main Issue Again: Computational Complexity

- Standard choice is to take $R \in \mathbb{C}^{n^2 \times n^2}$ as an approximate inverse of mid $r'(x)$.
- $R$ is needed explicitly. $I - R r'(x)$ is also needed explicitly.
- Cost is $\mathcal{O}(n^5)$!
- The number of arithmetic operations needed to implement the classical Krawczyk approach is at-least $\mathcal{O}(n^5)$!
Main Issue Again: Computational Complexity

- Standard choice is to take $R \in \mathbb{C}^{n^2 \times n^2}$ as an approximate inverse of mid $r'(x)$.
- $R$ is needed explicitly. $I - R r'(x)$ is also needed explicitly.
- Cost is $\mathcal{O}(n^5)$!
- The number of arithmetic operations needed to implement the classical Krawczyk approach is at-least $\mathcal{O}(n^5)$!
Main Issue Again: Computational Complexity

- Standard choice is to take \( R \in \mathbb{C}^{n^2 \times n^2} \) as an approximate inverse of \( \text{mid} \ r'(x) \).
- \( R \) is needed explicitly. \( I - R \ r'(x) \) is also needed explicitly.
- Cost is \( \mathcal{O}(n^5) \)!

The number of arithmetic operations needed to implement the classical Krawczyk approach is at-least \( \mathcal{O}(n^5) \)!
Main Issue Again: Computational Complexity

- Standard choice is to take $R \in \mathbb{C}^{n^2 \times n^2}$ as an approximate inverse of mid $r'(x)$.
- $R$ is needed explicitly. $I - R \cdot r'(x)$ is also needed explicitly.
- Cost is $\mathcal{O}(n^5)$!
- The number of arithmetic operations needed to implement the classical Krawczyk approach is at-least $\mathcal{O}(n^5)$!
Challenge

Reduce the cost to *cubic*!

The big question:

How to compute $R$ and $I_{n^2} - R \cdot r'(x)$ more cheaply?
Challenge

Reduce the cost to \textbf{cubic} !

The big question:

How to compute $R$ and $I_{n^2} - R \cdot r'(x)$ more cheaply?
Outline

1 Motivation
   - The Riccati Equation and Some Basic Tools
   - Our Main Problem
   - Previous Works

2 Our Results/Contribution
   - Main Results
   - Algorithms
   - Numerical Results
Essence of Krawczyk-Type Iterations

Theorem (Rump 1983, AND Frommer, H. 2009)

Assume that $f : D \subset \mathbb{C}^N \rightarrow \mathbb{C}^N$ is continuous in $D$. Let $\tilde{x} \in D$ and $z \in \mathbb{IC}^N$ be such that $\tilde{x} + z \subseteq D$. Moreover, assume that $\mathcal{P} \subset \mathbb{C}^{N\times N}$ is a set of matrices containing all slopes $P(\tilde{x}, y)$ for $y \in \tilde{x} + z =: x$. Finally, let $R \in \mathbb{C}^{N\times N}$. Denote $\mathcal{K}_f(\tilde{x}, R, z, \mathcal{P})$ the set

$$\mathcal{K}_f(\tilde{x}, R, z, \mathcal{P}) := \{-Rf(\tilde{x}) + (I - RP)z : P \in \mathcal{P}, z \in z\}. \quad (2)$$

Then, if $\mathcal{K}_f(\tilde{x}, R, z, \mathcal{P}) \subseteq \text{int } z$, the function $f$ has a zero $x^*$ in the set $\tilde{x} + \mathcal{K}_f(\tilde{x}, R, z, \mathcal{P}) \subseteq x$. Moreover, if $\mathcal{P}$ also contains all slope matrices $P(y, x)$ for the function $f$ and for $x, y \in x$, then this zero is unique in $x$. 
Theorem

Assume that $X$ is an Hermitian interval matrix and $X, Y \in X$. Then, the interval arithmetic evaluation of the Fréchet derivative of $R$ contains all its slopes.
Proof.

Suppose that $X, Y \in X$.

\[
R(Y) - R(X) = A^* Y + YA - YSY - A^* X - XA + XSX
= A^* (Y - X) + (Y - X)A
- \frac{1}{2} ((Y + X)S(Y - X) + (Y - X)S(Y + X)),
\]

So,

\[
r(y) - r(x) = \left[ I \otimes \left( A^* - \frac{1}{2} (Y + X)S \right) + (A^T - \frac{1}{2} (S(Y + X))^T \right) \otimes I \right] (y - x).
\]
Proof.

This means that

\[ P(y, x) = I \otimes (A^* - \frac{1}{2}(Y + X)S) + (A^T - \frac{1}{2}(S(Y + X))^T) \otimes I. \]

Since \( X, Y \in X \), by the enclosure property of interval arithmetic we have

\[ P(y, x) \in I \otimes (A^* - XS) + (A - SX)^T \otimes I. \]

Since \( X \) is Hermitian, \( X^*, Y^* \in X \). Moreover, \( S^* = S \). So, \( A^* - XS = (A - SX)^* \) and therefore

\[ P(y, x) \in I \otimes (A - SX)^* + (A - SX)^T \otimes I. \]

interval arithmetic evaluation of \( R'(X) \).
Slopes and Fréchet derivative of the function $R(X)$

**Proof.**

This means that

$$P(y, x) = I \otimes \left( A^* - \frac{1}{2} (Y + X)S \right) + \left( A^T - \frac{1}{2} (S(Y + X))^T \right) \otimes I.$$

Since $X, Y \in X$, by the enclosure property of interval arithmetic we have

$$P(y, x) \in I \otimes \left( A^* - XS \right) + \left( A - SX \right)^T \otimes I.$$

Since $X$ is Hermitian, $X^*, Y^* \in X$. Moreover, $S^* = S$. So, $A^* - XS = (A - SX)^*$ and therefore

$$P(y, x) \in I \otimes \left( A - SX \right)^* + \left( A - SX \right)^T \otimes I.$$

interval arithmetic evaluation of $R'(X)$
Proof.

This means that

\[ P(y, x) = I \otimes (A^* - \frac{1}{2}(Y + X)S) + (A^T - \frac{1}{2}(S(Y + X))^T) \otimes I. \]

Since \( X, Y \in X \), by the enclosure property of interval arithmetic we have

\[ P(y, x) \in I \otimes (A^* - XS) + (A - SX)^T \otimes I. \]

Since \( X \) is Hermitian, \( X^*, Y^* \in X \). Moreover, \( S^* = S \). So, \( A^* - XS = (A - SX)^* \) and therefore

\[ P(y, x) \in I \otimes (A - SX)^* + (A - SX)^T \otimes I. \]

interval arithmetic evaluation of \( R'(X) \)
Step 1: An as thin as possible enclosure for

\[ \mathcal{K}_f(\dot{\mathbf{x}}, R, \mathbf{z}, \mathcal{P}) := \{-R f(\dot{\mathbf{x}}) + (I - RP)\mathbf{z} : P \in \mathcal{P}, \mathbf{z} \in \mathbf{z}\}. \]
So, What Do We Need?

Step 1: An as thin as possible enclosure for

\[
\mathcal{K}_f(\dot{x}, R, z, P) := \left\{ -RF(\dot{x}) + (I - RP)z : P \in P, z \in z \right\}.
\]

FIRST TERM SECON Termin

Step 2: Check the relation \( \mathcal{K}_f(\dot{x}, R, z, P) \subseteq \text{int } z \).
So, What Do We Need ?

Step 1: An as thin as possible enclosure for

\[
\mathcal{K}_f(\dot{x}, R, z, \mathcal{P}) := \{ -R f(\dot{x}) + (I - RP)z : P \in \mathcal{P}, z \in z \}.
\]

FIRST TERM  SECOND TERM

Step 2: Check the relation \( \mathcal{K}_f(\dot{x}, R, z, \mathcal{P}) \subseteq \text{int} \ z \).
The Key: Spectral Decomposition of the Closed Loop Matrix

Let

\[ A - SX = V\Lambda W \quad \text{with} \]

\[ V, \Lambda, W \in \mathbb{C}^{n \times n}, \]
\[ VW = I, \]
\[ \Lambda = \text{Diag}(\lambda_1, \lambda_2, \cdots, \lambda_n) \quad \text{diagonal}. \]
Consequence of the Spectral Decomposition

Recall: \( r'(x) = I \otimes (A - SX)^* + (A - SX)^T \otimes I. \)

Another basic formula: \((A \otimes B) \cdot (C \otimes D) = (A \cdot C \otimes B \cdot D).\)
Consequence of the Spectral Decomposition

Recall: \( r'(x) = I \otimes (A - SX)^* + (A - SX)^T \otimes I \).

\[
\begin{align*}
    r'(x) &= (V^{-T} \otimes W^*). \\
          &\quad \left( I \otimes [W(A - SX)W^{-1}]^* + [V^{-1}(A - SX)V]^T \otimes I \right). \\
          &\quad (V^T \otimes W^{-*}).
\end{align*}
\]

Another basic formula: \( (A \otimes B) \cdot (C \otimes D) = (A \cdot C \otimes B \cdot D) \).
Consequence of the Spectral Decomposition

Recall: \( r'(x) = I \otimes (A - SX)^* + (A - SX)^T \otimes I. \)

\[
\begin{align*}
r'(x) &= (V^{-T} \otimes W^*) \cdot \\
&\quad \left( I \otimes [W(A - SX)W^{-1}]^* + [V^{-1}(A - SX)V]^T \otimes I \right) \cdot \\
&\quad (V^T \otimes W^{-*}).
\end{align*}
\]

Another basic formula: \((A \otimes B) \cdot (C \otimes D) = (A \cdot C \otimes B \cdot D).\)
Consequence of the Spectral Decomposition

\[ r'(x) = (V^{-T} \otimes W^*) \cdot \left( I \otimes \left[ W(A - SX)W^{-1} \right]^* + \left[ V^{-1}(A - SX)V \right]^T \otimes I \right) \cdot (V^T \otimes W^{-*}), \]
Consequence of the Spectral Decomposition: An Approximate Inverse for mid $r'(x)$

\[ R = (V^{-T} \otimes W^*) \cdot \left( I \otimes \Lambda^* + \Lambda^T \otimes I \right)^{-1} \cdot (V^T \otimes W^{-*}), \]
Consequence of the Spectral Decomposition: An Approximate Inverse for mid $r'(x)$

$$R = (V^{-T} \otimes W^*) \cdot \left( I \otimes \Lambda^* + \Lambda^T \otimes I \right)^{-1} \cdot (V^T \otimes W^{-*}),$$

*Extremely important:*

$$\Delta := I \otimes \Lambda^* + \Lambda^T \otimes I \in \mathbb{C}^{n^2 \times n^2} \text{ is diagonal.}$$
Consequence of the Spectral Decomposition for the SECOND TERM in $\mathcal{K}_r(\dot{x}, R, z, P)$

Recall: $R = (V^{-T} \otimes W^*) \cdot \Delta^{-1} \cdot (V^T \otimes W^{-*})$.

We have

$$l_{n^2} - R \cdot r'(x) =$$

$$l_{n^2} - R(l_n \otimes (A - SX)^* + (A - SX)^T \otimes l_n) =$$

$$(V^{-T} \otimes W^*) \Delta^{-1} \Omega (V^T \otimes W^{-*}),$$

where

$$\Omega = \Delta - l_n \otimes (W(A - SX)W^{-1})^* - (V^{-1}(A - SX)V)^T \otimes l_n.$$
Consequence of the Spectral Decomposition for the SECOND TERM in $\mathcal{K}_r(\dot{x}, R, z, \mathcal{P})$

Recall: $R = (V^{-T} \otimes W^*) \cdot \Delta^{-1} \cdot (V^T \otimes W^{-*})$.

We have

$$I_{n^2} - R \cdot r'(x) =$$

$$I_{n^2} - R(I_n \otimes (A - SX)^* + (A - SX)^T \otimes I_n) =$$

$$(V^{-T} \otimes W^*) \Delta^{-1} \Omega (V^T \otimes W^{-*}),$$

where

$$\Omega = \Delta - I_n \otimes (W(A - SX)W^{-1})^* - (V^{-1}(A - SX)V)^T \otimes I_n.$$
Outline

1 Motivation
   - The Riccati Equation and Some Basic Tools
   - Our Main Problem
   - Previous Works

2 Our Results/Contribution
   - Main Results
   - Algorithms
   - Numerical Results
Alg. 1: Compute an Interval Matrix $Z$ s.t. $z$ Encloses the FIRST TERM $-R \cdot r(\dot{x})$ with $R = (V^{-T} \otimes W^*) \cdot \Delta^{-1} \cdot (V^T \otimes W^{-*})$

1: Enclose $RES := A^*\dot{X} + \dot{X}A - \dot{X}S\dot{X} + Q$.
2: Enclose $G := I^*_W \cdot RES \cdot V$.
3: Enclose $H := G \cdot / D$.
4: Enclose $Z := -W^*HI_V$.
5: Output $Z$.

Cost of Alg. 1 is cubic.
Alg. 1: Compute an Interval Matrix $Z$ s.t. $z$ Encloses the FIRST TERM $-R \cdot r(\dot{x})$ with

$$R = (V^{-T} \otimes W^*) \cdot \Delta^{-1} \cdot (V^T \otimes W^{-*})$$

1: Enclose $RES := A^* \dot{X} + \dot{X}A - \dot{X}S\dot{X} + Q$.
2: Enclose $G := I_W^* \cdot RES \cdot V$.
3: Enclose $H := G \cdot D$.
4: Enclose $Z := -W^* HI_V$.
5: Output $Z$.

Cost of Alg. 1 is cubic.
Alg. 2: Compute an Interval Matrix $U$ s.t. $u$ Encloses the Set of SECOND TERMS with $x$ Replaced by $\dot{x} + y$

Recall: $(I_{n^2} - R \cdot r'(\dot{x} + y))y = (V^{-T} \otimes W^*) \Delta^{-1} \Omega (V^T \otimes W^{-*})y$, where

\[
\Omega = I_n \otimes \Lambda^* - I_n \otimes \left( W(A - S(\dot{X} + Y))W^{-1} \right)^* + \\
\Lambda^T \otimes I_n - \left( V^{-1}(A - S(\dot{X} + Y))V \right)^T \otimes I_n.
\]

1: Enclose $ZZ = I_w^* \cdot Y \cdot V$,
2: Enclose $P = W \cdot (A - S(\dot{X} + Y)) \cdot I_w$.
3: Enclose $Q = I_v \cdot (A - S(\dot{X} + Y)) \cdot V$.
4: Enclose $E = (\Lambda - P)^* \cdot ZZ + ZZ \cdot (\Lambda - Q)$.
5: Enclose $N = E \div D$.
6: Enclose $U = W^* \cdot N \cdot I_v$
7: Output $U$.

Cost of Alg. 2 is also cubic.
Alg. 2: Compute an Interval Matrix $U$ s.t. $u$ Encloses the Set of **SECOND TERMS** with $x$ Replaced by $\dot{x} + y$

Recall: $(I_{n^2} - R \cdot r'(\dot{x} + y))y = (V^{-T} \otimes W^*) \Delta^{-1} \Omega (V^T \otimes W^{-*})y$,
where

$$\Omega = I_n \otimes \Lambda^* - I_n \otimes (W(A - S(\dot{X} + Y))W^{-1})^* +$$

$$\Lambda^T \otimes I_n - \left(V^{-1}(A - S(\dot{X} + Y))V\right)^T \otimes I_n.$$

1: Enclose $ZZ = I_W^* \cdot Y \cdot V$,
2: Enclose $P = W \cdot (A - S \cdot (\dot{X} + Y)) \cdot I_W$.
3: Enclose $Q = I_V \cdot (A - S \cdot (\dot{X} + Y)) \cdot V$.
4: Enclose $E = (\Lambda - P)^* \cdot ZZ + ZZ \cdot (\Lambda - Q)$.
6: Enclose $U = W^* \cdot N \cdot I_V$
7: Output $U$.

Cost of Alg. 2 is also cubic.
Alg. 2: Compute an Interval Matrix $U$ s.t.

**The Set of SECOND TERMS** with $x$ Replaced by $\dot{x} + y$

Recall: $(I_{n^2} - R \cdot r'(\dot{x} + y))y = (V^{-T} \otimes W^*) \Delta^{-1} \Omega (V^T \otimes W^{-*})y,$

where

$$\Omega = I_n \otimes \Lambda^* - I_n \otimes \left( W(A - S(\dot{X} + Y))W^{-1} \right)^* + \Lambda^T \otimes I_n - \left( V^{-1}(A - S(\dot{X} + Y))V \right)^T \otimes I_n.$$ 

1: Enclose $ZZ = I_W^* \cdot Y \cdot V,$
2: Enclose $P = W \cdot (A - S \cdot (\dot{X} + Y)) \cdot l_W.$
3: Enclose $Q = I_V \cdot (A - S \cdot (\dot{X} + Y)) \cdot V.$
4: Enclose $E = (\Lambda - P)^* \cdot ZZ + ZZ \cdot (\Lambda - Q).$
5: Enclose $N = E / D.$
6: Enclose $U = W^* \cdot N \cdot I_V$
7: Output $U.$

Cost of Alg. 2 is also cubic.
Main Algorithm

1: Use a floating point algorithm to get an approximate solution $\tilde{X}$ of the Riccati equation (1).
2: Use a floating point algorithm to compute $V$, $W$ and $\Lambda$ in the spectral decomposition of $A - S\tilde{X}$.
3: Put $D \in \mathbb{C}^{n \times n}$ s.t. column $d_j$ of $D$ is $\text{diag}(\Lambda) + (\Lambda)_{jj}(1, \ldots, 1)^T$.
4: Compute interval matrices $I_W \ni W^{-1}$ and $I_V \ni V^{-1}$.
5: Use Alg. 1 to compute an enclosure $Z$ for $-R \cdot r(\tilde{X})$.
6: Put $X = Z$ and $k = 0$ {Prepare loop}
7: repeat
8: Put $Y = \Box((0, X \cdot [1 - \varepsilon, 1 + \varepsilon]))$, increment $k$ {\varepsilon-inflation}
9: Use Alg. 2 to compute an interval matrix $U$ such that $u$ is an enclosure for the set $\{(I_{n^2} - R \cdot r'(\tilde{x} + y))y : y \in Y\}$
10: Enclose $X = Z + U$
11: until $(X \subseteq \text{int } Y$ or $k = 15$)
12: if $X \subseteq \text{int } Y$ then {successful termination}
13: output $XX = \tilde{X} + X$
14: end if
Outline

1 Motivation
- The Riccati Equation and Some Basic Tools
- Our Main Problem
- Previous Works

2 Our Results/Contribution
- Main Results
- Algorithms
- Numerical Results
Example 16 (well-conditioned) from benchmark examples for Riccati equations by Benner, Laub and Mehrmann, 1995

\[ \text{ARESOLV from Matlab’s Robust Control Toolbox used for computing } \dot{X} \]

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \dot{X} )</th>
<th>Our Algorithm double prec. res.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time</td>
<td>time</td>
</tr>
<tr>
<td>100</td>
<td>( 2.3 \cdot 10^{-1} )</td>
<td>( 7.0 \cdot 10^{-1} )</td>
</tr>
<tr>
<td>200</td>
<td>( 4.6 \cdot 10^{-1} )</td>
<td>1.1</td>
</tr>
<tr>
<td>400</td>
<td>3.9</td>
<td>8.8</td>
</tr>
<tr>
<td>800</td>
<td>( 3.0 \cdot 10^{+1} )</td>
<td>( 6.4 \cdot 10^{+1} )</td>
</tr>
</tbody>
</table>
Example 5: A 9th-order continuous state space model of a tabular ammonia reactor

\[ n = 9, \ t_X = 0.01 \ \text{sec.}, \ t_X = 0.05 \ \text{sec.} \]

\[ mrp = 1.1 \times 10^{-12}, \ arp = 5.2 \times 10^{-14}. \]

Example 19: A model of 35 coupled springs, dashpots and masses

\[ n = 140, \ t_X = 2.2 \ \text{sec.} \]

\[ t_X = 5.5 \ \text{sec. after 7 iterations} \]

\[ mrp = 3.4 \times 10^{-9}, \ arp = 7.2 \times 10^{-13}. \]
Example 5: A 9th-order continuous state space model of a tabular ammonia reactor

\[ n = 9, \ t_X = 0.01 \text{ sec.}, \ t_X = 0.05 \text{ sec.} \]
\[ mrp = 1.1 \times 10^{-12}, \ arp = 5.2 \times 10^{-14}. \]

Example 19: A model of 35 coupled springs, dashpots and masses

\[ n = 140, \ t_X = 2.2 \text{ sec.} \]
\[ t_X = 5.5 \text{ sec. after 7 iterations} \]
\[ mrp = 3.4 \times 10^{-9}, \ arp = 7.2 \times 10^{-13}. \]
Example 17: A feedback controller

\( n = 21, \ t_{\dot{X}} = 0.01 \) sec.

Our algorithm fails because \( V \) is ill-conditioned

\[ \kappa_V = 2.4 \times 10^9, \]

condition number of Riccati equation: \( \kappa_{Ricc} = 1.3 \times 10^9. \)
Summary

- Reduction of the cost for verification to **cubic** via spectral decomposition of the closed loop matrix $\Rightarrow$ comparable to the cost for getting $\hat{X}$.
- Algorithm uses matrix-matrix operations $\Rightarrow$ fast in INTLAB.
- Algorithm will not succeed if the eigenvector matrix $V$ is ill-conditioned.

Outlook

- Verify stabilizing property of a solution to the CARE (1)
- Try recent algorithms for multiplication of interval matrices (by Rump & Ozaki, Ogita, Oishi & Nguyen, Revol and others)
- Discrete-time Riccati equations
Questions?

Comments?

Or suggestions?