A New Model Reduction Algorithm for Sequentially Semiseparable Matrices and Its Applications in System Identification

Yue Qiu[†], Martin B. van Gijzen[‡], Jan-Willem van Wingerden[†], Michel Verhaegen[†]

† Delft Center for System and Control
‡Delft Institute of Applied Mathematics
Delft University of Technology, Delft, the Netherlands

27th June, 2014, Strasbourg, France

TU Delft

Outline

Control and Identification of Distributed Systems

Control of spatially interconnected system arise in many applications

- multi-agent systems
- distributed power control
- control of PDEs
- ......

Challenge for control and identification of such systems

- High Computational Complexity

  N interconnected systems, each system with n states, $\mathcal{O}(n^3 N^3)$
- Memory Consumption

  $\mathcal{O}(n^2 N^2)$, not economical for large N.

Ways Out

Overcome the computational complexity

- Linear matrix inequality (LMI) approach

  Spatially invariant system transformed by fractional transformation, $\mathcal{O}(n^{2\alpha}N^{\alpha})$, where $2.5 < \alpha < 3.5$. [D'Andrea & Dullerud, TAC 2003.]

- Structured matrix approach

  Exploit the matrix structure, $\mathcal{O}(n^3N)$. [Rice & Verhaegen, TAC 2010, TAC 2011; Torres & van Wingerden, TAC 2014.]

- Approximate sparse inverse

  Exploit the sparsity of the system matrix, compute an approximate sparse inverse. [Haber & Verhaegen, TAC 2013; Lin & Jovanovic, TAC 2014.]

Reduce memory consumption

- Structured matrix approach

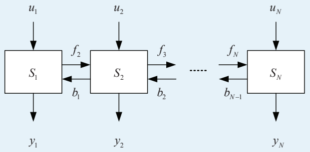
  $\mathcal{O}(n^2N)$ for sequentially semiseparable (SSS) matrix, $n \ll N$.

- Approximate sparse inverse

  Depends on the prescribed sparsity patten and approximation error, and $< \mathcal{O}(n^2N^2)$.

1-D Distributed Systems

Spatially Interconnected Systems



Local system

$$S_i: \begin{bmatrix} \dot{x}_i \\ f_{i+1} \\ b_{i-1} \\ y_i \end{bmatrix} = \begin{bmatrix} A_i & B_i^f & B_i^b & B_i \\ C_i^f & R_i & 0 & Q_i \\ C_i^b & 0 & W_i & V_i \\ C_i & G_i^f & G_i^b & D_i \end{bmatrix} \begin{bmatrix} x_i \\ f_i \\ b_i \\ u_i \end{bmatrix}$$

Global System Model

$$\bar{x} = \begin{bmatrix} x_1^T & x_2^T & \cdots & x_N^T \end{bmatrix}^T$$

$$\bar{u} = \begin{bmatrix} u_1^T & u_2^T & \cdots & u_N^T \end{bmatrix}^T$$

$$\bar{y} = \begin{bmatrix} y_1^T & y_2^T & \cdots & y_N^T \end{bmatrix}^T$$

Global system

$$\begin{bmatrix} \dot{\bar{x}} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} \bar{A} & \bar{B} \\ \bar{C} & \bar{D} \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{u} \end{bmatrix}$$

where,

$$\bar{A} = \begin{bmatrix} A_1 & B_1^b C_2^b & B_1^b W_2 C_3^b & B_1^b W_2 W_3 C_4^b \\ B_2^f C_1^f & A_2 & B_2^b C_3^b & B_2 W_3 C_4^b \\ B_3^f R_2 C_1^f & B_3^f C_2^f & A_3 & B_3 C_4^b \\ B_4^f R_3 R_2 C_1^f & B_4^f R_3 C_2^f & B_4^f C_3^f & A_4 \end{bmatrix}$$

SSS Matrices and Properties

Definition

Let A be an $N \times N$ block matrix with SSS structure, then A can be written in the following block partitioned form

$$A_{i,j} = \begin{cases} U_i W_{i+1} \cdots W_{j-1} V_j^T, & \text{if } i < j; \\ D_i, & \text{if } i = j; \\ P_i R_{i-1} \cdots R_{j+1} Q_j^T, & \text{if } i > j. \end{cases}$$

and denoted as $A = \mathcal{SSS}(P_S,\ R_S,\ Q_s,\ D_s,\ U_s,\ W_s,\ V_s)$.

Properties of SSS Matrices

- $+$, $\times$, and $()^{-1}$ are closed under such structure.
- Complexity of at most $\mathcal{O}(r^3 N)$
- $+$, $\times$ will lead to growth of the rank of the off-diagonal blocks, r_i, where $r = \max_i r_i$.
- Model order reduction (MOR) is necessary to keep the computational complexity low.

TUDelft

Model Reduction for SSS Matrices

Why Model Reduction?

The off-diagonal rank r defined by

$$r = max\left\{r_l, r_u\right\}$$

where

$$r_l = \max\left\{\text{size}(R_i)\right\}, \ r_u = \max\left\{\text{size}(W_i)\right\}$$

Basic matrix-matrix operations increase r (keep in mind $\mathcal{O}(r^3N)$).

What is Model Reduction?

To approximate the SSS matrix

$$A = \mathcal{SSS}(P_S,\ R_S,\ Q_s,\ D_s,\ U_s,\ W_s,\ V_s)$$

with

$$\tilde{A} = \mathcal{SSS}(\tilde{P}_S,\ \tilde{R}_S,\ \tilde{Q}_s,\ D_s,\ \tilde{U}_s,\ \tilde{W}_s,\ \tilde{V}_s)$$

where $\tilde{P}_S,\ \tilde{R}_S,\ \tilde{Q}_s,\ \tilde{U}_s,\ \tilde{W}_s,\ \tilde{V}_s$ have smaller size, and

$$\|A - \tilde{A}\| \leq \epsilon$$

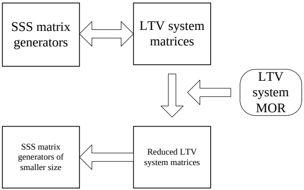
SSS Matrices and Its LTV System Realization

Motivation

Mixed-causal linear time-varying (LTV) system over finite time interval $[k_0,\ k_f]$,

$$\left.\begin{aligned}
\begin{bmatrix} x^c_{i+1} \\ x^a_{i-1} \end{bmatrix} &= \begin{bmatrix} R_i & \\ & W_i \end{bmatrix} \begin{bmatrix} x^c_i \\ x^a_i \end{bmatrix} + \begin{bmatrix} Q_i \\ V_i \end{bmatrix} u_i \\
y_i &= \begin{bmatrix} P_i & U_i \end{bmatrix} \begin{bmatrix} x^c_i \\ x^a_i \end{bmatrix} + D_i u_i \\
\bar{u} &= \begin{bmatrix} u_1^T, & u_2^T, & \dots & u_N^T \end{bmatrix}^T \\
\bar{y} &= \begin{bmatrix} y_1^T, & y_2^T, & \dots & y_N^T \end{bmatrix}^T
\end{aligned}\right\} \xRightarrow{N=4} \bar{y} = \begin{bmatrix} D_i & U_1 V_2 & U_1 W_2 V_3 & U_1 W_2 W_3 V_4 \\ P_2 Q_1 & D_2 & U_2 V_3 & U_2 W_3 V_4 \\ P_3 R_2 Q_1 & P_3 Q_2 & D_3 & U_3 V_4 \\ P_4 R_3 R_2 Q_1 & P_4 R_3 Q_2 & P_4 Q_3 & D_4 \end{bmatrix} \bar{u}$$

How to do Model Reduction?



Approximate Balanced Truncation for LTV Systems

Low Rank Approximation of the Gramians

Causal part of the mixed-causal LTV system,

$$\begin{cases} x_{k+1} = A_k x_k + B_k u_k \\ y_k = C_k x_k \end{cases}$$

over $[k_o, k_f]$, where $A_k \in \mathbb{R}^{M_{k+1} \times M_k}$, $B_k \in \mathbb{R}^{M_{k+1} \times m_k}$, and $C_k \in \mathbb{R}^{n_k \times M_k}$.

Controllability gramian $\mathcal{G}_c(k)$ and observability gramian $\mathcal{G}_o(k)$

$$\begin{aligned} \mathcal{G}_c(k+1) &= A_k \mathcal{G}_c(k) A_k^T + B_k B_k^T \\ \mathcal{G}_o(k) &= A_k^T \mathcal{G}_o(k+1) A_k + C_k^T C_k \end{aligned}$$

with $\mathcal{G}_c(k_o) = 0$ and $\mathcal{G}_o(k_f + 1) = 0$. $\mathcal{G}_c(k)$ and $\mathcal{G}_o(k)$ are symmetric positive semi-definite, usually of low numerical rank.

$$\begin{aligned} & \mathcal{G}_c(k) = L_k L_k^T, \quad L_k \in \mathbb{R}^{M_k \times r_k}, \quad r_k \le M_k, \\ \xrightarrow{\text{Low rank approx.}} \quad & \mathcal{G}_c(k) \approx \tilde{L}_k \tilde{L}_k^T, \quad \tilde{L}_k \in \mathbb{R}^{M_k \times r}, \quad r \le r_k. \end{aligned}$$

TUDelft

Approximate Balanced Truncation for LTV Systems (cont'd)

Low Rank Update

Approximate $\mathcal{G}_c(k) \approx \tilde{\mathcal{G}}_c(k) = \tilde{L}_k \tilde{L}_k^T$, update $\tilde{\mathcal{G}}_c(k+1)$

- $\left[\begin{array}{c|c} A_k \tilde{L}_k & B_k \end{array}\right] = U \Sigma V^T$
- $U = \left[\begin{array}{c|c} U_1 & U_2 \end{array}\right]$, $\Sigma = \left[\begin{array}{c|c} \Sigma_1 & \\ \hline & \Sigma_2 \end{array}\right]$
- $\tilde{L}_{k+1} = U_1 \Sigma_1$

Approximate Balanced Truncation

Balancing via

$$\tilde{\mathcal{G}}_c^T(k) \tilde{\mathcal{G}}_o(k) = U_k \Sigma_k V_k^T$$

truncation by

$$\Pi_l(k) = \tilde{\mathcal{G}}_o(k) V_k \Sigma_k^{-\frac{1}{2}}, \ \Pi_r(k) = \tilde{\mathcal{G}}_c(k) U_k \Sigma_k^{-\frac{1}{2}}$$

gives

$$\begin{cases} \tilde{x}_{k+1} = \Pi_l(k+1) A_k \Pi_r(k) \tilde{x}_k + \Pi_l(k+1) B_k u_k \\ y_k = C_k \Pi_r(k) \tilde{x}_k \end{cases}.$$

TUDelft

Extended Kalman Filter for 1-D System Identification

1-D Heat Equation

Heat equation over $x \in [0, 1]$

$$\frac{\partial u(x,t)}{\partial t} = \nabla \cdot (k(x)\nabla u(x,t)) - \frac{1}{2}u(x,t)$$

$u(x, t_0) = u_0$ on $\partial\Gamma$, $\nabla\cdot$ divergence operator and ∇ gradient operator, $k(x)$ the heat conduction coefficients.

$$\text{PDE} \overset{\text{FEM/FDM}}{\Longrightarrow} \text{ODE} \overset{\text{Euler Method}}{\Longrightarrow} \begin{cases} T_{k+1} = A(\theta)T_k + B(\theta)u_k \\ y_k = C(\theta)T_k \end{cases}$$

with $\theta = \{k(x), \Delta t\}$

$$\begin{cases} T_{k+1} = A(\theta)T_k + B(\theta)u_k + \omega_k \\ y_k = C(\theta)T_k + v_k \end{cases}$$

where $E(\omega_k\omega_j^T) = Q\delta_{kj}$, $E(v_k v_j^T) = R\delta_{jk}$, $E(\omega_k v_j^T) = S\delta_{kj}$, $E(x_0) = 0$, $E(x_0 x_0^T) = \Pi_x$ and $E(\theta\theta^T) = \Pi_\theta$.

TU Delft

Extended Kalman Filter for SysID

Innovations Form

$$\begin{cases} \widehat{T}_{k+1} = A(\widehat{\theta}_k)\widehat{T}_k + B(\widehat{\theta}_k)u_k + K_k\left(y_k - C(\widehat{\theta}_k)\widehat{T}_k\right) \\ \widehat{\theta}_{k+1} = \widehat{\theta}_k + L_k\left(y_k - C(\widehat{\theta}_k)\widehat{T}_k\right) \end{cases}$$

Kalman gains

$$\begin{bmatrix} K_k \\ L_k \end{bmatrix} = \left(\begin{bmatrix} A_k & M_k \\ 0 & I \end{bmatrix} \Xi_k \begin{bmatrix} C_k & N_k \end{bmatrix}^T + \begin{bmatrix} S \\ 0 \end{bmatrix} \right) P_k^{-1}.$$

and $A_k = A(\widehat{\theta}_k)$, $M_k = \frac{\partial}{\partial \theta}\left(A(\theta)\widehat{T}_k + B(\theta)u_k\right)|_{\theta=\widehat{\theta}_k}$, $C_k = C(\widehat{\theta}_k)$,
$N_k = \frac{\partial}{\partial \theta}\left(C(\theta)\widehat{T}_k\right)|_{\theta=\widehat{\theta}_k}$ where

$$P_k = \begin{bmatrix} C_k & N_k \end{bmatrix} \Xi_k \begin{bmatrix} C_k & N_k \end{bmatrix}^T + R$$

$$\Xi_{k+1} = \begin{bmatrix} A_k & M_k \\ 0 & I \end{bmatrix} \Xi_k \begin{bmatrix} A_k & M_k \\ 0 & I \end{bmatrix}^T - \begin{bmatrix} K_k \\ L_k \end{bmatrix} P_k \begin{bmatrix} K_k \\ L_k \end{bmatrix}^T + \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}$$

Numerical Experiments

Choose $k(x) \in 1 + \frac{2}{5}\mathcal{U}[-1,1]$, $\omega \sim \mathcal{N}(0, 10^{-6})$, $v \sim \mathcal{N}(0, 10^{-6})$

Set initial guess $\Xi_x = \Xi_\theta = I$, $k_0 = \mathbf{1}$. Stop criteria $\frac{\|\hat{\theta}_k - \theta^\star\|_2}{\|\theta^\star\|_2} \leq 5 \times 10^{-4}$

Random run the numerical experiments 5 times.

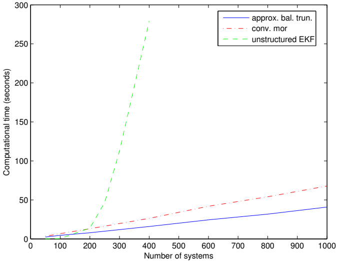


Figure: Average computational time

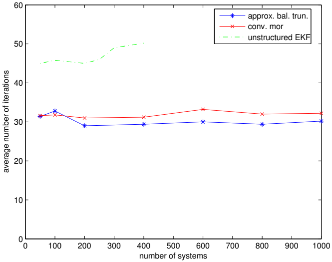


Figure: Average number of iterations

Numerical Experiments (cont'd)

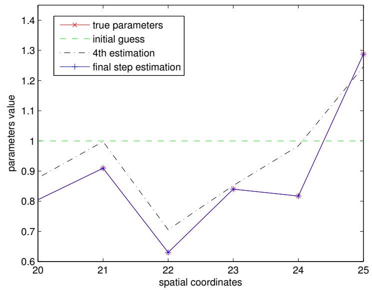


Figure: Estimated parameters

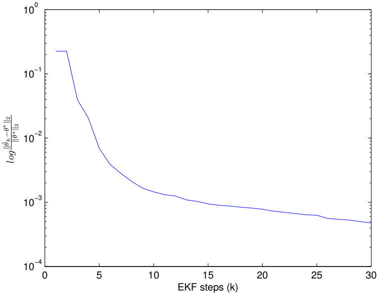


Figure: Asypototic convergence

TUDelft

Conclusions & Remarks

- SSS matrix computations yield linear computational complexity for EKF for 1-D SysID.
- The approximate balanced truncation is computationally cheaper than the conventional method.
- Both MOR algorithms give linear computational complexity.
- To extend to higher dimensional systems, such as 2-D or 3-D, multilevel SSS matrix is necessary.
- For multilevel SSS matrices, structure preserving MOR is the key, still an open problem.

Conclusions & Remarks

- SSS matrix computations yield linear computational complexity for EKF for 1-D SysID.
- The approximate balanced truncation is computationally cheaper than the conventional method.
- Both MOR algorithms give linear computational complexity.
- To extend to higher dimensional systems, such as 2-D or 3-D, multilevel SSS matrix is necessary.
- For multilevel SSS matrices, structure preserving MOR is the key, still an open problem.

Conclusions & Remarks

- SSS matrix computations yield linear computational complexity for EKF for 1-D SysID.
- The approximate balanced truncation is computationally cheaper than the conventional method.
- Both MOR algorithms give linear computational complexity.
- To extend to higher dimensional systems, such as 2-D or 3-D, multilevel SSS matrix is necessary.
- For multilevel SSS matrices, structure preserving MOR is the key, still an open problem.

TUDelft

Conclusions & Remarks

- SSS matrix computations yield linear computational complexity for EKF for 1-D SysID.
- The approximate balanced truncation is computationally cheaper than the conventional method.
- Both MOR algorithms give linear computational complexity.
- To extend to higher dimensional systems, such as 2-D or 3-D, multilevel SSS matrix is necessary.
- For multilevel SSS matrices, structure preserving MOR is the key, still an open problem.

Conclusions & Remarks

- SSS matrix computations yield linear computational complexity for EKF for 1-D SysID.
- The approximate balanced truncation is computationally cheaper than the conventional method.
- Both MOR algorithms give linear computational complexity.
- To extend to higher dimensional systems, such as 2-D or 3-D, multilevel SSS matrix is necessary.
- For multilevel SSS matrices, structure preserving MOR is the key, still an open problem.