Consistent and computationally efficient estimation for stochastic LPV state-space models: realization based approach.

Manas Mejari and Mihály Petreczky

financial support ERDF, French state, CNRS project PEPS blanc BayesRealForRNN

CDC 2019, Nice, France

USI/SUPSI
IDSIA
Istituto Dalle Molle di studi sull'intelligenza artificiale

CRIStAL
Centre de Recherche en Informatique, Signal et Automatique de Lille

Outline

Contributions

- *Realization* and *identification* problem for stochastic LPV-SSA representations.
- stochastic covariance realization algorithm.
- Identification algorithm
- Simulation case study

Problem formulation

LPV State-Space Affine (LPV-SSA) representations:

$$\mathbf{x}(t+1) = A(\boldsymbol{\mu}(t))\mathbf{x}(t) + K(\boldsymbol{\mu}(t))\mathbf{v}(t),$$
$$\bar{\mathbf{y}}(t) = C\mathbf{x}(t) + D\mathbf{v}(t)$$

- $(\mathbf{x}, \bar{\mathbf{y}}, \mathbf{v})$ denote state, output, noise *processes*,
- $\boldsymbol{\mu}$ scheduling signal process taking values in $\mathbb{R}^d$.

$$A(\boldsymbol{\mu}(t)) = A_1 + \sum_{i=2}^{d} A_i \boldsymbol{\mu}_i(t), \quad K(\boldsymbol{\mu}(t)) = K_1 + \sum_{i=2}^{d} K_i \boldsymbol{\mu}_i(t).$$

- Noise covariance $Q_i = \mathbb{E}\left[\mathbf{v}(t)\mathbf{v}^\top(t)\boldsymbol{\mu}_i(t)^2\right] \quad \forall i = 1, \cdots, d$

Problem formulation

LPV State-Space Affine (LPV-SSA) representations:

$$\mathbf{x}(t+1) = A(\boldsymbol{\mu}(t))\mathbf{x}(t) + K(\boldsymbol{\mu}(t))\mathbf{v}(t),$$
$$\bar{\mathbf{y}}(t) = C\mathbf{x}(t) + D\mathbf{v}(t)$$

- $(\mathbf{x}, \bar{\mathbf{y}}, \mathbf{v})$ denote state, output, noise *processes*,
- $\boldsymbol{\mu}$ scheduling signal process taking values in $\mathbb{R}^d$.

$$A(\boldsymbol{\mu}(t)) = A_1 + \sum_{i=2}^{d} A_i \boldsymbol{\mu}_i(t), \quad K(\boldsymbol{\mu}(t)) = K_1 + \sum_{i=2}^{d} K_i \boldsymbol{\mu}_i(t).$$

- Noise covariance $Q_i = \mathbb{E}\left[\mathbf{v}(t)\mathbf{v}^\top(t)\boldsymbol{\mu}_i(t)^2\right] \quad \forall i = 1, \cdots, d$

Problem formulation

Problem (Realization problem)

For observed process $(\mathbf{y}, \boldsymbol{\mu})$, find matrices $(\{A_i, K_i\}_{i=1}^{d}, C, D)$ and processes $\mathbf{x}, \mathbf{v}$ such that LPV-SSA is a realization of $\mathbf{y}$, i.e., $\mathbf{y}(t) = \bar{\mathbf{y}}(t),\ t \in \mathbb{Z}$.

Problem (Identification problem)

Given sample paths $\{y(t), \mu(t)\}_{t=1}^{N}$, compute the estimates $\{\{\hat{A}_i^N, \hat{K}_i^N, \hat{Q}_i^N\}_{i=1}^{d}, \hat{C}^N, \hat{D}^N\}$, such that as $N \to \infty$, the estimated matrices converge to matrices $\{\{A_i, K_i, Q_i\}_{i=1}^{d}, C, D\}$ of true system, which is a representation of $(\mathbf{y}, \boldsymbol{\mu})$.

Problem formulation

Problem (Realization problem)

For observed process $(\mathbf{y}, \boldsymbol{\mu})$*, find matrices* $(\{A_i, K_i\}_{i=1}^{d}, C, D)$ *and processes* $\mathbf{x}, \mathbf{v}$ *such that LPV-SSA is a realization of* $\mathbf{y}$*, i.e.,* $\mathbf{y}(t) = \bar{\mathbf{y}}(t),\ t \in \mathbb{Z}$*.*

Problem (Identification problem)

Given sample paths $\{y(t), \mu(t)\}_{t=1}^{N}$*, compute the estimates* $\{\{\hat{A}_i^N, \hat{K}_i^N, \hat{Q}_i^N\}_{i=1}^{d}, \hat{C}^N, \hat{D}^N\}$*, such that as* $N \to \infty$*, the estimated matrices converge to matrices* $\{\{A_i, K_i, Q_i\}_{i=1}^{d}, C, D\}$ *of true system, which is a representation of* $(\mathbf{y}, \boldsymbol{\mu})$*.*

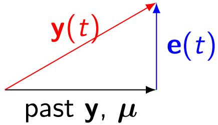
Assumptions

- The scheduling process $\boldsymbol{\mu} = [1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_d]^\top$ is i.i.d.
  $\mathbb{E}\left[\boldsymbol{\mu}_i^2\right] = p_i$
- $\begin{bmatrix}\mathbf{x}^\top & \mathbf{v}^\top\end{bmatrix}^\top$ satisfies technical conditions: is a Zero Mean Wide Sense Stationary w.r.t. scheduling Inputs
- The LPV-SSA representation $(\{A_i, K_i\}_{i=1}^d, C, D, \mathbf{v})$ satisfies technical condition: it is stationary for $\boldsymbol{\mu}$

  Stationary LPV-SSA: stable, additive white noise, statistical independence conditions for the noise, input and state.

Forward innovation form

- the *forward innovation process* $\mathbf{e}$
  $\mathbf{e}(t) = \mathbf{y}(t) - E_l[\mathbf{y}(t) \mid \text{past } \mathbf{y}, \boldsymbol{\mu}]$, E_l denotes projection.



- The stationary LPV-SSA $(\{A_i, K_i\}_{i=1}^{d}, C, D, \mathbf{v})$ in forward innovation form if $\mathbf{v} = \mathbf{e}$ and $D = I_p$, *i.e.*,

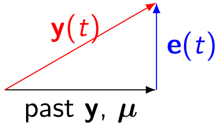
$$\mathbf{x}(t+1) = \sum_{i=1}^{d} \left(A_i \mathbf{x}(t) + K_i \mathbf{e}(t)\right) \boldsymbol{\mu}_i(t), \mathbf{y}(t) = C\mathbf{x}(t) + \mathbf{e}(t).$$

Theorem

Under suitable assumptions, every LPV-SSA representation can be transformed to minimal forward innovation form.

Forward innovation form

- the *forward innovation process* $\mathbf{e}$
  $\mathbf{e}(t) = \mathbf{y}(t) - E_l[\mathbf{y}(t) \mid \text{past } \mathbf{y}, \boldsymbol{\mu}]$, E_l denotes projection.



- The stationary LPV-SSA $(\{A_i, K_i\}_{i=1}^{d}, C, D, \mathbf{v})$ in forward innovation form if $\mathbf{v} = \mathbf{e}$ and $D = I_p$, *i.e.*,

$$\mathbf{x}(t+1) = \sum_{i=1}^{d} \left(A_i \mathbf{x}(t) + K_i \mathbf{e}(t)\right) \boldsymbol{\mu}_i(t), \mathbf{y}(t) = C\mathbf{x}(t) + \mathbf{e}(t).$$

Theorem

Under suitable assumptions, every LPV-SSA representation can be transformed to minimal forward innovation form.

Stochastic realization algorithm

- The covariance sequence $\Psi_{\mathbf{y}}(i_1 \cdots i_k)$ is defined as

$$\Psi_{\mathbf{y}}(i_1 \cdots i_k) = \mathbb{E}\left[\mathbf{y}(t)\frac{\mu_{i_k}(t-1)}{\sqrt{p_{i_k}}} \cdots \frac{\mu_{i_1}(t-k)}{\sqrt{p_{i_1}}}(\mathbf{y}(t-k))^\top\right] \quad k > 0$$

  constructed from past outputs and scheduling.
- $\Psi_{\mathbf{y}}(i_1 \cdots i_k)$ are a sub-Markov parameters of deterministic LPV-SSA: $\Psi_{\mathbf{y}}(i_1 \cdots i_k) = CA_{i_k} \cdots A_{i_1} B_{i_1}$

$$z(t+1) = \sum_{i=1}^{d}(A_i z(t) + B_i w(t))\mu_i(t), \; \mathbf{y}(t) = Cz(t)$$

- Let state and noise covariance be $P_i = \mathbb{E}\left[\mathbf{x}(t)\mathbf{x}^\top(t)\mu_i^2(t)\right], \; Q_i = \mathbb{E}\left[\mathbf{e}(t)\mathbf{e}^\top(t)\mu_i^2(t)\right]$
- $B_i = \frac{1}{\sqrt{p_i}}(A_\sigma P_i C^\top + K_i Q_i) \;\; \forall i = 1, \cdots, d$

Stochastic realization algorithm

- The covariance sequence $\Psi_{\mathbf{y}}(i_1 \cdots i_k)$ is defined as

$$\Psi_{\mathbf{y}}(i_1 \cdots i_k) = \mathbb{E}\left[\mathbf{y}(t)\frac{\boldsymbol{\mu}_{i_k}(t-1)}{\sqrt{p_{i_k}}} \cdots \frac{\boldsymbol{\mu}_{i_1}(t-k)}{\sqrt{p_{i_1}}}(\mathbf{y}(t-k))^\top\right] \quad k > 0$$

  constructed from past outputs and scheduling.
- $\Psi_{\mathbf{y}}(i_1 \cdots i_k)$ are a sub-Markov parameters of deterministic LPV-SSA: $\Psi_{\mathbf{y}}(i_1 \cdots i_k) = CA_{i_k} \cdots A_{i_1} B_{i_1}$

$$z(t+1) = \sum_{i=1}^{d} (A_i z(t) + B_i w(t))\boldsymbol{\mu}_i(t), \; \mathbf{y}(t) = Cz(t)$$

- Let state and noise covariance be $P_i = \mathbb{E}\left[\mathbf{x}(t)\mathbf{x}^\top(t)\mu_i^2(t)\right], \; Q_i = \mathbb{E}\left[\mathbf{e}(t)\mathbf{e}^\top(t)\mu_i^2(t)\right]$
- $B_i = \frac{1}{\sqrt{p_i}}(A_\sigma P_i C^\top + K_i Q_i) \;\; \forall i = 1, \cdots, d$

Stochastic realization algorithm

- The covariance sequence $\Psi_{\mathbf{y}}(i_1 \cdots i_k)$ is defined as

$$\Psi_{\mathbf{y}}(i_1 \cdots i_k) = \mathbb{E}\left[\mathbf{y}(t)\frac{\boldsymbol{\mu}_{i_k}(t-1)}{\sqrt{p_{i_k}}} \cdots \frac{\boldsymbol{\mu}_{i_1}(t-k)}{\sqrt{p_{i_1}}}(\mathbf{y}(t-k))^\top\right] \quad k > 0$$

  constructed from past outputs and scheduling.
- $\Psi_{\mathbf{y}}(i_1 \cdots i_k)$ are a sub-Markov parameters of deterministic LPV-SSA: $\Psi_{\mathbf{y}}(i_1 \cdots i_k) = CA_{i_k} \cdots A_{i_1} B_{i_1}$

$$z(t+1) = \sum_{i=1}^{d} (A_i z(t) + B_i w(t))\boldsymbol{\mu}_i(t), \ \mathbf{y}(t) = Cz(t)$$

- Let state and noise covariance be
  $P_i = \mathbb{E}\left[\mathbf{x}(t)\mathbf{x}^\top(t)\boldsymbol{\mu}_i^2(t)\right], \ Q_i = \mathbb{E}\left[\mathbf{e}(t)\mathbf{e}^\top(t)\boldsymbol{\mu}_i^2(t)\right]$
- $B_i = \frac{1}{\sqrt{p_i}}(A_\sigma P_i C^\top + K_i Q_i) \quad \forall i = 1, \cdots, d$

Stochastic realization algorithm

- The covariance sequence $\Psi_{\mathbf{y}}(i_1 \cdots i_k)$ is defined as

$$\Psi_{\mathbf{y}}(i_1 \cdots i_k) = \mathbb{E}\left[\mathbf{y}(t)\frac{\boldsymbol{\mu}_{i_k}(t-1)}{\sqrt{p_{i_k}}} \cdots \frac{\boldsymbol{\mu}_{i_1}(t-k)}{\sqrt{p_{i_1}}}(\mathbf{y}(t-k))^\top\right] \quad k > 0$$

  constructed from past outputs and scheduling.
- $\Psi_{\mathbf{y}}(i_1 \cdots i_k)$ are a sub-Markov parameters of deterministic LPV-SSA: $\Psi_{\mathbf{y}}(i_1 \cdots i_k) = CA_{i_k} \cdots A_{i_1} B_{i_1}$

$$z(t+1) = \sum_{i=1}^{d}(A_i z(t) + B_i w(t))\boldsymbol{\mu}_i(t), \;\; \mathbf{y}(t) = Cz(t)$$

- Let state and noise covariance be
  $P_i = \mathbb{E}\left[\mathbf{x}(t)\mathbf{x}^\top(t)\boldsymbol{\mu}_i^2(t)\right], \;\; Q_i = \mathbb{E}\left[\mathbf{e}(t)\mathbf{e}^\top(t)\boldsymbol{\mu}_i^2(t)\right]$
- $B_i = \frac{1}{\sqrt{p_i}}(A_\sigma P_i C^\top + K_i Q_i) \;\; \forall i = 1, \cdots, d$

Stochastic realization algorithm: Riccati equations

- The state and noise covariance satisfy Riccati-like equations

$$P_i = \sum_j p_i \left(A_j P_j A_j^\top + K_j Q_j K_j^\top \right)$$

$$Q_i = p_i \mathbb{E}\left[\mathbf{y}(t)\mathbf{y}^\top(t)\boldsymbol{\mu}_i^2(t) \right] - C P_i C^\top$$

$$K_i = \left(B_i \sqrt{p_i} - A_i P_i C^\top \right) (Q_i)^{-1}$$

- The equations can be solved iteratively $k = 1, 2, \cdots, \mathcal{I}$

$$\hat{P}_i^{k+1} = \sum_j p_i \left(\hat{A}_j \hat{P}_j^k \hat{A}_j^\top + \hat{K}_j \hat{Q}_j^j \hat{K}_j^\top \right)$$

$$\hat{Q}_i^k = p_i \mathbb{E}\left[\mathbf{y}(t)\mathbf{y}^\top(t)\boldsymbol{\mu}_i^2(t) \right] - \hat{C} \hat{P}_i^k \hat{C}^\top$$

$$\hat{K}_i^k = \left(\hat{B}_i \sqrt{p_i} - \hat{A}_i \hat{P}_i^k (\hat{C})^\top \right) \left(\hat{Q}_i^k \right)^{-1}$$

Stochastic realization algorithm: Riccati equations

- The state and noise covariance satisfy Riccati-like equations

$$P_i = \sum_j p_i \left(A_j P_j A_j^\top + K_j Q_j K_j^\top \right)$$

$$Q_i = p_i \mathbb{E}\left[\mathbf{y}(t) \mathbf{y}^\top(t) \boldsymbol{\mu}_i^2(t) \right] - C P_i C^\top$$

$$K_i = \left(B_i \sqrt{p_i} - A_i P_i C^\top \right) (Q_i)^{-1}$$

- The equations can be solved iteratively $k = 1, 2, \cdots, \mathcal{I}$

$$\hat{P}_i^{k+1} = \sum_j p_i \left(\hat{A}_j \hat{P}_j^k \hat{A}_j^\top + \hat{K}_j \hat{Q}_j^i \hat{K}_j^\top \right)$$

$$\hat{Q}_i^k = p_i \mathbb{E}\left[\mathbf{y}(t) \mathbf{y}^\top(t) \boldsymbol{\mu}_i^2(t) \right] - \hat{C} \hat{P}_i^k \hat{C}^\top$$

$$\hat{K}_i^k = \left(\hat{B}_i \sqrt{p_i} - \hat{A}_i \hat{P}_i^k (\hat{C})^\top \right) \left(\hat{Q}_i^k \right)^{-1}$$

Stochastic realization algorithm

Main steps:

1. Construct covariance sequence $\Psi_{\mathbf{y}}(i_1 \cdots i_k) = CA_{i_1}A_{i_2}\cdots A_{i_k}B$ from the past $\mathbf{y}$ and $\boldsymbol{\mu}$.
2. Construct Hankel matrices $H_{\Psi_{\mathbf{y}}}$ from the covariance sequence $\Psi_{\mathbf{y}}(i_1 \cdots i_k)$.
3. Estimate $(\{A_i, B_i\}_{i=1}^{d}, C)$ from the $H_{\Psi_{\mathbf{y}}}$, using Ho-Kalman like realization algorithm.
4. Estimate the $\{K_i\}_{i=1}^{d}$ and noise-covariances $\{P_i, Q_i\}_{i=1}^{d}$ solving Riccatti-like equations.

Stochastic realization algorithm

Main steps:

1. Construct covariance sequence $\Psi_{\mathbf{y}}(i_1 \cdots i_k) = CA_{i_1}A_{i_2} \cdots A_{i_k}B$ from the past $\mathbf{y}$ and $\boldsymbol{\mu}$.
2. Construct Hankel matrices $H_{\Psi_{\mathbf{y}}}$ from the covariance sequence $\Psi_{\mathbf{y}}(i_1 \cdots i_k)$.
3. Estimate $(\{A_i, B_i\}_{i=1}^{d}, C)$ from the $H_{\Psi_{\mathbf{y}}}$, using Ho-Kalman like realization algorithm.
4. Estimate the $\{K_i\}_{i=1}^{d}$ and noise-covariances $\{P_i, Q_i\}_{i=1}^{d}$ solving Riccatti-like equations.

Stochastic realization algorithm

Main steps:

1. Construct covariance sequence $\Psi_{\mathbf{y}}(i_1 \cdots i_k) = CA_{i_1}A_{i_2} \cdots A_{i_k}B$ from the past $\mathbf{y}$ and $\boldsymbol{\mu}$.
2. Construct Hankel matrices $H_{\Psi_{\mathbf{y}}}$ from the covariance sequence $\Psi_{\mathbf{y}}(i_1 \cdots i_k)$.
3. Estimate $(\{A_i, B_i\}_{i=1}^{d}, C)$ from the $H_{\Psi_{\mathbf{y}}}$, using Ho-Kalman like realization algorithm.
4. Estimate the $\{K_i\}_{i=1}^{d}$ and noise-covariances $\{P_i, Q_i\}_{i=1}^{d}$ solving Riccatti-like equations.

Stochastic realization algorithm

Main steps:

1. Construct covariance sequence $\Psi_{\mathbf{y}}(i_1 \cdots i_k) = CA_{i_1}A_{i_2} \cdots A_{i_k}B$ from the past $\mathbf{y}$ and $\boldsymbol{\mu}$.
2. Construct Hankel matrices $H_{\Psi_{\mathbf{y}}}$ from the covariance sequence $\Psi_{\mathbf{y}}(i_1 \cdots i_k)$.
3. Estimate $(\{A_i, B_i\}_{i=1}^{d}, C)$ from the $H_{\Psi_{\mathbf{y}}}$, using Ho-Kalman like realization algorithm.
4. Estimate the $\{K_i\}_{i=1}^{d}$ and noise-covariances $\{P_i, Q_i\}_{i=1}^{d}$ solving Riccatti-like equations.

Stochastic realization algorithm: selections

- The construction of full Hankel matrix $H_{\Psi_\mathbf{y}}$ is avoided.
- We use the idea of selections [Cox et. al. '18, Sontag et.al '70] to select specific entries of the Hankel matrix $H_{\Psi_\mathbf{y}}$
- The selection pair (α, β) is chosen to select the entries of the Hankel matrix $H_{\Psi_\mathbf{y}}$ to form a sub-Hankel matrix $H_{\alpha,\beta}$.
- The sub-Hankel matrix $H_{\alpha,\beta}$ is such that,

$$\text{rank } H_{\alpha,\beta} = n$$

Stochastic realization algorithm: selections

- The construction of full Hankel matrix H_{Ψ_y} is avoided.
- We use the idea of selections [Cox et. al. '18, Sontag et.al '70] to select specific entries of the Hankel matrix H_{Ψ_y}
- The selection pair (α, β) is chosen to select the entries of the Hankel matrix H_{Ψ_y} to form a sub-Hankel matrix $H_{\alpha,\beta}$.
- The sub-Hankel matrix $H_{\alpha,\beta}$ is such that,

$$\text{rank } H_{\alpha,\beta} = n$$

Stochastic realization algorithm: selections

- The construction of full Hankel matrix $H_{\Psi_{\mathbf{y}}}$ is avoided.
- We use the idea of selections [Cox et. al. '18, Sontag et.al '70] to select specific entries of the Hankel matrix $H_{\Psi_{\mathbf{y}}}$
- The selection pair (α, β) is chosen to select the entries of the Hankel matrix $H_{\Psi_{\mathbf{y}}}$ to form a sub-Hankel matrix $H_{\alpha,\beta}$.
- The sub-Hankel matrix $H_{\alpha,\beta}$ is such that,

$$\text{rank } H_{\alpha,\beta} = n$$

Identification algorithm

Main idea: estimate the covariance sequences $\Psi_{\mathbf{y}}$ from the observed sample paths $\{y(t), \mu(t)\}_{t=1}^{N}$.

1. Compute empirical covariances from data, approximating expectations with sample mean
2. Run stochastic realization algorithm using empirical covariances for $\mathcal{I}$ iterations to get
   - Model matrices $(\{\tilde{A}_i, \tilde{K}_i^{\mathcal{I}}\}_{i=1}^{d}, \tilde{C})$
   - Noise and state covariances $(\{\tilde{Q}_i^{\mathcal{I}}, \tilde{P}_i^{\mathcal{I}}\}_{i=1}^{d})$

Identification algorithm: Consistency

Theorem (Consistency)

The estimates satisfy

$$\tilde{K}_i = \lim_{\mathcal{I}\to\infty} \lim_{N\to\infty} \tilde{K}_i^{N,\mathcal{I}},$$
$$\tilde{A}_i = \lim_{N\to\infty} \tilde{A}_i^{N},$$
$$\tilde{C} = \lim_{N\to\infty} \tilde{C}^{N}$$

and $(\{\tilde{A}_i, \tilde{K}_i,\}_{i=1}^{d}, \tilde{C}, \hat{\mathbf{x}}, \mathbf{e})$ *is a minimal stationary LPV-SSA representation of* $(\mathbf{y}, \boldsymbol{\mu})$ *and*
$\mathbb{E}\left[\mathbf{e}(t)(\mathbf{e}(t))^\top \boldsymbol{\mu}_i^2(t)\right] = \lim_{\mathcal{I}\to\infty} \lim_{N\to\infty} \tilde{Q}_i^{N,\mathcal{I}}$, $i = 1, \cdots d$

Identification algorithm: Consistency

Theorem (Consistency)

The estimates satisfy

$$\tilde{K}_i = \lim_{\mathcal{I}\to\infty} \lim_{N\to\infty} \tilde{K}_i^{N,\mathcal{I}},$$
$$\tilde{A}_i = \lim_{N\to\infty} \tilde{A}_i^{N},$$
$$\tilde{C} = \lim_{N\to\infty} \tilde{C}^{N}$$

and $(\{\tilde{A}_i, \tilde{K}_i, \}_{i=1}^{d}, \tilde{C}, \hat{\mathbf{x}}, \mathbf{e})$ *is a minimal stationary LPV-SSA representation of* $(\mathbf{y}, \boldsymbol{\mu})$ *and*
$\mathbb{E}\left[\mathbf{e}(t)(\mathbf{e}(t))^\top \boldsymbol{\mu}_i^2(t)\right] = \lim_{\mathcal{I}\to\infty} \lim_{N\to\infty} \tilde{Q}_i^{N,\mathcal{I}}$, $i = 1, \cdots d$

Identification algorithm: Consistency

Theorem (Consistency)

The estimates satisfy

$$\tilde{K}_i = \lim_{\mathcal{I}\to\infty} \lim_{N\to\infty} \tilde{K}_i^{N,\mathcal{I}},$$
$$\tilde{A}_i = \lim_{N\to\infty} \tilde{A}_i^{N},$$
$$\tilde{C} = \lim_{N\to\infty} \tilde{C}^{N}$$

and $(\{\tilde{A}_i, \tilde{K}_i, \}_{i=1}^{d}, \tilde{C}, \hat{\mathbf{x}}, \mathbf{e})$ *is a minimal stationary LPV-SSA representation of* $(\mathbf{y}, \boldsymbol{\mu})$ *and*
$\mathbb{E}\left[\mathbf{e}(t)(\mathbf{e}(t))^\top \boldsymbol{\mu}_i^2(t)\right] = \lim_{\mathcal{I}\to\infty} \lim_{N\to\infty} \tilde{Q}_i^{N,\mathcal{I}}$, $i = 1, \cdots d$

Numerical Example

The LPV-SSA representation in forward innovation form

$$A_1 = \begin{bmatrix} 0.4 & 0.4 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.4 & 0.4 \\ 0 & 0.4 & 0.4 \end{bmatrix},$$

$$K_1 = \begin{bmatrix} -0.036 \\ 0 \\ 1 \end{bmatrix}, K_2 = \begin{bmatrix} 0 \\ 0.015 \\ 1.17 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix},$$

- Training data $N = 100,000$, validation $N_{\mathrm{val}} = 100000$.
- Scheduling signal process $\boldsymbol{\mu} = [\boldsymbol{\mu}_1 \;\; \boldsymbol{\mu}_2]$ such that $\boldsymbol{\mu}_1(t) = 1$ and $\boldsymbol{\mu}_2(t)$ is a white-noise process with $\mathcal{U}(-1.5, 1.5)$
- innovation noise $\mathbf{e} \sim \mathcal{N}(0, 1)$, *Signal-to-Noise Ratio* 4.7 dB

Numerical Example

- the first local model $\tilde{A}_1 = A_1 - K_1 C$ is not observable, which is a typical assumption in few PBSID algorithms.
- We run the proposed Algorithm with $\mathcal{I} = 50$ iterations and with the following n-selection pair (α, β), with $n = 3$,

$$\alpha = \{(\epsilon, 1), (1, 1), (21, 1)\}, \ \beta = \{(\epsilon, 1), (2, 1), (21, 1)\}$$

- *Best Fit Rate* (BFR) and *Variance Accounted For* (VAF)

BFR	93.98 %
VAF	99.74 %

- The PBSID [van Wingerden et.al. 2009] approach failed for this example.

Conclusions

- Computationally efficient algorithm is proposed based on the idea of selections.
- The proposed algorithm avoids the curse of dimensionality.
- The algorithm is provenly consistent for a class of scheduling signals.
- An extension of the proposed approach with exogenous inputs signals is recently presented in LPVS 2019, Eindhoven, The Netherlands.

Thank you