An integral architecture for identification of continuous-time state-space LPV models

Manas Mejari, Bojan Mavkov, Marco Forgione, Dario Piga

LPVS 2021, Milan

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

Motivation

- Majority of physical systems are naturally modelled in Continuous-time (CT), parameters of CT models have physical interpretation.
- Direct identification of CT systems has multiple advantages [Garnier et. al. 2014]:
  - numerical robustness,
  - handling non-uniformly sampled data.
- Several direct CT identification methods developed for LTI model class.
- In this work, we develop direct CT identification for Linear Parameter-Varying state-space models.

Problem Formulation

- CT LPV Data generating system:

$$\dot{\mathbf{x}}(t) = \mathcal{A}(\mathbf{p}(t))\mathbf{x}(t) + \mathcal{B}(\mathbf{p}(t))\mathbf{u}(t)$$
$$\mathbf{x}(0) = \mathbf{x}_0$$
$$\mathbf{y}^{\text{o}}(t) = \mathcal{C}(\mathbf{p}(t))\mathbf{x}(t) + \mathcal{D}(\mathbf{p}(t))\mathbf{u}(t)$$

$$\underbrace{\mathcal{A}(\mathbf{p}(t)) = A_0 + \sum_{i=1}^{n_p} A_i \mathbf{p}_i(t)}_{\text{Affine LPV}}$$

Problem Formulation

- CT LPV Data generating system:

$$\dot{\mathbf{x}}(t) = \mathcal{A}(\mathbf{p}(t))\mathbf{x}(t) + \mathcal{B}(\mathbf{p}(t))\mathbf{u}(t)$$
$$\mathbf{x}(0) = \mathbf{x}_0$$
$$\mathbf{y}^{\mathrm{o}}(t) = \mathcal{C}(\mathbf{p}(t))\mathbf{x}(t) + \mathcal{D}(\mathbf{p}(t))\mathbf{u}(t)$$

$$\underbrace{\mathcal{A}(\mathbf{p}(t)) = A_0 + \sum_{i=1}^{n_p} A_i \mathbf{p}_i(t)}_{\text{Affine LPV}}$$

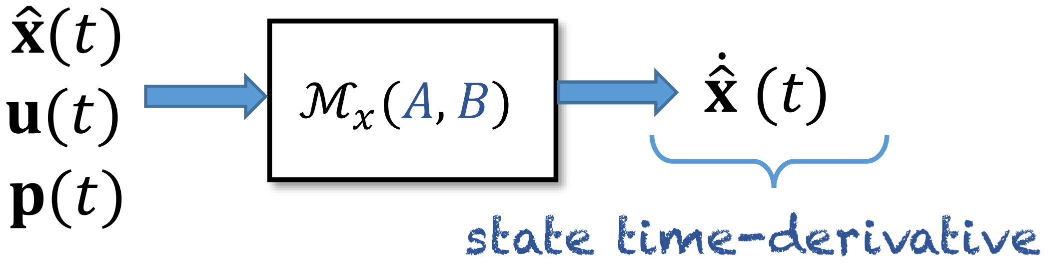
- **Objective:**

Given a training dataset $\{\mathbf{u}(t_k), \mathbf{p}(t_k), \mathbf{y}(t_k)\}_{k=0}^{N-1}$ taken at discrete time instances, identify a continuous-time LPV state-space affine model, such that output matches closely with $\mathbf{y}(t)$

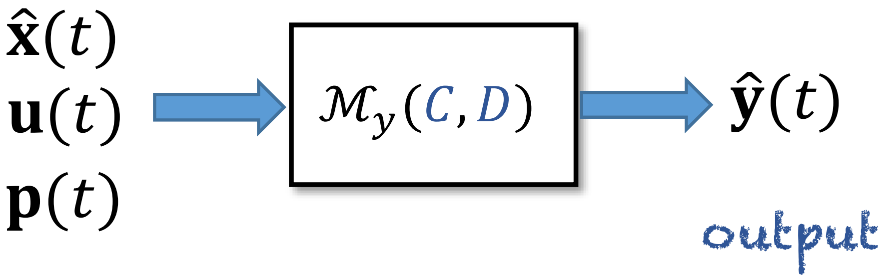
CT identification of LPV models

- We model the state and output maps with following LPV blocks

State map



Output map



$\mathcal{M}_x(\hat{A}, \hat{B})$:

$$\left(\hat{A}_0 + \sum_{i=1}^{n_p} \hat{A}_i \mathbf{p}_i(t)\right) \hat{\mathbf{x}}(t) + \left(\hat{B}_0 + \sum_{i=1}^{n_p} \hat{B}_i \mathbf{p}_i(t)\right) \mathbf{u}(t)$$

$\mathcal{M}_y(\hat{C}, \hat{D})$:

$$\left(\hat{C}_0 + \sum_{i=1}^{n_p} \hat{C}_i \mathbf{p}_i(t)\right) \hat{\mathbf{x}}(t) + \left(\hat{D}_0 + \sum_{i=1}^{n_p} \hat{D}_i \mathbf{p}_i(t)\right) \mathbf{u}(t)$$

CT identification of LPV models

- The resulting continuous-time LPV state-space model is given by

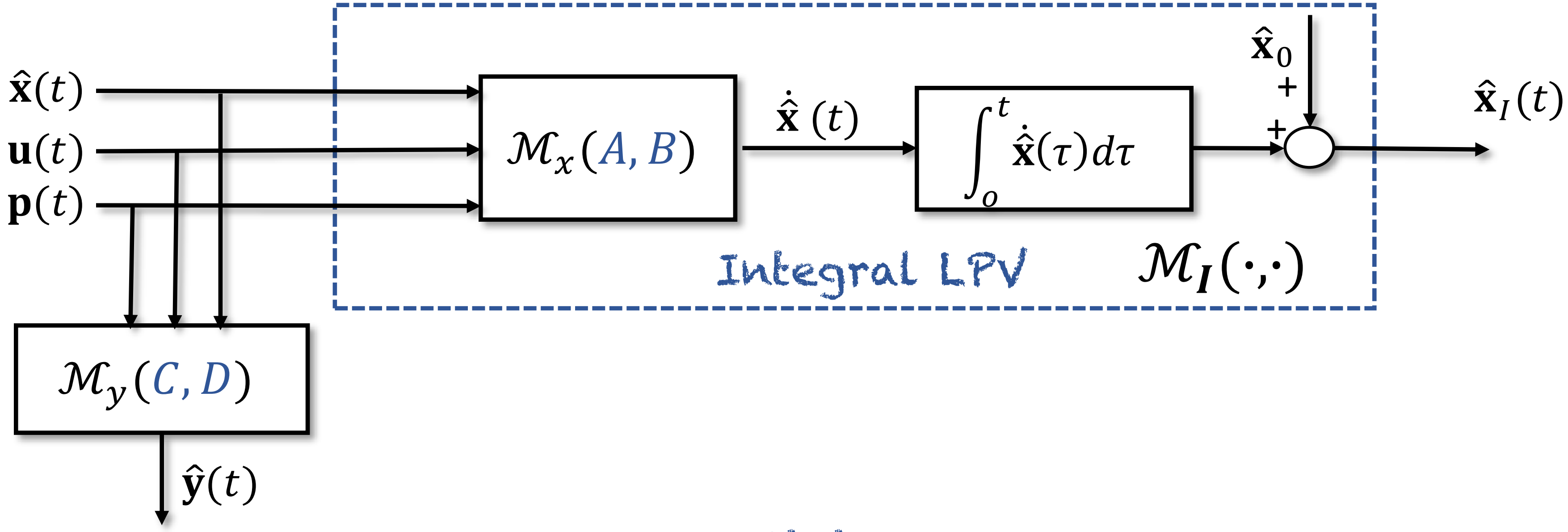
$$\begin{aligned} \dot{\hat{\mathbf{x}}}(t) &= \mathcal{M}_x(\hat{\mathbf{x}}(t), \mathbf{u}(t), \mathbf{p}(t); \hat{A}, \hat{B}) \\ \hat{\mathbf{x}}(0) &= \mathbf{x}_0 \\ \hat{\mathbf{y}}(t) &= \mathcal{M}_y(\hat{\mathbf{x}}(t), \mathbf{u}(t), \mathbf{p}(t); \hat{C}, \hat{D}) \end{aligned}$$

Affine in parameters

- In this work, we exploit the integral form of this Cauchy problem to identify the LPV state-space model.

Integral architecture

We define an integral LPV block

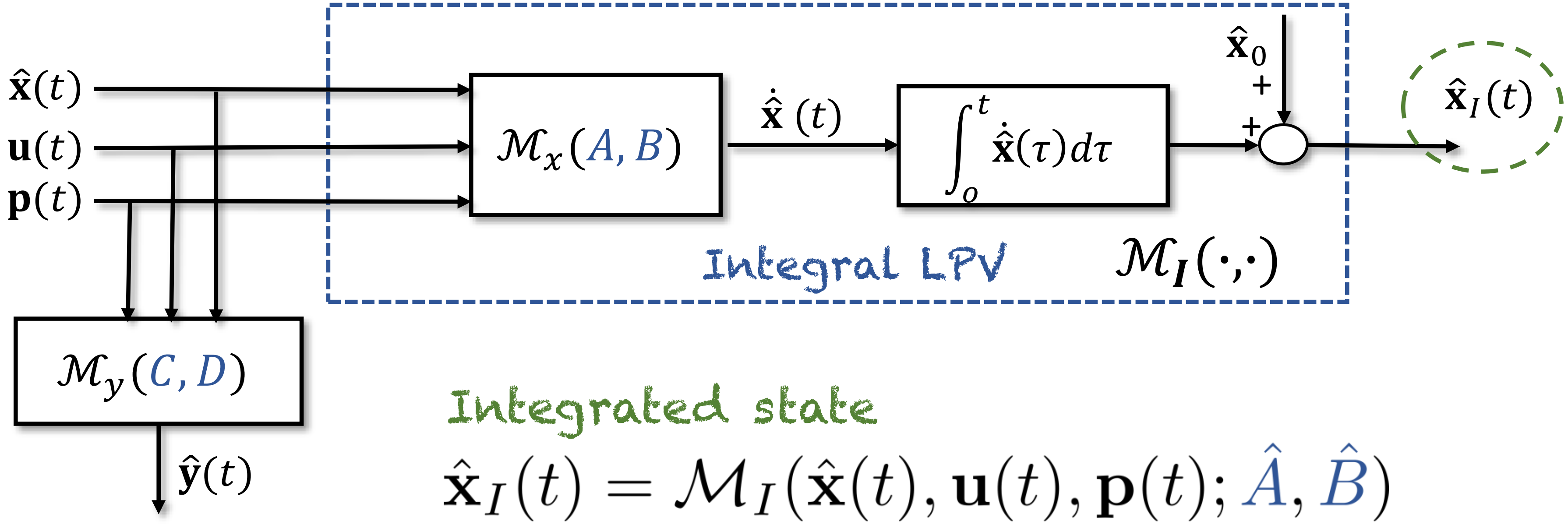


State map

$$\mathcal{M}_I(\cdot) = \hat{\mathbf{x}}(0) + \int_0^t \overbrace{\mathcal{M}_x(\hat{\mathbf{x}}(\tau), \mathbf{u}(\tau), \mathbf{p}(\tau); \hat{A}, \hat{B})}^{\text{State map}} d\tau.$$

Integral architecture

We define an integral LPV block

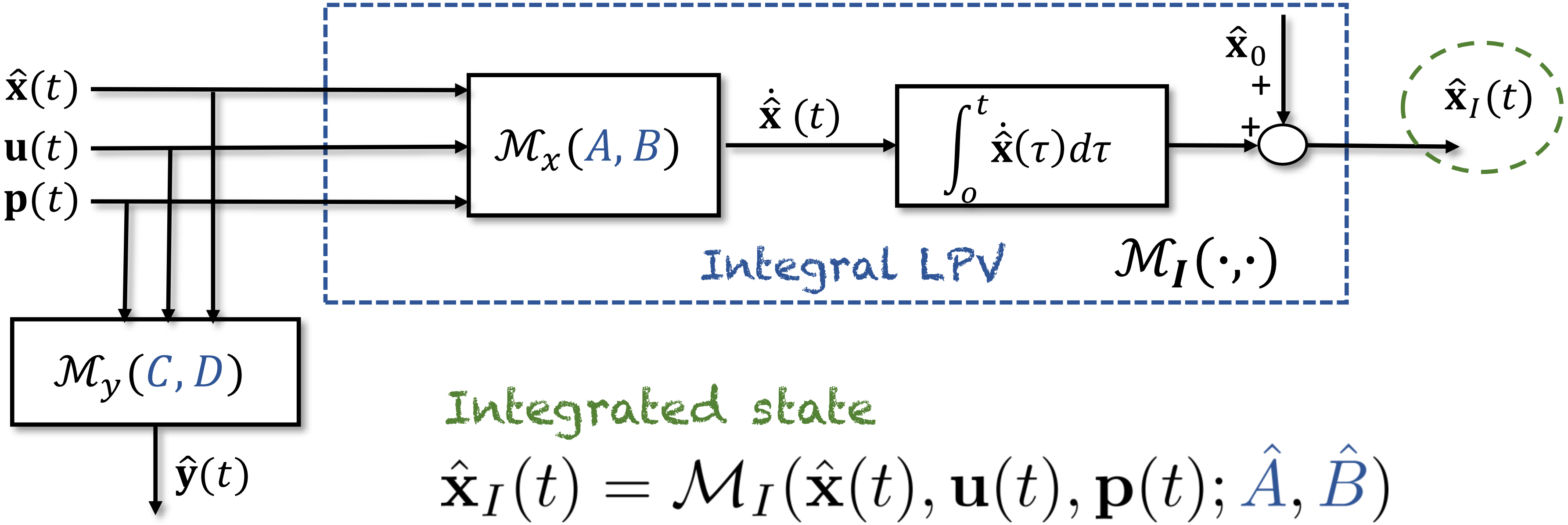


$$\hat{\mathbf{x}}_I(t) = \mathcal{M}_I(\hat{\mathbf{x}}(t), \mathbf{u}(t), \mathbf{p}(t); \hat{A}, \hat{B})$$

$$\mathcal{M}_I(\cdot) = \hat{\mathbf{x}}(0) + \int_0^t \mathcal{M}_x(\hat{\mathbf{x}}(\tau), \mathbf{u}(\tau), \mathbf{p}(\tau); \hat{A}, \hat{B})d\tau.$$

Integral architecture

We define an integral LPV block



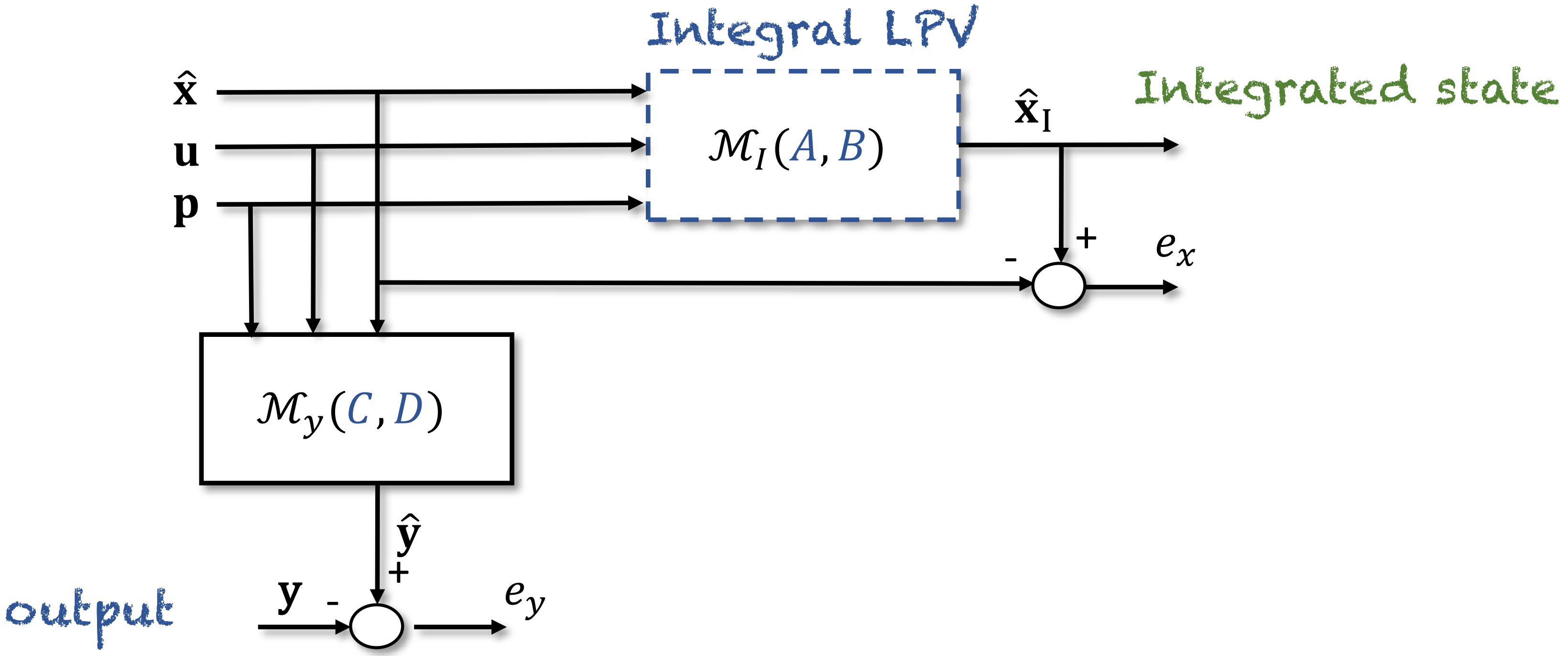
$$\hat{\mathbf{x}}_I(t) = \mathcal{M}_I(\hat{\mathbf{x}}(t), \mathbf{u}(t), \mathbf{p}(t); \hat{A}, \hat{B})$$

- If the states feeding the LPV block are actually generated by the model, integrated states exactly matches them, *i.e.*,

$$\hat{\mathbf{x}}(t) = \hat{\mathbf{x}}_I(t) \qquad \forall t \in [t_0 \; t_{N-1}].$$

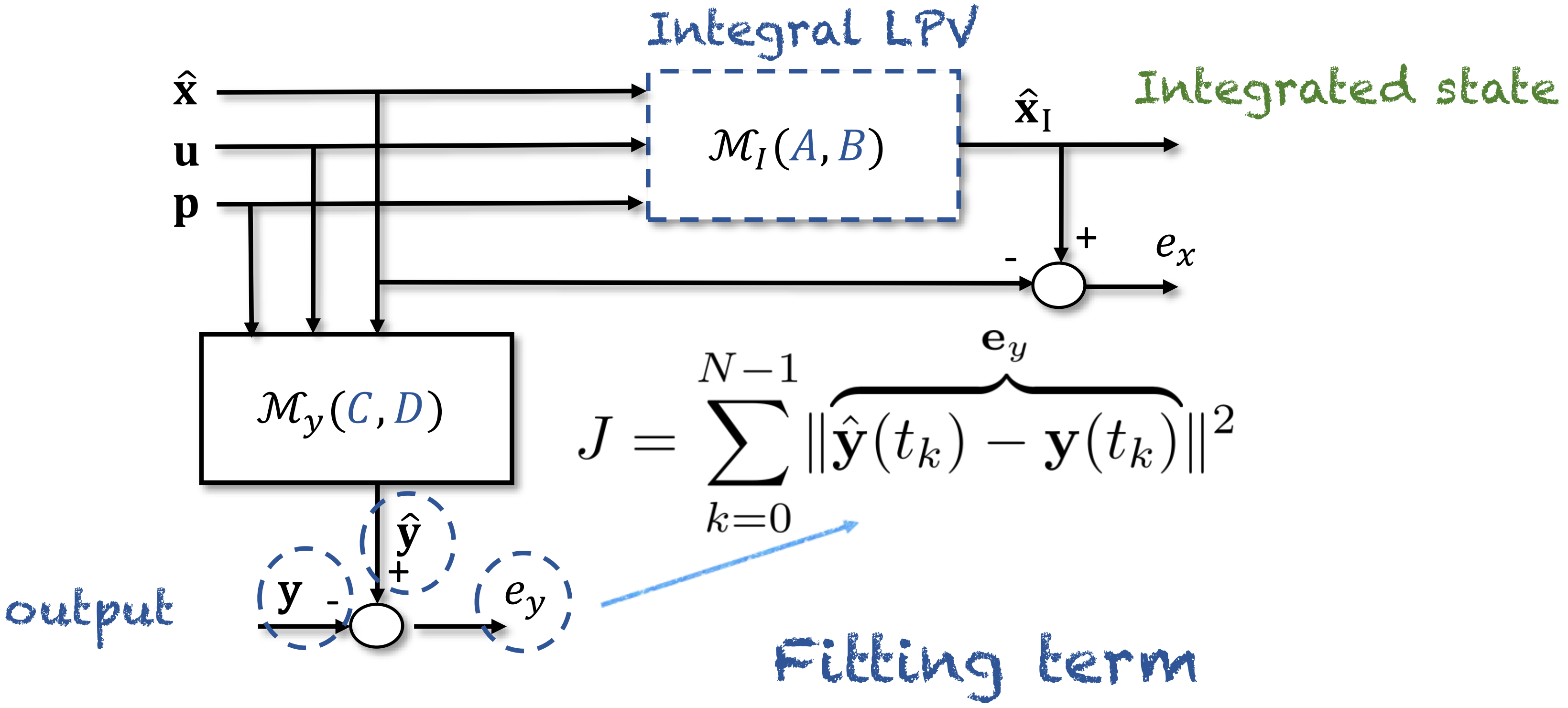
Fitting Criterion

The model matrices and states are jointly optimized according to a dual objective



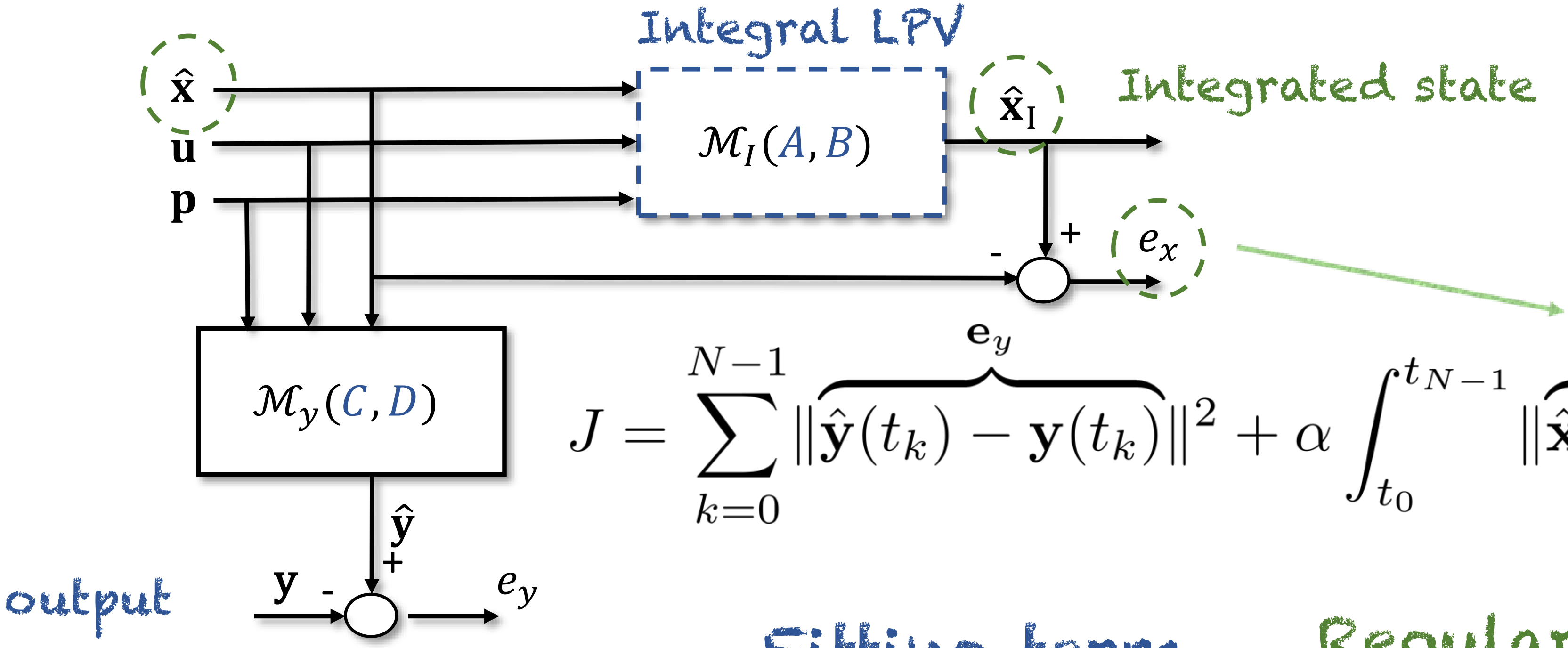
Fitting Criterion

The model matrices and states are jointly optimized according to a dual objective



Fitting Criterion

The model matrices and states are jointly optimized according to a dual objective



$$J = \sum_{k=0}^{N-1} \| \overbrace{\hat{\mathbf{y}}(t_k) - \mathbf{y}(t_k)}^{\mathbf{e}_y} \|^2 + \alpha \int_{t_0}^{t_{N-1}} \| \overbrace{\hat{\mathbf{x}}_I(\tau) - \hat{\mathbf{x}}(\tau)}^{\mathbf{e}_x} \|^2 \, d\tau,$$

Fitting term

Regularization term

Optimization problem

- The model matrices and states are jointly optimized by minimizing

$$\min_{\hat{\mathbf{x}}(\cdot),\hat{A},\hat{B},\hat{C},\hat{D}} J(\hat{\mathbf{x}}(\cdot),\hat{A},\hat{B},\hat{C},\hat{D})$$

$$J = \sum_{k=0}^{N-1} \|\overbrace{\hat{\mathbf{y}}(t_k) - \mathbf{y}(t_k)}^{\mathbf{e}_y}\|^2 + \alpha \int_{t_0}^{t_{N-1}} \|\overbrace{\hat{\mathbf{x}}_I(\tau) - \hat{\mathbf{x}}(\tau)}^{\mathbf{e}_x}\|^2 \, d\tau,$$

- continuous-time state signal is one of the decision variables!
- the optimization problem is infinite-dimensional and intractable.

Optimization problem

- The model matrices and states are jointly optimized by minimizing

$$\min_{\hat{\mathbf{x}}(\cdot),\hat{A},\hat{B},\hat{C},\hat{D}} J(\hat{\mathbf{x}}(\cdot),\hat{A},\hat{B},\hat{C},\hat{D})$$

$$J = \sum_{k=0}^{N-1} \|\overbrace{\hat{\mathbf{y}}(t_k) - \mathbf{y}(t_k)}^{\mathbf{e}_y}\|^2 + \alpha \int_{t_0}^{t_{N-1}} \|\overbrace{\hat{\mathbf{x}}_I(\tau) - \hat{\mathbf{x}}(\tau)}^{\mathbf{e}_x}\|^2 \, d\tau,$$

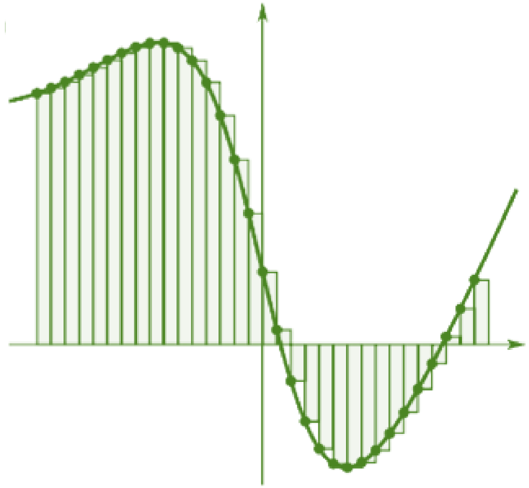
- continuous-time state signal is one of the decision variables!
- the optimization problem is infinite-dimensional and intractable.
- We employ numerical techniques to transform it into finite-dimensional problem.

Numerical techniques

- Signals are approximated using finite-dimensional parameterization, e.g., piecewise constant

- Integrals are approximated using rectangular quadrature.

$$\int_{t_0}^{t_{N-1}} \|\hat{\mathbf{x}}_I(\tau) - \hat{\mathbf{x}}(\tau)\|^2 d\tau \approx \sum_{k=1}^{N-1} \|\hat{\mathbf{x}}_I(t_k) - \hat{\mathbf{x}}(t_k)\|^2 \Delta t_k$$



- State is approximated with the Riemann sum:

$$\hat{\mathbf{x}}_I(t_k) = \hat{\mathbf{x}}(0) + \int_0^t \mathcal{M}_x(\hat{\mathbf{x}}(\tau), \mathbf{u}(\tau), \mathbf{p}(\tau);\ \hat{A}, \hat{B})\, d\tau.$$

$$\approx \hat{\mathbf{x}}(0) + \sum_{j=0}^{k-1} \Delta t_{j+1} \mathcal{M}_x(\hat{\mathbf{x}}(t_j), \mathbf{u}(t_j), \mathbf{p}(t_j);\ \hat{A}, \hat{B}).$$

.

Numerical techniques

- Signals are approximated using finite-dimensional parameterization, e.g., piecewise constant

- Integrals are approximated using rectangular quadrature.

$$\int_{t_0}^{t_{N-1}} \|\hat{\mathbf{x}}_I(\tau) - \hat{\mathbf{x}}(\tau)\|^2 d\tau \approx \sum_{k=1}^{N-1} \|\hat{\mathbf{x}}_I(t_k) - \hat{\mathbf{x}}(t_k)\|^2 \Delta t_k$$

- In general, more complex parameterizations piecewise polynomials with other quadrature rules, trapezoidal or Gaussian can be used.

Optimization algorithm

We employ coordinate descent algorithm to solve the problem

$$\min_{\{\hat{\mathbf{x}}(t_k)\}_{k=0}^{N-1}, \hat{A}, \hat{B}, \hat{C}, \hat{D}} J(\hat{\mathbf{x}}, \hat{A}, \hat{B}, \hat{C}, \hat{D})$$

1. **Iterate for** $n = 1, \ldots$

   1.1. $\hat{A}^{(n)}, \hat{B}^{(n)}, \hat{C}^{(n)}, \hat{D}^{(n)} \leftarrow \arg \min_{\hat{A}, \hat{B}, \hat{C}, \hat{D}} J(\hat{\mathbf{x}}^{(n-1)}, \hat{A}, \hat{B}, \hat{C}, \hat{D})$

   1.2. $\hat{\mathbf{x}}^{(n)} \leftarrow \operatorname{argmin}_{\hat{\mathbf{x}}} J(\hat{\mathbf{x}}, \hat{A}^{(n)}, \hat{B}^{(n)}, \hat{C}^{(n)}, \hat{D}^{(n)})$

Optimization algorithm

We employ coordinate descent algorithm to solve the problem

$$\min_{\{\hat{\mathbf{x}}(t_k)\}_{k=0}^{N-1}, \hat{A}, \hat{B}, \hat{C}, \hat{D}} J(\hat{\mathbf{x}}, \hat{A}, \hat{B}, \hat{C}, \hat{D})$$

1. **Iterate for** $n = 1, \ldots$

Least Squares

1.1. $\hat{A}^{(n)}, \hat{B}^{(n)}, \hat{C}^{(n)}, \hat{D}^{(n)} \leftarrow \arg \min_{\hat{A}, \hat{B}, \hat{C}, \hat{D}} J(\hat{\mathbf{x}}^{(n-1)}, \hat{A}, \hat{B}, \hat{C}, \hat{D})$

1.2. $\hat{\mathbf{x}}^{(n)} \leftarrow \operatorname{argmin}_{\hat{\mathbf{x}}} J(\hat{\mathbf{x}}, \hat{A}^{(n)}, \hat{B}^{(n)}, \hat{C}^{(n)}, \hat{D}^{(n)})$

The solutions at steps 1.1 and 1.2 can be computed analytically

Numerical example

- We consider MIMO CT LPV data-generating system:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathcal{A}(\mathbf{p}(t))\mathbf{x}(t) + \mathcal{B}(\mathbf{p}(t))\mathbf{u}(t)\\ \mathbf{x}(0) &= \mathbf{x}_0\\ \mathbf{y}^{\mathrm{o}}(t) &= \mathcal{C}(\mathbf{p}(t))\mathbf{x}(t) + \mathcal{D}(\mathbf{p}(t))\mathbf{u}(t)\end{aligned}$$

$$\begin{bmatrix} A_0 & A_1 & A_2 \end{bmatrix} = \left[\begin{array}{cc|cc|cc} 0 & 0.5 & 0.4 & 0 & 0.2 & 0 \\ -0.5 & 0 & 0 & 0.3 & 0 & 0 \end{array}\right],$$

$$B_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, C_0 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, D_0 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}.$$

- Outputs are corrupted by white Gaussian noise,
  - Signal-to-noise ratio (SNR) of {15, 25} dB are considered.

Numerical example

- We consider MIMO CT LPV data-generating system:

$$\dot{\mathbf{x}}(t) = \mathcal{A}(\mathbf{p}(t))\mathbf{x}(t) + \mathcal{B}(\mathbf{p}(t))\mathbf{u}(t)$$
$$\mathbf{x}(0) = \mathbf{x}_0$$
$$\mathbf{y}^{\mathrm{o}}(t) = \mathcal{C}(\mathbf{p}(t))\mathbf{x}(t) + \mathcal{D}(\mathbf{p}(t))\mathbf{u}(t)$$

$$\left[\begin{array}{ccc} A_0 & A_1 & A_2 \end{array}\right] = \left[\begin{array}{cc|cc|cc} 0 & 0.5 & 0.4 & 0 & 0.2 & 0 \\ -0.5 & 0 & 0 & 0.3 & 0 & 0 \end{array}\right],$$

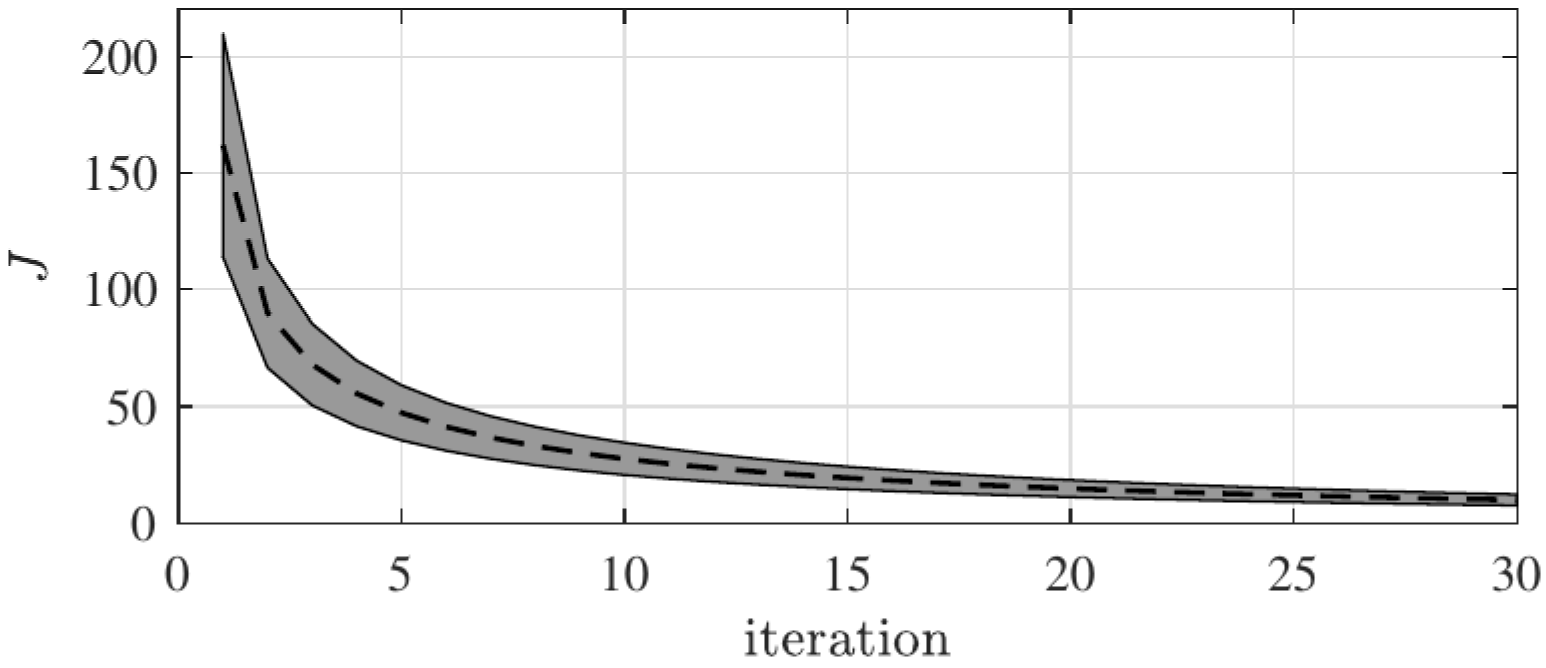
$$B_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, C_0 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, D_0 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}.$$

- Outputs are corrupted by white Gaussian noise,
  - Signal-to-noise ratio (SNR) of {15, 25} dB are considered.
- Monte-Carlo (MC) study with 50 MC runs are performed.
  - Training dataset: 800 samples
  - Validation dataset:500 samples is gathered.

Numerical example

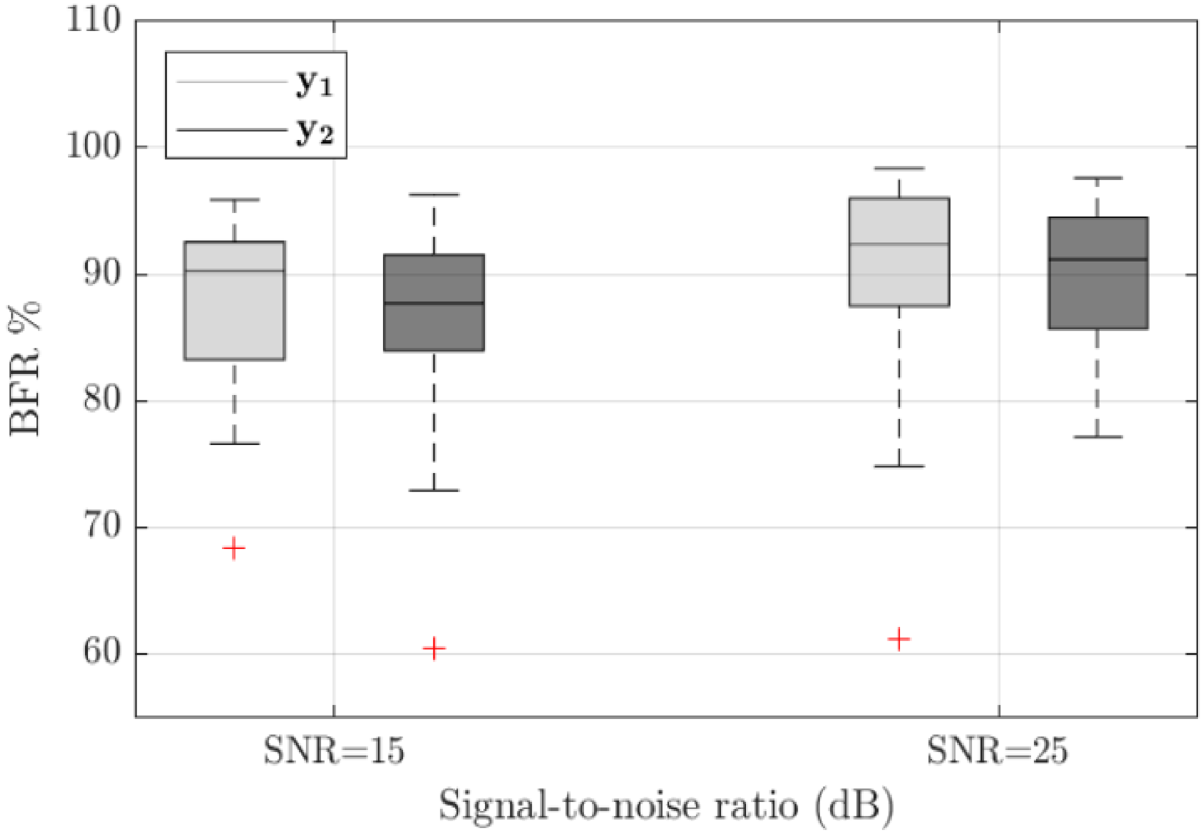
- We run the proposed algorithm for 30 iterations with a random initial guess for states.

Cost function vs iterations

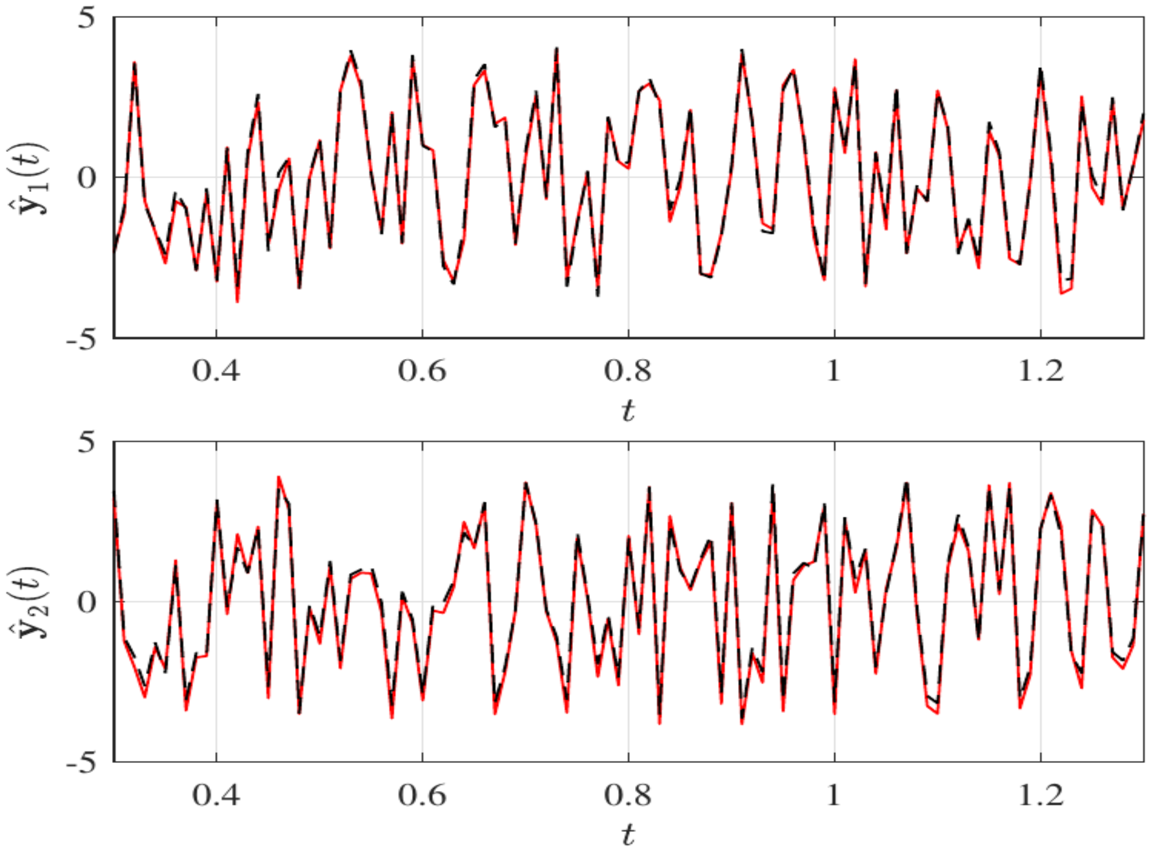


Numerical example

Best Fit Rate over 50 MC runs



True vs estimated output



Conclusions

- We have presented an integral architecture for continuous-time identification of LPV state-space models.
- A coordinate-descent algorithm exploiting linear-parametric structure is presented.
- The algorithm is computationally efficient as the solutions to the sub-problems can be obtained in closed-form.
- Future work will be focused on extending the proposed algorithm to other model classes.