Bayesian nonparametric identification of piecewise affine ARX systems

Johan Wågberg\textsuperscript{\textregistered}, Fredrik Lindsten\textsuperscript{\textregistered\textdagger}, Thomas Schö

\textsuperscript{\textregistered}Department of Information Technology, Uppsala University, Sweden
\textsuperscript{\textregistered\textdagger}Department of Engineering, University of Cambridge, UK

johan.wagberg@it.uu.se
Bayesian identification of ARX

\[ y_t = \theta^T \phi_t + e_t \]

\[ \phi_t = [x_t^T, y_{t-1}, \ldots, y_{t-n_a}, u_{t-1}, \ldots, u_{t-n_b}]^T \]

\[ e_t \sim N(0, \Lambda) \]

Unknown parameters \( \theta \) and \( \Lambda \).

Bayesian identification \( \Rightarrow \) need prior on \( \theta \) and \( \Lambda \).
Bayesian identification of ARX

ARX \( (n_a, n_b) \) system

\[
y_t = \vartheta^T \varphi_t + e_t
\]

\[
\varphi_t = [x_t^T \quad 1]^T
\]

\[
x_t = [y_{t-1}^T \quad \cdots \quad y_{t-n_a}^T \quad u_t^T \quad \cdots \quad u_{t-n_b}^T]^T
\]

\[
e_t \sim \mathcal{N}(0, \Lambda)
\]
Bayesian identification of ARX

ARX \((n_a, n_b)\) system

\[ y_t = \vartheta^T \varphi_t + e_t \]
\[ \varphi_t = [x_t^T \ 1]^T \]
\[ x_t = [y_{t-1}^T \ldots y_{t-n_a}^T \ u_{t}^T \ldots u_{t-n_b}^T]^T \]
\[ e_t \sim \mathcal{N}(0, \Lambda) \]

Unknown parameters \(\vartheta\) and \(\Lambda\).
Bayesian identification of ARX

ARX \((n_a, n_b)\) system

\[
y_t = \vartheta^T \varphi_t + e_t
\]

\[
\varphi_t = [x_t^T \ 1]^T
\]

\[
x_t = [y_{t-1}^T \ \cdots \ y_{t-n_a}^T \ u_t^T \ \cdots \ u_{t-n_b}^T]^T
\]

\[
e_t \sim \mathcal{N}(0, \Lambda)
\]

Unknown parameters \(\vartheta\) and \(\Lambda\).
Bayesian identification \(\Rightarrow\) need prior on \(\vartheta\) and \(\Lambda\).
Prior on $\vartheta$ and $\Lambda$

Assume $[x^T t, y^T t]^T$ to be jointly Gaussian

$[x^T t, y^T t]^T \sim \mathcal{N}(\mu, \Sigma) = \mathcal{N}(\begin{bmatrix} \mu^x \\ \mu^y \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix})$

then $y^t | x^t \sim \mathcal{N}(\mu^y | x, \Sigma^y | x)$, where

$\mu^y | x = \mu^y + \Sigma_{yx} \Sigma^{-1}_{xx} (x^t - \mu^x)$

$\Sigma^y | x = \Sigma_{yy} - \Sigma_{yx} \Sigma^{-1}_{xx} \Sigma_{xy}$

Identify $\vartheta = \begin{bmatrix} \Sigma_{yx} \\ \Sigma_{yx} \Sigma^{-1}_{xx} \mu^y - \Sigma_{yx} \Sigma^{-1}_{xx} \mu^x \end{bmatrix}$,

$\Lambda = \Sigma^y | x \Rightarrow y^t = \vartheta^T [x^T 1]^T + e^t = \vartheta^T \phi^t + e^t 
\sim \mathcal{N}(0, \Lambda)$
Prior on $\vartheta$ and $\Lambda$

Assume $[x_t^T, y_t^T]^T$ to be jointly Gaussian

$$
\begin{bmatrix}
  x_t \\
  y_t
\end{bmatrix} \sim \mathcal{N}(\mu, \Sigma) = \mathcal{N}
\left(
\begin{bmatrix}
  \mu_x \\
  \mu_y
\end{bmatrix},
\begin{bmatrix}
  \Sigma_{xx} & \Sigma_{xy} \\
  \Sigma_{yx} & \Sigma_{yy}
\end{bmatrix}
\right)
$$

Identify

$$
\vartheta = \begin{bmatrix}
  \Sigma_{yx} \\
  \Sigma_{yy}
\end{bmatrix},
\Lambda = \Sigma_{y|X} = \Rightarrow
$$

$$
y_t = \vartheta^T [x_t 1]^T + e_t = \vartheta^T \phi_t + e_t 
$$

$e_t \sim \mathcal{N}(0, \Lambda)$
Prior on $\vartheta$ and $\Lambda$

Assume $[x_t^T, y_t^T]^T$ to be jointly Gaussian

$$
\begin{bmatrix}
  x_t \\
  y_t 
\end{bmatrix}
\sim \mathcal{N}(\mu, \Sigma) = \mathcal{N}
\begin{pmatrix}
  \mu_x \\
  \mu_y \\
  \Sigma_{xx} & \Sigma_{xy} \\
  \Sigma_{yx} & \Sigma_{yy}
\end{pmatrix}
$$

then

$$
y_t \mid x_t \sim \mathcal{N}(\mu_{y|x}, \Sigma_{y|x}), \text{ where }
\begin{align*}
  \mu_{y|x} &= \mu_y + \Sigma_{yx} \Sigma_{xx}^{-1} (x_t - \mu_x) \\
  \Sigma_{y|x} &= \Sigma_{yy} - \Sigma_{yx} \Sigma_{xx}^{-1} \Sigma_{xy}
\end{align*}
$$
Prior on $\vartheta$ and $\Lambda$

Assume $[x_t^T, y_t^T]^T$ to be jointly Gaussian

$$[x_t, y_t] \sim \mathcal{N}(\mu, \Sigma) = \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}\right)$$

then

$$y_t \mid x_t \sim \mathcal{N}(\mu_{y|x}, \Sigma_{y|x}), \text{ where}$$

$$\mu_{y|x} = \mu_y + \Sigma_{yx} \Sigma_{xx}^{-1} (x_t - \mu_x)$$

$$\Sigma_{y|x} = \Sigma_{yy} - \Sigma_{yx} \Sigma_{xx}^{-1} \Sigma_{xy}$$

Identify

$$\vartheta = \begin{bmatrix} \Sigma_{yx} \Sigma_{xx}^{-1} \\ \mu_y - \Sigma_{yx} \Sigma_{xx}^{-1} \mu_x \end{bmatrix}$$

$$\Lambda = \Sigma_{y|x}$$

$$y_t = \vartheta^T \begin{bmatrix} x_t \\ 1 \end{bmatrix} + e_t = \vartheta^T \varphi_t + e_t$$

$$e_t \sim \mathcal{N}(0, \Lambda)$$

Bayesian nonparametric identification of PWARX systems
Prior on $\vartheta$ and $\Lambda$

A prior on $\vartheta$ and $\Lambda$ is directly given from a prior on $\mu$ and $\Sigma$. To ease computations, we use a conjugate prior. In this case: Normal inverse Wishart ($\mathcal{NIW}$).

Our generative model now looks like

$$(\mu, \Sigma) \sim \mathcal{NIW}(\mu_0, \lambda, \Psi, \nu)$$
$$(x_t, y_t) | \mu, \Sigma \sim \mathcal{N}(\mu, \Sigma)$$

Conjugate prior $\Rightarrow$ exist analytic expressions for $\mu_n, \lambda_n, \Psi_n, \nu_n$ such that

$$(\mu, \Sigma) | x_{1:n}, y_{1:n} \sim \mathcal{NIW}(\mu_n, \lambda_n, \Psi_n, \nu_n)$$
Bayesian identification of ARX

Example

Generate 100 samples from posterior distribution of $(\mu, \Sigma)$ given $n$ observations $(x_1:n, y_1:n)$ from the ARX system

$$y_t = u_t - 1 + 2 + e_t.$$
Bayesian identification of ARX

Example

Generate 100 samples from posterior distribution of \((\mu, \Sigma)\) given \(n\) observations \((x_{1:n}, y_{1:n})\) from the ARX system

\[ y_t = 1u_{t-1} + 2 + e_t. \]
Bayesian identification of ARX

Example

Generate 100 samples from posterior distribution of \((\mu, \Sigma)\) given \(n\) observations \((x_{1:n}, y_{1:n})\) from the ARX system

\[ y_t = 1u_{t-1} + 2 + e_t. \]
Bayesian identification of ARX

Example

Generate 100 samples from posterior distribution of \((\mu, \Sigma)\) given \(n\) observations \((x_{1:n}, y_{1:n})\) from the ARX system

\[
y_t = 1 u_{t-1} + 2 + e_t.
\]
Bayesian identification of ARX

Example

Generate 100 samples from posterior distribution of \((\mu, \Sigma)\) given \(n\) observations \((x_{1:n}, y_{1:n})\) from the ARX system

\[
y_t = 1 u_{t-1} + 2 + e_t.
\]
Bayesian identification of ARX

Example

Generate 100 samples from posterior distribution of \((\mu, \Sigma)\) given \(n\) observations \((x_{1:n}, y_{1:n})\) from the ARX system

\[
y_t = 1u_{t-1} + 2 + e_t.
\]
Bayesian identification of ARX

Example

Generate 100 samples from posterior distribution of $(\mu, \Sigma)$ given $n$ observations $(x_{1:n}, y_{1:n})$ from the ARX system

$$y_t = 1u_{t-1} + 2 + e_t.$$
Bayesian identification of ARX

Example

Generate 100 samples from posterior distribution of $(\mu, \Sigma)$ given $n$ observations $(x_1:n, y_1:n)$ from the ARX system

$$y_t = 1 u_{t-1} + 2 + e_t.$$
Bayesian identification of ARX

Example

Generate 100 samples from posterior distribution of \((\mu, \Sigma)\) given \(n\) observations \((x_{1:n}, y_{1:n})\) from the ARX system

\[ y_t = 1u_{t-1} + 2 + e_t. \]
Piecewise affine ARX

\( y_t = \begin{cases} 
\vartheta_1 + \phi_t + e_t & \text{if } x_t \in R_1, \\
\vartheta_K + \phi_t + e_t & \text{if } x_t \in R_K,
\end{cases} \)

where \( \{R_k\}_{k=1}^K \) partition of the regressor space.
Piecewise affine ARX

This was for a standard ARX system, but the title promises more.
Piecewise affine ARX

This was for a standard ARX system, but the title promises more. Piecewise affine ARX \((n_a, n_b)\) model with \(K\) modes

\[
y_t = \begin{cases} 
\vartheta_1^T \varphi_t + e_t & \text{if } x_t \in \mathcal{R}_1, \\
\vdots & \\
\vartheta_K^T \varphi_t + e_t & \text{if } x_t \in \mathcal{R}_K,
\end{cases}
\]

where \(\{\mathcal{R}_k\}_{k=1}^K\) partition of the regressor space.
Piecewise affine ARX

To model the partition, we introduce cluster labels $z_1: N$ such that $z_t = k \iff x_t \in R_k$.

$z_1: N$ unknown $\Rightarrow$ need to assign a prior.

Call this prior $\pi(\alpha)$

Our model of a PW-ARX system $z_1: N \sim \pi(\alpha)$

$(\mu_k, \Sigma_k) \text{i.i.d.} \sim \text{NIW}(\mu_0, \lambda, \Psi, \nu), k = 1, \ldots, K$

$(x_t, y_t) | z_t, \mu_{z_t}, \Sigma_{z_t} \sim N(\mu_{z_t}, \Sigma_{z_t})$
Piecewise affine ARX

To model the partition, we introduce cluster labels $z_{1:N}$ such that

$$z_t = k \iff x_t \in \mathcal{R}_k.$$
To model the partition, we introduce cluster labels $z_{1:N}$ such that

$$z_t = k \iff x_t \in \mathcal{R}_k.$$ 

$z_{1:N}$ unknown $\Rightarrow$ need to assign a prior
Piecewise affine ARX

To model the partition, we introduce cluster labels $z_{1:N}$ such that

$$z_t = k \iff x_t \in \mathcal{R}_k.$$ 

$z_{1:N}$ unknown $\implies$ need to assign a prior

Call this prior $\pi$ with parameter $\alpha$
Piecewise affine ARX

To model the partition, we introduce cluster labels $z_{1:N}$ such that

$$z_t = k \iff x_t \in \mathcal{R}_k.$$ 

$z_{1:N}$ unknown $\Rightarrow$ need to assign a prior

Call this prior $\pi$ with parameter $\alpha$

Our model of a ARX system

$$\begin{align*}
    (\mu, \Sigma) &\sim NIW(\mu_0, \lambda, \Psi, \nu), \\
    (x_t, y_t) \mid (\mu, \Sigma) &\sim \mathcal{N}(\mu, \Sigma)
\end{align*}$$
Piecewise affine ARX

To model the partition, we introduce cluster labels $z_{1:N}$ such that

$$z_t = k \iff x_t \in R_k.$$ 

$z_{1:N}$ unknown $\Rightarrow$ need to assign a prior
Call this prior $\pi$ with parameter $\alpha$
Our model of a PWARX system

$$z_{1:N} \sim \pi(\alpha)$$

$$(\mu_k, \Sigma_k) \sim \text{NIW}(\mu_0, \lambda, \Psi, \nu), \quad k = 1, \ldots, K$$

$$(x_t, y_t) \mid z_t, \mu_{z_t}, \Sigma_{z_t} \sim \mathcal{N}(\mu_{z_t}, \Sigma_{z_t})$$
Prior $\pi$ on $z_{1:N}$
The prior $\pi$ should have no upper bound on the number of clusters $K$. 
The prior $\pi$ should have no upper bound on the number of clusters $K$.

Let the prior $\pi(\alpha)$ on $z_{1:N}$ be

$$
\pi(z_{n+1} = k \mid z_{1:n}) = \left\{ \begin{array}{ll}
\frac{n_k}{n+\alpha} & k = 1, \ldots, K \quad \leftarrow \text{Join existing cluster} \\
\frac{\alpha}{n+\alpha} & k = K + 1 \quad \leftarrow \text{Create new cluster}
\end{array} \right.
$$

$$
n_k = \sum_{t=1}^{n} \mathbb{I}(z_t = k) \quad \leftarrow \text{Number of members in cluster } k
$$
Prior $\pi$ on $z_{1:N}$

The prior $\pi$ should have no upper bound on the number of clusters $K$.
Let the prior $\pi(\alpha)$ on $z_{1:N}$ be

$$
\pi(z_{n+1} = k \mid z_{1:n}) = \begin{cases} 
\frac{n_k}{n+\alpha} & k = 1, \ldots, K \quad \leftarrow \text{Join existing cluster} \\
\frac{\alpha}{n+\alpha} & k = K + 1 \quad \leftarrow \text{Create new cluster} 
\end{cases}
$$

$$
n_k = \sum_{t=1}^{n} \mathbb{I}(z_t = k) \quad \leftarrow \text{Number of members in cluster } k
$$

The number of clusters and hence also the number of parameters grows with data, which makes this a nonparametric model.
Inference

Inference

We are interested in the posterior distribution
\[ p(y|x, x_1:N, y_1:N) \]
We can write
\[ p(y|x, x_1:N, y_1:N) = \sum_{z_1:N} p(y, z_1:N|x, x_1:N, y_1:N) \]
This is intractable but can be approximated if we have
\[ M \]
independent samples of
\[ p(y|x, x_1:N, y_1:N) \approx \frac{1}{M} \sum_{m=1}^{M} p(y|x, z_1:N[m], x_1:N, y_1:N) \].
Inference

We are interested in the posterior distribution

\[ p(y \mid x, x_{1:N}, y_{1:N}) . \]
Inference

We are interested in the posterior distribution

\[ p(y \mid x, x_{1:N}, y_{1:N}). \]

We can write

\[
p(y \mid x, x_{1:N}, y_{1:N}) = \sum_{z_{1:N}} p(y, z_{1:N} \mid x, x_{1:N}, y_{1:N})
\]

\[
= \mathbb{E}_{z_{1:N} \mid x_{1:N}, y_{1:N}} [p(y \mid x, z_{1:N}, x_{1:N}, y_{1:N})]
\]
Inference

We are interested in the posterior distribution

\[ p(y \mid x, x_{1:N}, y_{1:N}). \]

We can write

\[ p(y \mid x, x_{1:N}, y_{1:N}) = \sum_{z_{1:N}} p(y, z_{1:N} \mid x, x_{1:N}, y_{1:N}) \]

\[ = \mathbb{E}_{z_{1:N} \mid x_{1:N}, y_{1:N}} [p(y \mid x, z_{1:N}, x_{1:N}, y_{1:N})] \]

This is intractable but can be approximated if we have \( M \) independent samples of \( z_{1:N} \mid x_{1:N}, y_{1:N} \)

\[ p(y \mid x, x_{1:N}, y_{1:N}) \approx \frac{1}{M} \sum_{m=1}^{M} p(y \mid x, z_{1:N}[m], x_{1:N}, y_{1:N}). \]
Inference

Independent samples is not necessary. Samples from an ergodic Markov chain is enough. Ergodic means it is possible to go from any state to any other state. One method to construct such a Markov chain is called Gibbs sampling, where each variable is sampled conditioned on all other.

Algorithm 1

Gibbs sampler for $p(z_1:N|x_1:N, y_1:N)$

Require:

Starting state $z_1:n[1]$

1: for $m = 2$ to $M$

2: for $t = 1$ to $N$

3: Sample $z_t[m] \sim p(z_t|z_1:t-1[m], z_t+1:n[m-1], x_1:N, y_1:N)$

4: end for

5: end for
Inference

Independent samples is not necessary. Samples from an ergodic Markov chain is enough. Ergodic means it is possible to go from any state to any other state. One method to construct such a Markov chain is called Gibbs sampling, where each variable is sampled conditioned of all other.
Inference

Independent samples is not necessary. Samples from an ergodic Markov chain is enough. Ergodic means it is possible to go from any state to any other state. One method to construct such a Markov chain is called Gibbs sampling, where each variable is sampled conditioned of all other.

Algorithm 1 Gibbs sampler for $p(z_{1:N} \mid x_{1:N}, y_{1:N})$

**Require:** Starting state $z_{1:n}[1]$
1: for $m = 2$ to $M$ do
2: for $t = 1$ to $N$ do
3: Sample $z_t[m] \sim p(z_t \mid z_{1:t-1}[m], z_{t+1:n}[m-1], x_{1:N}, y_{1:N})$
4: end for
5: end for
Let's revisit our previous ARX system, but change it into a PWARX system, make 50 observations and run the Gibbs sampler to collect 1000 samples.
Illustration of Gibbs sampling

Let’s revisit our previous ARX system,

\[ y_t = \begin{cases} \mu + \theta u_{t-1} & \text{if } y_{t-1} \leq 0 \\ -\mu + \theta u_{t-1} & \text{otherwise} \end{cases} \]

\[ u_t = u_{t-1} + \varepsilon_t \]

where \( \mu \) is the mean, \( \theta \) is the gain, and \( \varepsilon_t \) is the noise.
Let’s revisit our previous ARX system, but change it into a PWARX system,
Illustration of Gibbs sampling

Let’s revisit our previous ARX system, but change it into a PWARX system, make 50 observations
Illustration of Gibbs sampling

Let’s revisit our previous ARX system, but change it into a PWARX system, make 50 observations and run the Gibbs sampler to collect 1000 samples.
Let’s revisit our previous ARX system, but change it into a PWARX system, make 50 observations and run the Gibbs sampler to collect 1000 samples.
Illustration of Gibbs sampling

Let’s revisit our previous ARX system, but change it into a PWARX system, make 50 observations and run the Gibbs sampler to collect 1000 samples.
Illustration of Gibbs sampling

Let’s revisit our previous ARX system, but change it into a PWARX system, make 50 observations and run the Gibbs sampler to collect 1000 samples.
Illustration of Gibbs sampling

Let’s revisit our previous ARX system, but change it into a PWARX system, make 50 observations and run the Gibbs sampler to collect 1000 samples.
Let’s revisit our previous ARX system, but change it into a PWARX system, make 50 observations and run the Gibbs sampler to collect 1000 samples.
Illustration of Gibbs sampling

Let’s revisit our previous ARX system, but change it into a PWARX system, make 50 observations and run the Gibbs sampler to collect 1000 samples.
Illustration of Gibbs sampling

Let’s revisit our previous ARX system, but change it into a PWARX system, make 50 observations and run the Gibbs sampler to collect 1000 samples.
Illustration of Gibbs sampling

Let’s revisit our previous ARX system, but change it into a PWARX system, make 50 observations and run the Gibbs sampler to collect 1000 samples.

Iteration number: 500, Number of clusters: 3
Illustration of Gibbs sampling

Let’s revisit our previous ARX system, but change it into a PWARX system, make 50 observations and run the Gibbs sampler to collect 1000 samples.
Illustration of Gibbs sampling

We can now use the samples from the Gibbs sampler to compute quantities of interest. For example the conditional expected value and variance or the full conditional density.
Illustration of Gibbs sampling

We can now use the samples from the Gibbs sampler to compute quantities of interest. For example the conditional expected value and variance or the full conditional density.
Pick and place machine
Pick and place machine

- Pick and place machines are used to place electronic components on a printed circuit board (PCB).

Pick and place machine

- Pick and place machines are used to place electronic components on a printed circuit board (PCB).
- Data from the vertical position of the mounting head.

Pick and place machine

- Pick and place machines are used to place electronic components on a printed circuit board (PCB).
- Data from the vertical position of the mounting head.
- Hybrid system with at least four modes identified from physical modeling: free mode, impact mode and saturations.

Pick and place machine

15 seconds of data sampled at 50 Hz.

▶ Use the first 8 seconds to learn model.

▶ Simulate the last 7 seconds using the true input.

Fit of 77.5 %, slightly worse than existing methods, but with quantified uncertainty.
Pick and place machine

- 15 seconds of data sampled at 50 Hz.

$y(t)$

$u(t)$
Pick and place machine

- 15 seconds of data sampled at 50 Hz.
- Use the first 8 seconds to learn model.
Pick and place machine

- 15 seconds of data sampled at 50 Hz.
- Use the first 8 seconds to learn model.
- Simulate the last 7 seconds using the true input.
Pick and place machine

- 15 seconds of data sampled at 50 Hz.
- Use the first 8 seconds to learn model.
- Simulate the last 7 seconds using the true input.
- Fit of 77.5%, slightly worse than existing methods, but with quantified uncertainty.

Bayesian nonparametric identification of PWARX systems
We have looked at a Bayesian nonparametric identification method for ARX systems. Shows promising results on both synthetic and real data.

Future work:
- Expand to other piecewise affine systems, e.g. state space models.
- Learn hyperparameters from data.

Paper with all details:

johan.wagberg@it.uu.se
Conclusion

- We have looked at a Bayesian nonparametric identification method for ARX systems.
Conclusion

- We have looked at a Bayesian nonparametric identification method for ARX systems.
- Shows promising results on both synthetic and real data.

Future work
- Expand to other piecewise affine systems, e.g. state space models.
- Learn hyperparameters from data.

Paper with all details
Conclusion

- We have looked at a Bayesian nonparametric identification method for ARX systems.
- Shows promising results on both synthetic and real data.
- Future work
Conclusion

- We have looked at a Bayesian nonparametric identification method for ARX systems.
- Shows promising results on both synthetic and real data.
- Future work
  - Expand to other piecewise affine systems, e.g. state space models.
Conclusion

- We have looked at a Bayesian nonparametric identification method for ARX systems.
- Shows promising results on both synthetic and real data.
- Future work
  - Expand to other piecewise affine systems, e.g. state space models.
  - Learn hyperparameters from data.
Conclusion

- We have looked at a Bayesian nonparametric identification method for ARX systems.
- Shows promising results on both synthetic and real data.
- Future work
  - Expand to other piecewise affine systems, e.g. state space models.
  - Learn hyperparameters from data.

Paper with all details