Bayesian nonparametric identification of piecewise affine ARX systems

Johan Wågberg

Uppsala University

June 3, 2014
Bayesian identification of ARX

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

\[ \phi_t = \begin{bmatrix} x_{t-1} & \cdots & x_{t-n_a} & u_{t-1} & \cdots & u_{t-n_b} \end{bmatrix}^T \]

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

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

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

ARX \((n_a, n_b)\) system

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

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

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

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

ARX \((n_a, n_b)\) system

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

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

Assume $[x^T \ t, y^T \ t]^T$ to be jointly Gaussian $[x^T, y^T] \sim \mathcal{N}(\mu, \Sigma) = \mathcal{N}(\mu_x, \Sigma_{xx})$, where

- $\mu_y | x = \mu_y + \Sigma_{yx} \Sigma^{-1}_{xx} (x_t - \mu_x)$
- $\Sigma_y | x = \Sigma_y - \Sigma_{yx} \Sigma^{-1}_{xx} \Sigma_{xy}$

Identify $\vartheta = \begin{bmatrix} \Sigma_{yx} \\ \Sigma \end{bmatrix}$, $\Lambda = \Sigma_y \Rightarrow y_t = \vartheta^T \phi_t + e_t \sim \mathcal{N}(0, \Lambda)$.
Prior on $\vartheta$ and $\Lambda$

We can construct a prior on $\vartheta$ and $\Lambda$ in the following way.
Prior on $\vartheta$ and $\Lambda$

We can construct a prior on $\vartheta$ and $\Lambda$ in the following way. Assume $[x_t^T, y_t^T]^T$ to be jointly Gaussian

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

Identify $\vartheta = \begin{bmatrix} \Sigma_{yx} \\ \Sigma_{yy} \end{bmatrix}$, $\Lambda = \Sigma_{y|x}$. 

$y_t = \vartheta^T \phi_t + e_t$, where $e_t \sim N(0, \Lambda)$.
Prior on $\vartheta$ and $\Lambda$

We can construct a prior on $\vartheta$ and $\Lambda$ in the following way. Assume $[x_t^T, y_t^T]^T$ to be jointly Gaussian

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

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 $\Lambda$

We can construct a prior on $\vartheta$ and $\Lambda$ in the following way. Assume $[x_t^T, y_t^T]^T$ to be jointly Gaussian

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

then

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

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} \Rightarrow y_t = \vartheta^T \varphi_t + e_t \\
e_t \sim \mathcal{N}(0, \Lambda)
Prior on $\theta$ and $\Lambda$

A prior on $\mu$ and $\Sigma$ can be used as prior on $\theta$ and $\Lambda$.

To ease computations, we use a conjugate prior. In this case: Normal inverse Wishart (NIW).

Our generative model now looks like

$$(\mu, \Sigma) \sim \text{NIW}(\mu_0, \lambda, \Psi, \nu)$$

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

$$(\mu, \Sigma) | x_1:n, y_1:n \sim \text{NIW}(\mu_n, \lambda_n, \Psi_n, \nu_n)$$
Prior on $\vartheta$ and $\Lambda$

A prior on $\mu$ and $\Sigma$ can be used as prior on $\vartheta$ and $\Lambda$. 
Prior on $\vartheta$ and $\Lambda$

A prior on $\mu$ and $\Sigma$ can be used as prior on $\vartheta$ and $\Lambda$. To ease computations, we use a conjugate prior.
Prior on $\vartheta$ and $\Lambda$

A prior on $\mu$ and $\Sigma$ can be used as prior on $\vartheta$ and $\Lambda$. To ease computations, we use a conjugate prior. In this case: Normal inverse Wishart ($\text{NIW}$).
Prior on $\vartheta$ and $\Lambda$

A prior on $\mu$ and $\Sigma$ can be used as prior on $\vartheta$ and $\Lambda$. To ease computations, we use a conjugate prior. In this case: Normal inverse Wishart ($\text{NIW}$). Our generative model now looks like

$$(\mu, \Sigma) \sim \text{NIW}(\mu_0, \lambda, \Psi, \nu)$$

$$(x_t, y_t) \mid \mu, \Sigma \sim \mathcal{N}(\mu, \Sigma)$$
Prior on $\theta$ and $\Lambda$

A prior on $\mu$ and $\Sigma$ can be used as prior on $\theta$ and $\Lambda$.
To ease computations, we use a conjugate prior.
In this case: Normal inverse Wishart ($\mathcal{NIW}$).
Our generative model now looks like

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

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

$$(\mu, \Sigma) \mid x_{1:n}, y_{1:n} \sim \mathcal{NIW}(\mu_n, \lambda_n, \Psi_n, \nu_n)$$
Example - Bayesian ARX
Example - Bayesian ARX

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

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

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

\[ y_t = 1u_{t-1} + 2 + \epsilon_t. \]
Example - Bayesian ARX

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

\[ y_t = \mathbb{1} u_{t-1} + 2 + \epsilon_t. \]
Example - Bayesian ARX

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

\[
y_t = 1u_{t-1} + 2 + \epsilon_t.
\]
Example - Bayesian ARX

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

\[ y_t = 1u_{t-1} + 2 + \epsilon_t. \]
Example - Bayesian ARX

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

\[
y_t = 1u_{t-1} + 2 + \epsilon_t.\]

\((\mu, \Sigma)\)
Example - Bayesian ARX

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

\[ y_t = 1u_{t-1} + 2 + \varepsilon_t. \]
Example - Bayesian ARX

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

\[ y_t = \begin{cases} \vartheta_1 \phi_t + e_t & \text{if } x_t \in \mathbb{R}_1, \\ \vartheta_K \phi_t + e_t & \text{if } x_t \in \mathbb{R}_K, \end{cases} \]

where \( \{\mathbb{R}_k\}_{k=1}^{K} \) is a 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

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

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

\[\begin{array}{c}
\mathcal{R}_1 \\
\mathcal{R}_2 \\
\mathcal{R}_3 \\
\mathcal{R}_4 \\
\mathcal{R}_5 \\
\mathcal{R}_6
\end{array}\]
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

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

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

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

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

\[
(\mu, \Sigma) \sim \text{NIW}(\mu_0, \lambda, \Psi, \nu),
\]

\[
(x_t, y_t) \mid (\mu, \Sigma) \sim \mathcal{N}(\mu, \Sigma)
\]
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 PWARX system

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

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

$$(x_t, y_t) \mid z_t, \mu_{z_t}, \Sigma_{z_t} \sim \mathcal{N}(\mu_{z_t}, \Sigma_{z_t})$$
Prior $\pi$ on $z_1:N$

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

$$\pi(z_{n+1} = k | z_{1:n}) = \frac{\alpha_n + \alpha_k}{\alpha_n + \alpha_k + K - 1}, \ldots, \frac{\alpha_n + \alpha_k}{\alpha_n + \alpha_k + K - 1}$$

$\rightarrow$ Join existing cluster

$\alpha_n + \alpha_k + K + 1$ $\rightarrow$ Create new cluster

$\sum_{t=1}^{n} I(z_t = k)$ $\rightarrow$ 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.
The prior $\pi$ should have no upper bound on the number of clusters $K$. 
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

$$
\pi(z_{n+1} = k \mid z_{1:n}) = \begin{cases} 
\frac{n_k}{n+\alpha} & k = 1, \ldots, K \\
\frac{\alpha}{n+\alpha} & k = K + 1
\end{cases}
$$

$$
n_k = \sum_{t=1}^{n} \mathbb{I}(z_t = k)
$$
The prior $\pi$ should have no upper bound on the number of clusters $K$.
Let the prior $\pi(\alpha)$ on $z_{1:N}$ be

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

$$n_k = \sum_{t=1}^{n} \mathbb{I}(z_t = k)$$
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

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

$$
n_k = \sum_{t=1}^{n} \mathbb{I}(z_t = k)
$$
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 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

\[ p(y \mid x, x_1:N, y_1:N) . \]
Inference

We are interested in the posterior distribution

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

Having this, any point estimate can be computed.
Inference

We are interested in the posterior distribution

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

Having this, any point estimate can be computed. We can write

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

We are interested in the posterior distribution

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

Having this, any point estimate can be computed.

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

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

Having this, any point estimate can be computed.

We can write

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

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

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

We are interested in the posterior distribution

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

Having this, any point estimate can be computed. 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 of 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 \( M \) do

2: for \( t = 1 \) to \( N \) do

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.

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.
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.

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 $M$ do
2:   for $t = 1$ to $N$ do
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,
Illustration of Gibbs sampling

Let’s revisit our previous ARX system, but change it into a PWARX system,
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.
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.
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

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.
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.

\[ p(y | x = 2, x_{1:N}, y_{1:N}) \]
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
Pick and place machine

- 15 seconds of data sampled at 50 Hz.

```
y(t)
```

```
u(t)
```

Time [s]

June 3, 2014, 14 / 15
Reglernöte 2014: J. Wågberg - Bayesian nonparametric identification of PWARX systems
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 72.5 %, slightly worse than existing methods, but with quantified uncertainty.

![Graphs showing y(t) and u(t) over time](image-url)
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

Johan Wågberg, Fredrik Lindsten and Thomas B. Schön.

Bayesian nonparametric identification of piecewise affine ARX systems.

Submitted to the 53rd IEEE Conference on Decision and Control (CDC), Los Angeles, CA, USA, December, 2014.
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

Johan Wågberg, Fredrik Lindsten and Thomas B. Schön. **Bayesian nonparametric identification of piecewise affine ARX systems.** Submitted to the 53rd IEEE Conference on Decision and Control (CDC), Los Angeles, CA, USA, December, 2014.