Gaussian process SDE models

Roland Hostettler[1]

Department of Electrical Engineering
Uppsala University
E-Mail: roland.hostettler@angstrom.uu.se

December 7, 2022

[1] Based on joint work with Zheng Zhao, Filip Tronarp, and Simo Särkkä

Stochastic differential equations

A typical stochastic differential equation (SDE):

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, t)\,\mathrm{d}t + \mathbf{L}(\mathbf{x}_t, \mathbf{u}_t, t)\,\mathrm{d}\boldsymbol{\beta}_t$$

where

- $\mathbf{x}_t \in \mathbb{R}^{d_x}$ is the state,
- $\mathbf{u}_t \in \mathbb{R}^{d_u}$ is a deterministic input,
- $\mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, t)$is the drift function,
- $\mathbf{L}(\mathbf{x}_t, \mathbf{u}_t, t)$ is the diffusion function, and
- $\boldsymbol{\beta}_t$ is Brownian motion with diffusion matrix $\boldsymbol{\Sigma}$.

Modeling and identification using SDEs

Typical identification problems

- The SDE is given (e.g., from first principles);
- Data is given;
- Objective: Estimate the parameters in the SDE model.

What if...

...we do not know the SDE itself?

- Can we still use the SDE approach?
- How should we model the drift (and diffusion) function(s)?

Modeling and identification using SDEs (2/2)

Requirements

The model should…

- …be flexible;
- …account for different nonlinearities;
- …quantify uncertainty;
- …scale w.r.t. data and state dimensions.

A few approaches

- Parametric models (e.g., polynomials);
- neural SDEs and deep neural SDEs;
- Gaussian processes (GPs).

This talk: Gaussian processes!

Basic idea

GP SDE model (drift-only):

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, t)\,\mathrm{d}t + \mathrm{d}\boldsymbol{\beta}_t,$$

where

- Drift function

$$\mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, t) = \begin{bmatrix} f_1(\mathbf{x}_t, \mathbf{u}_t, t) & \dots & f_L(\mathbf{x}_t, \mathbf{u}_t, t) \end{bmatrix}^\mathsf{T},$$

- GP prior:

$$f_l(\mathbf{x}_t, \mathbf{u}_t, t \sim \mathcal{GP}(m(\mathbf{x}_t, \mathbf{u}_t, t), k(\mathbf{x}_t, \mathbf{u}_t, t, \mathbf{x}'_t, \mathbf{u}'_t, t')).$$

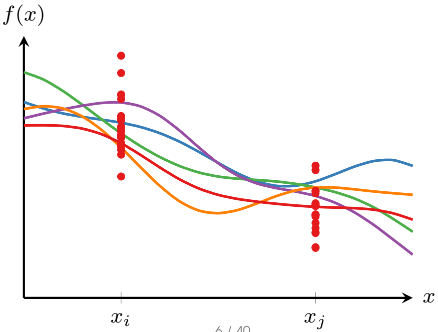
Gaussian processes

From random variables to random processes

- Assume: $f : \mathbb{R}^{d_x} \mapsto \mathbb{R}$ is a function of $\mathbf{x} \in \mathbb{R}^{d_x}$
- Let $f_i = f(\mathbf{x}_{\boldsymbol{i}})$
- Then, f is a Gaussian process if

$$p(f_i, f_j) = \mathcal{N}\left(\begin{bmatrix} f_i \\ f_j \end{bmatrix}; \begin{bmatrix} m_i \\ m_i \end{bmatrix}, \begin{bmatrix} \sigma_{ii}^2 & \sigma_{ij}^2 \\ \sigma_{ji}^2 & \sigma_{jj}^2 \end{bmatrix}\right)$$

for all $\mathbf{x}_i$ and $\mathbf{x}_j$



Definition

Definition 1: Gaussian process

A Gaussian process is a random function $f : \mathbb{R}^{d_x} \mapsto \mathbb{R}$ with mean function

$$\mathrm{E}\{f(\mathbf{x})\} = m(\mathbf{x})$$

and covariance function

$$\mathrm{Cov}\{f(\mathbf{x}), f(\mathbf{x}')\} = k(\mathbf{x}, \mathbf{x}')$$

for which it holds that

$$p(f(\mathbf{x}), f(\mathbf{x}')) = \mathcal{N}\left(\begin{bmatrix} f(\mathbf{x}) \\ f(\mathbf{x}') \end{bmatrix}; \begin{bmatrix} m(\mathbf{x}) \\ m(\mathbf{x}') \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}, \mathbf{x}) & k(\mathbf{x}, \mathbf{x}') \\ k(\mathbf{x}', \mathbf{x}) & k(\mathbf{x}', \mathbf{x}') \end{bmatrix}\right)$$

for all $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^{d_x}$. We denote f as

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right).$$

GP regression (1/2)

- Assume that

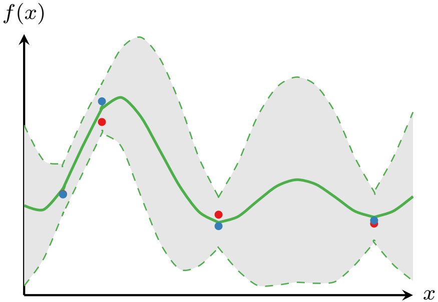
$$f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$$
$$y = f(\mathbf{x}) + r$$

  with $r \sim N(0, \sigma_r^2)$
- Given $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ and

$$\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_N \end{bmatrix}^{\mathsf{T}},$$

  we have

$$p(\mathbf{f}) = \mathcal{N}\left(\mathbf{f}; \mathbf{0}, \mathbf{K}_{1:N,1:N}\right)$$
$$p(\mathbf{y} \mid \mathbf{f}) = \mathcal{N}\left(\mathbf{y}; \mathbf{f}, \sigma_r^2 \mathbf{I}_N\right)$$



GP regression (2/2)

- Given $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ and $\mathbf{y}$, we have

$$
\begin{aligned}
p(\mathbf{f}) &= \mathcal{N}\left(\mathbf{f}; \mathbf{0}, \mathbf{K}_{1:N,1:N}\right) \\
p(\mathbf{y} \mid \mathbf{f}) &= \mathcal{N}\left(\mathbf{y}; \mathbf{f}, \sigma_r^2 \mathbf{I}_N\right)
\end{aligned}
$$

- Joint density:

$$
p(\mathbf{f}, \mathbf{y}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{f} \\ \mathbf{y} \end{bmatrix}; \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{1:N,1:N} & \mathbf{K}_{1:N,1:N} \\ \mathbf{K}_{1:N,1:N} & \mathbf{K}_{1:N,1:N} + \sigma_r^2 \mathbf{I}_N \end{bmatrix}\right)
$$

- Conditioning on the measurements $\mathbf{y}$ yields the posterior

$$
\begin{aligned}
p(\mathbf{f} \mid \mathbf{y}) &= \mathcal{N}\left(\mathbf{f}; \mathbf{m}_{\mathbf{f}|\mathbf{y}}, \mathbf{P}_{\mathbf{f}|\mathbf{y}}\right) \\
\mathbf{m}_{\mathbf{f}|\mathbf{y}} &= \mathbf{K}_{1:N,1:N} \left(\mathbf{K}_{1:N,1:N} + \sigma_r^2 \mathbf{I}_N\right)^{-1} \mathbf{y} \\
\mathbf{P}_{\mathbf{f}|\mathbf{y}} &= \mathbf{K}_{1:N,1:N} - \mathbf{K}_{1:N,1:N} \left(\mathbf{K}_{1:N,1:N} + \sigma_r^2 \mathbf{I}_N\right)^{-1} \mathbf{K}_{1:N,1:N}
\end{aligned}
$$

State-space Gaussian process drift model

GP drift model (1)

Assume the scalar, autonomous, time-invariant model

$$\begin{aligned} \mathrm{d}x_t &= f(x_t)\,\mathrm{d}t + \mathrm{d}\beta_t, \\ f(x_t) &\sim \mathcal{GP}(m(x_t), k(x_t, x_t')). \end{aligned}$$

Assumptions

- We perfectly observe a full length N trajectory $x_{0:N} = \{x_0, x_1, \dots, x_N\}$;
- Mean function is zero, $m(x_t) = 0$;
- $k(x_t, x_t')$ is stationary, $k(x_t, x_t') = k(\Delta x)$ ($\Delta x = x_t' - x_t$)

Sub-optimal solution (1/2)

Euler–Maruyama discretization of the model:

$$
\begin{aligned}
x_n &\approx x_{n-1} + f(x_{n-1})\Delta t + v_n \\
v_n &\sim \mathcal{N}(0, \Delta t \sigma^2)
\end{aligned}
$$

- Define $y_n = x_n - x_{n-1}$, then:

$$
\begin{aligned}
y_n &= f(x_{n-1})\Delta t + v_n \\
v_n &\sim \mathcal{N}(0, \Delta t \sigma^2)
\end{aligned}
$$

- GP regression:

$$
\begin{aligned}
p(\mathbf{f}_{1:N} \mid \mathbf{y}_{1:N}) &= \mathcal{N}\left(\mathbf{f}_{1:N}; \mathbf{m}_{\mathbf{f}|\mathbf{y}}, \mathbf{P}_{\mathbf{f}|\mathbf{y}}\right) \\
\mathbf{m}_{\mathbf{f}|\mathbf{y}} &= \frac{1}{\Delta t}\mathbf{K}_{1:N,1:N}\left(\mathbf{K}_{1:N,1:N} + \frac{1}{\Delta t}\sigma^2\mathbf{I}_N\right)^{-1}\mathbf{y}_{1:N} \\
\mathbf{P}_{\mathbf{f}|\mathbf{y}} &= \mathbf{K}_{1:N,1:N} - \mathbf{K}_{1:N,1:N}\left(\mathbf{K}_{1:N,1:N} + \frac{1}{\Delta t}\sigma^2\mathbf{I}_N\right)^{-1}\mathbf{K}_{1:N,1:N}^{\mathsf{T}}
\end{aligned}
$$

Sub-optimal solution (2/2)

Solution:

$$\begin{aligned} y_n &= f(x_{n-1})\Delta t + v_n \\ v_n &\sim \mathcal{N}(0, \Delta t \sigma^2) \\ p(\mathbf{f}_{1:n} \mid \mathbf{y}_{1:n}) &= \mathcal{N}\left(\mathbf{f}_{1:n}; \mathbf{m}_{\mathbf{f}|\mathbf{y}}, \mathbf{P}_{\mathbf{f}|\mathbf{y}}\right) \end{aligned}$$

with $y_n = x_n - x_{n-1}$.

Drawbacks

- Euler–Maruyama requires small Δt
  - ...and higher order discretization schemes require gradient information;
- Regression scales according to $\mathcal{O}(N^3)$
  - ...problematic for large N.

Spectral decomposition

Recall:

$$f(x_t) \sim \mathcal{GP}(0, k(x_t, x_t')).$$

- The covariance function is assumed stationary, $k(x_t, x_t') = k(\Delta x)$
- Power spectral density:

$$S_f(\omega) = \mathcal{F}_{\Delta x}\{k(\Delta x)\};$$

- Decomposition (exact or approximate):

$$S_f(\omega) = q_f H(\mathrm{i}\,\omega) H^*(\mathrm{i}\,\omega);$$

- $H(\mathrm{i}\,\omega)$ is a linear system $\Rightarrow$ state-space representation:

$$\begin{aligned} \mathrm{d}\mathbf{z}_x &= \mathbf{A}\mathbf{z}_x\,\mathrm{d}x + \mathbf{B}\,\mathrm{d}\varepsilon_x, \\ f(x) &= \mathbf{C}\mathbf{z}_x. \end{aligned}$$

Resulting strategy

Thus, we can:

1. Sort the state sequence $x_{0:N}$ in ascending order (pseudo-time) $\Rightarrow \{\tilde{x}_{0:N}, \tilde{y}_{1:N}\}$;
2. Consider the model

$$
\begin{aligned}
\mathbf{z}_n &= \mathbf{F}\mathbf{z}_{n-1} + \mathbf{w}_n, \\
\tilde{y}_n &= \Delta t \mathbf{C}\mathbf{z}_{n-1} + v_n, \\
v_n &\sim \mathcal{N}(0, \Delta t \sigma^2)
\end{aligned}
$$

3. Run a Kalman filter & Rauch–Tung–Striebel smoother to estimate $p(\mathbf{z}_n \mid \tilde{y}_{1:N})$ and thus $p(f_n \mid \tilde{y}_{1:N})$

Pros and cons

- Inference scales according to $\mathcal{O}(N)$ (can be implemented as $\mathcal{O}(\log(N))$);
- Other discretization schemes: Gradients available, but needs xKF/PF/...;
- Requires sorting the data (incurs a cost).

Example: Double well

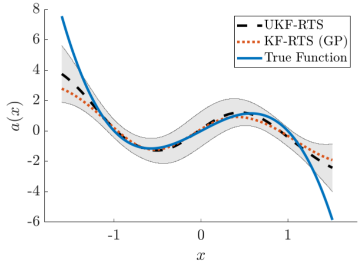
- SDE:

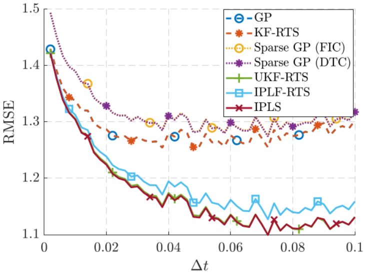
$$\mathrm{d}x_t = 3(x_t - x_t^3)\,\mathrm{d}t + \mathrm{d}\beta_t$$

  with $x_0 = 1$;
- Kernel: Matérn,

$$k(\Delta x) = \sigma_M^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\Delta x}{\ell} \right) K_\nu \left(\frac{\sqrt{2\nu}\Delta x}{\ell} \right)$$

  with $\nu = 5/2$, $\ell = 1.2$, and $\sigma_M = 3$.





Example: Beneš SDEs

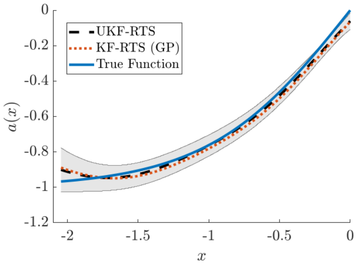
- SDE:

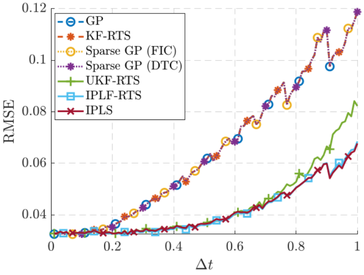
$$\mathrm{d}x_t = \tanh(x_t)\,\mathrm{d}t + 0.01\,\mathrm{d}\beta_t$$

  with $x_0 = 0$.
- Kernel: Matérn,

$$k(\Delta x) = \sigma_M^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\Delta x}{\ell} \right) K_\nu \left(\frac{\sqrt{2\nu}\Delta x}{\ell} \right)$$

  with $\nu = 5/2$, $\ell = 1.5$, and $\sigma_M = 0.3$.





Reduced-rank Gaussian process drift model

GP drift model (2)

How about the more general, d_x-dimensional model

$$\begin{aligned} \mathrm{d}\mathbf{x}_t &= \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, t)\,\mathrm{d}t + \mathrm{d}\boldsymbol{\beta}_t, \\ \mathbf{y}_n &= \mathbf{g}(\mathbf{x}_{t_n}, \mathbf{u}_{t_n}) + \mathbf{r}_n, \end{aligned}$$

with noisy observations $\mathbf{y}_n = \mathbf{y}(t_n)$ and

$$\begin{aligned} \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, t) &= \begin{bmatrix} f_1(\mathbf{x}_t, \mathbf{u}_t, t) & \dots & f_L(\mathbf{x}_t, \mathbf{u}_t, t) \end{bmatrix}^\mathsf{T}, \\ f_l(\mathbf{x}_t, \mathbf{u}_t, t) &\sim \mathcal{GP}(m(\mathbf{x}_t, \mathbf{u}_t, t), k(\mathbf{x}_t, \mathbf{u}_t, t, \mathbf{x}'_t, \mathbf{u}'_t, t'))? \end{aligned}$$

Assumptions

- Mean function is zero, $m(\mathbf{x}_t, \mathbf{u}_t, t) = 0$;
- Covariance function is separable,

$$k(\mathbf{x}_t, \mathbf{u}_t, t, \mathbf{x}'_t, \mathbf{u}'_t, t') = k_S(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}'_t, \mathbf{u}'_t) k_T(t, t');$$

- $k_T(t, t')$ is stationary, $k_T(t, t') = k_T(\tau)$.

Trouble on the horizon...

Model:

$$
\begin{aligned}
f_l(\mathbf{x}_t, \mathbf{u}_t, t) &\sim \mathcal{GP}(0, k_S(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}'_t, \mathbf{u}'_t) k_T(t, t')),\\
\mathrm{d}x_{l,t} &= f_l(\mathbf{x}_t, \mathbf{u}_t, t)\,\mathrm{d}t + \mathrm{d}\beta_{l,t},\\
\mathbf{y}_n &= \mathbf{g}(\mathbf{x}_{t_n}, \mathbf{u}_{t_n}) + \mathbf{r}_n.
\end{aligned}
$$

Problems

- $\mathbf{x}_t$ is a latent state, we only have access to $\mathbf{y}_n$;
- f_l depends on whole trajectories $\mathbf{x}_\tau$ and $\mathbf{u}_\tau$ for all $\tau \in [0, t]$;
- The model is non-Markovian.

Some solutions

- Sparse approximations (inducing points);
- Basis function expansion;
- ...

Basis function expansion (1/2)

Decompose

$$f_l(\mathbf{x}_t, \mathbf{u}_t, t) \sim \mathcal{GP}(0, k_S(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_t', \mathbf{u}_t')k_T(t, t')).$$

in $\mathbf{x}_t$, $\mathbf{u}_t$ such that

$$f_l(\mathbf{x}_t, \mathbf{u}_t, t) = \sum_{j=0}^{\infty} \alpha_{j,t}\psi_j(\mathbf{x}_t, \mathbf{u}_t),$$

where $\psi_j(\mathbf{x}_t, \mathbf{u}_t)$ are the eigenfunctions of $k_S(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_t', \mathbf{u}_t')$:

$$\langle \psi_i(\mathbf{x}_t, \mathbf{u}_t), \psi_j(\mathbf{x}_t, \mathbf{u}_{\boldsymbol{t}})\rangle = \begin{cases} 1 & i = j, \\ 0 & i \neq j, \end{cases}$$

$$\langle k_S(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_t', \mathbf{u}_t'), \psi_j(\mathbf{x}_t', \mathbf{u}_t')\rangle = \lambda_j \psi_j(\mathbf{x}_t, \mathbf{u}_t),$$

with

$$\langle g(\mathbf{x}), h(\mathbf{x})\rangle = \int g(\mathbf{x})h^*(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

Basis function expansion (2/2)

Basis function expansion of $f_l(\mathbf{x}_t, \mathbf{u}_t, t)$ in $\mathbf{x}_t$, $\mathbf{u}_t$:

$$f_l(\mathbf{x}_t, \mathbf{u}_t, t) = \sum_{j=0}^{\infty} \alpha_{j,t} \psi_j(\mathbf{x}_t, \mathbf{u}_t).$$

The (time-varying) coefficients are

$$\alpha_{j,t} = \langle f(\mathbf{x}_t, \mathbf{u}_t, t), \psi_j(\mathbf{x}_t, \mathbf{u}_t) \rangle.$$

- Then:

$$\begin{aligned} \operatorname{Cov}\{\alpha_{i,t}, \alpha_{j,t'}\} &= k_T(t, t') \lambda_j \delta_{ij} \\ \alpha_{j,t} &\sim \mathcal{GP}(0, k_T(t, t') \lambda_j) \end{aligned}$$

- $\alpha_{j,t}$s are still non-Markovian

Spectral decomposition

Coefficients:

$$\alpha_{j,t} \sim \mathcal{GP}(0, k_{\alpha,j}(t,t'))$$

with $k_{\alpha,j}(t,t') = k_T(t,t')\lambda_j = k_T(\tau)\lambda_j$

- Power spectral density:
$$S_{\alpha,j}(\omega) = \mathcal{F}_\tau\{k_{\alpha,j}(\tau)\};$$
- Decomposition:
$$S_{\alpha,j}(\omega) = q_{\alpha,j} H(\mathrm{i}\,\omega) H^*(\mathrm{i}\,\omega);$$
- $H(\mathrm{i}\,\omega)$ is a linear system $\Rightarrow$ state-space representation:

$$\begin{aligned} \mathrm{d}\mathbf{z}_{j,t} &= \mathbf{A}_j \mathbf{z}_{j,t}\,\mathrm{d}t + \mathbf{B}_j\,\mathrm{d}\varepsilon_{j,t}, \\ \alpha_{j,t} &= \mathbf{C}_j \mathbf{z}_{j,t}. \end{aligned}$$

Dynamic model: Summary (1/2)

- Basis function expansion with finite approximation:

$$\begin{aligned} f_l(\mathbf{x}_t, \mathbf{u}_t, t) &= \sum_{j=0}^{\infty} \alpha_{j,t} \psi_j(\mathbf{x}_t, \mathbf{u}_t) \\ &\approx \mathbf{\Psi}(\mathbf{x}_t, \mathbf{u}_t) \boldsymbol{\alpha}_{l,t}, \end{aligned}$$

  with
  - $\mathbf{\Psi}(\mathbf{x}_t, \mathbf{u}_t) = \begin{bmatrix} \psi_1(\mathbf{x}_t, \mathbf{u}_t) & \psi_2(\mathbf{x}_t, \mathbf{u}_t) & \dots & \psi_J(\mathbf{x}_t, \mathbf{u}_t) \end{bmatrix}$,
  - $\boldsymbol{\alpha}_{l,t} = \begin{bmatrix} \alpha_{1,t} & \alpha_{2,t} & \dots & \alpha_{J,t} \end{bmatrix}^\mathsf{T}$;
- Spectral decomposition:

$$\begin{aligned} \mathrm{d}\mathbf{z}_{j,t} &= \mathbf{A}_j \mathbf{z}_{j,t} \, \mathrm{d}t + \mathbf{B}_j \, \mathrm{d}\varepsilon_{j,t}, \\ \alpha_{j,t} &= \mathbf{C}_j \mathbf{z}_{j,t}. \end{aligned}$$

Dynamic model: Summary (2/2)

Complete model for $f_l(\mathbf{x}_t, \mathbf{u}_t, t)$:

$$\begin{aligned} \mathrm{d}\mathbf{z}_{j,t} &= \mathbf{A}_j \mathbf{z}_{j,t} \, \mathrm{d}t + \mathbf{B}_j \, \mathrm{d}\varepsilon_{j,t}, \\ \alpha_{j,t} &= \mathbf{C}_j \mathbf{z}_{j,t}, \\ \mathrm{d}x_l &= \mathbf{\Psi}(\mathbf{x}_t, \mathbf{u}_t) \boldsymbol{\alpha}_{l,t} \, \mathrm{d}t + \mathrm{d}\beta_{l,t}. \end{aligned}$$

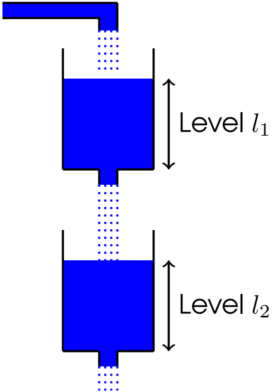
Observations

- (Partially linear) nonlinear state-space model;
- The full model is obtained by further stacking;

Encoding structure

- Prior knowledge of (physical) structure can be encoded;
- Example: Coupled system:

$$
\begin{aligned}
\mathrm{d}x_{1,t} &= f(x_{1,t}, u_t)\,\mathrm{d}t + \mathrm{d}\beta_{1,t}, \\
\mathrm{d}x_{2,t} &= f(x_{1,t}, x_{2,t})\,\mathrm{d}t + \mathrm{d}\beta_{2,t}.
\end{aligned}
$$



Estimation

Recall:

$$\begin{aligned} \mathrm{d}\mathbf{z}_{j,t} &= \mathbf{A}_j \mathbf{z}_{j,t}\,\mathrm{d}t + \mathbf{B}_j\,\mathrm{d}\varepsilon_{j,t}, \\ \alpha_{j,t} &= \mathbf{C}_j \mathbf{z}_{j,t}, \\ \mathrm{d}x_l &= \boldsymbol{\Psi}(\mathbf{x}_t, \mathbf{u}_t)\boldsymbol{\alpha}_{l,t}\,\mathrm{d}t + \mathrm{d}\beta_{l,t}, \\ \mathbf{y}_n &= \mathbf{g}(\mathbf{x}_{t_n}, \mathbf{u}_{t_n}) + \mathbf{r}_n \end{aligned}$$

Objectives

1. Estimate the Gaussian process' hyperparameters (found in $\mathbf{A}_j, \mathbf{B}_j, \mathbf{C}_j \ldots$);
2. Estimate the state $\mathbf{x}_t$ itself;
3. Predict the future state.

Standard tools such as Kalman filters/smoothers
or particle methods can be used.

Examples: Setup

- Two benchmark problems:
  - Bouc–Wen oscillator;
  - Cascaded tanks;
- Assumptions:
  - Linear observation models;
  - Euler–Maruyama discretization;
  - Fourier basis functions:

$$\psi_j(x) = \frac{1}{\sqrt{\gamma}} \exp\left(\frac{\mathrm{i}\, j 2\pi x}{\gamma}\right)$$

- Estimation:
  - Extended Kalman filter;
  - Hyperparameters: Maximization of the marginal likelihood $p(\boldsymbol{\theta} \mid \mathbf{y}_{1:N})$;
- Performance criterion: One-step ahead prediction error.

Example: Bouc–Wen oscillator — Model and data

Model

$$
\begin{aligned}
\mathrm{d}x_{1,t} &= x_{2,t}\,\mathrm{d}t, \\
\mathrm{d}x_{2,t} &= f(x_{1,t}, x_{2,t}, u_t, t)\,\mathrm{d}t + \mathrm{d}\beta_t;
\end{aligned}
$$

- Covariance functions:

$$
\begin{aligned}
k_S(\mathbf{x}_t, u_t, \mathbf{x}_t', u_t') &= k_{\mathsf{SE}}(\mathbf{x}_t, \mathbf{x}_t') + k_{\mathsf{SE}}(u_t, u_t') \\
k_T(t, t') &= k_{\mathsf{OU}}(t, t');
\end{aligned}
$$

- Truncate at 25 ($\mathbf{x}_t$) and 5 (u_t) eigenfunctions.

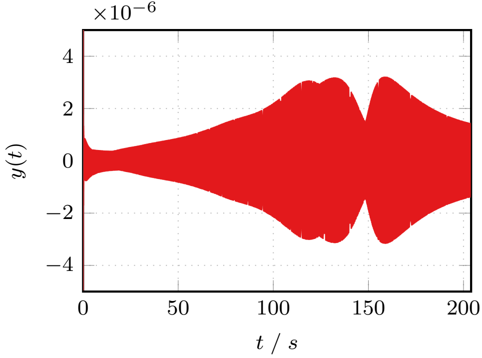
Data

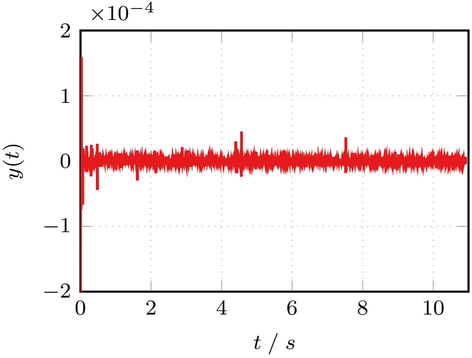
- Separate training and validation datasets;
- Two excitations: Swept sine and multi-sine.

Example: Bouc–Wen oscillator — Results (1/2)

Prediction RMSE:

- Training: 2.65×10^{-5}
- Multisine validation: 0.580×10^{-5}
- Swept sine validation: 0.096×10^{-5}





Example: Bouc–Wen oscillator — Results (2/2)

Comparison of the RMSE and R^2 values for the Bouc–Wen benchmark.

Model	Multisine		Swept sine	
	RMSE	R^2	RMSE	R^2
AR(1)	21.6×10^{-5}	0.676	20.9×10^{-5}	0.685
BLA[1]	1.13×10^{-5}	0.983	0.698×10^{-5}	0.989
Volterra[1]	0.895×10^{-5}	0.986	0.347×10^{-5}	0.994
Proposed	0.580×10^{-5}	0.991	0.096×10^{-5}	0.998

[1] Schoukens and Griesing-Scheiwe (2016)

Example: Cascaded tanks — Model and data

Model

$$\begin{aligned} \mathrm{d}x_{1,t} &= f_1(x_{1,t}, u_t, t)\,\mathrm{d}t + \mathrm{d}\beta_{1,t} \\ \mathrm{d}x_{2,t} &= f_2(x_{1,t}, x_{2,t}, t)\,\mathrm{d}t + \mathrm{d}\beta_{2,t} \end{aligned}$$

- Covariance function:

$$k(\mathbf{x}, \mathbf{x}', t, t') = k_{\mathsf{SE}}(\mathbf{x}, \mathbf{x}')k_{\mathsf{OU}}(t, t').$$

- Remaining parameters: Same as for Bouc–Wen

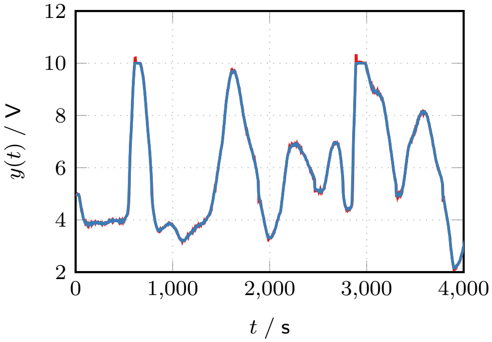
Data

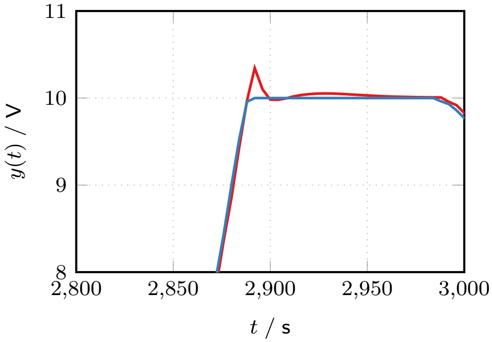
- Separate training and validation datasets;
- Arbitrary operational data.

Example: Cascaded tanks — Results (1/2)

Prediction RMSE:

- Training set RMSE: 51.5×10^{-3}
- Validation set RMSE: 57.6×10^{-3}.





Example: Cascaded tanks — Results (2/2)

Comparison of the RMSE and R^2 values for the cascaded tanks benchmark.

Model	RMSE	R^2
AR(1)	185.9×10^{-3}	0.911
BLA[1]	55.6×10^{-3}	0.974
Volterra[1]	49.4×10^{-3}	0.991
GP drift	57.6×10^{-3}	0.972

[1] Schoukens and Griesing-Scheiwe (2016)

Some pitfalls

- Truncation order J depends on the hyperparameters $\Rightarrow$ need to be chosen carefully and taken into account when estimating the hyperparameters;
- Fourier basis functions are local:

$$\psi_j(x) = \frac{1}{\sqrt{\gamma}} \exp\left(\frac{\mathrm{i}\, j 2\pi x}{\gamma}\right)$$

  - γ controls the size of the neighborhood;
  - Large $\gamma \Rightarrow$ large J;
- Large $J \Rightarrow$ high-dimensional model (but conditionally linear in most states):

$$\begin{aligned}
\mathrm{d}\mathbf{z}_{j,t} &= \mathbf{A}_j \mathbf{z}_{j,t}\, \mathrm{d}t + \mathbf{B}_j\, \mathrm{d}\varepsilon_{j,t}, \\
\alpha_{j,t} &= \mathbf{C}_j \mathbf{z}_{j,t}, \\
\mathrm{d}x_l &= \mathbf{\Psi}(\mathbf{x}_t, \mathbf{u}_t) \boldsymbol{\alpha}_{l,t}\, \mathrm{d}t + \mathrm{d}\beta_{l,t}, \\
\mathbf{y}_n &= \mathbf{g}(\mathbf{x}_{t_n}, \mathbf{u}_{t_n}) + \mathbf{r}_n
\end{aligned}$$

- Poor scaling of higher-dimensional basis functions that are based on cartesian products of low-dimensional ones.

Summary

State-space GP SDE model

- Model:

$$
\begin{aligned}
\mathrm{d}x_t &= f(x_t)\,\mathrm{d}t + \mathrm{d}\beta_t, \\
f(x_t) &\sim \mathcal{GP}(m(x_t), k(x_t, x_t')).
\end{aligned}
$$

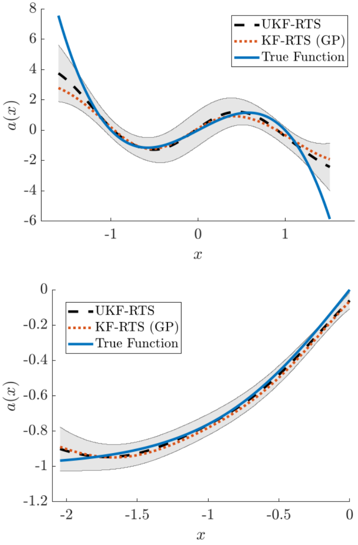
- Euler–Maruyama discretization:

$$
\begin{aligned}
y_n &= f(x_{n-1})\Delta t + v_n \\
v_n &\sim \mathcal{N}(0, \Delta t \sigma^2)
\end{aligned}
$$

  with $y_n = x_n - x_{n-1}$.
- Spectral decomposition (in Δx):

$$
\begin{aligned}
\mathbf{z}_n &= \mathbf{F}\mathbf{z}_{n-1} + \mathbf{w}_n, \\
\tilde{y}_n &= \Delta t \mathbf{C}\mathbf{z}_{n-1} + v_n, \\
v_n &\sim \mathcal{N}(0, \Delta t \sigma^2)
\end{aligned}
$$

- Estimation using RTS smoother



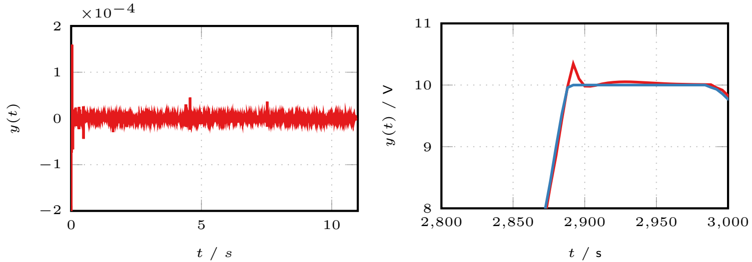
Reduced-rank GP SDE model

- Proposed model:

$$\begin{aligned} \mathrm{d}\mathbf{x}_t &= \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, t)\,\mathrm{d}t + \mathrm{d}\boldsymbol{\beta}_t \\ f_l(\mathbf{x}_t, \mathbf{u}_t, t) &\sim \mathcal{GP}(0, k_S(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}'_t, \mathbf{u}'_t)k_T(t, t')) \end{aligned}$$

- Basis function expansion and spectral decomposition yield:

$$\begin{aligned} \mathrm{d}\mathbf{z}_{j,t} &= \mathbf{A}_j\mathbf{z}_{j,t}\,\mathrm{d}t + \mathbf{B}_j\,\mathrm{d}\varepsilon_{j,t}, \\ \alpha_{j,t} &= \mathbf{C}_j\mathbf{z}_{j,t}, \\ \mathrm{d}x_l &= \mathbf{\Psi}(\mathbf{x}_t, \mathbf{u}_t)\boldsymbol{\alpha}_{l,t}\,\mathrm{d}t + \mathrm{d}\beta_{l,t} \end{aligned}$$



References

References

- C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.
- S. Särkkä, A. Solin, and J. Hartikainen, "Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering," *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 51–61, July 2013.
- A. Solin and S. Särkkä, "Hilbert space methods for reduced-rank Gaussian process regression," 2014, arXiv:1401.5508.
- R. Hostettler, F. Tronarp, and S. Särkkä, "Modeling the drift function in stochastic differential equations using reduced rank Gaussian processes," in *18th IFAC Symposium on System Identification (SYSID)*, Stockholm, Sweden, July 2018.
- Z. Zhao, F. Tronarp, R. Hostettler, and S. Särkkä, "State-space Gaussian process for drift estimation in stochastic differential equations," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, May 2020.