UPPSALA
UNIVERSITET

# Machine learning approaches for system identification

Thomas Schön, Uppsala University, Sweden.

Conference on Modelling Identification and Control of Nonlinear Systems (MICNON)

Guadalajara, Mexico,
June 22, 2018.

# Key lesson from contemporary Machine Learning

**Flexible models** often give the best performance.

How can we build and work with these flexible models?

1. Models that use a large (but fixed) number of parameters. (**parametric**, ex. deep learning)

   LeCun, Y., Bengio, Y., and Hinton, G. **Deep learning**, *Nature*, Vol 521, 436–444, 2015.

2. Models that use more parameters as we get access to more data. (**non-parametric**, ex. Gaussian process)

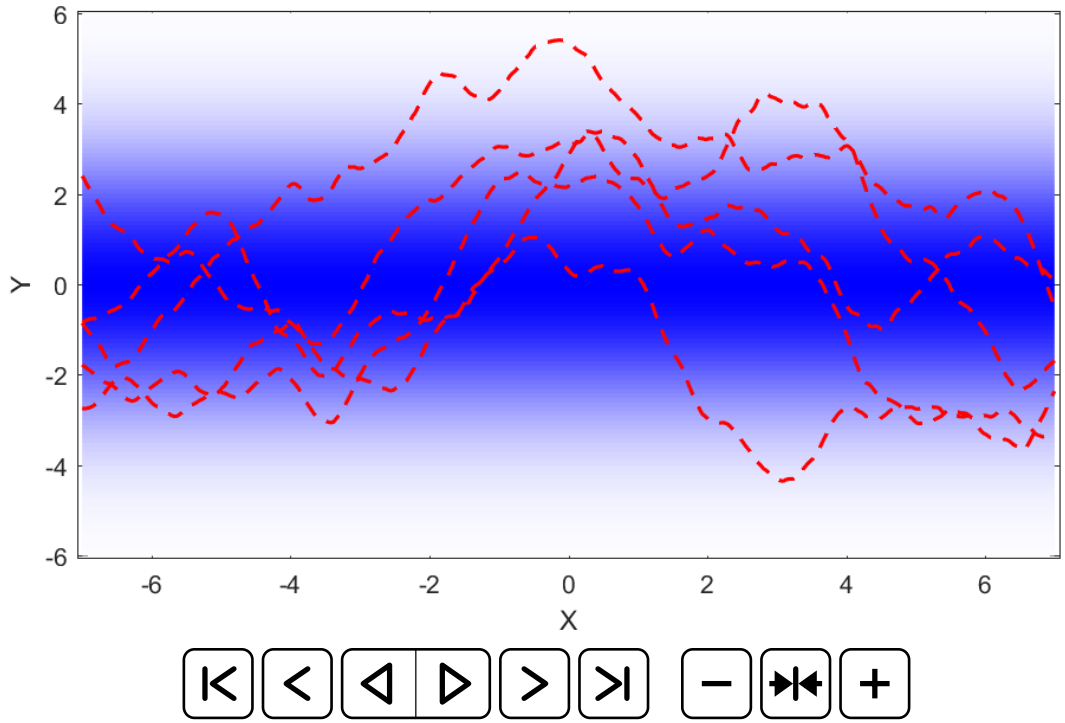
   Ghahramani, Z. **Bayesian nonparametrics and the probabilistic approach to modeling**. *Phil. Trans. R. Soc. A* 371, 2013.

   Ghahramani, Z. **Probabilistic machine learning and artificial intelligence**. *Nature* 521:452-459, 2015.

# Gaussian process

The Gaussian process is a **non-parametric** and **probabilistic** model of a nonlinear function.

- **Non-parametric** means that it does not rely on any particular parametric functional form to be postulated.
- **Probabilistic** means that it takes uncertainty into account in every aspect of the model.

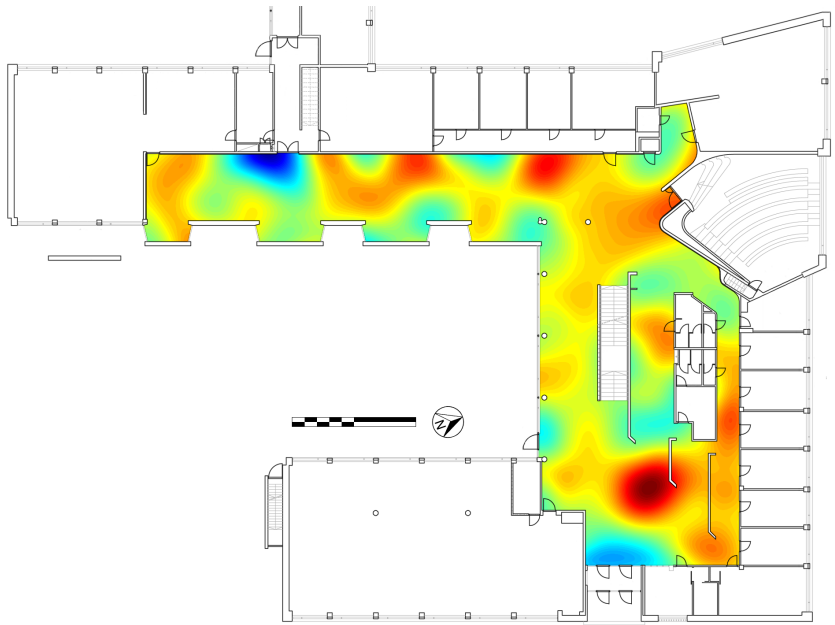


# Motivation 0 – Static model of the ambient magnetic field

The Earth's magnetic field sets a background for the ambient magnetic field. Deviations make the field vary from point to point.

**Aim:** Build a map (i.e., a model) of the magnetic environment based on magnetometer measurements.

**Solution:** Customized Gaussian process that obeys Maxwell's equations.



www.youtube.com/watch?v=enlMiUqPVJo


Arno Solin, Manon Kok, Niklas Wahlström, TS and Simo Särkkä. **Modeling and interpolation of the ambient magnetic field by Gaussian processes**. *IEEE Transactions on Robotics*, 2018. (in press)

Carl Jidling, Niklas Wahlström, Adrian Wills and TS. **Linearly constrained Gaussian processes**. *Advances in Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, December, 2017.

# Motivation 1 – GP-based linear impulse response estimation

Consider a linear time-invariant dynamical system described by

$$y(t_k) = \int_0^\infty g(\tau) u(t_k - \tau) d\tau + e(t_k).$$

**Task:** Learn a model of the true underlying impulse response $g(\tau)$.

Beats the "classical system identification approach".

Gianluigi Pillonetto and Giuseppe De Nicolao. **A new kernel-based approach for linear system identification**. *Automatica*, 46(1):81–93, 2010.

The GP offers a **data-driven model flexibility tuning**, an automatic **regularization** striking a bias-variance trade-off that is "just right".

The "classic" parametric approaches and the GP-based approach are linked via a **decision-theoretic** formulation.

Johan Wågberg, Dave Zachariah and TS. **Regularized parametric system identification: a decision-theoretic formulation**. In *Proceedings of the American Control Conference (ACC)*, Milwaukee, WI, USA, June, 2018.

# Motivation 2 – GP-based nonlinear ARX models

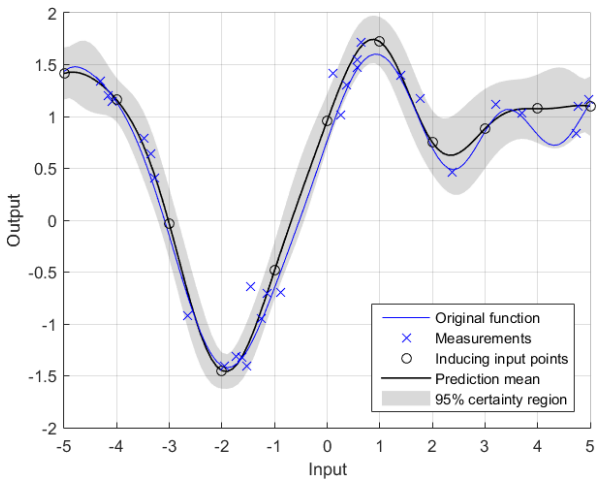
Standard nonlinear ARX model structure

$$\begin{aligned} y_t &= \varphi(y_{t-1}, \ldots, y_{t-n_y}, u_t, \ldots, u_{t-n_u}) + e_t \\ &= \varphi(z_t) + e_t, \end{aligned}$$

where $\varphi$ is some function and $z_t = (y_{t-1}, \ldots, y_{t-n_y}, u_t, \ldots, u_{t-n_u})$.

The GP can be used to represent the unknown nonlinear function $\varphi$,

$$\varphi(z) \sim \mathcal{GP}(0, \kappa_\eta(z, z')).$$





Jus Kocijan, Agathe Girard, Blaz Banko, and Roderick Murray-Smith. **Dynamic systems identification with Gaussian processes.** *Mathematical and Computer Modelling of Dynamical Systems*, 11(4):411–424, 2005.

Hildo Bijl, TS, Jan-Willem van Wingerden and Michel Verhaegen. **System identification through online sparse Gaussian process regression with input noise.** *IFAC Journal of Systems and Control*, 2:1–11, December, 2017.

# Motivation 3 – GP-based nonlinear state space model

"Inspired by the Gaussian process, enabled by the particle filter"

$$x_{t+1} = f(x_t) + v_t, \qquad \text{s.t.} \quad f(x) \sim \mathcal{GP}(0, \kappa_{\eta,f}(x, x')),$$
$$y_t = g(x_t) + e_t, \qquad \text{s.t.} \quad g(x) \sim \mathcal{GP}(0, \kappa_{\eta,g}(x, x')).$$

Results in a **flexible** non-parametric model where the GP prior takes on the **role of a regularizer**. Enables regularization also in nonlinear models.

We can now approximately recover the posterior distribution

$$p(f, g, Q, R, \eta \mid y_{1:T}),$$

(we use **SMC and MCMC**).

---


Frigola, Roger, Fredrik Lindsten, TS, and Carl Rasmussen. **Bayesian inference and learning in Gaussian process state-space models with particle MCMC**. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

Andreas Svensson and TS. **A flexible state space model for learning nonlinear dynamical systems**, *Automatica*, 80:189-199, June, 2017.

# Motivation 4 – GP-based maximum likelihood in nonlinear SSMs

Find the unknown parameters $\theta$ in a nonlinear SSM

$$x_t = f(x_{t-1}, \theta) + v_t,$$
$$y_t = g(x_t, \theta) + e_t,$$
$$x_0 \sim p(x_0 \mid \theta).$$

**Maximum likelihood** – model the unknown parameters as a deterministic variable $\theta$ and solve

$$\max_{\theta} p(y_{1:T} \mid \theta),$$

where $p(y_{1:T} \mid \theta) = \prod_{t=1}^{T} \int p(y_t \mid x_t, \theta) \underbrace{p(x_t \mid y_{1:t-1}, \theta)}_{\text{approx. by SMC}} \mathrm{d}x_t.$

**Challenge:** The non-convex optimization problem is stochastic!

Adrian G. Wills and TS. **On the construction of probabilistic Newton-type algorithms**, *Proceedings of the 56th IEEE Conference on Decision and Control (CDC)*, Melbourne, Australia, December 2017.

# Motivation – why use the GP in system identification?

**Static models:**

0. Estimating the ambient magnetic field

**Linear dynamical models:**

1. Impulse response estimation

**Nonlinear dynamical models:**

2. Nonlinear ARX models
3. Nonlinear state space model
4. Maximum likelihood learning of nonlinear SSM
    - Stochastic quasi-Newton algorithm (much more general)

**Perhaps most interesting:**

5. Situations where it has not yet been used...

# Outline

**Message:** The Gaussian process can be used to construct **new models and algorithms** for identification of **nonlinear** dynamical systems.

**Outline:**

Introductory motivation

Part 1 – Probabilistic modelling of nonlinear dynamical systems

Part 2 – Inferring the state via sequential Monte Carlo

Part 3 – Stochastic optimization

*"The Gaussian process (GP) is a non-parametric and probabilistic model of a nonlinear function."*

# Part 1 – Probabilistic modelling of dynamical systems

# Probabilistic modeling of dynamical systems

Probabilistic modeling allow for **representing and manipulating uncertainty** in data, models, decisions and predictions.

A **parametric** state space model (SSM) is given by:

$$
\begin{aligned}
x_t &= f_\theta(x_{t-1}, u_t) + v_{\theta,t}, \\
y_t &= g_\theta(x_t, u_t) + e_{\theta,t}, \\
x_0 &\sim p_\theta(x_0), \\
(\theta &\sim p(\theta)).
\end{aligned}
\qquad
\begin{aligned}
x_t \mid x_{t-1} &\sim p_\theta(x_t \mid x_{t-1}, u_t), \\
y_t \mid x_t &\sim p_\theta(y_t \mid x_t, u_t), \\
x_0 &\sim p_\theta(x_0), \\
(\theta &\sim p(\theta)).
\end{aligned}
$$

# SSM – full probabilistic model

The **full probabilistic model** is given by

$$p(x_{0:T}, \theta, y_{1:T}) = \underbrace{p(y_{1:T} \mid x_{0:T}, \theta)}_{\text{data distribution}} \underbrace{p(x_{0:T}, \theta)}_{\text{prior}}$$

Distribution describing a parametric nonlinear SSM

$$p(x_{0:T}, \theta, y_{1:T}) = \underbrace{\prod_{t=1}^{T} \underbrace{p(y_t \mid x_t, \theta)}_{\text{observation}}}_{\text{data distribution}} \underbrace{\prod_{t=1}^{T} \underbrace{p(x_t \mid x_{t-1}, \theta)}_{\text{dynamics}} \underbrace{p(x_0 \mid \theta)}_{\text{state}} \underbrace{p(\theta)}_{\text{param.}}}_{\text{prior}}$$

**Model = probability distribution!**

# Finding the states and the parameters

Based on our generative model, compute the **posterior distribution**

$$p(x_{0:T}, \theta \mid y_{1:T}) = \underbrace{p(x_{0:T} \mid \theta, y_{1:T})}_{\text{state inf.}} \underbrace{p(\theta \mid y_{1:T})}_{\text{param. learn.}}.$$

**Bayesian** formulation – model the unknown parameters as a random variable $\theta \sim p(\theta)$ and compute

$$p(\theta \mid y_{1:T}) = \frac{p(y_{1:T} \mid \theta) p(\theta)}{p(y_{1:T})}$$

**Maximum likelihood** formulation – model the unknown parameters as a deterministic variable and solve

$$\widehat{\theta} = \arg\max_{\theta \in \Theta} p(y_{1:T} \mid \theta).$$

# Central object – the likelihood

The likelihood is computed by marginalizing

$$p(x_{0:T}, y_{1:T} \mid \theta) = p(x_0 \mid \theta) \prod_{t=1}^{T} p(y_t \mid x_t, \theta) \prod_{t=1}^{T} p(x_t \mid x_{t-1}, \theta),$$

w.r.t the state sequence $x_{0:T}$,

$$p(y_{1:T} \mid \theta) = \int p(x_{0:T}, y_{1:T} \mid \theta) \mathrm{d}x_{0:T}.$$

(We are averaging $p(x_{0:T}, y_{1:T} \mid \theta)$ over all possible state sequences.)

Equivalently we have

$$p(y_{1:T} \mid \theta) = \prod_{t=1}^{T} p(y_t \mid y_{1:t-1}, \theta) = \prod_{t=1}^{T} \int p(y_t \mid x_t, \theta) \underbrace{p(x_t \mid y_{1:t-1}, \theta)}_{\text{key challenge}} \mathrm{d}x_t.$$

---

TS, Fredrik Lindsten, Johan Dahlin, Johan Wågberg, Christian A. Naesseth, Andreas Svensson and Liang Dai. **Sequential Monte Carlo methods for system identification.** In *Proceedings of the 17th IFAC Symposium on System Identification (SYSID)*, Beijing, China, October 2015.

# The model – learning relationship

Learning a model based on data leads to computational challenges:

- **Integration:** e.g. the HD integrals arising during marg. (averaging over all possible parameter values $z$):

$$p(y_{1:T}) = \int p(y_{1:T} \mid z) p(z) \mathrm{d}z.$$

- **Optimization:** e.g. when extracting point estimates, for example by maximizing the likelihood

$$\widehat{z} = \arg\max_{z} \; p(y_{1:T} \mid z)$$

Impossible to compute exactly, approximations are needed:

- Monte Carlo (MC), Markov chain MC, and sequential MC.
- Variational inference (VI).
- Stochastic optimization.

# Part 2 – Inferring the state via sequential Monte Carlo

# Learning the state – nonlinear filtering problem

**Aim:** Compute the nonlinear filtering distribution $p(x_t \mid y_{1:t})$.

---

The solution entails the **measurement update**

$$p(x_t \mid y_{1:t}) = \frac{\overbrace{p(y_t \mid x_t)}^{\text{measurement}} \overbrace{p(x_t \mid y_{1:t-1})}^{\text{prediction pdf}}}{p(y_t \mid y_{1:t-1})},$$

and the **time update**

$$p(x_t \mid y_{1:t-1}) = \int \underbrace{p(x_t \mid x_{t-1})}_{\text{dynamics}} \underbrace{p(x_{t-1} \mid y_{1:t-1})}_{\text{filtering pdf}} \mathrm{d}x_{t-1}.$$

**Key problem:** The integrals are intractable!

# Sequential Monte Carlo (SMC)

SMC provide approximate solutions to **integration** problems where there is a **sequential structure** present.

The **particle filter** approximates $p(x_t \mid y_{1:t})$ for

$$x_t = f(x_{t-1}) + v_t,$$
$$y_t = g(x_t) + e_t,$$

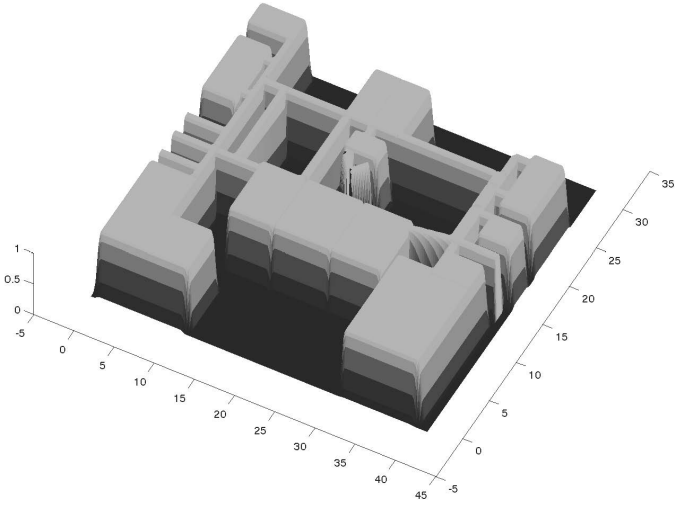
by maintaining an empirical distribution made up of $N$ samples (particles) $\{x_t^i\}_{i=1}^N$ and the corresponding weights $\{w_t^i\}_{i=1}^N$

$$\widehat{p}(x_t \mid y_{1:t}) = \sum_{i=1}^{N} \frac{w_t^i}{\sum_{j=1}^{N} w_t^j} \delta_{x_t^i}(x_t),$$

that converge to the true filtering distribution as $N \to \infty$.

# Application – indoor localization

**Aim:** Compute the position of a person moving around indoors using sensors (inertial, magnetometer, radio) located in an ID badge and a map.




Probability density function representing an office environment, the bright areas are rooms and corridors (i.e., walkable space).

**Show movie**

# Sequential Monte Carlo (SMC) – abstract

The distribution of interest $\pi(x)$ is called **target distribution**.

(Abstract) problem formulation: **Sample from a sequence** of probability distributions $\{\pi_t(x_{0:t})\}_{t\geq 1}$ defined on a sequence of spaces of increasing dimension, where

$$\pi_t(x_{0:t}) = \frac{\widetilde{\pi}_t(x_{0:t})}{Z_t},$$

such that $\widetilde{\pi}_t(x_t) : \mathcal{X}^t \to \mathbb{R}^+$ is known point-wise and $Z_t = \int \pi(x_{0:t})\mathrm{d}x_{0:t}$ is often computationally challenging.

1. Approximate the normalizing constant $Z_t$.
2. Approximate $\pi_t(x_t)$ and compute integrals $\int \varphi(x_t)\pi_t(x_t)\mathrm{d}x_t$.

**Important question:** How general is this formulation?

# Automation via a probabilistic programming language

1. Basic idea of **probabilistic programming**: equate probabilistic models with the computer programs that implement them.
2. Just as we can think of doing inference over models, we can now think of doing **inference over programs**.

---

Provides a means for **separating** the model and the learning algorithms.

We are developing a probabilistic programming language called **Birch**.

`birch-lang.org`

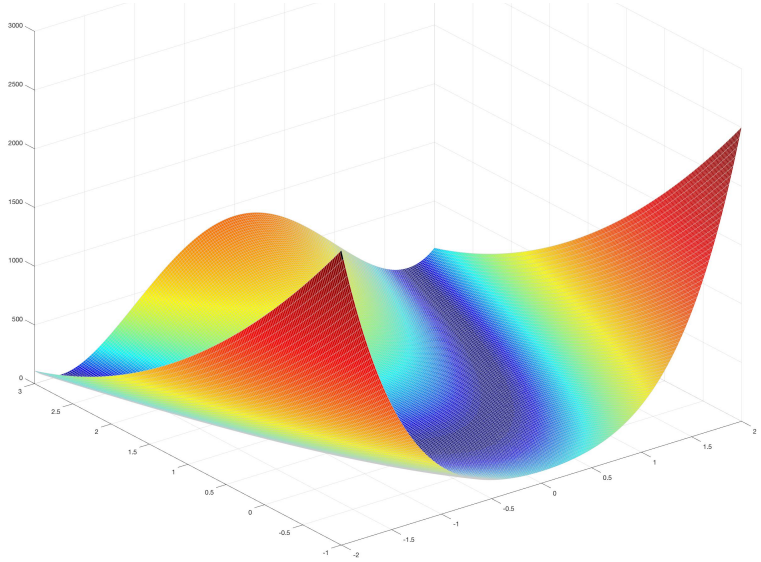
# Part 3 – Stochastic optimization

# Intuitive preview example – Rosenbrock's banana function

Let $f(\theta) = (1 - \theta_1)^2 + 100(\theta_2 - \theta_1^2)^2$.

**Deterministic problem**

$$\min_{\theta} f(\theta)$$

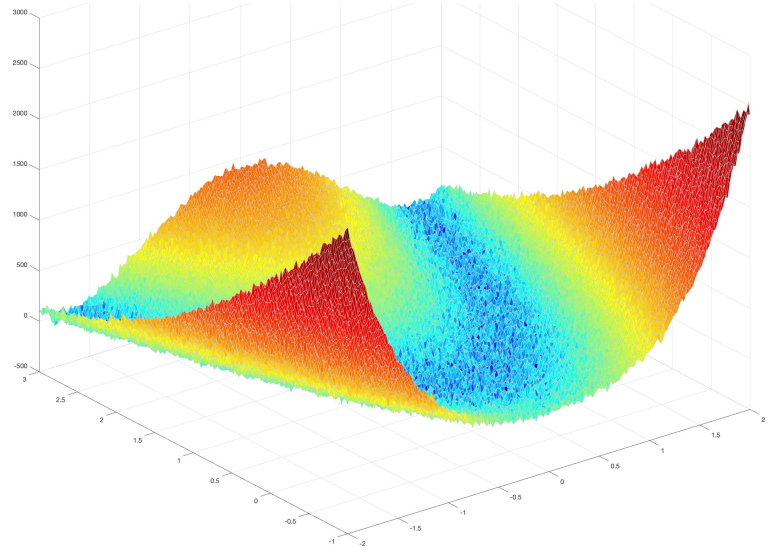


---

**Stochastic problem**

$$\min_{\theta} f(\theta)$$

when we only have access to noisy versions of the cost function ($\widetilde{f}(\theta) = f(\theta) + e$, $e = \mathcal{N}(0, 30^2)$) and its gradients.



# Quasi-Newton – A non-standard take

Our problem is of the form

$$\max_{\theta} f(\theta)$$

**Idea underlying (quasi-)Newton methods:** Learn a local quadratic model $q(\theta_k, \delta)$ of the cost function $f(\theta)$ around the current iterate $\theta_k$

$$q(\theta_k, \delta) = f(\theta_k) + g(\theta_k)^{\mathsf{T}}\delta + \frac{1}{2}\delta^{\mathsf{T}} H(\theta_k)\delta$$

$$g(\theta_k) = \nabla f(\theta)\big|_{\theta=\theta_k}, \qquad H(\theta_k) = \nabla^2 f(\theta)\big|_{\theta=\theta_k}, \qquad \delta = \theta - \theta_k.$$

---

We have measurements of

- the cost function $f_k = f(\theta_k)$,
- and its gradient $g_k = g(\theta_k)$.

**Question:** How do we update the Hessian model?

# Useful basic facts

Line segment connecting two adjacent iterates $\theta_k$ and $\theta_{k+1}$:

$$r_k(\tau) = \theta_k + \tau(\theta_{k+1} - \theta_k), \qquad \tau \in [0, 1].$$

---

1. The **fundamental theorem of calculus** states that

$$\int_0^1 \frac{\partial}{\partial \tau} \nabla f(r_k(\tau)) \mathrm{d}\tau = \nabla f(r_k(1)) - \nabla f(r_k(0)) = \underbrace{\nabla f(\theta_{k+1})}_{g_{k+1}} - \underbrace{\nabla f(\theta_k)}_{g_k}.$$

2. The **chain rule** tells us that

$$\frac{\partial}{\partial \tau} \nabla f(r_k(\tau)) = \nabla^2 f(r_k(\tau)) \frac{\partial r_k(\tau)}{\partial \tau} = \nabla^2 f(r_k(\tau))(\theta_{k+1} - \theta_k).$$

$$\underbrace{g_{k+1} - g_k}_{=y_k} = \int_0^1 \frac{\partial}{\partial \tau} \nabla f(r_k(\tau)) \mathrm{d}\tau = \int_0^1 \nabla^2 f(r_k(\tau)) \mathrm{d}\tau \underbrace{(\theta_{k+1} - \theta_k)}_{s_k}.$$

# Result – the quasi-Newton integral

With the definitions $y_k \triangleq g_{k+1} - g_k$ and $s_k \triangleq \theta_{k+1} - \theta_k$ we have

$$y_k = \int_0^1 \nabla^2 f(r_k(\tau)) \mathrm{d}\tau s_k.$$

**Interpretation:** The difference between two consecutive gradients ($y_k$) constitute a **line integral observation of the Hessian**.

**Problem:** Since the Hessian is unknown there is no functional form available for it.

# Solution 1 – recovering existing quasi-Newton algorithms

Existing quasi-Newton algorithms (e.g. BFGS, DFP, Broyden's method) assume the Hessian to be constant

$$\nabla^2 f(r_k(\tau)) \approx H_{k+1}, \qquad \tau \in [0, 1],$$

implying the following approximation of the integral (**secant condition**)

$$y_k = H_{k+1} s_k.$$

---

Find $H_{k+1}$ by **regularizing** $H$:

$$\begin{aligned} H_{k+1} = \min_{H} \quad & \|H - H_k\|_W^2, \\ \text{s.t.} \quad & H = H^\mathsf{T}, \quad H s_k = y_k, \end{aligned}$$

Equivalently, the existing quasi-Newton methods can be interpreted as **particular instances of Bayesian linear regression**.

Philipp Hennig. **Probabilistic interpretation of linear solvers**, *SIAM Journal on Optimization*, 25(1):234–260, 2015.

# Solution 2 – use a flexible nonlinear model

The approach used here is fundamentally different.

Recall that the problem is **stochastic** and **nonlinear**.

Hence, we need a model that can deal with such a problem.

**Idea:** Represent the Hessian using a **Gaussian process** learnt from data.

# Resulting stochastic qN integral and Hessian model

**Summary:** resulting stochastic quasi-Newton integral:

$$y_k = D_k \int_0^1 \widetilde{B}(r_k(\tau))\mathrm{d}\tau + e_k,$$

with the following model for the Hessian

$$\widetilde{B}(\theta) \sim \mathcal{GP}(\mu(\theta), \kappa(\theta, \theta')).$$

The Hessian can now be estimated using tailored GP regression.

Linear transformations (such as an integral or a derivative) of a GP results in a new GP.

# Resulting stochastic optimization algorithm

Standard numerical optimization loop with **non-standard components**.

**Algorithm 1** Stochastic optimization

1. **Initialization** ($k = 1$)
2. **while** *not terminated* **do**
   (a) Compute a **search direction** $p_k$ using the current approximation of the gradient $g_k$ and Hessian $B_k$.
   (b) **Stochastic line search** to find a step length $\alpha_k$ and set
   $$\theta_{k+1} = \theta_k + \alpha_k p_k.$$
   (c) **Update the Hessian model** (tailored GP regression).
   (d) Set $k := k + 1$.
3. **end while**

Curvature information is useful also for stochastic optimization.

# Conclusions

**Message:** The Gaussian process can be used to construct **new models and algorithms** for identification of **nonlinear** dynamical systems.

Motivation via 4 recent applications of the GP for dynamical systems.

Part 1 – Probabilistic modelling of nonlinear dynamical systems

Part 2 – Inferring the state via sequential Monte Carlo

Part 3 – Showed that the GP can be useful also for deriving methods (stochastic optimization)

**Take away:** There are still **many unexplored avenues** when it comes to combining these tools for nonlinear system identification and control!