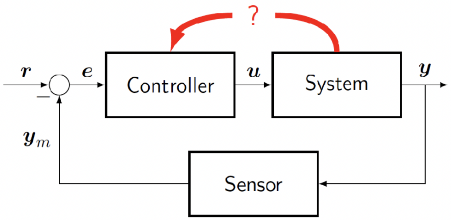
Safety-guarantees with data-driven controls based on Gaussian processes for quadrotors UAVs and multi-agent systems

Leonardo J. Colombo

Centre for Automation and Robotics
Spanish National Research Council

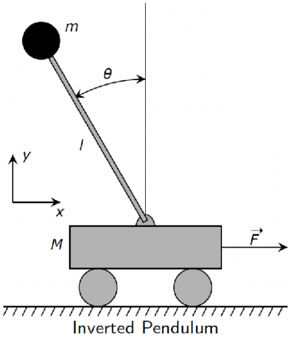
Motivation



Design of controller typically requires a precise model of the system

Motivation

How to model a system?



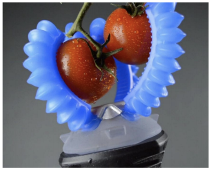
e.g. using Lagrange-equation:

$$(M+m)\ddot{\boldsymbol{x}} - ml\ddot{\theta}\cos\theta + ml\dot{\theta}\sin\theta = F$$

$$l\ddot{\theta} - g\sin\theta = \ddot{\boldsymbol{x}}\cos\theta$$

Motivation

Modeling control systems based on data


[Soft robotics]

[Kuka]

[Bitcraze]

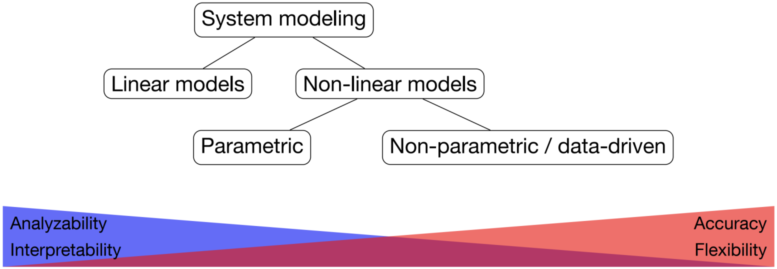
- The parametric modeling is very time-consuming or even unfeasible.
- Limited to the number of parameters to estimate.
- The modern control methods based on data-driven approaches as neural networks, in general, does not present safety guarantees (i.e., a quantification of the uncertainty in the modelization).

Lack of safety guarantees

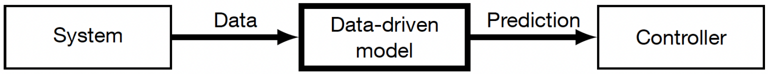
- There is a need to construct algorithms based on data guaranteeing the security of the system.

Overview modeling

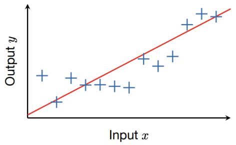
¿Why are data-driven models so interesting?



Data-driven control design



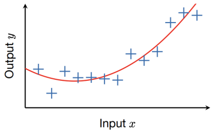
Parametric modeling



$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$

Input x and Output y
Model $\boldsymbol{f}$
Parameter $\boldsymbol{\theta}$

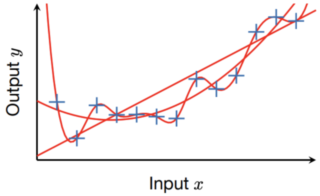
Linear: $y = ax + b$



$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$

Input x and Output y
Model $\boldsymbol{f}$
Parameter $\boldsymbol{\theta}$

Quadratic: $y = ax^2 + bx + c$

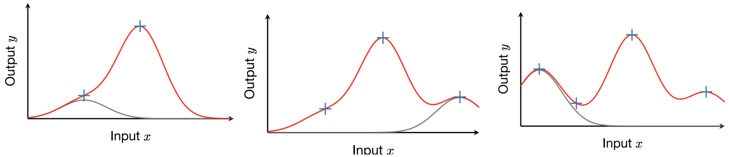


¿Which model is correct?

Number of fixed parameters to estimate $\boldsymbol{\theta}$, with prediction based on the parametric model estimated.

The complexity of the model is limited due to fixed number of parameters

Non-parametric modeling - Data driven modeling



Properties:

- The model scales/adapts with the number of data N.
- Depends on the set of data $\mathcal{D} = \{X, Y\}$,
- The complexity of the model is not limited, the model learns from the data
- Very flexible but often "black-box" behavior" (lost of interpretability).
- Probabilistic interpretation with Bayesian statistics. In particular, Gaussian regression models

Bayesian Statistics

- Combines previous knowledge with data to obtain an improved model through Bayes theorem
- Model $\mathcal{M}$, data $\mathcal{D}$, previous information of the model $\boldsymbol{\theta}$.

$$\underbrace{\Pr(\mathcal{M}|\mathcal{D},\boldsymbol{\theta})}_{\text{Posterior}} \propto \underbrace{\Pr(\mathcal{D}|\mathcal{M},\boldsymbol{\theta})}_{\text{Likelihood}}\underbrace{\Pr(\mathcal{M}|\boldsymbol{\theta})}_{\text{Prior}}$$

- Prior: Previous knowledge without data.
- Posterior: New knowledge with data.
- Likelihood: Likelihood of the data $\mathcal{D}$ with model $\mathcal{M}$ and parameters $\boldsymbol{\theta}$.

  Gaussian processes extends the concept of Gaussian distribution to an infinity collection of variables.

  - This extension permits to think a Gaussian process as a distribution over functions and not only over vectors of random variables.
  - In general, Gaussian processes are defined as a distribution over probability functions.

Overview of the course

- Session 1:

(I) Introduction, motivation and background for the course.

(II) Data-driven modeling with Gaussian Processes.

(III) Learning-based tracking control of Lagrangian systems with uncertain dynamics.

- Session 2:

(I) Online learning-based formation control of multi-agent systems.

(II) Online learning-based flocking control of multi-agent systems.

- Session 3:

(I) Online learning-based trajectory tracking for underactuated vehicles with uncertain dynamics.

(II) Learning-based fault-tolerant control for an hexarotor with model uncertainties.

Multivariate Gaussian distributions

Gaussian distributions can be extended to a finite collection of variables, considering $\mu \in \mathbb{R}^n$ which represents the different means, and a covariance matrix Σ with dimensions $n \times n$ which is symmetric and positive definite.

Definition

For $\vec{x} \in \mathbb{R}^n$, the density function for a multivariate Gaussian distribution is given by

$$P(\vec{x}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\vec{x}-\mu)^T\Sigma^{-1}(\vec{x}-\mu)\right)$$

where $|\Sigma|$ denotes the determinant of Σ.

Multivariate Gaussian distributions

Gaussian distributions can be extended to a finite collection of variables, considering $\mu \in \mathbb{R}^n$ which represents the different means, and a covariance matrix Σ with dimensions $n \times n$ which is symmetric and positive definite.

Definition

For $\vec{x} \in \mathbb{R}^n$, the density function for a multivariate Gaussian distribution is given by

$$P(\vec{x}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\vec{x}-\mu)^T\Sigma^{-1}(\vec{x}-\mu)\right)$$

where $|\Sigma|$ denotes the determinant of Σ.

Gaussian processes (GPs) extend the concept of Gaussian distributions to an infinite collection of variables.

- This extension allows us to think of a Gaussian process as a distribution over functions and not just over vectors of random variables.
- In general terms, Gaussian processes are defined as a distribution over probability functions.

Review on Lyapunov stability

Lyapunov's direct method is concerned with autonomous dynamical systems

$$\dot{x} = f(x), \tag{1}$$

where $f : \mathbb{R}^d \to \mathbb{R}^d$ is a continuous function.

An equilibrium of (1) is defined as a state $x \in \mathbb{R}^d$, such that $f(x) = 0$.

Without loss of generality, we assume that the origin is an equilibrium, i.e., $f(0) = 0$.

Note that this assumption does not pose a relevant restriction since a change of variables can always be used to ensure it.

Lyapunov stability theory investigates the asymptotic behavior of solutions $x(\cdot)$ of dynamical systems (1) in the neighborhood of equilibria.

Review on Lyapunov stability

Definition

The equilibrium point $x = 0$ of (1) is

- stable if, for each $c_1 \geq 0$, there is $c_2 > 0$ such that

$$||x(0)|| < c_2 \implies ||x(t)|| \leq c_1, \, \forall t \in \mathbb{R}_{\geq 0};$$

- asymptotically stable if it is stable and c_2 can be chosen such that

$$||x(0)|| < c_2 \implies \lim_{t \to 0} x(t) = 0,$$

- unstable if it is not stable.

Since we generally cannot determine closed-form solutions for nonlinear dynamical systems, directly using the conditions in the previous definition is not possible.

Therefore, the the behavior of a proxy function $V : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, the so-called Lyapunov candidate, is investigated. The theoretical foundation of this approach, which is commonly referred to as Lyapunov's direct method is given by the following theorem:

Review on Lyapunov stability

Theorem

Let $x = 0$ be an equilibrium point of (1) and $S \subset \mathbb{R}^d$ be a domain containing $x = 0$. Let $V : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ be a continuously differentiable function such that $V(0) = 0$, $V(x) > 0$ for all $x \in \mathbb{R}^d \backslash \{0\}$ and $\dot{V}(x) \leq 0$, for all $x \in S$, then $x = 0$ is stable. Moreover, if $\dot{V}(x) < 0$ for all $x \in S \backslash \{0\}$, then $x = 0$ is asymptotically stable.

Once a Lyapunov candidate has been chosen, this theorem provides stability conditions that can be directly evaluated. However, the results are only qualitative due to the definition of the stability concepts.

In order to obtain quantitative guarantees for the evolution of the system state $x(t)$ in the proximity of an equilibrium, comparison functions can be employed.

Lassalle's invariance principle

The set $M \subset \mathbb{R}^n$ is said to be (positively) invariant set if for all $y \in M$ and $t_0 \geq 0$, we have $s(t, y, t_0) \in M$, $\forall t \geq t_0$ where $s(t, x_0, t_0)$ is the solution of $\dot{x} = f(x)$ at time t starting from x_0 at t_0.

Theorem (Lassalle's principle):

Let $V : \mathbb{R}^n \to \mathbb{R}$ be a positive definite function such that on the compact set

$$\Omega_c = \{x \in \mathbb{R}^n : V(x) \leq c\}$$

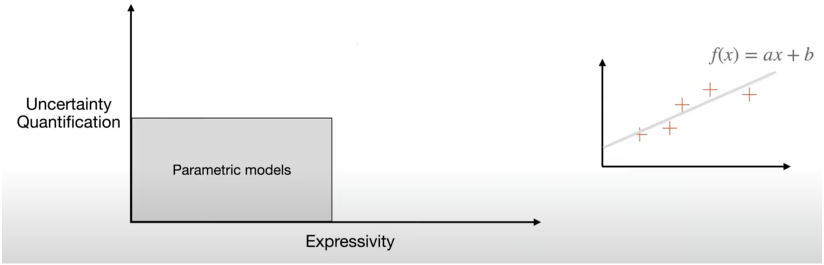
we have $\dot{V}(x) \leq 0$. Define

$$S = \{x \in \Omega_c : \dot{V} = 0\}.$$

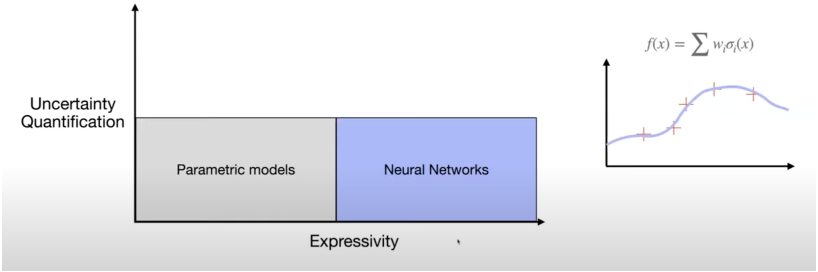
As $t \to \infty$, the trajectory tends to the largest invariant set inside S. In particular, if S contains no invariant sets other than $x = 0$, then 0 is asymptotically stable.

Learning methods

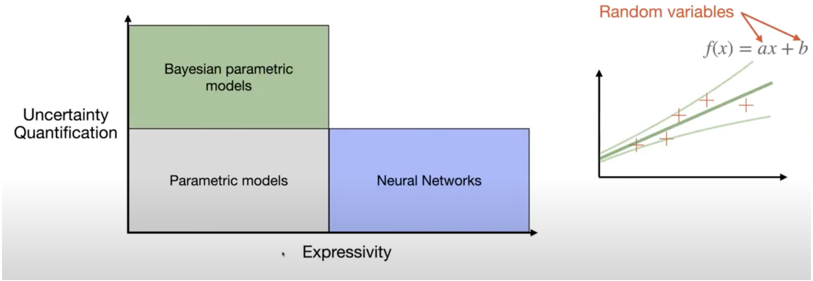
Learning methods



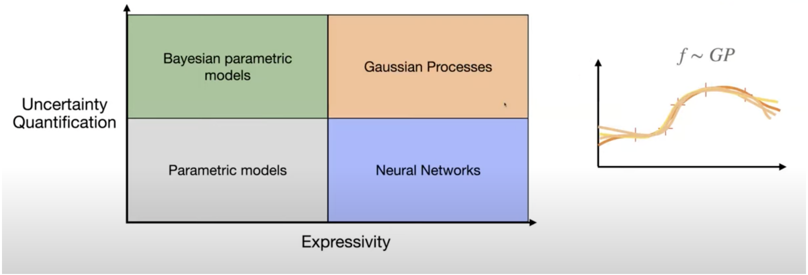
Learning methods



Learning methods



Learning methods



Gaussian Processes: Prior knowledge + nonparametric + uncertainty quantification

Gaussian process

What is a Gaussian Process? It is a Gaussian distribution over function space.

Gaussian Processes are defined by the mean and covariance functions:

$$\boxed{\varphi_{GP}(\xi) \sim \mathcal{GP}\left(m(\xi), k\left(\xi, \xi'\right)\right)}$$

- ξ values in the domain
- (ξ, ξ'),all possible pairs in the domain,
- $m(\xi)$ is the mean function
- $k\left(\xi, \xi'\right)$ is the covariance function.

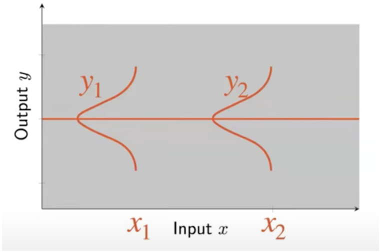
C.E. Rasmussen. *Gaussian processes for machine learning*. *MIT Press*. 2006.

C. Bishop. Pattern Recognition and Machine Learning. *Springer*, 2006.

Gaussian process

In other words, a Gaussian process is a collection of random variables, of which any finite number has a joint Gaussian distribution.

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ m(x_2) \end{bmatrix} \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) \\ k(x_2, x_1) & k(x_2, x_2) \end{bmatrix}\right)$$



Gaussian process

A precise, more explicit formulation is given in the following:

Definition

Let $\mathcal{X}$ be a (multidimensional) index set, and denote by $\{\varphi(\mathbf{x})\}_{\mathbf{x}\in\mathcal{X}}$ a real valued stochastic process over $\mathcal{X}$. Such a process is called Gaussian, if and only if, any finite collection of random variables $\{\varphi(\mathbf{x_1}), \ldots \varphi(\mathbf{x}_\nu)\}$ is a ν-dimensional multivariate Gaussian distribution.

A GP is fully specified by a mean function $m(\mathbf{x})$ and a kernel function $k(\mathbf{x}, \mathbf{x}')$ in function space

$$\begin{aligned} m(\mathbf{x}) =& \mathbb{E}[\varphi(\mathbf{x})], \\ k(\mathbf{x}, \mathbf{x}') =& \mathbb{E}[(\varphi(\mathbf{x}) - m(\mathbf{x}))(\varphi(\mathbf{x}') - m(\mathbf{x}'))], \end{aligned}$$

and thus, we can write

$$\varphi_{GP}(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

Gaussian processes

- To understand the basic functioning of the GP, we have to move our point of view from function to feature space.

- The GP model essentially circumvents the intrinsic limitations in expressiveness of linear modeling by first projecting the inputs $\mathbf{x}, \mathbf{x}'$ into some (high) dimensional feature space using the kernel function $k(\cdot, \cdot)$ before employing a linear model there.

- The so called kernel trick allows for computationally efficient implicit calculations in the feature space.

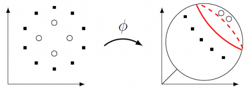


Figure: The mapping ϕ transforms the data points into a feature space where linear regressors can be applied to predict the output.

- Therefore, the essential part in GP modeling concerns the kernel $k(\cdot, \cdot)$. The mean function $m(\cdot)$ is often set to zero in practice, as this simplifies calculation without limiting the expressiveness of the process.

Different kernels for the covariance function

Let the covariance matrix be denoted by $K_{i,j} = k(x_i, x_j; \theta_K)$, where k refers to the function kernel or covariance function, which establish the correlations among different points of the process.

One of the most commonly used kernels in control theory is the square exponential (SE) kernel with the form

$$k_{\mathrm{SE}}\left(x_i, x_j; A, L\right) := A^2 \exp\left\{-\frac{\left(x_i - x_j\right)^2}{2L^2}\right\} + \sigma_n^2 \delta_{ij},$$

where $\theta_K = A, L, \sigma_n \in \mathbb{R}$ to be determined are called hyperparameters.

The hyperparameter A describes the signal variance which determines the average distance of the data-generating function from its mean. The length-scale L defines how far it is needed to move along a particular axis in input space for the function values to become uncorrelated. σ_n is a signal noise. The squared exponential kernel is infinitely differentiable, which means that the GPR exhibits a smooth behavior.

Different kernels for the covariance function

- Apart from few exceptions, the kernel functions can be divided into two classes: *stationary and dot-product kernels*.
- A kernel is called stationary, if the obtained covariances are invariant to translations in the input space.
- A dot-product kernel depends on the non-stationary inner product between inputs.
- Typically, non-constant stationary kernels depend on some kind of distance measure between input samples.
- Part of the most widely used stationary kernel functions in machine learning is the squared exponential kernel and Matérn kernel.

Other kernels: constant and linear kernels

The equation for the constant kernel is given by

$$k(z, z') = \varphi_1^2.$$

This kernel is mostly used in addition to other kernel functions. It depends on a single hyperparameter $\varphi_1 \in \mathbb{R}_{\geq 0}$.

The equation for the linear kernel is given by

$$k(z, z') = z^T z'.$$

The linear kernel is a dot-product kernel and thus, non-stationary. The kernel can be obtained from Bayesian linear regression. The linear kernel is often used in combination with the constant kernel to include a bias.

Other kernels: Polynomial and Matérn kernels

The equation for the polynomial kernel is given by

$$k(z, z') = (z^T z' + \varphi_1^2)^p, \; p \in \mathbb{N}.$$

The polynomial kernel has an additional parameter $p \in \mathbb{N}$, that determines the degree of the polynomial. Since a dot product is contained, the kernel is also non-stationary. The prior variance grows rapidly for $||z|| > 1$ such that the usage for some regression problems is limited. It depends on a single hyperparameter $\varphi_1 \in \mathbb{R}_{\geq 0}$.

The equation for the (stationary) Matérn kernel is given by

$$k(z, z') = \varphi_1^2 \exp\left(-\frac{\sqrt{2\check{p}}\|z - z'\|}{\varphi_2}\right) \frac{p!}{(2p)!} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{\sqrt{8\check{p}}\|z - z'\|}{\varphi_2}\right)^{p-i}$$

with $\check{p} = p + \frac{1}{2}, p \in \mathbb{N}$. The Matérn kernel is a very powerful kernel and presented here with the most common parameterization for $\check{p}$. Functions drawn from a GP model with Matérn kernel are p-times differentiable. This kernel is an *universal kernel* which will be explained next.

Universal Kernels

We are interested in kernels with the following universal approximating property:

Given any prescribed compact set $\mathcal{Z}$ of $\mathcal{X}$, any positive number ϵ and any function $f \in \mathcal{C}^1(\mathcal{Z})$, there is a function $g \in K(\mathcal{Z})$ such that

$$||f - g||_{\mathcal{Z}} \leq \epsilon,$$

where $K(\mathcal{Z}) := \overline{\text{span}}\{k_y : y \in \mathcal{Z}\}$, where k is a kernel function and $k_y : \mathcal{X} \to \mathbb{R}$ is the function defined at every $x \in \mathcal{X}$ by $k_y(x) := k(x, y)$.

The set $K(\mathcal{Z})$ consists of all functions in $\mathcal{C}^1(\mathcal{Z})$ which are uniform limits of functions of the form

$$f := \sum_{j \in \mathbb{N}} c_j k(\cdot, x_j)$$

with $c_j \in \mathbb{R}$ parameters typically obtained by training data.

Intuitively speaking, a GP with an universal kernel can approximate any continuous function arbitrarly exact on a compact set.

Matérn and squared exponential kernels are universal.

Bayesian approach

The Bayesian approach to learning models differs from other methods in two fundamental ways:

- an *a prior* distribution is selected based on prior knowledge,
- predictions about future observations are made by integrating the model's predictions regarding the *a posterior* distribution of the parameters, which is obtained by actualizing the *a prior* distribution with data.

This approach is actually very useful for Gaussian Process, as they have the ability to incorporate new data into the process in a very simple way, updating the *a priori* distribution .That is, as data observations are incorporated into the training, the model will learn and improve, adjusting to the data.

Intuitively, such an adjustment process can be observed in the following figures, where each symbol '$+$' represents the incorporation of new data:

Bayesian approach

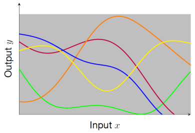


Figure: (I)

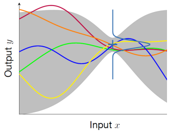


Figure: (II)

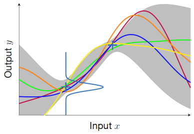


Figure: (III)

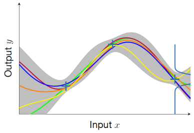


Figure: (IV)

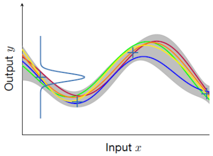


Figure: (V)

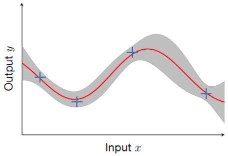
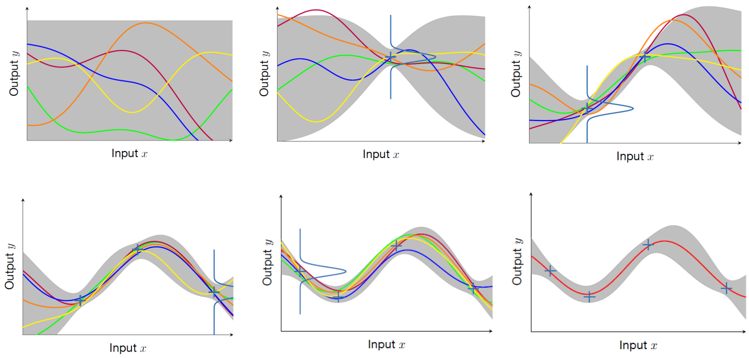


Figure: (VI)

In Figures (I)-(V) can be seen how the process learns from the known data points and adjusts to them as they are added, one by one. The more points it has, the more precise it becomes. Besides, the *a posteriori* distribution shown in (VI), once it is determined, is also a GP.

Training a model by using GPRs: An overview

Bayesian approximation: Update of the prior distribution with new data.



(a) Space of possible (parametric) a priori models. (b) Given a piece of data, the process learns from the data. Note that we not only have a point estimate but also a probabilistic distribution on the estimated data. (f) Posteriori distribution.

The posteriori distribution is again a Gaussian process

Specific kernels generates bounded trajectories (confidence of the model)

Training a model by using GPRs: An overview

Bayesian approach: Update of the prior distribution with new data

- $\mathcal{D} = \{X, Y\}$, $X = \left[\boldsymbol{x}^{\{1\}}, \boldsymbol{x}^{\{2\}}, \ldots, \boldsymbol{x}^{\{m\}}\right] \in \mathbb{R}^{p\times m}$ and $Y = \left[\boldsymbol{y}^{\{1\}}, \boldsymbol{y}^{\{2\}}, \ldots, \boldsymbol{y}^{\{m\}}\right] \in \mathbb{R}^{p\times m}$ are the training data.

- For the test input $\boldsymbol{x}^* \in \mathbb{R}^p$ the prediction for $\boldsymbol{f}(\boldsymbol{x}^*)$ are obtaining by conditioning over the data which gives rise to the posteriori distribution.

$$\mu\left(f_i \,|\, \boldsymbol{x}^*, \mathcal{D}\right) = \boldsymbol{k}\left(\boldsymbol{x}^*, X\right)^\top \left(K + I\sigma^2\right)^{-1} Y_{:,i},$$
$$\text{var}\left(f_i \mid \boldsymbol{x}^*, \mathcal{D}\right) = k\left(\boldsymbol{x}^*, \boldsymbol{x}^*\right) - \boldsymbol{k}\left(\boldsymbol{x}^*, X\right)^\top \quad \left(K + I\sigma^2\right)^{-1} \boldsymbol{k}\left(\boldsymbol{x}^*, X\right)$$

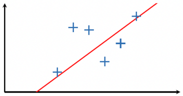
for all $i \in \{1, \ldots, p\}$, where $Y_{:,i}$ denotes the i-th column of Y.

- The kernel k measures the correlation between two entries $(\boldsymbol{x}, \boldsymbol{x}')$. The function $K\colon \mathbb{R}^{p\times m} \times \mathbb{R}^{p\times m} \to \mathbb{R}^{m\times m}$ is the Gram matrix with elements given by $K_{j',j} = k(X_{:,j'}, X_{:,j}) + \delta(j, j')\sigma^2$ for all $j', j \in \{1, \ldots, m\}$ with the delta function $\delta(j, j') = 1$ for all $j = j'$ and zero otherwise.

- The vector valued function $\boldsymbol{k}\colon \mathbb{R}^p \times \mathbb{R}^{p\times m} \to \mathbb{R}^m$, with elements $k_j = \boldsymbol{k}(\boldsymbol{x}^*, X_{:,j})$ for all $j \in \{1, \ldots, m\}$, express the covariance between $\boldsymbol{x}^*$ the training data with entry X.

Training a model by using GPRs: An overview

- The choice of the kernel and the determination of the corresponding hyperparameters can be seen as degrees of freedom of the regression procedure.

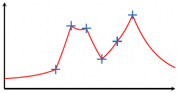


Kernel k:

- Linear

$$\mu(\boldsymbol{x}|\mathcal{D}) = \sum_{j=0}^{N} w_j k(\boldsymbol{x}, X_j)$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^\top \boldsymbol{x}'$$

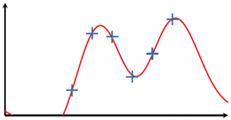


Kernel k:

- Linear
- Matern, continuous

$$\mu(\boldsymbol{x}|\mathcal{D}) = \sum_{j=0}^{N} w_j k(\boldsymbol{x}, X_j)$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|}{2l}\right)$$



Kernel k:

- Linear
- Matern, continuous
- Squared Exponential

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2l^2}\right)$$

$$\mu(\boldsymbol{x}|\mathcal{D}) = \sum_{j=0}^{N} w_j k(\boldsymbol{x}, X_j)$$

Kernel defines the properties of the model

Example 1

In this example, we use three GPRs with the same set of training data

$$X = [1, 3, 5, 7, 9],\ Y = [0, 1, 2, 3, 6] \tag{2}$$

but with different kernels, namely the squared exponential, the linear, and the polynomial kernel.

The Figure shows the different shapes of the regressions with the posterior mean (red), the posterior variance (gray shaded) and the training points (black). Even for this simple data set, the flexibility of the squared exponential kernel is already visible.

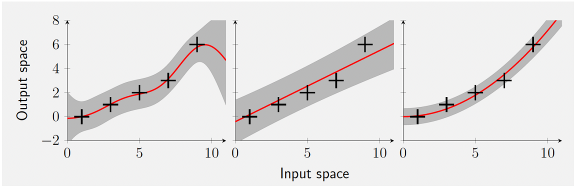


Figure: GPR with different kernels: squared exponential (left), linear (middle) and polynomial with degree 2 (right).

Example 2

We assume a GP with zero mean and a kernel function given by

$$k(z, z') = 0.3679^2 \exp\left(-\frac{(z - z')^2}{2 \cdot 2.7183^2}\right)$$

as prior distribution. The training data set $\mathcal{D}$ is assumed to be

$$X = \begin{bmatrix} 1 & 3 & 6 & 10 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -0.3 & 0.3 & -0.2 \end{bmatrix}^\top,$$

where the output is corrupted by Gaussian noise with $\sigma_n = 0.0498$ standard deviation and the test point is assumed to be $z^* = 5$. The Gram matrix $K(X, X)$ is calculated as

$$K(X, X) = \begin{bmatrix} 0.1378 & 0.1032 & 0.0249 & 0.0006 \\ 0.1032 & 0.1378 & 0.0736 & 0.0049 \\ 0.0249 & 0.0736 & 0.1378 & 0.0458 \\ 0.0006 & 0.0049 & 0.0458 & 0.1378 \end{bmatrix}$$

and the kernel vector $\boldsymbol{k}(z^*, X)$ and $k(z^*, z^*)$ are obtained to be

$$\boldsymbol{k}(z^*, X) = \begin{bmatrix} 0.0458 & 0.1032 & 0.1265 & 0.0249 \end{bmatrix}$$
$$k(z^*, z^*) = 0.1378.$$

Example 2

Finally, we compute the predicted mean and variance for $f_{\mathsf{GP}}(z^*)$

$$\mu(f_{\mathsf{GP}}(z^*)|z^*, \mathcal{D}) = 0.0278, \quad var(f_{\mathsf{GP}}(z^*)|z^*, \mathcal{D}) = 0.0015.$$

The figure shows the prior distribution (left), the posterior distribution with two training points (black crosses) in the middle, and the posterior distribution given the full training set $\mathcal{D}$ (right). The solid red line is the mean function and the gray shaded area indicates the 2σ-standard deviation. Five realizations (dashed lines) visualize the character of the distribution over functions.

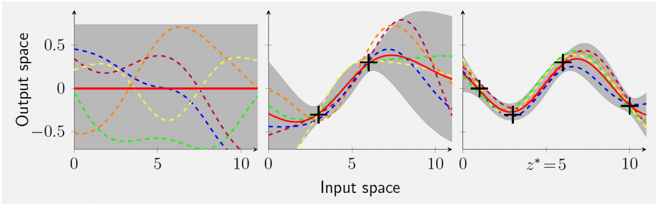
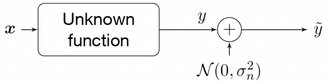


Figure: The prior distribution of a GP is updated with data that leads to the posterior distribution.

Model Training



- Data set $\mathcal{D} = \{\mathbf{x}^i, \tilde{y}^{(i)}\}_{i=1}^N$, with $N \in \mathbb{N}$, $\mathbf{x} \in \mathcal{X}$, $\tilde{y} \in \mathbb{R}$.
- Input data matrix $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$
- Output data matrix $Y^T = [\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_N]$

Output y^* for test input x^* follows a multivariate Gaussian distribution

$$\begin{bmatrix} y^* \\ Y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x^*) \\ m(X) \end{bmatrix}, \begin{bmatrix} k(x^*,x^*) & k^\top(x^*,X) \\ k(x^*,X) & K + \sigma_n^2 I_N \end{bmatrix}\right),$$

with $K_{i,j} = k(X_i, X_j)$.

Prediction based on conditioning

$$y^* \sim \mathcal{N}(\mu(y^*|x^*,\mathcal{D}), \Sigma(y^*|x^*,\mathcal{D}))$$
$$\mu(y^*|x^*,\mathcal{D}) = k(x^*,X)^T(K + I\sigma_n^2)^1 Y$$
$$\Sigma(y^*|x^*,\mathcal{D}) = k(x^*,x^*) - k(x^*,X)^T(K + I\sigma_n^2)^1 k(x^*,X).$$

Example 3

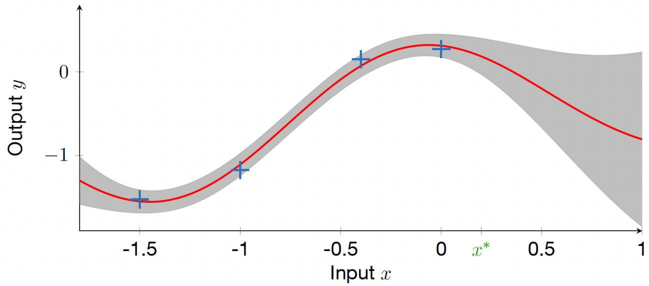
- Example: $N = 4$ data points Y at $X = [-1.5, -1, -0.40, 0]$
- Covariance of input data

$$K(X, X) = \begin{bmatrix} 1.70 & 1.42 & 0.87 & 0.51 \\ 1.42 & 1.70 & 1.34 & 0.97 \\ 0.87 & 1.34 & 1.70 & 1.48 \\ 0.51 & 0.97 & 1.48 & 1.70 \end{bmatrix}$$

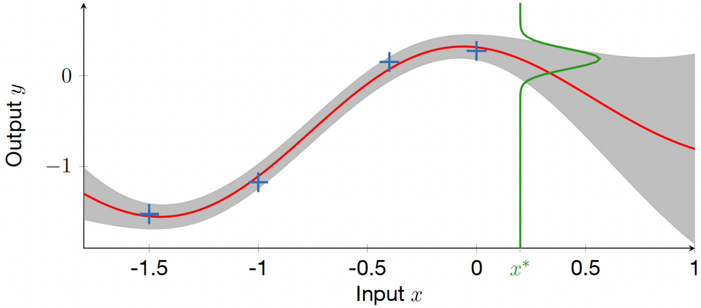
- Covariance with test input $x^* = 0.2$

$$k(x^*, x^*) = 1.70,\ k(x^*, X) = [0.38, 0.79, 1.35, 1.58]$$

Example 3



Example 3



Prediction for y^*: $\mu(y^*) = 0.18,\ \Sigma(y^*) = 0.01.$

Model training

- Model training refers to optimizing the parameters of the mean and the kernel function. We consider a smooth but unknown mapping f with $f(x) = y$.

- Optimal approximation of f, given the input values x_i, is obtained for the dimension-wise globally maximum likelihood of the observations y_i.

- Usually, the hyperparameter optimization is performed using standars gradiendt decent methods to minimize the negative marginal log likelihood, which can be calculated analytically,

$$h_l^* = \text{arg min}_{h_l} - \log p(\{(y_i)_{i=1}^{\nu}\}|\{x_i\}_{i=1}^{\nu}, h_l).$$

- Since an analytic solution of the derivation of $\log p(\{(y_i)_{i=1}^{\nu}\}|\{x_i\}_{i=1}^{\nu}, h_l)$ is impossible, a gradient based optimization algorithm is typically used to minimize the function.

- However, the negative log likelihood is non-convex in general such that there is no guarantee to find the optimum h_l^*, σ_n^*.

- In fact, every local minimum corresponds to a particular interpretation of the data. In the following example, we visualize how the hyperparameters affect the regression.

Reproducing Kernel Hilbert spaces

For computation of the model error we will need to assume some properties of the covariance functions. To stablish such properties we need first to introduce the notion of Reproducing Kernel Hilbert Space (RKHS).

Inner Product and Hilbert spaces

Let H be a vector space over $\mathbb{R}$: $\langle \cdot, \cdot \rangle_H : H \times H \longrightarrow \mathbb{R}$ is an inner product on H if it is:

- i) Linear: $\langle \alpha_1 f_1 + \alpha_2 f_2, g \rangle = \alpha_1 \langle f_1, g \rangle + \alpha_2 \langle f_2, g \rangle \ \forall \alpha_1, \alpha_2 \in \mathbb{R}, f_1, f_2, g \in H$,
- ii) Symmetric: $\langle f, g \rangle_H = \langle g, f \rangle_H \ \forall f, g \in H$, and
- iii) $\langle f, f \rangle_H \geq 0$ and $\langle f, f \rangle_H = 0$ if and only if $f = 0$.

The norm on H is induced by the inner product as $\|f\|_H = \sqrt{\langle f, f \rangle_H}$.

A Hilbert space is an space with an inner product which is complete (i.e., all Cauchy sequence in the space converge to a point in that space) with respect with the norm induced by the inner product.

Reproducing Kernel Hilbert spaces

Kernel functions

Let $\mathcal{X}$ be a non-empty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel if there exists a Hilbert Space H and a feature map $\phi : \mathcal{X} \to \mathcal{V}$ such that $\forall x, x' \in \mathcal{X}$

$$k(x, x') := \langle \phi(x), \phi(x') \rangle_H.$$

It can be noticed how there are almost no conditions on $\mathcal{X}$ (eg, $\mathcal{X}$ itself doesn't need an inner product). Besides, a single kernel can corresponds to several possible features, as in this trivial example for $\mathcal{X} := \mathbb{R}$ equipped with the usual inner product on $\mathbb{R}$

$$\phi_1(x) = x \text{ and } \phi_2(x) = \left[\frac{x}{\sqrt{2}}, \frac{x}{\sqrt{2}}\right]^T.$$

Positive definite functions

If we are given a function of two arguments, $k(x, x')$, how could be determined if it is indeed a valid kernel?

- a) Find a feature map. This is not always obvious. If the feature vector is infinite dimensional, e.g, the exponential quadratic kernel. Besides, we know that the feature map is not unique.
- b) A direct property of the function: Positive definiteness

A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive definite if $\forall n \geq 1$, $\forall (Q_1, ..., Q_n) \in \mathbb{R}^n,\ \forall (x_1, ..., x_n) \in \mathcal{X}^n$,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} Q_i Q_j k(x_i, x_j) \geq 0.$$

The function $k(\cdot, \cdot)$ is strictly positive definite if for mutually distinct x_i,the equality holds **only** when all the Q_i are zero.

The next result shows that kernels are positive definite functions.

Proposition: Let H be a Hilbert space, $\mathcal{X}$ a non-empty set and $\phi : \mathcal{X} \to H$ then $\langle \phi(x), \phi(y) \rangle_H =: k(x, y)$ is positive definite.

Reproducing Kernel Hilbert spaces

The reproducing property

Even though a kernel neither uniquely defines the feature map nor the feature space, one can always construct a canonical feature space, namely the reproducing kernel Hilbert space (RKHS) given a certain kernel.

The reproducing property (kernel trick):

$$\forall x \in \mathcal{X},\ \forall f(\cdot) \in H,\ \langle f(\cdot), k(\cdot, x) \rangle_H = f(x).$$

The feature map of every point is a function: $k(\cdot, x) = \phi(x) \in H$ for any $x \in \mathcal{X}$ and

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_H = \langle k(\cdot, x), k(\cdot, x') \rangle_H$$

An extremely useful property of Hilbert spaces is that they are equivalent to an associated kernel function. This equivalence allows to simply define a kernel, instead of fully defining the associated vector space.

Formally speaking, if a Hilbert space $\mathcal{H}$ is a RKHS, it will have a unique positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which spans the space $\mathcal{H}$.

Reproducing Kernel Hilbert spaces

Definition: Let H be a Hilbert space of $\mathbb{R}$-valued functions on a non-empty set $\mathcal{X}$. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a reproducing kernel of H, and H is a reproducing Hilbert space if:

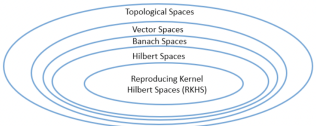
- $\forall x \in \mathcal{X}, k(\cdot, x) \in H$
- $\forall x \in \mathcal{X}, \forall f \in H, \langle f(\cdot), k(\cdot, x)\rangle_H = f(x)$(the reproducing property)

The RKHS is the smallest feature space of a kernel, and can serve as a canonical feature space.

Theorem (Moore-Aronszajn)

Every positive definite kernel k is associated with a unique RKHS $\mathcal{H}$.

By Moore Aronszajn theorem, if $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive definite, there exists a unique RKHS $\mathcal{H} \subset \mathbb{R}^{\dim(\mathcal{X})}$ with reproducing kernel k (recall that feature map is not unique, only the kernel is unique).



Reproducing Kernel Hilbert spaces

Importantly, any function $f_{\mathcal{H}}$ in $\mathcal{H}$ can be represented as a weighted linear sum of this kernel evaluated over the space $\mathcal{H}$, as

$$f_{\mathcal{H}}(\cdot) = \langle f_{\mathcal{H}}(\cdot), k(x, \cdot)\rangle_{\mathcal{H}} = \sum_{i=1}^{n_\phi} \alpha_i k\left(\mathbf{x}_{\text{dat}}^{\{i\}}, \cdot\right),$$

with $\alpha_i \in \mathbb{R}$ for all $i = \{1, \dots, n_\phi\}$, where $n_\phi \in \mathbb{N} \cup \{\infty\}$ is the dimension of the feature space H (H is the Hilbert space s.t. $\mathcal{H} \subset H$).

Thus, the RKHS is equipped with the inner-product

$$\langle f_{\mathcal{H}}, f'_{\mathcal{H}}\rangle_{\mathcal{H}} = \sum_{i=1}^{n_\phi} \sum_{j=1}^{n_\phi} \alpha_i \alpha'_j k(\mathbf{x}_{\text{dat}}^{\{i\}}, \mathbf{x}_{\text{dat}}'^{\{j\}}),$$

with $f'_{\mathcal{H}}(\cdot) = \sum_{j=1}^{n_\phi} \alpha'_j k\left(\mathbf{x}_{\text{dat}}'^{\{j\}}, \cdot\right) \in \mathcal{H}, \alpha'_j \in \mathbb{R}$. Now, the reproducing character manifests as

$$\forall \boldsymbol{z} \in \mathcal{X}, \forall f_{\mathcal{H}} \in \mathcal{H},\ \langle f_{\mathcal{H}}, k(x, \cdot)\rangle_{\mathcal{H}} = f_{\mathcal{H}}(\boldsymbol{z}), \tag{3}$$

in particular

$$k(\boldsymbol{z}, \boldsymbol{z}') = \langle k(\cdot, \boldsymbol{z}), k(\cdot, \boldsymbol{z}')\rangle_{\mathcal{H}}.$$

Reproducing Kernel Hilbert spaces

The RKHS is then defined as

$$\mathcal{H} = \{f_{\mathcal{H}} \colon \mathcal{X} \to \mathbb{R} | \exists \boldsymbol{c} \in H, f_{\mathcal{H}}(\boldsymbol{z}) = \langle \boldsymbol{c}, \phi(\boldsymbol{z}) \rangle_H, \forall \boldsymbol{z} \in \mathcal{X}\}, \tag{4}$$

where $\boldsymbol{\phi}(\boldsymbol{z})$ is the feature map constructing the kernel through $k(\boldsymbol{z}, \boldsymbol{z}') = \langle \phi(\boldsymbol{z}), \phi(\boldsymbol{z}') \rangle_H$.

Ingo Steinwart, Don Hush, and Clint Scovel. *An explicit description of the reproducing kernel Hilbert spaces of Gaussian RBF kernels. IEEE Transactions on Information Theory 52(10), 4635-4643.* 2006.

Example

We want to find the RKHS for the polynomial kernel with degree 2 that is given by

$$k(\boldsymbol{z}, \boldsymbol{z}') = (\boldsymbol{z}^\top \boldsymbol{z}')^2 = (z_1 z_1')^2 + 2(z_1 z_1' z_2 z_2') + (z_2 z_2')^2.$$

for any $\boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^2$. First, we have to find a feature map ϕ such that the kernel corresponds to the inner product $k(\boldsymbol{z}, \mathbf{y}) = \langle \phi(\boldsymbol{z}), \phi(\boldsymbol{y}) \rangle$.

A candidate for the feature map is $\phi(\boldsymbol{z}) = \left[z_1^2, \sqrt{2} z_1 z_2, z_2^2\right]^\top$ because

$$\langle \boldsymbol{\phi}(\boldsymbol{z}), \boldsymbol{\phi}(\boldsymbol{z}') \rangle_{\mathbb{R}^3} = \boldsymbol{\phi}(\boldsymbol{z})^\top \boldsymbol{\phi}(\boldsymbol{z}) = (z_1 z_1')^2 + 2(z_1 z_1' z_2 z_2') + (z_2 z_2')^2 = k(\boldsymbol{z}, \boldsymbol{z}').$$

We know that the RKHS contains all linear combinations of the form

$$\begin{aligned} f_{\mathcal{H}}(\boldsymbol{z}) &= \sum_{i=1}^{3} \alpha_i k\left(\mathbf{x}_{\text{dat}}^{\{i\}}, \boldsymbol{z}\right) = \sum_{i=1}^{3} \alpha_i \langle \boldsymbol{\phi}(\boldsymbol{z}'), \boldsymbol{\phi}(\boldsymbol{z}) \rangle_{\mathbb{R}^3} = \sum_{i=1}^{3} \langle \boldsymbol{c}, \boldsymbol{\phi}(\boldsymbol{z}) \rangle_{\mathbb{R}^3} \\ &= c_1 z_1^2 + c_2 \sqrt{2} z_1 z_2 + c_3 z_2^2, \end{aligned}$$

with $\boldsymbol{\alpha}, \boldsymbol{c}, \mathbf{x}_{\text{dat}}^{\{i\}} \in \mathbb{R}^3$. Therefore, a possible candidate for the RKHS $\mathcal{H}$ is given by

$$\mathcal{H} = \left\{ f_{\mathcal{H}} \colon \mathbb{R}^2 \to \mathbb{R} | f_{\mathcal{H}}(\boldsymbol{z}) = c_1 z_1^2 + c_2 \sqrt{2} z_1 z_2 + c_3 z_2^2, \boldsymbol{c} \in \mathbb{R}^3 \right\} \qquad (5)$$

Example

• Next, it must be checked if the proposed Hilbert space is the related RKHS to the polynomial kernel with degree 2. This is achieved in two steps:

• (i) Checking if the space is a Hilbert space (Homework) and

• (ii) confirming the reproducing property.

• The condition for an RKHS must be fulfilled, i.e., the reproducing property $f_{\mathcal{H}}(\boldsymbol{z}) = \langle f_{\mathcal{H}}(\cdot), k(\cdot, \boldsymbol{z})\rangle_{\mathcal{H}}$. Since we can write

$$\langle f_{\mathcal{H}}(\cdot), k(\cdot, \boldsymbol{z})\rangle_{\mathcal{H}} = \langle \boldsymbol{c}^\top \boldsymbol{\phi}(\cdot), k(\cdot, \boldsymbol{z})\rangle_{\mathcal{H}} = \sum_{i=1}^{3} c_i k(\cdot, \boldsymbol{z}) = \boldsymbol{c}^\top \boldsymbol{\phi}(\boldsymbol{z}) = f_{\mathcal{H}}(\boldsymbol{z}),$$

property (3) is fulfilled and, thus, $\mathcal{H}$ is the RKHS for the polynomial kernel with degree 2.

•Note that, even though the mapping $\boldsymbol{\phi}$ is not unique for the kernel k, the relation of k and the RKHS $\mathcal{H}$ is unique.

Reproducing Kernel Hilbert Space: Summary

- In summary, we investigated the unique relation between the kernel and its RKHS.

- The reproducing property allows us to write the inner-product as a tractable function which implicitly defines a higher (or even infinite) feature dimensional space.

- Next, the RKHS-norm is exploited to determine the error between the prediction of GPR and the actual data-generating function.

Model Error

One of the most interesting properties of GPR is the uncertainty description encoded in the predicted variance.

This uncertainty is beneficial to quantify the error between the actual underlying data generating process and the GPR.

Assume that there is an unknown function $f_{\mathsf{uk}}\colon \mathbb{R}^{n_z} \to \mathbb{R}$ that generates the training data. In detail, the data set $\mathcal{D} = \{X, Y\}$ consists of

$$\begin{aligned} X &= [\mathbf{x}_{\mathsf{dat}}^{\{1\}}, \mathbf{x}_{\mathsf{dat}}^{\{2\}}, \dots, \mathbf{x}_{\mathsf{dat}}^{\{n_{\mathcal{D}}\}}] \in \mathbb{R}^{n_z \times n_{\mathcal{D}}} \\ Y &= [\tilde{y}_{\mathsf{dat}}^{\{1\}}, \tilde{y}_{\mathsf{dat}}^{\{2\}}, \dots, \tilde{y}_{\mathsf{dat}}^{\{n_{\mathcal{D}}\}}]^\top \in \mathbb{R}^{n_{\mathcal{D}}}, \end{aligned} \tag{6}$$

where the data is generated by

$$\tilde{y}_{\mathsf{dat}}^{\{i\}} = f_{\mathsf{uk}}(\mathbf{x}_{\mathsf{dat}}^{\{i\}}) + \nu,\ \nu \sim \mathcal{N}(0, \sigma_n^2) \tag{7}$$

for all $i = \{1, \dots, n_{\mathcal{D}}\}$.

Without any assumptions on f_{uk} it is obviously not possible to quantify the model error. Loosely speaking, the prior distribution of the GPR with kernel k must be suitable to learn the unknown function.

Model error

• More technically, f_{uk} must be an element of the RKHS spanned by the kernel. This leads to the following assumption.

Assumption: The function f_{uk} has a finite RKHS norm with respect to the kernel k, i.e., $\|f_{\mathsf{uk}}\|_{\mathcal{H}} < \infty$, where $\mathcal{H}$ is the RKHS spanned by k.

• This sounds paradoxical as f_{uk} is assumed to be unknown. However, there exist kernels that can approximate any continuous function arbitrarily exact. Thus, for any continuous function, an arbitrarily close function is element of the RKHS of an universal kernel.

Next, we assume that a GPR is trained with the data set (6) and the assumption holds. The work

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. *Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. IEEE Transactions on Information Theory 58(5), 3250–3265.* 2012.

derives an upper bound for samples of the GPR on a compact set with a specific probability.

Model Error: Information-theoretical approach

Theorem (Srinivas et.al)

Given the assumption, the model error $\Delta \in \mathbb{R}$

$$\Delta = |\mu(f_{GP}|\boldsymbol{z}, \mathcal{D}) - f_{uk}(\boldsymbol{z})| \tag{8}$$

is bounded for all $\boldsymbol{z}$ on a compact set $\Omega \subset \mathbb{R}^{n_z}$ with a probability of at least $\delta \in (0,1)$ by

$$P\left\{\forall \boldsymbol{z} \in \Omega,\ \Delta \leq |\beta Var^{\frac{1}{2}}(f_{GP}|\boldsymbol{z}, \mathcal{D})|\right\} \geq \delta, \tag{9}$$

where $\beta \in \mathbb{R}$ is defined as

$$\beta = \sqrt{2||f_{uk}||_k^2 + 300\gamma_{max} \ln^3\left(\frac{n_{\mathcal{D}}+1}{1-\delta}\right)}. \tag{10}$$

The variable $\gamma_{max} \in \mathbb{R}$ is the maximum of the information gain

$$\gamma_{max} = \max_{\mathbf{x}_{dat}^{\{1\}}, \ldots, \mathbf{x}_{dat}^{\{n_{\mathcal{D}}+1\}} \in \Omega} \frac{1}{2} \log|I_{n_{\mathcal{D}}+1} + \sigma_n^{-2} K(\boldsymbol{z}, \boldsymbol{z}')| \tag{11}$$

with Gram matrix $K(\boldsymbol{z}, \boldsymbol{z}')$ and the input elements $\boldsymbol{z}, \boldsymbol{z}' \in \{\mathbf{x}_{dat}^{\{1\}}, \ldots, \mathbf{x}_{dat}^{\{n_{\mathcal{D}}+1\}}\}$.

Gaussian Process Regression: Summary

Idea:

- The posteriori distribution is Gaussian over the functional space.
- Update with new data.

Adventages:

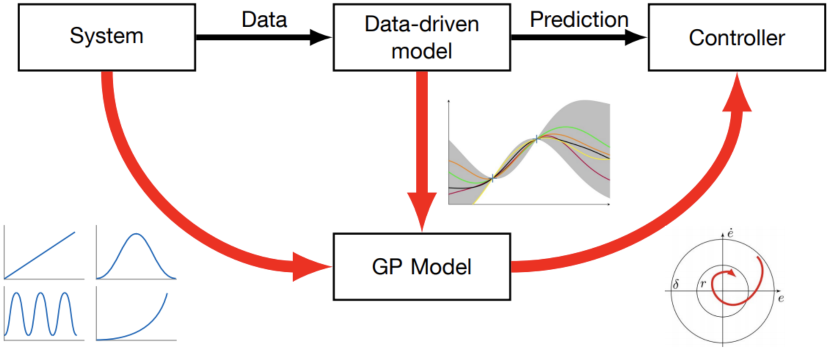
- Fully probabilistic.
- Robust against Gaussian noise.
- Explicit description of the uncertainty:Not only do I know the output data for the prediction, but, how I can vary it. This gives us a mitigation of the model uncertainty.

Disadventages:

- Computationally expensive

Prediction with knowledge of model uncertainty: **Safety guarantees**.

Data-driven control: Summary



What next? $\rightarrow$ Feedback control

$$\underbrace{\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial \mathcal{L}}{\partial \boldsymbol{q}}} = \underbrace{\quad \boldsymbol{u} \quad} + \underbrace{\quad \boldsymbol{f} \quad}$$

$$\mathbf{q} \in \mathbb{R}^n, \;\; \mathcal{L} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}.$$

Equivalent expression:

$$H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}) - f(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \mathbf{u}(t).$$

Classical approximation:

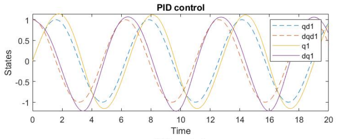
$$\mathbf{u}(t) = \hat{H}(\mathbf{q})\ddot{\mathbf{q}}_d + \hat{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \hat{g}(\mathbf{q}) - K_d\dot{e} - K_p e.$$

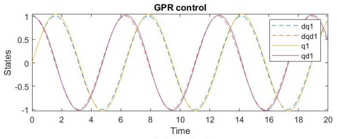
- For $\hat{H} = H$, $\hat{C} = C$, $\hat{g} = g$, $f = 0$, one can guarantee asymptotic stability.
- Another case: increase K_d, K_p to minimize the tracking error.
- But: (I) big errors in presence of noise, (II) saturation of the actuators, (III) the stability is not guaranteed/ dangerous.

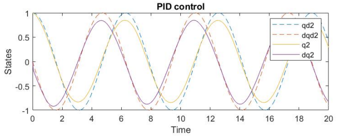
Feedback control based on data

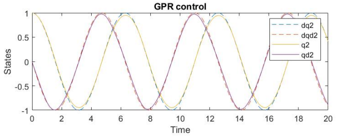
Control law: Consider $\tilde{\tau}(p) = \tilde{H}(q)\ddot{q} + \tilde{C}(q,\dot{q})\dot{q} + \tilde{g}(q) - f(p)$ with $\tilde{H} = H - \hat{H}$, $\tilde{C} = C - \tilde{C}$, $\tilde{g} = g - \tilde{g}$.

$$u(t) = \hat{H}\ddot{q}_d + \hat{C}\dot{q}_d + \hat{g} + \mu(\tilde{\tau} \mid \mathcal{D}) - K_d\dot{e} - K_p e.$$







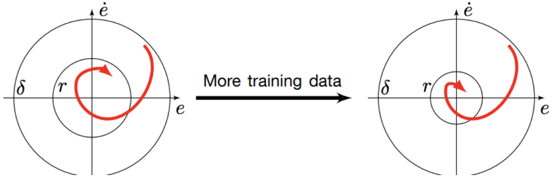


Safety guarantees

Boundedness of the tracking error: The data-driven controller guarantees that, for $||\dot{e}^T(t_0), e^T(t_0)|| < \delta$, we have that

$$P\{||\dot{e}^T(t_0), e^T(t_0)|| \leq r, \forall t \geq t_0 + T(\delta)\} \geq \rho$$

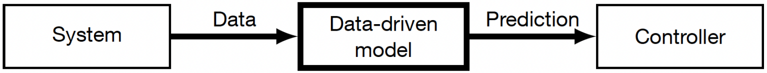
with $r = r(||u||_{\mathcal{H}}, \mathcal{D})$.



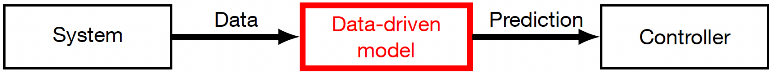
This result shows the evaluation between the tracking error bound, the model uncertainty and the feedback gains.

T. Beckers, L. Colombo, M. Morari, G. Pappas. *Learning-based Balancing of Model-based and Feedback Control for Second-order Mechanical Systems. 2022 IEEE 61th Annual Conference on Decision and Control (CDC).*

Data-driven control design



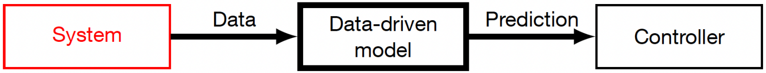
Data-driven control design



Outlook:

- Benefits of data driven models

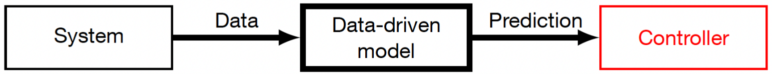
Data-driven control design



Outlook:

- Benefits of data driven models
- Training and prior knowledge

Data-driven control design



Outlook:

- Benefits of data driven models
- Training and prior knowledge
- Data driven based control

Position control and trajectory tracking

Consider the controlled Euler-Lagrange equations

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau \tag{12}$$

where $\tau \in \mathbb{R}^n$ denotes the torques for the system.

Let q_d the trajectory we wish to track. If we have a perfect model of the robot and $q(0) = q_d(0)$, $\dot{q}(0) = \dot{q}_d(0)$, then we may solve our problem by choosing

$$\tau = H(q_d)\ddot{q}_d + C(q_d,\dot{q}_d)\dot{q}_d + g(q_d).$$

Since both q and q_d satisfy the same differential equation and have the same initial conditions , it follows from uniqueness of solutions that $q(t) = q_d(t)$, $\forall\ t \geq 0$.

This is an example of open-loop control law: The current state of the robot is not used in choosing the control inputs.

This strategy is not very robust. If $q(0) \neq q_d(0)$, then the open-loop control law will never correct for this error.

Position control and trajectory tracking

There are no guarantees that if our starting configuration is near the desired initial configuration that the trajectory of the robot will never stay near the desired trajectory. For this reason, we introduce feedback into our control law.

This feedback must be chosen so that the actual robot trajectory converges to the desired trajectory. In particular, if our trajectory is a single set-point, the closed-loop system should be asymptotically stable about the desired set point.

Computed-torque control: Consider the following refinement of the open-loop control law: Given the current position and velocity of the robot, cancel all nonlinearities and apply exactly the torque needed to overcome the inertia of the actuatior,

$$\tau = H(q)\ddot{q}_d + C(q, \dot{q})\dot{q} + g(q).$$

Substituting this control law into the dynamic equation of the robot we see that $\ddot{q} = \ddot{q}_d$ since $H(q)$ is positive definite.

Computed-torque control

Hence, if the initial position and velocity of the robot matches the desired position and velocity, the robot will follow the desired trajectory.

This control law **will not** correct for any initial condition errors which are present. Next, we improve the control law with a feedback and consider the computed-torque control

$$\tau = H(q)(\ddot{q}_d - k_v \dot{e} - k_p e) + C(q, \dot{q})\dot{q} + g(q),$$

where $e = q - q_d$, and k_v, k_p are constant gain matrices.

Substituting in (68), the error dynamics can be written as

$$H(q)(\ddot{e} + k_v \dot{e} + k_p e) = 0$$

and since $H(q)$ is positive definite

$$\ddot{e} + k_v \dot{e} + k_p e = 0.$$

Computed-torque control

The computed torque control law can be written in terms of two components as:

$$\tau = \underbrace{H(q)\ddot{q}_d + C(q,\dot{q})\dot{q} + g(q)}_{feedforward} + \underbrace{H(q)(-K_v\dot{e} - k_p e)}_{feedback}$$

Feedforward component: Necessary torque to drive the system along its nominal path.

Feedback term: Correction torques to reduce any error in the trajectory of the robot.

Proportional derivative (PD) control

In its simple form the PD control has the form

$$\tau = -k_v \dot{e} - k_p e \tag{13}$$

with k_p, k_v positive definite matrices and $e = q - q_d$.

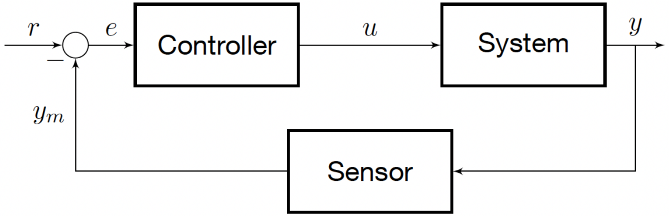
If $\dot{q}_d = 0$ and $k_v, k_p > 0$ the control law applied to the system renders the equilibrium $q = q_d$ asymptotically stable.

Since we are interested in tracking, we consider a modified version of the PD control law:

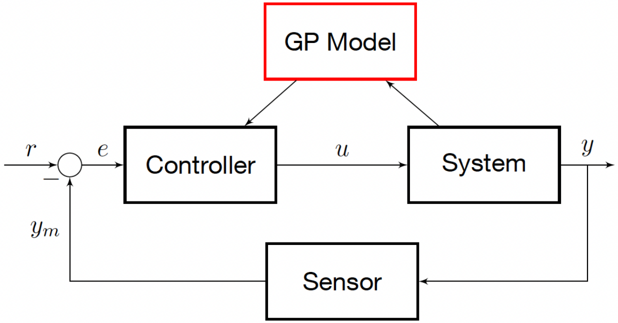
$$\tau = H(q)\ddot{q}_d + C(q, \dot{q})\dot{q}_d + g(q) = k_v \dot{e} - k_p e \tag{14}$$

The control law (14) applied to the controlled Euler-Lagrange equations results in asymptotically stable tracking if $k_v, k_p > 0$.

Data-driven control



Data-driven control



Goal: Improved performance with stability guarantees

Uncertain Lagrangian systems

$$\underbrace{\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial \mathcal{L}}{\partial \boldsymbol{q}}} = \underbrace{\quad \boldsymbol{u} \quad} + \underbrace{\quad \boldsymbol{f} \quad}$$

$$\mathbf{q} \in \mathbb{R}^n, \;\; \mathcal{L} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}.$$

Equivalent expression:

$$H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) - \boldsymbol{f}_{\boldsymbol{u}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \mathbf{u}(t).$$

where

- generalized coordinates: $\mathbf{q} \in \mathbb{R}^n$, control input $\mathbf{u} \in \mathbb{R}^n$,
- $\mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^T H(\mathbf{q})\dot{\mathbf{q}}$, $H(\mathbf{q}) : \mathbb{R}^n \to \mathbb{R}^{n \times n}$ the inertial matrix,
- the Coriolis matrix $C(\boldsymbol{q}; \dot{\boldsymbol{q}}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n \times n}$,
- $\mathbf{g}(\boldsymbol{q}) : \mathbb{R}^n \to \mathbb{R}$, $\mathbf{g}(\boldsymbol{q}) := -\frac{\partial \mathcal{V}}{\partial \mathbf{q}}$.

Classical Approach

Classical approximation:

$$\mathbf{u}(t) = \underbrace{\hat{H}(\mathbf{q})\ddot{\mathbf{q}}_d + \hat{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}_d + \hat{\mathbf{g}}(\mathbf{q})}_{\text{Estimated model}} - \underbrace{K_d\dot{\mathbf{e}} - K_p\mathbf{e}.}_{\text{Feedback control}}$$

where $\hat{H}$, $\hat{C}$ y $\hat{\mathbf{g}}$ are the parametric estimates of H, C and g respectively. The terms K_d y K_p are the feedback gains, where K_p is the proportional gain and K_d is the derivative gain and $\mathbf{e} = \mathbf{q_d} - \mathbf{q}$ is the error.

- For $\hat{H} = H$, $\hat{C} = C$, $\hat{\mathbf{g}} = \mathbf{g}$, $\boldsymbol{f_u} = 0$, one can guarantee asymptotic stability.
- Another case: increase K_d, K_p to minimize the tracking error.
- But: (I) Large errors in presence of noise, (II) saturation of the actuators, (III) the stability is not guaranteed/ dangerous.

Murray, R. M., Li, Z., Sastry, S. S., & Sastry, S. S. (1994). A mathematical introduction to robotic manipulation. CRC press.

Assumptions on the disturbance

• Let us consider a fully actuated system. In particular, we consider the controlled Euler-Lagrange equations, where the control now is decomposed as

$$H(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{u} + \boldsymbol{u_d}, \tag{15}$$

where $\boldsymbol{u} \in \mathbb{R}^n$ is the action of control and $\boldsymbol{u_d} \in \mathbb{R}^n$ is the effect of an unknown external force.

Assumption 1: $\boldsymbol{u_d}$ can be parameterized as $\boldsymbol{u_d} = \boldsymbol{f_u}(\boldsymbol{p})$ with $\boldsymbol{p} = [\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}]$ where $\boldsymbol{f_u} : \mathbb{R}^{3n} \to \mathbb{R}^n$ is a continuous function.

• Hence, equation (15) can be written as

$$H(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{f_u}(\boldsymbol{p}) = u. \tag{16}$$

• The assumption restricts $\boldsymbol{f_u}$ to be not directly time dependent which holds in many application scenarios. For example, the common unknown dynamics in robotic systems, i.e. Columb and viscous friction.

• The kinetic energy in the EL equation is of the form $\mathcal{K}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T H(\boldsymbol{q})\dot{\boldsymbol{q}}$ where $H(\boldsymbol{q})\colon \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the symmetric and positive definite generalized inertia matrix.

Assumptions on the estimates

- Property 1: The non-unique matrix $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is always defined such that $\dot{H}(\boldsymbol{q}) - 2C(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n \times n}$ is skew-symmetric $\forall \boldsymbol{q}, \dot{\boldsymbol{q}} \in \mathbb{R}^n$.

- Consider the EL system with the unknown dynamics $\boldsymbol{f_u}$. If a priori knowledge of the system is available, a hybrid learning approach can be used which is a combination of a parametric and a data-driven model.

- We consider the estimated model to be given by

$$\hat{\boldsymbol{u}}(t) = \hat{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \hat{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{g}}(\boldsymbol{q}), \tag{17}$$

where $\hat{H}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}, \hat{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n \times n}$ and $\hat{\boldsymbol{g}}(\boldsymbol{q}) \in \mathbb{R}^n$ are estimates of the true values which also satisfy the skew-symmetry property given before. Furthermore, the estimates must fulfill the following property.

Property 2. Structure of the estimates:

There exist $h_1, h_2, k_C \in \mathbb{R}_{>0}$ with $h_1 \|\mathbf{x}\|^2 \leq \mathbf{x}^\top \hat{H}(\boldsymbol{q}) \mathbf{x} \leq h_2 \|\mathbf{x}\|^2$, and $\|\hat{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\| \leq k_C \|\dot{\boldsymbol{q}}\|$ where $\hat{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{q}' = \hat{C}(\boldsymbol{q}, \boldsymbol{q}')\dot{\boldsymbol{q}}$ for all $\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{q}', \mathbf{x} \in \mathbb{R}^n$.

Assumptions on the estimates

• The identification of these estimates while guaranteeing Property 1 and Property 2 can be achieved following the identification procedures.

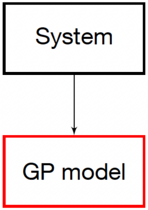
Aström, K.J. and Eykhoff, P. (1971). System identification-a survey. Automatica Vol 7(2), 123-162

• Note that "structure of the estimates" (Property 2) is required for the estimates only and not for the true system.

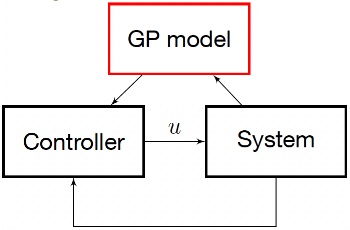
• Without prior knowledge of the system, the estimates are set to $\hat{H} = I, \hat{C} = 0, \hat{\boldsymbol{g}} = \mathbf{0}$.

GP based computed torque

1. Step:
Learning the system dynamics



2. Step:
Using the GP model in the controller



Learning

• After the parametric model is selected, a GP is trained with m data pairs $\mathcal{D} = \{\boldsymbol{p}^{\{i\}}, \tilde{\boldsymbol{\tau}}^{\{i\}}\}_{i=1}^{m}$ of the system consisting of $\boldsymbol{p} = \left[\ddot{\boldsymbol{q}}^\top, \dot{\boldsymbol{q}}^\top, \boldsymbol{q}^\top\right]^\top \in \mathbb{R}^{3n}$ as input data, and the difference between the real system dynamics and the estimated model as output data.

• This residual dynamic is denoted by

$$\tilde{\boldsymbol{\tau}}(\boldsymbol{p}) = \tilde{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \tilde{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \tilde{\boldsymbol{g}}(\boldsymbol{q}) - \boldsymbol{f}_{\boldsymbol{u}}(\boldsymbol{p}), \tag{18}$$

with $\tilde{H} = H - \hat{H}$, $\tilde{C} = C - \hat{C}$ and $\tilde{\boldsymbol{g}} = \boldsymbol{g} - \hat{\boldsymbol{g}}$.

• For the generation of training data, the real system can be operated by an arbitrary controller as shown in the following Figure.

• The only condition is that a finite sequence of training data of the system can be collected whereas stability is not necessarily required.

Model error

For the computation of the model error, we assume the following for the covariance function of the GP.

Assumption 2: The covariance function k is chosen such that the functions $\tilde{\tau}_1, \dots, \tilde{\tau}_n$ have a bounded reproducing kernel Hilbert Space (RKHS) norm on any compact set $\Omega \subset \mathbb{R}^{3n}$, i.e. $\|\tilde{\tau}_i\|_k < \infty$ for all $i = 1, \dots, n$.

• Assumption 2 requires that the covariance function must be selected in such a way that the residual $\tilde{\boldsymbol{\tau}}(\boldsymbol{p})$ is an element of the associated RKHS. This sounds paradoxical since the residual is unknown. However, there exist covariance functions (universal functions), which can approximate any continuous function arbitrarily precisely on a compact set.

• Therefore, any smooth residual dynamics can be covered by a universal covariance function, i.e. this assumption is not restrictive.

Model error

An upper bound for the distance between the mean prediction $\boldsymbol{\mu}(\tilde{\boldsymbol{\tau}})$ of the GPR and the true function $\tilde{\boldsymbol{\tau}}$ is given in

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. *Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. IEEE Transactions on Information Theory 58(5), 3250–3265.* 2012.

and is extended for multidimensional functions in the following lemma.

Lemma 1: Consider a Lagrangian system and a trained GP satisfying Assumption 2. The model error is bounded by

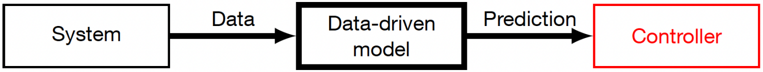
$$\mathsf{P}\left\{\|\,\boldsymbol{\mu}(\tilde{\boldsymbol{\tau}}|\boldsymbol{p},\mathcal{D})-\tilde{\boldsymbol{\tau}}(\boldsymbol{p})\|\leq\|\boldsymbol{\beta}^{\top}\Sigma^{\frac{1}{2}}(\tilde{\boldsymbol{\tau}}|\boldsymbol{p},\mathcal{D})\|\right\}\geq\delta \tag{19}$$

for $\boldsymbol{p}\in\Omega\subset\mathbb{R}^{3n}$ with $\delta\in(0,1),\boldsymbol{\beta},\boldsymbol{\gamma}\in\mathbb{R}^{n}$ and

$$\beta_j=\sqrt{2\|\tilde{\tau}_j\|_k^2+300\gamma_j\ln^3\left(\frac{m+1}{1-\delta^{1/n}}\right)}$$

$$\gamma_j=\max_{\boldsymbol{p}^{\{1\}},\dots,\boldsymbol{p}^{\{m+1\}}\in\Omega}\frac{1}{2}\log|I+\sigma_j^{-2}K(\mathbf{x},\mathbf{x}')|,\text{ with }\mathbf{x},\mathbf{x}'\in\left\{\boldsymbol{p}^{\{1\}},\dots,\boldsymbol{p}^{\{m+1\}}\right\}.$$

Data-driven control design



Outlook:

- Benefits of data driven models: Gaussian Process Regression
- Training and prior knowledge
- Data-driven based control

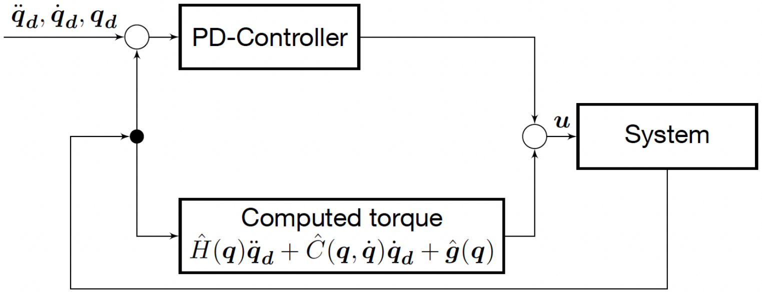
Tracking control with GPR

The goal of tracking control is to follow a desired trajectory with the closed loop system. We start with the following assumption for the desired trajectory.

Assumption 3: The desired state trajectory is bounded by $||\mathbf{q}_d|| \leq \bar{q}_d$, $||\dot{\mathbf{q}}_d|| \leq \dot{\bar{q}}_d$ with $\bar{q}_d, \dot{\bar{q}}_d \in \mathbb{R}_{\geq 0}$, and $\mathbf{q}_d \in \mathbb{R}^n$.

A bounded reference motion trajectories is a very natural assumption and does not pose any restriction in practice.

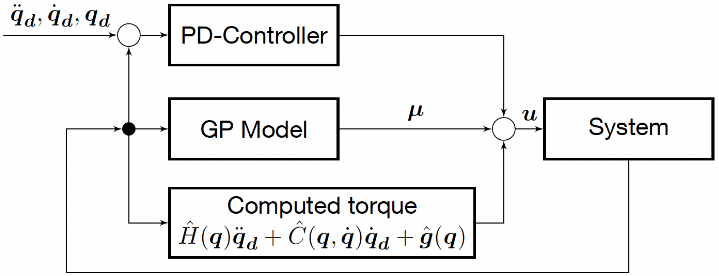
Tracking control with GPR



$\ddot{\boldsymbol{q}}_{\boldsymbol{d}}, \dot{\boldsymbol{q}}_{\boldsymbol{d}}, \boldsymbol{q}_{\boldsymbol{d}}$
PD-Controller
$\boldsymbol{u}$
System
Computed torque
$\hat{H}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \hat{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_{\boldsymbol{d}} + \hat{\boldsymbol{g}}(\boldsymbol{q})$

$$\boldsymbol{u}(t) = \underbrace{\hat{H}(\boldsymbol{q})\ddot{\mathbf{q}}_{\mathbf{d}} + \hat{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\mathbf{q}}_{\mathbf{d}} + \hat{\boldsymbol{g}}(\boldsymbol{q})}_{\text{computed torque}} - \underbrace{K_d\dot{\boldsymbol{e}} - K_p\boldsymbol{e}}_{\text{PD-controller}}$$

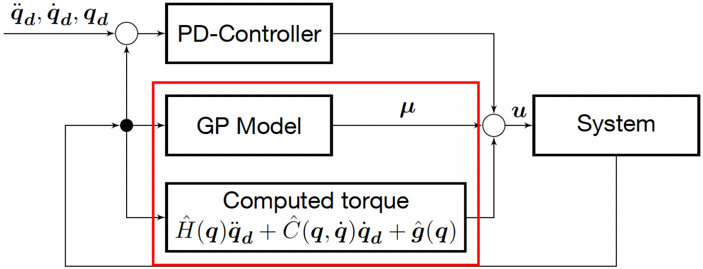
Tracking control with GPR



$\ddot{\boldsymbol{q}}_d, \dot{\boldsymbol{q}}_d, \boldsymbol{q}_d$
PD-Controller
GP Model
$\boldsymbol{\mu}$
$\boldsymbol{u}$
System
Computed torque
$\hat{H}(\boldsymbol{q})\ddot{\boldsymbol{q}}_d + \hat{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_d + \hat{\boldsymbol{g}}(\boldsymbol{q})$

$$\boldsymbol{u}(t) = \underbrace{\hat{H}(\boldsymbol{q})\ddot{\mathbf{q}}_{\mathbf{d}} + \hat{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\mathbf{q}}_{\mathbf{d}} + \hat{\boldsymbol{g}}(\boldsymbol{q})}_{\text{computed torque}} + \underbrace{\boldsymbol{\mu}(\tilde{\boldsymbol{\tau}}|\boldsymbol{p}, \mathcal{D})}_{\text{GP-model}} - \underbrace{K_d\dot{\boldsymbol{e}} - K_p\boldsymbol{e}}_{\text{PD-controller}}$$

Tracking control with GPR


$\ddot{q}_d, \dot{q}_d, q_d$
PD-Controller
GP Model
μ
u
System
Computed torque
$\hat{H}(q)\ddot{q}_d + \hat{C}(q,\dot{q})\dot{q}_d + \hat{g}(q)$

$$u(t) = \underbrace{\hat{H}(q)\ddot{\mathbf{q}}_{\mathbf{d}} + \hat{C}(q,\dot{q})\dot{\mathbf{q}}_{\mathbf{d}} + \hat{g}(q)}_{\text{computed torque}} + \underbrace{\mu(\tilde{\tau}|p,\mathcal{D})}_{\text{GP-model}} - \underbrace{K_d\dot{e} - K_p e}_{\text{PD-controller}}$$

Tracking control with GPR

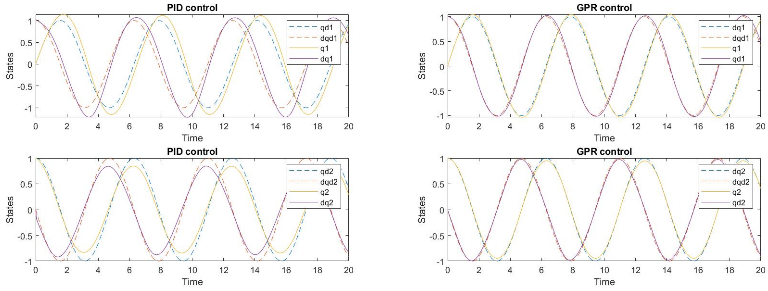


Figure: Controlled trajectory tracking (classic PID) for the two-link planar robotic arm.

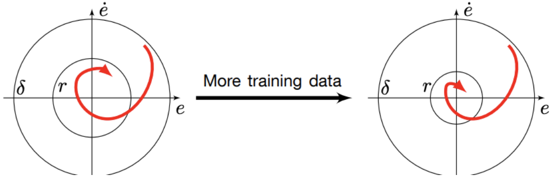
T. Beckers, D. Kulic, and S. Hirche, *Stable Gaussian Process based Tracking Control of Euler-Lagrange Systems. Automatica 103, pp. 390–397*, 2019.

Safety guarantees

Boundedness of the tracking error: The data-driven controller guarantees that, for $||\dot{e}^T(t_0), e^T(t_0)|| < \delta$, we have that

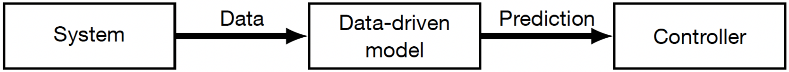
$$P\{||\dot{e}^T(t_0), e^T(t_0)|| \leq r, \forall t \geq t_0 + T(\delta)\} \geq \rho$$

with $r = r(||\tau||_{\mathcal{H}}, \mathcal{D})$.

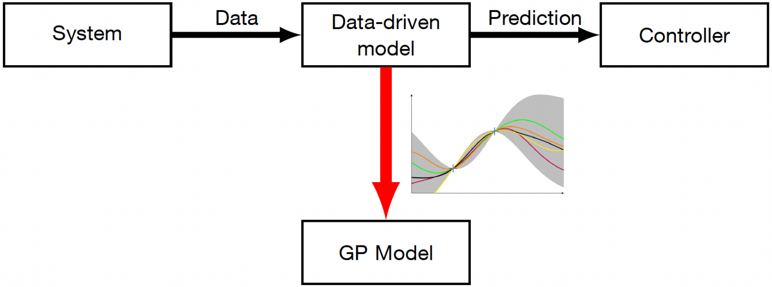


T. Beckers, L. Colombo, M. Morari, G. Pappas. *Learning-based Balancing of Model-based and Feedback Control for Second-order Mechanical Systems. 2022 IEEE 61th Annual Conference on Decision and Control (CDC).*

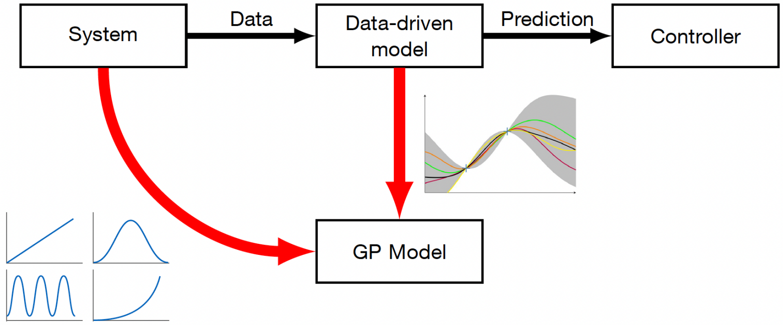
Tracking control with GPR: Summary



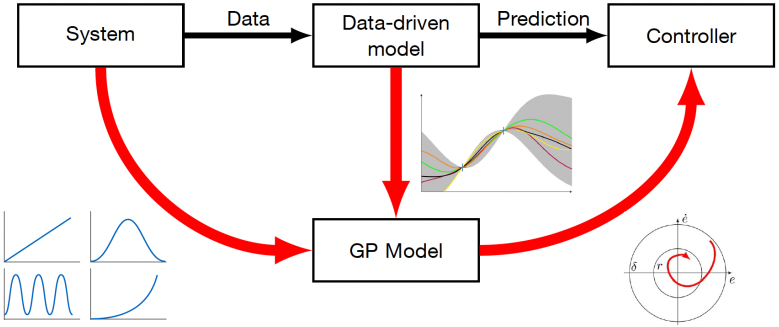
Tracking control with GPR: Summary



Tracking control with GPR: Summary



Tracking control with GPR: Summary



System
Data
Data-driven model
Prediction
Controller
GP Model
$\dot{e}$
δ
r
e