# Learning-Based Predictive Control

# Chapter 7
# Stochastic Learning-Based MPC

Dr. Lukas Hewing

ETH Zurich

The Exploration Company

2023

# Learning Objectives

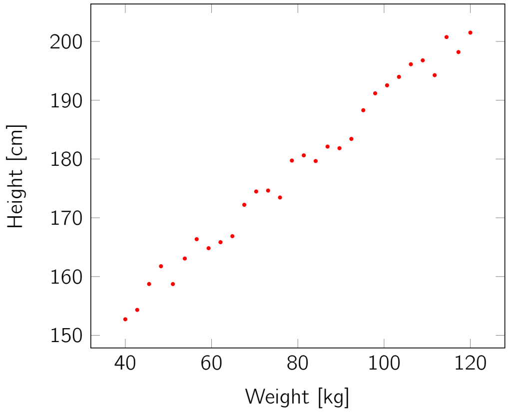
- Review Bayesian linear regression
- Derive multivariate intervals for Gaussian random variables
- Application to safety guarantees for parametric uncertainty
- Review Gaussian process regression
- Application to autonomous racing

# Outline

1. Bayesian Linear Regression for Dynamics Learning
2. Robustness in Probability
3. Non-parametric Bayesian Regression: Gaussian Processes
4. Application to Autonomous Racing

# Example

We want to predict the height $y$ of a person given its weight $x$.



# Model

**Assumption: Linear Model**

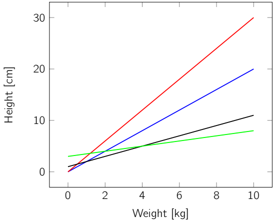
Consider the model

$$f(x) = \phi(x)\theta$$
$$y = f(x) + w$$

where $\phi(x)$ is the feature matrix, $\theta$ is a vector of weights, and $w \sim \mathcal{N}(0, \Sigma^w)$ is independent and identically distributed (iid) noise.

*Example:*

$$\phi(x) = [x \ 1]$$
$$\theta = [\theta_1, \ \theta_2]^\top$$

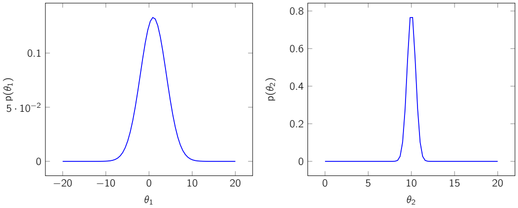


# Prior

**Prior**

A prior expresses one's beliefs about a quantity before some evidence is taken into account.

In the following, we will consider a Gaussian distributed prior on the parameter vector $\theta$, i.e. $\theta \sim \mathcal{N}(\mu^\theta, \Sigma^\theta)$, which allows the derivation of closed-form expression.




*Uninformative prior:* Assume $\Sigma_p = aI$. If $a \to \infty$, we obtain an uninformative prior, which still denotes an initial knowledge.

# Joint Distribution

$$y = \phi(x)\theta + w$$

We therefore have two independent Gaussian random variables $w$ and $\theta$ and

$$\begin{bmatrix} y \\ \theta \end{bmatrix} = \begin{bmatrix} I & \phi(x) \\ 0 & I \end{bmatrix} \begin{bmatrix} w \\ \theta \end{bmatrix}, \text{ with } \begin{bmatrix} w \\ \theta \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} 0 \\ \mu^\theta \end{bmatrix}, \begin{bmatrix} \Sigma^w & 0 \\ 0 & \Sigma^\theta \end{bmatrix} \right)$$

which, as a linear transformation of Gaussian random variables, is (jointly) Gaussian distributed with

$$\begin{aligned} \begin{bmatrix} y \\ \theta \end{bmatrix} &\sim \mathcal{N}\left( \begin{bmatrix} I & \phi(x) \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 \\ \mu^\theta \end{bmatrix}, \begin{bmatrix} I & \phi(x) \\ 0 & I \end{bmatrix} \begin{bmatrix} \Sigma^w & 0 \\ 0 & \Sigma^\theta \end{bmatrix} \begin{bmatrix} I & \phi(x) \\ 0 & I \end{bmatrix}^\mathsf{T} \right) \\ &\sim \mathcal{N}\left( \begin{bmatrix} \phi(x)\mu^\theta \\ \mu^\theta \end{bmatrix}, \begin{bmatrix} \Sigma^w + \phi(x)\Sigma^\theta\phi(x)^\mathsf{T} & \phi(x)\Sigma^\theta \\ \Sigma^\theta\phi(x)^\mathsf{T} & \Sigma^\theta \end{bmatrix} \right) \end{aligned}$$

# Posterior Parameter Distribution

$$\begin{bmatrix} y \\ \theta \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \phi(x)\mu^\theta \\ \mu^\theta \end{bmatrix}, \begin{bmatrix} \Sigma^w + \phi(x)\Sigma^\theta\phi(x)^\mathsf{T} & \phi(x)\Sigma^\theta \\ \Sigma^\theta\phi(x)^\mathsf{T} & \Sigma^\theta \end{bmatrix} \right)$$

we can find the posterior parameter distribution ($\theta$ given $y$) by Gaussian conditioning[1]

**Posterior Parameter Distribution**

The posterior distribution is Gaussian

$$p(\theta|y) = \mathcal{N}\left(\mu^{\theta|y}, \Sigma^{\theta|y}\right)$$

with $\mu^{\theta|y} = \mu^\theta + \Sigma^\theta\phi(x)^\mathsf{T}(\Sigma^w + \phi(x)\Sigma^\theta\phi(x)^\mathsf{T})^{-1}(y - \phi(x)\mu^\theta)$
and $\Sigma^{\theta|y} = \Sigma^\theta - \Sigma^\theta\phi(x)^\mathsf{T}(\Sigma^w + \phi(x)\Sigma^\theta\phi(x)^\mathsf{T})^{-1}\phi(x)\Sigma^\theta$

[1] Let $p(A, B) = \mathcal{N}\left(\begin{bmatrix} \mu^A \\ \mu^B \end{bmatrix}, \begin{bmatrix} \Sigma^A & \Sigma^{AB} \\ \Sigma^{BA} & \Sigma^B \end{bmatrix}\right)$ then $p(B|A) = \mathcal{N}\left(\mu^B + \Sigma^{BA}(\Sigma^A)^{-1}(A - \mu^A), \Sigma^B - \Sigma^{BA}(\Sigma^A)^{-1}\Sigma^{AB}\right)$

# Recursive Update

So far, we have only considered a single measurement $y$ and found

$$p(\theta|y) = \mathcal{N}(\mu^{\theta|y}, \Sigma^{\theta|y})$$

Given a new measurement $y'$, we can consider this our new 'prior'

Adjusting the notation, this directly leads to

**Recursive Parameter Update**

$$p(\theta|y_{i+1}, \ldots, y_1) = \mathcal{N}\left(\mu^{\theta}_{i+1}, \Sigma^{\theta}_{i+1}\right)$$

with $\mu^{\theta}_{i+1} = \mu^{\theta}_i + \Sigma^{\theta}_i \phi(x_{i+1})^{\mathsf{T}}(\Sigma^w + \phi(x_{i+1})\Sigma^{\theta}_i \phi(x_{i+1})^{\mathsf{T}})^{-1}(y_{i+1} - \phi(x_{i+1})\mu^{\theta})_i$
and $\Sigma^{\theta}_{i+1} = \Sigma^{\theta}_i - \Sigma^{\theta}_i \phi(x_{i+1})^{\mathsf{T}}(\Sigma^w + \phi(x_{i+1})\Sigma^{\theta}_i \phi(x_{i+1})^{\mathsf{T}})^{-1}\phi(x_{i+1})\Sigma^{\theta}_i$

# Example

In the weight-height regression problem, our model is

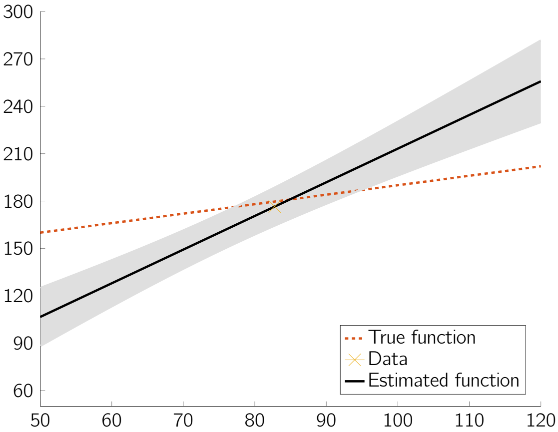
$$f(x) = \phi(x)\theta$$
$$y = f(x) + w$$

with

$$\phi(x) = [x \ 1]$$
$$\theta = [\theta_1, \ \theta_2]^\top.$$

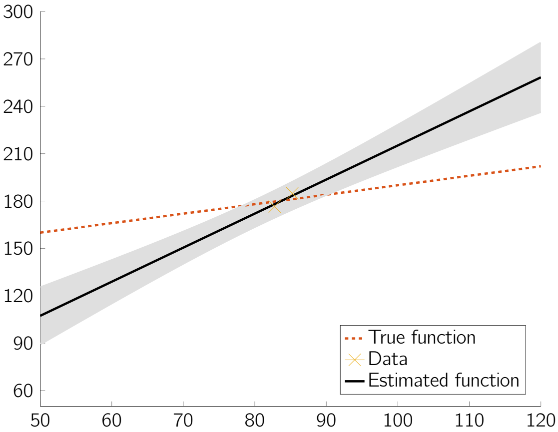
We assume $\Sigma_p = \begin{bmatrix} 200 & 0 \\ 0 & 200 \end{bmatrix}$ and $\sigma_n^2 = 16$.

# Example - $N = 1$



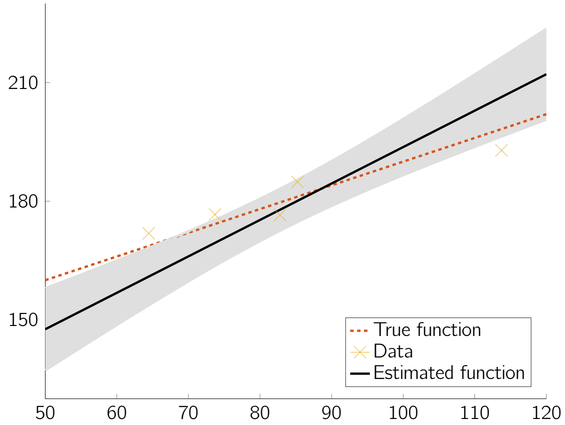
300
270
240
210
180
150
120
90
60
50
60
70
80
90
100
110
120
True function
Data
Estimated function

# Example - $N = 2$



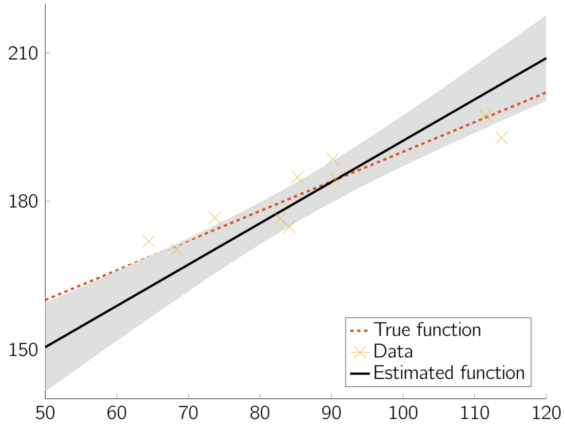
300
270
240
210
180
150
120
90
60
50
60
70
80
90
100
110
120
True function
Data
Estimated function

# Example - $N = 5$

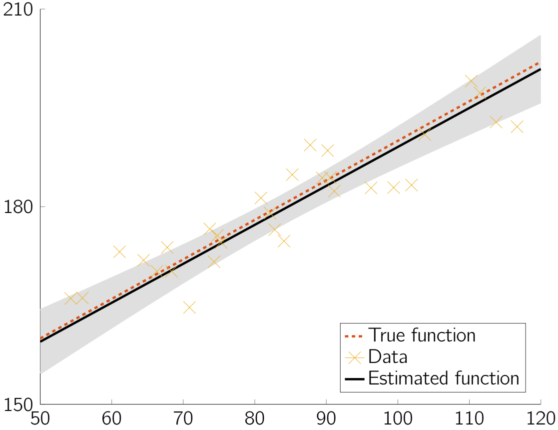


210
180
150
50
60
70
80
90
100
110
120
True function
Data
Estimated function

# Example - $N = 10$

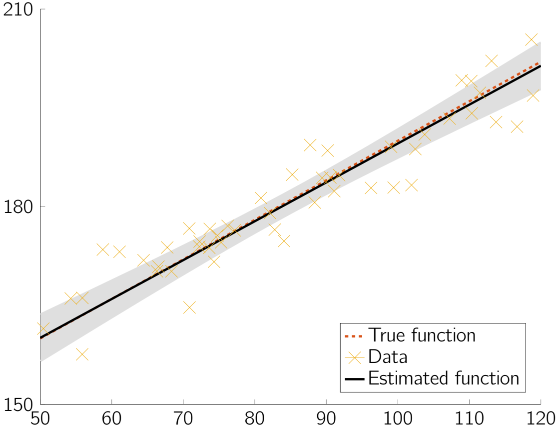

210
180
150
50
60
70
80
90
100
110
120
True function
Data
Estimated function

# Example - $N = 30$



210
180
150
50
60
70
80
90
100
110
120
True function
Data
Estimated function

# Example - $N = 50$


210
180
150
50
60
70
80
90
100
110
120
True function
Data
Estimated function

# Outline

1. Bayesian Linear Regression for Dynamics Learning

    Application to Parameter Identification

# General Remarks

$$x(k+1) = \phi(x(k), u(k))\theta + w(k)$$

- Uncertainty in the dynamics $\theta \sim \mathcal{N}(\mu^\theta, \Sigma^\theta)$
- Random (i.i.d.) disturbances $w(k) \sim \mathcal{N}(0, \Sigma^w)$

Assuming uncorrupted **state measurements**, this is standard form for Bayesian linear regression

| | | | |
|---|---|---|---|
| $x(k+1)$ | observed values | $\theta$ | parameter vector |
| $(x(k),\ u(k))$ | regressor | $\phi(x,u)$ | features |
| $w(k)$ | measurement noise | | |

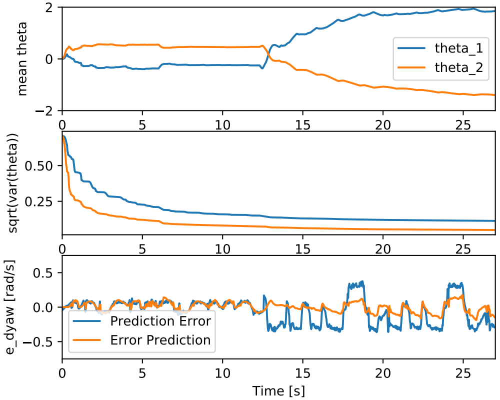
---

**Note:** In particular the lack of a measurement equation with (measurement) noise $y(k) = h(x(k)) + v(k)$ significantly simplifies the system identification problem.

- Generally requires simultaneous state & parameter estimation
- Approximate solutions (also in Linear-Gaussian case), see e.g. [1, 2]

# Example: Bayesian Linear Regression

Video: 03_BLR




Car dynamics change suddenly at 12 s → BLR adjusts slowly to changes

# Convergence of Estimate for Bayesian Linear Regression

Recall recursive (vector valued) Bayesian Linear Regression equations

$$y_k = \phi(x_k)\theta + w_k,\ \ w_k \sim \mathcal{N}(0, \Sigma_w) \text{ i.i.d.}$$

resulting in posterior parameter distribution $p(\theta|\{y_i, x_i\}_1^k) \sim \mathcal{N}\left(\mu_k^\theta, \Sigma_k^\theta\right)$ with

$$\mu_k^\theta = \mu_{k-1}^\theta + \Sigma_{k-1}^\theta \phi(x_k)^\mathsf{T} (\phi(x_k)\Sigma_{k-1}^\theta \phi(x_k)^\mathsf{T} + \Sigma_w)^{-1}(y_k - \phi(x_k)\mu_{k-1}^\theta),$$
$$\Sigma_k^\theta = \Sigma_{k-1}^\theta - \Sigma_{k-1}^\theta \phi(x_k)^\mathsf{T} \left(\phi(x_k)\Sigma_{k-1}^\theta \phi(x_k)^\mathsf{T} + \Sigma_w\right)^{-1} \phi(x_k)\Sigma_{k-1}^\theta$$

---

Note that estimate gets more confident over time and adaptation slows down

- Posterior parameter covariance decreases over time[2] $\lim_{k\to\infty} \Sigma_k^\theta = 0$
- Which then implies $\lim_{k\to\infty} \mu_{k+1}^\theta - \mu_k^\theta = 0$

Parameter estimates converge over time $\to$ learning/adaptation slows down and stops

[2] requires 'informativeness' condition on $\phi(x_k)$

# From Bayesian Linear Regression to Kalman Filtering

To alleviate this, we can introduce **parameter dynamics**

$$\theta(k+1) = \theta(k) + w_\theta(k), \qquad w_\theta(k) \sim \mathcal{N}(0, \Sigma^{w_\theta}) \text{ i.i.d.}$$
$$x(k+1) = \phi(x(k), u(k))\theta(k) + w(k), \qquad w(k) \sim \mathcal{N}(0, \Sigma^{w}) \text{ i.i.d.}$$

Modelling the possibility that the parameters can evolve randomly over time (typically slowly w.r.t. system dynamics, i.e. $\Sigma^{w_\theta} \ll \Sigma^{w}$)

---

Note that this corresponds exactly to Kalman filter setup:

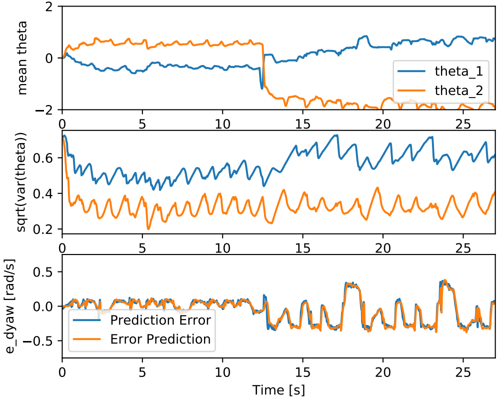
$$\bar{x}(k+1) = A_k \bar{x}(k) + \bar{w}(k), \qquad \bar{w}(k) \sim \mathcal{N}(0, \Sigma^{\bar{w}}) \text{ i.i.d.}$$
$$\bar{y}(k+1) = C_k \bar{x}(k) + \bar{v}(k), \qquad \bar{w}(k) \sim \mathcal{N}(0, \Sigma^{\bar{v}}) \text{ i.i.d.}$$

with $\bar{x}(k) := \theta(k)$, $\bar{y}(k) := x(k)$, $\bar{w}(k) := w_\theta(k)$, $\bar{v}(k) := w(k)$, $A_k = I$ and time varying measurement equation $C_k = \phi(x(k), u(k))$

Time-varying Kalman filter equations can be used for Bayesian parameter tracking

# Example: Kalman Filter for Parameter Estimation

Video: `04_KF`




Car dynamics change suddenly at $\approx 12\,\mathrm{s} \rightarrow$ Kalman filter adjusts quickly to change

# Outline

# Robustness in Probability

Assume that the **true** system can be described by

$$x(k+1) = \tilde{f}(x(k), u(k), \theta) + \tilde{w}(k)$$

which is subject to **parametric uncertainties** $\theta$ as well as **disturbances** $\tilde{w}(k)$

**Typical Robust Approach**

1. Define nominal model $f(x, u, \hat{\theta})$ for analysis and control
2. Assume/ensure bounded deviation of **true system** from **nominal model**

$$x(k+1) - f(x(k), u(k), \hat{\theta}) \in \mathcal{W} \ \forall k \geq 0$$

3. Carry out **robust analysis** or **design** for $x(k+1) = f(x(k), u(k), \hat{\theta}) + w(k)$ and all $w(k) \in \mathcal{W}$

$\rightarrow$ Robust guarantees for original system (if 2. holds)

# Robustness in Probability

Assume that the **true** system can be described by

$$x(k+1) = \tilde{f}(x(k), u(k), \theta) + \tilde{w}(k)$$

which is subject to **parametric uncertainties** $\theta$ as well as **disturbances** $\tilde{w}(k)$

**Typical Robust in Probability Approach**

1. Define nominal model $f(x, u, \hat{\theta})$ for analysis and control
2. Assume/ensure bounded deviation of **true system** from **nominal model** in probability

$$\Pr(x(k+1) - f(x(k), u(k), \hat{\theta}) \in \mathcal{W} \ \forall k \geq 0) \geq p$$

3. Carry out **robust analysis** or **design** for $x(k+1) = f(x(k), u(k), \hat{\theta}) + w(k)$ and all $w(k) \in \mathcal{W}$

$\rightarrow$ Robust guarantees for original system hold with probability $p$ (if 2. holds)

# Multivariate Gaussian Intervals: Chi-squared Distribution

Consider the standard $n$-dimensional multivariate Gaussian distribution

$$x \sim \mathcal{N}(0, I)$$

The distribution of $x^\mathsf{T} x$ is known as the chi-squared distribution with $n$ degrees of freedom

$$x^\mathsf{T} x = \sum_{i=1}^{n} x_i^2, \text{ with i.i.d.} x_i \sim \mathcal{N}(0, 1)$$

Quantile function is available (matlab: `chi2inv`)

$$\chi_n^2(p) = \bar{x}, \text{ such that } \Pr(x^\mathsf{T} x \leq \bar{x}) = p$$

$$\boxed{\Pr\left(x^\mathsf{T} x \leq \chi_n^2(p)\right) = p \text{ (generalizing 1-D (confidence) interval)}}$$

# Multivariate Gaussian Intervals: General Result

Consider the standard $n$-dimensional multivariate Gaussian distribution

$$x \sim \mathcal{N}(0, I) \text{ as well as a general } \tilde{x} \sim \mathcal{N}(\mu, \Sigma)$$

with $\Sigma > 0$. Then there exists[3] $\Sigma^{\frac{1}{2}} > 0$ such that

$$\tilde{x} \overset{d}{=} \mu + \Sigma^{\frac{1}{2}} x \Leftrightarrow \Sigma^{-\frac{1}{2}}(\tilde{x} - \mu) \overset{d}{=} x$$

From $\Pr\left(x^\top x \leq \chi_n^2(p)\right) = p$ we immediately get

$$\boxed{\Pr\left((\tilde{x} - \mu)^\top \Sigma^{-1}(\tilde{x} - \mu) \leq \chi_n^2(p)\right) = p \text{ for } \tilde{x} \sim \mathcal{N}(\mu, \Sigma)}$$

[3] e.g. Cholesky decomposition of $\Sigma$

# Application to Bayesian Linear Regression

$$x(k+1) = \phi(x(k), u(k))\theta + w(k), \text{ with i.i.d. } w(k) \sim \mathcal{N}(0, \Sigma^w)$$

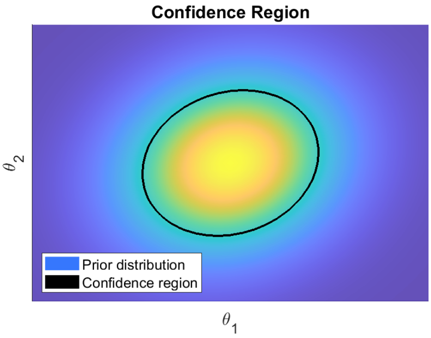
given a normal prior on $\theta$ and measurement data $\mathcal{D} = \{x(k+1), \phi(x(k), u(k))\}_{k=0}^{n_d}$,

posterior parameter distribution is Gaussian

$$p(\theta \,|\, \mathcal{D}) = \mathcal{N}(\mu_\theta, \Sigma_\theta)$$

and we have $\Pr(\theta \in \Omega \,|\, \mathcal{D}) = p$
with $\Omega = \{\theta \,|\, (\theta - \mu_\theta)^\mathsf{T} \Sigma_\theta^{-1} (\theta - \mu_\theta) \le \chi_n^2(p)\}$

Allows for **robust analysis** w.r.t $\theta \in \Omega$
(which holds with probability $\geq p$)



# Resulting Robust in Probability Approach

Assume now that the **true** system is given by

$$x(k+1) = \phi(x(k), u(k))\theta + \tilde{w}(k)$$

and we have found a parameter set $\Omega$ such that

$$\Pr(\theta \in \Omega) \geq p$$

By assuming/ensuring that $\tilde{w}(k) \in \mathcal{W}$ for all times $k \geq 0$ we can reduce this to the **robust problem**

$$x(k+1) = \phi(x(k), u(k))\theta + \tilde{w}(k) \text{ for all } \theta \in \Omega,\ \tilde{w}(k) \in \mathcal{W}$$

for which we have seen solutions e.g. in the last lecture $\rightarrow$ valid with probability $\geq p$

**Problems**
1. $\tilde{w}(k) \in \mathcal{W}\, \forall k \geq 0$ inconsistent with Gaussian assumption for estimation
2. Feasibility after uncertainty update can usually not be ensured

# Outline

1. Bayesian Linear Regression for Dynamics Learning

2. Robustness in Probability

3. Non-parametric Bayesian Regression: Gaussian Processes

4. Application to Autonomous Racing

# Gaussian Processes

| Definition: Gaussian Process |
| --- |
| A Gaussian Process (GP) defines a distribution over functions, any finite evaluations of which are jointly Gaussian distributed. |

A GP is completely defined by its mean $m(x)$ and covariance function $k(x, x')$

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

such that an evaluation of $f$ at $x_1, \dots, x_N$ is

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix} = \mathcal{N}\left( \begin{bmatrix} m(x_1) \\ \vdots \\ m(x_N) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_N) \\ \vdots & \ddots & \vdots \\ k(x_N, x_1) & \dots & k(x_N, x_N) \end{bmatrix} \right)$$

For simplicity, we typically assume $m(x) = 0$.

# GP Regression

$$y = f(x) + w, \text{where } f \sim \mathcal{GP}(m,k) \text{ and } w \sim \mathcal{N}(0,\sigma_w^2) \text{ (i.i.d.)}$$

- Observations $Y = [y_1, \ldots, y_N]$ at locations $X = x_1, \ldots, x_N$.
- Test points $X^* = [x_1^*, \ldots, x_m^*]$

$$\begin{bmatrix} Y \begin{cases} y_1 \\ \vdots \\ y_N \end{cases} \\ f(X^*) \begin{cases} f(x_1^*) \\ \vdots \\ f(x_m^*) \end{cases} \end{bmatrix} = \mathcal{N}\left( 0, \begin{bmatrix} \underbrace{\begin{bmatrix} k(x_1,x_1)+\sigma_w^2 & \ldots & k(x_1,x_N) \\ \vdots & \ddots & \vdots \\ k(x_N,x_1) & \ldots & k(x_N,x_N)+\sigma_w^2 \end{bmatrix}}_{K(X,X)+\sigma_w^2 I} & \underbrace{\begin{bmatrix} k(x_1,x_1^*) & \ldots & k(x_1,x_m^*) \\ \vdots & \ddots & \vdots \\ k(x_N,x_1^*) & \ldots & k(x_N,x_m^*) \end{bmatrix}}_{K(X,X^*)} \\ K(X^*,X) & \underbrace{\begin{bmatrix} k(x_1^*,x_1^*) & \ldots & k(x_1^*,x_m^*) \\ \vdots & \ddots & \vdots \\ k(x_m^*,x_1^*) & \ldots & k(x_m^*,x_m^*) \end{bmatrix}}_{K(X^*,X^*)} \end{bmatrix} \right)$$

# GP Regression

$$y = f(x) + w, \text{where } f \sim \mathcal{GP}(m, k) \text{ and } w \sim \mathcal{N}(0, \sigma_w^2) \text{ (i.i.d.)}$$

- Observations $Y = [y_1, \dots, y_N]$ at locations $X = x_1, \dots, x_N$.
- Test points $X^* = [x_1^*, \dots, x_m^*]$

$$\begin{bmatrix} Y \\ f(X^*) \end{bmatrix} = \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma_w^2 I & K(X, X^*) \\ K(X^*, X) & K(X^*, X^*) \end{bmatrix}\right)$$

Again, we can do inference by Gaussian conditioning!

**Posterior Distribution**

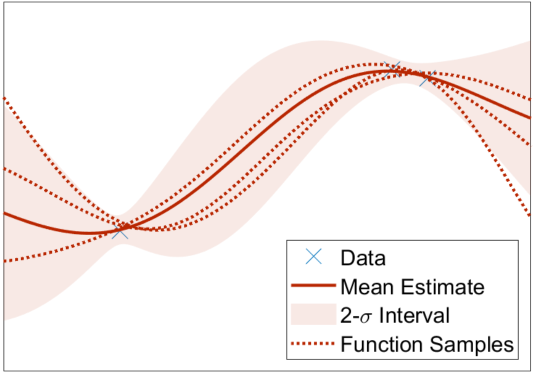
$$p(f(X^*)|Y) = \mathcal{N}(\mu(X^*), \Sigma(X^*))$$

with $\mu(X^*) = K(X^*, X)(K(X,X) + \sigma_w^2 I)^{-1} Y$
and $\Sigma(X^*) = K(X^*, X^*) - K(X^*, X)(K(X,X) + \sigma_w^2 I)^{-1} K(X, X^*)$

# Illustration

GP regression specifies **distribution of functions**

- Function values jointly Gaussian distributed $\begin{bmatrix} f(x_1^*) & \dots & f(x_N^*) \end{bmatrix} \sim \mathcal{N}(\mu(X^*), \Sigma(X^*))$
- We can draw **function samples** (choose $X^*$ as tight grid)



# Robust in Probability

Due to the Gaussian distribution of function values, it is easy to bound **marginal distribution**

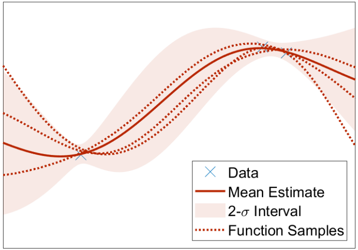
$$\Pr(f(x) \in \tilde{\Omega}(x)) = p \text{ with } \tilde{\Omega}(x) = [\mu(x) - c\sigma(x), \mu(x) - c\sigma(x)]$$

But for 'Robust in Probability' we require bound on **joint distribution** of function values

$$\Pr(f(x) \in \Omega(x) \, {\color{red}\forall x \in \mathcal{X}}) \geq p$$

For instance, if $f(x_1)$ and $f(x_2)$ are uncorrelated (e.g. $x_1$, $x_2$ far apart, $k(x_1, x_2) \approx 0$)

$$\Pr(f(x_1) \in \tilde{\Omega}(x_1) \textbf{ and } f(x_2) \in \tilde{\Omega}(x_2)) \approx p^2$$



# Bounding GPs over compact domains

Over compact domain $\mathcal{X}$, one can nevertheless find bounds of type

$$\Pr(f(x) \in \Omega(x) \, \forall x \in \mathcal{X}) \geq p$$

**[3, Theorem 3.1] (simplified)**

Consider a zero mean Gaussian process defined through kernel $k(x, x')$ with Lipschitz constant $L_k$ on the compact set $\mathcal{X}$. Then one can find constants $\beta$ and $\gamma$ such that

$$\Pr(|f(x) - \mu(x)| \leq \sqrt{\beta}\Sigma(x) + \gamma, \, \forall x \in \mathcal{X}) \geq p$$

where $\beta$ and $\gamma$ depend (among others) on $L_k$ and the 'size' of $\mathcal{X}$.

- Alternatives exists [4] which, however, require different assumptions true function $f$
- Obtained bounds are often conservative and challenging to directly apply in control applications

# Outline

1. Bayesian Linear Regression for Dynamics Learning
2. Robustness in Probability
3. Non-parametric Bayesian Regression: Gaussian Processes
4. Application to Autonomous Racing

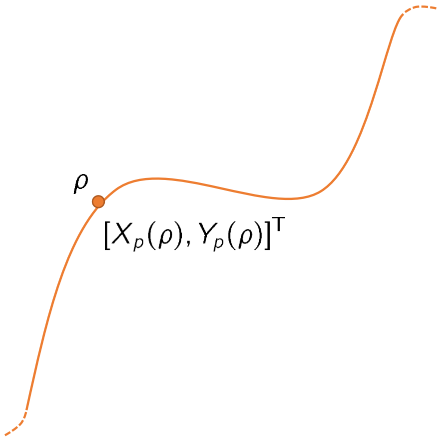
# Outline

# Model Predictive Contouring Control

Dynamic system with **position states** $X(k),\ Y(k)$

$$x(k+1) = f(x(k), u(k))$$
$$x(k) = [X(k),\ Y(k),\ \ldots]^\mathrm{T}$$

Path parameterized by $\rho$ : $X_p(\rho),\ Y_p(\rho)$




**Goal:** Follow path **accurately** and **quickly**

Applications in

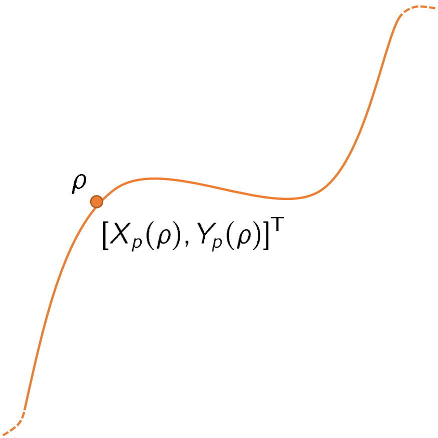
- Machining (e.g. laser cutting)
- Autonomous racing
- ...

# Model Predictive Contouring Control

Dynamic system with **position states** $X(k)$, $Y(k)$

$$x(k+1) = f(x(k), u(k))$$
$$x(k) = [X(k),\ Y(k),\ \ldots]^\mathsf{T}$$

Path parameterized by $\rho$ : $X_p(\rho)$, $Y_p(\rho)$




**Goal:** Follow path **accurately** and **quickly**

Applications in

- Machining (e.g. laser cutting)
- **Autonomous racing**
- ...

# Model Predictive Contouring Control

Progress as additional state $\rho_i$, increment as input $\nu_i$

$$\rho_{i+1} = \rho_i + \nu_i$$

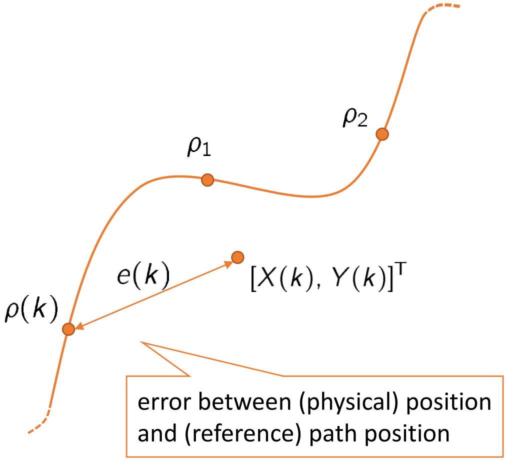
results in 'reference position' on path

$$(X_p(\rho_i), Y_p(\rho_i))$$

Design cost function $l(x, u)$ which

1. Encourages progress: $-\nu(k)$
2. Penalizes error: $\|e(k)\|^2$
   - Accurate tracking
   - Connects $\rho$ to position $X(k)$, $Y(k)$

and additionally introduce constraints (e.g. to stay on track)



# Example: ORCA Platform [5]

Video: `01_ORCA`

# Effects of Model Mismatch

Video: `02_ORCA_GPMPC`

# Effects of Model Mismatch

→ Systematic performance and safety issues resulting from model mismatch

# Outline

4. Application to Autonomous Racing

    Contouring MPC for Autonomous Racing

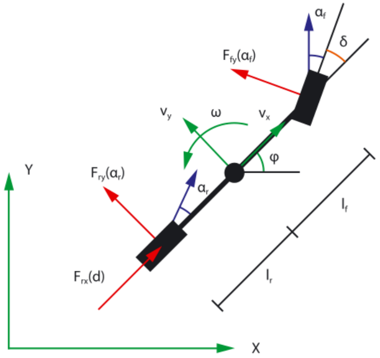
    Model Learning for Performance Improvement

# Nominal System Model

Nominal dynamics derived from first principles

$$\text{Kinematic} \begin{cases} \dot{X} &= v_x \cos(\varphi) - v_y \sin(\varphi) \\ \dot{Y} &= v_x \sin(\varphi) + v_y \cos(\varphi) \\ \dot{\varphi} &= \omega \end{cases}$$

$$\text{Dynamic} \begin{cases} \dot{v}_x &= \frac{1}{m}(F_{r,x} - F_{f,y} \sin(\delta) + m v_y \omega) \\ \dot{v}_y &= \frac{1}{m}(F_{r,y} + F_{f,y} \cos(\delta) + m v_x \omega) \\ \dot{\omega} &= \frac{1}{I_z}(F_{f,y} l_f \cos(\delta) - F_{r,y} l_r) \end{cases}$$




with nonlinear tire forces $F_{f/r,x/y}$

Learn model mismatch only on **dynamic** part of model: $B_d$

$$\boxed{x(k+1) = f(x(k), u(k)) + B_d(g(x(k), u(k) + w(k))}$$

# GP-based MPC

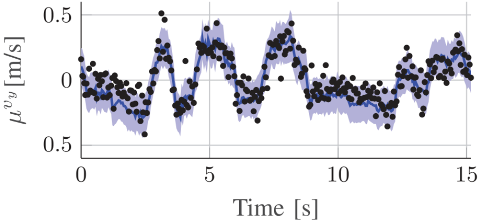
Assumption: Nominal model allows for (rudimentary) control of the system

1. Collect data with nominal controller (with $B_d^\dagger$ the pseudoinverse of $B_d$)

$$(y_j, x_j): \text{ where } y = B_d^\dagger(x(k+1) - f(x(k), u(k))),\ x = [x(k),\ u(k)]^\mathsf{T}$$

2. Use Gaussian Process regression to estimate
   - Model mismatch $g(x(k), u(k))$
   - Disturbances $w(k)$

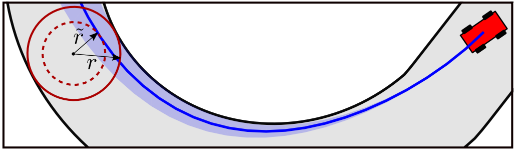
via Maximum likelihood hyperparameter optimization



# Uncertainty Propagation for Constraint Tightening

$$x_{i+1} = f(x_i, u_i) + B_d(\hat{g}(x_i, u_i) + w_i)$$

Use uncertainty description of GP for **constraint-tightening** MPC




Challenges:

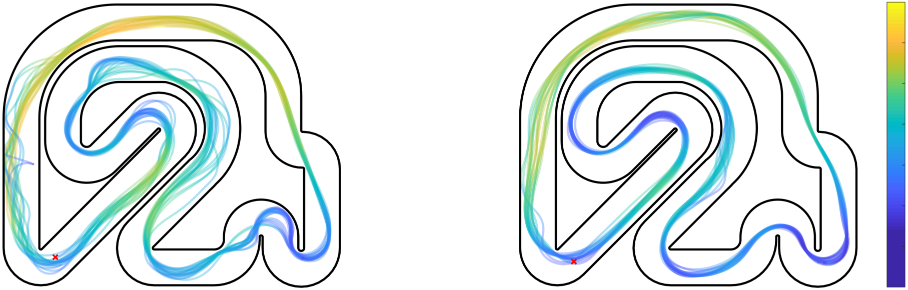
1. Stochastic uncertainty propagation through non-linear system[4]
   Practical solution: Linearize around 'mean prediction' and use linear techniques (Lecture 5)
2. Recursive feasibility under receding horizon control
   Practical solution: Make use of soft-constraints
3. Computational effort
   Practical solution: Use (shifted) solution from previous time step for precomputations & further approximations

[4] In particular under GP assumption, see e.g. Hewing et al., L4DC, 2020

# Resulting GP-Based Racing Controller [6]

Video: `02_ORCA_GPMPC`

# Resulting GP-Based Racing Controller [6]



Nominal Controller      GP-Based Controller

Systematic performance and safety issues addressed with automatic model adjustment

$\rightarrow$ Overall lap time & safety improvements

# Computational Approximations for GPs

GP regression is a **non-parametric** regression technique:

$\rightarrow$ Evaluation of $\hat{g}(x(k), u(k))$ becomes more expensive as we collect more data

This is usually addressed in one of two ways

1. Use sparse spectrum GP approximation [7,8]
   - Construct finite number of basis functions to approximate GP
   - Can then essentially be treated as **Bayesian Linear Regression**

   $\rightarrow$ Can be treated as parametric regression

2. Make use of reduced number of **inducing inputs** [9]
   - Construct inducing input locations to approximate GP
   - Computation times scale mainly with # of inducing inputs

   $\rightarrow$ Remains fully non-parametric

# Data-Point Management for Gaussian Processes

In non-parametric setting, one typically has to employ a **data-management system** keeping track of a finite number of data points.

Typical criteria for data point selection are

- Age of data points
  Simplest case: Keep last $N_d$ data points (windowing)
- Informativeness of data-points
  Can be directly related to posterior variance at data-point location $x_i$

$$\mathcal{I}(x_i) \propto k(x_i, x_i) - K(x_i, X_{-i})(K(X_{-i}, X_{-i}) + I\sigma_n^2)^{-1} K(X_{-i}, x_i)$$

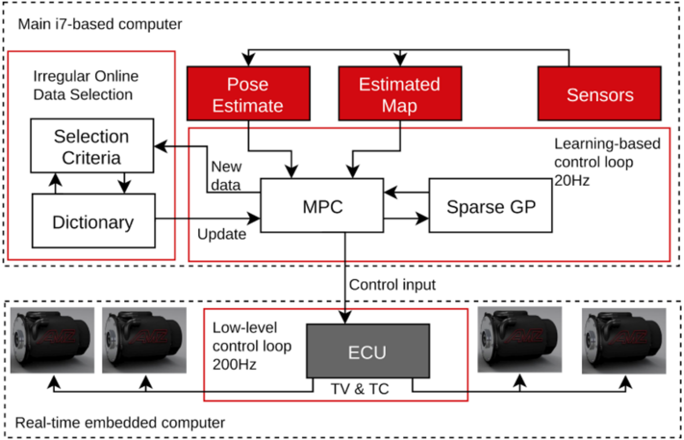
- A combination thereof [10]
  Compute score for each data-point based on informativeness and age and drop worst scoring point at each time step

# Example: GPMPC on AMZ Car [10]

Full-sized autonomous race car[5]

Continuous learning during operation

Sparse inducing inputs GPs




[5] Akademischer Motorsportverein Zürich (AMZ), `electric.amzracing.ch`

# Example: GPMPC on AMZ Car [10]

Video: `05_AMZ_GPMPC`

# Summary

## Model Learning in MPC

A number of regression techniques provide **uncertainty quantification**.
These can be used in **robust** & **stochastic** MPC formulations:

**Robust**

- Set membership estimation for **robust** uncertainty bounds
- Strong theoretical safety guarantees when used in MPC
- Practical feasibility and performance needs to be further evaluated

**Stochastic**

- Powerful regression and estimation techniques in stochastic settings
- Use in MPC theoretically challenging – but can be very effective in practice
- Option 1 (reasonably strong guarantees): Translate into robust setting
- Option 2 (weak/no guarantees): Work directly with stochastic description

# References and further reading


[1] Ljung, "System identification, theory for the user", 1999

[2] Schön et al., "System identification of nonlinear state-space models" Automatica, 2011

[3] Lederer et al., "Uniform Error Bounds for Gaussian Process Regression with Application to Safe Control," NIPS, 2019

[4] Koller et al., "Learning-based model predictive control for safe exploration," CDC, 2018

[5] Liniger et al., "Optimization-based autonomous racing of 1:43 scale RC cars ," Optimal Control Applications and Methods, 2015

[6] Hewing et al., "Cautious model predictive control using Gaussian process regression," TCST, 2020

[7] Rahimi and Recht et al., "Random Features for Large-Scale Kernel Machines," NIPS, 2007

[8] Lázaro-Gredilla et al., "Sparse Spectrum Gaussian Process Regression," JMLR, 2010

[9] Quinoñero-Candela and Rasmussen, "A Unifying View of Sparse Approximate Gaussian Process Regression" JMLR, 2005

[10] Kabzan et al., "Learning-Based Model Predictive Control for Autonomous Racing," RA-L, 2019

[11] Fröhlich et al., "Contextual Tuning of Model Predictive Control for Autonomous Racing," IROS, 2022