Machine Learning for Systems and Control

5SC28

Lecture 1B: Preliminaries

dr. ir. Maarten Schoukens & dr. ir. Roland Tóth

Control Systems Group

Department of Electrical Engineering

Eindhoven University of Technology

Academic Year: 2020-2021 (version 1.0)

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Version 13/04/2021

Poll: Study Background

A: Systems & Control / EE - Control Systems

B: Artificial Intelligence Engineering Systems

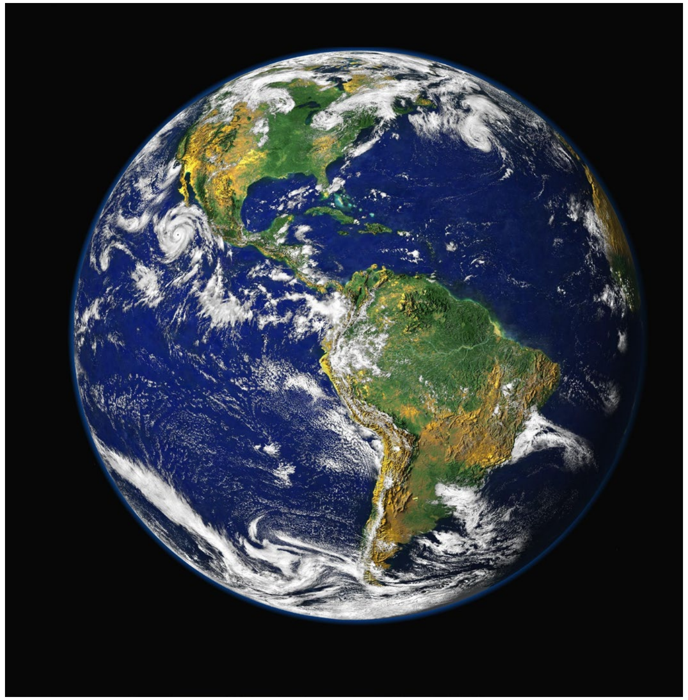
C: Electrical Engineering (other specialization)

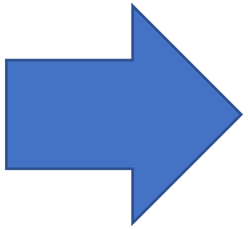
D: Mechanical Engineering

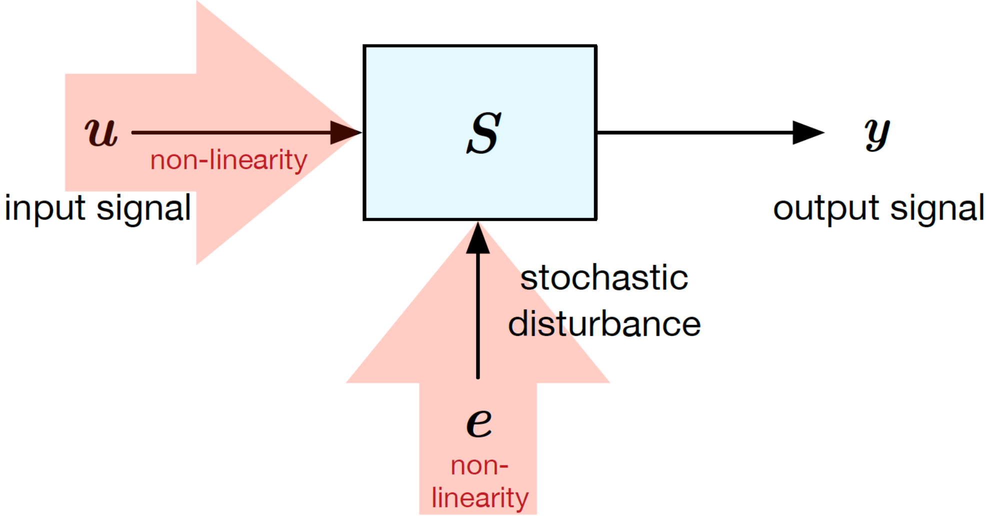
E: Applied Physics

F: Other (write in chat what background you have)

From Data to Model







Learning Objectives

Why Data-Driven Modelling?

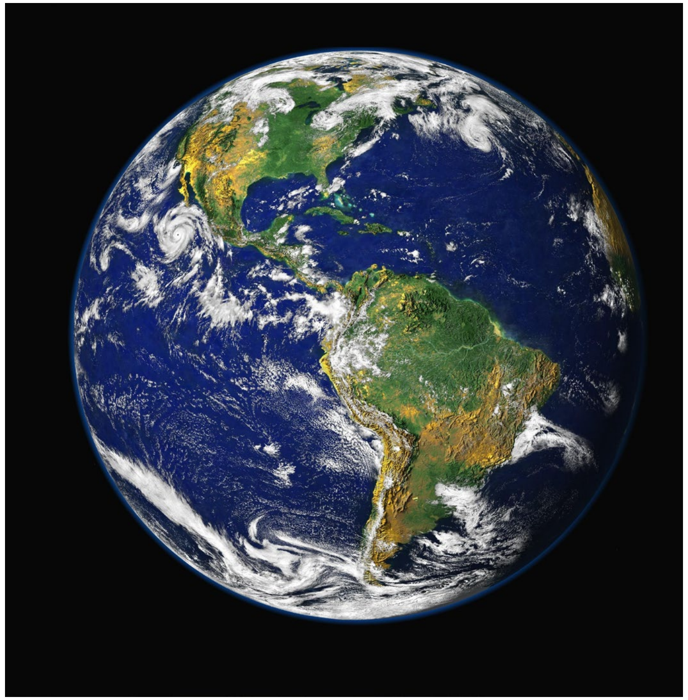
What Model Structures Are Used?

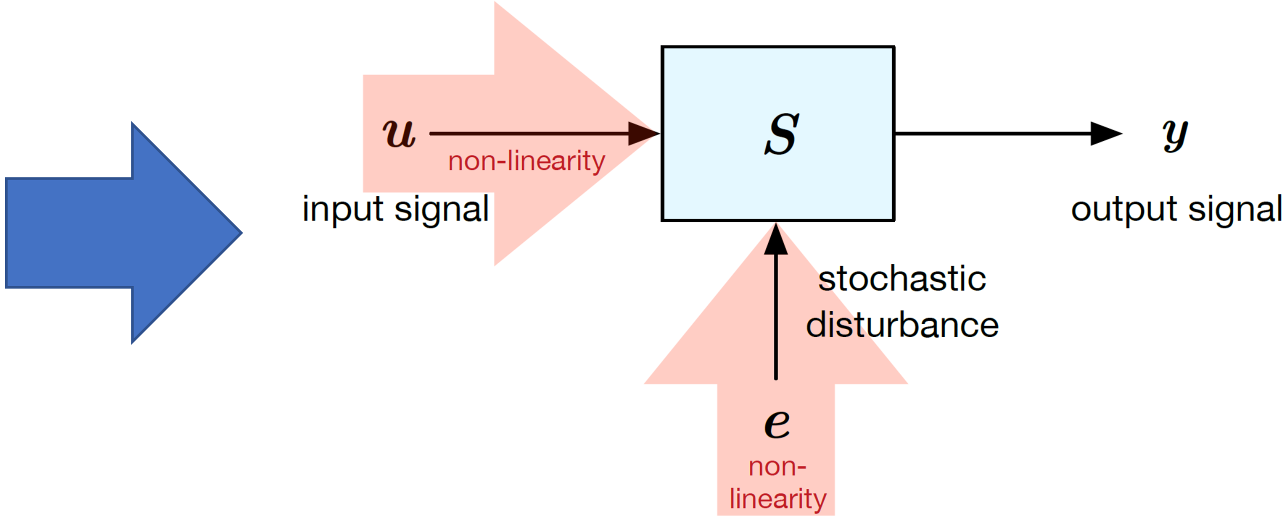
Estimation Framework

Contents

Contents

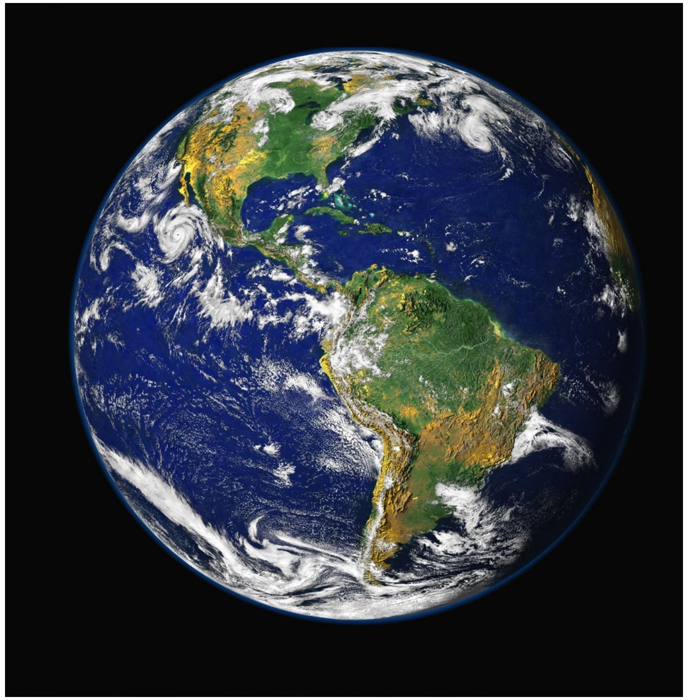
Models

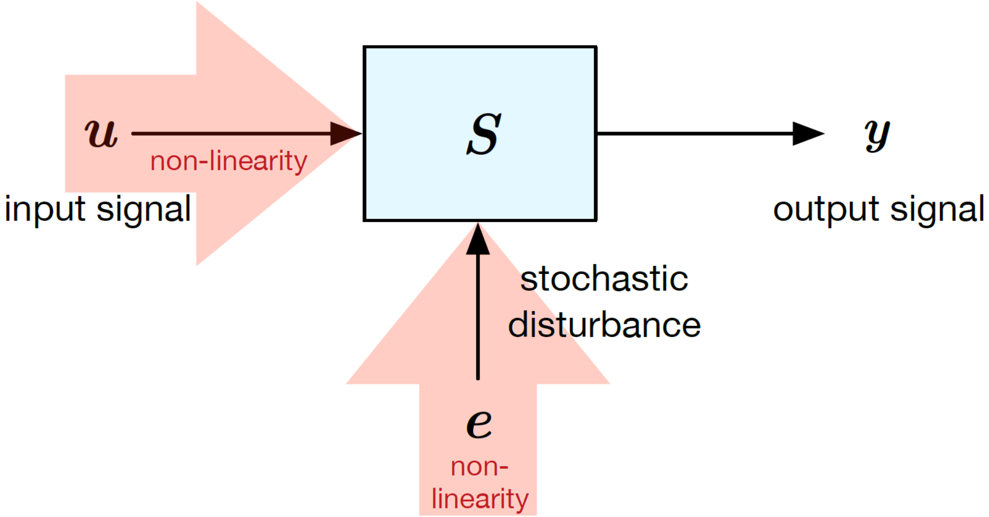




A model is an abstraction (approximation/simplification) of a part of reality, which is adequate for a given utilization purpose

Models





system (re)design, control design / optimization, prediction, simulation, diagnosis / fault detection, monitoring / soft sensing

Models in Engineering

Models are Needed Everywhere

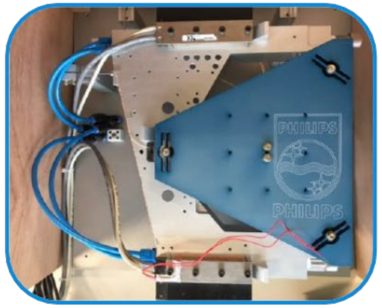
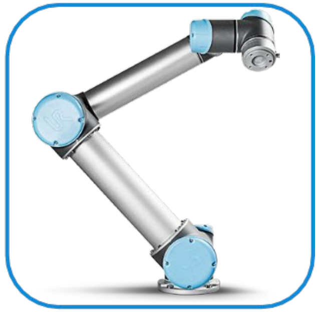
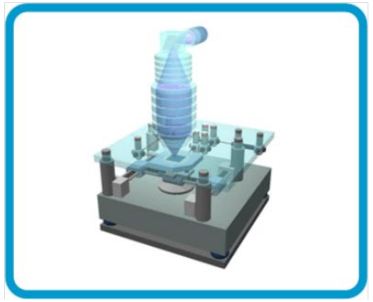
- Essential to Understanding
- Essential for Design Process
- Abstraction of Knowledge / Information

Utilization Perspective is Important

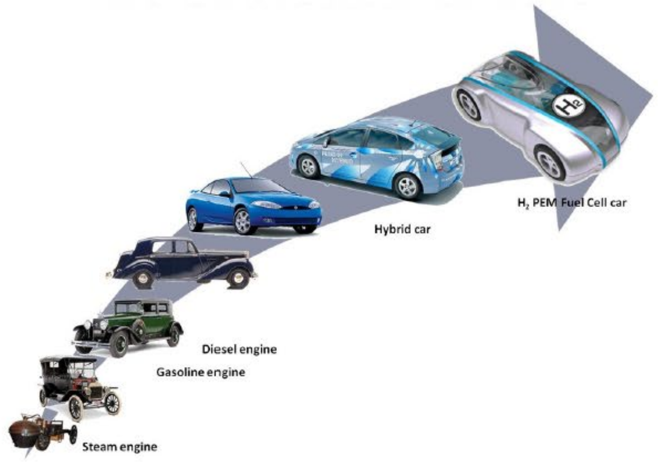
- Dynamic vs Static Models
- Temporal Aspects of the Dynamics
- Deterministic vs Stochastic
- Prediction / Simulation / Control





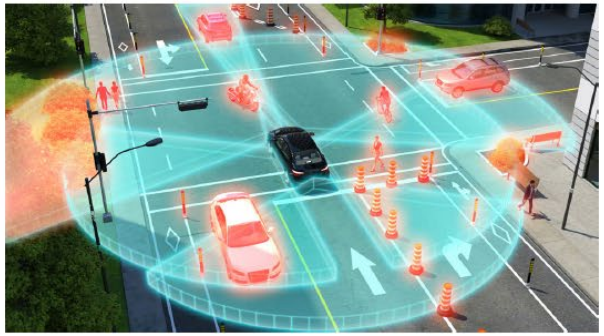

Growing Challenges



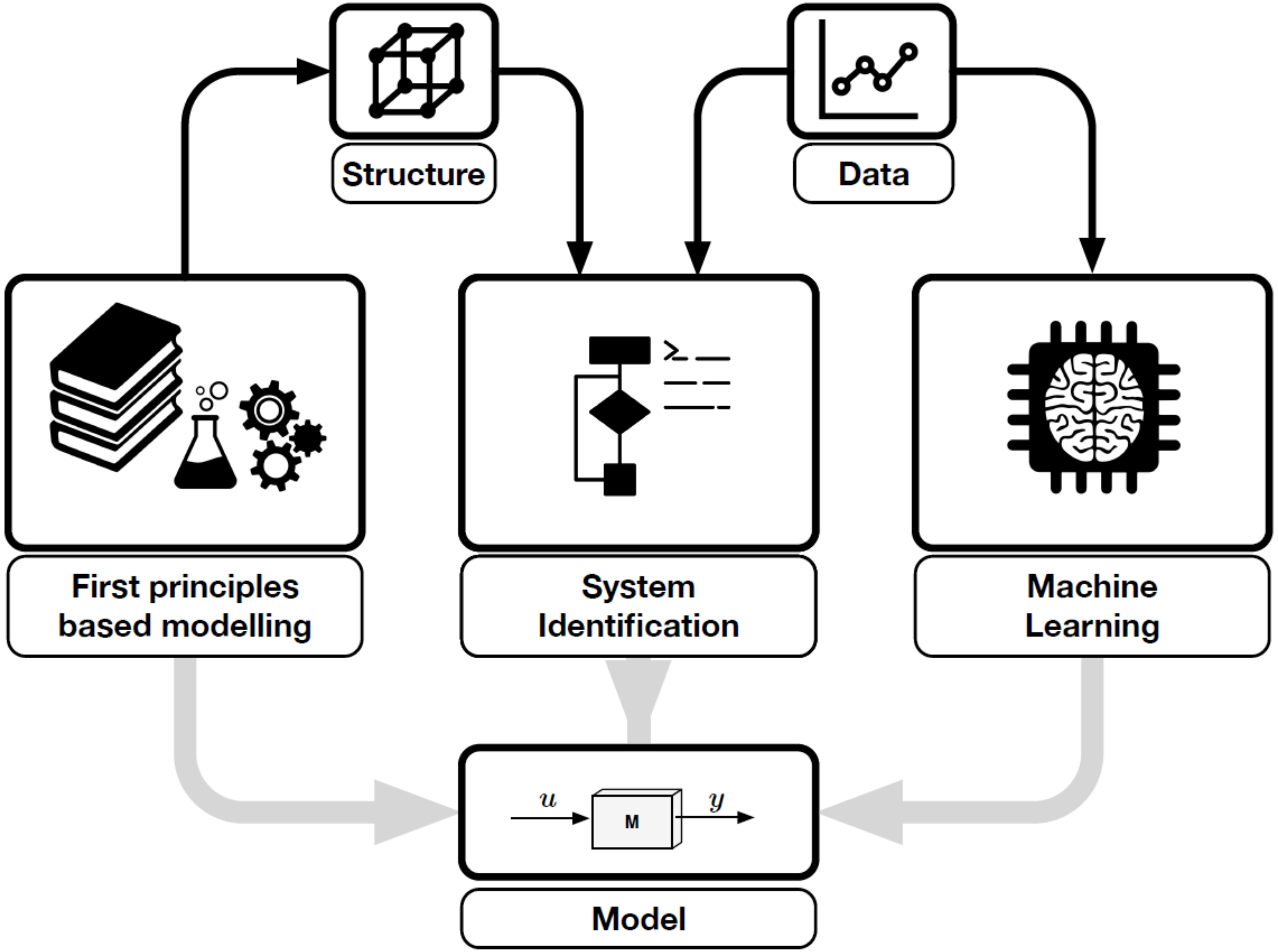
Complexity

Performance

Interconnectivity

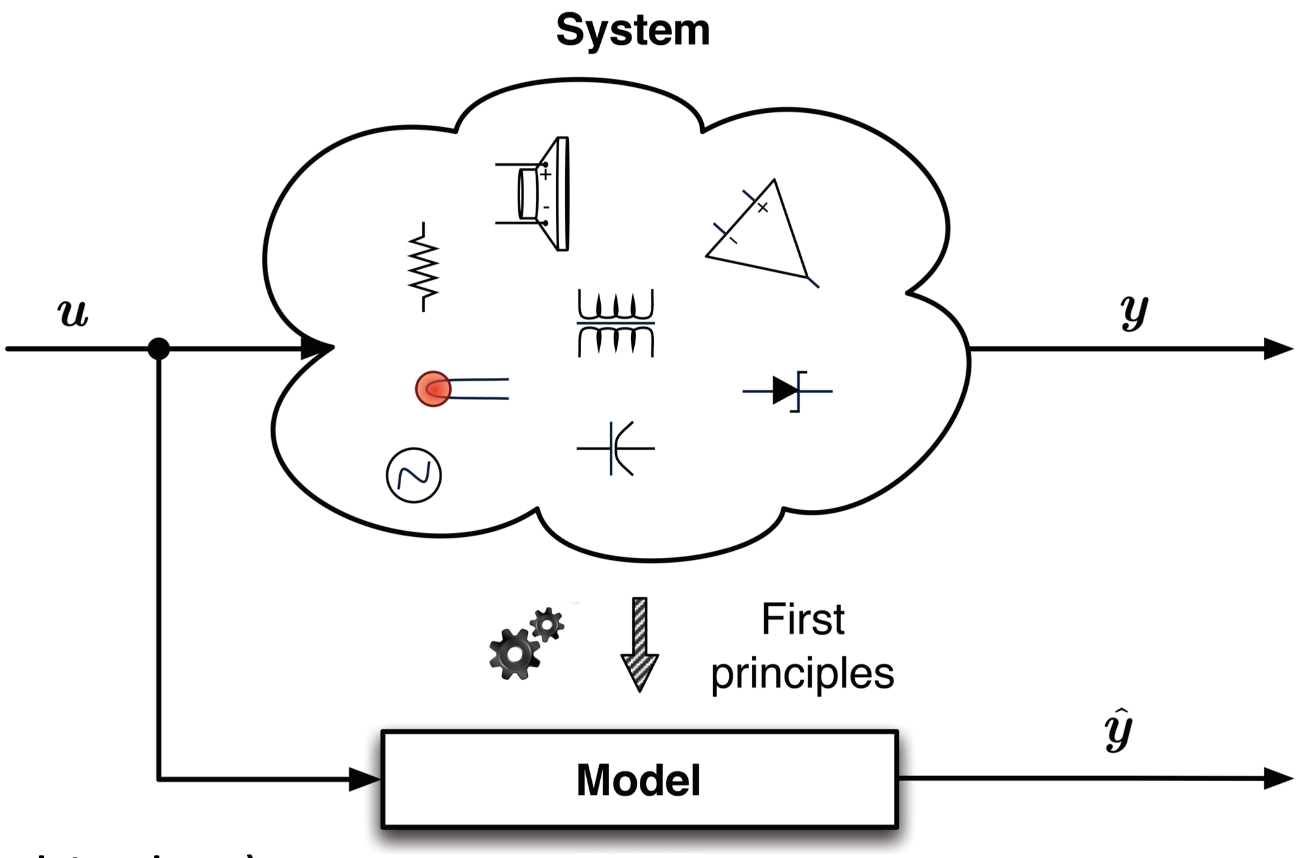


The Process



Structure
Data
First principles based modelling
System Identification
Machine Learning
u
M
y
Model

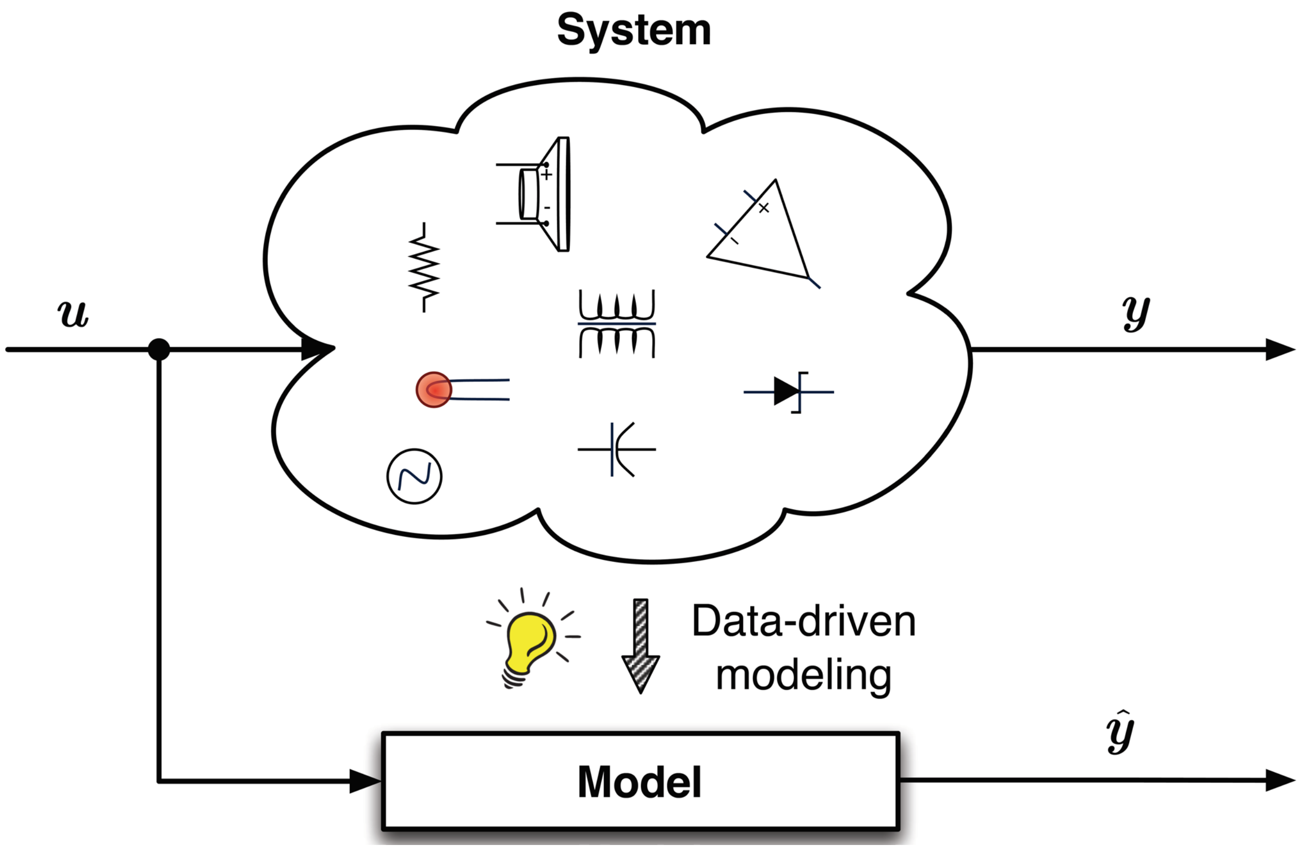
White-Box



First principles-based modeling (white box):

a priori knowledge and first principles laws
physical interpretation
high complexity and cost, but strong reliability

Data-Driven



Data-driven modeling (identification, black & grey box):

From measured data to model
Exploration of the dynamical structure: loss of physical interpretation
Low complexity and cost: model what you observe

Data-Driven Methods in Systems & Control

Data-Driven Modelling

- Capturing Unknown Dynamics
- Model Verification

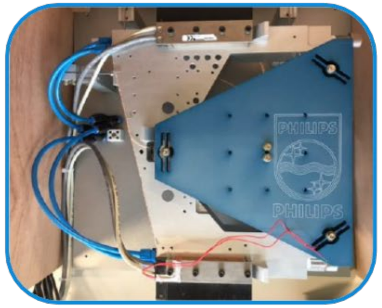
Model Maintenance and Adaptation

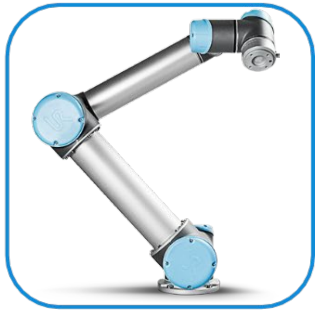
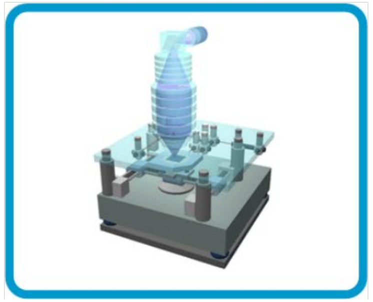
Extracting Control Relevant Behavior

Model Free Control Design





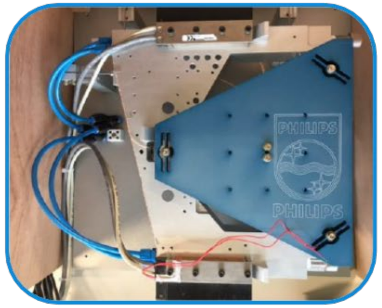


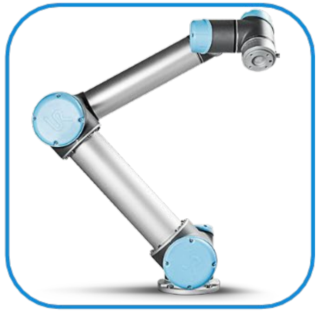



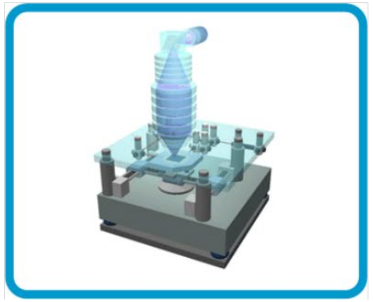

Data-Driven Methods in Systems & Control

How to scale with the surge of complexity & performance needs?

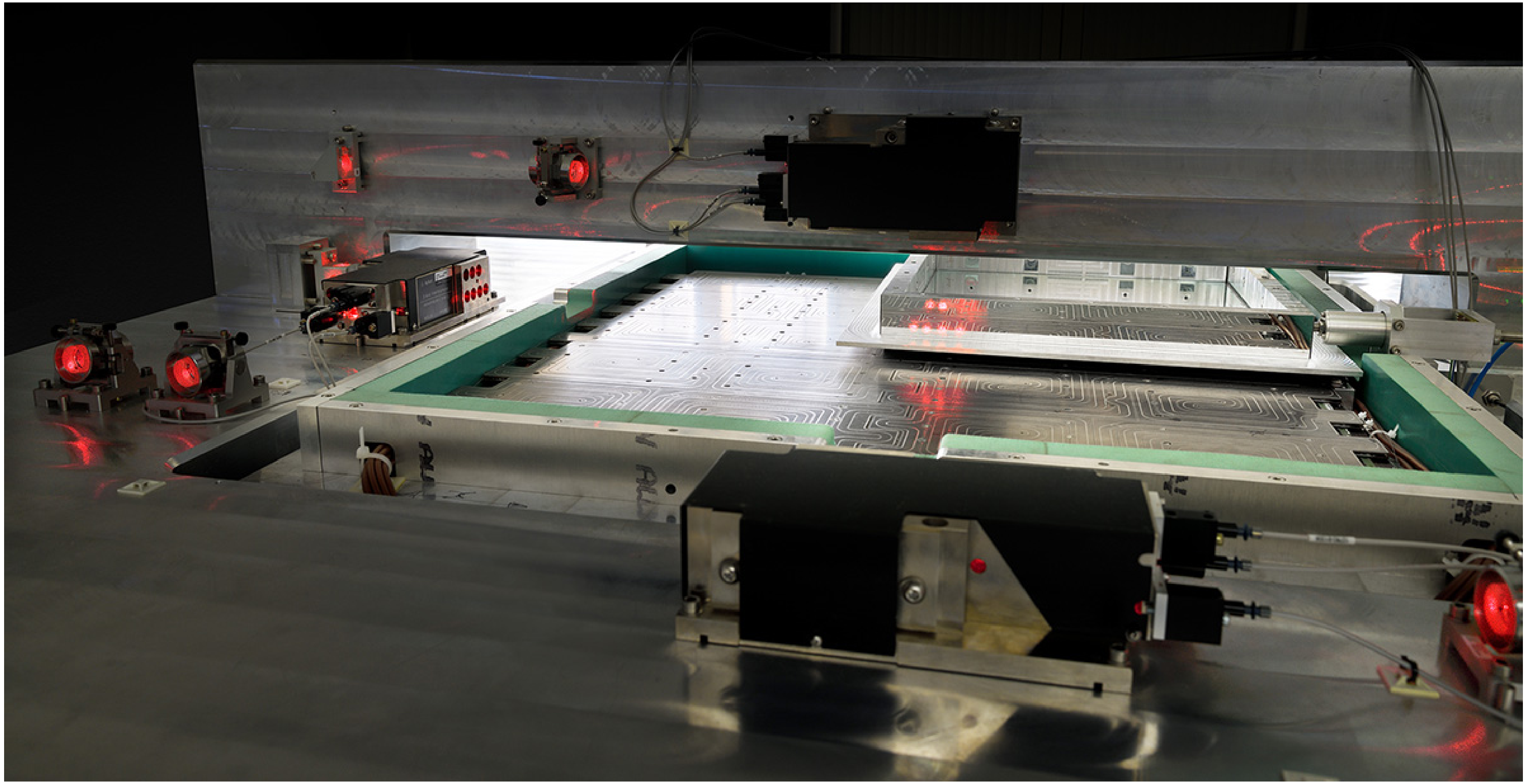
Can machine learning offer the same guarantees as the classical frameworks?





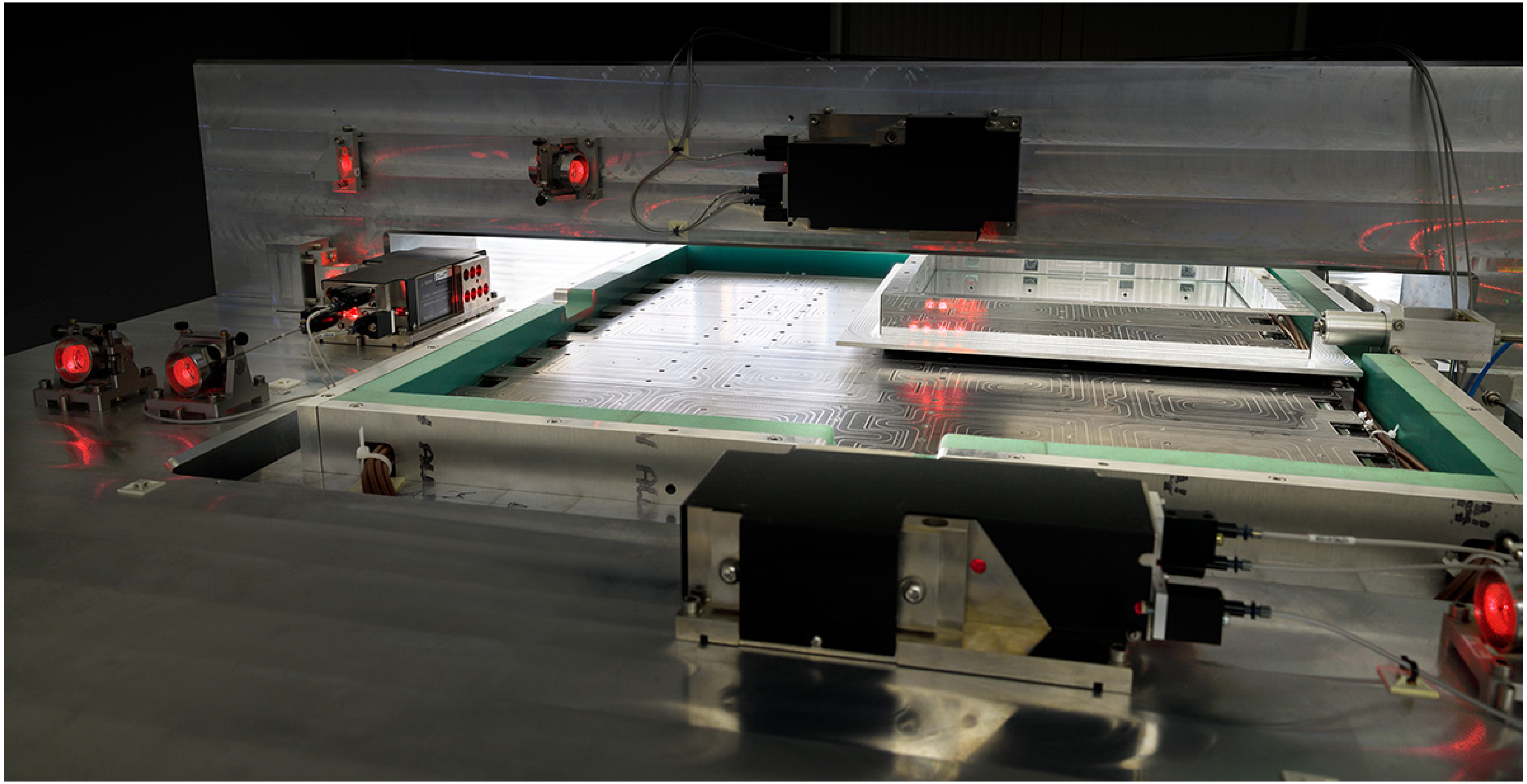


6-DOF High-Precision Motion System



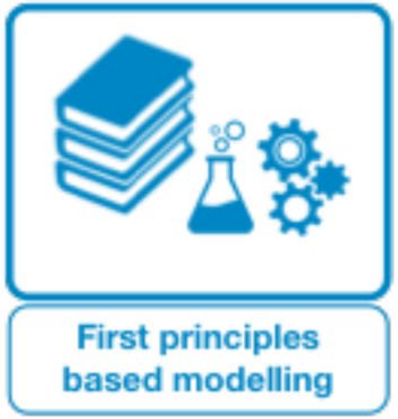
Developed by EPE and CS

6-DOF High-Precision Motion System

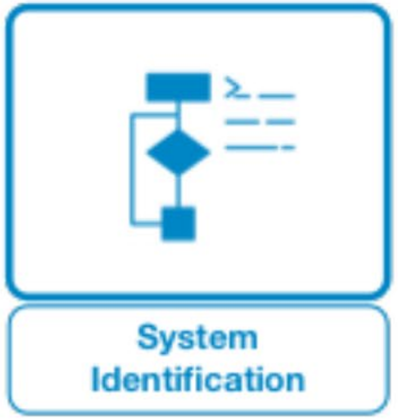


Achieve nm-accurate motion of a magnetically levitated stage with high-velocity and acceleration

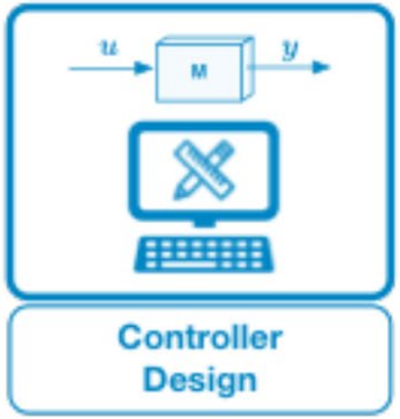
6-DOF High-Precision Motion System



High-fidelity ME & EM model
(Fourier series + FEM)

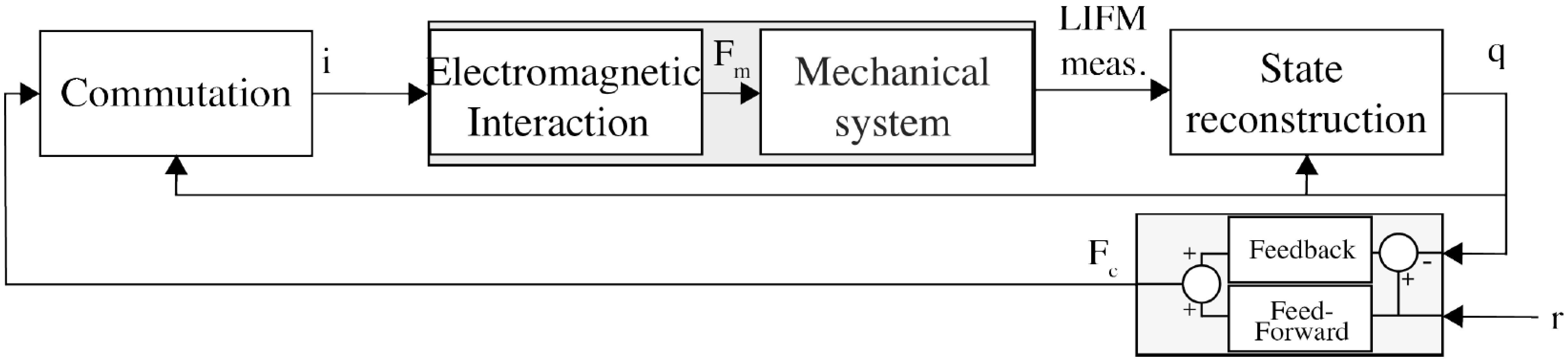


System identification
(FRF modelling)

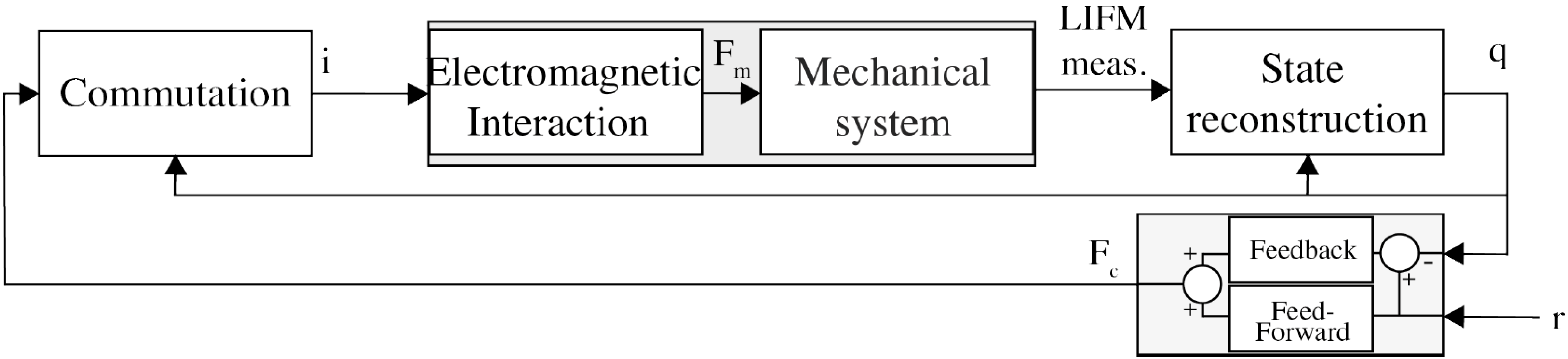


Sophisticated
control design

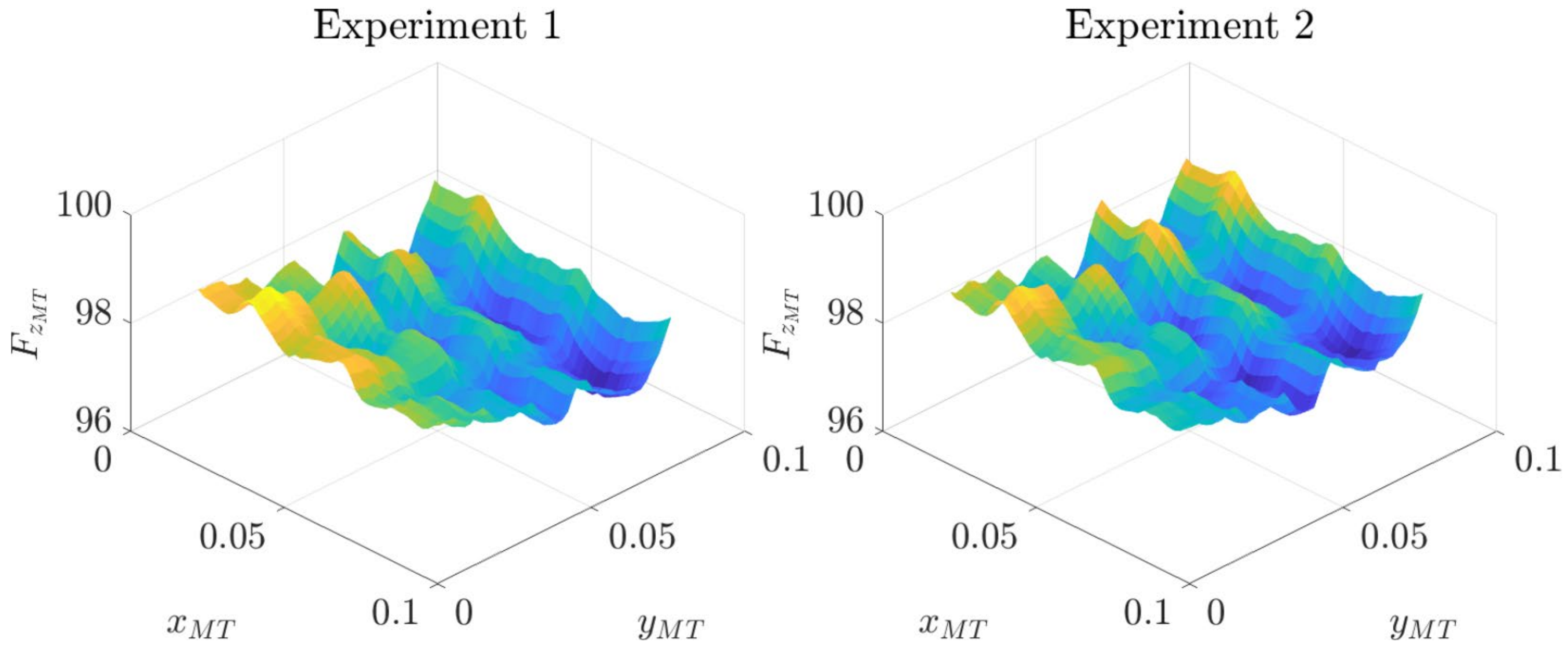
6-DOF High-Precision Motion System



6-DOF High-Precision Motion System

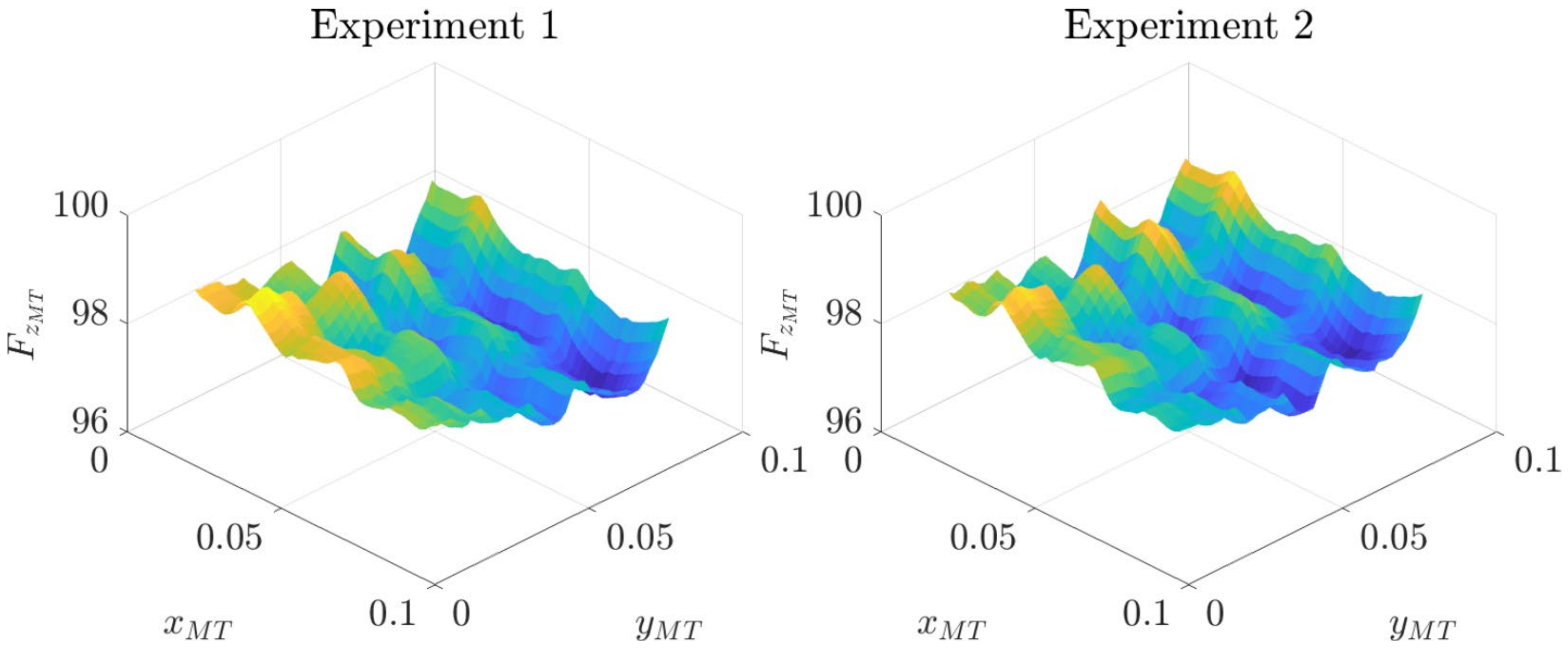


6-DOF High-Precision Motion System



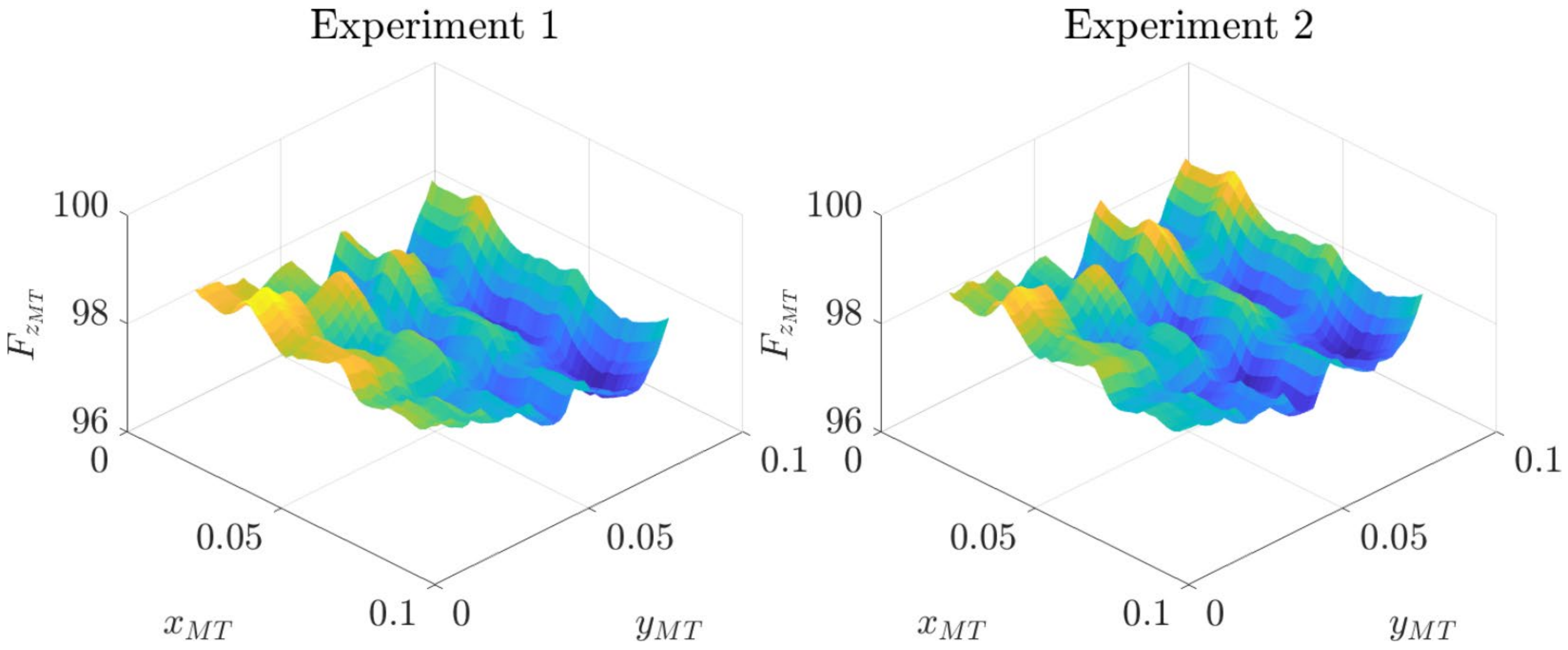
Reasons: Coil Pitch, Metrology frame / Stator Alignment, Manufacturing errors, …

6-DOF High-Precision Motion System



Question: How can we learn this offset for feed-forward compensation?

6-DOF High-Precision Motion System



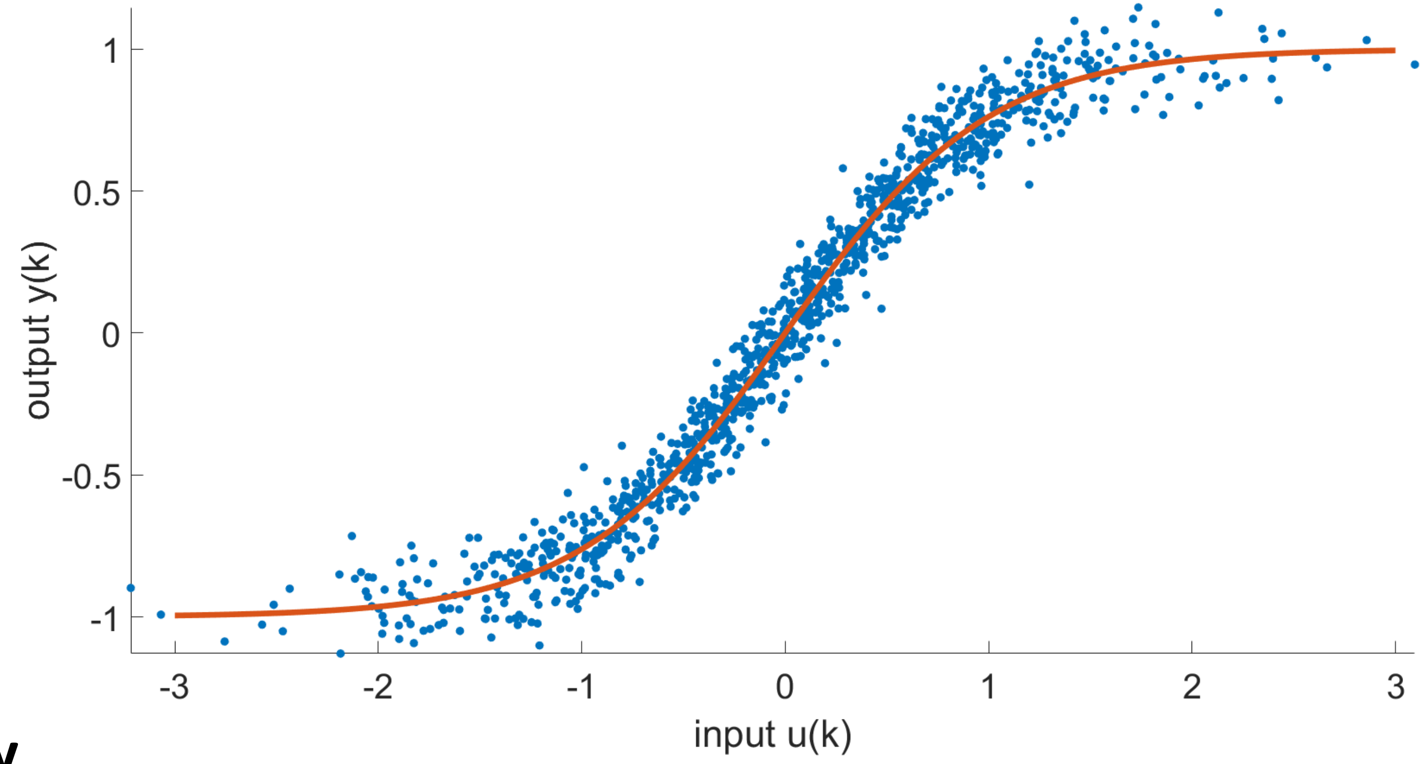
Question: How can we learn this offset for feed-forward compensation?

Machine Learning

Contents

Static vs. Dynamic

$$y_k = f(u_k) + e_k$$



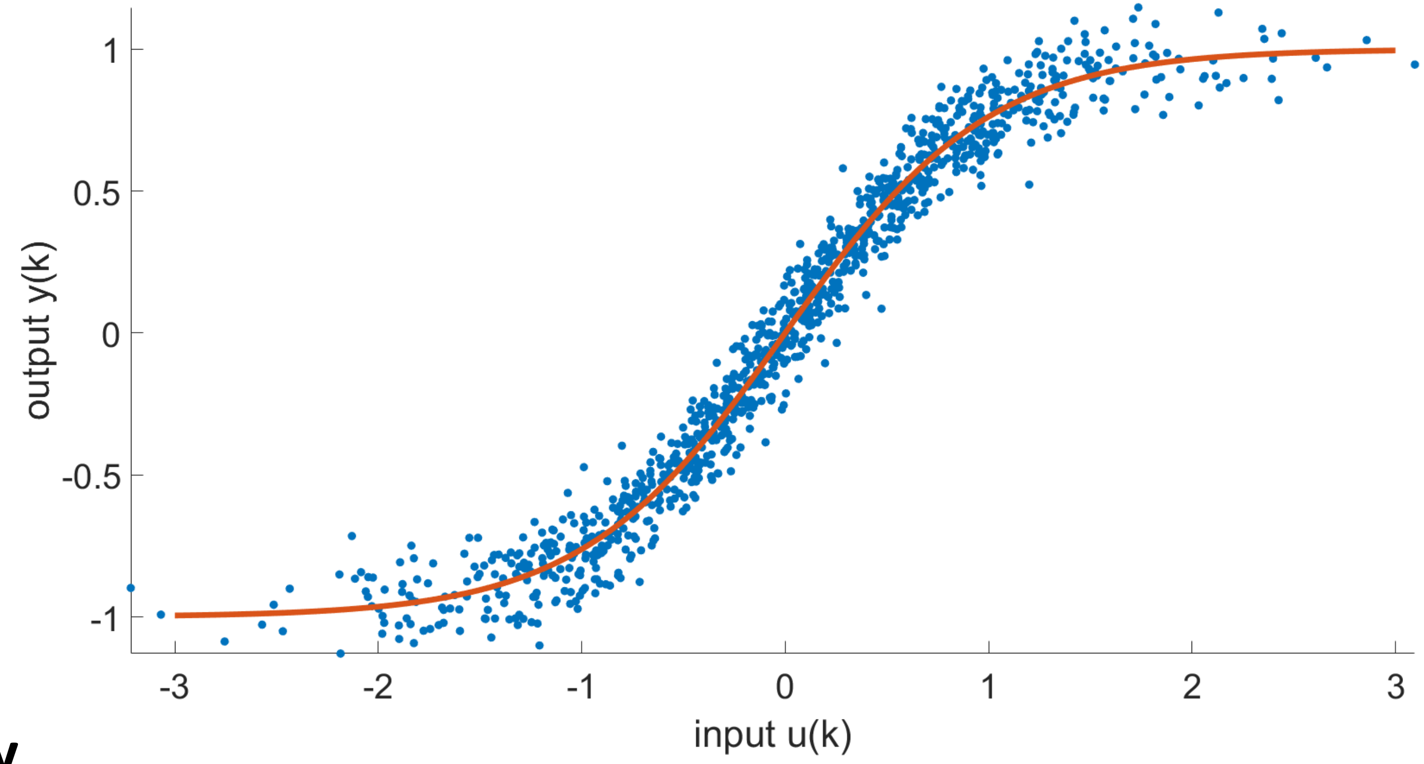
Static Nonlinearity

Output only depends on the instantaneous input(s)
Easy to analyze (for low-dimensional inputs)
Easy to estimate

Static vs. Dynamic

$$y_k = f(u_k) + e_k$$

Example: polynomial nonlinearity
(see exercise session)

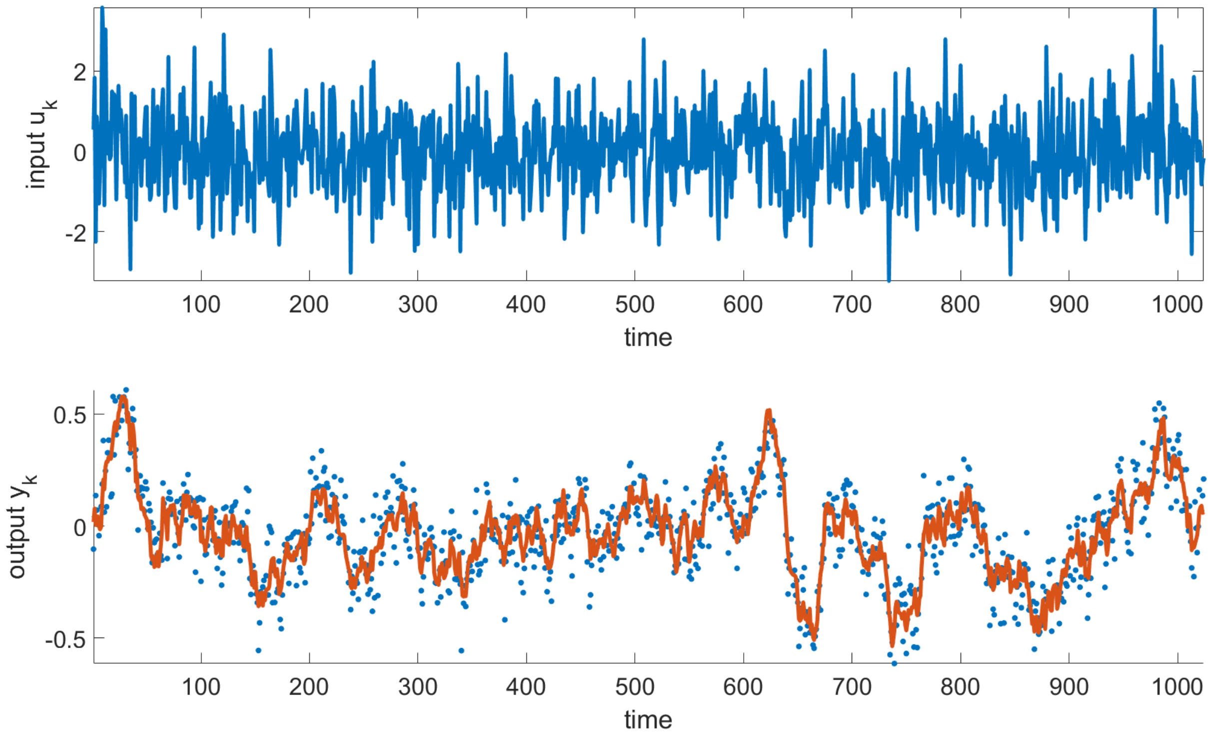


Static Nonlinearity

Output only depends on the instantaneous input(s)
Easy to analyze (for low-dimensional inputs)
Easy to estimate

Static vs. Dynamic

$$y_k = \sum_{i=1}^{n} g_i u_{k-i} + v_k$$



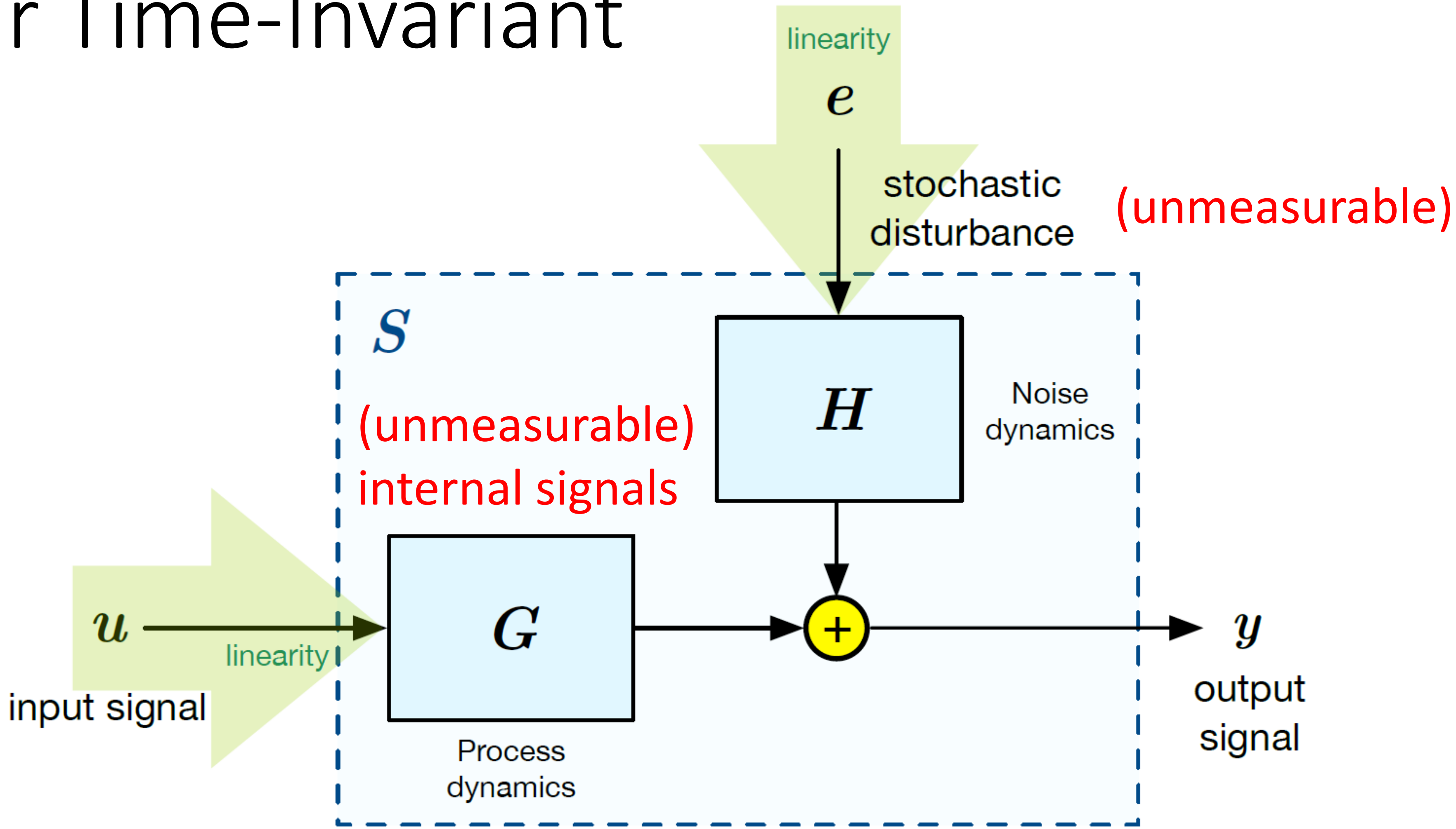
Dynamic Relation

Output depends past input, past output and/or system states

Analysis possible for linear dynamics (harder for nonlinear dynamics)

More difficult to estimate

Linear Time-Invariant



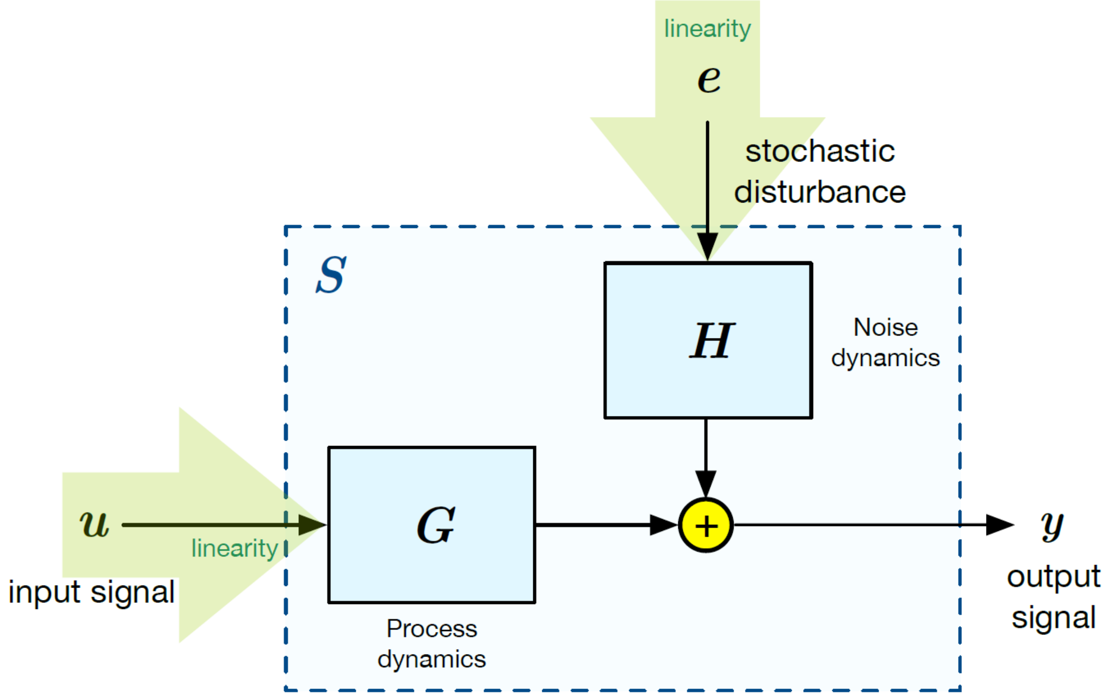
$u : \mathbb{Z} \to \mathcal{U} \subseteq \mathbb{R}$

(measurable) input signal

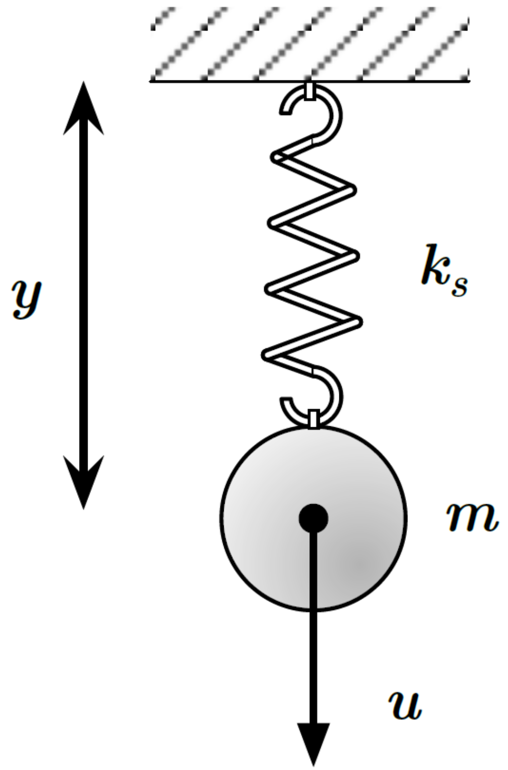
$y : \mathbb{Z} \to \mathcal{Y} \subseteq \mathbb{R}$

(measurable) output signal

Linear Time-Invariant



Example: Linear Spring



Choice of input / output partition might not be unique

Most often, causal relationships are considered

Stationary signals: statistical properties do not change over time

Representing LTI Dynamics

Impulse Response:

$$y_k = \sum_{i=1}^{n} g_i u_{k-i} + v_k$$

v_k: colored zero-mean noise process (induced by white noise e_k)

ARX:

$$y_k = \sum_{i=1}^{n_a} a_j y_{k-i} + \sum_{j=0}^{n_b} b_j u_{k-j} + e_k$$

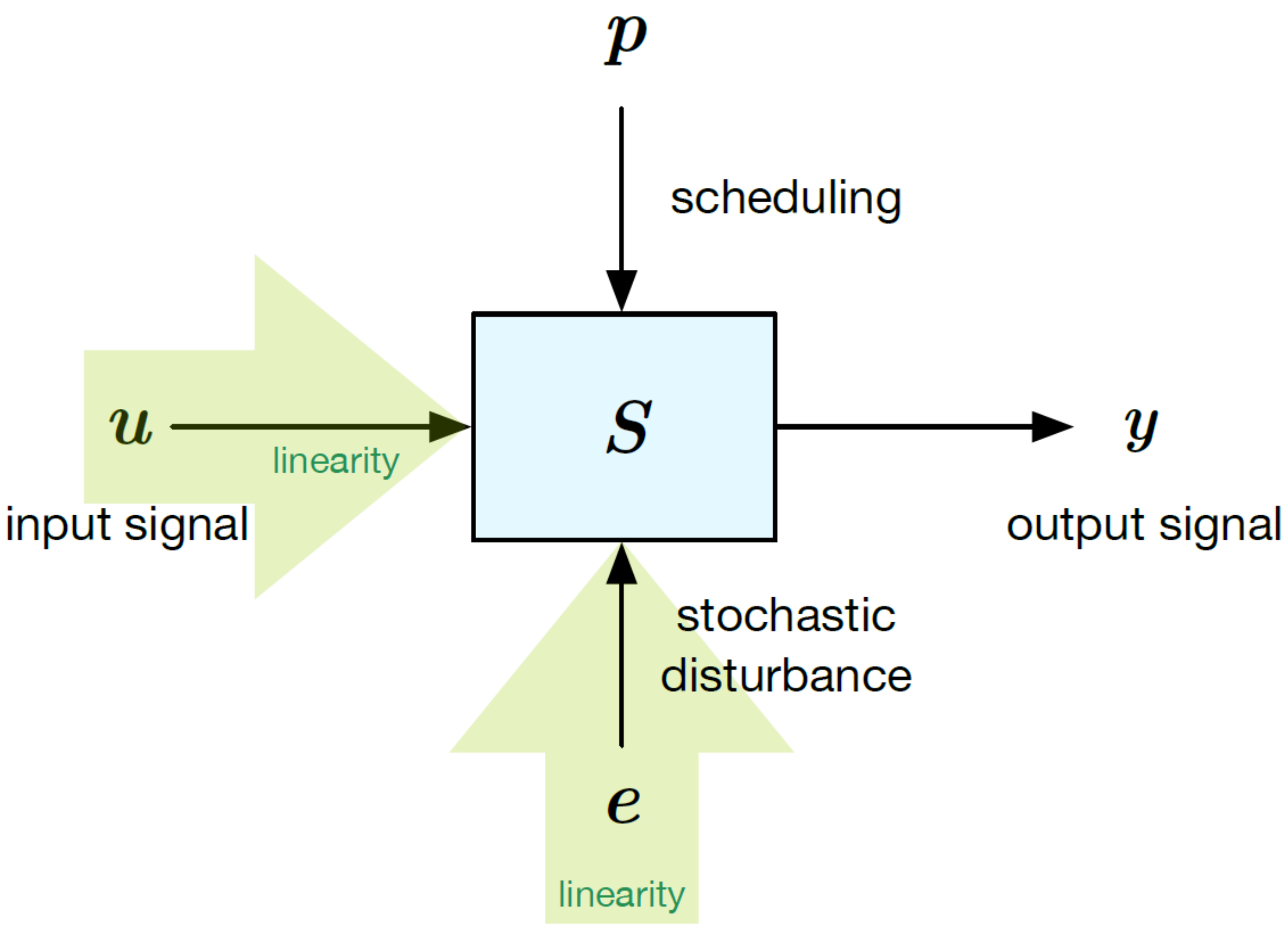
State-Space:

$$x_{k+1} = Ax_k + Bu_k + w_k$$
$$y_k = Cx_k + Du_k + e_k$$

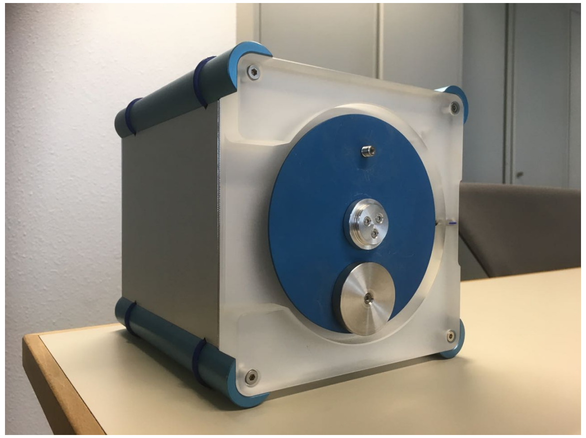
w_k: process noise

Many more: ARMAX, Output Error, Box Jenkins, Frequency Domain, ….

Linear Parameter-Varying



Example: Inverted Pendulum



Representing LPV Dynamics

Input-Output (IO) Representation:

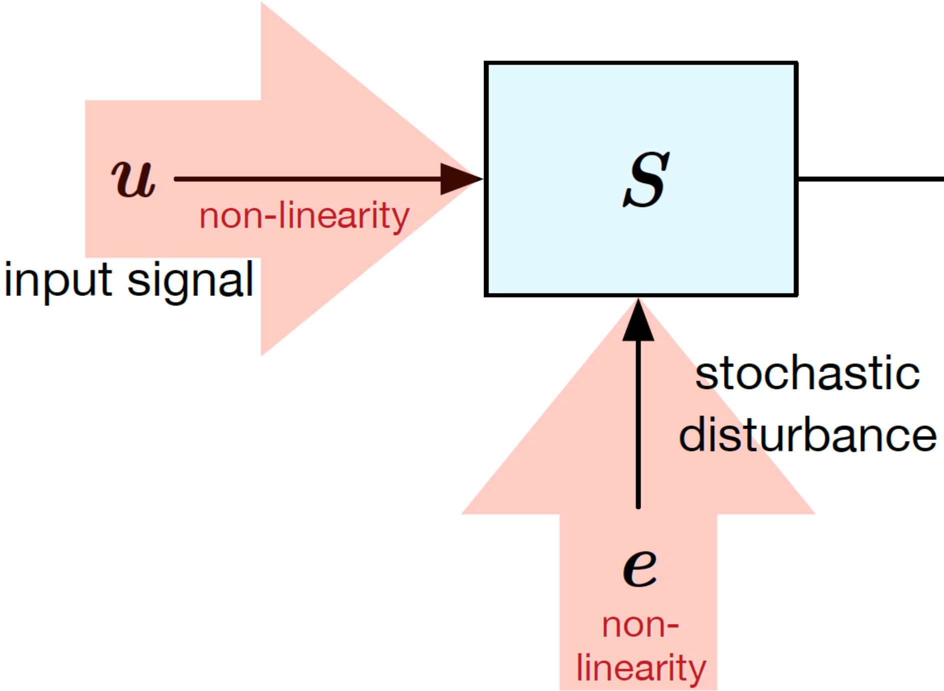
$$y_k = \sum_{i=1}^{n_\mathrm{a}} a_i(p_k) y_{k-i} + \sum_{i=0}^{n_\mathrm{b}} b_j(p_k) u_{k-j} + v_k$$

$\{a_i : \mathbb{P} \to \mathbb{R}\}_{i=1}^{n_\mathrm{a}}$ $\{b_j : \mathbb{P} \to \mathbb{R}\}_{j=1}^{n_\mathrm{b}}$ smooth coefficient functions

$p : \mathbb{Z} \to \mathbb{P} \subseteq \mathbb{R}^{n_\mathrm{p}}$ scheduling signal

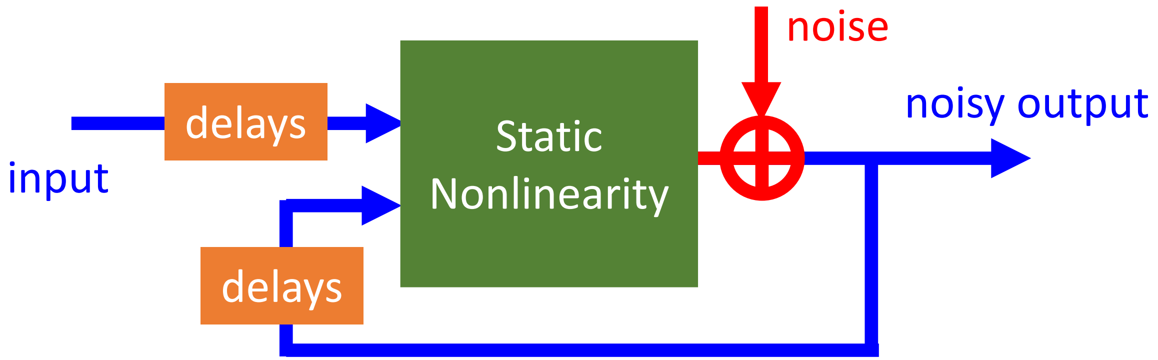
Many more: State-Space, LFR, Frequency Domain, ….

Nonlinear



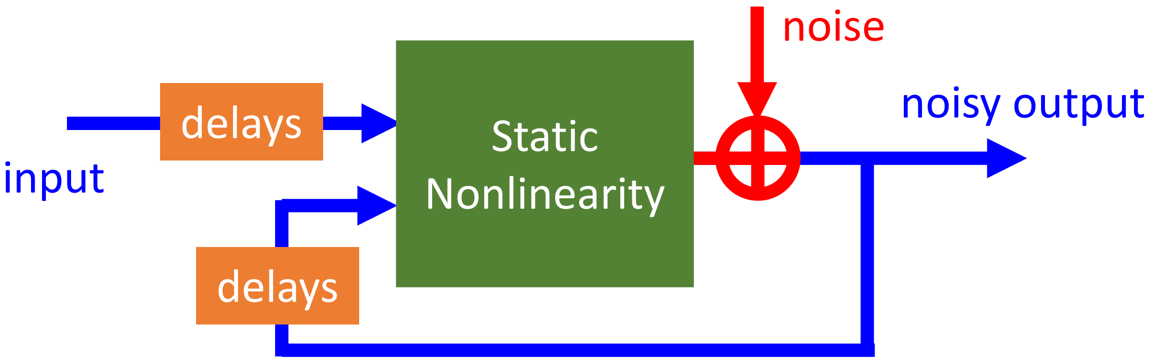
Example: Flexible Aerospace Structures

Nonlinear: NARX



$$y_k = f\left(u_k, u_{k-1}, \ldots, u_{k-n_b}, y_k, \ldots, y_{k-n_a}\right) + e_k$$

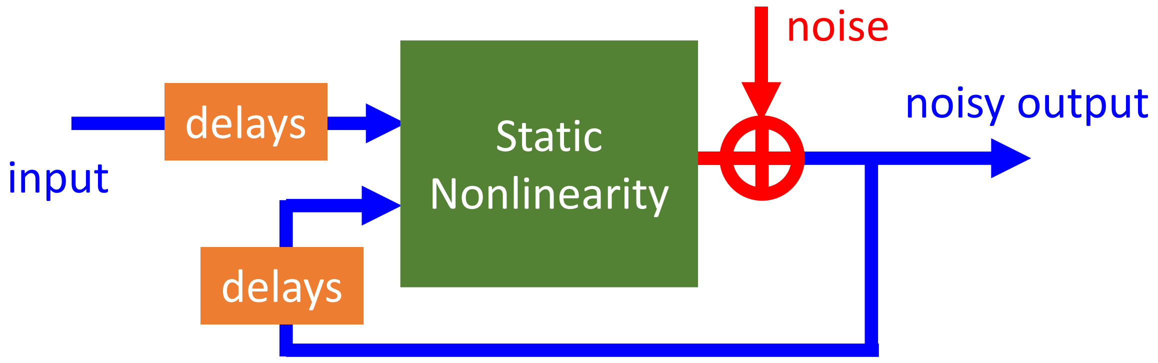
Nonlinear: NARX



$$\boxed{y_k} = \boxed{f}\boxed{(u_k, u_{k-1}, \ldots, u_{k-n_b}, y_k, \ldots, y_{k-n_a})}\boxed{+\, e_k}$$

To be estimated known unknown

Nonlinear: NARX



$$\boxed{y_k} = \boxed{f}\boxed{(u_k, u_{k-1}, \ldots, u_{k-n_b}, y_k, \ldots, y_{k-n_a})}\boxed{+\, e_k}$$

To be estimated known unknown

NARX (Nonlinear AutoRegressive with eXogenous input)

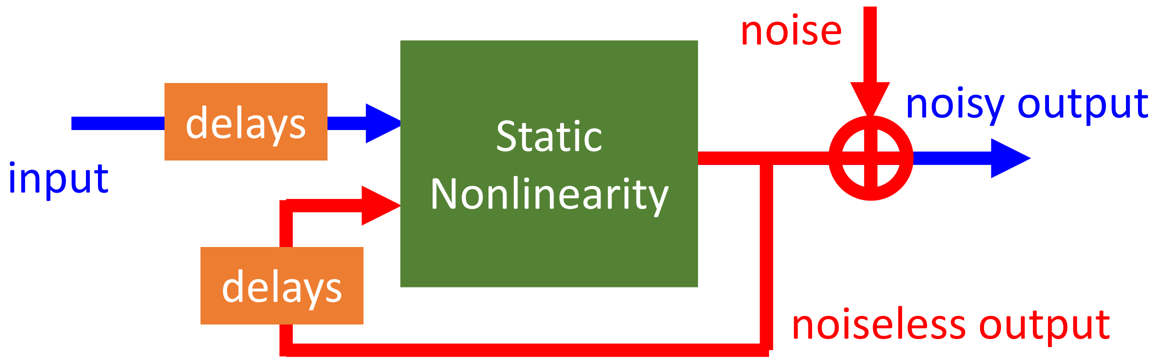
Output depends on past inputs and past measured noisy outputs

Difficult to analyze

Easy to estimate

Very particular noise structure

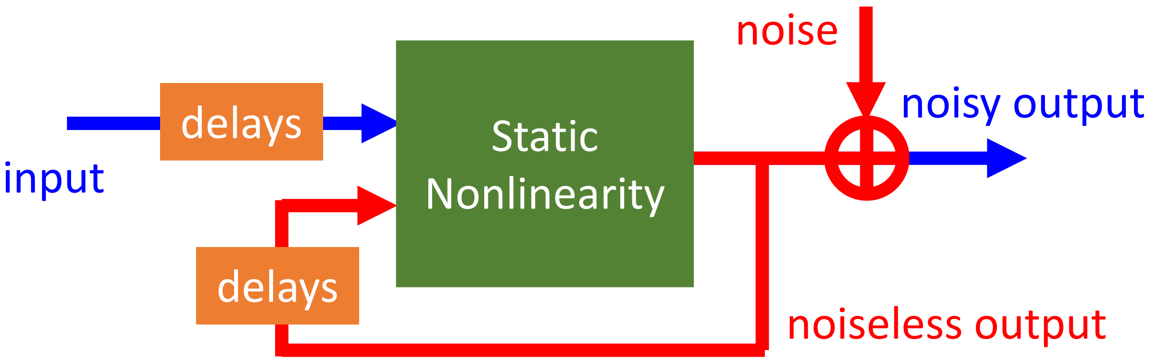
Nonlinear: NOE



$$s_k = f\left(u_k, u_{k-1}, \ldots, u_{k-n_b}, s_k, \ldots, s_{k-n_a}\right)$$

$$y_k = s_k + e_k$$

Nonlinear: NOE



To be estimated

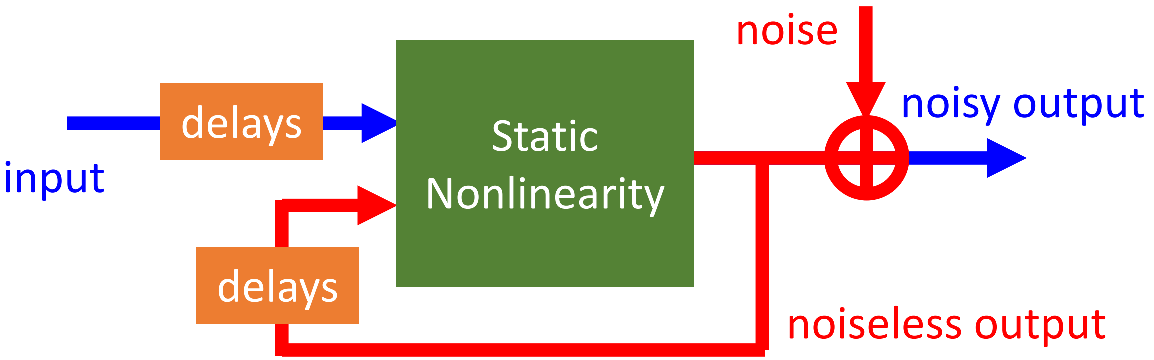
unknown

$$s_k = f(u_k, u_{k-1}, \ldots, u_{k-n_b}, s_k, \ldots, s_{k-n_a})$$

$$y_k = s_k + e_k$$

known

Nonlinear: NOE



To be estimated

unknown

$$\boxed{s_k} = \boxed{f}\boxed{(u_k, u_{k-1}, \ldots, u_{k-n_b}, s_k, \ldots, s_{k-n_a})}$$

$$\boxed{y_k} = \boxed{s_k + e_k}$$

known

NOE (Nonlinear Output Error)

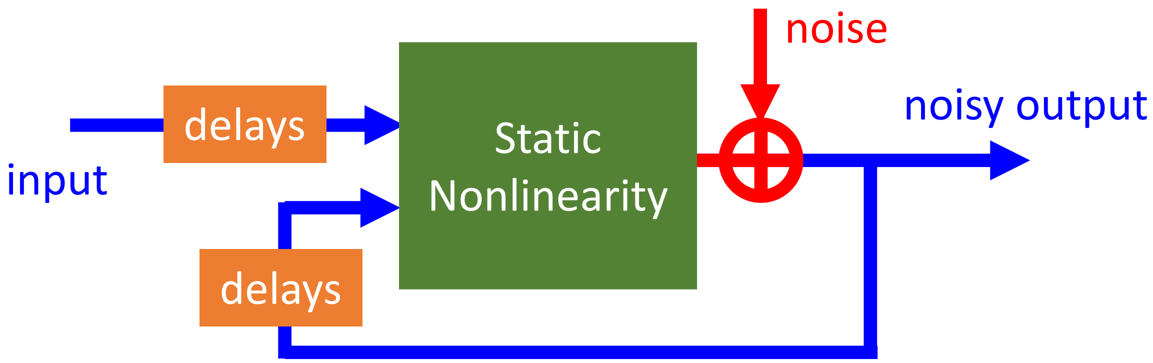
Output depends on past inputs and past unknown noiseless outputs

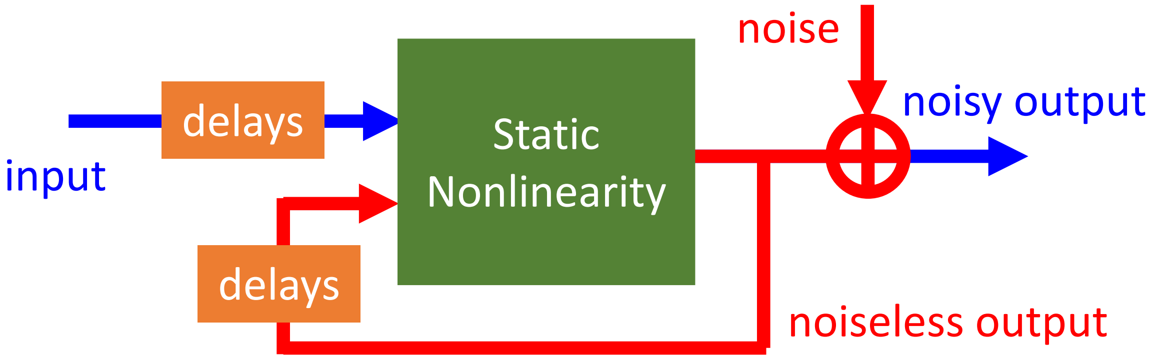
Difficult to analyze

Difficult to estimate

Straightforward noise structure

Nonlinear: NARX vs NOE





$$y_k = f\left(u_k, u_{k-1}, \ldots, u_{k-n_b}, y_k, \ldots, y_{k-n_a}\right) + e_k$$

$$s_k = f\left(u_k, u_{k-1}, \ldots, u_{k-n_b}, s_k, \ldots, s_{k-n_a}\right)$$
$$y_k = s_k + e_k$$

Different Noise Model

Nonlinear: State-Space

$$
\begin{aligned}
x_{k+1} &= f\left(x_k, u_k\right) + w_k \\
y_k &= h\left(x_k, u_k\right) + e_k
\end{aligned}
$$

Nonlinear: State-Space

unknown states

known input

$$x_{k+1} = f\left(x_k, u_k\right) + w_k$$

$$y_k = h\left(x_k, u_k\right) + e_k$$

process noise

output noise

known output

multivariate static nonlinearities

Nonlinear: State-Space

unknown states

known input

$$x_{k+1} = f\left(x_k, u_k\right) + w_k$$

$$y_k = h\left(x_k, u_k\right) + e_k$$

process noise

output noise

known output

multivariate static nonlinearities

Nonlinear State-Space

- Output depends on the instantaneous inputs and (unknown) states
- Some analysis possibilities
- Difficult to estimate (due to unmeasured states)
- Multiple noise structures possible (output and process noise)

Contents

Discrete Probability Distribution

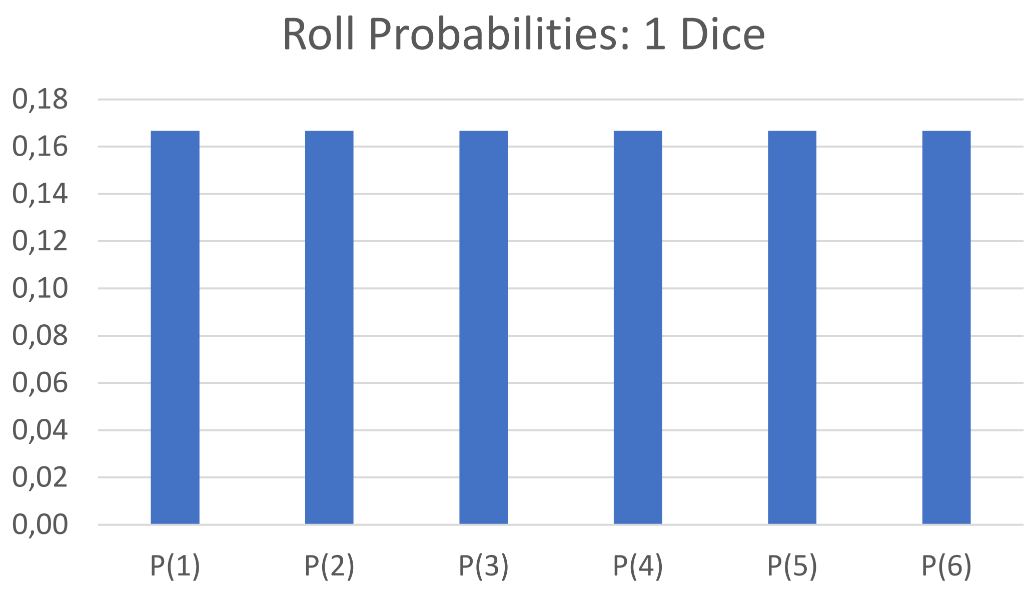
Expectation

Sum of the probability of all possible outcomes = 1

$$\sum_{x \in \mathbf{x}} P(\mathbf{x} = x) = 1$$

Expectation:

$$\mu_{\mathbf{x}} = \mathbb{E}\{\mathbf{x}\} = \sum_{x \in \mathbf{x}} x P(\mathbf{x} = x)$$



$\mathbf{x}$ is the random variable and x is a realization.

Random Variable

A variable whose possible values are the result of a random process.

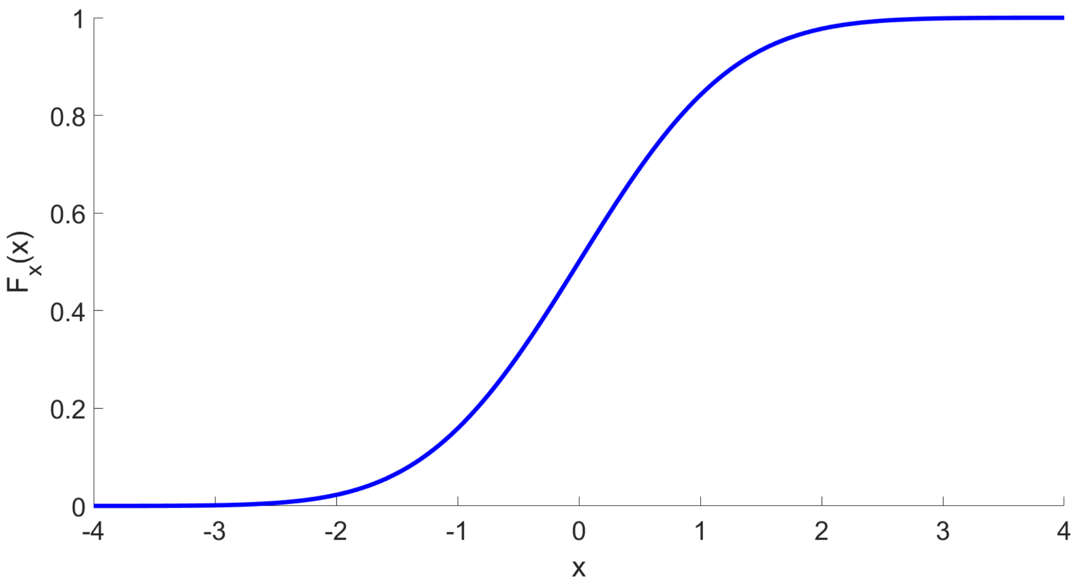
It is a function that maps the result of an unpredictable process to numerical quantities.

Random variable:

A random variable $\mathbf{x}$ is a function $S \to \mathbb{R}$, with S the sample space, such that for all $\lambda \in S$, $\mathbf{x}(\lambda) \in \mathbb{R}$ satisfies

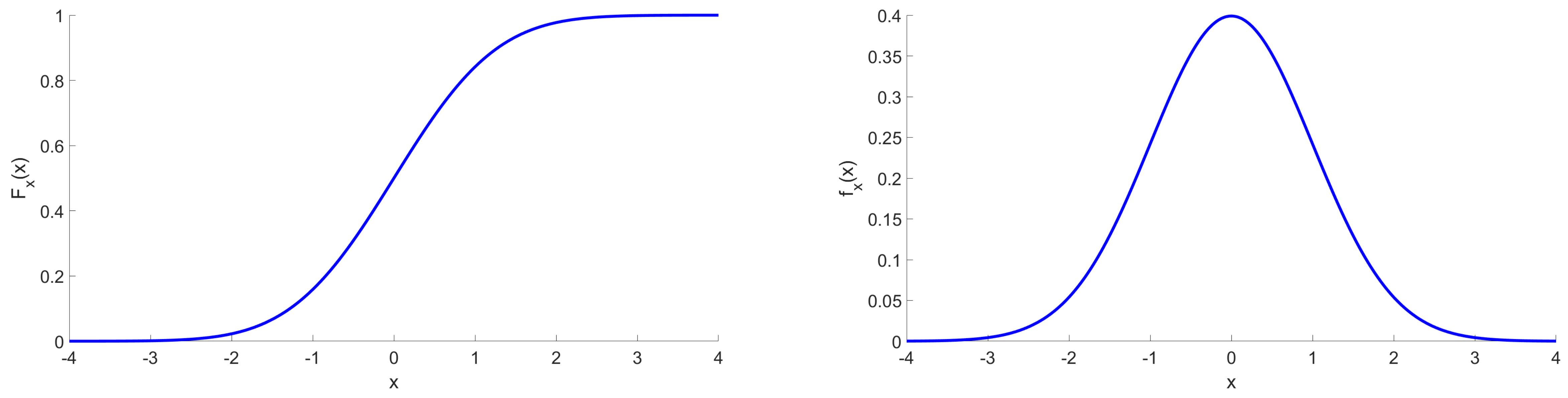
- The set $\{\lambda : \mathbf{x}(\lambda) \leq x\}$ is an event $\forall x \in \mathbb{R} \quad \Rightarrow \quad$ probability
- $P\{\lambda : \mathbf{x}(\lambda) = \infty\} = 0$; $P\{\lambda : \mathbf{x}(\lambda) = -\infty\} = 0$

(Cumulative) Distribution Function



Distribution function: $F_{\mathbf{x}}(x) := P(\mathbf{x} \leq x)$

Probability Density Function (PDF)



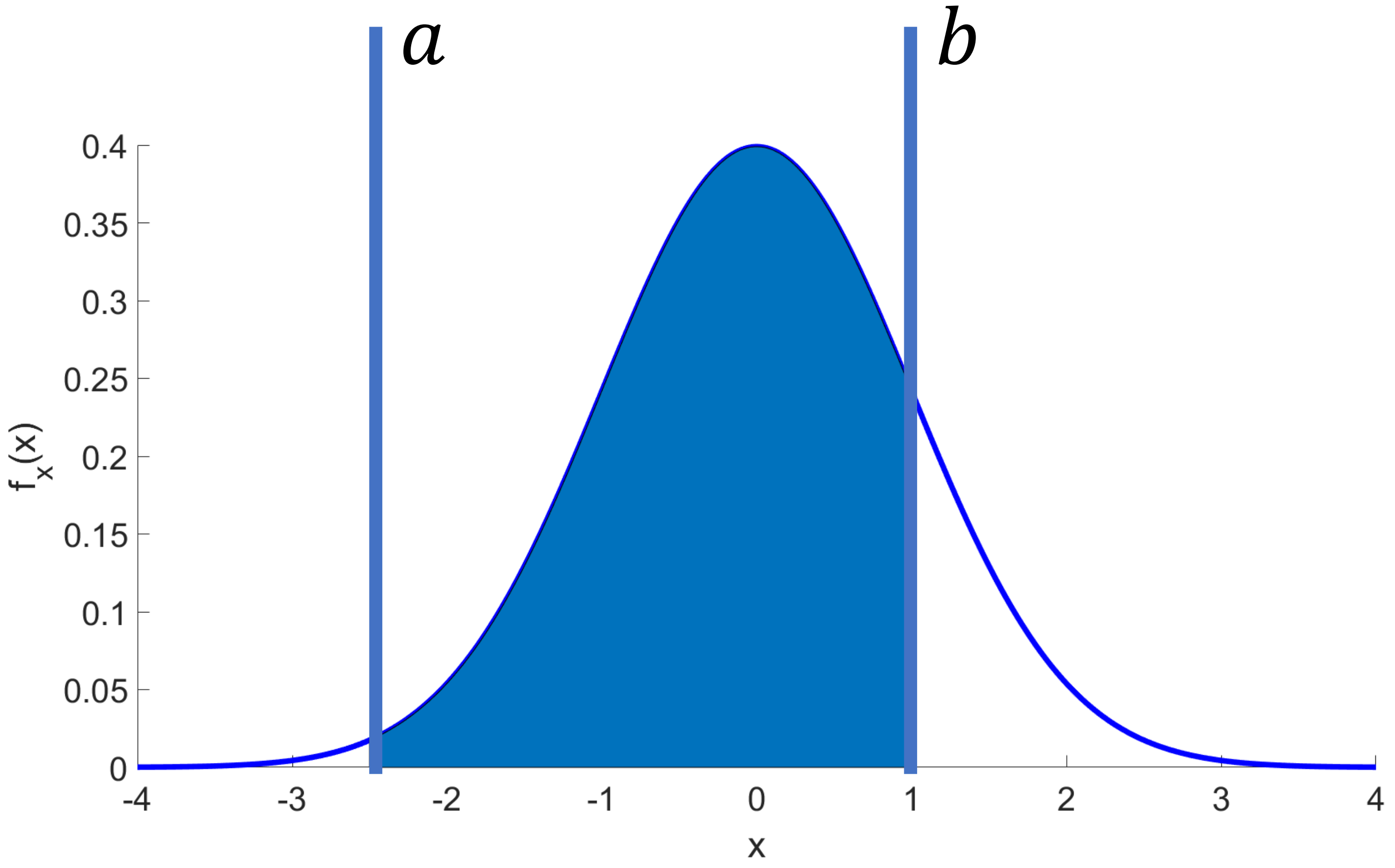
Distribution function: $F_{\mathbf{x}}(x) := P(\mathbf{x} \leq x)$

Probability density function (pdf): $f_{\mathbf{x}}(x) := \dfrac{dF_{\mathbf{x}}(x)}{dx}$

Probability Density Function (PDF)

$$\int_{-\infty}^{\infty} f_{\mathbf{x}}(x)\ dx = 1$$

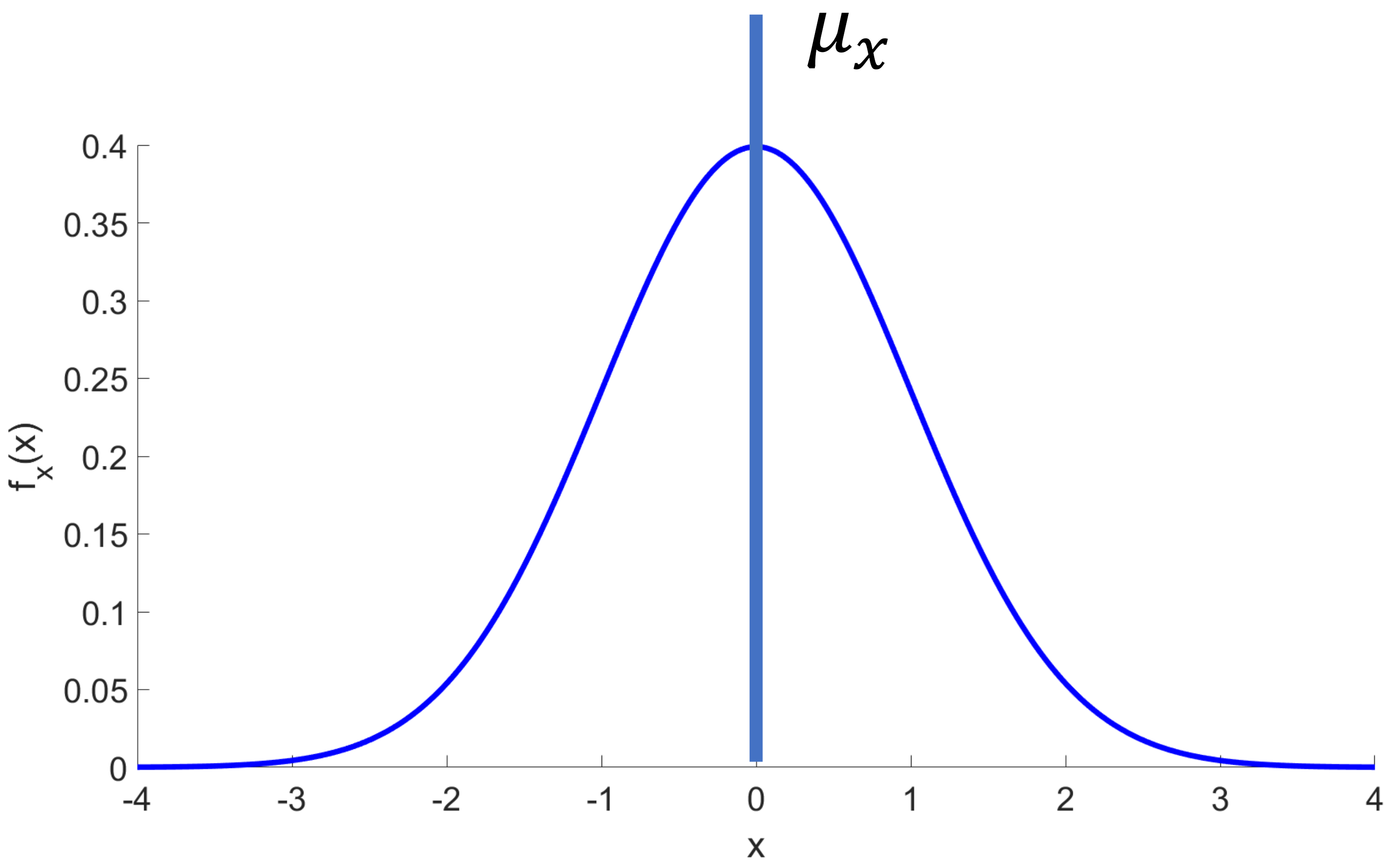
$$P(a \leq \mathbf{x} \leq b) = \int_{a}^{b} f_{\mathbf{x}}(x)\ dx$$



Expectation

$$\mu_{\mathbf{x}} = \mathbb{E}\{\mathbf{x}\} = \sum_{x \in \mathbf{x}} x P(\mathbf{x} = x)$$

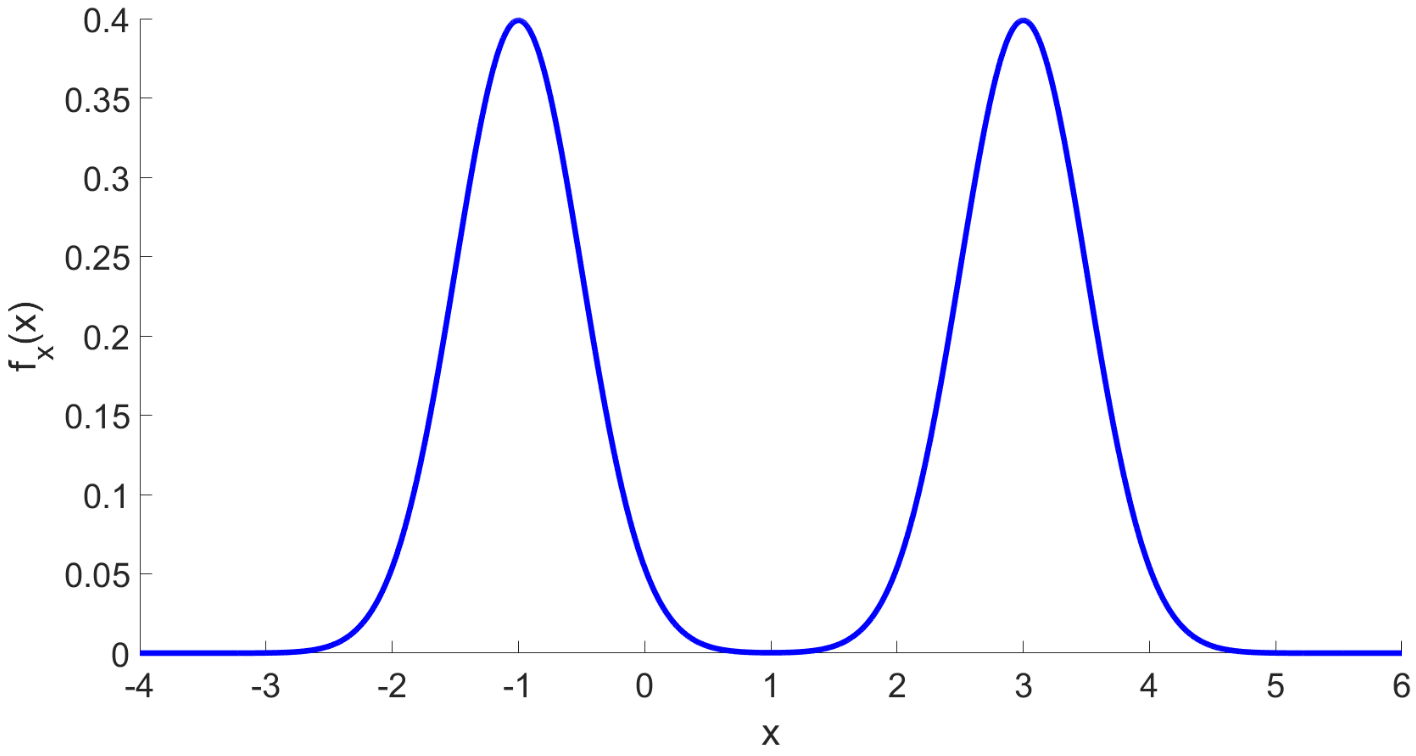
$$\mu_{\mathbf{x}} = \mathbb{E}\{\mathbf{x}\} = \int_{-\infty}^{\infty} x f_{\mathbf{x}}(x)\ dx$$



Expectation

$$\mu_{\mathbf{x}} = \mathbb{E}\{\mathbf{x}\} = \int_{-\infty}^{\infty} x f_{\mathbf{x}}(x)\ dx$$

What is the expected value?

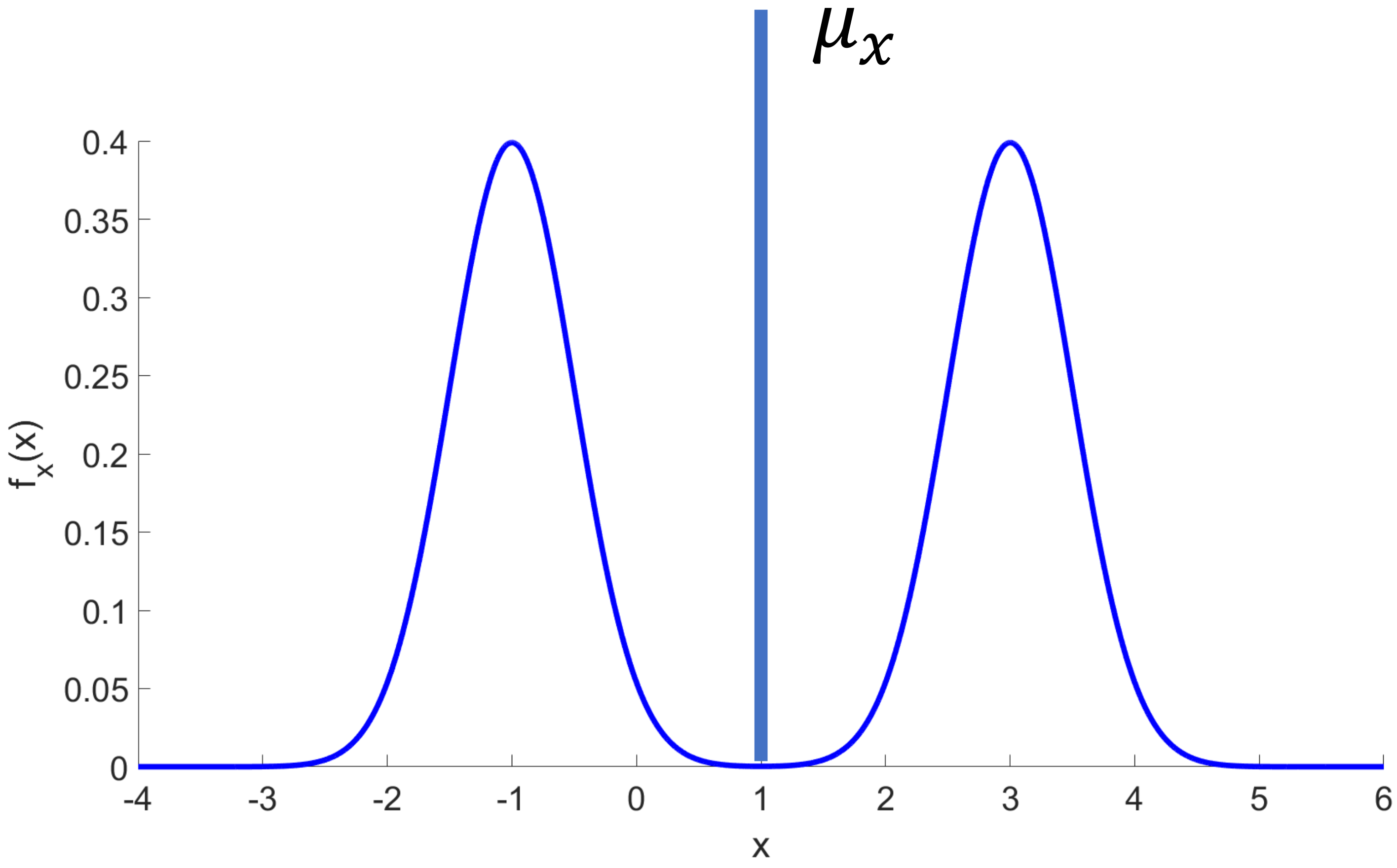


A: 1 B: -1 C: 3

D: Does Not Exist E: -1 and 3

Expectation

$$\mu_{\mathbf{x}} = \mathbb{E}\{\mathbf{x}\} = \int_{-\infty}^{\infty} x f_{\mathbf{x}}(x)\ dx$$

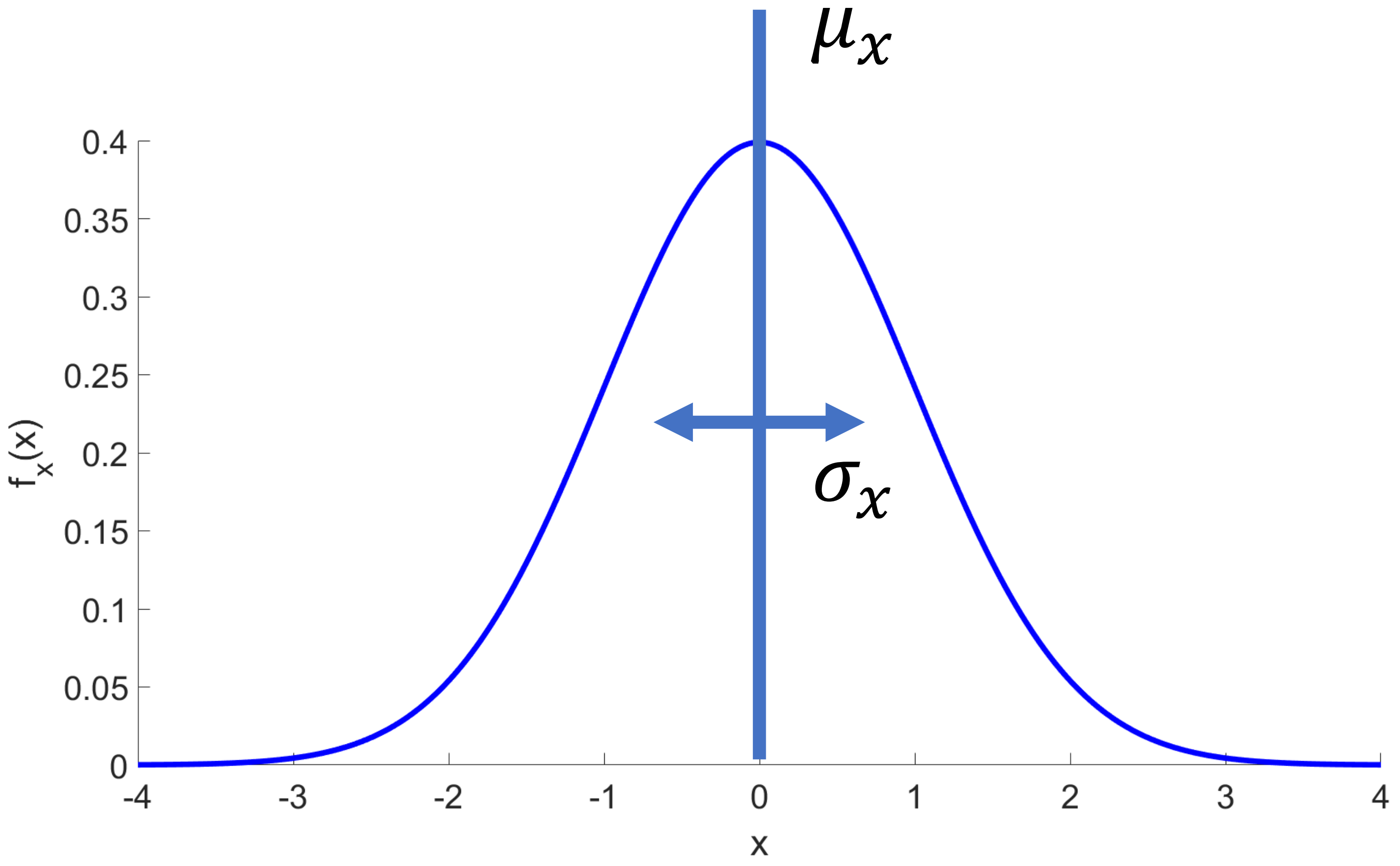


Expected value ≠ peaks or ‘modes’ of a distribution

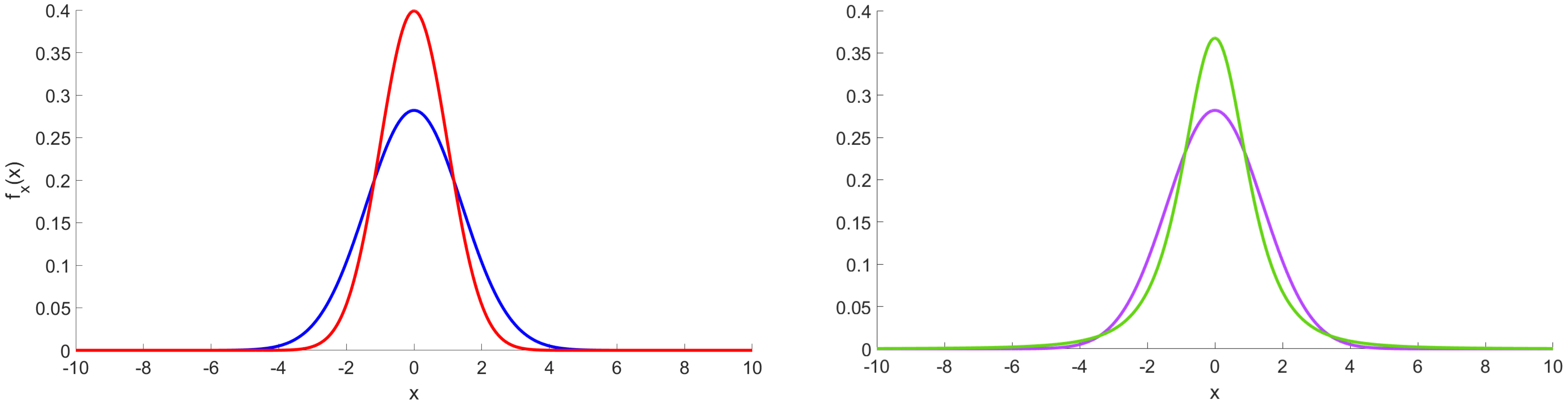
Variance

$$\mu_{\mathbf{x}} = \mathbb{E}\{\mathbf{x}\} = \int_{-\infty}^{\infty} x f_{\mathbf{x}}(x)\ dx$$

$$\begin{aligned}\sigma_{\mathbf{x}}^2 &= \mathbb{E}\{(\mathbf{x} - \mu_{\mathbf{x}})^2\} \\ &= \int_{-\infty}^{\infty} (x - \mu_{\mathbf{x}})^2 f_{\mathbf{x}}(x)\ dx\end{aligned}$$



Variance

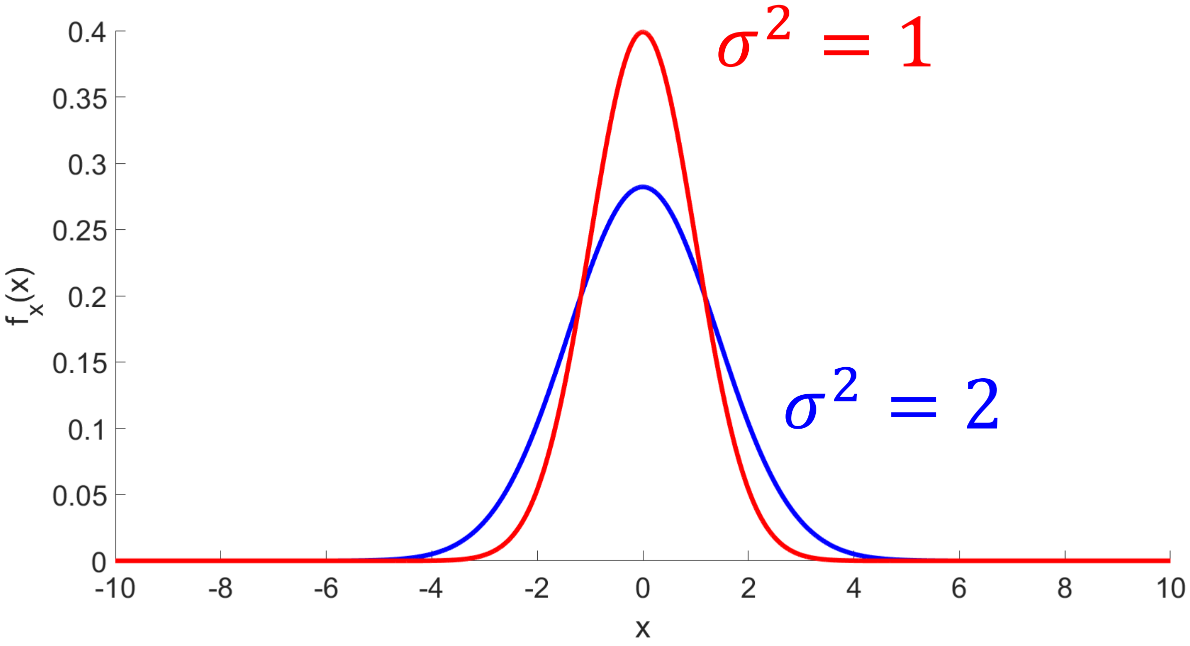


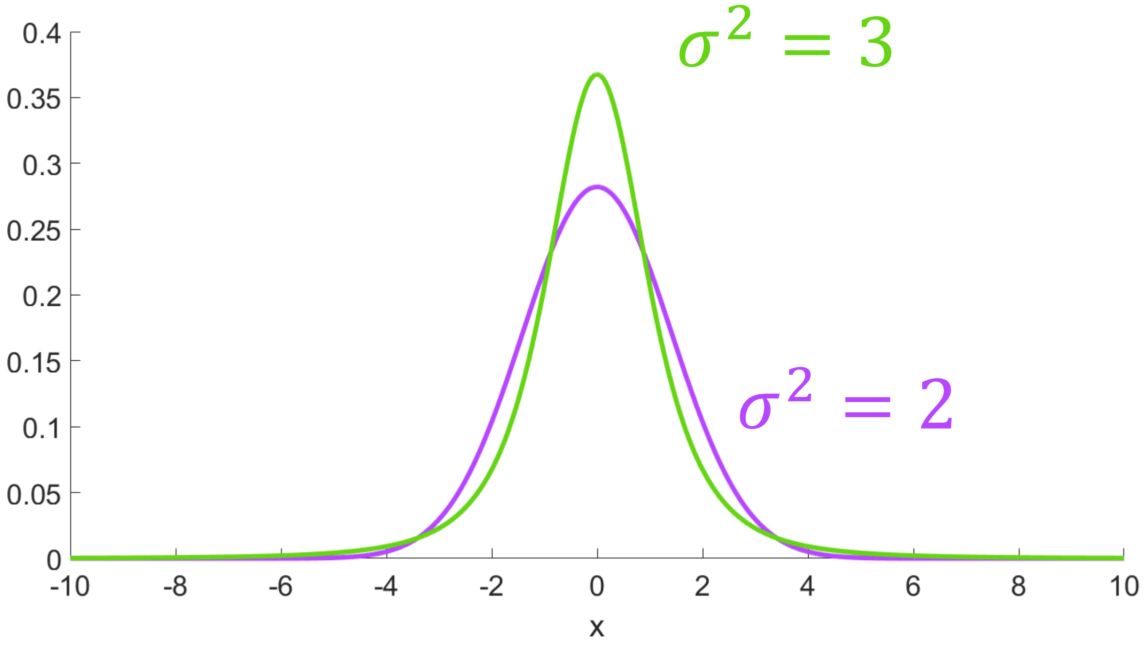
Highest variance?

A: Red and Green **B: Blue and Green**

C: Red and Magenta **D: Blue and Magenta**

Variance



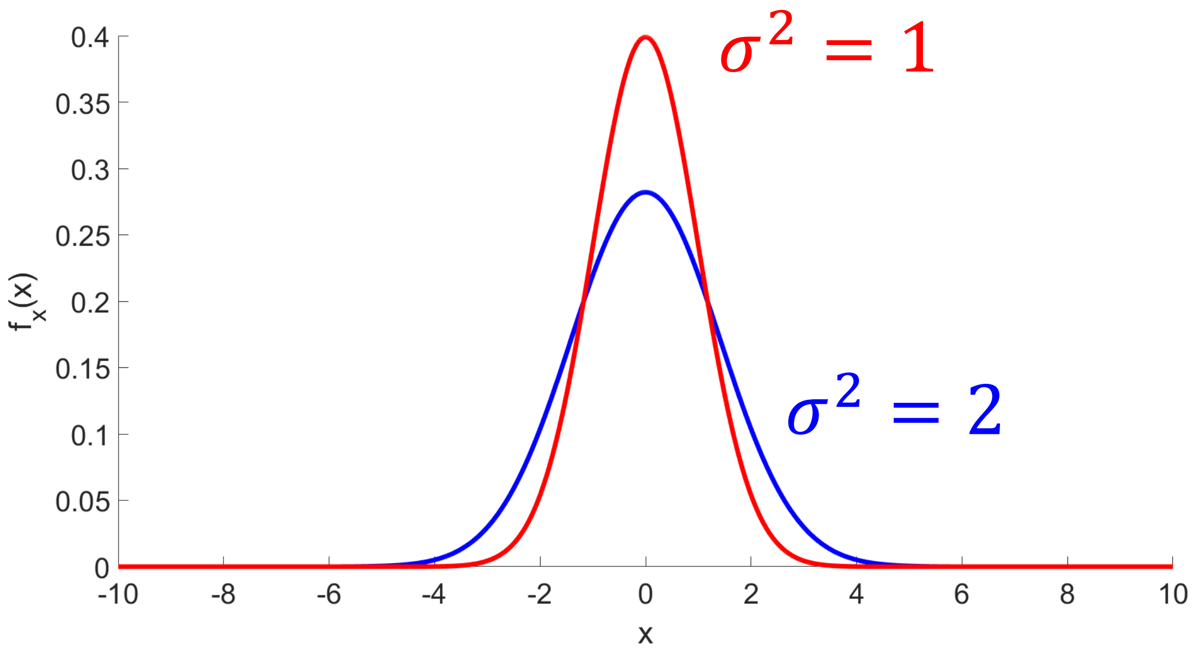


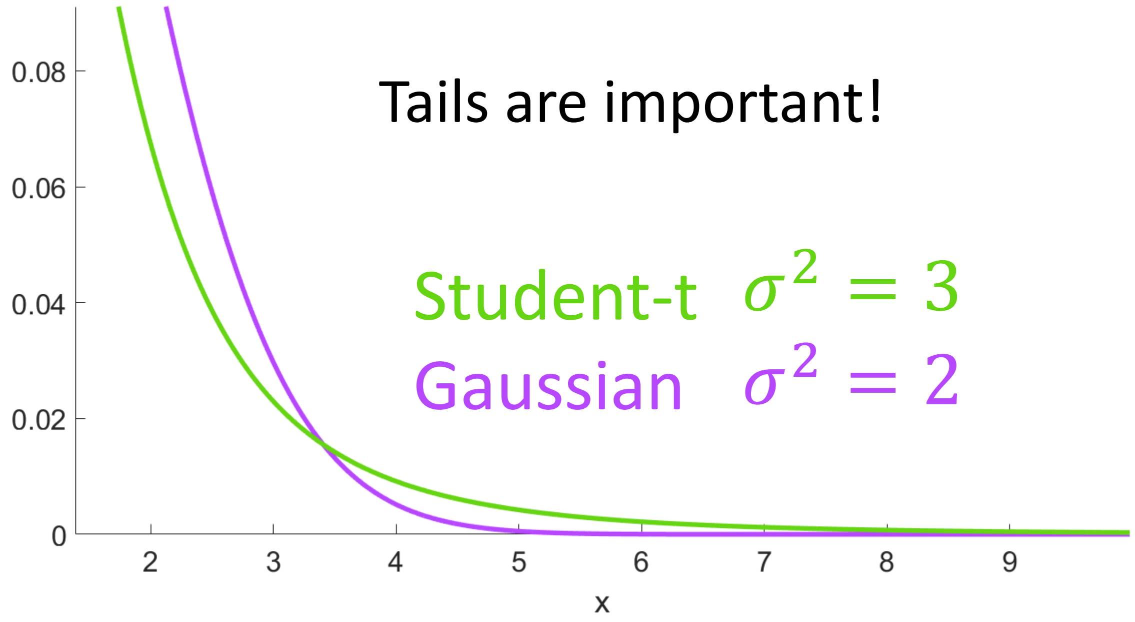
Highest variance?

A: Red and Green **B: Blue and Green**

C: Red and Magenta **D: Blue and Magenta**

Variance





Highest variance?

A: Red and Green **B: Blue and Green**

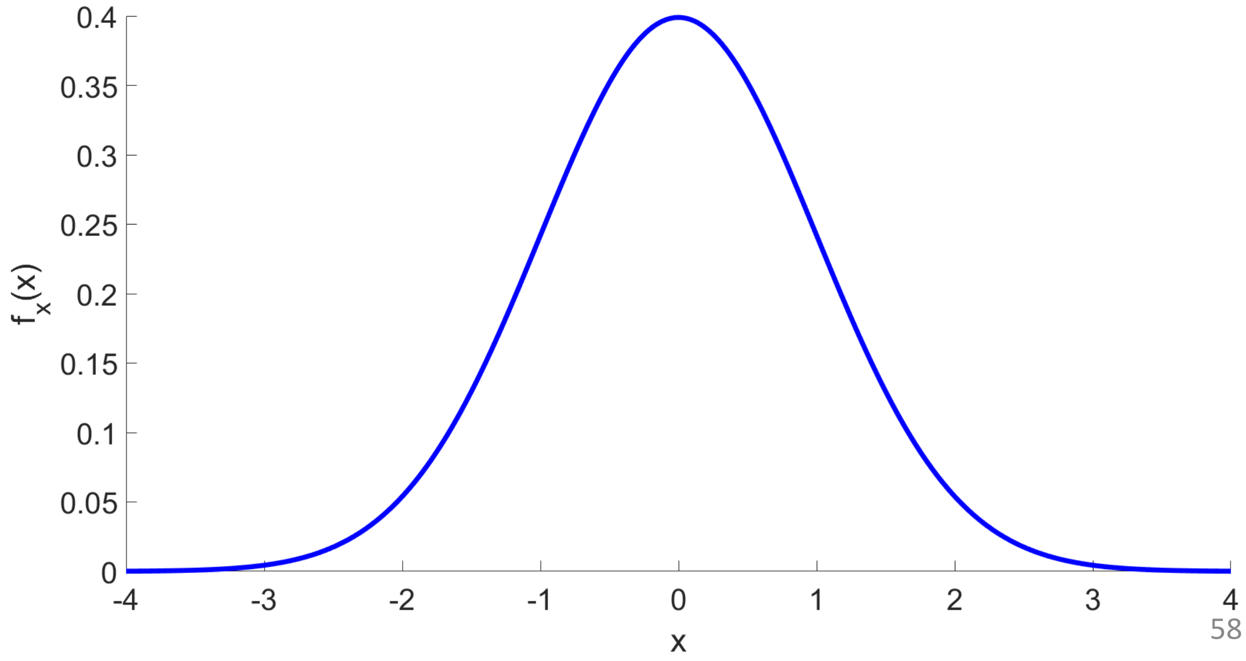
C: Red and Magenta **D: Blue and Magenta**

Gaussian (aka Normal) Distribution

$$\mathbf{x} \sim \mathcal{N}(\mu_{\mathbf{x}}, \sigma^2_{\mathbf{x}})$$

$$f_{\mathbf{x}}(x) = \frac{1}{\sqrt{2\pi\sigma^2_{\mathbf{x}}}}\, e^{-\frac{(x-\mu_{\mathbf{x}})^2}{2\sigma^2_{\mathbf{x}}}}$$

Completely defined by its mean and variance

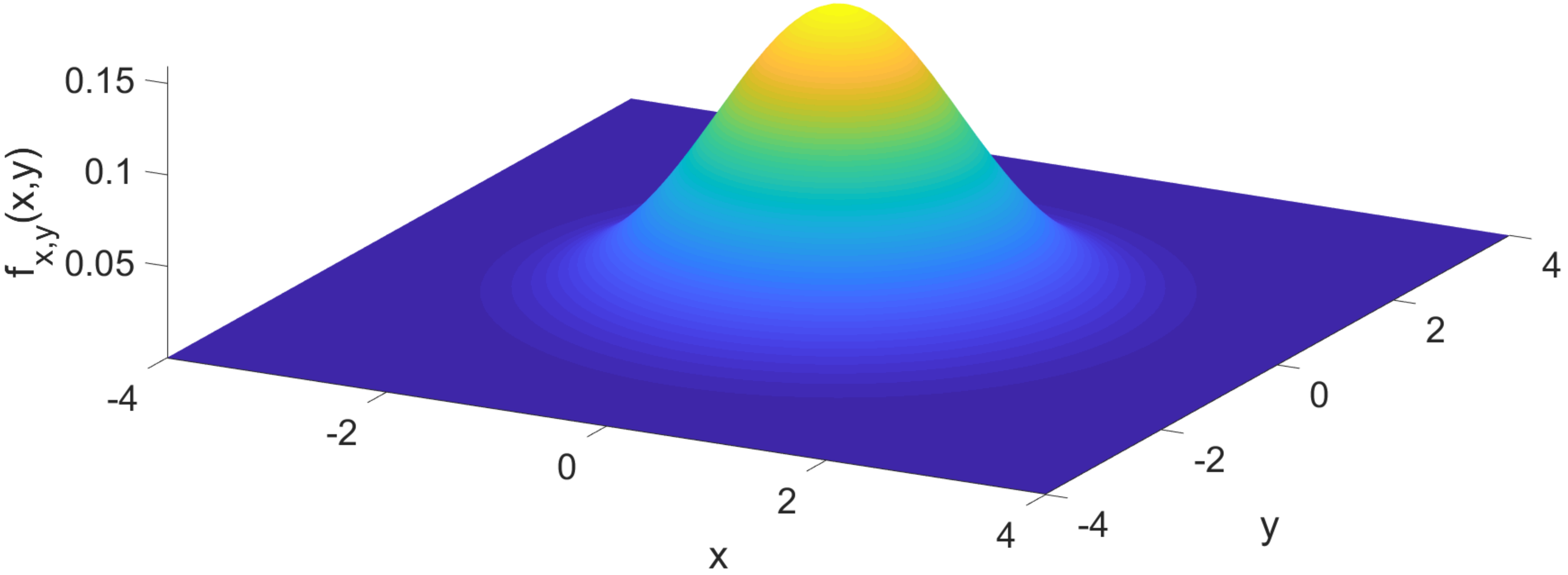


Other distributions:

Uniform

Student-t

Multivariate or Joint PDF



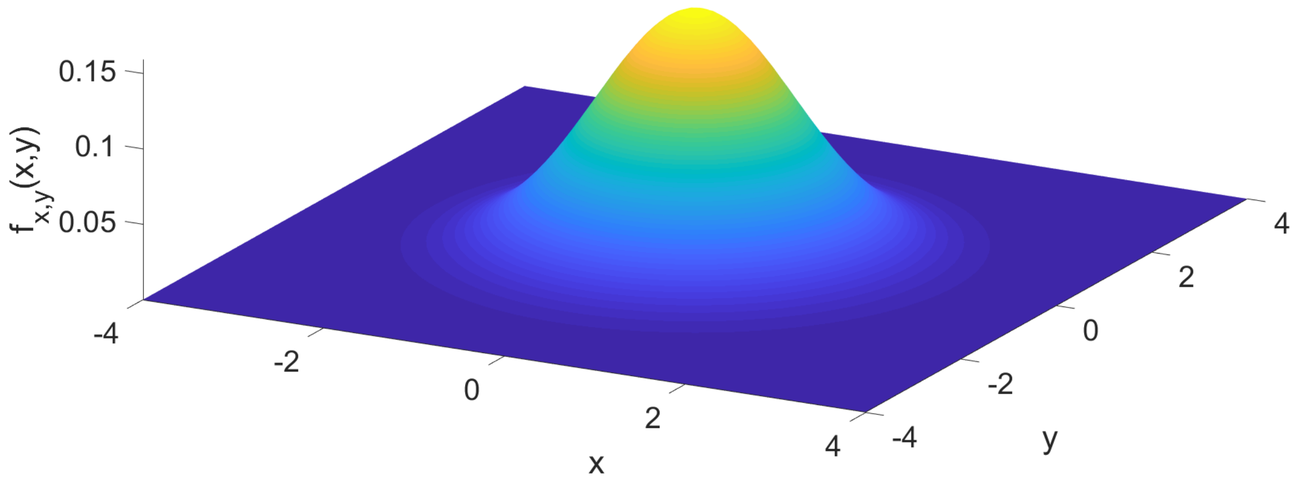
$$f_{\mathbf{x},\mathbf{y}}(x,y) := \frac{\partial^2 F_{\mathbf{x},\mathbf{y}}(x,y)}{\partial x \partial y}$$

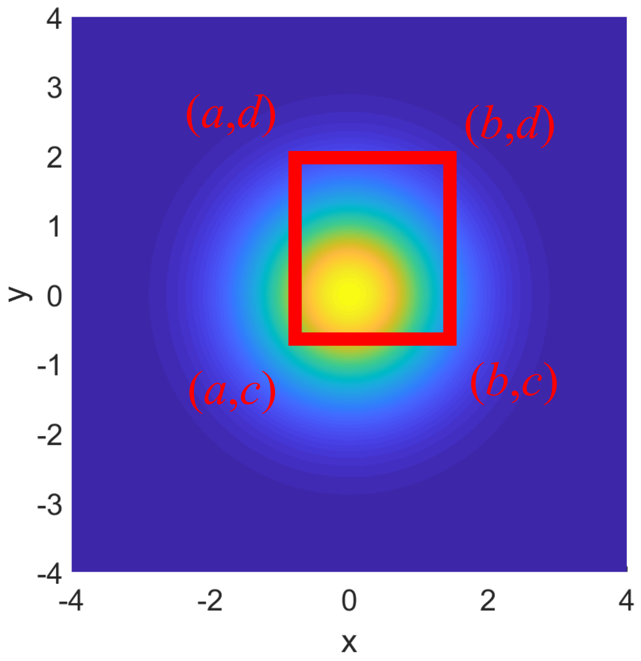
Multivariate PDF

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{\mathbf{x},\mathbf{y}}(x,y)\ dxdy = 1$$

$$P(a \le \mathbf{x} \le b, c \le \mathbf{y} \le d) = \int_a^b \int_c^d f_{\mathbf{x},\mathbf{y}}(x,y)\ dxdy$$

$$\mu_{\mathbf{x}} = \mathbb{E}\{\mathbf{x}\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x f_{\mathbf{x},\mathbf{y}}(x,y)\ dxdy$$





Covariance

$$\sigma_{\mathbf{x}}^2 = \mathbb{E}\{(\mathbf{x} - \mu_{\mathbf{x}})^2\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x - \mu_{\mathbf{x}})^2 f_{\mathbf{x},\mathbf{y}}(x, y)\ dxdy$$

$$\sigma_{\mathbf{y}}^2 = \mathbb{E}\{(\mathbf{y} - \mu_{\mathbf{y}})^2\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(y - \mu_{\mathbf{y}})^2 f_{\mathbf{x},\mathbf{y}}(x, y)\ dxdy$$

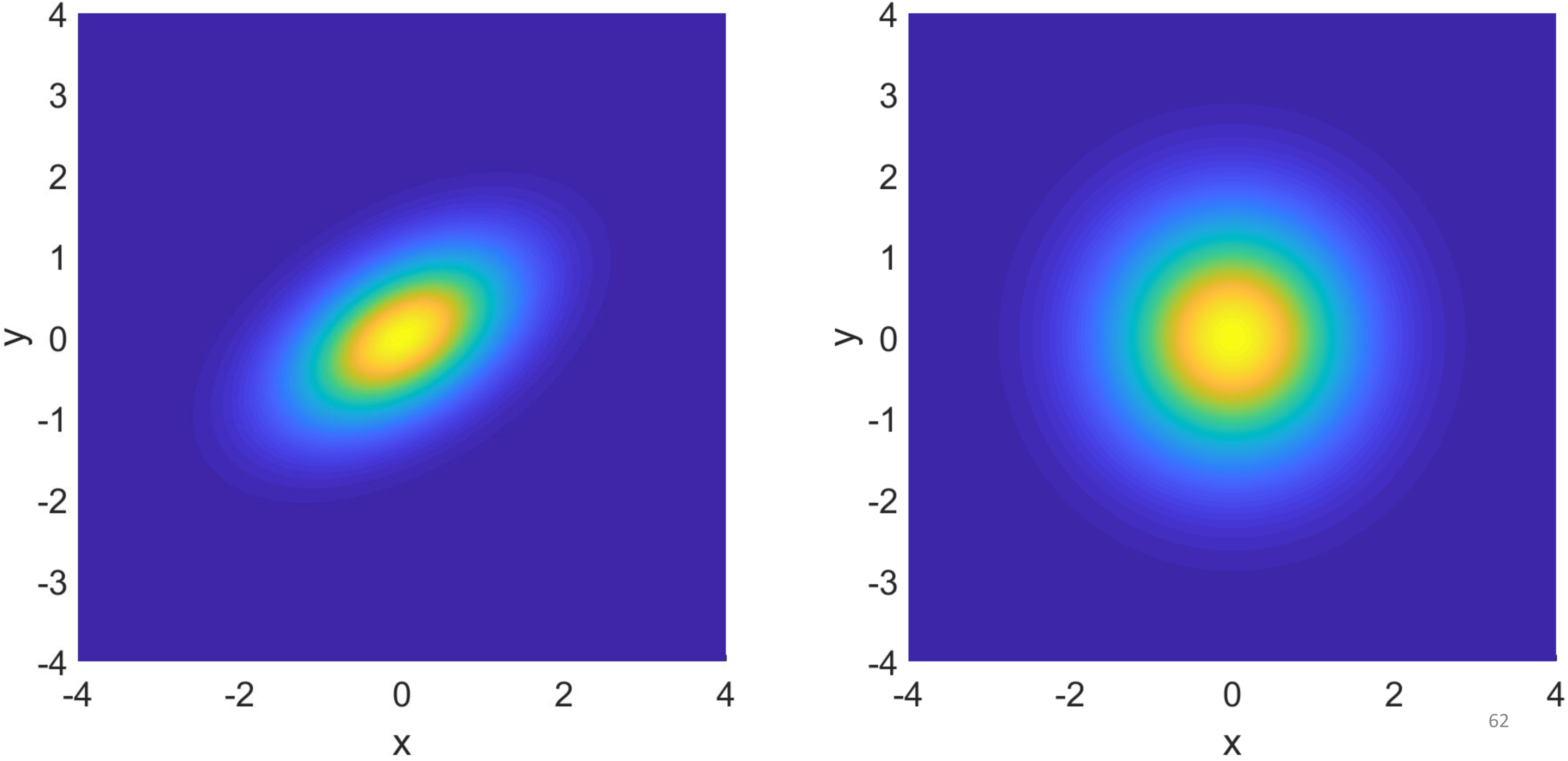
$$\sigma_{\mathbf{xy}} = \mathbb{E}\{(\mathbf{x} - \mu_{\mathbf{x}})(\mathbf{y} - \mu_{\mathbf{y}})\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x - \mu_{\mathbf{x}})(y - \mu_{\mathbf{y}}) f_{\mathbf{x},\mathbf{y}}(x, y)\ dxdy$$

Correlation

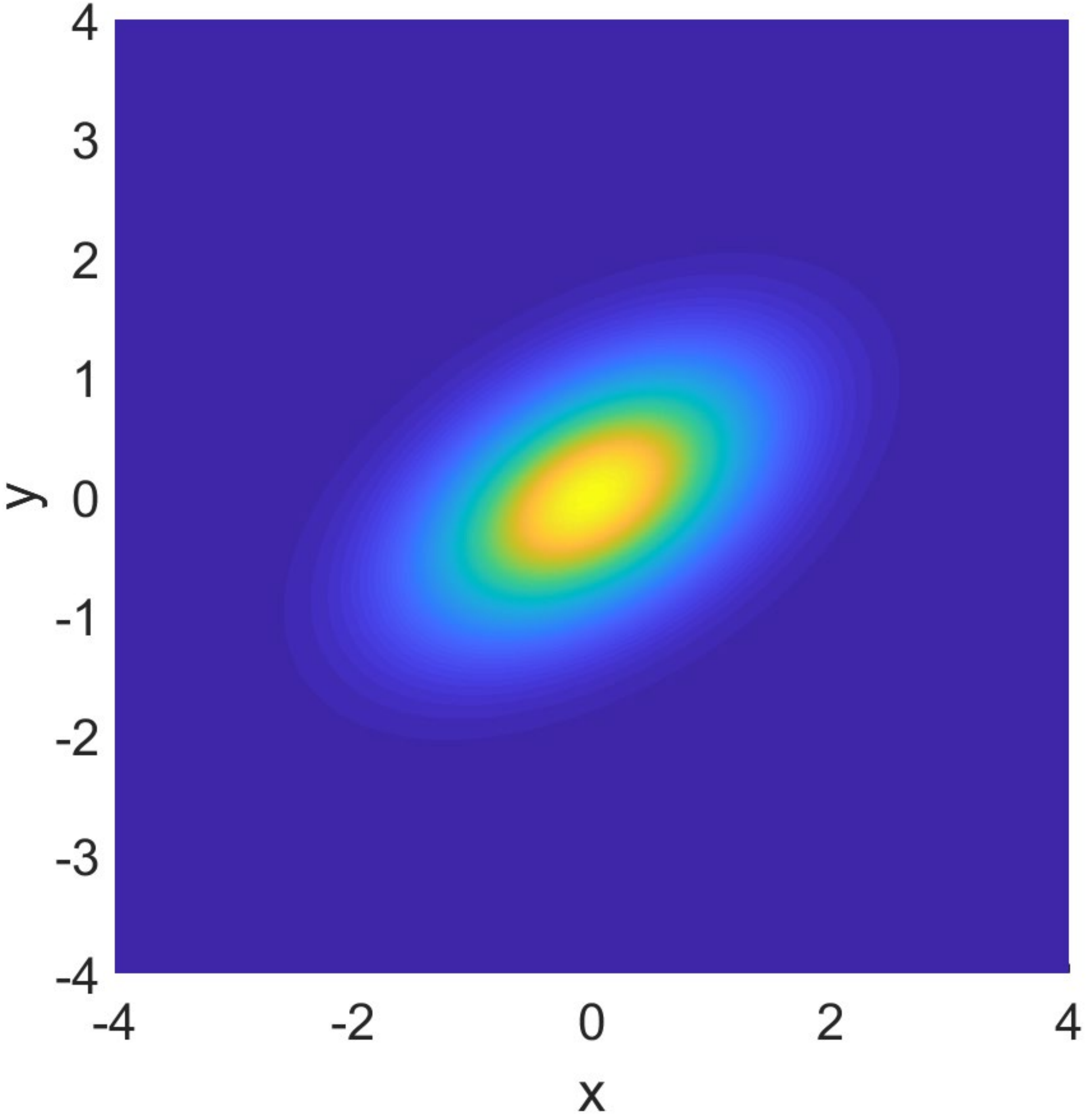
$$r_{\mathbf{xy}} = \frac{\mathbb{E}\{(\mathbf{x} - \mu_{\mathbf{x}})(\mathbf{y} - \mu_{\mathbf{y}})\}}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}}$$

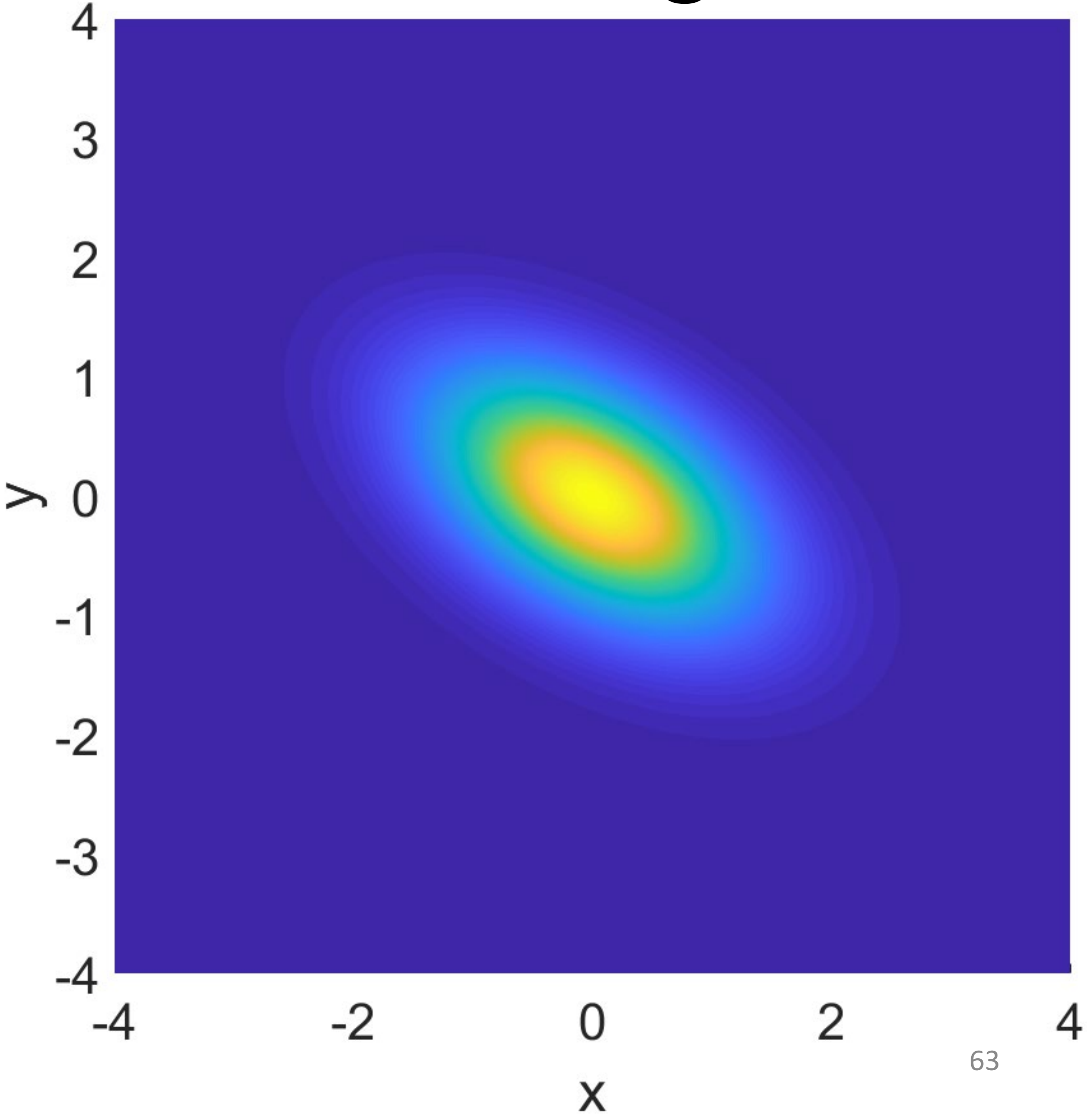
Normalized version of the covariance:
always between -1 and 1

Covariance and Correlation



Covariance and Correlation can be Negative!





Properties

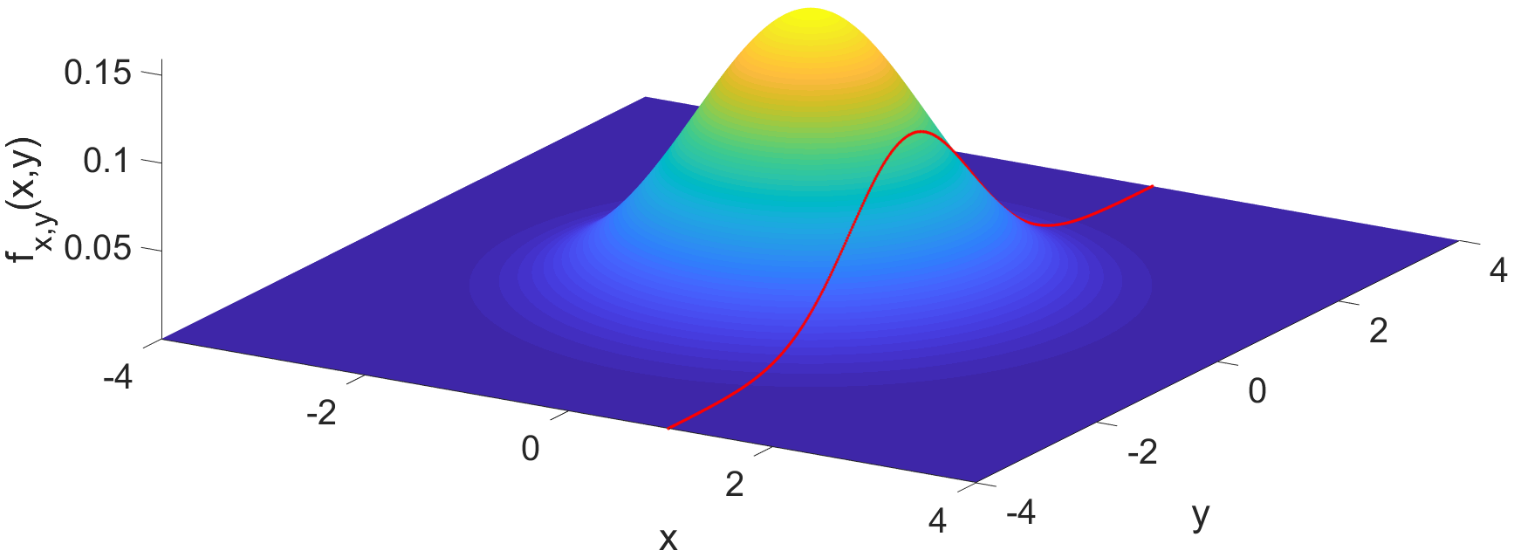
Random variables are called

- **independent** if $f_{\mathbf{x},\mathbf{y}}(x, y) = f_{\mathbf{x}}(x) \cdot f_{\mathbf{y}}(y)$
- **uncorrelated** if $\sigma_{\mathbf{xy}} = 0$ or equivalently $r_{\mathbf{xy}} = 0$

Conditional Distribution

PDF of $\mathbf{y}$, given that $\mathbf{x}$ has a particular value

$$f_{\mathbf{y}}(y|\mathbf{x}=x) = \frac{f_{\mathbf{x},\mathbf{y}}(x,y)}{f_{\mathbf{x}}(x)}$$



Take a slice and normalize

Still dependent on $\mathbf{x}$!

Marginal Distribution

$$f_{\mathbf{y}}(y) = \int_{-\infty}^{\infty} f_{\mathbf{x},\mathbf{y}}(x, y)dx = \int_{-\infty}^{\infty} f_{\mathbf{y}}(y|\mathbf{X} = x)f_{\mathbf{x}}(x)dx$$

Remove a variable from a joint distribution

Contents

Norm

Measures the 'size' of a vector

A norm is a non-negative function f mapping a (complex-valued) vector $\boldsymbol{x} \in \mathbb{C}^{n_x \times 1}$ to a scalar satisfying the following properties:

- $f(\boldsymbol{x}) = 0 \quad \Rightarrow \quad \boldsymbol{x} = 0$
- $f(\boldsymbol{x} + \boldsymbol{y}) \leq f(\boldsymbol{x}) + f(\boldsymbol{y})$ (triangle inequality)
- $\forall \alpha \in \mathbb{R},\; f(\alpha \boldsymbol{x}) = |\alpha| f(\boldsymbol{x})$

L^p Norm

$$||\boldsymbol{x}||_p = \left(\sum_i |x_i|^p \right)^{\frac{1}{p}} \quad \text{for } p \in \mathbb{R},\ p \geq 1$$

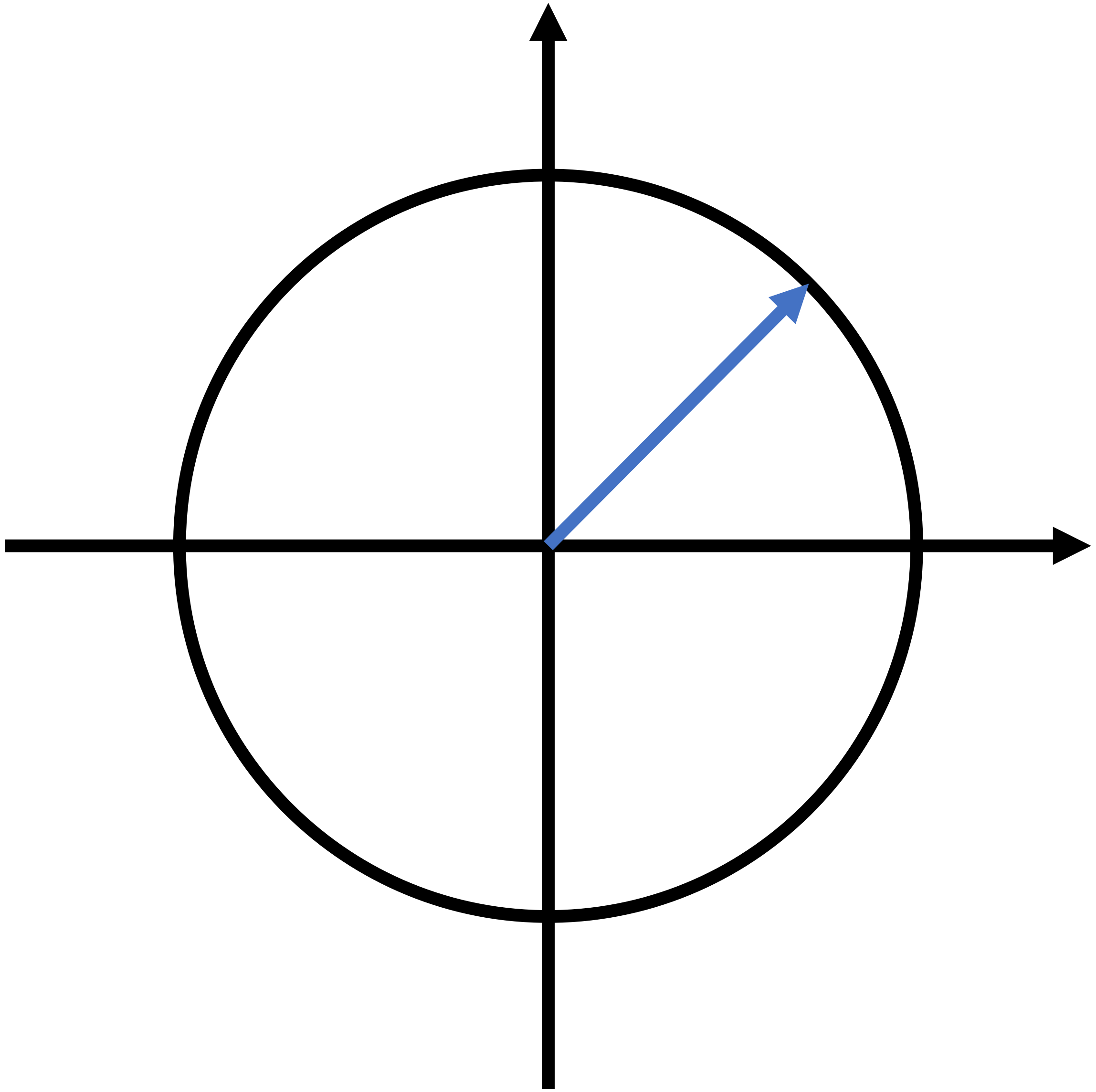
L^2 Norm

$$||\boldsymbol{x}||_2 = \sqrt{\left(\sum_i |x_i|^2\right)}$$

Most commonly used norm
also known as Euclidean norm

Simplified notation: $||\boldsymbol{x}||$

Short calculation: $||\boldsymbol{x}|| = \boldsymbol{x}^T\boldsymbol{x}$

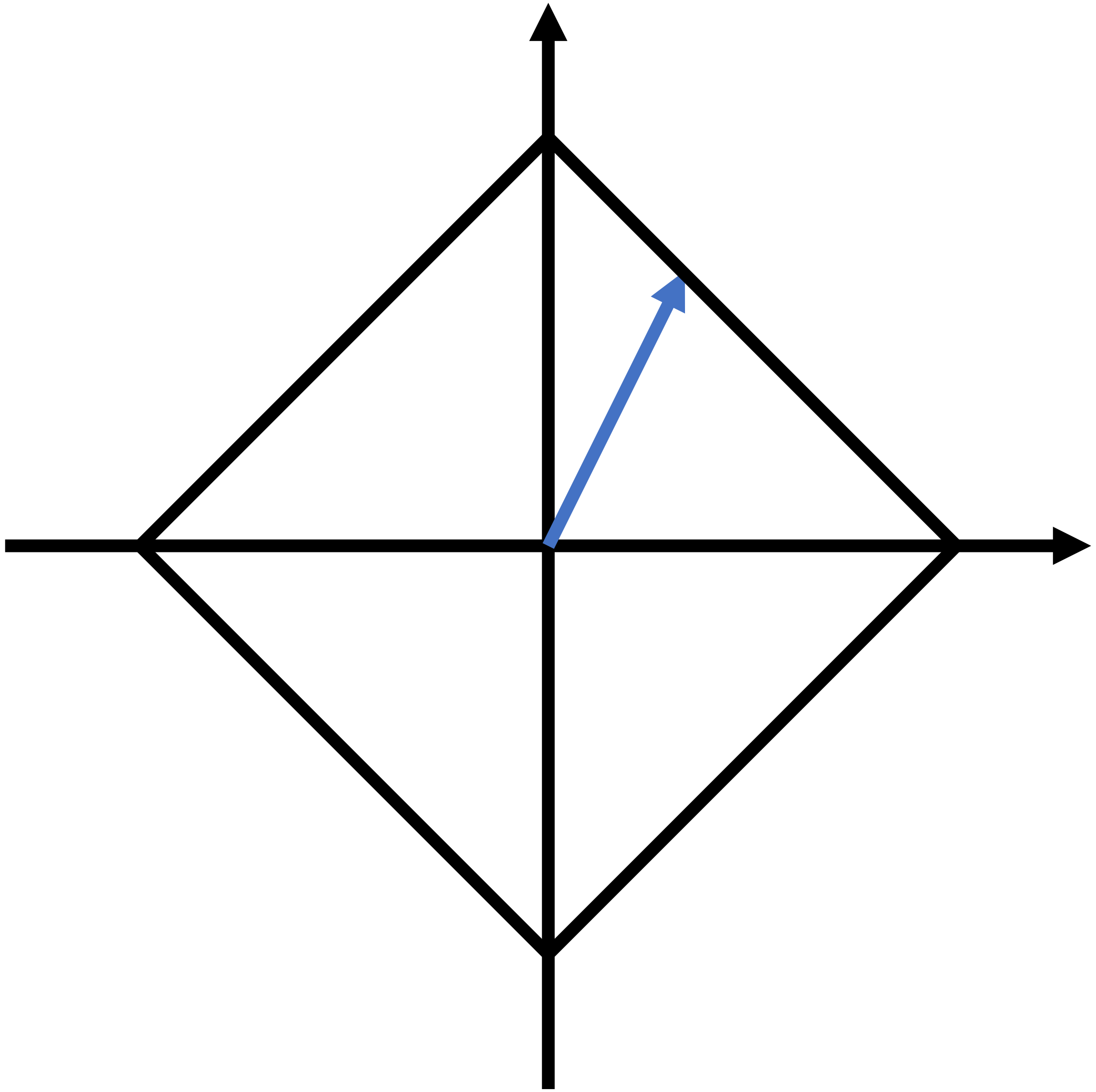


L^1 Norm

$$||\boldsymbol{x}||_1 = \sum_i |x_i|$$

Often used to induce sparsity

A substitute for counting the number of nonzero entries

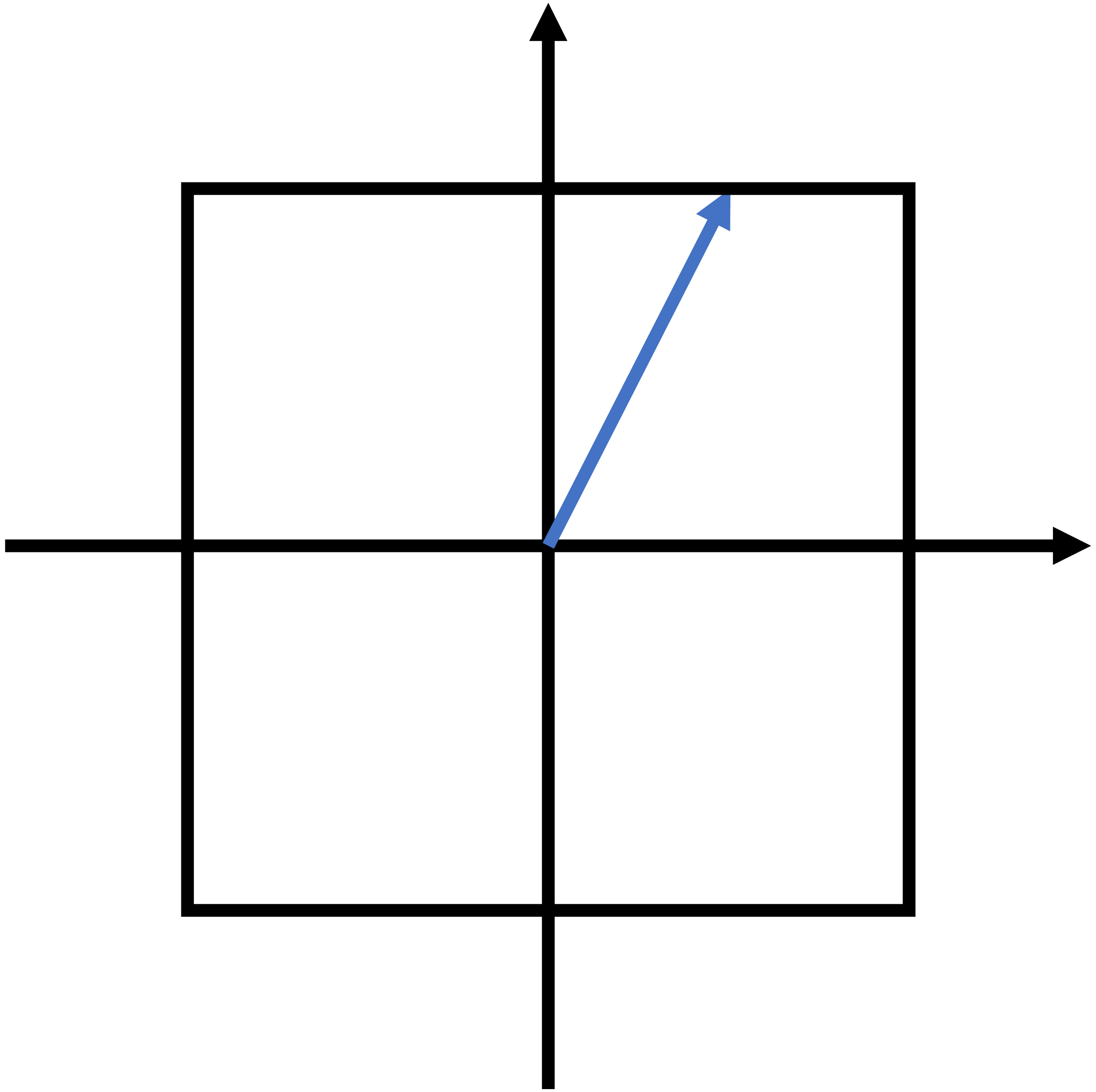


L^{∞} Norm

$$||\boldsymbol{x}||_{\infty} = \max_i |x_i|$$

Selects the largest value in the vector

Often used to assess worst case scenarios



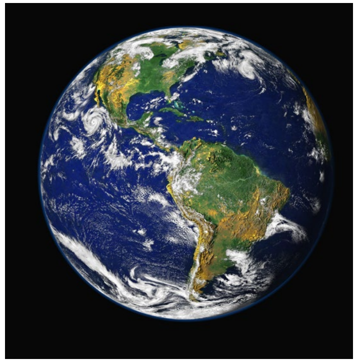
Contents

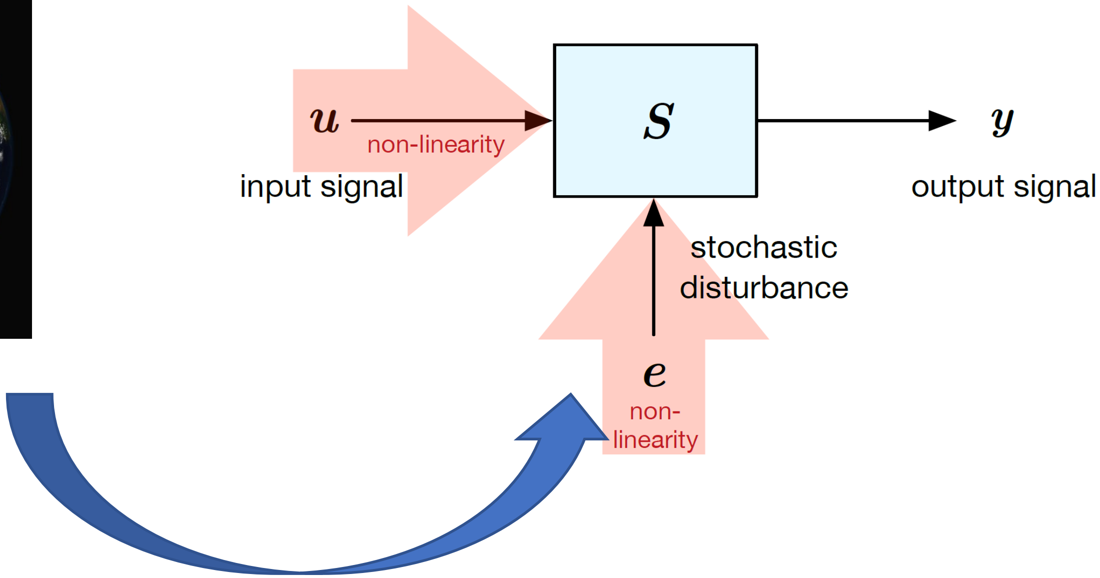
Estimation Criterion

The cost function describes the objective we want to minimize or maximize

Describes how well the model matches the data

Can be extended with extra penalty terms (regularization)





Cost Function
Objective Function
Loss Function

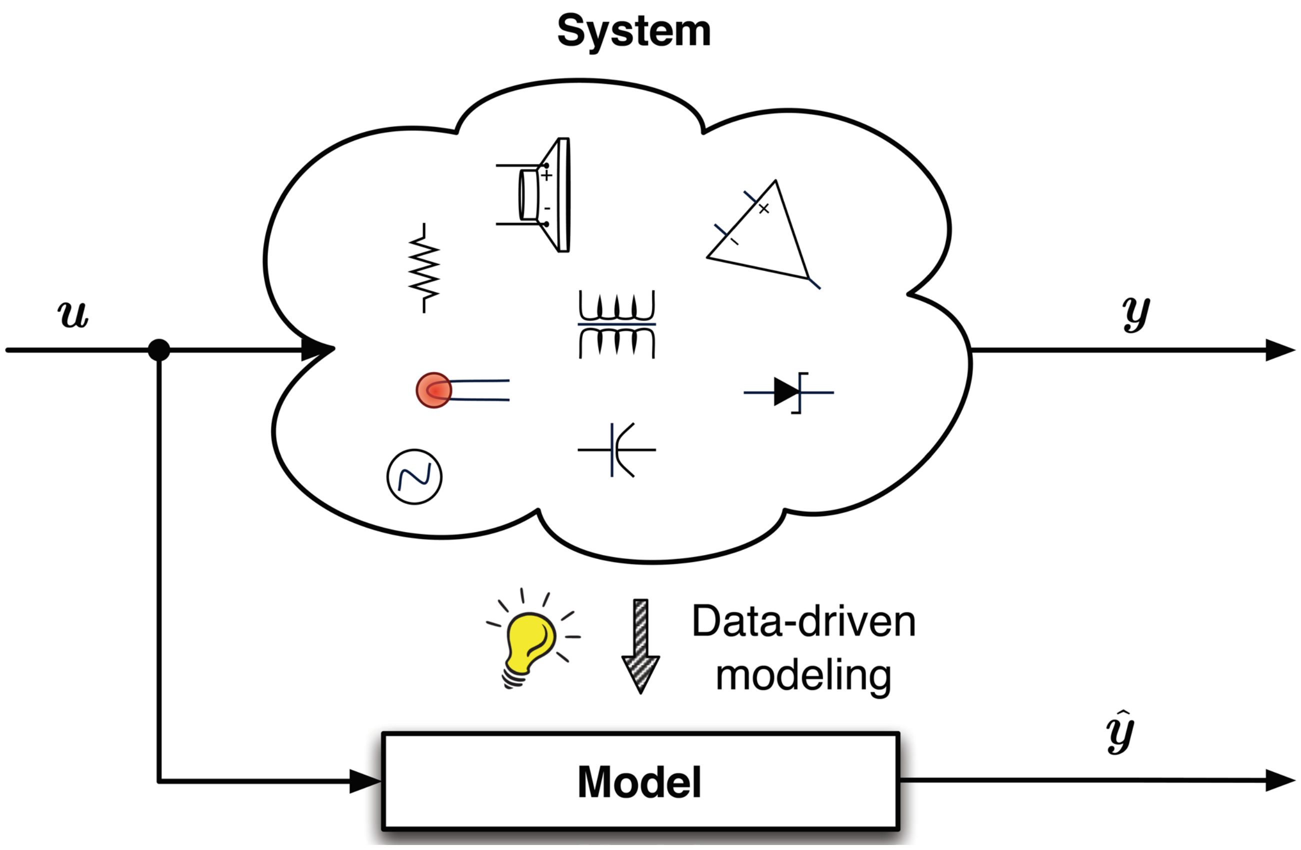
Typical Cost Function

Squared L^2 Norm

$$\frac{1}{N}\sum_{k=0}^{N-1}(y_k - \hat{y}_k)^2$$

normalize with
samples

simulation error
or
prediction error



Other cost functions:

Maximum Likelihood

Extensions:

Regularization Terms (e.g. L^1 or L^2 Norm of Parameter Vector)

1-Step Ahead Prediction

data up to timestep *k-1*

$$\underbrace{\hat{y}_k}_{\text{predicted output at timestep } k} = \mathbb{E}\left\{y_k \middle| \overbrace{\mathcal{D}_{k-1}}^{\text{data up to timestep } k\text{-}1}, \underbrace{u_k}_{\text{input at timestep } k}\right\}$$

predicted output
at timestep k

input at timestep k

Expectation taken over the stochastic disturbances

Simulation

$$\hat{y}_k = \mathbb{E}\left\{ y_k \middle| \{u_j\}_{j=1}^{k} \right\}$$

simulated output at timestep k

inputs from start up to timestep k

Expectation taken over the stochastic disturbances

Estimator

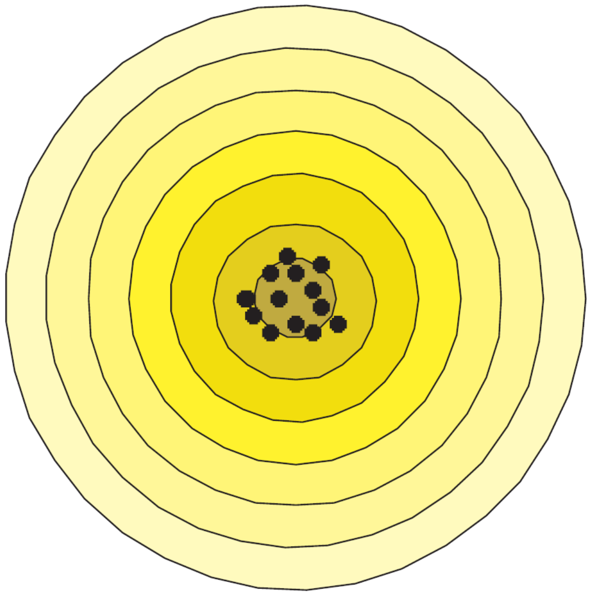
An **estimator** $\hat{\theta}_N$ of θ_o is a mapping from the (measured) data $\mathcal{D}_N$ to $\hat{\theta}_N$. If the data contains random variables, then $\hat{\theta}_N$ is a random variable.

Estimator

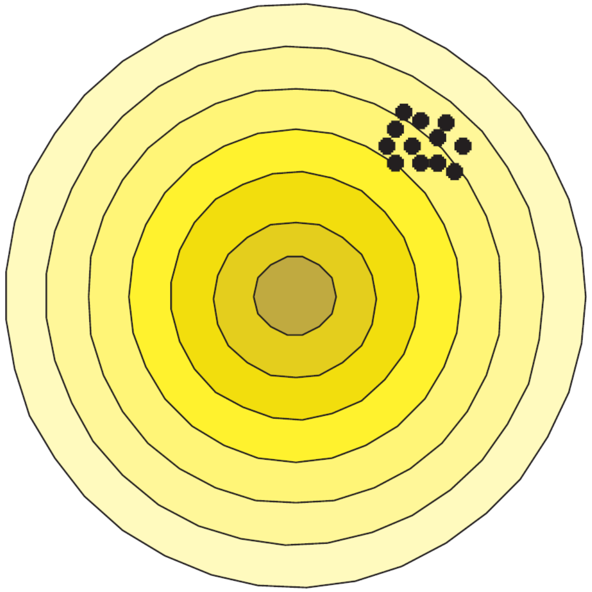
An **estimator** $\hat{\theta}_N$ of θ_o is a mapping from the (measured) data $\mathcal{D}_N$ to $\hat{\theta}_N$. If the data contains random variables, then $\hat{\theta}_N$ is a random variable.

- Unbiased: $\mathbb{E}\{\hat{\theta}_N\} = \theta_* = \theta_\text{o}$ bias $:= \theta_* - \theta_\text{o}$
- Variance: $P_\theta = \mathbb{E}\{(\hat{\theta}_N - \theta_*)(\hat{\theta}_N - \theta_*)^\top\}$
- Consistent: $\hat{\theta}_N$ is consistent if:
  - $P\{\lim_{N\to\infty} \hat{\theta}_N = \theta_\text{o}\} = 1$
  - $\hat{\theta}_N \to \theta_\text{o}$ with probability 1 for $N \to \infty$.

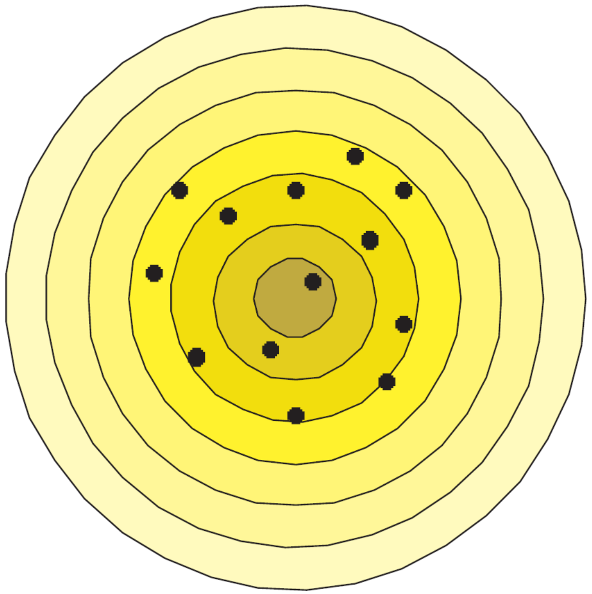
Estimator



Unbiased
Small variance



Biased
Small variance



Unbiased
Large variance

Consistency $=$ Asymptotically unbiased and variance tends to 0

Estimator

Unbiased
Small variance

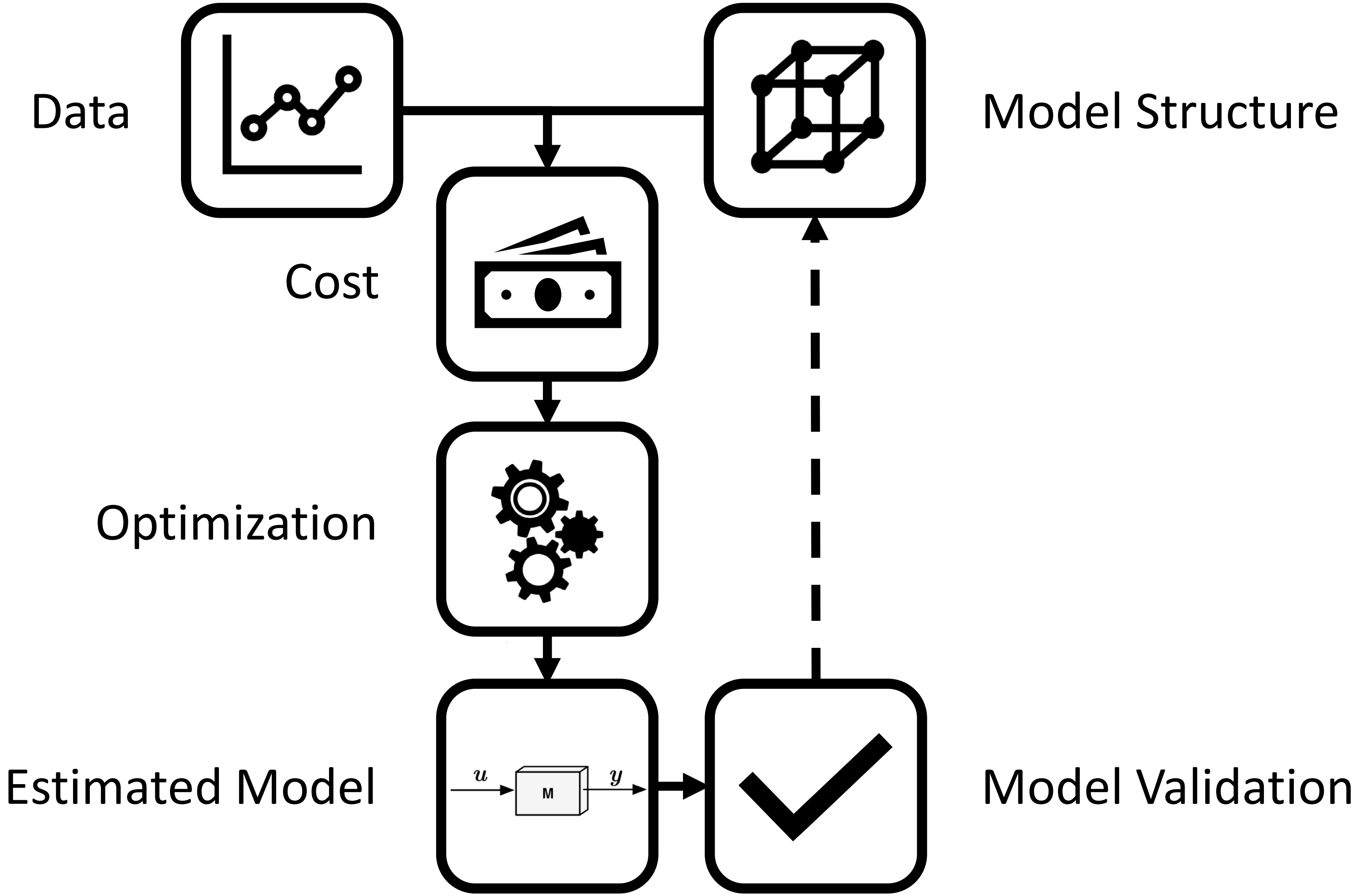
Biased
Small variance

Unbiased
Large variance

Consistency $=$ Asymptotically unbiased and variance tends to 0

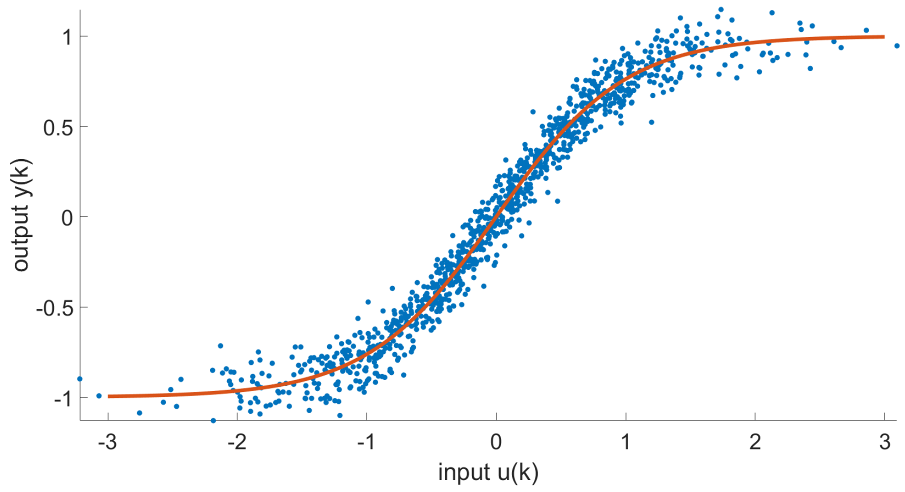
Contents

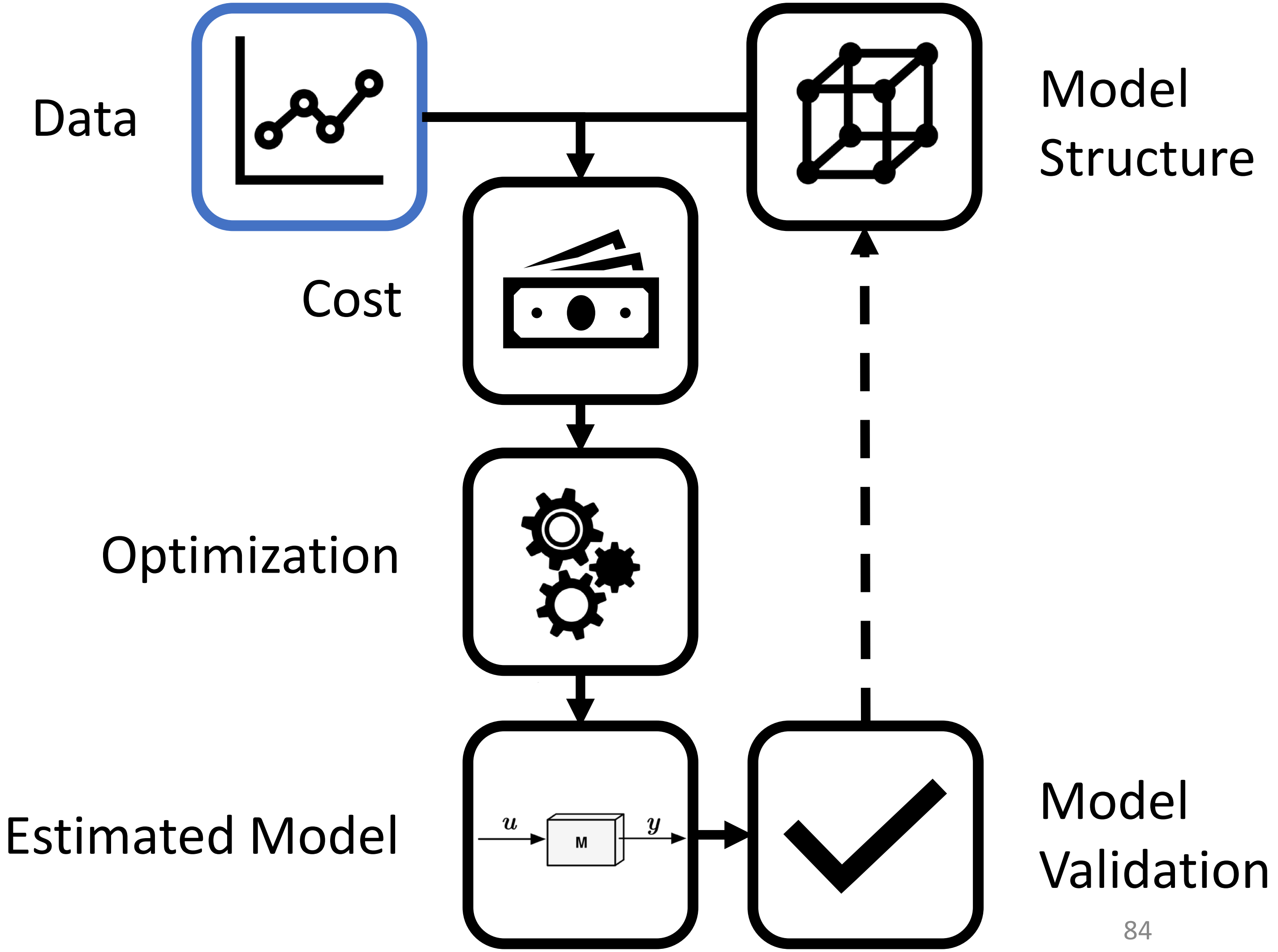
Data-Driven Modelling Process



Data-Driven Modelling Process: Data

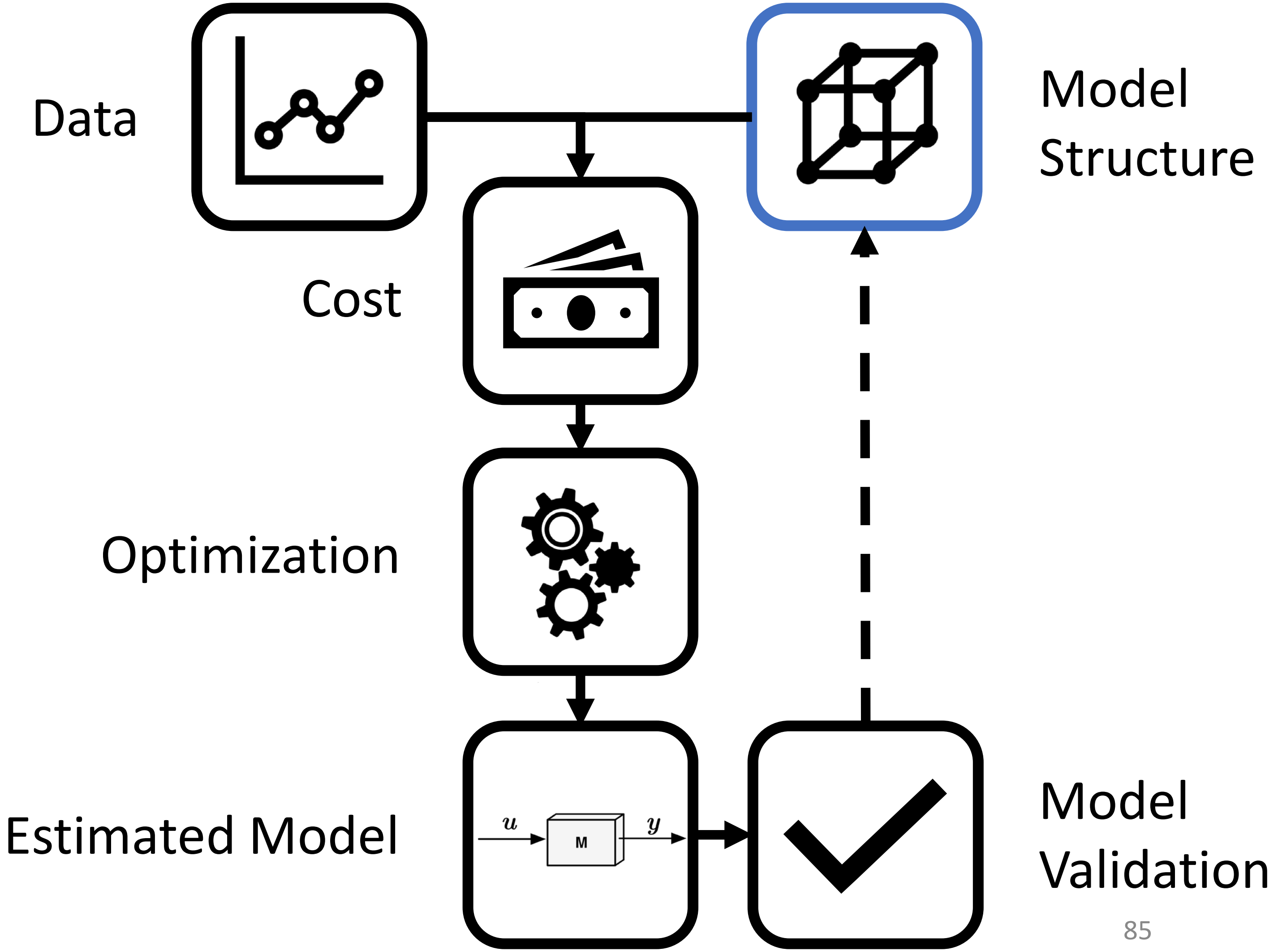
Measured data can be given to you or can be obtained after performing carefully designed experiments.





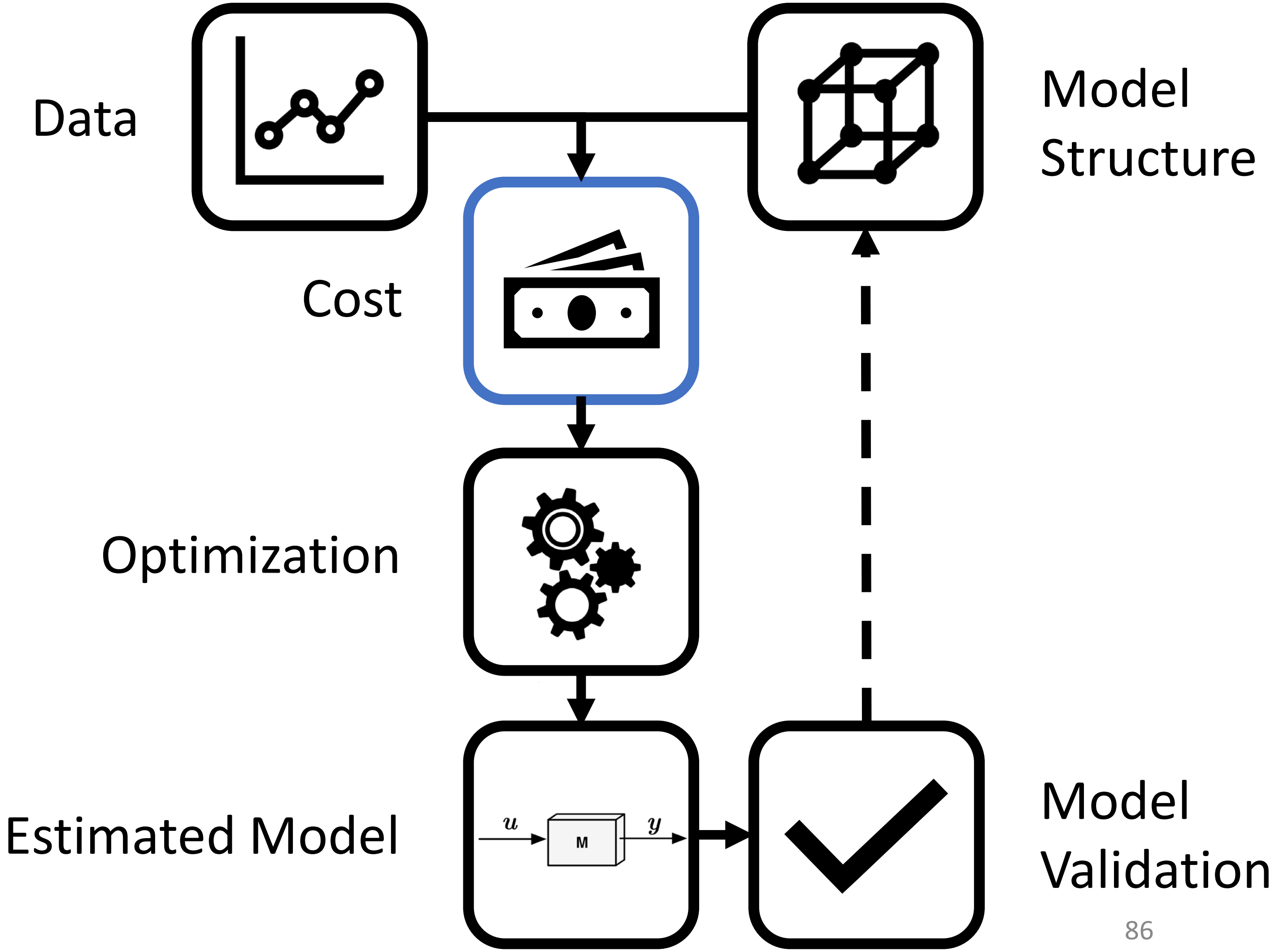
Data-Driven Modelling Process: Model Structure

Type of model you want to estimate with a chosen parametrization, also includes a noise model.



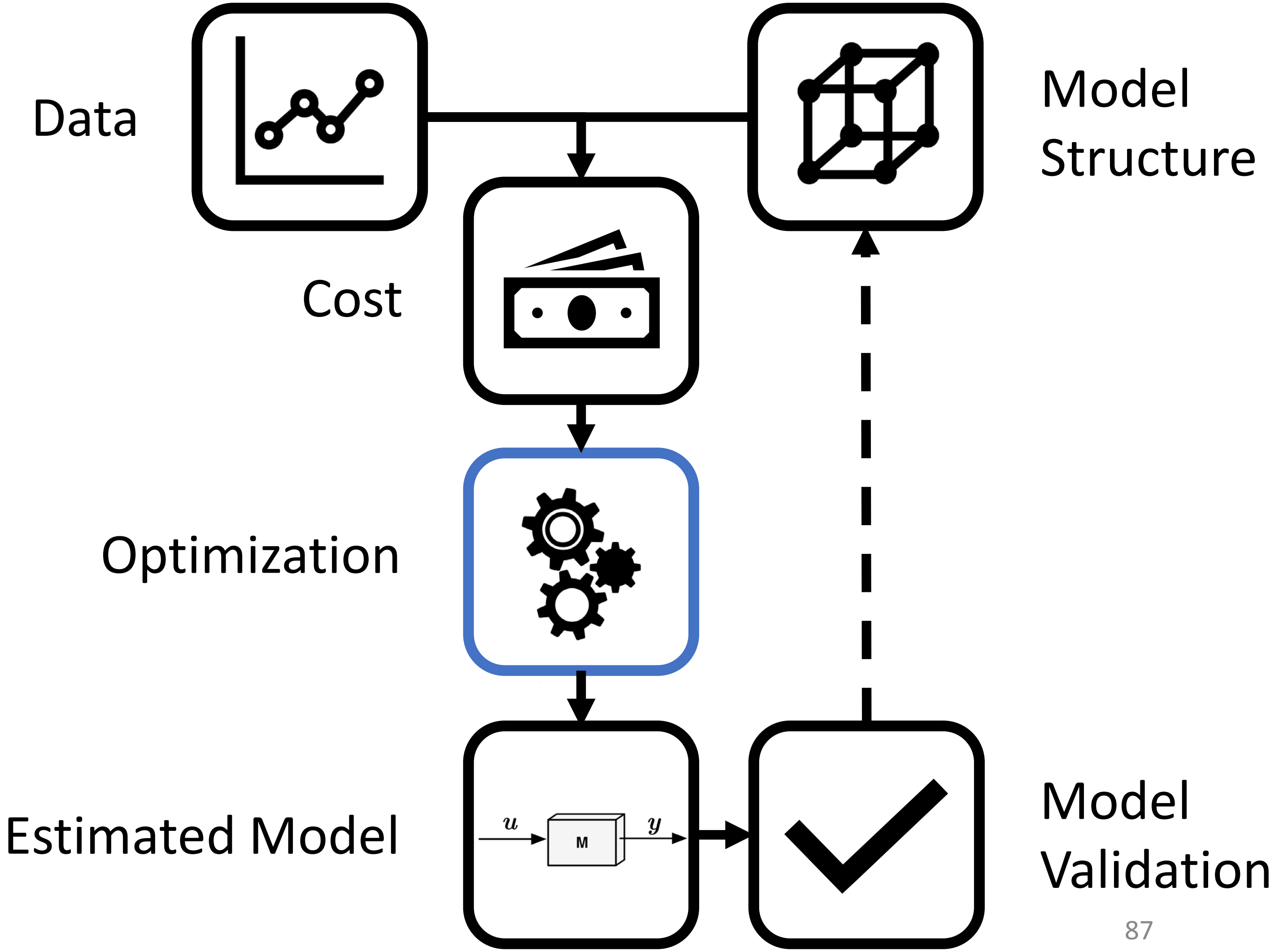
Data-Driven Modelling Process: Cost

Expresses how ‘far’ the model is from the data.



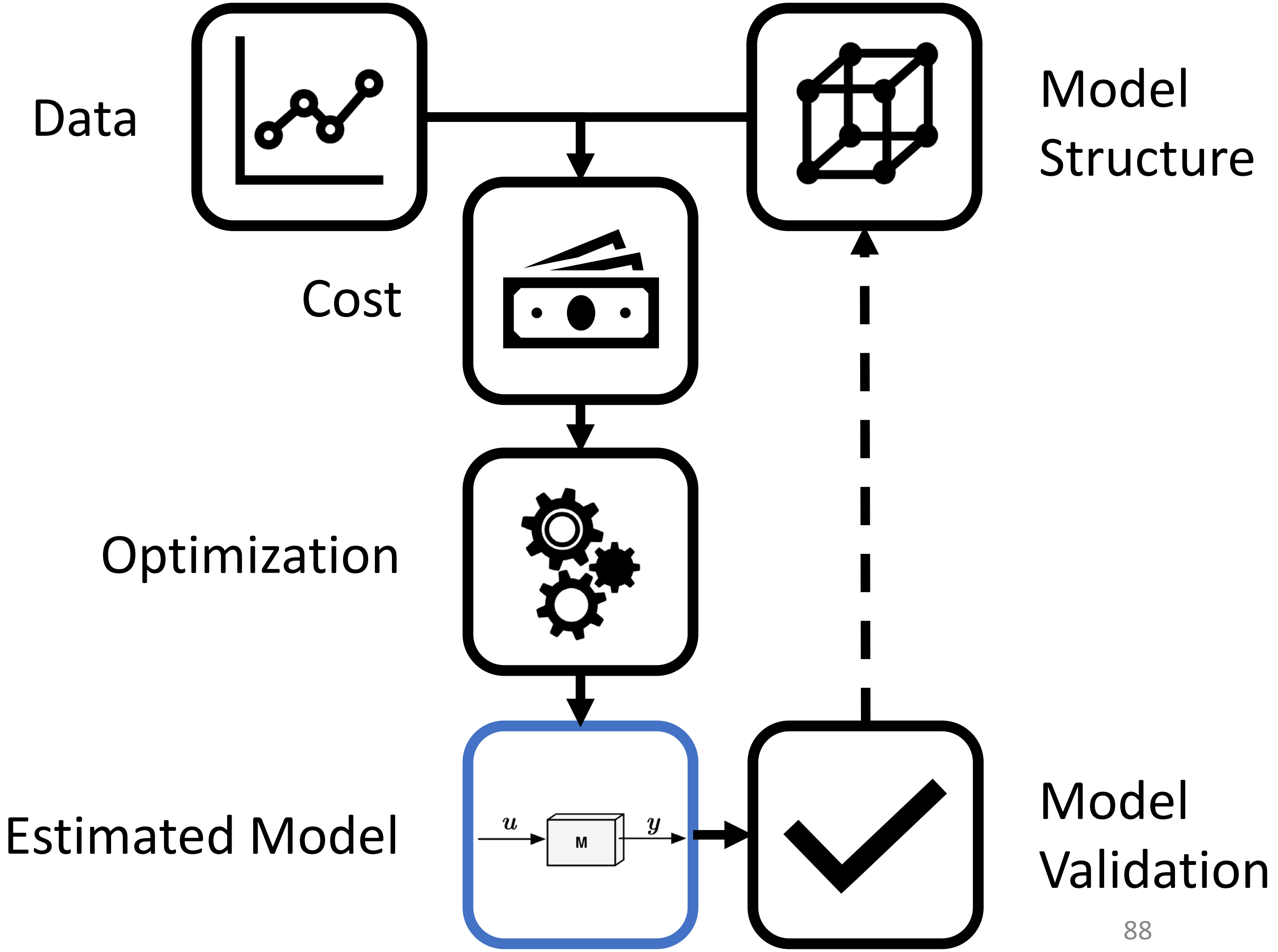
Data-Driven Modelling Process: Optimization

Minimizing the cost by modifying the model parameters results in the estimated model.



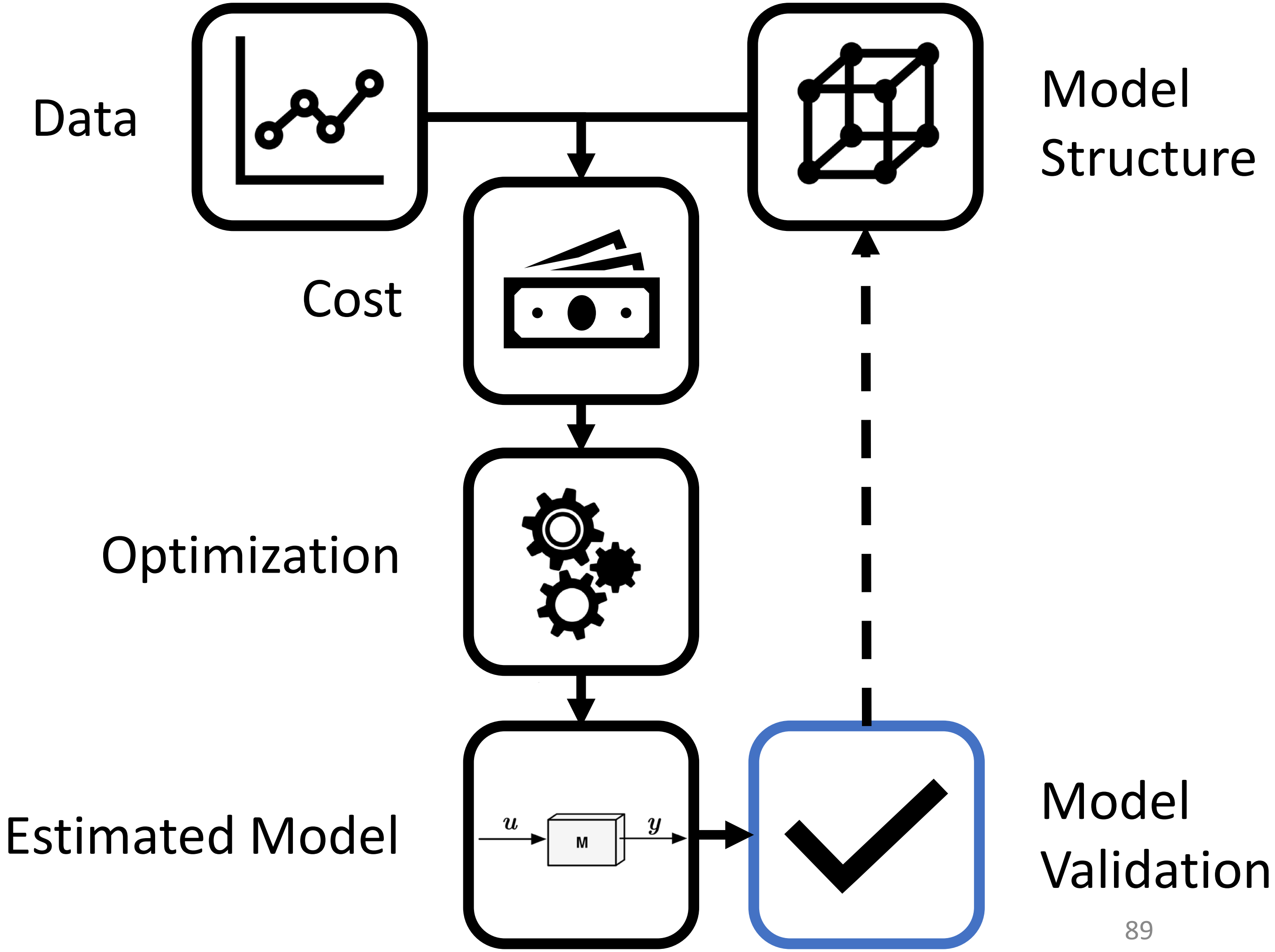
Data-Driven Modelling Process: Estimate

The estimated parameters combined with the previously chosen model structure result in the estimated model.

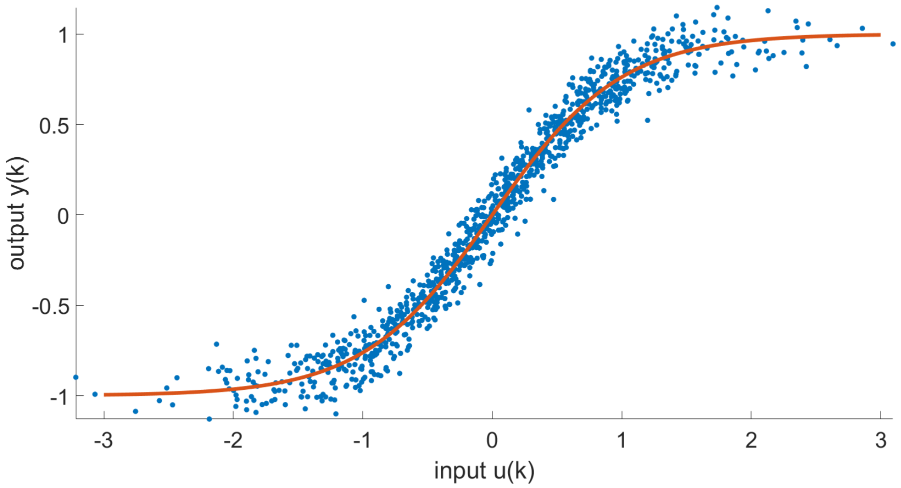


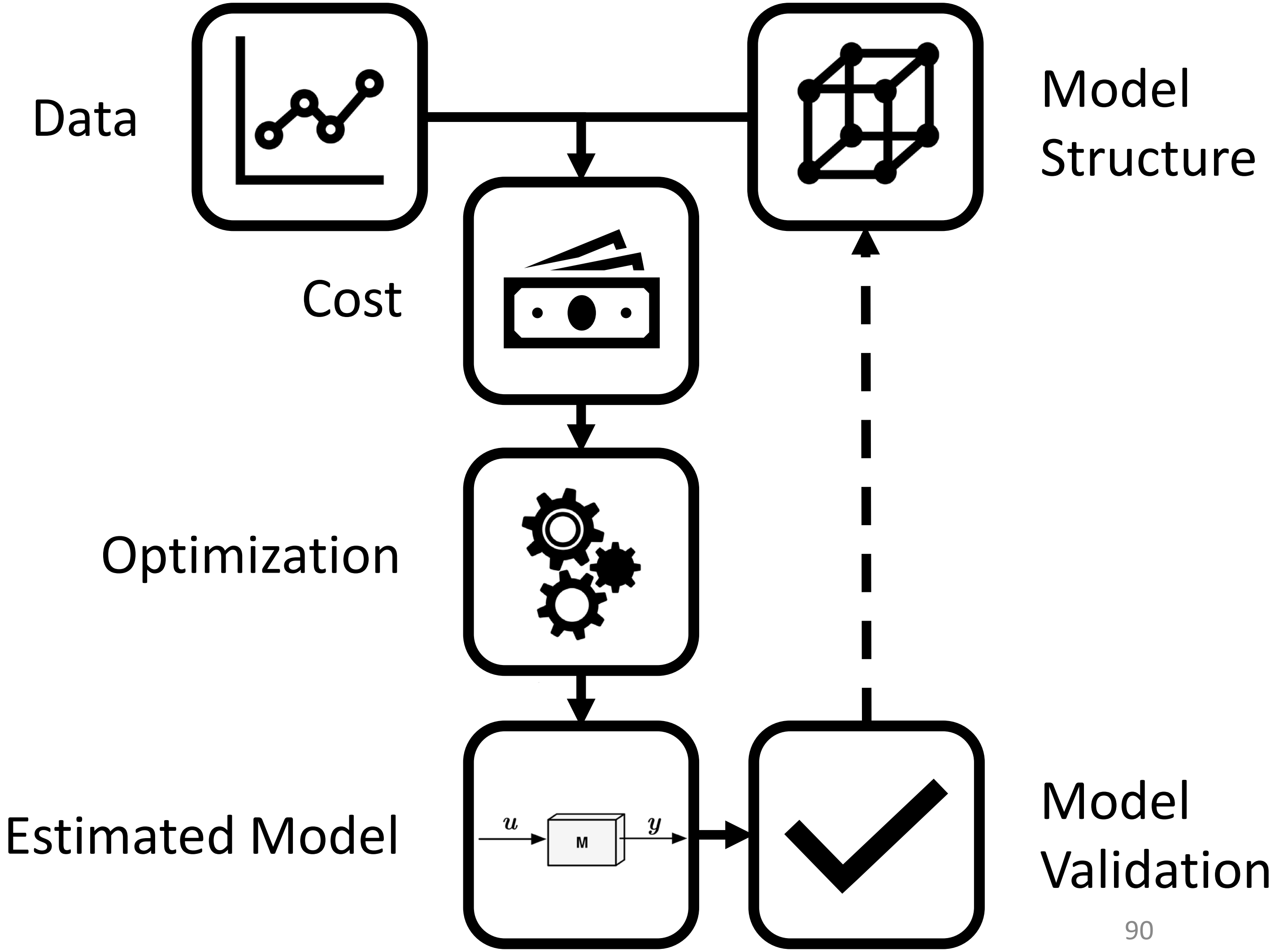
Data-Driven Modelling Process: Validation

Validate the model, e.g. on a validation dataset. Decide if the model is good, or if new data or a different model structure is needed.



Data-Driven Modelling Process: Example





How Can Machine Learning Contribute?

Model Completion (see motivating example)

Encoding Prior Knowledge

Function Representation

Automate the Data-Driven Modelling Process

Alternative Learning Methods

How Can Machine Learning Contribute?

Level 1: Function Estimation (supervised learning)

$$y_k = f\left(u_k, u_{k-1}, \ldots, u_{k-n_b}, y_k, \ldots, y_{k-n_a}\right) + e_k$$

Methods: GP, LS-SVM, IV-SVM, NN, …

Level 2: Estimate Dynamical Relation (self-supervised learning)

$$x_{k+1} = f\left(x_k, u_k\right) + w_k$$
$$y_k = h\left(x_k, u_k\right) + e_k$$

Methods: KCCA, KPCA, Deep NN, Bayesian Learning, MCMC, …

Level 3: Automated Structure-Free Modeling (self-supervised learning)

Level 4: Active-Interaction Based Learning (unsupervised learning)

Towards a “logic engine” to solve dynamic modeling problems

Contents

What did we not discuss?

How to represent the model (static nonlinearity)?

How to learn the model (minimize the cost)?

How can we estimate a policy instead of a model?

Machine Learning for Systems and Control

5SC28

Lecture 1B: Preliminaries

dr. ir. Maarten Schoukens & dr. ir. Roland Tóth

Control Systems Group

Department of Electrical Engineering

Eindhoven University of Technology

Academic Year: 2020-2021 (version 1.0)

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Version 13/04/2021