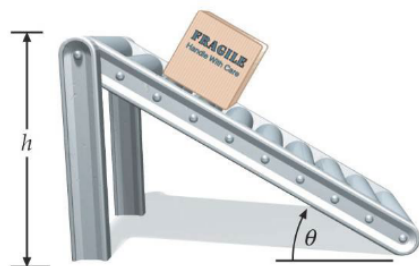


## SW 1: Introduction

### Model's three properties

- **Mapping:** models act as a representation of natural or artificial originals and can be models in turn;
- **Reduction:** models function as abstraction. They do not capture every attribute of the original; instead, they isolate and retain only those attributes relevant to the specific objective, intentionally omitting detail to manage complexity and focus on the problem at hand;
- **Pragmatic:** models function as utilitarian substitutes. They do not replace the original universally but serve as a representative for a specific user (subject), within a defined time frame, and for a particular purpose or operation.

### Example



- **Generaliz.:** point mass sliding down an inclined plane;
- **Mapping:** box as mass, conveyor slope as an angle  $\theta$ , vertical drop as height  $h$ , gravity;
- **Reduction:** no structure flexibility, no air movement, no friction, no rollers  $\rightarrow$  flat plane;
- **Pragmatic:** it allows  $a, v_f, t$  of the box to be calculated, it enables the prediction of how to build a belt mockup.

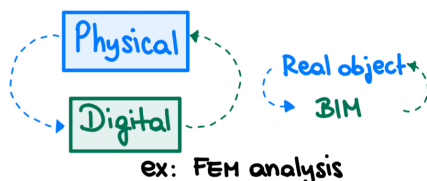
### Digital representation

----- Manual Data Flow (Offline)

----- Automatic Data Flow (Real-time)

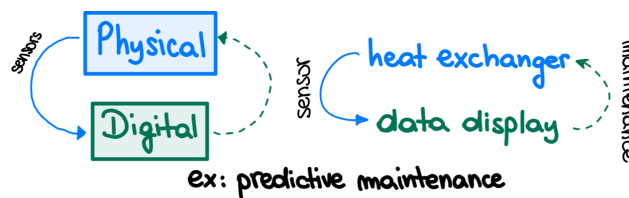
### Digital model (simulation)

No direct connection between digital and physical object:



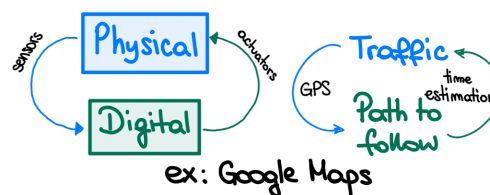
### Digital shadow

Unidirectional, automated data flow from physical object to digital model:



### Digital twin

Automated data exchange between physical object and model:



### Role of time

#### Stationary behavior

Steady-state operation:  $\dot{m}_\alpha = \dot{m}_\omega$

#### Dynamic behavior

Non stationary/transient/unsteady:  $\frac{dm}{dt} = \dot{m}_\alpha - \dot{m}_\omega$

### Governing dynamics

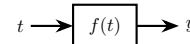
#### Empirical (black box)

Data based, without direct physics link. (ex: machine learning, fitting of functions)



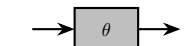
#### Physics-based (white box)

Based on physical laws. (ex: conservation of mass)



#### Grey-box (hybrid)

Combining physics and data parameters.



### Role of space

#### Point model (0D)

Assumes the whole system is perfectly mixed. (ex: ideal mixer with isotropic distribution). Software: Excel, MATLAB

### Linked point

Connects several simple models together to create a basic network or layout. (ex: space shown via linking of 0D-models). Software: Simulink, Modelica

### Spatial model (1-3D)

Considers real position of state variables or entities; spatial relationships affect the dynamics. (ex: real mixer with anisotropic, heterogeneous distribution). Software: COMSOL, ANSYS, AutoCAD, REVIT

### Example with a heat pump

- **Purpose:** digital shadow  $\rightarrow$  automated data;
- **Governing dynamics:** physics-based  $\rightarrow$  based on thermodyn. laws;
- **Time:** time dependent, dynamic behavior  $\rightarrow$  heating load, power of the hp, on/off cycles;
- **Space:** linked point  $\rightarrow$  el. inputs, thermal energy exchange, 4 components to monitor.

### Solvability of models

#### Analytical

Closed formula as solution. Only for simple problems.

$$A = \frac{x^3}{3} \Big|_0^2 = \frac{8}{3}$$

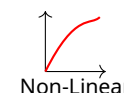
#### Numerical

Numerical approximation. For complex problems.

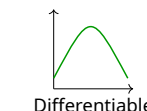
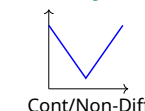
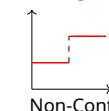
$$A \approx \sum_{i=1}^n f(x_i) dx \approx 2.6667$$

### Further modelling properties

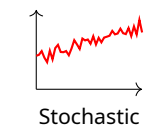
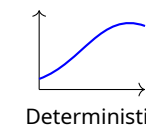
#### Linear vs Non-linear



#### Continuity vs Differentiability



#### Deterministic vs Stochastic



**Modelling approaches****Top-down**

Largest components broken down into smaller. ex: marble block sculpture, railway network.

⊕ Efficient model, ⊖ Misses details

**Bottom-up**

Individual components combined into larger. ex: LEGO model, human body.

⊕ Detailed model, ⊖ Complex

**SW2: How to model a system**

1. Problem formulation
2. Mathematical representation
3. Mathematical analysis
4. Interpretation and evaluation of results

**Problem formulation****Task 1 - Defining goals**

What do we want to achieve?

How well/closely does our model need to represent reality?

What could be the goals for this specific system?

**Task 2 - Characterize the system**

What are the relevant parameters and variables of the system?

What are the system boundaries?

What are the inputs and outputs of the system?

**Task 3 - Simplify and idealize the system**

Still reproduce the significant behaviors of the system, while reducing complexity.

Reduce model to the main parameters and variables (ex. for hp: COP? Max. power? Avg power? Yearly values? Temperature levels?).

**Mathematical formulation****Task 1 - Identify fundamental theories and laws**

If no laws are available, use ad-hoc or empirical data to derive relationships:

Thermodynamic laws, material properties, ad-hoc

**Task 2 - Derivation of relationships**

Transfer system into a mathematical formulation.

**Top-down (black/grey box):** Use generic relationship, data from measurement to determine parameters. For more complex systems, add more parameters. Use techniques such as machine learning.

**Bottom-up:** Detailed physical modelling of the device. Physical laws to describe each component. Exact geometry, material properties, boundary conditions.

**Task 3 - Reduce to standard mathematical problem**

Simple algebra, linear programming, differential equation, diffusion problem, wave propagation, FEM problem, using suitable methods and software/programming tools.

**Interpretation and evaluation of the results****Task 1 - Calibration of results**

Use existing data to calibrate the model.

**Task 2 - Validation**

Check underlying physics law, such as energy or mass conservation, compare to known solutions, look at extreme cases, compare to measured data.

→ What is it and why do we have to do it?

**Before the modelling:**

What do we model how?:

- a) Aims: does the model describe the process under test?
- b) Output: does the model provide the required output to describe the process?
- c) Type: is the type of the model suitable to describe the process?

**During modelling:**

Can we reproduce the measurements?

Does the model behave like to system under study?

- d) Fitting data: does the model reproduce the fitting data? How to measure accuracy?
- e) Reproducing novel data: does the model also predict novel measurement data correctly?
- f) Sensitivity analysis: does the model predict the behavior of the system correctly when system parameters are changed?

**After modelling:**

Does the model also work with new data?

- g) System potentially changed.
- h) Differences in system behavior is only manifest in new experiments.

**SW 3: Data-based modelling****Linear regression**

Used to find a linear function  $y = f(x) = a + bx$  that best fits a dataset  $(x_i, y_i)$ .

**Least squares method**

Minimize the sum of squared errors (SSE):

$$S = \sum_{i=1} (y_i - (a + bx_i))^2$$

If measurement uncertainties  $\Delta y_i$  exist, weight the error:

$$S_i = \left( \frac{y_i - y(x)}{\Delta y_i} \right)^2$$

**Optimal parameter formulas**

Finding  $a$  and  $b$  when  $S$  is minimal:

$$\frac{\partial S}{\partial a} = 0 \quad ; \quad \frac{\partial S}{\partial b} = 0$$

Slope  $b$ :

$$b = \frac{\sum_i x_i y_i - \frac{1}{n} (\sum_i x_i) (\sum_i y_i)}{\sum_i x_i^2 - \frac{1}{n} (\sum_i x_i)^2}$$

Intercept  $a$ :

$$a = \bar{y} - b\bar{x}$$

where:

$$\bar{x} = \frac{\sum_i x_i}{n} \quad ; \quad \bar{y} = \frac{\sum_i y_i}{n}$$

**Quality of fit ( $R^2$ )**

The coefficient of determination  $R^2$  indicates the percentage of variation explained by the model:

$$R^2 = \frac{\sum_i (y(x) - \bar{y})^2}{\sum_i (y_i - \bar{y})^2}$$

- $R^2 = 1$  (100%): the model explains all data;
- $R^2 = 0$  (0%): the model doesn't (random).

**Multilinear regression**

Used when the target depends on multiple variables:

$$y(x_1, \dots, x_n) = a + b_1 x_1 + \dots + b_n x_n = a + \sum_{j=1}^n b_j x_j$$

**Non-linear regression**

The goal is to fit data using non-linear functions when the underlying process is not linear.

**Linearization techniques**

Function	Equation	Trasformation	Variables
Exp	$y = ae^{bx}$	$\ln y = \ln a + bx$	$x$ vs $\ln y$
Power	$y = ab^x$	$\ln y = \ln a + x \ln b$	$x$ vs $\ln y$
Inverse	$y = \frac{a}{x}$	$\frac{1}{y} = \frac{x}{a}$	$x$ vs $\frac{1}{y}$
Square offset	$y = ax^2 + b$	$y = a(x^2) + b$	$x^2$ vs $y$
Root / Cubic	$y = \sqrt{ax^3 + b}$	$y^2 = ax^3 + b$	$x^3$ vs $y^2$

**Maximum likelihood method (MLE)**

Determines the parameters of a probability distribution that best describes a dataset, independent of histogram binning.

**Likelihood function**

Defines as the product of probability densities for all data points:

$$L(\sigma, \mu) = \prod_i f(x_i, \sigma, \mu)$$

**Log-likelihood**

To simplify calculation and avoid small numbers, minimize the negative logarithm:

$$-\log L = -\sum_i \log(f(x_i, \sigma, \mu))$$

**Common distribution****Normal distribution:**

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$

**Weibull distribution (Reliability):**

$$f(x) = \begin{cases} \lambda k (\lambda x)^{k-1} e^{-(\lambda x)^k}, & x > 0 \\ 0 & \text{else} \end{cases}$$

**Weibull cumulative distribution function**

$$F(x) = \int_{-\infty}^x f(u) du = \begin{cases} 1 - e^{-(\lambda x)^k} & \text{for } x > 0 \\ 0 & \text{else} \end{cases}$$

**SW4: Modelling with ODEs****Fundamentals of ODEs**

An ODE contains functions of one independent variable and their derivatives.

**Ordinary (ODE)**

Involves one independent variable:

$$\frac{d^2x}{dt^2} = -g$$

**Partial (PDE)**

Involves multiple independent variables:

$$\frac{d^2u}{dt^2} = c^2 \frac{d^2u}{dx^2}$$

**Analytical solution method****Separation of variables**

Used when terms involving  $y$  and  $x$  can be moved to opposite sides.

**Variation of parameters**

Used for inhomogeneous linear ODEs. General solution is the sum of the homogeneous solution and a particular solution.

**Numerical solution methods****Euler method**

A simple iterative method to approximate ODEs defined as  $\frac{df}{dx} = g(x)$ .

The approximation uses the finite difference slope:

$$\frac{df}{dx} \approx \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

Iterative steps:

$$f(x_0 + \Delta x) = f(x_0) + g(x_0)\Delta x$$

**Modelling principles****Balance equations**

Based on the conservation principle:

$$\frac{d}{dt}f(t) = f(t_\alpha) - f(t_\omega)$$

**Example in a capacitor**

$$U_0 = U_R + U_C \Rightarrow U_0 = RI + \frac{Q}{C} = R\frac{dQ}{dt} + \frac{Q}{C}$$

**Mechanics and forces**

Equation of motion is derived from Newton's second law  $F_{net} = ma$ .

**Example of a falling drop with drag**

$$m\dot{v} = mg - bv \Rightarrow v(t) = \frac{mg}{b} \left(1 - e^{-bt/m}\right)$$

**Growth and decay**

Describes processes where a quantity increases or decreases over time.

$$\frac{dN}{dt} = kN \Rightarrow N(t) = N_0 e^{kt}$$

with half-time / doubling factor  $\tau$ :

$$\tau = \left| \frac{\ln 2}{k} \right|$$

**Example of logistic growth**

$$\frac{dN}{dt} = KN(t) - \frac{K}{K} N^2 \Rightarrow N(t) = \frac{L}{1 + \left(\frac{L}{N_0} - 1\right) e^{-kt}}$$

**Recipe to derive the equation of motion**

1. Make a sketch of the situation;
2. Define the coordinate system and select variables of interest;
3. Identify all forces and momenta;
4. Formulate the equation of motion;
5. Solve it.

**Linear algebra and systems of ODEs****Matrix representation**

System of equations:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

Matrix form ( $Ax = b$ ):

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

If  $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ , then  $\dot{x} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix}$

**Inversion and diagonalization****Inverse matrix**  $R^{-1}$ :  $R \cdot R^{-1} = I$  (Identity matrix).

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

**Diagonalization:** Special matrices can be rewritten as:

$$A = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

This transforms the matrix into a diagonal matrix containing eigenvalues  $\lambda$ .**Why is it called linear algebra****Linearization:**Complex, non-linear functions can be approximated by linear functions in a small neighborhood of a point  $a$ :

$$f(x) \approx f(a) + f'(a)(x - a)$$

**Benefit of solving ODEs**If  $A$  were a number,  $\dot{x} = Ax$  would solve to  $x(t) = ke^{At}$ . Since  $A$  is a matrix, if we diagonalize it using eigenvalues  $\lambda$ , the solution becomes a mixture of exponentials:

$$x(t) = R^{-1} \begin{pmatrix} k_1 e^{\lambda_1 t} & 0 & 0 \\ 0 & k_2 e^{\lambda_2 t} & 0 \\ 0 & 0 & k_3 e^{\lambda_3 t} \end{pmatrix} R$$

**Solvability of linear systems****Geometric interpretation:**Solving  $Ax = b$  is finding the intersection of lines/planes.

- Case 1**, consistent: lines intersect at exactly one point;
- Case 2**, inconsistent: lines are parallel and distinct, there is no solution;
- Case 3**, infinite solutions: lines are identical and overlap completely.

**Determinant**A scalar value derived from a square matrix that tells us if it is invertible. If  $\det A = 0$ , the matrix is not invertible.**2x2 formula:** For  $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ ,  $\det A = ad - bc$ .**3x3 formula:** For  $A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$ ,

$$\det A = a_{11} \det \begin{bmatrix} e & f \\ h & i \end{bmatrix} - a_{12} \det \begin{bmatrix} d & f \\ g & i \end{bmatrix} + a_{13} \det \begin{bmatrix} d & e \\ g & h \end{bmatrix}$$

$$\det A = \sum_j^n a_{1j} C_{1j}, \quad \underbrace{C_{1j} = (-1)^{1+j} \det A_{ij}}_{\text{Cofactors}}$$

**The Eigenvalue problem**For a square  $n \times n$  matrix  $A$ , we look for a Eigenvector  $x$  and a Eigenvalues  $a$  such that:

$$Ax = \lambda x$$

**Calculation method:**

1. Solve the characteristic equation  $\det(A - \lambda I) = 0$
2. This result in an  $n$ -th order polynomial ( $a_1 \lambda^n + \dots = 0$ )
3. The roots of this polynomial are the Eigenvalues.

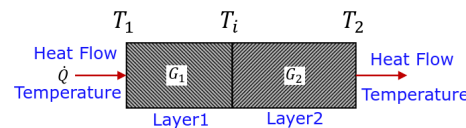
**SW5-10: Modelica****Equation-based modelling**

Describes a system by using physical relationships.

**Problem definition - Double layer wall**A wall consists of two layers with different thermal conductance values  $G_1$  and  $G_2$ .

We consider two steady-state cases:

1. A heat flow  $\dot{Q}_1$  passes through the wall and the right temperature is  $T_2$ . The interface temperature  $T_i$  and the left temperature  $T_1$  are unknown.
2. Both boundary temperatures  $T_1$  and  $T_2$  are given and the interface temperature  $T_i$  and the heat flow  $\dot{Q}$  are unknown.

**Formulas**

Heat conduction equation [W]:

$$\dot{Q} = G \Delta T = G (T_\alpha - T_\omega) = G_1 (T_1 - T_i) = G_2 (T_i - T_2)$$

Thermal conductance [W/K]:

$$G = \frac{\lambda}{L}$$

Conservation of energy:

$$\dot{Q}_1 = \dot{Q}_2 = \dot{Q}$$

**Component-based modelling**

Instead of rewriting equations each time, an instance of the needed physics law component is added.

**Thermal components****thermalConductor**Models heat linear heat flow between two ports determined by a constant thermal conductance  $G$ 

$$\dot{Q} = G (T_a - T_b) \quad ; \quad \dot{Q} = \frac{\lambda \cdot A}{L}$$

**fixedHeatFlow**

A source that injects a constant heat flow into the connected component



$$port.\dot{Q} = -\dot{Q}_{\text{component}}$$

**fixedTemperature**

Defines a constant temperature boundary condition (acting like an infinite heat reservoir).



$$port.T = T_{\text{parameter}}$$

**heatCapacitor**Thermal mass that stores energy, where temperature changes based on heat flow and heat capacity  $C$ .

$$C \cdot \frac{dT}{dt} = m \cdot c_p \cdot \frac{dT}{dt} = \dot{Q}$$

**convection**Models the heat transfer between a solid surface and a moving fluid based on a convection coefficient  $G_{\text{conv}}$ .

$$\dot{Q} = \alpha \cdot A \cdot \Delta T = G_{\text{conv}} \cdot (T_{\text{solid}} - T_{\text{fluid}})$$

**temperatureSensor**

Measures the absolute temperature at the thermal port and outputs that value as a real signal.



$$y = T_{\text{port}} \quad ; \quad \dot{Q} = 0$$

**Electrical components****resistor**

Resists the flow of electric current, creating a voltage drop proportional to the current.



$$U = R \cdot I \quad ; \quad \dot{Q} = P = U \cdot I$$

**constantVoltage**

An ideal voltage source that maintains a constant voltage difference between its positive and negative pins.



$$u_{\text{port}} = U_{\text{const}}$$

**ground**

Defines the reference potential (zero voltage) for an electric circuit.



$$u_{\text{port}} = 0$$



**Signal components****pulse**

Generates a signal that alternates between two values (amplitude and offset) with a defined period and pulse width.



$$y = \begin{cases} \text{offset} + \text{ampl.}, & \text{if } \in \text{pulse width} \\ \text{offset}, & \text{otherwise} \end{cases}$$

**constant**

A signal source that outputs a fixed numerical value.



$$y = k$$

**gain**

A signal block that multiplies the input signal  $u$  by a constant parameter  $k$  to produce the output signal  $y$ .



$$y = ku$$

**onOffController**

A logical controller that switches its output between true and false based on comparing a measured signal  $u$  to a reference value.



$$y = \begin{cases} \text{true} & \text{if } u < (\text{reference} - \frac{\text{bandwidth}}{2}) \\ \text{false} & \text{if } u > (\text{reference} - \frac{\text{bandwidth}}{2}) \end{cases}$$

**booleanToReal**

Converts a Boolean signal into a Real float number.



$$y = \begin{cases} \text{realTrue} & \text{if input is True} \\ \text{realFalse} & \text{if input is False} \end{cases}$$

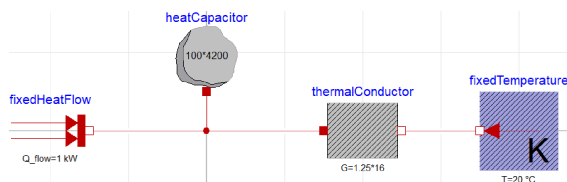
**Dynamic systems**

Two things can lead to time-varying behavior:

1. Transient boundary conditions
2. A dynamic system starting from a non-eq. state

**First-order thermal model**

A mass is heated by a constant source while simultaneously losing heat to a cooler environment

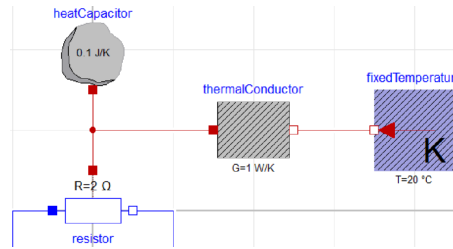


Conservation of energy at the central node:

$$C \cdot \frac{dT}{dt} = Q_{in} - G(T - T_{sink})$$

**Multi-domain modelling****Multi-domain model**

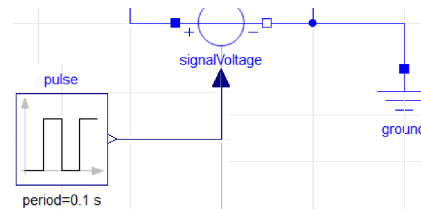
Allows representing different physical domains such as electrical, mechanical, thermodynamic, and fluid dynamics in a single model.



Resistor heat interacts with the thermal system

**Cyber-physical model**

A model combining physical domains with a software.

**One-dimensional model**

Simulation technique used to calculate spatial distribution by discretizing a continuous object into multiple discrete, lumped segments.

**About Modelica****Definition and structure**

Open source, equation-based, non-casual language for modelling dynamic behavior of multidisciplinary systems. Component-based (graphical connection), object oriented (inheritance), and hierarchical.

**Equation-based / non-casual modelling**

- Component diagram: topological (physical) structure;
- Equation-based: no fixed input/output direction;
- Connections: represent physical wiring/piping;
- Pros: reusable, multi-domain, closer to physics.

**Casual modelling**

- Block diagram: represents computational data flow;
- Assignment-based: fixed input/output;
- Connections: represent signal flow variables;
- Cons: prone to errors when modifying structure.

**Hierarchical structure**

Components are built from connected subcomponents and/or equations, allowing complex systems to be broken down into reusable parts.

**Object-oriented**

Allows creating general base definitions (superclasses) that specific components extend, rather than defining every component from scratch.

**Physical mapping**

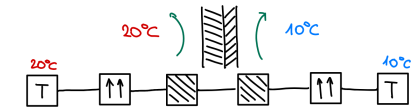
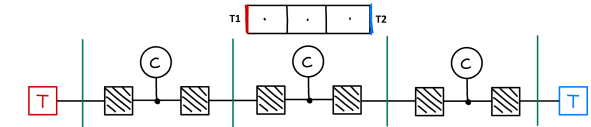
Icons represent physical components, connections represent actual physical couplings.

**Application examples**

- Multiphase flow: refrigeration systems;
- Multi-domain: Pneumatic piston pump;
- Compressible media: Medical pulse wave analysis.

**Examples wrap-up****Thermal circuit**

$$\frac{dE}{dT} = \frac{dU}{dT} = m \cdot c \cdot \frac{dT}{dt} = \dot{Q}$$

**Heat flow****Heaten up rod****Physical units**

Heat flow	$\dot{Q}$	[W]	Heat capacity	$C$	[J/K]
Thermal conductivity	$\lambda$	[W/mK]	Thermal conductance	$G$	[W/K]
Specific heat capacity	$c_p$	[J/kgK]	Convection coefficient	$\alpha$	[W/m²K]

**SW11: Model and control energy systems****Energy system model**

Model dynamics of energy systems to study consumption peaks, self-sufficiency and storage sizing, planning.

**How to build an energy system model****Single building**

Component list + properties; hydraulic connections; usage profiles; pricing information.

**Quarter/community/city**

Single-building info for ALL buildings + grid/connection properties + control scheme (balance demand/supply).

## Main challenges in energy system modelling

### Acquiring

Layout, parameters, installed-component details, control-system details, shading/clouds, usage profiles.

### Modelling

Reliable component behavior and coupling, environment interaction (solar/shading/clouds, wind shielding, contributions from neighboring buildings).

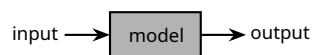
### Models availability

#### Software frameworks

Built-in, pre-made models for specific component types.

#### Simplified models

Basic blackbox representation:



### Literature

Models taken from published research and publications.

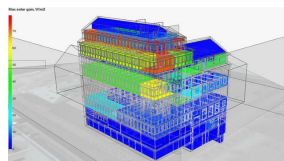
#### Data-science models

Models learned or fitted from data.

### White-/Grey-/Black-box models

- **White box:** theory-based, parameters from measurement or ab initio calculations;
- **Grey-box:** simplified theory, parameters estimated from data;
- **Black-box:** fit a model and parameters (data-based).

#### White-box



Each white-box model is a digital twin, but not all digital twins are white-box models.

#### Digital twin

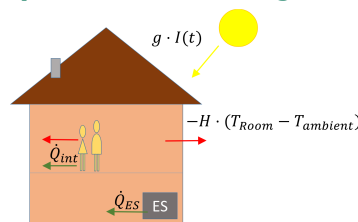
Virtual representation that serves as the real-time digital counterpart of a physical object or process:

- **Detailed:** detailed, ab initio / measured parameters;
- **Simplified:** key mechanisms, simple assumptions;
- **Numerical:** fitting data, parameters are fitted.

#### Heat pump modelling example (white to gray box)

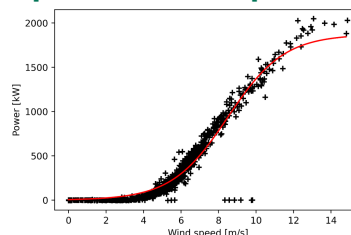
Evaporator and condenser cycles, refrigerant dynamics, expansion valve control algorithm. COP dependence, interpolate experimental data with non-linear regression.

### Grey-box example (1st order building / RC idea)



$$C \frac{dT_{\text{room}}}{dt} = \dot{Q}_{ES} + \dot{Q}_{\text{int}} + g I(t) - H (T_{\text{room}} - T_{\text{ambient}})$$

### Black-box example (wind turbine power curve)



$$P(w) = \frac{a}{1 + \exp(-b \cdot (w - s))}$$

### Control schemes overview

Control events = CE;  
Control decisions = CD;  
Control signal = CS.

- **Rule-based:** CE are chosen based on a rule-set;
- **Model-based:** predict the future behavior based on a model and decide such that the model performs best;
- **Neuronal network-based:** trains a neuronal network to take CD and act accordingly;
- **A posteriori optimisation-based:** takes historic data and optimise CD.

#### Rule-based

##### Fixed time plan controllers

CE are chosen based on a fixed time schedule.

##### Model predictive controllers

CE are chosen based on model prediction of system dynamics.

##### Set point controllers

CE is triggered if the target quantity is leaving the set point corridor. Ex:



$$\begin{cases} \text{ON} & \text{if } T_{\text{room}} < 19.5^\circ\text{C} \\ \text{OFF} & \text{if } T_{\text{room}} > 20.5^\circ\text{C} \end{cases}$$

### Feedback based controllers – P & I & D

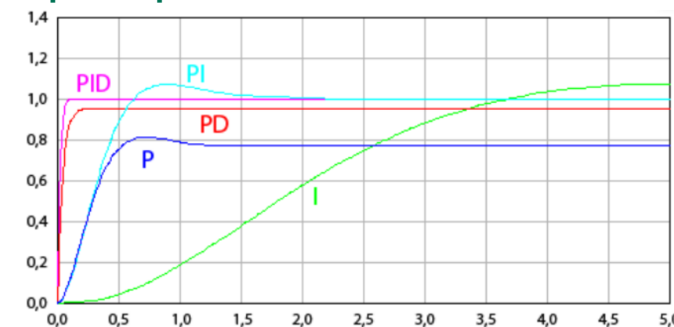
CE are initiated based on the interaction with the controlled quantity. We have P, PI, PID controllers.

A PID-type controller computes the actuator command  $u(t)$  from the error  $e(t) = r(t) - y(t)$

$$u(t) = \underbrace{K_P e(t)}_P + \underbrace{K_I \int_0^t e(\tau) d\tau}_I + \underbrace{K_D \frac{d}{dt} e(t)}_D$$

- **P:** reacts immediately to error:  
Bigger  $K_P \rightarrow$  faster response, but can overshoot / oscillate and typically leaves a steady-state error;
- **Integral:** accumulates error over time:  
Removes steady-state error (drives output to the set-point), but makes the response slower and can increase overshoot and “windup” if not handled.
- **Derivative:** reacts to how fast the error is changing:  
reduces overshoot and improves settling, but is sensitive to measurement noise.

### Graphical representation



The plot shows the closed-loop step response over time of the same plant controlled with different controllers with setpoint at 1.

- **P:** rises quickly but settles below 1  $\rightarrow$  steady-state error remains;
- **I:** rises very slowly but eventually reaches (and may slightly exceed) 1  $\rightarrow$  zero steady-state error, poor speed;
- **PD:** faster than P and with less overshoot / better damping, but still can have steady-state error;
- **PI:** reaches 1 but shows overshoot and slower settling;
- **PID:** fast rise, small overshoot, good settling, and zero steady-state error.

## SW12: Design and analysis of experiments

Design of experiments (DoE): systematics sweep of parameter space when no fully-analytical optimum is available. Build an empirical model  $y = F(x)$  from sampled runs and use it for optimisation.

### Setup of simulation study

**Aim:** what do you want to achieve?

- control optimisation for given energy
- retrofitting of existing energy system by adding or replacing technologies
- design of novel energy system

**Quantification:** choose KPI / cost function.

- total energy consumption
- cost for energy provision
- share/amount of renewable energy

**Decision variables:** which variable can be influenced.

- Ex: size, orientation, capacity, ...

**Measure of success:** optimal configuration with respect to the KPI.

- Ex: among all system configuration we find the solution with highest share of renewable energy

### Coarse workflow

1. Calculate total / monthly energy demand;
2. Calculate production for different sizes of the system;
3. Calculate share of renewable energy;
4. Pick optimum solution

**Challenges:** energy demand has to be fulfilled in each time point unless large storage system is considered. Storage capacity of battery often heavily overestimated.

### Detailed workflow

1. Setup system simulation;
2. Select different system sizes;
3. Run simulation measuring total and renewable energy consumption and calculate share of renewables;
4. Visualise the results;
5. Pick optimum solution.

**Challenges:** requires detailed knowledge about the building and the demand profiles, variable values for the system size is typically selected at random.

### Visualisation of results

- Display the share of renewable energy versus systems size;
- Select for the system size the optimum value and plot the two graphs.

## How to model a system (SW2)

### Task 1 - Defining goals

What do we want to achieve? → **Aim**

### Task 2 - Characterize the system

What are the relevant parameters and variables of the system? → **Variables / KPIs**

### Situation analysis

- System boundaries:
  - Depend on the question that needs answering;
  - Should include all parts with strong back-reaction;
- Structure of the system under consideration;
- Characterization of the influencing environment;
- Interpretation of actual state (SWOT);
- List of general restrictions / conditions;
- Summary of problem definition.

### Impact of system parameters

### Problem framing & sources of parameter influence

- What is the problem?: Example: external / non-controllable parameters may change the optimal solution sustainability;
- Where do these effects come from? Example: thermal energy storage integration in residential heating system with PV system and heat pump;
- Where is the parameter effect? Example: depending on utility or collaboration model, the compensation for PV injection is different.

### Pitfalls of visualisation approach

Parameters may have:

- co-dependencies;
- exclusive effect (one counteracts the second);
- infinitely many solutions.

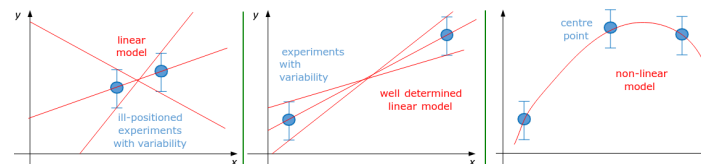
### Design of experiment (DoE)

#### Why do we need to design an experiment?

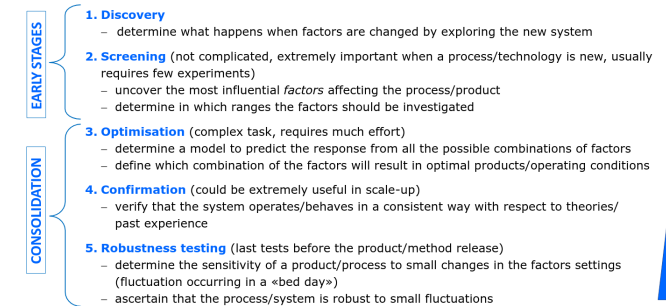
Analytical models are rare; optimisation typically involves many options and parameter ranges.

#### Impact of variability on DoE

Factor values and execution of experiments may have a major effect on results.



## Typical steps of a DoE



## Basic principles of DoE

- **Randomization**
  - both the allocation of the experimental material and the order of the individual runs of the experiment are to be performed at random
  - statistical methods require that the observations (or errors) are independently distributed random variables
  - “averaging out” the effects of extraneous factors
- **Replication**
  - making independent repeated runs of each factor combination → allows determining experimental error
  - replication is not repetition
  - repeated measurements reflect the inherent variability of the measurement system or gauge
  - replications reflect variability sources both within and between runs
- **Blocking**
  - separate experimental runs based on nuisance factors → reduces/eliminates variability from nuisance factors
  - one block is a set of relatively homogeneous experimental conditions

## Comment on factor selection

- Types of factors:**
- **design factors:** selected for the study
  - **nuisance factors:** they must be accounted for because they exert a large effect on the response
- Classification of factors:**
- **controllable:** its levels can be set by the experimenter → blocking can deal variability
  - **uncontrollable:** cannot be manipulated, but can be measured → analysis of covariance is used to compensate this effect (e.g. environmental humidity)
  - **noise:** natural and uncontrollable fluctuation that is not systematic
  - robustness studies usually minimise the noise effects
  - **held-constant factors:** despite exerting some effects on the response, they are kept constant
  - **allowed-to-vary factors:** factors that are applied in a nonhomogeneous fashion (e.g.: differences among processing units, effects of materials, etc...)

## Choice of experimental design model

Empirical model. **First-order polynomial:**

$$y = \beta_0 + \sum_{i=1}^I \beta_i x_i + \epsilon$$

$x$ 's are design factors,  $y$ 's are responses,  $\beta_i$  are parameters to be estimated, main effects are evaluated.

First-order with **interactions:**

$$y = \beta_0 + \sum_{i=1}^I \beta_i x_i + \sum_{i=1}^I \sum_{j=i+1}^J \beta_{ji} x_i \cdot x_j + \epsilon$$

**Second-order model:** Adequate for optimisation.

$$y = \beta_0 + \sum_{i=1}^I \beta_i x_i + \sum_{i=1}^I \sum_{j=i+1}^J \beta_{ji} x_i \cdot x_j + \sum_{i=1}^I \beta_{ii} x_i^2 + \epsilon$$

**Regression model:** data collected from experiments used for finding  $\beta$ , which are used in turn to estimate the response variable  $\hat{y}_{\text{NEW}}$  for new combination var.  $x_{\text{NEW}}$

**Full factor design**

Used in experiments involving several factors and perform experiments on all possible combinations of the levels of all the factors. When **L** levels are considered for **K** variables, the total **N**umber of experiments:  $N = L^K$

**SW13: Optimisation - Linear programming****Linear programming (LP)****Definition**

Linear programming is an optimization technique for problems with linear cost function  $x_{\text{opt}} = \max(a^T x)$  under the side conditions given by linear (in)equations

$$x_1 \leq 10, x_1 + x_2 \geq 3, x_1 + x_2 + x_3 = 5$$

**Why is LP relevant?**

Energy system operation schedules, energy saving potential by novel technologies, trading applications, routing of cars, busses, goods in a process.

**Prerequisites**

Linear and single cost function to optimize, linear constraints, continuous variables.

**Controller vs Linear programming****Controller**

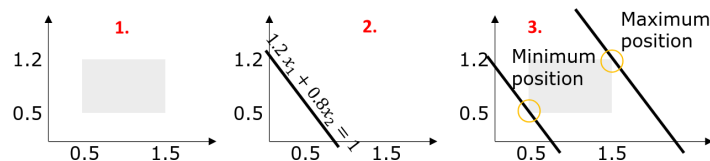
Decides based on current, historic, or predicted data.

**Linear programming application**

The optimum solution for a full period is investigated assuming perfect knowledge of the past, present, and future.

**Graphical solution of linear programs****Solving more complex systems**

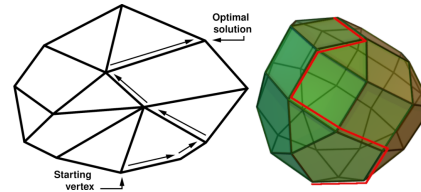
1. Identify the area of permitted variable combinations;
2. Find lines of constant cost function;
3. Optimize the cost function.

**Optimisation of a national energy system**

1. Variables, cost function and boundary conditions;
2. Energy system simulation;
3. System design + challenges;
4. Model setup;
5. Simulation results and optimized energy flows.

**The simplex algorithm**

Real-world LP problems involve very large numbers of variables and constraints, making graphical methods impractical. The simplex algorithm efficiently solves such problems by moving along the edges of the feasible region, each time selecting the direction that most improves the cost function, until no further improvements is possible.

**SW14: Mixed-integer and non-LP**