MODELLING AND SIMULATIONS

SYSID RECAP & SIMULATION & VALIDATION

AALBORG UNIVERSITY
DENMARK

Model based system identification

- *Grey-box*
  - Can be done by conventional physical experimentation and measurement methods, e.g.,
  - Example for first order system:
    - Estimate the time constant using step response
    - Estimate the DC-gain using steady-state response
- *Black-box*
  - Box-Jenkins (BJ) model
  - Output error (OE) model
  - ARMAX model
  - ARX model

Step Response Analysis *or* Transient Analysis

- Through this approach we can gain knowledge about
  - Cause and affect relationships, time delays, time constants (τ) and static gains
- Good industrial practice
- A lot of information is hidden
- Approximated model around a specific operating point
- Requires ability to manipulate the system inputs, and do it sufficiently to achieve a:
  - **Satisfactory Impulse Response**

Step Response Analysis *or* Transient Analysis

Vary the inputs

$$u(t) = u_0, \qquad t < t_0$$
$$u(t) = u_1, \qquad t \geq t_0$$

In the meanwhile measure the outputs:

$$y_0(t), \qquad t < t_0$$
$$y_1(t), \qquad t \geq t_0$$

Or let the inputs have a pule of short duration.

What can we learn from stepping the input

- **Its** effect on the system (i.e. which dynamics does it influence and what is significant to model)
  - Example: *Voltage to a motor on a quadrotor would influence the angle of the body, resulting in it have a specific acceleration, which can be translated into a force through Newtonian laws*.
- Time Constants
  - Which dynamics are most significant
    - Example: *Voltage input to a valve governing a level in a tank; the dynamic of the valve's position is insignificant compared to the level in the tank*
- Characteristics of the system (Oscilaion, damping, monotony…)

Pretreatment of Data

Sampling Interval

Filtering of data

Offset

Pretreatment of Data

Sampling Interval

- choose sufficient sampling interval
  - From frequency analysis find the bandwidth of the signal (FFT)

Pretreatment of Data

Sampling Interval
Filtering of data
Make an FFT of the signal, and use this as a baseline for filtering the signal
Design the filter and apply to your signal:

$$y_F(t) = L(q)y(t), \qquad u_F(t) = L(q)y(t)$$

Where the filter $L(q)$ is applied on both the input and output data

Be aware, The filter dynamics will influence your models dynamics
- Carefully design the filter
- Only filter the necessary frequencies, not too low:
  - Risk of removing some important system features
  - If the filtering too severe, some responses will be damped/prolonged

Be aware of filter delay!
  - Not important for offline analysis but very important for filters used in control solution

Pretreatment of Data

Sampling Interval

Filtering of data

Offset

- Remove means
- Add the offset when simulating the data

Closed Loop System

- Avoid Closed loop systems
- In some cases this is unavoidable
  - Eg. Control of valve position, from signal input
  - Make sure that the dynamics of the controller are far faster than the ones of the system characteristic which you wish to model

Choice of Model Order

- This can be done automatically in the software
- Be aware of too high order, may not be necessary
  - The noise can be modeled by the higher orders
- Use pole zero plot to check for cancelations
  - Surviving poles and zeros determine the necessary order to describe the dynamics

System Identification Toolbox

System identification toolbox:
`IDENT`

Provides:

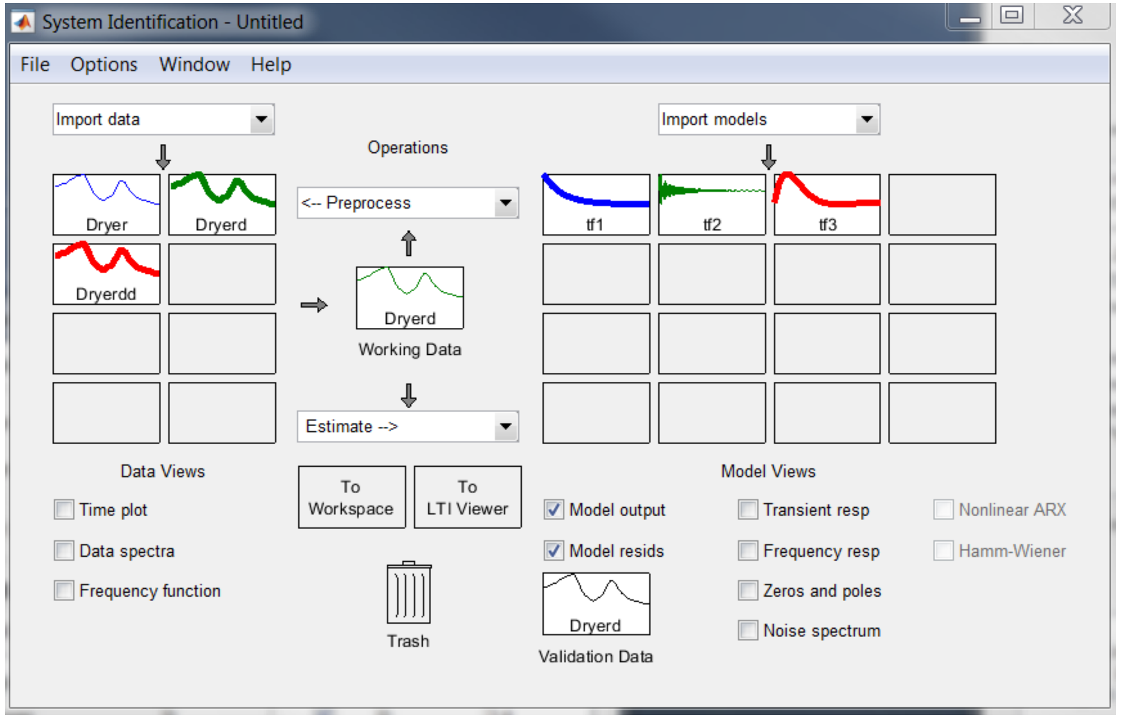
A. ***Handling of data, plotting, and the like:***
Filtering of data, removal of drift, choice of data segments, and so on

B. ***Nonparametric identification methods***
Estimation of covariances, Fourier transforms, correlation and spectral analysis, and so on.

C. ***Parametric estimation methods*** **:**
Calculation of parametric estimates in different model structures

D. ***Presentation of models:***
Simulation of models, estimation and plotting of poles and zeros, computation of frequency functions and plotting in Bode diagrams, and so on

E. ***Model validation:***
Computation and analysis of residuals $(\epsilon(t,\theta))$ comparison between different models' properties, and the like



AALBORG UNIVERSITY
DENMARK

Prepare Data for Toolbox

```
data = iddata(y,u,Ts)
```

Creates an `iddata` object containing a time-domain output signal `y` and input signal `u`, respectively. `Ts` specifies the sample time of the experimental data.

MODELLING AND SIMULATIONS

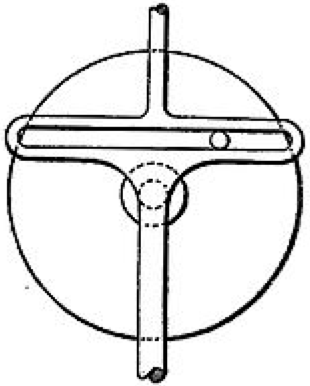
SIMULATION

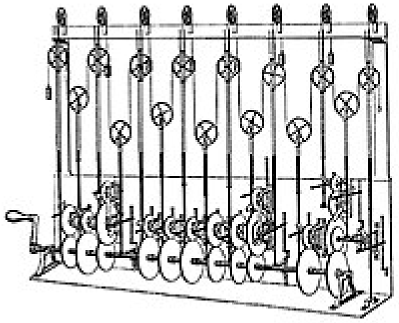
AALBORG UNIVERSITY
DENMARK

Simulation

- Solve the equation that we have found
- Present the results
- This is referred to as Simulation
- Equations are solved using numeric tools (Matlab)

- Originally done mechanically, eg. For sinusoidal function:
  - Expresses $A_1 \cos(\omega_1 t + \phi_1)$





https://en.wikipedia.org/wiki/Tide-predicting_machine

AALBORG UNIVERSITY
DENMARK

Numeric Methods

- Many versions exist:
- Simplest is the Euler's method

Euler's method

Approximating $\dot{x}(t)$, with a difference ratio

$$\frac{\boldsymbol{x_{n+1} - x_n}}{\boldsymbol{h}} \approx \boldsymbol{\dot{x}(t_n) = f(t_n, x_n),} \qquad \boldsymbol{where\ h = t_{n+1} - t_n}$$

Giving the following equation:

$$\boldsymbol{x_{n+1} = x_n + hf(t_n, x_n)}$$

An explicit one step method

  - as x_{x+1} is not included in the expression and we calculate one step ahead
- This method is not that effective
  - It has a small region of stability
  - It has a lower accuracy than other available methods, *'large local error'*

Numeric Methods

- Many versions exist:
- Simplest is the Euler's method

Euler's method

Approximating $\dot{x}(t)$, with a difference ratio

$$\frac{x_{n+1} - x_n}{h} \approx \dot{x}(t_n) = f(t_n, x_n), \qquad where\ h = t_{n+1} - t_n$$

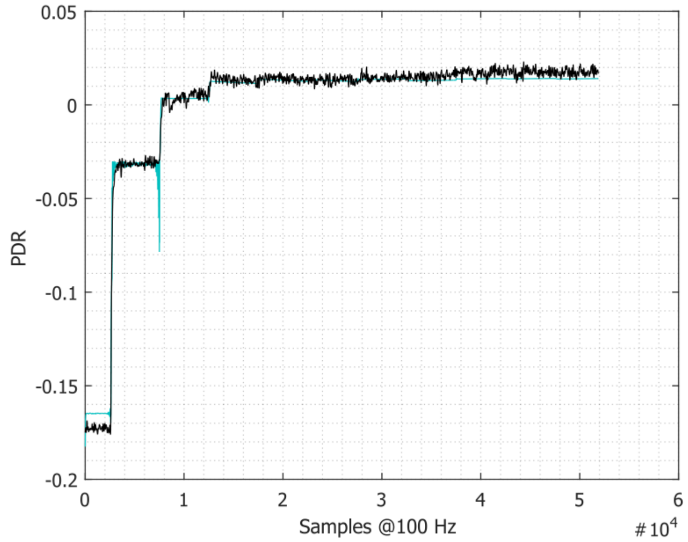
Giving the following equation:

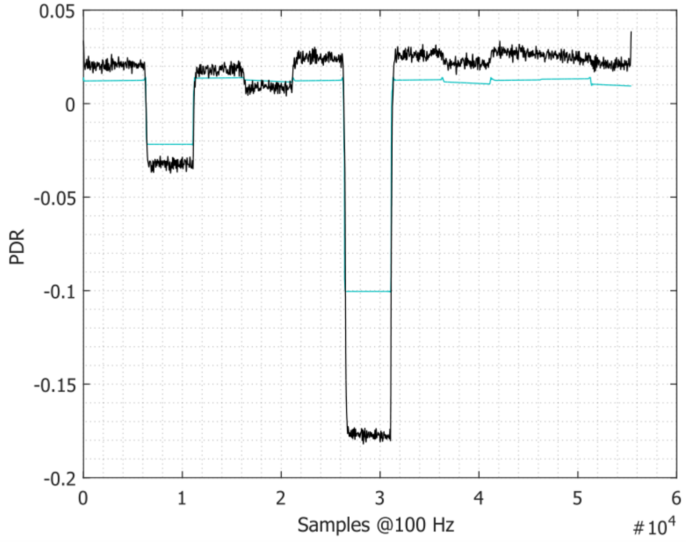
$$x_{n+1} = x_n + hf(t_n, x_n)$$

- This method is not that effective
- We use other methods such as: Runge Kutta, Adam's Method, Variable Step Length…
  - More info in section 11.6 (Modeling of Dynamic Systems, Ljung)

Presentation of the Simulations

- Plot of the model simulations
    - Remember to add offset
- We can do it in time or frequency domain
- We can choose to show multiple dimensions

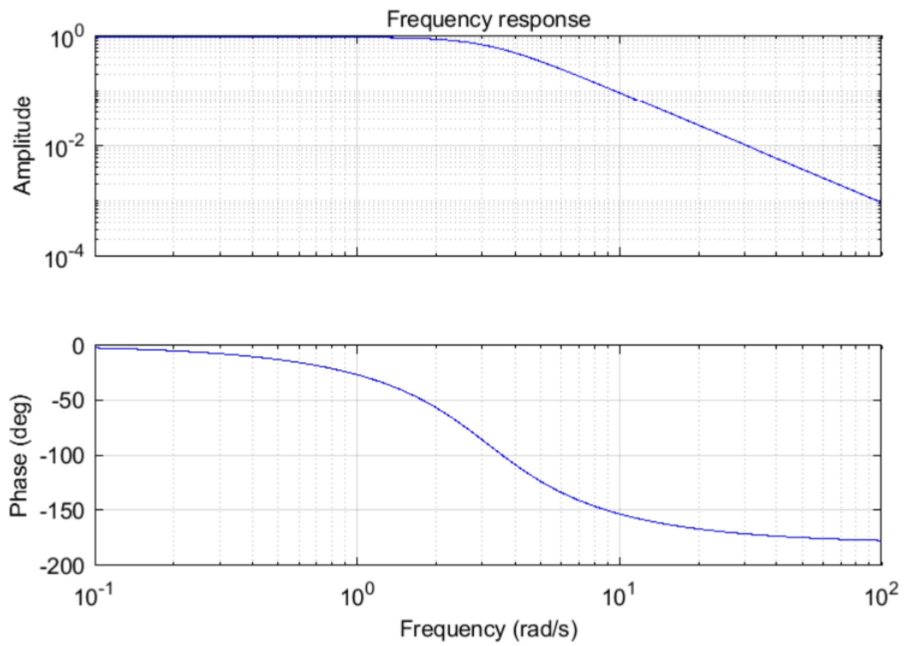




AALBORG UNIVERSITY
DENMARK

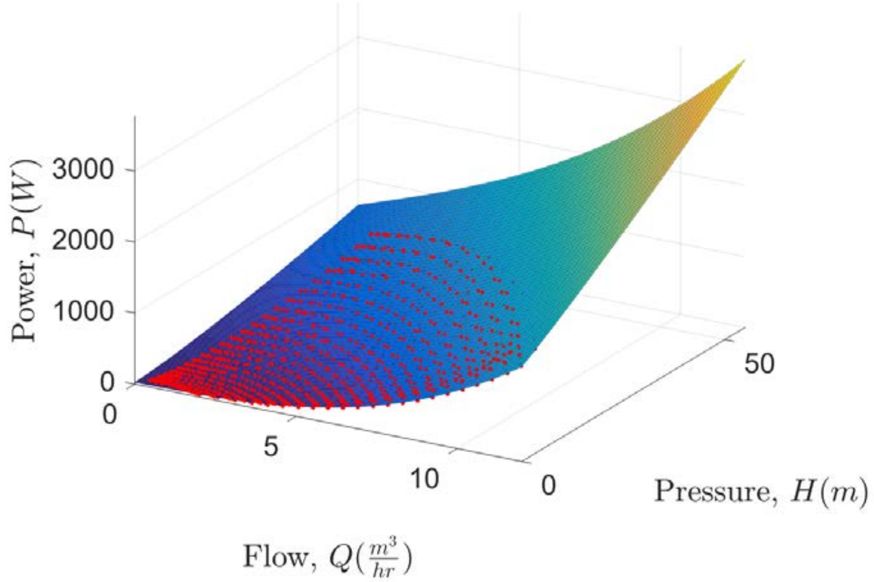
Presentation of the Simulations

- Plot of the model simulations
- We can do it in time or frequency domain
- We can choose to show multiple dimensions



Presentation of the Simulations

- Plot of the model simulations
- We can do it in time or frequency domain
- We can choose to show multiple dimensions



MODELLING AND SIMULATIONS

MODEL VALIDATION AND MODEL USE

AALBORG UNIVERSITY
DENMARK

Model Validation

- What is a Valid model?
- For our purpose it can yield good result
    - But it may be wrong
- We need to test the model’s validity

Model Validation

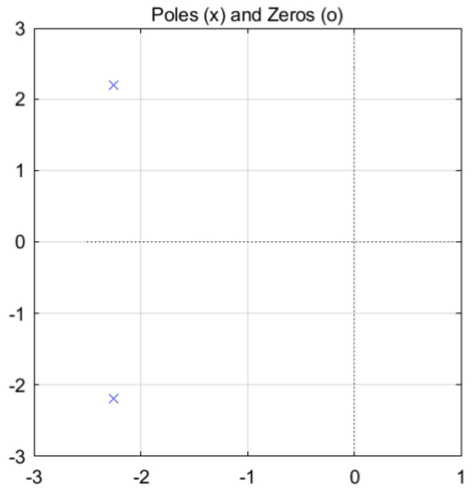
- What is a Valid model?
- For our purpose it can yield good result
  - But it may be wrong
- We need to test the model's validity

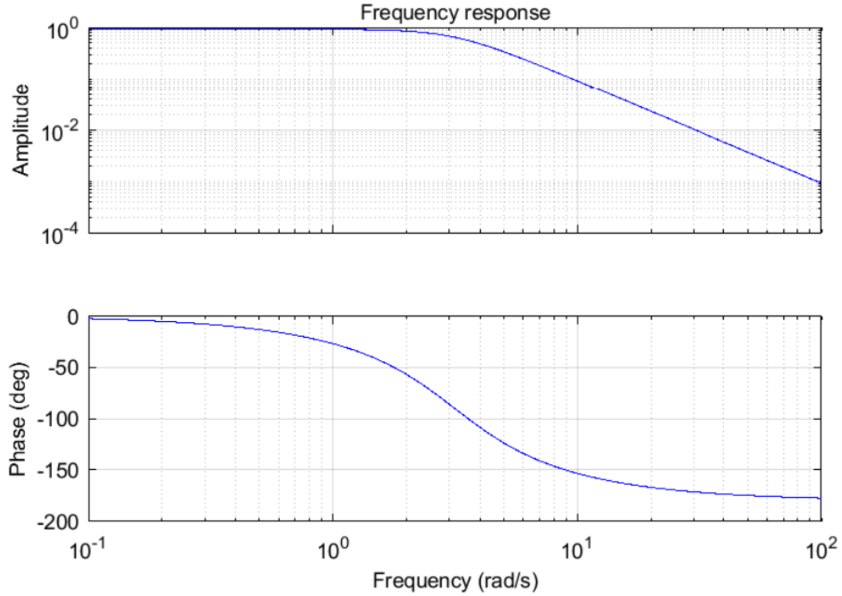
Model Validation

- What is a Valid model?
- For our purpose it can yield good result
  - But it may be wrong
- We need to test the model’s validity

Model Validation

1. Model Quality
   1. Stability
      1. Poles and Zeros
      2. Bode analysis, frequency function



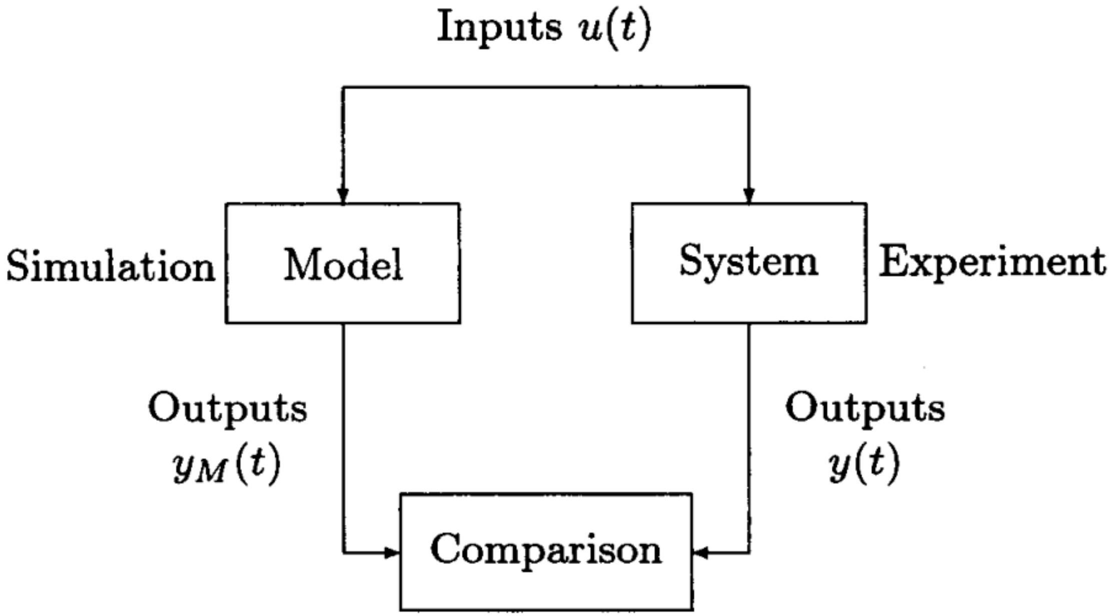


Model Validation

1. Model Quality
   1. Stability (Bode analysis, frequency function)
   2. Ability to reproduce system behavior
      1. Input output behavior comparison (simulated to real(new data set))
      2. k-step prediction (prediction horizon k larger than τ)
         1. Can be done using the residuals (`resid` ‘1 step ahead prediction errors’)

Test of Validity

- *Compare the **model's** output with the **experimental** results*
- *The **comparison** of the two **outputs** must be **small***



Test of Validity

- *Compare the **model's** output with the **experimental** results*
- *The **comparison** of the two **outputs** must be **small***

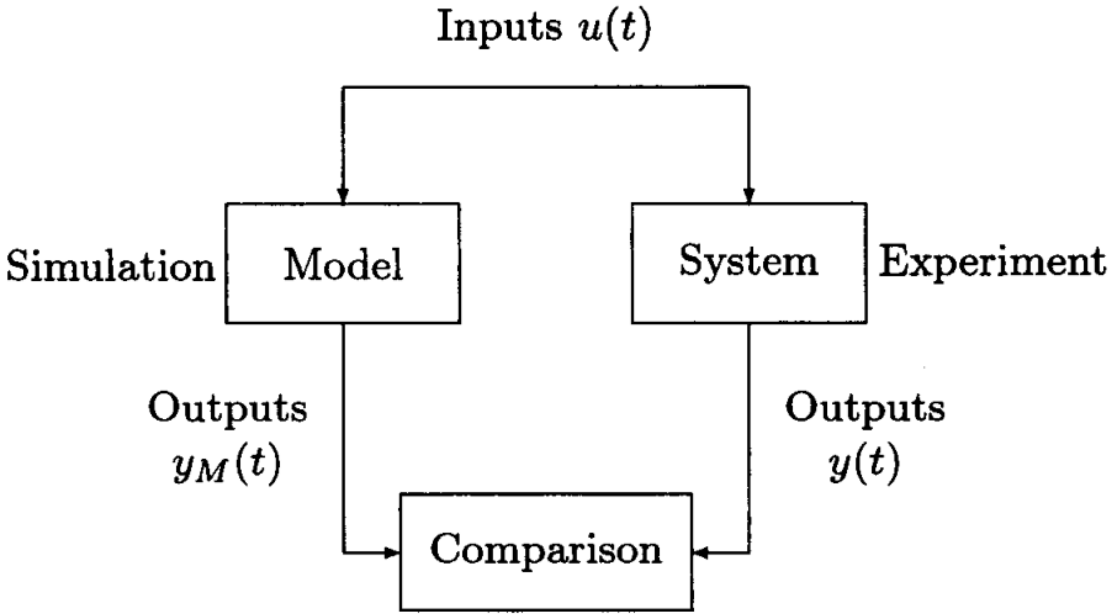
How to determine what is small enough?

Depends on:

- The purpose of the model
- Disturbances influencing the model

Which parameters influence the system most

  - Pay attention to these and update



Model Validation – Residual Analysis

- Goodness of fit between test and reference data
    - The goodness of fit is calculated using the normalized root mean square error as the cost function

`goodnessOfFit`

AALBORG UNIVERSITY
DENMARK

Model Validation

1. Model Quality
2. Residual Analysis

Model validation – Residual Analysis

This is a guess, or a prediction of $y(t)$ at time $(t-1)$

$$\hat{y}(t|\theta)$$

Where θ is the parameter vector which contains the parameters of the identified model, where parameters of θ are to be adjusted to collected data.

Then we can calculate the residuals, *(the parts of the data that the model could not reproduce)*

$$\epsilon(t) = \epsilon\left(t, \hat{\theta}_n\right) = y(t) - \hat{y}(t|\hat{\theta}_n)$$

In Matlab you can also call

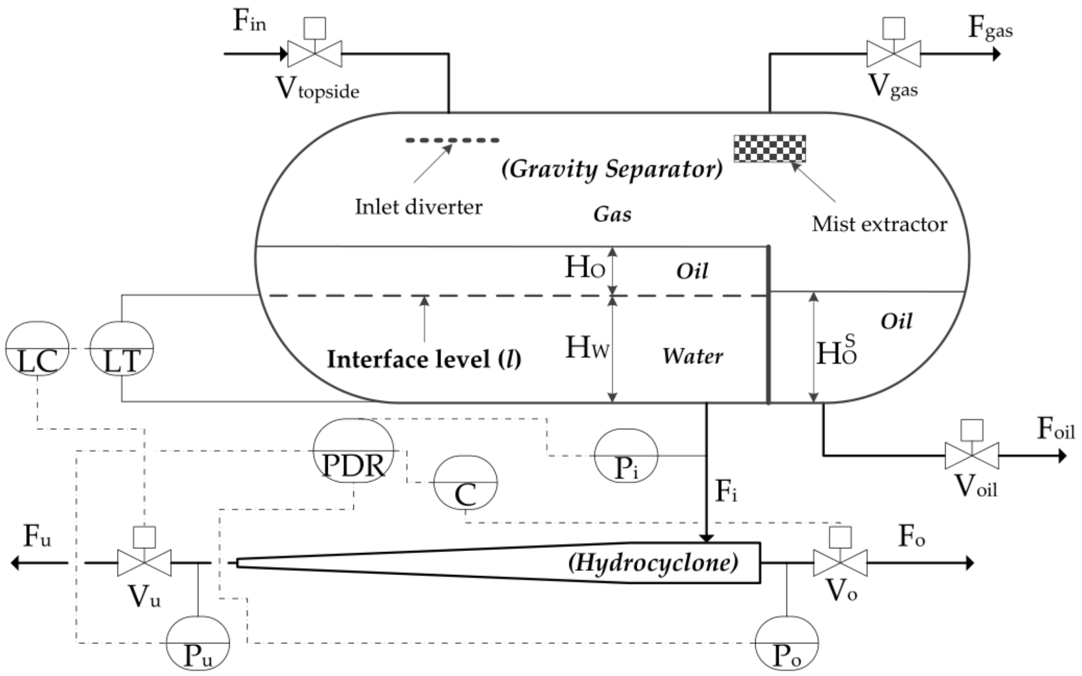
```
resid(Data,sys)
```

Domain of Validity

- Operating range
  - Right operating conditions
  - Must assure that

Domain of Validity

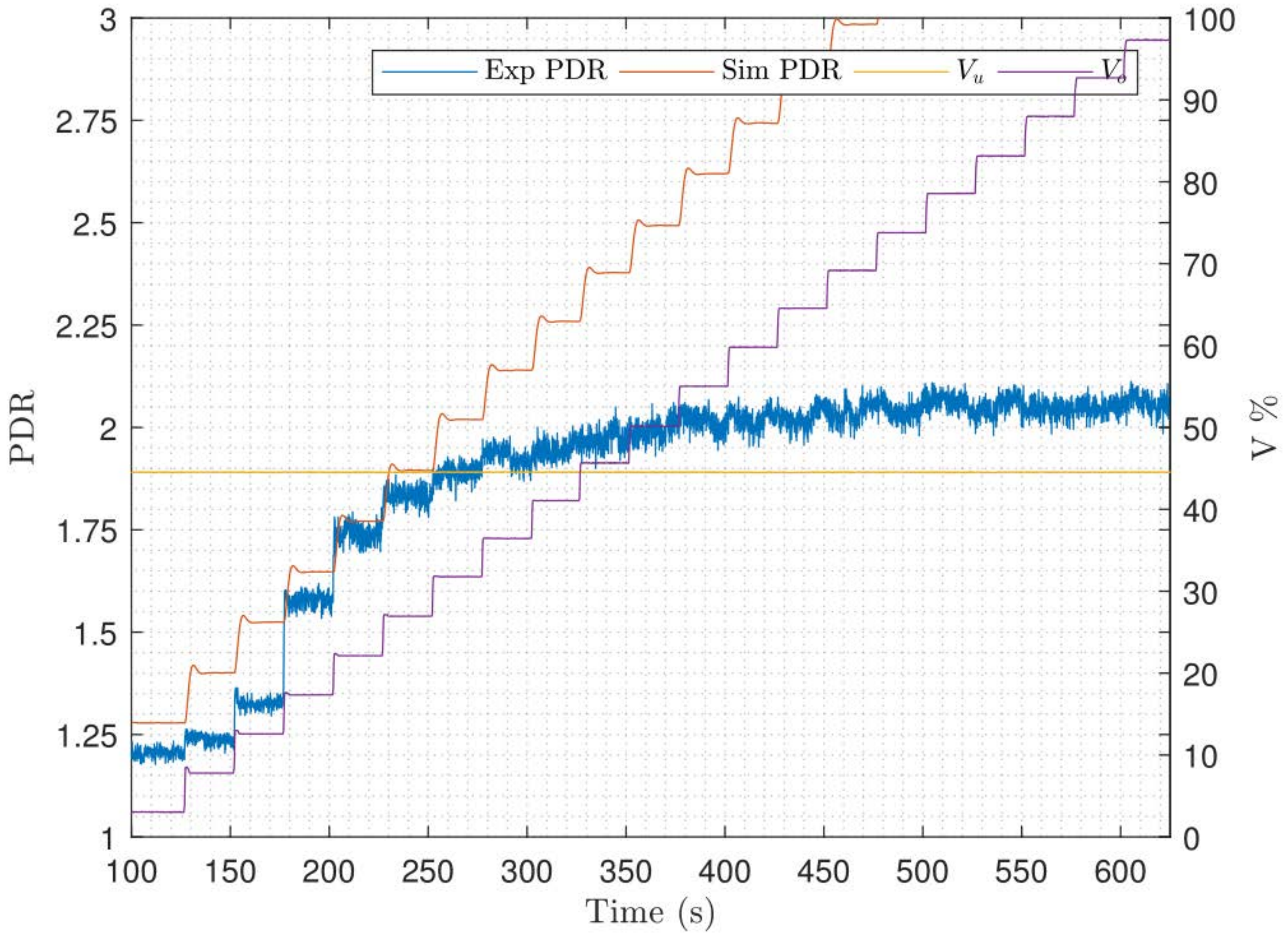
- Example with a model of a valve governing the PDR of a hydrocyclone
- Valve V_o is the input
- The pressure drop ratio (PDR) is the output
  - $\mathrm{PDR} = \frac{P_i - P_o}{P_i - P_u}$
- A linear model is generated from data collected



AALBORG UNIVERSITY
DENMARK

Domain of Validity

- Example with a model of a **valve** governing the **PDR** of a hydrocyclone
- The linear model is identified around a **operating point** of:
  - **PDR = 1.8%**
  - **Pressure 7 bars**
- Moving **too far** away from this point results in deviation from real data
- Thus the mode is only **valid** within a certain operating point



AALBORG UNIVERSITY
DENMARK

Models Validity and the Critical View

- A model may be invalid for changing operating conditions
  - Such as PDR, pressure, valve opening…
- A model cannot be made *Perfect*
  - *i.e. it cannot have a 100% fit for all conditions*
- Make sure that the model fits well for your use and operating conditions
- Be prepared to modify the model to fit to new operating conditions if necessary
- Do not over complicate the model, model the necessary dynamics

Exercise

- Finish the model design in SYSID from the last lecture
- Simulate the model (hint use here the toolbox)
  - Is the model valid
  - Does the model represent the system well
- Export the model and simulate the model externally, eg. Simulink
  - Hint for parameter extraction from *idtf*: *tf1.Numerator, tf1.Denominator*
  - Use the data from the example as input
    - Achieve the same result as the toolbox
  - Try different inputs and observe the output