Model Parameterization and Validation

Prof. Cesar de Prada
Dpt. Systems Engineering and Automatic Control
University of Valladolid
prada@autom.uva.es

Outline

- ✓ Introduction
- ✓ Parameterization / Calibration
- ✓ Sensitivities
- ✓ Identifiability
- ✓ Dynamic optimization
- ✓ Model validation

Modelling Methodology

There are specific methods and tools to facilitate the implementation of these steps

Process knowledge

Aims, Model kind

Hypothesis

Model formulation

Parameter estimation

Model Evaluation

Explotation

time

Simulation

		Tools
Range of validity	Hypothesis ↓	
		Knowledge
Model aims, required precision,...	Modelling ↓	
Formulation for a specific calculation	Coding ↓	Simulation language
	Parameterization ↓	Experimentation
Is it adequately solved?	Validation/ Experimentation	**All stages must be validated**

Modelling

$$\frac{d\,x(t)}{dt} = f(x(t), u(t), p, t)$$

$$y(t) = g(x, u(t), t)$$

Two steps:

- Model structure building
- Estimating the value of the model parameters

Generally, a choice among different possible model structures must be made, in agreement with the selected hypothesis

At the validation stage, several test can be proposed to select the best option

Parameterization

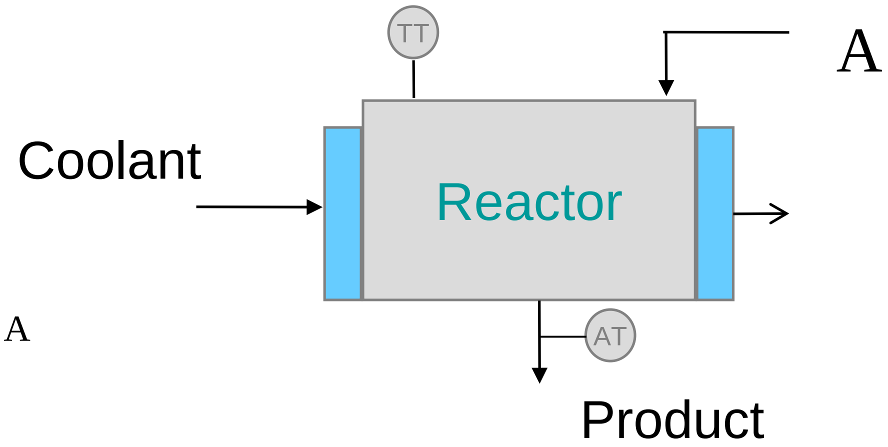
$$\frac{d\,x(t)}{dt} = f(x(t), u(t), p, t)$$

$$y(t) = g(x, u(t), t)$$

Some model parameters can be obtained from bibliography, documentation, etc. but there are always other ones that must be estimated from experimental data.

Example: Chemical reactor

$$A \rightarrow B$$



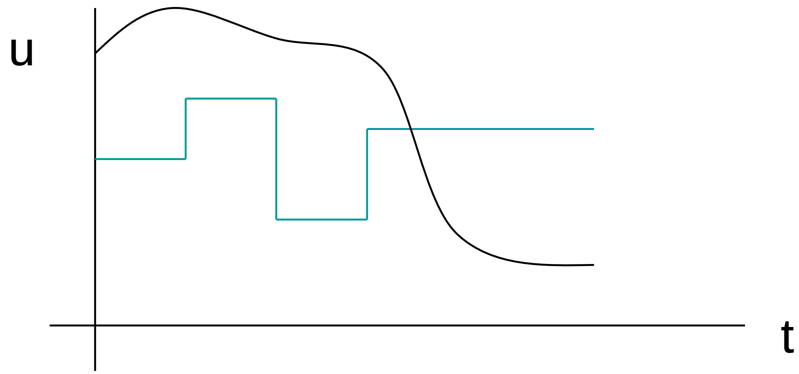
$$V \frac{dc_A}{dt} = Fc_{Ai} - Fc_A - Vke^{-E/RT} c_A$$

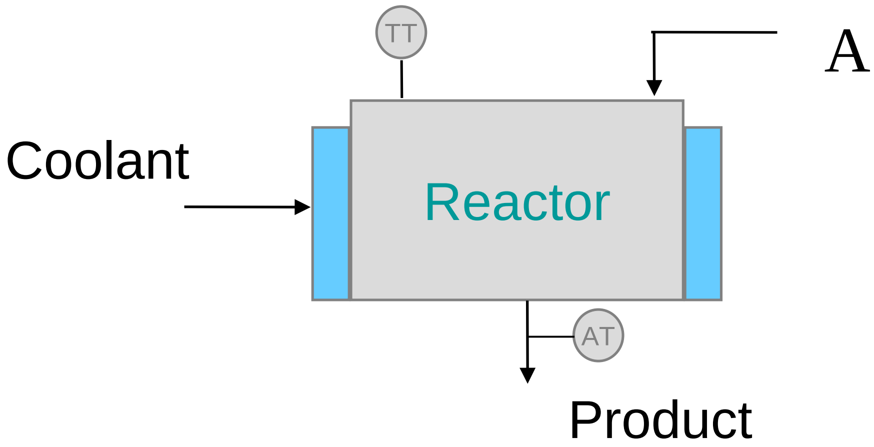
$$V \frac{dc_B}{dt} = -Fc_B + Vke^{-E/RT} c_A$$

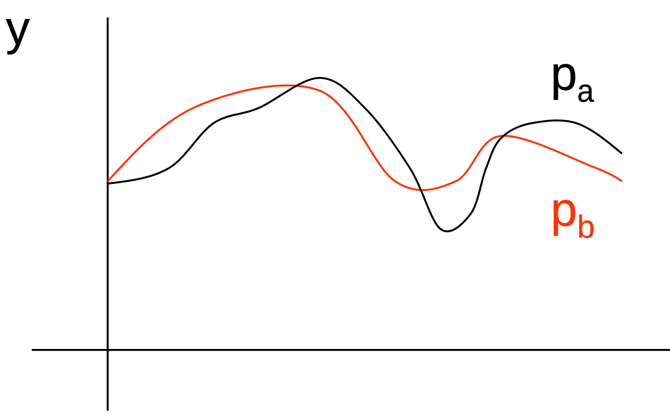
$$V\rho c_e \frac{dT}{dt} = F\rho c_e T_i - F\rho c_e T + Vke^{-E/RT} c_A \Delta H - UA(T - T_r)$$

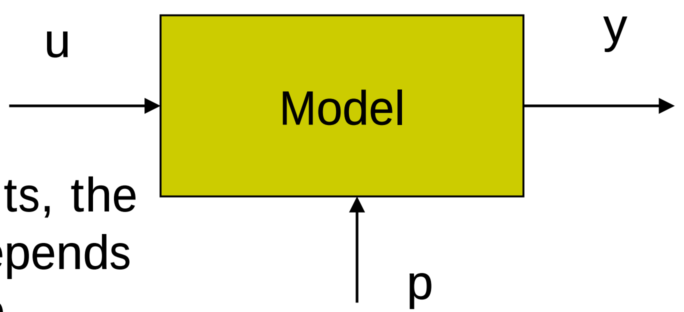
$$V_r \rho_r c_{er} \frac{dT_r}{dt} = F_r \rho_r c_{e_r} T_{ri} - F_r \rho_r c_{er} T_r + UA(T - T_r)$$

Parameterization (calibration)



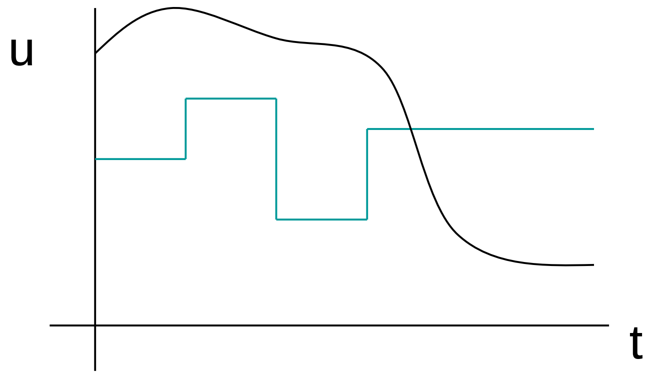


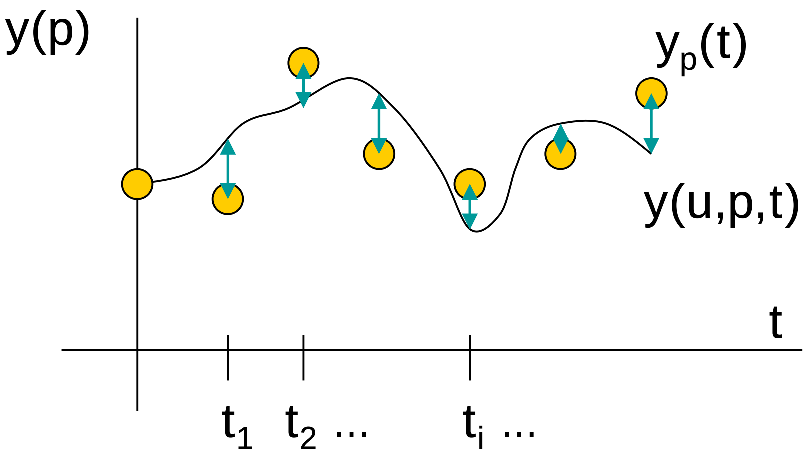




Given a set of inputs, the model response depends on the value of the parameters p

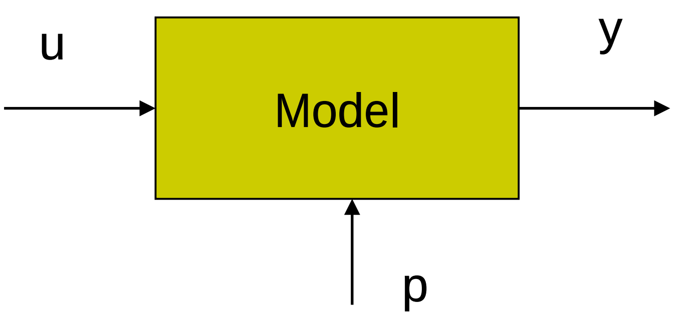
Parameterization (calibration)





If N measurements $y_p(t)$ of the process output have been collected, it is possible to define the error function:

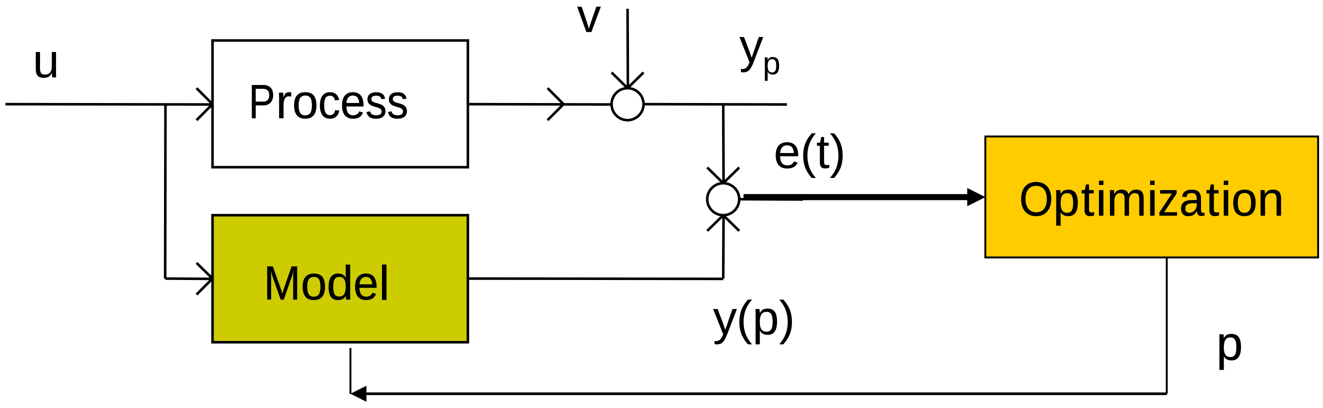
$$J = \frac{1}{N}\sum_{i=1}^{N} e(t_i)^2 = \frac{1}{N}\sum_{i=1}^{N}\left[y(u,p,t_i) - y_p(t_i)\right]^2$$



Other formulations can be made also for J

Parameterization (calibration)

$$\min_p J = \min_p \frac{1}{N}\sum_{i=1}^{N} e(t_i)^2 = \min_p \frac{1}{N}\sum_{i=1}^{N}\left[y(u,p,t_i) - y_p(t_i)\right]^2$$



J normally is a non-linear function of the parameters p, and the problem must be solved numerically using DO methods.

Dynamic model parameterization

$$\min_{p} J = \min_{p} \sum_{i=1}^{N} \left[y(u,p,t_i) - y_p(t_i) \right]^2$$

$$\dot{x}(t) = f(x(t),u(t),p) \qquad y(t) = g(x(t),u(t),p)$$

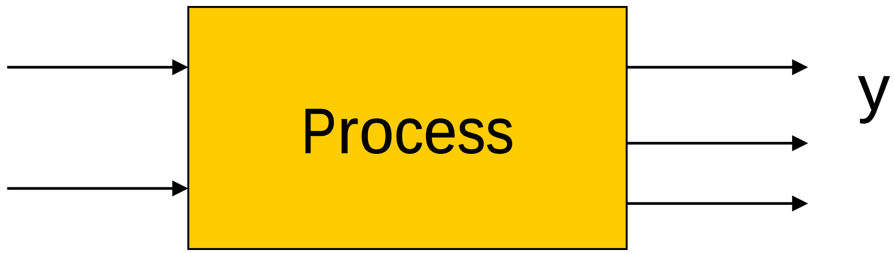
$$\underline{p} \le p \le \overline{p}$$

Dynamic optimization with constraints problem, DO

Besides the explicit model parameters, unknown initial states , disturbances or non measured inputs can also be included in the parameter estimation problem.

Parameterization

$$\min_p J = \min_p \frac{1}{N}\sum_{i=1}^{N} e(t_i)^2 = \min_p \frac{1}{N}\sum_{i=1}^{N}\left[y(u,p,t_i) - y_p(t_i)\right]^2$$



Generally, there are several measured process outputs:

$$\min_p \frac{1}{N}\sum_{i=1}^{N}\gamma_1\left[\frac{y_1(p,t_i) - y_{p1}(t_i)}{\bar{y}_1}\right]^2 + \gamma_2\left[\frac{y_2(p,t_i) - y_{p2}(t_i)}{\bar{y}_2}\right]^2 + \gamma_3\left[\frac{y_3(p,t_i) - y_{p3}(t_i)}{\bar{y}_3}\right]^2$$

Attention should be paid to avoid mixing variables with different units and orders of magnitude. Normalization is required. Weights reflect relative importance

Other possible cost functions

$$\min_{p} \sum_{i=1}^{N} \left| y(t_i, p) - y_p(t_i) \right|$$

Norm 1

All errors are equally weighted

$$\min_{p} \max_{i} \left| y(t_i, p) - y_p(t_i) \right|$$

∞ Norm

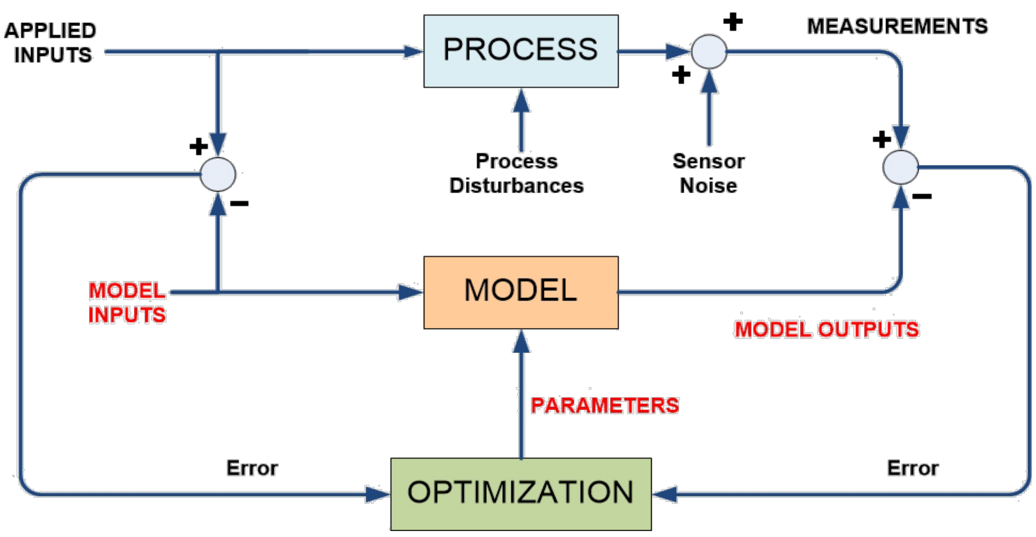
Minimizes the largest error

$$J(p) = \min_{p} \sum_{i=1}^{N} \frac{1}{\sigma(t)^2} \left(y(t_i, p) - y_p(t_i) \right)^2$$

Weighted LS

Errors are weighted inversely to the noise present in the data

Data reconciliation



Data reconciliation intends to:

- Estimate the values of all variables and model parameters coherent with a process model and as close as possible to the measurements
- Detect and correct inconsistencies in the measurements

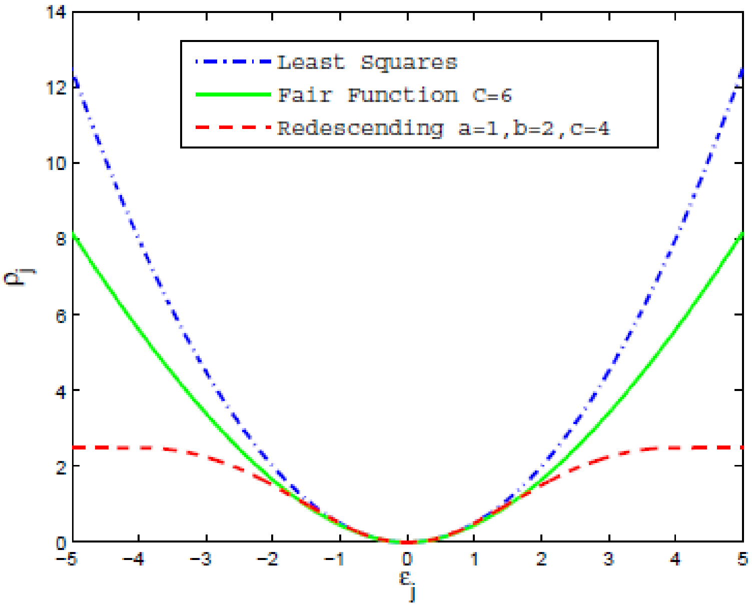
$$\min_{\theta=\{\hat{x},\hat{y},\hat{u},p\}} J = \sum_{i=1}^{n_i} \frac{|u_i-\hat{u}_i|}{\sigma_i} + \sum_{j=1}^{n_o} \frac{|y_j-\hat{y}_j|}{\sigma_j} \quad \text{subject to: } \begin{array}{c} \underline{\theta} \leq \theta \leq \overline{\theta} \\ \text{model} \end{array}$$

Robust estimators

Fair function

$$J = C^2 \sum_{i=1}^{n} \left(\frac{|\varepsilon_i|}{C} - \log\left(1 + \frac{|\varepsilon_i|}{C}\right)\right)^2$$

$$\varepsilon_i = \frac{y_i - \widehat{y_i}}{y_i}$$



Welsch

$$\mathrm{J} = \frac{c^2}{2}\left(1 - e^{-\left(\frac{\varepsilon_i}{c}\right)^2}\right)$$

Robust estimators

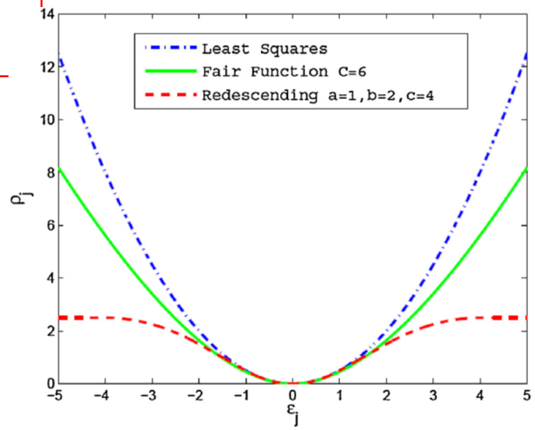
$$R_j = \begin{cases} 0.5\varepsilon_j^2 & 0 \le |\varepsilon_j| \le a \\ a|\varepsilon_j| - 0.5a^2 & a \le |\varepsilon_j| \le b \\ ab - 0.5a^2 + 0.5a(c-b)\left(1 - \left(\dfrac{c - |\varepsilon_j|}{c-b}\right)^2\right) & b \le |\varepsilon_j| \le c \\ ab - 0.5a^2 + 0.5a(c-b) & c \le |\varepsilon_j| \end{cases}$$

$c > b + 2a$

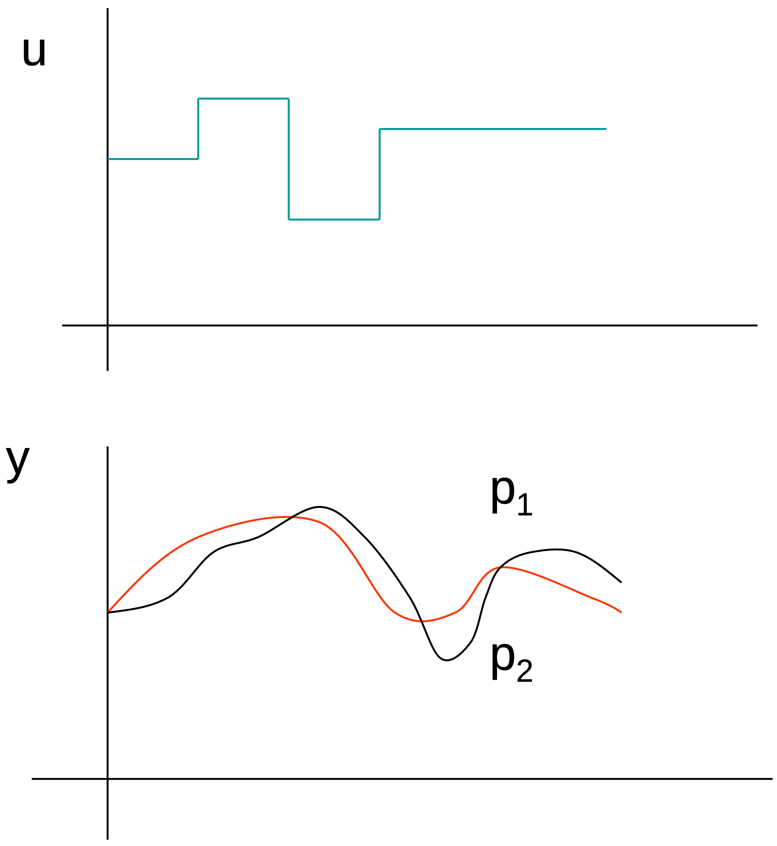
Hampel's Redescending estimator R

a, b, c tuning parameters

Smoothing functions



Which parameters should be estimated?



It may happen that the model response to a given set of inputs for two values p_1 and p_2 of a certain parameter does not change in a significant way.

A parameter should be included in the optimization only if the model output presents a sensible sensitivity to changes in the parameter.

Output sensitivities
Cost function sensitivity

$$J = \sum_{i=1}^{N} \left[y(p,u,t_i) - y_p(t_i) \right]^2$$

$$\dot{x}(t) = f(x(t), u(t), p) \qquad y(t) = g(x(t), u(t), p)$$

$$S_{ij}(t) = \frac{\partial y_i(t)}{\partial p_j} \qquad \frac{\partial J}{\partial p_j}$$

Sensitivity of the model output i with respect to parameter j in a given experiment. Notice that it is a time function.

Sensitivity of cost function J with respect to parameter j in a given experiment.

Summarize the effect of the parameter change over the whole experiment.

Output sensitivities

$$S_{ij}(t) = \frac{\partial y_i(t)}{\partial p_j}$$

It is difficult to compare output sensitivities due to the different units in which they are expressed. It is better to use relative sensitivities

$$s_{ij}(t) = \frac{p_j}{\bar{y}_i} \frac{\partial y_i(t)}{\partial p_j}$$

$$\begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1d} \\ s_{21} & s_{22} & \cdots & s_{2d} \\ \vdots & \vdots & \cdots & \vdots \\ s_{m1} & s_{m2} & \cdots & s_{md} \end{bmatrix}$$

The norm of column j of the output sensitivity matrix provides a measure of the importance of parameter p_j in the value of the model outputs.

Computing output sensitivities

They can be obtained using finite differences approximation or integrating the extended system.

One option is to use model simulations with small perturbations on each parameter involved

$$S_{ij}(t) = \frac{\partial y_i(t)}{\partial p_j} \approx \frac{y_i(p+\Delta p, t) - y_i(p, t)}{\Delta p}$$

The value of the sensitivities obtained depends on the point p considered and the experiment that was performed as u(t) is involved in the simulations.

Extended system

$$J = \sum_{i=1}^{N} \left[y(p,u,t_i) - y_p(t_i) \right]^2$$

sensitivities $s(t) = \dfrac{\partial x(t)}{\partial p}$

$$\dot{x}(t) = f(x(t), u(t), p) \qquad y(t) = g(x(t), u(t), p)$$

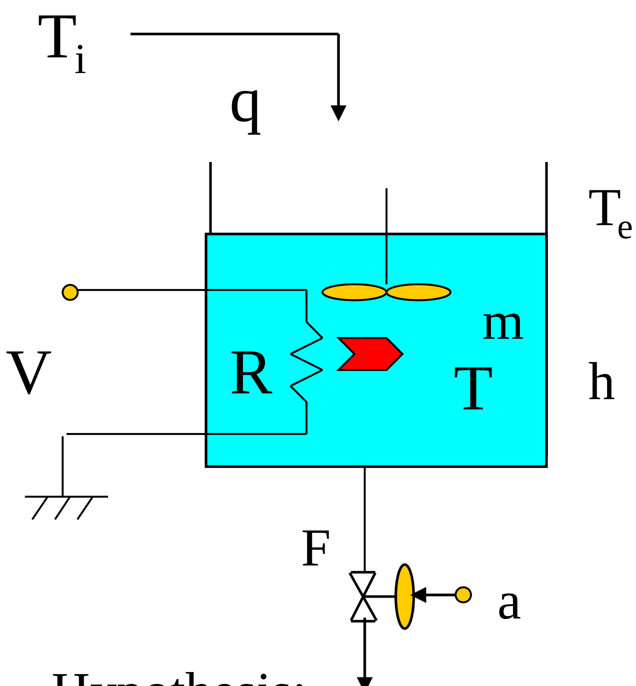
$$\frac{\partial J}{\partial p} = 2 \sum_{i=1}^{N} \left[y(p,u,t_i) - y_p(t_i) \right] \frac{\partial y}{\partial p}$$

$$\frac{\partial \dot{x}}{\partial p} = \boxed{\frac{d}{dt} \frac{\partial x}{\partial p} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial p} + \frac{\partial f}{\partial p}}$$

$$\frac{\partial y}{\partial p} = \frac{\partial g}{\partial x} \frac{\partial x}{\partial p} + \frac{\partial g}{\partial p}$$

Integrating this equation, besides the ones of the model, it is possible to obtain the time evolution of the sensitivities $\partial x / \partial p$ and, hence, the output sensitivities. IDAS

Example: Heated tank



Hypothesis:

T lumped temperature
ρ constant density
c_e constant specific heat

T temperature
m tank mass
h liquid level
V voltage
q Inflow
F Outflow
a valve opening
H entalphy
c_e specific heat
A tank cross section
ρ density
R resistence
T_e external temperature
T_i inflow temperature

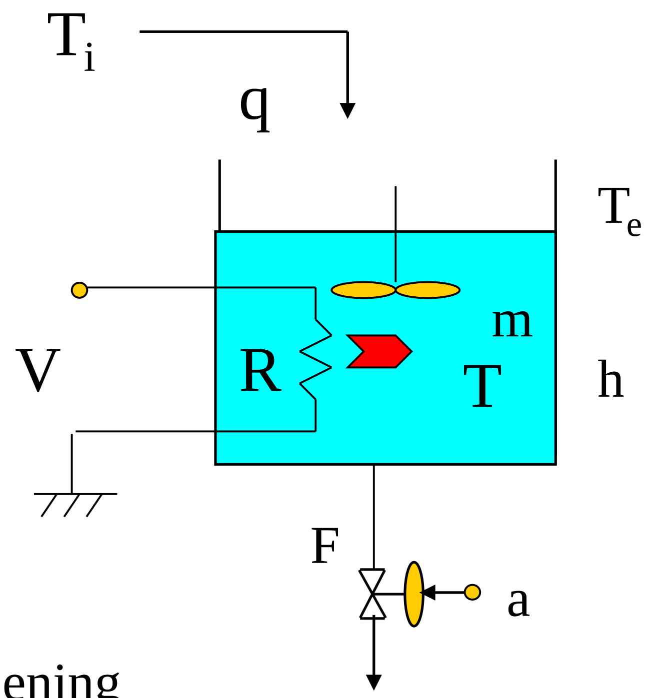
Dynamic model

State space non-linear model

$$\frac{dh}{dt} = \frac{1}{A}(q - F)$$

$$\frac{dT}{dt} = \frac{q}{Ah}(T_i - T) + \frac{V^2}{AhR\rho c_e} - \frac{U_{amb}}{Ah\rho c_e}(T - T_e)$$

$$F = ak\sqrt{h}$$



CV: T temperature, h level

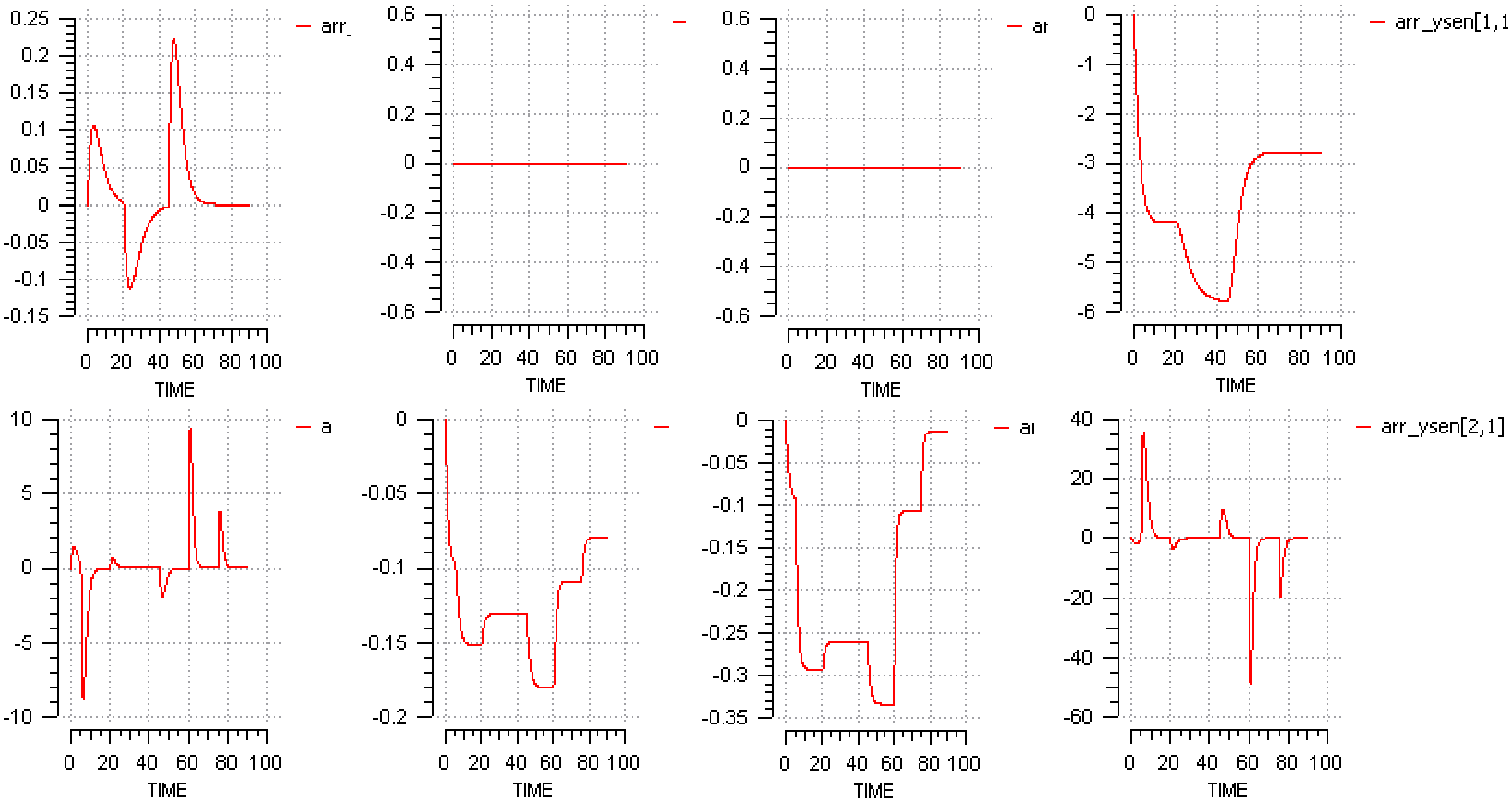
MV: V voltage, a valve opening

Disturbances: q Inflow, T_i Input temperature

States: h , T

Other variables: F outflow

Sensitivities



h

T

A   Uamb   R   k

Identifiability

We say a model is identifiable if it is possible to obtain the value of its parameters provided that sufficient number of process measurements is available.

In practice, identifiability means that a certain model output corresponds to a certain value of the parameter set. But it may happens that the same effect on the output can be attained either modifying one parameter or another. In this case, there is a certain co-linearity in that parameter set which makes very difficult the identification.

Identifiability is a structural property of the model, but the identification of particular parameter can depend also on the experimental data.

Identifiability examples

In a chemical reactor, without temperature measurements, parameters k and E, cannot be estimated independently

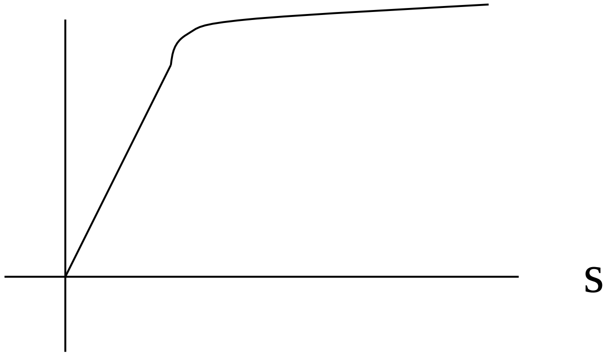
$$\frac{dc_B}{dt} = -\frac{F}{V}c_B + ke^{-E/RT}c_A$$

In the model, it is possible to identify the ratio F/V, but not F and V independently.

With steady state data, V cannot be identified.

$$\frac{\mu_m s}{K+s}$$

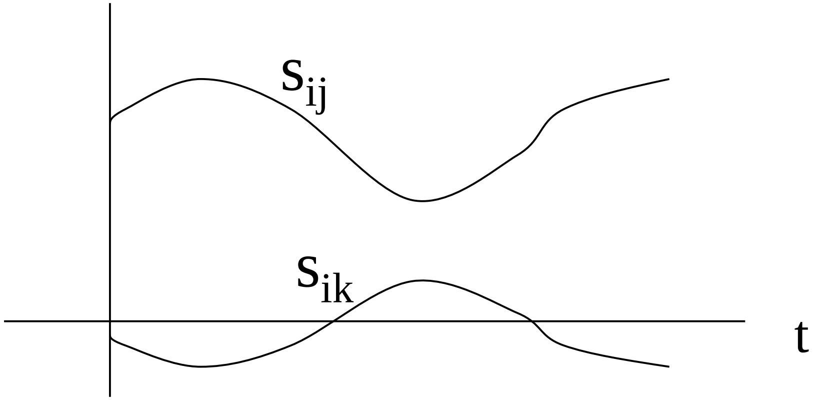
In the Monod model, using data limited to small values of s, only the ratio μ_m/K can be identified. With data containing only large values of s, only μ_m can be identified well.



Identifiability

With the output sensitivity matrix, it is possible to see if a certain degree of co-linearity exists, by inspecting range of its sub-blocks (by rows).

$$\begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1d} \\ s_{21} & s_{22} & \cdots & s_{2d} \\ \vdots & \vdots & \cdots & \vdots \\ s_{m1} & s_{m2} & \cdots & s_{md} \end{bmatrix}$$



Co-linearity makes parameter identification more difficult

Re-parameterization

Sometimes it is only possible to identify certain parameter combinations, or new variables can be defined in the model to obtain a model structure easier to identify.

If either of these alternatives has been implemented, it is necessary to remember that:

- ✓ Statistical characteristics of new variables are different to the ones of the data set.
- ✓ Confident regions of the new parameters are different to the ones of the original parameter.

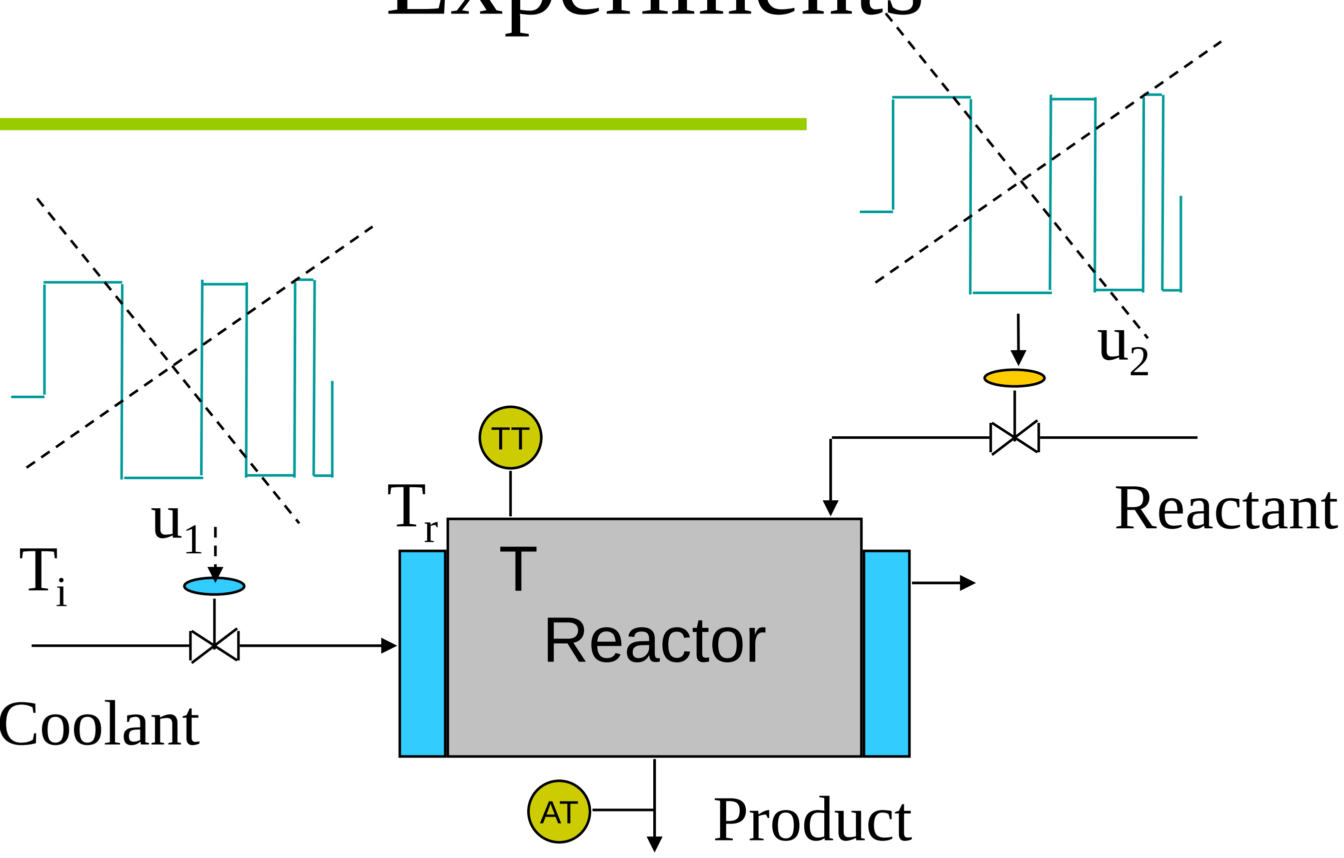
Experiments

- In order the model to capture the process dynamics, the experimental data used in the identification must contain information about that dynamics.
- Hence, the experiments should cover different operating conditions, exciting the different modes of operation of the process. This should be taken into account when selecting the excitation signals, planning its amplitude, signal to noise ratio and frequency .
- Historical records tend to be useless as they have poor dynamic information, as the operators or the control system try to maintain the process as stable as possible.

Experiments

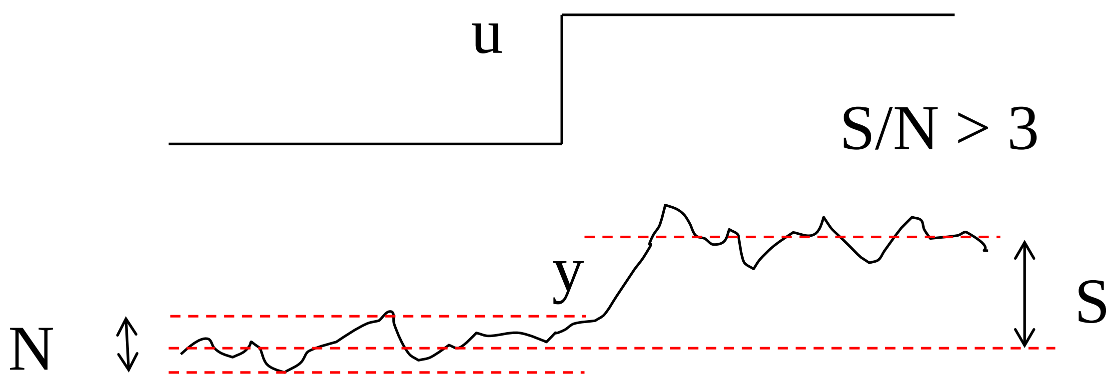
- Sampling period must be selected according to the intended use of the model and the process dynamics. For control applications, the desired closed loop settling time must be considered. For data acquisition, shorter sampling period can be used.
- Length of the experiment should consider data collection for identification and validation (1000 $\rightarrow$)
- Changes in the amplitude of the test signals should be large enough to obtain an adequate output signal / noise relation and covering all operational range of interest
- Test signals should cover the range of frequencies relevant for the process considered.

Experiments

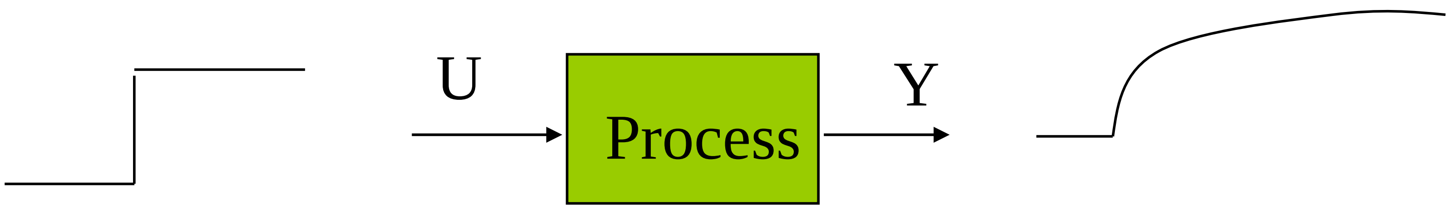


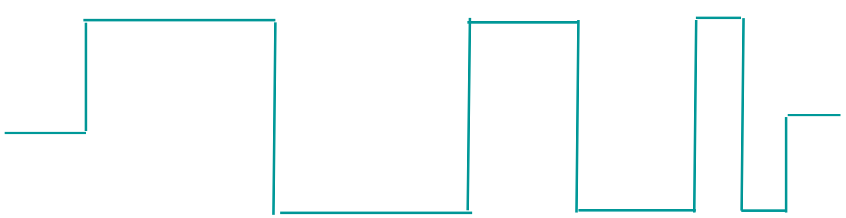
If the process has several inputs, changes applied to themshould be uncorrelated, sothat the optimization algorithm can distinguish the effects of every input on every output.

Experiments



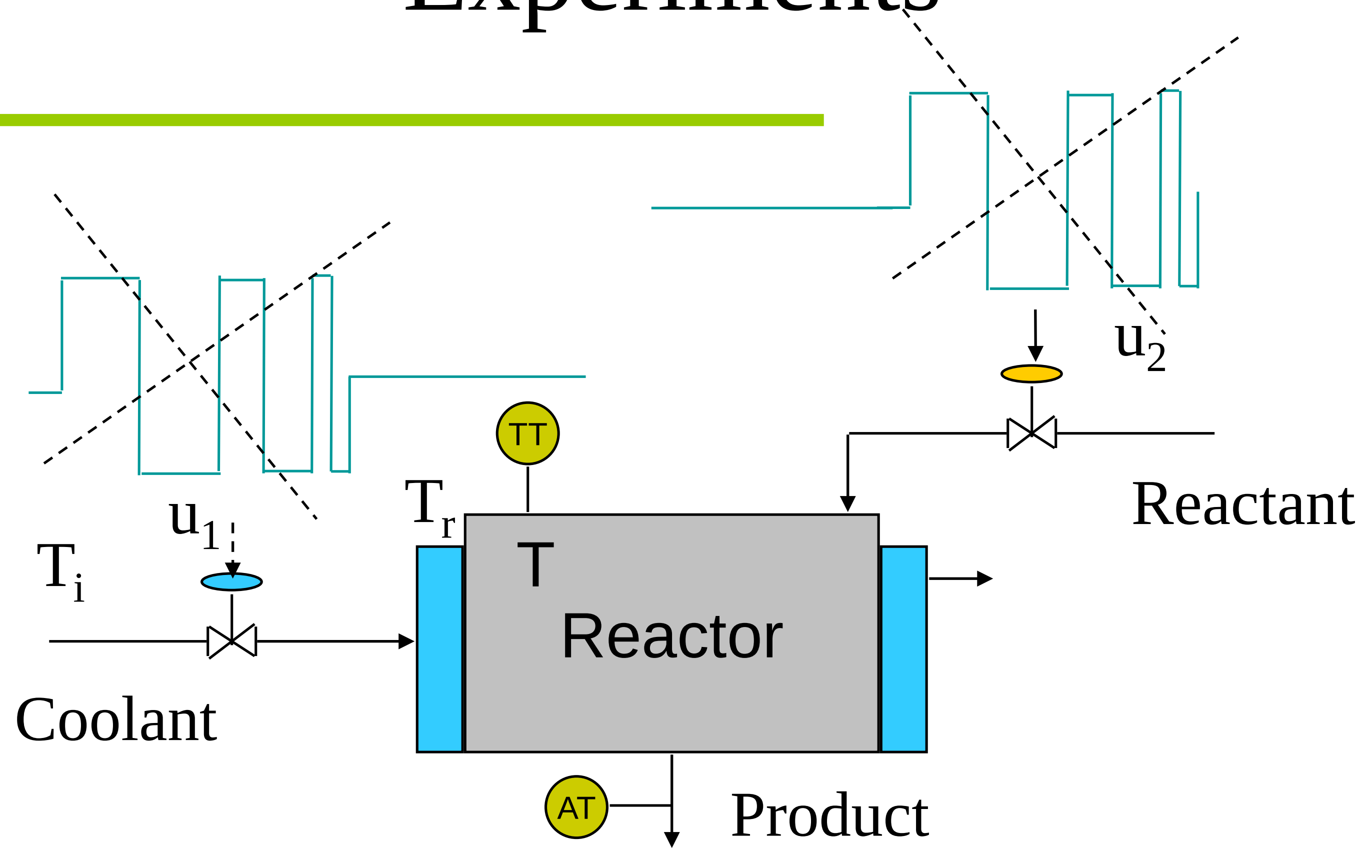
Pre-test can help planning the experiments





Typical test signal covering a range of frequencies

Experiments



Change one input while maintaining constant the others, and repeat the experiment agin with the remainder ones, it is not a good policy: it increases the time required for data collection and provides data in special operating conditions.

Solving parameterization problems

$$\min_{p} J = \sum_{i=1}^{N} \left[y(p, u, t_i) - y_p(t_i) \right]^2$$

$$\dot{x}(t) = f(x(t), u(t), p) \qquad y(t) = g(x(t), u(t), p)$$

$$\underline{p} \le p \le \overline{p}$$

✓Selection of cost function J

✓Numerical solution

- Sequential approach
- Simultaneous approach

✓Initialization of states

Simultaneous approach: Discretization

One important problem associated with the simultaneous approach is the discretization of the differential equations

$$\min_{u(t),x_0,t_f} \quad J(\mathbf{u}) = \int_{t_0}^{t_f} C(\mathbf{x},\mathbf{u})dt$$

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x},\mathbf{u},\mathbf{z}), \quad \mathbf{x}(t_0) = \mathbf{x}_0$$

$$\mathbf{h}(\mathbf{x},\mathbf{u},\mathbf{z}) = \mathbf{0}$$

$$\mathbf{g}(\mathbf{x},\mathbf{u},\mathbf{z}) \leq \mathbf{0}$$

Simple methods, such as the Euler discretization are not robust and lead to numerical problems with stiff systems

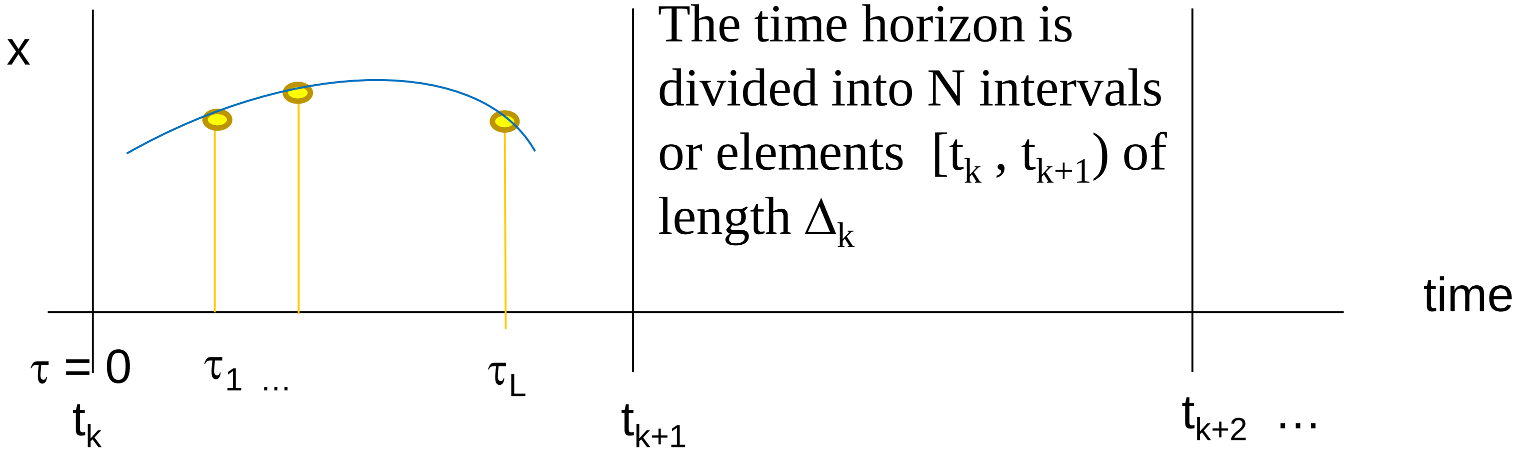
$$\frac{d\mathbf{x}}{dt} \approx \frac{\mathbf{x}(t+\Delta_t) - \mathbf{x}(t)}{\Delta_t} = \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\Delta_t}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{z}_k)\Delta_t$$

Other methods such as higher order implicit integration ones or collocation methods should be used

Collocation on finite elements

The time horizon is divided into N intervals or elements $[t_k , t_{k+1})$ of length Δ_k



The time evolution of the variables is approximated by polynomial interpolation on the values of the variable on L+1 collocation points located at fixed positions τ_j in every element k. Different methods exist. Using Lagrange polynomials, $x(t) = x_{kj}$

$$\mathbf{x}(t) \approx \sum_{j=0}^{L} P_j(\tau)\mathbf{x}_{kj}$$

$$t = t_k + \tau\Delta_k \qquad \tau \in [0,1)$$

$$\dot{\mathbf{x}}(t) \approx \sum_{j=0}^{L} \frac{\dot{P}_j(\tau)\mathbf{x}_{kj}}{\Delta_k}$$

Simultaneous approach

$$\min_{\mathbf{x}_k,\mathbf{u}_k,\mathbf{z}_k} J = \sum_{j=1}^{N} C(\mathbf{x}_k,\mathbf{u}_k)\Delta_k$$

$$F(\mathbf{x}_{k+1},\mathbf{x}_{k-i},\mathbf{u}_k,\mathbf{z}_k) = 0$$

$$\mathbf{h}(\mathbf{x}_k,\mathbf{u}_k,\mathbf{z}_k) = 0$$

$$\mathbf{g}(\mathbf{x}_k,\mathbf{u}_k,\mathbf{z}_k) \le 0$$

$$k = 0,1,2,\ldots.N$$

$$\min_{\mathbf{x}_k,\mathbf{u}_k,\mathbf{z}_k} J = \sum_{j=1}^{N} C(\mathbf{x}_k,\mathbf{u}_k)\Delta_k$$

$$F(\mathbf{x}_1,\mathbf{x}_0,\mathbf{u}_0,\mathbf{z}_0) = 0$$

$$F(\mathbf{x}_2,\mathbf{x}_1,\mathbf{u}_1,\mathbf{z}_1) = 0$$

$$\ldots.$$

$$h(\mathbf{x}_n,\mathbf{u}_n,\mathbf{z}_n) = 0$$

$$\ldots..$$

The number of equations increases by a factor of N and the number of decision variables increases from the CVP of u to u_k, x_k, z_k with respect to the sequential approach

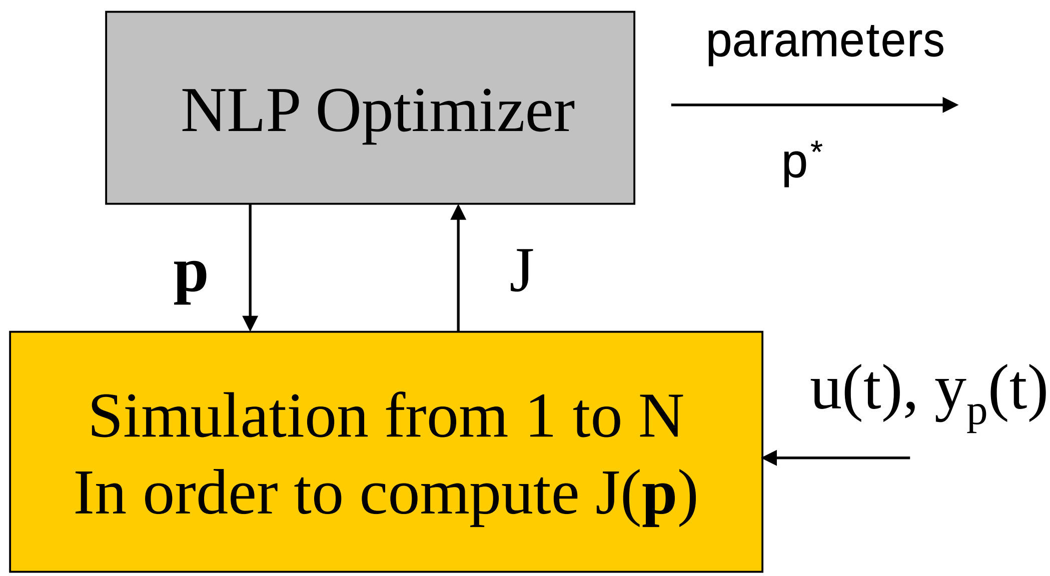
But it is easier to impose constraints on the time evolution of the states and algebraic variables (path constraints) by limiting , x_k, z_k at the discretization points

DO: Sequential approach

$$\min_{p} J = \min_{p} \sum_{i=1}^{N} \left[y(u,p,t_i) - y_p(t_i) \right]^2$$

$$\dot{x}(t) = f(x(t), u(t), p) \qquad y(t) = g(x(t), u(t), p)$$

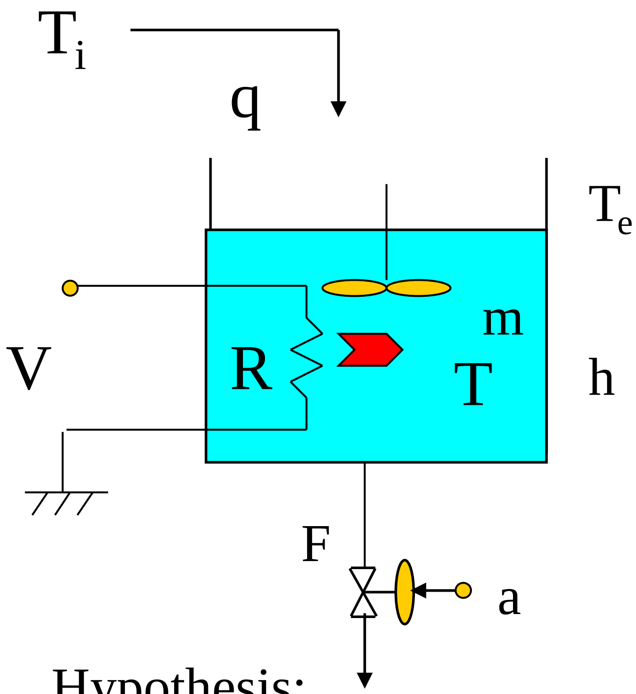
$$\underline{p} \le p \le \overline{p}$$



Parameterization / Methodology

- ✓ Experiments (data recorded for calibration and validation)
- ✓ Data analysis and filtering
- ✓ Choice of parameters to be identified
  - – Structural identifiability
  - – Sensitivities computation
- ✓ Optional model re-parameterization to avoid co-linearities
- ✓ Initial estimates and possible ranges of the parameters
- ✓ Select the cost function
- ✓ Estimate the parameters by optimization
- ✓ Validate the model. Estimate residuals and confident bands for the parameter.

Example: Heated tank



Hypothesis:

T lumped temperature
ρ constant density
c_e constant specific heat

T temperature
m tank mass
H liquid level
V voltage
q Inflow
F Outflow
a valve opening
H entalphy
c_e specific heat
A tank cross section
ρ density
R resistence
T_e external temperature
T_i inflow temperature

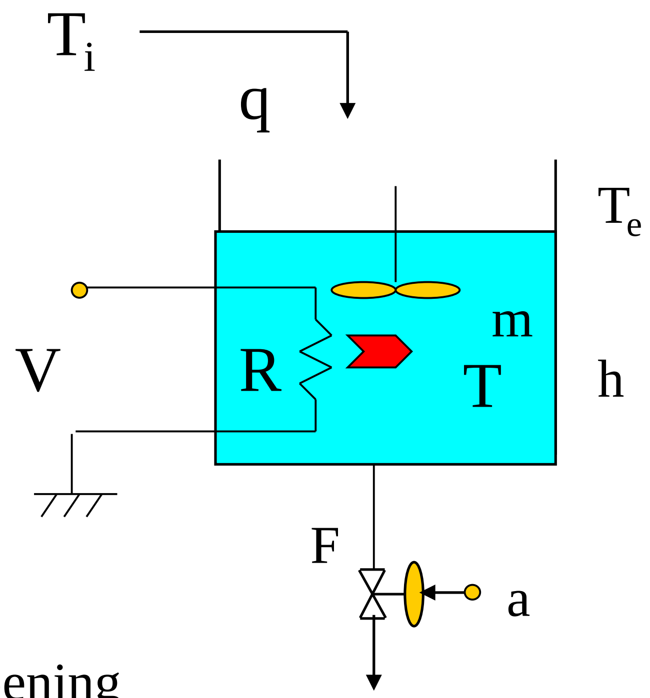
Dynamic model

State space non-linear model

$$\frac{dh}{dt} = \frac{1}{A}(q - F)$$

$$\frac{dT}{dt} = \frac{q}{Ah}(T_i - T) + \frac{V^2}{AhR\rho c_e} - \frac{U_{amb}}{Ah\rho c_e}(T - T_e)$$

$$F = ak\sqrt{h}$$



CV: T temperature MV: V voltage

h level a valve opening

Disturbances: q Inflow, T_i Input temperature

States: h , T

Other variables: F outflow

Model calibration

$$\frac{dh}{dt} = \frac{1}{A}(q - F)$$

$$\frac{dT}{dt} = \frac{q}{Ah}(T_i - T) + \frac{V^2}{AhR\rho c_e} - \frac{U_{amb}}{Ah\rho c_e}(T - T_e)$$

$$F = ak\sqrt{h}$$

Unknown parameters to be estimated

-- k friction factor

-- U_{amb} Heat transfer coefficient to ambient

-- A tank cross section

-- R electrical resistance

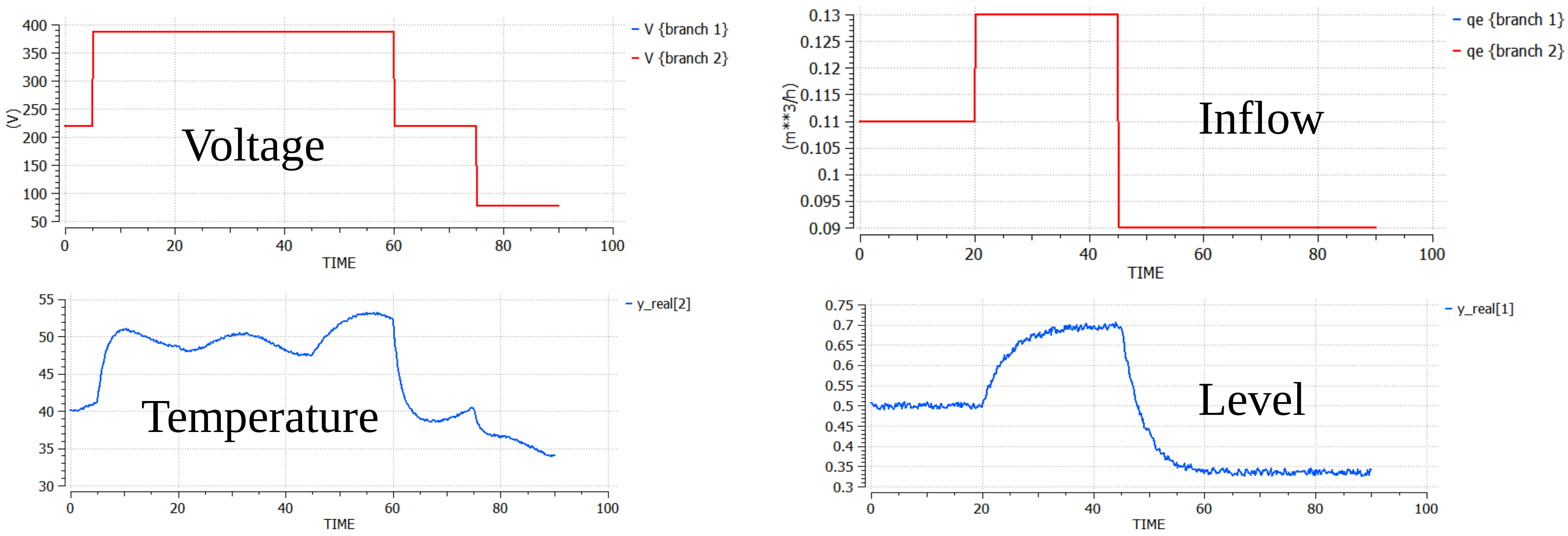
$$\min_p J = \min_p \sum_{i=1}^{N} \left[y(u, p, t_i) - y_p(t_i)\right]^2$$

$$\dot{x}(t) = f(x(t), u(t), p) \qquad y(t) = g(x(t), u(t), p)$$

$$\underline{p} \le p \le \overline{p}$$

In order to formulate the identification problem, experimental data are required

Experimental data



One experiment was performed where the inflow q and voltage V to the resistance were changed over time and the values of the liquid level and temperature in the tank were recorded

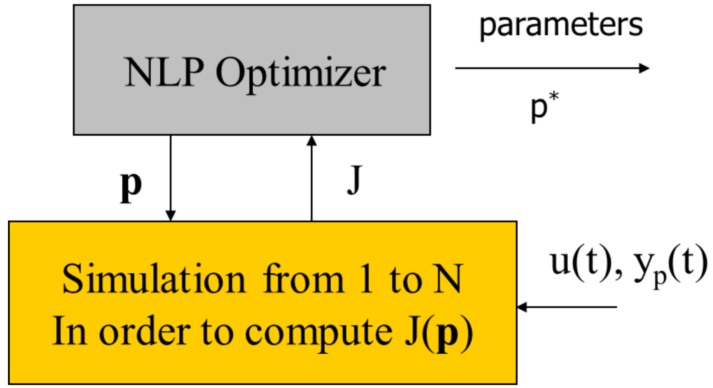
Disturbances in T_i were not recorded in the data set

Model calibration

$$\min_{p} J = \min_{p} \sum_{i=1}^{N} \left[T(u,p,t_i) - T_{exp}(t_i)\right]^2 + \sum_{i=1}^{N} \left[h(u,p,t_i) - h_{exp}(t_i)\right]^2$$

$$p = [k, U_{amb}, A, R]$$

$$\underline{p} \le p \le \overline{p}$$



$$\frac{dh}{dt} = \frac{1}{A}(q - F)$$

$$\frac{dT}{dt} = \frac{q}{Ah}(T_i - T) + \frac{V^2}{AhR\rho c_e} - \frac{U_{amb}}{Ah\rho c_e}(T - T_e)$$

$$F = ak\sqrt{h}$$

DO in EcosimPro

COMPONENT

PROCESS MODEL

$$\frac{dx(t)}{dt} = f(x(t), z)$$
$$y(t) = g(x(t), z)$$

CONSTRAINTS

$$g_1(x(t), z) \le 0$$
$$g_2(x(t), z) \le 0$$
$$\vdots$$

COST FUNCTION

$$J(z)$$

Dynanic simulator

EXPERIMENT

Constraints in the decision variables z

$$L_1 \le z_1 \le U_1$$
$$L_2 \le z_2 \le U_2$$
$$\vdots$$

Decision variables : z_1, z_2, ...

Value of the cost function *J* and constraints g

OPTIMIZATION ALGORITHM

NLP

z^*

The optimization algorithm obtains the values of the cost function J and constraints g when it needs them by calling the dynamic simulation module

Component / Variable

```
USE MATH
USE ESNOPT

COMPONENT estim_param_deposito (INTEGER nent =2, INTEGER nsal = 2)

   DATA
         REAL rho = 1000          "densidad del líquido (kg/m3)"
         REAL cp = 4168           "calor específico (J/kgºC)"
         REAL Tamb = 20           "temperatura del ambiente exterior (ºC)"
         REAL Te = 40             "temperatura de la corriente de entrada (ºC)"
         REAL a  = 100            "% de apertura de la valvula de descarga (%)"

 -- Parámetros a estimar sin escalar, aqui se dan unos valores inicial
         REAL k = 0.18            "factor de fricción (m 2.5/h)"
         REAL Uamb = 62           "coeficiente de pérdida de calor al ambiente (W/ºC)"
         REAL A =  0.47           "área del depósito (m2)"
         REAL R  = 52             "resistencia eléctrica (ohmios)"
```

Component name

Decision Variables

Component / Variables

```
…..
REAL tsamp = 0.2            -- Periodo de muestreo de recogida de datos (h)

INTEGER N1 = 1              -- Inicio horizonte de estimación
INTEGER NE = 451            -- Número de muestras, datos reales tomados
-- Pesos en el coste para los errores entre predicciones y datos reales
REAL pesos[2] = {1.0, 100.0}                    -- pesos[nsal]
-- Valores medios de los datos experimentales, factor de escala
REAL media[2] = {0.48, 45.4}               -- Nivel, Temperatura

REAL Liminfh = 0.3           -- Limites inferior y superior del nivel y temp
REAL Limsuph = 1
REAL LiminfT = 20
REAL LimsupT = 80
```

Additional parameters

Component / Variables

```
DECLS
        -- Variables manipuladas
        REAL V            "voltaje aplicado a la resistencia eléctrica (V)"
        REAL qe           "caudal de entrada al depósito (m**3/h)"
        -- Salidas medidas del proceso
        REAL h             "nivel de líquido en el depósito (m)"
        REAL T            "temperatura del depósito (ºC)"
        REAL F            "caudal de salida (m3/h)"

        REAL y_modelo[nsal]     -- Salidas del modelo que se quiere ajustar
        REAL y_real[nsal]       -- Salidas muestreadas experimentales
        REAL u_real[nent]       -- Entradas experimentales
```

Additional variables

Component / Variables

```
REAL coef = 0                  -- Coeficiente que activa la función de coste
DISCR REAL J1                  -- Subtotal del índice de coste
DISCR REAL J2                  -- Subtotal del índice de coste
DISCR REAL J_costo             -- La function de costo que se minimiza

BOOLEAN Sample = TRUE          -- Variable bucles
-- Tablas correspondientes a los datos experimentales
 TABLE_1D      tab1, tab2, tab3, tab4
-- Vector con la funcion objetivo y las restricciones no lineales
REAL F_optim[5]
-- Variable auxiliar para el calculo de maximos y minimos de nivel
REAL hmin, hmax
-- Variable auxiliar para el calculo de maximos y minimos de temperatura
REAL Tmin, Tmax
```

Additional variables

Component / INIT

INIT

```
J_costo = 0.0
J1 = 0.0
J2 = 0.0

-- Entradas
readTableCols1D ("datos_potencia.txt",1,2,tab1)
u_real[1] = linearInterp1D (tab1, TIME)
V = u_real[1]                    -- valor inicial para el modelo

readTableCols1D ("datos_caudal_entrada.txt",1,2,tab2)
 u_real[2] = linearInterp1D (tab2, TIME)
qe = u_real[2]                   -- valor inicial para el modelo
 ……
```

Reading data from a file to a table

Interpolation from a data table

Component / INIT

```
-- Salidas
readTableCols1D ("datos_nivel.txt",1,2,tab3)
y_real[1] = linearInterp1D (tab3, TIME)        -- valor real, leído de planta
h = y_real[1]                                  -- valor inicial del modelo
y_modelo[1] = y_real[1]                        -- valor inicial del modelo

readTableCols1D ("datos_temperatura.txt",1,2,tab4)
y_real[2] = linearInterp1D (tab4, TIME)        -- valor real, leído de planta
T = y_real[2]                                  -- valor inicial del modelo
y_modelo[2] = y_real[2]                        -- valor inicial del modelo

coef = 1    -- coeficiente que activa el índice de coste = 1 AFTER N1*tsamp
hmin = h                           --  inicializacion de variables auxiliares
hmax = h
Tmin = T
Tmax = T
```

Component / Cost function

```
        -- calculo de la funcion de costo a minimizar
DISCRETE

        WHEN (Sample) THEN

                -- Subtotales del índice de coste
                J1 += coef*((y_modelo[1] - y_real[1])/media[1])**2
                J2 += coef*((y_modelo[2] - y_real[2])/media[2])**2
                J_costo = pesos[1]*J1 + pesos[2]*J2

                Sample = FALSE
                Sample = TRUE AFTER tsamp
        END WHEN
```

Component / Model

CONTINUOUS

```
-- Entradas medibles
u_real[1] = linearInterp1D (tab1, TIME)
V = u_real[1]                          -- Voltaje aplicado a la resistencia
u_real[2] = linearInterp1D (tab2, TIME)
qe = u_real[2]                         -- Caudal de entrada
-- Salidas medibles
y_real[1] = linearInterp1D (tab3, TIME)        -- Nivel
y_real[2] = linearInterp1D (tab4, TIME)        -- Temperatura

-- Balance de materia
A*h' = qe - F
F = a*k*sqrt(h)/100
-- Balance de energía
A*h*rho*cp*T' = qe*rho*cp*(Te-T)  + (3600*V*V/R) - 3600*Uamb*(T-Tamb)
```

Interpolation from data tables

Dynamic model

Component / J, g

```
-- calcular las restricciones no lineales (expresiones  c(x) <= 0)

  F_optim[1] = J_costo
  F_optim[2] = ESNOPT.minabs(h,hmin)
  F_optim[3] = ESNOPT.maxabs(h,hmax)
  F_optim[4] = ESNOPT.minabs(T,Tmin)
  F_optim[5] = ESNOPT.maxabs(T,Tmax)

-- Asignación de valores del modelo
   y_modelo[1] = h
   y_modelo[2] = T

END COMPONENT
```

Cost function

Path constraints
Max(T(t)) < LimsupT

Experiment /Functions

```
USE OPTIM_METHODS

FUNCTION INTEGER coste_y_restricciones (IN REAL esnopt_x[], IN
INTEGER needF, OUT REAL esnopt_F[], IN INTEGER explicit_derivatives,
IN INTEGER needG, OUT REAL esnopt_G[])

  …..
END FUNCTION

EXPERIMENT ajuste ON estim_param_deposito.param

   …..
```

Experiment / Variables

```
EXPERIMENT ajuste ON estim_param_deposito.param
  DECLS
        INTEGER n_dec_vars = 4          --  numero de variables de decision
        INTEGER n_constraints = 4       --  numero restricciones
        INTEGER n_total            -- numero restricciones + una funcion objetivo
        REAL param_estim[4]      -- variables de decision, size n_dec_var
 REAL xlow[4]    -- valor inferior de las variables de decision, size n_dec_var
 REAL xupp[4]  -- valor superior de las variables de decision, size n_dec_var
 REAL Flow[5]    -- valor inferior de la funcion objetivo y las restricciones, size n_total
 REAL Fupp[5]    -- valor superior de la funcion objetivo y las restricciones, size n_total

  BOUNDS
    -- Set equations for boundaries: boundVar = f(TIME;...)
   coef = 1
```

Experiment / variables

BODY

-- Inicializaciones

k = 0.180 -- correct value 0.1556

Uamb = 62 -- correct value 40

A = 0.470 -- correct value 0.45

R = 52 -- correct value 60

n_total = n_constraints + 1

-- Formar vector de variables de decisión

param_estim[1] = k

param_estim[2] = Uamb

param_estim[3] = A

param_estim[4] = R

Specifying the decision variables and their range

Experiment

-- Limites de las variables de decision

-- k

xlow[1] = 0.077

xupp[1] = 0.24

-- Uamb

xlow[2] = 20

xupp[2] = 63

-- A

xlow[3] = 0.22

xupp[3] = 0.67

-- R

xlow[4] = 30

xupp[4] = 90

Specifying the decision variables and their range

Experiment

Flow[1] = -1.0e6
Fupp[1] = 1.0e6

Flow[2] = Liminfh
Fupp[2] = Limsuph

Flow[3] = Liminfh
Fupp[3] = Limsuph

Flow[4] = LiminfT
Fupp[4] = LimsupT

Flow[5] = LiminfT
Fupp[5] = LimsupT

Specifying constraints range

Experiment / SNOPT

```
                    --Optimization extern routine call
setSilentMode(TRUE)
SET_REPORT_ACTIVE("#MONITOR",FALSE)
esnopt_init (n_dec_vars, n_constraints)
esnopt_set_variables_bounds_and_initial_values (xlow, xupp, param_estim)
esnopt_set_constraints_bounds_and_initial_values (Flow, Fupp, F_optim)
esnopt_set_cost_function_and_constraints (coste_y_restricciones)
esnopt_set_explicit_derivatives (0)
esnopt_set_function_precision (1.0e-6)
REL_ERROR = 1.0e-7
                              -- for STEADY Calls  TOLERANCE = 1.0e-7
esnopt_set_iterations_limit (200)
esnopt ()
setSilentMode(FALSE)
SET_REPORT_ACTIVE("#MONITOR",TRUE)
RESET_VARIABLES ()
```

Calling the optimizer

Experiment /SNOPT

```
esnopt_get_variables_values(dec_var)
esnopt_free ()
esnopt_get_variables_values(param_estim)
esnopt_free ()

k =        param_estim[1]
Uamb =  param_estim[2]
A =        param_estim[3]
R =        param_estim[4]
 SET_INIT_ACTIVE(TRUE)
 TIME = 0
  TSTOP = 90
  CINT = 0.2
  IMETHOD = DASSL
  INTEG()
END EXPERIMENT
```

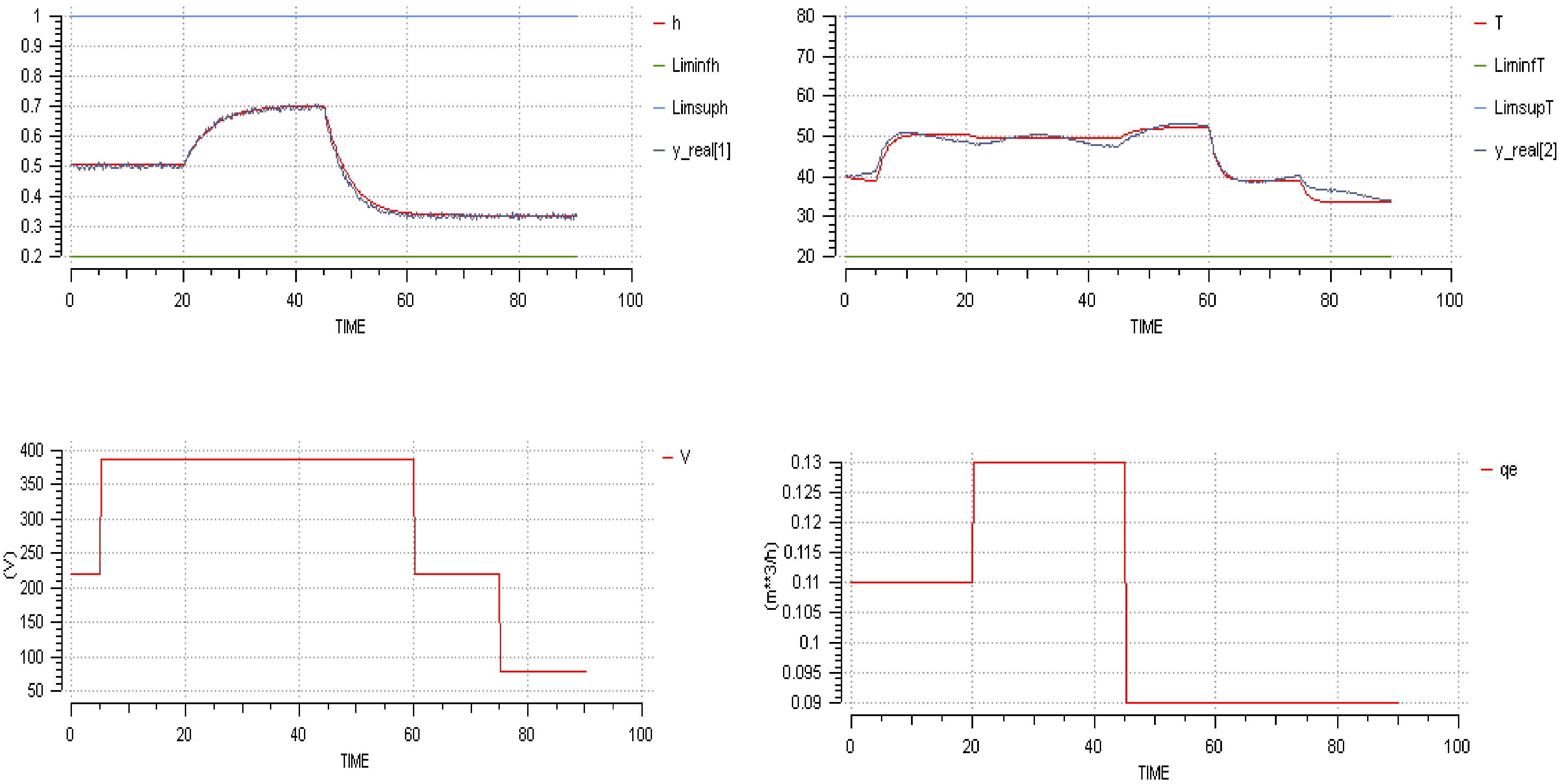
Getting the solution

Visualizing the solution

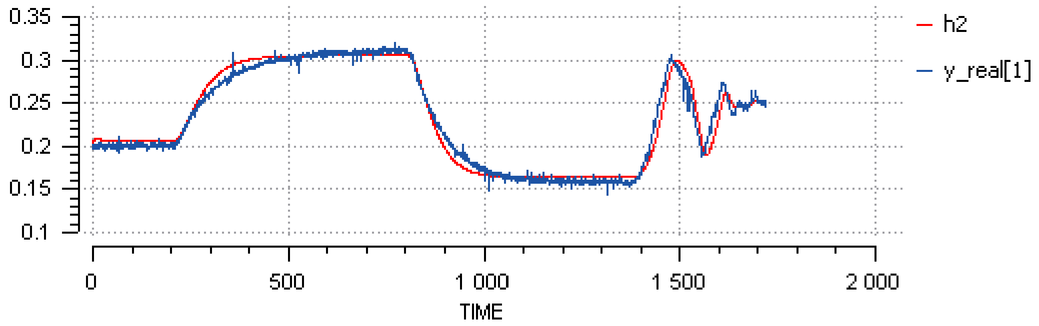
Call to the simulation

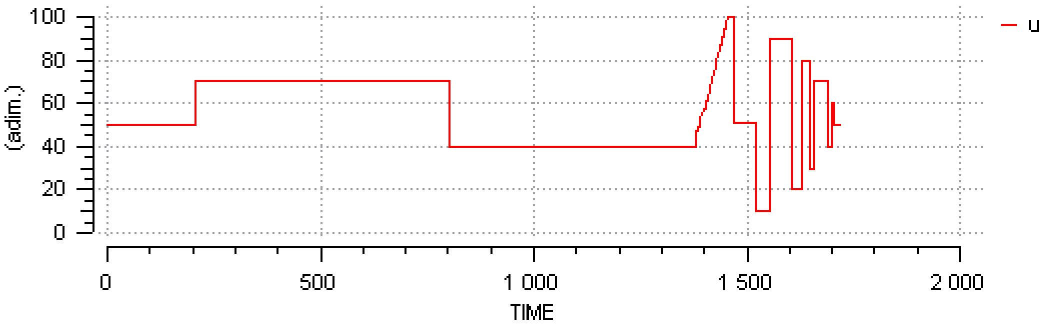
```
FUNCTION INTEGER coste_y_restricciones (IN REAL esnopt_x[], , IN
INTEGER needF, OUT REAL esnopt_F[], IN INTEGER explicit_derivatives, IN
INTEGER needG, OUT REAL esnopt_G[])
   BODY
         RESET_VARIABLES ()
         k        = esnopt_x[1]
         Uamb     = esnopt_x[2]
         A        = esnopt_x[3]
         R        = esnopt_x[4]
          SET_INIT_ACTIVE(TRUE)
         TIME = 0
         TSTOP = 90
         CINT = 0.2
         INTEG()

         esnopt_F[1] = F_optim[1]
         esnopt_F[2] = F_optim[2]
         esnopt_F[3] = F_optim[3]
         esnopt_F[4] = F_optim[4]
         esnopt_F[5] = F_optim[5]

         RETURN 0
     END FUNCTION
```

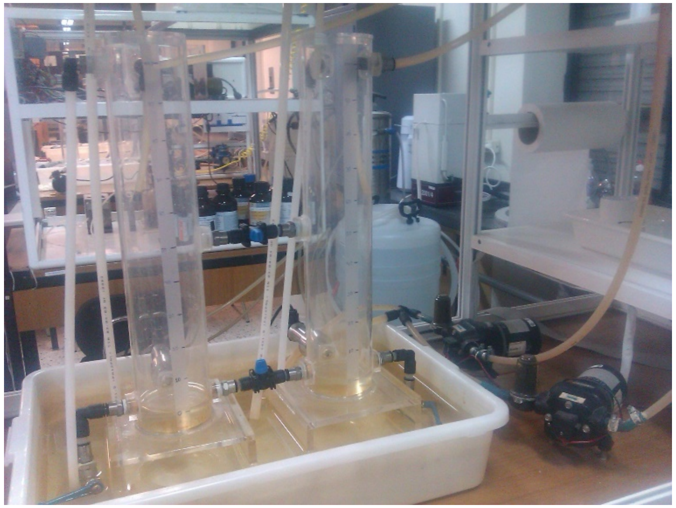
Calibration



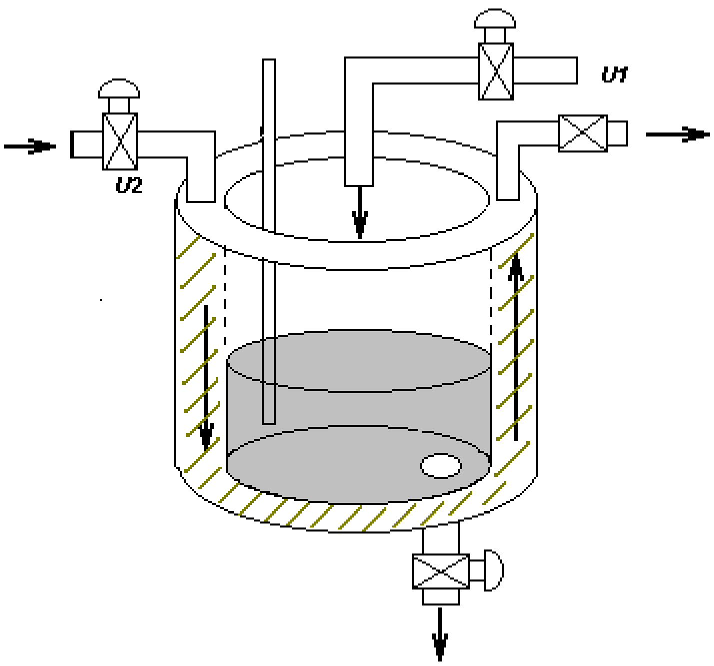
Two tank system







Example in EcosimPro: Van der Vusse Reactor



Highly non-linear reactor, difficult to control

$$A \rightarrow B \rightarrow C$$

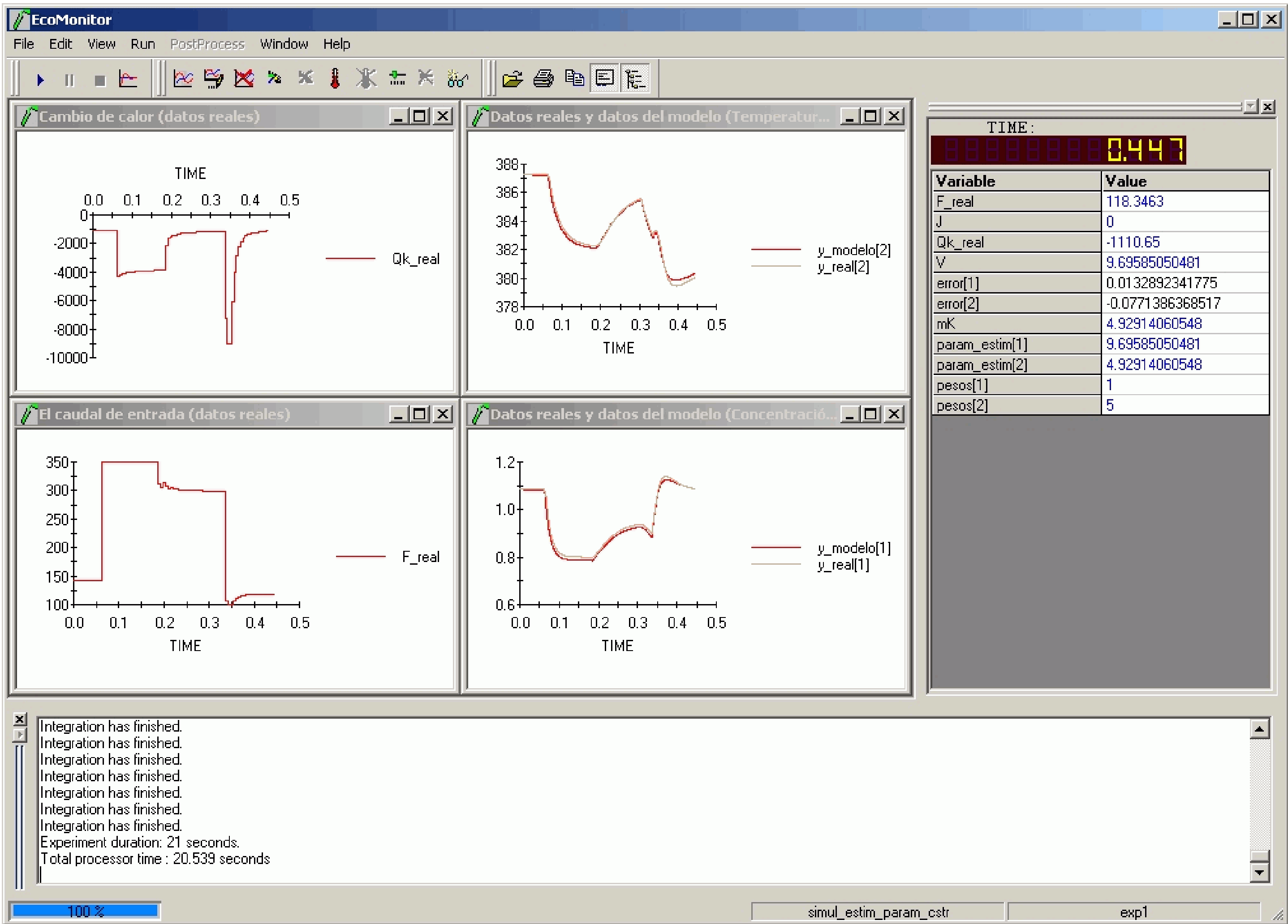
$$2A \rightarrow D$$

Parameters:

Volume: 10 l

Refrigerant mass: 5 Kg

Reactor Van der Vusse



Model validation

- Validating a model consists of implementing several test, so that, if the model responds adequately to them, a certain trust on its soundness for the aims it was designed for is obtained.
- There is no “prove” of model validity, but a certain degree of confidence in the model based on results of the tests.
- Model validity can be lost because of a single negative result in a test.

Validation

- The set of tests with positive and negative results allow to fix the range of validity of the model
- Different situations must be treated with different validation methods:
  - The process exists and a model must be built reproducing its behaviour
  - The process does not exists yet and the purpose of the model is to design it or predict future behaviour and optimize it.

Validation

- Set of tests of different nature
- Validation should be made at all model building levels:
  - Hypothesis
  - Formulation
  - Software coding
  - Numerical methods (verification)
  - Application

Validation levels

Validity range

Data recording / Reconciliation

How much sensitive is model to parameter changes?

Is the code robust, fast,...?

Does it provide sensible results?

Hypothesis ↓ **Modelling** ↓ **Parameterization** ↓ **Software coding** ↓ **Experimentation**

Does the model use fit the hypothesis?

Does the model reflect the key process aspects?

Which is the quality of the data?

Is the model well coded?

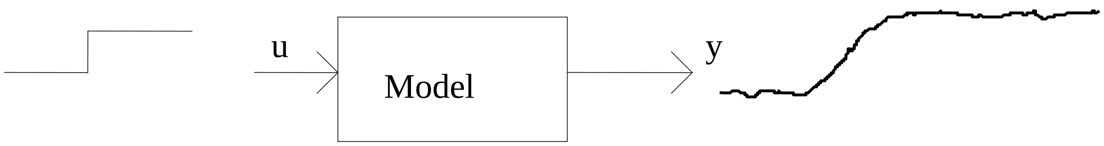
Do results match the reality?

Validation methods

- ✓ Qualitative model responses
- ✓ Experimental data fit
- ✓ Statistical tests
- ✓ Sensitivity analysis
- ✓ Prediction capabilities
- ✓ Parameter distortion
- ✓ Coherence between different models
- ✓ …

Qualitative model responses

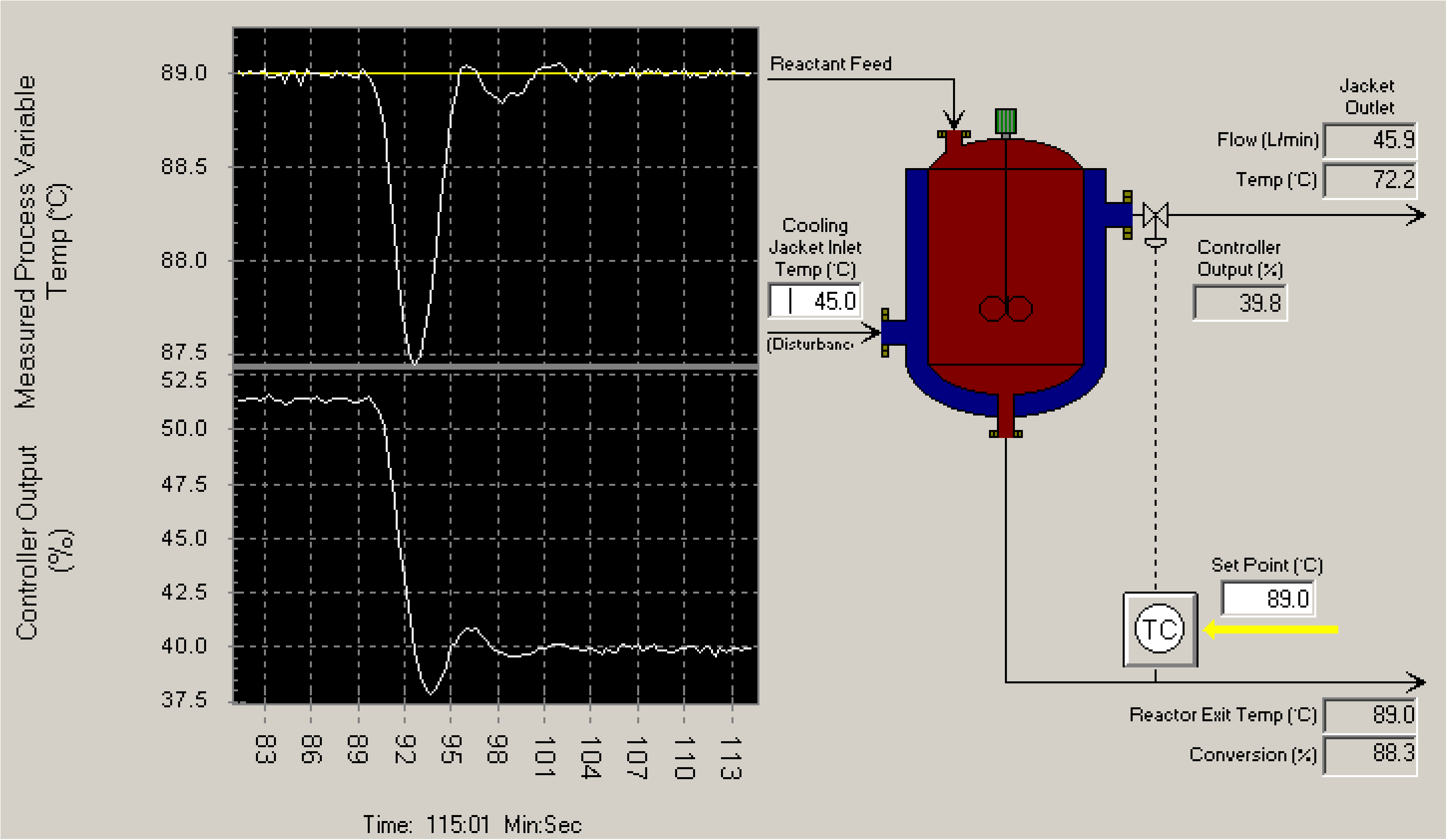
Evaluation of the model responses to standard inputs



Evaluation of the model responses with domain experts.

Qualitative model responses

Example of the model response to a step change in the cooling jacket input temperature to be analysed with domain experts

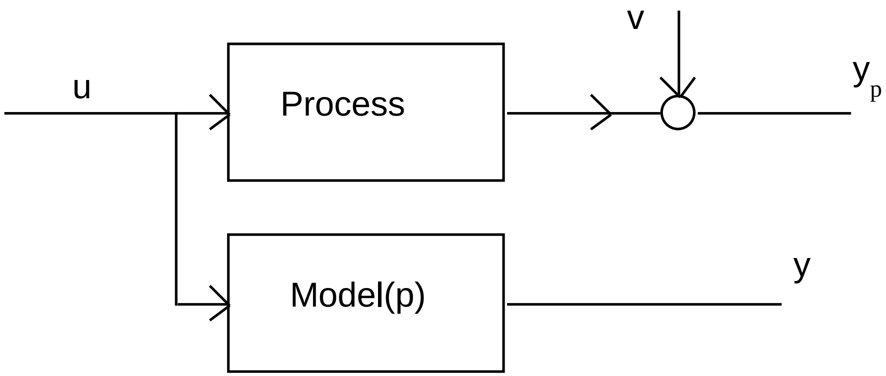


Qualitative model responses

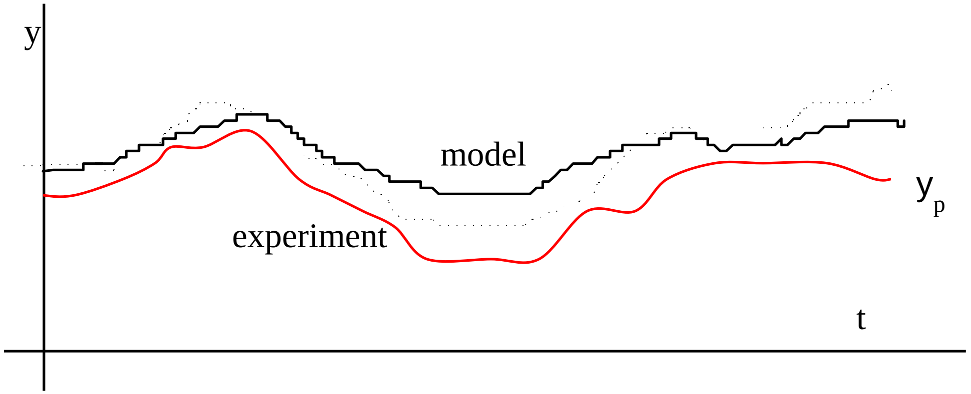
- Turing Test: Is it possible to distinguish between the model and process responses to the same input if they are presented in the same format?
- Examine the time evolution of variables such as flows through different components

Use of experimental data

Compare the process and model responses to the same inputs with data sets different to the ones used in the model parameterization



$$\frac{1}{N}\sum_{i=1}^{N}\left(y(t_i)-y_p(t_i)\right)^2$$



Errors due to:

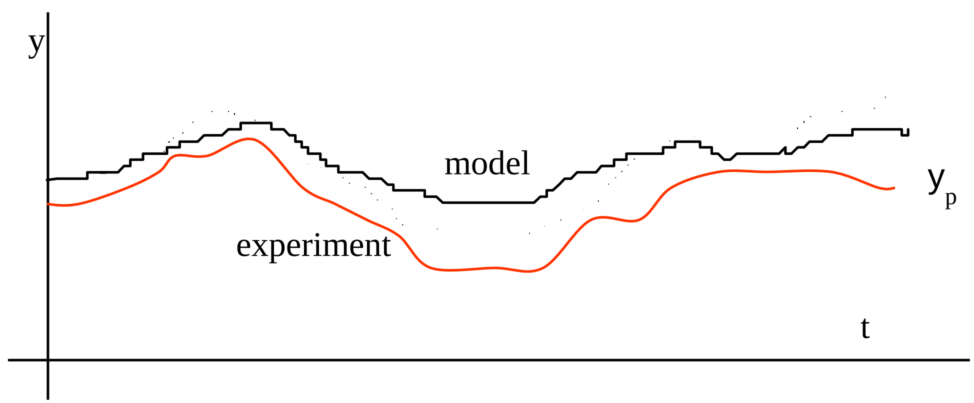
- ✓ Initial conditions
- ✓ Disturbances
- ✓ Model

Error indexes

Numerical measure of the matching between model responses y(t) and experimental data $y_p(t)$

Can be used to select the best model among several candidates to represent a process

$$\frac{1}{N}\sum_{i=1}^{N}\left(y(t_i) - y_p(t_i)\right)^2$$



Error indexes

Most of the indexes combine sum of errors and number of parameters in the model

Final Prediction Error FPE

$$\frac{1}{N}\left(\frac{1+d/N}{1-d/N}\right)\sum_{i=1}^{N}(y(t_i)-y_p(t_i))^2$$

Estimates of the prediction error variance with the new data

N number of data

d number of parameters in the model

Akaike Information Criterion AIC

$$(1+\frac{2d}{N})\sum_{i=1}^{N}(y(t_i)-y_p(t_i))^2$$

$$N\log\left(\frac{\sum_{i=1}^{N}(y(t_i)-y_p(t_i))^2}{N}\right)+2d$$

Error indexes

Bayesian Information Criterion BIC

$$N \log\left(\frac{\sum_{i=1}^{N} (y(t_i) - y_p(t_i))^2}{N} \right) + d \log(N)$$

Consistent: Probability of selecting an incorrect model tends to zero with N

Rissamen's Minimal Description lenght

$$(1 + \frac{2d}{N} \log N) \sum_{i=1}^{N} (y(t_i) - y_p(t_i))^2$$

Gives more weight to model complexity

Error indexes

F Test: Model j (with more parameters) is significantly better than model i, with a confidence level α

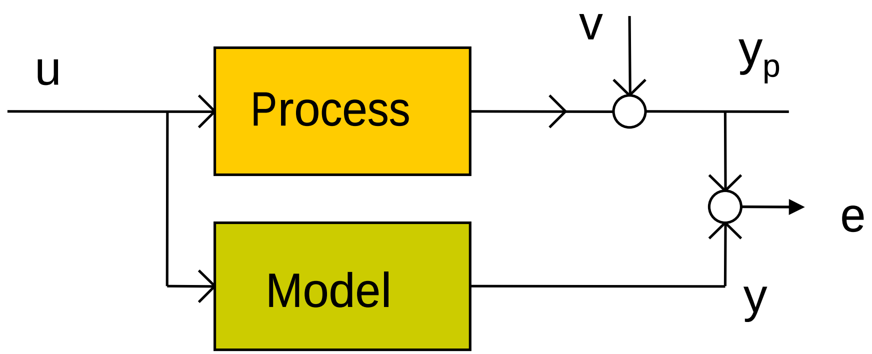
$$\frac{\left(\left[\sum_{i=1}^{N}(y(t_i)-y_p(t_i))^2\right]_{mod\,i}-\left[\sum_{i=1}^{N}(y(t_i)-y_p(t_i))^2\right]_{mod\,j}\right)/(d_j-d_i)}{\left[\sum_{i=1}^{N}(y(t_i)-y_p(t_i))^2\right]_{mod\,j}/(N-d_j)}$$

This statistical must be compared with the value provided by the F(d_j-d_i, N-d_j) distribution with a confidence level α

Data fitting errors

- Sources of data fitting errors are:
  - Measurements noise and disturbances. A characterization can be made measuring output variables while maintainig constant the process inputs.
  - Unknown initial conditions in the model. Its influence disappear after the initial transient.
  - Structural or parametric errors in the model. If the model were perfect, the residuals should have similar statistical characteristics as the output noise and disturbances.

Statistical test

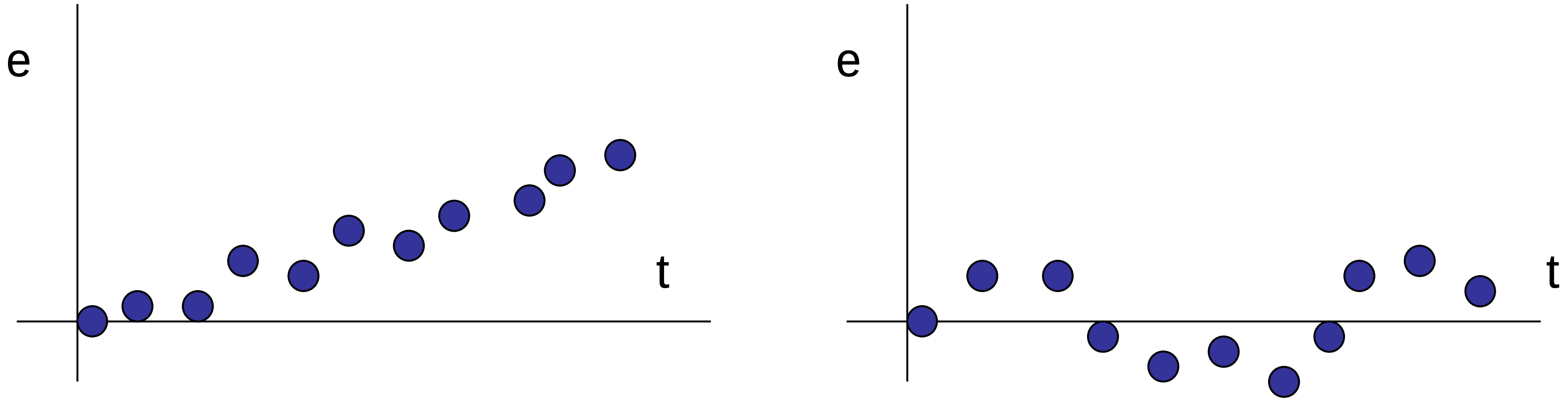


If the model is adequate, residuals should not show a "systematic" structure, but be the result of the stochastic noise and disturbances acting on the process.

Residuals $e(t) = y(t) - y_p(t)$

Statistical test

- Visual residual inspection



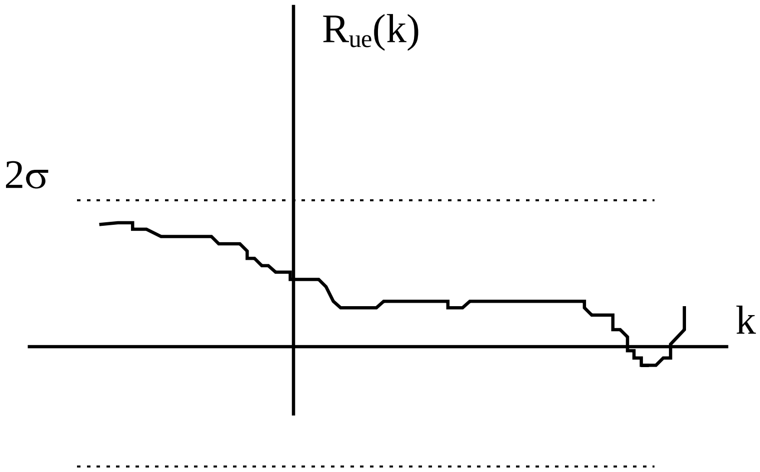
- Number of residual sign changes (ncs)
- Do they fit an AR model?

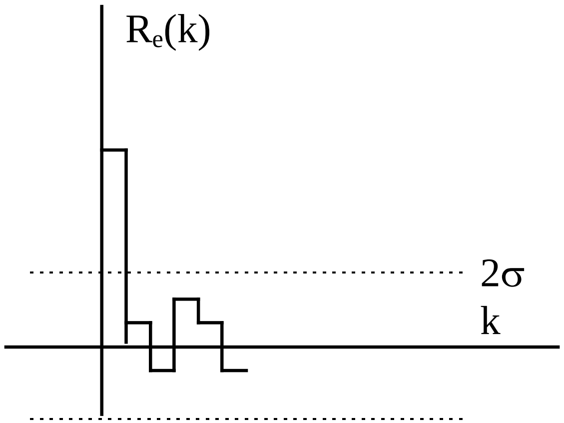
Test: $$\frac{ncs - \frac{N}{2}}{\sqrt{N/2}}$$

This statistic should follow a N(0,1) distribution if there is no bias in the number of sign changes

Statistical test

No correlation between input and residuals: With a good model, residuals represent the effect of noises and disturbances and they should be independent of a particular experiment given by u





Open loop, closed loop

Residuals whiteness

Parameter variance

Parameter confidence regions estimated in the model calibration provide a measure of its quality and the model goodness.

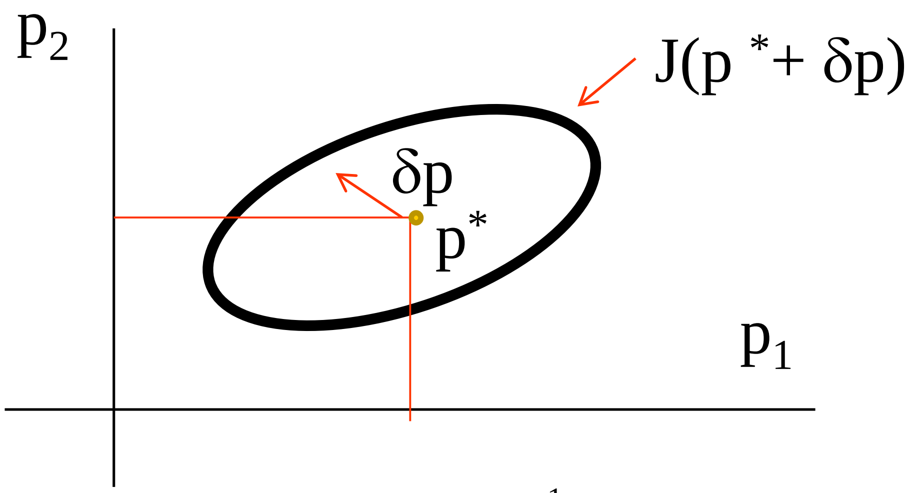
$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots \\ v_{21} & v_{22} & \cdots \\ \vdots & \vdots & \cdots \end{bmatrix}$$

Main diagonal elements of the covariance matrix obtained in the calibration step, give a measure of the confidence range of each parameter, while the off diagonal terms measure parameter independence, as ideally, they should be uncorrelated.

Parameter variance

A linear estimation of the change in the cost function is given by:

$$J(p^* + \delta p) = J(p^*) + \delta p' \sum_{t=1}^{N} \left[\left.\frac{\partial y_m}{\partial p}\right|_{p^*}' C^{-1} \left.\frac{\partial y_m}{\partial p}\right|_{p^*} \right] \delta p$$



This allows to draw the region on the parameter space that provides an error ΔJ

$$F^{-1} = \left[\sum_{t=1}^{N} \left(\frac{\partial y_m(t)}{\partial p} \right)' C^{-1} \left(\frac{\partial y_m(t)}{\partial p} \right) \right]^{-1}$$

The inverse of the Fisher matrix gives a lower bound of the error covariance matrix

Parameter variance

An better estimation of the error covariance matrix is given by:

$$C_e \approx \frac{2J(p^*)}{N-d}\left[\left.\frac{\partial^2 J}{\partial p \partial p'}\right|_{p^*}\right]^{-1}$$

With d the number of model parameters and N the number of data. Confidence intervals for the parameters are given by:

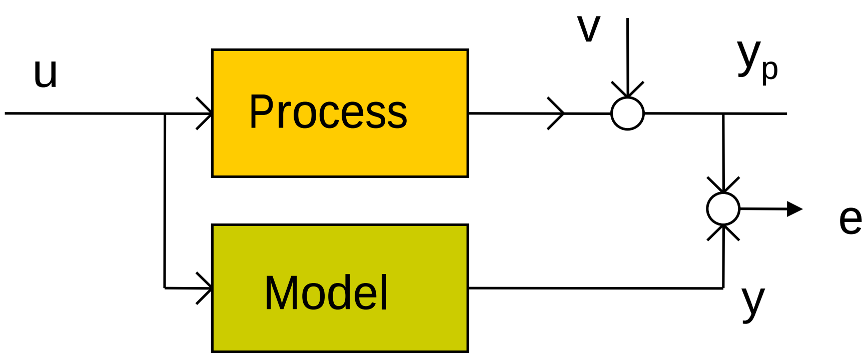
$p \pm 2.147\sqrt{c_{ii}}$ 90%

$p \pm 3.035\sqrt{c_{ii}}$ 99%

Care should be taken when considering multivariable effects

$p \pm t^{\alpha}_{N-d}\sqrt{c_{ii}}$ 100-α confidence degree. t Student dist.

Parameter variance



$$\operatorname{var}\left\{G(e^{jwt},\hat{\theta})\right\} \cong \frac{d}{N}\frac{\Phi_v(w)}{\Phi_u(w)}$$

In general terms, the variance of the estimated parameters increases with the number of parameters d in the model, and decreases with the ratio signal/noise and the number of data N used in the calibration

Sensitivity to parameters

How much J changes per unit change in a parameter p ?

How much a given output changes per unit change in a parameter p?

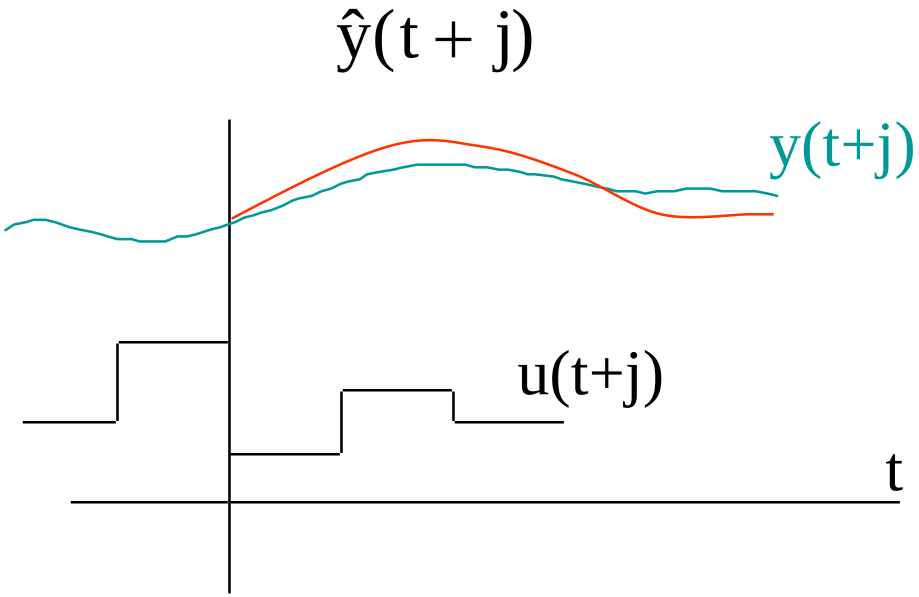
$$\min_{p} J = \sum_{i=1}^{N} \left[y(p, u, t_i) - y_p(t_i) \right]^2$$

$$\dot{x}(t) = f(x(t), u(t), p) \qquad y(t) = g(x(t), u(t), p)$$

$$\underline{p} \leq p \leq \overline{p}$$

They can be obtained from the sensitivities computed from integration of the extended system

Prediction capability



Future values can be predicted and compared with actual data. If the model is linear, explicit prediction formulas for DMC, GPC, etc. can be used.

Parameter distortion

$$\frac{d\,x(t)}{dt} = f(x(t), u(t), p, t)$$

$$y(t) = g(x, u(t), t)$$

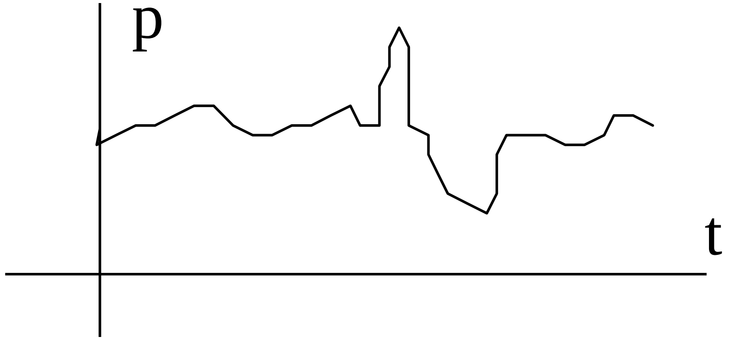
Calibration provides the value of the parameter set p^* (constant) giving the best fit to the experimental data

$$e^*(t) = y(p^*, t) - y_p(t)$$

Parameter distortion
Butterfield 1986

Assuming now that the set p could evolve over time,

Which would be the time evolution of model parameters p such that the model responses y would correspond exactly with the process ones for the same inputs?



Admissible distortions of parameters with physical meaning provide a measure of model credibility

$$\min_{\Delta \mathbf{p}} \quad y(t, \mathbf{p}^* + \Delta \mathbf{p}(t)) = y_p(t) \quad t = 1, \ldots, n$$