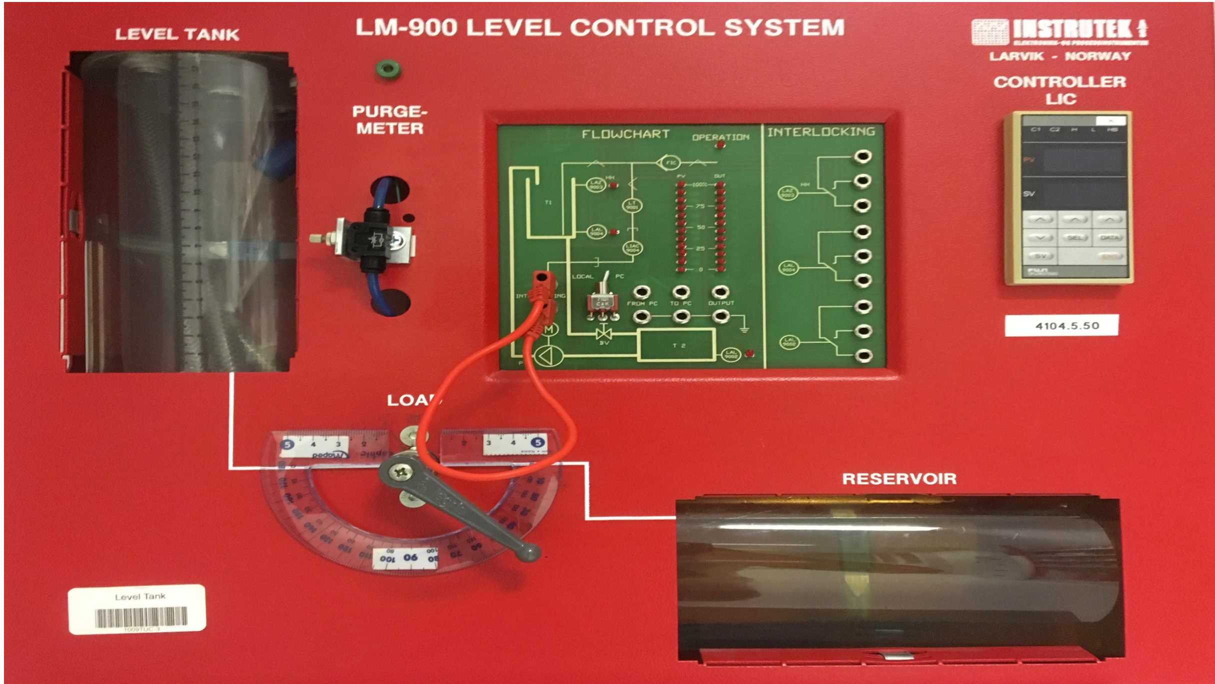
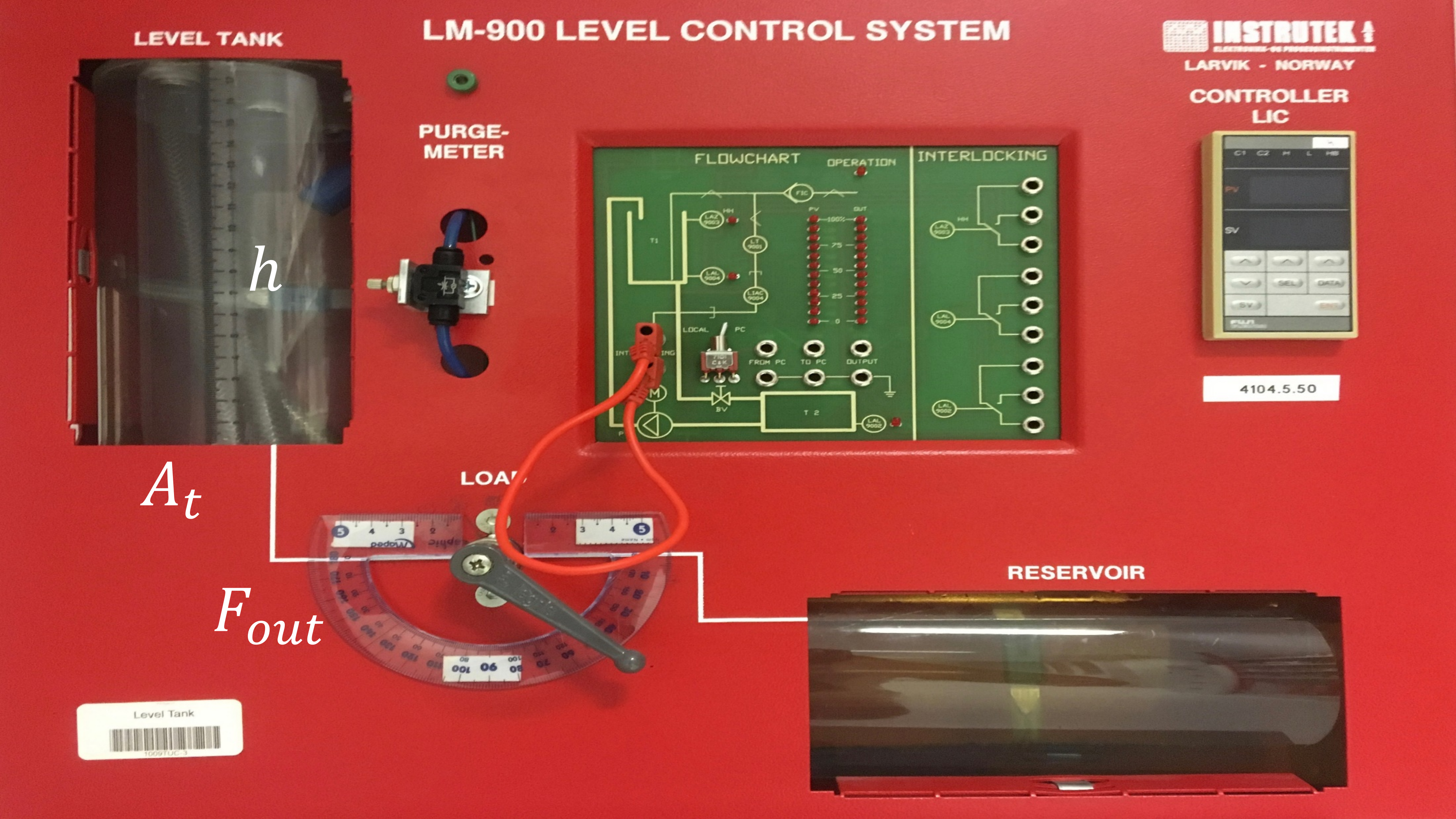
Level Tank Overview

Hans-Petter Halvorsen

Introduction

The Level Tank is a small-scale Laboratory Process




LEVEL TANK
LM-900 LEVEL CONTROL SYSTEM
INSTRUTEK
LARVIK - NORWAY
CONTROLLER
LIC
PURGE-
METER
FLOWCHART
OPERATION
INTERLOCKING
LOCAL
PC
FROM PC
TO PC
OUTPUT
T 2
PV
SV
4104.5.50
h
A_t
F_{out}
RESERVOIR
Level Tank

Level Tank

Level (h):

$$0 - 5V \;\; -> \;\; 0 - 20cm$$

This is approximately. Unless you need a more accurate relation, you can assume this range in your applications

- Use the “To PC” connectors

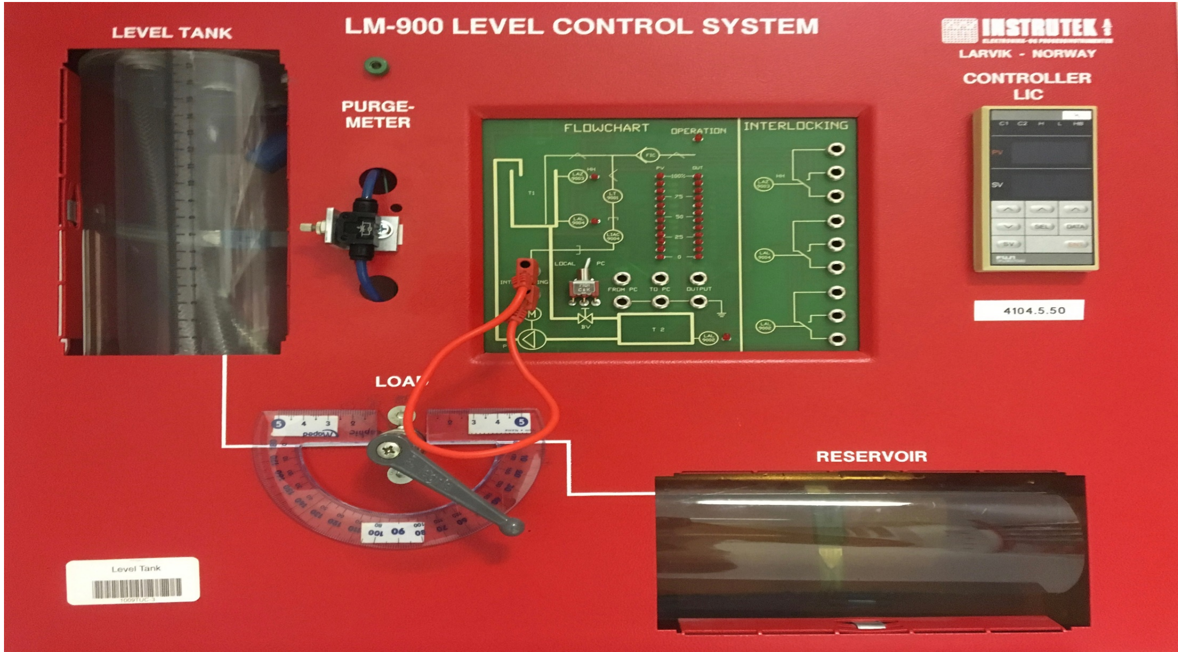
Control Signal (u):

$$0 - 5V$$

- Use the “From PC” connectors

The water level is measured by a level sensor

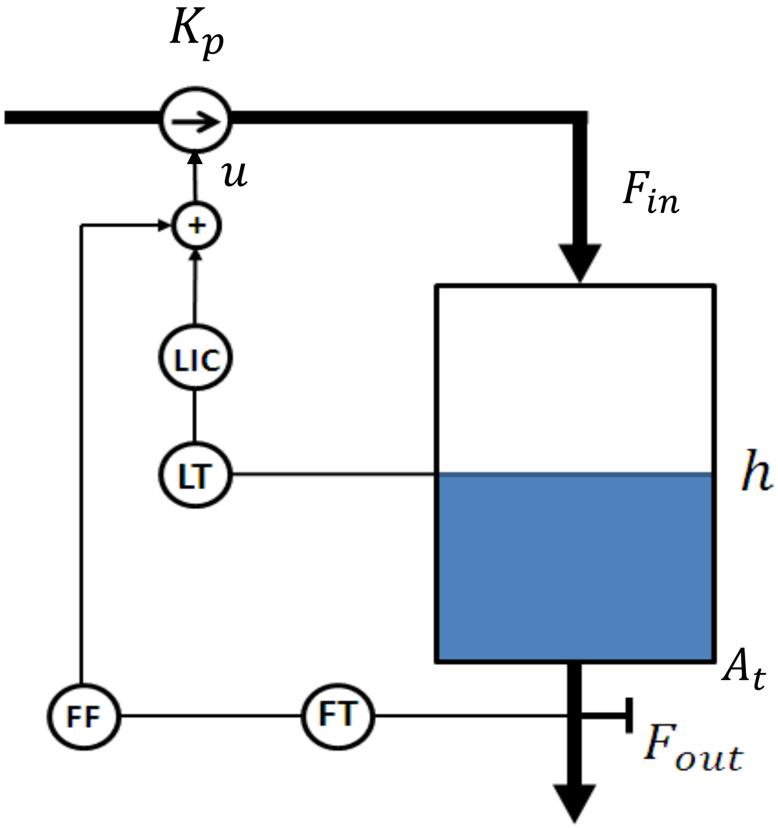
The pump should be controlled by an external voltage signal at the “From PC connector



Mathematical Model

Hans-Petter Halvorsen

Level Tank



$$A_t \frac{dh}{dt} = F_{in} - F_{out}$$

or:

$$\dot{h} = \frac{1}{A_t}(K_p u - F_{out})$$

Where:

- F_{in} - flow into the tank , F_{in}= $K_p u$
- F_{out} - flow out of the tank
- A_t is the cross-sectional area of the tank

LM-900 Level System

Can be manually adjusted

- For real system: a handle on the red tank
- For Simulator: A Numeric control on the Front Panel (HMI)

The level is measured

$$\dot{h} = \frac{1}{A_t}\left[K_p(u - u_0) - F_{out}\right]$$

u_0 is the bias voltage needed to get any flow (with u less than u_0 there is no flow into the tank)

We need to find the unknown model parameter(s) using System Identification methods

(A_t can be found by measuring the radius of the tank)

$$A_t \approx 78.5\ cm$$

Level Tank model – Integrator Model

$$\dot{h} = \frac{1}{A_t}\left[K_p(u - u_0) - F_{out}\right]$$

u_0 is the bias voltage needed to get any flow (with u less than u_0 there is no flow into the tank)

- K_p $[cm^3/s)/V]$ is the pump gain
- F_{out}[cm^3/s] is is the outflow through the valve
- A_t $[cm^2]$ is the cross-sectional area of the tank
- u $[V]$ is the control signal to the pump

You can use this model in your linear Kalman filter algorithm

Level Tank model - 1.order linear system

A more accurate model may, e.g., be:

$$\dot{h} = \frac{1}{A_t}\left[K_p(u - u_0) - K_v h\right]$$

where K_v is the valve gain on the outflow.

u_0 is the bias voltage needed to get any flow (with u less than u_0 there is no flow into the tank)

It is more normal to put it like this:

$$\dot{h} = -\frac{K_v}{A_t}h + \frac{K_p}{A_t}u$$

(The general term is $\dot{x} = ax + bu$)

The model above is a so-called Time-constant system (1.order linear system).

Level Tank model - 1.order Nonlinear Model

The following model is even more accurate:

$$\dot{h} = \frac{1}{A_t}\left[K_p(u - u_0) - K_v\sqrt{\rho g h}\right]$$

This is a so-called 1.order nonlinear model

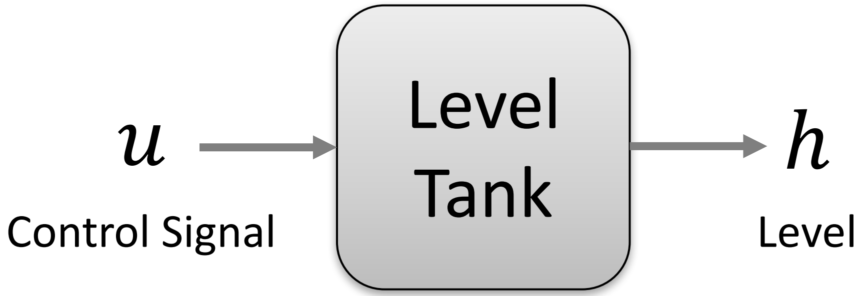
- $h\ [cm]$ is the level
- $u\ [V]$ is the pump control signal to the pump
- u_0 is the bias voltage needed to get any flow (with u less than u_0 there is no flow into the tank)
- $A_t[cm^2]$ is the cross-sectional area of the tank
- $K_p[(cm3/s)/V]$ is the pump gain
- K_v is the valve constant. It depends on the opening of the valve, but if the opening is constant, K_v is constant
- ρ is the is the density of the liquid (water: $1\ kg/m^3$)
- g is the is the gravity constant, $9.81\ \mathrm{m}/s^2$

You may find K_p and K_v using, e.g., the Least Square method

“Black Box Model”

- The Real Level Tank is only available in the Laboratory
- The Level Tank is also provided as a “black box”. Actually, it is just a LabVIEW SubVI where the Block Diagram and the Process Parameters are hidden for you.
- Useful when you are working outside the Laboratory

“Real Process” → “Black Box Model”



You can assume that the following model is a good representation of the Black Box Model:

$$\dot{h} = \frac{1}{A_t}\left[K_p u - F_{out}\right]$$

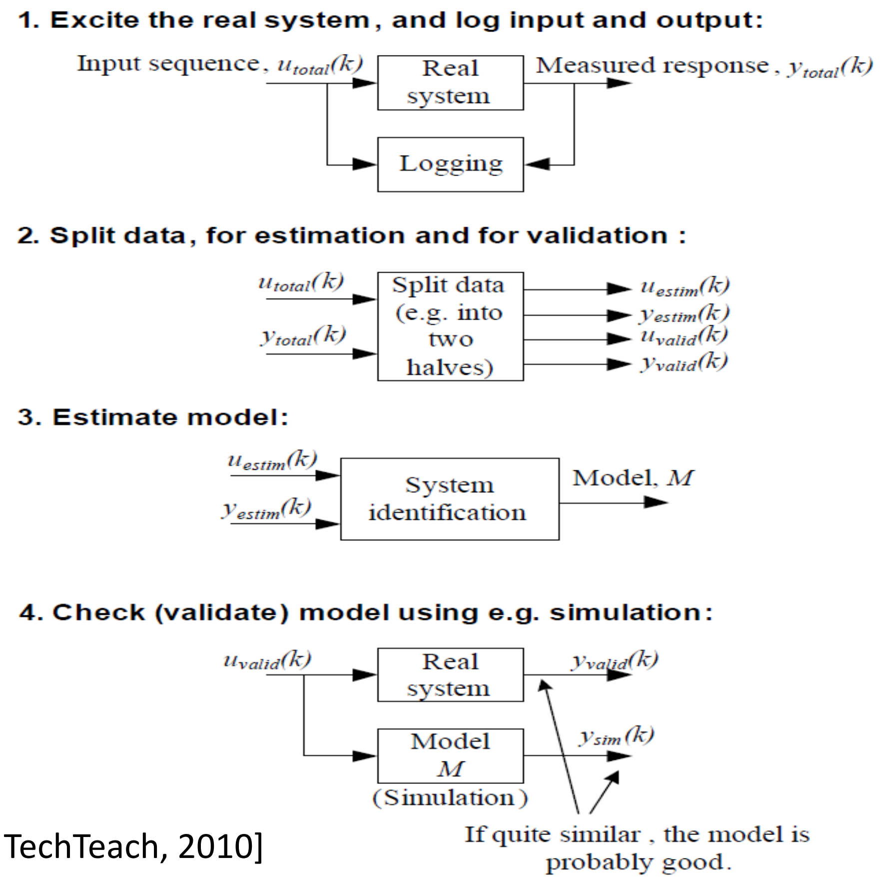
This means you need to unknown parameters using some kind of system identification method

System Identification

Theory

In general, System Identification consists of the following steps:

Make sure to include all these steps in your solution.



[Figure: F. Haugen, Advanced Dynamics and Control: TechTeach, 2010]

System Identification

Theory

You can find the Model Parameters using, e.g.,:

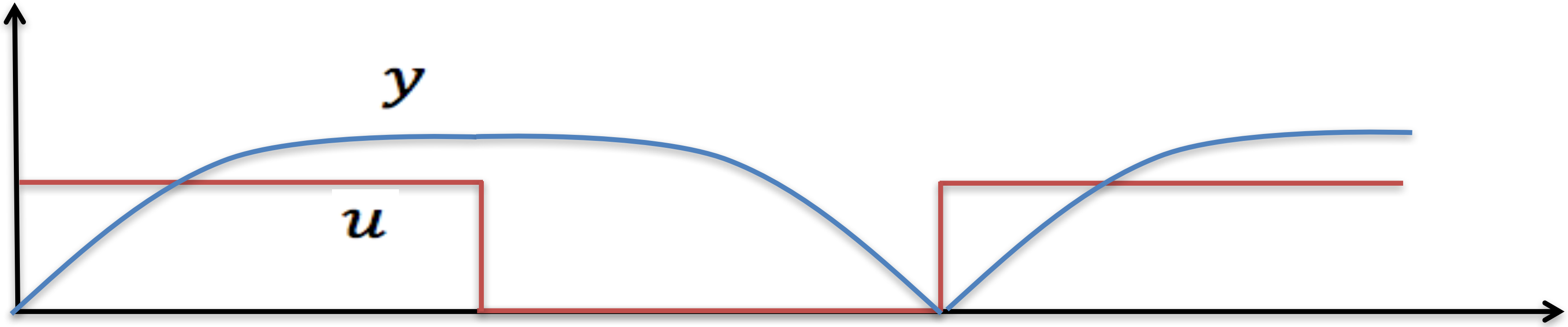
- The **Least Square Method**
- Then adjust and fine-tune the Model Parameters using the "**Trial and Error**" method if necessary

$$\theta_{LS} = (\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}\mathrm{Y}$$

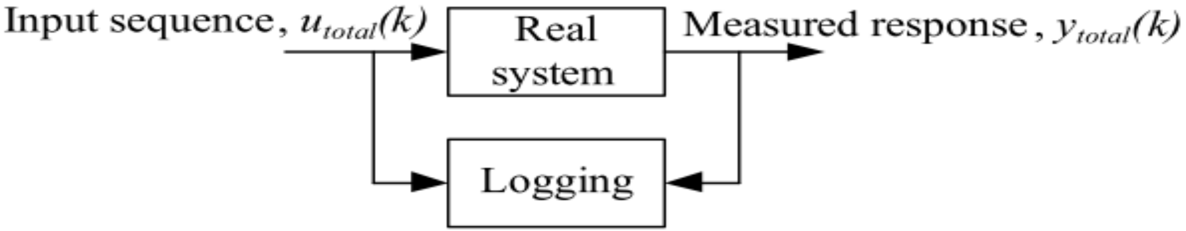
Data Logging

Theory

1. Exite the Real System, e.g.:



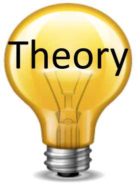
2. Log Data to File



[Figure: F. Haugen, Advanced Dynamics and Control: TechTeach, 2010]

3. Use the Logged Data to find the model or the model parameters

Least Square Example



Given:

$$\dot{x} = ax + bu$$

We want to find the unknown a and b.

This gives:

$$\underbrace{\dot{x}}_{y} = \underbrace{[x \quad u]}_{\varphi} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\theta}$$

i.e.,:

$$\theta = \begin{bmatrix} a \\ b \end{bmatrix}$$

Then we need to discretize:

$$\dot{x} \approx \frac{x_{k+1} - x_k}{T_s}$$

This gives:

$$\underbrace{\frac{x_{k+1} - x_k}{T_s}}_{y} = \underbrace{[x_k \quad u_k]}_{\varphi} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\theta}$$

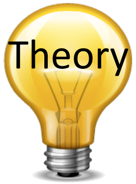
Based on logged data we get:

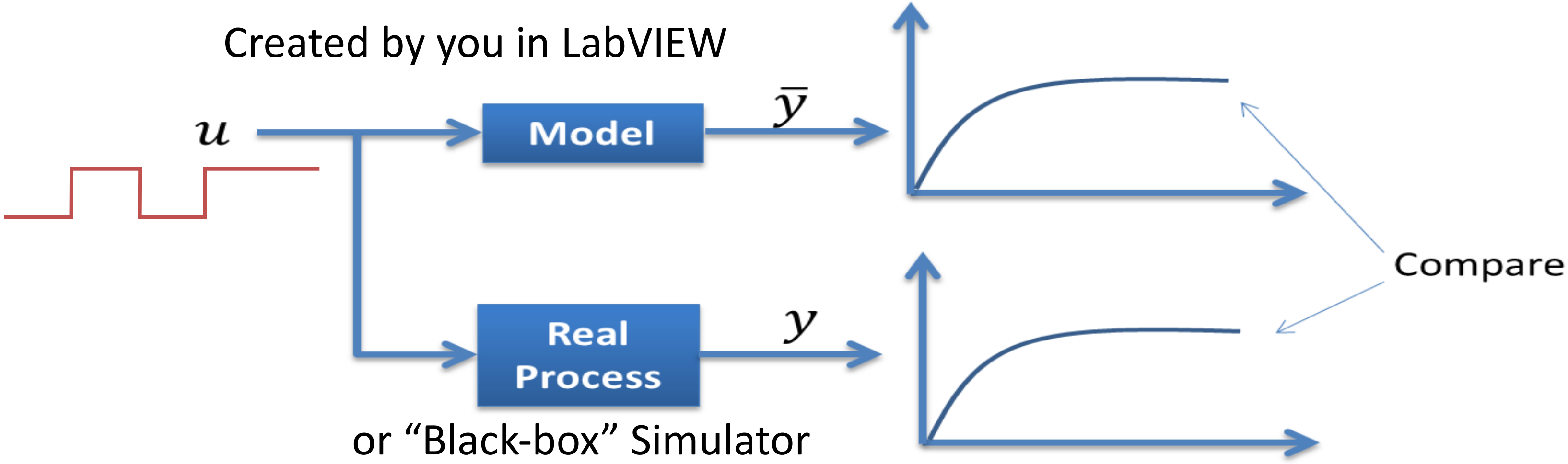
$$\underbrace{\begin{bmatrix} \vdots \\ \vdots \\ \frac{x_{k-1} - x_{k-2}}{T_s} \\ \frac{x_k - x_{k-1}}{T_s} \\ \frac{x_{k+1} - x_k}{T_s} \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} \vdots & \vdots \\ \vdots & \vdots \\ x_{k-2} & u_{k-2} \\ x_{k-1} & u_{k-1} \\ x_k & u_k \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\theta}$$

The we find the uknows a and b using LS:

$$\theta_{LS} = (\Phi^{T}\Phi)^{-1}\Phi^{T}Y$$

Trial & Error Method





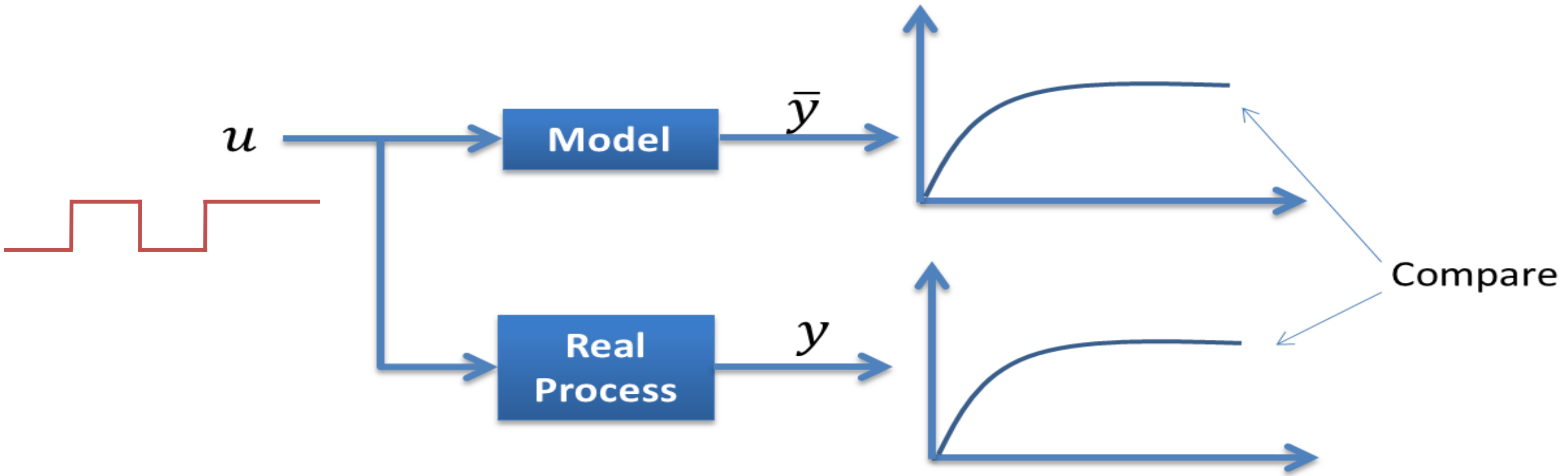
Adjust model parameters and then compare the response from the real system with the simulated model. If they are "equal", you have probably found a good model (at least in that working area)

Model Validation

Theory

Make sure to validate that your model works as expected

Example of simple model validation:



Model Values

$$\dot{h} = \frac{1}{A_t}[K_p u - F_{out}]$$

If you don't have the red Level Tank nearby, you may use the following values as a starting point for your simulations:

$$A_t = 78.5\ cm$$

$$K_p = 16.5\ cm^3/s$$

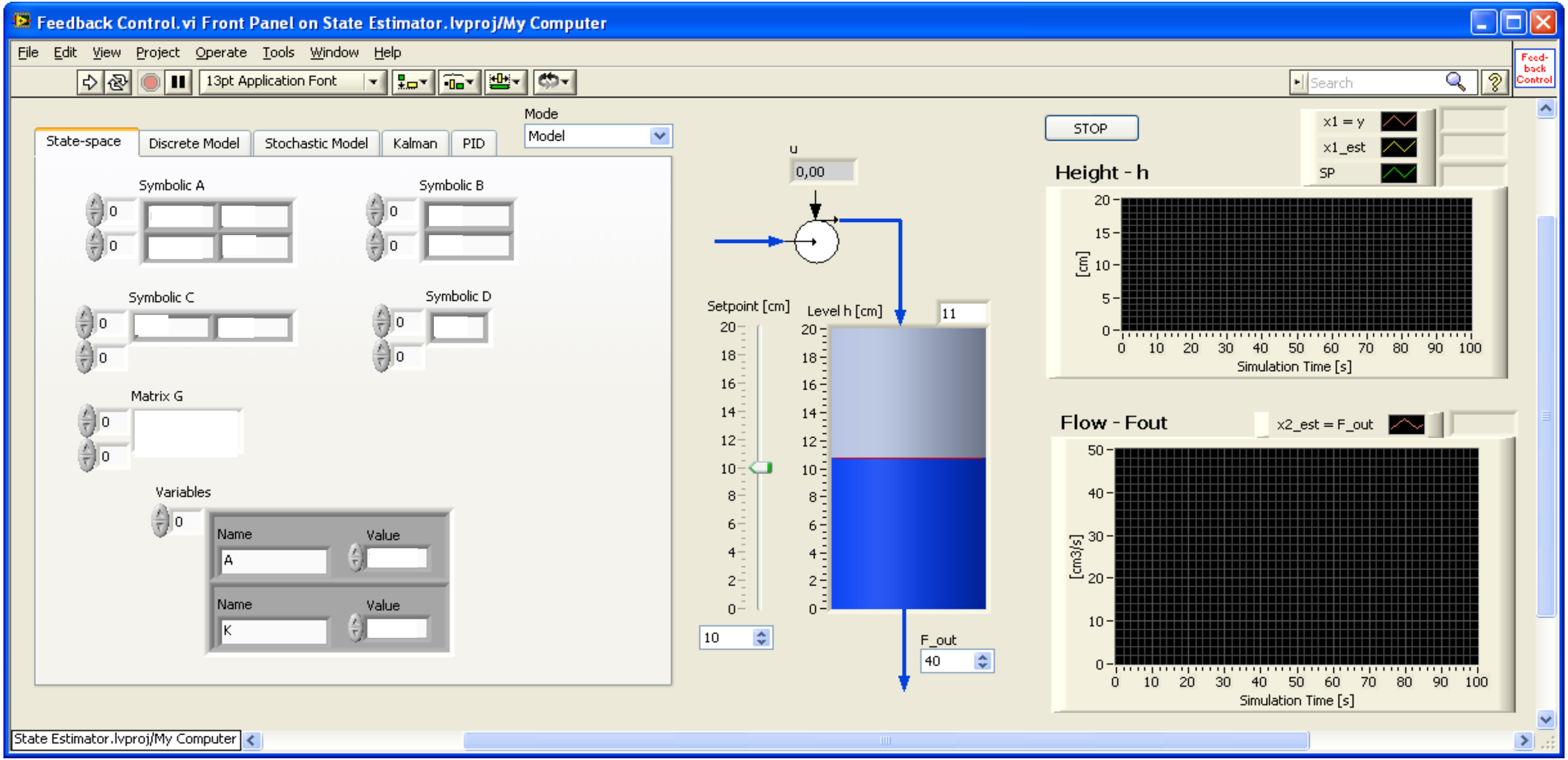
F_{out} should be adjustable from your Front Panel

The range for F_{out} could, e.g.., be $0 \le F_{out} \le 40 cm^3/s$

PID Control

Hans-Petter Halvorsen, M.Sc.

LabVIEW Example

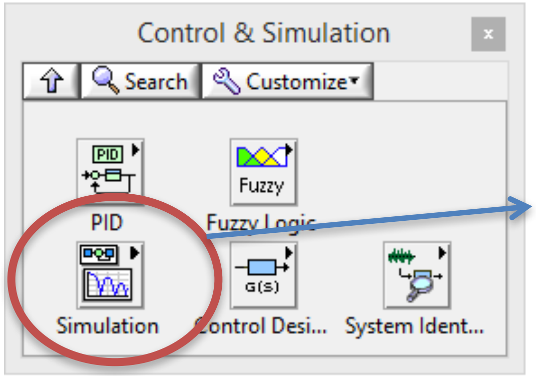


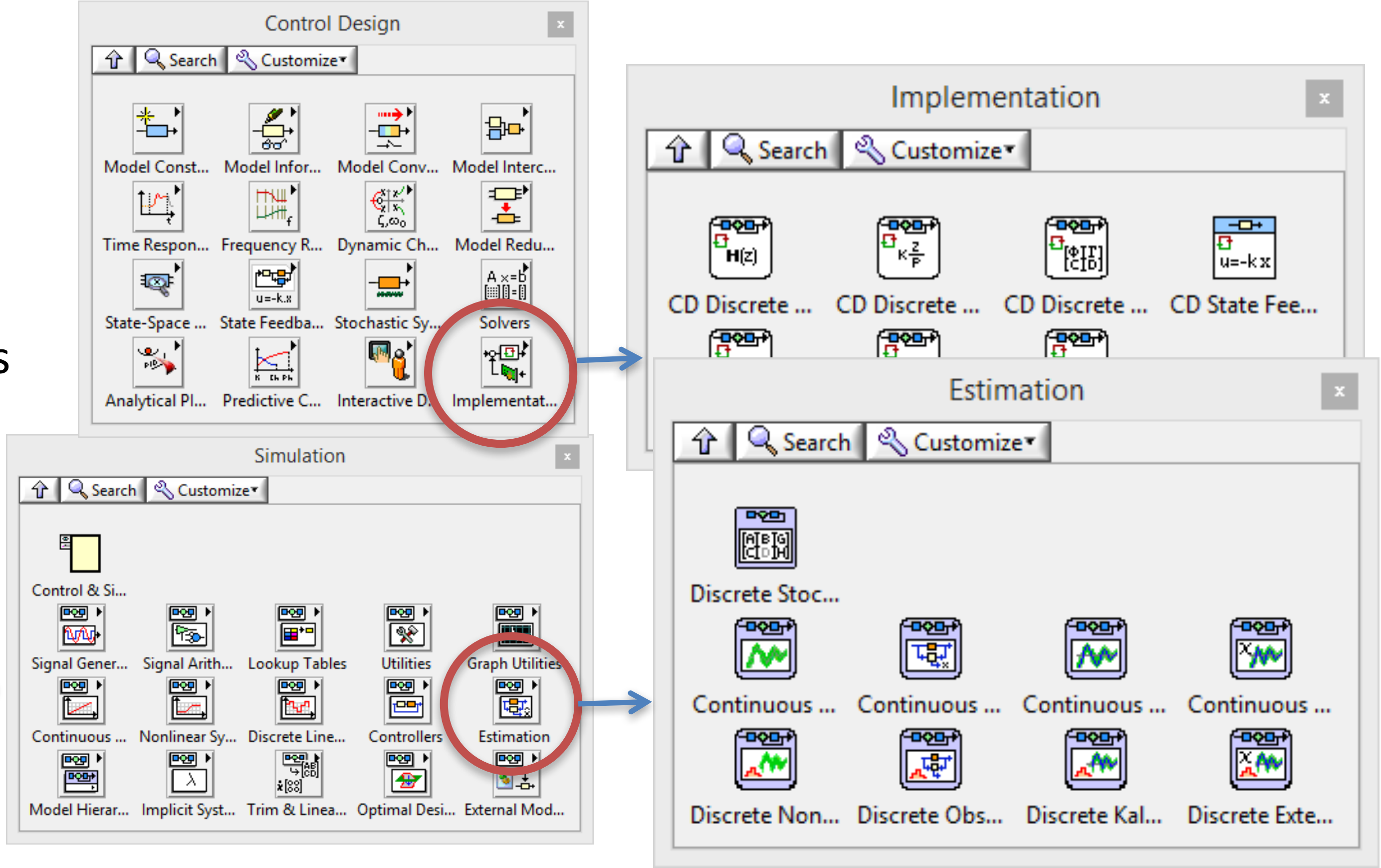
State Estimation and Kalman Filter

Hans-Petter Halvorsen

State Estimation in LabVIEW

"LabVIEW Control Design and Simulation Module" has built-in features for State Estimation, including different types of Kalman Filter algorithms





Level System Model

$$\dot{h} = \frac{1}{A_t}[K_p u - F_{out}]$$

For the real system, only the level (*h*) is measured, so we want to use a Kalman Filter for estimating the outflow (*Fout*) of the tank (Which we will use in a Feedforward control later).

1. Set *x1=h* and *x2=Fout* and assume that *Fout* is constant. Find the state-space model for the system.

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

2. Then you can find the discrete state-space model for the system as well

$$x_{k+1} = Ax_k + Bu_k$$

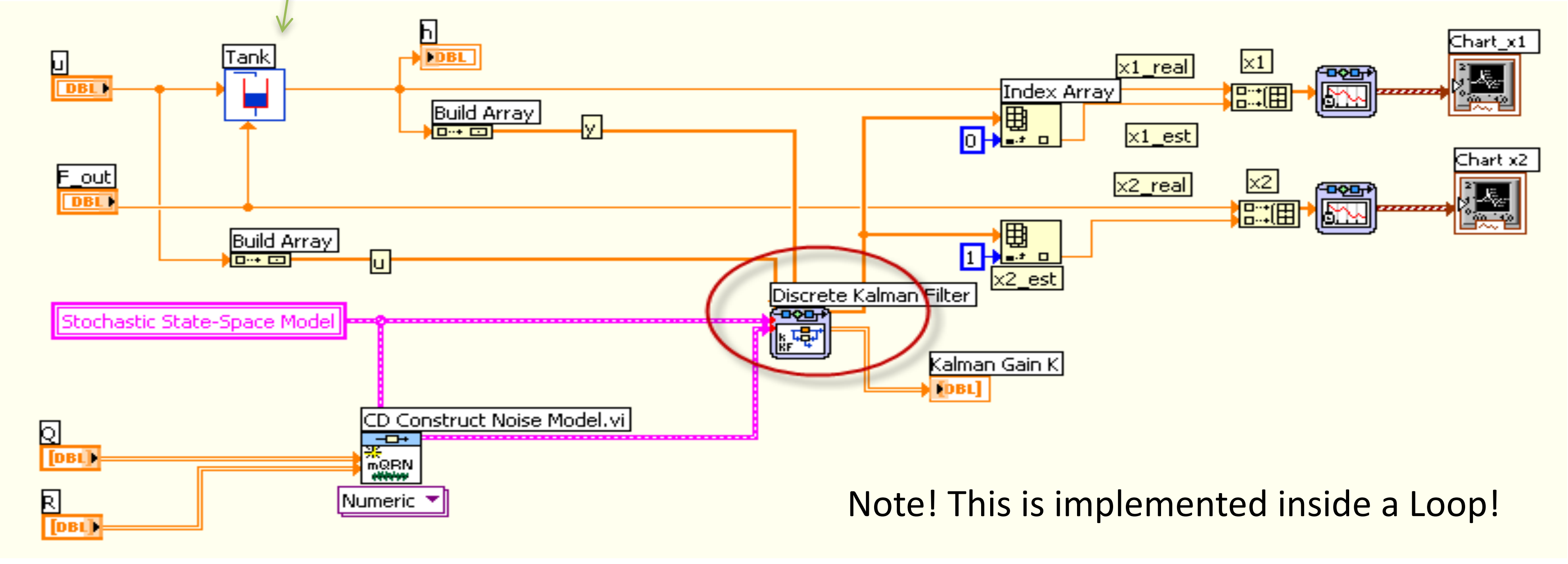
$$y_k = Cx_k + Du_k$$

3. The discrete state-space model can then be used in a Kalman Filter algorithm.

Kalman Filter in LabVIEW

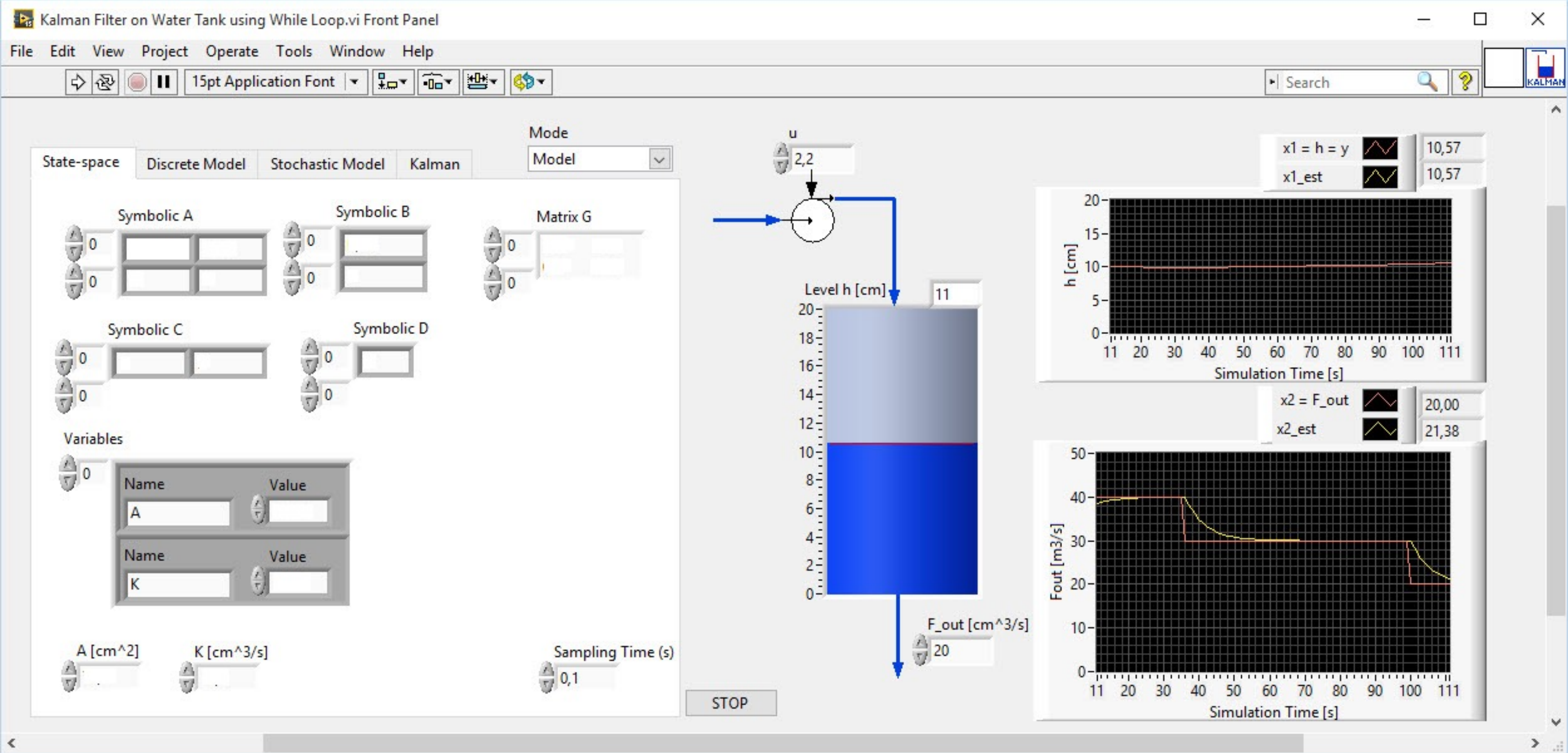
Example

Start using a simulator (model). When the simulator is working properly, switch to the real process. You may also add some noise to your model to make it more realistic.

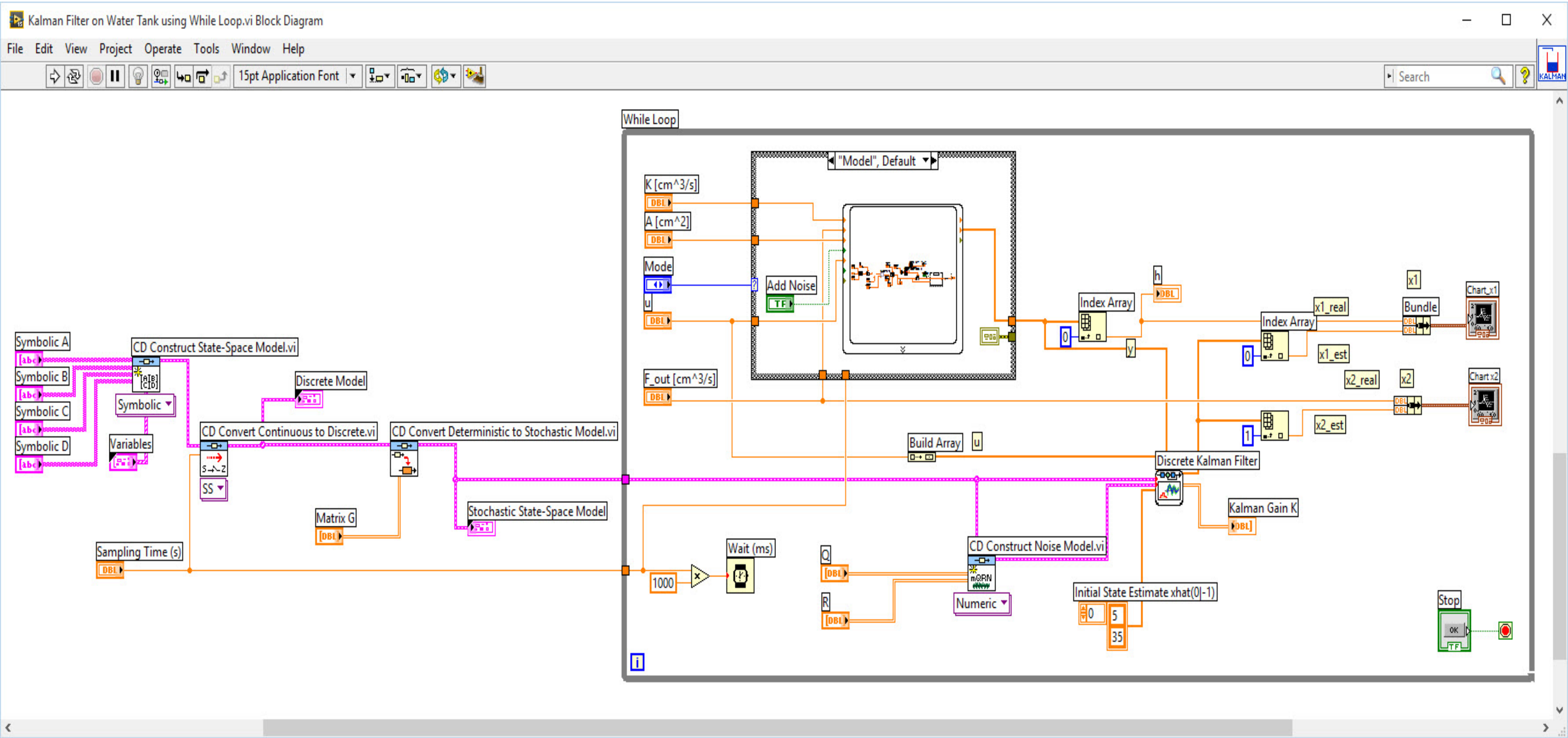


Note! This is implemented inside a Loop!

LabVIEW Example (Kalman Filter)



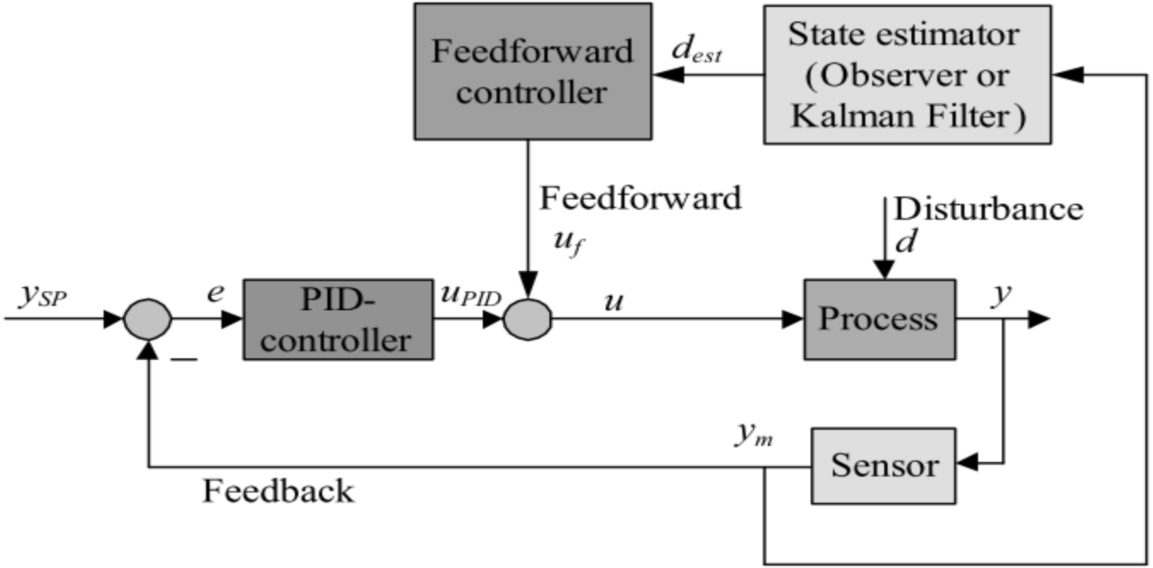
LabVIEW Example (Kalman Filter)



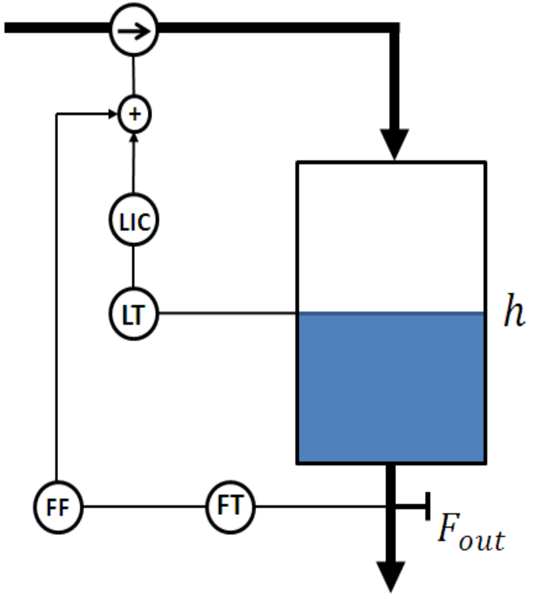
Feedforward Control

Hans-Petter Halvorsen, M.Sc.

Feedforward Control



[Figure: F. Haugen, Advanced Dynamics and Control: TechTeach, 2010]



$$\dot{h} = \frac{1}{A_t}\left[K_p u - F_{out}\right]$$

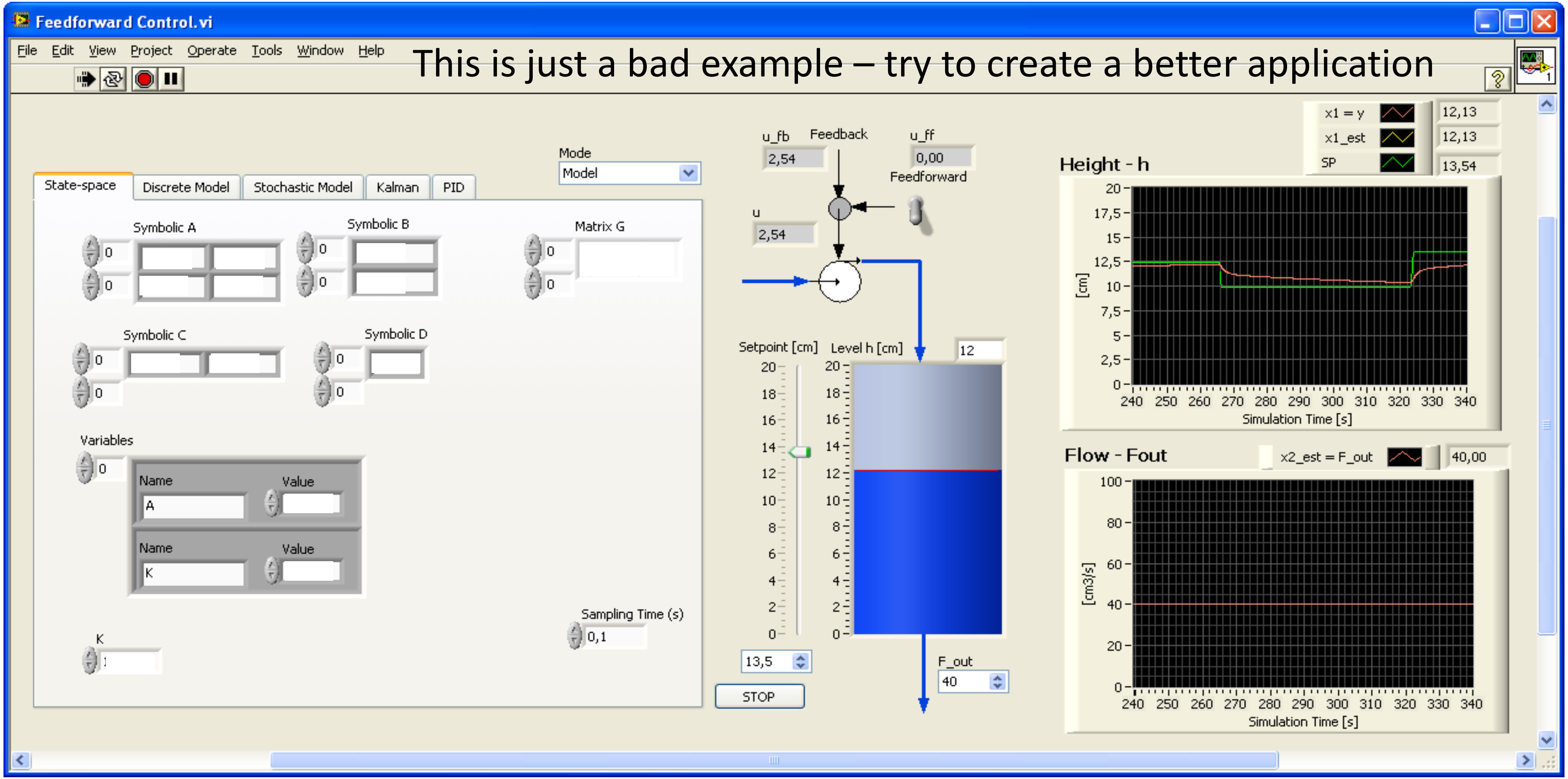
Feedforward Control

- In this model is *Fout* a noise signal/disturbance that we want to remove by using Feedforward.
- We want to design the Feedforward controller so that *Fout* is eliminated.
  - Solve for the control variable *u*, and substituting the process output variable *h* by its setpoint h_{sp}.
  - *Fout* is not measured, so you need to use the estimated value instead. Assume that the setpoint is constant.

We will use Feedforward Control in order to improve the control, compared to ordinary Feedback Control.

LabVIEW Example (PID + Kalman + FF)

This is just a bad example – try to create a better application



Hans-Petter Halvorsen

University of Southeast Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: http://www.halvorsen.blog