

<https://www.halvorsen.blog>



# Air Heater System

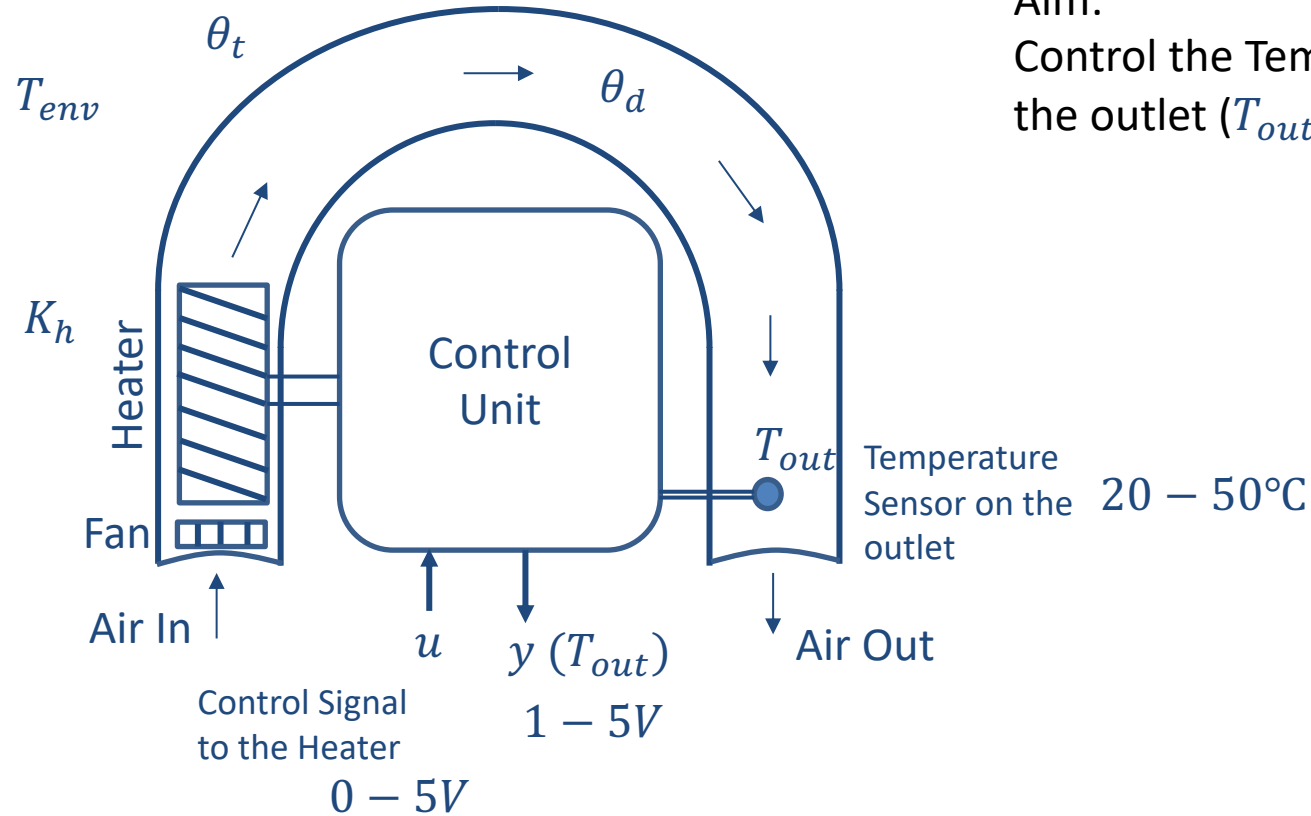
Simulation Examples

Hans-Petter Halvorsen

# Introduction

- We want to simulate an Air Heater System (1. order differential equation) using LabVIEW
- We will use 3 different alternatives:
  - MathScript Window (similar to MATLAB)
  - MathScript Node inside LabVIEW
  - LabVIEW Control Design and Simulation Module
- We should of course expect the same simulation results using the 3 different alternatives

# Air Heater System



Aim:  
Control the Temperature on  
the outlet ( $T_{out}$ )

20 – 50°C



# Model

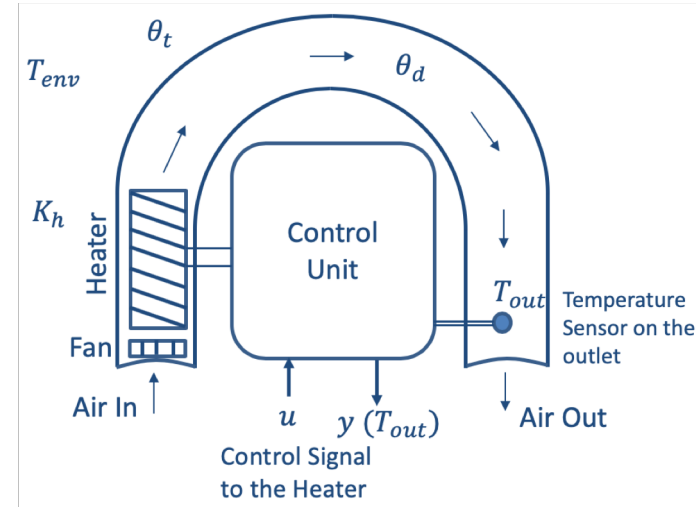
Hans-Petter Halvorsen

# Air Heater System

The system can be modelled as a 1. order system with time-delay

$$\dot{T}_{out} = \frac{1}{\theta_t} \{-T_{out} + [K_h u(t - \theta_d) + T_{env}]\}$$

- $T_{out}$  is the air temperature at the tube outlet
- $u$  [V] is the control signal to the heater
- $\theta_t$  [s] is the time-constant
- $K_h$  [deg C / V] is the heater gain
- $\theta_d$  [s] is the time-delay representing air transportation and sluggishness in the heater
- $T_{env}$  is the environmental (room) temperature. It is the temperature in the outlet air of the air tube when the control signal to the heater has been set to zero for relatively long time (some minutes)



# Model Values

You can assume the following values in your simulations:

$$\theta_t = 22 \text{ s}$$

$$\theta_d = 2 \text{ s}$$

$$K_h = 3.5 \frac{\text{°C}}{\text{V}}$$

$$T_{env} = 21.5 \text{ °C}$$

The range for  $T_{out}$  could, e.g., be  $20\text{°C} \leq T_{out} \leq 50\text{°C}$

Alternative #1



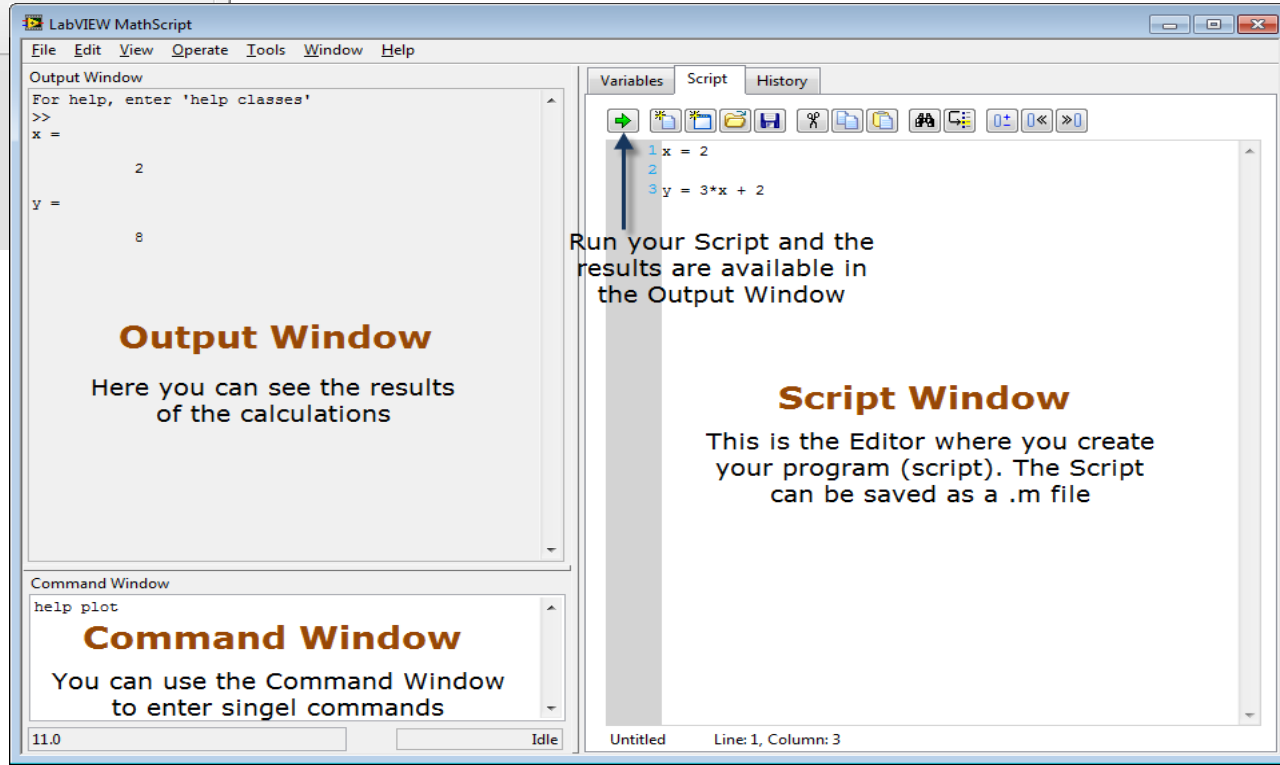
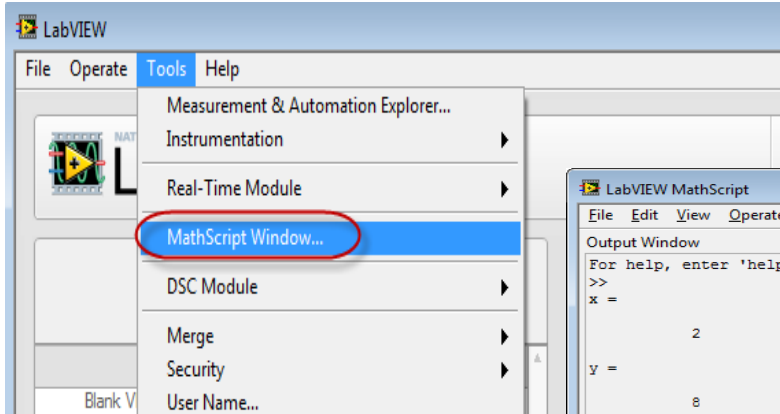
# MathScript Module

LabVIEW MathScript Window

Hans-Petter Halvorsen

# MathScript Window

## LabVIEW MathScript RT Module



The image shows the LabVIEW MathScript window. It has a menu bar with 'File', 'Edit', 'View', 'Operate', 'Tools', 'Window', and 'Help'. The window is divided into three main sections:

- Output Window:** Displays the results of calculations. It shows the prompt '>>' and the output 'x = 2' and 'y = 8'. Below the window, it says: **Output Window**  
Here you can see the results of the calculations
- Script Window:** The editor for creating and editing scripts. It shows a script with three lines: '1 x = 2', '2', and '3 y = 3\*x + 2'. A blue arrow points from the 'Run' button (a green play icon) to the first line of the script. Below the window, it says: **Script Window**  
This is the Editor where you create your program (script). The Script can be saved as a .m file
- Command Window:** Used for entering single commands. It shows the command 'help plot'. Below the window, it says: **Command Window**  
You can use the Command Window to enter single commands

The status bar at the bottom shows '11.0' on the left, 'Idle' in the middle, and 'Untitled Line: 1, Column: 3' on the right.

Add-on Module for LabVIEW  
where we can do text-based  
programming and simulations  
– very powerful!



# Discretization

- In order to simulate this system, we typically need to find the discrete differential equation (difference equation)
- We can use e.g., the **Euler** Approximation:

$$\dot{x} \approx \frac{x(k+1) - x(k)}{T_s}$$

# Discretization of Air Heater

Continuous Model:

$$\dot{T}_{out} = \frac{1}{\theta_t} \{-T_{out} + [K_h u(t - \theta_d) + T_{env}]\}$$

We can use e.g., the Euler Approximation in order to find the discrete Model:

$$\dot{x} \approx \frac{x(k+1) - x(k)}{T_s}$$

$T_s$  - Sampling Time       $x(k)$  - Present value  
 $x(k+1)$  - Next (future) value

The discrete Model will then be on the form:

$$x(k+1) = x(k) + \dots$$

We can then implement the discrete model in any programming language

# Discretization of Air Heater

We make a discrete version:

$$\dot{T}_{out} = \frac{1}{\theta_t} \{-T_{out} + [K_h u(t - \theta_d) + T_{env}]\}$$

$$\frac{T_{out}(k+1) - T_{out}(k)}{T_s} = \frac{1}{\theta_t} \{-T_{out}(k) + [K_h u(k - \theta_d) + T_{env}]\}$$

This gives the following discrete system:

$$T_{out}(k+1) = T_{out}(k) + \frac{T_s}{\theta_t} \{-T_{out}(k) + [K_h u(k - \theta_d) + T_{env}]\}$$

The Time delay  $\theta_d$  makes it a little complicated. **Note!** We can simplify by setting  $\theta_d = 0$

$$T_{out}(k+1) = T_{out}(k) + \frac{T_s}{\theta_t} \{-T_{out}(k) + [K_h u(k) + T_{env}]\}$$

# Discretization of Air Heater

Discrete version with Time delay  $\theta_d = 0$

$$T_{out}(k + 1) = T_{out}(k) + \frac{T_s}{\theta_t} \{-T_{out}(k) + K_h u(k) + T_{env}\}$$

We can use the following values in the simulation:

$$\theta_t = 22$$

$$K_h = 3.5$$

$$T_{env} = 21.5$$

We can set the Sampling Time  $T_s = 0.1s$

We need also to specify an initial value for  $T_{out}$  in our simulations., e.g., set  $T_{out,init} = 20$

You can of course also explore with other values for these parameters in your simulations

# MathScript Code

This is an example.  
You can implement it in many  
different ways

```
% Simulation of Air Heater System
clear, clc

% Model Parameters
Theta_t = 22;
Kh = 3.5;
Tenv = 21.5;

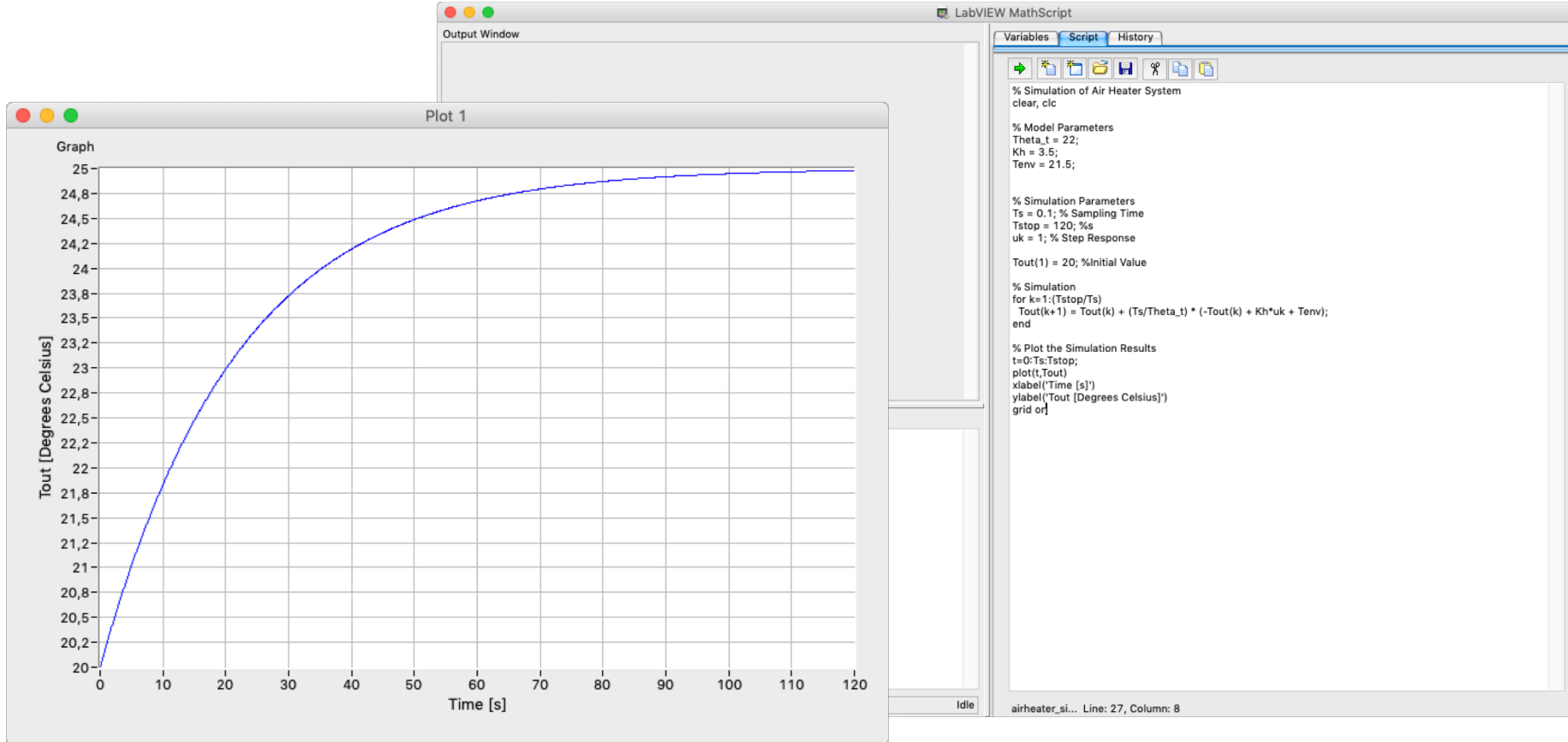
% Simulation Parameters
Ts = 0.1; % Sampling Time
Tstop = 120; %s
uk = 1; % Step Response

Tout(1) = 20; %Initial Value

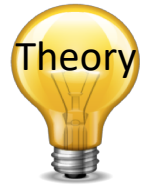
% Simulation
for k=1:(Tstop/Ts)
    Tout(k+1) = Tout(k) + (Ts/Theta_t) * (-Tout(k) + Kh*uk + Tenv);
end

% Plot the Simulation Results
t=0:Ts:Tstop;
plot(t,Tout)
xlabel('Time [s]')
ylabel('Tout [Degrees Celsius]')
grid on
```

# MathScript Code and Simulation Results

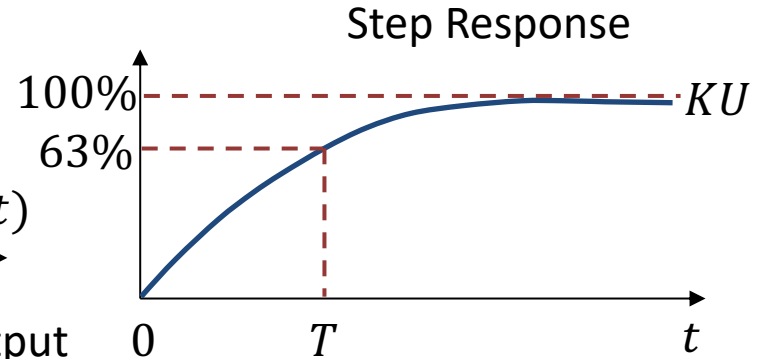
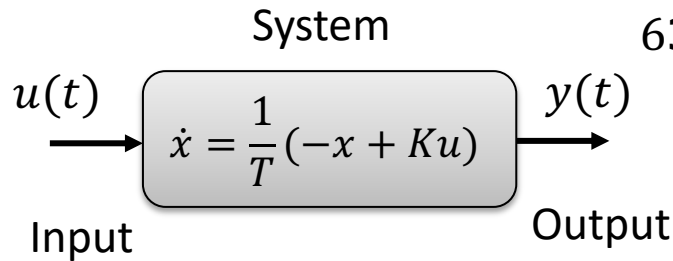
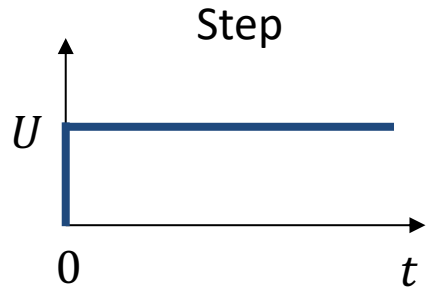


# Step Response 1. Order System



$$\dot{x} = \frac{1}{T}(-x + Ku)$$

We see that the Air Heater System (without time delay) is a standard 1.order system



Alternative #2



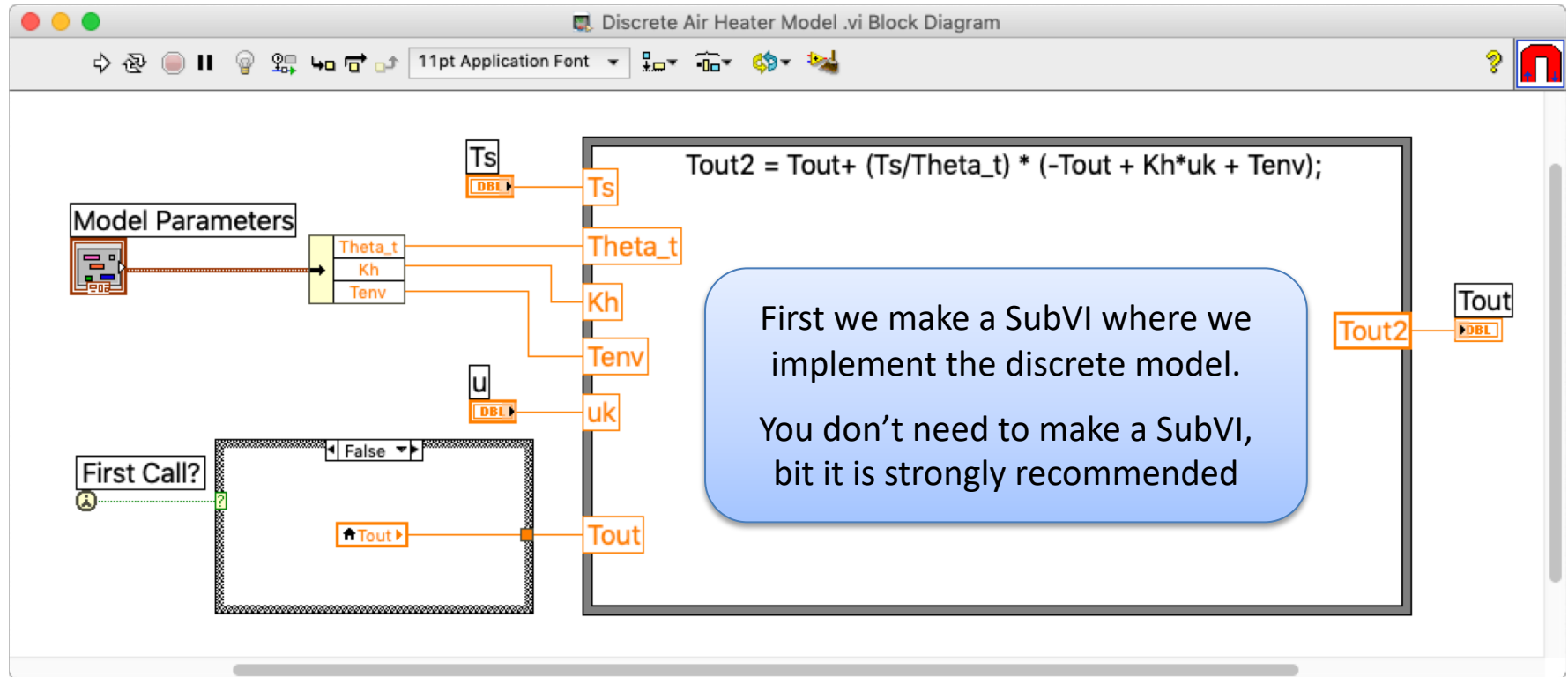
# MathScript Node

Hans-Petter Halvorsen

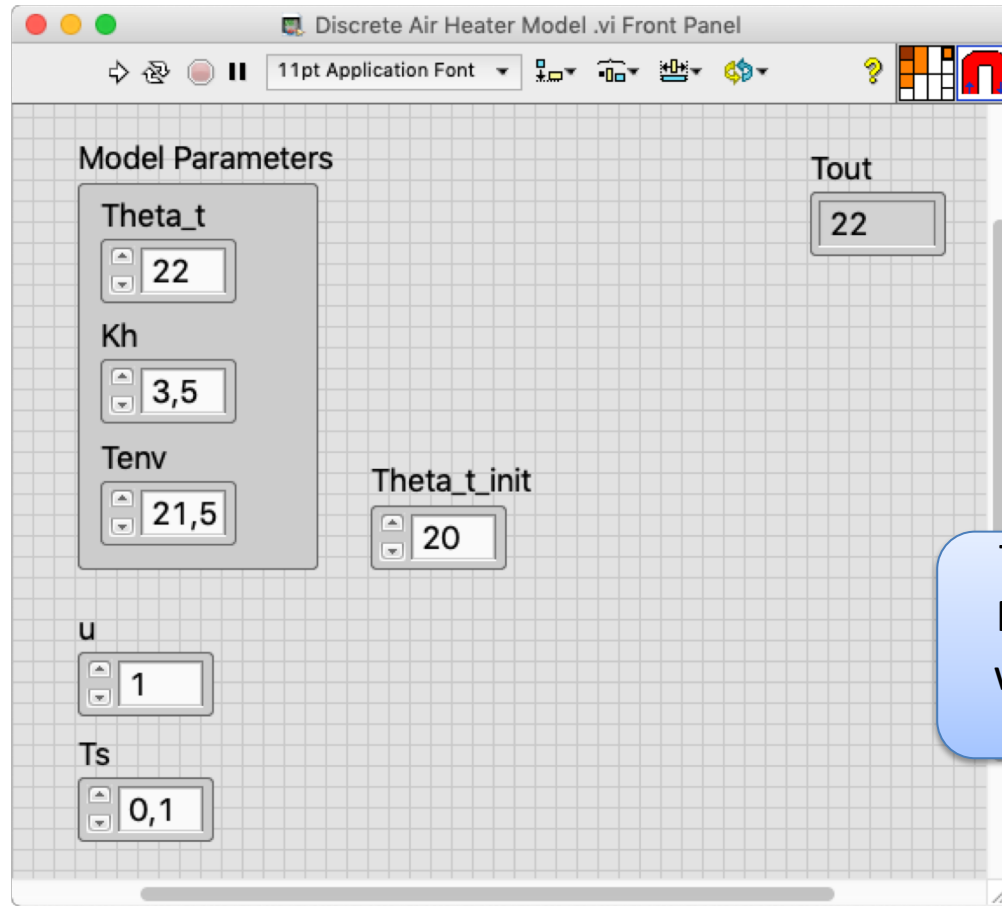


# MathScript Node Code Example

We just copy the previous code into a MathScript Node inside LabVIEW:



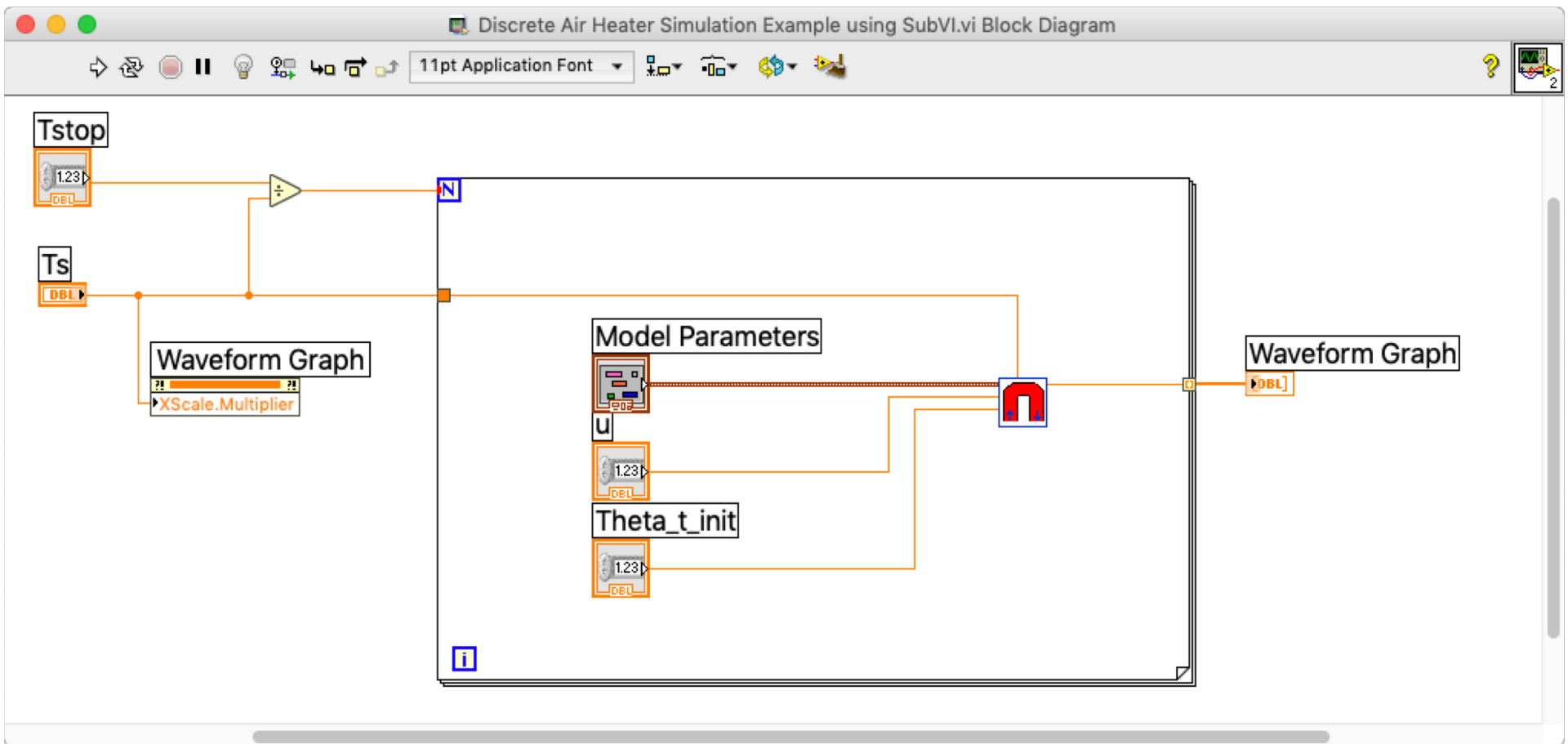
# MathScript Node Code Example



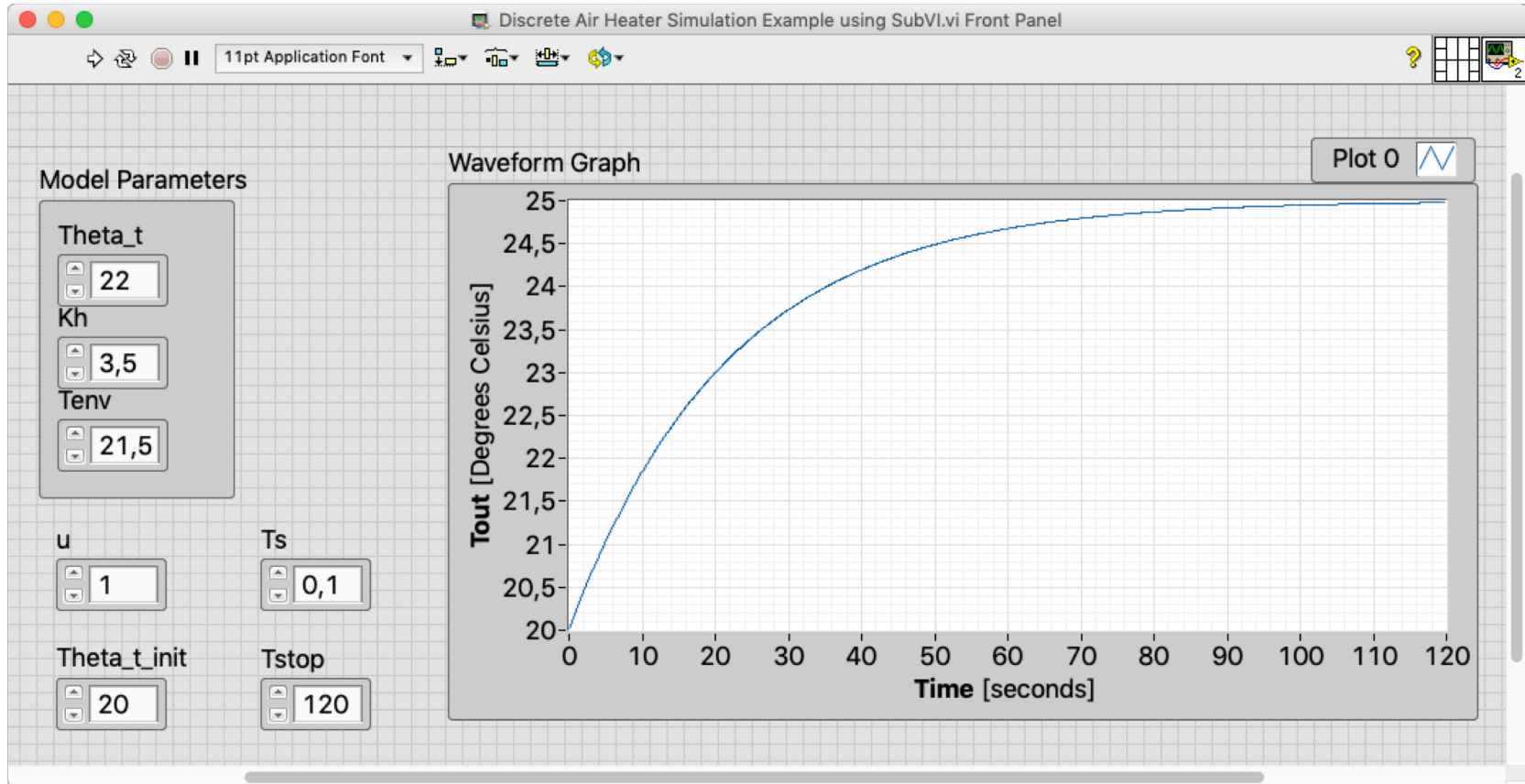
Make sure to make a descriptive Icon for the SubVI

This is an example of the Front Panel for the SubVI where we implement the discrete model

# MathScript Node Code Example



# MathScript Node Code Example



# Alternative #3



# LabVIEW

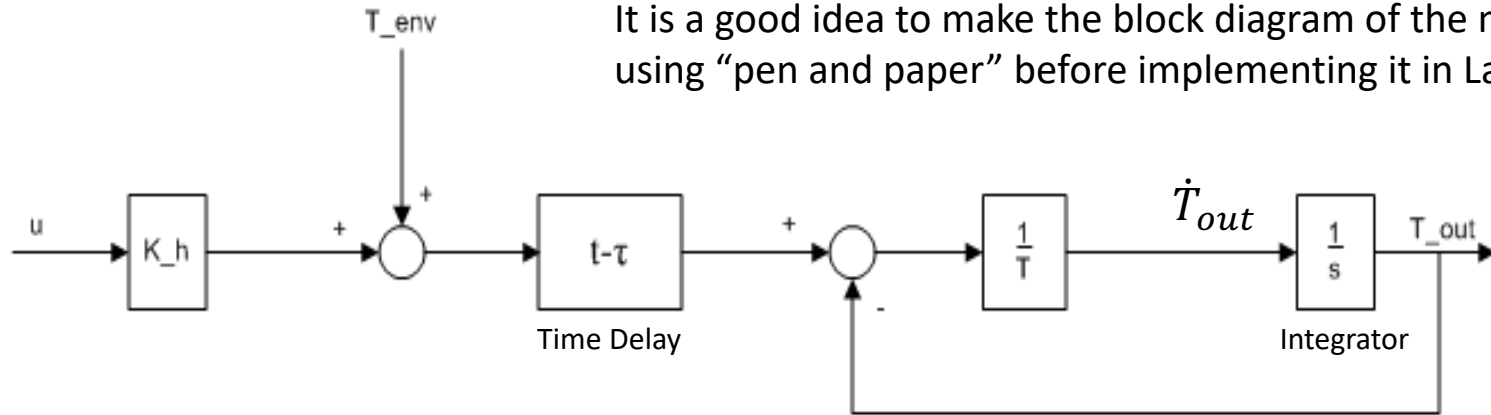
## LabVIEW Control and Design and Simulation Module

Hans-Petter Halvorsen

# Air Heater Block Diagram

$$\dot{T}_{out} = \frac{1}{\theta_t} \{-T_{out} + [K_h u(t - \theta_d) + T_{env}]\}$$

A Block Diagram for the model/differential equation above becomes:

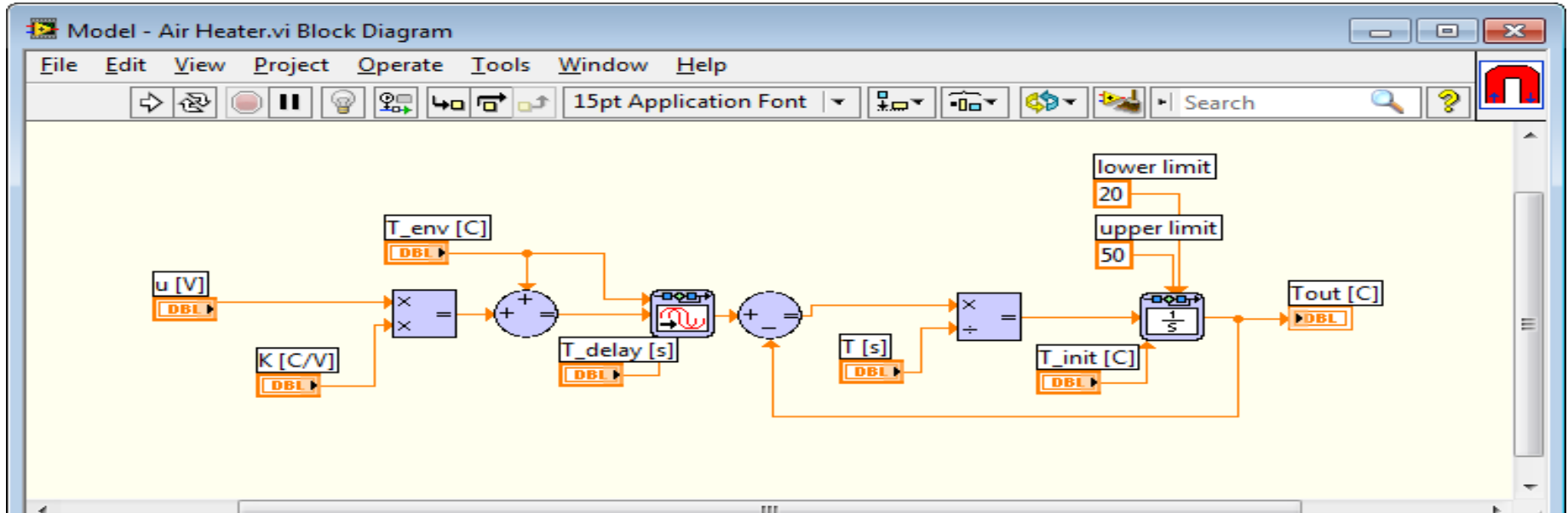


It is a good idea to make the block diagram of the model just using “pen and paper” before implementing it in LabVIEW

**Note!** Here is the time delay included, because there is a predefined time delay block that can be used in LabVIEW

# Air Heater Model in LabVIEW

Here is LabVIEW Control Design and Simulation Module used to create a block diagram model of the Air Heater System

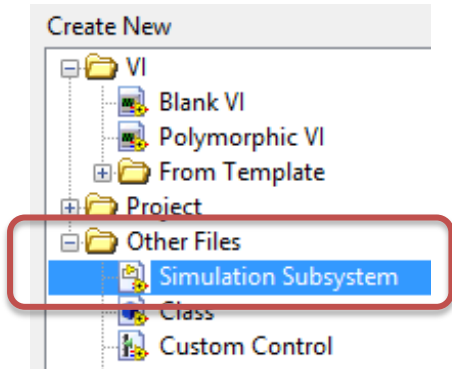


**Note!** This model is implemented in a so-called “**Simulation Subsystem**” (which is recommended, but not necessary). A Simulation Subsystem is just like an ordinary SubVI. Some advantages are that you can reuse the code in different VIs/applications and the code make more structured and easier to maintain, etc.

# Simulation Subsystem

A Way to structure your code, similar to SubVIs

This is the recommended way to do it! – You can easily reuse your Subsystems in different VIs and your code becomes more structured!

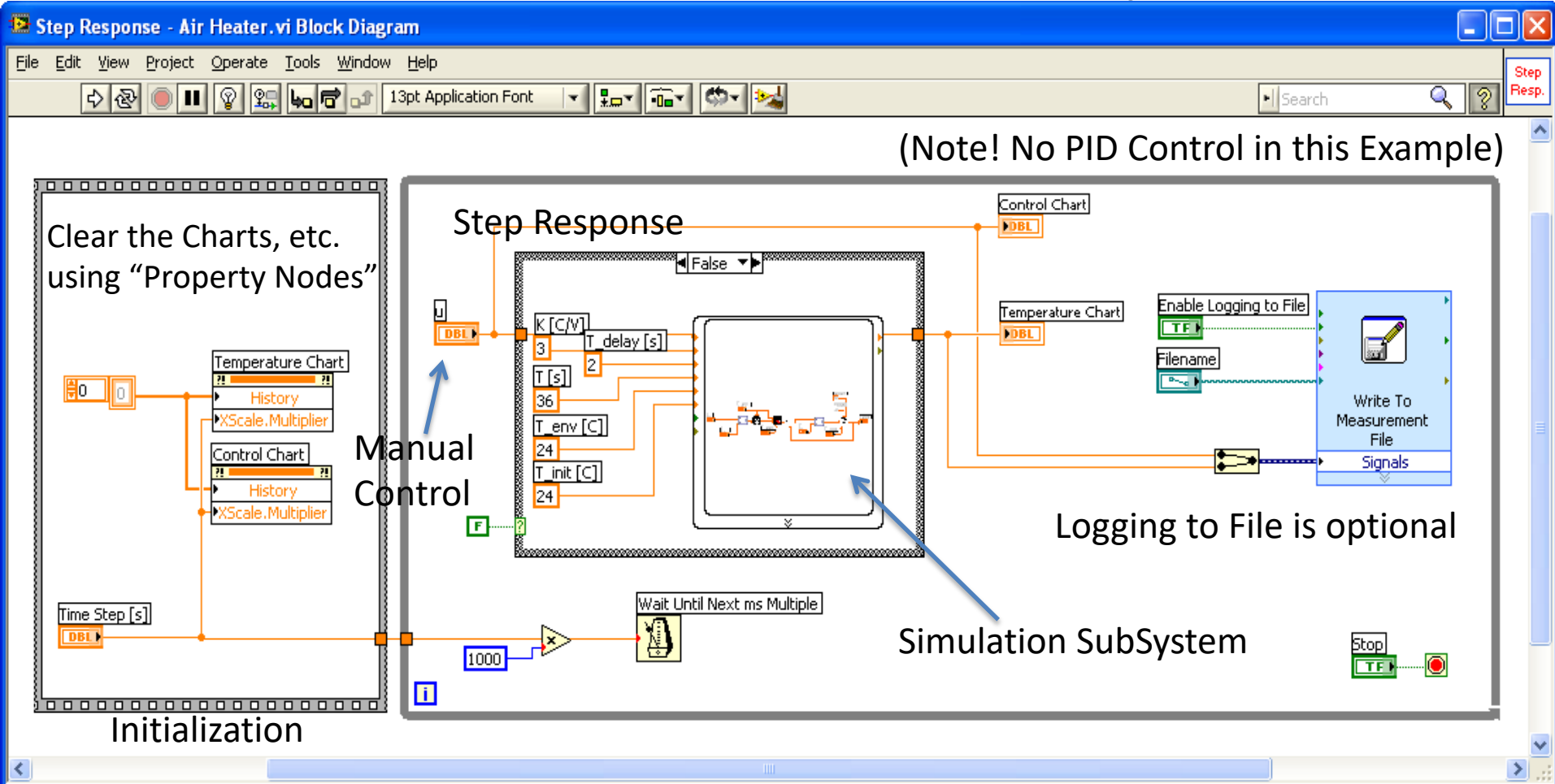


Select File -> New ..., Then choose “Simulation Subsystem”.

Create your Model within the Simulation Subsystem



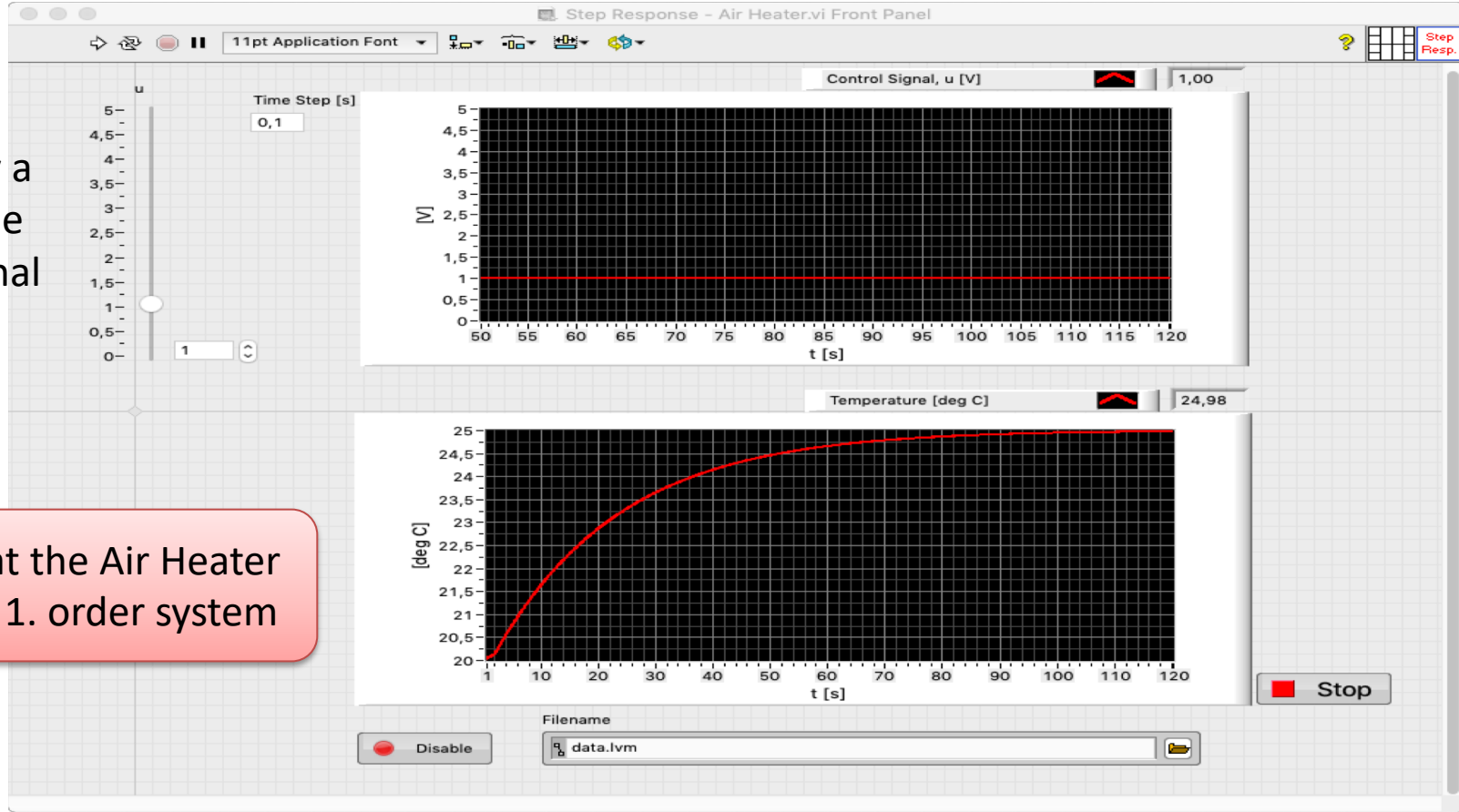
# Air Heater Simulation Example



# Air Heater Step Response (Front Panel)

We apply a step in the input signal

We see that the Air Heater is a typical 1. order system





# Summary

Hans-Petter Halvorsen

# Discussion/Summary/Conclusion

- We have simulated a Air Heater System using LabVIEW
- We have used 3 different alternatives:
  - MathScript Window (similar to MATLAB)
  - MathScript Node inside LabVIEW
  - LabVIEW Control Design and Simulation Module
- We got (of course) the same simulation results using the 3 different alternatives



# Further Work

Hans-Petter Halvorsen

- Improve your Graphical User Interface (GUI)
- Introduce the Time Delay in your Discrete model implementation of the Air Heater
- Etc.

# Discretization of Air Heater with Time Delay

We have the following discrete system:

$$T_{out}(k + 1) = T_{out}(k) + \frac{T_s}{\theta_t} \{-T_{out}(k) + [K_h u(k - \theta_d) + T_{env}]\}$$

The Time delay  $\theta_d$  makes it more complicated to implement

$\theta_d$  is in seconds and we need to convert it to discrete intervals in forms of  $k$

The discrete version of  $\theta_d$  is:  $\frac{\theta_d}{T_s}$

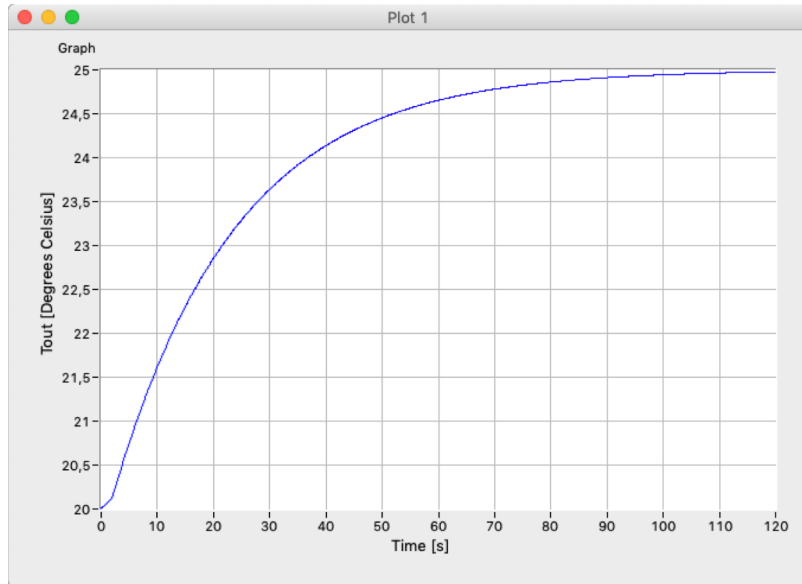
Then we get:

$$T_{out}(k + 1) = T_{out}(k) + \frac{T_s}{\theta_t} \left\{ -T_{out}(k) + \left[ K_h u \left( k - \frac{\theta_d}{T_s} \right) + T_{env} \right] \right\}$$

Assuming  $\theta_d = 2s$  and  $T_s = 0.1s$  we get  $u(k - 20)$

This mean that we have to remember the 20 previous samples of  $u(k)$

# MathScript Example with Time Delay



```
clear, clc
```

```
% Model Parameters
```

```
Theta_t = 22; % Seconds
```

```
Kh = 3.5; % degC/V
```

```
Tenv = 21.5; %degC
```

```
Delay = 2; % Seconds
```

```
% Simulation Parameters
```

```
Ts = 0.1; % Sampling Time
```

```
Tstop = 120; % Seconds
```

```
%uk = 1; % Step Response
```

```
u1 = zeros(Delay/Ts,1);
```

```
u2 = ones(Tstop/Ts,1);
```

```
uk = [u1;u2];
```

```
Tout(1) = 20; %Initial Value
```

```
% Simulation
```

```
for k=1:(Tstop/Ts)
```

```
    Tout(k+1) = Tout(k) + (Ts/Theta_t) * (-Tout(k) +  
    Kh*uk(k) + Tenv);
```

```
end
```

```
% Plot the Simulation Results
```

```
t=0:Ts:Tstop;
```

```
plot(t,Tout)
```

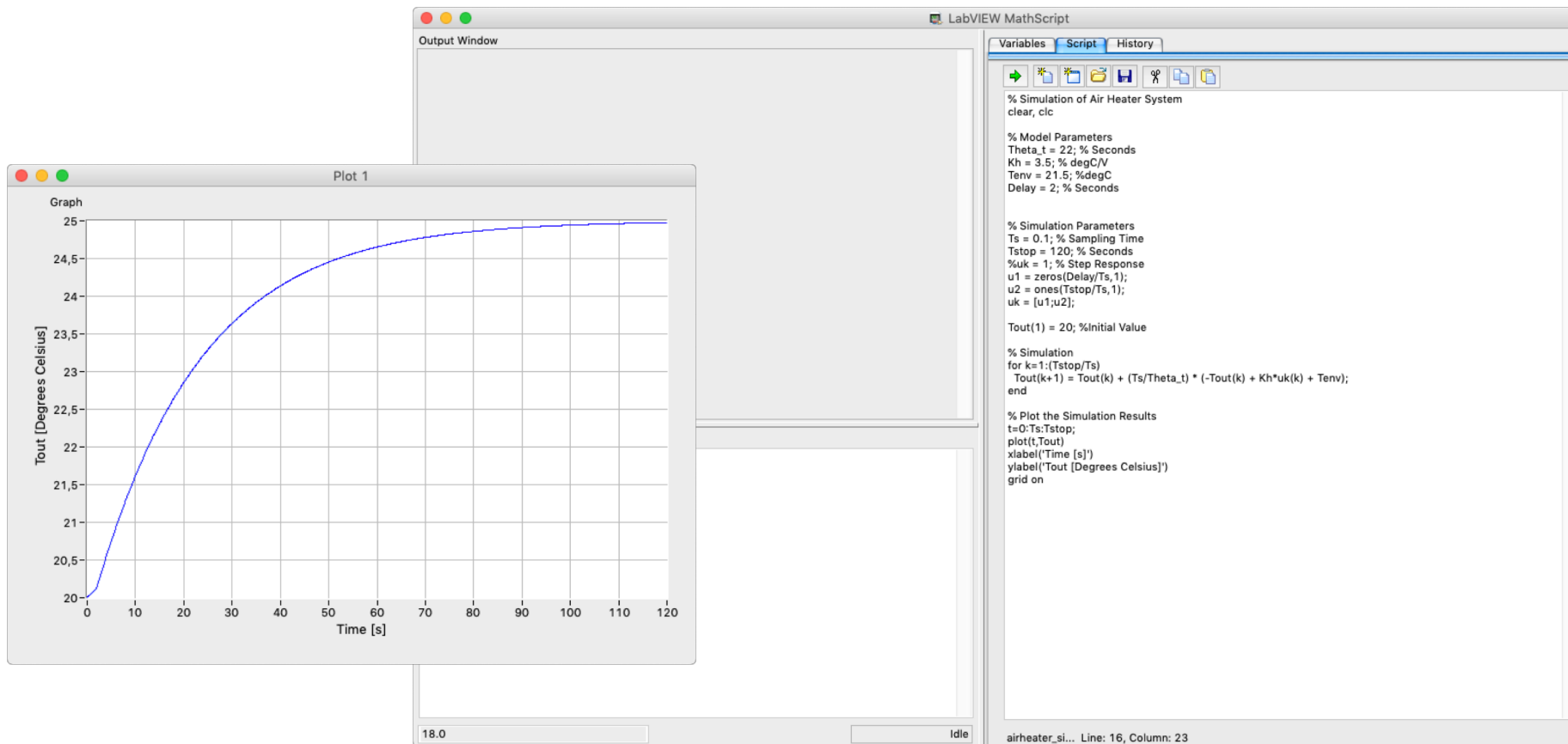
```
xlabel('Time [s]')
```

```
ylabel('Tout [Degrees Celsius]')
```

```
grid on
```



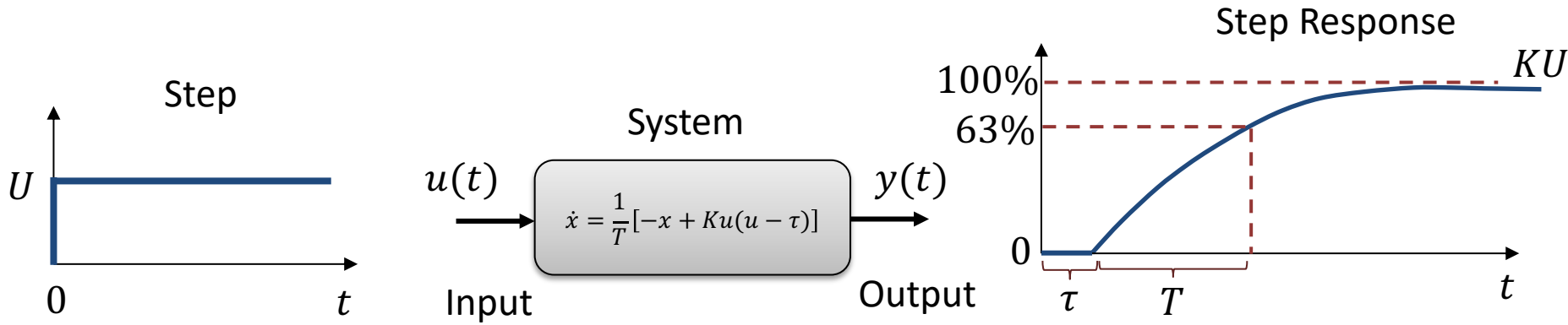
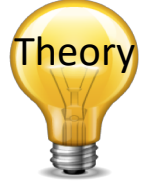
# MathScript Example with Time Delay



# Step Response 1. Order System with Time Delay

$$\dot{x} = \frac{1}{T} [-x + Ku(u - \tau)]$$

We see that the Air Heater System is a standard 1.order system with Time delay



$\tau$  - Time Delay  
 $T$  - Time Constant

# Hans-Petter Halvorsen

University of South-Eastern Norway

[www.usn.no](http://www.usn.no)

E-mail: [hans.p.halvorsen@usn.no](mailto:hans.p.halvorsen@usn.no)

Web: <https://www.halvorsen.blog>

