

GianAndrea Müller

# Air Massflow Sensor

## Lab Practice Control Systems Experiment Script

Institute for Dynamic Systems and Control  
Swiss Federal Institute of Technology Zurich

February 19, 2024



# Contents

<b>1 Introduction</b>	<b>1</b>
<b>2 Physical Principle</b>	<b>1</b>
2.1 Sensor principles .....	2
<b>3 Experimental Setup</b>	<b>4</b>
<b>4 System Model</b>	<b>4</b>
<b>5 Control Perspective</b>	<b>6</b>
<b>6 Simulink Intro</b>	<b>7</b>
<b>7 Simscape</b>	<b>7</b>
<b>8 Human in the Loop</b>	<b>9</b>
8.1 Overview of the GUI .....	9
<b>9 PID Controller</b>	<b>11</b>
<b>10 Mass Flow Estimation</b>	<b>11</b>
<b>11 System Identification</b>	<b>12</b>
<b>12 Validation</b>	<b>14</b>
<b>13 Conclusion</b>	<b>15</b>

## 1 Introduction

Please read through chapters 1-5 carefully, as a preparation for the lab session. You do not need to make a summary, however be prepared to discuss all homework tasks with the teaching assistant.

Imagine you are working for Sensirion. Initially an ETH spin-off, now the world's leading manufacturer of digital microsensors- and systems. After discovering a new sensor principle for air mass flow estimation you are now working in a small team, dedicated to show that your idea works! Your goal is to prove that it can be implemented and tuned for high precision as well as a small footprint. Finally you will adapt your sensor to large scale production and deliver proof of its performance, reliability and safety. At Sensirion all the needed competences for that process are available in house and you will cooperate with coworkers from all different departments helping your development with their expertise.

Right now you have received your first demonstrator from the automation department. Today you intend to deliver a proof of concept for the applied control algorithms.

## 2 Physical Principle

The change of energy over time, within a certain control volume can be described by the first law of thermodynamics for open systems:

$$\frac{dE}{dt} = \dot{Q} - \dot{W} + \sum \dot{m}_i \left( h_i + \frac{w_i^2}{2} + gz_i \right) - \sum \dot{m}_e \left( h_e + \frac{w_e^2}{2} + gz_e \right) \quad (1)$$

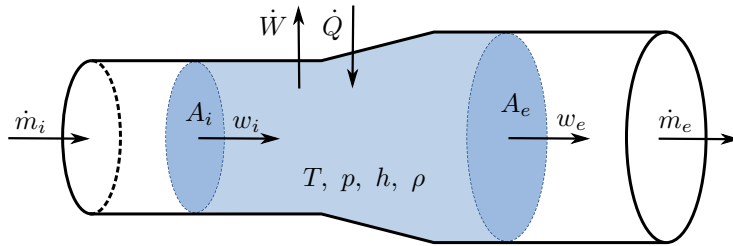


Figure 1: Conceptual control volume illustrating the application of the first law of thermodynamics.

$E$	Energy	[J]
$\dot{Q}$	Heat flow into the control volume	[W]
$\dot{W}$	Power input from work done on the system	[W]
$\dot{m}$	Mass flow into the system	[kg s <sup>-1</sup> ]
$h$	Specific enthalpy of the mass flow	[J kg <sup>-1</sup> ]
$w$	Velocity of the mass flow	[m s <sup>-1</sup> ]
$g$	Gravitational acceleration	[m s <sup>-2</sup> ]
$gz$	Potential energy of the mass flow	[J kg <sup>-1</sup> ]
$i$	Inlet quantity	
$e$	Exit quantity	

## 2.1 Sensor principles

**Homework:** Make sure to understand the main differences between a Venturi meter and the thermal measurement principle used in this experiment.

Measuring gas mass flow is possible in many ways. There exist a multitude of mechanical approaches featuring some moving part, i.e. a turbine, a propeller, a gear or even a piston, whose motion gives a measure of the amount of flow.

**Pressure based methods** do not require a moving part but rely on an orifice and measure the related pressure drop. Most notable among them is the Venturi meter.

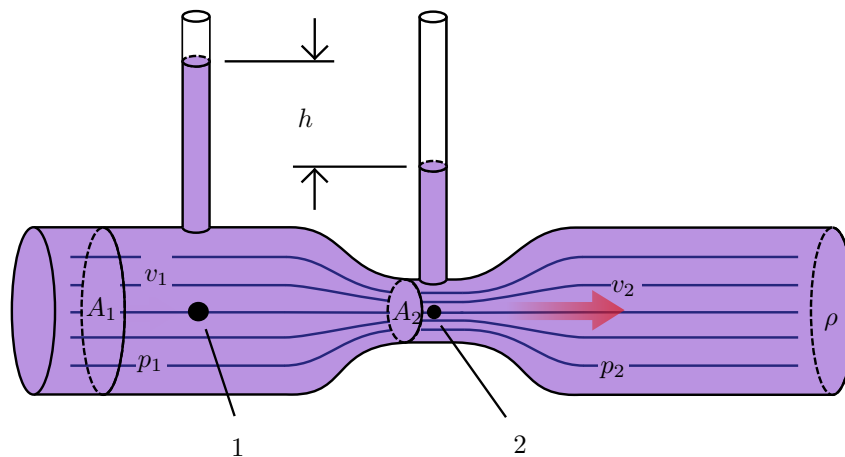


Figure 2: A Venturi meter. [Source: Wikipedia, Venturi Effect](#)

It is even possible to measure the flow of a gas by detecting particles moving with the gas by recording their passing through two subsequent laser beams and thus deriving the velocity of the gas carrying them. Those are **optical methods**. The same approach allows the implementation of particle measurements enabling determination of the air quality or even smoke detection.

Better applicable to fluids than to gases, but stunning in their beauty are **vortex methods**. They rely on placing a blunt body in the flow and recording the frequency of alternation of the generated vortex trail, also called Von Kármán's vortex street. This principle is actually applied in industry, check out this [link](#).

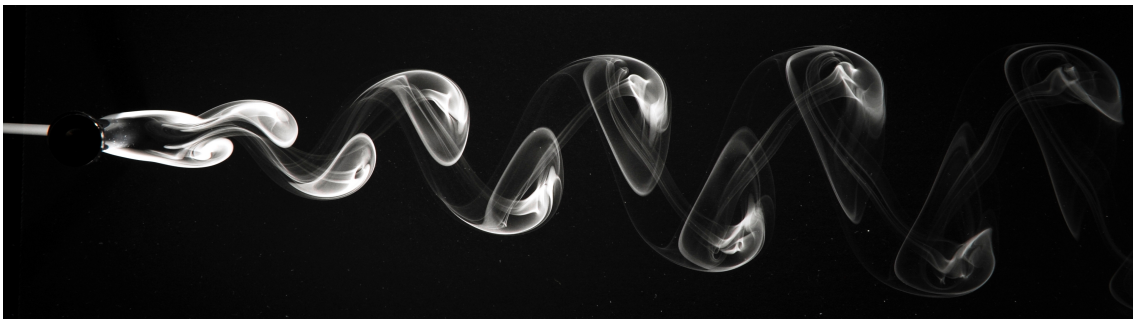


Figure 3: Image of Kármán's vortex street visualized with smoke in air flow. [Source: Wikipedia, Karman's Vortex Street](#)

Finally **acoustic methods** rely on the fact that the transit time of an acoustic wave through a fluid depends on the velocity of the fluid.

All the above mentioned methods are intriguing but lack the ability to directly measure the mass flow. They rather rely on a velocity or volume measurement, which in turn allow a mass flow estimate based on the assumed properties of the gas. This is where thermal mass flow meters and Coriolis flow meters come into play.

The latter relies on forcing the fluid through a oscillating tube section that deviates from its rotational axis. The radial acceleration of the fluid on its way away from and back to the axis results in a nonzero Coriolis force, which becomes measurable by the deformation of the tube and is in direct dependence of the mass flow.

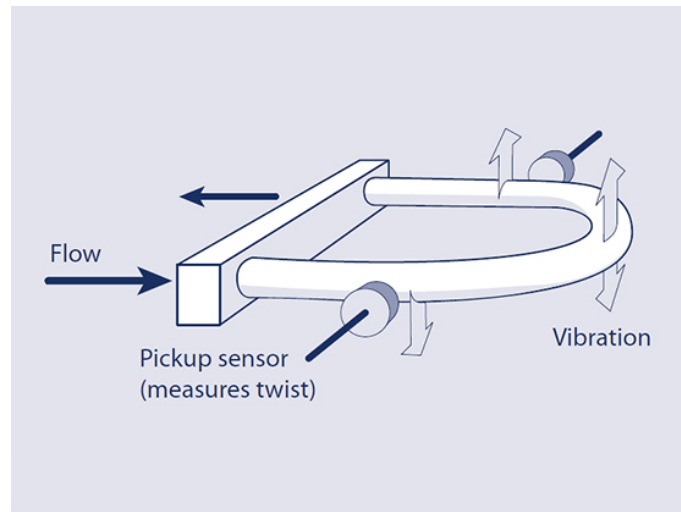


Figure 4: Operating principle of a coriolis mass flow meter. [Source: Bronkhorst](#)

The thermal measurement principle is in use for this experiment and will be covered in the following section.

### 3 Experimental Setup

A thermal mass flow meter uses a heater and a pair of temperature sensors. Knowledge of the specific heat capacity of the air at constant pressure and a measurement of the temperature difference over the heater at a certain power input allows to quantify the mass flowing through the channel directly. See figure 5 for a visualization of the concept.

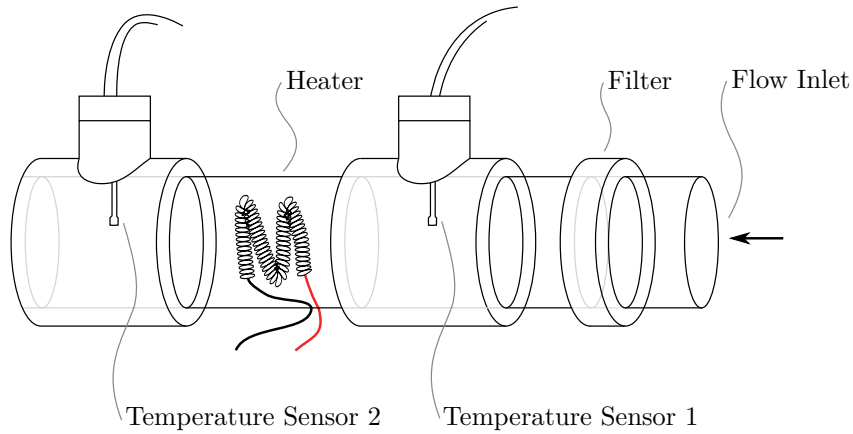


Figure 5: Schematic of the experimental setup.

### 4 System Model

To derive the simplified steady state system dynamics allowing a mass estimation the control volume is defined as seen in figure 6

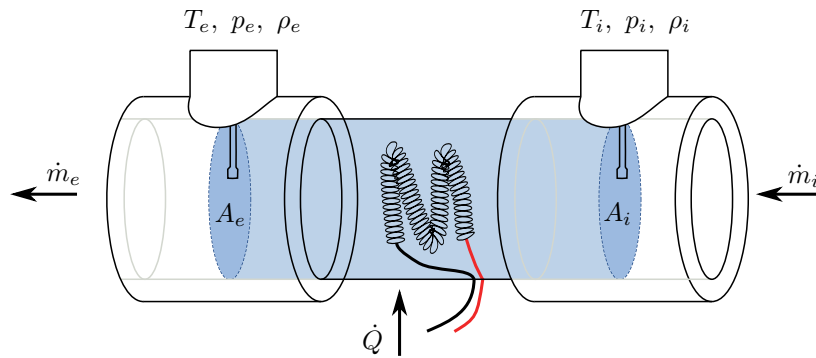


Figure 6: Control volume and physical quantities.

$$\frac{dE}{dt} = \dot{Q} - \dot{W} + \sum \dot{m}_i \left( h_i + \frac{w_i^2}{2} + gz_i \right) - \sum \dot{m}_e \left( h_e + \frac{w_e^2}{2} + gz_e \right)$$

**Homework: Go through all assumptions below.**

**For each one write down how realistic you think it is and provide a possible experiment to determine how much reality deviates from these idealizations.**

*Hint: We will be dealing with  $\Delta T$  of 6 °C to 15 °C and  $p \approx 1$  bar in the Tube*

**Assumptions:**

- The pressure drop from inlet to exit area is negligible.  
 \_\_\_\_\_  
 \_\_\_\_\_
- The heat flow through the pipe walls is very small.  
 \_\_\_\_\_  
 \_\_\_\_\_
- All considerations are made in steady state.  
 \_\_\_\_\_  
 \_\_\_\_\_
- The potential and kinetic energies of the fluid do not change within the control volume.  
 \_\_\_\_\_  
 \_\_\_\_\_
- The pipe is leak free, thus mass conservation is holding.  
 \_\_\_\_\_  
 \_\_\_\_\_
- The treated gas is assumed to be ideal.  
 \_\_\_\_\_  
 \_\_\_\_\_
- The heat transfer from coil to air is perfect and instantaneous.  
 \_\_\_\_\_  
 \_\_\_\_\_

Under the above assumptions equation 1 can be simplified to deliver a direct correspondence between the heating power, the temperature difference and the mass flow.

$$0 = \dot{Q} + \dot{m} (h_i - h_e) \tag{2}$$

Further, for an ideal gas, the enthalpy can be described as a function of the temperature, as long as the state change is isobaric.

$$\Delta h = h_e - h_i = c_p \Delta T \tag{3}$$

$c_p$  specific heat of air [J]

Finally the produced heat can be described directly by the power delivered by the electric heater and is a function of the related voltage and current.

$$\dot{Q} = P_{el} = UI = I^2 R \tag{4}$$

$P_{el}$	Electric input power	[W]
$U$	Electric voltage	[V]
$I$	Electric current	[A]
$R$	Electric resistance	[ $\Omega$ ]

Therefore the first approximation to the system dynamics can be described as follows:

$$\dot{m} = \frac{P_{el}}{c_p \Delta T} \quad (5)$$

Therefore a mass estimate can be made from the current temperature difference and the input power delivered by the heating coil.

## 5 Control Perspective

Without further knowledge about the system the power input can be freely chosen. This allows two basic approaches;

- Fix the power input,
- Fix the temperature difference.

To provide a good signal to noise ratio with the temperature measurement, it makes sense however to decide on a certain temperature difference. Just fixing the power input could result in an arbitrarily small temperature difference under large flow or vice versa in extreme heating in the opposite case. Especially the latter would violate the restrictive thermodynamic idealizations previously made.

This motivates the introduction of a temperature difference controller.

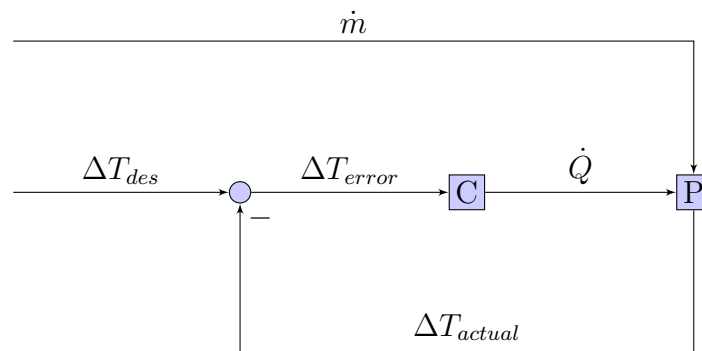


Figure 7: Visualization of the temperature difference control loop.

$\Delta T_{des}$	Desired temperature difference	[ $^{\circ}\text{C}$ ]
$\Delta T_{error}$	Temperature deviation	[ $^{\circ}\text{C}$ ]
$\Delta T_{actual}$	Actual temperature difference	[ $^{\circ}\text{C}$ ]

**This concludes the homework.**



## 6 Simulink Intro

The first part of this lab aims to show students some basic functionalities of Matlab Simulink.

Open the live script "MATLAB\_intro.mlx" and work through it:

1. Open the Simulink model by clicking on the button "open file"
2. Solve the exercise.
 

**Hint:** If you don't know what blocks you need, go to Simulation -> Library Browser
3. Save the File and close it
4. Check your answer by clicking on the button "Test input"
5. Once you have finished all tasks, click on "Finish exercise" at the bottom of the script and update your teaching assistant on your progress.

## 7 Simscape

Simscape is an extension of Simulink and allows to model the dynamics of physical systems directly instead of using the classic approach of modeling the underlying differential equations with Simulink. The basic principle is shown for a different system in figure 8. Note the following points:

- Just as in real life some sensors need to be mounted in parallel, others in series.
- For every physical domain there are sources, references, sensors and the elements defining the system dynamics, in this case a resistor.
- The solver configuration block must be present in any Simscape model. It tells Simulink how to solve the resulting equations.
- Simscape is based on information-sharing signals. Thus the blue lines in the plot simply connect what belongs together, but do not define a direction of the information flow. This is different for Simulink signals, which is why we need a 'PS-Simulink' converter to get one from the other.

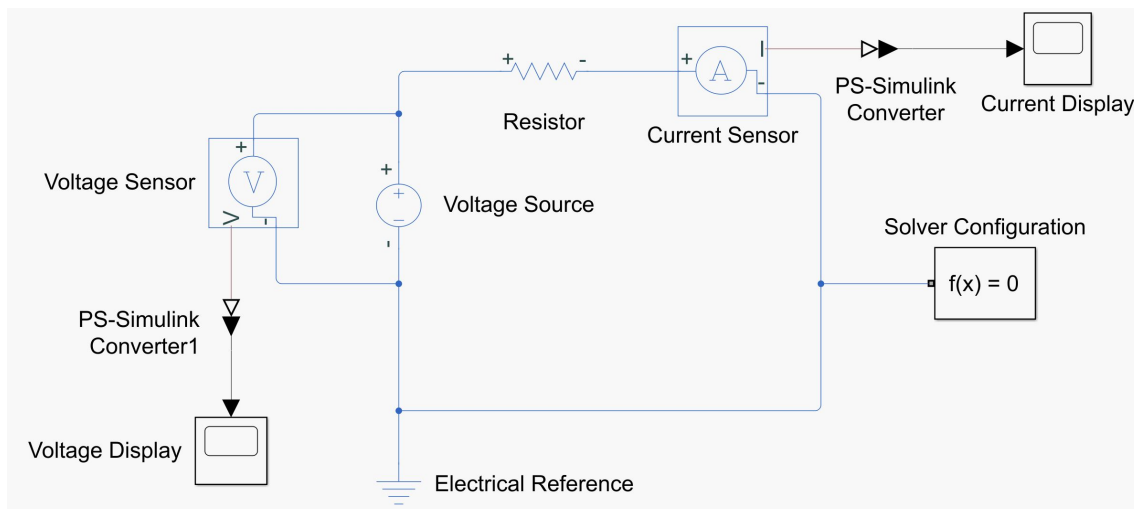


Figure 8: An example of a simple electrical network in Simscape.

**Exercise** Look into the Simscape model of the system and attach sensors to the system in order to measure mass flow and temperature difference by following the steps below:

1. Take a look at all the subsystems in the simulation. Answer the following questions:
  - Where does the air come from? Where does it go?
  - What does the  $f(x) = 0$  block do?
  - How is the voltage source programmed? Does it always deliver power? How could we change that?
  - Which blocks represent the heating coil? How exactly do they transfer heat to the air flow?
  - **Update the TA on your progress**
2. A reference mass flow measurement.
  - There are two commented out blocks and a commented out subsystem in the main simulation window. One of them is a “Mass and Energy Flow Rate Sensor”. You can find out more about it by double clicking it.

**Hint:** To enable it click right and choose “uncomment”.

**Hint:** To route a signal to different locations simply hold 'Ctrl' and drag an existing signal. It will fork.
  - Then you can start embedding the block into the model by connecting its ports to the rest of the simulation. Think about how the reference mass flow sensor (The green Sensirion Mass Flow Controller on the left of the setup) is connected to the flow path and do accordingly in the simulation.

**Hint:** There are already sensors in the Simulation measuring current and voltage. Try to find those to get some inspiration on how to connect sensors.

**Hint:** If you don't understand a block Matlab has extensive documentation online. Just search for the block name.

**Hint:** As a current sensor, a massflow sensor has to be mounted in series with the flow. Just like in the physical system the massflow has to go *through* the sensor in order to be measured.
  - Now use a “PS-Simulink” block to convert the sensor output, which is a Simscape signal into a Simulink signal.

**Hint:** To place a block whose name you know, just click somewhere on the white background of the Simulink window and start typing. Suggestions will appear.
  - Finally place a “Scope” block and connect it to the “PS-Simulink” output. This allows you to display the measured mass flow after running the simulation.

**Hint:** If you want to compare different signals, the scope block allows you to connect multiple inputs by simply dragging a signal close to it. It then spawns another input.
3. A mass flow estimate.
  - The second commented block is a “Pressure and Temperature Sensor”. Uncomment it.
  - Connect it to the simulation, such that it measures the pressure and temperature drop over the pipe.

**Hint:** In contrast to the massflow sensor, the pressure and temperature sensor measures a difference. In that way it is comparable to a voltage sensor.

- Now we need to hand over the temperature drop signal to the “Massflow Estimate” subsystem. This is done with a “Goto” block. For inspiration take another look at the voltage and current sensors and figure out how a Goto-From block connection works.

**Hint:** Also take a look at the “Massflow Estimate” subsystem, to find the corresponding “From” blocks

**Hint:** Goto and From blocks must be configured as ‘global’ in order to work across subsystem boundaries. This can be set when double clicking the block.

**Hint:** Goto and From blocks must have the same label to be connected with each other. They are marked in yellow if that connection is not set up correctly yet.

- Uncomment the “Massflow Estimate” subsystem. It contains certain constants as well as all “From” blocks you need to calculate a massflow estimate.

Note well that there is a tuning parameter `actual_transmitted_power`, which is to be multiplied with the electrical power, to represent a potential loss of power somewhere else in the system, we will use it later on for parameter fitting.

Implement the calculation of a massflow estimate, following equation 5.

**Hint:** By opening the Library Browser you can get an overview of the common Simulink blocks.

**Hint:** Check out equation 5 and think about which elementary mathematical operations you need to perform to calculate the right hand side. Then find corresponding Simulink blocks and connect them accordingly.

#### 4. Compare the reference mass flow to calculated estimate.

- Place a “Scope” block in the main MassflowModel.
- Connect the output of the reference massflow sensor you previously installed to it.
- Route the calculated massflow estimate from the subsystem back to the toplevel MassflowModel and drag it to the scope block. The scope block automatically sprouts a second input.

**Hint:** To hand signals through subsystem boundaries simply place an “Out” block within the subsystem. This will make the signal you connect to it available on the outside.

- Run the simulation by opening the top left tab “Simulation” and clicking the “Start” button.
- If the result does not match your expectations try finding where our model might go wrong.

#### 5. Update the TA on your progress

## 8 Human in the Loop

### 8.1 Overview of the GUI

**Exercise** Open the GUI window and read through the GUI description to familiarize yourself with its functionality.

1. Stop/Start measurement recording
2. Go back and forth between experiment pages
3. Reset plot windows
4. Switch to competition mode
5. Toggle heater output / massflow
6. Toggle reference step
7. Save the measurement buffer to a file
8. Current system status
9. Various sliders to control the setup

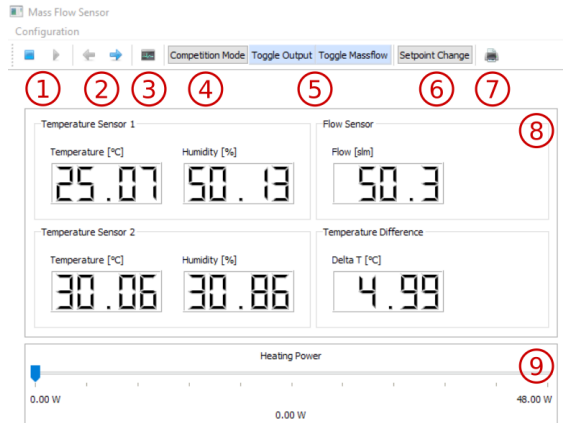
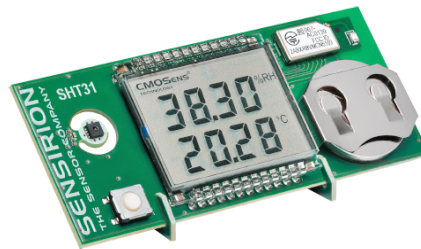


Figure 9: Central GUI elements

**Hint:** Plots are interactive! Try zooming and shifting by drag and drop.

**Hint:** Hovering over GUI elements will display hints in the lower left of the window.

**Exercise** Start up the physical system and on Experiment Page 1, try to keep the temperature difference at the desired level (in red) by varying the heating power by hand. You can also try reaching different setpoints by performing a reference step.



**Exercise** Win a Sensirion Humi-Gadget.

- Enable the competition mode.
- Make sure your target temperature difference is at the lower value, otherwise you will make your life harder.
- Start your trial by clicking the red button on the new widget. This will run the setup through a series of disturbance steps.
- Rinse and repeat until you feel like you achieved a sufficient score.
- **Update the TA on your progress** by showing your best score.

The competition mode grades your efforts according to the following equation,

$$\sqrt{\sum_{k=0}^N (\Delta T_{actual}[k] - \Delta T_{target}[k])^2}, \quad (6)$$

which implements the root mean squared error.

## 9 PID Controller

**Exercise** Switch to Experiment Page 2. This time you have a full PID controller at your disposal. Try out different settings and situations aiming to find one optimal combination of parameters and answer the following questions:

- How well does a simple P controller fare? What shortcomings does such a controller have?
- What is the purpose of the I part of the controller? Does it have disadvantages?
- Why would we need a D part? Is it necessary in this case?

**Exercise** See if the PID controller beats your performance.

- Turn on competition mode.
- Again be sure that the desired temperature difference is at the lower value, to have a situation comparable to the last exercise.
- Start recording the PID controller's performance with the red button.
- When finished click the printer symbol in the toolbar to save your run. This gives you the option to effortlessly compare your runs by executing `ComparePIDRuns.m` in Matlab. Feel free to adapt the script to your needs.
- **Update the TA on your progress** and be ready to discuss your answers regarding the above questions on PID control.

## 10 Mass Flow Estimation

**Exercise** After having set up a good control loop, the main question is: How well is our sensor performing?

- Keep your optimal controller settings from the previous exercise, turn on the mass flow and enable the controller output, such that the system can reach a stable steady state.
- Switch to Experiment Page 3. Now you can see the mass flow estimate as calculated by 5.
- Does the quality of the estimation meet your expectations? What problems do you see?
- A simple way of improving such an estimate is signal processing. Go through the following steps:
  - Locate the cutoff-frequency-slider and the FFT plot on the lower left of the screen. What information can you gather about the applied filter?
  - Test different filter settings to optimize the quality of your estimate.
  - Once you're satisfied go through different mass flow settings to test how fast the system responds to a change.
- Try a "setpoint change" now. Take a look at how that influences the frequency content of the flow estimate and the filtered estimate. Do you see how a step in the signal introduces a lot of high frequency content? If your filter performs well that should be much more prominent in the unfiltered signal. This also illustrates the delay introduced by the filter.

**Exercise** Now that we have powered up the test setup and have a feeling for how it works, we can test our previously made assumption, that the pressure drop over the heating coil is negligible. **Ask your TA to start the differential pressure measurement for you.**

## 11 System Identification

You now have knowledge of both the physical setup and the Simscape simulation. In a next step you will be fitting the simulation parameters to real world data making use of the Simulink Design Optimization toolbox.

For that reason we define a single input single output system from our model. We choose the input power of the coil as input and the temperature difference measured over the coil as output. The resulting model should capture the dynamics of the heater.

**Hint:** The input power is saved in a data field called “PWM”, which stands for pulse width modulation. This in turn is a simple way to get an analog output from a digital input. Read more on the subject here: [en.wikipedia.org/wiki/Pulse-width\\_modulation](https://en.wikipedia.org/wiki/Pulse-width_modulation).

**Exercise** Think about an experiment that will deliver useful data for identifying our system parameters.

1. Should we run our setup closed loop or open loop? Which scenario is more easily compared to the simulation?
2. How should we choose initial conditions? Which choice would be the easiest?
3. Would you like to identify only a steady state or rather see dynamics?

**Exercise** Execute your experiment and record the data.

1. Select either Experiment Page 1 (open loop) or Experiment Page 2 (closed loop).
2. Execute your experiment of choice and stop the recording when finished.
3. Save the measurement buffer to import it into Matlab later.

**Hint:** Check out section 8.1 to find the appropriate buttons for stopping the recording and saving it.

**Exercise** Set up the parameter estimator, following the steps below.

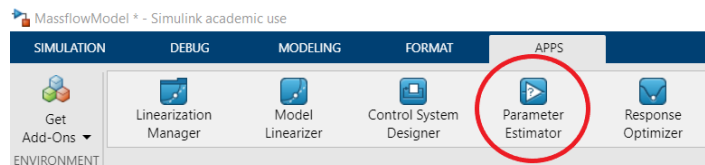
1. Make the simulation ready for accepting experiment data as input
  - Locate the “Scenario” block that provides the PWM input for the voltage source driving the heating coil. Delete it.
  - In its place connect an “Inport” block.
  - Leave the subsystem, going to the top level of the Simulation where now the “Voltage Source” subsystem should have another inport. Now place a second “Inport” block, connecting to the “Voltage source” subsystem. This allows us to hand over an input PWM signal given to the simulation to the voltage source block.

2. Make the simulation ready for accepting reference data for the output
  - Connect an “Out” block to the temperature difference signals, by simply forking the signal.

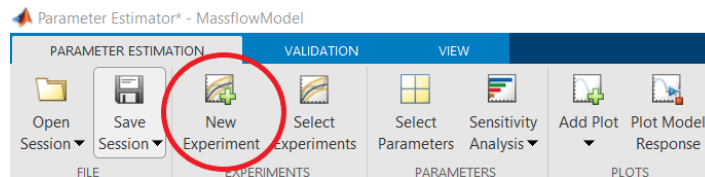
**Hint:** To route a signal to different locations simply hold 'Ctrl' and drag an existing signal. It will fork.

3. Import the measured data to Matlab by executing the first section of 'LoadDataForParameterEstimationAndValidation.m' and selecting the previously saved experiment data.
4. Set up a new “experiment”.
  - Go to the Simulink window.

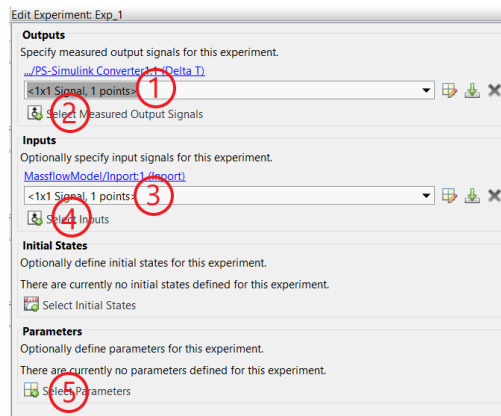
- Open the Matlab App “Parameter Estimator”.



- Start a “New Experiment”.



- Configure which signal in the experiment data are connected to which signal within the simulation and define the parameters to be estimated.

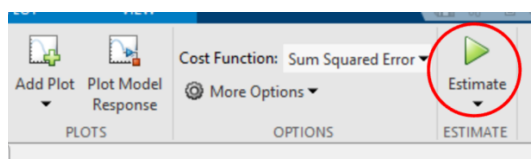


- Remember the workspace variables that were created when running the data loading script? Use those variables to define an output signal (the temperature difference), by typing directly in field 1.

**Hint:** A Matlab signal is made from a time vector and a data vector. Defining one is a matter of concatenating the two vectors as follows: `[time pwm]`.

- Field 2 now tells the estimator to which output in the simulation to connect the signal we just defined. Simply click it and select the previously defined output from the dropdown. Note that it might already be correctly selected by default.
- Field 3 and 4 do exactly the same for the input signal (pwm).
- Click field 5 to select all parameters that are not constant. These will be altered during the estimation such that the simulation fits the experiment data better.
- Scroll down and select “Plot and simulate”. This shows the measured data against the simulated data. Close the experiment editor by clicking “Ok”.

5. Run the estimation.



## 12 Validation

Now that we have fitted our model to the experimental data, we want to check if the estimated parameters are also valid for other heating power sequences.

**Exercise** Record another data set that is significantly different from the one you fitted your model with.

**Exercise** Import the data by executing the second section of 'LoadDataForParameterEstimationAndValidation.m'.

**Exercise** Validate your Estimation

1. Create a new experiment with validation data and select the same parameters as in the first experiment.
2. Copy the values for your parameters from your first experiment.
3. Press the button: `plot & simulate`
4. Does the data match your simulation?



## 13 Conclusion

The SFC5400 is the result of multiple years of development, during which measurement performance has been optimized while enabling large scale production and miniaturizing the whole design. Sensirion engineers gave some personal insight into that story:

- The **largest challenge** was setting up a reliable calibration sequence. The flow controller had to run through the desired flow range but setting the correct valve setpoint to achieve that range is hard with a completely uncalibrated device. The solution was to do a coarse calibration first and use that to produce a second more accurate run.
- Another issue was handling **deviating operating conditions**. To cope with temperatures different than used for the calibration a special compensation algorithm would make use of an additional onboard temperature sensor to predict the measurement deviation.
- The most prominent **source of noise** was caused by the air flow itself. Any turbulence would lead to highly increased noise in the measurement output. Therefore the current design aims to keep the air flow as laminar as possible. Besides trying to round all corners in the flow channels, introducing filtering elements is possible, but that might lead to additional pressure drops, which are to be minimized as well.
- To allow using the **same sensor for different flow regions** the setup is delivered as a DirectFlow variant, which means that the whole gas flow goes through the flow sensor. For higher throughputs there is a Bypass version that measures the flow on a narrow channel parallel to the main one.