Department of Electrical and Electronic Engineering
ELEN90066 Embedded System Design

Workshop One (W01)

Sensor Measurements

Welcome to Workshop 1 for Embedded System Design. In this workshop you will learn how to model the Sharp IR Sensor (GP2Y0A02YK0F) with Simulink, using Model-Based Design. This sensor is typically used as an IR range finder in embedded systems including ranging and object detection for robots, proximity sensing, and contact-less switches.

Once the sensor has been reliably modelled, its details can then be abstracted away as a "black box" component in the design, modelling and simulation of embedded systems.

Pre-workshop background

An Infra-red (IR) range finder uses a beam of reflected infrared light to sense the distance between the sensor and a reflective target. The range to an object is proportional to the reciprocal of the IR range finder's output voltage. The SHARP GP2Y0A02YK0F IR range finder, shown in Figure 1, is probably the most powerful sensor available for prototyping basic robotic systems. It is extremely effective, easy to use, affordable, very small, good range (centimetres to metres), and has low power consumption.

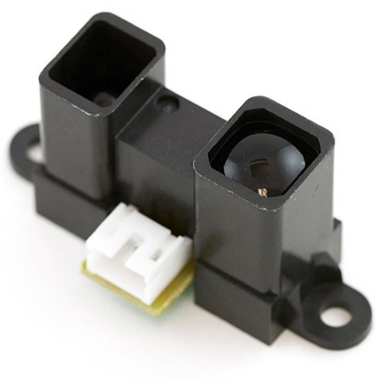

Figure 1: Infrared distance measuring sensor unit (GP2Y0A02YK0F)

The IR Range Finder works by the process of *triangulation.* A pulse of light (wavelength range of 850nm +/-70nm) is emitted and then reflected back (or not reflected at all). When the light returns it comes back at an angle that is dependent on the distance of the reflecting object. Triangulation works by detecting this reflected beam angle - by knowing the angle, distance can then be determined.

The IR range finder receiver has a special precision lens that transmits the reflected light onto an enclosed linear CCD array based on the triangulation angle. The CCD array then determines the angle and causes the rangefinder to then give a corresponding analog value that can be read by a microcontroller (via A/D conversion). Additional to this, the Sharp IR Range Finder circuitry applies a modulated frequency to the emitted IR beam. This ranging method provides some immunity to interference from ambient light, and offers indifference to the colour of the object being detected.

Have a look at the video IR Range Finder Interfacing Theory (9:59) to learn more about IR range finders (**note:** this is a different model but the same principles apply) including features, principle of operation, calculating range from sensor voltage based on the geometric principle of similar triangles, and calibrating the sensor with single- or multi-measurement techniques.

The values read from the sensor fluctuate quite a lot even with a fixed obstacle in front of the sensor, as shown in the oscilloscope trace in Figure 2

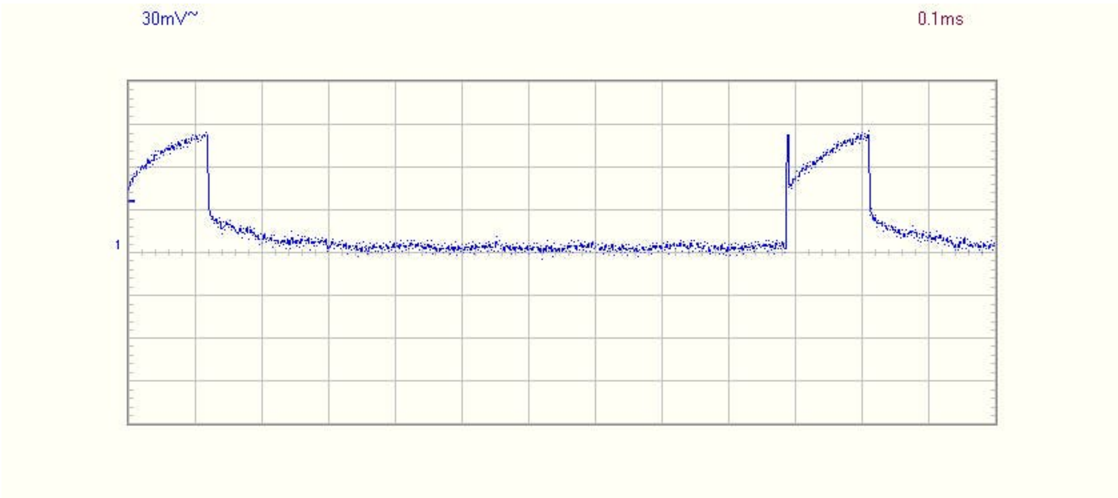

Figure 2: Infrared distance measuring sensor unit (GP2Y0A02YK0F) output

With a frequency of 1000 Hz there are peaks on the sensor output of about 90 mV. The sensor sends bursts of infrared lights every millisecond and outputs a voltage level based on which element of the sensor is struck by the reflected light. The peaks consists of one fast mode (the initial peak) and one slow mode and there is also some noise on the signal. The datasheet also shows that the output voltage level is non-linear according to the diagram in Figure 3.

In summary, the sensor has the following characteristics

- A non-linear function from distance to voltage level. The function goes from 0 cm to 150 cm which is the sensor maximum range.

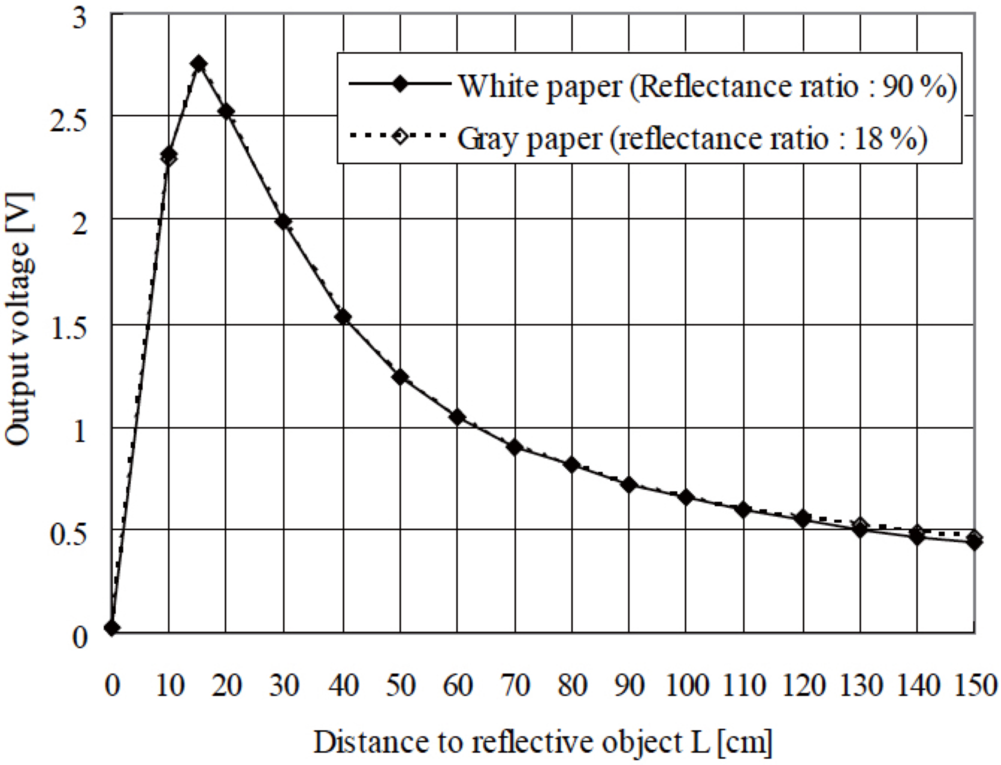

Figure 3: Distance measuring characteristics (output) (GP2Y0A02YK0F)

- Measurement noise
- Peaks in the response with fast and slow dynamics

In this workshop you will be constructing a model of this sensor and verifying its operation in Simulink.

Pre-workshop questions

1. What is the stated distance measuring range of the SHARP GP2Y0A02YK0F IR range finder sensor? (**Hint:** See data sheet on LMS).

2. (a) What is the typical output voltage range for the SHARP GP2Y0A02YK0F IR range finder sensor (i.e. the voltage outputs at both ends of its stated measuring range)?

   (b) Why would it be necessary to calibrate the sensor data directly from the sensor rather than use the data-sheet measurements?

Workshop Exercises

For this workshop, you will need the following software installed:

- MATLAB R2019a or later
- Simulink

In this workshop you will do the following

- Model the sensor **(TASK 1 - 5 marks)**
  - Construct the non-linear sensor function from distance to voltage level.
  - Model the sensor measurement noise.
  - Model peaks in the sensor response with fast and slow dynamics.
- Model a low-pass filter on the sensor output. **(TASK 2 - 5 marks)**
- Model sampling the analog sensor output signal. **(TASK 3 - 5 marks)**
- Verify the complete sensor model for a range of distances. **(TASK 4 - 10 marks)**

There are 25 marks available for successful completion of all tasks in this workshop, worth 1% of your final mark in the subject.

Task 1: Model-Based Design of the IR range sensor

Open MATLAB and create a new Simulink Model, starting with a blank model.

Constructing the non-linear sensor function

The non-linear distance-voltage function can easily be implemented with a 1-D Lookup Table in Simulink, shown in Figure 4 , found in the Library Browser under "Lookup Tables".

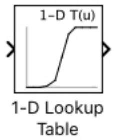

Figure 4: Simulink 1-D Lookup Table (LUT)

Insert the 1-D LUT into your project and double click on it. You will need to enter data points from the data sheet as vectors into the two data fields "Table Data" (for the Voltage values) and "Breakpoints 1" (for the distance measurement in m). You can further edit these values using the "Edit table and breakpoints" button.

Modelling measurement noise

The measurement noise can be modelled by a Band-Limited White Noise block, found under the "Sources" category in the Library Browser.

Insert this block and set the power parameter to `[0.0000000001]` and the sample time parameter to `0.0001/125` s, which are representative of measurements taken on the actual sensor.

Modelling the sensor dynamics

You now need to construct a transfer function for modelling the dynamics of the sensor, shown in Figure 2. For simplicity, we will chose to add just two transfer functions - one first-order transfer function for the slow dynamics and one second-order for the fast dynamics. A pulse generator will drive the transfer functions via the step response.

Based on measurements, the slow dynamics rises about 30 mV during the pulse. This means that the static gain should be about 0.03. The rise time for a first order function is:

$$t_{rise} = 0.67 \times (t_2 - t_1) \approx 0.67 \times 0.0001 \approx 0.000067s$$

With static gain $K = 0.03$ we get:

$$G(s) = \frac{(K/t_{rise})}{(s + 1/t_{rise})} = \frac{(K \times 14925)}{(s + 14925)} = \frac{448}{s + 14925}$$

These are good initial values and with some manual tweaking to make the step response look like the measurement we end up with:

$$G_1(s) = \frac{650}{s + 20000}$$

For the second order function we have:

$$G_2(s) = \frac{K_2(2\pi f)^2}{s^2 + 2\xi 2\pi f s + (2\pi f)^2}$$

As it should be faster than the first order function we can choose $f = 100kHz$ and $\xi = 0.3$. The gain is chosen as $K_2 = 0.06$ to ensure that the peaks are at about 90 mV, which mimics the real sensor output (see Figure 2).

Insert both of these transfer functions $G_1(s)$ and $G_2(s)$, as blocks into your Simulink project.

Connect a Pulse Generator block as an input to both of them to drive them, with the following parameters, which represent the sensor's operation

- Amplitude = 0.1
- Period = 0.001
- Pulse width = 13
- Phase delay = 0

Note: In a real world application, the sensor datasheet recommends to use a fairly large capacitor between Vcc and Gnd, as close to the sensor as possible, to limit the peaks during a burst. Thus, the amplitude of the pulse generator has been reduced by a factor of 10 to model this.

The slow and fast dynamics generate the response to a pulse generator (amplitude = 1 for reference) as shown in Figure 5.

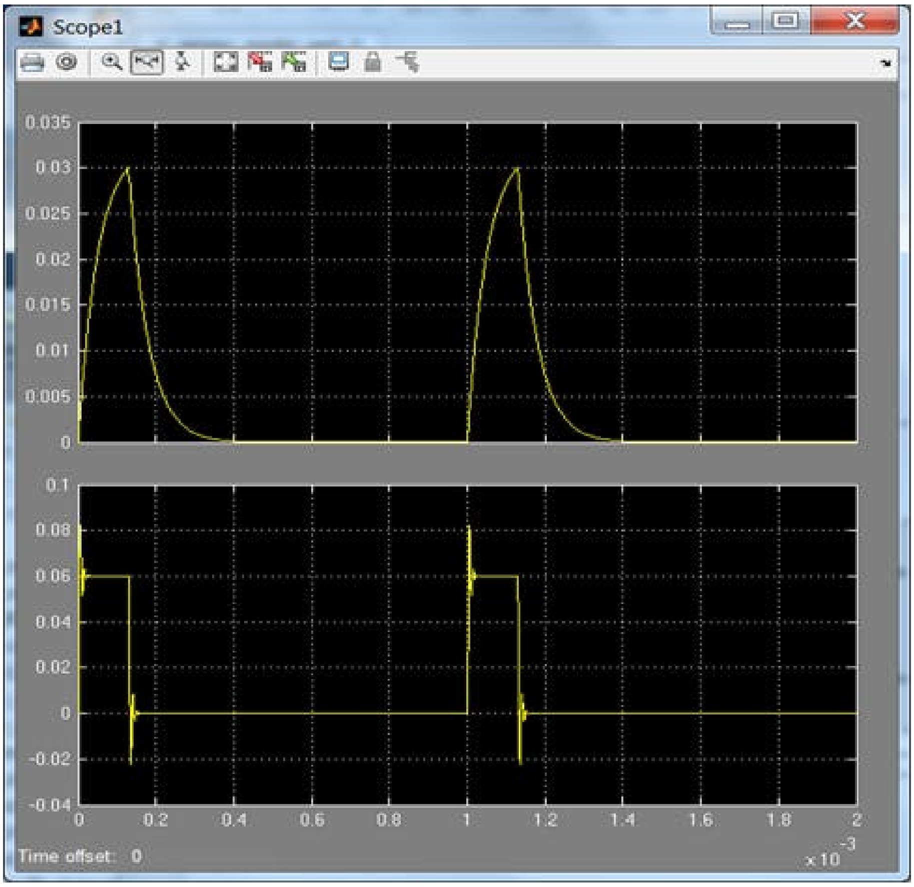

Figure 5: Sensor slow and fast dynamics

Putting it all together

Now you need to put everything together!

TASK 1 The components you have designed in the previous sections can simply be added together to form the sensor model. Add a constant block to represent the actual distance and a scope to the output so you can observe the model's behaviour. The resulting Simulink model should look like that shown in Figure 6.

Note : Your demonstrator will assess this task once you have finished it.

There are 5/25 marks for successful completion of this task.

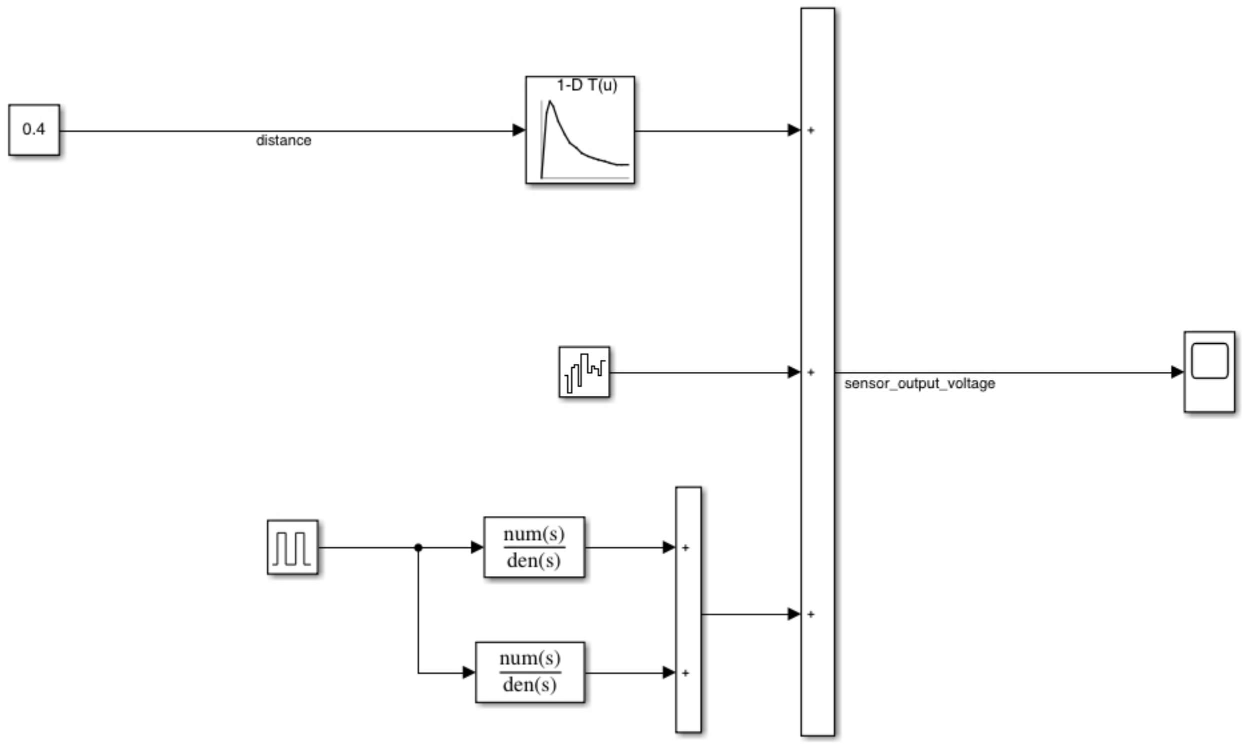

Figure 6: Model of the sensor (Task 1)

Task 2: Filtering the sensor signal

The data sheet for the sensor recommends to use a fairly large capacitor between V_{cc} and GND as close to the sensor as possible to limit the peaks during a burst transmission.

To further smooth out the sensor output we can add a simple passive filter, an RC-filter, to the circuit, shown in Figure 7, with transfer characteristic

$$V_{out}(s) = \frac{1/RC}{s + 1/RC} V_{in}(s)$$

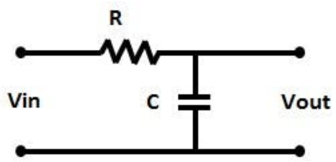

Figure 7: RC filter

This filter has a cutoff frequency of:

$$f_c = \frac{1}{2\pi RC}$$

To filter at least the fast dynamics, the cutoff frequency should be lower than 100 kHz. Also it can't be too low as changes in distance from the sensor would lag. Values of $R = 10\Omega$ and $C = 1\mu F$ are reasonable choices.

TASK 2 Add the low-pass RC filter to the model by creating a transfer function block and placing it at the sensor output. Connect a scope to the output of the filter and observe the signal for some varying distance values. You should observe a reduction in the size of the transient response in the output.

Note : Your demonstrator will assess this task once you have finished it.

There are 5/25 marks for successful completion of this task.

Task 3: Sampling the signal

An embedded system will usually have at its heart a microcontroller with ports that can read analog voltages. These port voltages are converted to a digital representation by an Analog to Digital (A/D) converter, before being able to be read as a (binary) register value by code executing on the microcontroller. You are going to model that conversion and incorporate it into the sensor model.

First the filtered continuous sensor value needs to be sampled, using the Rate Transition block, shown in Figure 8

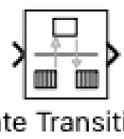

Figure 8: Rate Transition block

You can chose an output port sample time of 100 ms.

We will assume that the analog to digital conversion occurs with a default reference voltage (5V) and outputs values between 0 (0V) and 1023 (5V). This corresponds to 10-bit resolution. So, to convert the measured voltage we need to multiply the signal by 1023 and divide by 5.

The gain block doing the multiplication must also convert the signal to an unsigned 16-bit integer, to simulate its internal representation in the microcontroller, and saturates the signal to avoid wrapping for negative input voltages. This guards against the case when the input voltage is 0 and the noise is less than 0. Without this saturation, the result would be a very large number.

TASK 3 Complete the model of the sampled sensor signal by:

1. Inserting a rate transition block with output port sample time of 100 ms.
2. Adding some measurement noise by adding another Band-Limited White Noise block with representative values of the Noise power of `[0.000001]` and the sample time parameter to `0.1` s and summing this with the output of the rate transition block.
3. Adding a Gain block with :
   - Gain = `1023/5`
   - Output data type = `uint16`
   - The "Saturate on integer overflow" box checked.

Your model should look like that in Figure 9. Simulate your model to verify correct operation.

Note : Your demonstrator will assess this task once you have finished it.

There are 5/25 marks for successful completion of this task.

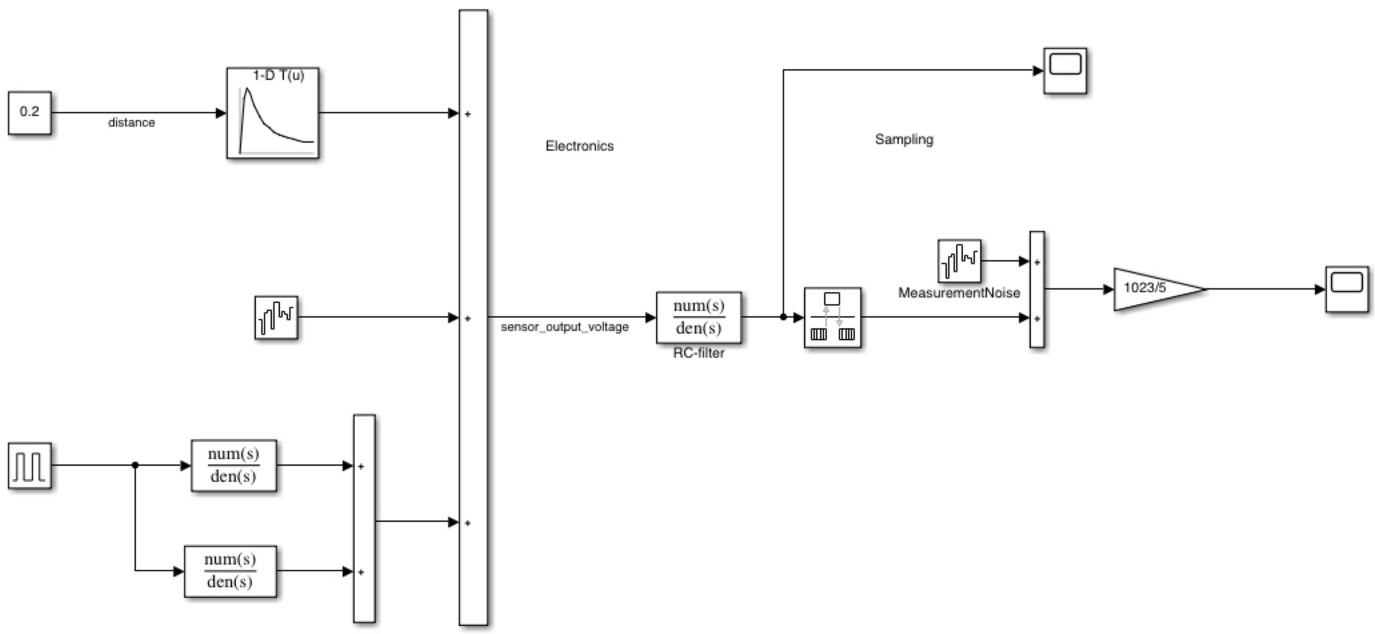

Figure 9: Model of the sensor with filtering and sampling (Task 3)

Task 4: Obtaining the distance

Reading the analog input gives a value between 0 and 1023 corresponding to 0 V to 5 V. There are many ways to convert this signal to a useful distance. You will be using a look-up-table because the distance is non-linear and a look-up-table is easy to calibrate.

The first step is to invert the look-up-table used for the sensor from before. This is problematic because the function is not monotonic when inverted. For example, the sensor gives approximately the same output voltage for a distance of 10 cm and 25 cm. To solve this we need to remove the first two values: 0 and 10 cm. The result is that only ranges between 20 and 150 cm could be measured. This is also the operating range of the sensor according to the specifications. This means that if we have an obstacle at 10 cm, the microcontroller software will think that it is at 25 cm. This has to be accounted for in the design of the rest of the system. Some ways to account for it are to stop before the minimum range, use another sensor for short ranges or place the sensor at the back of a robot facing forward.

In addition to inverting the table, we need to compensate for the sampling. The break points of the table contain voltages from 0.49 V to 2.75 V, but our read values are integers from 0 to 1023. Multiplying by 5 and dividing by 1023 should give us the correct values. However, that would give us a voltage between 0 and 5 and with integers that would not give sufficient resolution. Therefore we also multiply by 1000 to get the integer value in mV. We do the same thing with the distance and get the output range measurements in millimetres.

TASK 4 Add a 1-D LUT to complete the model of the sensor system as described in the section above and shown in Figure 10.

The output of your LUT should give the measured distance in mm.

Hint: You will need to multiply your voltage breakpoints by $1023/(5\times1000)$ as described above.

Simulate your model to verify correct operation for a range of distances. To further test the model, put a ramp signal in as the distance and check that output displays the correct distance.

Note : Your demonstrator will assess this task once you have finished it.

There are 10/25 marks for successful completion of this task.

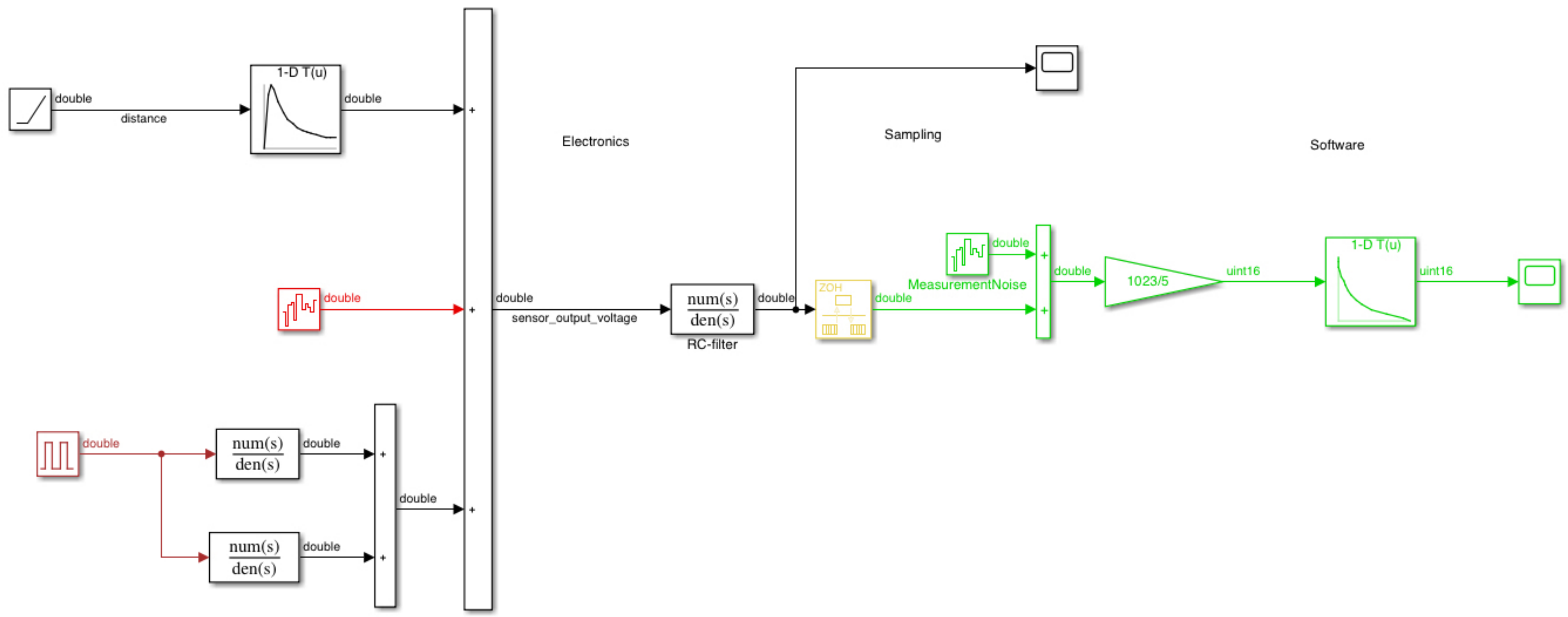

Figure 10: Complete model of the sensor (Task 4)