Smoothing Noisy Data in Distance Measurement

How can noisy measured distance data be smoothed to provide a reliable estimate of distance?

Contents
Initial Problem Statement  2  Narrative  3-8  Notes  9-11  Solutions  12-17  Appendices  18-21
Smoothing Noisy Data in Distance Measurement

Initial Problem Statement

There are many engineering applications where measuring the distance to an object is useful. One example is helping in a manual operation, such as parking a car, where bumper sensors let the driver know the distance to nearby objects. Another is in automated manufacturing where robots are used to handle and position items. The ability to measure distances allows the robot to perform such tasks without collisions.

The measured distance data supplied by a distance measuring device are often noisy, that is they do not give a constant value when measuring a fixed distance. How can the data be smoothed to provide a reliable estimate of distance?
NARRATIVE

INTRODUCTION

A typical infrared (IR) distance sensor is shown below.

![Image of infrared distance sensor](image)

The device is shown schematically below.

```
+5 V (VCC)
0 V (GND)
Output
```

Figure 2

For historical reasons the positive supply is commonly called VCC on many electronic devices; this stands for 'voltage common collector'.

GND is short for "ground".

There are three connections. Two are for power (VCC and GND). The third gives the output of the device in the form of a potential of between 0 and 5 volts above ground depending on how far away an object is from the sensor. The output is usually connected to a digital computer via a micro-controller which performs an analogue to digital conversion (ADC) and reports the values to a computer for logging and use.

See Note “Analogue to digital conversion” on page 9.
Multimedia
The video *Distance Sensor Video* is available to demonstrate the behaviour of a distance sensor.

The values from the ADC (range 0 – 1023) when an object is placed 20 cm in front of the detector are shown below:

Recorded ADC value (scale 0 - 1023; 1023 $\equiv 5$ V)

Discussion

1. What are the characteristics of these data?
2. Do you think these are good data?

Hint
Look at the vertical scale.
**Activity 1**
Quantify the quality of the data. What variation in recorded voltage does the recorded variation in signal represent (give results to 3 d.p.)?

**Discussion 3**
How could you improve the quality of the data? How could your idea be automated?

**Hint**
Do you need to keep all the values?

**Hint**
Do you need to use every individual value?

**Discussion 4**
Look at the data collected for an object placed 80 cm from the detector. Do they have similar characteristics to those collected when an object is 20 cm from the detector? Do they change your idea for improving the data?

---

**Figure 5**
Recorded ADC value (scale 0 - 1023; 1023 ≡ 5 V)

![Graph of data](image-url)
2. Using smoothed data

The company responsible for programming the micro-controller has suggested the following algorithms to smooth the data by averaging a number of measurements before sending the result to the computer.

Method 1

1. Record 10 consecutive measurements of the ADC value, \( a_1, a_2, \ldots, a_{10} \).
2. Find the average of the 10 values,
   \[ \bar{a} = \frac{a_1 + a_2 + \ldots + a_{10}}{10} \]
3. Send the value of \( \bar{a} \) to the computer.
4. Discard all recorded values.
5. Go back to step 1 and repeat.

Method 2

1. Record 20 consecutive measurements of the ADC value, \( a_1, a_2, \ldots, a_{20} \).
2. Find the average of the 20 values,
   \[ \bar{a} = \frac{a_1 + a_2 + \ldots + a_{20}}{20} \]
3. Send the value of \( \bar{a} \) to the computer.
4. Discard all recorded values.
5. Go back to step 1 and repeat.

When the measurement data are smoothed using the above averaging methods the following data are obtained from the micro-controller, where plots are shown for all data points, an average of 10 data points and an average of 20 data points.
Notice how the averaging of values reduces the variation in the signal and that averaging more points produces a smoother line.

The data points plotted above represent those received by the computer after they have been collected and averaged by the micro-controller. They are not plotted using the convention generally used in mathematics.

**Discussion 5**
What impact will the collecting of a number of data points and averaging the value before sending to the computer have on the ability of the device to record a distance?

**Activity 2**
The sensor response time is 39 milliseconds. This is the time between two successive measurements made by the sensor. Calculate the time interval between two successive measurements as sent to the computer when 10 data points are first measured then averaged. Calculate the time when 20 data points are averaged. How many updates or positions per second are sent for the three cases shown in Figure 6 (give your answers to 2 d.p.)?

**Discussion 6**
How could you improve the algorithm suggested above so that the same number of data points are averaged before sending a value to the computer but the time interval between sending the results is reduced? What is the shortest time interval that you can achieve? What are the limitations of your new method?
**Hint**
Step 4 in the algorithm discards all measured values after they have been used. Is this strictly necessary?

**Multimedia**
The multimedia resource *Smoothing Noisy Data in Distance Measurement Interactive* is available to help demonstrate how averaging a number of readings produces a smoothed result.
Notes

Analogue to digital conversion

At typical human engineering scales, variables in the physical world can be regarded as continuous, that is, they can take any value. Such variables are called analogue variables. Computers, however, store information digitally as a series of binary digits, or bits, that can take the value of 0 or 1.

When an analogue value, say the voltage of an output, is read by a digital device it must convert the value to the nearest digital equivalent. This process is called analogue to digital conversion and is carried out by an analogue to digital convertor (ADC). The accuracy with which the conversion takes place depends on how many bits the ADC uses. For example, suppose an analogue variable has a continuous range of 0 to 1. If the ADC uses 1 bit then the reading will be either 0 or 1. Any analogue value in between is rounded to the nearest integer.

<table>
<thead>
<tr>
<th>Digital value</th>
<th>Analogue equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>1</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Any analogue value in between is rounded to the nearest integer.

Suppose now that the ADC uses 2 bits. The possible combinations and representation of analogue values are

<table>
<thead>
<tr>
<th>Digital value</th>
<th>Analogue equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0.00</td>
</tr>
<tr>
<td>01</td>
<td>0.33...</td>
</tr>
<tr>
<td>10</td>
<td>0.66...</td>
</tr>
<tr>
<td>11</td>
<td>1.00</td>
</tr>
</tbody>
</table>

In this representation an analogue value in between is rounded to the nearest 1/3.

Most common ADCs use at least 8 bits and many use more. For example, an ADC used to digitise music for an MP3 player will typically use 16 or 24 bits.
Notes

Moving average

The data received by a computer after logging and averaging by a micro-controller are shown below.

- Recorded ADC value (scale 0 - 1023; 1023 ≡ 5v)
- Object distance = 20 cm

Notice that when averaging takes place, the micro-controller must first collect all the points it is going to average, take the average and send the value to the computer for use. This is why the 10 data point average is plotted after 10 individual measurements have been taken. In the above implementation the micro-processor then discards the measured data.

Figure 7
The use of the averaging of a number of points above is called a “moving average”. It is conventional to plot such data at the mid-point of the number of values being used in the average as this represents the value over the averaging range. Plotting the data in this way gives

Figure 8

Compare the two graphs. You can see that the averaged plots are moved to the left.
Solutions

Introduction

![Graph with data points and object distance = 20 cm]

Figure 9

Discussion 1 solution
The data show that the recorded ADC data are quite noisy. This means that the values change erratically with time. There are 98 data points and the majority (81 data points) are between 480 and 486. There are, however, occasional spikes above 486 (17 of them in this case).

Discussion 2 solution
The maximum recorded ADC value is 500, while the minimum is 480. This is a range of 20 which represents approximately a 4\% variation.

\[
\frac{20}{480} \times 100\% = 4.2\% \quad (1 \text{ d.p.})
\]

\[
\frac{20}{500} \times 100\% = 4\% \quad (\text{exactly})
\]

The vertical scale in the above graph gives the impression of a large variation. If the same data are plotted with a vertical scale ranging from 0 to 500 these percentage variations are more easily put in context.
Activity 1 solution

We may convert these to voltage values by noting that an ADC value of 1023 represents 5V. (Recall that an ADC device converts an analogue signal into a digital representation of the value.)

\[
\text{ADC} = 480: \text{Voltage} = 5 \times \frac{480}{1023} \approx 2.346 \text{ Volts (3 d.p.)}
\]

\[
\text{ADC} = 500: \text{Voltage} = 5 \times \frac{500}{1023} \approx 2.444 \text{ Volts (3 d.p.)}
\]

This is a range of about 0.1 Volts.

The majority of the data are between 480 to 486 which is a range of 6. This represents approximately a 1.25% variation and a voltage variation calculated as

\[
\text{ADC} = 480: \text{Voltage} = 5 \times \frac{480}{1023} \approx 2.346 \text{ Volts (3 d.p.)}
\]

\[
\text{ADC} = 486: \text{Voltage} = 5 \times \frac{486}{1023} \approx 2.375 \text{ Volts (3 d.p.)}
\]

This is a range of about 0.03 Volts.

You could conclude that the data are not as good as they could be as the majority of the values are within a narrow band while a few are outside of it, sometimes by a relatively large amount.

Discussion 3 solution

There are several possible methods of improving the quality of the data. One would be to exclude the spikes in the data. This is a process which a human is very good at given a set of data. Some analysis is required for this to be automated by a computer collecting the data.
When you look for outlying points you are probably imagining a band across the data and looking for points that are outside this. A computer can achieve a similar result by calculating the mean of all the data then excluding values outside a specified range. However, this requires some intervention to determine what a good value of the range is.

**Discussion 4 solution**

Applying the above approach to the data collected when the object is 80 cm from the detector could be problematical as there is much more variation in the data. If you applied the same acceptance range you would exclude many of the data. You can get around this by having an acceptance range that depends on the value but this is an added complication.

A simpler method is to just average all the data. The occasional spike will cause a bias in the results but this is not necessarily a problem as you still have to calibrate the sensor so that a given recorded voltage corresponds to a given measurement distance.

For the above solution you do not use every data point. Instead you collect a number of values, say 10 or 20, average them and only report the averaged value.

Note, there are more complicated methods that mathematically "filter out" rapid changes in the voltage data so that only the slow variations that are likely to arise from a genuine change in distance to an object are left.

You can also achieve this using electronics by connecting a capacitor and resistor to the signal line as follows

![Figure 12](image)

The effect of the resistor and capacitor is that rapid changes of the output signal are filtered out leaving only the underlying response.
2. Using smoothed data

**Discussion 5 solution**
When the micro-controller collects data for averaging prior to sending to the computer, the frequency with which measurements are supplied to the computer is reduced. If 10 data points are averaged by the micro-controller the computer will receive one item of data over a time interval that would previously have supplied 10.

**Activity 2 solution**
When every data point measured is sent to the computer, the time between successive values received is 39 milliseconds. If 10 data points are averaged the total time between successive values received is

$$t = 10 \times 39 = 390 \text{ (milliseconds)}$$

If 20 data points are averaged the total time between successive values received is

$$t = 20 \times 39 = 780 \text{ (milliseconds)}$$

To find the number of position updates per second, you must first convert the time between updates into seconds then find the reciprocal.

For no averaging: 

$$N = \frac{1}{39 \times 1 \times 10^{-3}} = 25.64 \text{ (2 d.p.)}$$

about 26 updates per second.

For average of 10: 

$$N = \frac{1}{390 \times 1 \times 10^{-3}} = 2.56 \text{ (2 d.p.)},$$

about 2.6 updates per second.

For average of 20: 

$$N = \frac{1}{780 \times 1 \times 10^{-3}} = 1.28 \text{ (2 d.p.)},$$

about 1.3 updates per second.
Discussion 6 solution
This algorithm is easy to implement but inefficient with data as all the points used are discarded after a single use. This is shown graphically below.

An alternative approach is to average the last 10 values from the ADC. As a new value becomes available, the oldest value is removed from the end of the list and the new value placed on top of the list. This is shown schematically below for an average of 10 values.

Just before a measurement is available

\[
\bar{a} = \frac{a_1 + a_2 + \cdots + a_{10}}{10}
\]
Just after a measurement is available

\[ a_{11} \]
\[ a_{10} \]
\[ a_{09} \]
\[ a_{08} \]
\[ a_{07} \]
\[ a_{06} \]
\[ a_{05} \]
\[ a_{04} \]
\[ a_{03} \]
\[ a_{2} \]

Oldest value discarded.

\[ \bar{a} = \frac{a_2 + a_3 + \cdots + a_{11}}{10} \]

Graphically this is

![Graph showing smoothing of noisy data](image)

Figure 14

The advantage of this approach is that a new average value can be sent to the PC from the microcontroller with the same frequency with which new measurements become available. For the case of averaging 10 measurements therefore, the time between data points received by the PC is 39 milliseconds rather than 390 milliseconds.

Note, this approach can only work when the table of values has been filled. This means there will be a delay of 390 milliseconds which the first 10 readings have been taken and averaged. After this a new 10-point average can be provided every 39 milliseconds. The time of 390 milliseconds represents the start-up time of the device.

**Multimedia**

The multimedia resource *Smoothing Noisy Data in Distance Measurement Interactive* is available to help demonstrate how averaging a number of readings produces a smoothed result. It also shows the difference between a fixed block and rolling average.
**Appendix 1**

**using the interactives**

**Smoothing Noisy Data in Distance Measurement Interactive**

The multimedia resource Distance Measurement Interactive is available to help demonstrate how averaging a number of readings produces a smoothed result.

The resource shows an example set of data points captured by the ADC directly from the device. When the “block average” option is selected the slider at the bottom can be moved to change the number of data points that are averaged by the micro controller before sending a result to the PC.

![Figure 15](image1.png)

![Figure 16](image2.png)
Notice that, in general, the data sent to the PC have less variation when averaging is used. Also note that the data points are spaced further apart, showing how the number of readings in a given time is reduced as a result of processing the data. For example, in the above there are 8 data points sent rather than 98. However, the data have much less variation.

If you placed the mouse pointer over one of the averaged points, the individual values that are used to calculate the average are highlighted.

![Figure 17](image1.png)

**Figure 17**
When the “rolling average” option is selected the slider at the bottom again changes the number of data points that are averaged by the micro controller before sending a result to the PC. However, this time average is taken over the \( N \) measurements made, where \( N \) is the number of data points to use in the average. Number of data points that are averaged by the micro controller before sending a result to the PC.

![Figure 18](image2.png)

**Figure 18**
Again the averaged data have less variation. In this method though, more data is supplied to the PC as data is re-used. Also, the first data point is not made available until a sufficient number of data points have been collected. This is the start-up time for the device (note it is the same time that elapses before the first block-average data point becomes available).

Again, placing the mouse pointer over one of the averaged points shows the individual values that are used to calculate the average.

If the two averaging results are super-imposed you can see that the rolling average has coincident points with the block average. However, it also provides values in between which shows some variation between two adjacent coincident points.
Appendix 2

mathematical coverage

Use algebra to solve engineering problems
- Be able to evaluate expressions
- Understand and work with percentages

Use statistics to solve engineering problems
- Understand that data collection may require sampling
- Be able to describe the main features of a displayed data set
- Be able to extract numerical information from a displayed data set
- Be able to compare data sets