An analysis of simple digital filters in robotic environments

Daniel Maximilian Swoboda[1*]

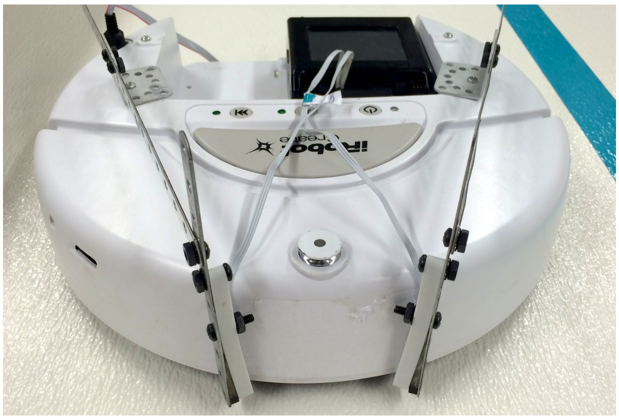
1 Höhere Technische Bundes Lehr- und Versuchsanstalt Wiener Neustadt, Abteilung Informatik (Secondary Technical College Wiener Neustadt, Department for Computer Science)
*i12032@student.htlwrn.ac.at

Abstract

This publication gives an overview of simple digital filters, their characteristics and their applications in robotic environments. There are many use cases for digital filters and filters in general. This publication focuses on their capabilities in noise reduction, the removal of distortions and general signal enhancement.
While there are many highly improved and mathematical complex digital filters, even very simple mathematical operations can be used to get sophisticating results. The introduction and analysis of these simple mathematical operations are the subject of this publication.

The first part focuses on the characteristics of digital filters in general, what their advantages are, how they are implemented and where they are commonly used. The second part introduces four basic digital filters and explains both their mathematical attributes as well as how they affect the data passed through them. The experimental part shows how the filters were experimentally tested and explains the process of data analyzation whereas in the analyzation part the results of the analyzation for each filter are introduced. In the last part the overall results are briefly discussed and suggestions for further research on this topic are made.

Keywords: digital filter techniques, noise reduction, simple digital filters, robotics



Graphic 1. The robot that was used in the experimental assembly with the sensor-mount clearly visible

Introduction

Digital filters are an important part of signal processing. They allow the reduction or enhancement of certain aspects of a signal or the removal of so called artifacts [1]. Digital filters work with sampled, discrete-time signals which is one of the major differences compared to analogue filters. By definition a digital filter is a mathematical operation which takes one sequence of numbers as an input and produces one modified sequence of numbers as an output [2].
When working with non repetitive signal sources like sensors in changing environments it’s often the case that there are distortions within the signal. These distortions can be caused by multiple things like voltage peaks in the sensor, fluctuations in light or temperature or just bad build quality of the sensors or other components [3].
Even the most simple filter could enhance the signal quality by reducing noise. Especially in robotics simple filters might be considered to improve the quality of the output signals from sensors.

1. Characteristics of digital filters

Digital filters work with discrete time signals which means that in a periodical sequence the signal consists of single impulses which are also an representation of the time. The number of values per second is equivalent to the sampling rate [2]. Another characteristic of a digital filter is that the resolution of the signal is finite and depends on the applied word size [4].

1.1. Advantages of digital filters

Digital filters feature a variety of advantages compared to analogue filters [1]:

- They can be implemented on the software side. Hence, no special hardware is needed.
- There is no drift due to the digital working principle.

- They are adaptive and therefore can adapt to the signal that's passed through them.

1.2. Implementation of digital filters

In contrast to analogue filters, digital filters can be implemented on both, the software and the hardware side. Both types of implementations depend on an analogue-to-digital-converter which can be found on almost any robotic controller since this is a critical part and is necessary to read data from sensors [5].
Software implementations of digital filters are normally as simple as a function which is fed with the data from a sensor and performs different mathematical operations to alter and modify the given data and return the result which can be further used in the process [2].
Hardware implementations usually consist of an circuit with logic chips to model the mathematical operation of the filter [5].

1.3. Common use cases for digital filters

Digital filters can be found in consumer products like digital sound recording applications, mobile phones, MP3-players, digital cameras or digital receivers as well as in high tech applications like robots, satellites, space probes and basically any application where signals are processed because of there flexibility [6].

2. Simple digital filters and their characteristics

While there are some highly advanced, specialized and complex filters, when focusing on simple filters fundamental mathematical concepts like averages can be used as a filter. Especially when working with low end computers like the ones used in educational robotics and computer science programs these simple approaches are important to consider.

There are 3 types of filters introduced here. The first type makes use of averages to filter the values. The second type uses a mathematical definition (the median) as a base while the third one makes use of polynomial functions.

2.1. Arithmetic mean filter

The arithmetic average is the most basic filter introduced here. It's the sum of a set of given values divided by the number of values within the set. [7]

$$y_n = \frac{1}{n}\sum_{i=1}^{n} x = \frac{1}{n}(x_1 + x_2 + \cdots + x_n)$$

The resulting signal is smoothed out to a grade equal to the number of values within the given set. This can drastically decrease the reliability of the signal.

2.2. Median filter

In difference to the arithmetic mean filter, the median filter doesn't make use of an global average. Instead the given set of *n* numbers gets sorted, afterwards the value at the position *n*/2 is determined and returned as the result of the filtering process, if the number of values is even the arithmetical mean of the the two middle values is used [8].

The median filter results in a very smoothed out signal because distortions far off the norm are completely ignored in the process. One major problem with the median is the sorting process which can slow down the calculation process of the median based on the size of the set.

2.3. Moving average filter

The moving average is a very commonly used filter [10]. Although there are different variants like the weighted moving average or the cumulative moving average only the simple moving average will be presented.
In difference to the arithmetic mean, where the average of a set of *n* numbers is calculated, the moving average takes the last result, adds the new incoming value and divides them by two [9].

$$y_n = \frac{1}{2}(y_{n-1}+x_n)$$

This leads to an improvement of the smoothing over time and works almost without delay. Most distortions will be filtered out this way but on an impulse like signal this filter technique is non-sensitive [10].

2.4. Savitzky-Golay filter (SG filter)

The most advanced and CPU-intensive filter introduced in this publication is the Savitzky-Golay filter. Originally used in spectroscopy, the filter works by using an polynomial function of the *n*-th grade [11]. The grade chosen has high influence on the actual outcome and can be chosen based on the requirements [12].

$$y_n = \frac{1}{35}(-3 \times x_{n-2} + 12 \times x_{n-1} + 17 \times x_n + 12\, x_{n+1} - 3 \times x_{n+2})$$

The signal passed through an Savitzky-Golay filter will almost remain it's basic form with strong reduction in noise but not as strong as with the moving average for example. The major difference is that impulse like signals won't decay over time. The SG-filter has a delay equal to the time it needs to get a number of data points equal to the chosen grade.

3. Experimental analysis of filter capabilities in robotic environments

3.1. Experimental assembly

To analyze the introduced filters a large set of real sensor data had to be gathered. An assembly consisting of an KIPR Link robotics controller, an iRobot Create and two infrared light reflection sensors to measure the reflectiveness of an rough PVC surface was used. The surface was chosen in order to get differentiating data so that the effects of the filters to fluctuating data sets could be determined.

For the first set of data the room was darkened and external sources of infrared light were removed from the assembly so that they could not interfere with the sensors. The subsequent tests were done under more realistic conditions with ambient light and possible infrared light emitters. Overall 10 different data sets on different parts of the surface and under different light conditions were collected to get a comprehensive overview of the filters on both, signals gathered in controlled environments as well as signals gathered in uncontrolled environments as they would be expected in robotics scenarios.

3.2. Analyzation process

The constructed robot drove with a speed of 5cm/s for 10s covering a distance of 50cm while gathering a point of data every 10ms. The collected 1000 data points of each test run were analyzed.

For the analyzation the introduced filters were implemented in Python using the NumPy and SciPy libraries. By passing the data through the introduced filters and visualizing the results using MatPlotLib, observations on the effects of the filters on the data could be made.

4. Data analysis

4.1. Visualized Data

The graphical visualizations represent parts of the data gathered with non-dimmed ambient light and on a plain part of the surface. The selected part was chosen because the input signal in this part has attributes that highlights the capabilities of each filter in an optimal way.

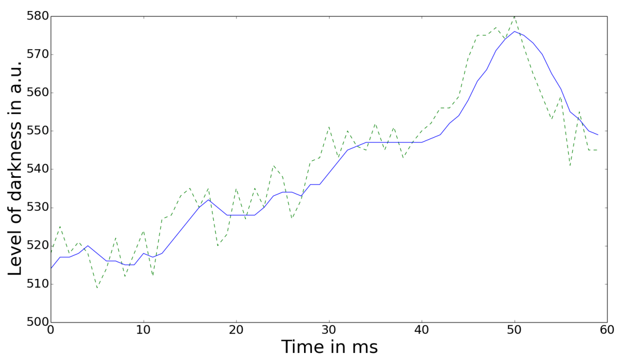


Chart 1. Input data (green, dashed) and arithmetical mean of the input data (blue)

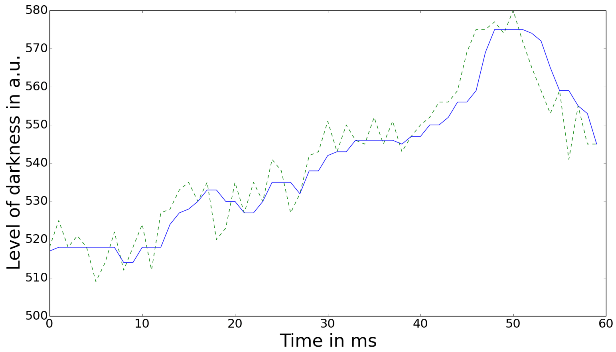


Chart 2. Input data (green, dashed) and median of the input data (blue)

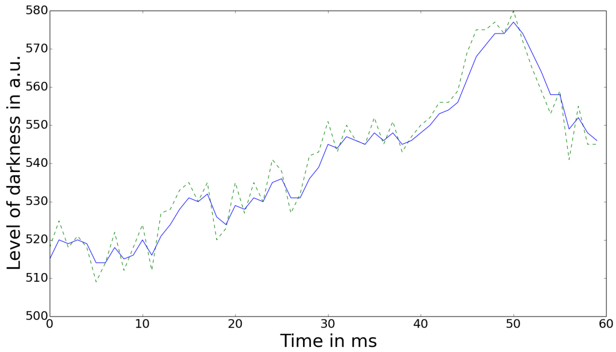


Chart 3. Input data (green, dashed) and moving average of the input data (blue)

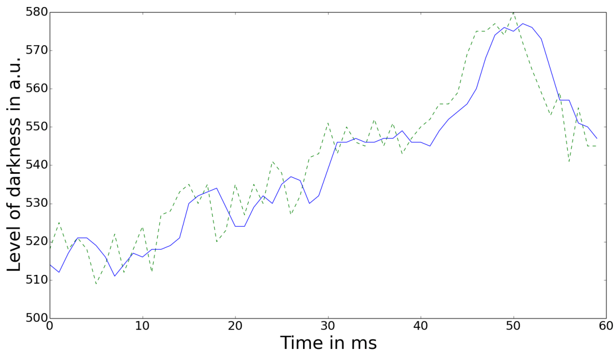


Chart 4. Input data (green, dashed) and Savitzky-Golay filtered data (blue)

4.2. Effects on the data

One result which covers all filters is that the range between 30 and 40 which seems to be undesired fluctuations is smoothed out. They all do a pretty good job in eliminating these noises. Another observation that can be made is that all the filters except for the moving average add a delay to some extent. One reason can be that these filters need, in the implementation used, a subset of 5 values before being able to perform any filtering .

It is noticeable that in practical use both, arithmetical-mean filter and median filter, have a similar effect on the signal passed through. Because of this the arithmetical mean should be preferred, as it requires less computational power and therefore is faster.

When comparing the moving average and the Savitzky-Golay filter to the arithmetical average, it can be noticed that, as expected, the results aren't as smooth.

For Savitzky-Golay it can be said that it's shape has more in common with the original signal except for the removal of noise while the moving average has a bigger impact on the shape.

Results

The results of the analysis give a big insight in the way the introduced simple digital filters modify an input signal to reduce noise. It shows that the most basic digital filters can enhance signal quality overall and therefore make operations that depend on this data much more reliable. They also show that the filter applied should be carefully chosen based on the specific use case.

In future the effects of the introduced filters on other signal sources can be analyzed as well as hybrid filters which make use of more than one filter or additional mathematical operations on the input signal. What also can be done is to analyze which filters fit best to the specific application.

Acknowledgements

The author would like to thank Dr. Michael Stifter for sharing his knowledge and his helpful support throughout the work on this publication, his helpful input on analyzation methods and for suggesting the Savitzky-Golay filter as an additional filter to think of; Daniel Honies, Christoph Käferle, Markus Pinter, Raphael Weinfurter and Florian Ungersböck for their support during the work on this publication and their help with the experimental assembly.

References

1. Georg Dorffner, "Digitale Filter" (german), http://www.meduniwien.ac.at/user/georg.dorffner/lv/Signal2.pdf, online presentation, 2008, accessed March 28th, 2015

2. J.O. Smith, "Introduction to Digital Filters with Audio Applications", https://ccrma.stanford.edu/~jos/filters/, online book, 2007 edition, accessed March 28th, 2015
3. Tom O'Haver, "A Pragmatic Introdction to Signal Processing", http://terpconnect.umd.edu/~toh/spectrum/SignalsAndNoise.html, online essay, 2015, accessed March 28th 2015
4. Steven W. Smith, "The Scientist and Engineer's Guide to Digital Signal Processing" Chapter 3, http://www.dspguide.com/CH3.PDF, online book, 1997 edition, accessed March 28th, 2015
5. Stephen Roberts, "Signal Processing and Filter Design" Lecture 5, http://www.robots.ox.ac.uk/~sjrob/Teaching/SP/l5.pdf, online book, 2003, accessed March 28th, 2015
6. Steven W. Smith, "The Scientist and Engineer's Guide to Digital Signal Processing" Chapter 1, http://www.dspguide.com/CH1.PDF, online book, 1997 edition, accessed March 28th, 2015
7. Eric W. Weisstein, "Arithmetic Mean", http://mathworld.wolfram.com/ArithmeticMean.html, web resource, accessed March 28th, 2015
8. R. Fisher, S. Perkins, A. Walker, E. Wolfart, "Median Filter", http://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm, online book, 2003, accessed March 28th, 2015
9. Steven W. Smith, "The Scientist and Engineer's Guide to Digital Signal Processing" Chapter 15, http://www.dspguide.com/CH15.PDF, online book, 1997 edition, accessed March 28th, 2015
10. "Programming Low End Robots", B. Tiefengraber, C. Jung, and M. Stifter, Physics Competitions Vol. 15 No 1 & 2 2013, p. 33-40.
11. Ronald W. Schafer, "What is a Savitzky-Golay filter", http://www-inst.eecs.berkeley.edu/~ee123/fa11/docs/SGFilter.pdf, online article, 2011, accessed March 28th, 2015
12. „Numerical Recipes in C", William H. Press, Saul A. Teukolsky, William T. Vettterling, Brian P. Flannery, ISBN 0-521-43108-5, p. 650 - 655