

Design and Optimization of Hardware-Efficient Filters for Active Safety Algorithms

Rafal Tomasz Dlugosz and Michał Szulc
Delphi Poland

Marta Kolasa
University Technology and Life Sciences

Paweł Skruch, Krzysztof Kogut, Paweł Markiewicz, Mateusz Orlowski,
Maciej Różewicz, Anna Ryszka, and Dominik Sasin
Delphi Poland

Tomasz Talaska
University Technology and Life Sciences

ABSTRACT

In this paper we present an example design process of filters used in automotive industry. Signal preprocessing is very important operation in active safety algorithms. Such algorithms usually take into account the vehicle state that includes position, velocities and accelerations of the car. On the basis of these data, as well as the parameters and trajectories of external objects “observed” by the car, the algorithms make decisions about various safety actions. Designer of such algorithms must assure an appropriate quality of such signals, which usually means a proper filtering. In this paper we focus on selected important aspects of the filter design process. The main objectives of the presented investigations is to obtain such filters that ensure a sufficient rejection of undesired components from the signal and at the same time that do not introduce too high delay to the processed signals. In our work we consider linear filters such as finite impulse response (FIR) and infinite impulse response (IIR) filters of different types and lengths, as well as nonlinear filters - in this case median filters. Selection of these filters results from the parameters of the signal. Since the filters have to be finally implemented in low cost and low power devices, in the paper we focus also on hardware implementation issues. Selected filters have been implemented in Field Programmable Gate Array to verify the concept.

CITATION: Dlugosz, R., Szulc, M., Kolasa, M., Skruch, P. et al., "Design and Optimization of Hardware-Efficient Filters for Active Safety Algorithms," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 8(1):2015, doi:10.4271/2015-01-0152.

INTRODUCTION

Filtering is one of the basic tasks that can be found in all engineering areas that include, for example, electronics, control, audio and image processing [1,2,3,4,5]. A typical role of a filter is to enhance a signal or to prepare it for a further processing. Enhancement usually means removal of undesirable spectrum components from the signal, e.g. the noise. Filters are widely used, for example, in multirate systems in such tasks as decimation and interpolation [6]. Of particular importance is a proper filtering before the decimation. Decimation is for example used in data compression performed by means of wavelet transform [7].

Various filtering techniques and filter types can be found in the literature. Classification of filters can be made according to various criteria [8,9]. For example, filters can be divided into those that work with continuous or discrete time signals. In the group working with

continuous signals one can distinguish passive (RLC), as well as active filters. In case of discrete time signals, the most commonly used filters include finite impulse response (FIR) and infinite impulse response (IIR) filters [10]. Of particular interest are FIR filters that feature many important advantages, such as: the possibility to obtain the linear phase response and absolute stability.

Filters can process either analog or digital (quantized) signals. Taking into account the properties, one can distinguish low pass and high pass filters that can be combined into pass-band and stop-band filters. Filters can be linear or nonlinear. Although in majority of situations linear filters are used in some cases non-linear filters can lead to better results. We discuss this issue in the paper.

A different issue is the implementation of the filters. The most common techniques include electronics and microelectronics devices as well as software systems. In the first case we can distinguish

analog and digital filters, while in the second case we have filters implemented in microcontrollers, digital signal processors (DSP), filed programmable gate arrays (FPGA) or very often on PC [11,12].

Design process of any filter is composed of several phases. The basic problem is to determine parameters of the signal and then to select an appropriate filter (with proper frequency responses) that will be able to process this signal accordingly. This is a system level design. At this stage we usually do not care about the implementation issues which are important in the next phase. The purpose of the next stage is an efficient and low cost implementation of the filter, which is of particular importance in the industry.

Various trade-offs can be indicated at particular stages. More selective filters with narrower pass-band usually require more complex structure with larger number of coefficients. In ASIC (application specific integrated circuit) implementation this means larger area occupied by the filter and larger power dissipation. In ASIC and in FPGA, that enable parallel data processing, larger complexity of the system usually does not affect significantly the data rate offered by the filter. Reduced data rate is a crucial problem in case of any sequential systems like DSPs, microcontrollers and PC. On the other hand, filters realized in programmable environments are more flexible and can be reprogrammed anytime if there is a need of that. Such a possibility is more difficult to obtain in hardware.

The motivation behind the presented work is a development of algorithms used in active safety area in the automotive industry. Modern algorithms of this type have to be fast enough to assure an appropriate safety level in systems that feature high dynamics. On the other hand, such algorithms more and more frequently must be realizable in hardware with limited computation resources. For this reason, there is a trade-off between these two aspects and thus both system level and hardware level investigations have to be performed in parallel. Proper signal conditioning may facilitate the work of safety algorithms that in turn can be simplified in this way.

Since the algorithms are realized in software systems we focus mainly on discrete time filters such as FIR, IIR and nonlinear median filters. During the design process of such filters one of the main issues is the group delay that results from the filter length, steepness of the transient band, attenuation in the stop-band, etc. Group delay is of particular importance in case of systems that operate in closed loop.

The paper is organized as follows. In next Section we present the problem. The signals that are used by the realized algorithm feature some parameters that are difficult to foresee. This signal depend on the road curvature as well as on some other parameters. In the following Section we focus on the filter design. The main effort has been put on minimizing the filter length as to suppress the delay introduced by the filter. Both FIR and IIR linear filters are considered in this context. We observe that in many situations nonlinear median filter leads to better results and therefore we also

investigate such option. The next important issue that has been studied is an efficient hardware implementation, which usually means using a fixed-point arithmetic. Several example filters have been implemented in the FPGA using the VHDL language. However the outcomes of the presented analysis can be easily ported either to DSP/microcontroller implementation or to an application specific integrated circuit (ASIC) realized in full custom style. The conclusions are formulated at the end.

DEFINITION OF THE PROBLEM

In this paper we focus on a development of the algorithms that are used in emergency braking (EB), as well as in adaptive cruise control (ACC) systems used in the active safety area. In general, the quality of the output signals calculated by any algorithm depends on the quality of the signals provided to the system. Such signals can be provided either directly by a measurement equipment or by a subsystem responsible for data preprocessing and conditioning. The EB and ACC systems are very complex and perform various tasks. Some of the subsystems are responsible for a precise detection of various objects in the surrounding environment (mainly on the road) and for a determination of their parameters (types, trajectories, distances, etc.). Objects detected in this way become one of data sources. Another source is a vehicle state vector that can contain velocities, yaw rate, accelerations, etc. On the basis of these data the algorithms determine if some of the objects (cars, trucks) are potentially dangerous for our car, or if our car can be dangerous for other objects (cyclists, pedestrians, animals). On the basis of such estimates the algorithms react accordingly, for example, emerging the driver, decreasing the velocity or stopping the car.

The problems that frequently occur in mentioned systems are various undesired components in the input signals. Noise is one of the problems, but the input signals can be contaminated by other signals. The main challenge is that such undesired signals can have various parameters that are not necessarily constant in time. For example, we observe some oscillations that appear in the input signals for given values of these signals. A typical way of removing such components is by using filters. However, such filters have to be designed very carefully to avoid distortion of the input signals. One can point out various trade-offs between particular filter parameters. In case of filters working with discrete-time signals (FIR and IIR filters) typical parameters that are the subject of the optimization include the attenuation in the stop-band, steepness of the transient band, relation of the cut-off frequency to the sampling frequency ($k_f = f_{\text{cut}} / f_s$), pass-band and stop-band ripples. All these parameters depend on the filter length (the number of the coefficients in the delay line), while the filter order has a strong influence on the delay introduced by the filter. If a filter is properly designed, then the larger is the number of the coefficients, the more selective filter, with smaller k_f factor can be obtained. Unfortunately, longer filters introduce larger delays. Taking it into account one can notice that low frequency components that

contaminate the input signals with large amplitudes are the most difficult signals to be removed. Filters that can perform this task must feature small k_f factor.

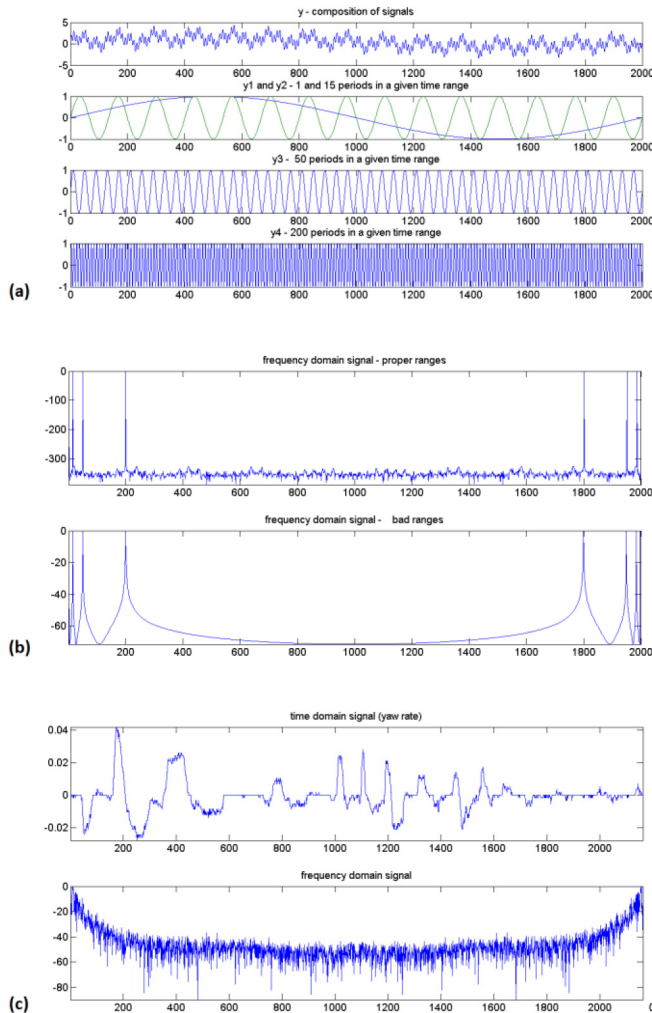


Figure 1. Analysis of the signals: (top) a typical YR (not ideal) signal in time and frequency (FFT) domains; (middle) an ideal periodic signal composed of several harmonics in time domain; (bottom) FFT analysis of the ideal signal in case of proper ranges of particular components and in case if one of the components has improper ranges.

DESIGN OF THE FILTERS AT THE SYSTEM LEVEL

Before starting designing a filter, typically we should know the parameters of the signal that will be filtered. This is not always possible. In the presented example, the yaw rate (YR) signal is not a signal which can be divided into constant harmonics with parameters invariant in time. YR depends on the curvature of the road, as well as the vehicle speed. For this reason, it may happen that over a long period of time the YR signal will be zero (for straight road) or will contain a DC component, if the curve has a constant curvature and the vehicle moves at a constant speed. In case of quick turns at a relatively high speed (for example while overtaking in some situations), high frequency components will appear in the signal. Another problem is that the YR signal can be contaminated by other signals of different frequencies. In such a situation, it is necessary to

find such situations that may be the, so-called, worst case. In this paper we focus on the YR signal, but similar problems we observe for other signals like vehicle speed and accelerations.

As a test case we selected two example signals, say “A” and “B”, that illustrate the described problems. The signal A is an example of driving on a two-lane road when the driver quickly overtakes other vehicles that move at low speed. Signal B is for riding on the turn. In this case, the signal is contaminated by an unspecified component with a relatively low frequency.

To show difficulty of the analysis of the YR signal, in Figure 1 we present (as a short revision) an FFT analysis of two example signals. The first one is the sum of sinusoidal signals with periods perfectly matched to a considered time interval, while the second one is an example real YR signal.

In the FFT analysis, it is good if the length of a considered time interval can be divided, without a remainder, by the lengths of the periods of all components. Figure 1(a) shows such an ideal signal and its components. Figure 1(b) (top panel) shows this signals in the frequency domain. The first sample is reserved for a DC component. The second sample is reserved for the component (y1 in this case), whose period equals the considered time interval, the third sample for that which has two full periods, etc. Figure 1(b) (bottom panel) illustrates a situation where there is no match (the period of the y1 component has been shortened by several samples) and how it affects the ratio (in dB) between the stripes and the background. In the ideal case there is 350 dB difference, while in the second about 60 dB.

Figure 1(c) illustrates an example YR signal and its FFT analysis. Here, there is never a perfect fit for the periods and hence the FFT gives results that are difficult to interpret in the context of the selection of the filters.

In the analysis, shown in next Section, we investigated three groups of filters: FIR, IIR and median defined respectively as:

$$H_{\text{FIR}}(z) = \frac{Y(z)}{X(z)} = b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N} \quad (1)$$

$$H_{\text{IIR}}(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-M}} \quad (2)$$

$$Y_{\text{MED}} = \text{median}(x(n), x(n-1), \dots, x(n-N)) \quad (3)$$

A general objective of our investigations is an efficient implementation of selected filters in hardware. For this reason, we focused on such filters, whose coefficients can be relatively simply implemented as parameters with low numbers of bits. In case of the FIR filters we considered the filters with equal coefficients (two cases):