20th of July 2013

Dr. tech. Benny Sällberg
CEO Sällberg Technologies e.U.

Abstract

This pamphlet discusses signal processing as we see it. It discusses our point of view and how our unique view may help bringing your products to the market.

Signal Processing — Our View

The way one approaches signal processing to solve a given problem is critical. A detailed knowledge as well as a broad overview of the field is necessary in order to find a suitable and optimal solution fitting the given prerequisites. Our Company has this knowledge and level of expertise to deliver you outstanding solutions at any stage of Your product development and in Your product's life-cycle. The following sections discuss some of the signal processing aspects we take into consideration during processing of signals. The discussion intends to give You an insight into how we practically deal with signal processing.

Signal Dimensionality

One classification we need to consider is if a signal contains one- or multiple channels. A multiple -channel signal is treated fundamentally different than a one-channel signal. This distinction leads us to the first physical classification of signal processing methods. Having access to one signal implies that a temporal processing (processing in time and frequency) can be carried out whereas the access to several signals may lead to spatio-temporal processing (simultaneously processing in space, time and frequency). The latter case usually results in dramatically increased performance than the one-signal case due to the availability of spatial processing, e.g., beamforming. In other words, we have one extra degree of freedom in which to find Your solution.

We can help You select preferred signal dimensionality in your application, or propose signal processing methods suiting the given dimensionality and problem at Your hand.

Signal Accuracy and Precision

The bandwidth of a signal, the voltage range of the Analog-to-Digital signal Converter (ADC), together with the number of quantization steps in a digitizer are all crucial factors to gain a high processing performance. The bandwidth should be high enough to encompass the dynamics of the stated problem, but not more than that to avoid noise gain, wasteful resource usage etc. A too low bandwidth may drastically limit the performance of the processing method. The input voltage range of the ADC should be set to avoid signal clipping under normal operating conditions, else a (nonlinear) Automatic Gain Control (AGC) unit may be required. An overly quantized signal may render quantization noise that lie above the allowed noise margin.

Normally, these factors are given as prerequisites to our work. The factors must be taken into account for, while they have a large impact on the performance of signal processing algorithms. Or, to put it in another way, we have to select a signal processing algorithm that fits into the boundaries set by these parameters.

However, some of these factors can be compensated for in most cases by means of added signal processing. For instance, a digital filter can be designed so as to minimize the quantization errors that arise from a limited precision implementation. An adaptive filter can be iteratively updated so as to avoid filter stalling due to a poor bit resolution. To give You another example, a sub-sampled speech signal can have its bandwidth synthetically extended by interpolating estimated pitch-formants but at the cost of introduced distortion to the bandwidth-extended speech signal. Many of the available compensation techniques that compensate for these nonlinear effects introduce some form of distortion. Better is of course to avoid these effects from the start, by setting up a proper design specification.

We can assist You with these aspects during Your system design phase. Or, we provide our signal processing expertise to tune Your product settings towards the target application.

Statistical Properties

The statistical properties of a signal to process play a crucial role in most signal processing algorithms. The more knowledge we have about a signal and its statistical properties on beforehand, the better processing we can do. To highlight the impact of the signal statistical properties we give some examples.

The extraction of a speech signal from a noisy background is possible even in negative SNR environments, due to the different higher-order statistical properties that distinguishes speech from noise, and provided we are using multiple microphones. If the target signal and the noise have the same statistical properties, e.g., same shape class of distribution, we may have to take other approaches. If both the noise and the target signal are Gaussian, we cannot use higher order information to separate them but must rely upon second order information. In this case, the target signal must be assumed more dominant (have more power) than the noise.

Another example. If we are to suppress noise in a signal we need to know something about the noise (and the target signal component). If the noise is deterministic, e.g., can be modeled by a sinusoidal model (Fourier series expansion), we can suppress all of it, generally. If the noise is stochastic, we can only give an optimal solution based on e.g. the correlation matrix of the noise. If the noise is ergodic, we can assume a simpler model, than if the noise is bursty.

We can help You determine the key statistical properties of the signals You face. Or we may help You select a suitable signal processing approach given the statistical properties.

Performance

How many deciBels (dB) should a signal be attenuated, or how often is a signal detector allowed to fail? The accuracy required to solve a specific problem steers which processing needs to be carried out and the level of performance it must satisfy. Generally, the higher required accuracy the more sophisticated processing is needed. If we are allowed to reduce performance just slightly can have a great impact on, e.g., implementation complexity reduction and power consumption.

We can help You tune the performance of Your products either by adjusting the existing algorithms (and their parameters) or by proposing new algorithms.

Complexity

Our focus is to deliver sustainable solutions to Your real problems. It is therefore clear that the signal processing must be implemented on a device that somehow realizes the desired solution. Most often the signal processing is implemented on a Digital Signal Processor (DSP) connected to signal converters, but can equally well be implemented on a micro controller, in analog hardware, or in a software running on a personal computer. The end platform gives a certain complexity budget for the signal processing algorithm. For instance, it is usually not feasible to have very complex and advanced algorithms implemented on a microcontroller with limited resources. Then, we have to settle for less complexity and less performance. Also, the current consumption and clock rate of a device tells us what processing budget we can afford to not void specified constraints

The complexity of a specific signal processing method can be reduced by carefully selected approximations, without having a large impact on end performance. This type of algorithm streamlining is delivered by our optimization services to You.

Robustness

How often do You allow a processing algorithm to fail? Never, sometimes, or as seldom as it is possible? What operating conditions are considered as normal, and what are considered as extreme operating conditions. What should happen if we face extreme conditions. These are questions related to product robustness. The level of required robustness is a key parameter during the development and design of a signal processing algorithm. Or, seen it from another perspective, the level of specified robustness may tell us if supplementary signal processing is needed (e.g., by including pre-post-processing, or additional control structures).

An important part relating to robustness is methods for error detection and recovery. Say it has been detected that an adaptive filter has diverged or stalled, how should it be recovered to its normal state?

We can help You increase product robustness.

Physical Limits

Any signal that is sensed from the real world, e.g., a signal from a sensor, is bound to physical properties of a product and the environment in which it is captured. A fundamental limit in signal processing says that we cannot do more than what the physical properties allows us to, without introducing more or less gross artifacts.

For instance, Nyquist's sampling theorem bounds the signal bandwidth to half the sample rate in order to have an error free signal representation. The sampling theorem is also valid in the spatial domain, where the physical positioning of sensors control the spatial and temporal bandwidth of a signal, i.e., the spatial resolution. Note that there are attempts to extend signal bandwidth synthetically. But these attempts rely upon specific signal properties, and are bound to introduce signal artifacts.

Other physical limits arise from the signal itself. To give an example, a stochastic signal can only be predicted up to a certain degree that is determined by correlation properties of the signal. This limits the amount of active control we can carry out, for instance.

Implementation Aspects

The last, but definitively not least, of important factors we take into consideration during signal processing are implementation aspects. The implementation aspects lay out the boundaries in which we can operate. Some of these factors have already been directly or indirectly discussed before, but are repeat here for clarity. It must be added that, the implementation details are normally a part of the product specification. So, the implementation aspects have usually to be considered first, before any actual signal processing is designed or developed.

You can use our experience and expertise to help select or design a signal processing system fulfilling not only the signal processing requirements of your application but also considering Your implementation requirements, e.g., resources budget.

Hardware Platform

The hardware platform that is used to realize the signal processing effect onto a signal plays a key role here. The hardware platform defines the resources we can use, and what we can do within the platform. For instance, the instruction set of a microcontroller or processor tells us what methods we should use and which we should avoid, it also tells us how a method can be implemented to maximize throughput or performance.

Software Development Language

The language used to develop a signal processing software has a large impact on implementation performance. Indeed, most processor vendors allow the software to be written in C or C++, but those implementations will never be as efficient as a software written in Assembler.

Efficiency, or implementation performance is here measured as the average number of signal processing operations per clock cycle. It is normally desired to maximize implementation efficiency by either minimizing clock cycles per second, or maximizing throughput, i.e., maximizing the number of signal processing operations per clock cycle. The high efficiency which a low-level language (e.g., assembler) provides must be traded off the narrower source code overview you get as opposed to a high level language such as C or C++. A mixture between high- and low-level code usually results in a winning concept. For instance, having the overall framework written in C/C++ and core routines in Assembler.

We can help You in any stage of signal processing software development.

Word Length

The word length of a digital processor, let it be 8, 16, 24 or 32 bits, controls the level of detail we can expect from a signal processing algorithm. If a resolution is required that lie outside the available word length, additional steps must be taken in order to virtually extend the precision where it is needed.

Fixed Point vs Floating Point

Digital numbers (e.g., signal sample values) are represented in fixed point format or in floating point format. The floating point format allows a wide dynamic range of the signal while the decimal point is variable within the digital word. A fixed point format, on the other hand, has its decimal point fixated at one point in the digital word, hence, it has a predetermined and fixed precision. A combination called block floating point format can be used in fixed point structures to extend their precision. Not all digital processors support both numerical formats, where some support only the fixed point number format.

It is, with few exceptions, much less affording to implement signal processing on a floating point architecture. However, the floating point numerical format require additional control hardware to keep track on the mantissa and exponent (i.e., the decimal point) of the digital word. This additional control hardware adds to the current consumption of a floating point device as opposed to the fixed point device. We usually find fixed point devices in products that require a minimum of current consumption. In devices that support both formats, we can, as developers, decide which format to use, and thereby throttle the consumed power.

You may use our services to implement your algorithm on a signal processor using fixed point numerical format or floating point numerical format. We can also help You translate an existing algorithm from one numerical system to another, or to balance the numerical load between the two numerical systems.

Resources

The amount of available memory residing in a digital system puts a harsh limit to what we can accomplish. The available system resources are also often shared with other processing, or software, leaving a certain resource budget over for doing the signal processing. If an entire signal can be uploaded to memory, then processed, and then, when the processing is done, finally output the result we would be able to do very much. However, we usually have to settle for a limited working memory in practice, especially in real-time applications. This means that we need to represent long segments of data in a compact form, e.g., by parameterizing the data using statistical models or by considering small blocks of data at a time.

Delay

Delay can be split into two sub-categories which are normally mistakenly combined into one overall delay measure in the product specification.

First, delay can be defined as the inertia of a signal processing system to react on a given input. If a system is to detect a critical signal; what is the reaction time from the critical signal enters the system until the system flags its presence? Inertia may be a product of many parameters including, e.g., signal block length, number of filter taps, frequency resolution, etcetera. Inevitably, system inertia renders a certain delay between the input and output signals which need to be accommodated for, or, at least, taken into account. The tolerated system inertia should be well specified in the system specification.

Second, delay may also refer to how a signal is propagating through a signal processing system. If we borrow terms from digital filter theory, where phase refers to the delay, we may distinguish a delay as a linear phase (all frequencies are delayed equally), a zero phase (no delay at all), or a minimum phase (a minimal and close to linear delay over the target frequency band, but nonlinear elsewhere). Linearity of the delay is important in some applications sensitive to nonlinear phase distortions such as in signal coding, signal recognition, pattern matching, or signal classifying. Other applications are tolerant to phase nonlinearities and simply aim at minimizing the overall signal delay. The target application specifies the overall delay requirement and delay linearity.

Clock Rate

The more complex a signal processing algorithm is, the more clock cycles are required per second of data to process. The signal bandwidth also steers the clock rate of a digital implementation. It is, with few exceptions, always desired to keep the clock rate of a system low. First, dynamical current consumption is proportional to the system core voltage and the clock rate (due to the capacitive load of the clock grid). I.e., the dynamic power consumption is proportional to Vcc^2 F C, where Vcc is the core voltage [V], F is the clock rate [Hz], and C is the average clock grid capacitance [F]. Since the core voltage of a chip is bound to its manufacturing process, we can usually not do much about it. However, by throttling the clock rate to a minimum reduces the dynamic power consumption of our solution, and most chips support dynamic clock rate adjustment. Second, special care has to be taken in high-frequency clock signals as opposed to low-frequency clock signals if they are propagated on a circuit board. Unmatched conductor impedance may lead to clock jitter or clock skew with unpredictable results.

Current Consumption

As previously stated, with few exceptions, it is desired to keep the current consumption of a device at a minimum. A minimal current consumption prolongs the battery recharge time interval in portable equipment. It is also very important not to waste resources to yield a sustainable and environmentally friendly product for the end-user. An effort to minimize the current consumption of a device can even pose a strong marketing advantage over competitors.

We can help reduce the current consumption of Your products.

About the company

Sällberg Technologies e.U. offers services within advanced digital signal processing, hardware development, prototyping and related fields (see Services). Our partners are supported in all stages of their products' life cycle. We can help You at the research desk, during prototyping or production. Our long experience within the field of signal processing, together with an extensive third-party network, will bring Your products to the market.

Our motto is "*Innovating the Future*". This means that our broad and deep knowledge in signal processing represents the current state of the art, providing You innovative and sustainable solutions for the future. The solutions we deliver will meet and go beyond Your expectations!

We are looking forward to be able to service you. Please do not hesitate contacting us for more information.

With kind regards,

Benny Sällberg, PhD
Managing Director - Sällberg Technologies e.U.

Contact information
Sällberg Technologies e.U.
Dr. tech. Benny Sällberg
Friedrich Schiller-Str. 11
4840 Vöcklabruck

Web: http://www.sallberg.at
Email: office@sallberg.at
Tel.: +43(0)660 4849 960