Single Bit Radar Systems for Digital Integration

Øystein Bjørndal
Department of Informatics
University of Oslo (UiO)
&
Norwegian Defence Research Establishment (FFI)

September 2013 – September 2017

Thesis submitted for the degree of Philosophiae Doctor

Abstract

Small, low cost, radar systems have exciting applications in monitoring and imaging for the industrial, healthcare and Internet of Things (IoT) sectors. We here explore, and show the feasibility of, several single bit square wave radar architectures; that benefits from the continuous improvement in digital technologies for system-on-chip digital integration. By analysis, simulation and measurements we explore novel and harmonic-rich continuous wave (CW), stepped-frequency CW (SFCW) and frequency-modulated CW (FMCW) architectures, where harmonics can not only be suppressed but even utilized for improvements in down-range resolution without increasing on air-bandwidth. In addition, due to the flexible digital CMOS implementation, the system is proved by measurement, to feasibly implement pseudo-random noise-sequence and pulsed radars, simply by swapping out the digital baseband processing. Single bit quantization is explored in detail, showing the benefits of simple implementation, the feasibility of continuous time design and only slightly degraded signal quality in noisy environments compared to an idealized analog system.

Several iterations of a proof-of-concept 90 nm CMOS chip is developed, achieving a time resolution of 65 ps at nominal 1.2 V supply with a novel Digital-to-Time converter architecture. In pulsed mode, the chip features programmable pulses with a minimum width of 130 ps and a time step of 65 ps. In CW mode, we can transmit arbitrary signals up to 3.8 GHz all the way down to DC. With a continuous time single bit receiver, the backscattered signal can be mixed with the on-chip XOR gate and integrated with the on-chip counters, to provide a system-on-chip CW platform.

Acknowledgments

I am really grateful to everyone who has collaborated, motivated and supported me throughout this work. In particular, my supervisors Svein-Erik Hamran from FFI and Tor Sverre Lande (Bassen) from UiO; whose insight not only in their respective fields but also cross-discipline, goes without saying, has been invaluable. Thank you to Svein-Erik for always having an open door, your motivation, guidance and realistic goals have helped me grow as a researcher. Thanks to Bassen for pushing boundaries and scrutinizing my suggestions, both the long and short conversations, in-person and by email; has not only been motivating but also enlightening.

My colleagues both at FFI (Norwegian Defence Research Establishment) and at the Nanoelectronics group at UiO (University of Oslo) for your support, insight and lunch breaks: Tor Berger, Sverre Brovoll, Leif Damsgård and Mats Jørgen Øyan from FFI and Philipp Häfliger, Yngvar Berg, Joar Martin Østby, Kristian Gjertsen Kjelgård, Olav Stanly Kyrvestad, Mathias Tømmer, Jon Håvard Eriksrød, Tohid Moradi Khanshan, Amir Hasanbegovic and Sohail Musa Mahmood from UiO. Especially Kristian and Olav, without whom this thesis would be purely theoretical, for your assistance and support with chip tapeout, circuit board design, fabrication, lab setup and debugging.

I would like to acknowledge some of the previous PhD students working on the swept threshold principle and implementation for pulsed radar: Håkon Andre Hjortland, Malihe Zarre Dooghabadi, Shanthi Sudalaiyandi and Nguyen Thanh Trung whose circuit blocks have all been vital for the chip prototype in this work. In particular, to Håkon, for paving the way with the swept threshold principle and for your invaluable and insightful comments. I am also grateful to Novelda for the comparator.

Financial support from FFI from September 2013 through December 2016 and UiO from January 2017 through June 2017 is gratefully acknowledged.

Lastly to the love of my life, Ine, thank you for your patience and understanding and to my family for your support and encouragement.

Contents

Nomenclature

ADC	Analog-to-Digital Converter
AWGN	Additive white Gaussian noise
BW	Bandwidth
CAD	Computer-Aided Design
CIC	Cascaded Integrator-Comb
CML	Current Mode Logic
CMOS	Complementary metal-oxide-semiconductor
COTS	Commercial off-the-shelf
CW	Continuous-Wave
DAC	Digital-to-Analog Converter
DCO	Digitally Controlled Oscillator
DDS	Direct Digital Synthesizer
DTC	Digital-to-Time Converter
ENOB	Effective Number Of Bits
FMCW	Frequency-Modulated Continuous-Wave
FPGA	Field-Programmable Gate Array
GPR	Ground Penetrating Radar
HDL	Hardware Description Language

LFM	Linear Frequency-Modulated
LFSR	Linear Feedback Shift Register
LNA	Low Noise Amplifier
mux	multiplexer
OSR	OverSampling Rate
PA	Power amplifier
PDM	Pulse-Density Modulation
PLL	Phase Locked Loop
PRF	Pulse Repetition Frequency
PSD	Power Spectral Density
PVT	Process, Voltage and Temperature
PW	Pulse-Width
RMS	Root Mean Square
SAR	Synthetic Aperture Radar
SerDes	Serializer/Deserializer
SFCW	Stepped-Frequency Continuous-Wave
SFDR	Spurious Free Dynamic Range
SNDR	Signal to Noise and Distortion Ratio
SNR	Signal to Noise Ratio
SPI	Serial Peripheral Interface bus
SQNR	Signal to Quantization Noise Ratio
SR	Stochastic Resonance
SSR	Suprathreshold Stochastic Resonance

TDC	Time-to-Digital Converter
VCO	Voltage Controlled Oscillator
WP	Wave-Pipeline

Part I

Thesis

Chapter 1

The first chapter

When interfacing with the real world, an ideal analog circuit can track the environment with infinite speed and infinite precision. The traditional challenge is then to build an analog circuit that is as close to ideal as we need. This usually leads to a circuit that can handle a specific bandwidth (speed) at a specific input range and limited by a noise level. The analog circuit requirements are then traded off between cost (technology), power consumption (vs noise), size/portability, existing vs custom components and so on. There is no doubt that fully analog circuits can achieve remarkable feats. Many of the digital devices we take for granted today started out as a fully analog device, such as clocks, film cameras, record players and radios.

Today clocks, cameras, music players and radios take on an increasing number of digital components, where we “interface” the analog world with Analog-to-Digital Converter (ADC) and Digital-to-Analog Converter (DAC). The first advantage of going digital is in storage, analog storage mediums are lossy and may degrade with time and playback/copy, whereas digital storage promises to be cheaper, smaller and more reliable. The biggest advantage is however the flexibility, once a signal is digitized, we can not only store the information for future playback, but we can process (modify, merge and simplify) and we can easily exchange/share across the world by connecting to the internet. In addition, digital design, allows for a greater abstraction layer between the designer and complex electrical effects; enabling a much larger degree of automation and re-use of multi-purposes digital building blocks than feasible in analog design.

Radar systems have evolved in the same manner as consumer devices, from the first fully analog radar systems that outputs to a Cathode-ray tube PPI display, to systems that does an increasing amount of processing and storage in the digital domain. Processing and storage can be achieved in the analog domain, such as the early Synthetic Aperture Radar (SAR) systems that recorded to film and used optics for the processing [ULB+14]. Optical processing of SAR data was commonplace starting in the 1960s and extending into the early 1990s [Joh06], but is now completely replaced by digital processing. Digital processing allows for increasingly sophisticated image processing, without the bulk and imperfections through an optical system. Figure 1.1 shows one such optical system and the caption confirms what we stated above, that an analog system can be bulky, expensive and application specific.

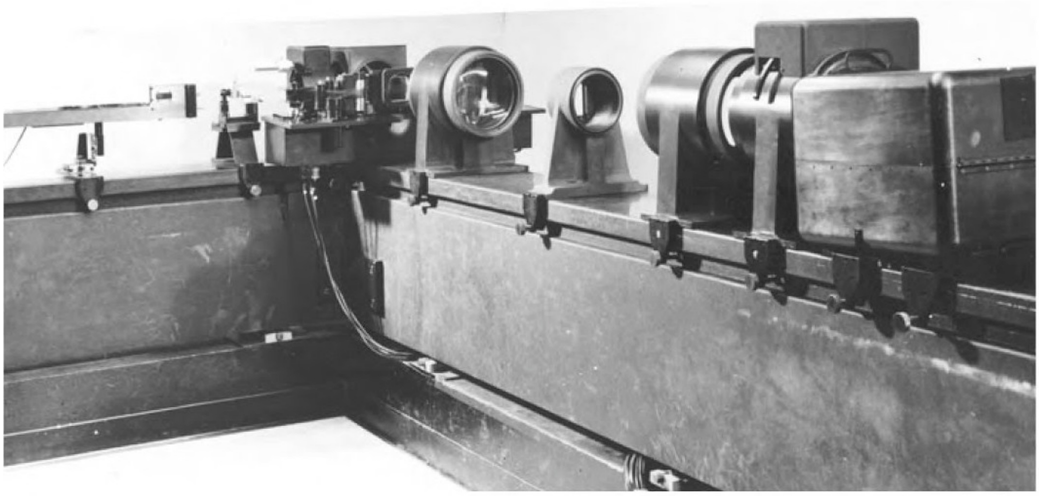


Fig. 4.3. Optical processor at Willow Run, *c.*1960. The large aperture optics were mounted on steel slabs fastened to granite supports and steel I-beams. The large defect-free lenses cost some $10,000 to design and a further $10,000 to fabricate, and were optimized for the wavelength of mercury green light (546 nm) (Leith collection).

Figure 1.1: SAR optical processing from the 1960s [Joh06]. Permission to reproduce this image in this thesis given by Oxford University Press May 2017. All other uses restricted.

Radar transmitters started out as simple spark gaps, but have evolved to sophisticated waveform generators that allow fine control over the waveforms frequency content. Digital transmitters can be built as a Direct Digital Synthesizer (DDS), which allow for flexible waveform synthesis with precise phase control. A DDS based Frequency-Modulated Continuous-Wave (FMCW) radar is planned for both the NASA 2020 rover and Chinese 2020 rover [HBB+15, ZSJ+16]. Both solutions rely on frequency multiplication; as a DDS is limited in its maximum output frequency, but its digital nature ensures reliability and high accuracy.

A previously analog intensive technique, a Phase Locked Loop (PLL), is also being digitized as seen by [Sta11, BVGC10]. A PLL can directly generate higher frequencies than a DDS, where the DDS is limited by the clock rate, a PLL is limited by the frequency range of the oscillator. The disadvantage of a PLL is that it lacks the same phase accuracy as a DDS and requires a lock-time when changing the frequency. Hybrid systems combining DDS as a reference to a PLL are appearing [Gol00, WSJ06], aiming to combine the advantages of both approaches.

In this thesis, we propose a digital frequency modulated radar, that can take full advantage of the rapid advancement of digital technologies. By restricting the signal to a single bit and removing the high frequency clock, we get something similar to a DDS, but operated at a much higher speed. This allows direct frequency generation, only limited in speed, by the (decreasing) gate delay of the technology.

A single bit/square wave signal is rich in harmonics and must be carefully accounted for in the system design process. Traditionally, non-linearity (harmonics) is minimized on the hardware level by careful component selection and filtering, the remaining harmonics are either looked at as distortion (a time domain view often seen in ADC literature) or intermodulation. We will here take a system approach where we carefully analyses how the harmonics affect the system output and how we can circumvent or even utilize the harmonics. These solutions rely on adjustable delays, something which is readily available when working with single bit digital signals, opening up some interesting signal processing solutions that allow the harmonics to be present in every component of the design.

In addition, due to the flexibility of a digital circuit, we also show possible implementations of non-frequency modulated radars, both pulsed time of flight and noise based correlator architectures. We also propose and analyze multiple baseband solutions, that again leverages the single bit digital nature of the signal.

1.1 Introduction to the proposed radar

We will here walk through the thought process of converting a traditional analog frequency modulated radar into a fully digital radar solution that will benefit from modern digital technologies. The Frequency-Modulated Continuous-Wave (FMCW) principle is explained in more detail in the next chapter, but as seen in figure 1.2 the architecture simply requires a frequency source, a mixer, an ADC and some amplifiers. Readers unfamiliar with these concepts may want to skip ahead to chapter 2.

1.1.1 Analog radar

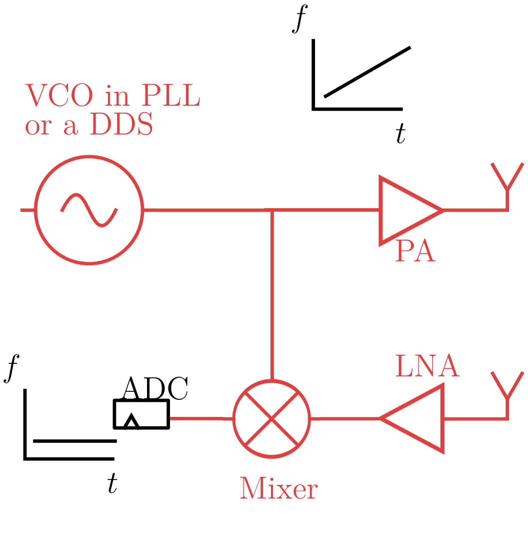


Figure 1.2: Example traditional analog FMCW radar, analog components in red. A VCO (Voltage Controlled Oscillator) is controlled by a PLL (Phase Locked Loop) to generate a low phase noise, linear chirp. A linear PA (Power amplifier) amplifies the transmitted signal while the linear LNA (Low Noise Amplifier) amplifies the returns. The transmitted and received signal is mixed and digitized by a high speed multi–bit ADC (Analog to Digital Converter). Filters not shown.

We note that the only digital component in figure 1.2 is the ADC, and as we will discuss further in chapter 5, even the ADC is not ideally suited for modern digital integration. As for the remaining components, we have mentioned that there is work to digitize the frequency source, by either a digital PLL or by DDS, but these are still circuit blocks with an analog output. Both PAs and LNAs can be integrated in modern Complementary metal-oxide-semiconductor (CMOS), but suffer from poor output power, linearity and noise figure compared to alternative (more expensive) technologies such as GaAs and SiGe.

In a traditional analog design, we want all of the circuit blocks to be linear. Linearity is the basis of linear circuit analysis, giving us superposition and simple, straight forward characterization and measurements techniques. Unfortunately, no real active circuit block will ever be fully linear (for all inputs), as any real active circuit will always eventually clip in amplitude. This becomes even more challenging in modern digital technologies as the supply voltage goes down. A workaround for non-linear blocks can be a filter on the output and analog intensive solutions often need multiple filters throughout the design. Filters are typically in-flexible and depending on frequency not suited for direct on-chip integration.

It should be mentioned that a mixer is a non-linear component, as it produces output frequencies not present on the inputs. That said, a mixer is usually followed by a filter, leaving either the sum or difference only. In addition, one of the inputs either have a single constant frequency or a single modulated frequency signal (as in figure 1.2) allowing us to treat the mixing operation as input-output linear, as it works simply as a frequency shifter. By this, we mean that any linear combination of input signals, will give a scaled (and shifted!) output signal. This shift can be time dependent, as in an FMCW radar.

We will therefore, at times, refer to a mixer as a linear component. In addition, the reader should not confuse a linear chirp or sweep, with a linear component. A linear chirp simply follows a straight line $f(t) = a + bt$ (in the time-frequency domain), while a linear component satisfy $H(ax + y) = aH(x) + H(y)$. Where H is the components transfer function, a and b are constants and x and y are either voltages or currents.

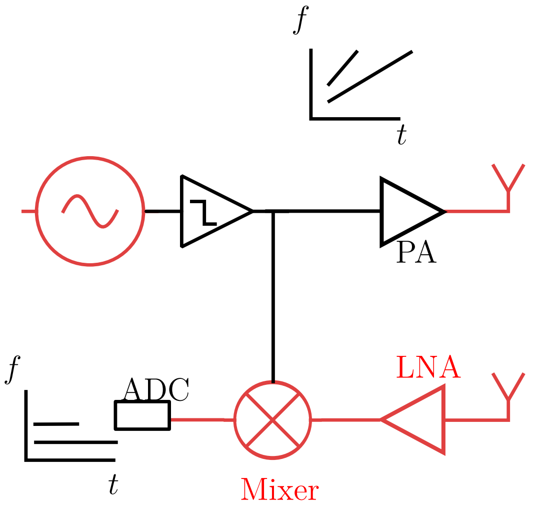


Figure 1.3: Limiting the transmitted signal to two levels. As will be discussed in section 2.2.1 the only information we care about in the transmitter is the zero crossings. We no longer care if the PA is linear and so we can view it as digital.

1.1.2 Digital transmitter

As a starting block, we begin by replacing the analog frequency source for a (digital) amplitude clipped version as shown in figure 1.3. We will discuss this transition more in section 2.2.1, but the essence is that we are not doing any amplitude modulation and we are only interested in a single instantaneous frequency, so we can safely amplitude limit the transmitted signal. If the transmitter has a bandwidth that is greater than an octave, then we have introduced harmonics both in the transmitted signal and in the final mixer output. The only analog solution for removing these is a tunable/selectable filter bank for each octave, before the mixer on at least 1 of the mixer inputs.

The filter bank only needs to remove harmonics on 1 of the mixer inputs since the harmonics we wish to remove is the mixing between harmonics on each of the inputs; as we will see shortly in section 1.1.5. The Power amplifier (PA) now has a digital input and can be class-E/class-F if the bandwidth is sufficiently low.

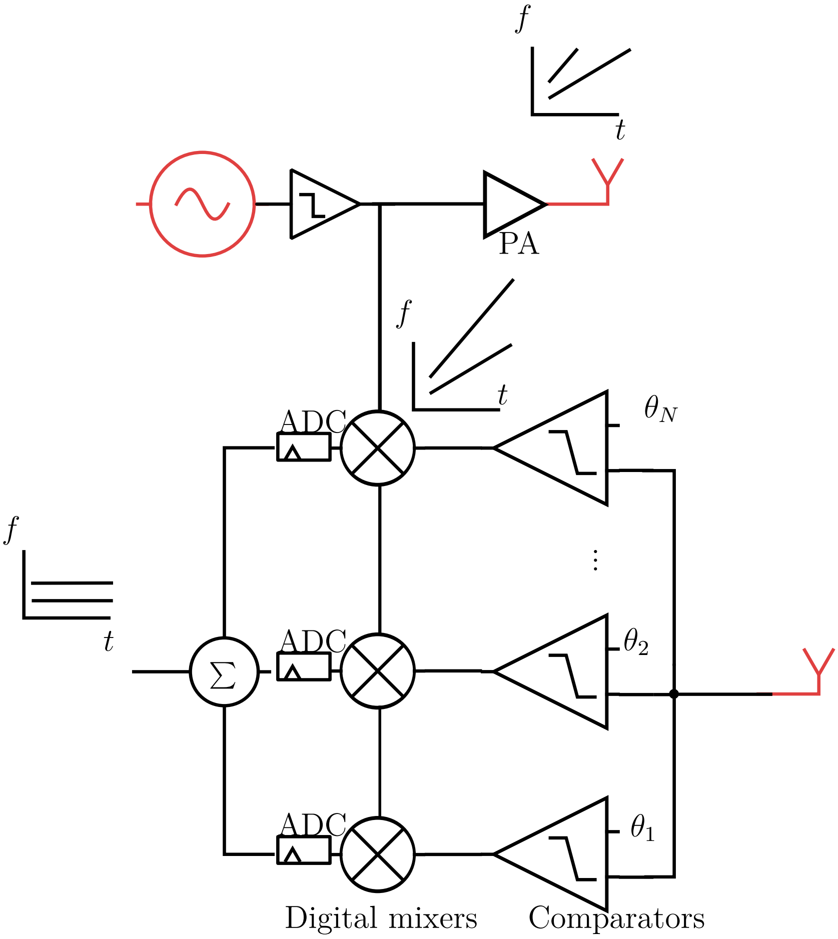


Figure 1.4: Principle of a swept threshold receiver, where each comparator, mixer and sampler can be done serially if the scene is stationary or, as depicted here, in parallel. If the comparator levels θ_i ($i \in [1, N]$) are fixed, linearly separated levels, we have the equivalence of a flash-ADC. Due to linearity of the operations, the processing (mixing and sampling) can either be done before averaging (as shown here) or after, which would be the case for a traditional flash ADC.

1.1.3 Digital transmitter and receiver

To digitize the receiver, we must (as, again, will be detailed in section 2.2) use a number of threshold levels; as simply amplitude limiting the received signal yields a non-linear system. This arrangement is shown in figure 1.4. For low noise scenarios, these threshold levels should be linearly distributed, covering the signals amplitude peak-to-peak level. As will be seen, the levels can also be fixed at the signals mean, if the noise is sufficiently high (instantaneous noise level comparable or greater than the signal level).

At this point we not only have a digital receiver, we also have a unique opportunity to process the signal while it is still in its single bit form. This is illustrated in figure 1.4 where we do the mixing and sampling before we combine the information obtained by the different threshold levels. We will use this extensively in Paper II and Paper III to implement a wide range of receivers using the single bit bitstream.

The mixing operation is much simpler in implementation for a single bit signal than a multi-bit digital signal. We can (1) do the operation in continuous time as there is no synchronization between bits required, (2) implement it as a single digital gate (a XOR gate, as will be shown in section 5.3.1) and (3) have most of the benefits of a digital component which is robustness, general purpose, scalable speed, area and power footprint. The single bit decomposition is explored in detail by Tsividis [Tsi06].

We now notice a small but significant change in the harmonics when going from figure 1.4 to figure 1.5. In the first case, the harmonics will only appear in the beat spectrum if the bandwidth of the system is greater than one octave and even in this case it can be removed if a tunable filter is applied. When the receiver is digital, the situation changes as we are effectively re-creating the harmonics on the receiver side. In essence, the digital gates will have sufficient bandwidth to drive the signal to saturation and give harmonics. And so in figure 1.5, the mixer(s) will have harmonics on both inputs, regardless of the transmit PA and channel and regardless of whether the fundamental is single or multi-octave. This is good news, as we can utilize the harmonics constructively.

1.1.4 Proposed radar solution

We are now ready to arrive at figure 1.5. For maximum flexibility, the transmitted waveform is computed offline, in software, allowing us complete free-

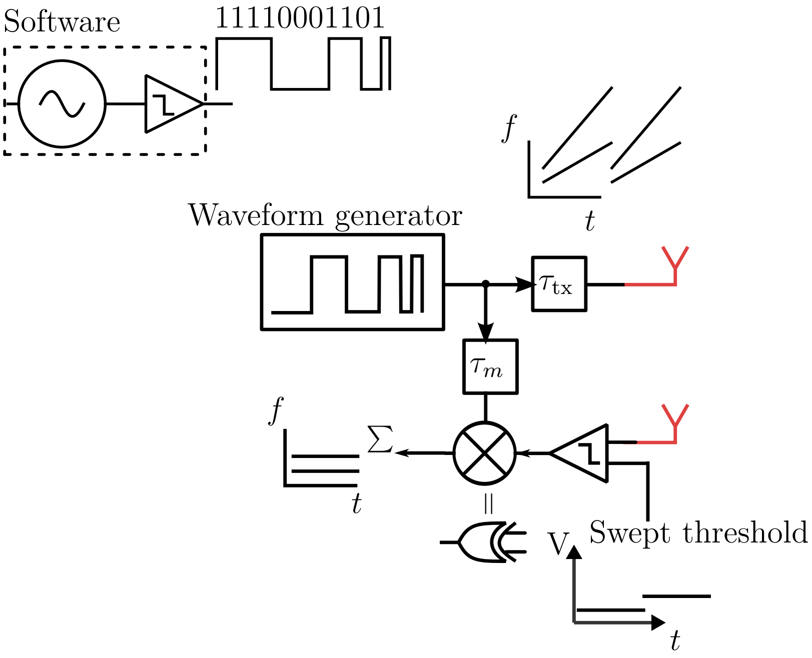


Figure 1.5: The proposed all digital FMCW radar solution. The quantization of the transmitted waveform is now done in software and the transmitter is a digital serializer. The receiver is a single swept threshold comparator with a digital mixer (xor gate). Two adjustable delays inserted in the transmitter and mixing path. The transmitting path (τ_{tx}) can be used to separate the harmonics from the fundamental by lengthening the two-way travel time, while τ_m can be utilized to shorten the time and align the transmitted signal up to the received.

dom when generating the transmitted single bit signal. This flexibility will be a cornerstone in Paper II and Paper III as it allows us not only a frequency modulated waveform but also pulses and noise sequences. As will be discussed in the papers, the adjustable delay τ_{tx} allows us to separate the harmonics from the fundamental by ensuring we do not get any scatters appearing to be close to the radar. This gives us a single band for the fundamental and a separate band for the harmonics.

The parallel receiver structure in figure 1.4 is now replaced with a single swept threshold receiver. As we will see in the implementation chapter, our comparator is quite bulky and, as it has two analog inputs, our only analog block. So at least for our prototype, employing multiple parallel receivers was not economical.

A swept threshold receiver has some interesting properties that we will explore further in section 2.2 and chapter 3. In low noise, the flash-ADC analogous fits well for a stationary scene, while some interesting effects arise when the noise is increased (and is independent between runs). With a fixed threshold, the system exhibit a Stochastic Resonance (SR) phenomena (coined Suprathreshold Stochastic Resonance (SSR)), where the performance peaks at a non-zero noise level. In addition, as the noise becomes comparable or larger than the signal, a performance very close to an ideal analog system is observed. This peculiarity of a swept threshold receiver is explored by the author that coined the term, Hjortland [HWL^{+}06] in his Ph.D. [Hjo16]. Stochastic Resonance is explained well by Mcdonnel [MA09] which hold a PhD on the subject [McD06], while an introduction to SSR can be found by Stocks [Sto00]. Performance characterization in noise when correlating is studied by Watts [Wat62], as will be discussed on page 28.

The same occurs if we compare our single bit system to a multi-bit system, where in excessive noise the multi-bit system can be said to be “wasteful” as the extra bits produced does not carry any meaningful information about the signal, it simply represents the signal+noise with a high degree of accuracy. This observation can be used in the radar system design, so that we intentionally work close or below the thermal limit, to conserve energy both on the transmitter and on the receive side.

The flexibility of a discrete time bitstream transmitter does come at a price. For our frequency source, the discrete time nature does imply a finite time resolution which leads to an ambiguous frequency spectrum and, due to the square wave nature, aliasing. We point out this aliasing in the papers and a work-around by dithering is presented in section 3.2.1.

1.1.5 Beneficial harmonics

We have in the above talked about harmonics in the radars range spectrum. To show why the harmonics appear in the radar output, the solution and also a potential benefit, we here present figure 1.6 which attempts to address all of the above.

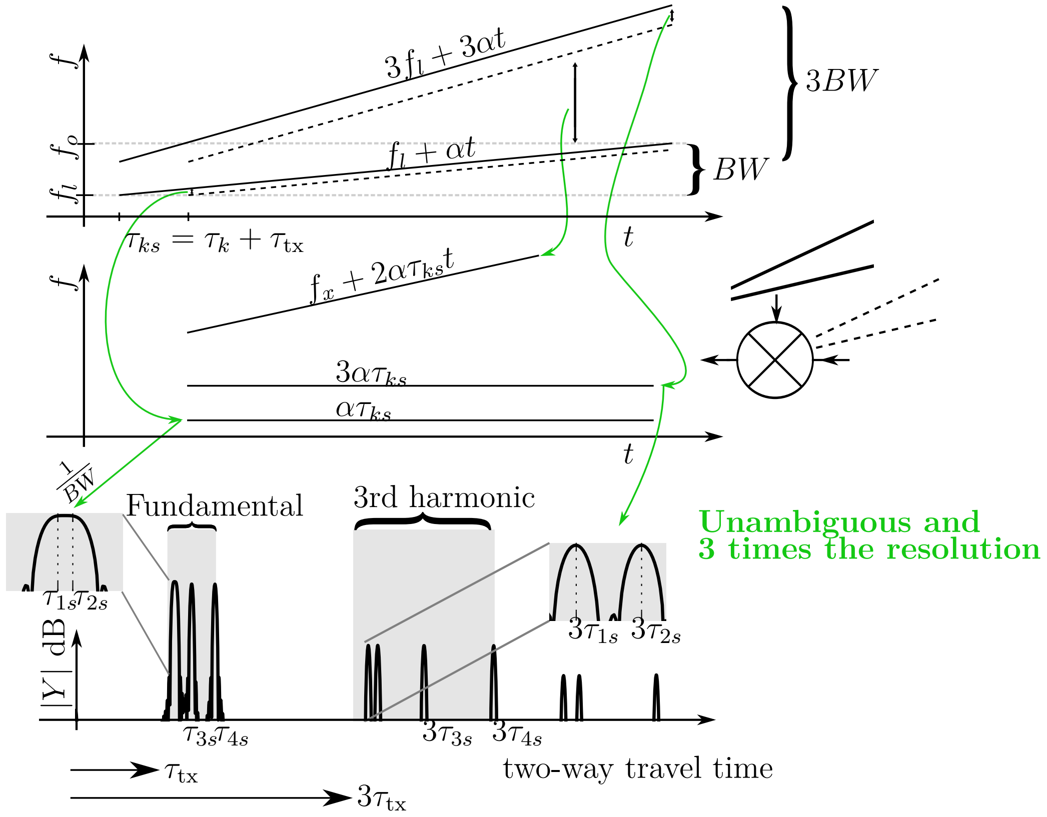


Figure 1.6: Top: a linear frequency sweep with third harmonic, transmitted signal in black and received in stippled. Middle: Mixer difference, showing mixing of the fundamental, 3rd harmonics and one of the intermixing products. Bottom: Resulting beat spectrum when using a delay τ_{tx} in the channel path to separate the third harmonic for a unambiguous view and looking at the harmonics for improved resolution. Bottom panel shows 4 targets of equal amplitude at two-way travel times $\{\tau_1, \tau_2, \tau_3, \tau_4\}$ where the fundamental becomes shifted by τ_{tx}. Note that the top panels only show a single target τ_k, for clarity.

Figure 1.6 starts off with the inputs to the mixer, one time delayed with respect to the other by the two-way travel time to the target τ_k plus the radar transmit path delay τ_{tx}, giving the shorthand

$$\tau_{ks} \equiv \tau_k + \tau_{\text{tx}}. \tag{1.1}$$

As the signal is digital it has harmonics, simplified here to only the third harmonic. Higher order odd harmonics will also be present and non-ideal circuits will also have even order harmonics, but the principle in figure 1.6 holds.

Mixing the reference and delayed version and looking at the mixer difference, we obtain the lines in the middle panel of figure 1.6. The frequency difference of the fundamentals of each input create the classical FMCW range response which is directly proportional to the two-way travel time ($\alpha\tau_{ks}$). In an equivalent manner, mixing the third harmonics with each other, creates another harmonic response at $3\alpha\tau_{ks}$. In addition to these, we will have a number of cross-products. These and other details are treated more carefully in Paper III. One of the cross terms is drawn here with a slope of $2\alpha\tau_{ks}$. Care must be taken such that the cross terms are easy to filter out and the reader is again encouraged to see Paper III.

1.1.6 Summary

Harmonics can be reduced or eliminated by filtering or by dithering/staggering. One of the key findings in this work, is that the harmonics are not necessarily unwanted and we can (a) shift them outside the band of interest, creating an unambiguous view, and (b) as the harmonics cover a wider bandwidth, this gives us improved resolution; without needing to transmit and receive this wider bandwidth.

By dithering away the harmonics, we lose the potential advantage and we make it much more difficult to shift the unwanted part out of the band of interest, making the case for keeping the predictable harmonics as opposed to whitening and randomizing the energy.

The findings in this work also applies to traditional analog radar systems that want to explore multi-octave bandwidths and systems that struggles with the inevitable non-linearity of active components. The only requirement to utilize the harmonics (if they are present in the system) is some way of offsetting (in time or frequency) the transmitter and mixing path.

We should also point out that digitizing the transmitter can be done independently of digitizing the receiver. The concept drawn in figure 1.3, where only the transmitter is digital, still provides some of the benefits presented in this work and avoids the multiple-sweeps of a swept threshold receiver.

1.2 Applications

Cheap, digital radar solutions, have countless applications as short-range radar sensors. They can also conceivably serve as "backend" for traditional long-range radar systems. These systems will typically require more expensive and power-hungry high-power amplifiers on the transmit side and/or low noise amplifiers on the receiver side, but can still benefit from a digital backend for waveform agility and adaptivity. Cheap digital radars are ideal for beamforming/imaging applications, where one can duplicate and use a single chip per antenna element, or co-integrate several transmitter/receiver channels on the same die.

Low sensitivity and movement sensitive radars can be used as simple presence detection for lighting and air-condition control, and simple CW based radars mounted inside light-bulbs are already available commercially. These CW modules get some interesting reviews, as the light bulb may detect your presence before you enter the room (since it views you through the door). It therefore appears to the user as if the light is always on, which may even be the case, as some users report sensitivity to moving trees outside.

More reliable radar sensors, that can also get the range and not just movement and with clever on-chip digital signal processing; can have great benefits for elderly monitoring or smart-homes, where a radar sensor is less privacy intrusive than a camera and can provide through wall-monitoring. Conceivably these monitoring radars can not only provide detection of presence and movement (and hence unusual activity) but also, falling instances and medical monitoring of the gait, heart and breathing pattern [CRC$^+$11].

Radar sensors also have several applications within environmental monitoring, chiefly looking for water content (high dielectric contrast) in soil, ice or even farmland. Or as metal/conductor detection for land-mines, re-bars in concrete or archaeological surveys. The same Ground Penetrating Radar (GPR) imaging techniques can also be applied to the human body, for breast-cancer detection [BBGN12] and even brain-imaging [TKL16, LNLC15].

Chapter 2

Background

2.1 Radar architecture fundamentals

This chapter introduces the basic radar architectures in use today, these architectures will be re-visited in Paper III in a single bit form, but is here presented in a more idealized manner; to cover the basic principle.

We categorize different radaR architecture based on the mechanism used to extract the Range; be it by time difference, mixer or correlator as shown in figure 2.1. All of the architectures will yield the same end result, where the performance will ideally depend only on the time and bandwidth, but will differ in implementation details and challenges.

We restrict ourselves to coherent radars, where we define a coherent radar; as a radar that retains the full signal, that is, we avoid extracting only the amplitude, we also retain the phase. The coherent assumption is critical for the sweep-threshold receiver as we need that synchronization to recreate the incoming signal. A coherent result is also beneficial for further processing of the radar return, allowing us to use the phase when comparing two radar frames separated either in space (2D imaging) or time (movement/speed estimation).

This chapter starts with the Pulsed time-of-flight radar, which is the most intuitive radar architecture, before introducing a family of radar architectures that can transmit and receive continuously. These either rely on a frequency modulation and a mixer or a full correlation circuit, to extract the transfer function of the environment.

Figure 2.1 shows an overview of the radar architectures covered in this chapter, these architectures will be presented in a single bit form in Paper

III but where here focus on the idealized principle. A pulsed system extracts the range by looking at the time-difference between when the pulse was transmitted and when it was received. Pulse compression radars on the other hand, extract the range from a beat frequency (in the FMCW case) or as an index-lag in the correlation; of a correlator based radar.

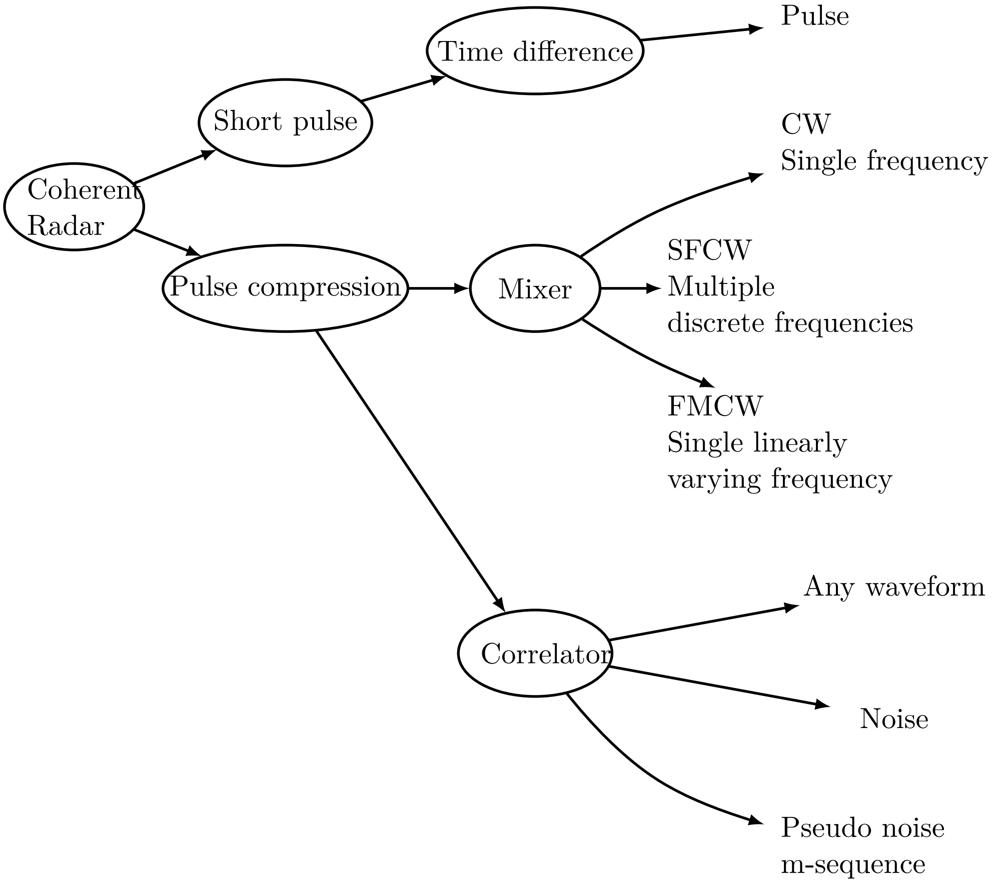


Figure 2.1: Categorizing all of the radar architectures discussed in this thesis; based on the waveform and mechanism to extract the range (time of flight).

Hybrids between these systems are possible, which can either be looked at as a pulsed radar with coding or as a time-gated pulse-compression radar. Both in an attempt to get the best of both worlds. The main motivation for not continuously transmitting being that the transmitter should be off when receiving (at least for the direct coupling), to avoid saturating the receiver.

A detailed analysis of these hybrid techniques is beyond the scope of this thesis.

2.1.1 Pulsed time-of-flight radar

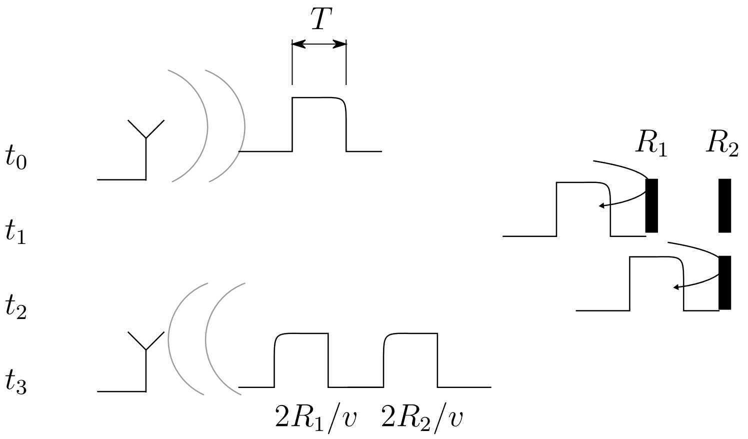


Figure 2.2: Principle of a pulsed radar with two equal reflectors and a sufficiently short pulse length so that the receiver gets two separate pulses.

The first and most intuitive radar type is a pulsed radar where the time-of-flight is extracted by the time difference between transmission and reception. As illustrated in figure 2.2, the transmitted pulse is reflected from two equal reflectors at a radial distance R_1 and R_2 respectively. As the pulse width $T = 1/BW$ is shorter than the two-way travel time between the targets we say that these targets are "resolved" as we can distinctly detect two pulses on the receiver. The required pulse width can be formulated to give a range resolution of [ULB+14]

$$R_2 - R_1 = \Delta R = \frac{vT}{2} = \frac{v}{2BW} \tag{2.1}$$

where v is the speed-of-light in the medium, T the pulse length and $BW = 1/T$ is the bandwidth. We will in this work, without loss of generality, assume the medium is vacuum or air and hence $v = c \approx 3 \times 10^8$ m/s.

To change the frequency spectrum of the pulse, the pulse can be up-converted to a sine-wave carrier before reception and down-converted on the receiver (a process known as heterodyning). This does however *not* change the principle in figure 2.2 and we will for brevity exclusively deal with non-up-converted waveforms in this thesis. The choice of carrier frequency will have a major impact on the front-end, including amplifiers, antenna (size) and the physical objects that can be characterized. Carrier frequency is hence important when a given *application* is envisioned; this thesis tries to be more general and focuses on the traditional versus single bit radar aspect without the constraint of any particular application.

Pulsed time-of-flight radar systems are mentioned in this thesis for the sake of completeness, our main focus is on radars that code the transmitted waveform and correlate on the receiver. In addition, as mentioned, we will neglect any frequency shift as illustrated in the system diagram of figure 2.3. It should be mentioned that the bandwidth in the resolution equation (2.1), is the bandwidth "on-air" and so, a heterodyne radar with up-and-down conversion can usually provide finer range separation at the cost of hardware complexity and increased environment attenuation at higher frequencies.

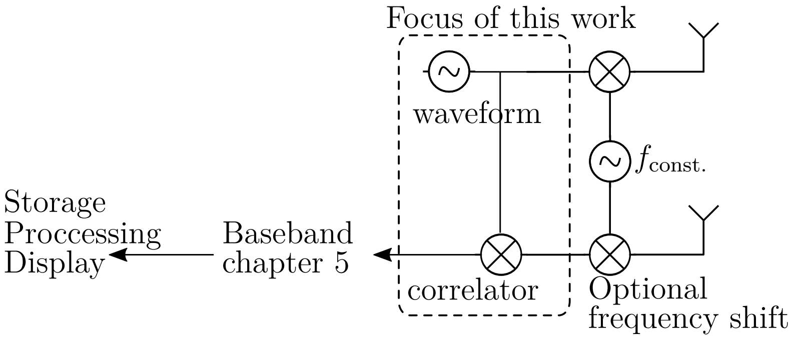


Figure 2.3: A complete radar system, where this thesis is focused mainly on the waveform and correlation method.

2.1.2 Continuous-Wave (CW) Radar

One of the fundamental disadvantages of the naive pulsed radar, is that if we transmit pulses too rapidly (before all of the reflected pulses have been received), we will have a hard time distinguishing between what we just transmitted and what is reflected from some far-off target. We call this a "range ambiguity", which is a very important consideration when selecting radar architecture and parameters. When comparing a pulsed radar to a radar that is continuously transmitting and assuming a given peak transmitted power, receiver gain and noise limit, the only way to increase the sensitivity and range in a given time slot is to increase the percentage of time we are transmitting energy. A pulse radar will therefore need to tradeoff the pulse length (which gives the resolution) and the transmitted energy.

The idea behind a correlation radar, is that we can code the signal that is transmitted and look for that code in the received signal. Hence, we can transmit continuously, or at least a larger percentage of the time; increasing our sensitivity without sacrificing bandwidth.

We will in this thesis, use the Continuous-Wave (CW) term for a constant frequency modulated radar that transmits continuously. To explore the continuous-wave concept, we therefore write

$$y_{\mathrm{CW}}(t) = A\cos(\omega_0 t) = A\cos(2\pi f_0 t) = A\cos\left(2\pi\frac{\lambda_0}{c}t\right) \tag{2.2}$$

which, results in a received scaled and delayed copy for reflector k (assumed stationary)

$$x_{\mathrm{CWk}} = B_k\cos(\omega_0(t-\tau_k)) \tag{2.3}$$

In the above, $\omega_0 = 2\pi f_0$ is the angular frequency, λ_0 the wavelength and c the speed-of-light in the medium.

The received amplitude B_k can be found by accounting for the transmitted amplitude A, the antenna gain, the path loss (which will have at least an $1/R^4$ spreading loss), the reflected amplitude, polarization and the receiver gain. For realistic scenes, estimating the received signal can become its own research topic, which we will avoid in this thesis, by dealing with scatterers of a given amplitude B_k and two-way-travel time τ_k exclusively; as this is sufficient to deal with any linear scenes with stationary targets. Though it is important to keep in mind, that B_k is likely frequency dependent.

Clearly, the range information is contained in the phase of the returned signal, a hardware friendly way of extracting the phase is with an I/Q mixer as illustrated in figure 2.4.

2.1.2.1 Why I/Q?

We will here take a small digression to explore the need for an I/Q receiver. In signal-processing, dealing with signals as complex exponentials are often

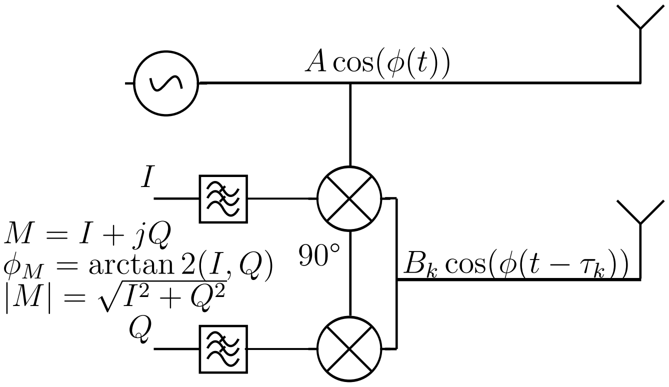


Figure 2.4: Principle of a I/Q continuous wave transceiver, the received signal is mixed with the transmitted signal and a 90° phase shifted version of the transmitted signal (in this case $\sin(\phi(t))$). This gives us sufficient information to extract both the phase and amplitude of the mixer product M.

mathematically convenient, especially since they map nicely to the Fourier transform. Too often however, the real I and imaginary Q parts are extracted by the hardware; in the analog front-end, without any consideration of its advantages and downsides.

If we start by attempting to write the mixer product as

$$M(t) = A(t)\cos(\phi_M(t)) \tag{2.4}$$

then the phase is impossible to extract, since

$$\phi_M(t) = \arccos\left(\frac{M(t)}{A(t)}\right) \tag{2.5}$$

and $A(t)$ is unknown. We therefore see the need for at least *two* values, to solve for your two unknowns, assuming of course we actually need the instantaneous phase $\phi_M(t)$ and/or amplitude $A(t)$. We therefore take two measurements, one shifted θ degrees from the other

$$M_I(t) = A(t)\cos(\phi_M(t)) \tag{2.6}$$

$$M_Q(t) = A(t)\cos(\phi_M(t) + \theta). \tag{2.7}$$

Allowing us to write

$$\frac{M_Q(t)}{M_I(t)} = \frac{A(t)\cos(\phi_M(t)+\theta)}{A(t)\cos(\phi_M(t))} \tag{2.8}$$

which is easy to solve if $\theta = 90°$, yielding

$$\phi_M(t) = \arctan\left(\frac{M_Q(t)}{M_I(t)}\right). \tag{2.9}$$

We note that two measurements shifted by 90° is sufficient and convenient, but any shift not equal to integer ratios of 180° would work. We can also use more than two measurements, giving us an overdetermined problem, reducing the noise.

An alternative to a phase shift of $\theta = \pi/2$, is to observe that for a constant frequency

$$M_Q(t) = A(t)\cos(\omega_0 t + \theta)$$

we obtain the same value by waiting $t_\theta = \theta/\omega_0 = \frac{\pi/2}{2\pi f_0} = 1/4f_0$

$$\begin{aligned} M_Q(t+t_\theta) &= A(t)\cos(\omega_0(t+t_\theta)) \\ &= A(t)\cos(\omega_0 t + \omega_0 t_\theta) \end{aligned}$$

as Nyquist already requires us to sample at least every $1/(2f_0)$ seconds, the new requirement to sample every $1/(4f_0)$ seconds to obtain complex samples is in many cases not excessive.

It should be re-stated that this assumes the beat signal is a constant frequency, if the mixer output is constant, like in a CW radar with a stationary target, increasing the sample frequency will not help as we do not get any “new” information about the signal. In addition, for a CW radar, extracting the phase allows us to distinguish between a change in reflected amplitude and a change in phase (movement). Hence for a CW radar, an I/Q receiver does make sense, but let us first check if a CW radar can give any useful range information.

2.1.2.2 Range ambiguity of a CW radar

If we return to the transmitted CW signal in (2.2) and the received in (2.3), we can relate the captured low-pass filtered mixer phase ϕ_M to the two-way travel time by a simple scaling

$$\phi_M = \omega_0 t - \omega_0 (t - \tau_k) \tag{2.10}$$

$$\Rightarrow \tau_k = \frac{-\phi_M}{\omega_0} \tag{2.11}$$

or, in meters,

$$R_k = -\phi_M \frac{\lambda}{4\pi} \tag{2.12}$$

which on the surface looks great, since we can now extract the range to the target. Unfortunately, the phase (2.9) is bounded by ±π, which means the range estimate is ambiguous and we must write

$$R_k = -\phi_M \frac{\lambda}{4\pi} \pm \frac{n\lambda}{4} \quad \text{for } n = 0, 1, 2 \tag{2.13}$$

The reader is reminded that at 1 GHz the ambiguity in air is ±75 mm.

A CW radar is on the other hand, a useful and simple radar architecture for detecting the relative velocity of moving targets. We will, with the exception of section 3.3.3, limit the scope of this thesis by assuming targets are stationary for the duration of a radar measurement. Hence for our purposes the CW radar is not very practical.

2.1.3 Stepped-Frequency Continuous-Wave (SFCW)

To solve this range ambiguity, we introduce the Stepped-Frequency Continuous-Wave (SFCW) radar, sketched in figure 2.5. The idea is that a single frequency is ambiguous, but by probing the environment with a large number of discrete frequencies we can resolve the ambiguity. A different perspective is in the frequency domain, where a SFCW radar recreates the environments transfer function by discretely probing with CW waveforms, hence building up the transfer function one frequency at the time. This is how most network analyzers work, hence a network analyzer makes a great lab-based radar systems.

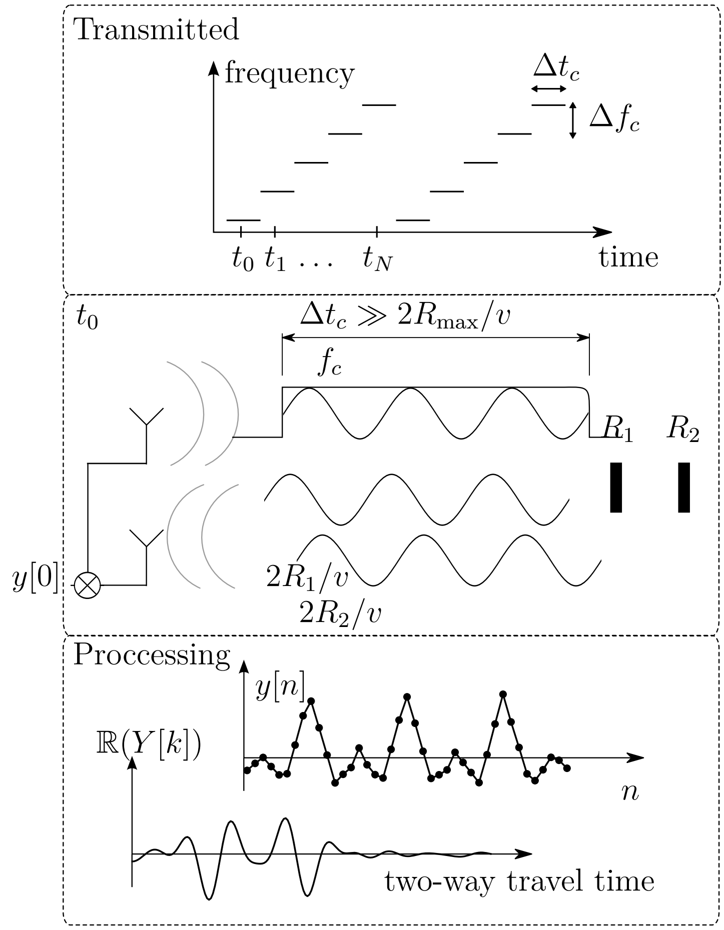


Figure 2.5: Principle of a SFCW radar, showing N frequency steps spaced Δf_c apart. Note that the pulse time Δt_c is much longer than the two-way travel time so that the receiver compares the phase of the transmitted and received signal for each frequency step. After a low-pass filter (not shown) we are left with a single (possibly complex) samples $y[n]$ for each step which by a Fourier transform and a scaling gives the range response.

A stepped frequency radar that steps through $n \in [0, \ldots, N-1]$ frequency steps, from f_l to $f_l + (N-1)\Delta f_c$, will, for each step, obtain after mixing and low-pass filtering

$$y_{\text{SFCW beat}}[n] = AB_k \cos\left(2\pi \left(f_l + n\Delta f_c\right) \tau_k\right) \tag{2.14}$$

for a single target at τ_k. After all of the $y[n]$ samples are recorded, we can extract the delay τ_k by a Fourier transform of (2.14). We note that by superposition, multiple targets can be extracted.

A SFCW radar has a bandwidth of $N\Delta f_c$ and hence a resolution of

$$R_2 - R_1 = \frac{v}{2BW} = \frac{v}{2N\Delta f_c} \tag{2.15}$$

where the ambiguity goes from

$$\text{range ambiguity CW} = \pm\frac{\lambda}{4} = \pm\frac{c}{4f_c} \tag{2.16}$$

$$\text{range ambiguity SFCW} = \pm\frac{c}{\Delta f_c} \tag{2.17}$$

as outlined in Paper III, which quickly gives a much more useful ranging system.

The receiver for a SFCW radar is often I/Q, but can either be just a single phase as shown here, or as many phases as can practically be implemented.

2.1.3.1 SFCW with frequency offset

Especially for a square wave radar, we will see in Paper III, that a heterodyne implementation will be beneficial. We will therefore briefly introduce the principle here.

A CW heterodyne transceiver is shown in figure 2.6, where we note a need for two *different* frequencies for the mixer inputs (hence hetero-.) and in principle two frequency translation stages. We note that depending on the selected ω_o, the last frequency stage can be done after digitization in the digital (software) domain.

It may seem that we have added additional unnecessary complexity, but there are several advantages to down convert to an intermediate non-zero ω_{IF} as opposed to direct conversion to DC. Heterodyne architectures have

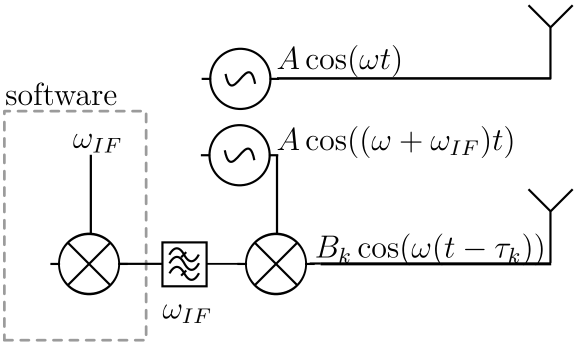


Figure 2.6: CW transceiver with a frequency offset ω_{IF} between the transmitted/received and the second mixer input.

several benefits for I/Q mismatch, flicker noise and DC-offset [Raz97], but we will here concentrate on the idealized case

We have noted in the "Why I/Q" section that an I/Q receiver is required if the frequency is the same on both of the mixer inputs, while as we have argued, we can avoid an I/Q conversion in the analog domain when we have a frequency offset. To show this in the heterodyning case, we write

$$\begin{aligned} M &= AB_k\cos((\omega+\omega_{IF})t)\cos\left(\omega(t-\tau_k)\right) && (2.18)\\ &= \frac{AB_k}{2}\left[\cos(\omega_{IF}t+\omega\tau_k)+\cos(2\omega t+\omega_{IF}t-\omega\tau_k)\right] && (2.19)\end{aligned}$$

which after a low-pass or bandpass filter is the wanted phase shift $\omega\tau_k$ around ω_{IF}

$$M_{\mathrm{IF}} = \frac{AB_k}{2}\cos(\omega_{IF}t+\omega\tau_k) \tag{2.20}$$

where it is trivial to extract B_k and τ_k as long as we have two or more measurements separated in time/phase and we know ω_{IF}.

For our square wave radar, this form of (2.20) is beneficial since the harmonics at $3\omega_{IF}$, $5\omega_{IF}$, ..., can easily be filtered, which is not the case if we have a zero ω_{IF}. An alternative method to obtain two different frequencies at the mixer inputs is a frequency swept transmitter, where a non-zero delay τ_k gives us a non-zero beat-frequency proportional to the delay, which is the focus of the next section.

2.1.4 Frequency-Modulated Continuous-Wave (FMCW)

Instead of discretely stepping through N frequencies, we can do a linear frequency sweep, obtaining a FMCW radar, where the signal is often called a Linear Frequency-Modulated (LFM) pulse or simply a “chirp”. The resolution is then the same as in (2.1) and (2.15), being inversely proportional to the bandwidth of the chirp, while, since we are not discretely stepping, the range ambiguity is removed entirely[1]. The majority of this work is focused on implementing a FMCW radar, but the system is flexible enough to easily adapt to the other radar architectures that we mention in this section.

We now transmit

$$\begin{aligned} y_{\text{chirp}} &= A\cos(\phi_{\text{chirp}}(t)) \qquad\qquad 0 \le t \le T_m && (2.21)\\ \phi_{\text{chirp}}(t) &= 2\pi\left(f_l t + \frac{f_o - f_l}{2T_m}t^2\right) && (2.22)\\ &\equiv 2\pi\left(f_l t + \frac{\alpha}{2}t^2\right) && (2.23) \end{aligned}$$

where the chirp goes from f_l to f_o in T_m seconds and α is the chirp rate.

Similar to a heterodyne system, the resulting low-pass filtered mixer product is no longer a constant, but a varying signal with a frequency directly relate to the two-way travel time, often called the beat frequency $f_{\text{beat FMCW}}$

$$\begin{aligned} \phi_{\text{beat FMCW}} &= 2\pi t\left(\alpha\tau_k\right) + 2\pi\left(f_l\tau_k - \alpha\tau_k^2\right) && (2.24)\\ f_{\text{beat FMCW}} &= \alpha\tau_k && (2.25) \end{aligned}$$

The concept and the chirp parameters are illustrated in figure 2.7. Comparing figure 2.7 and figure 2.5, we note that the processing and output is identical, in that we take our received (mixed and low-pass filtered) samples and do a Discrete Fourier Transform yielding a pulse for each scatterer. The major conceptual difference is that the SFCW radar outputs a single sample (or two for a complex I/Q receiver) for each frequency step, while a FMCW radar gives you the entire frequency response in one sweep. This naturally requires a higher sample rate to digitizer than a SFCW radar, but still far below the transmitted bandwidth BW.

FMCW radar systems are sometimes called a direct-conversion (or homodyne) architectures, but in the view presented in the previous section, the

[1] assuming the chirp length is much longer than the maximum-two-way travel time

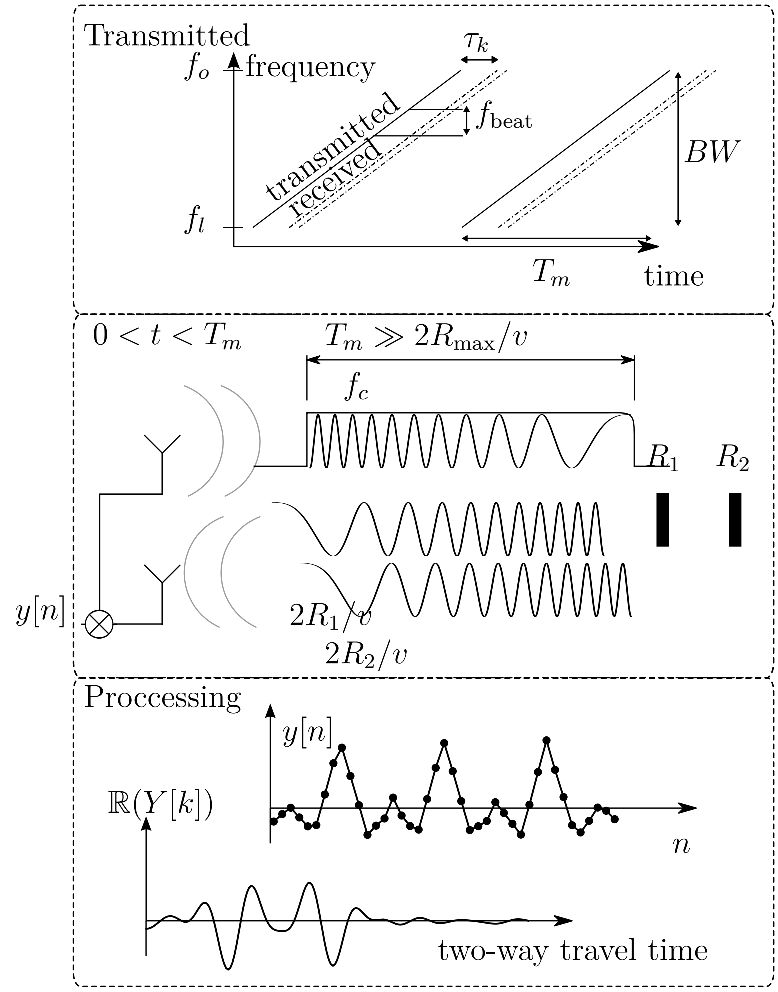


Figure 2.7: Principle of a FMCW radar, showing a linear frequency sweep from f_l to f_o in T_m seconds.

system is heterodyne. The frequency of the two mixer inputs will be different for all non-zero range-delays. In addition, the architecture is in essence doing two frequency translation steps, if we consider the FFT as a set of mixers and filters.

The heterodyne classification is particularly valid when we insert a delay in the transmit path. As we saw in the introduction, a positive $\tau_{\text{tx}} - \tau_m$ ensures the beat spectrum is moved up in frequency. Making the separation of harmonics and signals trivial and ensuring a non-zero difference in the frequency of the mixer inputs.

An architecture that requires the processing of the entire BW is presented next, where we are no longer limited to simple frequency modulated waveforms, but can employ any arbitrary signal.

2.1.5 Correlator based radar

By capturing the entire received signal and doing a correlation with the entire transmitted signal, we have what we will refer to as a correlator based radar. In discrete time

$$c_{xy}[k] = \sum_{n=0}^{N} x[n]y[k+n] \qquad \text{for } -N/2 < k < N/2. \tag{2.26}$$

We notice that for $k = 0$ this is a single mixer and a low-pass filter. A correlation radar can be implemented with a single mixer if we step the delay, effectively giving us a sequential sampling architecture [SSFS08], where each range cell is scanned one at a time.

The correlation inputs, x and y, can be in the baseband, where the signal “on-air” is a phase coded signal on some carrier frequency, as was illustrated in figure 2.3 (page 18), frequency coded or even amplitude coded.

A correlator will work as a matched filter for *any waveform* and is therefore highly flexible. The disadvantage of a full correlation compared to a mixer based radar is that for a digital correlation we must digitize the entire signal bandwidth which is why the sequential sampling architecture is often used [Sac13]. As an alternative, we propose to take advantage of the ease of computing the correlation when we only have a single bit, this means we can cover the entire range view at a time and can use multiple sweeps to reduce the noise and non-linearity as further discussed in Paper III.

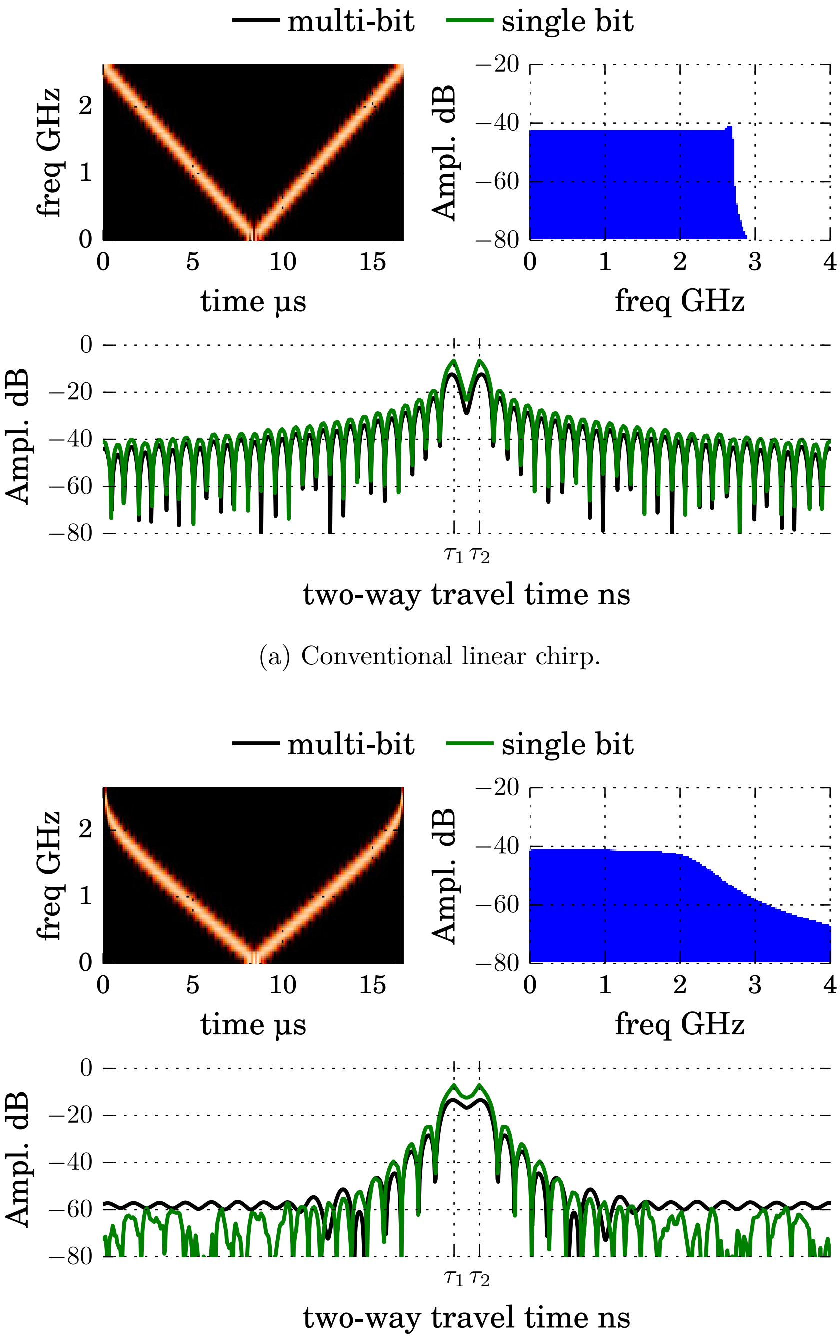
Only correlating with a single bit is an old technique that has been well studied, Watts [Wat62] discusses analog correlation, digital and single bit (called polarity-coincidence), Ekre [Ekr63] the effects of sampling in time. Remley [Rem66] gives a further list of references in the introduction and studies both Gaussian and sinusoidal noise. Weinreb [Wei63] wrote his PhD on a single bit auto-correlation to obtain a power-spectrum which we will return to in section 5.5. The earliest work we have found is by Vleck in 1943, republished in [VM66], which similar to Weinreb looks at the autocorrelation. More recent work include the patent by Reves [Ree10] for a noise radar, which (for some odd reason) proposes to do the correlation after averaging, neglecting the computational advantage of a single bit correlation, but which covers the advantage of adding noise. Axelsson [Axe01] shows how adding noise (or working in a noisy environment), improves the linearity of the single bit correlation and also notes that the concept works for "chirps, step frequency and phase code modulation" [Axe01].

The above literature studies different applications and hence have different input probability density assumptions, but all conclude with either simulation or theory that correlating with single bit does work, though with mixed tradeoffs. A common result found in [Wei63] for Gaussian statistics is a reduction in SNR of $10\log_{10}(\pi/2) \approx 1.96\,\mathrm{dB}$ for a single bit correlator in a noisy environment. Ekre deals with a few different input statistics and sample rates and finds a reduction in SNR between 1 dB and 10 dB while [Hjo16] shows both the 1.96 dB result for Gaussian statistics and a novel result of near zero SNR loss for filtered noise. We will return to this metric in section 3.2 where we show that a single bit FMCW radar has the same property of only a minor degradation in SNR in a noisy environment.

What sets the correlation system proposed in Paper II and Paper III apart is the use of a sweep threshold which will linearize the system even in high SNR scenarios.

2.1.5.1 Any waveform

To demonstrate that any waveform can be used, figure 2.8 shows a simulated scenario with a standard linear chirp (a) as defined in (2.22) and a modified



(a) Conventional linear chirp.

(b) Chebyshev chirp.

Figure 2.8: The two top panels shows the transmitted signal as a spectogram and as a Fourier transform (no-window). The bottom shows the resulting correlation with two targets at τ_1 and τ_2 using both a multi-bit correlation (limited by computer simulation) and over sampled single bit.

chirp (b)

$$\phi_{\text{cheby chirp}}(t) = \frac{\pi\alpha_L}{T_m}\left(t - \frac{T_m}{2}\right)^2 - \frac{\pi\alpha_C T_m}{2}s \tag{2.27}$$

$$\text{where} \quad s = \sqrt{1 - \frac{4}{T_m^2}\left(t - \frac{T_m}{2}\right)^2} \tag{2.28}$$

from [Ric14], where we have used $\alpha_L = 1.5f_o$ and $\alpha_C = 0.075f_o$ in the example. We see from figure 2.8, that by modifying the transmitted signal we change the obtained range profile side-lobe structure. We note that we are effectively applying a window function but in place of adjusting the amplitude we simply modify the frequency to sweep faster through the higher frequencies. For a single target, this is equivalent to studying the autocorrelation of the transmitted signal.

The goal is then to find a signal with a "good" autocorrelation. As it turns out, a random signal can be a good choice, often referred to as a noise-radar.

2.1.5.2 Noise radar, random signal

The goal achieved by a noise radar is to suppress the side lobes, we would ideally like to have a autocorrelation (or correlator output with a return at zero delay) as a Dirac

$$c_{xx}[k] = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{elsewhere} \end{cases} \tag{2.29}$$

A perfect autocorrelation cannot be achieved with a binary sequence longer than length 4 [BA91], but a random sequence will approach this goal for longer sequences. We will, similar to Sachs [Sac13], limit our study to Maximum length sequence (m-sequence), which is a *pseudo-noise* sequence. It is pseudo, because it repeats itself after $2^m - 1$ bits and is hence deterministic. It can also be categorized as a *shift-register* sequence as it can be computed with a shift-register with feedback. Figure 2.9 shows a Galois shift-register configuration of length $m = 4$, where the g_i ($i \in [0, 1, 2, 3, 4]$) is the tap weights with values $g_i \in [0, 1]$. Taps that yield a maximum length sequences are available, [Ins05] gives values up to length $m = 32$.

With infinite bandwidth, a m-sequence of length $N = 2^m - 1$ has the

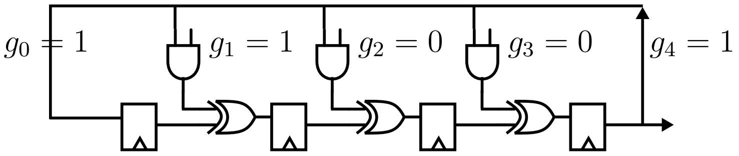


Figure 2.9: Galois shift-register configuration of length 4 with taps $g = [1, 1, 0, 0, 1]$ to yield a m-sequence. Not depicted but the registers must avoid being reset to the all-zero state, all other starting sequences will yield a m-sequence.

autocorrelation

$$c_{xx_m}[k] = \begin{cases} 1 & \text{for } k = 0 \\ -1/N & \text{elsewhere} \end{cases}. \tag{2.30}$$

An interesting note in [Sac13, page 84], is that the $-1/N$ is not a "sidelobe" level, in that it does not hinder the detection of weaker scatterers. In particular, Sachs shows that the $-1/N$ term vanishes for a non-DC coupled channel. To show this, a similar simulation to Paper III is setup, with an order 4 m-sequence and a swept threshold receiver (128 sweeps) with single bit correlation.

One would assume that a scatterer with an amplitude of $1/N$ would be below the sidelobe level and hence undetectable, but as seen in figure 2.10, this scatterer is still visible. In addition, by band-limiting the channel (see figure 2.11, we approach the ideal characteristic of (2.29), the downside is that we now have *true* sidelobes that depend on the channel filter/transfer function and quantization. Note that we here use a band-pass filter, the high-pass response ensures we get rid of the $-1/N$ level, while the low-pass portion of the filtering gives ripples in the correlation output; moving us away from the idealized triangular shape.

In particular for communication, one not only cares about the autocorrelation (2.29), but also the cross-correlation between different users. Example of codes having both good autocorrelation and cross-correlation properties are Gold codes. These codes should also be considered for radar applications with multiple users, such as automotive radars.

As a side-note. In automotive radar, FMCW seems to be the dominating architecture and interference from identical chirps are problematic; Brooker [Bro07] shows this and also discusses some mitigating techniques.

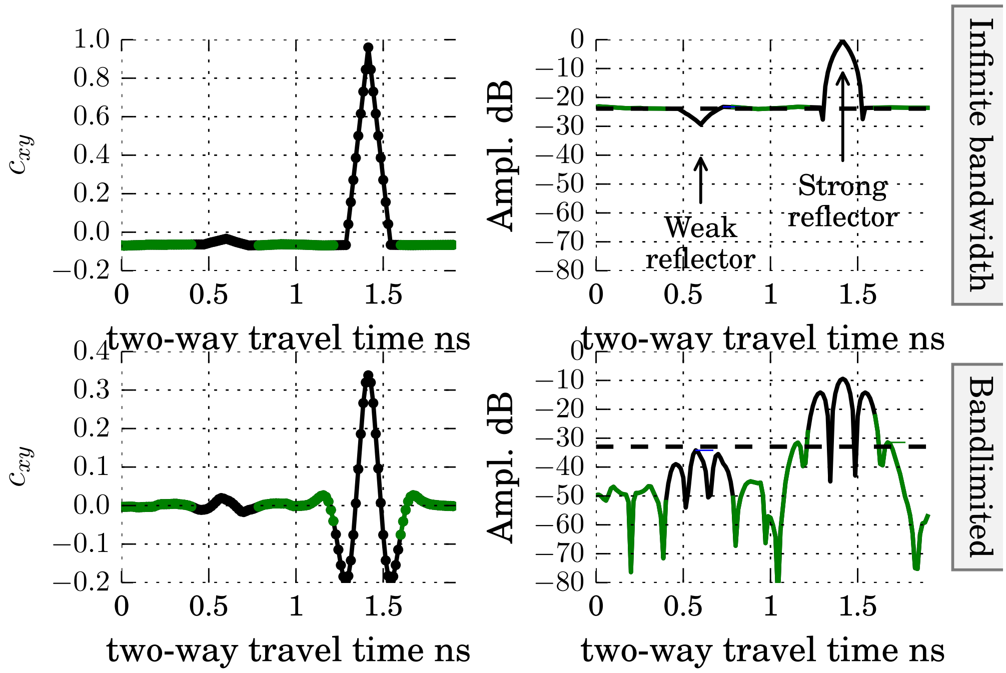


Figure 2.10: M sequence order 4 (N=15) simulation with 2 targets (one with amplitude 1/15=−23.5 dBc marked with strippled line.) in linear (left) and log (right) y-scaling. . Expected return colored black. Top is an ideal channel while the bottom is band-pass limited (see figure 2.11).

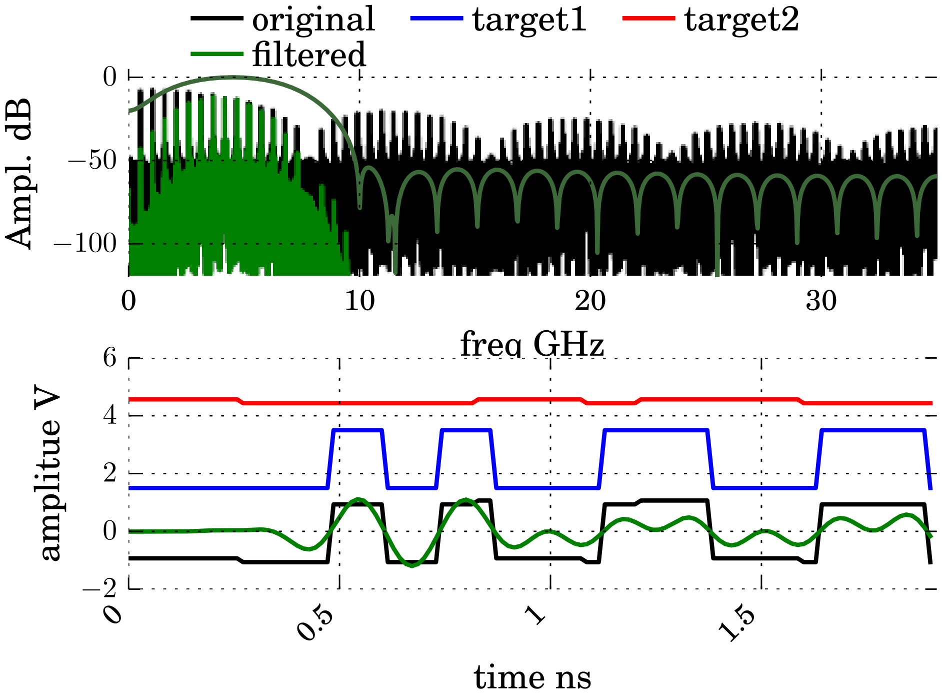


Figure 2.11: Original and filtered (band-limited) channel for a order 4 M-sequence repeated 100 times. Bottom shows the first sequence with the two targets separated in y-scale for clarity.

Pseudo or true random sequences are implementation wise more challenging, but should be carefully considered for automotive radar. It should be stated that for some of the cases discussed by Brooker, a single bit radar will have an advantage over a conventional architecture since it will naturally clip large short time interference signals without any post or pre-processing.

2.2 Swept threshold

A central aspect of the proposed radar system is the sweep threshold quantizer which amplifies the signal and compares it to a (possibly fixed) threshold. We will here discuss some peculiarities of such a quantizer, both for the single quantizer used on the transmitter and the sweep-threshold quantizer on the receiver. We will see that for the transmitter, a single frequency modulated signal can be clipped in amplitude without loss of information, while the receiver requires multiple thresholds to remain linear. We will discuss noise and focus on the benefits it brings to the receiver.

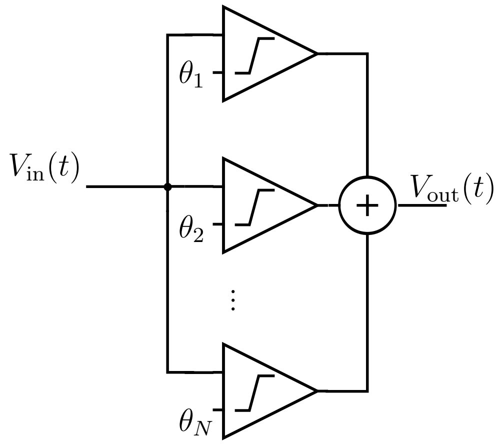


Figure 2.12: Single bit decomposition. Assuming a stationary scene this can be serialize by repeating the sweep while thresholding at different levels

With the arrangement in figure 2.12, there are several points of view depending on the input voltage

- A single sinusoidal can be hard clipped with a single quantizer with the threshold set to the mean level. This creates harmonics, but the original sinusoidal phase and frequency is unambiguously maintained.

- An arbitrary signal can be de-composed with a set of linearly distributed comparator levels, in a Flash-ADC manner. Linear processing can be applied both before or after the summing operation [Tsi06].

- In noise levels comparable to the signal, the threshold levels can even be fixed at the mean level, giving us a SSR system [Sto00].

In addition, we not only have amplitude quantization, but it is natural to include sampling in time. The above points are the focus of this section.

2.2.1 A single quantizer

We start by investigating a single quantizer without noise and the threshold set to the DC level of the signal. In the Fourier domain, we can write

$$\begin{aligned}\operatorname{sign}\left[A\cos(\omega t)\right] &= \sum_{n=1,3,5,\ldots}^{\infty} a_n \cos(n\cdot\omega t) &\quad (2.31)\\ &= \frac{4}{\pi}\left(\cos(\omega t) + \sum_{n=3,5,\ldots}^{\infty}\frac{1}{n}\cos(n\cdot\omega t)\right) &\quad (2.32)\end{aligned}$$

Equation (2.32) shows that the sign (zero crossings) of $\cos(\omega t)$, will yield $\cos(\omega t)$ with additional odd harmonics $\cos(n\omega t)$. We can therefore reconstruct our original frequency $\cos(\omega t)$, by low-pass filtering. We therefore see, that the sign function is sufficient when we are only dealing with a single frequency.

Note also that the original amplitude A in (2.31) is lost, as the resulting square wave will always be ±1 peak-to-peak. Hence, we lose the ability to do amplitude modulation with this simple scheme; for our radar transmitter, this is actually an advantage, as we also gain robustness against amplitude errors, as long as we can keep the digital signal at high and low.

2.2.1.1 Looking at the zero crossings

An alternative explanation, that does not need a Fourier transform, is that we can extract the original frequency $\cos(\omega t)$ by looking at the zero crossings. Clearly the sign function does not change the number of zero crossings and by using the “The Dominant Frequency Principle” [Ked86] we can uniquely extract the frequency, by counting the number of zero crossings. This property is further explored in figure 2.13 where we give two examples of detecting a frequency even in Gaussian noise and sinusoidal interference.

The zero crossing principle can be extended to a linearly increasing/decreasing frequency

$$y_{\text{FMCW}} = \cos\left(2\pi\left(f_l t + \frac{\alpha}{2}t^2\right)\right)$$

which, following [Wil81] is zero at

$$\begin{aligned} n\pi &= 2\pi\left(f_l t_n + \frac{\alpha}{2}t_n^2\right) \quad \text{for } n = 0, \pm 1, \pm 2, \dots \\ \Rightarrow t_n &= \frac{f_l}{\alpha}\left(-1 \pm \sqrt{1 + \frac{\alpha n}{f_l^2}}\right). \end{aligned}$$

Giving us the instantaneous frequency as

$$f_{n+1} = \frac{1}{2\left|t_{n+1} - t_n\right|}$$

As the waveform is linearly modulated, we can linearly interpolate between the $\{f_n\}$ values to reconstruct y_{FMCW}. Meaning we can safely clip a linearly modulated waveform to only 2 levels, as long as this process keeps the zero crossings.

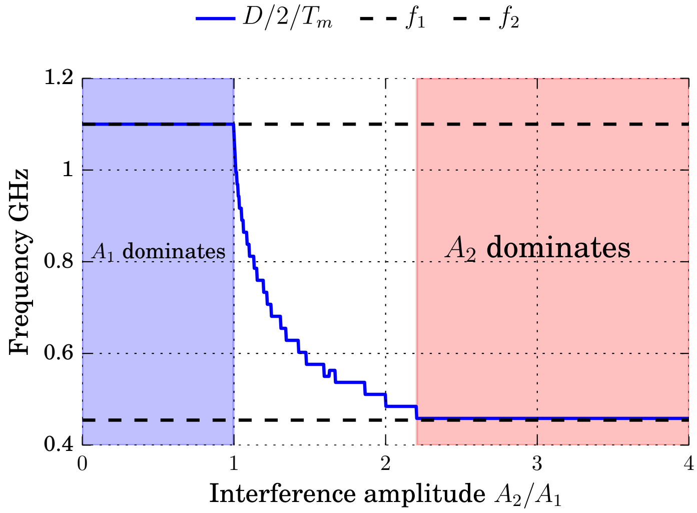
2.2.1.2 Multiple frequencies

Unfortunately, since the sign function is not linear, we cannot expect equality with multiple frequencies

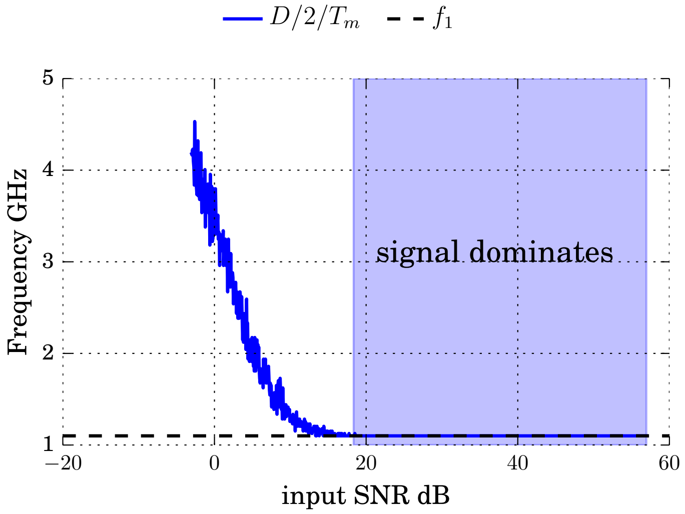
$$\operatorname{sign}\left[A\cos(\omega_1 t) + B\cos(\omega_2 t)\right] \equiv \sum_{n=1,3,5,\dots}^{\infty} a_n \cos(n\omega_1 t) + \sum_{n=1,3,5,\dots}^{\infty} b_n \cos(n\omega_2 t) \tag{2.33}$$

we instead obtain

$$= \sum_{n=1,3,5,\dots}^{\infty} \alpha_n \left[A\cos(\omega_1 t) + B\cos(\omega_2 t)\right]^n \tag{2.34}$$



(a) Extracting the dominant frequency by counting the number of zero crossings D for $A_1 \cos(2\pi f_1 t) + A_2 \cos(2\pi f_2 t)$, as long as A_2 is less than A_1 (and $f_1 > f_2$), the zero count gives the frequency of A_1.



(b) Extracting the dominant frequency by counting the number of zero crossings D for $A_1 \cos(2\pi f_1 t) + N(0, \sigma)$, where $N(0, \sigma)$ is band-limited gaussian noise with zero mean. With sufficient SNR (low σ), the zero count gives the frequency of A_1.

Figure 2.13: Two examples showing the robustness of the “The Dominant Frequency Principle” [Ked86], by using the number of zero crossings to find the dominant frequency in (a) sinusoidal interference and (b) Gaussian noise.

To gain some insight, we can expand the even 2nd harmonic, which will be present for a “non-ideal” clipping

$$\begin{aligned}
&[A\cos(\omega_1 t) + B\cos(\omega_2 t)]^2 && (2.35)\\
&= A^2\cos(\omega_1 t)\cos(\omega_1 t) \\
&+ 2AB\cos(\omega_1 t)\cos(\omega_2 t) && (2.36)\\
&+ B^2\cos(\omega_2 t)\cos(\omega_2 t) \\
&= A^2/2 + B^2/2 \\
&+ A^2/2\cos(2\omega_1 t) + B^2/2\cos(2\omega_2 t) && (2.37)\\
&+ AB\cos(\omega_1 t - \omega_2 t) + AB\cos(\omega_1 + \omega_2 t)
\end{aligned}$$

we now have some DC terms, a frequency doubling of each input and intermixing products $\omega_1 \pm \omega_2$. We note that the original signal $\cos(\omega_1 t) + \cos(\omega_2 t)$ is lost in the above expansion, but this *will* be present for odd order non-linearity.

We see that non-linearity can quickly become unmanageable when we have an arbitrary input signal. Not only do we get harmonic products, we also get various intermixing products that will depend on the input signal. These are non-trivial to filter out and as such we need a different strategy.

2.2.2 Multiple quantizers

We have seen that a single quantizer without noise is unable to deal with two (or more) sine wave signals without introducing significant intermixing products. There are 2 ways of linearizing the quantization process, both based on the principle in figure 2.12. We will start with the intuitive flash-ADC principle in section 2.2.2.1, before venturing into the stochastic resonance world in section 2.2.2.3.

2.2.2.1 Single bit quantization in a noise free environment

As seen in figure 2.14, the swept threshold quantizer can be used as a flash-ADC, where we need to use $2^{N_{\text{bits}}}$ unique threshold levels to get an N_{bits} representation of the input signal.

Modeling the deviation between the quantized and the “true” signal as an error which is white and assuming our threshold levels span a sinusoidal

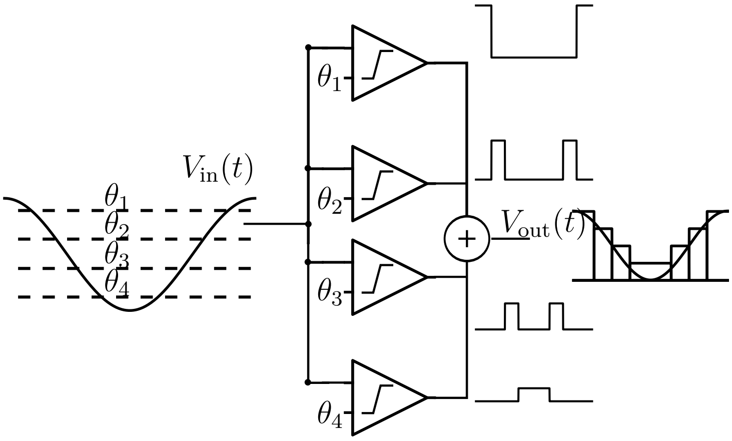


Figure 2.14: Single bit decomposition as a flash-ADC, illustrated with a 2 bit (4 comparators) quantizer. The threshold levels are selected to linearly span the input signals range. Assuming a stationary scene this can be serialize by repeating the sweep while sweeping the threshold and averaging coherently.

signal linearly and sampling at the Nyquist rate of $2f_{\mathrm{signal}}$ we obtain the classical maximum signal to quantization ratio [JM08]

$$SQNR_{\mathrm{db}} = 6.02N_{\mathrm{bits}} + 1.76 \tag{2.38}$$

That is, for every bit we add, the theoretical SNR limit is improved by 6.02 dB. At $N_{\mathrm{bits}} = 16$ bits, the quantization noise level will be at 98 dB and our flash-ADC would need $2^{16} = 65\,536$ comparators.

Implementing 2^{16} comparators on a single chip is not feasible, especially at RF frequencies, but alternatives exist. The method used in this thesis uses a single quantizer and repeats the sweep $2^{N_{\mathrm{bits}}}$ times, coherently averaging the results. A hybrid solution can also be used, where $N_{\mathrm{comparators}}$ reduces the number of sweeps to $2^{N_{\mathrm{bits}}}/N_{\mathrm{comparators}}$. Lastly, as will be discussed in section 2.2.2.2, oversampling and noise shaping can improve the SNR without adding more bits.

At this point, a swept-threshold quantizer may not sound overly appealing, as the quantization noise floor puts a severe limit on the systems theoretically achievable Signal to Noise Ratio (SNR). In addition, to achieve this limit requires (1) a stationary scene during a long integration window, (2) a precise adjustable threshold and (3) synchronization to achieve a coherent integration (integrating in-phase). We could end the thesis at this somber point, but let us first consider oversampling and then the signal in non-coherent noise.

2.2.2.2 Oversampling

We will here briefly cover the effects of sampling and pay particular attention to sampling above the Nyquist rate. In essence, Nyquist promises that we can perfectly reconstruct a signal band-limited to f_{signal} if we sample at least $2f_{\text{signal}}$. A "caveat" mentioned by most signal processing textbooks, is that a band-limited signal implies a signal that extends infinitely in time [Wes16]. Any practical system will process a finite set of samples band limited by a finite filter. An often applied solution to a finite anti-aliasing filter, filter distortion and a reconstruction that reflect the harmonic content of a signal is therefore to sample well above the Nyquist rate [Wes16]. There is however a second advantage to oversampling, namely a possible reduction in quantization noise, which is the focus of this discussion.

A perfect reconstruction assumes a "perfect" quantization, any deviation in the sampling will show up as a deviation in the reconstruction. For a digital quantization, this leads to an unavoidable quantization in amplitude. In particular for the coarse discretization proposed here, with only a single bit, we clearly need an alternative method for obtaining a reasonable resolution. Oversampling is a well known technique to achieve an improvement in resolution, especially combined with single bit quantization as a single bit quantization allows the sample rate to be pushed orders of magnitude higher than the signal frequency.

With the same assumptions as for equation (2.38), the quantization noise will have the same total power when oversampling, but spread (evenly) over a larger bandwidth. If we then low-pass filter (usually done digitally), down to the Nyquist rate, we obtain an improved SQNR of [JM08]

$$SQNR_{\text{db}} = 6.02N_{\text{bits}} + 1.76 + 10\log_{10}\left(\frac{f_s}{2f_{\text{signal}}}\right) \tag{2.39}$$

where $\frac{f_s}{2f_{\text{signal}}}$ is usually referred to as the OverSampling Rate (OSR).

If we shape the quantization noise, which can be done with a Delta-Sigma modulator, the benefits of oversampling is improved. A similar order of magnitude expression as (2.39) can be found in [JM08] which is plotted in figure 2.15 together with a simulation with the python delta sigma

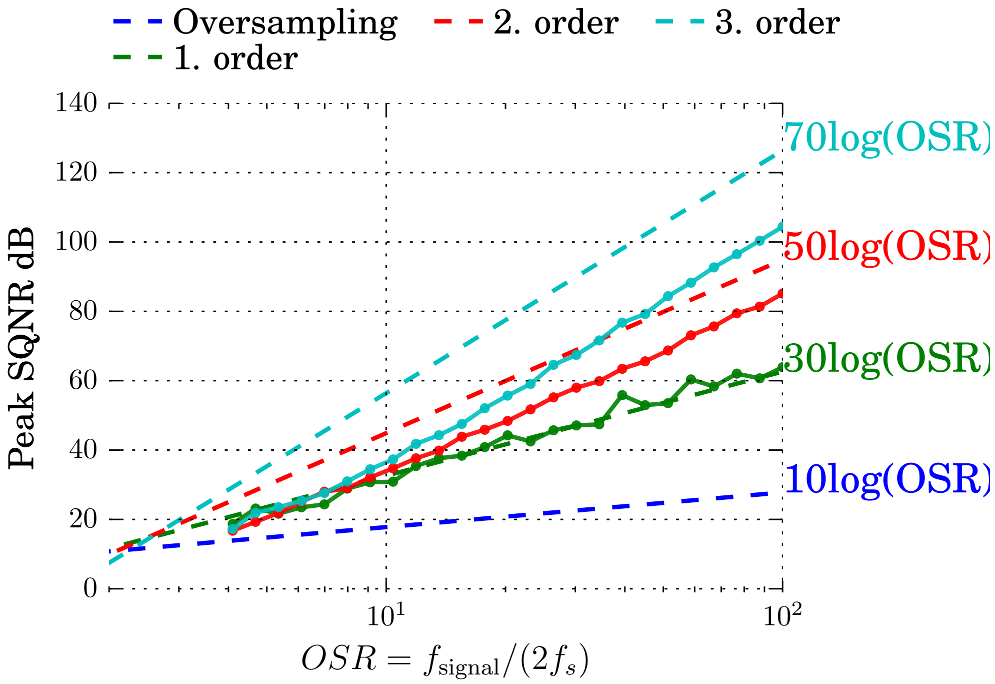


Figure 2.15: Theoretical and simulated, peak SQNR as a function of oversampling and Delta-Sigma noise shaping order. Simulated values uses the [VH15] package with a single sine wave input with an optimized input amplitude. Theoretical lines from [JM08], where Oversampling is shown in this work as (2.39).

module[VH15]. We note a stark contrast between the idealized expressions from [JM08] and the simulated values for noise shaping above first order. In general, higher order noise shaping can become unstable (due to the inherent feedback) and as seen in figure 2.15 do not live up the theoretical improvement in SQNR. In addition, straight forward (single-bit) oversampling is much easier to achieve than higher order noise shaping methods.

An example simulation is shown in figure 2.16, where both "straight-forward" oversampling is compared to a Delta-Sigma simulation when the time quantization is an integer ratio of the signal frequency. We note that the harmonics are effectively shaped to a higher frequency and can be removed with a low-pass filter. The simulation is slightly misleading due to the integer ratio between sample rate and signal frequency, for a non-integer ratio, the straight forward oversampling has numerous down-aliased components in-band.

2.2.2.3 Single bit quantization in a noisy environment

According to (2.38), a single bit quantizer is unable to get a SNR better than roughly 8 dB. This quantization noise floor cannot be reduced by averaging without some stochastic noise, as the quantization process is deterministic. As just discussed we can get some improvement by oversampling and noise shaping, but let us first consider real world noise.

Without noise, we can in theory get infinite range, but due to noise, we will always have a limited dynamic range and hence a limited down-range. We define *noise* as any non-coherent signal (not synchronized to the radar), be it thermal noise, interference and even the part of component nonidealities such as flicker noise, jitter or phase noise; in essence anything that will decrease in amplitude when coherently integrating and averaging.

The "nice" thing about non-coherent noise is that we can always reduce the non-coherent noise by coherently averaging. Ideally, we would capture the entire analog signal (with infinite resolution) and integrate each range cell to reduce noise. In a FMCW radar, this can in theory be achieved with only analog components if we have a narrow band-pass filter, a down conversion mixer and an integrator for each beat bin, but practical systems today usually brings the signal to the digital domain. (We will return to the band-pass filter bank idea in section 5.4). As we have note above, if we digitize with only a single bit, the quantization noise floor will be the

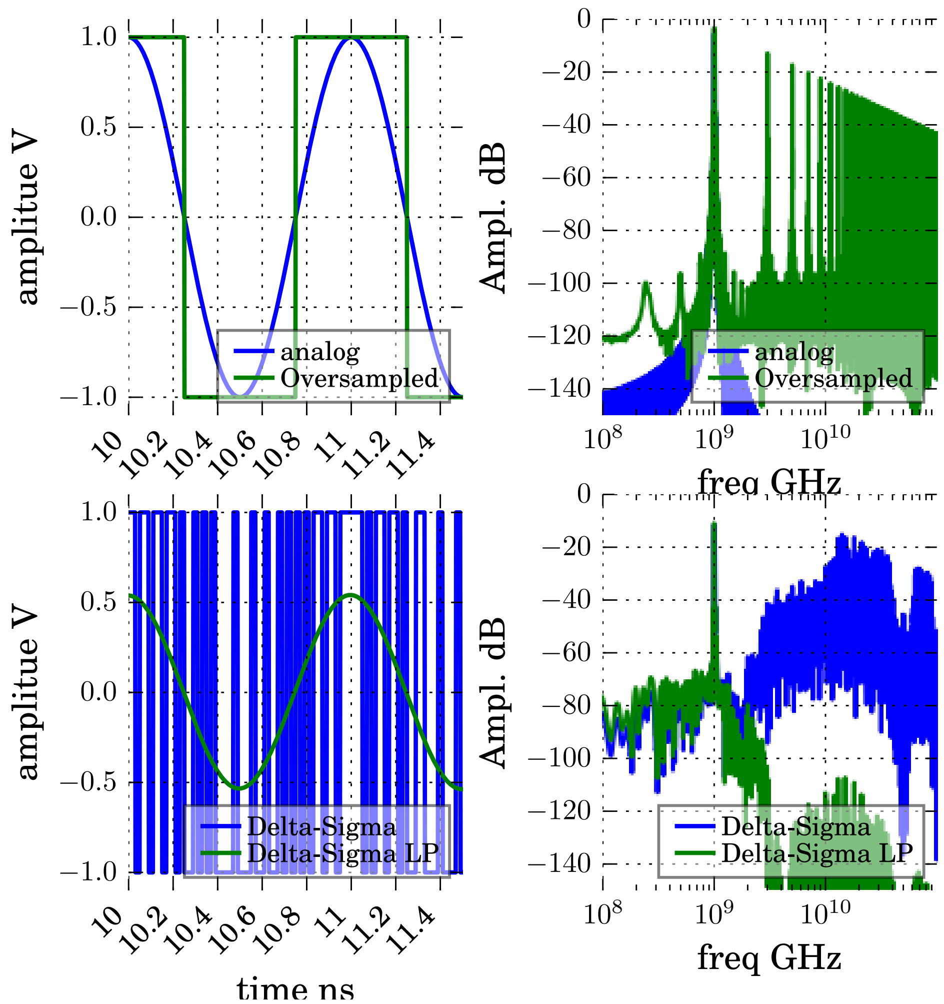


Figure 2.16: A 1 GHz sine wave in time and frequency. Top: Analog vs a single bit amplitude quantized, time sampled to $50 \cdot 1\,\text{GHz} = 50\,\text{GHz}$. Bottom: Example Delta-Sigma bitstream with the same time sampling and the low-pass filtered version.

limiting factor *as long as the non-coherent noise is below the quantization floor.* A logical question to ask is therefore, what performance (-degradation) we can expect if we limit the amplitude resolution to a single bit when the quantization floor is not a limiting factor due to other noise sources.

Table 2.1: Example values and description of symbols for eq. (2.40).

symbol	example value	unit	Description
k_B	1.38×10^{-23}	J/K	Boltzmann's constant
T_0	293	K	Temperature (assumed 20 °C)
E_T	-30	dBm/Hz	Transmitted signal energy si-unit $J = \text{Ws}$
$G = G_t = G_r$	5	dB	Transmitting and receiving antenna gain in look direction (dimensionless: relative to isotropic radiator)
λ	0.3	m	Free space wavelength at 1 GHz
σ	-22.6	dB	radar cross section (si-unit m^2). Assumed a $r = 42$ mm metal sphere.
R	100	m	Down-range
$e^{-2R\alpha_{\text{loss}}}$	0	dB	propagation loss in medium (assumed lossless)
F	8	dB	Noise figure (dimensionless)
$N_0 = k_B T_0 F$	-166	dBm/Hz	Thermal noise floor, si-unit $J = \text{Ws}$

As a motivation, we will briefly cover the effects of thermal noise, by studying the classical radar equation

$$SNR_{\text{thermal}} = \frac{G_t E_t}{4\pi R^2} \frac{\sigma}{4\pi R^2} e^{-2R\alpha_{\text{loss}}} \frac{G_r \lambda^2}{4\pi} \frac{1}{k_B T_0 F}. \tag{2.40}$$

Where the symbols are defined in table 2.1 together with a numerical example to get an order of magnitude feel for the thermal noise floor. With these parameters, we obtain a SNR of 0 dB at 100 meters; when looking for the reflection of an object with the radar cross section of −23 dB. The equation accounts for the transmitted energy along the look direction $\frac{G_t E_t}{4\pi R^2}$, the reflection of the scatterer $\frac{\sigma}{4\pi R^2}$, lossy medium $e^{-2\alpha_{\text{loss}} R}$ (where we here assume a lossless-material such as air $\alpha_{\text{loss}} = 0$), the portion of energy received $\frac{G_r \lambda^2}{4\pi}$ and lastly the thermal noise floor $4k_B T_0 F$ where F is the noise figure of the system.

The transmitted energy is found by

$$E_t = \int_0^{T_m} P_t(t)\ dt \tag{2.41}$$

which for $P_t(t) = P_{tp}$ simplifies to

$$= T_m P_{tp} \tag{2.42}$$

where T_m is the duration and P_t is the transmitted power. We note that depending on the constraints of the application, we may increase E_t (and hence the SNR) by either increasing the power or lengthening the measurement time, but the optimal system will use as little energy as possible.

For an intelligent and optimal radar system, this means that we want to minimize the transmitted energy (by either lowering the transmit power or decreasing the measurement time) so that we are able to detect the scatterer of interest. Hence having a system capable of detection in power levels comparable to the (thermal) noise floor is of practical importance.

To illustrate that discussing the radar equation in terms of the energy as opposed to the more traditional power and time is meaningful; we will briefly review what the energy level of $-30\,\text{dBm/Hz} = 1\,\mu\text{J}$ could represent. For a pulsed system we might be transmitting 100, 1 ns long pulses at a peak power of 10 W with a receiver bandwidth of 1/(1 ns). To obtain the same resolution, a pulse compression radar would spend the same amount of energy by transmitting at 1 mW for 1 ms over a 1 GHz modulated bandwidth and the same receiver bandwidth of 1/(1 ns). Alternatively, if the transmitter is peak power limited to 1 mW, the pulsed system would need to transmit 1×10^6 pulses or sacrifice resolution by using longer pulses or sacrifice max unambiguous range by transmitting more frequently.

If the noise is comparable or greater than the signal level, an interesting observation arises. If we quantize with an ADC where the quantization floor (Signal to Quantization Noise Ratio (SQNR)) is below the signal (coherent) to noise (non-coherent) ratio than we are effectively wasting energy by using

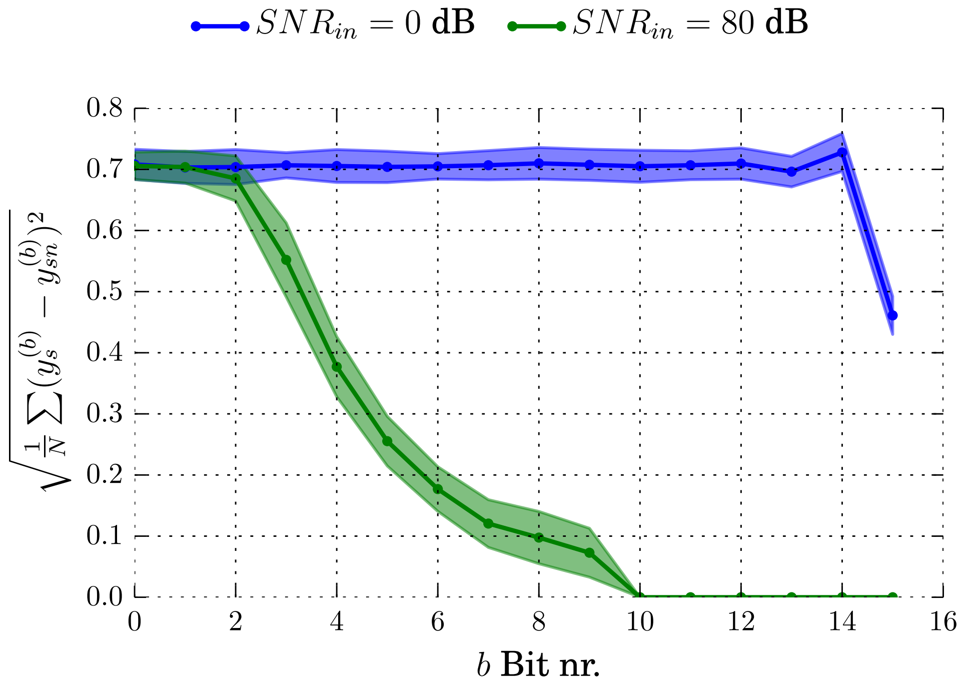


Figure 2.17: Looking at each of the 16 output bits b of a 16 bit ADC. y-axis is the mean square deviation between an ADC with a signal $y_s = \cos(2\pi n/(N/10.1))$ for $n = 0, 1, \ldots, N-1$, $N = 200$ and the signal plus noise $y_{ns} = y_s + \mathcal{N}(0, \sigma)$. Plotted for two different levels of gaussian standard deviations σ where the shaded region signify simulations with different random noise seeds.

an ADC with too many bits. Taking this to the extreme, if we use an analog circuit, which we can view as a circuit with infinite number of bits, than this representation can also be viewed as "wasteful" compared to a lower resolution ADC.

This point is illustrated in figure 2.17, where we compare the output of two 16 bit ADCs, one with just the signal and another with signal plus Gaussian noise. Where the Gaussian noise is comparable to the signal strength for the $\text{SNR}_{\text{db}} = 0\,\text{dB}$ line and much weaker than the signal in the SNR= 80 line. As an "informal" measure of the information contained in each of the 16 output bits, we then take the mean-square difference between each bit and note that for the $\text{SNR}_{\text{db}} = 80\,\text{dB}$ line the most significant bits match exactly due to the low noise level, while the least-significant bits (0-2) show little resemblance after adding the noise. Note that when we said "wasteful" in the previous paragraph, the 16 bit ADC *is* doing a good job of representing its input, i.e. the mean-square difference between the original signal plus noise input compared to the quantized output, but it is wasteful to us since we are only interested in the signal.

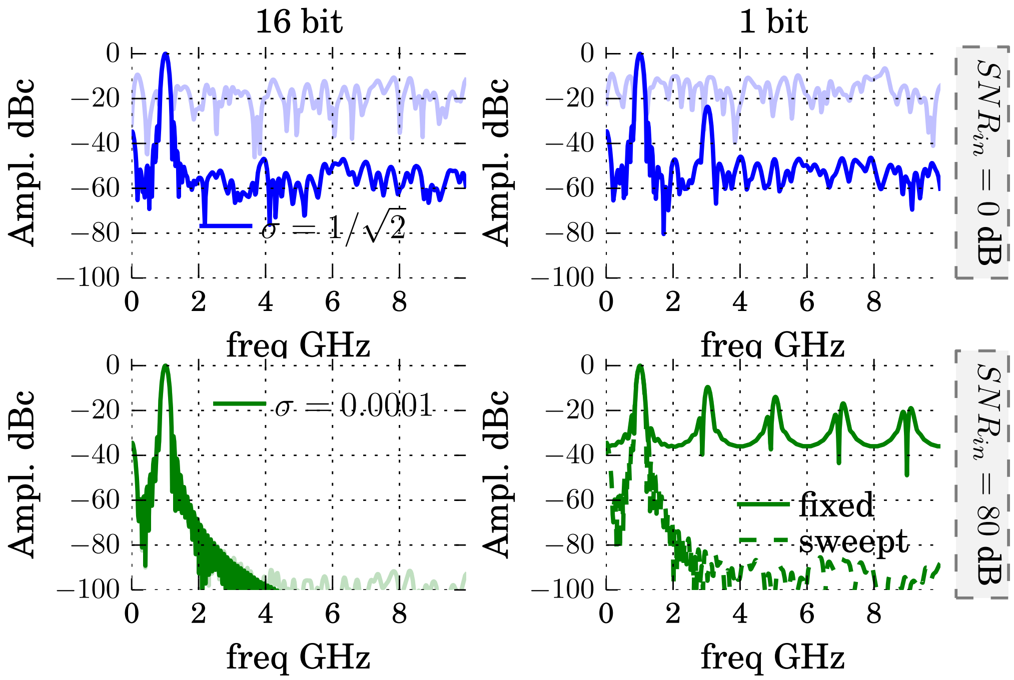


Figure 2.18: Studying coherent averaging with 16 and 1 bit resolution in high and low noise scenario. Averaging 4096 times. Same signal and noise model as in figure 2.17.

As mentioned, an elegant and robust way of separating the coherent signal and the non-coherent noise is by averaging and so, with the same signal and noise model as above, we compare the case of a 16 bit ADC with averaging and a 1 bit ADC with averaging in figure 2.18. We note that in the noise limited scenario (top row), there is negligible difference in the resulting noise floor after averaging. In the low noise scenario (bottom row) a fixed threshold quantizer (lower right) suffers from poor linearity (odd harmonics) and low dynamic range, one remedy to this is to sweep the threshold, as we then obtain the equivalent of a 12 bit ADC ($2^{12} = 4096$).

There is some non-linearity visible in the single bit output (visible as a weak third harmonic) which is because the noise is Gaussian, had the noise been uniform the single bit quantization would be perfectly linear but the bell shape of the noise, shapes the transfer-function, yielding a slightly non-linear output response, the reader is highly encouraged to see [Hjo16, page 102-103] for a detailed discussion and visualization. A simple solution to decrease this effect is to decrease the signal gain to avoid the (non-linear) edges of the transfer function.

As a concluding discussion, we note that an analog or high resolution digital representation is optimal for cases where we only consider the coherent radar signal; as it allows us to view both very weak and very strong scatterers. If we include non-coherent noise (be it thermal or interference), we firstly note that we can decrease the absolute noise floor by spending more energy (either time or power), but optimally we would like to spend as little energy as possible. We also find that having a resolution that exceed the noise floor is wasteful and that for this case, averaging a lower resolution digital representation can perform about the same as a higher resolution averaging.

Chapter 3

System perspective

We will in this chapter briefly cover some system simulations, showing some of the peculiarities of a swept threshold FMCW radar. The other radar architectures mentioned in the previous chapter (pulsed, SFCW and correlator based) will have similar considerations and performance and the reader is refereed to Paper III and our discussion in section 2.1. for the difference between them.

We will start with the influence of the transmitters time resolution (1/clock rate) in section 3.1, before moving on to noise in section 3.2 and ending with 3 different cases of moving targets in section 3.3.

In the publications and for the rest of this work, we have used a very simple SNR estimation method. To quantify the discussion, we have estimate the output SNR of the system by looking at the FFT output, the signal strength is taken as the amplitude at the expected target location while the noise level is found by averaging the remaining spectrum. This estimate is not reliable for high SNR (above ≈ 70 dB) scenarios, as the sidelobes of the signal influences the noise estimate; due in part to the target delay not being an integer of the FFT sample rate. The estimate is also incorrect for low SNR values (below ≈ 10 dB) since the signal estimation becomes incorrect. Despite these inaccuracies, the simple to implement SNR estimate, readily works for a large signal analysis and accounts for both harmonics, intermixing products and noise.

In this chapter however, the estimation method is slightly improved, by being similar to the algorithm used by Hjortland [Hjo16]. As the main objective of this chapter is to investigate the difference between an idealized analog system and a single-bit system, we will in this chapter estimate the SNR by

1. Computing the radar output of both an idealized analog noise-free radar y_{ideal} and the single bit radar being investigated y_{DUT}.

2. Computing the band-limited compressed pulse response by a discrete Fourier transform of the beat spectrum (after applying a Hanning window and zero-padding), covering the minimum and maximum range of interest, yielding Y_{ideal} and Y_{DUT} respectively.

3. Since we do not care about a difference in gain between the two systems, the mean square difference is found by minimizing [Hjo16]

$$\min_{A\in(0,\infty]} RMSE(A) = \sqrt{\frac{1}{N}\sum_{n}^{N}\left|A \cdot Y_{\text{ideal}} - Y_{\text{DUT}}\right|^2} \tag{3.1}$$

4. Yielding the estimated SNR

$$SNR = \frac{Y_{\text{ideal peak}}}{\min_{A\in(0,\infty]} RMSE(A)} \tag{3.2}$$

The difference between this estimation method and the simple one mentioned in the previous paragraph is found to be between 5 dB to 0 dB, where the biggest difference is that this estimate is more stable.

It should be noted that the output SNR does not directly imply the ability to detect weaker scatterers, as this ability is further limited by the number of sweep-threshold levels (giving us a resolution) and the noise (where the appropriate noise level will aid the detection of weaker targets).

3.1 Time resolution

We will here re-visit the advantage of oversampling which was mentioned in section 2.2.2.2, by presenting a FMCW system simulation where we vary the time resolution of the transmitter. We will also briefly cover the continuous time case, where we avoid any time sampling in the transmitter.

Ideally, we would like the continuous time arrangement in figure 3.1(a) but to get the full advantage, including the great flexibility, of a digital system, the time-sampled system in (b) is implemented in this thesis. We will return to the receiver ADC and FFT in chapter 5.

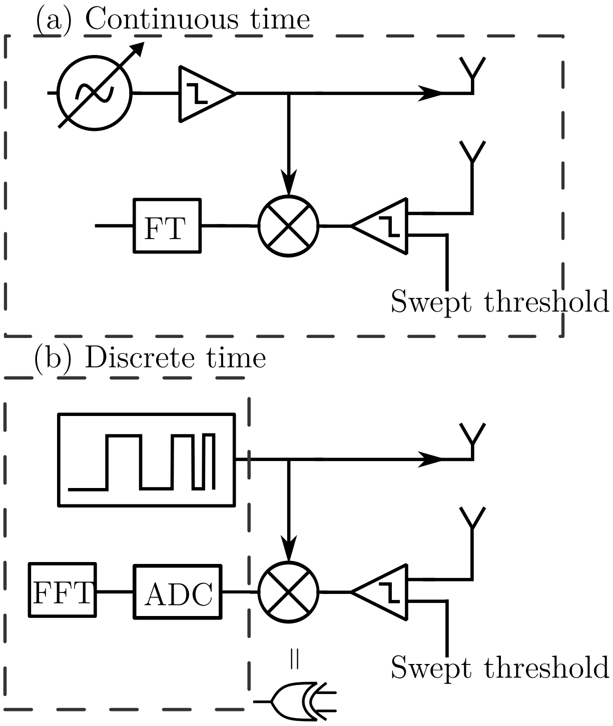


Figure 3.1: The continuous and discrete (digital) implementation of a square wave radar, (a) an idealized entirely continous time architecture with a VCO driven with a continous ramp which is quantized to a single bit(-stream), transmitted and received at the bottom where it is quantized again, mixed and fourier transformed. (b) Digital implementation, where a digitally generated chirp (quantized in both time and amplitude) is transmitted and where the mixer difference is sampled and discrete Fourier transformed.

As mentioned in section 2.2.1, for a single instantaneous frequency (continuous phase), we can safely amplitude quantize the sine wave to a square wave and still retain an unambiguous reconstruction of the frequency (and phase). The problem does get a bit more complicated when we consider time-quantization as well, though the zero-crossing reconstruction in section 2.2.1.1 only requires a sample rate above the Nyquist rate, a Fourier analysis as shown in section 2.2.2.2 is not as kind without going orders of magnitude above the Nyquist rate.

The difference between the zero-crossing reconstruction and a (discrete) Fourier analysis is in the starting assumptions. If we assume we are only

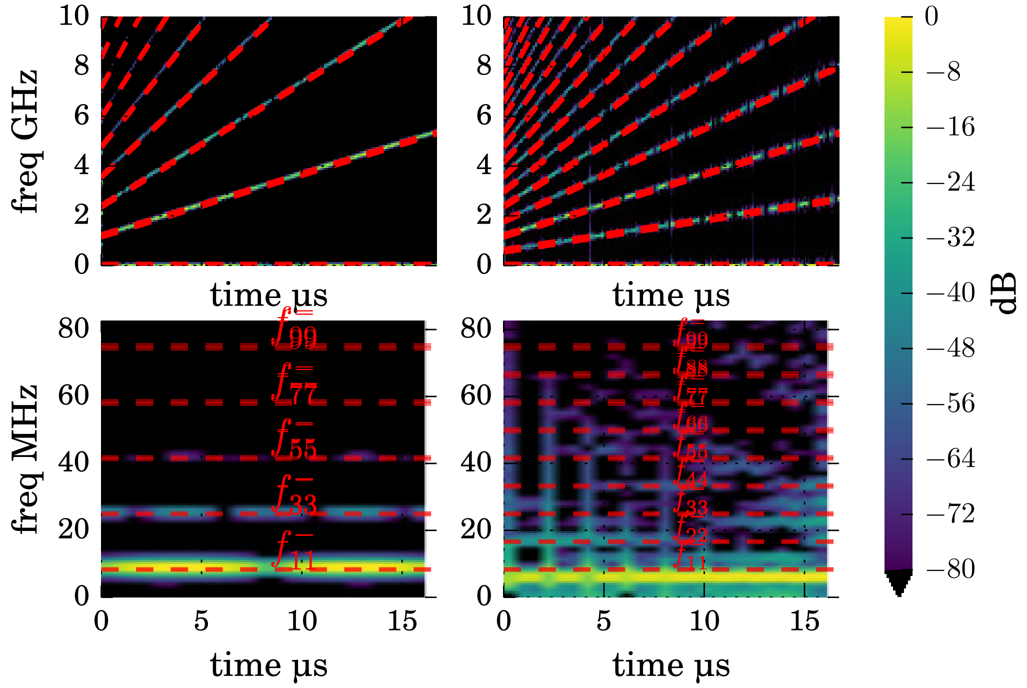


Figure 3.2: Simulated beat spectogram of a digital FMCW radar with two close targets and analytical expressions up to the 9th harmonic overlayed as red dashed lines. Left: python simulation with odd order analytical frequency expressions overlaid, right: post layout XOR simulation with odd and even order analytical expressions overlaid.

looking for a single frequency, then counting the number of zero crossings is sufficient. If on the other hand, we would like a general amplitude weighted sum of a number of frequencies then some form of Fourier transform is required. For our receiver, we generally cannot assume only a single scatterer, so the zero-crossing method fails.

3.1.1 Continuous time

We show in Paper III that a FMCW mixer with amplitude quantized inputs generate harmonics both as multiples of the beat frequency and as multiples of the beat sum, these are shown graphically in figure 3.2. Figure 3.3 shows the beat spectrum, computed with a simulation using two different sample rates, where we note that the “noise floor” is here just an artifact of the

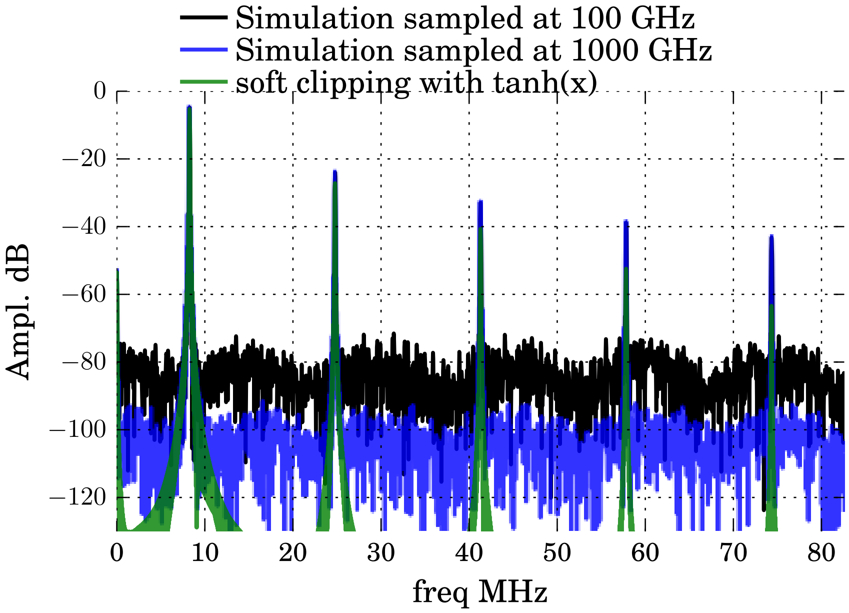


Figure 3.3: Simulated continous time beat spectum with a single scatterer, using 2 different simulation sample rates and using soft clipping to minimize artifical simulation aliasing.

discrete simulation.

We include here a soft-clipping $\tanh(x)$ clipping operator for comparison, to be sure we avoid the aliasing inherent in modeling a square wave on a discrete computer, but we should warn that simulating with a soft clipping such as $\tanh(x)$ will give results that are "too optimistic" for simulations that include weaker scatterers since the weaker targets will simply propagate through the simulation as an analog signal.

Doing a simulation of a continuous time system correctly on a discrete computer is involved, but "fortunately" our system is discrete and as such a thorough study of a continuous time system is beyond the scope of this thesis. It should be re-stated, that the simulation in figure 3.3 does not show that we have an infinite dynamic range in the sense that weak scatterers are detectable as this will depend on the number of thresholds on the receiver, but it does attempt to show that a continuous time system will not have the noise floor level that we see in the discrete transmitter case.

3.1.2 Discrete time

If we sample, either by transmitting a sampled bitstream, or sample on the receiver without an anti-aliasing filter then anything above the sample rate is going to fold down. Figure C.2 (page 183) and figure 3.4 shows what this folding looks like when the mixer sum folds down into the range profile. Avoiding aliasing on the receiver side is best done with an analog low-pass filter and a multi-bit ADC as will be discussed in chapter 5, while aliasing on the transmitter can be mitigated with intentional dithering as will be discussed in section 3.2.1.

Of course, the higher we can push the oversampling rate, less energy is going to fold down; which is why Paper IV focuses on a high speed serializer. To quantize the transmitters time resolution and relate it to the discussion on oversampling we had in section 2.2.2.2, we have created figure 3.5 which shows a sweep of the transmitters time resolution, again using 2 different simulation sample rates as an indicator that we are simulating using a sufficiently small time-step. We note that for simulating a fine time resolution radar (above $10^{11} = 10\,\mathrm{GHz}$), the selected simulation sample rate is not sufficient.

The straight line with a $20 \log_{10} OSR$ slope confirms (2.39) (which is a power quantity, while we here use a voltage convention), while the vertical

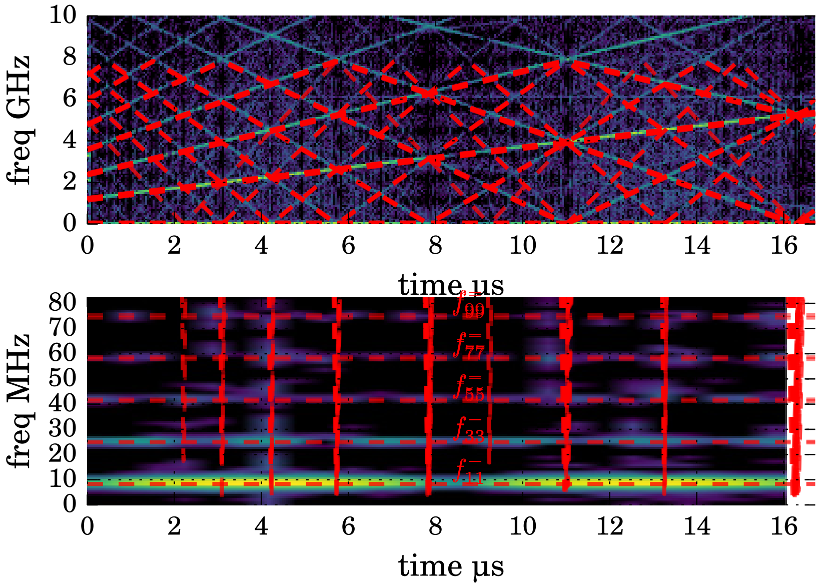


Figure 3.4: Same setup as the simulation in figure 3.2, but with a discrete transmitter, with a time resolution of 64 ps, giving folding around 1/(2·64 ps)=7.8 GHz.

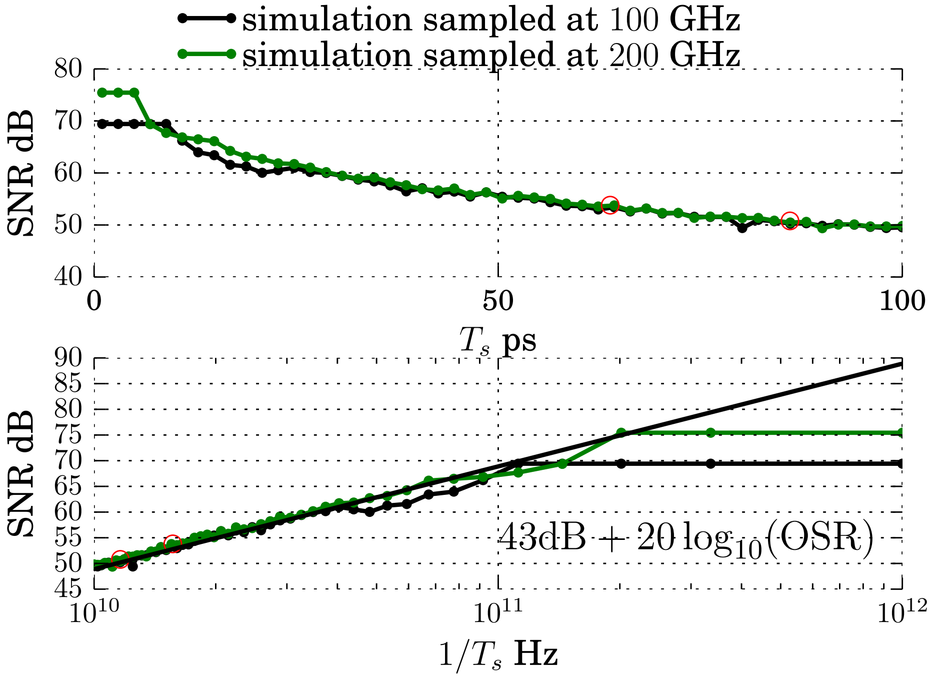


Figure 3.5: Top: linear view as a function of time resolution of the transmitter T_s, bottom: logarithmic view of the same data as a function of equivalent clock rate of the transmitter $1/T_s$. Simulated using 1 FMCW sweeps from 600 MHz to 2.67 GHz in 16.7 µs with 1 scatterers. highlighted in red is time quantized to 64 ps and 86 ps.

intercept (43 dB), deviates only 2 dB from the ideal gain-bandwidth-product of $10\log(2.07\,\text{GHz} \cdot 16.7\,\mu\text{s}) = 45\,\text{dB}$.

We note that this "noise" seen in the discrete transmitter simulations above, is entirely deterministic, and hence one should be able to subtract away, this "noise-floor". This is complicated by the fact that the simulation is optimistic, in that no system will be noise free and hence the deterministic part will be intertwined with true random noise. In addition, the exact amplitude of, especially the higher harmonics, is difficult to accurately predict as it depends on the circuits driving strength (fall/rise time of the digital signal). Currently, our only proposed solution other than oversampling, is presented in section 3.2.1 below, but we will first present the two main random contributions, namely jitter and thermal/white noise.

3.1.2.1 Jitter

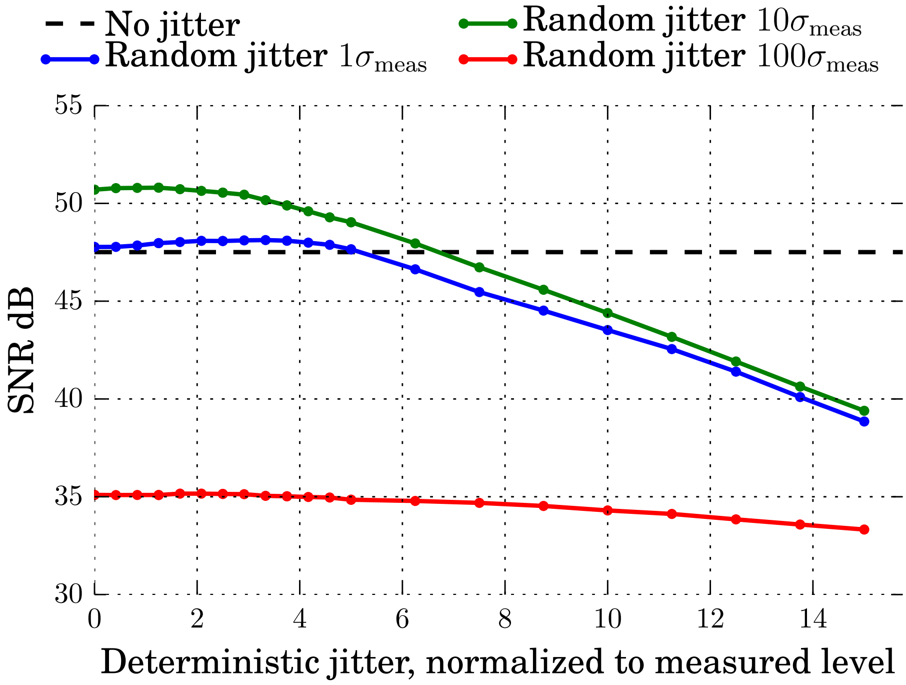


Figure 3.6: Simulating a non-ideal transmitter with both deterministic and random jitter (see Paper I). The blue line corresponds to the measured random jitter level. Using 16 FMCW sweeps from 600 MHz to 2.67 GHz in 22.6 µs time quantized to 86 ps with 1 scatterers.

If we include the effect of random variations in the transmitters an interesting effect emerges. As seen in figure 3.6, where we use the measured jitter levels and time resolution of Paper I, stochastic variations in the transmitter is actually beneficial for the SNR level.

This is a good example of Stochastic Resonance (SR) which we will see again in section 3.2, where a certain noise level can be beneficial to the performance. The skeptical engineer should keep in mind the previous sections on continuous and discrete time, the “benefit” we gain from the random jitter is only possible due to the performance hit we get when moving from a continuous to a discrete (and non-linear) system. The noise merely aids us in randomizing the coherent and deterministic error we make when we time-quantize the signal.

Figure 3.6 also underlines the statement made in Paper I that the time resolution is more important than accuracy and so the next chip iterations, which is detailed in Paper IV, improves the time resolution; without adding any feedback or other on-line calibration/filter methods.

3.2 Thermal noise

We have already alluded to the facts that (1) a single bit system can benefit from a certain “optimal” noise level and (2) the performance of a single bit system approaches a multi-bit or analog system when the noise level is high. It should also be repeated that thermal noise considered here, could just as easily be other non-coherent noise sources, such as interference or as just seen; jitter. Other noise sources may have different probability densities and as such see a slightly different result, but the main concept is the same. The interested reader is referred to the literature discussion we had on page 28 and the thesis by Hjortland [Hjo16].

A similar system simulation as before is presented in figure 3.7, where we compare both a fixed (that is, keeping the threshold level at the mean of the signal (0)) and a linear sweep where the threshold level is set to a uniform step between ±1 giving a serialized flash ADC. We also include the performance of a fully analog radar.

As should be expected from the discussion in section 2.2.2.3, figure 3.7 shows that at low noise levels an analog system is far superior to our time and amplitude quantized digital radar. It also shows that for input signal to

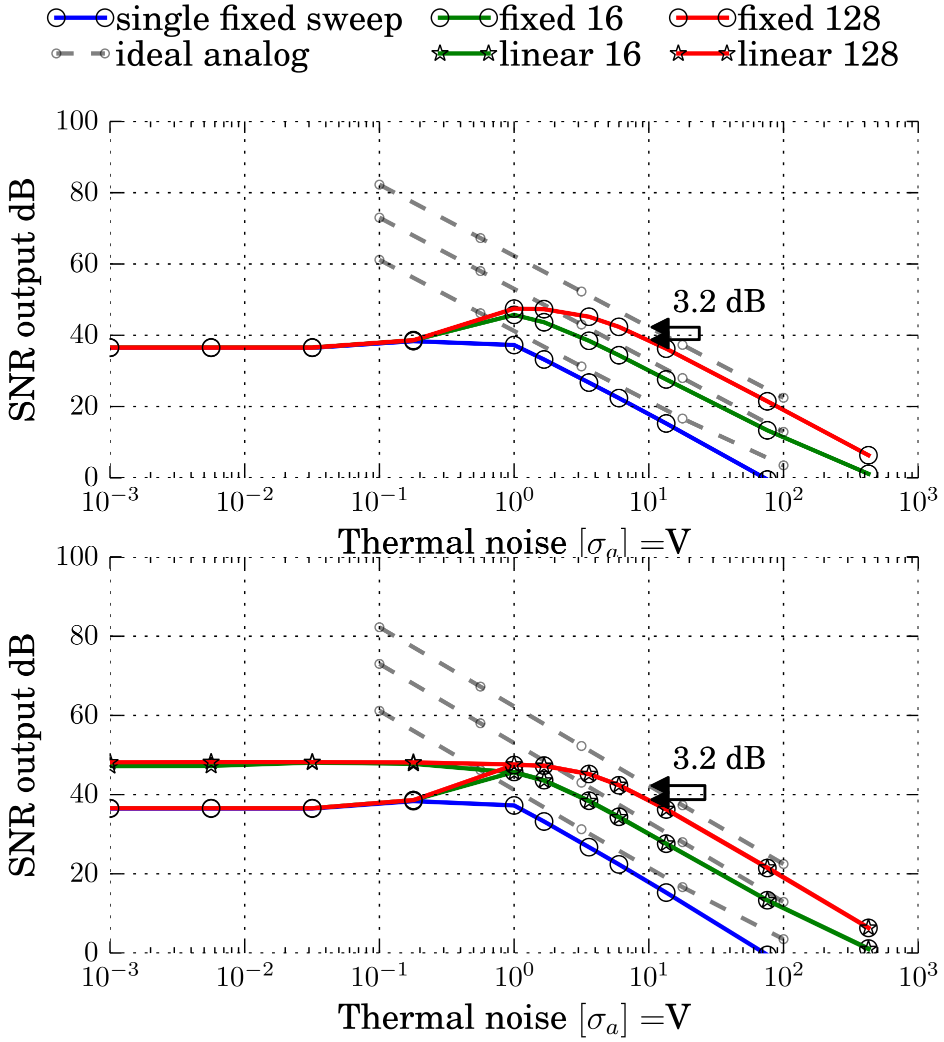


Figure 3.7: Fixed and linear sweeps for 10 targets with equal amplitude as a function of band-limited thermal noise. Simulated ideal analog radar shown as strippled lines. Using FMCW sweeps from 600 MHz to 2.67 GHz in 16.7 µs time quantized to 64 ps. Top panel shows only the threshold fixed at the DC level (0), showing the SSR effect, while the bottom compares fixed to linear distibuted threshold values.

noise ratios close to 1, there is almost negligible difference between a fully analog and our cheaper digital solution.

We also note a slight difference between the “linear” and “fixed” threshold modes, where the fixed threshold exhibit a SR phenomena coined Suprathreshold Stochastic Resonance (SSR), where again, the correct noise amount is beneficial (top panel of figure 3.7). As seen in figure 3.7 and considering that a fixed threshold at the signals mean is simpler to implement, a fixed threshold is preferred if we know the noise level is high (0 dB or less) while the linear threshold give the same performance and also works better when the noise level is low.

In the presented simulation, 10 scatters are used, if we use only a single scatterer then all of the low-noise simulations give the same results since both a fixed and a linear sweep keeps returning identical contributions regardless of the number of sweeps. Band-limiting the channel changes this some, as the linear threshold will return slightly varying duty cycled signals, but the concept remains the same.

As will be clearer in the next section, adding additional scatterers will increase the noise floor, because each target will carry with it the unwanted down-aliased harmonic intermixing and mixer sums. One possible solution to this is to de-correlate this unwanted noise.

3.2.1 Intentional dithering

A logical improvement, which has not been discussed in any of the published papers, is to add intentional randomness to the transmitted bitstream, which is the goal of this section.

With excessive randomness, we can even dither away the harmonics, the disadvantage of dither the harmonics away as opposed to the delay based methods given in the published papers is that the harmonic energy ends up increasing the noise floor. It can therefore be preferable to have the harmonic energy concentrated at multiples of the fundamental rather than spreading it evenly into an increased noise floor. A possible improvement would be to use some form of (Delta-Sigma) noise shaping, but this does require some method of filtering the shaped noise, which is non-trivial if we want to keep the single bit nature as will be seen in chapter 5.

Dithering is a logical step when we recognize that we repeat the measurement multiple times and to recognize that our time discretization is far

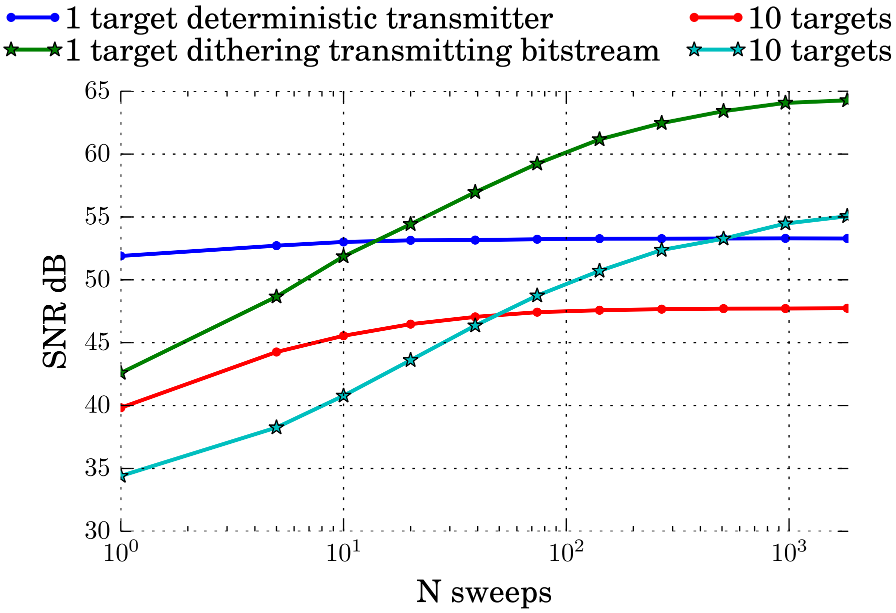


Figure 3.8: Simulated output SNR as a function of number of sweeps N, comparing a fixed bitstream being transmitted N times to a bitstream that is slightly different between each sweep.

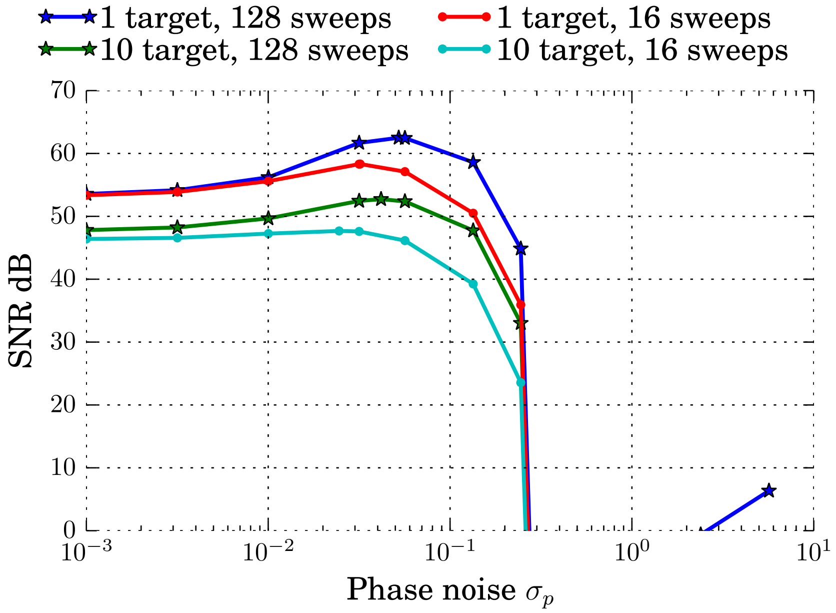


Figure 3.9: Simulated output SNR as a function of intentional dithering (phase noise when generating bitstream).

from ideal in that it leads to unwanted *coherent* noise. We therefore want to de-correlate this noise, so that it will reduce as we increase the number of unique sweeps. This is illustrated in figure 3.8 where the single target scenario without any form of randomness does not in any way benefit from sweeping multiple times without adding dithering.

A simple method to achieve this goal is to modify the bitstream slightly between each sweep, in the rest of this work, we calculate the bitstream values as

$$y[n] = \operatorname{sign}\cos(\phi_{\text{chirp}}(n\Delta t)) \tag{3.3}$$

where ϕ_{chirp} is given in (2.22) on page 26 and

$$\operatorname{sign} x = \begin{cases} -1 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}. \tag{3.4}$$

We dither this bitstream by re-calculating the following for each sweep iteration

$$y_{\text{dithered}}[n] = \operatorname{sign}\cos(\phi_{\text{chirp}}(n\Delta t) + 2\pi N(\mu = 0, \sigma_p)). \tag{3.5}$$

In essence, we are here adding intentional phase noise, where we for simplicity only consider a simple normal distribution $N(\mu, \sigma)$ with zero mean $\mu = 0$.

A detailed study on the optimal dithering to add is beyond the scope of this thesis, but figure 3.9 shows a simple sweep of the phase noise variance σ_p where we note a maximum around 0.1 regardless of the number of sweeps and targets. It is expected that the intentional dithering must take into account (unintentional) random circuit jitter to perform optimally.

3.3 Moving targets

We have up to this point assumed targets are stationary while the measurement takes place, we will in this section briefly explore what happens if this is not the case. As we will see in section 3.3.2, this section also touches on a 2D imaging radar, as one of the possible processing steps are identical. We start of by mentioning tracking in section 3.3.1 and we end up with Doppler frequency shift in section 3.3.3.

3.3.1 Tracking

For completeness, we here mention *tracking*, as a method of extracting the velocity and range of moving targets. As seen in figure 3.10, tracking is applied much later in the processing chain than the main focus of this thesis. Tracking can be applied without coherent phase data, as it can simply work on thresholded (e.g. CFAR) signal envelopes. This makes the radar simpler to implement and reduces the information rate. Usually some form of application dependent information is used to divide scatterer information from the radar, into clutter and target bins. Such as an expected velocity range, scatter magnitude and statistics over time.

Tracking can be applied to a 2D image in space as illustrated in figure 3.10, where separate frames can originate from multiple radars at different physical locations, a single radar that moves or a single rotating radar/antenna. One

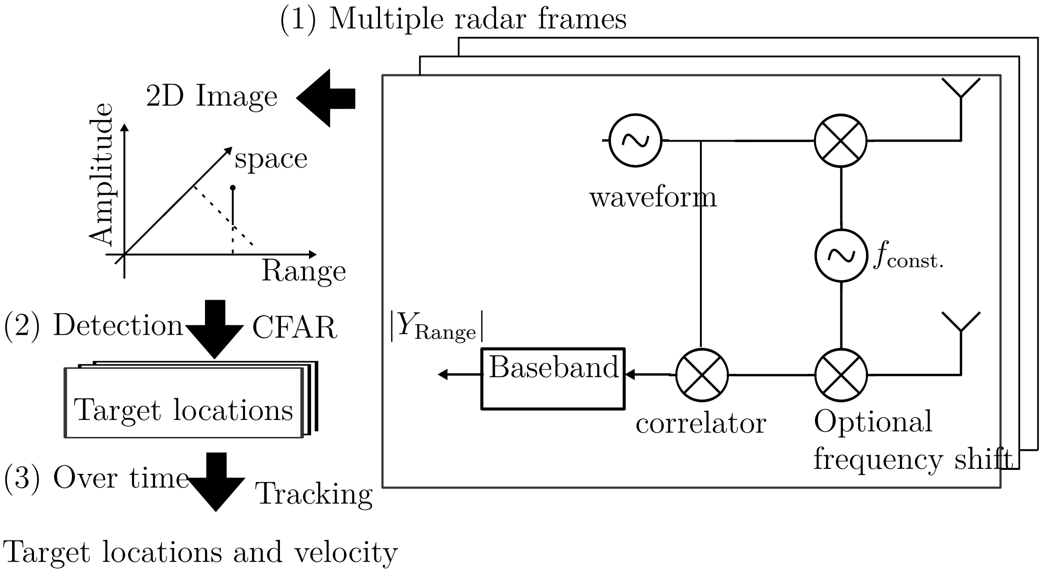


Figure 3.10: Principle of incoherent radar tracking. A (possibly 2D) image is created by combining frames that are separated in space (can be angle or position), this data is then processed by a detection step to yield possible target locations. If we now compare target locations over time we can employ tracking algorithms in software to yield both location and velocity (including a heading if our data is 2D).

could also apply tracking to a 1D image, yielding a velocity only in the radial direction. An example incoherent tracking paper can be found in [GNB$^+$14].

One can also apply tracking to coherent data, each “expected” velocity and range profile is then matched against the measured data and a map that is proportional to the match between expected profiles and measured responses can be generated. See section “Correlation Processing” in [Wei94, page 19-20] for an example matched filter implementation.

As an alternative to a 2D image in space, we can apply one of the two procedures below to create a 2D image in Doppler/velocity; by either frame processing as seen in the next section or by Doppler shift as seen in section 3.3.3. This could then be passed to a tracking algorithm to follow targets over a longer time period.

3.3.2 Comparing frames in time

If the targets moves sufficiently slow in respect to the measurement time, we can extract the targets velocity by comparing different measurements (frames) in time. The assumption is here that the target does not leave the range resolution cell before we can take another measurement and additionally, that the signal does not experience a significant Doppler frequency shift “on-air”; as will be discussed in the next section. This gives us a scenario that from a processing stand point is identical to an imaging radar, where instead of comparing measurement “frames” in time we compare frames separated in space.

A computation efficient method to compare frames is illustrated in figure 3.11 and consists of taking a 2 dimensional FFT; one to obtain the range view and another to obtain a velocity, combined these will give a range-velocity map. Note that this assumes we have coherent phase data to work with and that the spacing (in either range or time) is uniform.

As was discussed in the Background chapter, all of the presented radar architectures will result in a pulse, regardless of the waveform used to probe the environment and as such all of the presented architectures will behave the same for slow moving targets (though the measurement time required for each architecture will differ and as such the definition of a “slow target”). The only difference in the procedure outlined in figure 3.11 is that for a correlator based radar the range FFT is replaced by the correlator, while a pulsed radar can skip this step altogether. In a mixer based radar the pulse

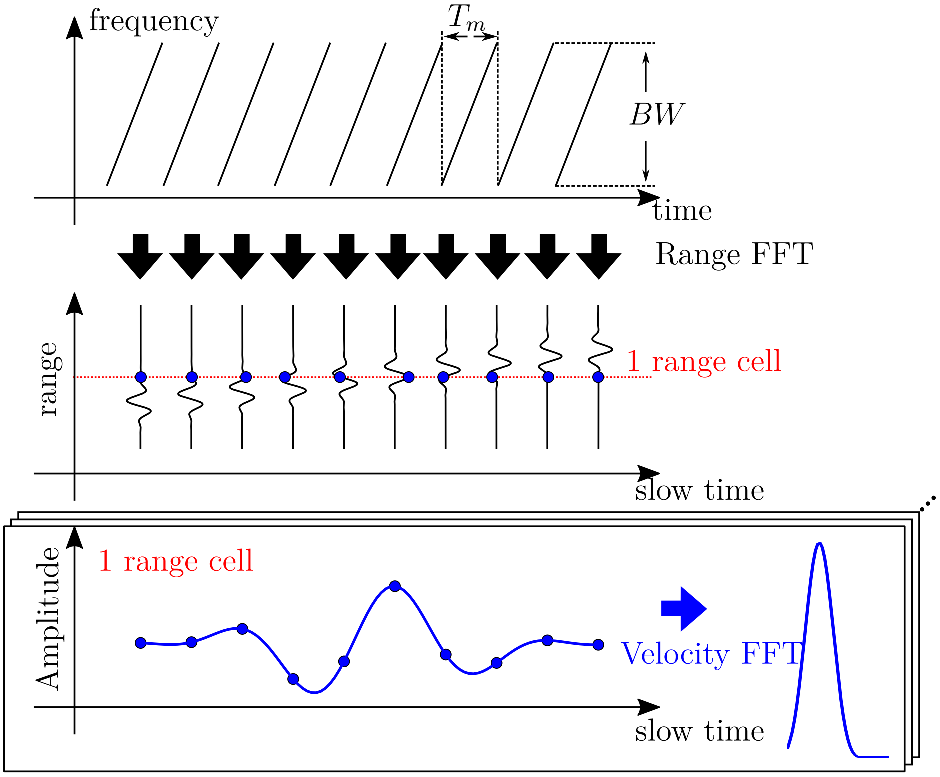


Figure 3.11: Principle of a FMCW 2D Fourier transform, figure inspired by [HJL16]. Top: transmitted chirp signals. Middle: Each frame is mixed and Fourier transformed with a range FFT. Bottom: a new FFT is taken at every range cell over the gathered frames to give a range-velocity map.

shape is determined by the window applied in the range FFT (a Hanning window was used to create figure 3.11), while a correlator radar will depend on the channel bandwidth as seen in figure 2.10. In a pulsed based radar the pulse shape is often determined by regulations and hardware. We therefore note that a mixer based radar with a range FFT is the most flexible as we can simply swap window function in software to trade-off resolution and sidelobe levels.

For our digital swept-threshold radar the principle in figure 3.11 should be adjusted. As discussed in the previous chapter, using a single threshold will cause non-linearity and inter-mixing between targets in high SNR scenarios while low SNR scenarios benefit from averaging multiple frames. We must therefore use multiple sweeps to obtain a single frame. With this in mind, a single frame will take

$$T_{\text{frame}} = N_{\text{sweeps}} \cdot T_m \tag{3.6}$$

to complete. The target must not leave the range cell before we can make another measurement, so we must require

$$v_R < \frac{\delta_R}{N_{\text{sweeps}} \cdot T_m} \tag{3.7}$$

$$\text{where} \quad \delta_R = \frac{c}{2BW}. \tag{3.8}$$

A numerical example with a $BW = 1\,\text{GHz}$ chirp taking $T_m = 10\,\mu\text{s}$ and repeated 300 times yields a maximum velocity of $v_{R_{\max}} = 50\,\text{m/s} = 180\,\text{km/h}$ in air.

In addition, the sweep threshold principle assumes the scene is stationary and so the maximum velocity should be treated with care; as we might see smearing and in the worst case non-linearity. A detailed study is beyond the scope of this thesis.

In the lower range, we are limited by the sampling interval and observation time. A Doppler FFT will have a Doppler frequency resolution

$$\delta_{f_D} = \frac{1}{N_{\text{frames}} \cdot T_{\text{frame}}} \tag{3.9}$$

giving a velocity resolution of

$$\delta_v = \delta_{f_d} \cdot \frac{c}{2f_c} \tag{3.10}$$

where f_c is the sweep center frequency. Targets moving slower than δ_{f_D} will be difficult to distinguish from stationary targets. With the same numerical example as above around $f_c = 500\,\text{MHz}$ and using $N_{\text{frames}} = 10$ frames we obtain $\delta_v \approx 10\,\text{m/s}$. As with the range FFT, this velocity resolution will be degraded if we utilize a non-rectangular window when doing the velocity Fourier transform.

3.3.3 Doppler shift

If, on the other hand, the target moves fast with respect to the measurement time; the utilized waveform will see a frequency shift, called a Doppler shift. The output response now depends on the waveform used, in particular for a FMCW radar where the frequency shift can become indistinguishable from a target delay (range). We are now at a point where we can extract velocity from a single measurement.

To compare waveforms the "ambiguity function" tells us how a particular waveform will look like on the receiver for a single target at various velocities and positions. It gives us both range-ambiguity and velocity ambiguity, determined mainly by the Pulse Repetition Frequency (PRF) as a fast PRF gives ambiguity in range while a slow PRF give undersampling of moving objects and hence ambiguity in velocity. The ambiguity function also tells us the signal energy at a "mismatched" velocity and range, which for the range view is the range-sidelobe structure. Allowing us to extract the effective resolution and sidelobe level in range and velocity. [ULB+14].

The wideband ambiguity function is

$$\chi(v, R) = \sqrt{\eta} \int_{-\infty}^{\infty} s(t) s^* \left(\eta \left(t - \tau\right)\right) \, dt \tag{3.11}$$

$$\text{where} \quad \eta(v) = \frac{c - v}{c + v} \tag{3.12}$$

and reduces to the narrowband ambiguity function for non-relativistic $\eta(v) \approx 1$, constant velocity and small time bandwidth product $2v/c \ll 1/(BWT_m)$ [Wei94]

$$\chi_n(v, R) = \int_{-\infty}^{\infty} s(t) s^* \left(t - \tau\right) e^{-j2\pi f_D t} \, dt \tag{3.13}$$

$$\text{where} \quad f_D = v \cdot \frac{2f_c}{c} \tag{3.14}$$

where f_D is the (approximate) frequency shift of a narrow-band signal. We note that we go from a time scaling of $\eta \cdot t$ to a frequency shift $e^{-j2\pi f_D t}$, where the narrowband assumptions is that all frequencies are shifted equally [Wei94]. Some authors further restrict the ambiguity function to the absolute value $|\chi(v, R)|^2$, but we will see that the real part can give some intuition.

Figure 3.12 shows a simulated wideband ambiguity function of an analog and a single bit amplitude quantized radar. The figure is created by a series of simulations, one for each velocity for a given target range. Each scatterer return is found by a velocity scaled chirp

$$s_{\text{chirp}}(\eta(t - \tau)) = \operatorname{sign} \cos(\phi_{\text{chirp}}(\eta(t - \tau))) \tag{3.15}$$

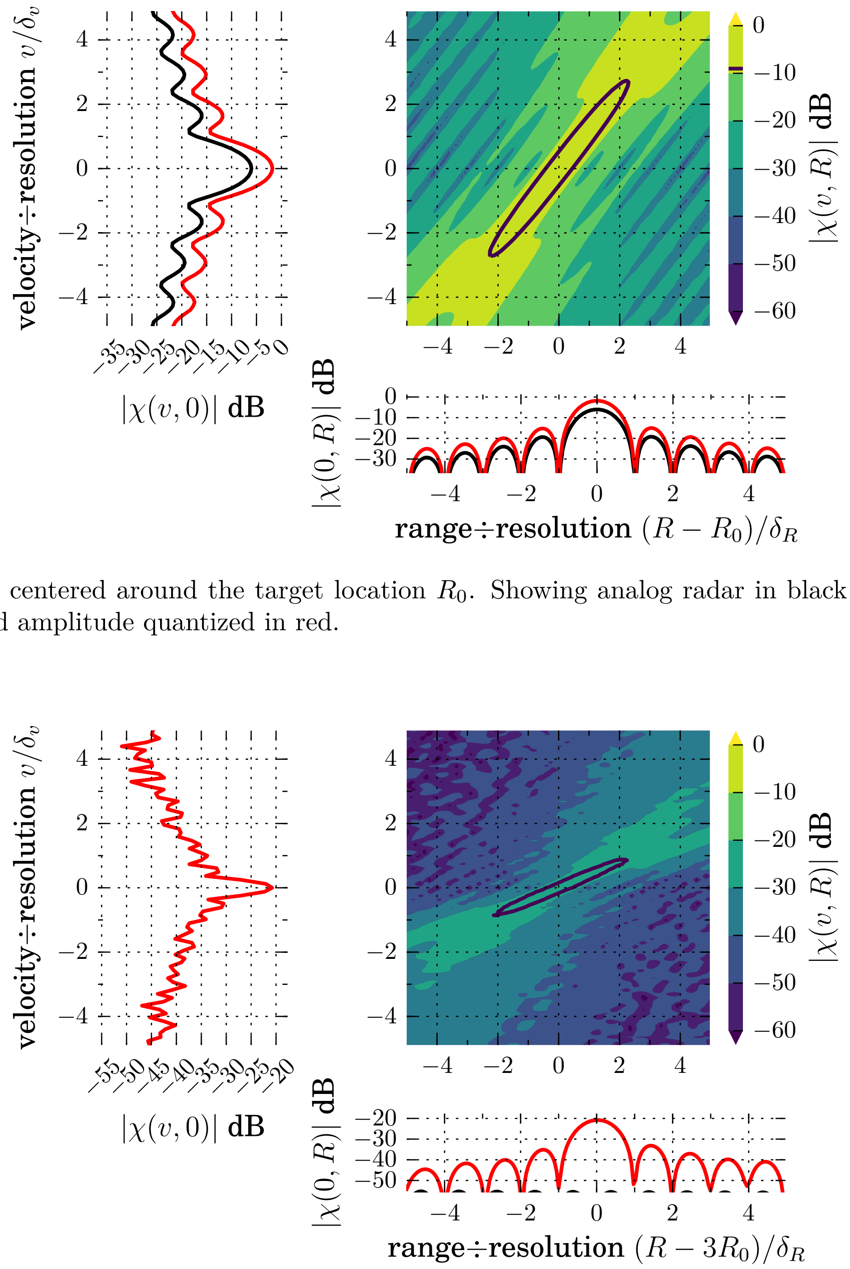
where the analog radar is simulated without the sign function. We then obtain the range profile in the usual way of mixing and then Fourier transforming (with zero padding). As we only consider a single scatterer, we will; for further simplicity, only consider a single sweep with the receiver threshold fixed at 0. As discussed in the previous chapter, only using a single threshold is acceptable for situation when we only have a single scatterer.

In figure 3.13, we show three range slices, one at zero Doppler and two at the velocity $\pm\delta_v$ given by (3.10), while also including a “ridge” cut along the FMCW range-velocity coupling

$$v_{\text{ridge}}(R) = \frac{BW}{f_c T_m} \cdot R. \tag{3.16}$$

Figure 3.13 includes both the absolute value of the ambiguity function and the real part, where the latter is equivalent to the time domain (compressed) output pulse.

Closed form expressions for the narrowband ambiguity function is available for a range of waveforms, see e.g. [Lev88, chapter 7]. The wideband



(a) centered around the target location R_0. Showing analog radar in black and amplitude quantized in red.

(b) centered around the third harmonic at $3R_0$.

Figure 3.12: Simulated wideband ambiguity of a digital and analog FMCW radar without any window applied to the FFT (boxcar). Normalized in velocity by the doppler resolution δ_v and normalized in range to the range resolution δ_R.

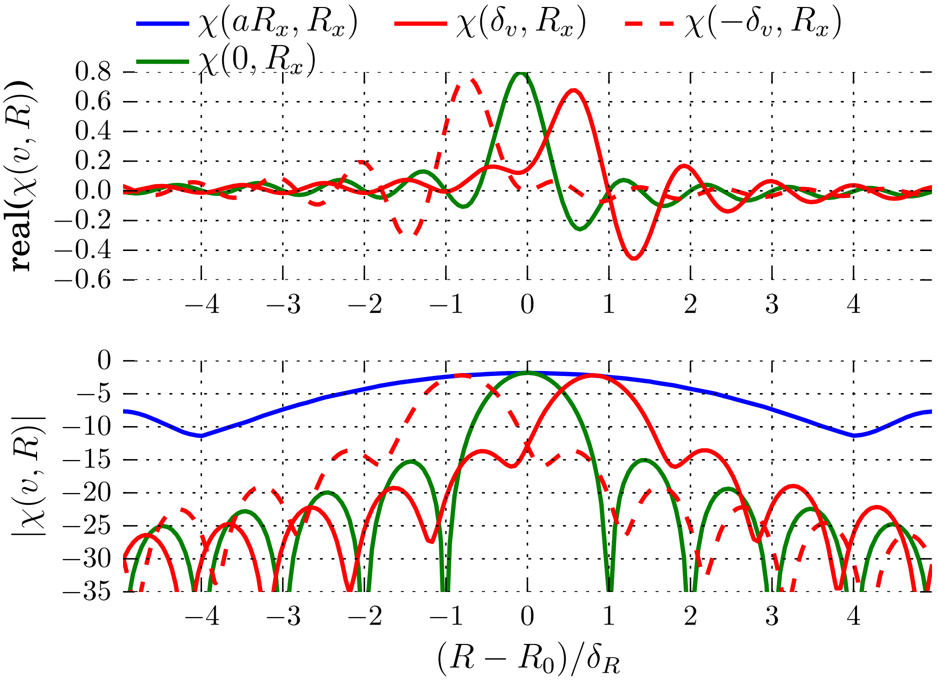
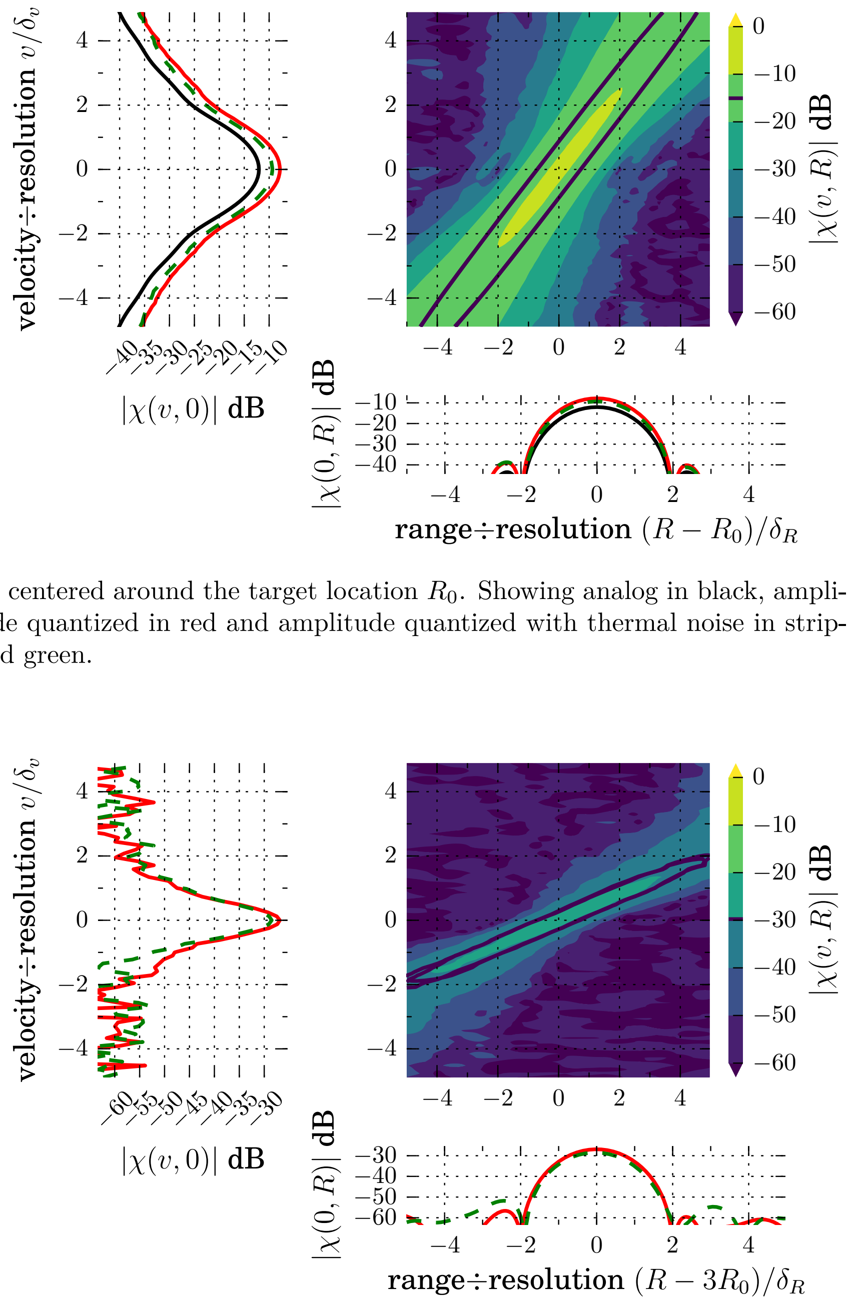


Figure 3.13: Simulated ambiguity cuts for a digital FMCW radar, at zero doppler (in blue), at some positive and negative doppler (in red) and along the FMCW ridge (in blue). Found by interpolating figure 3.12.



(a) centered around the target location R_0. Showing analog in black, amplitude quantized in red and amplitude quantized with thermal noise in strippled green.

(b) centered around the third harmonic at $3R_0$. Showing amplitude quantized in red and amplitude quantized with thermal noise in strippled green.

Figure 3.14: Same as figure 3.12 but using a Hanning window and also including simulated thermal noise to a input SNR of 0 dB.

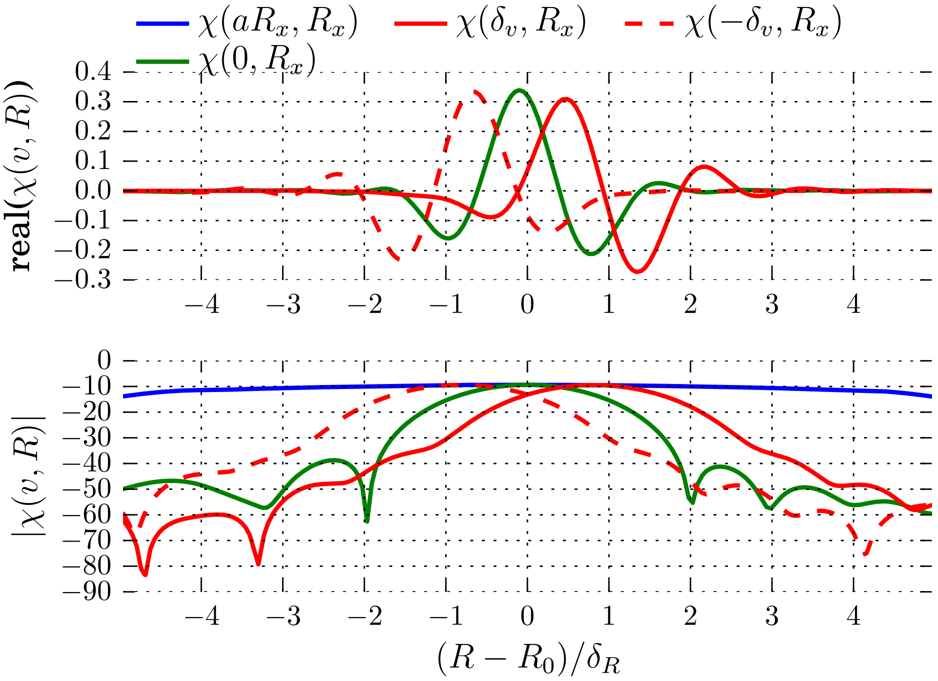


Figure 3.15: Same as figure 3.13 but using a hanning window. Showing the wideband ambiguity function at zero doppler (in blue), at some positive and negative doppler (in red) and along the FMCW ridge (in blue).

ambiguity function is not readily available as it is more challenging and not always necessary. We have therefore opted for a purely numerical study here.

To show the flexibility of this numerical study, we can include additional radar system effects, such as the window utilized in the range compression and, as an example, thermal noise. These effects are shown in figure 3.14 and figure 3.15. We note that the thermal noise (green stippled simulation) only creates a slight deformation of the response at the harmonic (figure 3.14(b)), while the Hanning window (applied in the range domain) actually reduces both the effective range and velocity resolution.

For the parameters utilized here, the velocity resolution (and hence the simulated velocities) are on the order of,

$$\delta_v = \frac{c}{2 f_c T_m} = 5 \times 10^3 \,\mathrm{m/s} \tag{3.17}$$

which is almost 16 times the speed of sound! This velocity therefore necessitates a wideband ambiguity function since the narrowband assumption

$$\frac{2\delta_v}{c} \ll \frac{1}{BWT_m} \tag{3.18}$$

is comparable in magnitude. We believe the analysis holds for more reasonable velocities and therefore normalize the figures to δ_v.

For the Doppler shift to become significant at reasonable speeds, we must either be working with

- A longer chirp sweep time, such as 1 ms-10 ms, and/or:
- A higher center frequency, by e.g. the arrangement in figure 2.3 to mix the center frequency up in the 60 GHz-100 GHz range.

Lastly we should emphasis that we only consider FMCW here, the ambiguity function will be different for other radar architectures and the reader is referred to the extensive literature.

3.3.4 Summarizing moving targets

To conclude our discussion on moving targets, we can use the following checklist when faced with moving objects

- Does the waveform exhibit significant frequency shift “on-air”? Study the ambiguity function: for a FMCW radar the velocity obtain by (3.17) gives an order of magnitude estimate.
- Is the expected velocity and PRF such that a frame by frame comparison of the phase makes sense? Here (3.7) and (3.10) gives an estimate.
- Is the radar incoherent (without any phase)? Or is there a need to classify and track targets over a long period? Then tracking algorithms must be applied.

Chapter 4

Implementation

We will in this chapter walk through the fabricated 90 nm CMOS chip used to verify the proposed single bit digital radar concepts.

Measurements from 2 chip runs can be found in the papers, where Paper I presents measurements from the first chip which included the “waveform generator”, which in essence is a clock-free 64 to 1 MUX. Paper II lays the foundation for Paper III; which include measurements for the second chip which includes a simple receiver with a quantizer, XOR gate (as mixer) and a counter (as a LP filter). The first and last chip differ in the transmitter and the difference (one or two inverters) is presented and discussed in Paper IV; where we have taken a single “row” of the waveform generator and presented it as a Digital-to-Time Converter (DTC). We will in this chapter focus on the second chip.

4.1 System overview

The proposed radar solution was shown in figure 1.5 and a simplified implementation schematic is displayed in figure 4.1. Transmitter is implemented as a memory to hold the waveform information and the memory content is first serialized to 4 single bit lines which are combined with a OR operation. On the receiver side, the comparator for the swept threshold operation is visible, followed by a XOR gate as a mixer and a counter as a low-pass filter. Also visible in the diagram is the two path delays, implemented as a cascade of inverters where the “tapping point” is selectable with a mux.

Serial Peripheral Interface bus (SPI) is used to reduce the pin count and controls various digital aspects of the chip. Figure 4.1 presents the top level diagram of the second chip. Not accounting for the miscellaneous debugging pins, the chip features a

- $\mathrm{TX}_{\mathrm{delay}}$, which is the waveform generator output with a programmable delay.
- RX and I_{comp} as the inputs to the comparator and LNA.
- IF for the XOR output bitstream
- Sample for the CW receiver counter/integrator.
- 4 SPI pins for reading and writing to all registers on the chip. Including loading the on-chip memory with a wanted bitstream.
- Transmit signal to start/stop the waveform generator.

A chip photo, giving a sense of the size and placement of each block, is shown in figure 4.2. Each block has its own supply voltage and ground, in particular the transmitter, receiver and digital logic is separated within its own deep-N-well. The different domains, analog/mixed signal comparator, continuous time mixer, asynchronous transmitter and conventional digital blocks (memory and SPI interface) are placed as far apart from each other as feasible, to minimize coupling. Care is also taken in the routing, by keeping critical lines short and separating clocked digital from the critical analog and continuous time connections.

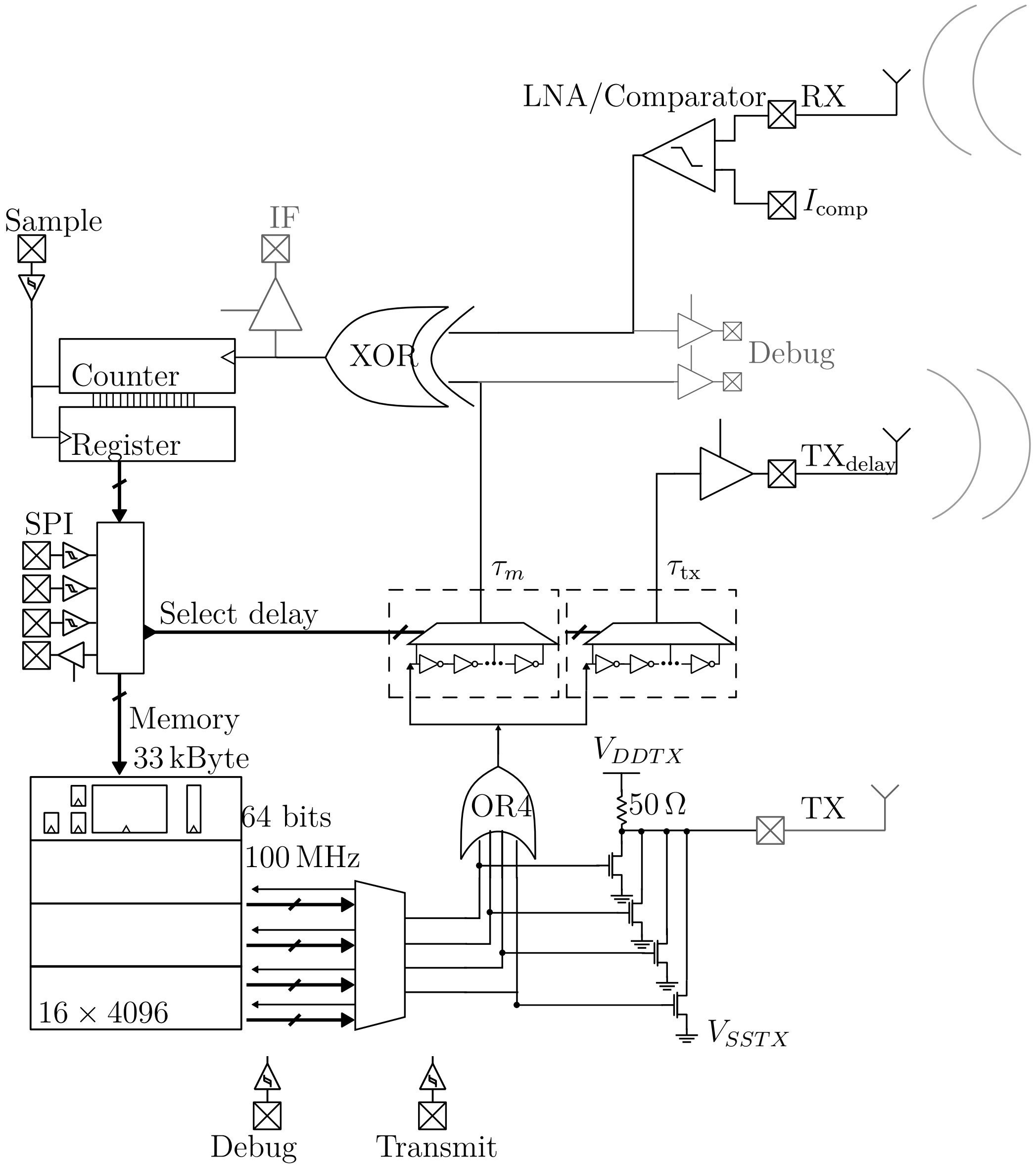


Figure 4.1: Schematic for the prototype single chip square wave radar with the CW receiver. The TX signal is available either as a delayed version and, as a debug measure, available with the common source OR connection in the lower right. Starting at the lower left we have the waveform memory, serializer and the two OR implementations that can be used as TX. Starting at the upper right the RX is compared to an externally set I_{comp} before being mixed (XORed) with a (possibly delayed) version of the transmitted signal.

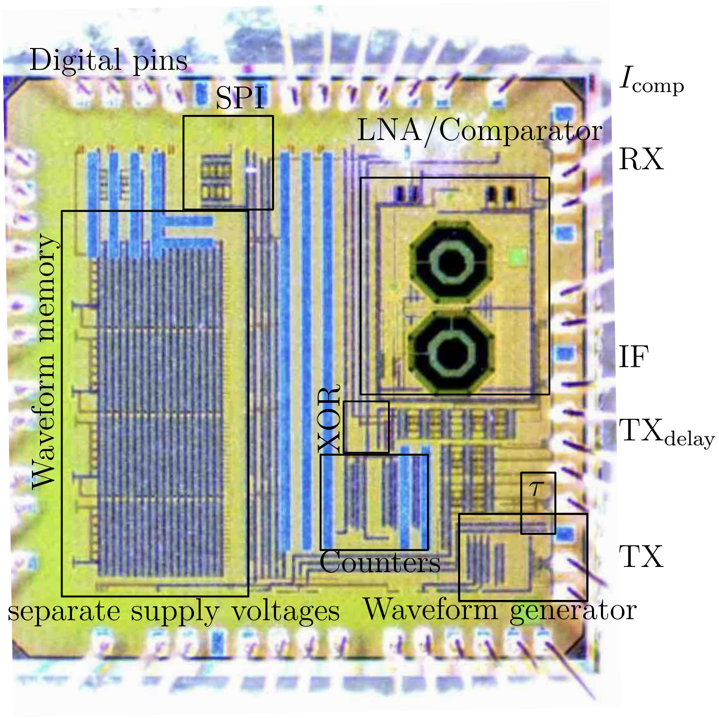


Figure 4.2: Chip photo of the single chip square wave radar with the CW receiver with bonding wires for a QFN48 package. The chip measures 2 mm × 2 mm

4.1.1 SPI configuration

The on-chip SPI module was created by Hjortland [Hjo16] and so was part of the software running on a connected single board computer. The module allows us both read and write to every digital registry on the chip while taking up only 4 chip pads. The primary advantage over a naive shift register implementation is that we can separately address only the (8 bit) register we want to read (non-destructively) or write to.

A speed issue was faced when programming the entire 33 kByte memory, as the single board computer (a Raspberry Pi 3) ended up having long pauses between each SPI command. This could be improved both in software (by sending larger "packages") and in the chip implementation, by using separate pins dedicated to memory programming using some custom made communication protocol. This speed issue is mainly a disadvantage for the SFCW radar, as it relies on re-programming the chip for each frequency step. It could also be problematic for the dithering method in section 3.2.1 as that to relies on modifying the memory content for each sweep.

4.1.2 Transmitter

The transmitter consists of a memory bank to hold the bitstream waveform, the waveform generator from Paper I, two methods of OR-ing the waveform generator output and programmable delay.

4.1.2.1 Memory bank

The memory consists of 4 banks that are 16 bits wide and 4096 bits deep, resulting in a total storage of around 33 kBytes, the width was determined as a tradeoff with the waveform generator to slow down the required memory speed while the depth was mostly limited by area. Memory modules where supplied by TSMC as a "black-box" with only a Verilog timing model and a layout placeholder.

The memory bank from TSMC is supplemented with some simple logic for reading and writing the register from SPI and a counter to automatically increment the address when reading. These where implemented using standard synchronous design principles.

With a readout with an effective clock rate of 64 ps, the memory can hold a waveform of 17 µs in length, at which point the waveform will repeat. The "clock rate" is decided by the waveform generator and will therefore scale with process, voltage and temperature of the waveform generator and can be tuned with the backgate voltage as discussed in Paper IV. As the memory is effectively $4 \times 16 = 64$ bits wide, the memory works at the reduced speed of 15.7 GHz/64 = 245 MHz which is acceptable for on chip memory but excludes using external memory (especially when considering the required 64 data pads).

4.1.2.2 Waveform generator

A high level description of the waveform generator is presented in Paper I while a lower level circuit description is the focus of Paper IV and the reader is refereed to these papers for details. We will here briefly comment on the asynchronous nature of the serializer as it connects to the memory bank discussed above and influences how the radar is used before showing the Monte Carlo results with backgate tuning.

As the chip does not feature *any* absolute time reference, it is up to external circuity to extract the absolute clock rate if required. In this project this is done by either measuring a known (short) sequence with a high speed oscilloscope or by processing the baseband signal and looking for the discontinuity when the waveform repeats. A useful feature that was not added is to bring either the "overflow" memory counter value or the memory bank clocks out of the chip, which would provide a third alternative for extracting the clock rate.

A further refinement to this would be on-chip circuitry to adjust the (backgate) tuning voltage dynamically to match some external stable clock reference. In addition, the frequency estimation technique in section 2.2.1.1, implemented by a simple count-and-dump; could be used as a simple to implement on-chip test and calibration method.

Note that to achieve coherent integration we do not require an absolute time reference and so an asynchronous system is fully compatible with a coherent radar. The only requirement is an accurate trigger signal, which is also non-trivial. To work around an accurate trigger signal an additional processing step was done, the baseband signal was filtered and interpolated before short segments where correlated to figure out the offset before aligning and averaging. This processing step was used in the creation of figure C.12.

An absolute timing reference was not needed in this work and working around the fact that no absolute timing information was available proved to be feasible. This also exemplifies the flexibility of a digital system as one can mitigate circuit non-idealities in software.

That said, one downside of an asynchronous waveform generator as implemented in this thesis is that a duplicate generator cannot be expected to run in phase and with the same absolute clock rate. A duplicate generator can be useful as it eliminates the need for delaying the reference signal, as one could instead use two separate generators with an offset in initial state as exemplified in[Sac13, figure 3.34]. In addition, two separate chips cannot be expected to transmit the exact same waveform, in the very least necessitating a calibration step for some imaging applications.

One weakness, not discussed in the publications, is the synchronization mechanize that we use to select which of the four rows should be the transmitting row, shown as the “select loop” in figure A.4 and figure A.6 around page 150. The concept is based on duplicating the chain of muxes and assuming equal delay, each row was assumed finished when the select pulse had propagated through the same number of equal mux-stages. This equal delay assumption proved to be acceptable and worked surprisingly well, as no significant correlation in the jitter was found when the transmitting row is switched. In addition, any change in temperature (not measured) and voltage, affects both the reference delay and the transmitting row, ensuring that the delay stays matched over chip-to-chip process variations, temperature and voltage.

What did cause some issues, but again worked well enough for us to keep it in place between the first and last chip prototype, is that the pulse that is supposed to circulate continuously in the “select loop”. It did however have a tendency, on some chips, to either shrink or grow until it disappears; stopping the entire transmitter. This was the main use of the backgate tuning, as each chip required a slightly different backgate voltage on the wavefront-generator part to continue the circulation. This tuning was done manually, but as the main objective was to visually see on an oscilloscope if the circuit stopped or not, and as the tuning was not particularly sensitive, could be easily performed by any slow on-line digital calibration loop that simply checks for output activity.

To characterizes the ability of the backgate tuning to mitigate the pulse from disappearing, a set of Monte Carlo simulations was done. For each Monte Carlo seed, multiple backgate and supply voltages was iterated over:

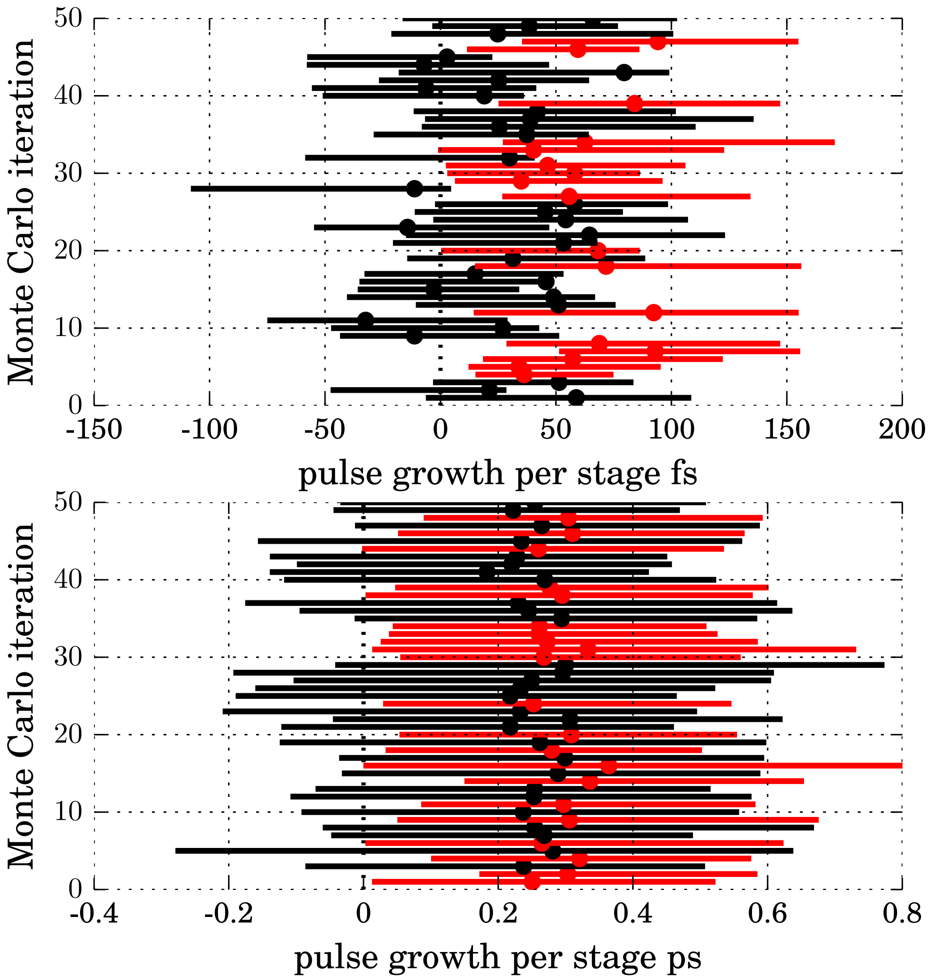


Figure 4.3: Showing the tuning potential for each MC scenario. Dot marks the nominal (no tuning). Red lines denote chips that are not sufficiently tunable (does not include 0 pulse growth). Top: Chip 2 (one inverter), bottom: chip 1 (two inverters). Note the difference in x-scale.

- positive bulk VDDB$\in \{0.5, 0.8, 1.2, 1.5, 2\}$
- negative bulk GNDB$\in \{-0.2, 0, 0.4\}$
- supply voltage VDD$\in \{1.1, 1.2, 1.3\}$.

The resulting tunability range for each of the 50 MC iteration is shown in figure 4.3. We have colored the scenarios that have the ability to be tuned to 0 pulse growth black, while those with only positive pulse growth is colored red. Do note the x-axis, in that when moving from the non-inverting design (two inverters) to an inverting design (one inverter) as presented in Paper IV, we see an order of magnitude decrease in variation, but, unfortunately an equal order of magnitude decrease in backgate-tunability. We discuss a workaround for this lack of tunability in Paper IV, but this was never implemented.

To summarize, the novel waveform generator, in essence a serializer, does show both the great benefits of an asynchronous design but also show design challenges that are not present in a conventional clocked design.

4.1.2.3 Programmable delay

The programmable delay originates from the work of Dooghabadi [Doo15], but with a corrected mux to avoid pulse swallowing. The cell can be programmed to $2^6 = 64$ different delays with a step of 2.9 ns in the nominal corner.

Using the delay on the transmit path (τ_{tx} in figure 1.5), this delay cell gives a free-space unambiguous range of 28 meters if we use the “moving out of band” harmonic removal technique and 64 different delay settings for the staggering technique. Using it on the mixer path (τ_m in figure 1.5), gives us 64 different range cells between 0 and 28 meters that we can down-convert to DC for a sequential sampling architecture as mentioned around equation (2.26) on page 28.

We note that for the primary use of this delay (on the transmit path), the delay accuracy is not critical, as the moving out of band technique simply needs some large delay, while the staggering technique only needs a set of delays that are different from each other.

4.1.3 Receiver

The receiver only consists of the comparator for the swept threshold receiver and a XOR gate for mixing the transmitted and received signal.

4.1.3.1 LNA/comparator

The comparator acts both as a gain stage, to amplify the backscattered signal and as a comparator yielding a continuous time digital output. This circuit block originates from Novelda, with some modifications by Tuan Anh to get a current instead of a voltage threshold input. A simplified diagram is presented in figure 4.4. We note the two inputs, RX and I_{comp}, the threshold current is converted to a voltage which is used to shift the DC point of the amplified RX input signal. The gain of the first stage is adjustable, since we do not want to clip the signal before the correct threshold is set, this is not an issue for the last stage, where we want the output to clip.

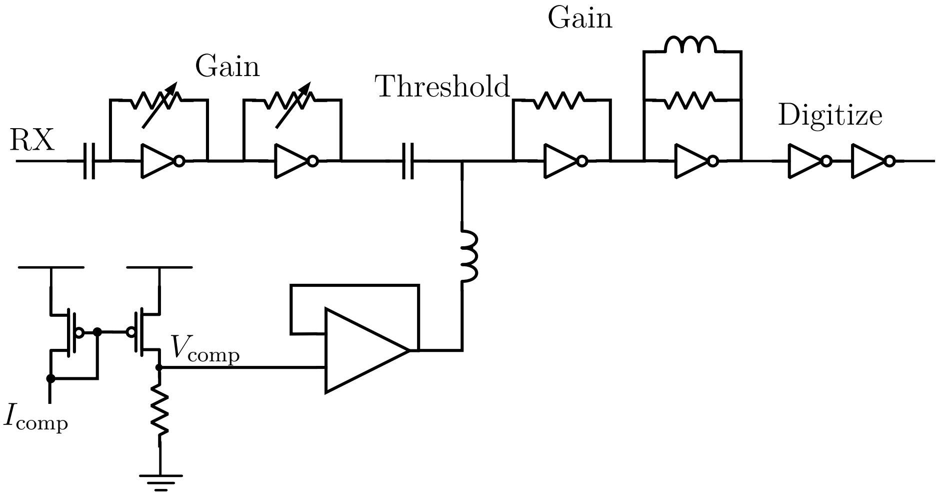


Figure 4.4: Simplified diagram of the thresholder/comparator circuit from Novelda and modified by Tuan Anh. The circuit consists of an initial (digitally tunable) gain stage before the DC level is shifted by the comparison voltage V_{comp} before being amplified again and “digitized” by conventional digital inverters.

Characterizing this mixed-signal (analog inputs and digital outputs) and non-linear devices can be a bit unusual. As the device is not-linear and since we are interested in a large-scale output, a small-signal AC or S-parameter analysis is out of the question. A transient analysis is therefore selected here, though a harmonic balance simulation would most likely be faster. We start by finding the correct threshold (current), this is achieved here by a python script that (1) modifies the Spectre netlist (2) runs the simulation and (3) analyses the output. This is repeated until an adequate threshold is found (to within 30 µV). These steps are then repeated for a range of input (peak-to-peak) amplitudes and frequencies.

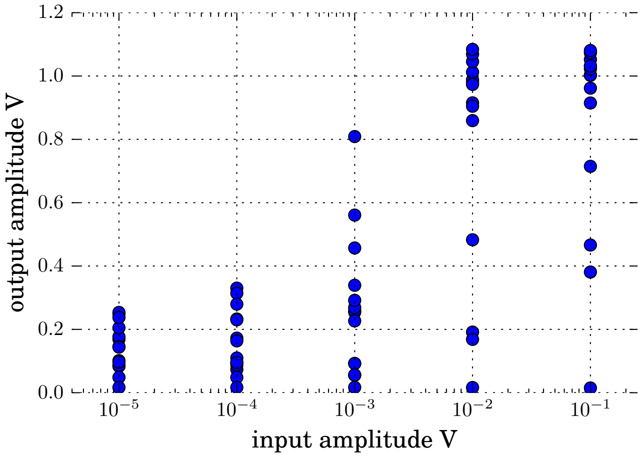


Figure 4.5: Post layout simulation of the quantizer, showing the input and output amplitude of the fundamental. Each point represents different frequencies, where for each point, the I_{com} is optimized to maximize the output amplitude.

The results of the above analysis are shown in figure 4.5 and figure 4.6. In the first image, we show the 5 input amplitudes that was simulated and the corresponding peak-to-peak output amplitude (at the fundamental frequency). This highlights a big challenge; the circuit block is able to take a signal from 1×10^{-5} V $= -100$ dB to 0.25 V, an 88 dB gain, but a 0.25 V signal is not particularly useful to any digital gate interpreting the output. That said a digital gate does have quite good gain and can conceivably take the signal up from 0.25 V to the supply voltage of 1.2 V, but this assumes a correct DC level and is therefore not ideal.

With this in mind, we create a series of gain lines in figure 4.6, each with a different “cutoff”, of what we consider as a useful output signal. Requiring a 1 V peak-to-peak output, we note a gain of at least 40 dB (but not greater

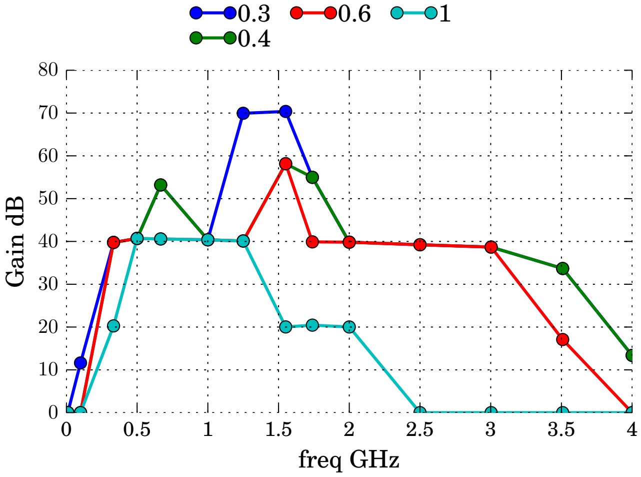


Figure 4.6: The same data as in figure 4.5 but simplified to a gain plot as a function of frequency. Each line uses a different definition of "useful" gain, where data-points below the required output amplitude is removed. Note that the gain takes on discrete peak values only because we test with discrete input amplitudes.

than 50 dB) between 0.5 GHz and 1.25 GHz. The interpretation of this plot is a bit unusual, as we are only testing with a discrete set of input amplitudes and discarding gain figures if the output amplitude is too low.

The above analysis does, more or less, ignore noise, which should be beneficial in the correct amount for weak-signal detection in a SR manner. Further emphasizing the complexity of quantifying the comparators performance.

4.1.3.2 XOR gate

We briefly show in section 5.3.1, that a multiplication of two numbers that only takes the values $\{-1, 1\}$ can be implemented by a single XOR gate. To implement the XOR function, there exists a large number of transistor-level implementations, especially considering that one could potentially implement it as an analog mixer with digital buffers on inputs and outputs or utilize the vast literature on phase comparator utilized in e.g. PLLs. We have here used the same reasoning that was used in Paper IV to arrive at the static NAND implementation in figure 4.7. A static gate gives us superior robustness and gain and the NAND choice avoids large pMOS transistors.

The NAND based XOR design is not ideal for a standard digital im-

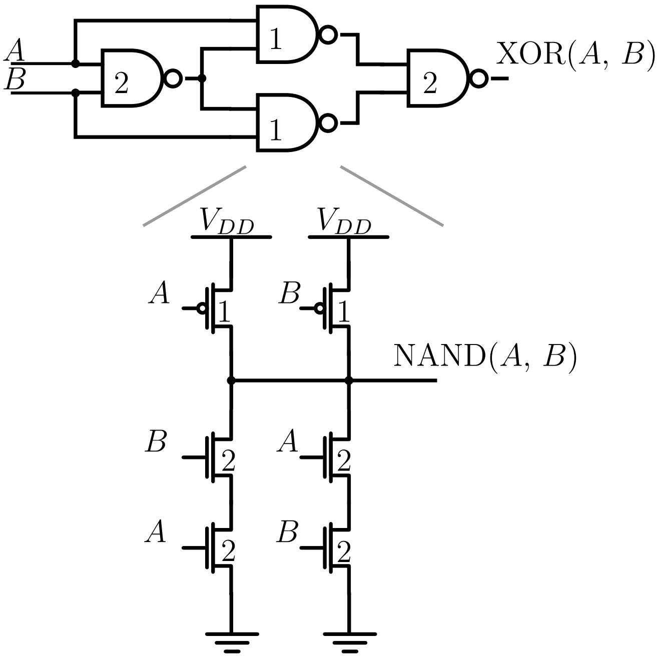


Figure 4.7: 4 NAND XOR implementation, using a symmetric NAND gate. Number in the top denote the relative sizing of each NAND gate, while the number on the NAND gate denote the relative sizing compared to a balanced inverter with $W_N = 4.7\,\mu\text{m}$ and $W_P = 12.75\,\mu\text{m}$.

plementation, where one primarily is concerned with the delay through the gate. For our use, as a continuous time mixer, the delay is not an issue, as it simply shifts the beat in time slightly. By delaying both inputs by the same amount (by using symmetric NAND gates in a symmetric XOR) we ensure the mixer does not introduce any false increase or decrease in the extracted two-way-travel time.

Simulation of the XOR gate as a mixer for FMCW applications can be found in figure B.4 in Paper II. The mixer does exhibit even-order harmonics (2nd, 4th, . . .) and a significant noise skirt; it is not known if this can be avoided by a more careful XOR implementation or if this is a modeling/simulation error. It should also be emphasized that the noise skirt does not originate from the frequency sources, as these are modeled as ideal.

4.1.3.3 Counter and readout

As mentioned in Paper III, for a CW radar (or SFCW), we want to extract the mean of the mixer output. This is simple to implement with a low-pass filter/integrator, which in the digital domain is just a counter that records how often the signal is high compared to how often the signal is low.

The implementation uses a 7 stage ring-oscillator, implemented with 6 inverters and 1 NAND gate to be able to turn the oscillator off. The oscillator is used to sample the signal and 2 counters one that increments if the sampled signal is high and another that increment if the sampled signal is low. 7 stages where selected due to the (slow) speed of the flip-flops used.

A readout is signaled by clocking the counter signals to a SPI readable register and resetting the counter.

4.2 Design flow

One of the great benefits of conventional digital design is the layers of abstractions that can be put between the digital designer and the physical world. Today's synthesis tools can take high level descriptions of the desired digital function and quickly compare and implement for either Field-Programmable Gate Array (FPGA) or CMOS hardware. Such tools are lacking when we design without clocks and this lack of support from the major EDA vendors is a leading reason few companies leverages the advantages gained by a non-clocked design [MAZ+14]. We start by outlining the design flow that was used in this work and make a case for improved support in standard digital design tools.

Except for the quantizer, which has analog inputs, all of the circuits used in this thesis can be completely described in a digital Hardware Description Language (HDL) (either pure VHDL or Verilog). VHDL is actually used in this work, as an initial verification of the design idea, to confirm delays, signal polarity and inter-block level communication. Existing tools to take this VHDL to synthesis and place and route, is to the authors knowledge not available. Existing tools will, as said, assume the circuit is clocked and 1) ignore the VHDL statements

```
A <= 0; B <= 1;
wait for 10 ns;
A <= B after 10 ns;
```

by simply tying both A and B to 1 as that is the quickest solution and 2) simplify any “redundant” logic, such as one-shot circuits above.

The synthesis tool does this, because it assumes one is expressing some combinational logic that will be clocked, hence the only important values are the signal level at the active edge of the clock. There is however nothing wrong with removing this assumption and having a synthesis tool that respect the **wait for** and **after** statements with additional support for glitches and other “hazards”.

All synthesis tools are able to recognize patterns from the description language, such as flip-flops and (often unintended) latches. Flip-flops are asynchronous components, so the idea of recognizing delays and one-shots is feasible. Modern synthesis and place and route tools must increasingly deal with complex (asynchronous) issues such as clock gating and dynamic hazards (glitching) to save power [Rab09] and so support for asynchronous gate based descriptions would in the authors opinion make sense.

4.2.1 Analog design

The design flow in this thesis is therefore very similar to a standard analog design, starting with a schematic of the transistor implementation and initial verification, the design is transferred to layout. For the critical blocks (and in particular for the circuit block in Paper IV), the final transistor size tuning is done on the extracted layout (post layout), where the parasitics associated with the transistors and interconnect gets included. Transistor sizes in this extended netlist can then be fine tuned for optimal performance.

These parasitics are especially important to include for delay based circuits. A schematic only simulation on the wavefront train serializer fails to operate correctly for the waveform 00110011, due (in part) to the short delay a schematic only simulation gives. While a post layout simulation has a 100 % functional yield for 100 Monte Carlo simulation over a range of temperatures and voltages at a significantly lower speed than predicted by a schematic only simulation.

4.2.2 Testbenches

Testing is paramount with any design, especially for a large and partly analog intensive design such as this. Testing can be divided into two parts, a purely functional verification and a detailed characterization to optimize the design and determine power consumption and delay.

For the digital blocks around the memory and the counters, a purely functional test was done with a Verilog based testbench. Stimulus was written in Verilog and the circuits output was verified by bringing it back into the Verilog block. This allows for simple verification over a range of speeds and process corners. To co-simulate both Spectre (for the transistor based circuits) and Verilog, the AMS co-simulator in Cadence Virtuoso is used. This is slower than simply running a Verilog-A block in Spectre, but the Verilog language provides features not present in Verilog-A.

A functional test was also done for the SPI module (with sub-parts connected), where the SPI software was modified to optionally output a PWL file with the 3 SPI input signals. This had the added bonus of verifying (and debugging) the software before entering the lab, though at a steep computational cost.

For a detailed characterization we have already mentioned the importance of extracting the parasitics, in addition, care must be taken to

- model the expected (capacitive) load,
- model a finite supply current and
- model the input stimulus (frequency and driving ability).

To characterize a delay based circuit block, there is very few characteristics of interest, these are power consumption, delay (measured at the 50 % crossing) for both falling and rising edges and also the rise/fall times. The rise/fall times are not as interesting, as equalizing the rise and fall time does not result in a working circuit. The rise/fall time does give an indication of a circuit that is too heavily loaded and the maximum frequency, but so does the delay. In addition, for a first order model, the measurement point for rise/fall time does not matter (e.g. 80 %-20 % or 90 %-10 %), but in a real circuit this choice matters.

The delay based measures can be used to express the input and output Pulse Width (PW) which we define in figure 4.8. We note that one could also extract the negative Pulse Width, but the negative pulse growth/shrinkage measure would carry the same information. It is also trivial to note that minimizing the PW growth is identical to minimizing the difference in rise and fall time as seen below:

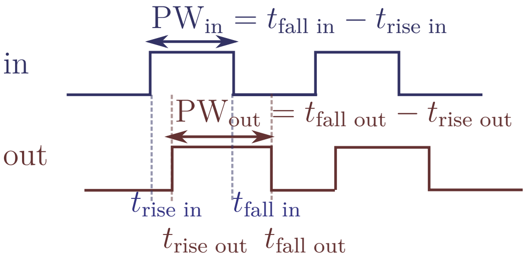

$\mathrm{PW_{in}} = t_{\text{fall in}} - t_{\text{rise in}}$
in
$\mathrm{PW_{out}} = t_{\text{fall out}} - t_{\text{rise out}}$
out
$t_{\text{rise in}}$
$t_{\text{fall in}}$
$t_{\text{rise out}}$
$t_{\text{fall out}}$

Figure 4.8: How to extract the (positive) Pulse Width (PW), to find the PW Growth=$\mathrm{PW_{in}} - \mathrm{PW_{out}}$

$$\begin{aligned}\text{PW growth} &= \mathrm{PW_{in}} - \mathrm{PW_{out}} && (4.1)\\ &= (t_{\text{fall in}} - t_{\text{rise in}}) - (t_{\text{fall out}} - t_{\text{rise out}}) && (4.2)\\ &= (t_{\text{fall in}} - t_{\text{fall out}}) + (t_{\text{rise out}} - t_{\text{rise in}}) && (4.3)\end{aligned}$$

To show why the frequency of the stimulus matters, figure 4.9 shows a series of simulations where the input frequency (1/input period) is varied. For this particular circuit (which was a pseudo-differential mux that was discarded), the optimal nMOS/pMOS ratio depends on the input frequency. When simulating with a sufficiently slow input (below 2 GHz), the optimal nMOS/pMOS ratio is 2.9, but if we do a single simulation at 3 GHz, we might extract an optimal nMOS/pMOS ratio of 4.3. In a time domain simulation

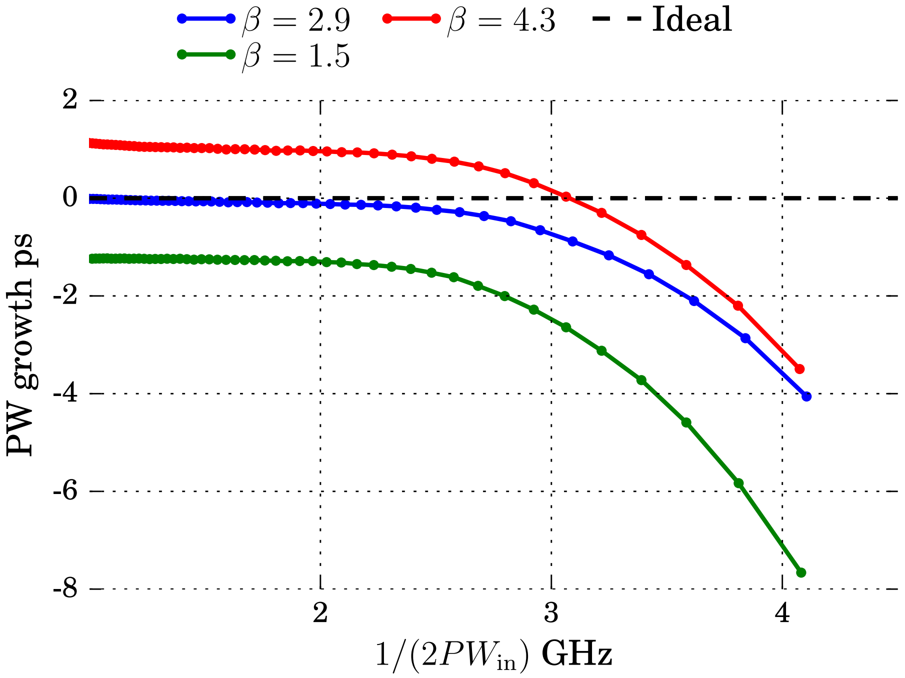


Figure 4.9: Example simulation of the pulse growth as a function of input pulse width and for a few different β = pMOS/nMOS ratios.

at 3 GHz (not shown), we note that the signal is closer to a sinusoidal than a square wave and that the nMOS/pMOS ratio that works for a lower frequency fails to propagate this nearly sinusoidal signal.

In a first order transistor model, the only difference between a nMOS and pMOS transistor is the carrier mobility, which we account for by using a larger pMOS width. Unfortunately, the world is not as simple, and there is a difference in channel-modulation, velocity saturation and threshold voltage between the two transistor types. This complicates an accurate extraction of the optimal nMOS/pMOS ratio and lead to our use of the PW growth measure as this is the “system” measure we care most about. (We do not mind if the rise and fall time differ, or that the DC characteristic is non-symmetric; as long as pulses do not de-form).

4.3 Concluding implementation

We have briefly outlined not only the implementation of the system given in the introduction chapter, but also highlighted some of the design choices, challenges and alternatives while also giving some insight into the design of a continuous time digital system. This has resulted in single-chip CMOS prototype that confirm the feasibility of a digital intensive radar for system-on-chip integration.

In the next chapter, we will look at the remaining circuit block, namely the digitization and transform for a FMCW receiver.

Chapter 5

Baseband

We will in this chapter review some possible baseband implementations, focusing mostly on a square wave FMCW radar, but also touching on a m-sequence and CW/SFCW for on chip-baseband processing. One fundamental aspect we have taken for granted in the papers, is the use of a Fourier transform to get the mixer output into the desired radar range profile. We will here explore exactly how we can achieve this, and also check if we can employ a different transform, namely the Walsh transform. We will also explore a novel filter bank solution, where we use a number of digital band-pass filters tuned to the FMCW beat spectrum range gates.

5.1 Analog filter and ADC baseband path.

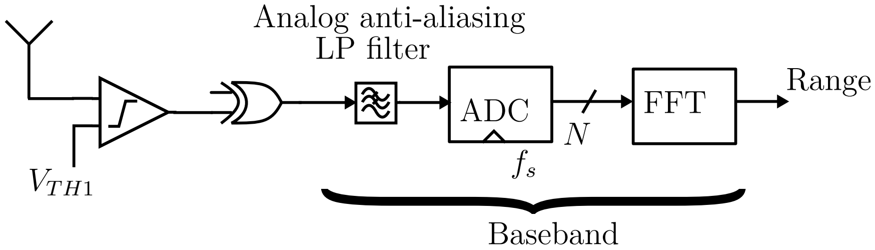


Figure 5.1: Conventional baseband path, the beat is low-pass filtered and quantized to N-bits at a sample rate f_s.

We start with the most straightforward baseband solution, drawn in figure 5.1, where we use a conventional high resolution ADC with an analog filter in front. Measurements presented in this work uses this setup, where the XOR output is taken out of the chip and digitized with a lab-oscilloscope. It is also suited for a Commercial off-the-shelf (COTS) solution, as both ADCs and anti-aliasing filter are readily available for circuit board integration.

One of main advantages of a frequency modulated radar, is the reduction in signal bandwidth achieved by spreading the information in time (using a long pulse). This enables a reasonable sample rate f_s, assuming we image a reasonable range over a sufficiently long time. In more exact terms, we can express the maximum beat frequency $f_{\mathrm{beat}_{\mathrm{un}}}$ as a function of maximum range r_{un} and chirp rate α as

$$f_{\mathrm{beat}_{\mathrm{un}}} = \alpha \cdot 2r_{\mathrm{un}}/c = \frac{BW}{T_m} \cdot 2r_{\mathrm{un}}/c \tag{5.1}$$

The chirp parameters can therefore be adjusted to suit available ADCs, for a fast time-to-market solution.

With an ideal anti-aliasing filter with a cutoff at the maximum wanted beat frequency $f_{\mathrm{beat}_{\mathrm{un}}}$ and an ideal ADC with infinite resolution, we can perfectly capture the wanted information by setting the sample rate f_s to the Nyquist rate of $2f_{\mathrm{beat}_{\mathrm{un}}}$. A realistic anti-aliasing filter with a finite rollover will however benefit from a faster sample rate, giving a straight forward tradeoff between energy suppressed by the filter (determined by the filter order, type and cutoff point) and the ADC clock rate. In addition, the finite ADC resolution, set by the effective number of output bits, gives another tradeoff.

The effective number of output bits of an ADC, can be found by rearranging (2.38)

$$\mathrm{ENOB} = \frac{\mathrm{SNR}_{\mathrm{db}} - 1.67}{6.02}. \tag{5.2}$$

where $\mathrm{SNR}_{\mathrm{db}}$ is the measured ADCs SNR. For an oversampled ADC, we first low pass filter the digital output and decimate, before computing the $\mathrm{SNR}_{\mathrm{db}}$ and ENOB. The ENOB figure of merit will also include various ADC non-idealities and in essence gives us a sense of how many bits that we can discard.

To quantify the feasibility of an on-chip CMOS ADC, we have used the published ADC performance parameters from [Mur16] to create figure 5.2. We have overlaid the required Nyquist frequency (four times (5.1), to get complex samples) for an example design with the following constraints

- Max unambiguous range $r_{\text{un}} = 100\,\text{m}$ in free space.
- Bandwidth 2.1 GHz

As we increase the chirp time, we also increase the processing gain of the FFT operation. To quantify this, we assume an idealized processing gain of $N_{\text{FFT FMCW}}$, which assumes a sinusoidal in white noise, where

$$\begin{aligned} N_{\text{FFT FMCW}} &= F_s \cdot T_m = 4 f_{\text{beat}_{\text{un}}} \cdot T_m && (5.3) \\ &= 4 \left(\frac{BW}{T_m} \cdot 2 r_{\text{un}}/c \right) \cdot T_m && (5.4) \\ &= 8 BW \cdot r_{\text{un}}/c && (5.5) \end{aligned}$$

giving

$$G_{\text{FFT FMCW db}} = 10 \log_{10} \left(8 BW \cdot r_{\text{un}}/c \right) \qquad (5.6)$$

Which is $G_{\text{FFT FMCW db}} \approx 37\,\text{dB}$ for the above parameters. It may look un-intuitive to have a processing gain proportional to maximum range and bandwidth, but keep (5.3) in mind. When we increase the maximum range, we also increase the beat frequency and hence the sample rate, capturing more information about the signal in a given time-span, effectively identical to oversampling. The same reasoning applies to increasing the bandwidth, where an increased bandwidth requires a faster sample rate. Increasing the chirp time, does not increase the number of FFT points, as long as we also reduce the sample rate by the same amount.

Now, as we have seen in the Background chapter, sampling with a high resolution ADC in a low SNR environment is wasteful, additionally a single bit radar is best suited for low SNR scenarios (close or below the thermal limit) and so we may intentionally be transmitting at low energy levels. In addition, as seen in chapter 3, the expected SNR for a discrete time FMCW radar is relatively low per sweep, using the simulated value from chapter 3 we obtain an expected SNR of 43 dB for a 17 µs long sweep. We thus see that most of the SNR performance (with the presented discrete time FMCW radar), originates from the coherent averaging and the FFT processing gain.

Now what does this imply about our required ADC Signal to Noise and Distortion Ratio (SNDR) performance? As the instantaneous SNR is

$G_{\text{FFT FMCW db}}$ lower than the per sweep SNR, the ADC SNDR only needs to be better than

$$SNDR_{\text{FMCW ADC}} > SNR_{\text{FMCW per sweep}} - G_{\text{FFT FMCW db}}$$

which is only $SNDR_{\text{FMCW ADC}} > 5.6\,\text{dB}$ in this numerical example. This requirement is included in figure 5.2 as a red line, all ADCs above this line should have sufficient SNDR for our required instantaneous SNR. We note that this requirement should be near trivial to achieve. The reader should however keep the assumptions in mind, as we are assuming white non-coherent noise, ignoring coherent spurs and so some margin above the found values is advisable.

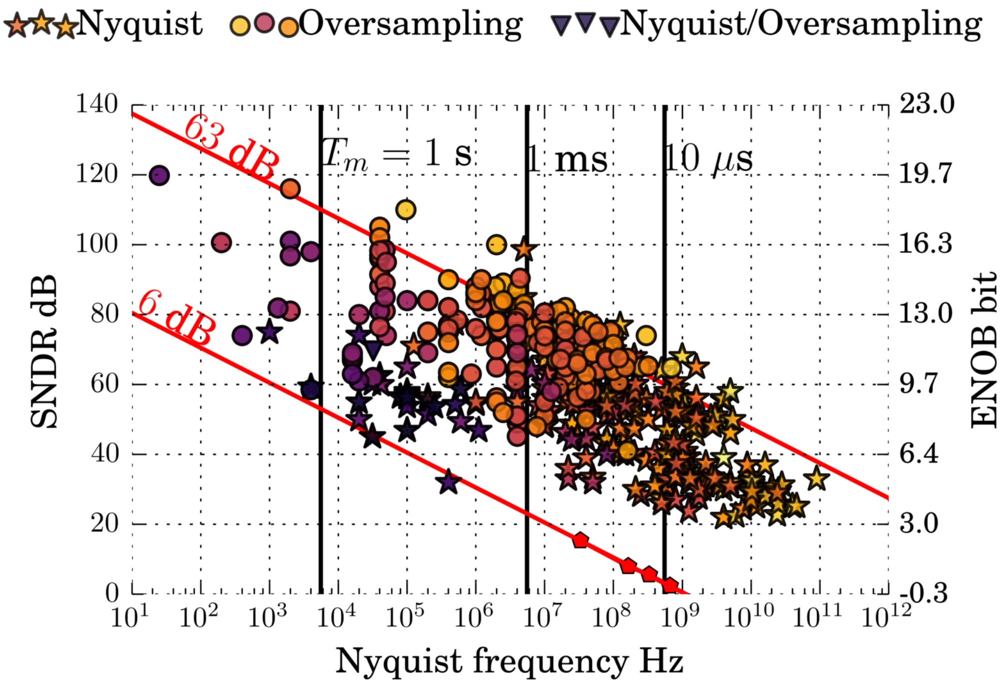


Figure 5.2: Symbols are ADC performance values from [Mur16] showing pure CMOS implementations published between 1997 and 2016 in ISSCC and VLSI Symposium. Color indicate power consumption (darker uses less power). Vertical lines are drawn to indicate the required beat spectrum Nyquist frequency for a FMCW radar with fixed bandwidth of 2.07 GHz and a maximum unambiguous range of 100 m (in air) for different sweep times using four times eq. (5.1). Red lines denote the lower bound for ADC SNDR for a wanted single sweep SNR of 43 dB and 100 dB after the range-FFT.

We also include the required SNDR if the single sweep SNR is 100 dB, (the 63 dB line in figure 5.2), which gives a much stricter ADC requirement. This high level of instantaneous SNR, would require a continuous time transmitter as shown in section 3.1 and most likely an analog receiver and mixer as was depicted in figure 1.3 and with a transmitted energy at least 100 dB above the noise floor.

Despite CMOS not being an ideal platform for ADC integration [Jon10, Sha00], we see that a discrete time FMCW system with on chip ADC; is feasible, due to the relaxed requirements that we obtain when we consider both swept threshold averaging and DFT processing gain. These two coherent averaging techniques helps us have a reasonable system SNR, without each component in the system being highly accurate.

One drawback is that the presented digital FMCW radar is fully configurable for a wide range of sweeps with different chirp rates and delays, yielding a wide array of possible beat bandwidths and SNDR requirements, a (possibly external) analog filter and ADC will lack the flexibility to handle this flexibility in an efficient manner. The remainder of this chapter will therefore focus on alternative baseband architectures, where we will explore several novel solutions that do not need the classical anti-aliasing and ADC solution presented above.

5.1.1 Double swept threshold baseband

A natural single bit realization of figure 5.1 is drawn in figure 5.3(a), where in-place of the multi-bit ADC we employ another swept threshold comparator.

To begin, we must first understand what the mixer output signal looks like. Despite being a single bit signal, sampling is non-trivial, since the mixer output contains a range of frequencies that we are not interested in. Without filtering, these will alias down into the sampled signal. Among these unwanted terms is the mixer sum, which for the analog case goes from approximately from $2f_l$ to $2BW$ (see the f_{11}^{+} term in (C.7) on page 182), while a digital radar also has additional unwanted terms above this range. We will explore these briefly by also considering the simple arrangement in figure 5.3(b), where the anti-aliasing filter is removed.

After low-pass filtering, if the harmonics are placed outside the range-view by a e.g. delay as shown in figure 1.6, the remaining signal will be a superposition of sine waves. If no delay is used, the harmonic amplitudes are

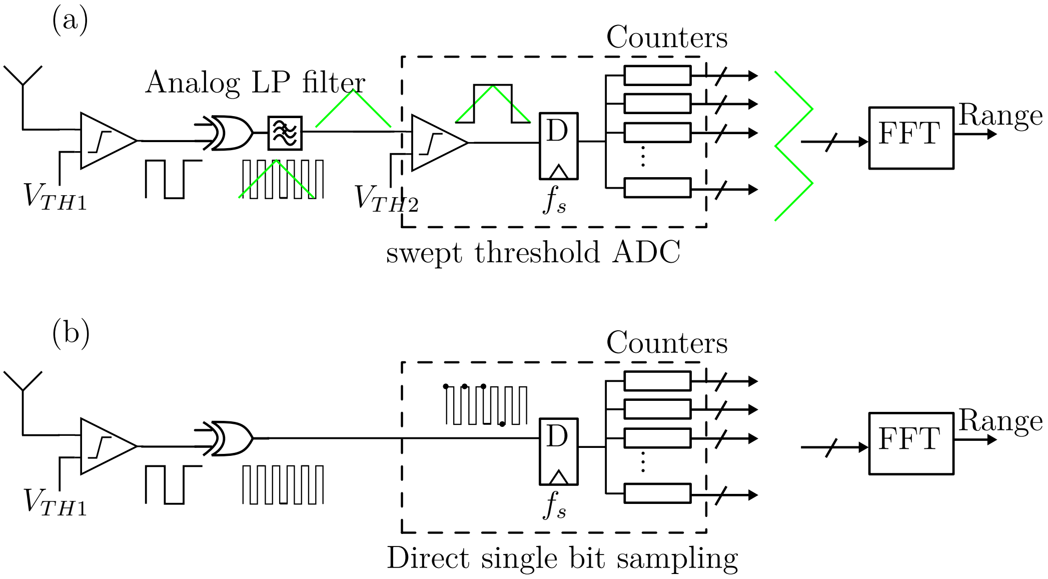


Figure 5.3: (a) Alternative implementation of figure 5.1, where the ADC is a sweep-threshold quantizer; consisting of a second comparator, a sampler and a digital integrator (drawn as a bank of counters). (b) Simplified implementation, ignoring the anti-aliasing filter, leaving a single-bit signal for sampling.

the product of the harmonic amplitude on the mixer inputs. For square waves with approximately −10 dBc third harmonics, the output will have −20 dBc third harmonics, which in the time domain takes the form of a triangular wave, which is what we have drawn in green. Multiple scatters will then be a superposition of either triangular or sine-waves. Thus, we see that we go from a square wave digital signal; to an analog signal that needs to be amplitude quantized.

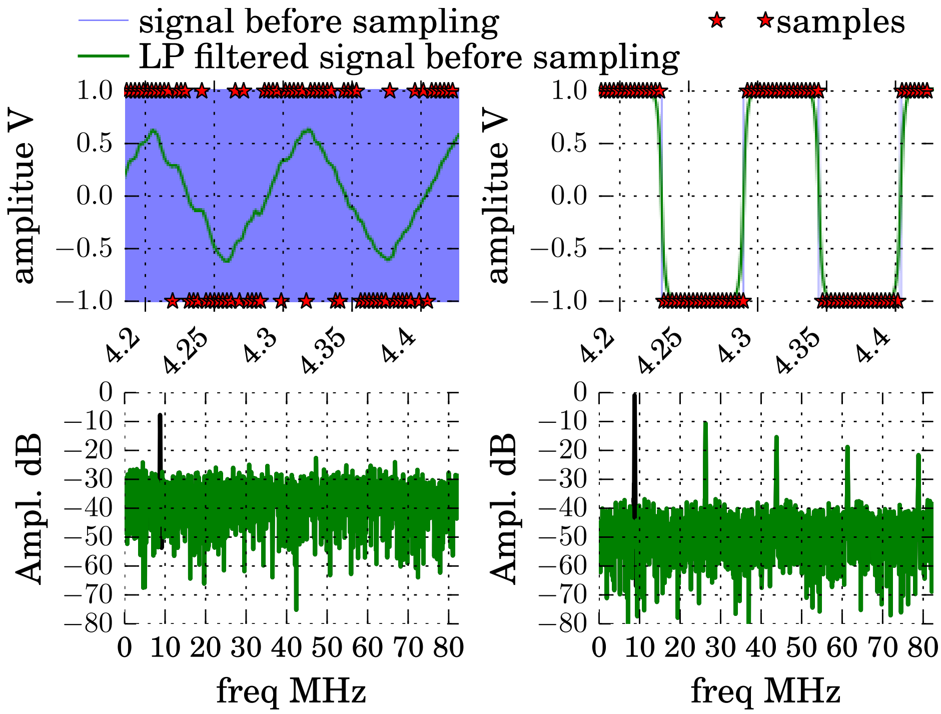
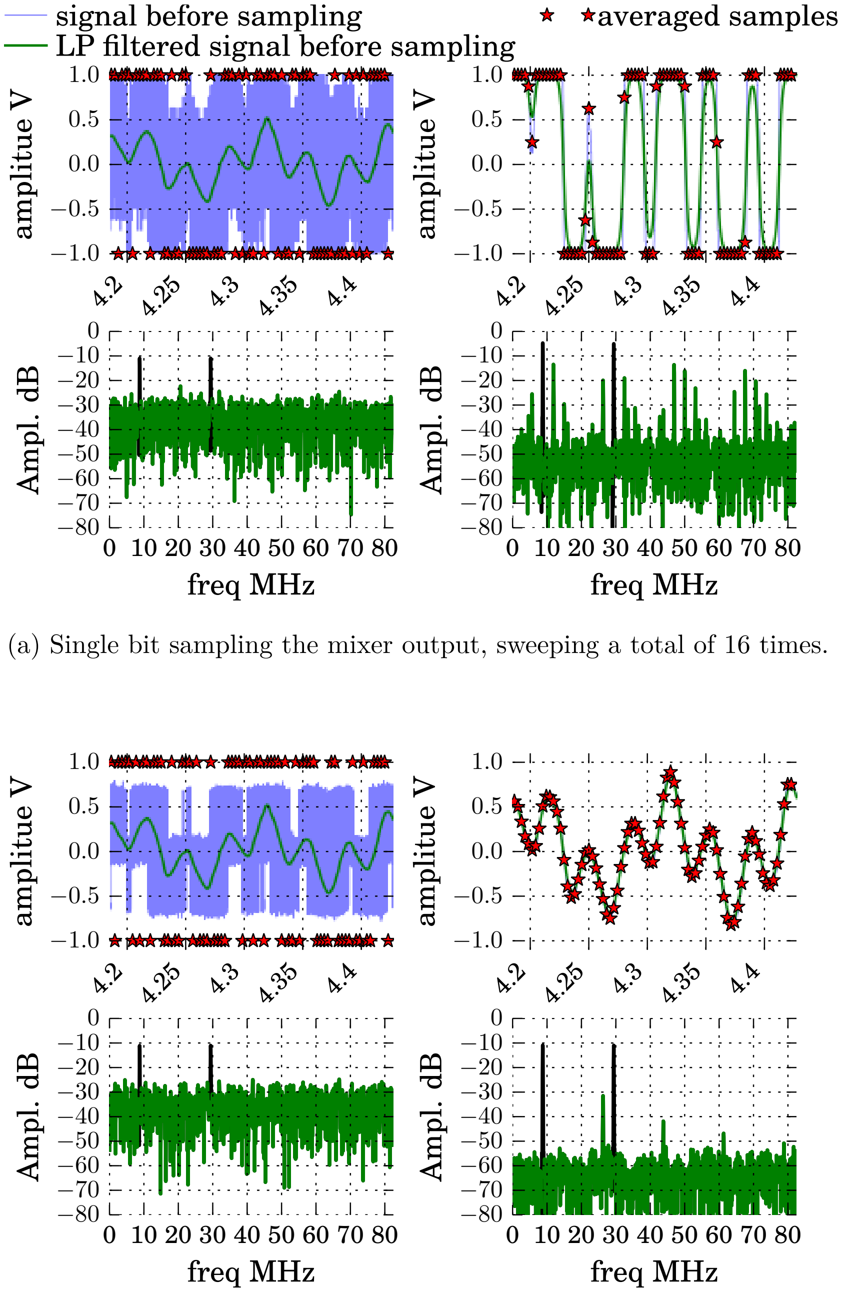


Figure 5.4: Simulating a single scatterer at 8.76 MHz with a single threshold. Left: directly sampling the XOR output bitstream at $4f_{\mathrm{beat}_{\mathrm{un}}}$, right: using an idealized anti-alliasing filter first, before single-bit sampling at $4f_{\mathrm{beat}_{\mathrm{un}}}$.

To visualize this sampling process, an example simulation is shown in figure 5.4, where we show a single sweep that is single bit sampled both with and without anti-aliasing filtering. As only a single scatterer is simulated here, the only peaks are the scatterers fundamental and harmonics, for the anti-aliased case, notice that the harmonics are at −10 dBc, since we have effectively thresholded the input triangular waveform into a square wave. This simulation is however very optimistic and does not show that the system is indeed not input output linear.



(a) Single bit sampling the mixer output, sweeping a total of 16 times.

(b) Using 16 in the second swept thresholder for a total of 256 sweeps.

Figure 5.5: Simulating two scatterers with 16 thresholds in the first quantizer. Left: directly sampling the XOR output bitstream at $4f_{\text{beat}_{\text{un}}}$, right: using an idealized anti-alliasing filter first, before swept-threshold sampling at $4f_{\text{beat}_{\text{un}}}$.

We therefore include multiple targets in figure 5.5 and also sweep multiple times in an attempt to linearize the system. In figure 5.5(a) we notice that with anti-aliasing, the output now contains multiple inter-mixing products of the two target beat frequencies due to the single bit sampling of our now analog signal. In figure 5.5(b) we therefore use multiple comparators (serialized) on the second thresholder, leaving us with a linear system that only has the expected fundamental and harmonic peaks.

We note that not including an anti-aliasing filter is disastrous for our SNR level, as we get excessive aliasing when sampling in time. That said, for applications that can live with the poor target dynamic range and for few scatterers (such as an altimeter or level sensors, where we are not necessarily trying to detect a weaker scatterer close to a strong one), the roughly 10 dB SNR could be acceptable, providing a very cheap and easy to implement radar system that could even forgo the anti-aliasing filter.

As we saw in the previous section, the instantaneous SNR requirement is not necessarily excessive, and so we would expect that the second thresholder in figure 5.3 does not need too many levels. This is seen in figure 5.6, where we adjust the number of threshold levels (and hence the number of sweeps) and calculate both SNR and Spurious Free Dynamic Range (SFDR). We note that we reach the “ideal” no sampling case after just a few comparators. This ideal level is constant, as we have not included any randomization in the transmitter and as such the remaining “noise” is coherent, but we expect the phase noise method of section 3.2.1 to work for this case as well. We note that at $N_{\text{sweep2}} = 1$ (single bit sampling the beat), not having an anti-aliasing filter is actually slightly beneficial for the SFDR value, as the folded noise helps to smear out the otherwise non-linear peaks.

5.1.1.1 Summary

For a single target, like a radar altimeter (or the short range “liquid level” equivalent application), the solution in figure 5.3 will most likely work acceptably, as a general purpose radar, the solution does have some major drawbacks.

Firstly, for each receiver threshold V_{TH1}, we need to sweep V_{TH2}, requiring a very long integration time. This can be counteracted by using multiple comparators in parallel, effectively creating a N-bit flash ADC. In the altimeter example, only a single threshold is required as the zero crossings are

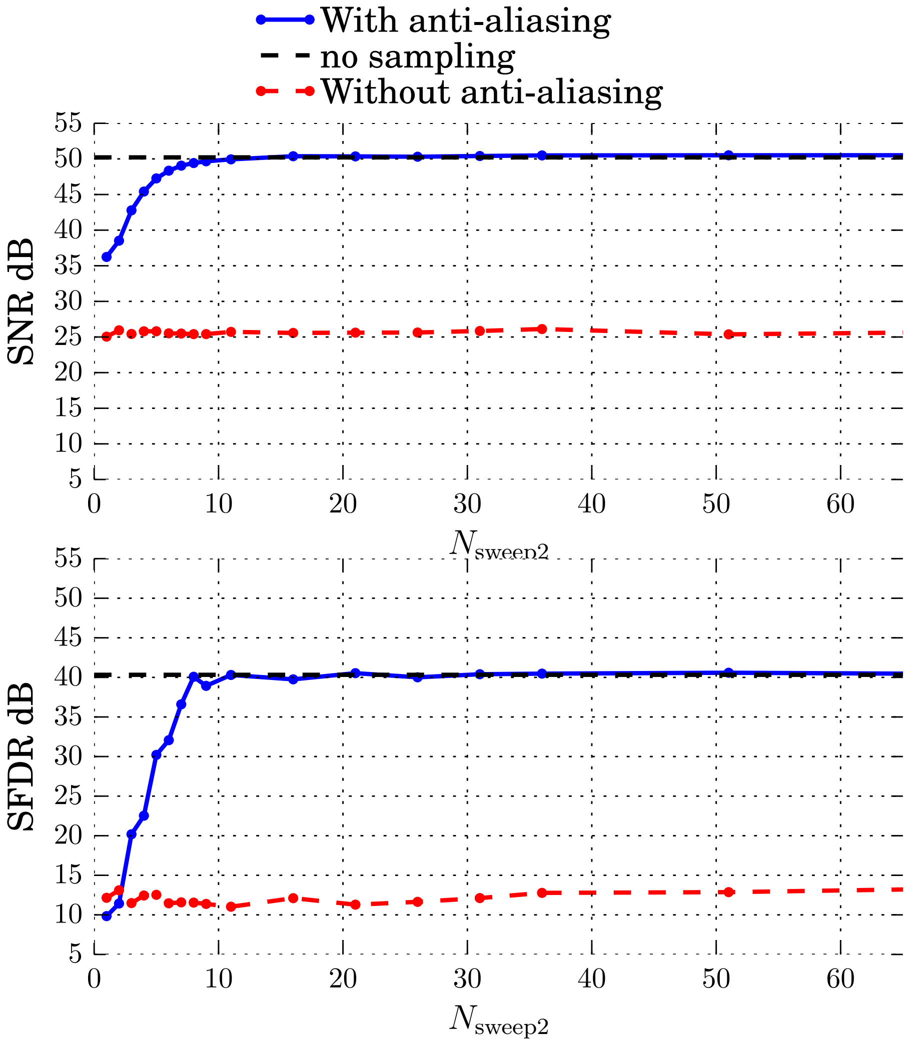


Figure 5.6: Simulating 2 targets of equal amplitude, showing both estimated SNR and SFDR (Spurious Free Dynamic Range) as a function of the number of thresholds in the second swept-threshold ADC in figure 5.3. A channel thermal noise SNR of 0 dB is used and the sweep is repeated $16 \cdot N_{\text{sweep2}}$ times from 600 MHz to 2.67 GHz in 16.7 µs time quantized to 64 ps with a delay in the channel path of 667 ns to remove harmonics.

sufficient (as discussed in section 2.2), showing that the idea is sound for a single return.

The second drawback, is that the sweep-threshold technique gives an improvement in SNR if the sweeps are un-correlated, this will be the case for the first thresholder if the input SNR is low; but may not be fully un-correlated on the second thresholder since the bandwidth is effectively much lower after the anti-aliasing filter. One workaround for this is to add intentional noise for the second thresholder, either on the input or the threshold input.

Lastly, it should be mentioned that the arrangement with an analog LP filter and a thresholder, is very similar to how we would expect a digital gate to behave. A digital gate will have a finite bandwidth and the signal will settle to either a "0" or a "1" resulting in the thresholding action. As we have seen, this is both good and bad news, the filtering action will reduce the bandwidth and aliasing problems, while the thresholding action will result in a non-linear behavior. Luckily the non-linear behavior will be reduced as we average multiple sweeps in a noisy environment as we have previously discussed.

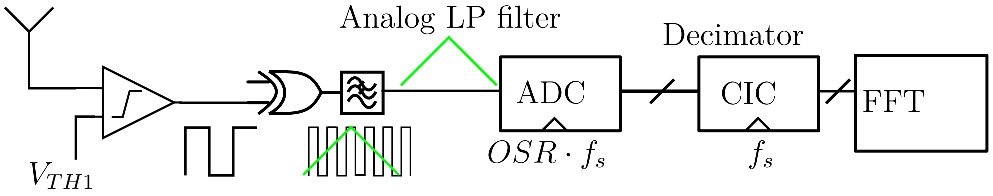
5.2 Cascaded Integrator-Comb filters (CIC)

"lean, mean filtering machines" – [Lyo]

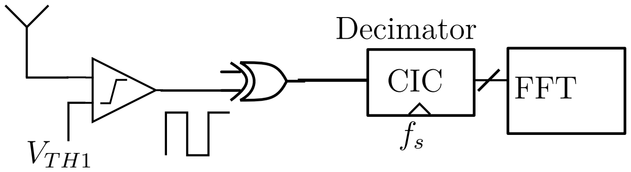
We have seen in the previous section that simply sampling the XOR output beat bitstream at some reduced sample rate, causes aliasing and possibly non-linearity. As an alternative method of circumventing this, we will in this section investigate single bit oversampling with a digital filter, where the goal is a circuit block that can output a multi-bit filtered signal at some reduced sample frequency, without an analog filter.

A conventional Cascaded Integrator-Comb (CIC) decimation stage is shown in figure 5.7(a), where the decimator reduces the sample rate (but typically increases the bit-width) of an over-sampled digital signal. The main advantage of a CIC decimator over other decimation methods is the simple structure and implementation. What we would like to propose is a method that circumvents the analog conversion altogether, which is shown in figure 5.7(b), where the single bit continuous time bitstream is (over-)sampled and decimated by a CIC like structure.

The conventional CIC structure is shown in figure 5.8(a) and consists of an integrator, sampler and differentiate stages. With a sample rate S/R, and



(a) Conventional CIC usage after a (oversampling) ADC



(b) Proposed single bit input CIC used right after the XOR gate without filtering

Figure 5.7: Showing both the conventional (and ideal) way of utilizing a decimator after an oversampling ADC and in (b) the proposed method of simply single bit sampling the un-filtered beat spectrum.

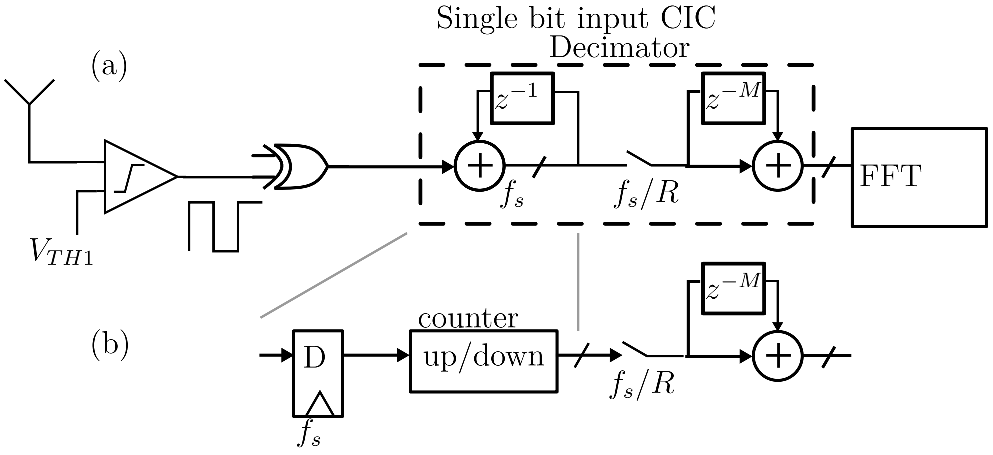


Figure 5.8: Showing the (a) CIC structure and (b) a simple single bit integrator stage.

a frequency f, both in Hz, a CIC decimation filter that decimates by R will have a voltage transfer function

$$G(f) = \left| \frac{\sin\left(\frac{\pi M f R}{S}\right)}{\sin\left(\frac{\pi f}{S}\right)} \right|^N \tag{5.7}$$

where M is the "differential delay" used in the comb section of figure 5.8 and N is the number of stages [Hog81]. One can compensate for the sinc shaped transfer function in a later stage by applying an inverse of (5.7).

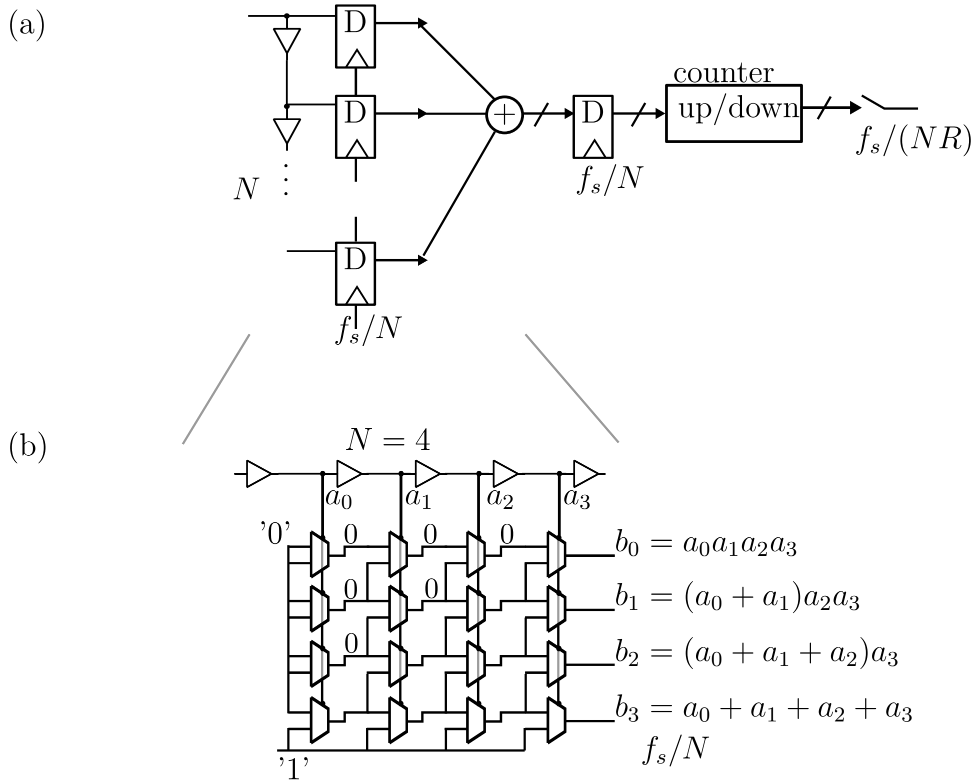


Figure 5.9: Conceptual methods of implementing a high speed single bit integrator, (a) a parallel implementation of figure 5.7(b) and (b) a non-clocked implementation from [SDV$^+$10, SVH$^+$12]

Now as the input signal is not band-limited, we want to push the sample rate f_s as high as possible. The structure in figure 5.8(b) will mainly be limited by the clock rate of the initial flip-flop. As a method of reducing this limitation we can employ multiple sampling flip-flops in parallel as shown in figure 5.9(a). To take this even further, we can employ a "continuous time" integrator shown in figure 5.9(b), where we in place of clocked synchronization rely on equal gate delays.

In the previous section, we explored single bit Nyquist sampling. To show the benefit of oversampling on the receiver (and then CIC decimating), we create figure 5.10. The sample rate in Hz will depend on the chirp rate and so a lower chirp rate will make the sampling requirement more relaxed as seen in the top panel where we compare our 2 GHz sweep with a 200 MHz sweep. So that we only modify the baseband sample rate we also scale the max range and chip time by the same factor of 10. We note that for sample rates around 10 GHz, which is feasible with the circuits discussed above, does give some reduction in SNR, but this tradeoff may be worthwhile considering we only need single bit sampling without analog filters.

5.3 Fourier and Walsh transform of square wave signals

Fourier analysis is the go-to transform in electrical engineering and signal processing but is inherently unsuited for the analysis of square waves. The reason for this is that an infinitely sharp square wave needs an infinite number of Fourier coefficients, making it cumbersome in both closed form and numerically. Luckily, a real world square waves has a finite slope and hence a finite number of Fourier coefficients, making Fourier analysis feasible, if not exactly suited. We will here define and discuss the benefits and drawbacks of Fourier analysis, while also exploring an alternative transform, namely the Walsh transform[1].

The Walsh transform is enticing, as it only deals with integers of ±1, whose multiplication is trivial as explained in section 5.3.1, making for a very attractive on-chip frequency transform. Unfortunately, as we will see, expressing the beat spectrum as a sum of square waves is not necessarily beneficial and the postulated benefit turns out to be smaller than expected. The reader may therefore skip this section without much loss in continuity.

The discrete Fourier transform is used to express a given signal $x[n]$ as a

[1] See Wikipedia for a long list of alternative names for the same transform `https://en.wikipedia.org/wiki/Hadamard_transform`.

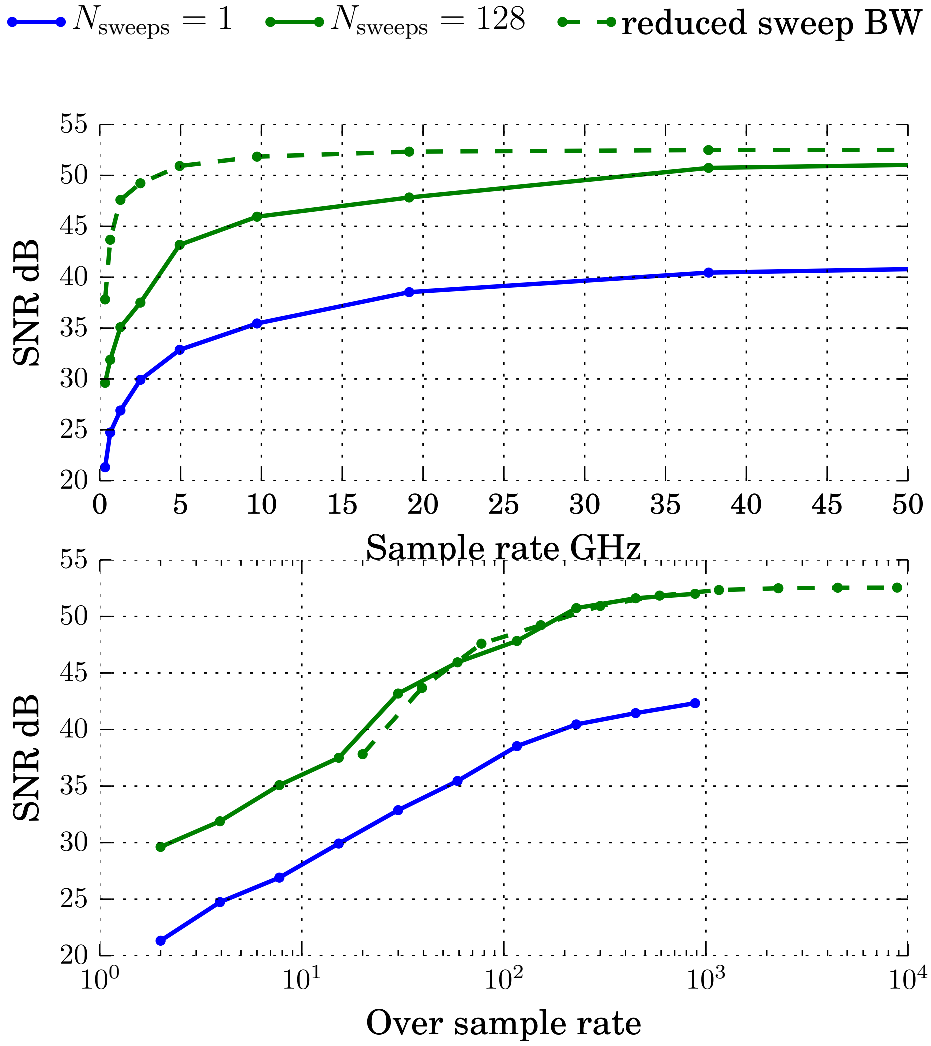


Figure 5.10: Direct single bit sampling without any filtering, shown both as a zoomed absolute linear frequency scale and at the bottom: normalized to the Nyquist rate 165 MHz for the $BW/T_m = 2.07\,\text{GHz}/16.7\,\mu\text{s}$ chirp and 16.5 MHz for the reduced bandwidth chirp $BW/T_m = 207\,\text{MHz}/167\,\mu\text{s}$ with a scaled chirp time to maintain the same gain-bandwidth product and a scaled maximum range to maintain the number of FFT points.

linear combination of complex exponentials

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X_f[k] \cdot e^{jk\omega_k \Delta t}, \tag{5.8}$$

while the (discrete) Walsh transform can express $x[n]$ as a linear combination of Walsh rows

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X_w[k] \cdot W[n,k] \tag{5.9}$$

The Walsh matrix $W[n,k]$ can be generated in a number of ways, see [HS79, appendix A]. As an example, the Walsh matrix of size $2^3 \times 2^3$ in Sequency order is

$$W = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & - & - & - & - \\ 1 & 1 & - & - & - & - & 1 & 1 \\ 1 & 1 & - & - & 1 & 1 & - & - \\ 1 & - & - & 1 & 1 & - & - & 1 \\ 1 & - & - & 1 & - & 1 & 1 & - \\ 1 & - & 1 & - & - & 1 & - & 1 \\ 1 & - & 1 & - & 1 & - & 1 & - \end{pmatrix} \tag{5.10}$$

where "$-$" is used in place of -1. As we can see, the Walsh matrix contains only ± 1 and is also symmetric $W[n,k] = W[k,n]$ and orthogonal $W = NW^{-1}$ (where N is the normalization equal to the number of rows/columns in W); so the Walsh transform is identical in form to the inverse transform in (5.9)

$$X_w[k] = \sum_{n=0}^{N-1} x[n] \cdot W[k,n]. \tag{5.11}$$

While the discrete Fourier transform takes the form

$$X_f[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-jn\omega_k \Delta t}, \tag{5.12}$$

As is known, the Fourier transform can be re-written to express a signal as sin and cos terms; we can do something similar with the Walsh transform which can be divided into the even and odd functions Cal and Sal. In table 5.1, we further explore the Walsh matrix of order 8, where the Cal and Sal naming convention is shown, together with the number of zero crossings and the relative frequency. The relative (Fourier) frequency is found by Fourier transforming each of the Walsh rows, which is shown in figure 5.11. We note that the Sal[i] and Cal[i] have the same Fourier frequency, but a phase shift of 90, so a Sequency ordered Walsh amplitude spectrum can be displayed as the quadratic mean of the two, giving a similar visualization as a single sided Fourier amplitude spectrum as we will explore later.

Table 5.1: Relative peak Fourier frequency for Walsh functions of order 8 and number of zero crossings.

Walsh	even, odd	relative frequency	nr. of zero crossings
$W[0, n]$	Cal[0, n]	0	0
$W[1, n]$	Sal[1, n]	1/8	1
$W[2, n]$	Cal[1, n]	1/8	2
$W[3, n]$	Sal[2, n]	2/8	3
$W[4, n]$	Cal[2, n]	2/8	4
$W[5, n]$	Sal[3, n]	3/8	5
$W[6, n]$	Cal[3, n]	3/8	6
$W[7, n]$	Sal[4, n]	4/8	7

Figure 5.11 shows the Walsh rows in time and in both Walsh and Fourier domain, we also introduce the continuous Walsh functions which are obtained by a zero-order-hold interpolation of the discrete Walsh rows. We specified that the matrix in (5.10) was Sequency ordered, this is apparent in both figure 5.11 and table 5.1 as an increase in number of zero crossings and an increase in peak frequency; as we travel down the matrix. Different orderings are possible but will not be discussed here.

Some interesting observations can be made by looking at the Fourier view in figure 5.11, where it is clear that the Fourier transform is ill-suited in expressing a square wave. We firstly note the number of odd harmonics for row 1,2,3,4 and 7 which is visible when we use the oversampled continuous time view, showing that a Fourier transform needs a large number of coefficients to describe a square wave. In addition, we note a discrepancy in amplitude between the Fourier transform of the discrete (stars) and continuous (lines) Walsh rows, due to the undersampled nature of the Fourier view.

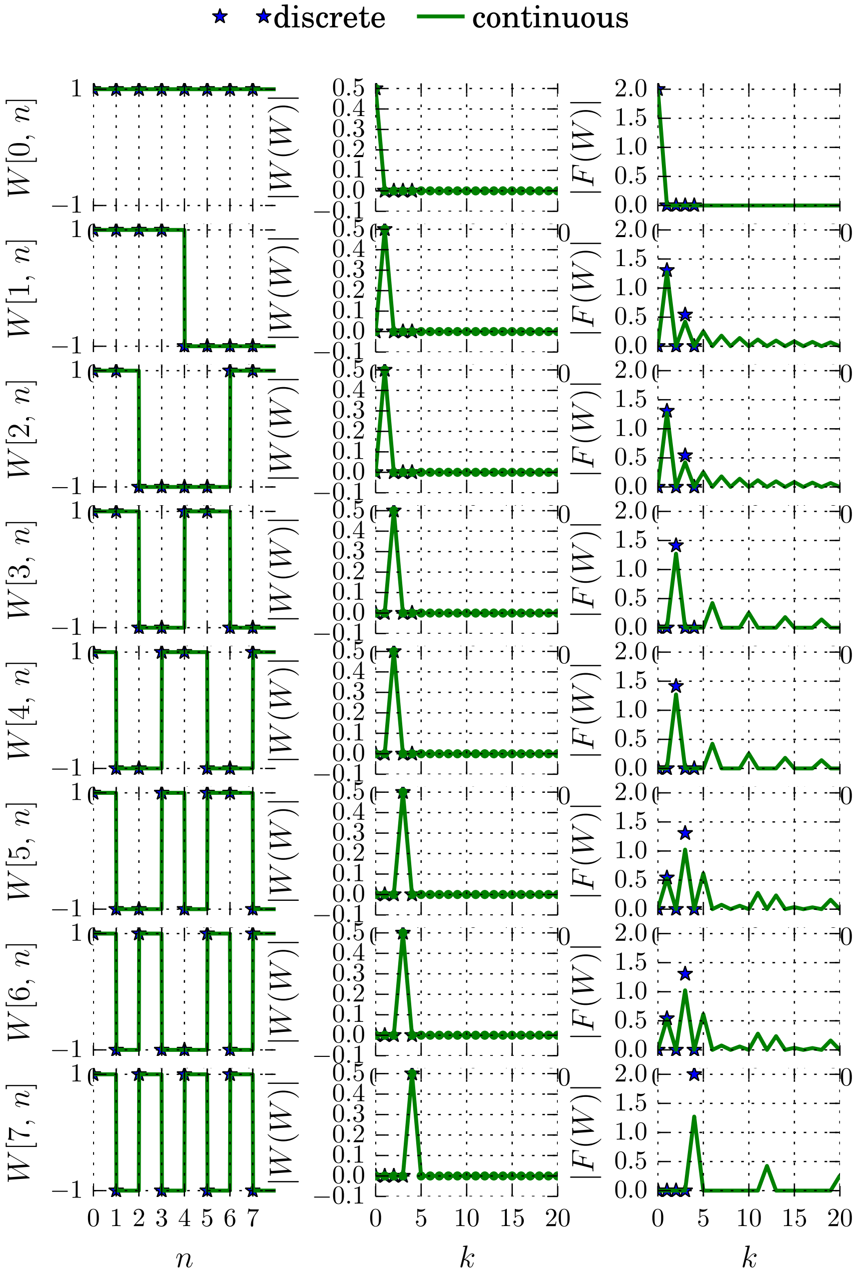


Figure 5.11: Visualizing the Walsh matrix in (5.10) and the Walsh function. On the left, the time domain view, middle: the positive Walsh spectrum and on the right a positive Fourier spectrum.

5.3.1 Implementation, XOR as a multiplication circuit.

It should be apparent that the Walsh transform is a lot more amendable to simple hardware realization, where, instead of multiplication with $e^{-jk\omega\Delta t}$, which is 2 floating point values, we only have the set $\{-1, 1\}$.

If $x[n]$ is a single bit, and interpreting $\{0, 1\}$ as $\{-1, 1\}$ the multiplication can be carried out by an XOR gate. The interpretation is that two number of opposite signs; results in a negative result, while equal signs; become positive. This is shown in table 5.2.

Table 5.2: Truth table of a digital XOR gate and multiplication of ±1.

a	b	XOR(a, b)
0	0	0
0	1	1
1	0	1
1	1	0

(a)

a	b	$a \cdot b$
-1	-1	-1
-1	$+1$	$+1$
$+1$	-1	$+1$
$+1$	$+1$	-1

(b)

5.3.2 Utilizing Walsh in a square wave radar

We will in this section explore the feasibility of utilizing the Walsh transform in a square wave FMCW radar. As the Fourier beat spectrum has a number of harmonics, due to the square wave nature of the signals, the hope is that a Walsh beat spectrum will be free from harmonics and only give a single peak. Unfortunately, as seen in figure 5.12, this is not the case. We will below explain the challenges in attempting to utilize the Walsh transform and why it is not perfectly suited for the task.

Firstly, as we saw in the previous two sections, the mixer output is a pulse-density modulated triangular waveform and not a square waveform. So expressing a sum of triangular waveforms as a sum of square Walsh rows is not ideal and does not give us the single sharp peak we were hoping for. Secondly, the Walsh transform is not shift-invariant, and so a small phase

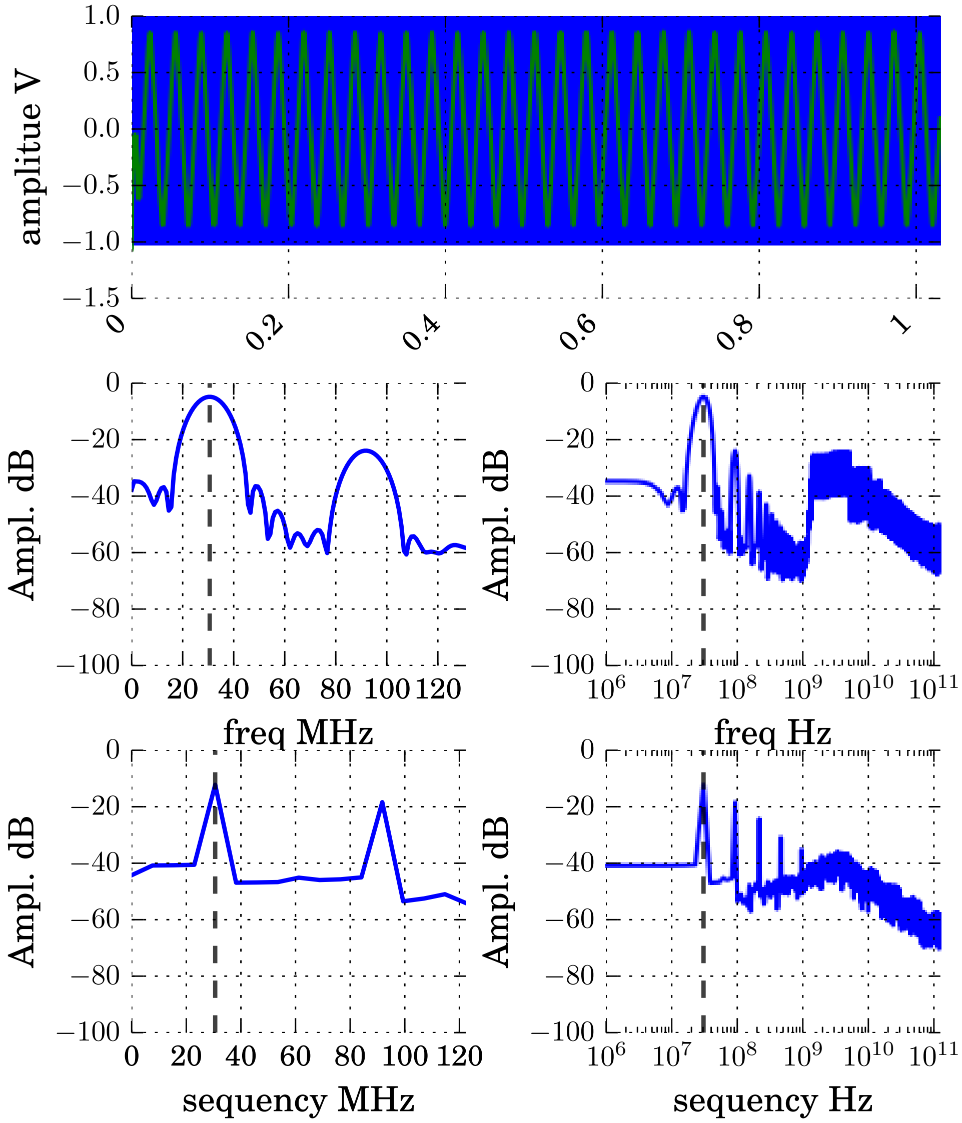


Figure 5.12: Time view (top), Fourier view and Walsh (bottom) amplitude spectrum of an example digital beat spectrum where the target is placed exactly a power-of-two integer ratio from the simulated sample spacing. The beat spectrum is shown on the left while the entire spectrum in log scale is shown on the right.

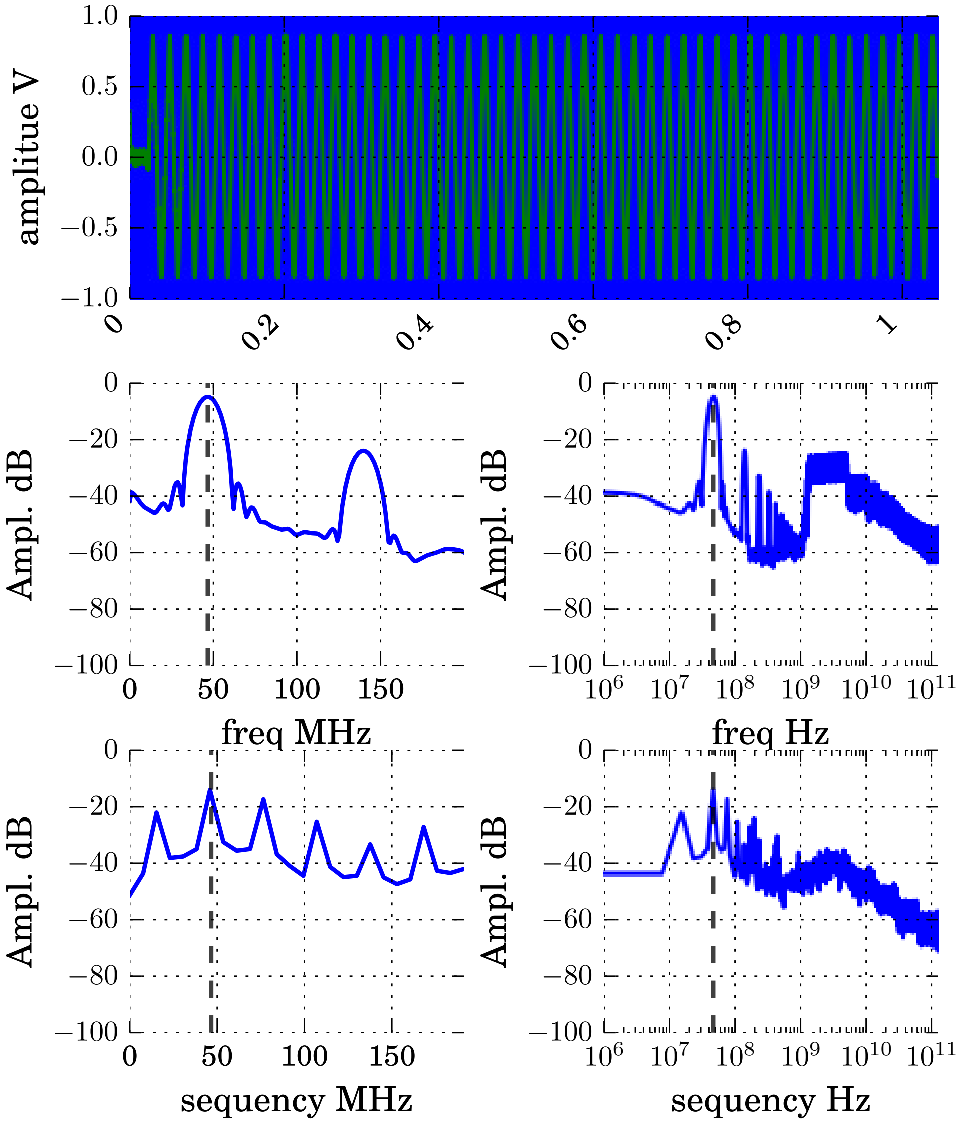


Figure 5.13: Time view (top), Fourier view and Walsh (bottom) amplitude spectrum of an example digital beat spectrum with a non-integer target delay and without removing the initial discontinuity. The beat spectrum is shown on the left while the entire spectrum in log scale is shown on the right.

shift away from any of the Walsh rows give a significantly more crowded Walsh spectrum. This shift dependence can be seen by comparing figure 5.12 and figure 5.13, where we in the first case is very careful to simulate a target located at the correct range bin, while we in the second case show a more realistic randomly selected target delay.

A workaround for the first challenge is to employ a single bit sampling, either by oversampling or by filtering and clipping as was shown in figure 5.3 and figure 5.1 respectively. The signal is now truly a square wave, but this still leaves the problem of shift-variance.

As seen in the above discussed graphs, the Walsh spectrum is not (Walsh)-band-limited (even if filtered and quantized), so any reconstruction using only the lowest Walsh bins will lead to potentially significant loss of information. This necessitates a rather high sample rate and number of points and so even though the transform is simple, can become problematic to implement, which is confirmed by [Bla74].

A clear tradeoff between the computational advantage of a Walsh transform, the physical implementation and the system performance is beyond this thesis. The Walsh view could potentially have benefits for specific applications, for instance an altimeter (since the peak Walsh index seems to corresponds to the largest scatterer) or in a setting where you looking for one (or more) known signatures, similar to [HHM82].

Similarly to other 1 bit concepts, the Walsh transform was popular in the 1970s, an audio synthesizer was proposed by Hutchins [HJ73] where the Walsh functions are weighted and summed for arbitrary audio output. A rigorous master thesis can be found in [Del75], covering the differences between a standard FFT and a Walsh transform, both software (Fortran) and hardware implementations (mostly XOR gates in TTL integrated logic) and the different Walsh function orderings. The master-thesis does confirm what we found here, that a Walsh transform is dependent on the time base and only integer divisions of the time base makes sense. Put simply, “the function $\cos(1.386, \phi)$ is easily defined” [Del75, page 17], while $\mathrm{Cal}[1.386, \phi]$ is not. As far as we can tell, this excludes using the Walsh transform as a replacement for a FFT in a radar receiver, as we have attempted to propose here.

More recently, a PhD thesis by Bouassida proposes the same concepts as above in waveform synthesis, but scaled to a modern 28 nm SOI process [Bou16] for an energy efficient software defined radio transmitter intended for 5G. Comparing the transistor speeds of [Del75] and [Bou16] gives us a clear perspective on the power of continuous digital scaling for the past 40 years; it also highlights that some of the 1 bit techniques from the 1970 is being re-evaluated today.

Walsh sequences have uses in communication, where the orthogonality of the codes can be utilized as long as the sequences can be time-aligned. There is also a link to maximum-length sequences which is briefly discussed below.

5.3.3 Walsh and m-sequence

There exists a link between the Walsh transform and a m-sequence. M-sequences can be used to generate a “Cyclic S-matrix” [HS79], which can be re-arranged into a Walsh/Hadamard matrix. This transformation enables the use of the Fast Walsh transform and hence an efficient m-sequence correlation processor. The reader is referred to the literature for details [Xia92].

The Fast Walsh transform is identical to the Fast Fourier transform, reducing the number of operations from $O(N^2)$ to $O(N\log(N))$. It should be noted that a Fast Walsh transform reduces the number of *operations*, but that the operations themselves goes from the trivial multiplication of ±1 to integer addition/subtraction at the additional layers. This distinction does not matter on a general purpose processor, but for a custom implementation we can take advantage of the triviality of multiplying ±1 and summing the results with an increment by-one circuit. One can therefore assume that for a custom implementation there exists a tradeoff depending on the number of coefficients, where the saving obtained by going from $O(N^2)$ to $O(N\log(N))$ operations will eventually outweigh the advantage of working with a single bit. So for a reasonable N a single bit realization should be carefully considered, especially since it can lend itself to an asynchronous implementation. This tradeoff is however outside the scope of this thesis, and so we leave this as a challenge to future generations.

We will now turn our attention to two alternative baseband FMCW architectures, these also avoids the traditional anti-aliasing filter and ADC and the first avoids a Fourier transform by a bank of digital filters.

5.4 N-path filter bank

We will here briefly discuss using a bank of n-path filters as a direct method of obtaining the Fourier transformed beat view, without using a discrete

Fourier transform. To the best of our knowledge the bank-of-digital n-path filters is a novel concept. The idea is sketched in figure 5.14 and consists of taking the beat signal and filtering with a set of n-path bandpass filters. Each n-path filter mixes the input with a n-phase clock and low-pass filters the result. The filter is amendable to a fully digital implementation as seen in figure 5.15.

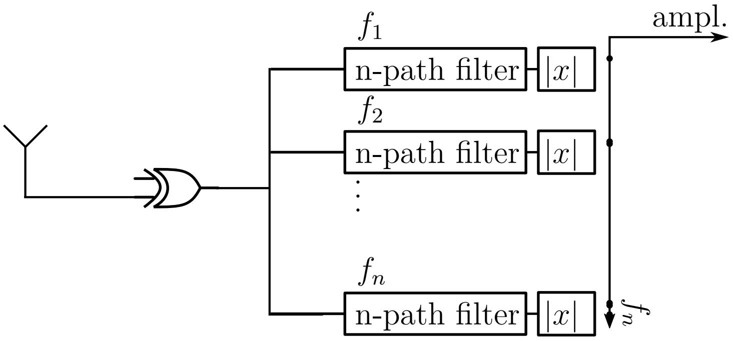


Figure 5.14: Filter bank as Fourier transform, using n-path filters.

The technique used is simply a down-conversion and low-pass filtering, each "filter-bank" uses a different clock frequency in the down-conversion. We can either employ a bank of multiple filters, or a single or subset of filters with an adjustable/selectable frequency.

A n-path filter has a bandwidth of [GKSN11]

$$BW_{\text{N-path filter}} = \frac{1}{2\pi NRC} \tag{5.13}$$

around f_{clk} and an idealized Q factor of

$$Q = 2\pi RCNf_{\text{clk}} \tag{5.14}$$

the RC time constant sets the integration time, the longer we wait, the sharper the filter becomes, which is multiplied by the number of phases N.

One of the conventional drawbacks of a n-path filter is that the filter has harmonic responses [GKSN11], for a square wave input however, the input signal already has harmonics, and so the harmonic responses are expected and must be dealt with by different means. This effect is illustrated in figure 5.16, where we have a single n-path filter with a constant n-phase-clock,

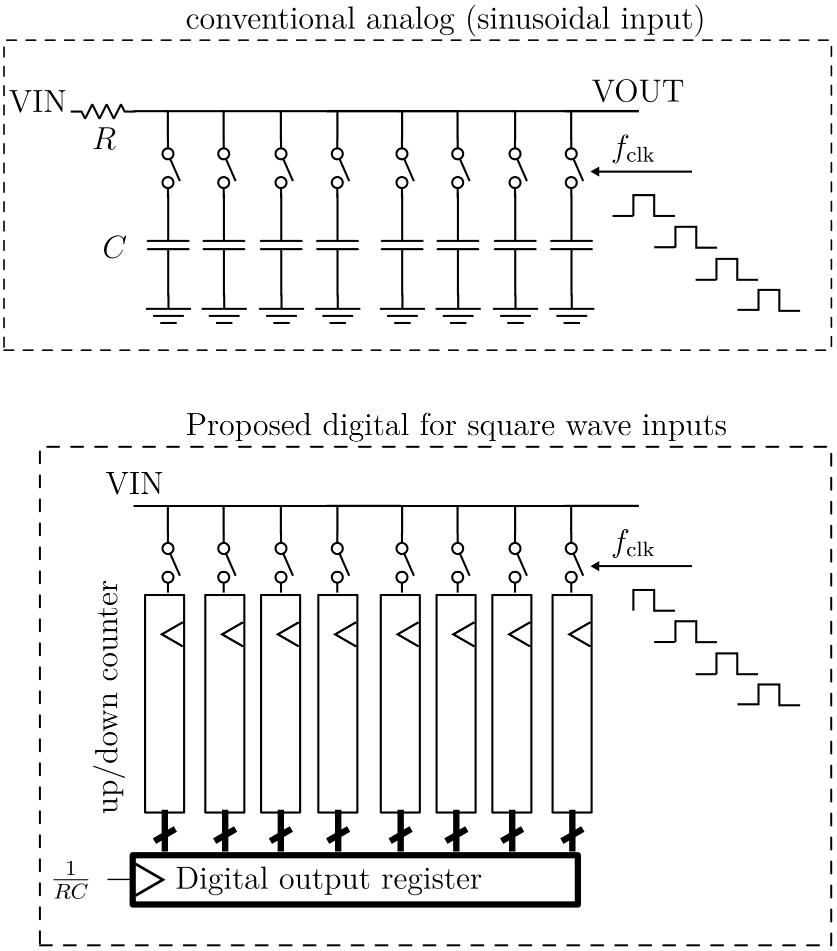


Figure 5.15: Conventional and proposed n-path filter. The digital implementation uses up/down counters as the capacitive storage elements, after a desired integration time $T_{\text{int}} = RC$, the values can be read out and the counters reset (reset not depicted).

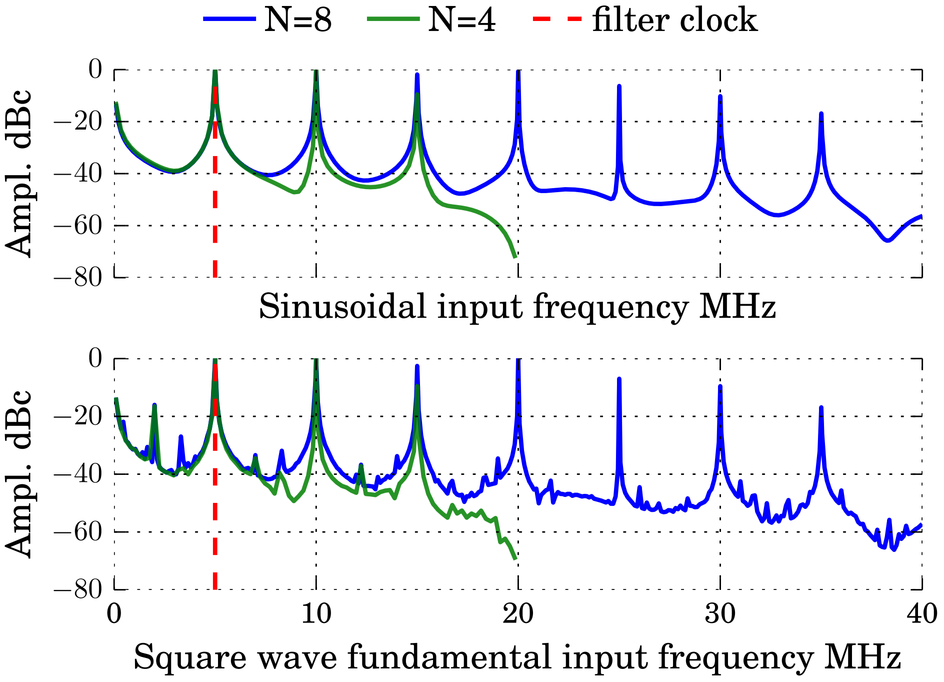


Figure 5.16: Transfer function of a idealized single n-path-filter (sweeping the input frequency) with RC=16.7 µs. Top: sinusoidal input, bottom: square wave input (sweeping the fundamental frequency of a square wave).

we then sweep the frequency of a sine-wave (top panel) and the fundamental frequency square wave (bottom panel) and record the Root Mean Square (RMS) amplitude of the output. As is seen, the filter is unable to distinguish between a square wave input and a sine wave input, which for our applications is fine, as we have ways to avoid the ambiguities of the harmonics by for instance ensuring the beat spectrum is high enough in frequency to only contain fundamentals (see Paper I).

A second drawback is folding, around Nf_{clk}, which causes the spurs seen in the simulations. Again, taking multiple measurements with different delays might be beneficial to suppress these spurs.

The filter bank concept is tested on a simulated FMCW beat spectrum in figure 5.17, where the top panel shows the result of a conventional digitization and Fourier transform while the bottom panel shows the output of a filter bank. We note that we can use the integration time, RC, to reduce the resolution and therefore cover the entire spectrum with fewer filters. In principle, as RC is relatively easy to adjust, as it is only the slow digital clock as depicted in figure 5.15, we can use a fast RC to “search” for responses with a coarse bandwidth, and then focus on regions of interest to resolve nearby targets. Creating a smart and adaptable single chip digital radar.

5.4.1 Relationship to a superheterodyne receiver

The proposed architecture in figure 5.14 reduces to a superheterodyne receiver if we do the following simplifications

- Use only a single filter at f_i
- Use 2 phases, spaced 90° apart, giving us I and Q.

As a radar, this gives us a system that responds to targets at a set range, giving us not only the amplitude, but also the phase. The I and Q is here crucial for a smooth output, as we need it to determine the amplitude without fading when going in and out of phase.

Using additional phases, as we do here, gives us increased averaging and hence lower noise and it pushes the frequency folding to a higher frequency (by N).

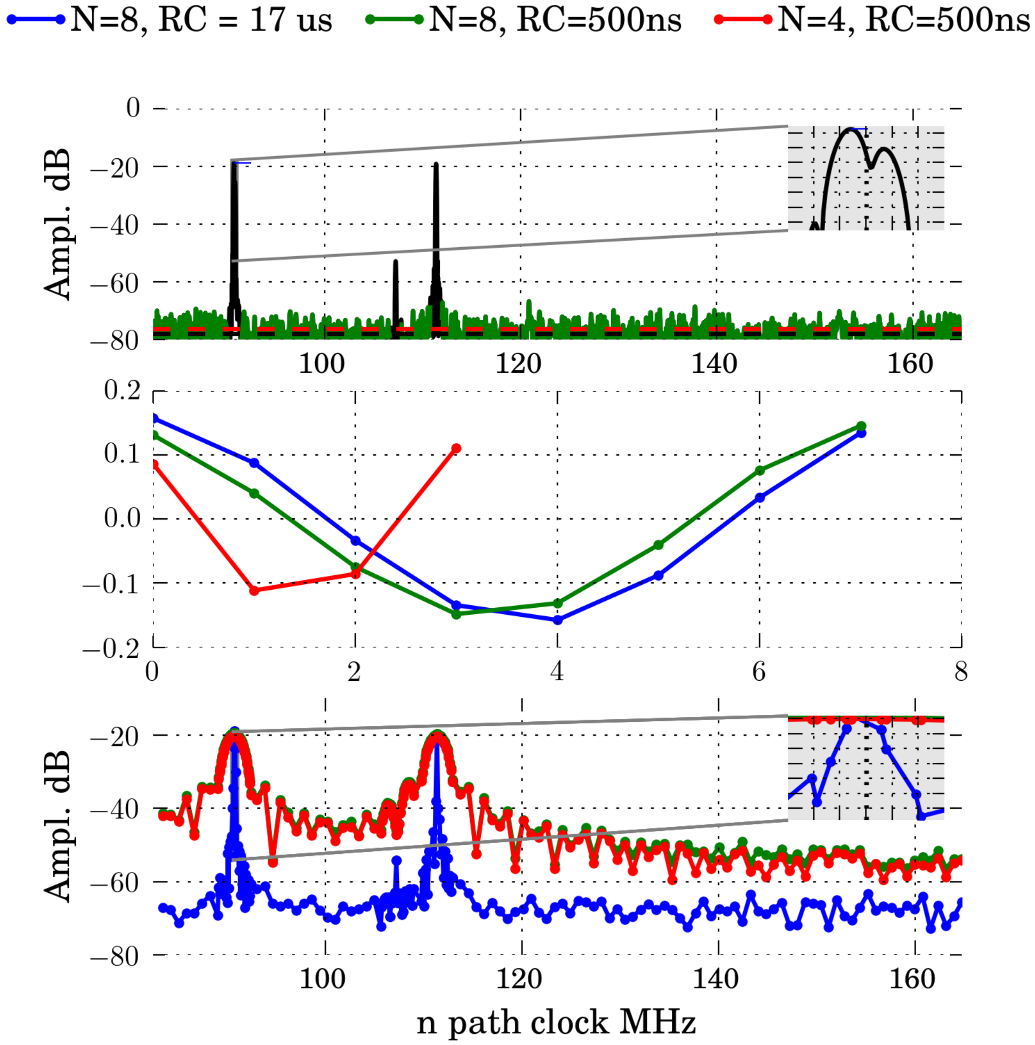


Figure 5.17: Top: Example output FFT spectrum of a square wave FMCW radar with delay to avoid harmonics in band, 4 target, one weaker and 2 close together. Middle: Output value for each of the N counters when the n-path filter is tuned to one of the beat frequencies. Bottom: Recording the RMS amplitude of the counter values of each n-path clock frequency, leading to a Fourier transform by n-path filters.

5.5 From autocorrelation to Fourier domain

This section is a direct application of the Wiener Khinchin theorem, where the Power Spectral Density (PSD) of a signal can be found as the Fourier transform of the signals autocorrelation. What this state, is that instead of computing

$$X_f[k] = \left(\sum_{n=0}^{N-1} x[n] \cdot e^{-jn\omega_k \Delta t} \right)^2 , \tag{5.15}$$

we achieve the same result by first computing the autocorrelation

$$r_{xx}[n] = \frac{1}{N} \sum_{k=0}^{N-n-1} x[k]x^*[k-n] \tag{5.16}$$

and computing the Fourier transform of $r_{xx}[n]$ instead of $x[n]$

$$X_f[k] = \sum_{n=0}^{N-1} r_{xx}[n] \cdot e^{-jn\omega_k \Delta t}, \tag{5.17}$$

On the surface, this may seem a little pointless, but we here want to explore a hardware solution that *estimates* the autocorrelation, by only computing a subset of the lag-terms $r_{xx}[n]$ with single bit processing. The idea is then instead of low-pass filtering and sampling the sequence $x[n]$; we would like a single bit realization that computes the $r_{xx}[n]$ terms (which is just a XOR gate and a counter) directly. The signal can then be Fourier transformed to get a PSD view of the beat spectrum.

For brevity, we here ignore the time sampling of $x[n]$ and view it as a continuous time signal $x(t)$, this idealizes the counter to an integral so we obtain

$$r_{xx}[n] = \frac{1}{T_m} \int_{t=0}^{T_m - n\tau_a} x(t)x^*(t - n\tau_a)\, dt \tag{5.18}$$

$$\tag{5.19}$$

Now if we look at the discrete Fourier transform of this, it is clear that we only need time-lags at the Nyquist rate $1/(2\tau_a)$ (where the factor of 2 is due to using real samples). The maximum time lag $(N-1)\tau_a$ (where $n \in [0, 1, \ldots, N-1]$), should optimally be the entire length of the signal, so $N = T_m/\tau_a$, but since the autocorrelation falls off in amplitude a shorter maximum time lag might be permissive [BT59].

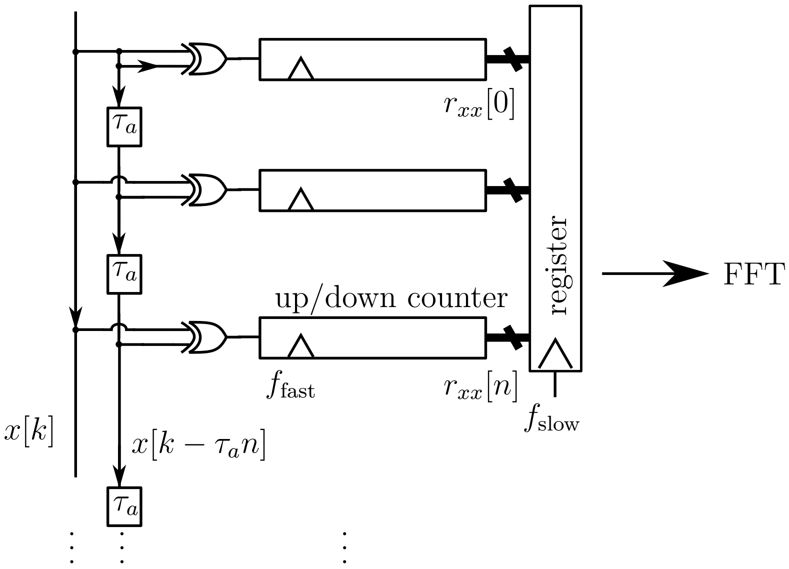


Figure 5.18: Principle of the proposed single bit autocorrelation circuit, shown with a continous time delay line. To avoid the large fanout on the $x[k]$ line, additional (equal) buffers can be added to each stage for pipelined processing. For brevity we here consider the counter clock (f_{fast}) equal to the simulation time-spacing (in the 100 GHz range), while the register and FFT clock speed only needs a single transition for every frame (so $f_{\text{slow}} = 1/(N_{\text{sweeps}} \cdot T_m)$).

In hardware, the autocorrelation can be implemented by the scheme presented in figure 5.18. We note that, from a hardware perspective the solution is very similar to the n-path-filter bank solution presented in the previous section (with $n = 1$). The similarity between autocorrelation and filter-bank is elaborated more in [Wei63]. It should be noted that a single bit autocorrelation needs the following correction [Wei63]

$$\hat{r}_{xx} = \sin\left(\pi/2 \cdot r_{xx}\right). \tag{5.20}$$

An initial proof of concept simulation is shown in figure 5.19, where we compare:

- Top: An idealized receiver with a structure similar to figure 5.1.

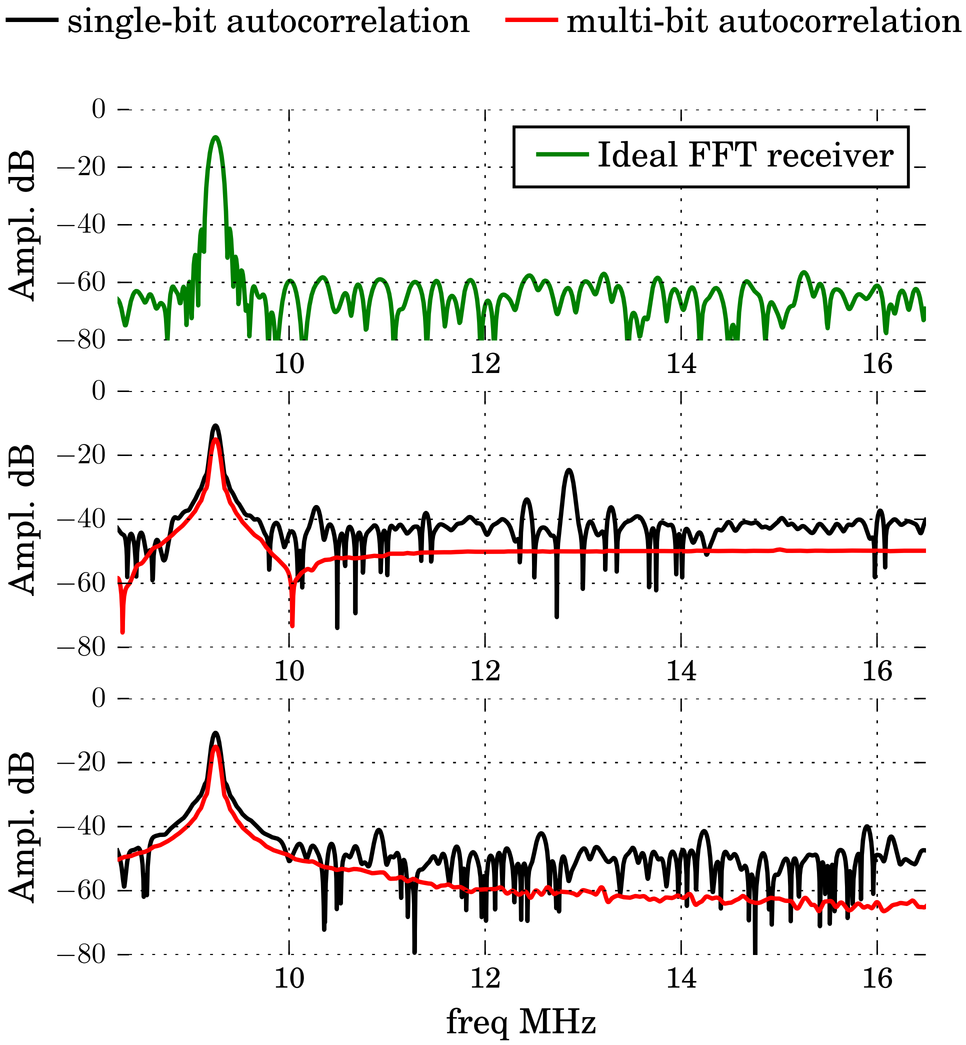
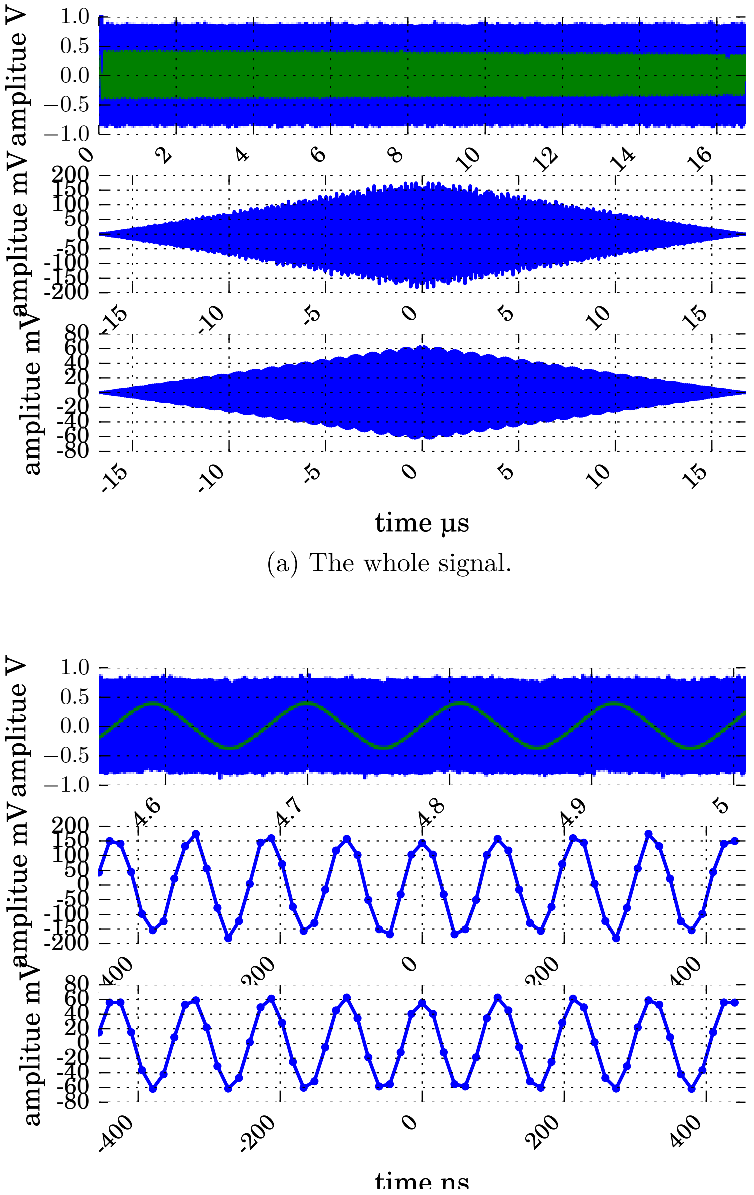


Figure 5.19: top: ideal baseband, single scatterer with harmonics out of band. Instantaneous SNR 0 dB with 128 averaged sweeps from 600 MHz to 2.67 GHz in 16.7 µs with the maximum unambiguous range set to 10 m. Next: comparing a single and multi-bit autocorrelating baseband receiver with 1.1×10^3 time lags with a $1/4 f_{\mathrm{beat}_{\mathrm{un}}}$ =15 ns time spacing. At the bottom, the time-lag sampling frequency is increased to 4.42×10^3 lags at a $1/16 f_{\mathrm{beat}_{\mathrm{un}}}$ =3.8 ns spacing.



(a) The whole signal.

(b) Zoomed view of (a).

Figure 5.20: Time domain view of the simulation in figure 5.19. Top panel: the time domain signal, filtered in green for visualization. Next: averaged single bit autocorrelation output and at the bottom the multi-bit autocorrelation output as a function of time lag τ_a.

- Middle: Autocorrelation receiver similar to figure 5.18 where the time lags are computed for every $1/4f_{\mathrm{beat_{un}}}$.

- Bottom: decreased time lag spacing and hence number of points (in essence-oversampling).

For the autocorrelation, we include an analog/multi-bit simulation, showing the similar performance of a 1 bit autocorrelation and a fully analog as pointed out in the literature. Referring to the architecture in figure 5.18, for analog autocorrelation, we need an impractically long analog tapped delay-line, a large number of analog mixers and low-pass filters and an equal number of ADCs if we want to digitize the resulting autocorrelation. Clearly highlighting the benefits of single bit processing.

For reference, the time domain view of figure 5.19 is shown in figure 5.20, we note from (a) that the autocorrelation gets weighted by a triangular window due to the $1/N$ normalization, as we get further out into the sum of (5.16). We have here replaced the $r[0]$ term (which is 1 by definition), by the cubic interpolation of its neighbors, since a value of 1 created a large discontinuity that affected the FFT output. Since the autocorrelation is symmetric, only half the terms are computed and the remainder is found by symmetry $r[-k] = r[k]$.

We should note that the Wiener Khinchin theorem only holds for band-limited signals (and hence infinite length sequences), neither of which is true in our case and we therefore see improvements when oversampling the number of lag terms. In addition, the estimate is said to be an inconsistent estimator, in that it is only true in the statistical mean sense and not theoretically suited for deterministic signals. Some initial simulations do however confirm that the results presented above have negligible variance when modifying the simulation noise seed. Using a shorter maximum time lag as hinted above reduces the variance further, at the cost of reduced resolution.

5.6 Concluding baseband

We have shown the conventional FMCW baseband solution, consisting of an anti-aliasing filter and a multi-bit ADC. This conventional solution is shown to be viable for low instantaneous SNR digital radars by a simplified idealized analysis and comparing to available literature.

The conventional solution is however in-flexible and may not be optimal, especially when considering the required anti-aliasing filter. Simply sampling the single bit XOR output at Nyquist is shown to yield sub-optimal results, but oversampling and decimating by novel continuous time CIC structures can be a viable option.

To avoid a FFT, we explored the Walsh transform, which is shown to be very promising in some areas, but have significant drawbacks such as shift-variance and the fact that our mixer output is more efficiently described as a sum of triangular or sine waves. A straight-forward implementation that avoid the FFT is a bank-of filters, where a novel digital n-path filter is proposed as a viable alternative. Similar in implementation is an auto-correlation processor, that "samples" and integrates the signal by computing the autocorrelation while the signal is still in its single bit form.

Other than the conventional anti-aliasing filter and ADC, none of the above methods have been implemented and extensively studied. This chapter merely serves as ideas to potentially novel, energy efficient and flexible baseband solutions.

Chapter 6

The last chapter

"For the impatient"

We will in this last chapter give an overview of the previous pages and also give a brief overview of the attached and re-formatted publications. See page 141 for the publication list. We will give some thoughts on future work, discussions and limitations of the presented study before we conclude in section 6.4.

6.1 Summary of thesis and published papers

We have walked through the thought process of converting a traditional analog intensive radar into an almost entirely digital radar. The radar requires an analog comparator, but is otherwise realized with digital gates only. We have also laid the foundation for understanding a range of classical radar architectures based on the range-extraction mechanism, namely pulsed and pulse compression architectures. The published papers show our proposed digital intensive version of these architectures, which can all be feasibly integrated on a single chip radar system. Measurements from two prototype single-chip implementations are also reported in the publications, where we also explore novel asynchronous continuous time design techniques.

Digital realizations of the frequency source have appeared in the literature, either as a DDS (first proposed in [TRG71]) or as a digitally intensive PLL (using a Digitally Controlled Oscillator (DCO)) [Sta11], where the majority of the sub-blocks are digital; while the output attempts to be analog without harmonics. DDS architectures are even appearing in low-pass filtered single bit output form by PWM [WMR^{+}04] or delta-sigma modulation [SADRH08, Sot16]. These architectures can take some advantage of the continuous digital scaling, not only benefiting from the increased speed of sub-micron processes but also the integration of power and area efficient digital logic in the same technology. They do however suffer from reduced flexibility as the frequency range and noise requirements must still be decided early in the design process.

We have here taken a system approach, where we allow harmonics to be in every component block and where we even take advantage of the increased bandwidth of the harmonic sweep. This enables all circuit blocks to be clipping, greatly easing the implementation complexity and ensuring we can take full advantage of the increased time over amplitude resolution in modern processes while being very flexible in terms of not only bandwidth but also radar architecture.

We have focused mostly on a digital FMCW radar, where the pulse compression is achieved by Fourier transforming the constant beat frequency when mixing a reference chirp to a range delayed-chirp. The novelty is that the waveform is not a sinusoidal, but a square wave, enabling the use of digital components in the signal chain. The architecture relies heavily on averaging and will have comparable performance to an idealized fully analog system when the noise level is high. This observation leads us to a radar with low instantaneous transmit power (to get close or below the thermal limit) and considerable coherent averaging. A low instantaneous SNR gives relaxed requirements to the entire signal chain, allowing us to implement sub-optimal components for reduced costs and complexity, while still giving us a system capable of high SNR and resolution radar images after averaging.

Throughout this work and especially in the baseband chapter, we have not worried about a mathematically optimal design. Although an optimal design can be a worthwhile goal to purse, the reality is that a digital system needs quantization in both amplitude and time and that filters, if they are even feasible to implement, are not infinitely sharp. Real world noise and interference will also always be a limiting factor and we have argued that high resolution systems are wasteful in low-SNR scenarios. We therefore explore everything from poor amplitude resolution ADCs, to single bit sampling with and without filters and novel single bit digital filter banks and autocorrelation processors with inherent aliasing and amplitude resolution issues. Since the instantaneous SNR can be low in our radar system, these seemingly sub-optimal designs can greatly improve the energy efficiency and the integration flexibility as we are not wasting energy computing and converting with a higher than needed resolution.

A short summary of the published papers is given next

- **Paper I** Introduces the square wave FMCW radar and focuses on the transmitter, showing some preliminary measurements from the first prototype chip. The chip did feature a novel shift-register based register, which failed to work correctly and so the publication only shows the simple sequence 11001100... instead of a true FMCW chirp. From this simple sequence, we then extract jitter, decomposing it into deterministic and random. We could then bring these jitter statistics into the high level radar simulation and come to the conclusion that a finer time-resolution was more important than the jitter. Section 3.1.2.1 shows this conclusion even clearer, where the jitter level is beneficial and not detrimental to the SNR performance. The simplest method of moving the harmonics to a separate band was also shown.

- **Paper II-III.** Conference paper II was extended in journal paper III, so we cover both here. These papers add an additional harmonic removal method, using a staggering delay in the transmit path to move the harmonics and fundamental by different amounts. The results can the either be averaged (summed) or correlated (multiplied) for a harmonic free range spectrum. The papers also show a post layout XOR mixing and the Journal paper include several measurements, some just on the transmitter and also a full-system FMCW measurement where the oscilloscope acts as the baseband ADC. Main contribution of these papers is the introduction of square wave realizations of CW, SFCW, PN-sequence and pulsed radars in addition to the previously introduced FMCW concept.

  Measurements from a new chip realization is included in III, this time with a memory module provided by the foundry and also with a receiver front-end, consisting of a comparator and XOR mixing circuit. The reader may want to skip II, though the text is re-phrased, all of the ideas presented in II are repeated and extended in III.

  The journal article also includes analytical treatment, which was particularly essential in understanding the SFCW case.

- **Paper IV** is a technical paper on the transistor level implementation of our asynchronous transmitter, which we promised to present "in a future work" in I. The circuit was re-purposed as a DTC, so that we could focus solely on a single "row", without focusing on the system aspects. To the best of our knowledge, the wavefront-architecture used; has previously only been presented as a serializer for communication and not recognized as a digital to time converter. It should be noted that as presented, the circuit takes a rising edge on the MODE/select input and, depending on the digital thermometer code, gives a new delayed rising edge. The falling edge is not-modified. In addition, some of the design choices made (to support arbitrary digital codes), does not necessarily make sense if the digital codes are used solely as thermometer coded and so for a finer resolution DTC one may want to focus even more on the delay; rather than the driving strength of each stage.

  We should note that some of the reviewers indicated that the whole circuit was not shown, we took this as a complement to the simplicity of the proposed DTC.

6.2 Discussion and limitations of this work

A mathematically rigorous analysis of the proposed radar architectures is beyond this authors ability, but we have analyzed the following cases. The noise-free, non-quantized, case is straight forward as the square wave nature only adds harmonics that can be tedious, but not difficult, to include in a time-frequency/phase analysis. In a noise free case, the receiver is also simple, since it acts as a flash-ADC with a resolution of $2^{N_{\text{sweeps}}}$ and where processing can either be done before or after averaging (in single or multi-bit form), as long as each operation is linear. Though a proof of the linearity of the mixing operation has not been attempted and only shown by simulation to be linear for sufficient number of threshold levels.

Quantizing in time is also straight-forward to analyses, as it adds folding, which is the main drawback of the proposed implementation. The proposed methods of dealing with harmonics do not take folding into account. A truly continuous time implementation with a transmitter similar to figure 1.3 should be carefully considered, while introduction of randomness and noise shaping could (as hinted in section 3.2.1) provide a more digitally friendly alternative. In addition, since the folded "noise" is deterministic, a simple FFT of the data provides a sense of poor SNR and dynamic range, but more advanced signal processing, taking the expected folding into account, should be further explored.

The main advantage of the single-bit system is for high noise scenarios, where a purely simulation based approach is taken. This stochastic and non-linear problem has however been studied in detail by the Stochastic Resonance (SR)/SSR community, though not necessarily for the input statistics and output figure-of-merits of a radar system.

As a figure of merit, we have focused mostly on the SNR, where the estimation method is mostly a naive integration of the non-signal part of the spectrum, see details on page 51. The estimated SNR then includes all non-idealities of the output, namely the remaining thermal noise, distortion of the signal and the matched-filter-mismatch. The last effect is the mixer products that does not result in a constant beat frequency, but rather appears as non-constant interfering signals that will land in the beat spectrum due to the sampled transmitter. Lastly, a given SNR level does not guarantee the detection of weaker scatters, which among other factors, will depend on the noise since the system does contain a non-linear swept-threshold receiver (a SSR system).

Especially for the FMCW case, the resolution will degrade as the overlap between the transmitted and received signal decreases (as the two-way-travel time increases). This is especially a concern for the "moving out of band" harmonic technique, where depending on the wanted unambiguous range and the sweep time, one could potentially end up removing the overlap entirely. The technique and especially the reported 3 times increase in resolution; does therefore assume a sweep time much longer than the maximum-two-way travel time $T_m \gg 4r_{\text{un}}/c$. In addition, the resolution is never extracted and the slight decrease in the resolving ability when including jitter, (as clearly seen in the insets of figure A.9, page 155) is never quantified or analyzed.

An idealized view of the world is taken and our main concern is the radar hardware and not the signal propagation in the channel, the reader should keep figure 2.3 in mind. We should re-state that we do not need to assume the harmonics of a square wave radar "survives" the channel, as long as the fundamental can be amplitude clipped at the receiver.

6.3 Future work

We have above pointed out some limitations of the current work and all of the above makes for interesting future challenges, we will here give a brief list of additional interesting future research or technical realizations.

We have in this work opted for a continuous time/asynchronous implementation. A conventional synchronous/clocked design cannot deliver the same time-resolution at the same power-efficiency but opens the door for

- FPGA or other general purpose digital chips for implementations or prototyping, requiring only an (external) comparator and an antenna driver (PA).
- Multi-bit, we have only compared the extreme “analog” vs “single-bit”, but intermediate levels of bits can provide the needed performance when the clock frequency is the limiting factor. For an asynchronous solution, the overhead of synchronizing multiple bits quickly becomes excessive, which is why we have focused only on single bit; as no-synchronization is needed.
- Especially the SFCW architecture would benefit from two synchronous transmitters that each transmits its own frequency, this is challenging in an asynchronous implementation but trivial in a synchronous design.

The asynchronous implementation does not have any true time-reference and so will vary with process, temperature and voltage and only give a “relative” distance estimate. This ensures the circuit always functions and at its maximum speed, but some care must be taken depending on the application.

For measurements with a known (fixed) scatterer in the scene, we only need a relative distance and so the lack of time reference is not problematic. The radar could be improved by including a known reference path, which can be used for occasional calibration by occasionally connecting the transmitter and receiver together, through a known delay. In addition, depending on the expected temperature and voltage changes and the required (absolute) accuracy, a simple one-time look-up-table calibration can be sufficient. If the only goal is to extract what the absolute time-reference of the transmitted signal is, then we discuss a few simple options in section 4.1.2.2.

Future work if the above is not acceptable, naturally includes

- a synchronous implementation with a stable reference clock

- a tunable transmitter, such as the current starved wavefront Serializer/Deserializer (SerDes) transmitter in [LKK$^+$05] that reduces the supply voltage sensitivity with an on-chip-reference voltage
- a feedback frequency generation, such as a PLL with a stable reference clock as input.

Lastly, many of the implementation choices made, in particular the transmitter being a circuit that reads a bitstream from memory, is a very flexible design. An interesting task for future generations, would be a less flexible circuit, aimed for some specific application and only supporting a single (square-wave) radar architecture. That said, the flexibility of the precomputed bitstream has not been taken full advantage off and future studies on the optimal bitstream for a specific application would also be an interesting topic.

6.4 Conclusion

We have presented a digital future for radar, showing that we can not only use a large number of digital circuit block, but also a digital signal, namely a single bit square wave, to realize a flexible radar system for digital integration. For a mixer/frequency based radar, a square wave has harmonics that we have shown to be non-problematic to remove and how to constructively utilize them for increased resolution. An interesting single-bit receiver is used, the sweep-threshold receiver is especially useful for high noise scenarios, where not only noise is beneficial to the architecture, but where performance is close to an analog system. The single bit receiver also allows for single-bit processing, greatly easing the implementation complexity and energy efficiency and providing great flexibility.

A continuous time implementation is integrated on a single CMOS chip, with measured results confirming the feasibility of several digital radar architectures. Pulse compression radars, utilizing either a single mixer or a correlator, is a key enabler when peak power is limiting the range and dynamic-range in a given time-span. Coherent integration, in the form of averaging and pulse-compression processing-gain, can not only improve the SNR, but allow the instantaneous SNR to be low enough for cheap and simple CMOS integration.

Part II

Publications included in the thesis

Appendix A

Paper I. UWB waveform generator for digital CMOS radar

The following is a re-formatted version of
Øystein Bjørndal, Svein-Erik Hamran, and Tor Sverre Lande. UWB waveform generator for digital CMOS radar. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1510–1513, 2015. doi: 10.1109/ISCAS.2015.7168932

PhD Candidate contribution: Wrote the majority of the paper, implemented and measured the presented solution.

Supervisors contributions: Contributed to the original system concept, both the radar/signal-processing and the implementation.

Abstract

A digital asynchronous waveform generator is presented and analysed for FMCW chirp generation. A novel delay line based sequencer is proposed, where the maximum output frequency scales with the gate delay of the process. At 2.9 GHz, 0 dBm is measured over a 50 Ω load, drawing 8 mA from a single 1.2 V supply. System simulation show the flexible generator can be programmed for either large bandwidth (resolution) or improved dynamic range. With a sweep to 2.9 GHz a free space resolution of 10 cm is feasible.

A.1 Introduction

Technology scaling in CMOS is driven by the continuous demand for higher performance digital circuits. Therefore modern technology favor resolution in time over resolution in amplitude [1], disfavoring conventional analog and RF circuits. For FMCW signal generation, feedback solutions like PLLs are often employed to reduce phase noise but at the cost of slow chirp rate and limited bandwidth due to the tunability of the VCO. As an alternative, a DDS (Direct Digital Synthesizer) can be utilized as a feed forward solution, at the cost of power consumption limiting the maximum frequency.

In this paper we extend the concept of time average frequency [2] to FMCW chirp generation, which is evaluated and measured, preparing for an all digital radar in CMOS. Single bit FMCW chirps, appears as early as in 1984 [3], but was discarded due to the low clock rate of digital circuits at the time [4]. More recently single bit chirps have appeared in the bio-impedance community for wideband excitation [5]. Section A.2 addressees the side-effect of introducing harmonics and outlines a novel solution.

Figure A.1 shows the proposed waveform generator used in a fully digital FMCW radar. The waveform generator is preset with a linear chirp, which is transmitted from the upper antenna. Echoes are then received by the lower antenna and immediately digitized by a sign detector (a comparator with a threshold as second input). Since all signals are digital, the analog mixer can be replaced by a digital XOR gate. The XOR will produces the difference (and sum) between the generated waveform and received waveform. The difference in frequency will be directly proportional to the two-way travel time, producing the beat spectrum.

The presented waveform generator is however fully programmable for any digital bit sequence, enabling CW, FMCW, M-sequence or a pseudo noise sequence radar. The only limiting factor on chirp rate is the desired frequency range of the beat spectrum, the proposed system can e.g. sweep from 1 GHz to 10 GHz in 10 µs, yielding a beat spectrum from 0 to 600 MHz for targets between 0 and 100 meters.

For radar applications the bitstream needs to be shifted out of the chip at a high data-rate to enable wideband excitation. Shifting out a bitstream at 6 GHz, is hard to do with a clock based approach. To avoid the high frequency clock, a Continuous Time Binary Value (CTBV) [6] solution is sought. In CTBV the time resolution is decided by the gate delay of the

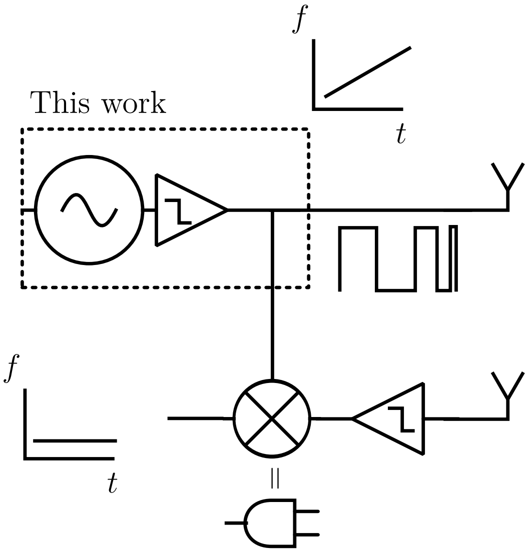


Figure A.1: Digital FMCW radar with a square wave chirp and a sign detector on the input, the mixer can then be a digital AND or XOR gate. Harmonics are not shown.

process and not on the maximum clock frequency. A similar continuous time scheme is proposed by [1, 2, 7]. The use of a global clock to synchronize all of the elements makes sense and eases the design process as it can be fully automated by modern synthesis tools. It does however limit the maximum frequency, as ample margins must be inserted to ensure the slowest blocks can finish in the slowest process corner. The clock is also a large power drain, for a signal which in itself does not carry any real information.

Using delays as a sequencing element gives much greater flexibility and avoids high frequency global signals. Subject to slow process corners the delay line will "move" slower, but ideally stay in sync. Therefore the delay line based solution in figure A.2 was implemented. The circuit works as follows: In load mode, the delay line gets new values from the "memory", which is loaded in parallel onto the delay line. When the select goes high, the bits will "flow" out of the chip. It should be noted that the solution relies entirely on standard static cells, without any feedback for latches nor refresh for dynamic logic. A similar serializer was proposed in [8] for SerDes applications. Our implementation is different from [8] and will be presented in a future work. Our novel usage of the serializer, in the waveform generator, is presented in section A.3.

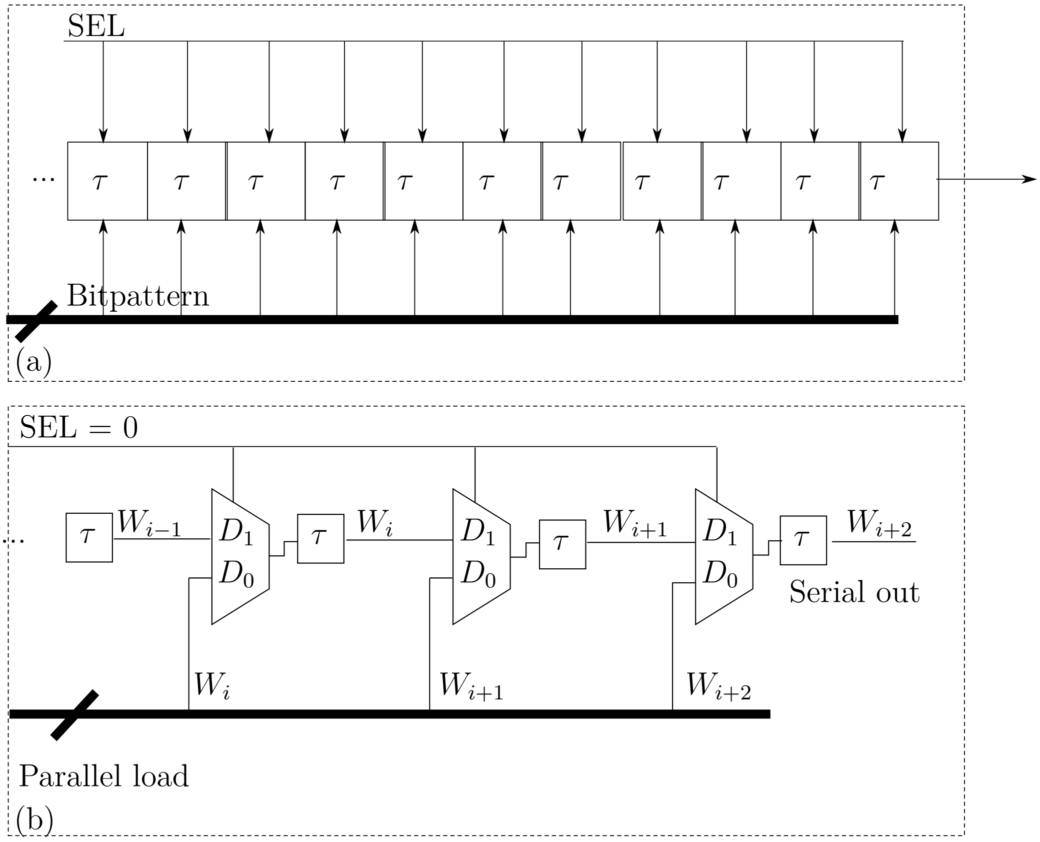


Figure A.2: The utilized serializer works by reading out a bitstream using a delay line as a sequencer. (a) shows the delay line with a “load” feature, where a bitpattern can be loaded in parallel onto the delay line. In (b) the circuit implementation is shown. The circuit has two modes, when SEL is low a new bitpattern W_i is loaded into the mux delay line, and when SEL goes to 1 the bitpattern will “flow” out.

The “memory” can be a long shift register, a standard memory with an address accumulator or any circuit which generates the desired bit sequence. In this prototype, this part has not been successfully implemented, so only a fixed pattern is presented.

A.2 Quantization in amplitude and time

Quantization in time, or sampling, is well understood, leading to folding around the sampling frequency. Quantization in amplitude is however a non-linear problem which is not as straight forward, but the specific case of an FMCW chirp is discussed in [3]. In the work by Johnston the harmonics are filtered out, but in this work we aim for a wideband waveform where the first harmonic is allowed to overlap with higher harmonics. Without any compensation, the higher harmonics will appear in the beat spectrum as false targets. This is illustrated in figure A.3(a) and can be expressed as a (weighted) sum of Dirac pulses,

$$X_{\text{beat}_n}(f) = \sum_{m \in [1,3,5,\dots]}^{\infty} a_m \delta\left(f - m f_{\text{target}_n}\right) \tag{A.1}$$

for every target n. This is particularly troublesome for close targets (e.g. the direct coupling), as multiple false targets will appear in the beat spectrum.

It should be noted that these harmonics will appear as long as the mixer is digital, so simply filtering the output before transmitting will not alleviate the situation. In addition, such a filter would need to be tuned with changing frequency.

One straightforward solution to avoid the false targets, is to insert a delay on the input side, as depicted in figure A.3(b). The delay element will move the beat spectrum up in frequency, changing the beat in (A.1) to $m(f_{\text{target}_n} + \alpha\tau)$. This moves the first harmonics of interest to one band and the harmonics to a higher band that can be filtered out in the baseband or ignored. Since the waveform is a single bit digital bitstream the delay can be realized by a string of inverters as in [6].

A.3 Fabricted chip

In the original sketch in figure A.2, the delay line would need to be as long as the desired sequence. To work around this, a pipelined system was implemented and is depicted in figure A.4. A minimum of $M = 2$ rows is needed for the pipeline to work, one row is reading out values, while the last is loading new values from memory. For symmetry and flexibility, as will be seen later, $M = 4$ rows was implemented.

With M fixed, the "depth" N decides the speed of the load mode and the bus width to memory. In addition, it should be kept at a manageable number since the column is highly critical and subject to disappearing pulses. With a delay of $\tau =$ (sample rate of $1/\tau =$), the implemented $M = 4$ rows and

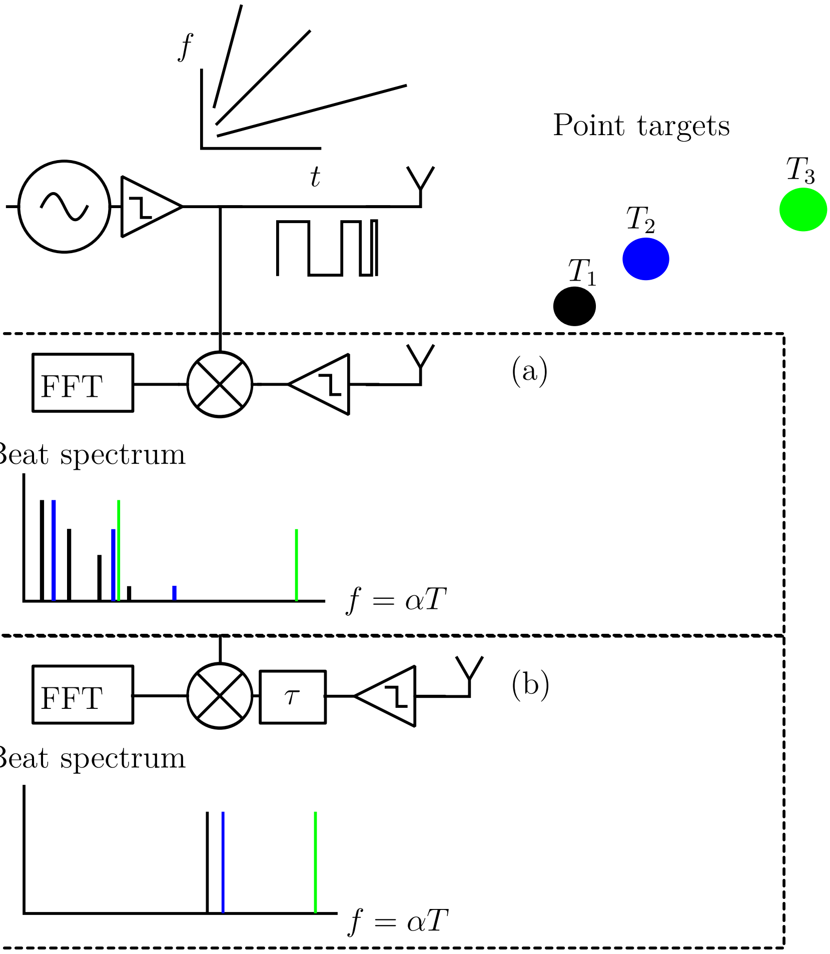


Figure A.3: A digital FMCW radar visualized with 3 point targets, direct coupling and path loss is ignored. After a frequency analysis of the mixer product the beat frequency f is directly proportional to the time of flight T for each target. The digital inputs to the mixer creates odd harmonics which appear as false targets in the beat spectrum (see (a)), by appropriate delay on the input-side the beat spectrum is clean again (see (b)).

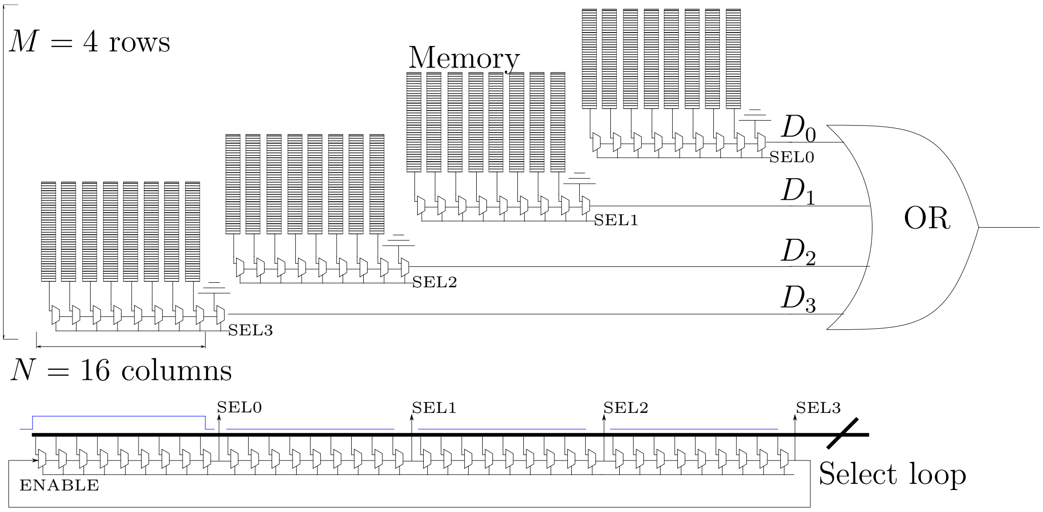


Figure A.4: Proposed waveform generator. Each of the 4 rows, depicted as 8 muxes, is implemented as the serializer in figure A.2. At the bottom, the select loop is made with the same muxed delay line, but is programmed at startup with a wider select pulse as depicted in blue.

$N = 16$ columns gives a load window of $\tau N(M-1) =$, which is a manageable speed for a 90 nm CMOS process. The timing diagram for the implemented solution is shown in figure A.5. With the simple OR solution, each row in sequence, transmit at full bandwidth. To reduce the speed of the on chip components, an alternative approach is also possible. If each of the 4 outputs are brought out of the chip, the select loop can be programmed to select two rows at the same time. By sending out e.g. 11001100 and 011001100 an external XOR/comparator can be used to up-convert the waveform frequency

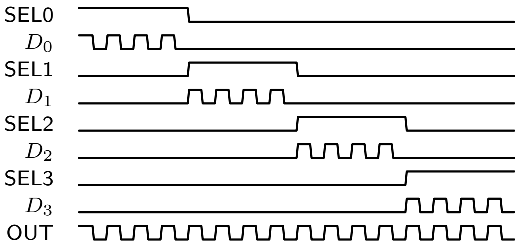


Figure A.5: Timing diagram for the implemented waveform generator in figure A.4. Each row OUTi is selected in turn by the corresponding SELi signal, OUT is simply created by OR-ing D_0, D_1, D_2 and D_3.

by a factor of 2.

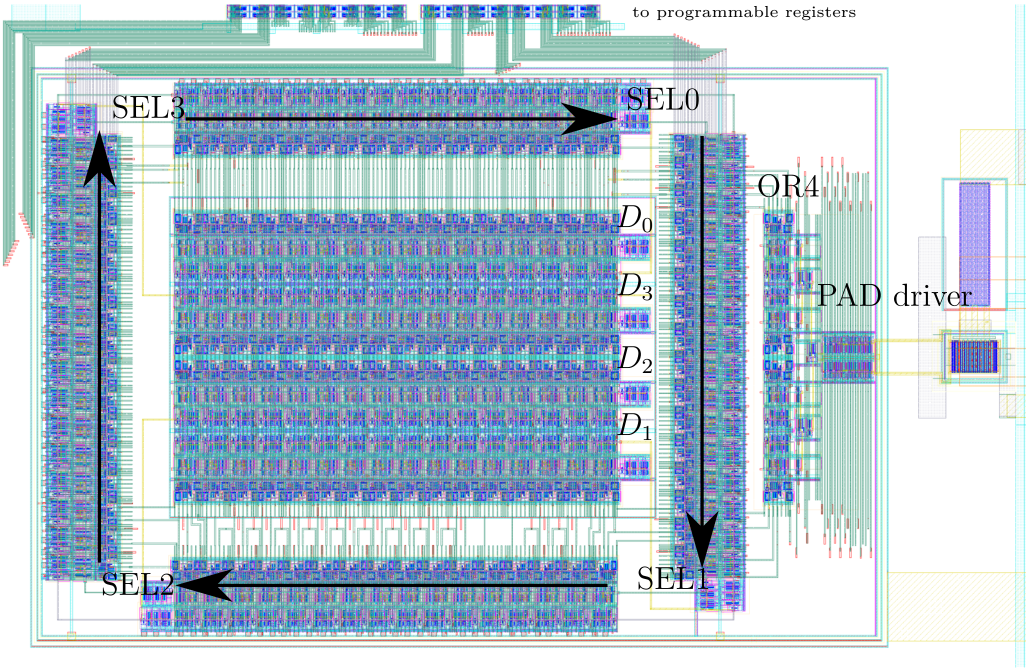


Figure A.6: Realized waveform generator. The SEL loop is connected around the four rows, with the symmetrical OR4 and the pad driver visible on the right. Only the lower metal layers and poly (in blue) are shown for clarity. The block has separate substrate connections and supply for tunability. (VDD bulk, VDD source, GND bulk, GND source)

The chip layout is shown in figure A.6. As can be seen, the choice of $M = 4$ gives a symmetrical select ring moving around the rows and columns in the middle. The circuit block measures $180\,\mu\text{m} \times 100\,\mu\text{m}$ including the output stage, which is a single common source NMOS transistor with $W = 80\,\mu\text{m}$ and a $50\,\Omega$ drain resistor.

The critical block is placed in a separate deep n-well with separate supply voltage and has the nMOS and pMOS bulk connections accessible on pads. In addition to noise reduction this enables adjusting of the nMOS and pMOS thresholds, allowing some tunability against chip to chip variations and non-symmetric layout parasitics; by adjusting the relative strength of the pull-up and pull-down.

The OR4 gate consists of three symmetrical NAND gates from Weste and Harris [9] (with appropriate inversions), where the layout ensures equal path length for all four inputs.

A.4 Measurement results

Programming the waveform generator to transmit the sequence 11001100... the output voltage was recorded by probing with a Lecroy WwaveMaster 830Z1 oscilloscope. The resulting bitstream is shown in figure A.7, in both time and frequency. While transmitting, the chip draws 8 mA from a 1.2 V supply. The data is presented without averaging nor filtering, but a DC block capacitor was used to eliminate the static current of the output stage.

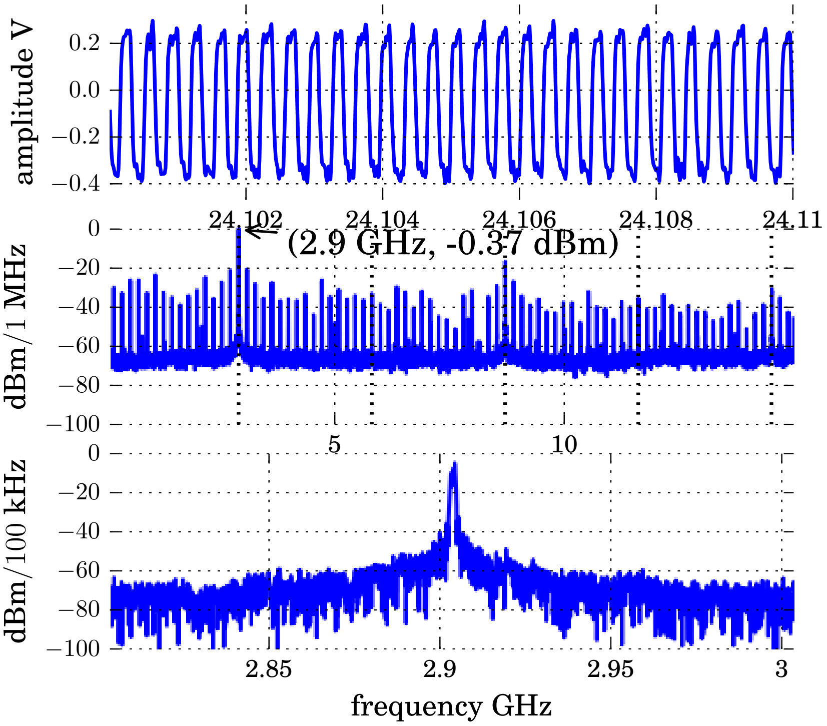


Figure A.7: Measured output when programmed to 11001100.... Top: Zoomed in view of the output. Middle: Power spectrum estimated using Welch's method [10], bottom: zoomed in view of the spectrum around the peak frequency.

When extracting the period and time interval jitter, a repeating pattern for every 16th edge was observed. This is also evident in figure A.7 (middle), as pattern noise. The "16" is simply the number of edges in one loop. The pattern noise can be explained by non-symmetric layout and process mismatch and is also present in post layout simulations. The effect is a combination of non-symmetries in the select loop, pulse "shaping" in the readout rows and non-symmetries in the OR gate. When analyzing all of the edges

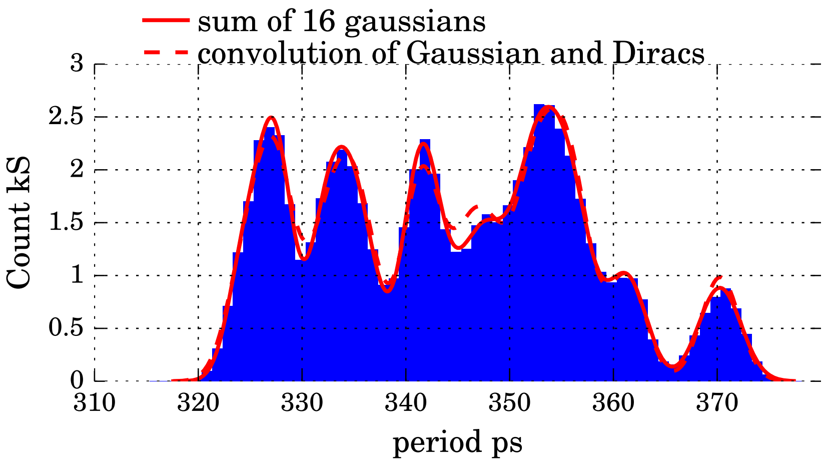


Figure A.8: Period jitter PDF estimation assuming each 16th stage follows a Gaussian distribution. Top: Extracting mean and sigma for each block and summing these Gaussions we obtain the total pdf. Alternativly, using the mean and an average standard devation the total pdf can be recreated by convolving 16 Dirac pulses with a Gaussian.

(by looking at the crossing of the mean), the total histogram in figure A.8 emerged. To recreate the distribution, each 16th sample was selected to create 16 new distributions, each found to be normally distributed. The resulting mean and standard deviation for these distributions can be found in table A.1, where the top-left values represents the distribution of the first edge/pulse while the bottom-right the last edge. By summing 16 Gaussians with the mean and standard deviation from the table, the total probability density (PDF) was recreated, see the solid red line in figure A.8.

Table A.1: Mean (a) and standard devation (b) of the extracted period in pico seconds. Based on 4539 edges.

(a) Mean, ps.

355	324	341	353
332	356	327	361
342	328	352	370
334	348	336	346

(b) Standard deviation, ps.

1.9	1.6	1.7	1.9
1.6	2.0	1.5	1.9
1.8	1.7	1.8	2.0
1.7	2.0	1.9	2.4

The standard deviation can be seen to be about the same differing only by 0.8 ps, while the mean varies from 324 ps to 370 ps. To divide the jitter into "random" and "deterministic", a single average standard deviation of 1.8 ps, was combined with each of the mean values, this model results in the red dashed line in figure A.8. The agreement in estimate PDFs allows us to postulate that the random jitter has the same root cause, i.e. this is not 16 different phenomenas. The random jitter is believed to originate from amplitude noise (thermal noise) and $1/f$ noise. The exact distribution of the deterministic components is still under investigation, in particular how they change from chip to chip and how they change when the supply and substrate voltages are adjusted.

A.5 System simulation

Based on measured performance and jitter a high level simulation was carried out to asses the complete radar system performance. Following the diagram in figure A.3(b), a linear single bit chirp was generated, 4 copies where delayed to simulate 4 targets and the input threshold was changed in discrete steps and the beat spectrum averaged (16 times).

To properly model the waveform generator, we re-sample the signal into a bitstream which is allowed to change value only every 172 ps, effectively sampling it at 2.9 GHz. This is done without any filtering, so higher harmonics gets folded down into the spectrum, creating unwanted coherent time quantization "noise" in the spectrum. The measured deterministic and random jitter was then brought into the simulation and a CW test showed the same pattern noise as the measurements, although with a higher "noise floor"; due to the larger number of harmonics being folded down when the output is perfectly square.

A full sweep from 250 MHz to 2.9 GHz in 10 µs was then simulated and the resulting beat spectrum is shown in figure A.9. The figure compares the case of only amplitude quantization (top) and added time quantization with jitter (bottom). The noise level can be reduced by increasing the integration time (chirp length). The rms quantization noise has gone from 55 dB to 37 dB, meaning the time quantization degraded the dynamic range by 55 dB−37 dB = 18 dB. Adding the measured jitter, no degradation in SNR level was observed, but for the full sweep the targets seems to have moved slightly (see figure A.9 bottom).

By lowering the bandwidth, we loose resolution but obtain a cleaner spectrum, a similar analysis shows that with a sweep from 25 MHz to 290 MHz,

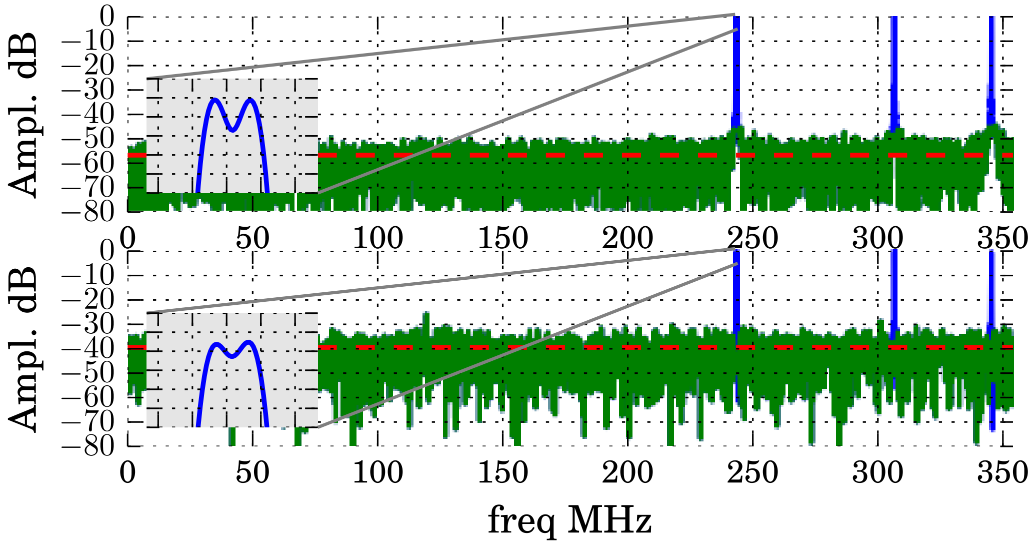


Figure A.9: Simulated FMCW beat spectrum (with a Hanning window), sweep from 250 MHz to 2.9 GHz in 10 µs, a delay is used to move the beat spectrum to [177 MHz, 354 MHz], inset shows two targets separated by only 124 mm being (barely) resolved by a 2 dB dip. Expected signals are in blue, while the red dotted line denotes the rms quantization noise level. Top: amplitude quantized, bottom: amplitude and time quantized with measured jitter.

the achieved dynamic range goes from 37 dB to 43 dB, a 5.5 dB improvement. Adding the jitter now slightly improves the dynamic range, by 2.2 dB (not shown).

Despite the transmitted waveform being far from a clean sine-chirp, the simple harmonic removal, matched filter and sweept threshold, provide a promising radar solution.

A.6 Conclusion

A flexible single bit waveform generator is presented for UWB applications. With measurements and simulations the feasibility of an FMCW single bit radar is analyzed and discussed. Simulations show that the timing jitter is insignificant in comparision to the tradeoff done in amplitude and time quantization. Future work should therefore focus on improving the time resolution over the timing accuracy. The flexibility in waveform enables bandwidth to be traded off with dynamic range, where a wide bandwidth allows fine resolution and oversampled waveform increases the dynamic range.

Acknowledgment

The authors would like to thank Kristian Gjertsen Kjelgård for assistance with measurements and insightful discussions.

Bibliography

[1] G. Roberts, “Reducing the analog-digital productivity gap using time-mode signal processing,” in *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*, June 2014, pp. 782–785.

[2] L. Xiu, “The concept of time-average-frequency and mathematical analysis of flying-adder frequency synthesis architecture,” *Circuits and Systems Magazine, IEEE*, vol. 8, no. 3, pp. 27–51, Third 2008.

[3] J. A. Johnston, “Generating analogue FM pulses using a 1 bit digital technique,” *Communications, Radar and Signal Processing, IEE Proceedings F*, vol. 131, no. 4, pp. 349–356, July 1984.

[4] R. J. Durrant and S. Parkes, “The implementation of high time bandwidth chirp pulses using digital techniques,” in *High Time-Bandwidth Product Waveforms in Radar and Sonar, IEE Colloquium on*, May 1991.

[5] M. Min, T. Paavle, P. Annus, and R. Land, “Rectangular wave excitation in wideband bioimpedance spectroscopy,” in *Medical Measurements and Applications, IEEE International Workshop on*, May 2009, pp. 268–271.

[6] H. Hjortland and T. Lande, “CTBV Integrated Impulse Radio Design for Biomedical Applications,” *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 3, no. 2, pp. 79–88, April 2009.

[7] Y. Tsividis, “Mixed-domain systems and signal processing based on input decomposition,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 53, no. 10, pp. 2145–2156, Oct 2006.

[8] S.-J. Lee, K. Kim, H. Kim, N. Cho, and H.-J. Yoo, “Adaptive network-on-chip with wave-front train serialization scheme,” in *VLSI Circuits, 2005. Digest of Technical Papers. 2005 Symposium on*, June 2005, pp. 104–107.

[9] N. H. Weste and D. M. Harris, *Integrated circuit design*, 4th ed., M. Hirsch, Ed. Pearson, 2011.

[10] E. Jones, T. Oliphant, P. Peterson *et al.*, “SciPy: Open source scientific tools for Python,” 2001–. [Online]. Available: http://www.scipy.org/

Appendix B

Paper II. Square wave architectures for radar-on-chip

The following is a re-formatted version of
Øystein Bjørndal, Svein-Erik Hamran, and Tor Sverre Lande. Square wave architectures for radar-on-chip. In *2016 46th European Microwave Conference (EuMC)*, pages 1485–1488, 2016. doi:10.1109/EuMC.2016.7824636

PhD Candidate contribution: Wrote the majority of the paper, implemented and measured the presented solution.

Supervisors contributions: Wrote most of the introduction. Contributed to the original system concept, both the radar/signal-processing and the implementation.

Abstract

Square waves enable radar implementation in modern digital technology giving power-efficent, compact and flexible radar solutions. Using a simple square wave generator and a sweep threshold quantizer we can realize a PN-sequence radar as well as a CW, SFCW, FMCW or a pulsed radar. The square wave receiver can utilize a digital XOR gate as a mixer and we present post layout simulations confirming our high level modelling.

B.1 Introduction

Advantages of modern technology are features like miniaturization, low-power consumption, substantial computational power and integration of complicated processing systems on a small piece of silicon (system-on-chip). For modern radar systems, the full advantage of modern technology has been difficult to utilize in spite of faster devices and higher computational speed. As indicated in [1] modern radar systems are still fairly large modules with substantial size and power consumption.

Over the last century, several viable radar architectures are explored. Trade-offs between usage, available technology and computational power; as well as regulations have changed over time and different radar architectures has emerged. Pulsed ranging radar, Frequency-Modulated Continuous-Wave (FMCW) radar, Stepped-Frequency Continuous-Wave (SFCW) radar, noise-radar systems and even Continuous-Wave (CW) movement radar sensors are just a few examples of architectures in use for different applications.

Some integrated radar systems are reported in the literature. The first single-chip CMOS radar was reported in 2006 by Hjortland et al. [2] featuring a pulsed radar system. Later an integrated SFCW radar in 65 nm CMOS by Caruso [3] and a M-sequence radar was reported by Sachs [4] using SiGe BiCMOS technology. Furthermore, challenging partial radar systems are emerging like the 77GHz FMCW radar front-end reported in [5] and a K-band front-end reported in [6] intended for automotive applications. Several short-range radar systems are reported [7, 8] indicating a number of potential sensing applications provided a compact, low-power and high-performance radar is available; preferably in low-cost standard technology.

In this work we are proposing a generic, programmable system-on-chip radar adaptable to different radar architectures. The classical ranging radar architecture is shown in figure B.1. Here a FMCW coded signal path is indicated, but other architectures like SFCW, CW as well as noise radar will be analyzed with respect to square wave signal coding.

By changing or sweeping the threshold voltage during measurements, different quantization levels are scanned and as such recreate any incoming signal (limited by the resolution of the threshold voltage). Results of each sweep is averaged coherently, increasing the SNR while building up the return. This combination of 1-bit sampling, averaging and threshold sweeping is called “swept-threshold” sampling [9].

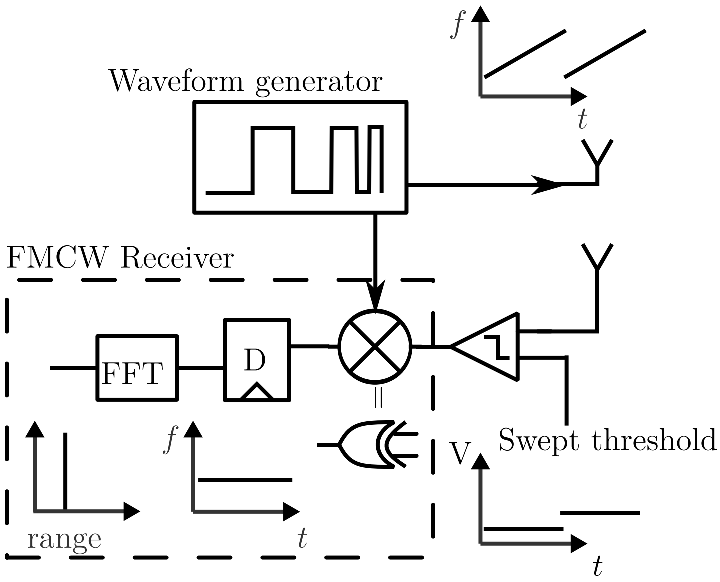


Figure B.1: Principle of a square wave generator utilized in a digital FMCW radar with a swept threshold receiver. Harmonics not shown, two up-sweeps depicted with different threshold levels for each sweep.

As indicated in the receiver part of figure B.1, the mixer or multiplier required in analog FMCW radars may be substituted by a simple and compact XOR gate! In fact, large numbers of XOR gates may be combined with delays for high-speed running cross-correlation as will be seen in section B.5. Not only is the digital XOR gate fast (and getting faster with new technologies); enabling high speed operations, it is also free from any lower frequency filtering and constant current sources, enabling power efficient radars operated at a user defined Bandwidth (BW).

For the frequency modulated radar architectures, the frequency or phase of the mixer output is of interest (called the beat frequency in an FMCW radar) and the harmonics inherent in a square wave signal must be carefully considered. In the following we will show how these techniques may be explored in a generic DDS-like radar architecture implementing a variety of well know radar systems. The FMCW radar is explored in section B.2, where techniques for removing, and even utilizing the harmonics are shown, a CW radar with a SFCW processor is then briefly covered in section B.4 before we show a correlation based radar in section B.5.

B.2 Dealing with harmonics

In [10], we briefly showed how a delay on the receiver would enable us to separate the ambiguity of harmonics and signals. This is achieved by increasing the apparent two-way travel time of the target and noting that the higher harmonics will move further than the fundamental. We will in this work, show how the harmonics can be utilized constructively when pushing them out of band and a correlation technique that suppresses them in-band.

It is equivalent to insert the delay either on the receiver, as was shown in [10], or to generate two equal, but delayed, sequences; as long as the two way travel time is increased relative to the mixer input. The delay can be implemented either as a series of digitally selectable inverters or two identical generators can be used with a delayed start signal.

B.2.1 Moving out of band

The required delay to separate the fundamental from the harmonics is simply the two way travel time of the furthest target of interest $2r_{\text{un}}/c$. This places the closest target (the direct return) at $f_0 = \alpha\tau_{\text{un}}$, with its second harmonic at the unambiguous range.

To show the viability of this approach a set of high level simulations where set up, with a digital chirp, from 600 MHz to 2.67 GHz, in 22.3 µs and repeated 20 times with linearly varying thresholds for each sweep. For illustration, 5 targets with equal amplitude is distributed between 0 m and the maximum unambiguous range $r_{\text{un}} = 100\,\text{m}$ (assuming air). The targets will then appear at the beat frequencies $f_i = \alpha\tau_i$, where $\alpha = (2.67\,\text{GHz} - 600\,\text{MHz})/22.3\,\mu\text{s} = 9 \times 10^{13}\,\text{Hz/s}$ and τ_i is the two way travel time. For simplicity, the beat spectrum is not low-pass filtered and re-sampled.

The resulting beat spectrum is displayed in figure B.2. In the first simulation, without delay, the harmonics from the closest targets overlap with the returns from the targets further away. By adding a delay of $2r_{\text{un}}/c$ we can separate the fundamental from the 3rd harmonics as shown in the middle panel. By increasing the delay further, we can also separate the 3rd harmonics from the 5th harmonics as shown in the bottom panel.

As an alternative to simply discarding the harmonics, we could utilize that for a sweep with bandwidth $BW = f_o - f_l$, the harmonics sweeps out $3f_o - 3f_l = 3BW$, giving us 3 times the resolution. This is illustrated in

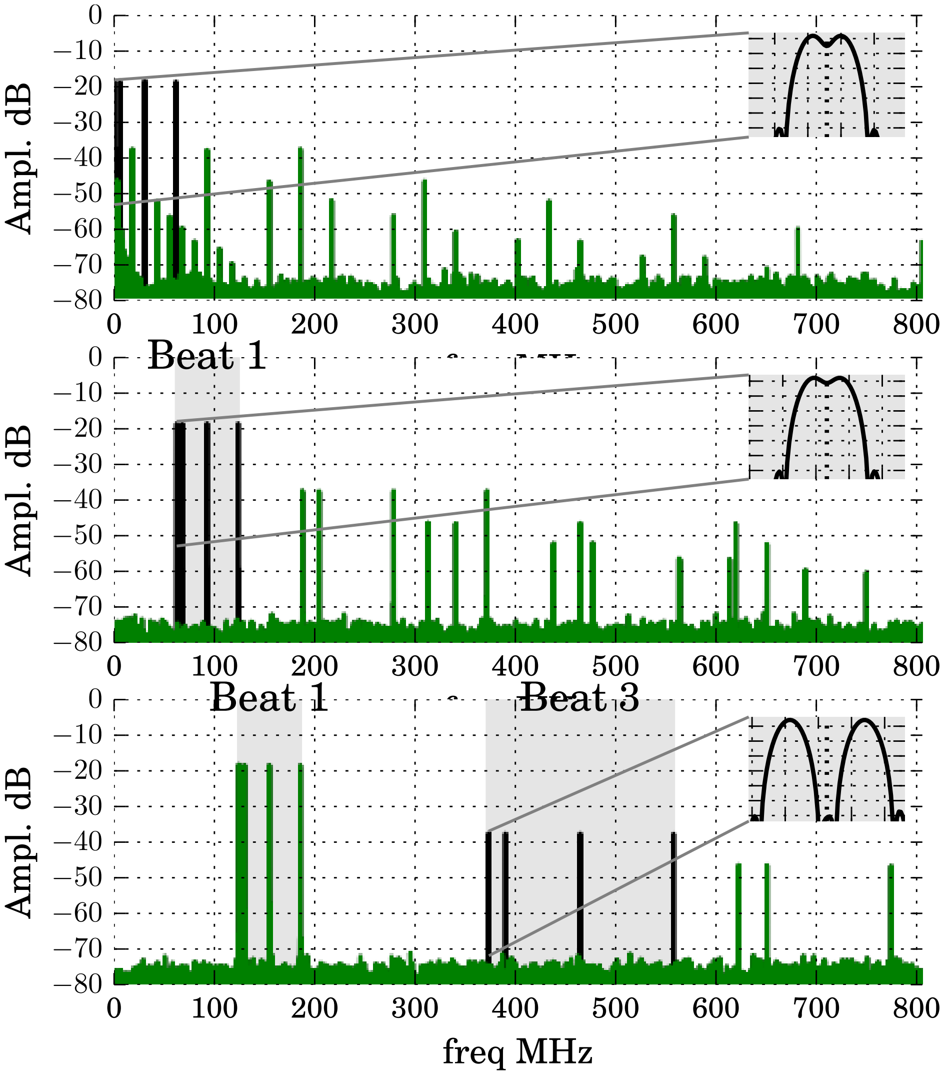


Figure B.2: By increasing the apparent range to the targets, the harmonics gets separated into bands. The band of interest is highlighted, while the expected targets are colored black. Top: no delay. Middle: delay is used to push the third harmonics away from the fundamental, giving an unambiguous beat spectrum. Bottom: the delay is increased to push the fifth harmonic away from the third harmonics, allowing us to utilize the third harmonics spectrum. Inset shows two targets separated by 243 mm.

the lower part of figure B.2, where strong scatters are easy to distinguish. Once the bands are separated, more advanced processing may be applied to combine the responces from each band.

B.2.2 Resolving in band ambiguity

As an alternative to pushing the harmonic out of band, we can use the frequency shift of the spectrum moving both the fundamental and the harmonics by an equal amount, while, as utilized above, a delay will move the fundamental and harmonics by different amounts. Assuming the non-delayed response is

$$[f_i, 3f_i, 5f_i, \ldots]$$

then a frequency shift of f_{shift} moves it to

$$[f_i + f_{\text{shift}}, 3f_i + f_{\text{shift}}, 5f_i + f_{\text{shift}}, \ldots]$$

while a delay of τ results in

$$[f_i + \alpha\tau, 3(f_i + \alpha\tau), 5(f_i + \alpha\tau), \ldots]$$

We can utilize this to see which peaks move by f_{shift} and which peaks move by multiples of $3, 5, \ldots$. Although this may give us a visual way to identify harmonics and signals, a less manual way is to correlate the two spectrums. To show this, a similar high level simulation as in section B.2.1 is set up, but with 3 targets, where one target is placed 40 dB below the close targets. To detect the weaker return, 128 thresholds are averaged.

To maintain the amplitude, the correlation is calculated as

$$X_{\text{corr}} = \sqrt{X_{f_{\text{shift}}} X_{\tau}}.$$

The result is shown in figure B.3 where an arbitrary delay of 42 ns is used. We see that the fundamentals overlap and retain their amplitude, while the harmonics are reduced by the noise level. It is clear that the weaker return can not be distinguished from the harmonics of the stronger returns, without doing the correlation.

For denser radar scenes with more targets, we may need to do the correlation multiple times with different delays. The correlation technique can

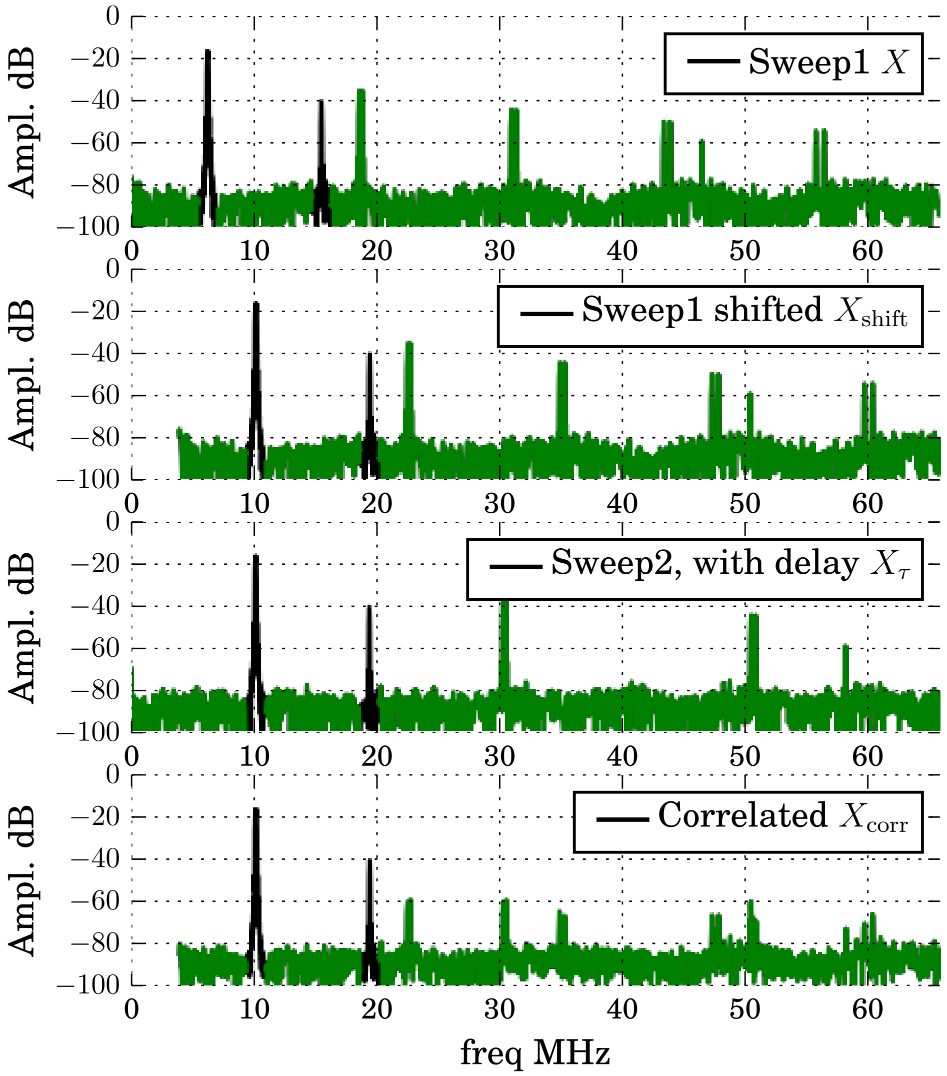


Figure B.3: We start with a non delayed sweep (top) which we shift by $f_{\text{shift}} = 3.9\,\text{MHz}$. The sweep is then repeated, but now with a delay of $\tau = f_{\text{shift}}/\alpha = 42\,\text{ns}$. By correlating (point wise multiplication of the frequency responce), the shifted and delayed spectrums we obtain the bottom spectrum.

also be combined with the previous technique to push either all or at least some of stronger harmonics out of band and correlating away the remaining in-band harmonics. Instead of the computationally expensive correlation, one could average multiple measurements with different delays, making sure to compensate for the expected shift.

It is important to realize, that we can utilize the harmonics even in the case where the channel attenuates the harmonics. The only requirement for the harmonics to "exists", is that the mixer is digital (has a digital input and outputs). We will show this in the next section, with a post layout simulation of an XOR gate with sinusoidal input.

B.3 Post layout XOR simulation

We have shown theoretically that a digital XOR gate will behave as a mixer, making it suitable as a matched filter in an FMCW radar. We will here briefly present a post-layout simulation, realized in a low power 90 nm CMOS process.

A CMOS XOR gate with inverters/buffers on the input and output is shown in figure B.4, in addition, a simple model for the supply pads are included. When we feed in sinusoidal inputs, the inverters have more than sufficent gain to drive the output to saturation, giving us a square digital signal. These square waves will have harmonics, which when mixed, produces not only the fundamental beat frequencies, but multiples (mostly) at the 3rd. and 5th. beat frequencies. For a perfect 50 % high/low waveform we expect to only see odd harmonics, which is what we have observed in the idealized system simulations. Due to slight differences in the rise and fall delay, we also see even harmonics when using real devices.

B.4 Stepped Frequency Continuous Wave (SFCW)

Instead of a continuously frequency modulated output, we can employ the same transmitter as before, but programmed for a series of CW (clipped sine wave) outputs. A single frequency does not give any significant range resolution, but by combining the output of multiple CW measurements we have a SFCW radar. The architecture is then a CW receiver combined with some additonal processing of the return.

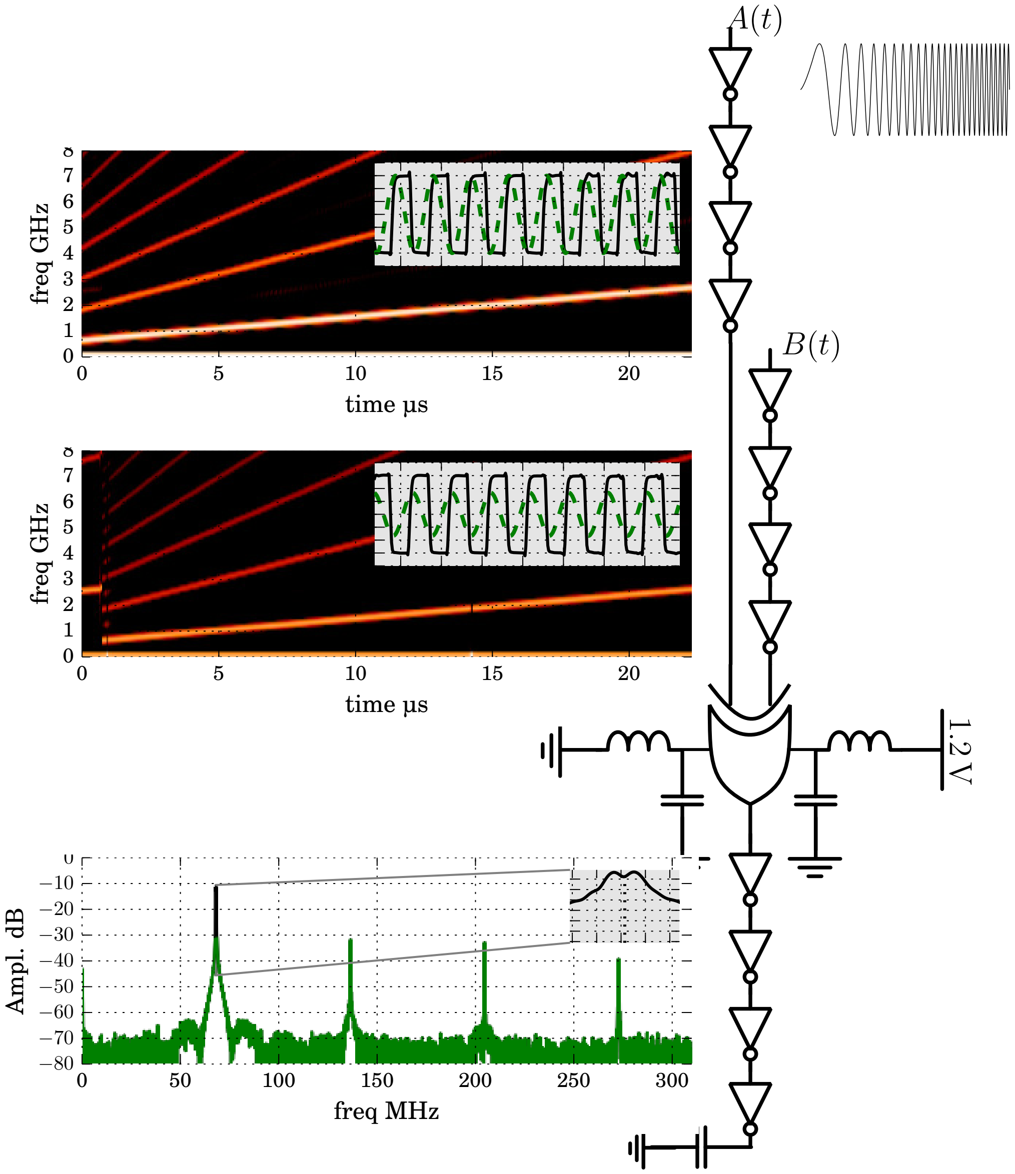


Figure B.4: Post layout simulation setup and results for a chirped sinusoidal input $A(t)$, and a delayed sinusoidal responce from 2 targets $B(t) = A(t - \tau_1)/2 + A(t - \tau_2)/2$. The insets of the two top panels, shows the sinusoidal inputs in time, where the dashed curve is the sinusoidal input and the whole black line is after the 4 inverters. Bottom plot is a Fourier transform of the buffered XOR output, showing that we can distingush the two targets with a single sweep.

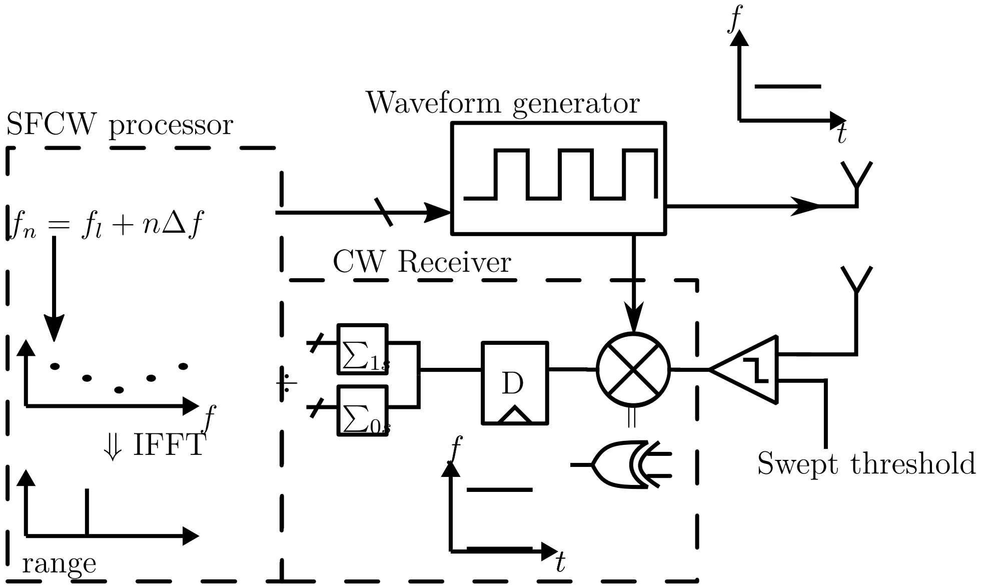


Figure B.5: A fully digital CW radar is controlled by a SFCW processor. For each frequency f_n, the CW radar outputs the mean (DC) value, which is arranged and transformed with an IFFT yielding the range spectrum.

To obtain the DC output after the mixer, the traditional method is a low pass filter (to get rid of the mixer sum) followed by a high resolution ADC. In figure B.5 we propose to take advantage of the mixer sum, the mixer output will be a PWM waveform, where the percentage of low values to high values give the DC level.

By stepping from 600 MHz to 2.67 GHz in 1381 steps of length 5 µs with a 1 MHz offset between transmitted and mixer input, we obtain figure B.6.

It is interesting to study how the square wave radar compares to an idealized sine wave radar. In addition to amplitude clipping, the waveform generator will realistically output a *sampled* digital signal. This is explored in figure B.6, where the top panel shows the ideal single 'dirac' response (with a Hanning window). When amplitude clipping to two levels, we get harmonics, which can be dealt with in the same manner as the FMCW case. At the bottom we include the effect of time quantizing the transmitted waveform.

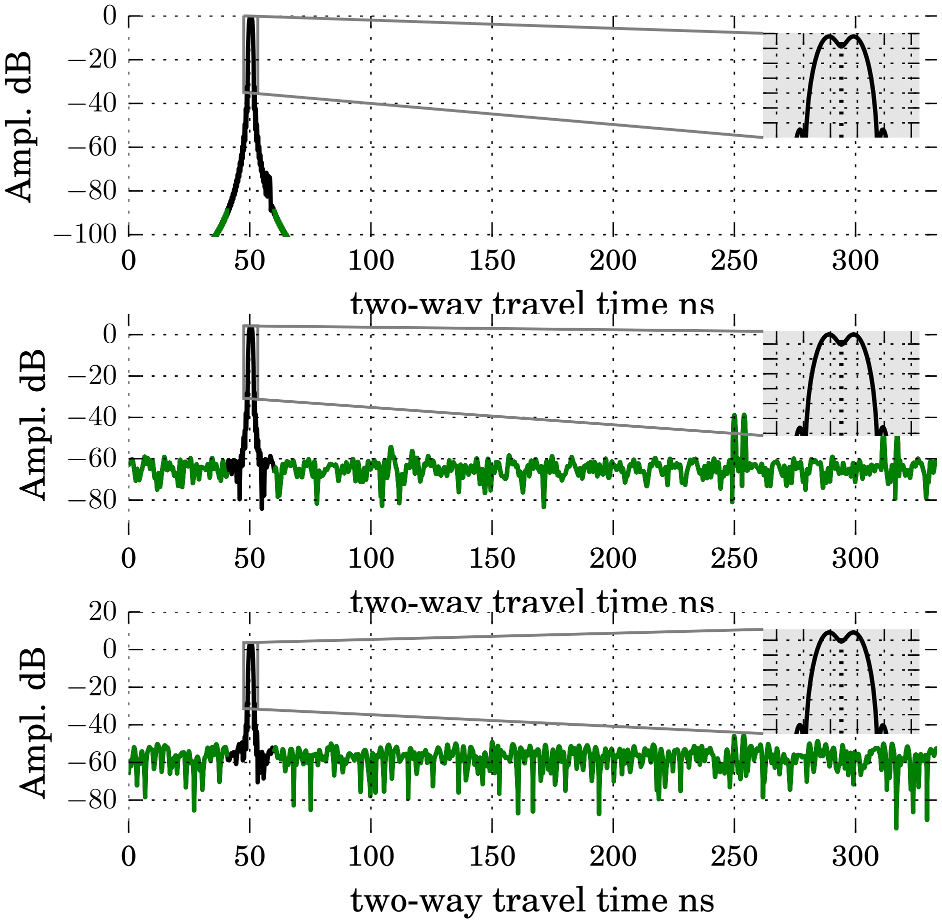


Figure B.6: Top: ideal (sinusoidal) SFCW radar range responce, middle: amplitude quantized (square wave), bottom: added finite time resolution. The expected return is colored black and the inset shows a zoomed in view around the expected return, with two targets separated and the −31 dB sidelobe level of the Hanning window barely visible.

B.5 Correlation based radar

As an alternative to the mixer based radars we have discussed, we can replace the matched filter operation of the mixer with a more general correlation circuit. In a correlation based radar, we can utilize *any* waveform, where random (or pseudo-random) are most common. The correlation circuit can be in continous time, as shown by the correlating circuit in [9], or sampled.

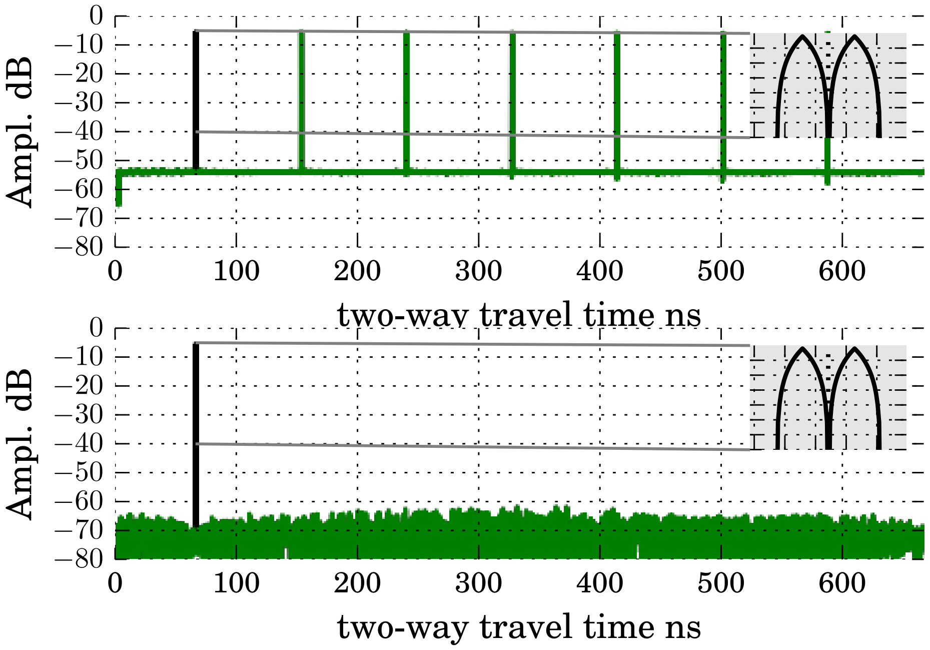


Figure B.7: Simulated output of an m-sequence radar for 2 close targets. Top: $m = 9$, bottom: $m = 12$, expected return colored black.

For brevity we will only touch upon M-sequences, which are particulary simple to generate with a single Linear Feedback Shift Register (LFSR) and have nice periodic autocorrelation properties. A system simulation similar to those presented earlier in the paper is shown in figure B.7 for an m-sequence generator of length 9 and 12. The sequences are read out continuously at a rate of 1/(2·170 ps)=2.9 GHz. The receiver is no longer a single XOR gate (mixer), but a full single bit correlation. In a PN sequence radar, the bandwidth (and hence the resolution) is determined by the sequence *rate* while the range ambiguity by the sequence *length*, which is why the resolution is identical for the two presented cases, but the shorter $2^9 - 1$ sequence has multiple ambiguous responces.

B.6 Conclusion

By exploiting the flexibility of a square wave transmitter, we have shown the feasibility of several classical radar architectures for single bit CMOS realization. The most obvious limitation of a square wave radar, is the harmonics, where we have proposed several ways not only to remove them, but also utilize them constructively. A digital CMOS XOR gate is shown with post layout simulations to perform the required mixer operation.

Bibliography

[1] J. Browne. (2014, July) Looking further with modern radar systems. [Online]. Available: http://defenseelectronicsmag.com/systems-subsystems/looking-further-modern-radar-systems

[2] H. Hjortland, D. Wisland, T. Lande, C. Limbodal, and K. Meisal, “CMOS impulse radar,” in *Norchip Conference, 2006. 24th*, Nov 2006.

[3] M. Caruso, M. Bassi, A. Bevilacqua, and A. Neviani, “A 2-to-16GHz 204mW 3mm-resolution stepped-frequency radar for breast-cancer diagnostic imaging in 65nm CMOS,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*, Feb 2013.

[4] J. Sachs, *Handbook of Ultra-wideband Short-range Sensing: Theory, Sensors, Applications*, 1st ed. Weinheim: Wiley-VCH Verlag & co., 2012.

[5] T. Mitomo, N. Ono, H. Hoshino, Y. Yoshihara, O. Watanabe, and I. Seto, “A 77 GHz 90 nm CMOS transceiver for FMCW radar applications,” *Solid-State Circuits, IEEE Journal of*, vol. 45, no. 4, April 2010.

[6] K. Kjelgård and T. Lande, “A CMOS UWB k-band impulse radar receiver,” in *Microwave Conference (EuMC), 2015 European*, Sept 2015.

[7] Q. Lv, Y. Dong, Y. Sun, C. Li, and L. Ran, "Detection of bio-signals from body movement based on high-dynamic-range doppler radar sensor (Invited)," in *RF and Wireless Technologies for Biomedical and Healthcare Applications (IMWS-BIO), 2015 IEEE MTT-S 2015 International Microwave Workshop Series on*, Sept 2015.

[8] D. Zito, D. Pepe, B. Neri, D. De Rossi, and A. Lanata, "Wearable System-on-a-Chip pulse radar sensors for the health care: System overview," in *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, vol. 2, May 2007.

[9] H. Hjortland and T. Lande, "CTBV integrated impulse radio design for biomedical applications," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 3, no. 2, April 2009.

[10] Ø. Bjørndal, S.-E. Hamran, and T. S. Lande, "UWB waveform generator for digital CMOS radar," in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, May 2015.

Appendix C

Paper III. Bitstream radar waveforms for generic single-chip-radar

The following is a re-formatted version of Øystein Bjørndal, Svein-Erik Hamran, and Tor Sverre Lande. Bitstream radar waveforms for generic single-chip-radar. *International Journal of Microwave and Wireless Technologies*, 9(6):1325–1337, August 2017. doi:10.1017/S1759078717000782

PhD Candidate contribution: Wrote the majority of the paper, implemented and measured the presented solution.

Supervisors contributions: Contributed to the original system concept, both the radar/signal-processing and the implementation.

Abstract

Bitstreams, square wave digital signals, enables flexible radar implementations in modern digital technology. By using bitstreams in place of analog sinusoidal waveforms, we can realize continuous wave (CW), stepped-frequency CW, frequency-modulated CW or even pseudo random noise-sequence and pulsed radars, all with a single bit of amplitude resolution. The building blocks are a programmable waveform generator, a sweep threshold quantizer, digital delay and a digital XOR gate as a mixer. This gives us a novel, almost fully digital (requiring only a comparator) system, as previously proposed and which is extended here. The flexibility of the transmitter allows for easy switching between waveforms and the bitstream signal can be processed with single bit digital gates. Single bit signals allows for exploration of novel continuous time non-clocked digital implementations to maximize speed and energy efficiency. Mixing frequencies with a digital XOR gate creates harmonics, which are explored for multiple solutions utilizing digital delay. Analytical as well as simulation results are presented. Initial measurements from a 90 nm CMOS chip is provided for the transmitter and the full system, proving the feasibility of a digital future in radar.

Keywords: Radar architecture and systems, Si-based devices and IC technologies

Corresponding author: Øystein Bjørndal; email: oystebjo@ifi.uio.no

C.1 Introduction

Modern digital technology bring miniaturization, low-power consumption, substantial computational power and integration of complex processing on a small piece of silicon (system-on-chip). For modern radar systems, the full advantage of modern technology has been difficult to utilize in spite of faster devices and higher computational speed. As indicated in [1] modern radar systems are still fairly large modules with substantial size and power consumption.

The basic principles of radar have been known for more than a century, over the years several radar architectures have been explored; each with its own tradeoff in application and hardware complexity. Radar architectures differ mainly by the utilized waveform, example architectures contains, Continuous-Wave (CW) radar for velocity determination, pulsed ranging radar, Frequency-Modulated Continuous-Wave (FMCW) radar, Stepped-Frequency Continuous-Wave (SFCW) radar and noise-radar. The frequency modulated architectures needs a frequency source and a mixer, but requires only a modest sampling rate of the mixer difference. The pulsed and noise radar are more amendable to digital implementation, but requires a receiver that samples the entire transmitted bandwidth.

Several integrated radar systems have appeared in the literature, some with commercial interest. The first single-chip CMOS radar was published by Hjortland et al. [2] in 2006, featuring a pulsed radar system now commercially available from Novelda [3]. Later an integrated SFCW radar was reported in 65 nm CMOS by Caruso [4] for breast cancer detection. Sachs [5] has written extensively on a M-sequence radar using SiGe BiCMOS technology. A growing application is automotive radars in the 60 GHz/70 GHz bands, where multiple single chip solutions have been presented and some are commercially available [6], CMOS realizations in the literature include [7, 8, 9, 10]. Several short-range radar systems are reported [11, 12, 13] indicating a number of potential sensing applications provided a compact, low-power and high-performance radar is available; preferably in low-cost standard technology.

Figure C.1 shows the proposed generic, programmable system-on-chip radar with a FMCW receiver. Other architectures like CW, SFCW as well as noise radar will be explained later in the paper, by simply replacing the dashed receiver circuit.

The proposed solution uses a programmable waveform generator and a swept threshold receiver, the frequency modulated architectures (CW, SFCW and FMCW) uses a single XOR gate as a mixer and the CW/SFCW system uses counters to obtain the desired averaged DC output.

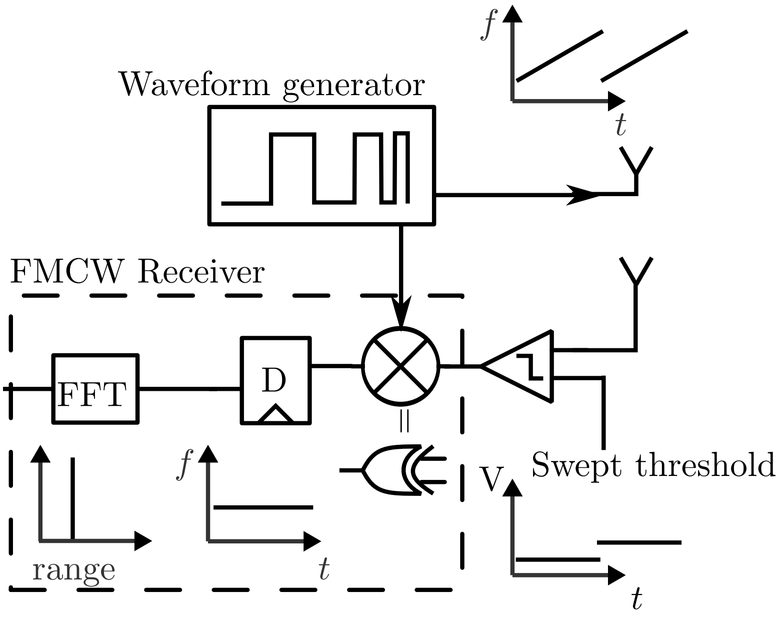


Figure C.1: Principle of a bitstream waveform generator utilized in a digital FMCW radar with a swept threshold receiver. The range spectrum is obtained after sampling, averaging and a frequency transform. Harmonics not shown, two up-sweeps depicted with different threshold levels for each sweep. [14]

The receiver amplifies the backscattered signal and quantizes the signal to a bitstream by comparing it with a changeable threshold voltage. This operation is non-linear, but by repeating the measurement with different threshold voltages, we re-create the incoming signal (limited to the resolution of the threshold voltage). As we repeat the measurement, noise will be uncorrelated and the signal can be averaged coherently (assuming the scene is stationary during the measurement period). We therefore increase the Signal to Noise Ratio (SNR), while recreating the received signal. The concept is called “swept-threshold” sampling [15] and combines single bit amplitude clipping, averaging and threshold sweeping. The comparator is the only analog component, having two continuous time analog voltages as inputs and a continuous time square wave digital output (bitstream).

In principle, the waveform generator and the swept threshold receiver could implement all of the proposed architectures, assuming we could sample the entire bitstream comparator output and de-modulate the signal in software. Although this sampling is made easier by the fact that we only need to sample a single bit in amplitude resolution, this is still a non-optimal implementation. Part of the novelty in this paper, is that we instead do the processing while the signal is still a bitstream; replacing the conventional analog mixer, with a simple and compact XOR gate! This allows us to take advantage of the matched filter property of a frequency modulated radar and lowers the required sample rate (or increases the oversampling rate). Bitstream processing also enables a high speed running cross-correlation circuit for a pseudo noise-radar, by simply combining multiple XOR gates with delays, as will be seen in section C.5.

The frequency modulated architectures extracts the phase or frequency difference by mixing the transmitted and received signal (giving the beat frequency in an FMCW radar). Mixer output is then Fourier transformed, yielding harmonics due to the square wave nature of both inputs, these harmonics must be carefully considered. We analytically derived the harmonics in section C.2 and show ways to remove or even utilize the harmonic in section C.3. We then briefly cover a CW radar with a SFCW processor in section C.4, before we show a correlation based radar in section C.5 and an unmodulated pulsed radar in section C.6. The chip and measurement setup is reviewed in section C.7, with a post layout simulation and a full FMCW transceiver measurement.

Preliminary simulations and handling of harmonics was first published in [14], this is an extended paper with analytical treatment and measurements from a prototype chip implementation.

C.2 Analytical mixing of frequency modulated waveforms

We will here outline the theoretical frequency behavior when using a digital XOR gate as a mixer. This theoretical foundation is then used in a digital FMCW radar in section C.2.1 and a digital CW/SFCW radar in section C.2.2. Starting with the behavior of an analog mixer with inputs $X(t)$ and $Y(t)$ performs the simple multiplication

$$M(t) = X(t) \cdot Y(t).$$

For two sinusoidal inputs, with arbitrary phases $\phi_X(t)$ and $\phi_Y(t)$, the mixer output can be expressed as

$$M(t) = A_X \cos(\phi_X(t)) \cdot A_Y \cos(\phi_Y(t)),$$

which can be re-expressed by applying Euler's formula

$$\begin{aligned} M(t) &= \frac{A_X A_Y}{2} \left[\cos\left(\phi_X(t) + \phi_Y(t)\right) + \cos\left(\phi_X(t) - \phi_Y(t)\right)\right] \\ &\equiv \frac{A_X A_Y}{2} \left[\cos\left(\phi^+(t)\right) + \cos\left(\phi^-(t)\right)\right] \end{aligned}$$

leaving us with the sum and difference between the phases of X and Y. Of particular interest in a frequency modulated radar and for brevity in the derivations below, we will often look at the differentiation only, namely the frequency

$$f(t) \equiv \frac{1}{2\pi} \frac{\partial}{\partial t} \phi(t).$$

A square wave can be expressed as an infinite sum of odd harmonics, so a square wave mixer (an XOR gate), will be doing

$$\begin{aligned} M(t) = &\left(\sum_{n=1,3,5,\ldots} A_{Xn} \cos(n\phi_X(t)) \right) \cdot \\ &\left(\sum_{n=1,3,5,\ldots} A_{Yn} \cos(n\phi_Y(t)) \right) \end{aligned} \tag{C.1}$$

As the signals are rail to rail digital signals, we can; for notational convenience, assume $A_{Xn} = A_{Yn} = A_n$. Some algebra will yield the following phase/frequency terms

$$\begin{aligned} \phi_{ij}^{\pm} &= i \cdot \phi_X(t) \pm j \cdot \phi_Y(t) \\ f_{ij}^{\pm} &= i \cdot f_X(t) \pm j \cdot f_Y(t) \end{aligned} \tag{C.2}$$

for $i, j \in [1,\ 3,\ 5,\ \dots]$ with amplitudes A_iA_j.

C.2.1 Analytical FMCW

In an FMCW radar, the phase and frequency takes the form

$$\phi_{\text{FMCW}}(t) = 2\pi \left(f_l t + \frac{f_o - f_l}{2T_m} t^2 \right) \tag{C.3}$$

$$\begin{aligned} f_{\text{FMCW}}(t) &= f_l + \frac{f_o - f_l}{T_m} t \\ &\equiv f_l + \alpha t \end{aligned} \tag{C.4}$$

where the frequency goes from f_l to f_o in T_m seconds, and α is the chirp rate. For a single return at the two-way travel time τ_k, the mixer product becomes

$$M_{\tau_k}(t) = A_X \cos(\phi_X(t)) \cdot A_Y \cos(\phi_X(t - \tau_k))$$

leading to the sum and difference frequencies

$$\begin{aligned} f_{11}^{+} &= \frac{1}{2\pi} \frac{\partial}{\partial t} \left(\phi_X(t) + \phi_X(t - \tau_k) \right) \\ &= (f_l + \alpha t) + (f_l + \alpha(t - \tau_k)) \\ &= 2\,(f_l + \alpha t) - \alpha \tau_k \end{aligned} \tag{C.5}$$

$$\begin{aligned} f_{11}^{-} &= \frac{1}{2\pi} \frac{\partial}{\partial t} \left(\phi_X(t) - \phi_X(t - \tau_k) \right) \\ &= \alpha \tau_k \end{aligned} \tag{C.6}$$

the beat sum, f_{11}^{+}, can be filtered out as long as $2\,(f_l + \alpha t) \gg \alpha \tau_k$, which is normally assured by having a slow sweep time T_m, leaving us with the desired beat frequency $\alpha \tau_k$ which is easily identified by a Fourier transform.

In a square wave FMCW radar, the mixer will, as seen in (C.1), not only mix the fundamental frequency terms, but also mix the harmonics. Writing out (C.2) when we include the third harmonic, we obtain

$$\begin{aligned}
&f_{11}^{-} = \alpha\tau_k \quad f_{11}^{+} = 2\alpha t - \alpha\tau_k + 2f_l \\
&f_{33}^{-} = 3\alpha\tau_k \quad f_{33}^{+} = 6\alpha t - 3\alpha\tau_k + 6f_l \\
&f_{13}^{-} = -2\alpha t + 3\alpha\tau_k - 2f_l \quad f_{13}^{+} = 4\alpha t - 3\alpha\tau_k + 4f_l \\
&f_{31}^{-} = 2\alpha t + \alpha\tau_k + 2f_l \quad f_{31}^{+} = 4\alpha t - \alpha\tau_k + 4f_l
\end{aligned} \tag{C.7}$$

Even when including higher order harmonics, the resulting frequencies are, as seen above, linear in time t. The third harmonic from each input mixes together and creates multiples of the beat difference (and beat sum) $f_{33}^{\pm} = 3f_{11}^{\pm}$, while the cross-terms results in various integer slopes.

These slopes can be seen in the left column of figure C.2, where we have plotted the entire mixer output spectogram as the first row, a zoomed view in the middle row and a Fourier transform of the beat spectrum in the bottom row. In addition, we have included the effect of a sampled transmitter, where the down-aliasing is visible as a folding around half the transmitters clock rate.

The depicted analytical model use terms up to the 30th harmonic, where we only draw up to the 9th in the spectogram view. The spectogram views uses the polynomials in (C.7) but aliased above the transmitters Nyquist rate of 5.2 GHz. To create the Fourier view the analytical model first sums up the phase terms

$$\begin{aligned}
y_{\text{analytical}}[nT] &= \sum_{i \in [1,\,3,\,5,\,\ldots]} A_i \left[\sum_{j \in [1,\,3,\,5,\,7]} A_j \cos\left(\phi_{ij}^{\pm}[nT]\right) \right] \\
\phi_{ij}^{\pm}[nT] &= i \cdot \phi_{\text{FMCW}}[nT] \pm j \cdot \phi_{\text{FMCW}}[nT - \tau_i]
\end{aligned}$$

where ϕ_{FMCW} is found in eq. (C.3), $T = 97\,\text{ps}$, $T_m = 25.3\,\mu\text{s}$ and $n \in [0,\, 1,\, \ldots,\, T_m/T]$. The waveform is then upsampled and zero padded before a Hanning windowed Fourier transform.

A square wave mixer has a large number of intermixing products. In addition to handle harmonics, care must be taken when selecting the sweep parameters f_l, f_o and T_m to minimize the interference of the mixer cross-terms. As the mixer cross-terms are not constant in time, their interference does not cause false targets, but rather raises the noise floor.

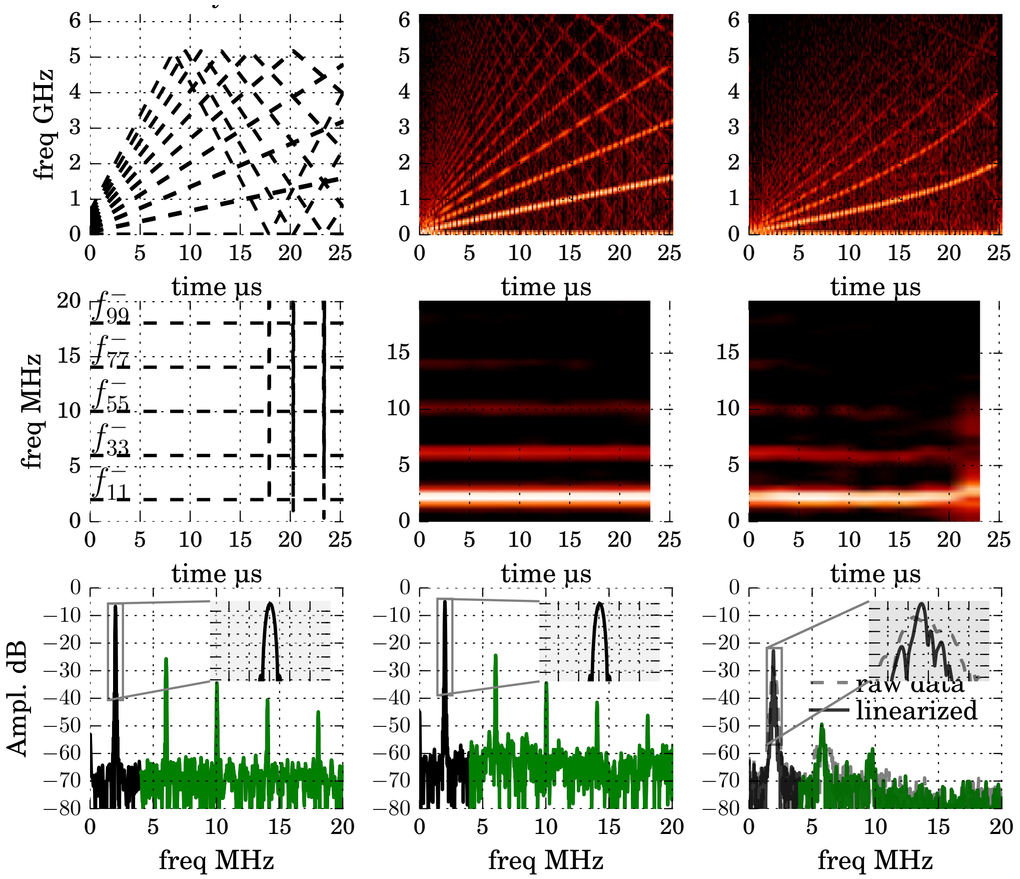


Figure C.2: Analytical, simulated and measured beat spectrum for a square wave FMCW sweep from 38.1 MHz to 799 MHz in 25.3 µs. Top: Spectogram of the entire mixer spectrum, middle row: Spectogram of the mixer difference (beat spectrum), with a (Hanning windowed) FFT at the bottom. From left to right: Analytical model, python simulation and measured results on the right. The transmitter is "clocked" at 1/97 ps=10.4 GHz, hence the visible aliasing around 5.2 GHz. The measurements are here only the transmitter, where the mixing and delay is done entirely in software. The beat spectrum is linearized with the technique presented in section C.7.

C.2.2 Analytical CW/SFCW

Transmitting a single frequency f_c, a square wave mixer sees the inputs

$$\phi_X = 2\pi f_c t \tag{C.8}$$
$$\phi_Y = 2\pi f_c(t - \tau_k). \tag{C.9}$$

We notice that a frequency domain investigation is insufficient, as we simply get

$$f_{ii}^{-} = f_{jj}^{-} = 0 \quad \text{for } i, j \in [1,\, 3,\, 5,\, \dots] \tag{C.10}$$

i.e., the fundamental and all of the harmonic differences will mix down to DC. Looking instead at the phase, we obtain

$$\begin{aligned} &\phi_{11}^{-} = 2\pi f_c \tau_k \quad \phi_{11}^{+} = 2\pi f_c \left(2t - \tau_k\right) \\ &\phi_{33}^{-} = 6\pi f_c \tau_k \quad \phi_{33}^{+} = 6\pi f_c \left(2t - \tau_k\right) \\ &\phi_{13}^{-} = 2\pi f_c \left(-2t + 3\tau_k\right) \quad \phi_{13}^{+} = 2\pi f_c \left(4t - 3\tau_k\right) \\ &\phi_{31}^{-} = 2\pi f_c \left(2t + \tau_k\right) \quad \phi_{31}^{+} = 2\pi f_c \left(4t - \tau_k\right) \end{aligned} \tag{C.11}$$

The "classical" result is here the phase ϕ_{11}^{-}, which gives a (ambiguous) range estimate at

$$\phi_{11}^{-} = 2\pi \left(f_c \tau_k + n\right) \quad \text{for } n \in \pm[0,\, 1,\, 2,\, 3, \dots] \tag{C.12}$$

which can be solved for the two-way travel time of target k

$$\tau_k = \frac{\phi_{11}^{-}}{2\pi f_c} - \frac{n}{f_c} \tag{C.13}$$

we notice that the ambiguity of phase wrapping creates false targets every $1/f_c$. To improve this, we need to measure with multiple frequencies f_c, say N frequencies spaced Δf_c apart, each frequency being transmitted for Δt_c seconds. We now have a SFCW radar, where the change in measured phase gives the range information [16]

$$f_{\text{SFCW}} = \frac{1}{2\pi} \frac{\partial \phi_{11}^{-}(f_c)}{\partial t} \tag{C.14}$$
$$= \frac{\partial}{\partial t} \left(f_c \tau_k a + n\right) \tag{C.15}$$

$$= \tau_k \frac{\partial}{\partial t} f_c \tag{C.16}$$

which is just the discrete equation

$$= \tau_k \frac{\Delta f_c}{\Delta t_c} \tag{C.17}$$

In a similar manner to the FMCW radar, we now have a direct link between a frequency and the two-way travel time of the target. This is however inherently sampled, following [16], the ambiguity of Δf_c discrete frequency steps creates aliasing around $\tau = 1/(2\Delta f_c)$. A tight number of steps is therefore required to avoid far off returns appearing as aliased responses.

There is no strict requirement for I/Q sampling, as the phase at each step $\phi_{11}^{-}(f_c)$ is never explicitly needed in the processing, we simply use the real valued samples for each SFCW step to recreate the environments transfer function. We therefore achieve the same final result as an I/Q SFCW radar by sampling with twice the number of frequencies. From (C.10) we notice that the output of interest lies at DC, so for each SFCW step, we simply store the mean mixer output value. The f_{SFCW} is then found be a Fourier transform, alternatively, we can get τ_k directly from an Inverse Fourier transform.

The aliasing does however represent a challenge for a square wave SFCW radar. In the FMCW case the ambiguous/additional responses at $i\alpha\tau_k$ (for $i \in [3, 5, \ldots])$ can be filtered by an anti-aliasing filter before sampling the beat frequency. In the SFCW case, the undesired phase terms $i \cdot 2\pi f_c \tau_k$ will create the sampled frequencies $i\tau_k \Delta f_c/\Delta t_c$ which will fold into signal band if left un-managed.

The solution is identical to a superheterodyne receiver, where instead of using a single frequency, we transmit one frequency and mix with a frequency that is offset by $f_c + f_{\text{offset}}$. The mixer difference will produce the offset frequency, and the harmonics mix to create $[3f_{\text{offset}}, 5f_{\text{offset}}, \ldots]$ which can be filtered out.

We will return to a square wave SFCW radar in section C.4, but first, we must deal with the harmonics.

C.3 Dealing with harmonics

As we have seen; a frequency modulated radar, will after a Fourier transform, have ambiguous returns at integer multiples of the true target. Although these ambiguous returns are lower in amplitude, they will appear indistinguishable from weaker true targets. A method to circumvent the ambiguity was first presented in [17], where we briefly presented how a delay on the receiver separated the harmonics from the signals. This was then extended in [14], where we not only showed how to move the targets out of band, but also how to utilize the harmonics constructively and an alternative technique that suppresses the harmonics in-band. In this paper, we have refined the simulation setup and the correlation technique.

The advantage of using a bitstream as the waveform, is that implementing a programmable time delay is realistic. This enables an array of new radar signal processing opportunities, since we can now freely adjust the range response, making targets appear closer or further away depending on which path is delayed. In a square wave frequency modulated radar this functionality is useful, since it gives multiple methods for separate the harmonics from the fundamental.

There are multiple ways to implement the programmable delay. In continuous time, we can use a series of digitally selectable inverters. In a synchronous implementation, we can duplicate the generator and program in a time-offset between them. One generator then feeds the mixer, while the other is used for transmission. A delay inserted in the on-chip path between the generator and mixer will decrease the apparent two-way travel time of the response, while a delay in the channel path will make the environment response appear further away. For the channel path, which is the programmable delay we will use here, the delay can either be inserted on the receiver (after the threshold, as depicted in [17]) or, equivalently, on the transmitter; as both placements increases the apparent two way travel time. We will denote this adjustable delay as τ_{tx}.

C.3.1 Moving out of band

The first method we will present require a delay equal (or greater) than the desired unambiguous range r_{un}. A delay of $\tau_{\text{tx}} = \tau_{\text{un}} = 2r_{\text{un}}/c$ will move the fundamental of the direct return to $\alpha\tau_{\text{un}}$ with its second harmonic now moved to $2\alpha\tau_{\text{un}}$. This creates a unambiguous band for the fundamental, allowing us to ignore (filter away) the harmonics.

To illustrate this technique, we simulate a square wave FMCW radar (as indicated in figure C.1) and obtain the beat spectrums in figure C.3. For simplicity we here view everything as continuous-time and care is taken in the simulation to minimize aliasing when modeling the square waves in a discrete simulation. We do not include the typical low-pass filtering and sampling of the beat spectrum.

In figure C.3, we simulate 5 targets distributed between 0 meters and $r_{\text{un}} = 100$ meters using increasing delay settings in the channel path. Without delay, the close in targets will have harmonics interwoven with targets further out. As we increase the delay we can not only get an unambiguous frequency range for the fundamental (middle panel), but we can also get an unambiguous range for the individual harmonics.

Remember that the 3rd harmonic components are created by the mixing the 3rd harmonics of the two mixer inputs, this means that a sweep with bandwidth $BW = f_o - f_l$ has 3rd harmonics sweeping $3f_o - 3f_l = 3BW$. This gives us 3 times the bandwidth, and hence 3 times the resolution! We show this resolution in the bottom panel of figure C.3, where two targets separated by 136 mm are barely separated when we use the fundamental, but is clearly separated when looking at the 3rd harmonic in the beat spectrum.

The only requirements for utilizing the harmonics is that a) the mixer is digital and b) that we can look at the harmonics without ambiguity. We will show the first requirement in section C.7 C.7.2, where we simulate a digital XOR gate with sinusoidal inputs, but we will first show a second way of getting a unambiguous spectrum.

C.3.2 Resolving in band ambiguity

If the desired unambiguous range is large, the above approach may necessitates an impractically long delay. As an alternative approach we have therefore proposed a second solution. The idea is that instead of moving the harmonics all the way out of the band, we can make repeated measurements with different delays, in essence giving us staggering or jittering in the beat spectrum. If we do a frequency shift of the spectrum, both the fundamental and harmonics will shift by the same frequency, while a delay will cause the fundamental to shift by a different amount than the harmonics. Assuming

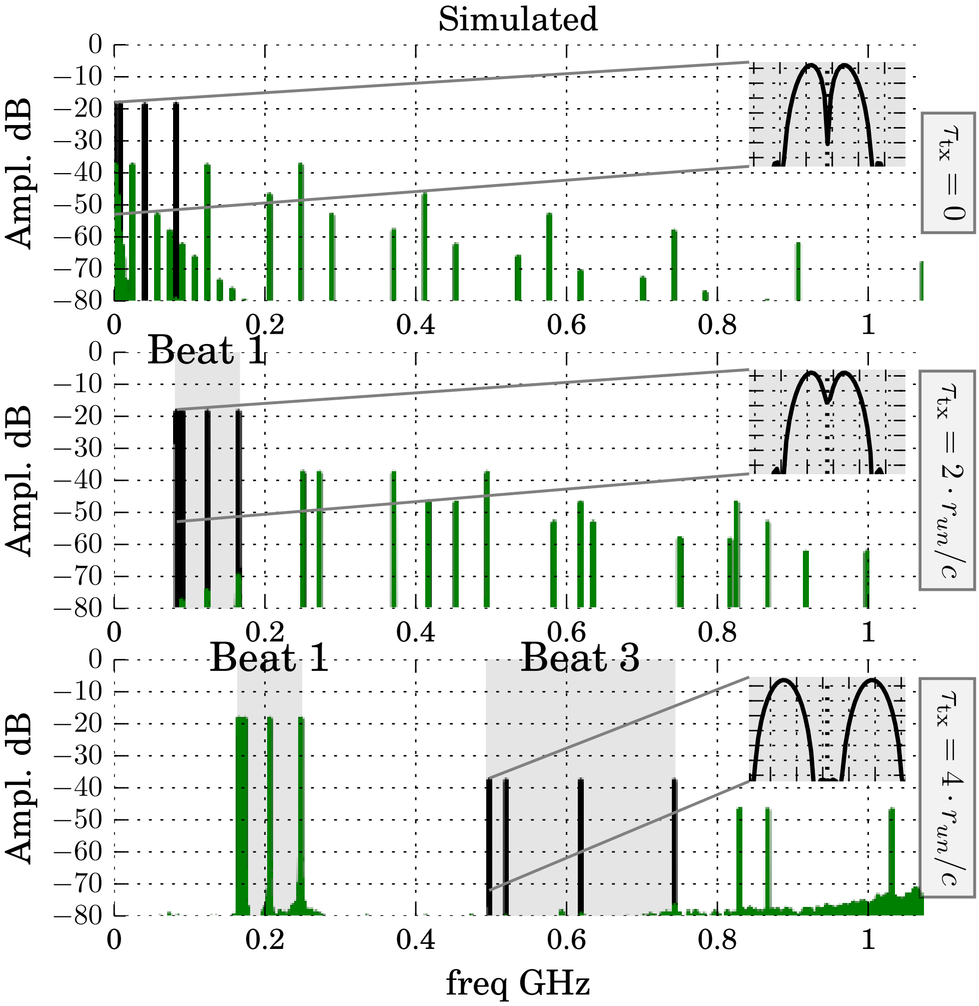


Figure C.3: Simulated scenario to illustrate how a delay in the channel path can separate the fundamental from the harmonics. A square wave chirp, where the fundamental goes from 600 MHz to 2.67 GHz in 16.7 µs is repeated 20 times with a linearly varying thresholds for each chirp. The first panel uses no delay and shows 5 equal amplitude targets interwoven with harmonics. By increasing the delay, by $2 \cdot 100\,\mathrm{m}/c = 667\,\mathrm{ns}$, the next panel (middle) shows the harmonics moved out of the highlighted fundamental band. At the bottom we increase the delay even further to separate the third and fifth harmonic, allowing us to utilize the higher resolution of the third harmonics. Originally proposed in [14].

the non-delayed response is

$$F = [f_i,\, 3f_i,\, 5f_i,\, \ldots]$$

a new measurement with a delay τ_{tx} on the receiver results in

$$F_{\tau_{\mathrm{tx}}} = [f_i + \alpha\tau_{\mathrm{tx}},\, 3(f_i + \alpha\tau_{\mathrm{tx}}),\, 5(f_i + \alpha\tau_{\mathrm{tx}}),\, \ldots]$$

Shifting this by back by $f_{\mathrm{shift}} = \alpha\tau_{\mathrm{tx}}$ results in

$$\begin{aligned} F_{f_{\mathrm{shift}}} &= F_{\tau_{\mathrm{tx}}} - f_{\mathrm{shift}} \\ &= [f_i,\, 3(f_i + \alpha\tau_{\mathrm{tx}}) - \alpha\tau_{\mathrm{tx}},\, 5(f_i + \alpha\tau_{\mathrm{tx}}) - \alpha\tau_{\mathrm{tx}},\, \ldots] \end{aligned}$$

We notice that the fundamentals of F and $F_{f_{\mathrm{shift}}}$ lines up at f_i, while the harmonics do not. The shift is simplest to achieve after a Fourier transform, as a re-arrangement of indexes.

By measuring with different delay settings of τ_{tx} and compensating for the expected shift the range profiles can be averaged. This effectively dithers away the undesired harmonic responses, while improving the SNR.

As an alternative to averaging, we can in some cases, reduced clutter level with fewer measurements by correlating the shifted and delayed spectrum. To maintain the voltage unit, the correlation is calculated as

$$X_{\mathrm{corr}} = \sqrt{|X_{f_{\mathrm{shift}}} X_{\tau_{\mathrm{tx}}}|}.$$

Where, without the square root, voltage units would be transformed to power units.

An example simulation is shown in figure C.4. As in section C.3.1 each sweep covers a bandwidth of $2.67\,\mathrm{GHz} - 600\,\mathrm{MHz} = 2\,\mathrm{GHz}$ in $16.7\,\mu\mathrm{s}$, but is here repeated 128 times to detect the weaker return placed 40 dB below the two stronger returns. In addition, Additive white Gaussian noise (AWGN) is added to the channel, with the same rms amplitude as the signal (giving an input SNR of 0 dB). The time bandwidth product of the chirp and the coherent averaging of the sweep threshold receiver gives a theoretical SNR improvement of $2\,\mathrm{GHz} \cdot 16.7\,\mu\mathrm{s} \cdot 128 = 66\,\mathrm{dB}$, which is consistent with the here simulated SNR of 70 dB.

We see in figure C.4, that the fundamentals lines up while the harmonics are reduced (limited by the noise floor). It is clear that the weak signal, is

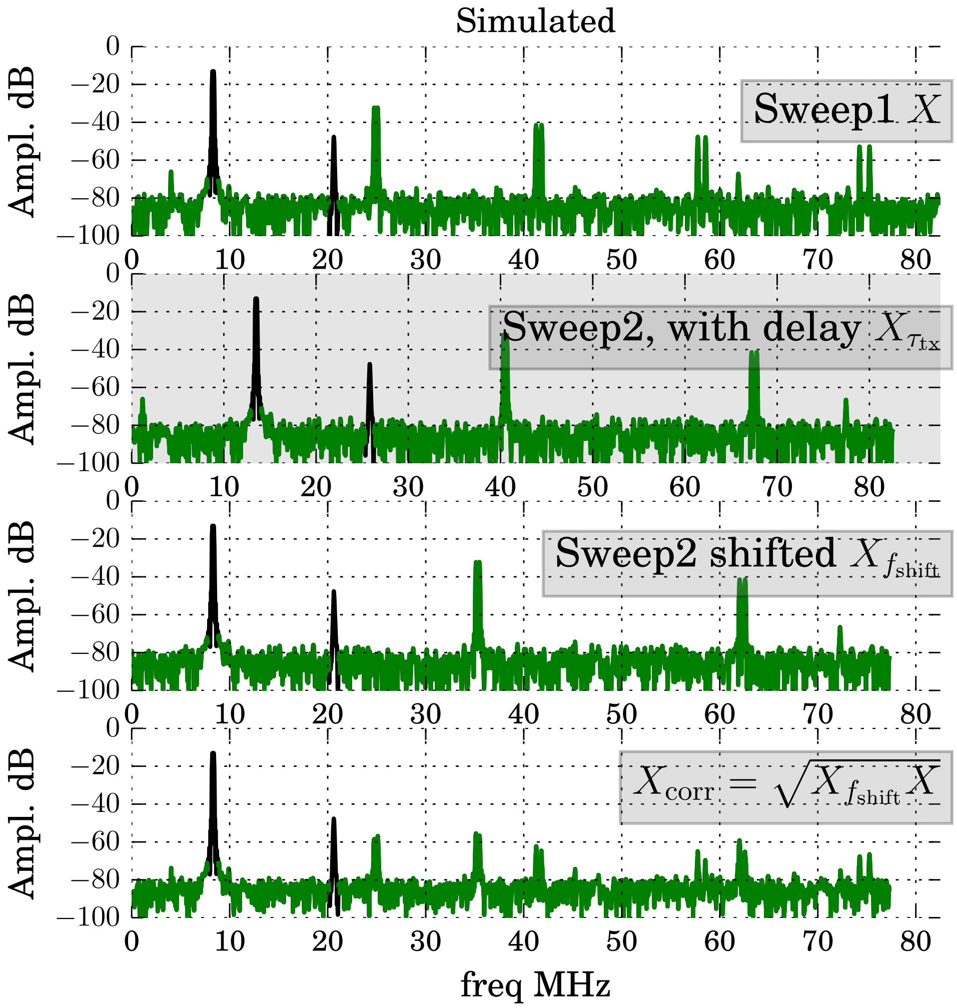


Figure C.4: Simulated scenario to illustrate the correlation technique to suppress harmonics in-band. 3 targets are simulated, two closely spaced targets and one weaker return. AWGN noise is added to the channel with the same rms amplitude as the signal (input SNR of 0 dB). The top panel shows a sweep where the radar is programmed without any delay. In the next panel, the sweep is repeated with a delay of $\tau_{\text{tx}} = 42\,\text{ns}$ in the channel path, this is then shifted back by $f_{\text{shift}} = \alpha\tau_{\text{tx}} = 5.2\,\text{MHz}$. The bottom panel shows the point wise multiplication (correlation) of the original and shifted spectra. Similar to [14].

ambiguous with the harmonics of the stronger scatters, without the correlation. In this example, we line up the fundamental, but the technique can also be used to line up the harmonics, giving the same improved in resolution as was demonstrated in the previous section.

The correlation technique relies on the harmonics lining up with “empty” parts of the range spectrum and is therefore best suited for “sparse” scenes with few returns, or at least scenes with some empty regions. The two presented techniques can be combined to alleviate this, by moving at least some of the harmonics up and correlating or averaging away the rest. An optimal choice of delay settings and harmonic suppression technique will depend on the image scene, which, due to the ease of digital integration; can conceivably be optimized automatically.

C.4 Stepped Frequency Continuous Wave (SFCW)

As was explained in the analytical section (section C.2 C.2.2), a SFCW radar is implemented as a CW radar that steps the output frequency. The proposed architecture to achive this is shown in figure C.5, consisting of a CW transmitter and receiver with some control logic.

Traditionally a CW receiver low pass filters the mixer output and digitizes the remaining DC value with a high resolution ADC. To avoid this analog voltage level, and the filter, we can make the observation that the mixer output will toggle like a Pulse-Density Modulation (PDM) signal. The desired DC level can then be found by measuring (counting) the ratio of high to low values. This is trivial to implement as two counters, one that increments if the signal is high and another that increments when the signal is low. The sampling clock does not need to be synchronous, we are simply interested in the average value, so a local free running ring oscillator can serve as the clock source.

An example simulation is shown in figure C.6, where we step the CW frequency 1381 times from 600 MHz with a 1.5 MHz frequency step. A number of cases is illustrated, as a reference an idealized sinusoidal simulation is shown in the top panel, giving a single peak for each simulated target. The next panel shows what happens if we do not deal with the harmonics, due to the sampled nature of a SFCW radar the harmonics will fold around $1/(2 \cdot 1.5\,\mathrm{MHz}) = 334\,\mathrm{ns}$. By taking this folding into account we can annotate

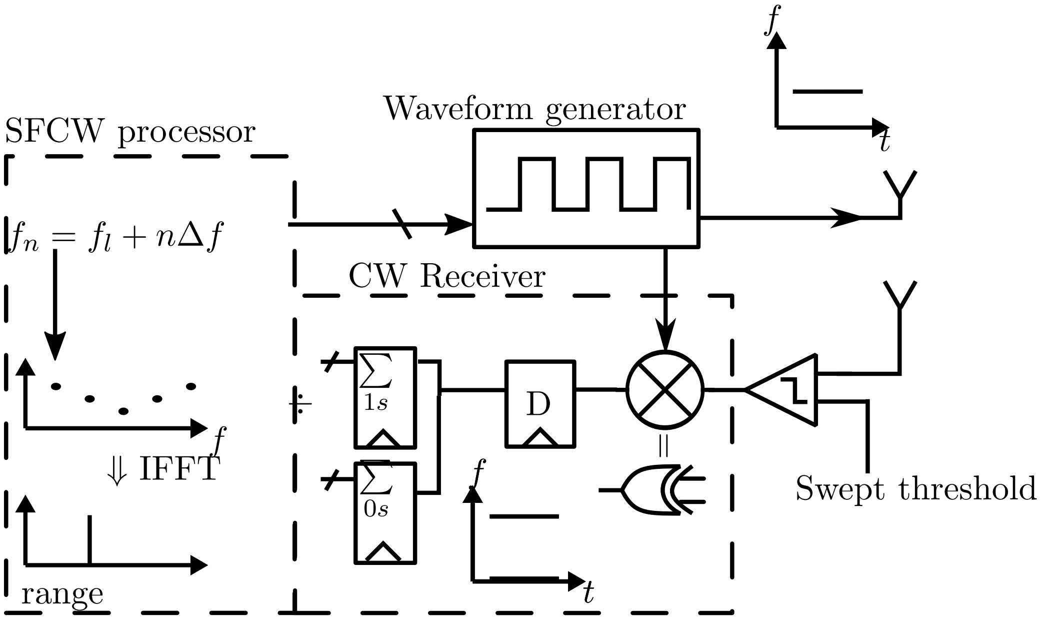


Figure C.5: Principle of a bitstream CW radar controlled by a SFCW processor. For each frequency f_n, the CW radar outputs the mean (DC) value, which is arranged and transformed with an inverse fast Fourier transform (IFFT) yielding the range spectrum.

the peaks by using the fundamental peak as a reference, peaks up to the 19th harmonic is visually recognizable.

As was mentioned in the anlytical section, the theoretical solution is to add a frequency offset between the transmitted and mixed signal, the low-pass characteristic of taking the average then attenuates the higher order harmonics as shown in the next panel. An alternative to attenuating the harmonics is to jitter them away, this could be added explicitly by modifying the bitstream, but we first consider some real world deviations from a perfect square wave.

The first such non-ideal behavior is a finite time resolution, the bitstream consists of discrete bits which is read out with a finite clock rate. In addition, the transmitter will have jitter, we here include the measured jitter level from [17] and also added some AWGN noise to the channel. These effects create the bottom panel of figure C.6. We note that these real world effects jitters away the higher harmonic peaks, without requiring a frequency offset.

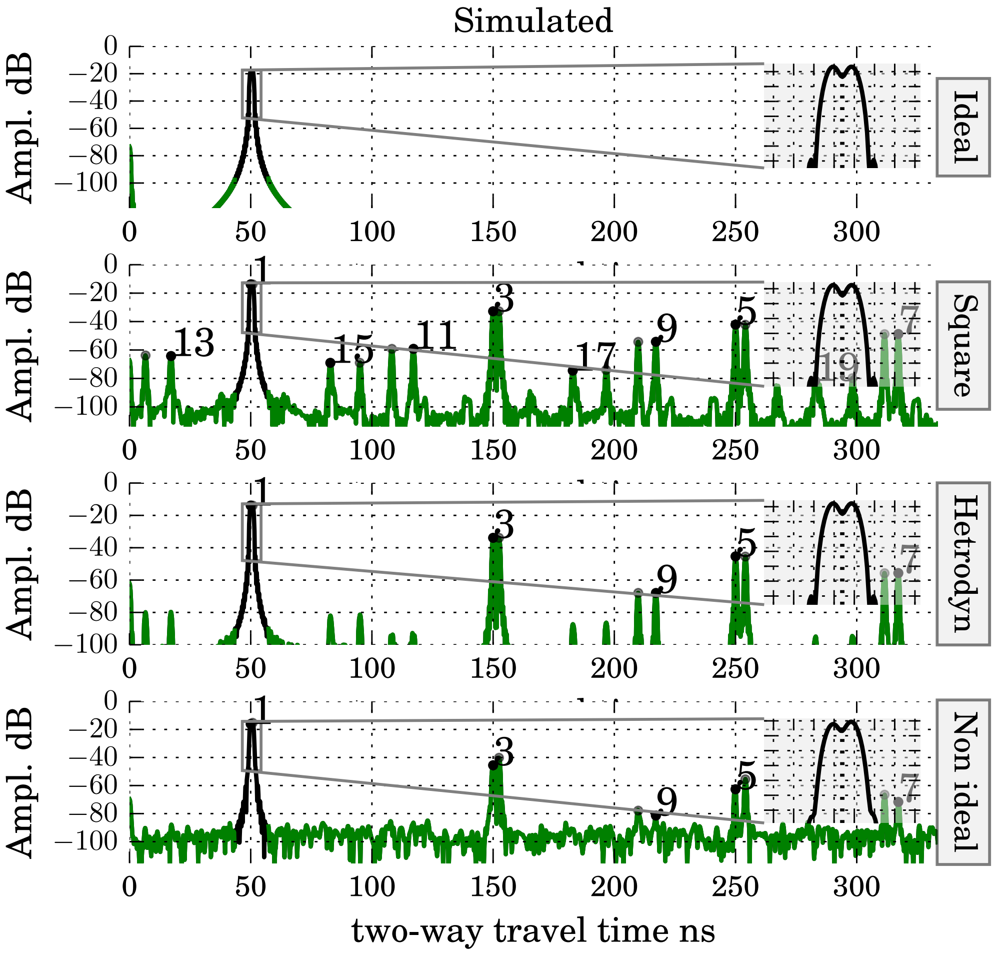


Figure C.6: Simulated SFCW radar with 2 close targets, the radar is stepped from 600 MHz to 2.67 GHz in 1381 steps of length 53 µs (divided into 16 different threshold settings). The top panel shows an idealized (sinusoidal) simulation, with an inset showing the two simulated targets separated by 121 mm and the −31 dB sidelobe level of the Hanning window barely visible. In the next panel, a square wave SFCW radar is simulated and harmonics peaks up to the 19th harmonic is annotated. The harmonics are attenuated by adding a frequency offset of 30 kHz between the two mixer inputs and using averaging as a simple to implement filter. In the bottom panel, we do not use a frequency offset, but simulate the transmitter with a finite time resolution of 64 ps, jitter levels measured in [17] and a input SNR of 0 dB. This effectivly dithers away the higher order harmonic peaks.

C.5 Correlation based radar

We have up to now discussed radar architectures that are frequency modulated. These all rely on a single mixer to down-convert the received signal and a Fourier transform of the captured data to obtain a range spectrum. A more general architecture is proposed in figure C.7, where we have replaced the mixer with a correlation circuit. In a correlation based radar, we can utilize *any* waveform, where "noise-radars" (usually pseudo noise) are most common. The correlation circuit can be in continuous time, as shown by the correlating circuit in [15], or discrete.

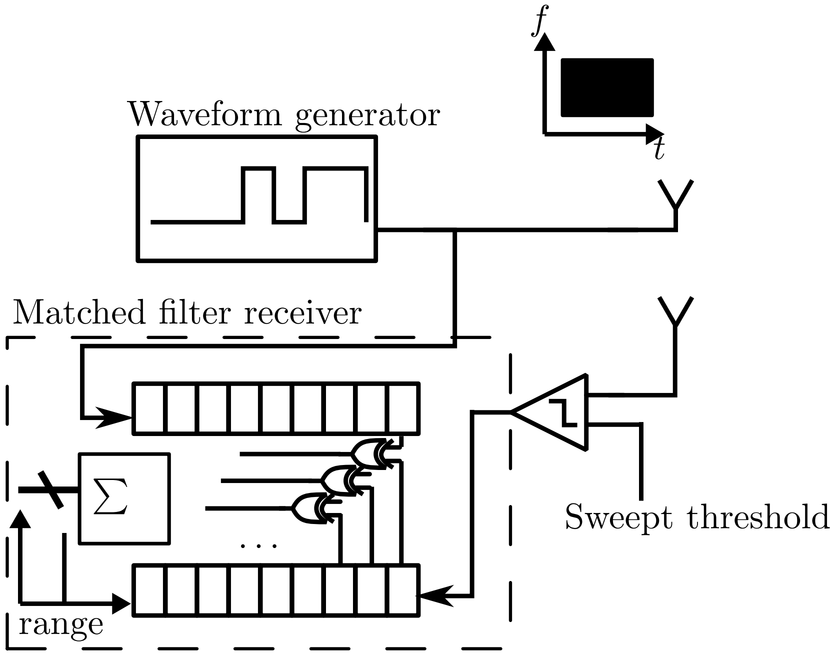


Figure C.7: Principle of a bitstream based radar that does a full correlation between the transmitted and received signal. The correlation circuit can either be sampled; by using a chain of D-flip-flops, or continuous time; by using inverters as delay elements.

A particularly nice set of pseudo random noise (PN) sequences are coined Maximum length sequences (M-sequences) and can be generated with a single Linear Feedback Shift Register (LFSR). Sachs [5] gives a thorough overview on M-sequence radar. The repetition of a pseudo random sequence gives a range ambiguity given by the sequence *length* while the bandwidth (and hence the resolution) is proportional to the sequence *rate*.

In addition, an M-sequence has “perfect” autocorrelation, with a peak of 1 and the value $-1/N$ for all other delays. This gives us the dynamic range for a single target scenario. For a more realistic scenario, with multiple returns, the floor $-1/N$ ads up, decreasing the dynamic range for each return.

Figure C.8 shows a simulated and measured m-sequence after correlation. Two sequences are compared, a shorter $2^9 - 1 = 511$ sequence and a $2^{12} - 1$ sequence. The sequences are repeated continously and read out at a clock rate of $1/(2 \cdot 88.4\,\text{ps}) = 5.7\,\text{GHz}$. The first sequence repeates after 90 ns while the other after 724 ns

C.6 Unmodulated pulsed radar

The previously presented radar types can easily be used in pulsed mode (or interrupt mode), where the transmitter is occasionally turned off. Shutting down the transmitter is traditionally done to avoid saturating the receiver, long range radars can transmits long “pulses” of a modulated waveform and then shut the transmitter off before capturing the backscattered data. The modulation can be any of the mentioned radar architectures, either frequency modulated or even a pseudo random sequence.

In this section we will have a brief look at a simpler unmodulated radar, where no pulse compression is performed. An unmodulated radar is in principle simpler to implement, we simply transmit a single pulse and get the range response by synchronously capturing the received signal. We do however loose the advantage of a modulated waveform (either pulsed or continuous), in that the receiver takes advantage of a matched filter, presented in this paper as either a mixer or a correlator, theoretically maximizing the SNR. An additional disadvantage of pulsing the transmitter, with a limited peak output power, is the reduction in average transmitted power and hence maximum range.

In an unmodulated pulsed radar, the resolution is directly proportional to the pulse width T as

$$\Delta R = \frac{cT}{2}.$$

As seen in figure C.9, the prototype waveform generator can transmit pulses as short as 120 ps, giving us a theoretical resolution of 18 mm. The inherent flexibility of a programmable bitstream allows for adjustable bandwidth as

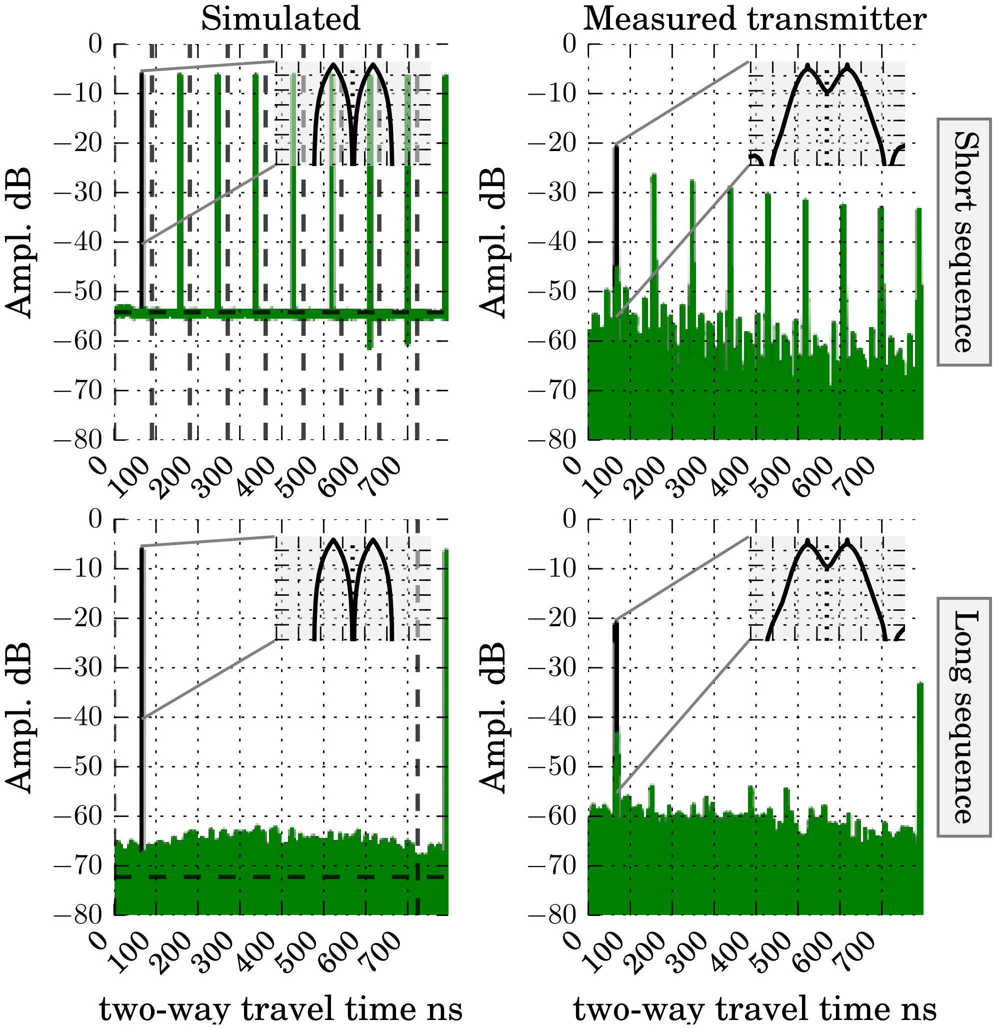


Figure C.8: Simulated m-sequence radar with 2 close targets. The top panels uses a 9 stage LFSR while the bottom panels uses a $m = 12$ stage. On the left, the entire system is simulated [14], on the right, the waveform generator is measured and used in the simulation. The simulation is averaged over 32 different threshold levels and include band limited noise with $\sigma = 0.1$.

shown by the difference between the “11” and “111” waveform. To conform to regulations, we can also insert notches in the mainlobe, as seen by two example double pulses “110011” and “1110000111”.

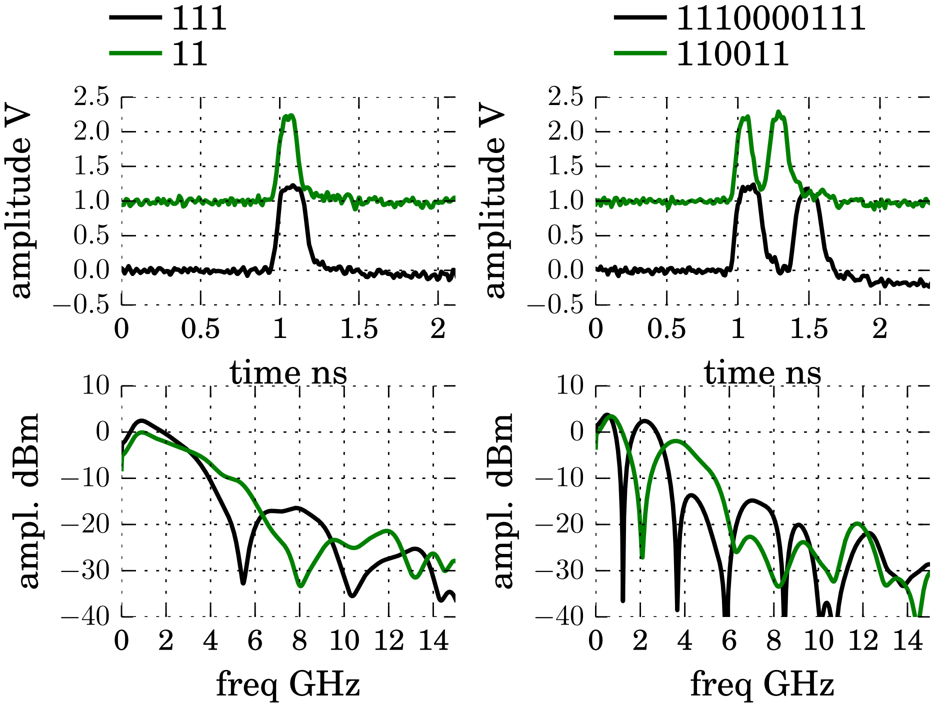


Figure C.9: Measurements of the waveform generator as a flexible pulse generator. Top: time domain view (different waveforms offset vertically for clarity). Bottom: Frequency domain view, found by the Welch method with a Hanning window and zero padding. The shortest pulse (“11”) has a measured 50 % pulse width of 120 ps and a 10 dB bandwidth of 5 GHz.

C.7 Measurements

In the preceding sections we have included measurements of the transmitter; programmed for a FMCW chirp bitstream in figure C.2, m-sequence bitstream in figure C.8 and pulsed in figure C.9. The transmitter is a slightly improved version of the one published in [17]. The chip also features a receiver, with a thresholder from Novelda [3], a XOR gate and the CW receiver counters depicted in figure C.5. The prototype is realized in a low power commercial 90 nm process. We will show the XOR gate in section C.7.2 with a post layout simulation and a FMCW measurement utilizing the generator, thresholder and mixer in section C.7.3.

The chip is programmed using a Raspberry Pi communication over Serial Peripheral Interface bus (SPI) and the threshold is set using an external Digital-to-Analog Converter (DAC). The CW counters can be accessed with the SPI interface, so with a SPI capable device we can realize a standalone SFCW radar. The chip can be used as an FMCW radar by externally sampling the XOR output as depicted in figure C.10 and doing a FFT on the captured data.

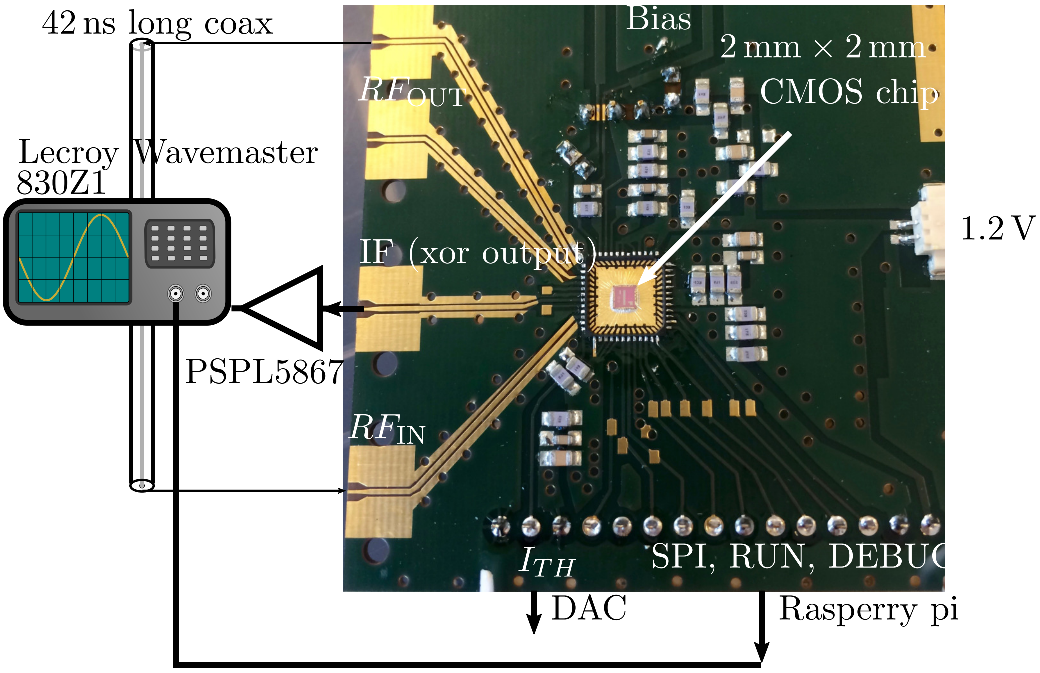


Figure C.10: Test setup when measuring the entire system. The chip is configured via SPI to the desired bitstream, it then transmits once the digital RUN signal enables. The channel is here emulated by a long coax. After going though the coax, the signal is compared to the externally set threshold current (I_{TH}) before being mixed and sent out of the chip again. The intermediate frequency (IF) is then sampled by an oscilloscope, which has an amplifier in front; to reduce noise. The chip is mounted in a standard QFN48 package (pictured with the lid of) and all of the surface mount device (SMD) components are decoupling capacitors. Note that there is no external clock/frequency reference as the chip is self-timed.

The waveform generator is programmed by writing the bitstream waveform to a 32 Kbytes on-chip-memory. Under typical supply and bias settings (supply and N-well at 1.2 V) the chip will read out this bitstream once every 16.7 µs. This can be slowed down by increasing the N-well voltage or decreasing the supply voltage to the transmitter.

C.7.1 FMCW linearity correction

Due to the transmitter being open loop, the transmitted chirp does not linearly increase in frequency. We correct for this using the efficient technique presented in [18]. The linearization re-samples the beat spectrum and corrects for first order non-linearity in the chirp, the idea starts by equating the measured beat with a constant frequency sine wave

$$2\pi f_{\text{beat}} \cdot t_{\text{old}}[n] = 2\pi f_{\text{meas}}[n] \cdot t_{\text{new}}[n] \tag{C.18}$$

where $f_{\text{beat}} = \alpha_1 \tau$ is a constant. If we assume the measurements to follow

$$f_{\text{meas}}[n] = \tau(\alpha_1 + \alpha_2 t_{\text{new}}[n]), \tag{C.19}$$

where α_2 is the unwanted first order non-linearity term. We can solve (C.18) to obtain our new sampling locations

$$t_{\text{new}}[n] = -\alpha_{12} \pm \sqrt{\alpha_{12}^2 + 2\alpha_{12} t_{\text{old}}[n]} \tag{C.20}$$

$$\text{where} \quad \alpha_{12} \equiv \frac{\alpha_1}{2\alpha_2} \tag{C.21}$$

Possibly due to a software–bug, we have found the above equation to give complex time locations, removing a factor of 2 has empirically given better results:

$$t_{\text{new}}[n] = -\alpha_{12} \pm \sqrt{\alpha_{12}^2 + \alpha_{12} t_{\text{old}}[n]}. \tag{C.22}$$

α_1 and α_2 is extracted from the beat spectrum by fitting a second order polynomial through the extracted phase. The phase is extracted by a Hilbert transform of the beat spectrum when there is only one target.

C.7.2 XOR gate as a digital mixer

We have stated that the only requirement for the harmonics to exist in a frequency modulated radar is that the mixer is digital. This implies that attenuation of the transmitters harmonics have little consequence to the presented principles, in addition, the digital mixer can even be used in a traditional analog radar. To show this we present a post-layout simulation from a commercial low power 90 nm CMOS process, where the inputs are sinusoidal.

A static and symmetric XOR gate is realized with 4 symmetric NAND gates [19] a layout is created and the parasitics are extracted. For realistic drive and load conditions, buffers are added to the XOR inputs and output as shown in figure C.11. In addition, a simply model for the supply pads are included in the circuit model, consisting of a 50 fF capacitor and a 2 nH inductor.

As is seen in figure C.11, the buffers have more than sufficient gain to drive the output to saturation, giving us a square wave digital signal with harmonics. These harmonics will mix, creating additional peaks in the beat spectrum at multiples of the expected beat frequency. We note that a perfect 50 % high/low waveform only has odd harmonics, while even harmonics are seen in the simulated XOR output.

C.7.3 Full system measurements

To show the full system, we connect the chip as shown in figure C.10 and produce figure C.12. The transmitter is here programmed for a shorter sweep, only 418 ns long. The xor output (IF) is captured by a oscilloscope and the result is coherently averaged and Fourier transformed on a computer. The transmitter is connected to the receiver through a 43 ns long coax, the resulting beat frequency peak at 51 MHz corresponds to a two-way travel time of 42 ns. The reader should note that the post layout simulation uses the extracted measured delay in the simulation, hence the match between simulated and measured peak beat frequency comes as no surprise (figure C.12).

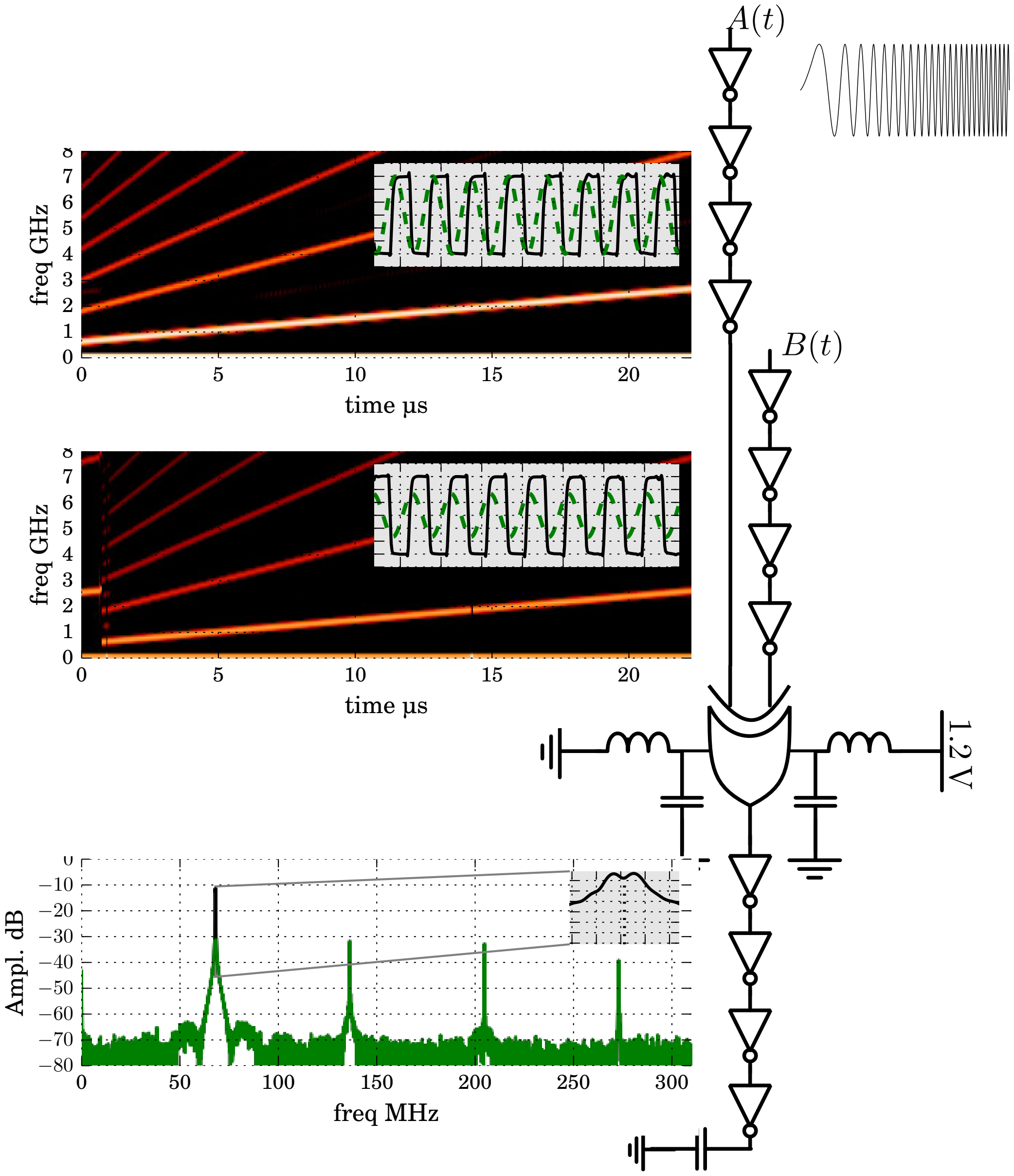


Figure C.11: Post layout simulation setup and results for a chirped sinusoidal input $A(t)$, and a delayed sinusoidal responce from 2 targets $B(t) = A(t - \tau_1)/2 + A(t - \tau_2)/2$. The insets of the two top panels, shows the sinusoidal inputs in time, where the dashed curve is the sinusoidal input and the whole black line is after the 4 inverters. Bottom plot shows a zoomed spectogram of the buffered XOR output, where the inset includes a low pass filtered responce as a visual reference. [14]

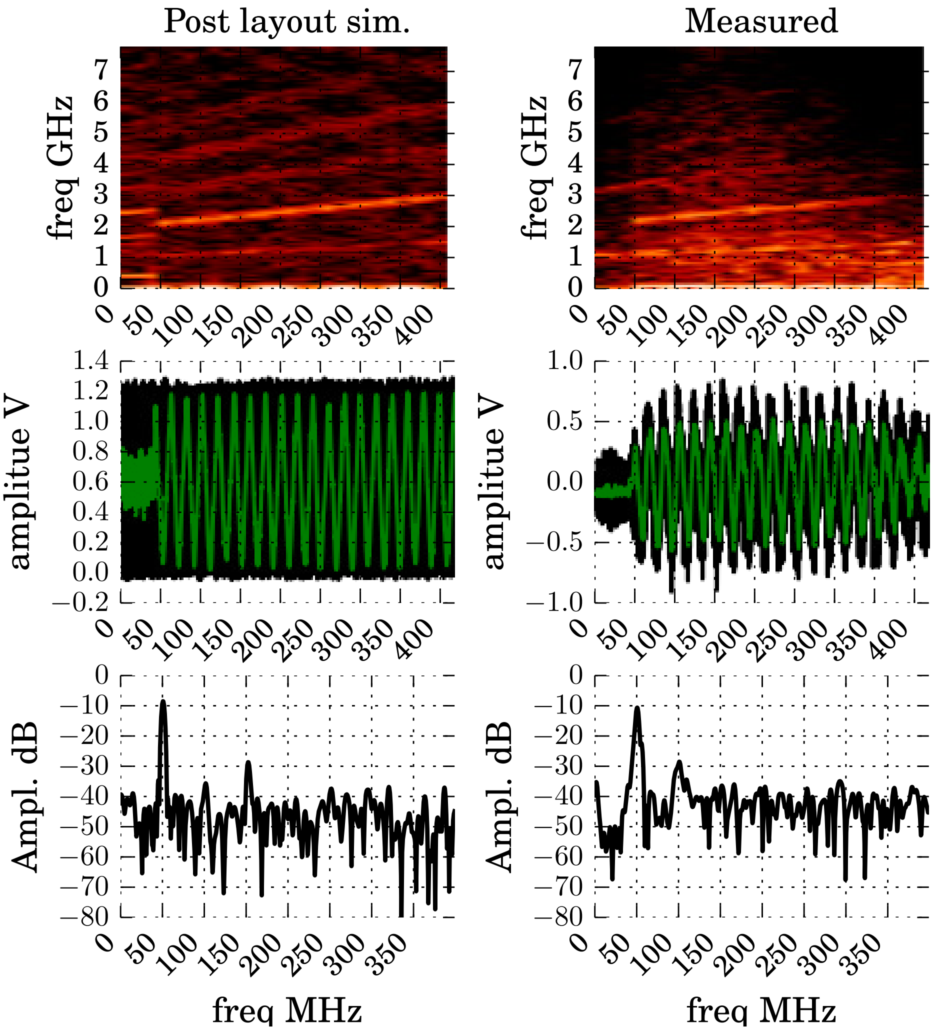


Figure C.12: Post layout simulation (left) and measurements (right) of a digital XOR gate mixing bitstreams. The measurements are for the full system, the waveform generator is transmitting a bitstream chirp from 1 GHz to 1.5 GHz in 418 ns and clocked out at 15.7 GHz though a 43 ns long coax, the return is then quantized and mixed with the transmitted copy before being sent out of the chip, amplified and captured by a oscilloscope. The post layout result is from a single simulation, while the measurements are the result of coherently averaging 283 times with different threshold levels.

C.8 Discussion

Previous all digital designs like those outlined in [20, 21], focus on digitally assisting in the generation and capture of analog waveforms. The novelty in this work is that the waveform itself is a digital bitstream, allowing for digital generation and processing with simple digital gates.

The implementation chosen in this work is digital, asynchronous and (mostly) continuous time.[1] Concepts presented in this paper is however fully compatible with a synchronous digital implementation, or even traditional analog. Of particular note is the increase in resolution when mixing the harmonics, as have been shown, the harmonics can be taken advantage of without a fully digital implementation. The transmitting and receiving antennas and amplifiers does not need to support the larger bandwidth of the harmonics. The drawback is that the fundamental and harmonics must be separated and that the various mixer cross-terms will lower the dynamic range.

A digital asynchronous implementation has the advantage of pushing modern technology to its limits, as it can take full advantage of the fine time delay of digital gates without the added overhead of timing margins and fast clock distribution. The achieved equivalent clock rate of 15.7 GHz would not be possible using standard synchronous design as a standard cell flip flop has a maximum clock rate below 8 GHz in the utilized 90 nm process[2]. Clock margins is avoided by self-timed chip design, so that the circuit will naturally slow down if the supply voltage drops or the temperature increases. This enables an open loop transmitter, where the output frequency will depend on voltage and temperature, making precise and stable frequency generation under voltage and temperature variations impossible without some form of feedback. For the frequency modulated architecture feedback can be added by using a Phase Locked Loop (PLL) with a stable frequency reference and a adjustable oscillator and divider, at the cost of significantly reduced flexibility as one is now limited to frequency modulated waveforms within the oscillators frequency range. Alternatively the waveform generator can use a fast and stable clock to read out the bitstream, at the cost of significantly lower sample rate and an increase in power consumption.

Programming the bitstream waveform by storing the bitstream in mem-

[1] the waveform generator is discrete, as it transmits a sequence of bits.

[2] according to the data-sheet under fast-fast conditions

ory gives maximum flexibility and is ideal for our prototype. That said, dedicated on-chip generators for the trivial m-sequence bitstream or even a Direct Digital Synthesizer (DDS) like frequency and phase counter would most likely be far more area and power efficient for these dedicated architectures.

In a conventional FMCW radar the chirp rate is kept low (typically on the order of $100\,\mathrm{MHz}/1\,\mathrm{ms} = 1 \times 10^{11}\,\mathrm{Hz/s}$), this reduces the sample rate of the beat spectrum. In our work, the single bit beat spectrum is easier to sample at high speeds and the waveform memory imposes a maximum limit on our chirp length, in addition, the lack of a frequency lock means we can easily transmit chirp rates on the order of $2\,\mathrm{GHz}/20\,\mu\mathrm{s} = 1 \times 10^{14}\,\mathrm{Hz/s}$. Our high chirp rate allows the chirp to be repeated 50 times and still get a frame-rate similar to a conventional radar, easing our assumption of a stationary scene.

This high chirp rate and low center frequency also allows us to safely ignore Doppler shift. For a target moving at the speed of sound $v_r = c_{\text{sound}} = 340\,\mathrm{m/s}$ and a chirp time of $T_m = 17\,\mu\mathrm{s}$ around $f_c = 1\,\mathrm{GHz}$, we expect a Doppler shift of only around $2v_r f_c/c = 2.3\,\mathrm{kHz}$, which is in essence a DC signal considering our chirp time and the assumed (fast!) speed.

Further study is required before the proposed principles can be applied to complex scenes with multiple moving targets (both with and without Doppler shift). One workaround is to apply multiple single bit parallel receivers, which is very feasible since the single bit processing is so area efficient. This will reduce the time required and hence easing the assumption of a stationary scene during the measurement period.

C.9 Conclusion

We have shown the feasibility of implementing several classical frequency modulated architectures, a pseudo-noise correlation radar and a simple pulsed radar; all in single bit CMOS implementation. Using bitstreams enables mixer operation of a frequency modulated radar to be implemented as a single CMOS XOR gate, which is proven by analytical treatment, post layout simulations and measurements. The digital nature of the radar also allows such operations as delays, which are proven useful to deal with the ambiguity created by harmonics as well as increasing the theoretical resolution by a factor of 3. Preliminary measurements from a digital asynchronous implementation in CMOS, prove the feasibility of bitstreams for generic single-chip-radar.

Bibliography

[1] J. Browne, "Looking further with modern radar systems." http://defenseelectronicsmag.com/systems-subsystems/looking-further-modern-radar-systems, July 2014.

[2] H. A. Hjortland, D. T. Wisland, T. S. Lande, C. Limbodal, and K. Meisal, "CMOS impulse radar," in *2006 NORCHIP*, pp. 75–79, 2006.

[3] "Single-chip radar sensors with sub-mm resolution - XeThru." http://www.novelda.no.

[4] M. Caruso, M. Bassi, A. Bevilacqua, and A. Neviani, "A 2-to-16GHz 204mW 3mm-resolution stepped-frequency radar for breast-cancer diagnostic imaging in 65nm CMOS," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*, pp. 240–241, Feb 2013.

[5] J. Sachs, *Handbook of Ultra-Wideband Short-Range Sensing*. Weinheim, Germany: John Wiley & Sons, 2013.

[6] I. Gresham, N. Jain, T. Budka, A. Alexanian, N. Kinayman, B. Ziegner, S. Brown, and P. Staecker, "A compact manufacturable 76-77-GHz radar module for commercial ACC applications," *IEEE Transactions on Microwave Theory and Techniques*, vol. 49, no. 1, pp. 44–58, 2001.

[7] T. Mitomo, N. Ono, H. Hoshino, Y. Yoshihara, O. Watanabe, and I. Seto, "A 77 GHz 90 nm CMOS transceiver for FMCW radar applications," *Solid-State Circuits, IEEE Journal of*, vol. 45, pp. 928–937, April 2010.

[8] J. Lee, Y. A. Li, M. H. Hung, and S. J. Huang, "A Fully-Integrated 77-GHz FMCW radar transceiver in 65-nm CMOS technology," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 12, pp. 2746–2756, 2010.

[9] R. J. Evans, P. M. Farrell, G. Felic, H. T. Duong, H. V. Le, J. Li, M. Li, W. Moran, and E. Skafidas, "Consumer radar: Opportunities and challenges," in *European Radar Conference (EuRAD), 2014 11th*, pp. 5–8, 2014.

[10] D. Guermandi, Q. Shi, A. Medra, T. Murata, W. V. Thillo, A. Bourdoux, P. Wambacq, and V. Giannini, "19.7 a 79GHz binary phase-modulated continuous-wave radar transceiver with TX-to-RX spillover cancellation in 28nm CMOS," in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, pp. 1–3, 2015.

[11] Q. Lv, Y. Dong, Y. Sun, C. Li, and L. Ran, "Detection of bio-signals from body movement based on high-dynamic-range doppler radar sensor (Invited)," in *RF and Wireless Technologies for Biomedical and Healthcare Applications (IMWS-BIO), 2015 IEEE MTT-S 2015 International Microwave Workshop Series on*, pp. 88–89, Sept 2015.

[12] D. Zito, D. Pepe, B. Neri, D. De Rossi, and A. Lanata, "Wearable System-on-a-Chip pulse radar sensors for the health care: System overview," in *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, vol. 2, pp. 766–769, May 2007.

[13] H. C. Kuo, C. C. Lin, C. H. Yu, P. H. Lo, J. Y. Lyu, C. C. Chou, and H. R. Chuang, "A fully integrated 60-GHz CMOS Direct-Conversion doppler radar RF sensor with clutter canceller for Single-Antenna non-contact human Vital-Signs detection," *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 4, pp. 1018–1028, 2016.

[14] Ø. Bjørndal, S.-E. Hamran, and T. S. Lande, "Square wave architectures for radar-on-chip," in *2016 46th European Microwave Conference (EuMC)*, pp. 1485–1488, 2016.

[15] H. Hjortland and T. Lande, "CTBV integrated impulse radio design for biomedical applications," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 3, pp. 79–88, April 2009.

[16] C. Nguyen and J. Park, *Stepped-Frequency Radar Sensors: Theory, Analysis and Design.* AG Switzerland: Springer, 2016.

[17] Ø. Bjørndal, S.-E. Hamran, and T. S. Lande, "UWB waveform generator for digital CMOS radar," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1510–1513, 2015.

[18] S. Scheiblhofer, S. Schuster, and A. Stelzer, “Signal model and linearization for nonlinear chirps in FMCW radar SAW-ID tag request,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 4, pp. 1477–1483, 2006.

[19] N. H. Weste and D. M. Harris, *Integrated circuit design.* Boston, MA, USA: Pearson, fourth ed., 2011.

[20] R. B. Staszewski, “State-of-the-Art and future directions of High-Performance All-Digital frequency synthesis in nanometer CMOS,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 7, pp. 1497–1510, 2011.

[21] A. Pozsgay, T. Zounes, R. Hossain, M. Boulemnakher, V. Knopik, and S. Grange, “A fully digital 65nm CMOS transmitter for the 2.4-to-2.7GHz WiFi/WiMAX bands using 5.4GHz delta sigma RF DACs,” in *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, pp. 360–619, 2008.

Appendix D

Paper IV. Power-Efficient, Gate-Based Digital-to-Time Converter in CMOS

The following is a re-formatted version of
Øystein Bjørndal and Tor Sverre Lande. Power-Efficient, Gate-Based Digital-to-Time converter in CMOS. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2017. doi:10.1109/ISCAS.2017.8050433

PhD Candidate contribution: Wrote the majority of the paper, implemented and measured the presented solution.

Supervisors contributions: Contributed to the original implementation concept and helped with the writing.

Abstract

A Digital-to-Time converter (DTC) based on static CMOS multiplexers is presented, achieving a time resolution of 65 ps consuming 0.5 mW. The DTC relies on gate delay for programmability, ensuring robustness, linearity and wide delay range. Details of the circuit implementation and optimization is given, with transistor sizes, post layout simulations with Monte Carlo, voltage and temperature variations are given with measurements from two different chip realizations in 90 nm CMOS.

D.1 Introduction

Time-to-digital conversion (TDC) is used in high precision timing measurement application exploring high-speed properties of digital processes. TDC circuits are often used in time-of-flight applications, such as pulsed laser rangefinders [1], in clock data recovery, digital Phase Locked Loop (PLL) [2] and even Analog-to-Digital Converter (ADC) [3]. The inverse conversion, namely a Digital-to-Time Converter (DTC), has gained less attention [4] and is the focus of this paper. The implementation of a non-clocked (continuous-time) TDC is in many ways similar to DTC design, exploring analog device properties. For proper performance detailed circuit analysis is required.

In our [5] paper, we explored a DTC for a digital Frequency-Modulated Continuous-Wave (FMCW) radar implementation. A simple delay-line based DTC, or serializer, of the type shown in figure D.1 was used in in parallel to generate a continuous bitstream output at GHz frequencies. In this paper we elaborate on circuit details allowing for more general use of the DTC, The DTC is also redesigned for improved performance documented both by simulations and measurements including robustness with respect to supply interference as well as temperature dependence. Fine tuning of delay elements is done by back-gate (well) tuning. All reported results are for a low power 90 nm CMOS by TSMC.

A delay line serializer is used by [6, 7, 8] for serial communication and called a Wave-Pipeline (WP). Our design goal is somewhat different from a serial communication system, as the temporal resolution is more important than maximum bitrate. We have measured an equivalent clock period of 65 ps, but are limited to a maximum data-rate of $7.7\,\text{Gb/s} = 1/(2 \cdot 65\,\text{ps})$.

Searching literature, few real WP implementations with measurements are found. The fastest simulated data-rates presented in the literature is based on Current Mode Logic (CML), in [8] a data-rate of 67 Gb/s is presented with 175 mW of power consumption. Lower power is claimed by [9], but without Monte Carlo simulations or measured results.

A simple, digital CMOS implementation, without current sources or other analog elements is considered more robust and versatile [10] than CML. In addition, a clockless design reduces power consumption since no clock power is consumed, [6] reports a 47 % power saving by removing the clock. With a 8 bit serializer transmitting 00110011, we have a post-layout simulated power consumption of 0.5 mW, while our entire 1:64 serializer with output

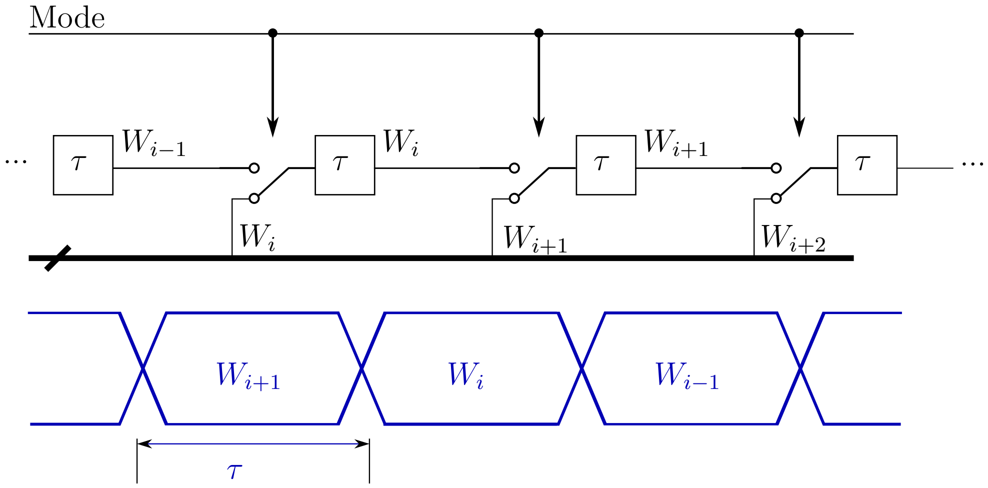


Figure D.1: Principle of a delay line based serializer. The delay line is preset with the bit values $\{W_i\}$, all of the switches are then switched so that the delay line becomes "whole", enabling the bits to flow out with an equivalent clock rate of $1/\tau$.

pads consume 13 mW. As we will show, analog tuning of the back-gate, will mitigate Process, Voltage and Temperature (PVT) variations and could even be used for fine delay steps.

We start with a quick look at how the Wave Pipeline (WP) architecture can be used as a delay generator in section D.2, before we dive into a detailed implementation of the mux/delay element which is at the heart of the WP DTC in section D.3. We then provide simulations and measurements of both our previous design and the improved inverting design in section D.4.

D.2 WP as delay generator

As an alternative use of our DTC, is as a delay generator. If we program the $\{W_i\}$ bits as a thermometer code, each thermometer increment will increase the delay by 65 ps with a simulated nominal ±1 ps DNL and INL. Results of 100 Post layout Monte Carlo simulations is shown in figure D.2. The delay step is not a sub-gate delay resolution in our process, but is still comparable to some of the work presented in the recent overview of delay lines in [11], while the DNL and INL is less than 1 LSB, ensuring linearity across mismatch and process variations.

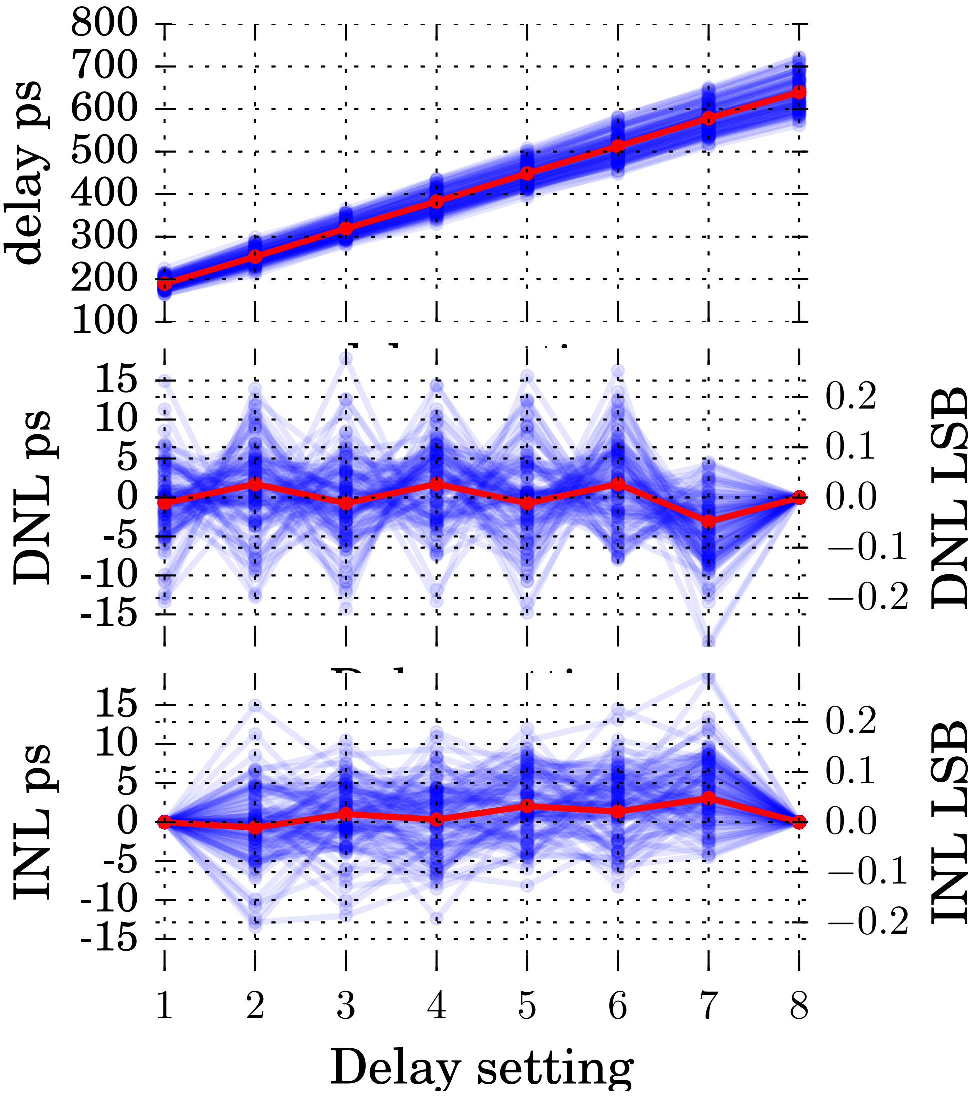


Figure D.2: Showing the WP serializer as a delay generator nominal simulation in red and 100 Monte Carlo simulations in transparent blue. Top: Digital delay setting and resulting delay, below that, Integral Nonlinearity (INL) and Differential Nonlinearity (DNL).

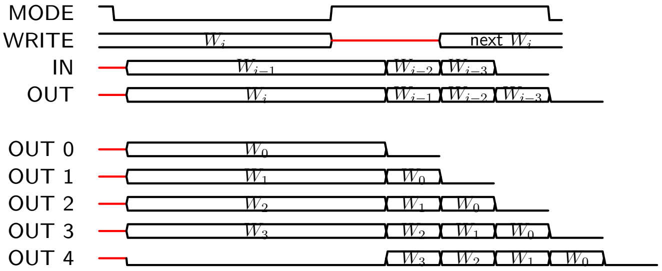


Figure D.3: Timing diagram for serializer row. The 4 first rows shows the signals for a single mux, while the bottom 5 shows the output for a 5 stage serializer. In the write mode (MODE=0), the input comes from the previous stage and stabilizes to W_{i-1}, while the output stabilizes to W_i. After MODE goes high, the output goes from W_i to W_{i-1} and then W_{i-2} and W_{i-3}.

D.3 Implementing the MUX

D.3.1 Choice of circuit

In [6] the multiplexer (mux) was implemented by two pass-transistors, although pass-transistor logic has very little delay, this comes at a cost of reduced robustness and gain, reducing the fall and rise time, especially when cascaded. In [7], an improvement in throughput is reported with a tri-state structure, but this stacks pMOS transistors and has a large capacitance on the output node. We here choose a fully static NAND based design, as it only stacks 2 NMOS transistors, giving us sharp transitions and a relatively short delay.

D.3.2 Sizing for the critical transitions

The serializer circuit will have 2 modes, a "load" mode, where the bits are set on the delay line and a "run" mode, when the bits are flowing out. As is seen from the timing diagram in figure D.3, the load mode is much less critical than the run mode, so it makes sense to optimize the transistor sizing for the run mode.

To optimize the transistor sizing in the mux, we start by identifying the critical transitions. As there are 3 input signals, where each signal can either be 0, 1 or transitioning from 0 → 1 or 1 → 0, we have a total of 56 combinations with at least one transitioning. If we limit ourselves to assuming only 1 input changes at a time, and consider only changes that actually affect the output, the number of combinations reduces to 12 and can be further simplified to the 8 transitions in table D.1 by inserting some do not care states X.

Table D.1: All transitions for a multiplexer with inputs WRITE and IN. Critical, non-critical and illegal (transitions that will not occur in our use of the mux)

nr.	MODE	WRITE	IN	OUT	
(0)	0	X	0 → 1	0 → 1	run mode (critical)
(1)	0	X	1 → 0	1 → 0	run mode (critical)
(2)	1	1 → 0	X	1 → 0	write mode (illegal)
(3)	1	1 → 0	X	1 → 0	write mode (illegal)
(4)	0 → 1	0	1	0 → 1	enable run (critical)
(5)	0 → 1	1	0	1 → 0	enable run (critical)
(6)	1 → 0	0	1	1 → 0	disable run (illegal)
(7)	1 → 0	1	0	0 → 1	disable run (non-critical)

The circuit is not a general purpose mux, but one optimized for our specific use, this does imply that the driver of MODE and WRITE must be designed with some care since we do make some assumptions. When MODE is stable, transitions on the selected input is transitioned to the output. We assume that WRITE signal is stable before we enter write mode, so we mark transitions (2-3) as illegal as these signify the write signal changing when we are in write mode. In addition, we assume the row is finished transmitting before we enter a new write mode, so we mark (6) as illegal because IN is high. The main goal is that we have identified transitions (0-1) and (4-5) as critical, these are the transitions that directly appear at the output of the row and we should optimize these for speed.

In figure D.4 we have drawn the three NAND gates on the transistor level. Starting with the mode (transition (0-1) in table D.1), we identify the critical path. In the run mode, the mux is basically two inverters, consisting of M2-M3 and M10-M11, the additional transistors are then sized either small, for low capacitive loading, or large for small on-resistance depending on the state these are in when in run mode.

Another set of critical transitions is when the mode changes to run, transition (4) in table D.1, require that M12 turns off (due to M5), before M11

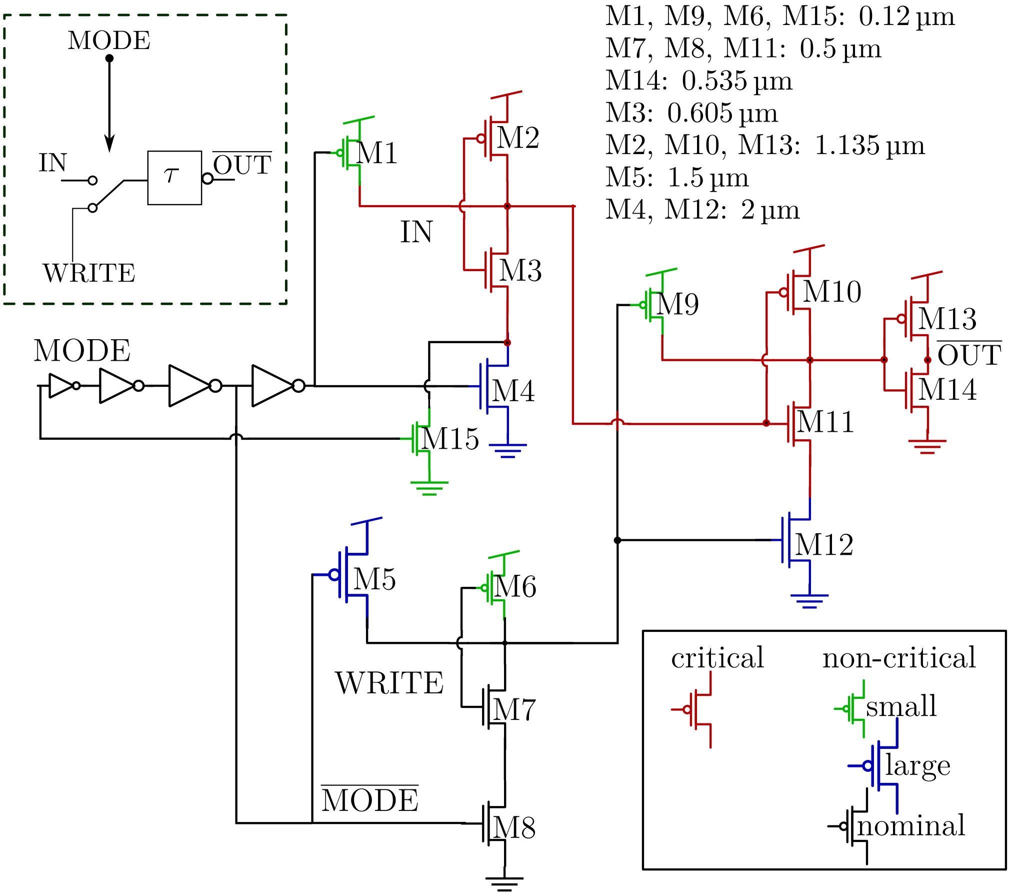


Figure D.4: Sizing guide for a NAND based mux with one inverter on the output, where the IN path (run mode) is much more critical than the WRITE path (write mode). Critical transistors/path drawn in red, additional transistors connected to the red path (M1 and M9) should be minimum width (drawn green) to avoid capacitive loading of the output while stacked transistors (M4 and M12) should be large for minimal on-resistance. Black transistors (M6, M7, M8) are in the WRITE path and does not load the IN path and can therefore be sized normally. All transistors are minimum length (100 nm drawn), final widths are indicated in the top right corner.

turns on to avoid a glitch which is avoided by buffering (delaying) the MODE signal, from the $\overline{\text{MODE}}$ signal. Case (5) is easier, as we only need M5 to be strong enough to turn M12 on, this is ensured by making M5 large. We pre-charge M4's drain node with M15 to alleviate the driving requirements of the fairly large device. The complete optimized circuit is shown in figure D.4.

In figure D.4 we have included local buffering of the MODE signal, minimizing the load capacitance. Note that the latency introduced by this delay has little consequence, adding a minor time-lag to all elements.

A single inverter (M13-M14) is added to the output, which will be explained in the next section.

D.4 Characterization

When characterizing the waveform generator circuit, there are a three key speed parameters. We have 1) delay, which sets the time resolution, which combined with the fall/rise time determines 2) the maximum frequency, or Pulse-Width (PW). Lastly 3) pulse distortion, measured as PW growth, is given by unmatched falling and rising delays.

Although related, the maximum frequency is not necessarily given by the delay. In theory the maximum frequency is set by the bitstream 010101, but if the delay through the mux is too short, the edges will overlap causing the short pulses to disappear. The original circuit used in [5] used two inverters as delays in an attempt to circumvent this, but Monte Carlo simulations and measurements proved that the circuit was unable to correctly transmit 010101 and the presented measurements are therefore the sequence 001100110011. For our use, in a single bit waveform generator, we need a certain level of oversampling, so it makes sense to maximize the clock rate at the cost of being unable to transmit 010101. As we will see, we found it beneficial to have a inverting mux, so one inverter is used in the optimized design. Each stage can be inverting as long as the chain is of even length and every odd stage has an inverted WRITE signal.

The third speed parameters is related to pulse deformation, if we input a pulse(-train) we want the pulses to maintain the shape. One major challenge when dealing with asynchronous signals is that when traveling down a delay-line pulses may shrink (or grow) slightly in each stage, until they completely disappear. It should be noted that for a pulse train both pulse width growth and pulse width shrinkage is bad, as both leads to deformation and may eventually lead to pulse losses.

D.4.1 N/P ratio

When setting the nMOS to pMOS (N/P) ratio, there are a couple of important points to keep in mind. The first is that we want to equalize the delay of the falling and rising transition, or equivalently, have a PW growth of 0 seconds. It should be noted that this is not the same as equalizing the fall and rise-time.

The second concern depend on the capacitive loading of the circuit, as was discussed by [12]. The optimal N/P ratio when considering varying capacitive load in a non-inverting circuit, yields a different zero crossing for each capacitive load, making the optimization process challenging as the optimal point depends on both intentional and parasitic load. The N/P ratio is made less important with the inverting design, as a falling edge becomes a rising edge in the next block, making the optimization process easier.

D.4.2 Supply sensitivity

It is well known that a simple static single ended CMOS circuit is sensitive to both supply voltage and temperature. One advantage of building a self-timed circuit is that this sensitivity only affects the absolute time/frequency of the output, not the relative. So if the voltage drops, or the temperature increases, the circuit will naturally slow down without any complex feedback loop. To show this, we simulate the waveform 001100110011, while measuring the absolute period and pulse deformation. For a functional chip, the relative pulse deformation needs to be within a fraction of the period, so we normalize the pulse growth to the pulse width and multiply by 100 to get a percentage.

Both post layout and measured results are presented in figure D.5 as a function of supply voltage, where Monte Carlo results are also visible. Sweeping the temperature, yields similar results, where the pulse width for the one inverter circuit varies linearly from 119 ps to 143 ps for −40 °C to 100 °C while the pulse growth is less than ±0.007 % of the pulse width.

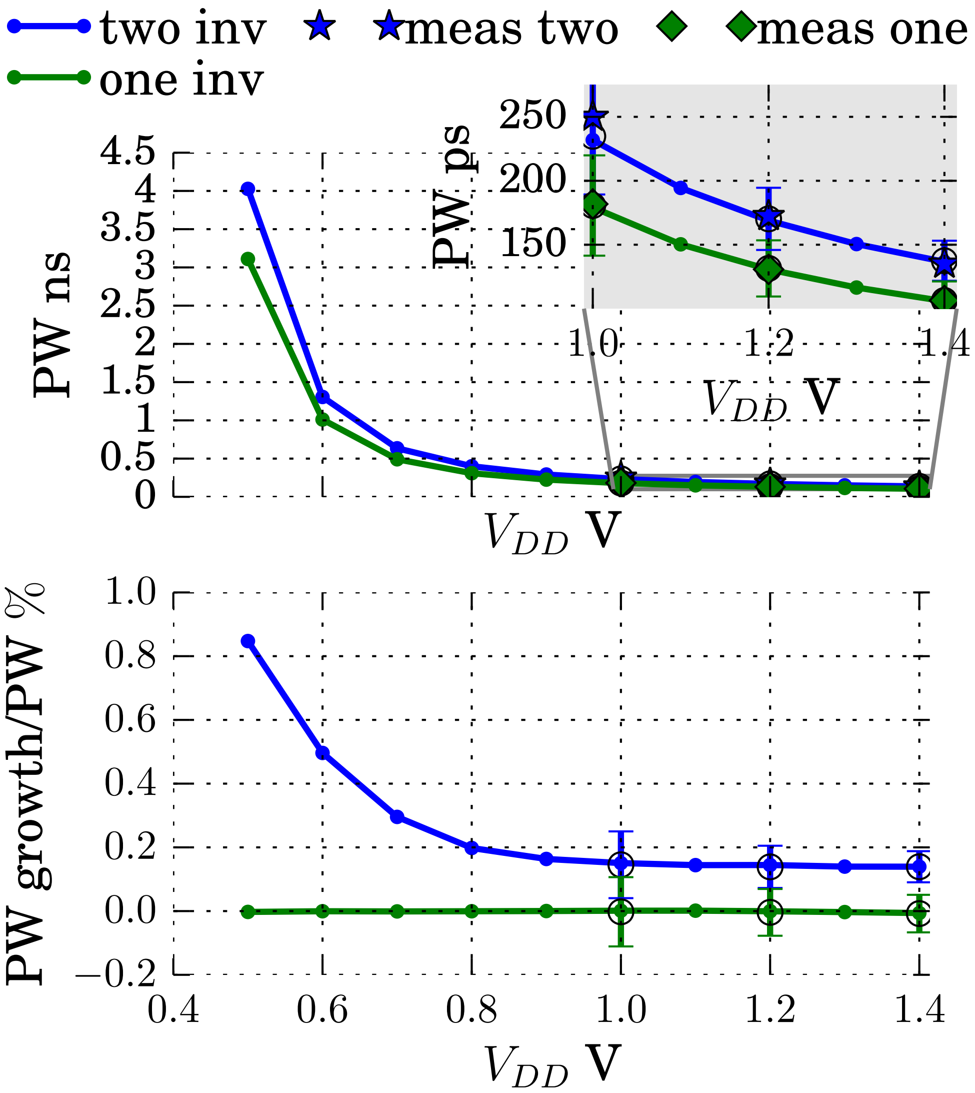


Figure D.5: Pulse Width (PW) and distortion as a function of supply voltage. Nominal simulation shown as dots connected with straight lines, and Monte Carlo simulations for selected voltages shown as circle (mean) and ± three standard deviation as error bars. The top panel also includes measured results, which are on average 1.7 % slower than the simulated values.

D.4.3 Back-gate tuning

A simple dynamic way of tuning the nMOS to pMOS ratio is to adjust the p-well and n-well voltages (back-gate). The advantages compared to current starving or varying the capacitive loading is that this does not reduce the maximum achievable speed when no tuning is required.

Back-gate tuning does however have a limited tuning range; with the original design with two inverters, adjusting the p-well from 600 mV to 1.4 V and the n-well from −300 mV to 300 mV only results in a change in pulse growth from 150 fs to 924 fs. This gets worse in the inverting design, as we are less sensitive to the N/P ratio, so adjusting all of the nMOS and pMOS transistors leaves us with a tuning range of 31 fs.

To remedy this limitation, we may adjust only every other stage equally. This does have some layout challenges, as wells must be separated, increasing the area and the parasitic wiring capacitance, but gives a much larger tuning range. An optimistic, initial, post layout simulation without the well separation; gives a tuning range of 9.5 ps, which corresponds to 292 Monte Carlo standard deviations. Our 90 nm process is a triple well process, so multiple p-wells can be adjusted independently without adjusting the entire chip; but as seen from table D.2 a n-well only tuning is also sufficient for mitigating PVT variations in the pulse width and eliminate PVT variations in the pulse growth.

Table D.2: Tunability of the optimized one inverter circuit when adjusting the back-gate of every other element. The last three rows show PVT variations; as the mean μ pluss/minus 3 standard deviation σ based on 200 Monte Carlo simulations, supply-. and temperature-variations.

	PW ps		PW growth ps	
	min	max	min	max
p-well only	120	144	−3.99	3.85
n-well only	124	136	−2.18	2.14
n-well and p-well tuning	116	148	−4.84	4.62
Monte Carlo $\mu \pm 3\sigma$	109	153	−0.10	0.09
Supply 1.2 V ± 0.2 V	106	181	−0.01	−0.00
Temp −40 °C to 100 °C	120	144	0.00	−0.01

D.4.4 Comparison

At this point, it makes sense to question if the design effort of section D.3 was worth it. To do so, we compare a naive symmetric NAND mux implementation, without any inverters on the output, with our optimized and skewed NAND mux including an extra inverter to obtain a inverting block. The simulated result give the same PW (128 ps to 131 ps in the naive and optimized version respectively), despite adding an extra gate. The optimized design has a pulse deformation within ±15 fs with a worst case standard deviation of ±73 fs over a range of Monte Carlo simulations with the supply ranging from 1 V to 1.4 V and the temperature from −40 °C to 100 °C. While the naive has a 122 times larger mean and a 6.6 times larger standard deviation over the same process conditions. In addition we have a improvement in the average fall/rise time (found from 10 % to 90 %) from 44 ps to 28 ps.

The fact that our optimized solution has the same delay despite an extra gate can largely be attributed to the skewed sizing and the improved fall and rise time. The improvement in pulse deformation is attributed to the inverting design, where instead of matching nMOS vs pMOS, we rely on matching equal devices.

D.5 Conclusion

A robust wave-pipeline is proposed as a low power DTC for on-chip serial communication or sub-clock delay-line applications. With a time resolution of 65 ps and a voltage, temperature, process and mismatch insensitive pulse deformation by inverting design and backgate tuning, static CMOS provide a robust solution. Details on the simple to implement multiplexer is given, together with post layout simulations with PVT and mismatch variations and measured results.

Bibliography

[1] P. Palojarvi, K. Maatta, and J. Kostamovaara, “Pulsed time-of-flight laser radar module with millimeter-level accuracy using full custom receiver and TDC ASICs,” *IEEE Transactions on Instrumentation and Measurement*, vol. 51, no. 5, 2002.

[2] R. Staszewski, S. Vemulapalli, P. Vallur, J. Wallberg, and P. Balsara, "Time-to-digital converter for RF frequency synthesis in 90 nm CMOS," in *Radio Frequency integrated Circuits (RFIC) Symposium, 2005. Digest of Papers. 2005 IEEE*, June 2005.

[3] J. Daniels, W. Dehaene, M. S. J. Steyaert, and A. Wiesbauer, "A/D conversion using asynchronous Delta-Sigma modulation and Time-to-Digital conversion," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 9, 2010.

[4] P. A. Nuyts, P. Reynaert, and W. Dehaene, "Continuous-time digital design techniques," in *Continuous-Time Digital Front-Ends for Multistandard Wireless Transmission.* Springer, 2014, ch. 4.

[5] Ø. Bjørndal, S.-E. Hamran, and T. S. Lande, "UWB waveform generator for digital CMOS radar," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015.

[6] S.-J. Lee, K. Kim, H. Kim, N. Cho, and H.-J. Yoo, "Adaptive network-on-chip with wave-front train serialization scheme," in *VLSI Circuits, 2005. Digest of Technical Papers. 2005 Symposium on*, June 2005.

[7] B. C. Hien, S.-M. Kim, and K. Cho, "Design of a wave-pipelined serializer-deserializer with an asynchronous protocol for high speed interfaces," in *Quality Electronic Design (ASQED), 2012 4th Asia Symposium on*, July 2012.

[8] R. Dobkin, M. Moyal, A. Kolodny, and R. Ginosar, "Asynchronous current mode serial communication," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 18, no. 7, July 2010.

[9] A. Jaiswal, D. Walk, Y. Fang, and K. Hofmann, "Low-power high-speed on-chip asynchronous wave-pipelined CML SerDes," in *System-on-Chip Conference (SOCC), 2014 27th IEEE International*, Sept 2014.

[10] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital integrated circuits.* Prentice hall Englewood Cliffs, 2002, vol. 2.

[11] B. I. Abdulrazzaq, I. Abdul Halin, S. Kawahito, R. M. Sidek, S. Shafie, and N. A. M. Yunus, "A review on high-resolution CMOS delay lines:

towards sub-picosecond jitter performance," *SpringerPlus*, vol. 5, no. 1, 2016.

[12] S. Sudalaiyandi, H. A. Hjortland, and T. S. Lande, "A continuous-time IR-UWB RAKE receiver for coherent symbol detection," *Analog Integrated Circuits and Signal Processing*, vol. 77, no. 1, 2013.

Part III

References

Bibliography

[Axe01] S. R. J. Axelsson. Noise radar for range/Doppler processing and digital beamforming using binary ADC. In *IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No.01CH37217)*, volume 5, pages 2001–2005 vol.5, 2001. doi:10.1109/IGARSS.2001.977884.

[BA91] L Bömer and M Antweiler. Perfect energy efficient sequences. *Electronics Letters*, 27(15):1332–1334, 1991.

[BBGN12] M. Bassi, A. Bevilacqua, A. Gerosa, and A. Neviani. Integrated SFCW transceivers for UWB breast cancer imaging: Architectures and circuit constraints. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 59(6):1228–1241, 2012. doi:10.1109/TCSI.2011.2173400.

[BHL15] Øystein Bjørndal, Svein-Erik Hamran, and Tor Sverre Lande. UWB waveform generator for digital CMOS radar. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1510–1513, 2015. doi:10.1109/ISCAS.2015.7168932.

[BHL16] Øystein Bjørndal, Svein-Erik Hamran, and Tor Sverre Lande. Square wave architectures for radar-on-chip. In *2016 46th European Microwave Conference (EuMC)*, pages 1485–1488, 2016. doi:10.1109/EuMC.2016.7824636.

[BHL17] Øystein Bjørndal, Svein-Erik Hamran, and Tor Sverre Lande. Bitstream radar waveforms for generic single-chip-radar. *International Journal of Microwave and Wireless Technologies*, 9(6):1325–1337, August 2017. doi:10.1017/S1759078717000782.

[BL17] Øystein Bjørndal and Tor Sverre Lande. Power-Efficient, Gate-Based Digital-to-Time converter in CMOS. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2017. doi:10.1109/ISCAS.2017.8050433.

[Bla74] N. M. Blachman. Sinusoids versus walsh functions. *Proceedings of the IEEE*, 62(3):346–354, 1974. doi:10.1109/PROC.1974.9428.

[Bou16] Nassim Bouassida. *Design of a full software transmitter based on walsh sequences*. PhD thesis, Bordeaux, 2016. URL: https://tel.archives-ouvertes.fr/tel-01497771/document.

[Bro07] G. M. Brooker. Mutual interference of Millimeter-Wave radar systems. *IEEE Transactions on Electromagnetic Compatibility*, 49(1):170–181, 2007. doi:10.1109/TEMC.2006.890223.

[BT59] R. B. Blackman and John W Tukey. *The measurement of power spectra : from the point of view of communications engineering*. Dover Publications, New York, 1959.

[BVGC10] J. Borremans, K. Vengattaramane, V. Giannini, and J. Craninckx. A 86MHz-to-12GHz digital-intensive phase-modulated fractional-N PLL using a 15pJ/Shot 5ps TDC in 40nm digital CMOS. In *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, pages 480–481, 2010. doi:10.1109/ISSCC.2010.5433840.

[CRC$^+$11] T. S. Chu, J. Roderick, S. Chang, T. Mercer, C. Du, and H. Hashemi. A short-range UWB impulse-radio CMOS sensor for human feature detection. In *2011 IEEE International Solid-State Circuits Conference*, pages 294–296, 2011. doi:10.1109/ISSCC.2011.5746325.

[Del75] Edward John Delp. Spectral analysis and sythesis using walsh functions. Master's thesis, University of Cincinnati, 1975. URL: https://engineering.purdue.edu/~ace/thesis/delp/delp-ms-thesis.pdf.

[Doo15] Malihe Zarre Dooghabadi. *IR-UWB Transmitter for Ranging Application an IR-UWB MIMO Radar-Based Imaging Array*. PhD thesis, University of Oslo, 2015.

[Ekr63] H. Ekre. Polarity coincidence correlation detection of a weak noise source. *Information Theory, IEEE Transactions on*, 9(1):18–23, 1963. doi:10.1109/TIT.1963.1057811.

[GKSN11] A. Ghaffari, E. A. M. Klumperink, M. C. M. Soer, and B. Nauta. Tunable High-Q N-Path Band-Pass filters: Modeling and verification. *IEEE Journal of Solid-State Circuits*, 46(5):998–1010, 2011. doi:10.1109/JSSC.2011.2117010.

[GNB+14] K. Granström, A. Natale, P. Braca, G. Ludeno, and F. Serafino. PHD extended target tracking using an incoherent x-band radar: Preliminary real-world experimental results. In *17th International Conference on Information Fusion (FUSION)*, pages 1–8, 2014.

[Gol00] Bar-Giora Goldberg. *Digital frequency synthesis demystified*. Newnes, 2000.

[HBB+15] Svein-Erik Hamran, Tor Berger, Sverre Brovoll, Leif Damsgård, Øystein Helleren, Hans Erik Amundsen, Lynn Carter, Rebecca Ghent, Jack Kohler, Michael Mellon, David Paige, and Dirk Plettemeier. RIMFAX: a GPR for the mars 2020 rover mission. In *Advanced Ground Penetrating Radar (IWAGPR), 2015 8th International Workshop on*, pages 1–4. IEEE, 2015. doi:10.1109/IWAGPR.2015.7292690.

[HHM82] G. Hermann, L. Horvath, and L. Monostori. Real-time monitoring of machine tools via Walsh-Hadamard tranform. In *ICASSP '82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 7, pages 343–346, 1982. doi:10.1109/ICASSP.1982.1171756.

[HJ73] Bernard A Hutchins Jr. Experimental electronic music devices employing walsh functions. *Journal of the Audio Engineering Society*, 21(8):640–645, 1973.

[HJL16] Eugin Hyun, Young-Seok Jin, and Jong-Hun Lee. A pedestrian detection scheme using a coherent phase difference method based on 2D Range-Doppler FMCW radar. *Sensors (Basel, Switzerland)*, 16(1):124, January 2016. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4732157/, doi:10.3390/s16010124.

[Hjo16] Håkon A Hjortland. *Sampled and Continuous-Time 1-Bit Signal Processing in CMOS for Wireless Sensor Networks*. PhD thesis, University of Oslo, 2016. URL: http://urn.nb.no/URN:NBN:no-53031.

[Hog81] E. Hogenauer. An economical class of digital filters for decimation and interpolation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(2):155–162, 1981. doi:10.1109/TASSP.1981.1163535.

[HS79] Martin Harwit and Neil J. A. Sloane. *Hadamard transform optics*. Academic Press, Inc. New York, 1979.

[HWL$^+$06] H. A. Hjortland, D. T. Wisland, T. S. Lande, C. Limbodal, and K. Meisal. CMOS impulse radar. In *2006 NORCHIP*, pages 75–79, 2006. doi:10.1109/NORCHP.2006.329248.

[Ins05] New Wave Instruments. Linear feedback shift registers. implementation, M-Sequence properties, feedback tables [online]. 2005. URL: http://www.newwaveinstruments.com/resources/articles/m_sequence_linear_feedback_shift_register_lfsr.htm [cited 2017-03-30].

[JM08] David A Johns and Ken Martin. *Analog integrated circuit design*, chapter 14. John Wiley & Sons, 2008.

[Joh06] S.F. Johnston. *Holographic Visions: A History of New Science*. H. Spencer Lecture. OUP Oxford, 2006. URL: https://books.google.no/books?id=iPfU_powAgAC.

[Jon10] B. E. Jonsson. On CMOS scaling and A/D-converter performance. In *NORCHIP 2010*, pages 1–4, 2010. doi:10.1109/NORCHIP.2010.5669483.

[Ked86] B. Kedem. Spectral analysis and discrimination by zero-crossings. *Proceedings of the IEEE*, 74(11):1477–1493, 1986. doi:10.1109/PROC.1986.13663.

[Lev88] Nadav Levanon. *Radar principles*. Wiley-Interscience, New York, 1988.

[LKK$^+$05] Se-Joong Lee, Kwanho Kim, Hyejung Kim, Namjun Cho, and Hoi-Jun Yoo. Adaptive network-on-chip with wave-front train serialization scheme. In *VLSI Circuits, 2005. Digest of Technical Papers. 2005 Symposium on*, pages 104–107, June 2005.

[LNLC15] T. Lauteslager, N. Nicolaou, T. S. Lande, and T. Constandinou. Functional neuroimaging using UWB impulse radar: A feasibility study. In *2015 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4, 2015. doi:10.1109/BioCAS.2015.7348387.

[Lyo] Richard Lyons. Understanding cascaded integrator-comb filters. March,2005.

[MA09] Mark D McDonnell and Derek Abbott. What is stochastic resonance? definitions, misconceptions, debates, and its relevance to biology. *PLoS Comput Biol*, 5(5):e1000348, 2009.

[MAZ$^+$14] M. T. Moreira, M. Arendt, A. Ziesemer, R. Reis, and N. L. V. Calazans. Automated synthesis of cell libraries for asynchronous circuits. In *2014 27th Symposium on Integrated Circuits and Systems Design (SBCCI)*, pages 1–7, 2014. doi:10.1145/2660540.2660984.

[McD06] Mark Damian McDonnell. *Theoretical Aspects of Stochastic Signal Quantisation and Suprathreshold Stochastic Resonance*. PhD thesis, The University of Adelaide, Australia, February 2006. URL: http://hdl.handle.net/2440/85501.

[Mur16] B. Murmann. ADC performance survey 1997-2016 [online]. 2016. URL: https://web.stanford.edu/~murmann/adcsurvey.html [cited 2017-03-03].

[Rab09] Jan Rabaey. *Low power design essentials*. Springer Science & Business Media, 2009.

[Raz97] B. Razavi. Design considerations for direct-conversion receivers. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 44(6):428–435, 1997. doi:10.1109/82.592569.

[Ree10] B.A. Reeves. Noise augmented radar system. https://www.google.com/patents/US7656341, February 2010. US Patent 7,656,341. URL: https://www.google.com/patents/US7656341.

[Rem66] Winslow R Remley. Some effects of clipping in array processing. *The Journal of the Acoustical Society of America*, 39(4):702–707, 1966.

[Ric14] Mark A. Richards. *Fundamentals of Radar Signal Processing*, chapter 4. McGraw-Hill, New York, second edition, 2014.

[Sac13] Jürgen Sachs. *Handbook of Ultra-Wideband Short-Range Sensing*. John Wiley & Sons, Weinheim, Germany, 2013.

[SADRH08] K. A. Shehata, M. A. Aboul-Dahab, S. H. El Ramly, and K. A. Hamouda. An FPGA based 1-bit all digital transmitter employing Delta-Sigma modulation with RF output for SDR. In *2008 2nd International Conference on Signals, Circuits and Systems*, pages 1–6, 2008. doi:10.1109/ICSCS.2008.4746928.

[SDV$^+$10] S. Sudalaiyandi, M. Z. Dooghabadi, T. A. Vu, H. A. Hjortland, Ø. Næss, T. S. Lande, and S. E. Hamran. Power-efficient CTBV symbol detector for UWB applications. In *2010 IEEE International Conference on Ultra-Wideband*, volume 1, pages 1–4, 2010. doi:10.1109/ICUWB.2010.5614422.

[Sha00] G. G. Shahidi. Challenges of CMOS scaling at below 0.1 mu;m. In *ICM 2000. Proceedings of the 12th International Conference on Microelectronics. (IEEE Cat. No.00EX453)*, pages 5–8, 2000. doi:10.1109/ICM.2000.916402.

[Sot16] P. P. Sotiriadis. Spurs-Free Single-Bit-Output All-Digital frequency synthesizers with forward and feedback spurs and noise cancellation. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63(5):567–576, 2016. doi:10.1109/TCSI.2016.2525058.

[SSFS08] S. Schuster, S. Scheiblhofer, R. Feger, and A. Stelzer. Signal model and statistical analysis for the sequential sampling pulse radar technique. In *2008 IEEE Radar Conference*, pages 1–6, 2008. doi:10.1109/RADAR.2008.4720961.

[Sta11] R. B. Staszewski. State-of-the-Art and future directions of High-Performance All-Digital frequency synthesis in nanometer CMOS. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 58(7):1497–1510, 2011. doi:10.1109/TCSI.2011.2150890.

[Sto00] NG Stocks. Suprathreshold stochastic resonance in multilevel threshold systems. *Physical Review Letters*, 84(11):2310, 2000.

[SVH$^+$12] S. Sudalaiyandi, T. A. Vu, H. A. Hjortland, Ø. Næss, and T. S. Lande. Continuous-time single-symbol IR-UWB symbol detection. In *2012 IEEE International SOC Conference*, pages 198–201, 2012. doi:10.1109/SOCC.2012.6398347.

[TKL16] M. Tømmer, K. G. Kjelgård, and T. S. Lande. Body coupled wideband monopole antenna. In *2016 Loughborough Antennas Propagation Conference (LAPC)*, pages 1–5, 2016. doi:10.1109/LAPC.2016.7807483.

[TRG71] J. Tierney, C. Rader, and B. Gold. A digital frequency synthesizer. *IEEE Transactions on Audio and Electroacoustics*, 19(1):48–57, 1971. doi:10.1109/TAU.1971.1162151.

[Tsi06] Y. Tsividis. Mixed-Domain systems and signal processing based on input decomposition. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 53(10):2145–2156, Oct 2006. doi:10.1109/TCSI.2006.882822.

[ULB$^+$14] Fawwaz Tayssir Ulaby, David Gardner Long, William J Blackwell, Charles Elachi, Adrian K Fung, Chris Ruf, Kamal Sarabandi, Howard A Zebker, and Jakob Van Zyl. *Microwave radar and radiometric remote sensing*, volume 4. University of Michigan Press Ann Arbor, 2014.

[VH15] Giuseppe Venturini and Shayne Hodge. python-deltasigma: python-deltasigma v.0.2.1. http://dx.doi.org/10.5281/zenodo.18529, jun 2015. doi:10.5281/zenodo.18529.

[VM66] J. H. Van Vleck and D. Middleton. The spectrum of clipped noise. *Proceedings of the IEEE*, 54(1):2–19, 1966. doi:10.1109/PROC.1966.4567.

[Wat62] D.G. Watts. A general theory of amplitude quantization with applications to correlation determination. *Proceedings of the IEE - Part C: Monographs*, 109(15):209–218, 1962. doi:10.1049/pi-c.1962.0027.

[Wei63] Sander Weinreb. *A digital spectral analysis technique and its application to radio astronomy*. PhD thesis, Massachusetts Institute of Technology, 1963.

[Wei94] L. G. Weiss. Wavelets and wideband correlation processing. *IEEE Signal Processing Magazine*, 11(1):13–32, 1994. doi:10.1109/79.252866.

[Wes16] Tim Wescott. Sampling: what nyquist didn't say, and what to do about it. *Wescott Design Services, Oregon City, OR*, June 2016. URL: http://www.wescottdesign.com/articles/Sampling/sampling.pdf.

[Wil81] R. Wiley. Approximate FM demodulation using zero crossings. *IEEE Transactions on Communications*, 29(7):1061–1065, 1981. doi:10.1109/TCOM.1981.1095091.

[WMR$^+$04] P. Wagh, P. Midya, P. Rakers, J. Caldwell, and T. Schooler. An all-digital universal RF transmitter [CMOS RF modulator and PA]. In *Custom Integrated Circuits Conference, 2004. Proceedings of the IEEE 2004*, pages 549–552, Oct 2004.

[WSJ06] C. Wagner, A. Stelzer, and H. Jager. PLL architecture for 77-GHz FMCW radar systems with Highly-Linear Ultra-Wideband frequency sweeps. In *2006 IEEE MTT-S International Microwave Symposium Digest*, pages 399–402, 2006. doi:10.1109/MWSYM.2006.249555.

[Xia92] Ning Xiang. Using m-sequences for determining the impulse responses of LTI-systems. *Signal Processing*, 28(2):139–152, 1992. URL: http://symphony.arch.rpi.edu/~xiangn/Papers/SignalProc1992Xiang.pdf.

[ZSJ$^+$16] B. Zhou, S. X. Shen, Y. C. Ji, W. Lu, F. Zhang, G. Y. Fang, Y. Su, and S. Dai. The subsurface penetrating radar on the rover of china's mars 2020 mission. In *2016 16th International Conference on Ground Penetrating Radar (GPR)*, pages 1–4, 2016. doi:10.1109/ICGPR.2016.7572700.