A Primer on Software Defined Radios

Dimitrie C. Popescu, *Senior Member, IEEE* and Rolland Vida, *Senior Member, IEEE*

***Abstract*—The commercial success of cellular phone systems during the late 1980s and early 1990 years heralded the wireless revolution that became apparent at the turn of the 21st century and has led the modern society to a highly interconnected world where ubiquitous connectivity and mobility are enabled by powerful wireless terminals. Software defined radio (SDR) technology has played a major role in accelerating the pace at which wireless capabilities have advanced, in particular over the past 15 years, and SDRs are now at the core of modern wireless communication systems. In this paper we give an overview of SDRs that includes a discussion of drivers and technologies that have contributed to their continuous advancement, and presents the theory needed to understand the architecture and operation of current SDRs. We also review the choices for SDR platforms and the programming options that are currently available for SDR research, development, and teaching, and present case studies illustrating SDR use. Our hope is that the paper will be useful as a reference to wireless researchers and developers working in the industry or in academic settings on further advancing and refining the capabilities of wireless systems.**

***Index Terms*—Software defined radio, field programmable gate array, digital signal processing, wireless communication networks.**

I. INTRODUCTION

OVER the past three decades wireless communication systems have revolutionized the modern society, becoming essential components of our daily lives. Today's wireless devices provide much more than the mobile phone service enabled by the first generation of cellular phones available during the 1980s. They make extensive use of the Internet with capabilities that include accessing business and financial data, providing email, text messaging, and videoconference capabilities, enabling online shopping and entertainment with augmented reality features, assisting drivers with navigation and up-to-the-minute traffic information, and many more. Consumers preference of a wireless device and design has even become a personal statement about their status and social identity.

This unprecedented revolution in wireless communication systems occurred over multiple generations of wireless technologies that succeeded since the late 1980s and has been fueled by two main factors that have acted in synergy:

- Advances in hardware, starting from the clumsy, brick-like mobile phone terminals in the first generation to the sleek smartphones of the current generation that bring the Internet to our finger tips.
- Demand from consumers and society for applications that evolved from providing basic voice service using mobile phones and enabling wireless networking over short distances, and have advanced to supporting ubiquitous connectivity and edge computing through a vast heterogeneous infrastructure of interconnected wired and wireless networks.

A significant shift in the design paradigm of wireless systems occurred during the mid 1990s with the transition between second and third generations, when the SDR concept was formally introduced by visionary engineer and wireless pioneer Joseph Mitola [1], [2]. According to the Wireless Innovation Forum [3], a SDR is defined as "a radio in which some or all of the physical layer functions are software defined". We note that the physical layer of a communication system has been traditionally associated with the hardware, and any changes to physical layer functions such as modifying the modulation scheme or changing the frequency band associated with a particular system for example, would require hardware changes. Thus, in order to support multiple wireless standards on a conventional radio, all the corresponding hardware blocks would have to be built in, which would increase the manufacturing cost and limit flexibility to a predefined set of choices. By contrast, SDRs have a minimal set of hardware components and can change their operating parameters as needed through programming, providing a cost-effective alternative to multifunctional wireless devices.

In the three decades that have passed since the introduction of the SDR concept, SDRs have facilitated major advances in wireless communication systems through low-cost rapid prototyping, becoming the building blocks of modern communication systems. We note that, despite the fact that three decades of existence is expected to be a significant life time in the realm of modern electrical and electronic technologies, SDRs continue to thrive and are an ubiquitous presence in all aspects of research, development, and teaching of wireless communication systems and networks.

Motivated by the vitality of SDR technologies, in this paper we provide an overview of their salient aspects that can be used as a self-guided introduction to SDRs. We start by reviewing, in Section II, the drivers and enabling technologies that have shaped the SDR evolution over the past three decades, highlighting the current trends that maintain SDRs in the focus of the wireless communications research and development communities. We continue with a brief theoretical background, in Section III, that is indispensable to understanding SDR operation. This includes representation of bandpass signals in terms of in-phase and quadrature components along with heterodyning for frequency up- and down-conversions, and is

---

This work was completed while the first author was on a Fulbright US Scholar fellowship at the Budapest University of Technology and Economics.

D. C. Popescu is with the Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, VA 23529, USA. (e-mail: dpopescu@odu.edu)

R. Vida is with the Department of Telecommunications and Media Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Magyar tudósok körútja 2, 1117 Budapest, Hungary. (e-mail: vida@tmit.bme.hu)

DOI: 10.36244/ICJ.2022.3.3

followed by presentation of SDR architectures in Section IV. SDR choices that are currently available on the commercial market along with programming alternatives are reviewed in Section V. Two case studies illustrating the use of SDR platforms in academic projects are also reviewed, in Section VI, before concluding the paper with final remarks in Section VII.

We hope that this SDR primer will become a useful reference to wireless researchers and developers working in the industry or in academic settings on future generations of wireless communication systems.

II. SDR DRIVERS AND ENABLING TECHNOLOGIES

Similar to cellular wireless systems, which have matured over multiple generations, SDRs have also seen the succession of multiple generations over which they have developed and have been refined. The timeline of SDR generations, however, does not align with that of the cellular wireless systems that have succeeded in the commercial/consumer market. Rather, SDR generations started in the late 1990s and are defined in terms of their increasing volume and presence in the overall wireless industry as outlined in [4] and illustrated in Fig. 1.

A. First SDR Generation

In the early days of SDR, during the late 1990s, the main driver was the defense industry with its efforts aimed at replacing existing radios used by the US military with a single one that was dubbed the Joint Tactical Radio System (JTRS) [5]. The idea behind the JTRS was that the new system could be programmed for multimode radio operation to eliminate the need for multiple radio units in a single military vehicle, and system upgrades would also be performed through software updates rather than through hardware changes. Besides the JTRS, other drivers of the initial development of SDRs include public-safety communications [6] along with spectrum monitoring and signal intelligence (SIGINT) [7], [8].

In terms of enabling technologies, the late 1990s and early 2000 years witnessed significant advances in integrated circuits (ICs) for radio frequency (RF) applications (also referred to as RFICs) [9] as well as in field programmable gate array (FPGA) technology [10]. These advances supported the needs of the defense-related SDR applications while also impacting the commercial market. Specifically, RFIC manufacturers were able to overcome important design challenges related to practical implementations of highly-integrated RF transceivers using CMOS technology, and RFICs advanced towards system-on-chip (SoC) solutions that combined complex RF analog and digital functionality, making possible the “ultimate transmission” [11]. At the same time, implementations of digital signal processing (DSP) algorithms using FPGA-based hardware also advanced to the point where they would be able to compete with application-specific integrated circuits (ASICs) and application-specific standard products (ASSPs) used in the current wireless communication systems [12], [13].

B. Subsequent SDR Generations

The advances made in RFIC and FPGA technologies led to the emergence of a commercial ecosystem of providers supporting SDR applications and prompted a second generation of SDRs in the early 2000 years. Equipment providers came to the realization that SDR architectures are also beneficial to the development of cellular wireless systems [14] and SDRs made their way into the fourth generation (4G) equipment that was aligning with the long-term evolution (LTE) and LTE-advanced (LTE-A) standards [15], [16]. As a result, the LTE base station infrastructure was developed using SDR RFIC and FPGA technology, and new concepts such as software defined networking (SDN) and network function virtualization (NFV) were introduced as an approach to decouple the various network functions and services from the underlying hardware components of the network in order to support legacy services as well as future evolutions [17]. In addition, a software communications architecture (SCA) was established as a distributed systems architecture that allows the distinct components of a SDR application to run on different processors, which communicate with each other based on the Common Object Request Broker Architecture (CORBA) middleware [18].

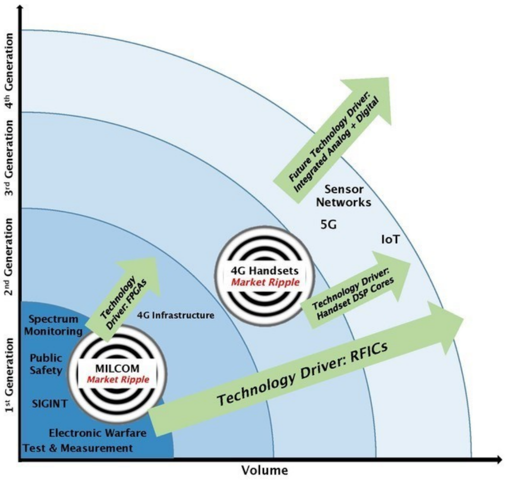


Fig. 1. The succession of SDR generations along with the drivers and enabling technologies for each generation [4].

Further advances in low-power high-performance ICs prompted the move of SDR technology to the handset segment of the 4G LTE networks starting in the early 2010 years. Specifically, low-power RFICs [19] in conjunction with high-performance FPGAs optimized to function as DSP cores [20] have started to be used in consumer handsets, significantly increasing the volume of SDRs on the commercial market. This marked the third generation of SDRs that also resulted in the SDR technology becoming a de facto industry standard for radios.

C. Future Trends

Currently, emerging systems such as the fifth generation (5G) of cellular systems and the Internet of Things (IoT), provide impetus for further development of SDR technology that will include advances on both sides of SDR platforms, hardware and software.

In terms of technology drivers, advances are expected to occur on the hardware side of SDRs that will bring the analog and digital sides closer together [21], by combining them in a single monolithic chip that will result in integration of the FPGAs or of the ASICs with the analog-to-digital converters (ADC) and digital-to-analog converters (DAC) [22], which will likely lower the overall size and cost of the SDR platforms, making them even more affordable and widespread in practice. At the same time, on the software side of the SDRs, the programming tools used by developers and researchers will evolve to enable the implementation of more complex tasks and novel DSP algorithms on increasingly more powerful FPGAs and ASICs.

The wireless industry will continue to rely on SDRs in the development of 5G systems, using them for various purposes that include practical experimentation and prototyping [23], as well as for enabling reconfigurable wireless networks with efficient spectrum utilization where the SDRs provide the programmable RF front-end needed for adjusting modulation schemes for operation in different frequency bands [24]. In this direction we note the performance evaluation of the non-orthogonal multiple access (NOMA) approach using SDR platforms in [25] and the 5G radio prototypes that are based on SDRs [26], [27].

5G systems are also expected to support the IoT with its specific requirements implied by the need to interconnect a multitude of sensors operating on strict energy and latency constraints [28], and SDRs will also be beneficial to the development of IoT networks by enabling rapid prototyping and experimentation. In this direction we note the SDR implementations of time-sensitive IoT networks in [29] and of RF identification (RFID) readers in [30], [31]. In addition, SDR implementations of receivers for the proprietary long range low power (LoRa) modulation technique [32] have recently been presented [33], [34].

Other emerging applications that have started to influence SDR development and evolution in recent years include satellite communications, where the SDR cost and versatility makes them attractive for implementing reconfigurable radio links that can deliver high data rate with low power consumption in small satellite systems [35]. In addition, satellite communications are also envisioned to support the Internet of Remote Things (IoRT), where sensors or other smart devices are located in remote areas or they are dispersed over a wide geographical area such that they are inaccessible to terrestrial networks [36] and SDR-based gateways are used to connect them to a satellite network [37].

III. THEORETICAL BACKGROUND

Like any other type of radio system, a SDR is used to transmit and receive bandpass signals that carry information. In order to have a complete picture of SDR operation, a good understanding of the canonical representation of bandpass signals [38, Appendix 2] is a necessary prerequisite. This need is also emphasized in references that discuss the more general concept of “software defined electronics” [39], [40], which includes SDRs as well as other types of modern measurement systems that rely on converting bandpass RF signals to lowpass baseband equivalent ones and then using software approaches for further processing.

Bandpass signals are formally defined as signals with spectrum concentrated in a band of frequencies that is centered at some frequency f_c, usually much larger than the bandpass signal bandwidth, which is denoted by $2W$ and extends over the frequency interval $[f_c - W, f_c + W]$. They are frequently encountered in communication systems and are obtained from a baseband information-bearing signal by applying a specific modulation scheme to a sinusoidal carrier signal with frequency f_c. To illustrate bandpass signals with a practical example, Fig. 2 shows the instantaneous spectrum along with the waterfall plot corresponding to a RF scan of the FM broadcast band displaying three active FM stations. We note that, the three distinct stations that are active display different patterns of frequency use in time as seen in the waterfall plot part of Fig. 2, which correspond to the distinct music and/or talk shows broadcast at scan time on the three stations. Their instantaneous spectra, however, look similar as they correspond to bandpass signals obtained by applying the same type of modulation (frequency modulation – FM) to baseband signals that contain similar information (music and speech signals).

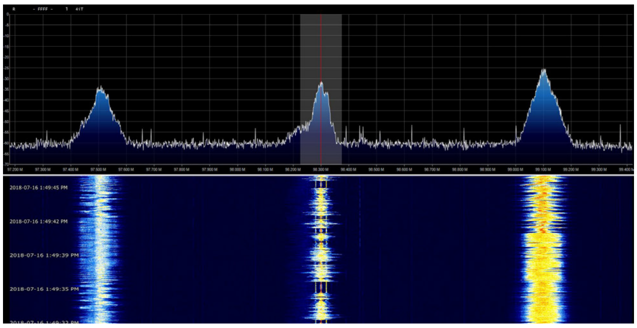


Fig. 2. Example of bandpass signal spectra collected in a RF scan of the FM broadcast bands [41]. The RF spectrum monitoring application displays only the spectral images that correspond to positive frequencies.

A. Pre-Envelope and Complex Lowpass Equivalent Signals

We consider an arbitrary bandpass signal $s(t)$ with a generic amplitude spectrum $|S(f)|$ shown in Fig. 3(a)[1], and we note that the fist step in obtaining the canonical representation of bandpass signals is to construct the *pre-envelope signal*, which is a complex-valued signal whose real part consists of the original bandpass signal $s(t)$, while its imaginary part consists of the Hilbert transform $\hat{s}(t)$ of the bandpass signal $s(t)$:

$$s_+(t) = s(t) + j\hat{s}(t) \tag{1}$$

We note that the Hilbert transform performs a phase shift of $\pm\pi/2$ on all components of $s(t)$ and may be obtained by passing $s(t)$ through a linear filter[2] with impulse response $h(t) = 1/(\pi t)$ and transfer function $H(f) = -j\mathrm{sgn}(f)$, where $\mathrm{sgn}(\cdot)$ denotes the signum function [38, Appendix 2]. We also note some properties of the Hilbert transform that are relevant to the canonical representation of bandpass signals:

- A signal $s(t)$ and its Hilbert transform $\hat{s}(t)$ are orthogonal, that is
$$\int_{-\infty}^{\infty} s(t)\hat{s}(t)dt = 0 \tag{2}$$

---

[1] The bandpass signal $s(t)$ is assumed to be real-valued, hence the symmetry of its amplitude spectrum $|S(f)|$ with respect to the vertical axis that can be noticed in Fig. 3(a).

[2] This linear filter is referred to as a *Hilbert transformer* [38, Appendix 2].

The orthogonality property of the Hilbert transform goes along with the intuition that the real and imaginary parts of a complex-valued quantity are real-valued quantities corresponding to two orthogonal dimensions that are represented by the horizontal and vertical axes of the Cartesian representation of the complex plane, and support the construction of the pre-envelope signal (1) having

$$s(t) = \mathfrak{Re}\{s_+(t)\} \quad \text{and} \quad \hat{s}(t) = \mathfrak{Im}\{s_+(t)\}. \tag{3}$$

- A signal $s(t)$ and its Hilbert transform $\hat{s}(t)$ have the same amplitude spectrum

$$|S(f)| = |\hat{S}(f)|, \tag{4}$$

where $|S(f)|$ and $|\hat{S}(f)|$ denote the Fourier transforms of $s(t)$ and $\hat{s}(t)$, respectively.

Using simple algebra one can easily show that the Fourier transform $S_+(f)$ of the pre-envelope signal $s_+(t)$ can be expressed in terms of the Fourier transform $S(f)$ of the bandpass signal $s(t)$ as

$$S_+(f) = \begin{cases} 0 & \text{for } f < 0 \\ S(0) & \text{for } f = 0 \\ 2S(f) & \text{for } f > 0, \end{cases} \tag{5}$$

which shows that $S_+(f)$ has no components with negative frequencies. Thus, for the bandpass signal $s(t)$ with generic amplitude spectrum shown in Fig. 3(a), the amplitude spectrum $|S_+(f)|$ of its corresponding pre-envelope signal $s_+(t)$ looks like the one shown in Fig. 3(b) and can be obtained through a shift in frequency by f_c of the amplitude spectrum $|\tilde{S}(f)|$ shown in Fig. 3(c) that corresponds to signal $\tilde{s}(t)$.

The signal $\tilde{s}(t)$ with amplitude spectrum shown in Fig. 3(c) is referred to as the *complex lowpass equivalent signal* of the bandpass signal $s(t)$, and the frequency shifting relationship between the $|S_+(f)|$ and $|\tilde{S}(f)|$ amplitude spectra,

$$S_+(f) = \tilde{S}(f - f_c) \tag{6}$$

translates into multiplication by a complex exponential of the complex lowpass equivalent signal in time domain, that is

$$s_+(t) = \tilde{s}(t)e^{j2\pi f_c t}. \tag{7}$$

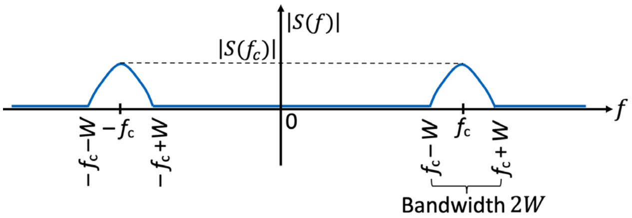
Noting that, by construction (3), the original bandpass signal $s(t)$ corresponds to the real part of the pre-envelope signal, we can now write the relationship between $s(t)$ and $\tilde{s}(t)$ as

$$s(t) = \mathfrak{Re}\{\tilde{s}(t)e^{j2\pi f_c t}\}. \tag{8}$$

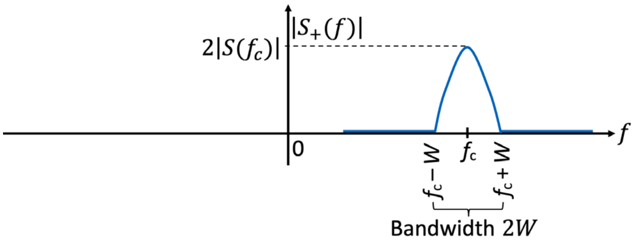
Expression (8) highlights the two components of the canonical representation of bandpass signals:

- The information content of bandpass signal $s(t)$, which is implied by the spectrum of its complex lowpass equivalent signal $\tilde{s}(t)$ with bandwidth $2W$, and
- The frequency band where the bandpass signal occurs, which is centered at f_c, the frequency of the complex exponential term.

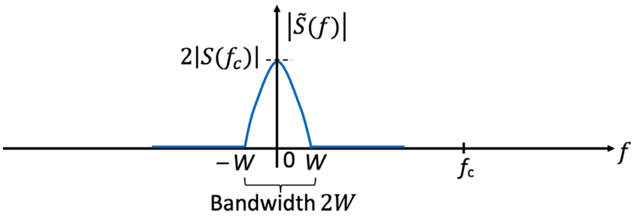
Thus, the canonical representation of bandpass signals enables their analysis in terms of complex lowpass equivalent signals and it is independent of the center frequency at which the bandpass signals occur. From a SDR perspective, the implication is that the transmitter can focus on implementing a modulation scheme for information transmission without considering the band of frequencies in which the modulated signal should be transmitted, while the receiver can extract the information contained in the bandpass signal by baseband processing of the complex lowpass equivalent signal.



(a) Amplitude spectrum for bandpass signal $s(t)$.



(b) Amplitude spectrum for pre-envelope signal $s_+(t)$.



(c) Amplitude spectrum for complex lowpass equivalent signal $\tilde{s}(t)$.

Fig. 3. Amplitude spectra for signals used in the canonical representation of bandpass signals.

B. The In-Phase and Quadrature Signal Components

The downside of the canonical representation of bandpass signals based on the complex lowpass equivalent signal is the fact that, due to its complex-valued nature, its characteristics cannot be directly visualized using measurement equipment such as an oscilloscope or spectrum analyzer. Nevertheless, the complex lowpass equivalent signal can be used to provide an alternative representation in terms of the two real-valued signals that make up its real and imaginary parts, $s_I(t)$ and $s_Q(t)$, respectively, which are referred to as the in-phase (I) and quadrature (Q) components of the bandpass signal. Thus, for bandpass signal $s(t)$ with complex lowpass equivalent signal $\tilde{s}(t)$ we have that

$$s_I(t) = \mathfrak{Re}\{\tilde{s}(t)\} \quad \text{and} \quad s_Q(t) = \mathfrak{Im}\{\tilde{s}(t)\}, \tag{9}$$

such that equation (8) can be rewritten as

$$s(t) = \mathfrak{Re}\{[s_I(t) + js_Q(t)]e^{j2\pi f_c t}\}. \tag{10}$$

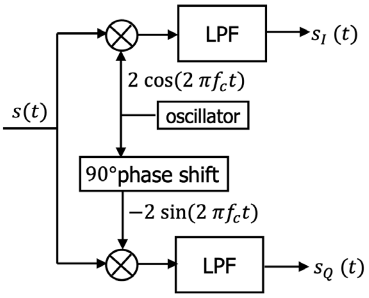


Fig. 4. Obtaining the I and Q components of a bandpass signal.

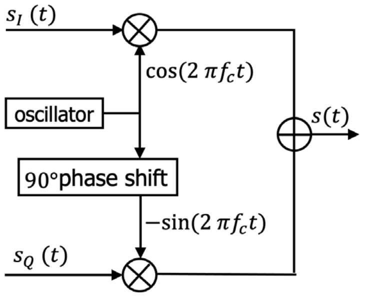


Fig. 5. Synthesizing a bandpass signal from its I and Q components.

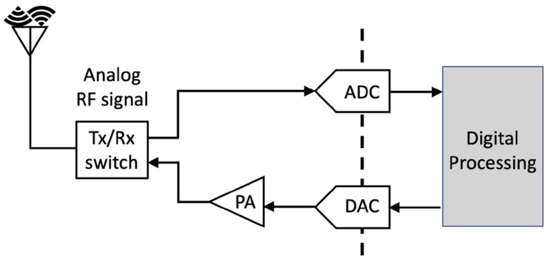


Fig. 6. Ideal architecture of a SDR. The ADC and DAC are performed on the RF signal, and a power amplifier (PA) is used on the transmit side to ensure desired RF transmit power level.

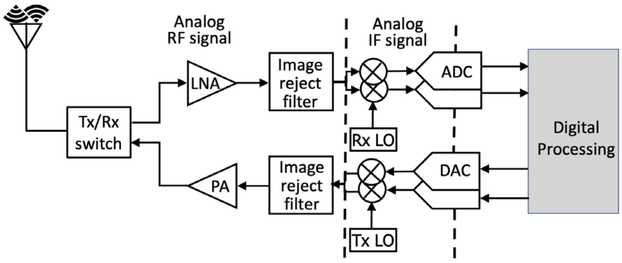


Fig. 7. The architecture of existing SDRs. The RF signal is shifted to IF where ADC and DAC are performed. A low noise amplifier (LNA) is used on the receive side prior to IF, with a PA on the transmit side.

Using Euler's formula $e^{j2\pi f_c t} = \cos(2\pi f_c t) + j\sin(2\pi f_c t)$ in (10) we obtain the equivalent expression of $s(t)$ in terms of the I and Q components as

$$s(t) = s_I(t)\cos(2\pi f_c t) - s_Q(t)\sin(2\pi f_c t), \qquad (11)$$

which, similar to (8), provides the I/Q signals as an alternative way of characterizing the information content of bandpass signal $s(t)$, in terms of real-valued signals $s_I(t)$ and $s_Q(t)$, both with lowpass spectrum and bandwidth $2W$ as implied by the spectrum of $\tilde{s}(t)$.

Given the bandpass signal $s(t)$, its I and Q components can be obtained by multiplying it with $\cos(2\pi f_c t)$ and $-\sin(2\pi f_c t)$ respectively, followed by lowpass filtering (LPF), as shown in Fig 4, where the bandwidth of the LPF used is the same for both $s_I(t)$ and $s_Q(t)$ and is equal to the bandwidth W of the complex lowpass equivalent signal $\tilde{s}(t)$.

Alternatively, when the I and Q components of the bandpass signal are available, the bandpass signal $s(t)$ can be synthesized by directly implementing (11) as shown in Fig. 5.

C. Heterodyning and Frequency Down/Up-Conversion

Heterodyning, also referred to as frequency changing or mixing [38, Section 2.4], consists of multiplying a bandpass signal $s_1(t)$ with center frequency f_{c1} with a sinusoidal signal produced by a local oscillator with frequency f_{LO} followed by an appropriate bandpass filtering operation to produce a new bandpass signal $s_2(t)$ with a different center frequency $f_{c2} = f_{c1} \pm f_{LO}$. When $f_{c2} < f_{c1}$ the operation is referred to as *frequency-down conversion*, and when $f_{c2} > f_{c1}$ the operation is referred to as *frequency-up conversion*.

A major application of heterodyning is in the superheterodyne receiver [38, Section 2.9], which has been used for decades in the reception of radio signals by converting the received RF signal to an intermediate frequency (IF) where it would be further filtered and amplified before being processed by the demodulator to extract the information. The same approach that consists of processing at an IF, which is used in the superheterodyne receiver, is also used in SDRs [42] as well as in other applications such as the emerging 5G systems [43].

IV. THE SDR ARCHITECTURE

Ideally, a SDR platform should perform all processing digitally, with the conversions from analog-to-digital and digital-to-analog occurring directly on the RF signal at the antenna, as shown in Fig. 6. This architecture, which corresponds to the one envisioned by Mitola [2], is applicable currently to lower frequencies, mostly in the high-frequency (HF) and very high frequency (VHF) bands, due to limitations of existing ADC and DAC converter technology. We note that, to support operation over a wide RF range, from HF (tens of MHz) to super high frequencies (SHF) up to 6 GHz, the ADC and DAC used must have extremely high resolutions with a wide dynamic range. This is a critical requirement that was acknowledged in the microelectronics and IC community from the early days of SDRs [44].

A. The IF Stage

During the late 1990s and early 2000 years, it was realized that capabilities of integrated ADCs and DACs [45] were increasing at a slower pace than those of other types of ICs, which were following Moore's law [46]. Thus, alternative architectures for SDRs had to be pursued. The solution was found in the form of the SDR architecture shown in Fig 7, which is present in current SDRs and uses an IF processing stage where the ADC and DAC take place [42].

The IF stage bridges the RF front end of the SDR with its digital processing core where information is extracted from received signals or embedded in signals synthesized for transmission:

- On the receiver side, the IF stage translates the analog RF signal to the IF and enables a subsampling (or sampling translation) approach for ADC [47], which takes advantage of the fact that a conventional ADC can digitize an analog signal with a compact spectrum, as is the case with bandpass signals, using an undersampling approach relative to the RF or IF frequencies, but oversampling with respect to the information bandwidth of the bandpass signal [48], [49]. The RF front end of the SDR may include on the receiver side a LNA to strengthen weak signals without significantly impacting the signal-to-noise ratio (SNR) [50].
- On the transmitter side, the I and Q components of the modulated signal are digitally synthesized and translated to IF, followed by DAC and analog frequency translation to RF. The RF front end of the SDR includes on the transmitter side also a PA, which is a critical component as it impacts the power consumption and overall cost of the SDR [51].

We note that in current implementations of SDR platforms, the two analog stages of the SDR (the RF front end and the IF) are usually integrated on the same chip. For example, this is the case with Ettus Research bus series SDR platforms, which use AD 936x agile transceiver chips [52][3], as well as with the RTL-SDR receive only SDR, which uses the R820T tuner [53].

B. Digital Processing

On the digital processing side of a SDR platform one can also distinguish two distinct stages, the digital front end and the baseband processing stage [54].

The digital front end performs two functions [55]:

- Sample rate conversion, which adapts the sampling rate corresponding to the IF stage and the sampling rate at which the digital baseband processing is accomplished in the subsequent signal processing stage.
- Channelization, which includes channel filtering to extract specific frequency bands and conversion between the digital IF and baseband.

The baseband processing stage is where the actual operations related to the communication signal synthesis/analysis takes place and covers functions that include physical layer processing such as implementing modulation/demodulation and error correction encoding/decoding as well as MAC layer functions that connect the physical layer with the upper protocol layers.

In existing implementations of SDR platforms, the digital front end is implemented on an FPGA that can be co-located on the same board as the RF front end and IF chip or on a different board, while baseband processing is accomplished on a general purpose processor (GPP), which, in most cases, is a host computer programmed to run specific applications handling the digital stream of I/Q data. In this case, the FPGA includes also the communication interface between the digital front end and the host computer, which can be over Universal Serial Bus (USB), Ethernet, or PCIe [56].

Baseband processing can also be implemented on the same FPGA as the digital front end if the FPGA fabric has sufficient resources available, which is the case with high-end FPGAs and Systems-on-a-Chip (SoCs) that integrate powerful FPGAs with ARM processing cores on the same IC enabling standalone SDR platforms that can be deployed in the field [54].

V. SDR CHOICES AND PROGRAMMING

A wide range of SDR platforms are currently available on the commercial market, and providing a comprehensive listing of all SDR choices is beyond the scope of the paper. Rather, we would like to highlight several SDR platforms that have attracted the attention of a wider audience and have been used for wireless systems research, development, and teaching in industry and academic settings.

A. The Universal Software Radio Peripheral – USRP

The USRP is among the most widely used SDR platforms for wireless research and teaching [57], being available in many flavors [58]:

- At the low end, the USRP family has the bus series, which provides a fully integrated, single board SDR platform with continuous frequency coverage from 70 MHz to 6 GHz and up to 56 MHz of real-time bandwidth.
- At the high end of the USRP spectrum, the X series offers a high-performance scalable architecture that includes large user-programmable FPGAs and the RF front end covering the range from DC to 6 GHz with up to 120 MHz of baseband bandwidth.
- The top member of the USRP family, the X410, features a Zynq Ultrascale RFSoC that includes a quad-core ARM Cortex-A53 processor for standalone applications and is designed for frequencies from 1 MHz to 7.2 GHz with 4 independent transmit/receive channels and a two-stage superheterodyne architecture, being capable of supporting up 400 MHz of instantaneous bandwidth on each channel.

Over the past decade the USRP SDRs have become a leading choice for teaching fundamental concepts in communication systems and for hands-on experimentation with wireless communications, and many references are available in the published literature [59], [60], [61], [62].

B. The RTL-SDR Receiver

Another popular SDR choice is the RTL-SDR, which is a receive only platform, with the name acronym coming from the use of the RealTek RTL2832U chip for its digital front-end. Different versions of the RTL-SDR are available, that are distinguished by the different tuner chips used to receive the RF signal, which include [53]:

- The Rafael Micro R820T covers the frequency range from 24 MHz to 1.766 GHz and uses an IF processing stage to provide a down-converted IF signal with a bandwidth of about 6 MHz to the RTL2832U, which extracts the digital I/Q data.

---

[3] The AD 9364 implements a single transceiver and is used in the USRP B200. The AD9361 provides two independent transceivers and is used in the USRP B210.

- The Elonics E4000 operates from 52 MHz to 2.2 GHz, with a gap between 1.1 GHz and 1.25 GHz, and has no IF stage, converting the analog RF signal to a baseband one with a roughly 10 MHz bandwidth and feeding the analog I/Q signals to the RTL2832U, which samples them to extract the I/Q data.

The I/Q data at the output of the RTL2832, which has a bandwidth of about 2.8 MHz and is encoded on 8 bits, is provided over USB to the host computer for baseband processing.

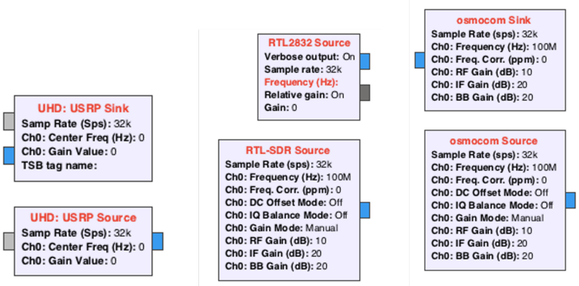
The RTL-SDR receiver is very affordable, with kits that include the RTL-SDR USB dongle, antennas and cables, available online for very low prices. Despite its lower capabilities in terms of frequency range or bandwidth when compared to the USRP, the RTL-SDR is a main choice for radio enthusiasts, with numerous projects using it featured on the internet [63]. In addition, reference book [53], which can be used for hands-on teaching SDR concepts, has also contributed to the popularity of the RTL-SDR receiver.

C. Other SDR Choices

The ADALM-PLUTO is a SDR platform that aims academic teaching and is marketed as "*an active learning module*" that "*helps introduce electrical engineering students to the fundamentals of SDR, RF, and wireless communications*" [64]. Its RF front end features an AD9363 highly integrated RF agile transceiver with the digital front end using a Zynq FPGA, operating over the frequency range from 325 MHz to 3.8 GHz with up to 20 MHz of real-time bandwidth and communicating with the host computer over USB. The ADALM-PLUTO is supported in MATLAB and Simulink, and is a good candidate for integrating it in the electrical engineering curriculum to support teaching a wide range of concepts related to RF and wireless systems, digital communications and signal processing, or embedded systems [65], [66].

Two other SDRs have also been mentioned alongside the USRP in a recent study of SDR platforms that meet minimum specifications for existing wireless technologies [56], the HackRF One [67] and the Lime SDR [68]. They have also been used in academic projects [69], [70] and are also popular with radio enthusiasts, with various projects using them also featured online [63]. Their main characteristics are:

- The HackRF operates over the frequency range from 1 MHz to 6 GHz with an instantaneous bandwidth of 20 MHz, communicating with the host computer over USB 2.0. It features a MAX2837 chip for the RF front end, which has no IF and converts the RF signals to baseband, followed by the MAX5864 ADC/DAC and the LPC4300 series ARM Cortex-M4 microcontroller for its digital front end.
- The Lime SDR operates over the frequency range from 100 kHz to 3.8 GHz with an instantaneous bandwidth of 160 MHz, communicating with the host computer over USB 3.0. The RF front end uses the LMS7002M field programmable RFIC dual transceiver, which supports 2×2 MIMO configurations and has on chip integrated 12-bit ADC and DAC to provide the digital I/Q signal data to an Alterra Cyclone IV FPGA.



UHD: USRP Sink
Samp Rate (Sps): 32k
Ch0: Center Freq (Hz): 0
Ch0: Gain Value: 0
TSB tag name:

UHD: USRP Source
Samp Rate (Sps): 32k
Ch0: Center Freq (Hz): 0
Ch0: Gain Value: 0

RTL2832 Source
Verbose output: On
Sample rate: 32k
Frequency (Hz):
Relative gain: On
Gain: 0

RTL-SDR Source
Sample Rate (sps): 32k
Ch0: Frequency (Hz): 100M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

osmocom Sink
Sample Rate (sps): 32k
Ch0: Frequency (Hz): 100M
Ch0: Freq. Corr. (ppm): 0
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

osmocom Source
Sample Rate (sps): 32k
Ch0: Frequency (Hz): 100M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

(a) (b) (c)

Fig. 8. GNU Radio blocks required for programming SDR platforms: (a) Source and sink blocks for USRP SDR; (b) Different versions of source blocks for RTL-SDR; (c) Osmocom source and sink blocks that can be used with various SDRs.

Finally, the Myriad RF SDR board is also worth mentioning [71]. The original Myriad–RF 1 is a multi-band, multi-standard RF module from Lime Microsystems that is based on their LMS6002D transceiver IC, featuring one RF broadband output and one RF broadband input with digital baseband interface. The Myriad-RF 1 contained everything needed for it to be connected to baseband chipsets, FPGAs, or to run in standalone mode. Currently, the Myriad-RF 1 design has been adapted for use with the Novena open hardware computing platform, which can be configured for embedded applications [72].

D. SDR Programming

Software development toolkits available to program SDR platforms include GNU Radio [73], MATLAB and Simulink [74], and LabVIEW [75]. They provide graphical interfaces in which blocks performing specific signal processing functions are interconnected to implement various physical and MAC layer functions on SDRs and to run standalone applications. We note that GNU Radio is open source and free to use, while MATLAB and Simulink as well as LabVIEW require a valid license to be able to use them.

Behind every block available in GNU Radio there is a Python script supporting it. Due to the open source nature of GNU Radio, code can be modified by the user as needed by adding out-of-tree (OOT) modules containing new functionalities and blocks [76], thus effectively leveraging the power of open-source SDR community [73]. We note that GNU Radio was originally designed for use with the open-source Linux operating system and the Ubuntu distribution of Linux continues to be preferred for developing applications with GNU Radio by SDR developers. Nevertheless, installation options for running GNU Radio under Windows and Mac operating systems are also available, albeit taking advantage of the open-source features of GNU Radio such as adding OOT modules may not be as friendly under these operating systems as under Ubuntu. We also note that, while GNU Radio can be used without any hardware as a simulation and development environment, its power lies in the ability to simulate complete transmit/receive chains that include RF, analog, and other relevant impairments encountered in practical systems and implementations. Thus, using GNU Radio with specific SDR platforms requires that the manufacturers provide support for

GNU Radio to ensure that blocks corresponding to their specific SDR platforms such as the ones illustrated in Fig. 8 are available for use:

- Sink blocks represent transmitters and correspond to the RF front end of the SDR platform that synthesizes the RF signal. The input to a SDR sink is in general a complex variable whose real and imaginary parts, respectively, are sampled versions of the I and Q components of the RF signal that is transmitted by the SDR, with the sampling rate specified as one of the parameters of the sink block.
- Source blocks represent receivers and correspond to the RF front end of the SDR platform that acquires the RF signal. The output of a SDR source is in general a complex variable whose real and imaginary parts, respectively, are sampled versions of the I and Q components of the RF signal that is acquired by the SDR, with the sampling rate specified as one of the parameters of the source block.

Programming SDR platforms using MATLAB and Simulink requires also the Communications Toolbox, which needs to be included with the MATLAB and Simulink license to be available for use. In addition, hardware support packages specific to the SDR platform that needs to be programmed should be installed. These provide Simulink blocks similar to the source and sink blocks in GNU Radio, which communicate with external SDR devices to process live radio signals captured over the air. Currently, the USRP, the RTL-SDR, and the ADALM-PLUTO are supported with MATLAB, but the HackRF One and the Lime SDR are not [74]. Since MATLAB is not open source, the blocks available for SDR programming cannot be modified by the users. However, free open-source MATLAB and Simulink code published by users is available on the MATLAB Central File Exchange [77].

Programming using LabVIEW is currently limited to the NI and Ettus USRP SDRs, and other SDR platforms are not officially supported. However, some examples of using LabVIEW with RTL-SDR can be found by searching the NI Community website [78].

VI. CASE STUDIES

We illustrate the use of SDR platforms with two case studies that have been completed in recent years in academic projects outside of a formal course on wireless communications:

- In the first project a Lime SDR platform is used as receiver to collect RF measurements for an empirical characterization of man-made noise in the 900 MHz frequency band [70]. This study demonstrates one of the many applications of SDR platforms, to replace conventional equipment used for performing RF measurements, which leads to lighter and more portable system and is beneficial for reducing overall system costs.
- In the second project an experimental study of the RF transmit power for a USRP B200 is presented [79]. The study is important since it highlights the need to test SDR platforms prior to using them in practical implementations, to confirm the RF power level at which they are programmed to transmit, and to understand dependence of RF power on frequency and other parameters, such as transmit gain(s) for example, which may be available when programming the SDRs.

Fig. 9. Full size van converted into a mobile platform to perform noise measurements in the early 1990s [70].

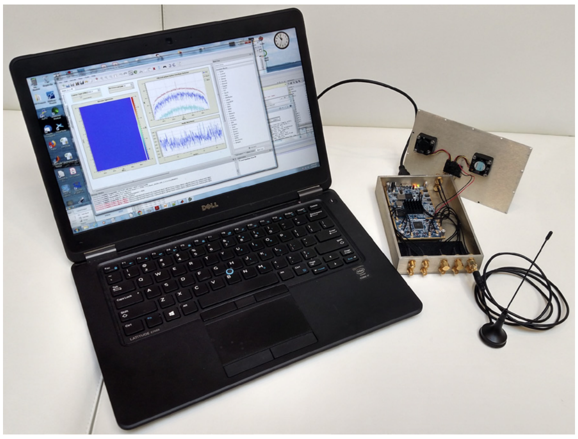


Fig. 10. The mobile platform used to perform noise measurements in [70].

A. Using SDR Platforms for Empirical Noise Characterization

The items needed to perform an empirical characterization of impulsive noise include a radio receiver, a spectrum analyzer, a logarithmic detector, a digital oscilloscope, and a computer. We note that performing an empirical noise characterization study during the early 1990s required access to a well-equipped lab dedicated to communication systems where all the items, which had significant price tags at the time, had to be available. Furthermore, in order to incorporate this equipment into a mobile platform a full size van had to be converted into a measurement vehicle as shown in Fig. 9.

By contrast, the setup used for taking measurements of impulsive noise these days can be accomplished using inexpensive components that can be acquired with the limited budget of an undergraduate research project, consisting of a SDR platform such as the Lime SDR, along with an average laptop computer as seen in Fig. 10. The SDR platform is configured as receiver providing access to the I/Q noise data, while all of the other items (the logarithmic detector, the spectrum analyzer, and the digital oscilloscope) are integrated in the signal processing software running on the laptop.

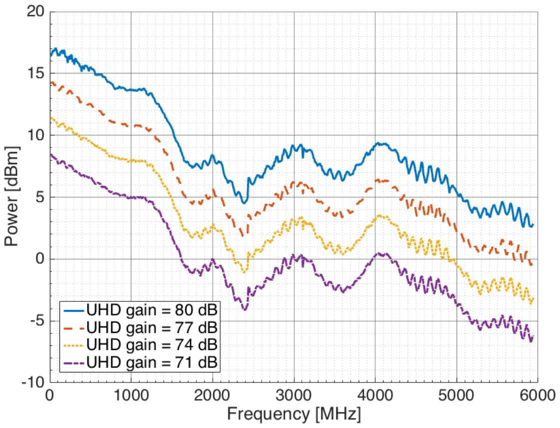
The specific application presented in [70] focuses on a narrowband system with RF bandwidth of 250 Hz operating in the 900 MHz band, but the setup can be easily adapted to other applications due to the versatility of the Lime SDR platform, which operates over a wide range of frequencies, covering the HF, VHF and UHF bands. Thus, the measurement setup in [70] requires minimal software changes to be applied to impulsive noise measurements and characterization in other scenarios, such as complementing the numerical simulation results in [80] with actual RF measurements of the noise radiated by a microwave oven over a narrowband channel with 300 kHz bandwidth at 2 GHz, similar to the one in [81]. Alternatively, the same system in [70] can be adapted to characterize wideband UHF digital TV channels with a bandwidth of 10 MHz in the 700 MHz band [82].

The Lime SDR, or other similar SDR platform, can also be used in transmit mode to synthesize impulsive noise signals using computer generated I and Q components as outlined in [83]. Accomplishing this requires in essence developing the software for generating the I and Q components of the artificial impulsive noise along with programming the SDR platform to transmit it over the frequency band of interest, and can be useful for testing purposes in a lab setup.

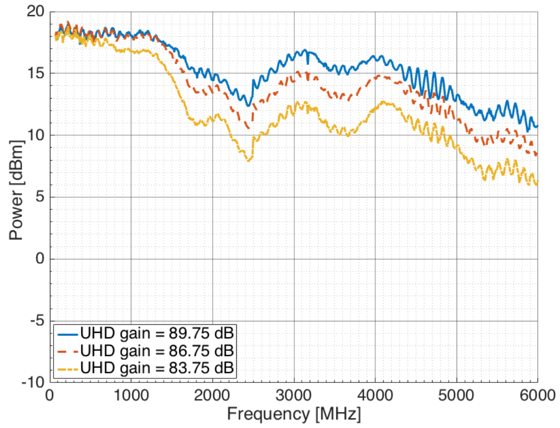
The noise measurement approach presented in [70] allows RF site-surveyors to accumulate noise data and to identify best and worst-case noise scenarios by using a lightweight and highly portable battery-powered measurement setup that includes a SDR platform and a laptop computer. The setup can be easily adapted to perform measurements of broadband or narrowband electromagnetic interference emissions, which are usually performed by experts at nationally recognized testing laboratories. Such measurements can be both expensive and time consuming, since they require specialized personnel and test fixtures, as well as hours of calibration and measurements to produce highly accurate certified results. However, when stringent accuracy and certification are not absolutely necessary, as may be the case with consumer-type applications, taking advantage of the versatility of SDR platforms, to establish dedicated systems for electromagnetic compatibility testing can significantly reduce cost, while still providing useful information for system design.

B. Transmit Power Variation for USRP B200 SDR

The study of transmit RF power in [79] is motivated by the fact that the USRP SDRs are not calibrated devices that can be used for measurement and/or testing, and USRP data sheets are vague when it comes to the specification of the RF transmit power level. A USRP B200 is considered in [79], for which the RF specifications mention only that its RF transmit power is above 10 dBm [52]. Furthermore, the transmit power of the USRP B200 is programmable through the transmit gain parameter in the USRP Hardware Driver (UHD), but exact specification of RF power as a function of the UHD gain is elusive. Nevertheless, precise knowledge of the RF transmit power is desirable for both experimentation and practical implementations, in particular for radio links with low margins such as, for example, those occurring in satellite communications [84], where every dB matters.



(a) Lower UHD gain values.



(b) Higher UHD gain values.

Fig. 11. Measured output RF power for USRP B200 SDR [79].

To perform the measurements of transmit RF power in [79] the USRP B200 was programmed using MATLAB and Simulink to transmit tones with frequency starting from 70 MHz to 6 GHz in 10 MHz increments. The USRP was configured to transmit using various UHD gain settings with values between 70 dB and the maximum allowed UHD value of 89.75 dB, and the RF power of the corresponding transmitted signal was recorded using a spectrum analyzer. A separate Matlab script is run to collect the power measurements automatically using the Instrument Control Toolbox, and the results obtained are separated into two categories as follows:

- The first category, shown in Fig. 11(a), includes lower UHD gains with values starting at 71 dB and increasing in 3 dB increments to 80 dB.
- The second category, shown in Fig. 11(b), includes high UHD gains with values starting at 83.75 dB and increasing in 3 dB increments to a UHD gain value of 89.75 dB, which is the maximum allowed UHD gain setting for the USRP B200.

The experimental results show that, with the UHD gain set at the maximum allowed value of 89.75 dB, the transmit power of the USRP B200 is indeed above 10 dBm over all its operating range as specified by the manufacturer [52]. Results in Fig. 11(b) also show that for UHD gains above 83 dB, the transmitter appears to be outside its linear operating range as changes in the UHD gain value do not reflect linearly in the transmit power variations. For UHD gains below 80 dB results in Fig. 11(a) show that the transmitter operation is linear as one can observe the clear 3 dB separation between neighboring curves, which is expected from the UHD gain settings under linear operation. It is thus conceivable to extrapolate the measurements done for these UHD gain values (71 dB, 74 dB, 77 dB, and 80 dB) to estimate transmit power levels for UHD gains below 70 dB.

Another interesting observation that can be based on the plots in Fig. 11 is that the transmit RF power of the USRP B200 varies with frequency. For example, looking at the curves for lower UHD gain values in Fig. 11(a), one can notice a variation of about 12 dB between the output power values at lower frequencies (70 MHz – 100 MHz) and those at higher frequencies (5.5 GHz – 6 GHz). The decrease in the transmit RF power of the USRP B200 observed when the operating frequency increases is likely due to the fact that the transmitter impedance is better matched at lower frequencies than at higher ones, and should be considered when the USRP is programmed to transmit at a specific power level in a given frequency band.

The specific application setup presented in [79], which focuses on a USRP B200 SDR, can be easily adapted to other SDR platforms. If MATLAB hardware support package for the SDR platform is not available, which is currently the case with the HackRF One and the Lime SDR platforms, then the SDR may be programmed to sweep its operating range using GNU Radio, while collecting the power measurements can still be accomplished using the MATLAB script that calls functions in the Instrument Control Toolbox.

VII. CONCLUSIONS

In this paper we provided a comprehensive introduction to SDRs, which are the building blocks of modern communication systems and are used in research, development, implementation, and teaching of wireless communications. The paper starts with a brief review of drivers and enabling technologies that contributed to the advancement of SDR platforms and to their ubiquitous presence in current and emerging wireless systems and networks, mentioning also future trends that will continue to keep SDRs at the forefront of communication technologies. Next, a brief theoretical background is given, reviewing bandpass signal representations in terms of I/Q components along with heterodyning for frequency up- and down-conversions, which are concepts that are essential to the understanding of SDR operation. The current architecture of SDR platforms is then presented, with details on the various processing stages in SDRs that include the RF front end, the IF processing, the digital front end, and the baseband processing. Finally, the paper presents several SDR platforms that have emerged as preferred choices for research, development, teaching, and radio enthusiast projects, reviews SDR programming alternatives, and presents two case studies demonstrating SDR applications and uses.

We are confident that, with the continued interest for SDRs in existing and future generations of wireless communication systems, the paper will serve as a useful reference for wireless researchers and developers working in the industry as well as for instructors teaching courses on wireless communications.

ACKNOWLEDGEMENT

The work of Dimitrie Popescu was supported by a Fulbright US Scholar grant awarded for the Spring 2022 semester.

REFERENCES

[1] J. Mitola, “Software Radios: Survey, Critical Evaluation and Future Directions,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 8, no. 4, pp. 25–36, 1993. **DOI**: 10.1109/62.210638

[2] ——, “The Software Radio Architecture,” *IEEE Communications Magazine*, vol. 33, no. 5, pp. 26–38, 1995. **DOI**: 10.1109/35.393001

[3] Wireless Innovation Forum, “What is a Software Defined Radio,” available online at https://www.wirelessinnovation.org/assets/documents/SoftwareDefinedRadio.pdf, accessed: April 24, 2022.

[4] National Instruments, “Software Defined Radio: Past, Present, and Future,” available online at https://www.ni.com/hu-hu/innovations/white-papers/17/software-defined-radio--past--present--and-future.html, accessed: April 24, 2022.

[5] J. Melby, “JTRS and the Evolution Toward Software-Defined Radio,” in *Proceedings 2002 IEEE Military Communications Conference*, vol. 2, pp. 1286–1290, Anaheim, CA, 2002. **DOI**: 10.1109/MILCOM.2002.1179664

[6] F. Vergari, “Software-Defined Radio: Finding Its Use in Public Safety,” *IEEE Vehicular Technology Magazine*, vol. 8, no. 2, pp. 71–82, 2013. **DOI**: 10.1109/MVT.2013.2252292

[7] M. Öner and F. Jondral, “Air Interface Recognition for a Software Radio System Exploiting Cyclostationarity,” in *Proceedings 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2004)*, vol. 3, pp. 1947–1951, Barcelona, Spain, September 2004. **DOI**: 10.1109/PIMRC.2004.1368338

[8] O. A. Dobre, “Signal Identification for Emerging Intelligent Radios: Classical Problems and New Challenges,” *IEEE Instrumentation Measurement Magazine*, vol. 18, no. 2, pp. 11–18, 2015. **DOI**: 10.1109/MIM.2015.7066677

[9] G. Retz, H. Shanan, K. Mulvaney, S. O’Mahony, M. Chanca, P. Crowley, C. Billon, M. Kalimuddin Khan, J. J. Lopez Orive, and P. Quinlan, “Radio Transceivers for Wireless Personal Area Networks Using IEEE 802.15.4,” *IEEE Communications Magazine*, vol. 47, no. 9, pp. 150–158, 2009. **DOI**: 10.1109/MCOM.2009.5277469

[10] M. Cummings and S. Haruyama, “FPGA in the Software Radio,” *IEEE Communications Magazine*, vol. 37, no. 2, pp. 108–112, February 1999. **DOI**: 10.1109/35.747258

[11] S. Balasubramanian, S. Boumaiza, H. Sarbishaei, T. Quach, P. Orlando, J. Volakis, G. Creech, J. Wilson, and W. Khalil, “Ultimate Transmission,” *IEEE Microwave Magazine*, vol. 13, no. 1, pp. 64–82, 2012. **DOI**: 10.1109/MMM.2011.2173983

[12] L. Pucker, “Is There Really Such a Thing as a “DSP” Anymore?” *IEEE Communications Magazine*, vol. 44, no. 9, pp. 34–36, 2006. **DOI**: 10.1109/MCOM.2006.1705976

[13] C. Dick and F. Harris, “FPGA Signal Processing Using Sigma-Delta Modulation,” *IEEE Signal Processing Magazine*, vol. 17, no. 1, pp. 20–35, 2000. **DOI**: 10.1109/79.814644

[14] P. Burns, Software Defined Radio for 3G. Norwood, MA: Artech House, 2002.

[15] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas, “LTE-Advanced: Next-Generation Wireless Broadband Technology,” *IEEE Wireless Communications*, vol. 17, no. 3, pp. 10–22, 2010. **DOI**: 10.1109/MWC.2010.5490974

[16] R. Schneiderman, “LTE Base Stations, Mobile Devices Flood Telecom, Consumer Markets,” *IEEE Signal Processing Magazine*, vol. 29, no. 4, pp. 9–14, 2012. **DOI**: 10.1109/MSP.2012.2186185

[17] Y. Kyung, T. M. Nguyen, K. Hong, J. Park, and J. Park, “Software Defined Service Migration Through Legacy Service Integration Into 4G Networks and Future Evolutions,” *IEEE Communications Magazine*, vol. 53, no. 9, pp. 108–114, 2015. **DOI**: 10.1109/MCOM.2015.7263353

[18] T. Ulversøy, “Software Defined Radio: Challenges and Opportunities,” *IEEE Communications Surveys & Tutorials*, vol. 12, no. 4, pp. 531–550, 2010. **DOI**: 10.1109/SURV.2010.032910.00019

[19] B. van Liempd, J. Borremans, E. Martens, S. Cha, H. Suys, B. Verbruggen, and J. Craninckx, “A 0.9 V 0.4–6 GHz Harmonic Recombination SDR Receiver in 28 nm CMOS With HR3/HR5 and IIP2 Calibration,” *IEEE Journal of Solid-State Circuits*, vol. 49, no. 8, pp. 1815–1826, 2014. **DOI**: 10.1109/JSSC.2014.2321148

[20] G. Wang, B. Yin, K. Amiri, Y. Sun, M. Wu, and J. R. Cavallaro, “FPGA Prototyping of a High Data Rate LTE Uplink Baseband Receiver,” in *Proceedings of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, pp. 248–252, Pacific Grove, CA, November 2009. **DOI**: 10.1109/ACSSC.2009.5470112.

[21] R. G. Machado and A. M. Wyglinski, “Software-Defined Radio: Bridg- ing the Analog–Digital Divide,” *Proceedings of the IEEE*, vol. 103, no. 3, pp. 409–423, 2015. **DOI**: 10.1109/JPROC.2015.2399173

[22] C. Erdmann, D. Lowney, A. Lynam, A. Keady, J. McGrath, E. Cullen, D. Breathnach, D. Keane, P. Lynch, M. De La Torre, R. De La Torre, P. Lim, A. Collins, B. Farley, and L. Madden, “A Heterogeneous 3D-IC Consisting of Two 28 nm FPGA Die and 32 Reconfigurable High-Performance Data Converters,” *IEEE Journal of Solid-State Circuits*, vol. 50, no. 1, pp. 258–269, 2015. **DOI**: 10.1109/JSSC.2014.2357432

[23] F. Gringoli, P. Patras, C. Donato, P. Serrano, and Y. Grunenberger, “Performance Assessment of Open Software Platforms for 5G Prototyping,” *IEEE Wireless Communications*, vol. 25, no. 5, pp. 10–15, October 2018. **DOI**: 10.1109/MWC.2018.1800049

[24] K. Lin, W. Wang, X. Wang, W. Ji, and J. Wan, “QoE-Driven Spectrum Assignment for 5G Wireless Networks Using SDR,” *IEEE Wireless Communications*, vol. 22, no. 6, pp. 48–55, December 2015. **DOI**: 10.1109/MWC.2015.7368824

[25] X. Xiong, W. Xiang, K. Zheng, H. Shen, and X. Wei, “An Open Source SDR-Based NOMA System for 5G Networks,” *IEEE Wireless Communications*, vol. 22, no. 6, pp. 24–32, December 2015. **DOI**: 10.1109/MWC.2015.7368821

[26] S. Gökceli, P. P. Campo, T. Levanen, J. Yli-Kaakinen, M. Turunen, M. Aléln, T. Riihonen, A. Palin, M. Renfors, and M. Valkama, “SDR Prototype for Clipped and Fast-Convolution Filtered OFDM for 5G New Radio Uplink,” *IEEE Access*, vol. 8, pp. 89 946–89 963, May 2020. **DOI**: 10.1109/ACCESS.2020.2993871

[27] Y. Liu, C. Li, X. Xia, X. Quan, D. Liu, Q. Xu, W. Pan, Y. Tang, and K. Kang, “Multiband User Equipment Prototype Hardware Design for 5G Communications in Sub-6-GHz Band,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 67, no. 7, pp. 2916–2927, July 2019. **DOI**: 10.1109/TMTT.2019.2904234

[28] G. P. Fettweis, “5G and the Future of IoT,” in *Proceedings 42nd European Solid-State Circuits Conference – ESSCIRC 2016*, Lausanne, Switzerland, September 2016., pp. 21–24. **DOI**: 10.1109/ESSCIRC.2016.7598234

[29] J. Liang, H. Chen, and S. C. Liew, “Design and Implementation of Time-Sensitive Wireless IoT Networks on Software-Defined Radio,” *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 2361–2374, February 2022. **DOI**: 10.1109/JIOT.2021.3094667

[30] P. Solic, Z. Blazevic, M. Skiljo, L. Patrono, R. Colella, and J. J. P. C. Rodrigues, “Gen2 RFID as IoT Enabler: Characterization and Performance Improvement,” *IEEE Wireless Communications*, vol. 24, no. 3, pp. 33–39, June 2017. **DOI**: 10.1109/MWC.2017.1600431

[31] G. Saxl, L. Görtschacher, T. Ussmueller, and J. Grosinger, “Software-Defined RFID Readers: Wireless Reader Testbeds Exploiting Software-Defined Radios for Enhancements in UHF RFID Systems,” *IEEE Microwave Magazine*, vol. 22, no. 3, pp. 46–56, March 2021. **DOI**: 10.1109/MMM.2020.3042408

[32] SEMTECH, “What is LoRa®?” available online at https://www.semtech.com/lora/what-is-lora, accessed: May 23, 2022.

[33] R. Ghanaatian, O. Afisiadis, M. Cotting, and A. Burg, “Lora Digital Receiver Analysis and Implementation,” in *Proceedings 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1498–1502., Brighton, UK, May 2019. **DOI**: 10.1109/ICASSP.2019.8683504

[34] C. Bernier, F. Dehmas, and N. Deparis, “Low Complexity LoRa Frame Synchronization for Ultra-Low Power Software-Defined Radios,” *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 3140–3152, May 2020. **DOI**: 10.1109/TCOMM.2020.2974464

[35] P. I. Theoharis, R. Raad, F. Tubbal, M. U. Ali Khan, and S. Liu, “Software-Defined Radios for CubeSat Applications: A Brief Review and Methodology,” *IEEE Journal on Miniaturization for Air and Space Systems*, vol. 2, no. 1, pp. 10–16, March 2021. **DOI**: 10.1109/JMASS.2020.3032071

[36] M. De Sanctis, E. Cianca, G. Araniti, I. Bisio, and R. Prasad, “Satellite Communications Supporting Internet of Remote Things,” *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 113–123, February 2016. **DOI**: 10.1109/JIOT.2015.2487046

[37] C. Gavrilă, V. Popescu, M. Alexandru, M. Murroni, and C. Sacchi, “An SDR-Based Satellite Gateway for Internet of Remote Things (IoRT) Applications,” *IEEE Access*, vol. 8, pp. 115423–115436, July 2020. **DOI**: 10.1109/ACCESS.2020.3004480

[38] S. Haykin, *Communication Systems*, 4th ed. NewYork, NY: John Wiley & Sons, Inc., 2001.

[39] G. Kolumban, T. I. Krebesz, and F. C. Lau, “Theory and Application of Software Defined Electronics: Design Concepts for the Next Generation of Telecommunications and Measurement Systems,” *IEEE Circuits and Systems Magazine*, vol. 12, no. 2, pp. 8–34, 2012. **DOI**: 10.1109/MCAS.2012.2193435

[40] G. Kolumban, “Software Defined Electronics: A Revolutionary Change in Design and Teaching Paradigm of RF Radio Communications Systems,” *ICT Express*, vol. 1, no. 1, pp. 44–54, 2015. **DOI**: 10.1016/S2405-9595(15)30021-7

[41] Wikipedia, “FM Broadcasting,” available online at https://en.wikipedia.org/wiki/FM_broadcasting, accessed: April 30, 2022.

[42] P. Cruz, N. B. Carvalho, and K. A. Remley, “Designing and Testing Software-Defined Radios,” *IEEE Microwave Magazine*, vol. 11, no. 4, pp. 83–94, June 2010. **DOI**: 10.1109/MMM.2010.936493

[43] A. Udalcovs, M. Levantesi, P. Urban, D. A. A. Mello, R. Gaudino, O. Ozolins, and P. Monti, “Total Cost of Ownership of Digital vs. Analog Radio-Over-Fiber Architectures for 5G Fronthauling,” *IEEE Access*, vol. 8, pp. 223 562–223 573, December 2020. **DOI**: 10.1109/ACCESS.2020.3044396

[44] B. Brannon, “Wideband Radios Need Wide Dynamic Range Converters,” Analog Dialogue, vol. 29, no. 2, pp. 11–12, April 1995, available online at https://www.analog.com/en/analog-dialogue/articles/wideband-radios-need-wide-dynamic-range-converters.html, accessed May 3, 2022.

[45] R. Walden, “Analog-to-Digital Converter Survey and Analysis,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 539–550, 1999. **DOI**: 10.1109/49.761034

[46] R. Schaller, “Moore’s Law: Past, Present and Future,” *IEEE Spectrum*, vol. 34, no. 6, pp. 52–59, 1997. **DOI**: 10.1109/6.591665

[47] A. A. Abidi, “The Path to the Software-Defined Radio Receiver,” *IEEE Journal of Solid-State Circuits*, vol. 42, no. 5, pp. 954–966, 2007. **DOI**: 10.1109/JSSC.2007.894307

[48] A. I. Hussein and W. B. Kuhn, “Bandpass ΣΔ Modulator Employing Undersampling of RF Signals for Wireless Communication,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 7, pp. 614–620, 2000. **DOI**: 10.1109/82.850420

[49] N. Beilleau, H. Aboushady, F. Montaudon, and A. Cathelin, “A 1.3V 26mW 3.2GS/s Undersampled LC Bandpass ΣΔ ADC for a SDR ISM-band Receiver in 130nm CMOS,” in *Proceedings 2009 IEEE Radio Frequency Integrated Circuits Symposium*, Boston, MA, June 2009., pp. 383–386., **DOI**: 10.1109/RFIC.2009.5135563

[50] A. Aneja and X. J. Li, “Multiband LNAs for Software-Defined Radios: Recent Advances in the Design of Multiband Reconfigurable LNAs for SDRs in CMOS, Microwave Integrated Circuits Technology,” *IEEE Microwave Magazine*, vol. 21, no. 7, pp. 37–53, July 2020. **DOI**: 10.1109/MMM.2020.2985189

[51] F. M. Ghannouchi, “Power Amplifier and Transmitter Architectures for Software Defined Radio Systems,” *IEEE Circuits and Systems Magazine*, vol. 10, no. 4, pp. 56–63, fourth quarter 2010. **DOI**: 10.1109/MCAS.2010.938639

[52] Ettus Research, "The USRP B200/B210/B200mini/B205mini," available online at https://kb.ettus.com/B200/B210/B200mini/B205mini, accessed: May 2, 2022.

[53] R. W. Stewart, K. W. Barlee, D. S. W. Atkinson, and L. H. Crockett, *Software Defined Radio Using MATLAB & Simulink and the RTL-SDR*, 1st ed. Glasgow, Scotland, UK: Strathclyde Academic Media, 2015, available online at https://www.desktopsdr.com.

[54] R. Akeela and B. Dezfouli, "Software-Defined Radios: Architecture, State-of-the-Art, and Challenges," *Computer Communications*, vol. 128, pp. 106–125, September 2018. **DOI**: 10.1016/j.comcom.2018.07.012

[55] T. Hentschel, M. Henker, and G. Fettweis, "The Digital Front-End of Software Radio Terminals," *IEEE Personal Communications*, vol. 6, no. 4, pp. 40–46, August 1999. **DOI**: 10.1109/98.788214

[56] D. M. Molla, H. Badis, L. George, and M. Berbineau, "Software Defined Radio Platforms for Wireless Technologies," *IEEE Access*, vol. 10, pp. 26 203–26 229, March 2022. **DOI**: 10.1109/ACCESS.2022.3154364

[57] A. M. Wyglinski, D. P. Orofino, M. N. Ettus, and T. W. Rondeau, "Revolutionizing Software Defined Radio: Case Studies in Hardware, Software, and Education," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 68–75, 2016. **DOI**: 10.1109/MCOM.2016.7378428

[58] Ettus Research, "Products," available online at https://www.ettus.com/products/, accessed: May 2, 2022.

[59] T. B. Welch and S. Shearman, "Teaching Software Defined Radio Using the USRP and Labview," in *Proceedings 2012 IEEE International Conference on Acoustics, Speech, and Signal Processing – ICASSP 2012*, pp. 2789–2792, Kyoto, Japan, March 2012. **DOI**: 10.1109/ICASSP.2012.6288496

[60] D. Pu and A. Wyglinski, Digital Communication Systems with Software-Defined Radio. Boston, MA: Artech House, 2013.

[61] M. El-Hajjar, Q. A. Nguyen, R. G. Maunder, and S. X. Ng, "Demonstrating the Practical Challenges of Wireless Communications Using USRP," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 194–201, May 2014. **DOI**: 10.1109/MCOM.2014.6815912

[62] R. Heath, *Introduction to Wireless Digital Communication: A Signal Processing Perspective*. Hoboken, NJ: Prentice Hall, 2017.

[63] "RTL-SDR and Software Defined Radio News and Projects," available online at https://www.rtl-sdr.com, accessed: May 26, 2022.

[64] Analog Devices, "ADALM-PLUTO: Software-Defined Radio Active Learning Module," available online at https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/ADALM-PLUTO.html, accessed: May 26, 2022.

[65] S. G. Bilén, A. M. Wyglinski, C. R. Anderson, T. Cooklev, C. Dietrich, B. Farhang-Boroujeny, J. V. Urbina, S. H. Edwards, and J. H. Reed, "Software-Defined Radio: A New Paradigm for Integrated Curriculum Delivery," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 184–193, May 2014. **DOI**: 10.1109/MCOM.2014.6815911

[66] M. Rice and M. McLernon, "Teaching Communications with SDRs: Making It Real for Students," *IEEE Communications Magazine*, vol. 57, no. 11, pp. 14–19, November 2019. **DOI**: 10.1109/MCOM.001.1900185

[67] Great Scott Gadgets, "HackRF One," available online at https://greatscottgadgets.com/hackrf/one/, accessed: May 26, 2022.

[68] Lime Microsystems, "LimeSDR," available online at https://limemicro.com/products/boards/limesdr/, accessed: May 26, 2022.

[69] M. Gummineni and T. R. Pollipali, "Implementation of Reconfigurable Transceiver using GNU Radio and HackRF One," *Wireless Personal Communications*, vol. 112, pp. 889–905, January 2020. **DOI**: 10.1109/MCOM.2014.6815911

[70] O. Popescu, J. Musson, and D. C. Popescu, "Using Open-Source Software Defined Radio Platforms for Empirical Characterization of Man-Made Impulsive Noise," *IEEE Electromagnetic Compatibility Magazine*, vol. 9, no. 4, pp. 54–61, 4th Quarter 2020. **DOI**: 10.1109/MEMC.2020.9327997

[71] Lime Microsystems, "Myriad RF," available online at https://limemicro.com/initiatives/myriad-rf/, accessed: Sep. 12, 2022.

[72] Myriad RF, "Novena Open Hardware Computing Platform," available online at https://myriadrf.org/projects/component/novena-rf/, accessed: Sep. 12, 2022.

[73] "GNU Radio," available online at https://gnuradio.org, accessed: May 26, 2022.

[74] MathWorks, "Supported Hardware – Software Defined Radio," available online at https://www.mathworks.com/help/comm/supported-hardware-software-defined-radio.html, accessed: May 26, 2022.

[75] National Instruments, "What is LabVIEW?" available online at https://www.ni.com/en-us/shop/labview.html, accessed: May 26, 2022.

[76] "GNU Radio Out of Tree Modules," available online at https://wiki.gnuradio.org/index.php/OutOfTreeModules, accessed: May 26, 2022.

[77] MathWorks, "MATLAB Central File Exchange," available online at https://www.mathworks.com/matlabcentral/fileexchange/, accessed: May 26, 2022.

[78] "NI Community," available online at https://forums.ni.com, accessed: May 26, 2022.

[79] M. W. O'Brien, J. S. Harris, O. Popescu, and D. C. Popescu, "An Experimental Study of the Transmit Power for a USRP Software-Defined Radio," in *Proceedings 12th IEEE International Communications Conference – COMM 2018*, pp. 377–380, Bucharest, Romania, June 2018. **DOI**: 10.1109/ICComm.2018.8484809.

[80] S. Miyamoto, M. Katayama, and N. Morinaga, "Performance Analysis of QAM Systems Under Class A Impulsive Noise Environment," *IEEE Transactions on Electromagnetic Compatibility*, vol. 37, no. 2, pp. 260–267, 1995. **DOI**: 10.1109/15.385891

[81] ——, "Receiver Design Using the Dependence Between Quadrature Components of Impulsive Radio Noise," in *Proceedings 1995 IEEE International Conference on Communications (ICC)*, vol. 3, pp. 1784–1789, Seattle, WA, June 1995. **DOI**: 10.1109/ICC.1995.524506

[82] M. G. Sanchez, L. de Haro, M. C. Ramon, A. Mansilla, C. M. Ortega, and D. Oliver, "Impulsive Noise Measurements and Characterization in a UHF Digital TV Channel," *IEEE Transactions on Electromagnetic Compatibility*, vol. 41, no. 2, pp. 124–136, 1999. **DOI**: 10.1109/15.765101

[83] P. Torío, M. G. Sanchez, and I. Cuinas, "An Algorithm to Simulate Impulsive Noise," in *Proceedings 19th IEEE International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Dubrovnic, Croatia, September 2011.

[84] O. Popescu, "Power Budgets for CubeSat Radios to Support Ground Communications and Inter-Satellite Links," *IEEE Access*, vol. 5, pp. 12 618–12 625, July 2017. **DOI**: 10.1109/ACCESS.2017.2721948

Dimitrie C. Popescu received the Engineering Diploma and PhD degree in electrical and computer engineering from Polytechnic Institute of Bucharest and Rutgers University, respectively. He is currently a Professor in the ECE Department, Old Dominion University, Norfolk, Virginia. His research interests are focused on wireless communication systems and include software defined and cognitive radios, spectrum sensing and dynamic spectrum access for cognitive radios, modulation classification, interference mitigation and transmitter/receiver optimization to support quality of service, vehicular networks, and signal processing for communications and radar systems. He is an active Senior Member of the IEEE currently serving as an associate editor for IEEE Open Journal of the Communications Society and participating regularly in the technical program and organizing committees for the IEEE Global Telecommunications Conference (GLOBECOM), the IEEE International Conference on Communications (ICC), the IEEE Wireless Communications and Networking Conference (WCNC), and the IEEE Vehicular Technologies Conference (VTC).

Rolland Vida graduated as valedictorian at the Faculty of Mathematics and Computer Science, Babes-Bolyai University, Cluj, Romania, in 1997. He received the PhD degree in Computer Networks from Université Pierre et Marie Curie, Paris, France, in 2003. He is currently an Associate Professor with Budapest University of Technology and Economics, Department of Telecommunications and Media Informatics, and the Head of HSN Lab, an academic research laboratory that is a strategic partner of Ericsson since 1992. In the last ten years, he has held different leading positions in IEEE Communications Society, IEEE Smart Cities, and IEEE Sensors Council. He is a member of the Steering Committee of IEEE Internet of Things journal, and Associate Editor of IEEE Sensors Letters. He served as the TPC Co-Chair for the IEEE Sensors Conference in 2019, 2020, and 2022. Rolland Vida is a Senior Member of IEEE.