Progress In Electromagnetics Research Letters, Vol. 101, 1–10, 2021

Radio-Propagation Measurement Based on a Low-Cost Software Defined Radio

Marcelo B. Perotoni[1, *], Felipe A. A. Silva[1], and Marcos S. Vieira[2]

Abstract—This article reports the development and test of a radio-propagation measurement system based on an 8-bit software-defined radio. Tests are performed in an urban area at the frequency of 733 MHz and compared with numerical prediction from the Altair WinProp commercial suite. The system is portable (1.2 kg), low-cost, based on non-proprietary open-source tools and has the capability of tracking the GPS coordinates of the measured points. Frequency limit of the system is bounded by the software-defined radio in use, and the limit of the present case spans 24 MHz to 1700 MHz. The integrated system does not need user intervention after its initial setup can be operated autonomously.

1. INTRODUCTION

Initial applications of software-controlled hardware blocks applied to radio communication were mentioned in the open literature starting from mid-1980s, aiming the use of a single instrument able to provide the interoperability among different combat field equipment. That was the essence of the US Army-funded SPEAKEasy I and II projects, which officially spanned the period from 1992 to 1997 [1, 2]. However, by the time the project reached its end, the prototype was already obsolete due to the hardware evolution. The specific term Software Defined Radio (SDR), however, was coined by Joe Mitola in 1991 [3]. In general, the SDR is attractive due to its versatile operation with regard to various modulation schemes and frequency ranges, given its software control acting upon the hardware. The concept of SDR enables a further paradigm change, the so-called Cognitive Radio [4], where the frequency and modulation in use are dynamically chosen by selecting open spots in the spectrum, enabling a more efficient use of the available frequency range.

An accidental finding involving a USB dongle distributed as a Digital Video Broadcasting — terrestrial (DVB-T) receiver paved the way to the popularization of SDRs. Its RF-converted samples could be transferred to the USB channel without demodulation by means of a hardware bypass and with software control of some its features. That is the origin of the RTL SDR, used in this study. Other SDRs offer the possibility of operation in different frequency ranges, instantaneous bandwidths, and also the possibility of operation in transmitting mode (full or half-duplex). They also have a widespread range of applications, due to their inherent versatility and software-based operation, without the need of complex and costly RF instrumentation. Some fields worth mentioning are medical imaging [5], radar [6], ground-penetrating radar [7], for educational purposes [8] and as a radioastronomy receiver [9].

There are two common alternatives to measure power in outdoor propagation scenarios:

- Spectrum Analyzers and
- Specialized integrated systems.

Received 6 July 2021, Accepted 22 October 2021, Scheduled 28 October 2021
* Corresponding author: Marcelo Bender Perotoni (mperoconsult@gmail.com).
[1] UFABC, Brazil. [2] Mackenzie Presbyterian University, Brazil.

The first option, spectrum analyzer, presents a bulky and inconvenient alternative in case that it is not battery-powered. It does not provide geo-reference to the measurements, and it is a relatively high-cost instrument. In addition to it, a visual inspection for each measured point should be performed. For the cases where a geo-reference unit is used with the spectrum analyzer, a laptop running a control software (e.g. Labview) is needed to operate both instruments simultaneously. Specialized integrated systems, in turn, have a GPS receiver which can deliver a map with the automatically acquired power samples and are usually employed in vehicles. However, they are sophisticated instruments, not affordable to most research laboratories.

This study presents an alternative based on a low-cost 8-bits SDR (RTL), battery-powered and low-weight (approximately 1.2 kg), therefore portable. It automatically computes the power level at the chosen frequency while assigning the measured value to its geographic coordinates, by means of a GPS module. The complete system is controlled by a Raspberry PI 3, which has 1 GB RAM and uses an 1.2 GHz 64-bit ARM processor, running Raspberry PI OS. Its software application was developed with GNU Radio, a Python-based, open-source tool. So both operational system and application software are non-proprietary and free. In contrast to most spectrum analyzers, it can be programmed to automatically acquire arbitrary samples per second.

Initial investigations had a hardware-only system based on a logarithm-broadband power meter integrated circuit (AD8318, covering 1 MHz to 8 GHz, dynamic range between −65 and 5 dBm). Though operating with high dynamic range, this class of ICs requires a very steep out-of-band rejection, by means of high-Q filters, customized to the desired range, since they integrate the power at their inputs in a broadband fashion. Besides the frequency selection, they also require an LNA (low-noise amplifier) stage for outdoor radio-propagation measurements. SDRs, however, enable a more convenient and affordable option with all the benefits of its software control, from the digital filtering of incoming data to internal amplification in case that it is needed. The hardware-only approach, on the other hand, requires interconnections with RF cables, and also DC-powering of active circuits might involve different voltage levels which need to be converted from a battery. The final block diagram is shown in Fig. 1, along with the picture of its real deployment.

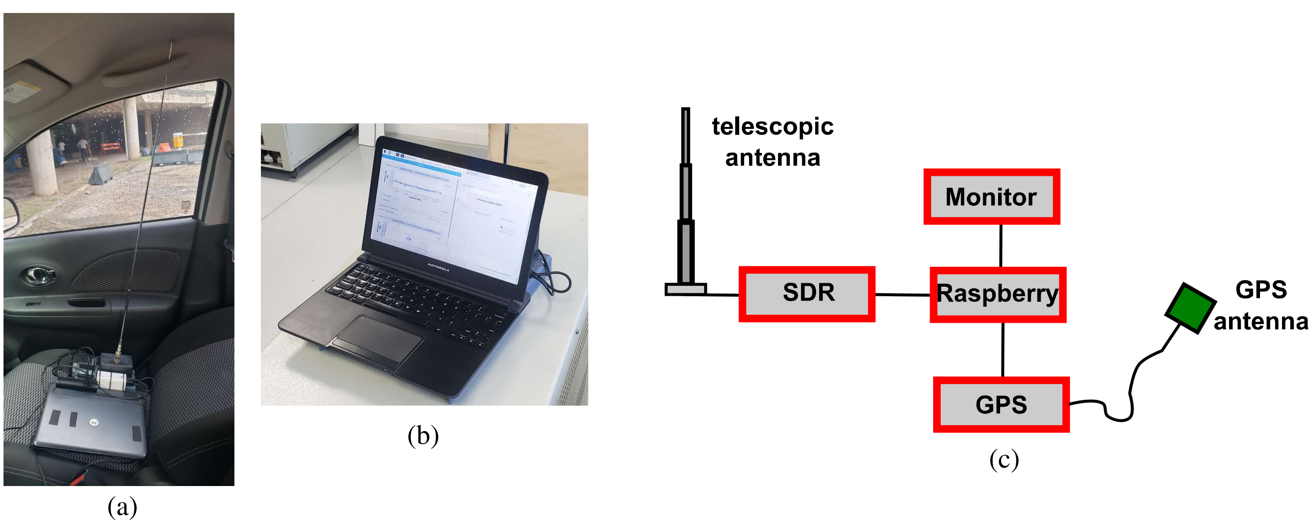


Figure 1. (a) System in a car; (b) dumb monitor with the application screenshot and (c) block diagram.

The GPS module (Stoton GNSS), SDR and a 11.6” dumb monitor (i.e., without processing) are connected to the Raspberry PI by means of USB connectors, occupying three of the available four slots. The dumb monitor allows the visualization of incoming data, checking whether the transmitter is actually operating as well as if there are other nearby emissions — this spectrum area can be legally occupied by 4G mobile transmissions. In the open literature some similar projects are reported. A propagation measurement system using the same SDR is reported [10], with the additional inclusion of an RF switch to select between two antennas, since it was originally devised to be of dual-band operation. In comparison to the present study, it does not include the monitor and does not contain a TCP/IP client-server option for the data-communication. Another report [11] uses 3 different SDRs (8-bits HackRF One, 12-bits Pluto and 14-bits NI USRP 2930) for the European TETRA (Terrestrial Trunked Radio) system measurement, operating around 390 MHz. It also contains a georeference board (GLONASS) and is controlled by a Raspberry PI. Using the same SDR and microprocessor board as this work — an RTL SDR connected to a Raspberry PI, a radio surveillance module was built, intended to search for anomalous electromagnetic emissions, SAS (Spectrum Alerting System) [12]. It works by comparing the acquired data in the frequency domain to a baseline emission, setting up a warning in case that an unidentified carrier is identified. The radiant system is an omnidirectional antenna, which sweeps the complete SDR 1.7 GHz bandwidth spectrum, sending the data real-time to a remote location. In spite of the low-cost processing microprocessor, filtering, FFT and DC-removal operations are locally performed by a GNU Radio flowgraph [13]. SDRs can also replace heavier and bulky instrumentation in other systems. For instance, a UHF/VHF signal strength level in tunnels and mines was measured by a system based on a wheeled robot transporting a conventional spectrum analyzer — its replacement by an SDR-based counterpart would be positive from the viewpoint of price, weight, and also the possibility of mechanical control by the microprocessor (Raspberry PI) [14].

The frequency range of the proposed system is limited by the SDR only. It does not need additional filtering unless strong FM broadcast carriers are close to the desired range — fact not observed during the tests. The RTL SDR can cover 24 MHz to 1.7 GHz, so that FM, Digital TV, UHF RFID, LORA/SigFox are possible to be analyzed.

For the present study, the frequency of 733 MHz was chosen due to the proximity to both lower-frequency 5G band and the Brazilian Digital TV range. Next sections cover the SDR and its sensitivity, and the software application that controls the integrated unit. A real-world test of it is shown in an urban area.

2. SDR

Within the SDR block diagram, the ADC (analog-to-digital converter) is one of the main factors that define its global dynamic range, a particularly relevant parameter in case of a measurement system. It is defined in terms of maximum and minimum input amplitudes; the former is found after the onset of intermodulation products and the latter by the instrument noise floor. The ADC signal-to-noise ratio (SNR) in dB can be expressed as [1]:

$$\mathrm{SNR} = 6.02N + 1.76 \tag{1}$$

with N defined as the ADC number of bits. The expression defines an upper-bound (optimistic) value considering the noise as AWGN type (Additive White Gaussian Noise). In practice, the parameter $ENOB$ (effective number of bits), usually smaller than N, represents a more realistic number of bits which takes into account other factors. For the case of 8-bit RTL SDR it results in an SNR of approximately 50 dB, which can be increased at expenses of techniques such as averaging and oversampling.

Another important parameter in the SDR operation is its bandwidth, and for the RTL unit its maximum BW is approximately 3 MHz, though it is reported that a safe upper limit of 2.4 MHz is suggested as to avoid sample losses in the USB channel. It is worth mentioning that due to the presence of filters and other elements in the loop, SDR noise behavior is unlike that of resistors, whose noise level is proportional to the incoming bandwidth. It means that the SDR operates with an optimum BW that minimizes the Noise Floor (NF). In fact, Dither techniques take advantage of the incoming noise in larger bandwidths (outside the region of interest) to increase the ADC sensitivity [15]. In the context of accurate readings, a higher-end 12-bits USRP SDR was shown able to generate measurements as precise as commercial RF instrumentation [16]. Another particularity of SDRs, conversion from RF to IF (intermediate frequency), is usually done with interleaved complex samples, providing I (in-phase) and Q (quadrature) data. It enables applications where phase information is needed, such as digital modulation and microwave imaging.

SDRs usually contain internal LNAs, switchable by software. In the case of RTL, it can go from 0 dB (where the amplifier is bypassed) up to 40 dB. It also contains a temperature-compensated crystal oscillator (TXCO), with nominal frequency stability of 0.5 ppm.

A measurement was performed in order to quantify the RTL noise floor, in dBm, for different frequencies across its operating range. The used setup involved an RF generator connected to the SDR (Fig. 2). The RF generator output power was set to 10 dB above the noise floor level at each frequency, this value chosen to ease the visualization against time-varying noise fluctuations. Measured peak power at these discrete frequencies was then subtracted from 10 dB to find the current *NF*. Two different SDR bandwidths (1 MHz and 2 MHz) were used. Internal SDR amplifiers were set to 0 dB, and neither filtering nor averaging was employed to keep the stream as raw as possible. From the results, an inverse relation between the SDR sensitivity and the frequency is observed. Frequencies in the range of 1 GHz to 1.5 GHz were shown to have higher sensitivity for the 1 MHz bandwidth case, probably due to the ADC and associated circuitry optimal performances.

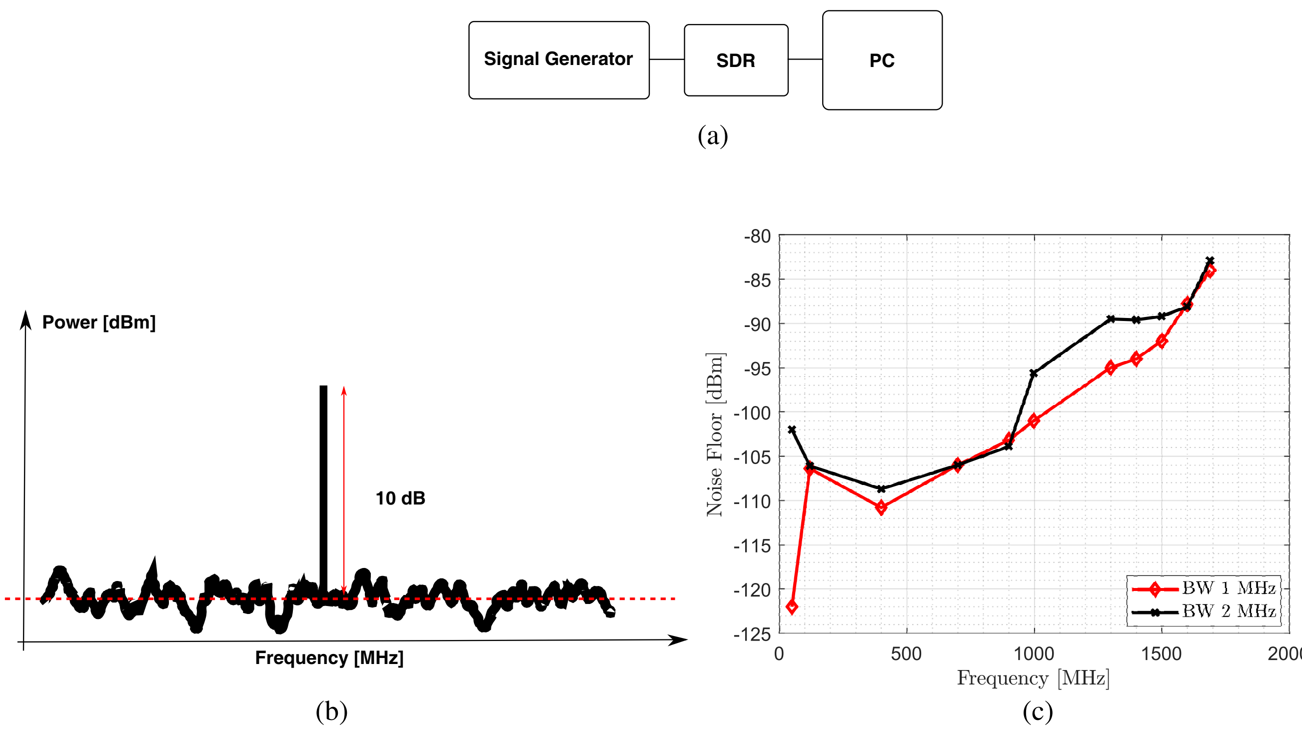


Figure 2. (a) Instrument connection, (b) pictorial visualization of the SDR power reading and (c) results.

3. SOFTWARE APPLICATION

The application is developed within GNU Radio, an open-source suite which contains the GNU Radio Companion, a development tool that controls the SDR interface, performs signal processing, and visualization functions by means of graphic block diagram programming. GNU Radio is based on C++ and Python, the latter operating as wrapper and user interface. The final developed flow graph can be run independently of GNU Radio as a Python script. Fig. 3 contains the final block diagram. It starts with the interface to the SDR (block named osmocom) which sets the bandwidth (1 MHz), frequency (733 MHz), LNA gain, and other parameters. Band-pass filtering (100-kHz wide) and a moving average routine reduce signal fluctuations due to noise, cut occasional nearby carriers, and attenuate amplitude oscillations due to fading. The data are sent to a display (shown in the dumb monitor) and also to a TCP/IP channel (via a ZeroMQ interface), which enables a remote control of the system in a client-server format. In case that the communication protocol contains several frequency slots, a different approach should be followed, like the one reported in [11].

In addition to GNU Radio, the RTL is also possible to be controlled from Matlab [17], with a

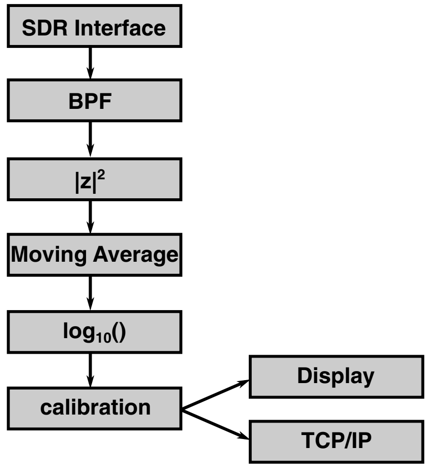


Figure 3. Flowgraph of the developed GNU radio code.

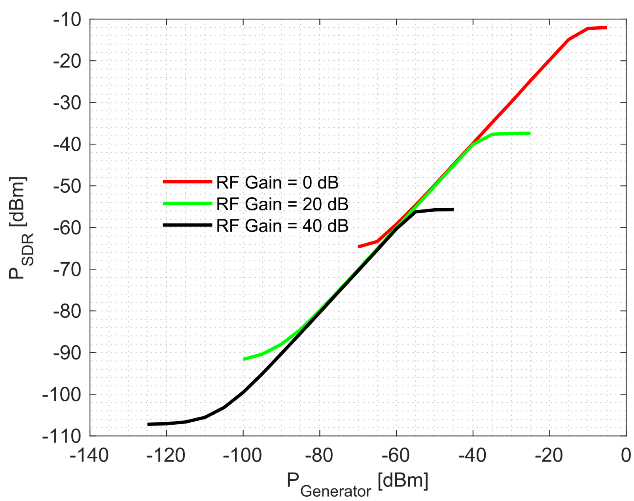


Figure 4. Calibration curve, parameterized with three different LNA gain settings.

thorough review involving this specific SDR presented in [18]. It is, however, a commercial package that would require more from the point of view of processing power, in this present case, the Raspberry PI.

GNU Radio expresses the power in frequency domain using the dBr unit (relative dB). In order to convert the final data to dBm a calibration process is performed by means of an external RF signal generator. Fig. 4 shows the calibration curve, where it is possible to see that the SDR has a linear dynamic range from -105 dBm to -20 dBm, depending upon the choice of LNA gain. This range can be further expanded by an external LNA if necessary.

The GPS module has its geographic coordinates acquired by the GPSD within the Raspberry PI, and its Python library records both the coordinates along the time they were read.

4. FIELD TEST MEASUREMENTS

Real-world measurements took place in the area close to UFABC, Santo Andre, Brazil (coordinates $-23.644107, -46.526872$), a relatively dense urban area. The transmitter was located in the eighth floor of a building (30 m height), continuous wave, consisting of an RF generator followed by a power amplifier, 1 W RMS output power. The application tool was set to get 1 sample per second along the desired route, which covers an area whose most points follow the Rayleigh statistic, i.e., without line of

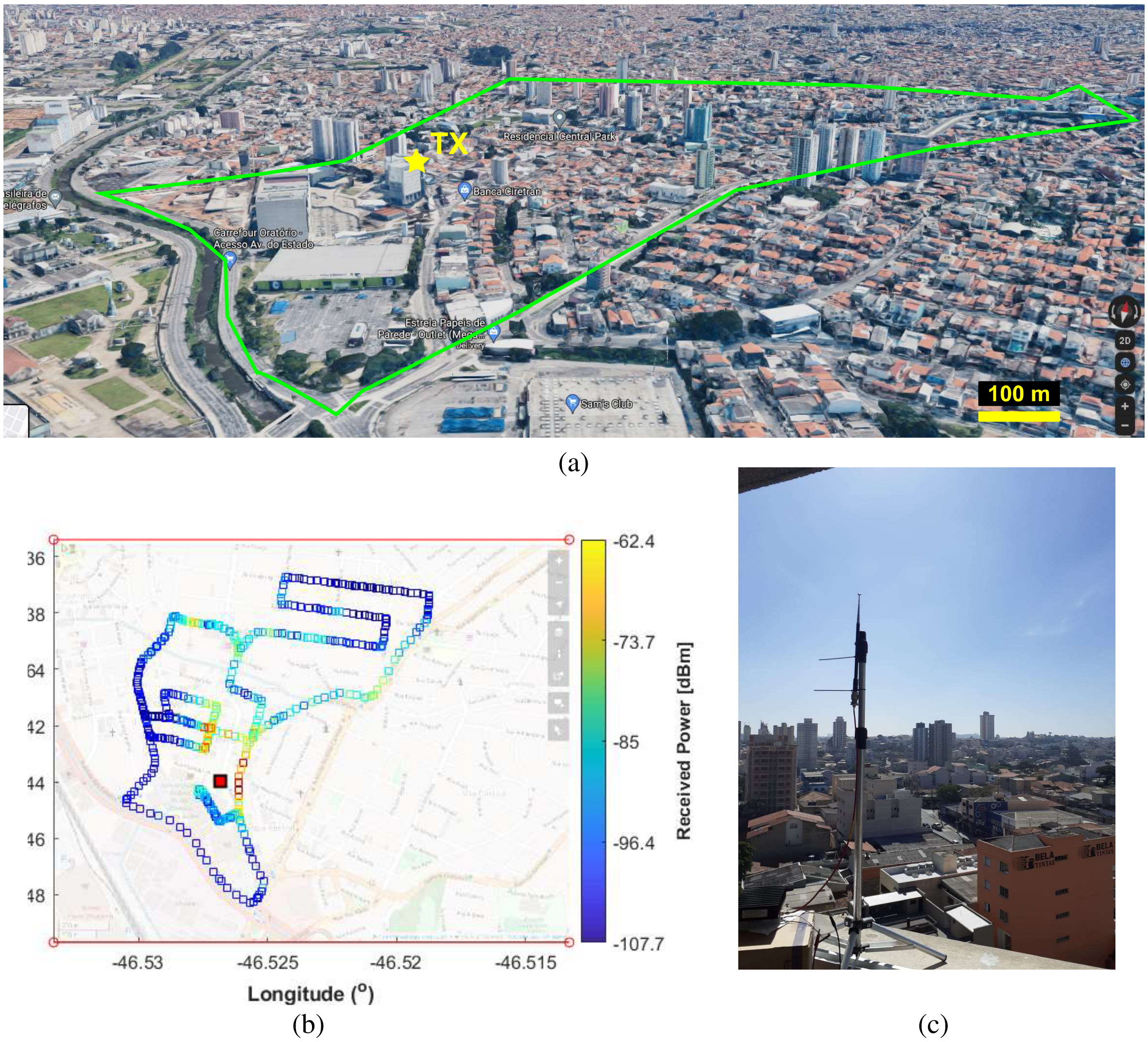


Figure 5. (a) 3D view of the measured area and (b) measured results and (c) transmitting antenna on the window ledge.

sight between receiver and transmitter. The maximum measured distance was limited to 1.2 km since it reaches the SDR sensitivity margin. Both transmitter and receiver antennas were telescopic monopoles, omnidirectional and with vertical polarization. Fig. 5 depicts the route and respective measurements, overlayed with the ©Open Street Map. The complete test was performed in 32 minutes, resulting in 1100 valid points. Assuming a maximum vehicle speed of 40 km/h, it results in an offset of approximately ±97 kHz due to the Doppler effect, still captured by the system bandwidth, set by the bandpass filter. A thorough study on the Doppler effect on GPS signals (1.53 GHz) received in a vehicle is reported [19], with the three speeds of 60 km/h, 90 km/h, and 120 km/h, larger than this study maximum observed speed.

Friis free-space formula is a first-order modeling equation used in radio-propagation, where the received power falls with the second power of the distance [20]. Though originally applied to an ideal reflectionless environment it was used to model the acquired real-world data with a slight modification. The received power in dBm P_{RX} was fitted to the equation:

$$P_{RX} = 10\log(P_{TX}G_{TX}G_{RX}) - 10n\log\left(\frac{4\pi d}{\lambda}\right) = B - 10n\log(d) \qquad (2)$$

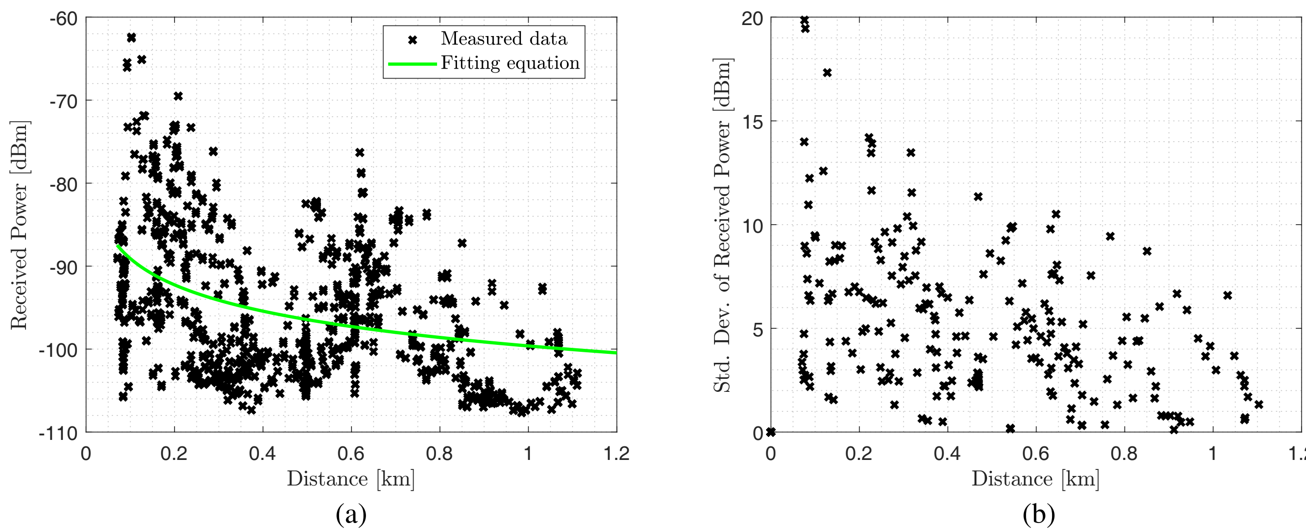


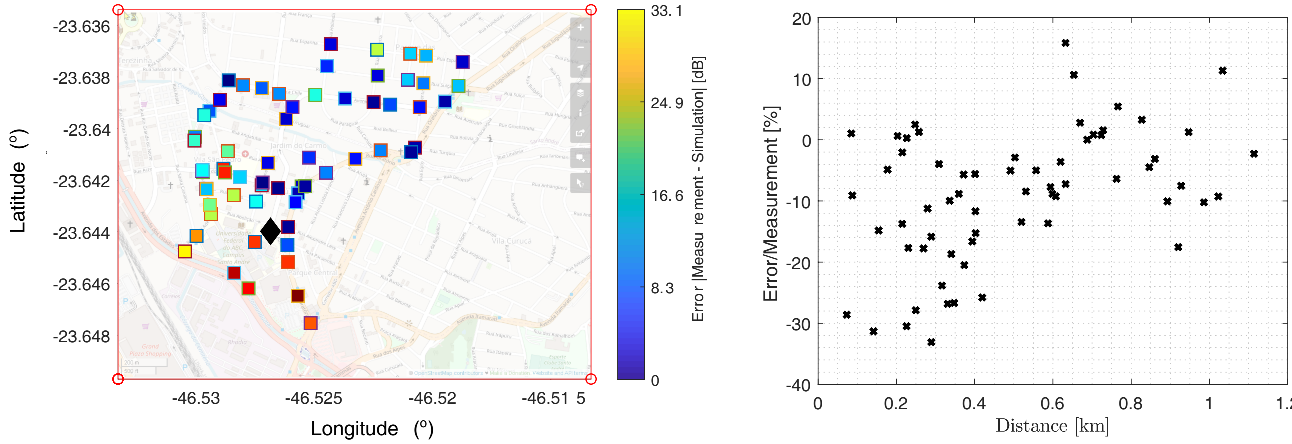
Figure 6. (a) Distribution of the measured data vs. distance and its fitting equation and (b) standard deviation of the measurements.

where P_{TX} is the transmitted power; G_{TX} and G_{RX} are respectively the transmitter and receiver antenna gains; d is the distance and λ the wavelength. The B term contains all the distance-independent data whereas n, the main unknown, accounts for the power law. Fig. 6 contains the measured data alongside the fitted curve, which results in $n = 1.05$, smaller than the original $n = 2$ in the Friis law due to the multiple reflections, from the ground and obstacles. The standard deviation of the measurement is also shown, taken at every 5 consecutive samples. As the receiver moves farther away from the transmitter site, the received power becomes more homogeneous.

Other popular and more sophisticated formulations (Okumura-Hata and COST Hata) have their validity only for distances larger than 1 km, so they are applied to a small fraction of the measured data [20].

For the sake of comparison, the radio propagation prediction suite Altair WinProp was employed, using the DPM (Dominant Path Model) Method, where only the most significant ray is taken into account for each spatial sample on the area of interest [21]. DPM offers advantages in contrast to RT (Ray Tracing) method — both of deterministic type — in terms of processing demand and simulation time and also more precision than empirical models. Since only the most significant ray (or rays) are considered, the need for a very realistic and precise 3D modeling is relaxed, unlike Ray Tracing method. A virtual model was constructed on the same area where the measurements were collected, with buildings manually designed in an approximate way due to the lack of a more precise (and costly) 3D scenario model. The simulation considered the area as flat, disregarding the maximum 30 meters of elevation difference in the route. Added complexity of elevation information would not pay off particularly for an outdoor scenario like the studied (Fig. 5), and it is a common approach in practical numerical predictions for mobile phone field evaluations. The simulation setting considered the receiving points to be placed at a 1.5 meters height in open space, with the vehicle influence on the receiving site not taken into account [22]. Fig. 7 shows the absolute error between the measured and simulated data, for 64 points along the route. It also shows the error in relation to the distance to the transmitter site. It can be seen that the largest differences lie close to the transmitter site and on the opposite side to where the TX antenna was placed (lower left of the map). Observed error increases for distances close to the transmitter, effect caused by the non-ideal effects of the antenna, modeled as omnidirectional in the simulation. The WinProp model considered the TX antenna deployed on the building rooftop, whereas the real world had the antenna projected from the eighth floor window opening, as shown in Fig. 5. Actual antenna placement generated a distortion in both vertical and horizontal planes, in contrast to ideal coverage case analyzed in the simulation.

The histogram map of the differences is shown in Fig. 8. It can be seen that the simulation estimates



(a) (b)

Figure 7. (a) Absolute error between simulation and measurements and its relative distribution with the distance from the transmitter site. (b) TX site is marked as a black diamond on the map.

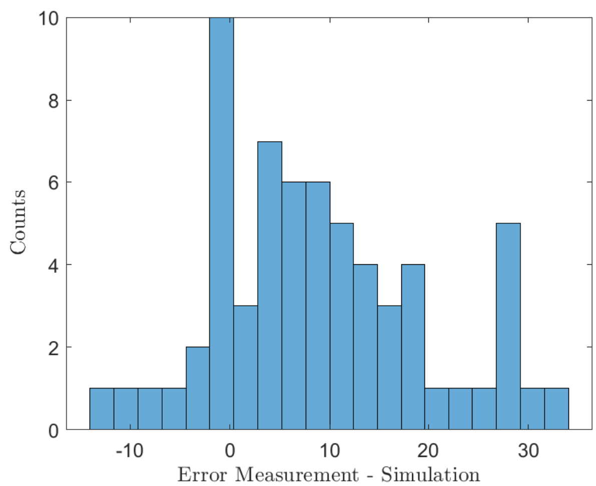


Figure 8. Histogram of the difference between measurement and simulation data.

are generally optimistic in relation to the measurements, since most difference samples are larger than zero. This error can be ascribed to the vehicle presence partially shielding the receiving energy and the transmitting antenna not placed on the rooftop, with its energy blocked by the concrete walls in the real case.

5. CONCLUSIONS

This article showed the design, integration, and test of a radio-propagation measurement system based on a low-cost software-defined radio, a GPS module, and a Raspberry PI. The prototype was entirely based on open-source software tools. The ascii data comprising the automatically acquired power, geographic coordinates and respective time were retrieved after the outdoor test was finished by means of a simple USB connection. It was used without the need of human intervention during a route performed in a car. Frequency and samples per second are the only parameters that need to be set in advance. Range of operation can cover from 24 MHz to 1700 MHz, and the minimum sensitivity reached -105 dBm with the use of the SDR internal LNAs. The comparison to a numerical prediction suite is used to validate the data.

REFERENCES

1. Collins, T. F., R. Getz, D. Pu, and A. M. Wyglinski, *Software-defined Radio for Engineers*, Artech House, Norwood, 2018.
2. Lackey, R. J. and D. W. Upmal, "Speakeasy: The military software radio," *IEEE Comm. Mag.*, Vol. 33, No. 5, 56–61, 1995.
3. Mitola, III, J., *Software Radio Architecture: Object Oriented Approaches to Wireless Systems Engineering*, John Wiley and Sons, New York, 2000.
4. Raut, R. D. and K. D. Kulat, "SDR design for cognitive radio," *Fourth International Conference on Modeling, Simulation and Applied Optimization*, 1–8, Apr. 2011.
5. Marimuthu, J., K. S. Bialkowski, and A. M. Abbosh, "Software-defined radar for medical imaging," *IEEE Trans. Micr. Theory Techn.*, Vol. 64, No. 2, 643–652, 2016.
6. Costanzo, S., F. Spadafora, G. Di Massa, A. Borgia, A. Costanzo, G. Aloi, P. Pace, V. Loscri, and H. O. Moreno, "Potentialities of USRP-based software defined radar systems," *Progress In Electromagnetics Research B*, Vol. 53, 417–435, 2013.
7. Ralston, J. and C. S. Rargrave, "Software defined radar: An open source platform for prototype GPR development," *14th International Conference on Ground Penetrating Radar (GPR)*, 172–177, Jun. 2012.
8. Del Barrio, A. A., J. P. Manzano, V. M. Maroto, A. Villarin, J. Pagan, M. Zapater, J. Ayala, and R. Hermida, "Hack-RF + GNU radio: A software-defined radio to teach communication theory," *International Journal of Electrical Engineering & Education*, 1–18, 2019.
9. Supriyatno, B. I., T. Hidayat, A. B. Susksmono, and A. Munir, "Development of radio telescope receiver based on GNU radio and USRP," *1st International Conference on Wireless and Telematics (ICWT)*, 1–4, Apr. 2016.
10. Wright, D. P. and E. A. Ball, "Highly portable, low-cost SDR instrument for RF propagation studies," *IEEE Trans. Instrum. Meas.*, Vol. 69, No. 8, 5446–5457, 2020.
11. Helbet, R., P. Bechet, V. Monda, S. Miclaus, and I. Bouleanu, "Low-cost sensor based on SDR platforms for TETRA signals monitoring," *Sensors*, Vol. 21, 3160, 2021.
12. Ball, D., N. Naik, and P. Jenkins, "Spectrum alerting system based on software defined radio and raspberry Pi," *Proceedings of the 2017 Sensor Signal Processing for Defence Conference (SSPD)*, 15, Dec. 2017.
13. Pfammatter, D., D. Giustiniano, and V. Lenders, "A software-defined sensor architecture for large-scale wideband spectrum monitoring," *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, 71–82, Apr. 2015.
14. Zhou, C., J. Waynert, T. Plass, and R. Jacksa, "Attenuation constants of radio waves in lossy-walled rectangular waveguides," *Progress In Electromagnetics Research*, Vol. 142, 75–105, 2013.
15. Reed, J. H., *Software Radio: A Modern Approach to Radio Engineering*, Prentice Hall, Upper Saddle River, 2002.
16. Andrich, C., A. Ihlow, J. Bauer, N. Beuster, and G. Del Galdo, "High-precision measurement of sine and pulse reference signals using software-defined radio," *IEEE Trans. Instrum. Meas.*, Vol. 67, No. 5, 1132–1141, 2018.
17. Stewart, R. W., L. Crockett, D. Atkinson, K. Barlee, D. Crawford, I. Chalmers, M. McLernon, and E. Sozer, "A low-cost desktop software defined radio design environment using MATLAB, simulink, and the RTL-SDR," *IEEE Commun. Mag.*, Vol. 53, No. 9, 64–71, 2015.
18. Stewart, B., K. Barlee, D. Atkinson, and L. Crockett, *Software Defined Radio Using Matlab & Simulink and the RTL-SDR*, Strathclyde, Glasgow, 2017.
19. Hamid, A. F. A., M. T. A. Rahman, A. Rahman, and M. M. M. Zabidi, "Path loss analysis considering doppler shift effect on cellular communication for connected car application at rural area," *IOP Conf. Ser.: Mater. Sci. Eng.*, 1–8, Feb. 2019.
20. Haslett, C., *Essentials of Radio Wave Propagation*, Cambridge, Ofcom, 2008.

21. Wahl, R., G. Wölfle, P. Wertz, P. Wildbolz, and F. Landstorfer, "Dominant path prediction model for urban scenarios," *German Microwave Conference (GeMiC 2005)*, 1–5, Apr. 2005.
22. Kaul, S., K. Ramachandran, P. Shankar, S. Oh, M. Gruteser, I. Seskar, and T. Nadeem, "Effect of antenna placement and diversity on vehicular network communications," *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 112–121, Jun. 2007.