URSI GASS 2021, Rome, Italy, 28 August - 4 September 2021

Open-Source Software-Defined Radio Receiver Platform for Harmonic Radar Applications to Track Airborne Insects

Diego Penaloza-Aponte(1), Julio Urbina(1), and Jose D. Fuentes(2)
(1) School of Electrical Engineering and Computer Science
(2) Department of Meteorology and Atmospheric Science
The Pennsylvania State University, State College, PA, USA

Abstract

We report on the development and integration of a cost-effective, open-source software-defined radio (SDR) receiver into a harmonic radar system to study the movement patterns of insect pollinators. A Raspberry Pi along with the SDR collects data to process it and subsequently transfers it wirelessly via an ad-hoc WiFi network to a host computer. This remote access receiver was programmed fully using Python. We describe the SDR system along with preliminary testing of the radar receiver in a laboratory setting.

1 Introduction

Harmonic radars have been successfully used to monitor foraging patterns of insect pollinators [1]-[5]. A harmonic radar transmits a pulse at a fundamental frequency and detects the second harmonic echo produced by a transponder carried by targeted insects. In this way, the clutter produce by the fundamental frequency is avoided and the detection tuned at the second harmonic just identifies a few objects that can generate harmonics. Riley and Smith [5] provided all the design considerations for a harmonic radar to track airborne insects using a fundamental frequency of 9.41 GHz, with a transmission power of 25kW, and a loop dipole soldered to a Schottky diode (nonlinear component) as a transponder. Results showed detection of up to 900 meters. Based on the success of this technology, several studies have used it as the main reference for new designs [6]-[7]. In 2011, [8] presented a harmonic radar capable to track targets at a maximum distance of 125 meters. This sensor uses a maritime commercial radar from the Furuno brand and operates at the frequency of 9.41 GHz. It has a customized receiver system mounted on top of the transmitting antenna. The use of the Furuno radar as the fundamental frequency generator avoids the necessity of designing a new transmitter. Then we can focused our efforts on the design of the receiver. The receiver presented in [8] is based on legacy technology and difficult to reproduce. Here, we report a radar receiver using open-source Software-Defined Radio (SDR) tools. The Adalm-Pluto SDR [10] and the Raspberry Pi 4, provide a low-cost ($\sim$ \$250), open-source platform, from which typical applications in SDR can be realized. The design described in this paper builds upon the system reported in [8]. For completeness, a general overview of the harmonic radar system architecture and data processing is provided. The paper will conclude with observational results from an indoor data collection experiment.

2 Harmonic Radar System Architecture

In general, a harmonic radar has three main subsystems: 1) transmission, 2) transponder, and 3) receiver, as illustrated in Fig. 1, and further described below. This figure shows further details of each component and also indicates the corresponding signal paths.

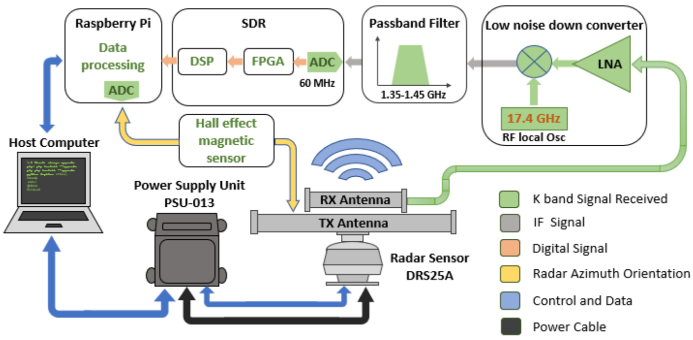


Figure 1. Functional diagram of the Harmonic Radar system.

Considering the work reported in [8], we selected the Furuno radar (model DRS25A), as the transmitter of the radar. This model can generate a pulsed carrier at a fundamental frequency $f_o = 9.41$ GHz with a peak power of 25 KW. The TimeZero Navigator software is used to set up the operational modes of the transmitter, such as pulse width, etc., as indicated in its corresponding manual.

In general, a harmonic radar receiver relies on the backscatter electromagnetic signal from a transponder mounted on the target. In this report, a transponder was constructed using a Schottky diode (SMS7630-079LF) that is soldered to a loop dipole antenna fabricated with copper wire of 0.25 mm diameter as described in ([5], [8]). The loop dipole antenna captures electromagnetic energy at f_o, which in turn excites the diode device, and electromagnetic energy is emitted by the loop antenna at $2f_o$. Numerical analysis of the loop antenna were performed to characterize gain and impedance values to understand the maximum detection range of the system using the harmonic radar analysis ([5], [9]). The outcome of this analysis produced a theoretical maximum range detection of about 120 meters, assuming all the conditions are favorable for the target detection.

The receiver consists of a 12×4 rectangular microstrip patch antenna array constructed using a 20 mils Rogers RO4350 substrate, modeled using Ansys HFSS [11], resonating at $2f_o$ = 18.82 GHz. The electromagnetic energy collected by this antenna is passed to a Low Noise Block (LNB) Norsat 9000HX-O3B-BN Ka-Band Duo, which has a local oscillator centered at f_{LO} =17.4 GHz that when it is mixed with the second harmonic of the radar at $2f_o$ =18.82 GHz, it yields an intermediate frequency f_{IF} = 1.42 GHz. This signal is subsequently passed through a band-pass filter, and fed into the Adalm-Pluto SDR, where the signal is further down converted and then digitized to produce quadrature and in-phase signals. The output data of SDR is sent to a Raspberry Pi 4 that hosts GnuRadio and the corresponding kernel drivers to communicate with the Pluto SDR. The azimuth position of the transmitter antenna as it rotates at a speed of 48 RPM is sensed with a hall-effect sensor (Honeywell 103SR) and sampled with the Raspberry Pi Hat ADS1015. The Raspberry Pi manages all collected data and broadcasts them through a WiFi network to a nearby host computer for further data analysis, processing, and visualization.

3 Software-Defined Radio Receiver

Due to the pandemic of COVID-19 and lockdown regulations, the experimental results presented here were limited and conducted inside a lab. The receiver subsystem was assembled and carefully mounted on top of the Furuno radar antenna. Fig. 2 shows the physical elements of the prototype receiver.

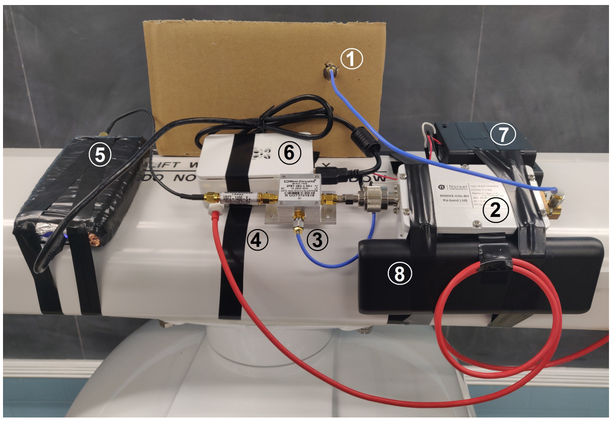
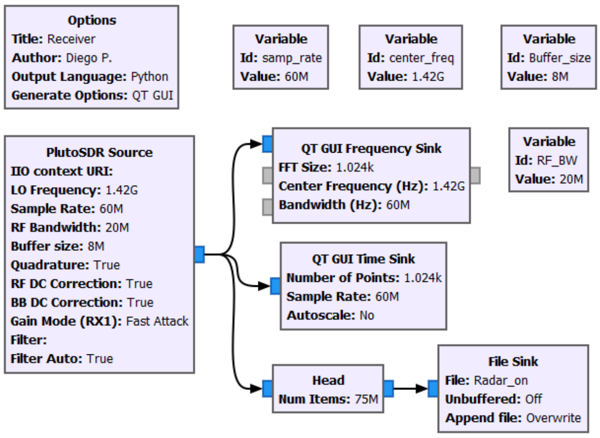


Figure 2. Harmonic radar implementation back view shows: (1) RX antenna, (2) LNA, (3) Bias TEE, (4) Band-pass Filter, (5) SDR, (6) Raspberry Pi, (7) 12 VDC battery, and (8) 5 VDC battery.

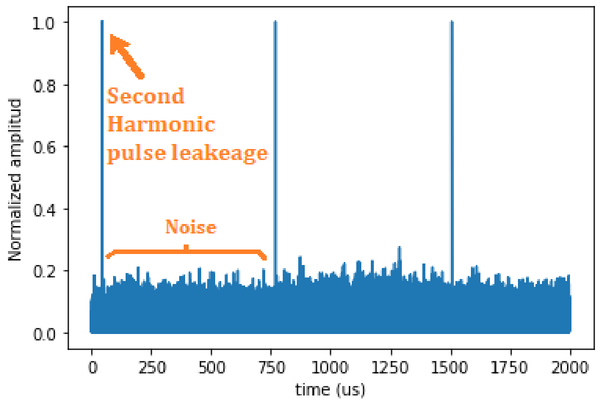
Typically, in a radar system both the transmitter and receiver share the same reference clock to determine accurately the remote location of a target. However, in using the Furuno radar as the transmitter of the harmonic radar, it limited our access to internal hardware details due to proprietary constrains. A solution to this limitation was to take advantage of the harmonic leakage produced by the power amplifier of the Furuno radar. Therefore, when in transmission at f_o, the receiver captured the leaked energy of the second harmonic at $2f_o$, which provided the clock reference (in a semi-coherent mode at $2f_o$) needed to interpret the echo generated from the transponder when emitting energy at $2f_o$. This echo signal is processed in software to calculate the time delay echo produced by the target and translate it into its current position relative to the radar position. This process is implemented as follows. The Adalm-Pluto SDR captures the output signal from the output of the down-converter using the widely known open-source GNU-radio [12]. Fig. 3a shows a graphical implementation using signal processing flowgraphs from GNUradio. In our experiment, the local oscillator was set to 1.42 GHz, which is the carrier frequency from the output of the down-converter, as described in Section II. Since the minimum pulse width the Furuno radar can generate is 0.08 μs, this value translates to a signal with a minimum bandwidth of 12.5 MHz in baseband. The sampling rate is set to 60 MHz to guarantee Nyquist sampling conditions of at least twice the signal bandwidth to properly digitized an analog signal without aliasing. Fig. 3a, RF bandwidth represents a low pass filter in the Adalm-pluto's SDR to limit the bandwidth of the input signal. This value is set to 20 MHz to keep the noise power low. Then, a block called *Head* limits the number of samples to 75 millions which corresponds approximately to the number of samples in one full rotation of the Furuno radar at a rate of 48 RPM. The sampled in-phase and quadrature data is stored in the Raspberry Pi in a binary file format using a block called *File Sink*. A time-domain plot of the data collected using these steps are illustrated in Fig. 3b. The vertical blue thin line, with a normalized amplitude of 1, represents the leaked energy of the transmitter at $2f_o$, repeating at a rate of 1500 Hz (or about 667μs). This plot illustrates the successful implementation of the SDR feature of the receiver but lacks echo returns from the transponder since the measurement was conducted inside a lab and in exceedingly proximity of near field of the electromagnetic energy. In an actual field deployment experiment of the radar, more complex graphical visualizations such as target position and tracking can be implemented by a laptop that can receive all collected data through a WiFi connection.

4 Discussion

Despite the limitations of the indoor deployment of the radar, the measurements depicted in Fig. 3b, demonstrate the functionality of the SDR receiver at capturing the second harmonic $2f_o$ that is leaked by the transmitter. The translation, from data collected and stored in the Raspberry pi, to a 2-D plot shown in Fig. 3b is accomplished by an algorithm that was developed as part of this work, to care-



(a)



(b)

Figure 3. (a) GnuRadio Signal processing flowgraph of the receiver, (b) Second Harmonic pulse leakage detected by the SDR.

fully detect the $2f_o$ energy of the leaked pulses. We have also developed the software to routinely reference the $2f_o$ leaked pulses with their corresponding azimuth angles to uniquely identify the target's position. Post-COVID-19, we expect to conduct several sensitivity studies of the system to determine maximum detection range, error detection, probability of false alarm, etc., and plan to continue improving the system.

Notice that using both Raspberry Pi and Adalm Pluto as SDR platforms offers a significant advantage when developing software. One of them is using a single language (Python) to communicate and control the two devices. For example, Adalm pluto can be programmed with Python, via *PyADI-IIO*, which is a Python interface capable of simplifying the configuration of the SDR receiver. In addition, Python can easily be used to create additional applications to handle, process, and display real-time data in another computer connected directly to the Rasberry Pi of the SDR receiver. Data can also be uploaded to the cloud enabling additional remote visualization tools, which are quite useful for outdoor deployment of the radar. We plan to add a geo-referencing feature using a GPS (to accurately locate the radar system position) and produce airborne insect flying trajectories in an interactive map. We also intend to construct long term data storage and management tools to study insect trajectories with colleagues from Entomology, Climate Analysis, etc.

In summary, the SDR receiver platform reported here is a simple and inexpensive solution for a receiver implementation; offering a straightforward method to apply signal processing at base-band. The integration of both Adalm-Pluto SDR and Raspberry Pi can rapidly enable open-source platforms, easy to program with Python, portable, and simple to scale-up with other systems. Although the receiver measurements were limited, we plan to perform outdoor experiments as soon as COVID restrictions are less severe. We expect to report on up-graded measurements at the time this paper is presented at URSI.

5 Acknowledgements

The Institutes of Energy and the Environment at Penn State University partly supported the research. The Penn State College of Engineering Seed ROCKET Program provided partial funding to this project.

References

[1] J. L. Osborne, S. J. Clark, R. J. Morris, I. H.Williams, J. R. Riley, A. D. Smith, D. R. Reynolds, and A. S. Edwards, “A landscape-scale study of bumblebee foraging range and constancy, using harmonic radar,” J. Appl. Ecol., vol. 36, pp. 519–533, 1999.

[2] Drake, V. & Reynolds, Don. (2012). Radar Entomology: Observing Insect Flight and Migration.

[3] Mascanzoni, Daniel & Wallin, Henrik. (2008). The harmonic radar: a new method of tracing insects in the field. Ecological Entomology. 11. 387 - 390. 10.1111/j.1365-2311.1986.tb00317.x.

[4] Capaldi EA, Smith AD, Osborne JL, Fahrbach SE, Farris SM, Reynolds DR, Edwards AS, Martin A, Robinson GE, Poppy GM, Riley JR. Ontogeny of orientation flight in the honeybee revealed by harmonic radar. Nature. 2000 Feb 3;403(6769):537-40. doi: 10.1038/35000564. PMID: 10676960.

[5] J.R Riley, A.D Smith, Design considerations for an harmonic radar to investigate the flight of insects at low altitude, Computers and Electronics in Agriculture, Volume 35, Issues 2–3, 2002, Pages 151-169, ISSN 0168-1699, https://doi.org/10.1016/S0168-1699(02)00016-9.

[6] Psychoudakis, Dimitris & Moulder, William & Chen, Chi-Chih & Zhu, Heping & Volakis, John. (2008). A Portable Low-Power Harmonic Radar System and Conformal Tag for Insect Tracking. Antennas and Wireless Propagation Letters, IEEE. 7. 444 - 447. 10.1109/LAWP.2008.2004512.

[7] Tahir, Nazifa & Brooker, Graham. (2015). Towards The Development of Millimeter Wave Harmonic Sensors for Tracking Small Insects. IEEE Sensors Journal. 15. 1-1. 10.1109/JSEN.2015.2445933.

[8] Milanesio D, Saccani M, Maggiora R, Laurino D, Porporato M. Design of an harmonic radar for the tracking of the Asian yellow-legged hornet. Ecol Evol. 2016 Mar 2;6(7):2170-8. doi: 10.1002/ece3.2011. PMID: 27069583; PMCID: PMC4782260.

[9] K. Rasilainen, J. Ilvonen, A. Lehtovuori, J. Hannula and V. Viikari, "On Design and Evaluation of Harmonic Transponders," in IEEE Transactions on Antennas and Propagation, vol. 63, no. 1, pp. 15-23, Jan. 2015, doi: 10.1109/TAP.2014.2366193.

[10] Analog Devices Inc., 'ADALM-PLUTO Overview', 2019. [Online] Available: https://wiki.analog.com/university/tools/pluto. [Accessed: 8-Oct-2020].

[11] Ansys Inc., '3D Electromagnetic Field Simulator for RF and Wireless Design',2020. [Online] Available: https://www.ansys.com/products/electronics/ansys-hfss. [Accessed: 8-Oct-2020].

[12] GNUradio Project, 'Documentation', 2020. [Online] Available: https://www.gnuradio.org/docs/. [Accessed: 8-Oct-2020].