Demo: POWDER over-the-air digital communications virtual learning lab

Neal Patwari*†, Meles Weldegebriel*, Jie Wang*, Frost Mitchell‡,
Aarti Singh*, Cassie Jeng*, Jacobus (Kobus) Van der Merwe‡, and Kirk Webb‡
*Electrical & Systems Engineering; †Computer Science & Engineering
Washington University in St. Louis; Email: {npatwari, g.weldegebriel, jie.w, aartisingh, jeng.c}@wustl.edu
‡ School of Computing; University of Utah
Email: frost.mitchell@utah.edu, {kobus, kwebb}@cs.utah.edu

***Abstract*—This demo will show attendees how to use POWDER as a wireless lab, to provide hands-on student learning of real-world over-the-air digital communications.**

I. Overview of Demo

Most digital communications labs are performed on a lab bench with test equipment, requiring a large investment of resources to own and maintain. This demo[1] exhibits how such labs can be performed on the platform for open wireless data-driven experimental research (POWDER) [1] on links in a city-scale, real-world environment. This lab also provides specific skills in software-defined radio (SDR) and repeatable experimentation.

As an example wireless lab exercise, we present the transmission and reception of a narrowband quadrature phase-shift keying (QPSK) signal at 3.4 GHz. The lab starts with using Python to encode a text message into a QPSK packet. We prepend a preamble and sync word, and modulate using a QPSK constellation and a square root-raised cosine (SRRC) pulse shape, shown in Fig. 1.

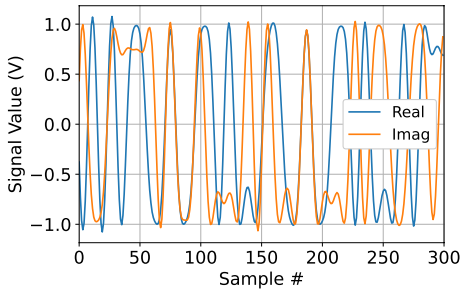

Fig. 1. Part of signal transmitted by Hospital rooftop node, I/Q components, for a QPSK signal with SRRC pulse shape w/ $\alpha = 0.5$.

We create an experiment on POWDER using the Shout [2] measurement profile[2]. Shout is software to orchestrate networked computers, each connected to an SDR, to synchronously transmit or receive as specified. We show in the demo how to specify desired Shout parameters, to use ssh to connect to each compute node, and finally to run Shout and collect the complex baseband received signal (.iq) files from each reception.

Finally, we use a Jupyter notebook to operate a receiver from the .iq files. We provide a library of functions for common signal plots: a power spectral density plot, a temporal plot of the IQ signal, an eye diagram, and a constellation plot; examples are shown in Fig. 2.

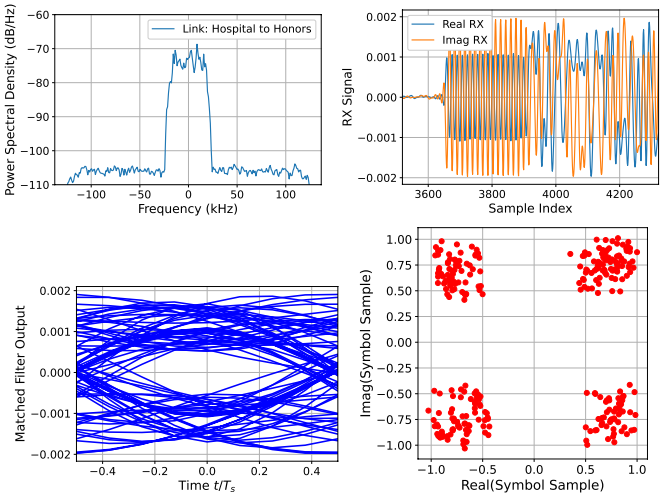

Fig. 2. Signal as received by Honors, transmitted from Hospital: (top left) PSD, (top right) start of packet in time domain; (bottom left) eye diagram prior to phase synch; (bottom right) constellation plot.

Demo participants can operate our QPSK receiver implementation, one step at a time, including frequency, phase, and symbol synchronization; matched filter; and symbol decision block, to obtain the estimate of the text message.

This material is based upon work supported by the US NSF under Grants #2232464, 2229427, and 1827940.

[1]https://github.com/npatwari/tx_rx_processing

[2]https://gitlab.flux.utah.edu/frost/proj-radio-meas

References

[1] J. Breen, A. Buffmire, J. Duerig, K. Dutt, E. Eide, A. Ghosh, M. Hibler, D. Johnson, S. K. Kasera, E. Lewis, D. Maas, C. Martin, A. Orange, N. Patwari, D. Reading, R. Ricci, D. Schurig, L. B. Stoller, A. Todd, J. Van der Merwe, N. Viswanathan, K. Webb, and G. Wong, "POWDER: Platform for open wireless data-driven experimental research," *Computer Networks*, vol. 197, Oct. 2021.

[2] K. Webb, J. Van der Merwe, S. K. Kasera, and N. Patwari, "WiMatch: wireless resource matchmaking," in *IEEE INFOCOM CNERT Workshop*, May 2021.