End-to-End ISAC Simulation: OFDM vs OTFS under Delay–Doppler Channels

Haytam Zerdani

17/09/2025

Abstract

This report documents a minimal end-to-end Integrated Sensing and Communication (ISAC) simulator comparing CP-OFDM and OTFS waveforms under parametric delay–Doppler channels. It produces ambiguity-like range–Doppler (RD) maps and proxy BLER vs Doppler curves from hard decisions. The code is modular (Python), fast to run, and designed for later insertion of proper equalizers and CFAR-based sensing KPIs.

1 Objectives

- Generate OFDM and OTFS frames with shared simulation blocks.
- Pass both through the same multi-path delay–Doppler channel.
- Produce RD maps and proxy BLER vs Doppler plots.
- Provide clean hooks for true BLER and sensing ROC in later work.

2 System Model

A discrete-time complex baseband signal $x[n]$ passes a parametric delay–Doppler channel

$$y[n] = \sum_k \alpha_k \, x[n - \tau_k] \, e^{j2\pi\nu_k n/f_s} + w[n], \tag{1}$$

with fractional delays τ_k, Dopplers ν_k, complex gains α_k, sampling rate f_s, and AWGN $w[n]$.

OFDM. QPSK on N_{sc} subcarriers and N_{sym} symbols; IFFT + CP per symbol.

OTFS. QPSK on a delay–Doppler grid $M \times N$; ISFFT to TF domain then per-time-slot IFFT to time domain.

3 Methodology

Ambiguity proxy. For each waveform, split the time series into overlapping blocks of length L. Compute a frequency-domain autocorrelation per block (fast-time), then a slow-time FFT to obtain a magnitude RD map.

Proxy BLER. OFDM: FFT per symbol after CP removal; compare hard decisions to the known transmit grid (no equalization). OTFS: SFFT back to DD; compare hard decisions to the transmitted DD symbols (no equalizer). These are proxies; proper BLER needs equalizers.

4 Implementation Layout

- `waveform/otfs.py` — OTFS ISFFT/SFFT + QPSK.
- `comm_rx/simple_ofdm.py` — OFDM Tx/Rx and hard-decision proxy.
- `channel/parametric_dd.py` — delay–Doppler paths.
- `metrics/ambiguity.py` — RD proxy.
- `metrics/plots2.py` — plotting helpers.
- `sims/e2e_isac.py` — runs the comparison and saves figures.

5 Receivers (baseline)

OFDM Rx. CP removal, FFT, hard decisions by sign on I/Q; no channel estimation or equalization.

OTFS Rx. Reshape to TF, SFFT to DD, hard decisions by sign on DD symbols; no equalization.

6 Key Parameters

Parameter	Default
f_s	15.36 MHz
OFDM: $(N_{\text{sc}}, N_{\text{sym}}, \text{CP})$	$(256, 32, 32)$
OTFS: (M, N)	$(32, 32)$
Doppler sweep	$\{0, 100, 200, 300, 400\}$ Hz
Noise SNR	20 dB
Ambiguity: $(L, K, \text{overlap})$	OFDM $(\min(2048, \lvert x\rvert), 32, 0.5)$; OTFS $(\min(1024, \lvert x\rvert), 32, 0.5)$

7 Results

8 Limitations and Next Steps

- **Equalization.** Add LS/MMSE for OFDM and a DD-domain equalizer for OTFS to turn proxies into real BLER/throughput metrics.
- **Sensing KPIs.** Add CFAR on RD maps to report P_d vs P_{fa} and RMSE(range, velocity).
- **Resource split.** Reserve α PRBs or power for sensing pilots and plot throughput vs P_d Pareto curves.
- **Impairments.** CFO, phase noise, quantization, clutter models.
- **Validation.** Ingest real IQ logs via `data/iq_logs/`.

9 Reproducibility

Run.

```
python - <<'PY'
import sys
sys.path.append('/mnt/data/isac')
from sims.e2e_isac import run
run() # saves figures under data/results/
PY
```

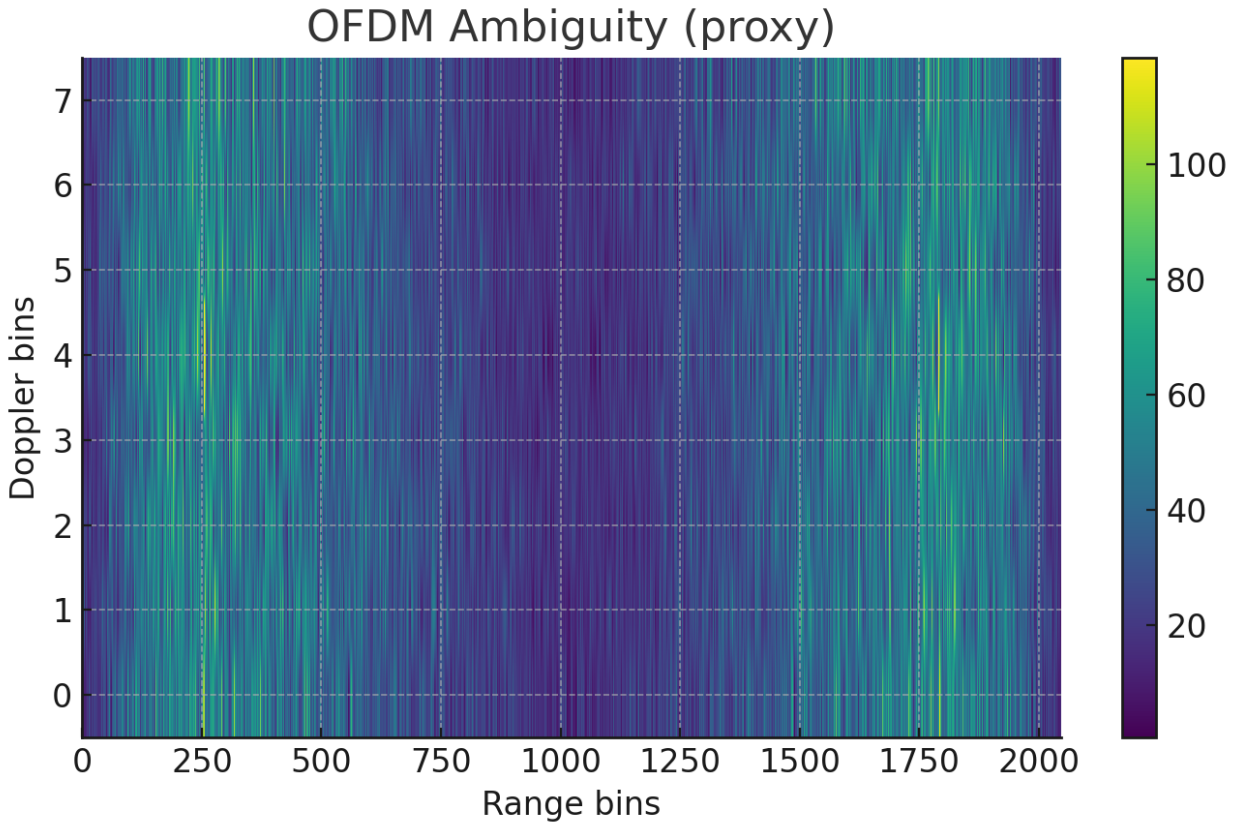

Figure 1: OFDM ambiguity proxy (range–Doppler).

Artifacts. Figures saved as `ofdm_ambiguity.png`, `otfs_ambiguity.png`, `bler_vs_doppler.png`.

10 Conclusion

The simulator compares OFDM and OTFS under the same delay–Doppler conditions and produces interpretable RD maps and Doppler stress curves. It is ready for equalization, CFAR, and resource-sharing studies needed for ISAC trade-offs.

Repo: `/mnt/data/isac` **Zip:** `ISAC_EndToEnd_OFDM_vs_OTFS_Demo.zip`

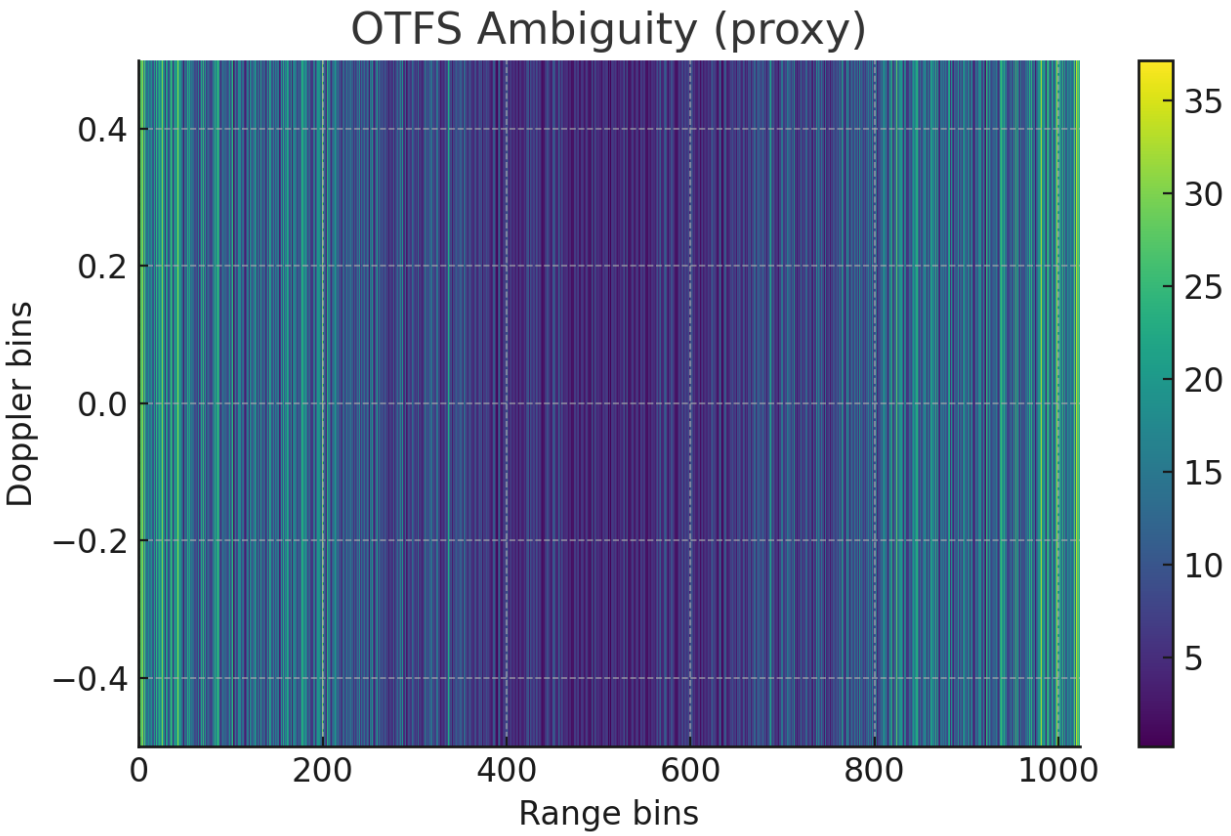

Figure 2: OTFS ambiguity proxy (range–Doppler).

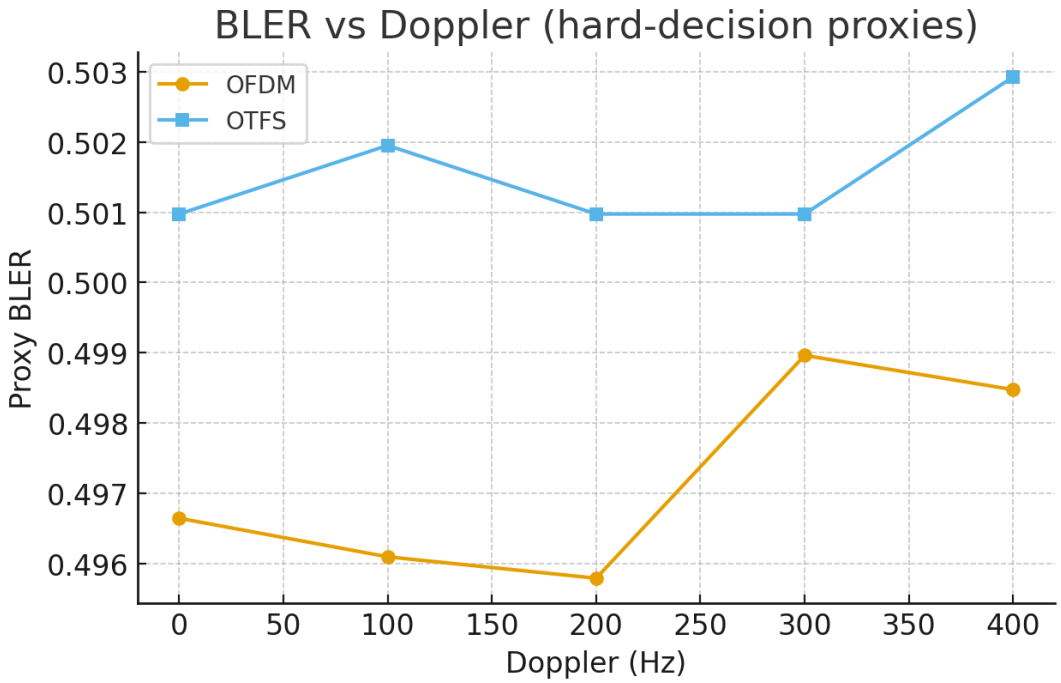

Figure 3: Proxy BLER vs Doppler (hard decisions, no equalization).