

## Pulse-Doppler Radar

Functions: pulse compression, 3-pulse cancellation, Doppler computation, FFT

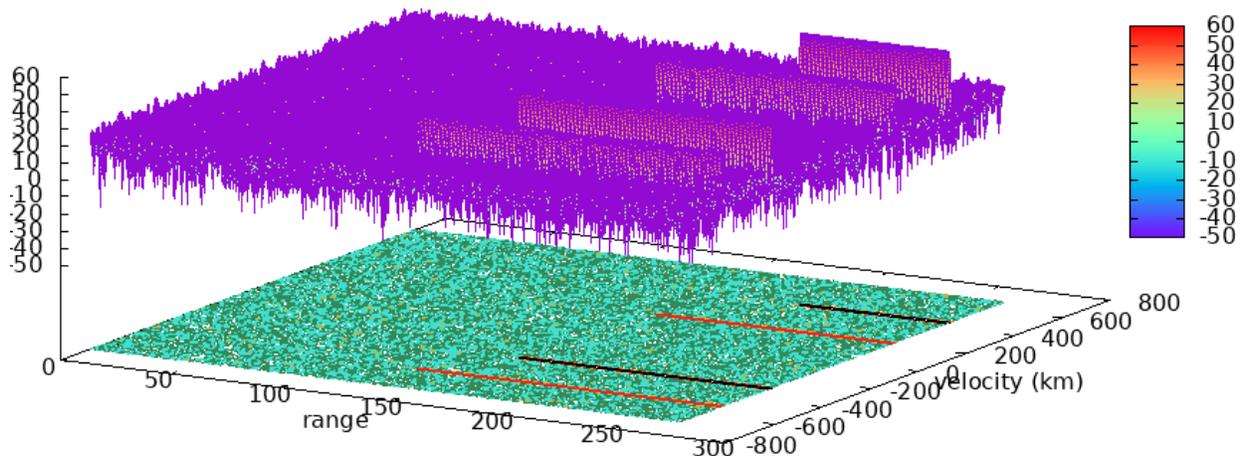
Inputs: simulated radar returns

Metrics: per-stage and end-to-end performance, target position/velocity accuracy

### 1. Overview

Pulse-Doppler radar transmits a series of radar pulses and uses the received signal reflections to determine the position and velocity (per the Doppler effect) of objects it detects. It is commonly used for tracking airborne targets or in weather applications. The characteristics both of the signal processing pipeline and of the radar pulses themselves (frequency, duration, repetition interval, etc.) strongly affect the ability of the system to detect and disambiguate objects at different ranges or moving at different speeds. For instance, a system that performs well with close-range, fast-moving objects may struggle to accurately detect long-range, slow-moving targets. For further reading, see [https://www.eetimes.com/document.asp?doc\\_id=1278808#](https://www.eetimes.com/document.asp?doc_id=1278808#).

RANGE-DOPPLER PLOT OF UNPROCESSED DATA



MilSpec's pulse-Doppler radar reference implementation is based on <https://github.com/yanyanggithub/doppler>. This implementation generates a simulated radar signal and then processes it with a four-step pipeline to find Doppler peaks and thereby detect object positions/velocities. The MilSpec version of this code simplifies usage, refactors various aspects of the code, includes enhanced facilities for custom radar signal generation (along with multiple predefined test scenarios), and adds performance instrumentation for processing speed and detection accuracy. A future revision will add constant false alarm rate (CFAR) processing.

## 2. System requirements

Platform: Ubuntu 18.04 LTS with g++ 7.4.0. Code may build and run successfully on other versions/platforms, but has not been tested with them.

Storage: ~5MB for code and sample radar returns. Plots for these sample returns (plus their intermediate work products) may require up to ~60MB.

Dependencies: None for baseline simulation. gnuplot is required to view plots of radar data (tested with version 5.2, patchlevel 2).

## 3. Build and run

To benchmark:

- Download and extract the zipfile from [www.adacenter.org/milspec](http://www.adacenter.org/milspec)
- From the PulseDoppler/ directory, make clean && make
- Choose a testcase: basic (few, slow-moving targets), mid (many targets), or fast (few, fast-moving targets).
- From the PulseDoppler/ directory, ./simulate [testcase].signal
- Results are displayed in the terminal, as below. Reference output for comparison is available in [testcase].expect:

```
Inputted Targets:
Target #      Range (m)      SNR      Velocity (m/s)
0             800           -3       -500
1             950            5       -300
2            1000          -1        210
3            1250            7        440

Pulse Compression Duration: 0.221168s
3-Pulse Canceller Duration: 0.631161s
Range Bins Search Duration: 0.00144333s
Doppler Peak Search: 4.523e-05s
Total doppler process time: 0.853817s

Detected targets:
Range (m)      Velocity (m/s)
800            -499.251
950            -299.247
993.75         210.612
1243.75        440.382

Error Percentages:
Target #      Range (%)      Velocity (%)
0             0.00          0.15
1             0.00          0.25
2             0.62          0.29
3             0.50          0.09
```

To view plots:

- Install gnuplot: `sudo apt-get install gnuplot`
- From the PulseDoppler/ directory, `make clean && make plot`
- Choose a testcase: `basic` (few, slow-moving targets), `mid` (many targets), or `fast` (few, fast-moving targets).
- From the PulseDoppler/ directory, `./signal_plot [testcase].in`
- From the PulseDoppler/ directory, `./simulate_plot [testcase].signal`
- From the PulseDoppler/ directory, `gnuplot plot.plt`
- Assorted `.png` images are generated in the working directory.

#### **4. Code structure**

Coming soon!

#### **5. MilSpec development notes, errata, changelog**

v0.9:

- Changes from baseline pulse-Doppler code for MilSpec:
  - Simplified command-line usage and code internals
  - Split out signal generation code into a separate, highly configurable program
  - Created baseline test scenarios
  - Added performance metrics and position/velocity error calculation
  - Created makefile

#### **6. Acknowledgements**

MilSpec is a project under development at the University of Michigan – Ann Arbor by Pete Ehrett, Bing Schaefer, Adrian Berding, John Paul Koenig, Pranav Srinivasan, Todd Austin, and Valeria Bertacco. This project is supported by the Applications Driving Architectures Center, one of six centers of JUMP, a Semiconductor Research Corporation program co-sponsored by DARPA.

The original pulse-Doppler radar processing implementation upon which the MilSpec version is based is available at <https://github.com/yanyanggithub/doppler>.