Modernization of a Toolbox for Assessing HF Radar and Drifter Derived Surface Current Velocities

Timothy Stolarz
Center for Ocean Observing Leadership
Rutgers University
New Brunswick, NJ USA
tstolarz@marine.rutgers.edu

Hugh Roarty
Center for Ocean Observing Leadership
Rutgers University
New Brunswick, NJ USA
hroarty@marine.rutgers.edu

Laura Nazzaro
Center for Ocean Observing Leadership
Rutgers University
New Brunswick, NJ USA
nazzaro@marine.rutgers.edu

***Abstract*— The Python hfr-drifters toolbox is a conversion of a MATLAB toolbox designed to analyze and visualize ocean surface velocities by comparing drifter-derived velocities against measurements from High Frequency (HF) Radars. Drifters provide in-situ measurements for comparison with HF Radars. The original MATLAB toolbox (hfr-drifters) was created for validating Rutgers owned HF radar systems by assessing both an individual site's radial velocity and the networks combined total velocity accuracy. This conversion seeks to modernize the toolbox by making it accessible in the widely adopted programming language Python, while also introducing new features and enhanced flexibility for data output. The MATLAB toolbox and Python conversion contribute to the National Oceanic and Atmospheric Administration's (NOAA) initiative to advance Quality Assurance Real Time Oceanographic Data (QARTOD) standards. The toolbox incorporates quality control methods such as velocity rate-of-change, standard deviation, and positional distance checks (or peak tests), ensuring accurate and reliable data comparisons on the radial and total velocity data. Additionally, the toolbox provides analytical capabilities for assessing total velocity products, which integrate radial velocity data from multiple radar sites into comprehensive surface current maps. Visualization capabilities of the toolbox include time series plots comparing drifter velocities with HF radar radials and totals, scatter plots displaying velocity correlations, geospatial visualizations of drifter trajectories with velocity vectors, and current roses which offer insights into directional and speed distributions. Rutgers researchers utilize these insights to verify radar data quality and support ongoing research and projects. Converting the MATLAB toolbox to Python creates pathways to integrate numerous other oceanographic packages and libraries into the analyses while also making the validation methodology more available to the community.**

Keywords—High Frequency Radar, Drifters, MATLAB, Python, Currents

I. Introduction

The coastal ocean of the Mid-Atlantic Bight (MAB) is a complex system that influences the lives of 61 million people within the region. Understanding the surface current environment of the MAB can help address Coast Guard Search and Rescue operations, harmful algal blooms, pollutant tracking, and ecosystem research [1, 2]. Rutgers University owns and operates a network of High Frequency Radars (HFR) which are coastal sensors that can measure surface current velocities in near real time. This network of 23 HFRs spans from Nantucket Island, MA, to Lewes Delaware, DE. Rutgers University's HFRs operate as part of a larger network within the Mid-Atlantic Regional Association Coastal Ocean Observing System (MARACOOS) covering more than 190,000 km^2 of coastal ocean [3]. HFRs in the Rutgers network collect data at three frequencies, each with differing spatial resolutions and coverage extents, reflecting a compensation between resolution and range. The 5 MHz stations provide the lowest resolution (6 km) but cover a broad area from the coastline to the shelf-break. In contrast, the 13 MHz stations offer higher resolution (3 km) but cover a narrower area from the New Jersey coast to mid-shelf. Finally, the 25 MHz stations provide the highest resolution (1 km), focusing coverage specifically around the mouths of Raritan Bay and Delaware Bay [4].

In-situ ocean drifters are another source of surface current data [5] and offer an ideal Lagrangian dataset for comparison with HFR data. Drifters that pass through the coverage field of HFRs can provide a means of comparing surface velocity data from the HFR against the drifter to verify instrument accuracy. Unfortunately, products of this kind of comparison are not produced in an automated, routine fashion.

In 2021, a MATLAB toolbox was created to process surface drifter data into a flexible format and then compare that data against HFR radial velocity and total vector data [6]. This tool is a powerful aid for HFR operators for assessing the accuracy of their reported surface current data. The MATLAB toolbox outputs a variety of figures for each selected HFR site in the drifter's track at the user's discretion as well as total vector data comparison figures.

This work introduces a modular framework that translates the MATLAB toolbox into Python. The Python toolbox provides updates in the form of user-set QC metrics, additional comparative figures and offers the user the ability to choose the desired output images if only a certain type of analysis is requested. This is the first version of the Python toolbox, and it is expected to be updated over time with user requested features of new comparative figures and additional forms of statistical analysis.

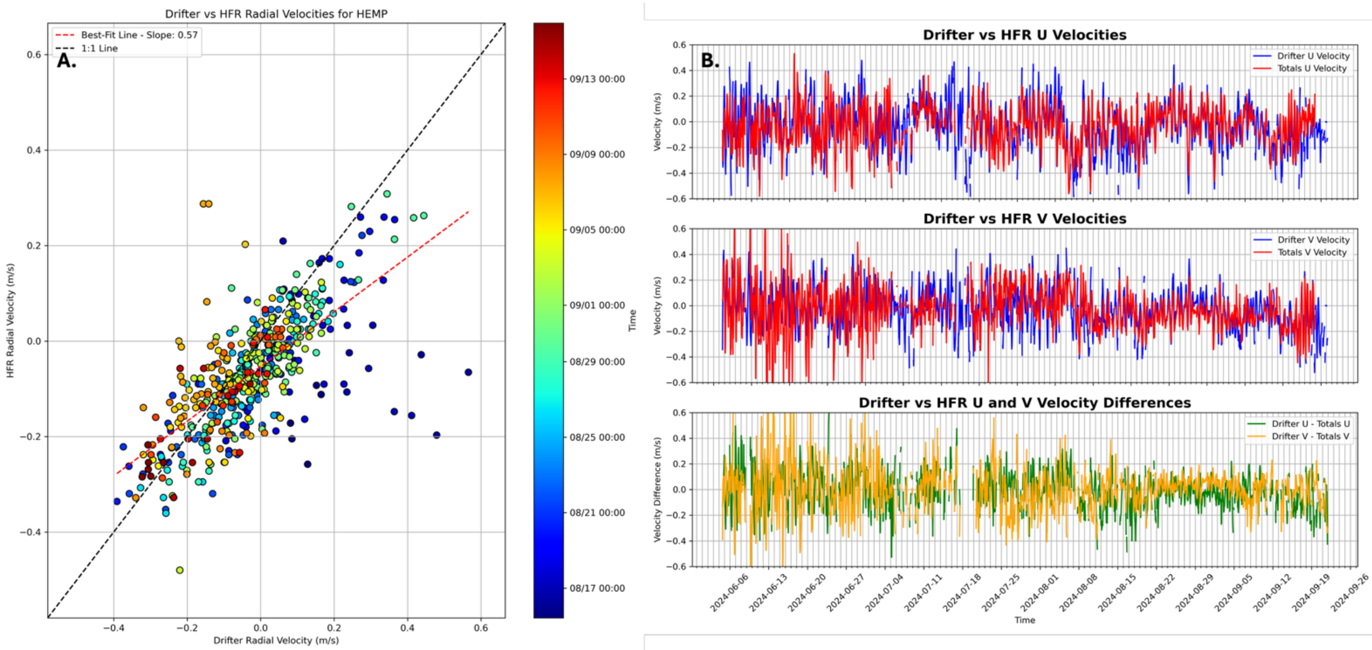

Figure 1. Two example figures from the new hfr-drifters Python toolbox. (A.) showcases a scatter plot of derived drifter radial velocity vs the radial velocities of the HFR station with a 1:1 and calculated line of best fit for the 5 MHz site in Hempstead, New York (HEMP). (B.) showcases time series comparisons of the totals u and v velocity components against the drifter-derived u and v velocity components. The final subplot in green and yellow showcases a time series of the difference of the u and v velocities calculated with drifter velocity minus totals velocity

II. Methods

The Python hfr-drifters toolbox translation seeks to replicate two of the major functions of the MATLAB toolbox:

1. The comparison of derived drifter radial velocities against radial velocities of selected nearby HFR sites.

2. The comparison of derived drifter velocities against nearby HFR total vectors.

Drifter data input into the toolbox must be preprocessed to remove duplicate times and resampled to hourly timestamps. The methods used for cleaning the drifter data can be found at the following GitHub repository: https://github.com/Tstolarz/clean_drifter_data. Velocities for the drifter were calculated using finite difference methods between consecutive fixed positions [7]. Distances were computed using the WGS84 ellipsoid via the pyproj geodesic calculator, and bearing angles were determined from the forward azimuth. Velocity magnitudes were calculated as distance divided by time interval, with eastward (u) and northward (v) components derived from:

$$u = |v| \times \sin(\theta)$$

$$v = |v| \times \cos(\theta)$$

where $|v|$ is velocity magnitude and θ is the bearing angle relative to true north in radians.

Three quality control filters can be applied to the drifter dataset depending on user preferences. The first filter is the rate of change filter which removes data that has a rate of change greater than a set threshold (default: 0.15 m/s) between consecutive hourly measurements [8]. The second filter is the standard deviation which removes velocity values deviating more than a set standard deviation (default: 3) from the mean velocity. The final filter is the distance filter, or effectively a velocity spike test, which removes position changes exceeding a threshold (default 10 km) which are likely produced due to GPS reporting errors [8].

HF Radar data for analysis were gathered from two sources. Radial files for site radial velocity comparisons are hosted and archived on a file server at Rutgers and are gathered from the server for processing. HFR total vector data were gathered from Rutgers University's Department of Marine and Coastal Sciences THREDDS server (https://tds.marine.rutgers.edu/thredds/catalog/cool/codar/cat_totals.html). Data for each resolution of the HF Radar systems is available for use with the Python drifter toolbox.

For radial data processing, the script determines the subset of HFR sites that fall within a frequency-dependent proximity radius of the drifter's position. Proximity thresholds of the site frequencies are as follows: 5 MHz (175 km), 13 MHz (100 km), 25 MHz (40 km). These proximity thresholds are based on the expected range for those systems at the selected frequency. For each selected site, an additional constraint is applied to ensure a closer data comparison. Each moment the drifter is within range of the radar, if the drifter's location is not within the range resolution of a velocity vector, that data is not used for analysis.

For every drifter position, the script identifies all HFR sites' data whose location to the drifter is less than the user-defined search radius. Range and bearing from the antenna are computed with the WGS84 ellipsoid, and the corresponding range cell and bearing bin are extracted from each site's radial file. The drifter radial velocity is then calculated using the azimuth from the antenna. Once a candidate drifter has been cleaned and velocity estimates computed, the toolbox proceeds through four sequential stages: 1) site selection, 2) spatio-temporal colocation, 3) statistical evaluation, and 4) diagnostic output. Each stage is controlled by keyword arguments read from a user-editable configuration file, allowing the workflow to be reproduced without modifying source code.

Only radial vectors with a vector quality flag of VFLG = 0 are retained. Vectors flagged VFLG > 0 are removed because they have been excluded by the operator-defined AngSeg masks which limit processing to selected bearings, ranges, or individual range–bearing bins or because the processing software detected a location error (e.g., a vector plotted over land) [4]. AngSeg masks are configured independently for each radar site, so this screening's flags are applied differently on a site-by-site basis. Additionally, drifter locations must fall within a frequency-specific distance threshold in order to compare the radial velocity measurement with the radial drifter velocity. The distance thresholds are as follows: 5 MHz (6 km), 13 MHz (3 km), and 25 MHz (1 km).

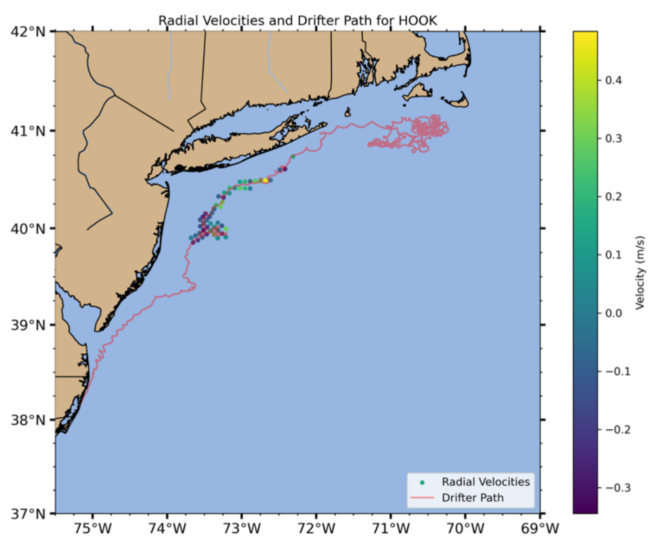

Figure 2. A map showing the entire track of the GPS drifter (red line) and locations where radial data was pulled (colored dots) for comparison with the 5 MHz site in Sandy Hook, NJ (HOOK). The colorbar indicates radial velocity at the time of comparison with the drifter.

Surface current totals data are obtained from one of three THREDDS server datasets corresponding to radar operational frequencies outlined previously. Users select their preferred dataset by setting the appropriate configuration flag. Gathering the nearest total vector to the drifter uses either nearest-grid point extraction or bilinear interpolation methods, configurable via user preference. For a drifter velocity to be used for comparison, it must be within the proximity of a total vector determined by the frequency-range resolution of the total product used, consistent with the radial processing.

III. Results

Matched pairs of drifter-derived radial velocities and radar-observed radial velocities are aligned into two separate time series. These time series are evaluated to produce statistical metrics, including root mean square error (RMSE), correlation coefficients, and linear regression values. The toolbox generates configurable image outputs including radial velocity time series comparisons (Figure 1), radial velocity difference plots, radial velocity timeseries with distance-to-site overlays, radial velocity correlation scatterplots, and drifter trajectory maps overlaid with points where radial data was taken from coverage fields (Figure 2). These visualizations are selected via a user-editable dictionary.

Once matched, standard comparison metrics, including total velocity $RMSE_u$, $RMSE_v$, and complex correlation are computed to assess agreement between datasets. Configurable image outputs for the totals toolbox include drifter trajectory maps with totals vectors overlay, current rose diagrams comparing directional distributions, u and v component time series, side-by-side animation frames for dynamic analysis, and correlation scatterplots for u and v velocity components. All outputs are configurable via a user-editable dictionary.

All parameters utilized in the drifter and radar data processing workflows, including file paths, QC thresholds (rate-of-change, standard deviation, distance-spike), figure generation flags, and dataset selection criteria, are defined in a central configuration section at the top of a configuration file, again ensuring reproducibility and ease of use without the need to modify underlying source code directly.

IV. Discussion

Drifter data provide an essential baseline for evaluating and improving the accuracy of HFR surface current measurements [9]. By directly comparing drifter velocities with radar measurements, users can identify and quantify errors due to a variety of potential factors including instrument calibration, spatial and temporal coverage/resolution discrepancies, noise, and interference. The calibration potential of drifter data is particularly significant in maintaining and enhancing the operational quality of HF radar systems. Cross-validation with drifters ensures radar systems remain reliable and accurate, facilitating effective monitoring of coastal ocean conditions.

The resulting images generated from the toolbox offer visual assessments of data quality. By inspecting trajectory overlays, velocity differences, and correlation scatterplots, operators can assess anomalies, spatial biases, or systematic errors in radar measurements, enabling operators to take actions to improve data quality where possible or suggest improvements to the system manufacturer.

V. Conclusion

The Python-based hfr-drifters toolbox is a flexible and intuitive package that enhances HF radar operators' ability to quality control and validate HF radar-derived surface currents. By leveraging drifter measurements for validation, radar data accuracy and reliability can be improved, directly benefiting oceanographic research, operational coastal monitoring efforts and search and rescue. The resulting images facilitate finding times and areas of higher error between the two datasets to assess the health and data quality of the operating HF radar systems. This toolbox represents an essential modernization of oceanographic toolboxes into the more accessible, open-source format using Python.

Acknowledgments

This work was funded by NOAA Award Number NA16NOS0120020 "Mid-Atlantic Regional Association Coastal Ocean Observing System (MARACOOS): Powering Understanding and Prediction of the Mid-Atlantic Ocean, Coast, and Estuaries". Sponsor: National Ocean Service (NOS), National Oceanic and Atmospheric Administration (NOAA) NOAA-NOS-IOOS-2021-2006475, Integrated Ocean Observing System Topic Area 1: Implementation and Development of Regional Coastal Ocean Observing Systems.

References

[1] J. Kohut *et al.*, "The Mid-Atlantic Regional Coastal Ocean Observing System: Serving coast guard needs in the mid-atlantic bight," in *2008 IEEE/OES US/EU-Baltic International Symposium*, 27–29 May 2008 2008, pp. 1–5, doi: 10.1109/BALTIC.2008.4625502.

[2] H. Roarty, A. Allen, S. Glenn, J. Kohut, L. Nazzaro, and E. Fredj, "Evaluation of Environmental Data for Search and Rescue II," in *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, 28–31 May 2018 2018, pp. 1–3, doi: 10.1109/OCEANSKOBE.2018.8559228.

[3] H. Roarty *et al.*, "Annual and Seasonal Surface Circulation Over the Mid-Atlantic Bight Continental Shelf Derived From a Decade of High Frequency Radar Observations," *Journal of Geophysical Research: Oceans,* vol. 125, no. 11, p. e2020JC016368, 2020, doi: https://doi.org/10.1029/2020JC016368.

[4] H. Roarty, T. Updyke, L. Nazzaro, M. Smith, S. Glenn, and O. Schofield, "Real-time quality assurance and quality control for a high frequency radar network," (in English), *Frontiers in Marine Science,* Original Research vol. Volume 11 - 2024, 2024–May–06 2024, doi: 10.3389/fmars.2024.1352226.

[5] D. Kelly, "Largest Drifter Project Ever Improves Oil Movement Forecasts in the Gulf of Mexico," *Gulf of Mexico Research Initiative (GoMRI) News,* 2012/11/27 2012. [Online]. Available: https://gulfresearchinitiative.org/largest-drifter-project-ever-improves-oil-movement-forecasts-in-the-gulf-of-mexico/.

[6] H. Roarty and L. Nazzaro, "Mid Atlantic Drifter Program: Development of Software Toolbox to Manage Drifter Data," in *OCEANS 2021: San Diego – Porto*, 20–23 Sept. 2021 2021, pp. 1–5, doi: 10.23919/OCEANS44145.2021.9706065. [Online]. Available: https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp=&arnumber=9706065&ref=

[7] D. V. Hansen and P.-M. Poulain, "Quality Control and Interpolation of WOCE/TOGA Drifter Data," *Journal of Atmospheric and Oceanic Technology,* vol. 13, no. 4, pp. 900–909, 1996, doi: 10.1175/1520-0426(1996)013<0900:QCAIOW>2.0.CO;2.

[8] "Manual for Real-Time Quality Control of In-Situ Current Observations QARTOD manual version 2.1," Manuals & Handbooks 2019, doi: https://doi.org/10.25923/sqe9-e310.

[9] C. Ohlmann, P. White, L. Washburn, B. Emery, E. Terrill, and M. Otero, "Interpretation of Coastal HF Radar–Derived Surface Currents with High-Resolution Drifter Data," (in English), *Journal of Atmospheric and Oceanic Technology,* vol. 24, no. 4, pp. 666–680, 01 Apr. 2007 2007, doi: https://doi.org/10.1175/JTECH1998.1.