Fast linearized coronagraph optimizer (FALCO) I: a software toolbox for rapid coronagraphic design and wavefront correction

A J Eldorado Riggs[a], Garreth Ruane[b], Erkin Sidick[a], Carl Coker[a], Brian D. Kern[a]

[a]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109

[b]California Institute of Technology, Pasadena, CA 91125

ABSTRACT

The Fast Linearized Coronagraph Optimizer (FALCO) is a free toolbox of routines for coronagraphic focal plane wavefront correction. The goal of FALCO is to provide a modular, open-source framework for the simulation and/or testbed operation of several common types of coronagraphs. FALCO includes routines for pair-wise probing estimation of the complex electric field and Electric Field Conjugation (EFC) control, and we ask the community to contribute other wavefront correction algorithms. FALCO utilizes and builds upon PROPER, an established optical propagation library. The key innovation in FALCO is the rapid computation of the linearized response matrix for each deformable mirror (DM), which facilitates re-linearization after each control step for faster DM-integrated coronagraph design and wavefront correction experiments. FALCO is freely available online as source code in MATLAB at *github.com/ajeldorado/falco-matlab* and, in a later release, will be available in Python.

Keywords: wavefront control, wavefront estimation, coronagraph, vortex, apodizer, Lyot, segmented, exoplanet, PROPER, optical modeling

1. FALCO DIRECTORY STRUCTURE

- **config**: Functions for loading default variable values (for the chosen coronagraph type) and generating the initial workspace.
- **data**: Parent directory for directories storing large files that should not be pushed to git.
  - **configs**: Minimalistic Matlab workspaces containing the key variable values for each trial.
  - **ws**: Matlab workspaces from each trial of the output values.
  - **ws_inprogress**: Minimalistic Matlab workspaces containing the DM commands and average contrast values after each correction iteration. These workspaces are a safety net in case the Matlab crashes partway through a trial.
- **docs**: Documentation slides and manual go here.
- **DoesNotSync**: Directory that does not push any contents to git. Useful for local test/development code.
- **eval**: User-written evaluation scripts and functions to analyze stored output data.
- **falco_wfsc_loop.m**: High-level function in which the WFSC loop and other high-level calculations are performed.
- **lib**: Parent directory for directories of function libraries.

Send correspondence to A.J. Riggs: `aj.riggs@jpl.nasa.gov`

- **cvx**: (OPTIONAL) If you want, download and install CVX professional from here. Note that the free CVX is too slow to use with FALCO. The CVX controller is still in development, so most users will not need or want to download CVX for FALCO.
- **dm**: DM model functions for creating influence function datacubes or creating a surface from those datacubes. Does not contain *prop_dm.m* since that is in the PROPER library folder.
- **masks**: Functions for generating various pupil plane and focal plane masks, and files of existing mask representations (especially pupils).
- **propcustom**: Custom functions for optical propagation, such as some for matrix Fourier transforms (MFTs). Inputs to these functions are NOT the same as for PROPER functions.
- **PROPER**: (Only if you don't have PROPER installed elsewhere and its path included in your Matlab path.) Folder for John Krist's Matlab PROPER optical propagation library. Obtain from here and download in this folder.
- **utils**: Various functions
- **wfsc**: Functions for estimation or control in the final, post-coronagraph focal plane.

- **main**: Folder to contain main scripts for WFSC. This is where you keep your main scripts to run individual trials or looped trials over different parameters. Not synced into git except for template files starting with TEMPLATE_*.
- **models**: Each coronagraph type has 3 model types which let you use different types of sampling for each model. As such, they run at different speeds.

  Really only used to simulate the bench as if the intensity images were coming directly from the bench camera. Not used for testbed experiments (since the testbed is the full model). The example "full" models in FALCO are really just the same as the compact models in form. Use PROPER to simulate a true lab setup with many optical surfaces and aberrations.

  1. Jacobian model: Efficient model that directly computes the first-order Taylor series expansion term of the change in E-field at the final focal plane for each deformable mirror actuator.
  2. Compact model: Simplified model that includes everything known that you want to include in the model. Can be made lower resolution as desired at the expense of accuracy.
  3. Full model: Comprehensive, high-resolution model used only in simulation to produce a "true" electric field in lieu of a testbed. This model can include all optics and aberrations; PROPER is a good tool for building this full model of a testbed or instrument.

- **testing**: Scripts and functions to test or validate specific parts of FALCO. Currently, the only testing scripts are to compute the accuracy of the Jacobian compared to the compact and full models.

1

Acknowledgements

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

[1] Krist, J., "Proper: an optical propagation library for IDL," in [*Proc. of SPIE*], **6675**, 66750P (2007).