

# The pyradi radiometry toolkit

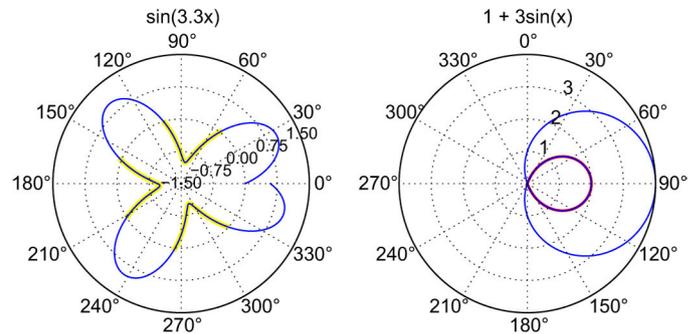
Cornelius Willers, Johannes Calitz, Alta de Waal,  
Azwitamisi Mudau, Maria Willers, Pieter van der Merwe, and  
Ricardo Tavares Santos

*The pyradi toolkit is designed to simplify many common, repetitive calculations researchers must perform while modeling electro-optical systems.*

Modeling and designing in electro-optical systems entails the calculation of several (often interrelated) parameters. Many of these calculations are repetitive, suitable for including in a generic toolkit. A well-designed kit would facilitate work flow and increase productivity during the modeling and design process. The concept of an extendable toolkit lends itself naturally to the open-source philosophy, where users cooperatively develop new tools to add to an ever-expanding set for the mutual benefit of all. The pyradi toolkit is an extendable, integrated and coherent collection of basic functions that can be applied towards diverse calculations in the electro-optics domain.<sup>1</sup> The name pyradi is derived from the combination of 'Python' and 'Radiometry.'

We initially considered two candidate languages for pyradi, MATLAB and Python. MATLAB has a strong following in the scientific community. In recent years Python and associated modules have been well tested, and have stabilized and matured sufficiently to support mainstream tool development. But after extensive use of both languages for radiometric calculation and modeling, we decided to continue only with Python. With this application and its features in mind, Python provides better capability as a general purpose language, and its data visualization tools, Matplotlib and Mayavi, are the most powerful available today. While the wider range of toolboxes (including Simulink) might be a compelling reason to use MATLAB, those capabilities are not required here.

We have already designed a number of pyradi modules covering basic electro-optical system calculations. For instance, the `ryplanck` module provides functions for Planck Law emittance calculations, as well as the Planck Law temperature derivative functions. Given the temperature and spectral vector, the functions provide spectral emittance in  $W/(m^2 \cdot *)$  or  $q/(s \cdot m^2 \cdot *)$ , with spectral variable  $*$  in wavelength,



**Figure 1.** Example polar plots: note axes conventions and yellow/red highlight of negative values.

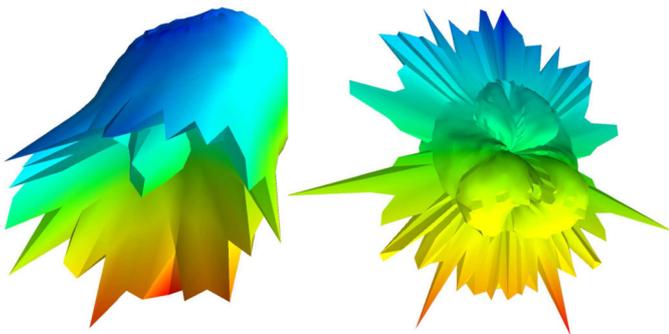
wavenumbers and frequency units. The `ryfiles` module provides functions for file input/output. Functions are provided to save a two-dimensional array to a text file, load selected columns of data from a text file, load a column header line, and read and write raw binary files. Matplotlib functions for Cartesian plots, polar plots (see Figure 1), three-dimensional plots (Figure 2) and image display (Figure 3) are wrapped by the `ryplot` module, which handles negative values in polar plots so as not to confuse negative values with a 180° phase shift. We included various utility functions for radiometry calculations in the `ryutils` module. Modeling a maximally flat spectral filter or a simple photon detector spectral response (see Figure 4), effective value calculation, spectral convolution, and conversion between spectral domain variables can all be handled by `ryutils`. The `ryptw` module provides the capability to read FLIR Inc.'s \*.ptw (Cedip) files, used for thermal imaging data. All information in the file header is read and made available with the image frames. Calculating three-dimensional noise parameters in an image sequence can be done with the `ry3noise` module (see Figure 3). The `rydetector` module provides a bulk material detector model for substances such as silicon, germanium, or indium antimonide, based on academic and well known parameters and models found in the classical literature. We also included functionality to read atmospheric spectral data from Modtran tape7 files given the column header names with the `rymodtran` module (see Figure 5). All four variations of the atmospheric scattering mode (IEMSCT) are accounted for, hence all formats of tape7

*Continued on next page*

files can be read. And the `ryplotspherical` module provides the capability to do three-dimensional spherical plots of target signatures or color ratios (Figure 2).

We manage the `pyradi` code in accordance with general practice in the open source community, and we make it available under the terms of the Mozilla Public License 1.1. The code is available on Google Code.<sup>2</sup> The toolkit files can be downloaded one-by-one using a web browser. We recommend, however, that a subversion client be used to check out and commit changes to the existing code or to add new features (instructions are given on the website).

The `pyradi` documentation is available using a link from the `pyradi` website.<sup>2</sup> Documentation is extracted from the comments in the Python code files, using the Sphinx documentation gener-



**Figure 2.** Three-dimensional spherical plots of signature color ratios.

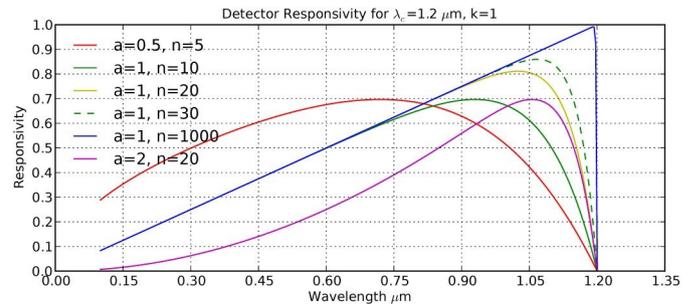


100 Frames read from data/PyradiSampleMWIR.ptw

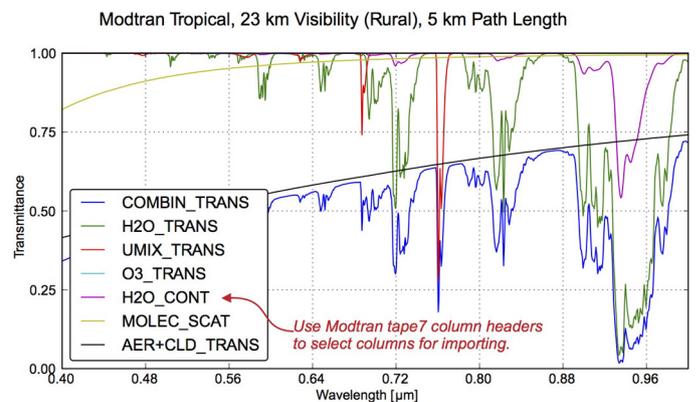
Image average S : 6269.154  
Total system noise : 49.338

Spatial noise	Temporal noise	Variation
Nh : 5.396	Nth : 0.430	Column
Nv : 45.784	Ntv : 1.994	Row
Nvh : 17.126	Ntvh : 3.381	Pixel
	Nt : 0.243	Frame

**Figure 3.** 3-D noise calculation on measured medium-wave infrared (MWIR) images. The *N* values refer to spatial and temporal noise measurements along columns (*h*) and rows (*v*) in the video sequence. A single frame of the video is shown in the figure.



**Figure 4.** Example of detector spectral shapes. The parameters '*a*' and '*n*' have no physical meaning.



**Figure 5.** Atmospheric data recorded using Modtran, read and plotted with `pyradi`. Abbreviations in the legend box represent standard Modtran `tape7` labels for atmospheric data.

ator. At the moment there is no user guide, but complete commented example code is included at the end of each module file.

The `pyradi` toolkit provides researchers and academia with a set of tools to simplify complex radiometry calculations and data visualization. We plan to grow the toolkit from its current modest beginning with the addition of new functionality in the future. Users are encouraged to join in, use, and contribute!

### Author Information

**Cornelius Willers, Johannes Calitz, Alta de Waal, and Azwita-misi Mudau**

Council for Scientific and Industrial Research (CSIR)  
Pretoria, South Africa

Cornelius Willers served as technical lead in a program to establish imaging seeker technology at Denel Dynamics until 2004. More recently, he worked on missile countermeasure

*Continued on next page*

development at the CSIR. He is codeveloper of a major physics-based infrared simulation system, and currently teaches a course in radiometry at the University of Pretoria.

**Maria Willers and Pieter van der Merwe**

Denel Dynamics  
Pretoria, South Africa

**Ricardo Tavares Santos**

Instituto Tecnológico de Aeronáutica  
São José Dos Campos, Brazil

**References**

1. C. J. Willers, M. S. Willers, R. A. T. Santos, P. J. van der Merwe, J. J. Calitz, A. de Waal, and A. E. Mudau, *Pyradi: an open-source toolkit for infrared calculation and data processing*, **Proc. SPIE 8543**, 2012. doi:10.1117/12.976043
2. <http://code.google.com/p/pyradi> Pyradi Python toolkit. Accessed 19 November 2012.