

NASA/TM—2019-220234



# Updating the Space Communications and Navigation (SCaN) Link Tool Executable Software to Version 5

*Jack L. Green, Bryan W. Welch, and Robert M. Manning  
Glenn Research Center, Cleveland, Ohio*

## NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Technical Report Server—Registered (NTRS Reg) and NASA Technical Report Server—Public (NTRS) thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers, but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., “quick-release” reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- E-mail your question to [help@sti.nasa.gov](mailto:help@sti.nasa.gov)
- Fax your question to the NASA STI Information Desk at 757-864-6500
- Telephone the NASA STI Information Desk at 757-864-9658
- Write to:  
NASA STI Program  
Mail Stop 148  
NASA Langley Research Center  
Hampton, VA 23681-2199

NASA/TM—2019-220234



# Updating the Space Communications and Navigation (SCaN) Link Tool Executable Software to Version 5

*Jack L. Green, Bryan W. Welch, and Robert M. Manning  
Glenn Research Center, Cleveland, Ohio*

National Aeronautics and  
Space Administration

Glenn Research Center  
Cleveland, Ohio 44135

---

October 2019

## Acknowledgments

This work was done in collaboration with the Strategic Center for Networking, Integration, and Communications (SCENIC) project office, specifically Bryan Welch. The mathematical models were derived and provided by Dr. Robert Manning and Bryan Welch. The work was also made possible by the Lewis' Educational and Research Collaborative Internship Project (LERCIP) at NASA Glenn Research Center.

Trade names and trademarks are used in this report for identification only. Their usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

*Level of Review:* This material has been technically reviewed by technical management.

Available from

NASA STI Program  
Mail Stop 148  
NASA Langley Research Center  
Hampton, VA 23681-2199

National Technical Information Service  
5285 Port Royal Road  
Springfield, VA 22161  
703-605-6000

This report is available in electronic form at <http://www.sti.nasa.gov/> and <http://ntrs.nasa.gov/>

# Updating the Space Communications and Navigation (SCaN) Link Tool Executable Software to Version 5

Jack L. Green\*, Bryan W. Welch, and Robert M. Manning  
National Aeronautics and Space Administration  
Glenn Research Center  
Cleveland, Ohio 44135

## Summary

NASA's Space Communications and Navigation (SCaN) program is responsible for providing space communication channels in low Earth orbit, geosynchronous orbit, and deep space for a variety of space missions. The SCaN Link Tool is a standalone, executable, and personal computer (PC)-based software operated via a user interface, which provides NASA civil servants and contractors, as well as the public by way of the tool's inclusion in the NASA Software Catalog, with indepth satellite communications link analysis capability. Version 4 of the tool was built using Python™ (Python Software Foundation) with the help of libraries such as NumPy and SciPy for numerical calculations, Matplotlib for graphical visualizations, and PyQt5 for the graphical user interface. It utilizes radiofrequency (RF) and optical communications link analysis calculations to give users the ability to input known link parameters and calculate select link performance outputs. With the development of the next-generation architecture for space satellite communications in the coming decade, NASA will benefit by having more indepth communications link analysis tools at its disposal. The tool's update from version 4 to version 5 aims to provide higher output value accuracy, the ability to solve for a more diverse set of output variables, as well as analog and digital repeater capabilities. Options included in the tool's functionality, such as the ability to save configuration parameter values, graphs, and results, as well as the provision of default parameter values, increase the tool's versatility. Users will have access to indepth, accurate communications link analysis as more advanced satellite constellations are designed and deployed by NASA and the growing commercial aerospace community.

## Introduction

NASA's Space Communications and Navigation (SCaN) program both engineers and operates communication ground stations and relay satellites to enable reliable space communications for various NASA missions and commercial applications. Effective optical and radiofrequency (RF) communication links are the drivers behind vital technologies such as the global positioning system (GPS), as well as cutting-edge NASA missions like the InSight Mars lander. Any object in space is not considered an asset unless it has the ability to both receive instructions from ground station signals and relay data back to Earth, which requires establishing communication links between ground antennas and antennas fashioned on the object in space. The performance of the link, and thus the quality and timing of the data received on Earth, is affected by an abundance of variables ranging from the modulation method and coding scheme being used to transmit and receive the signal to certain atmospheric conditions for links propagating through the atmosphere. Variables such as link margin, data rate, transmitter power, and received signal-to-noise ratio, among many others, are enlightening when characterizing a communication link and its efficacy. In addition, optical and RF communication links have vastly different properties and

---

\*Summer Intern in Lewis' Educational and Research Collaborative Internship Project (LeRCIP).

behaviors due to the difference in wavelength between the signals. Despite these differences, a mission may utilize either RF or optical communications, or a hybrid of both, causing a need for both types of analyses in certain mission architectures.

Mathematical equations that predict the performance of a communications link must be used iteratively for a variety of scenarios to sufficiently analyze a proposed mission. A tool that significantly speeds up this process would simplify analysis of any space mission utilizing RF and/or optical communication. The SCaN Link Tool is an executable software program that provides both RF and optical link analysis equations, along with a user interface, to provide easy link analysis to any user with a need for such capabilities. Version 4 of the SCaN Link Tool was implemented in Python™ (Python Software Foundation) using a host of open source libraries such as Matplotlib (Ref. 1), NumPy, SciPy, and PyQt5, and accessed via either an .exe or an .app file for use on the Microsoft Windows or macOS (Apple, Inc.) operating systems. The efforts outlined in this report describe updating the tool to version 5 by incorporating atmospheric attenuation calculations, RF analog repeater capabilities, and providing more calculation modes.

## **Background**

A brief background on RF and optical communications will be provided in this section, followed by a short discussion on the steps to convert the calculation algorithms from the MATLAB® (The Mathworks, Inc.) language to the Python™ language. Additionally, details of the Python™ NumPy array operations will be discussed, in comparison to MATLAB® operations.

### **Radiofrequency and Free-Space Optical Communications**

RF and free-space optical communications links are electromagnetic waves encoded with data and transmitted by an antenna. The propagating wave is then received by the receiver antenna upon which the data are read. Factors such as distance between antennas, frequency of the signal, and transmitter and receiver antenna aperture diameters impact the performance of a link.

RF and free-space optical communications differ in their wavelength, and thus frequency, of the propagating wave. While RF waves have a frequency of 300 kHz to 300 GHz, optical waves have a frequency of around 300 THz (Ref. 2). Therefore, optical waves have a much smaller wavelength. Optical beams have a narrower beam width when compared to RF signals, and as a result, a higher portion of the transmitted power is collected by the receiver, leading to greater signal-to-noise ratios (Ref. 2). However, the small beam divergence of an optical beam also makes beam pointing more difficult, hindering the effectiveness of free-space optical signals propagating through deep space over large distances (Ref. 2). In addition, the smaller wavelength of an optical signal makes it more susceptible to atmospheric attenuation due to specific atmospheric phenomena such as fog (Ref. 2).

### **Conversion From MATLAB® to Python™**

The equations used in the SCaN Link Tool version 5 are also used in the Strategic Center for Networking, Integration, and Communications (SCENIC) software, a SCaN web-based mission simulation and analysis software (Ref. 3). The RF and optical link analysis equations used in SCENIC were implemented in MATLAB® before their integration into the web-based software. Thus, the first step in the update of the SCaN Link Tool from version 4 to version 5 was converting the MATLAB® code, implementing the equations in Python™. A portion of the MATLAB® code included in the RF link calculations can be seen in Figure 1, with its corresponding translated Python™ code shown in Figure 2.

```

142 %Check p_ex
143 - pSet = [1, 2, 3, 5, 10, 20, 30, 50, 60, 70, 80, 90, 95, 99];
144 - if not isempty(find(pSet == p_cloud, 1))
145 -     mapstr = ['Lred_', monthStr, '_', num2str(p_cloud, '%u'), '_v5'];
146 -     map = load('Lred.mat', mapstr);
147 -     eval(['map = map.', mapstr, ';']);
148 -     L = interp2(lonGrid, latGrid, map, gs(2), gs(1));

```

Figure 1.—MATLAB® (The Mathworks, Inc.) code carrying out part of radiofrequency atmospheric attenuation calculations.

```

42 p_set = [1, 2, 3, 5, 10, 20, 30, 50, 60, 70, 80, 90, 95, 99]
43 long_grid = np.arange(0, 361, 1.125)
44 lat_grid = np.arange(-90, 91, 1.125)
45 if p_cloud in p_set:
46     monthstr = str(int(month))
47     if len(monthstr) == 1:
48         monthstr = '0' + monthstr
49     pstr = str(int(p_cloud))
50     mapstr = 'Lred_' + monthstr + '_' + pstr + '_v5.txt'
51     map_ = pd.read_csv(mapstr, sep = '\t', header = None).values
52     f = interpolate.interp2d(long_grid, lat_grid, map_)
53     L = f(longitude, -(latitude))

```

Figure 2.—Python™ (Python Software Foundation) code performing the same radiofrequency atmospheric attenuation calculations as the MATLAB® (The Mathworks, Inc.) code in Figure 1.

The focus of MATLAB® is to make matrix operations more efficient, therefore the default operations are matrix operations as seen in linear algebra. However, the standard data type provided by the NumPy library is a multidimensional array like in MATLAB®, the default operations are element-based (Ref. 4). The equations used to calculate RF and optical link performance do not use linear algebra, so NumPy’s default element-based operations are used in the SCaN Link Tool. A difference arose between the two languages in the atmospheric attenuation calculations; this was due to the difference between MATLAB® and NumPy’s interpolation functions. The implementation of both one- and two-dimensional interpolation functions in MATLAB® have the user input both the arrays of values and the query points as function parameters, which returns the interpolated values. However, NumPy only accepts the arrays of values as parameters, returning a function, which then accepts the query points as parameters. This prevailing function must then be called to calculate the interpolated values.

### NumPy Array Operations

NumPy is a powerful Python™ library often used for scientific computing and machine learning that provides mathematical operations on multidimensional arrays similar to MATLAB®. When performing operations on arrays in Python™, elements must typically be accessed one by one using a “For loop” and indexing. However, Python’s™ flexibility as a programming language means that altering an array in this way requires many function calls per iteration, prohibiting run-time effectiveness (Ref. 5). NumPy offers more efficiency when performing array operations as it utilizes C functions, offering less flexibility but with the benefit of speed, compared to the Python™ list implementation (Refs. 6). An example of a C function found in NumPy’s GitHub repository and used in the implementation of a NumPy array is seen in Figure 3 (Ref. 7). As a result of its utilization of C, NumPy can perform mathematical operations on multidimensional arrays much more efficiently, in terms of time and code syntax, than would be possible otherwise.

```

1752  static PyObject *
1753  array_iter(PyArrayObject *arr)
1754  {
1755      if (PyArray_NDIM(arr) == 0) {
1756          PyErr_SetString(PyExc_TypeError,
1757                          "iteration over a 0-d array");
1758          return NULL;
1759      }
1760      return PySeqIter_New((PyObject *)arr);
1761  }

```

Figure 3.—C function involved in the implementation of a NumPy array (Ref. 7).

NumPy’s abilities are crucial to the SCaN Link Tool version 5’s ability to quickly calculate link performance, especially when a repeater-enabled link is under consideration. These calculations contain operations done on arrays containing values for a specific variable for each sublink. In addition to the speed increase afforded by NumPy, the library offers mathematical functions such as various bases of the logarithm equation. These provided functions are used throughout the tool’s calculations.

## Methodology

A description of the graphical user interface (UI) of the SCaN Link Tool version 5 is provided, followed by descriptions of the dynamic aspects of the interface that change based on user selections. The various link calculation modes of operation are described, and several images from the tool are shown.

### Graphical User Interface

The SCaN Link Tool version 5 uses the PyQt5 framework along with its Qt Designer interface for the purpose of designing and creating the UI of the application. Qt5 is a C++ framework that simplifies the UI design and coding process. PyQt5 provides Python™ bindings that enable the power of Qt5 for Python™ applications (Ref. 8). PyQt5 UIs can be designed through Qt Designer, which uses a drag and drop widget system for easy organization of a UI window. The command line can then be used to convert the Qt Designer .ui file to a .py file, creating a Python™ UI ready to use in an application. The SCaN Link Tool version 5 is composed of three different UI components generated in Qt Designer. The most complex of the three components is the main window shown when the application is run (Figure 4). This window contains the menu options including the save and default value functions, as well as the tabs containing the parameter boxes for each type of link analysis, whether it be a nonrepeater optical link, nonrepeater analog link, or an analog repeater RF link. The second component is a noninteractive user guide that is displayed upon the click of the help button located in the top left corner of the main window. The third component is the repeater sublink widget.

PyQt5 allows one to create custom widgets not inherently included in the framework for integration into a UI (Ref. 9). This property is crucial to the operation of the repeater capabilities of the SCaN Link Tool version 5. Because a repeater analysis is an analysis of any  $n$  sublinks, where  $n$  is any positive integer, there cannot be a limit to the number of sublinks that the UI can keep track of. Therefore, a reusable widget containing the sublink parameters was created by designing a window in Qt Designer, converting the .ui file to a .py file, and altering the .py file so that the sublink widget is a child class of the QWidget class. This custom widget is presented on each page of a stacked widget, and a new instance is



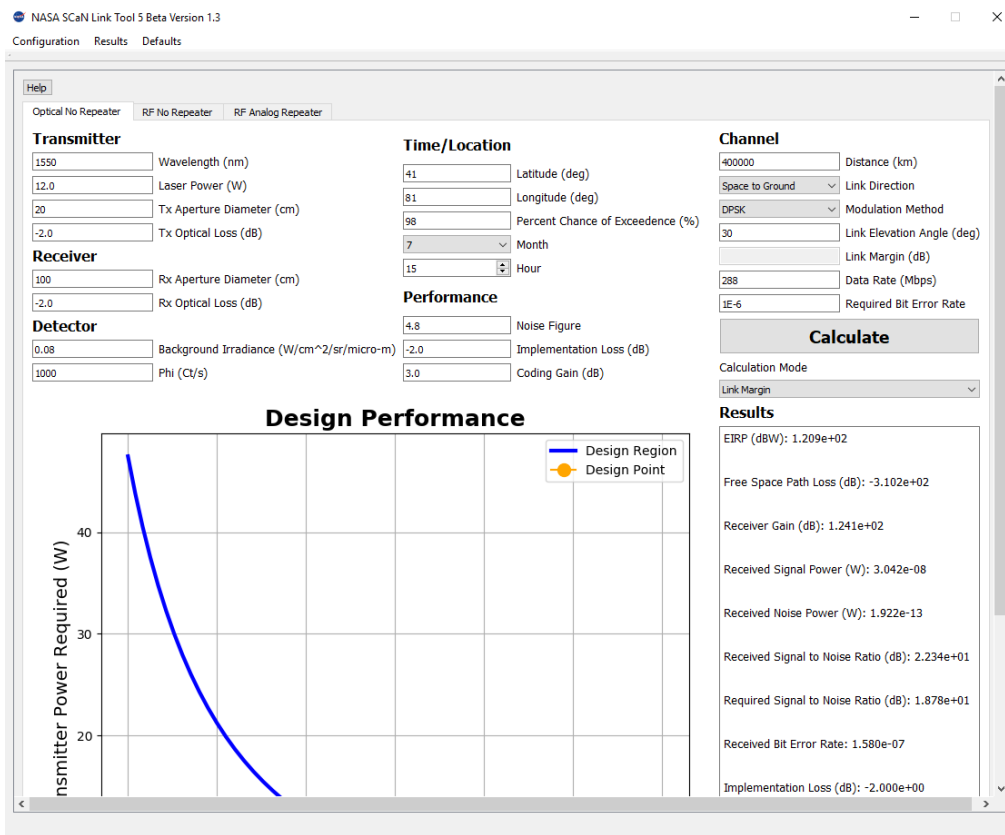


Figure 4.—Optical link calculation in link margin mode using the default optical link margin values.

created when a new sublink is added. Custom widgets can also be created by other frameworks, such as the plot widgets enabling the graphical representation included with each type of link analysis. These widgets were created with the Matplotlib framework and subsequently initialized in a child class of the QWidget class for use in the UI.

Difficulties with Qt Designer arose when trying to lay out widgets on the UI. PyQt5 provides layout widgets that can be used to organize widgets on a UI, however, the size and placement of the widgets within the UI is difficult to alter, as the layout places and sizes widgets by itself. Without a layout, the components of the UI do not get resized in accordance to a change in window size. These challenges were hindrances in the design of the UI's tabs' layouts.

## Dynamic User Interface Components

The link calculations made by the SCA-N Link Tool version 5 provide more calculation modes than version 4, which increases the complexity of the UI and its presentation of link-parameter input boxes. Many input parameters are only needed for specific link configurations. For example, the location of a ground station is only necessary if the link is not a space-to-space link.

PyQt5 provides a mechanism that allows enablement and disablement of certain UI widgets depending on the inputs entered in other widgets, which would allow the location parameters of the ground station to only show if the link is not a space-to-space link, as an example. Signals, such as the click of a specific button, can be connected to slots, which are callable Python™ functions (Ref. 10). In order to provide maximum usability, the tool only enables parameter input boxes if they are necessary for the current link configuration. Dropdown box indices are connected to functions that enable and disable input boxes depending on the currently selected index of the dropdown menu.

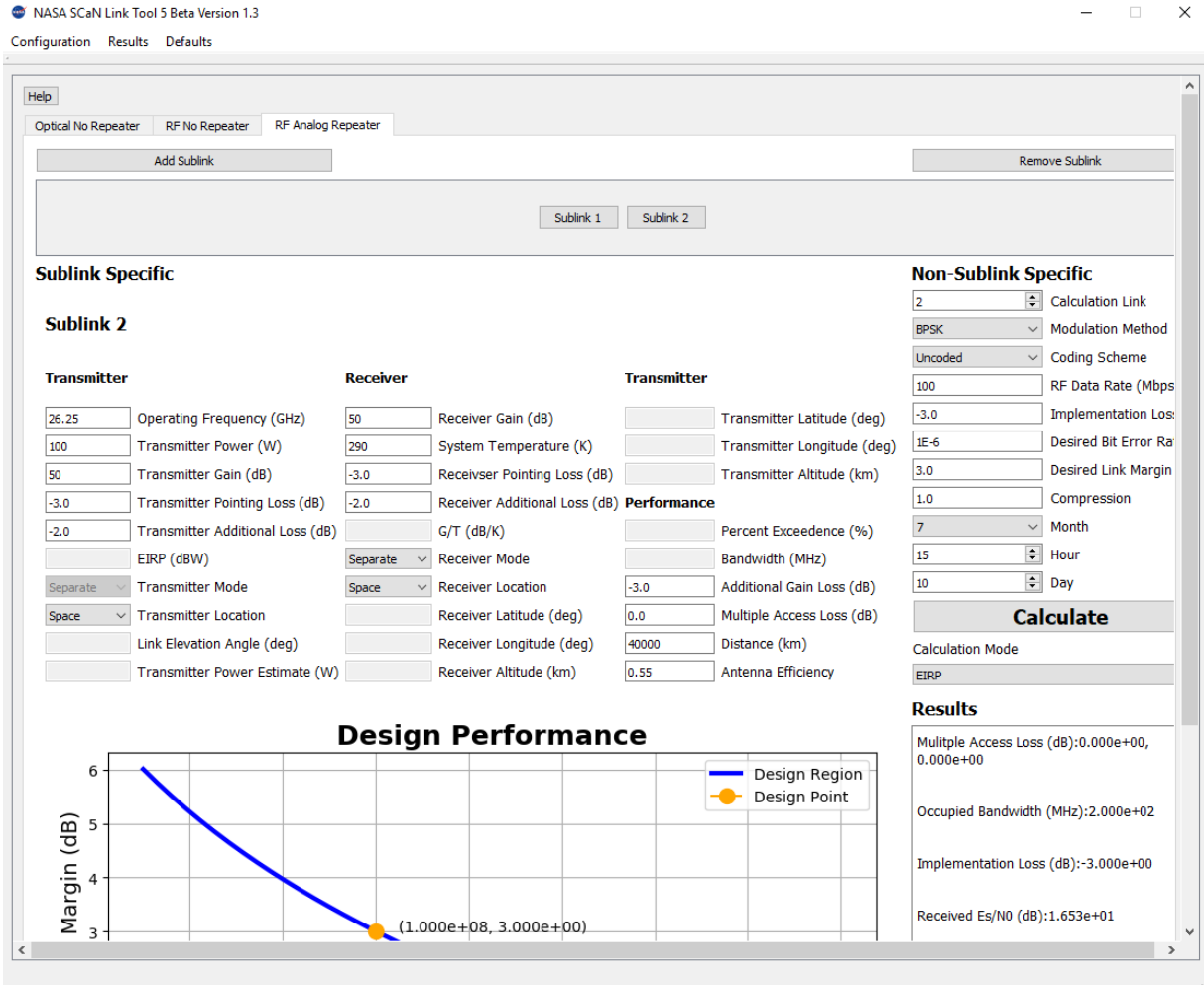


Figure 5.—The radiofrequency (RF) analog repeater calculations using the equivalent isotropically radiated power (EIRP) RF analog repeater default values.

The repeater link calculations provided by the SCan Link Tool version 5 make even greater use of dynamic changes to the UI than the nonrepeater capabilities by way of signals and slots. When a sublink is added, a new instance of the custom sublink widget is initialized and placed on a new layer of the stacked widget. A new sublink button is added as well, providing access to this new page of sublink parameter boxes. The function called upon the click of the add sublink button sets connections between signals and slots that ensure the receiver of one sublink has the same location as the transmitter of the next sublink and prohibits ground-to-ground links. As a result, when sublink widgets are removed, the slots connected to that sublink's click signal must also be removed so that the other sublink connected to those slots do not try to access the removed sublink's parameter boxes. Arrays of the sublink specific input parameter boxes are maintained so that each sublink's parameter values can be accessed in the calculations. Figure 5 shows the tool and its display of the second sublink's parameter widget.

## Link Calculations

The optical link calculation capabilities includes six calculation modes, each one omitting a certain parameter input, allowing it to be calculated using the other parameter values as inputs to the equations. The modes include link margin, data rate, distance, transmitter power, transmitter gain, and receiver gain.

The RF calculations also include equivalent isotropically radiated power (EIRP) and gain-to-temperature (G/T) in addition to the optical calculation modes for both analog repeater and nonrepeater analyses. In the case of the RF analog repeater calculations, a specific sublink is chosen as the calculation link for which the calculation mode output parameter is solved. Details such as ground station location for ground-to-space or space-to-ground links, modulation method, and coding scheme increase the fidelity of the calculations.

A portion of the calculations introduced in version 5 are the atmospheric attenuation calculations. Each ground-to-space or space-to-ground link analyzed considers the location of the ground station, elevation of the propagating signal, and the percent chance of exceedance to calculate atmospheric loss, increasing the accuracy of the prevailing outputs (Ref. 11). In addition to the calculation mode output value, other outputs include free-space path loss, received signal-to-noise ratio, and received bit error rate.

A host of data stored in .txt files is relied on for certain calculations. Required bit error rate and required signal-to-noise ratio are both calculated using data included in these .txt files. The latitudes and longitudes, as well as the atmospheric visibility data associated with each set of latitude and longitude, are included in the data and used in calculating the atmospheric attenuation of an optical link that travels through the atmosphere. The RF atmospheric attenuation calculations utilize Lred values included in the .txt files specific to the month and percent chance of exceedance included in the input parameters.

The graphical output along with the output parameter values for an optical link plots the required link transmitter power on the y-axis and the product of the transmitter and receiver aperture diameters on the x-axis. The required transmitter power is established by recalculating the link under the condition of a transmitter power calculation mode while using the previously calculated outputs as inputs for the recalculation. An example of a graph produced by an optical link calculation is shown in Figure 6. The design region is the curve on which the link would perform identically to its performance produced by the

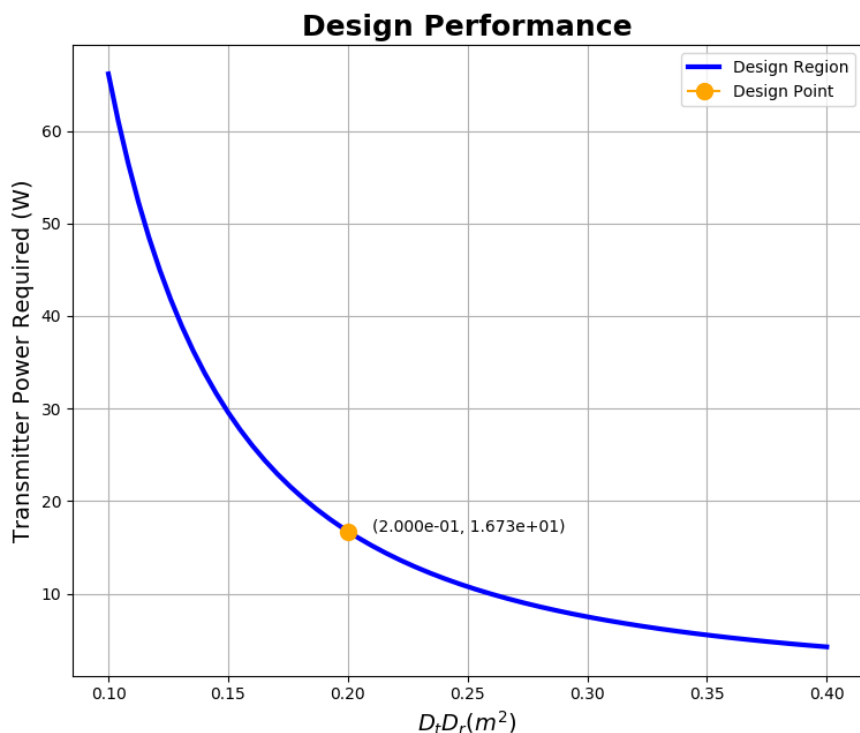


Figure 6.—Space Communications and Navigation (SCaN) Link Tool version 5 sample design performance plot showing data resulting from calculation of the default values for the transmitter power radiofrequency mode.

actual link parameters, which is illustrated by the design point plotted on the curve. A similar process is undertaken for RF link calculations, with the difference being that the recalculation uses the calculation process for the link margin calculation mode so that required link margin values can be plotted on the y-axis with data rate on the x-axis.

## Results

Throughout the update process, certain milestones prompted the creation of an executable beta version of the software for the purpose of testing. The first beta release took place after the completion of the optical and RF nonrepeater capabilities, both of which are present in version 4 of the tool but altered here. This release prompted feedback about issues such as font size and graphical visualizations. The most important update made at this stage of testing was the introduction of labeled points on the graphical plot. It took six iterations of this stage of the beta to correct bugs found when testing.

A second major iteration of the beta was released upon completion of the RF analog repeater capability. This capability included mechanisms new to the tool, as the elements of the UI necessary to organize the sublink data were not present in version 4. The sublink addition and removal process will need to be tested more rigorously as the calculations are more complicated given the lack of a limit to the number of sublinks that can be analyzed.

The output results produced by the tool in its current state match those calculated by the MATLAB<sup>®</sup> scripts given identical input parameters. Some outputs are slightly off and accurate to about three decimal places, most likely due to differences in the precision of the dot product functions of NumPy and MATLAB<sup>®</sup>.

## Conclusions

The Space Communications and Navigation (SCaN) Link Tool's purpose—to provide indepth radiofrequency (RF) and optical communication link calculations with a user-friendly user interface (UI)—is well serviced by Python<sup>™</sup> language (Python Software Foundation) and its available libraries and frameworks. Highly accurate mathematical operations and easy to use UI tools are made possible by NumPy, SciPy, and PyQt5. Output results have been shown to be accurate upon comparison with the outputs produced by the MATLAB<sup>®</sup> (The Mathworks, Inc.) code used in the Strategic Center for Networking, Integration, and Communications (SCENIC) software.

Usability will continue to be analyzed through testing by select beta testers, especially as the tool is further developed with the addition of RF, optical, and hybrid RF/optical digital repeater capabilities. Once these capabilities have been added and the tool has gone through the process of verification and validation, it will be submitted for inclusion in NASA's public facing software catalog.

## References

1. Hunter, John D.: Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.*, vol. 9, no. 3, 2007, pp. 90–95.
2. Sburlan, Suzana: Introduction to Optical Communications for Satellites. Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 2016.  
<http://kiss.caltech.edu/workshops/optcomm/presentations/Sburlan.pdf> Accessed Sept. 17, 2019.
3. SCENIC, NASA, <https://scenic.ndc.nasa.gov/>
4. NumPy for Matlab Users. SciPy, Enthought, Inc., Austin, TX, 2019.  
<https://docs.scipy.org/doc/numpy/user/numpy-for-matlab-users.html> Accessed Sept. 17, 2019.

5. 11 PEP 357: The ‘`__index__`’ method. Python Software Foundation, Wilmington, DE, 2006. <https://docs.python.org/2.5/whatsnew/pep-357.html> Accessed Sept. 17, 2019.
6. Frequently Asked Questions. SciPy, Enthought, Inc., Austin, TX, 2019. <https://scipy.org/scipylib/faq.html#what-advantages-do-numpy-arrays-offer-over-nested-python-lists> Accessed Sept. 17, 2019.
7. NumPy. Python Software Foundation, Wilmington, DE, 2005. <https://github.com/numpy/numpy/blob/master/numpy/core/src/multiarray/arrayobject.c> Accessed Sept. 17, 2019.
8. PyQt5. PyPI, River Bank Computing Limited, Dorchester, England, 2019. <https://pypi.org/project/PyQt5/> Accessed Sept. 17, 2019.
9. Boddie, David: Designing Custom Controls With PyQt. Qt Quarterly, Nokia Corporation, Espoo, Finland, 2008. <https://doc.qt.io/archives/qq/qq26-pyqt designer.html> Accessed Sept. 17, 2019.
10. Support for Signals and Slots. Riverbank Computing Limited, Dorchester, England, 2019. [https://www.riverbankcomputing.com/static/Docs/PyQt5/signals\\_slots.html](https://www.riverbankcomputing.com/static/Docs/PyQt5/signals_slots.html) Accessed Sept. 23, 2019.
11. Green, Jack L.; Welch, Bryan W.; and Manning, Robert M.: Optical Communication Link Atmospheric Attenuation Model. NASA/TM—2019-219991, 2019. <http://ntrs.nasa.gov>





