Link Budget

Release 2020

Igor Freire

Mar 22, 2023

CONTENTS:

A Python-based link budget calculator for satellite communications and radar systems.

CHAPTER

ONE

INSTALLATION

Package *link-budget* provides the link budget calculator as a command-line tool. To install it, run:

```
pip install link-budget
```

CHAPTER

TWO

BASIC USAGE

The *link-budget* utility accepts a range of parameters. The following example computes the link budget for a 52 dBW EIRP at 12.45 GHz coming from a GEO satellite at a longitude of -101° (or 101 West), with a nominal bandwidth of 24 MHz. The Rx dish has a diameter of 0.46 m, while the LNB has a conversion gain of 40 dB and a noise figure of 0.6 dB. Furthermore, the LNB connects to a receiver with a noise figure of 10 dB over a coaxial cable with 110 ft. Lastly, this example receiver station is at longitude -82.43 degrees and latitude 29.71 degrees.

```
link-budget \
   --sat-long -101 \
   --eirp 52 \
   --freq 12.45e9 \
   --bw 24e6 \
   --rx-dish-size 0.46 \
   --lnb-noise-fig 0.6 \
   --lnb-gain 40 \
   --coax-length 110 \
   --rx-noise-fig 10 \
   --rx-long -82.43 \
   --rx-lat 29.71
```

2.1 Command-line Utility

Link Budget Calculator

```
usage: link-budget [-h] [--json] [-v] [--min-cnr MIN_CNR]
                   [--impl-margin IMPL_MARGIN] --freq FREQ --bw BW
                   [--polarization {circular,linear}]
                   (--eirp EIRP | --tx-power TX_POWER) [--obo OBO]
                   [--carrier-peb CARRIER_PEB] [--tp-bw TP_BW]
                   [--tx-dish-size TX_DISH_SIZE | --tx-dish-gain TX_DISH_GAIN]
                   [--tx-dish-efficiency TX_DISH_EFFICIENCY]
                   (--rx-dish-size RX_DISH_SIZE | --rx-dish-gain RX_DISH_GAIN)
                   [--rx-dish-efficiency RX_DISH_EFFICIENCY]
                   [--atmospheric-loss ATMOSPHERIC_LOSS]
                   [--availability AVAILABILITY] [--asi]
                   [--asi-eirp-ratio ASI_EIRP_RATIO]
                   [--asi-long-separation ASI_LONG_SEPARATION]
                   [--antenna-noise-temp ANTENNA_NOISE_TEMP]
```

(continues on next page)

(continued from previous page)

```
                  [--mispointing-loss MISPOINTING_LOSS]
                  (--lnb-noise-fig LNB_NOISE_FIG | --lnb-noise-temp LNB_NOISE_TEMP)
                  --lnb-gain LNB_GAIN --rx-noise-fig RX_NOISE_FIG
                  --coax-length COAX_LENGTH [--lna-feed-loss LNA_FEED_LOSS]
                  [--rx-long RX_LONG] [--rx-lat RX_LAT]
                  [--rx-height RX_HEIGHT] [--sat-long SAT_LONG]
                  [--sat-lat SAT_LAT] [--sat-alt SAT_ALT]
                  [--slant-range SLANT_RANGE] [--ref-ellipsoid {WGS84,GRS80}]
                  [--obs-time OBS_TIME] [--sat-tle-name SAT_TLE_NAME]
                  [--sat-tle-group {active,stations,geo,intelsat,ses,iridium,iridium-
↪NEXT,starlink,oneweb,orbcomm,globalstar,swarm,amateur,x-comm,other-comm,satnogs,
↪gorizont,raduga,molniya,weather,noaa,goes,resource,sarsat,dmc,tdrss,argos,planet,spire,
↪gnss,gps-ops,glo-ops,galileo,beidou,sbas,nnss,musson,military,radar,cubesat,other}]
                  [--tle-save-dir TLE_SAVE_DIR] [--tle-no-save] [--radar]
                  [--radar-cross-section RADAR_CROSS_SECTION]
```

2.1.1 General Options

--json Print results in JSON format.

Default: False

-v, --version show program's version number and exit

2.1.2 Margin Options

--min-cnr Target minimum carrier-to-noise ratio (CNR) in dB for the link margin computation.

--impl-margin Implementation margin in dB accounting for the non-ideal behavior of the receiver system.

Default: 0

2.1.3 Frequency Options

--freq Downlink carrier frequency in Hz for satellite signals or simply the signal frequency in Hz for radar (passively reflected) signals.

--bw, --if-bw Nominal signal bandwidth in Hz.

2.1.4 Polarization Options

--polarization Possible choices: circular, linear

Polarization of the transmitted electromagnetic wave.

Default: "linear"

2.1.5 Tx Power Options

--eirp Carrier or transponder EIRP in dBW. If an output backoff is defined, this EIRP corresponds to the amplifier's saturated operation. Otherwise, it represents the actual operating EIRP.

--tx-power Power feeding the Tx antenna in dBW. If an output backoff is defined, this parameter represents the amplifier's saturated output power. Otherwise, it refers to the actual Tx power.

--obo Carrier or transponder output backoff in dB.

Default: 0

2.1.6 FDMA Carrier Power Options

Parameters to determine the power allocated to an FDMA carrier.

--carrier-peb Power-equivalent bandwidth (PEB) in Hz assigned for the FDMA carrier. When provided, the EIRP computed from –eirp or –tx-power refers to the transponder, while the PEB determines the fraction of this transponder EIRP allocated to the carrier. In this case, note the output backoff must refer to the transponder, not the carrier. If the output backoff represents the carrier backoff, do not inform the carrier PEB.

--tp-bw Transponder bandwidth in Hz, required if the PEB is provided.

2.1.7 Antenna Options

--tx-dish-size Diameter in meters of the parabolic antenna used for transmission. Used when the power is specified through option –tx-power.

--tx-dish-gain Gain in dBi of the parabolic antenna used for transmission. Used when the power is specified through option –tx-power.

--tx-dish-efficiency Aperture efficiency of the parabolic antenna used for transmission. Determines the antenna gain when the dish is specified by size (option `--tx-dish-size`). Otherwise, when the gain is defined directly by option `--tx-dish-gain`, this parameter is used to infer the diameter of an equivalent parabolic reflector.

Default: 0.56

--rx-dish-size Diameter in meters of the parabolic antenna used for reception.

--rx-dish-gain Gain in dBi of the parabolic antenna used for reception.

--rx-dish-efficiency Aperture efficiency of the parabolic antenna used for reception. Determines the antenna gain when the dish is specified by size (option `--rx-dish-size`). Otherwise, when the gain is defined directly by option `--rx-dish-gain`, this parameter is used to infer the diameter of an equivalent parabolic reflector.

Default: 0.56

2.1.8 Propagation Options

--atmospheric-loss Attenuation in dB experienced through the atmosphere. It should always include the clear air attenuation, and it could include other effects such as rain and cloud attenuation. When analyzing a radar system, note this option should determine the one-way attenuation, not the two-way. When omitted, the program assumes a reasonable atmospheric attenuation based on models from ITU-R recommendations.

--availability Target link availability in % to consider on the atmospheric attenuation model. For instance, when targeting at a 99.9% availability, the analysis is based on the atmospheric attenuation exceeded 0.1% of the time.

Default: 99

2.1.9 Interference Options

--asi Include adjacent satellite interference (ASI) in the link budget considering neighbor satellites with overlapping coverage, frequency and polarization.

Default: False

--asi-eirp-ratio Ratio between the aggregate downlink EIRP from adjacent satellites and the wanted signal's EIRP.

Default: 1.0

--asi-long-separation Longitudinal orbit separation in degrees between the wanted satellite and the adjacent satellite(s).

Default: 2.0

2.1.10 Rx Options

--antenna-noise-temp Receive antenna's noise temperature in K. When omitted, this parameter is derived based on the atmospheric attenuation.

--mispointing-loss Loss in dB due to antenna mispointing.

Default: 0

--lnb-noise-fig LNB's noise figure in dB.

--lnb-noise-temp LNB's noise temperature in K.

--lnb-gain LNB's gain.

--rx-noise-fig Receiver's noise figure in dB.

--coax-length Length of the coaxial transmission line between the LNB and the receiver in ft.

--lna-feed-loss Loss in dB of the passive feed (typically waveguide) connecting the antenna to the LNA. Applicable when a discrete LNB is used instead of an integrated LNBF.

Default: 0

2.1.11 Earth Station Position Information

Note longitudes are positive east of the prime meridian and negative otherwise (west), while latitudes are positive above the equator (north) and negative otherwise (south).

--rx-long Rx station's longitude in degrees.

--rx-lat Rx station's latitude in degrees.

--rx-height Rx station's height in meters above mean sea level.

Default: 0

2.1.12 Satellite or Radar Object Position Information

Note longitudes are positive east of the prime meridian and negative otherwise (west), while latitudes are positive above the equator (north) and negative otherwise (south).

--sat-long Longitude in degrees of the satellite or radar object.

--sat-lat Latitude in degrees of the satellite or radar object.

Default: 0

--sat-alt Satellite or radar object's altitude in meters above the reference ellipsoid.

Default: 35786000.0

--slant-range Slant path length in km between the Rx station and the satellite or reflector.

--ref-ellipsoid Possible choices: WGS84, GRS80

Reference Earth ellipsoid to use in the computation.

Default: "WGS84"

--obs-time Observation time for the satellite position's prediction. Requires the '–sat-tle-name' option and must be given in ISO 8601 format.

2.1.13 Satellite TLE Information

--sat-tle-name Satellite name on CelesTrak's TLE database.

--sat-tle-group Possible choices: active, stations, geo, intelsat, ses, iridium, iridium-NEXT, starlink, oneweb, orbcomm, globalstar, swarm, amateur, x-comm, other-comm, satnogs, gorizont, raduga, molniya, weather, noaa, goes, resource, sarsat, dmc, tdrss, argos, planet, spire, gnss, gps-ops, glo-ops, galileo, beidou, sbas, nnss, musson, military, radar, cubesat, other

CelesTrak satellite group to restrict the search. When undefined, searches the entire database.

--tle-save-dir Directory where downloaded TLE datasets should be saved.

Default: "/home/docs/.link-budget/tle"

--tle-no-save Do not save the downloaded TLE datasets locally. A new download will be required every time.

Default: False

2.1.14 Radar Options

--radar Activate monostatic radar mode so that the link budget considers the path loss to and back from object.

Default: False

--radar-cross-section Radar cross-section of the radar object.

2.2 Calculation Library

A collection of link budget and other RF calculations.

References:

- [1] L. W. Couch, "Digital & Analog Communication Systems," 8th Ed., 2013.
- [2] M. Lindgren, "A 1296 MHz Earth–Moon–Earth Communication System," 2015.
- [3] T. Pratt and J. E. Allnutt, "Satellite Communications," 3rd Ed., 2019.
- [4] D. M. Pozar, "Microwave Engineering," 4th Ed., 2012.

linkbudget.calc.**antenna_noise_temp**(*attn_db*, *T_medium=270*, *coupling_eff=1.0*)

Compute the antenna noise temperature

Follow the theory in Section 4.5.2 of [3].

Parameters

- **attn_db** – Total path attenuation (in dB) experienced through the atmosphere, including clear air and rain attenuation.
- **T_medium** – Medium temperature (in K) assumed for the rain.
- **coupling_eff** – Sky noise coupling coefficient determining the fraction of incident sky noise energy output by the antenna.

Note: In [3], a medium temperature of 270 K is considered when computing the sky noise temperature for rain. In contrast, a temperature of 290 K is considered when analyzing clear sky. The rationale is to be confirmed.

---

linkbudget.calc.**capacity**(*snr_db*, *bw*)

Compute the channel capacity in bps

Parameters

- **snr_db** – signal-to-noise ratio in dB.
- **bw** – nominal bandwidth.

Returns

Capacity in bits per second (bps).

linkbudget.calc.**carrier_eirp**(*sat_eirp*, *obo*, *peb=None*, *tp_bw=None*)

Compute the carrier EIRP using transponder/amplifier information

Converts the saturated EIRP obtained by an amplifier or transponder into the corresponding carrier EIRP, after output backoff and power allocation.

There are two main use cases for this function:

1. To compute the carrier EIRP from the transponder's saturated EIRP and carrier output backoff (OBO).
2. To compute the carrier EIRP based on the transponder's saturated EIRP, the transponder's OBO, the carrier's power-equivalent bandwidth (PEB), and the transponder's bandwidth.

In scenario 1, we start with the transponder's EIRP in dBW obtained when its amplifier operates in saturation. This metric is often given as the transponder's downlink EIRP at beam peak (BP) or beam center (BC). Then, assuming the transponder is shared by multiple FDMA carriers, we subtract the carrier OBO to obtain the EIRP assigned to the carrier of interest. In this case, note the **carrier OBO** is often a much larger value than the **transponder OBO**. The carrier OBO is easily in excess of 10 dB, depending on how much of the transponder it occupies. In contrast, as discussed in [3], the transponder OBO is typically within the range from 1 to 7 dB in FDMA systems.

The OBO interpretation changes in scenario 2 when considering FDMA systems. In this case, the OBO is interpreted as the **transponder OBO**, not the **carrier OBO**. Hence, the first step in the computation is to obtain the transponder's operating EIRP, which is given by the saturated EIRP minus the transponder OBO. The second step is to allocate a fraction of the available transponder EIRP to the carrier of interest. This fraction is determined by the ratio between the PEB and the transponder bandwidth.

To compute scenario 1, call this function with parameter *obo* as the carrier OBO, while leaving the PEB undefined. To compute scenario 2, call this function with *obo* as the transponder OBO, while informing the PEB and transponder bandwidth. In any case, however, the *sat_eirp* parameter represents the transponder's saturated EIRP in the direction of interest.

Note that, for convenience, the saturated EIRP is **in the direction of interest**, not necessarily at BP or BC. For example, it could be at beam edge (BE) or any arbitrary coverage contour. Furthermore, not this function takes the saturated EIRP (i.e., including the on-axis antenna gain), instead of the amplifier's saturated output power. This is intentional because it is often easier to find the EIRP information for a given satellite contour than to find the transponder and antenna gain separately.

Lastly, note that both use cases are equivalent in TDMA systems. Since there is a single carrier in TDMA, the transponder OBO is the same as the carrier OBO. Furthermore, the PEB is the full transponder bandwidth in TDMA, so the *peb* parameter does not need to be informed.

Parameters

- **sat_eirp** – Saturated amplifier/transponder EIRP in dBW in the direction if interest.
- **obo** – Output backoff in dB.
- **peb** – Power equivalent bandwidth if the carrier shares the amplifier/transponder with other FDMA carriers.
- **tp_bw** – Transponder bandwidth used when the PEB is given.

Returns

Carrier EIRP in dBW.

linkbudget.calc.**carrier_to_asi_ratio**(*rx_dish*, *long_separation*, *asi_eirp_ratio*)

Compute the carrier to adjacent satellite interference (ASI) ratio

Parameters

- **rx_dish** (Antenna) – Rx antenna object.
- **long_separation** (*float*) – Longitudinal separation in degrees between the wanted satellite and the adjacent interferer(s).
- **asi_eirp_ratio** (*float*) – Ratio between the aggregate adjacent downlink EIRP and the wanted signal's EIRP.

Returns

Carrier-to-ASI ratio in dB.

Return type
(float)

linkbudget.calc.**cascaded_input_noise_temp**(*temps*, *gains*)

Equivalent input noise temperature of cascaded linear devices

Computes the overall equivalent input noise temperature for cascaded linear devices according to Equation 8-37 from [1].

Parameters

- **temps** – List with the individual input noise temperatures (in K) corresponding to each of the devices in order from input to output.
- **gains** – List with the gains (in dB) of the cascaded linear devices, excluding the last, in the same order as given for the input noise temperatures.

Note: The list of gains should not include the gain of the last device in the chain because this gain is irrelevant for the computation.

Returns
The overall input noise temperature in K.

linkbudget.calc.**cascaded_noise_figure**(*nfs*, *gains*)

Compute the overall noise figure of cascaded linear devices

Based on Equation 8-34 from [1].

Parameters

- **nfs** – List with the noise figures (in dB) corresponding to each of the devices in order from input to output.
- **gains** – List with the gains (also in dB) of the cascaded linear devices, excluding the last, in the same order as given for the noise figures.

Note: The list of gains should not include the gain of the last device in the chain because this gain is irrelevant for the computation.

Returns
The overall noise figure in dB.

linkbudget.calc.**cnir**(*cnr*, *ci*)

Compute the carrier to noise plus interference ratio

Based on the reciprocal CNR formula in Eq. (4.42) from [3].

Parameters

- **cnr** (*float*) – Carrier-to-noise ratio in dB.
- **ci** (*float*) – Carrier-to-interference ratio in dB.

Returns
Carrier to noise plus interference ratio C/(N+I) in dB.

Return type
(float)

linkbudget.calc.**cnr**(*P_rx_dbw*, *N_dbw*)

Compute the carrier-to-noise ratio (CNR) in dB

Parameters

- **P_rx_dbw** – Received (carrier) power in dBW
- **N_dbw** – Receiver noise power in dBW

Returns
CNR in dB.

linkbudget.calc.**coax_loss_nf**(*length_ft*, *Tl=290*)

Compute the loss and noise figure of a coaxial RG6 transmission line

Parameters

- **length_ft** – Line length in feet.
- **Tl** – temperature of the line in Kelvin.

Returns
Tuple with line loss (dB) and noise figure (dB).

linkbudget.calc.**eirp**(*tx_power*, *tx_dish_gain*)

Compute the effective isotropically radiated power (EIRP)

EIRP (dB) = Tx Power (dB) + Tx Antenna Gain (dB).

Parameters

- **tx_power** – Transmit power feeding the antenna (dBW).
- **tx_dish_gain** – Transmit antenna gain (dB).

Returns
The EIRP in dBW.

linkbudget.calc.**g_over_t**(*rx_ant_gain_db*, *T_sys_db*)

Compute the G/T ratio in dB/K

Compute the ratio between the Rx antenna gain and the receiver system noise temperature, known as G/T. This ratio determines the quality of the satellite receiving system. The CNR is proportional to it, and the G/T ratio represents the part of the CNR that can be improved based on the receiver hardware alone. The other terms of the CNR are EIRP, slant range, noise bandwidth, and downlink frequency, which are usually fixed for a specific location and service.

Parameters

- **rx_ant_gain_db** – Receiver antenna gain in dB.
- **T_sys_db** – Receiver system noise temperature in dBK.

Returns
G/T in decibels with units of dBK^1 (or dB/K).

linkbudget.calc.**noise_fig_to_noise_temp**(*nf*)

Convert noise figure to the effective input-noise temperature

Note the noise figure is always referenced to a noise source at the standard noise temperature of T0 = 290 K. In contrast, the noise temperature is independent of the temperature of the noise source.

Parameters
nf – Noise figure in dB.

Returns
Noise temperature in K.

linkbudget.calc.**noise_power**(*T_sys_db*, *bw*)

Compute the receiver noise power in dBW

According to Equation 8-40 in [1], the noise power is given by *N = k*Tsyst*bw*, where k is the Boltzmann constant, Tsyst is the receiver system noise temperature (in absolute units) and bw is the IF equivalent bandwidth in Hz.

Parameters

- **T_sys_db** – Receiver system noise temperature in dBK.
- **bw** – Nominal signal bandwidth (also known as noise bandwidth).

Returns
Receiver noise power in dBW.

linkbudget.calc.**noise_temp_to_noise_fig**(*Te*)

Convert an effective input-noise temperature to a noise figure in dB

Parameters
Te – Noise temperature in K.

Returns
Noise figure in dB.

linkbudget.calc.**passive_attn_in_noise_temp**(*attn_db*, *T*)

Input noise temperature of a passive two-port attenuator

Compute the equivalent input noise temperature of a passive two-port lossy component, such as a transmission line or waveguide.

When considering the equivalent **input** noise temperature, the model is as follows:

```
Signal --> Sum --> Noiseless Passive Component --> Signal + Noise Out
            ^
            |
            |
      Noise Source (Resistor at Temperature Te)
```

The noise source is at the input of the passive two-port element, and the latter is assumed noiseless but lossy with gain G or attenuation L. The noise produced by the hypothetical noise source, when amplified by G (or attenuated by L), yields the same noise power on the component's output as the noisy (but otherwise equivalent) component produces by itself.

The output noise power produced by the noisy passive two-port component is modeled as equal to GkT_eB, namely the input noise power kT_eB amplified by gain G. Finally, T_e is the physical temperature of a hypothetical resistor used to represent the noise source. The resistor produces noise power kT_eB over bandwidth B, and T_e is called the equivalent input noise temperature, given by:

$$T_e = (L - 1)T,$$

where T is the physical temperature of the two-port component, and L is the component's attenuation (or loss) in absolute units. See Equation (8-31a) in [1] or Equation (10.15) in [4].

Parameters

- **attn_db** (*float*) – Attenuation in dB.
- **T** (*float*) – Physical temperature of the attenuator.

Returns
The equivalent input noise temperature of the passive two-port attenuator.

Return type
float

linkbudget.calc.**passive_attn_noise_fig**(*attn_db*, *T*)

Noise figure of a passive two-port attenuator

Compute the noise figure of a passive two-port lossy component, such as a transmission line or waveguide.

The equivalent input noise temperature of such a component is equal to:

$$T_e = (L-1)T,$$

where T is the physical temperature of the component, and *L* represents its attenuation (or loss) in absolute units.

Hence, the noise factor referenced to the standard temperature T0=290 K is equal to:

$$F = 1 + \frac{T_e}{T0} = 1 + \frac{T}{T0}(L-1).$$

See Equation 8.32a in [1] or Equation 10.16 in [4].

Note: The noise factor approaches unit if the component's physical temperature T is close to 290 K. This is typically the case in environments inhabitable by humans. In this scenario, the noise figure (in dB) is approximately equal to the device's loss in dB. For instance, a waveguide or transmission line with 2 dB loss would have a noise figure close to 2 dB.

Parameters

- **attn_db** (*float*) – Attenuation in dB.
- **T** (*float*) – Physical temperature of the attenuator.

Returns
The noise figure of the passive two-port attenuator.

Return type
float

linkbudget.calc.**passive_attn_out_noise_temp**(*attn_db*, *T*)

Output noise temperature of a passive two-port attenuator

Compute the equivalent output noise temperature of a passive two-port lossy component, such as a transmission line or waveguide.

When considering the equivalent **output** noise temperature, the model is as follows:

```
Signal --> Noiseless Passive Component --> Sum --> Signal + Noise Out
                                           ^
                                           |
                                           |
                     Noise Source (Resistor at Temperature Teo)
```

The noise source is at the output of the passive two-port element, and the latter is assumed noiseless but lossy with gain G or attenuation L. The hypothetical noise source produces the same noise power as the noisy (but otherwise equivalent) two-port component produces by itself.

The output noise power produced by the noisy passive two-port component is modeled as equal to $kT_{eo}B$, and T_{eo} is called the equivalent output noise temperature. Finally, T_{eo} is interpreted as the physical temperature of a hypothetical resistor representing the noise source, and is given by:

$$T_{eo} = (1 - G)T,$$

where T is the physical temperature of the two-port component, and *G* is the component's gain in absolute units (less than unit because the component is lossy). See Equation (4.21) in [3].

Note: The equivalent output noise temperature T_{eo} is equal to the equivalent input noise temperature T_e multiplied by the gain G of the lossy two-port component or network.

Parameters

- **attn_db** (*float*) – Attenuation in dB.
- **T** (*float*) – Physical temperature of the attenuator.

Returns
The equivalent output noise temperature of the passive two-port attenuator.

Return type
float

linkbudget.calc.**path_loss**(*d*, *freq*, *radar=False*, *obj_gain=None*, *bistatic=False*, *d_rx=None*)

Calculate the free-space path loss (or transmission loss)

This function supports radar mode, in which case it computes the transmission loss considering the path loss in the forward and reverse paths (to and from the radar object) as well as the scattering at the radar object based on its radar cross section.

Parameters

- **d** – Distance in meters between transmitter and receiver (e.g., satellite to ground station) or between the transmitter and the radar object.
- **freq** – Carrier frequency in Hz.
- **radar** – Radar mode.
- **obj_gain** – Gain of the radar object in dB.
- **bistatic** – Bistatic radar mode.
- **d_rx** – Bistatic radar mode only: distance between the radar object and the Rx that is not collocated with the Tx.

Returns
Path loss in dB.

linkbudget.calc.**radar_obj_gain**(*freq*, *rcs*)

Compute the gain of the radar object

Parameters

- **freq** – Carrier frequency in Hz.
- **rcs** – Radar cross section (RCS).

Returns
Gain of the radar object in dB.

Note: The RCS definition repeated in [2] is the following: "the RCS of a radar object is the hypothetical area intercepting that amount of power which, when scattered isotropically, produces a power density at the receiver equal to that from the actual object."

linkbudget.calc.**rx_flux_density**(*eirp_db*, *distance*, *atm_loss_db=0*)

Compute the received flux density in in dBW/m^2

Note the Rx flux density refers the incident power density (power per area) arriving at the receiving antenna, i.e., at the antenna's input. In contrast, the value computed by function "rx_power" refers to the power collected by the antenna, i.e., at the antenna's output.

Parameters

- **eirp_db** – EIRP in dBW.
- **distance** – Distance between the Tx and Rx in meters.
- **atm_loss_db** – Total atmospheric loss in dB.

Returns

Received flux density in dBW/m^2.

Note: The flux density unit of dBW$/m^2$ should be interpreted as $10\log 10(\frac{W}{m^2})$, and not $\frac{10\log 10(W)}{m^2}$. With a flux density F in dBW$/m^2$ and a given area A in m^2, the Rx power can be obtained by $F + 10\log 10(A)$. Equivalently, if the area is given as A_{db} in dBm^2 (decibels greater than 1 square meter), the Rx power can be obtained $F + A_{db}$.

linkbudget.calc.**rx_power**(*eirp_db*, *path_loss_db*, *rx_ant_gain_db*, *atm_loss_db=0*, *mispointing_db=0*, *feed_loss_db=0*)

Compute the received carrier power in dBW

Parameters

- **eirp_db** – EIRP in dBW.
- **path_loss_db** – Free-space path loss in dB.
- **rx_ant_gain_db** – Receiver antenna gain in dB.
- **atm_loss_db** – Total atmospheric loss in dB.
- **mispointing_db** – Antenna mispointing loss in dB.
- **feed_loss_db** – Antenna-to-LNA feed loss in dB.

Returns

Received power in dBW.

linkbudget.calc.**rx_sys_noise_temp**(*Tar*, *Te*, *feed_loss_db=0*, *feed_temp=290*)

Compute the receiver system noise temperature

The receiver system noise temperature is equal to the effective input noise temperature (Te) of the entire receiver (seen as a blackbox) added to the antenna noise temperature (Tar). The Te term represents the noise introduced by the cascaded linear components of the receiver (e.g., LNB, coax line, and radio interface). Meanwhile, The Tar component represents the cosmic noise and Earth blackbody radiation captured by the antenna. Hence, the simplified model is as follows:

```
Rx Antenna (Tar) -----> Sum ----> Noise-free Gain Stage ---> Detector
                         ^
                         |
                         |
                  Receiver Noise (Te)
```

Note the antenna is not treated as another cascaded device within the receiver. Instead, the antenna noise adds to the equivalent input noise of the cascaded devices, see Figure 8-24 in [1]. This is because the observation point is at the input to the receiver system. As explained in [2], around equation 4.39, this is a convenient choice in terms of where the effective input noise temperature is observed.

By definition, an equivalent (or effective) input noise temperature represents the thermodynamic temperature of a noisy resistor connected to the input of a noiseless two-port element, which gives the same output noise power as the noisy (but otherwise equivalent) two-port element with an ideal noiseless source at its input [2]. Hence, if we group the entire receiver into a single equivalent two-port element, the Te term represents the noise generated by the entire receiver, which has power k*Te*B. When combined to the noise collected by the antenna, the result becomes the total system noise temperature.

Since the observation point is at the Rx input, the carrier-to-noise ratio (CNR) becomes the carrier power captured by the antenna (i.e., at the Rx input) divided by k*Te*B. If the observation point were instead at the detector's input, both the carrier and noise powers would be multiplied by the combined gain of the cascaded linear components. Hence, the CNR would remain the same (see Equation 4.15 in [3]), which confirms the convenience of using the receiver's input as the reference point.

Besides, note the input to the receiver typically represents the input to the LNA, which is assumed to be directly connected to the antenna. However, in practice, there is a lossy passive two-port element (waveguide or transmission line) connecting the antenna to the LNA. When the loss in this segment is non-negligible, the model becomes:

```
Antenna (Tar) --> Waveguide/Line --> Sum --> Noiseless Rx --> Detector
                                      ^
                                      |
                                      |
                                Receiver Noise (Te)
```

In this case, the lossy waveguide or transmission line attenuates the antenna noise, but also generates noise of its own. This function takes both effects into account when a non-zero feed loss is specified.

Parameters

- **Tar** (*float*) – Antenna noise temperature in K.
- **Te** (*float*) – Effective input-noise temperature in K.
- **feed_loss_db** (*float*) – Loss (in dB) over the passive two-port feed connecting the antenna to the LNA.
- **feed_temp** (*float*) – Physical temperature of the feed.

Returns

Receiver system noise temperature in K.

linkbudget.calc.**spectral_density**(*power_dbw*, *bw*, *label=''*)

Compute the power spectral density (PSD) of a given signal

Assume the signal has a constant power spectral density over the given bandwidth. Equivalently, assume the signal behaves as white noise, i.e., with uncorrelated zero-mean samples in time.

Parameters

- **power_dbw** – Signal power in dBW.
- **bw** – Bandwidth in Hz.
- **label** – Optional label used for logging.

Returns
PSD in dBW/Hz.

2.3 Antenna Parameters

Antenna Calculations

References:

- [1] Timothy Pratt, Jeremy E. Allnutt, "Satellite Communications", 3rd Ed.
- [2] Couch, Leon W.. Digital & Analog Communication Systems.
- [3] Recommendation ITU-R S.465-6.

class linkbudget.antenna.**Antenna**(*freq*, *gain=None*, *diameter=None*, *efficiency=None*, *label='Dish'*)

Parabolic antenna

Parameters

- **freq** (*float*) – Operating frequency in Hz.
- **gain** (*float*) – Antenna gain in dB.
- **diameter** (*float*) – Diameter in m.
- **efficiency** (*float*) – Aperture efficiency.
- **label** (*str*) – Label used for logs.

Note: According to [1], the aperture efficiency is typically in the range 0.5–0.75 for parabolodial reflector antennas, lower for small antennas and higher for large Cassegrain and Gregorian antennas. For instance, Table 8-4 in [2] assumes an aperture efficiency of 0.56.

---

effective_aperture

Effective aperture area.

gain_db

Antenna gain in dBi.

_calc_eff_aperture(*diameter*, *aperture_efficiency*)

Compute the antenna's effective aperture area

The effective aperture area is given by:

$$A_e = \eta A,$$

where A represents the antenna's physical aperture area and η is the aperture efficiency.

If the aperture is circular with a diameter D in meters (or radius r), the physical aperture area is given by:

$$A = r^2 = \frac{D^2}{4}.$$

Hence, the effective aperture area becomes:

$$A_e = \frac{\eta D^2}{4}.$$

Parameters

- **diameter** (*float*) – Diameter in m.
- **aperture_efficiency** (*float*) – Aperture efficiency.

Returns
Effective aperture area in square meters (m^2).

_calc_gain(*freq*, *effective_aperture*)

Calculate the parabolic dish gain

The gain in linear units is given by:

$$G = \frac{4A_e}{^2},$$

where A_e is the effective aperture and is the wavelength.

Parameters

- **freq** (*float*) – Operating frequency in Hz.
- **effective_aperture** (*float*) – Effective aperture in square meters.

Returns
Gain in dB.

_infer_diameter(*effective_aperture*, *aperture_efficiency*)

Infer the diameter of an equivalent parabolic reflector

The physical aperture area A can be expressed in terms of the effective aperture A_e and the aperture efficiency η, as follows:

$$A = \frac{A_e}{\eta}.$$

If the aperture is circular with a diameter D in meters, the physical aperture area is given by:

$$A = \frac{D^2}{4}.$$

Hence, it follows that:

$$D = \sqrt{\frac{4A_e}{\eta\pi}}.$$

Note: This inference is useful when working with a non-parabolic antenna, such as a flat-panel antenna. In this case, you may know the antenna gain and aperture efficiency specifications, but not the physical diameter and aperture of the antenna. Meanwhile, the diameter may still be required for computations such as the tropospheric scintillation model from Recommendation ITU-R 618 (see, e.g., ITU-Rpy).

Parameters

- **effective_aperture** (*float*) – Effective aperture in square meters.

- **aperture_efficiency** (*float*) – Aperture efficiency.

Returns
Diameter in meters (m) of an equivalent parabolic reflector with the same aperture efficiency.

Return type
float

_infer_eff_aperture(*freq*, *gain_db*)

Infer the effective aperture area from the antenna gain

The effective aperture area can be inferred from the gain and wavelength, as follows:

$$A_e = \frac{G^2}{4}.$$

Parameters

- **freq** (*float*) – Operating frequency in Hz.
- **gain_db** (*float*) – Antenna gain in dB.

Returns
Effective aperture area in square meters (m^2).

Return type
float

off_axis_gain(*angle*)

Compute the off-axis co-polar antenna gain

Based on the reference earth station radiation pattern from ITU-R S.465-6 (01/2010 version) and the APEREC026V01 standard from the ITU-R antenna pattern list in https://www.itu.int/en/ITU-R/software/Pages/ant-pattern.aspx.

Parameters
angle (*float*) – Off-axis angle in degrees relative to the boresight. Must be within the [0, 180°) range.

Returns
(float) Off-axis antenna gain in dBi.

2.4 Antenna Look Angles

Satellite look angle computations

References:

- [1] https://www.ngs.noaa.gov/CORS/Articles/SolerEisemannJSE.pdf.
- [2] https://en.wikipedia.org/wiki/Earth_radius.
- [3] Maral, Gerard., Sun, Zhili., Bousquet, Michel. Satellite Communications Systems: Systems, Techniques and Technology. 3rd ed.

linkbudget.pointing.**get_default_tle_dataset_dir**()

Get the default directory for saving TLE datasets

linkbudget.pointing.**get_sat_pos_by_tle**(*name*, *group=None*, *obs_time=None*, *save_dir=None*, *no_save=False*)

Get satellite position from Two-Line Element (TLE) predictions

First, searches for the satellite on CelesTrak's TLE database. Then, computes the expected satellite position on the given observation time using the implementation from the Skyfield package.

The TLE dataset is queried using the format described in https://celestrak.org/NORAD/documentation/gp-data-formats.php. More specifically, it is either requested with:

https://celestrak.org/NORAD/elements/gp.php?NAME=VALUE&FORMAT=tle

or

https://celestrak.org/NORAD/elements/gp.php?GROUP=VALUE&FORMAT=tle

The former is used when the satellite is specified by name only (group set to None). The second format is used when the specific group on CelesTrak's database is also informed. See the groups at https://celestrak.org/NORAD/elements/. For example, the 'active' group holds a long list of active satellites, whereas the 'geo' group focuses on active geosynchronous satellites, and so on.

This function downloads the TLE dataset before processing, and the downloads are saved as txt files at *~/.link-budget/tle/* by default. When the dataset is already available locally (downloaded previously), this function does not need to download it again. This behavior can be customized using the *save_dir* and *no_save* arguments.

Parameters

- **name** (*str*) – Satellite name on the TLE database.
- **group** (*str, optional*) – Group to which the satellite belongs on the CelesTrak database. Defaults to None, in which case the satellite is searched by name over the entire database.
- **obs_time** (*datetime, optional*) – Observation time. Defaults to None, in which case the current time is used.
- **save_dir** (*str, optional*) – Directory on which the downloaded TLE dataset should be saved. Defaults to None, in which case the *~/.link-budget/tle/* directory is used.
- **no_save** (*bool, optional*) – Whether to save the downloaded TLE dataset permanently on the save directory (save_dir) to speed up future queries.

Returns
Satellite longitude (degrees), latitude (degrees), and altitude (meters).

Return type
tuple

linkbudget.pointing.**look_angles**(*sat_long*, *sat_lat*, *sat_alt*, *rx_long*, *rx_lat*, *rx_height=0*, *ellipsoid='WGS84'*)

Calculate look angles (elevation, azimuth) and slant range

Computes the angles relative to a reflector, either active (satellite) or passive (radar object). Compute using the rigorous ellipsoidal approach discussed in [1].

Parameters

- **sat_long** – Subsatellite point's longitude in degrees.
- **sat_lat** – : Subsatellite point's geodetic latitude in degrees.
- **sat_alt** – Satellite/reflector altitude in meters above the reference ellipsoid.
- **rx_long** – Longitude of the receiver station in degrees.
- **rx_lat** – Geodetic latitude of the receiver station in degrees.

- **rx_height** – Receiver station's orthometric height (height above sea-level) in meters. Defaults to zero.
- **ellipsoid** – Reference ellipsoid for the computation, GRS80 or WGS84. Defaults to WGS84.

Note: Positive longitudes are east of the prime meridian and negative longitudes are west of the prime meridian. Positive latitudes are north of the equator and negative latitudes are south of the equator.

Returns
Tuple with elevation (degrees), azimuth (degrees) and slant range (m).

linkbudget.pointing.**polarization_angle**(*sat_long*, *sat_lat*, *rx_long*, *rx_lat*)

Compute the polarization angle (skew) at a given location

A linearly polarized satellite transmission has the electric field oriented at a constant angle relative to a reference plane. For a satellite antenna, the reference plane is the equatorial plane. If the polarization is horizontal, the electric field is parallel to the equatorial plane. Otherwise, when the linear polarization is vertical, it is perpendicular to the equatorial plane.

The earth station antenna feed must have its polarization aligned with the polarization plane of the received wave. However, the local vertical (normal) plane does not match the equatorial plane due to the curvature of the earth. Hence, there is an angle difference between the polarization of the signal transmitted by the satellite and the apparent polarization of the received signal, which is known as the polarization angle or polarization skew (sometimes also referred to as "polarity", "polarity skew", "LNB skew", or just "skew").

When pointing an linearly-polarized earth station antenna, this angle has to be taken into account. In contrast, if pointing a circularly-polarized antenna, there is no need to compensate for the polarization skew.

For geostationary satellites, [3] presents the skew formula that follows:

$$\psi = \tan^{-1}\left(\frac{\sin(L)}{\tan(l)}\right)$$

where L is the relative longitude (difference between the earth station's longitude and the satellite longitude) and l is the earth station's latitude.

As indicated by the equation, the skew is 0 when the earth station and the satellite are at the same longitude (when the relative longitude is zero).

Note that, for a positive skew value, the LNB must be rotated clockwise, whereas, for a negative polarization angle, the LNB must be rotated counterclockwise. Furthermore, note that the clockwise and counterclockwise directions are relative to the front face (the feed horn) of the LNB. In other words, for a positive skew value, someone standing behind the dish and facing the satellite in the sky would rotate the LNB clockwise.

Parameters

- **sat_long** (*float*) – Subsatellite point's longitude in degrees.
- **sat_lat** (*float*) – Subsatellite point's geodetic latitude in degrees.
- **rx_long** (*float*) – Earth station's longitude in degrees.
- **rx_lat** (*float*) – Earth station's geodetic latitude in degrees.

Returns
Polarization angle in degrees.

Return type
float

2.5 Propagation Models

By default, the link budget tool assumes the location of interest experiences the atmospheric attenuation given by models from ITU-R P. Recommendations. In particular, it uses the model implementation imported from the ITU-Rpy package. Nevertheless, it is possible to override the modeled values and specify the atmospheric attenuation directly through option `--atmospheric-attenuation` (see *Propagation Options*).

The atmospheric attenuation models include the effects of rain, clouds, atmospheric gases, and tropospheric scintillation. Each of these effects involves several ITU-R P. recommendations. For instance, the following recommendations are used to model the rain attenuation:

1. **ITU-R P.838-3:** Specific attenuation model for rain for use in prediction methods (1992-1999-2003-2005).
2. **ITU-R P.618-13:** Propagation data and prediction methods required for the design of Earth-space telecommunication systems (12/2017).
3. **ITU-R P.1511-2:** Topography for Earth-space propagation modelling (08/2019).
4. **ITU-R P.839-4:** Rain height model for prediction methods (09/2013).
5. **ITU-R P.837-7:** Characteristics of precipitation for propagation modelling (06/2017).

The sequence used to predict the rain attenuation exceeded over 0.01% of an average year is as follows:

1. Find the rainfall rate in mm/h exceeded for 0.01% of an average year on the specific location of interest using the model and the database provided by Recommendation ITU-R P.837-7.
2. Using the carrier frequency, find the coefficients to obtain the rain attenuation per kilometer following Recommendation ITU-R P.838-3.
3. Find the mean annual rain height at the given location, i.e., how high is the melting layer, where the frozen precipitation turns into liquid precipitation. Use the model and the database from Recommendation ITU-R P.839-4.
4. Find the earth station's height at the given latitude/longitude by interpolating on the database from ITU-R P.1511-2.
5. Using the parameters from steps 2, 3, and 4, as well as the elevation, latitude, and carrier frequency, compute the effective slant path subject to rain attenuation using the procedure from Section 2.2.1.1 of Recommendation ITU-R P.618-13.
6. Finally, multiply the attenuation in dB/km (from step 2) by the effective slant path subject to rain attenuation (from step 5) to obtain the final rain attenuation in dB.

Distinct procedures apply to the other forms of atmospheric attenuation. Section 2.5 from Recommendation ITU-R P.618-13 outlines the references used for each computation. Refer to the ITU-Rpy's docs for further information.

`linkbudget.propagation.`**`atmospheric_attenuation`**(*lat*, *lng*, *elevation*, *pol_skew*, *freq*, *availability*, *dish_size*, *dish_eff*)

Total attenuation due to multiple sources of atmospheric attenuation

A wrapper to obtain the total atmospheric attenuation through ITU-Rpy.

Parameters

- **`lat`** (*`float`*) – Earth station's latitude in degrees.
- **`lng`** (*`float`*) – Earth station's longitude in degrees.
- **`elevation`** (*`float`*) – Elevation angle in degrees.
- **`freq`** (*`float`*) – Carrier frequency in Hz.
- **`availability`** (*`float`*) – Target link availability within [95 to 99.999%].

- **dish_size** (*float*) – Earth station's dish diameter in m.
- **dish_eff** (*float*) – Dish aperture efficiency within [0, 1].

Returns
Total atmospheric attenuation in dB.

Return type
float

2.6 Utility Functions

linkbudget.util.**db_to_lin**(*val_db*)

Decibels to linear value

Parameters
val – Value in decibels.

Returns
Corresponding value in linear scale.

linkbudget.util.**dbm_to_dbw**(*val_dbm*)

Convert dBm to dBW

Parameters
val_dbm – Value in dBm (decibels above 1 milliwatt).

Returns
Value in dBW (decibels above 1 Watt).

linkbudget.util.**dbw_to_dbm**(*val_dbw*)

Convert dBW to dBm

Parameters
val_dbw – Value in dBW (decibels above 1 Watt).

Returns
Value in dBm (decibels above 1 milliwatt).

linkbudget.util.**format_area**(*area*)

Format area given in m2

Parameters
area – Area in square meters (m2).

Returns
String with the area and an adjusted unit such as cm2, dm2, or m2.

linkbudget.util.**format_power**(*power*)

Format power given in Watts

Parameters
power – Power in Watts (W).

Returns
String with the power value and an adjusted unit such as kW or mW.

linkbudget.util.**format_rate**(*rate*)

Format data rate given in bps

Parameters
rate – Data rate in bits per second (bps).

Returns
String with the data rate and an adjusted unit such as Gbps, Mbps, or kbps.

linkbudget.util.**get_default_lb_dir**()

Get the default directory for link-budget-specific files

linkbudget.util.**get_home_dir**()

Get the user's home directory even if running with sudo

linkbudget.util.**lin_to_db**(*val*)

Linear value to decibels

Parameters
val – Linear scale value.

Returns
Corresponding value in decibels.

linkbudget.util.**log_header**()

Log header to form a table with result logs

linkbudget.util.**log_result**(*parameter*, *value*)

Log parameter-value result

Parameters

- **parameter** – Parameter to log.
- **value** – Corresponding value.

linkbudget.util.**wavelength**(*freq*)

Compute the radio wavelength for a given frequency

Parameters
freq – Radio frequency in Hz.

Returns
Wavelength in meters.

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX

Symbols

A

C

D

E

F

G

L

M

N

O

P

R

S

W