Satellite Link Budgets and P_E

Eren Can Gungor
Riccardo Miccini
Technical University of Denmark - DTU

December 15, 2016

Contents

1 Link budget model

A *link budget model* is a way of estimating the power P_R of a received signal by taking into account the possible sources of gain and loss over the trasmission. Since the aforementioned gains and losses are typically expressed in the form of ratios, the calculation of a link budget becomes trivial if employing decibels.

In the case of a satellite link, the following factors are taken into account in the budget:

$$P_{R,dB} = 20 \log_{10} \left(\frac{4\pi d}{\lambda} \right) + ERP_{dB} + G_{R,dB} - L_{0,dB} \tag{1}$$

$(\frac{4\pi d}{\lambda})^2$ Free-space loss

ERP Effective radiated power

G_R Gain of receiver antenna

L_0 Other losses (e.g. atmospheric absorption)

The following subsections will attempt to explain some of these factors.

1.1 Free-space loss

The free-space loss is the signal strength loss occurring in a line-of-sight path through free space (usually air), without accounting for reflection or diffraction. The equation for the free-space loss is given by:

$$FSPL = \left(\frac{4\pi d}{\lambda} \right)^2 \tag{2}$$

where d represents the distance between transmitter and receiver, and λ is wavelength of the signal with frequency f. In decibels, it can also be expressed as:

$$FSPL_{dB} = 20 \log_{10}(d) + 20 \log_{10}(f) - 147.55 \tag{3}$$

The equation results in a loss term, meaning that its value will always be confined between 0 and 1, or always negative when expressed in decibels.

It is clearly noticeable how the equation resembles that of the sphere surface area; with higher values of distance or signal frequency, the loss increases.

1.2 Effective radiated power

The effective radiated power is the equivalent power transmitted equally in all directions, from a theoretical spherically radiating source, also known as isotropic radiator. It is obtained by:

$$ERP_{dB} = 10\log_{10}(P_T) + G_{T,dB} \tag{4}$$

where P_T is the output power of the transmitter and G_T is the gain of its antenna in respect to the isotropic radiator model.

1.3 Link budget margin

The link margin is the additional amount of attenuation that a transceiving system can tolerate while still functioning. It is recommended to include a margin of a few decibels in the link budget, in order to account for unpredicted losses.

In the case of satellite communication, the margin might comprise of rain fade or atmospheric absorption. Other common sources of signal degradation which are included in the link margin are components tolerances and wear.

2 SNR and P_E

2.1 Calculation of signal-to-noise ratio

The SNR of the satellite transmission can be calculated as:

$$SNR_{dB} = P_{R,dB} - P_{int,dB} \tag{5}$$

where P_R is the previously introduced power of the received signal, and P_{int} is the power of the receiver output noise. The latter is provided by:

$$P_{int} = kT_R B \tag{6}$$

where T_R is the noise temperature of the receiver, B is the bandwidth of the transmission, and k is the Boltzmann constant.

2.2 Calculation of bit error probability

The bit error probability of a biphase-shift keying ($BPSK$) transmission is calculated as:

$$P_E = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{7}$$

It is possible to extrapolate the required energy-per-bit-to-noise spectral density ratio E_b/N_0 from the SNR figure. Multiplying numerator and denominator by the bit duration T_b gives:

$$SNR = \frac{P_R T_b}{k T_R T_b} \rightarrow \frac{E_b}{N_0 B T_b} \tag{8}$$

In case of a binary $BPSK$ (biphase-shift keying) transmission, the bandwidth B is $2/T_b$, so the factor BT_b is 2.

2.3 Impact of parameters d, λ, and B

This section will investigate the different roles of the receiver and transmitter distance, the wavelength of the signal, and its bandwidth, in the calculation of the signal-to-noise ratio and the bit error probability. The following graphs will show the surfaces generated by the combination of two of the aforementioned parameters:

- d (distance) is given values between 1 km and 10000 km

- λ (wavelength) is given values for frequencies between 300 MHz and 300 GHz

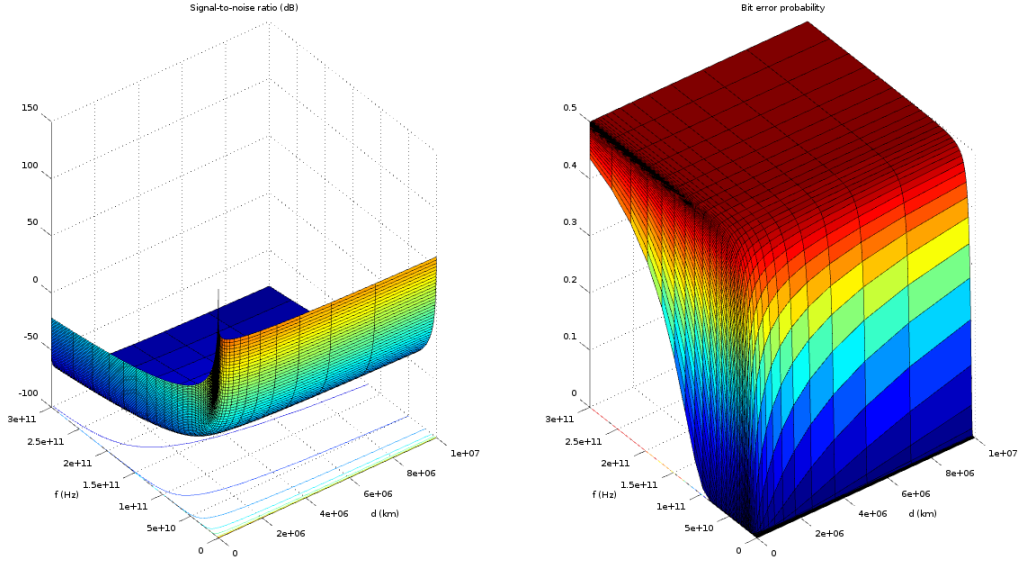- B (bandwidth) is given values between 10 kHz and 10 MHz

Figure 1: SNR_{dB} and P_E for different values of distance and wavelength

As previously introduced, increasing distance and frequency (thus decreasing wavelength) negatively affects the SNR and P_E values. The performance decay is quite drastic due to the power-of-two factor in the free-space loss.
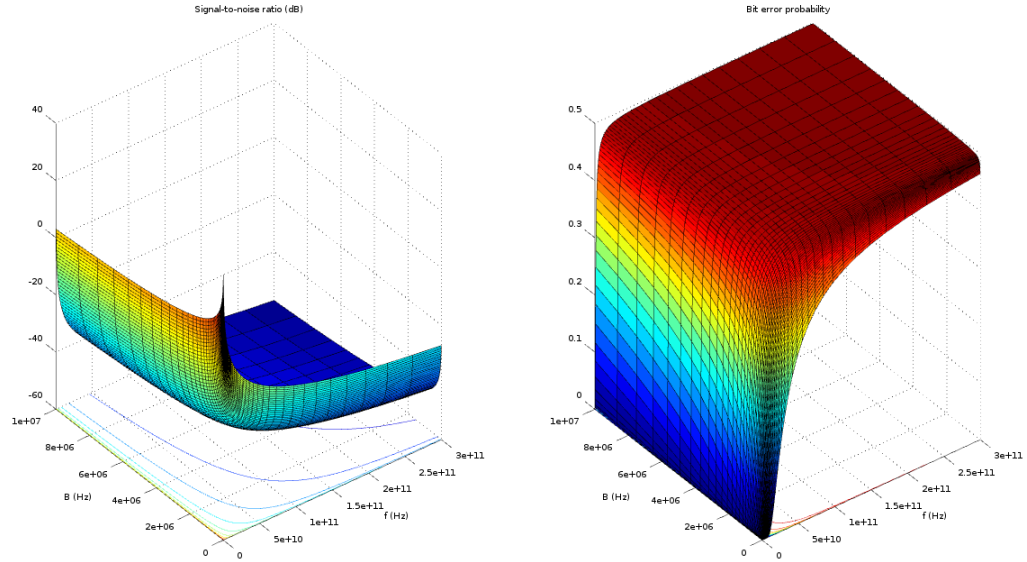

Figure 2: SNR_{dB} and P_E for different values of wavelength and bandwidth
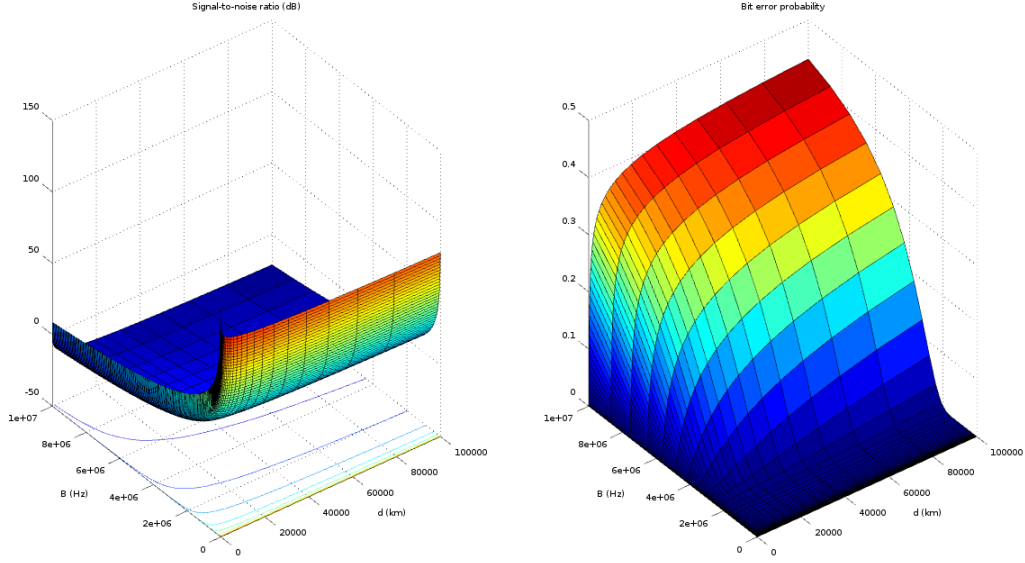
Figure 3: SNR_{dB} and P_E for different values of distance and bandwidth

In the four plots above, it can be noticed how a wider bandwidth does also negatively affect the performance of the system. However, it has a minor overall impact compared to distance and wavelength.

The MATLAB code used to generate the figures closely resembles the one used to calculate SNR and P_E, and can be found in the appendix.

2.4   Bit error probability at different transmission powers

In this section, the impact of the transmitter power will be analyzed, drawing from a sequence of given values. All the other parameters in the link budget model will use the values in the literature examples $A.8$ and $A.9$.

For transmitted power P_T values of 50 W, 5 W, and 500 mW, the bit error probability P_E results in:

```
p_e =
    1.4062e-05   9.2684e-02   3.3768e-01
```

With a power of 50 W, the bit error probability is barely adequate, while it reaches 9.3% and 34% at 5 W and 500 mW respectively. These last two figures would cause the transmission to be erratic beyond what is commonly considered usable.

Here follows a graph of the bit error probability across a range of transmitted power values:
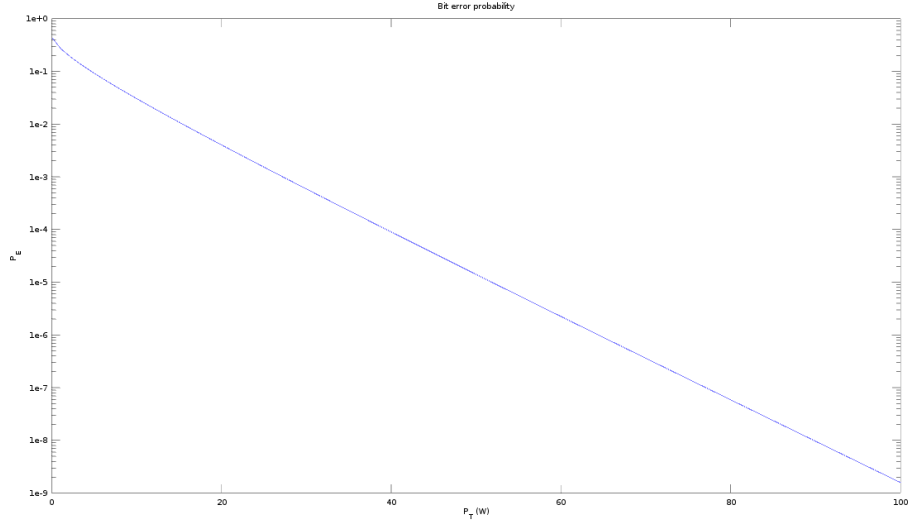
6

Figure 4: P_E for different values of transmitted power

As it can be seen here, it is necessary to transmit about 60 W of power in order to obtain a reliable link (bit error probability of less than 0.0001%)

3 Alternative modulation methods

This section will explore and discuss alternative modulation techniques, and determine their properties in terms of signal-to-noise ratio and bit error probability.

3.1 Method 1: Amplitude-shift keying

Amplitude-shift keying (ASK) is a modulation technique which consists in gating a sinusoidal signal on or off. The resulting symbol waveforms are:

- 0 for a 0 bits

- $A\cos(2\pi f_c t)\Pi(\frac{t-T/2}{T})$ for 1 bits

where $\Pi(x)$ is the rectangular function, and f_c is the carrier frequency.

It is possible to derive the bit error probability for ASK using its generic equations:

$$P_E = Q\left(\sqrt{(1 - R_{12})\frac{E_b}{N_0}}\right) \tag{9}$$

7

$$R_{12} = \frac{\sqrt{E_1 E_2}}{E_b} \rho_{12} \tag{10}$$

$$\rho_{12} = \frac{1}{\sqrt{E_1 E_2}} \int_{-\infty}^{\infty} s_1(t) s_2(t) dt \tag{11}$$

where ρ_{12} is a normalized correlation coefficient which expresses how similar the two symbol waveforms are, and R_{12} is related to the former.

In the case of ASK modulation, it holds that $R_{12} = \rho_{12} = 0$, meaning that the bit error probability equation becomes:

$$P_E = Q\left(\sqrt{\frac{E_b}{N_0}}\right) \tag{12}$$

Whilst the formula is similar to the one previously introduced for the calculation of P_E in a BPSK transmission, it lacks a $\sqrt{2}$ term. Therefore, the signal-to-noise radio of a ASK transmission has to be 3 dB higher in order to maintain the same performances.

For the examined satellite link, the bit error probabilities at 50 W, 5 W, and 500 mW would result in:

```
p_e =
    0.0015135   0.1742226   0.3834248
```

The values above show a considerable loss of performances when adopting ASK over BPSK.

3.2   Method 2: Frequency-shift keying

This modulation technique involves the transmission of sinusoidal waves with different frequency for each symbol. The resulting waveforms are:

- $A\cos(\omega_c t)$ for a 0 bits

- $A\cos((\omega_c + \Delta\omega)t)$ for 1 bits

For simplicity sake, it holds that:

- $\omega_c = \frac{2\pi n}{T}$

- $\Delta\omega = \frac{2\pi m}{T}$

where n and m are integer coefficients, so that the number of cycles within each symbol length T is integer too. These assumptions result in $R_{12} = 0$, and therefore the bit probability error equations is the same as for ASK:

$$P_E = Q\left(\sqrt{\frac{E_b}{N_0}}\right) \tag{13}$$

However, when deriving the E_b/N_0 ratio from the signal-to-noise ratio, the required bandwidth is given by $2.5/T$, so the factor BT_b becomes 2.5, or around 4 dB.

In the case of a FSK satellite link, the bit error probabilities at 50 W, 5 W, and 500 mW would result in:

```
p_e =
    4.5830e-04   1.4726e-01   3.7014e-01
```

Although this can be seen as an improvement over the ASK system, the calculated values, corresponding to probabilities of 0.0005%, 15%, and 37%, are still not adequate for a transmission link.

4 Source code

Here follows a thoroughly commented version of the MATLAB script used for the above calculations and observations.

4.1 SNR and P_E

The script below computes the signal-to-noise ratio and bit error probability for the free-space transmission characterized by its input parameters.

Listing 1: ../scripts/satellite.m

```
% script setup
clear all
pkg load communications

%%% constants
% standard temperature
t_0 = 290;
% boltzmann constant
k = 1.38064852e-23;
% B * T_b
bt_b_db = 3;

%%% input parameters
d = 41e6;
```

```
lambda = 0.15;
bw = 50e3;
p_t = 100;
g_t_db = 30;
a_r = 0;
g_r_db = 0;
l_0_db = 3;
t_r = 700;

% mag2db: 20 * log(x)
% pow2db: 10 * log(x)

% calculate free-space loss
fsl_db = mag2db(lambda / (4 * pi * d));

% calculate received signal power (A.70)
p_r_db = fsl_db + pow2db(p_t) + g_t_db + g_r_db - l_0_db;

% calculate noise level power
p_int_db = pow2db(k * t_r * bw);   % (A.72)
% p_int_db = pow2db(k * t_0) + pow2db(t_r / t_0) + pow2db(bw);
    % (A.73)

% calculate SNR at receiver output
snr_0_db = p_r_db - p_int_db


% calculate energy-per-bit-to-noise spectral density ratio
eb_n0_db = snr_0_db + bt_b_db;

% calculate bit error probability
p_e = qfunc(sqrt(2 * db2pow(eb_n0_db)))
```

4.2 d, λ, and B

The scripts below are used to investigate the relationships between the parameters d, λ, and B, and the signal-to-noise ratio and bit error probability.

Listing 2: ../scripts/satellite_plot_d_lambda.m

```
% script setup
clear all
pkg load communications

%%% constants
% standard temperature
t_0 = 290;
% boltzmann constant
```

```matlab
k = 1.38064852e-23;
% speed of light
c = 299792458;
% B * T_b
bt_b_db = 3;

%%% input parameters
% distance: logaritmically spaced values between 1km and 100000
    km
d = logspace(3, 8, 50);
% frequency: logaritmically spaced values between 300MHz and 300
    GHz (RF range)
f = 3.*logspace(8, 11, 50);
bw = 50e3;
p_t = 100;
g_t_db = 30;
a_r = 0;
g_r_db = 0;
l_0_db = 3;
t_r = 700;

% calculate free-space loss
fsl_db = mag2db(c ./ (4 * pi * f .* d'));

% calculate received signal power (A.70)
p_r_db = fsl_db + pow2db(p_t) + g_t_db + g_r_db - l_0_db;

% calculate noise level power
p_int_db = pow2db(k * t_r * bw);   % (A.72)

% calculate SNR at receiver output
snr_0_db = p_r_db - p_int_db;


% calculate energy-per-bit-to-noise spectral density ratio
eb_n0_db = snr_0_db + bt_b_db;

% calculate bit error probability
p_e = qfunc(sqrt(2 * db2pow(eb_n0_db)));

subplot(1, 2, 1);
surfc(d/1000, f, snr_0_db);
title('Signal-to-noise␣ratio␣(dB)');
xlabel('d␣(km)');
ylabel('f␣(Hz)');

subplot(1, 2, 2);
surfc(d/1000, f, p_e);
title('Bit␣error␣probability');
```

11

```
xlabel('d␣(km)');
ylabel('f␣(Hz)');
```

Listing 3: ../scripts/satellite_plot_lambda_b.m

```
% script setup
clear all
pkg load communications

%%% constants
% standard temperature
t_0 = 290;
% boltzmann constant
k = 1.38064852e-23;
% speed of light
c = 299792458;
% B * T_b
bt_b_db = 3;

%%% input parameters
d = 41e6;
% frequency: logaritmically spaced values between 300MHz and 300
    GHz (RF range)
f = 3.*logspace(8, 11, 50);
% bandwidth: logaritmically spaced values between 10kHz and 10
    kMHz
bw = logspace(1, 4, 50).*1000;
p_t = 100;
g_t_db = 30;
a_r = 0;
g_r_db = 0;
l_0_db = 3;
t_r = 700;

% calculate free-space loss
fsl_db = mag2db(c ./ (4 * pi * f * d));

% calculate received signal power (A.70)
p_r_db = fsl_db + pow2db(p_t) + g_t_db + g_r_db - l_0_db;

% calculate noise level power
p_int_db = pow2db(k * t_r .* bw');  % (A.72)

% calculate SNR at receiver output
snr_0_db = p_r_db - p_int_db;


% calculate energy-per-bit-to-noise spectral density ratio
eb_n0_db = snr_0_db + bt_b_db;
```

12

```octave
% calculate bit error probability
p_e = qfunc(sqrt(2 * db2pow(eb_n0_db)));

subplot(1, 2, 1);
surfc(f, bw, snr_0_db);
title('Signal-to-noise␣ratio␣(dB)');
xlabel('f␣(Hz)');
ylabel('B␣(Hz)');

subplot(1, 2, 2);
surfc(f, bw, p_e);
title('Bit␣error␣probability');
xlabel('f␣(Hz)');
ylabel('B␣(Hz)');
```

Listing 4: ../scripts/satellite_plot_d_b.m

```octave
% script setup
clear all
pkg load communications

%%% constants
% standard temperature
t_0 = 290;
% boltzmann constant
k = 1.38064852e-23;
% speed of light
c = 299792458;
% B * T_b
bt_b_db = 3;

%%% input parameters
% distance: logaritmically spaced values between 1km and 100000
   km
d = logspace(3, 8, 50);
f = 2e9;
% bandwidth: logaritmically spaced values between 10kHz and 10
   kMHz
bw = logspace(1, 4, 50).*1000;
p_t = 100;
g_t_db = 30;
a_r = 0;
g_r_db = 0;
l_0_db = 3;
t_r = 700;

% calculate free-space loss
fsl_db = mag2db(c ./ (4 * pi * f .* d'));
```

```octave
% calculate received signal power (A.70)
p_r_db = fsl_db + pow2db(p_t) + g_t_db + g_r_db - l_0_db;

% calculate noise level power
p_int_db = pow2db(k * t_r * bw);  % (A.72)

% calculate SNR at receiver output
snr_0_db = p_r_db - p_int_db;


% calculate energy-per-bit-to-noise spectral density ratio
eb_n0_db = snr_0_db + bt_b_db;

% calculate bit error probability
p_e = qfunc(sqrt(2 * db2pow(eb_n0_db)));

subplot(1, 2, 1);
surfc(d/1000, bw, snr_0_db);
title('Signal-to-noise ratio (dB)');
xlabel('d (km)');
ylabel('B (Hz)');

subplot(1, 2, 2);
surfc(d/1000, bw, p_e);
title('Bit error probability');
xlabel('d (km)');
ylabel('B (Hz)');
```

4.3 P_T

The scripts below are used to investigate the impact of the transmitter power on the bit error probability.

<div align="center">Listing 5: ../scripts/satellite_pt.m</div>

```octave
% script setup
clear all
pkg load communications

%%% constants
% standard temperature
t_0 = 290;
% boltzmann constant
k = 1.38064852e-23;
% B * T_b
bt_b_db = 3;
```

```matlab
%%% input parameters
d = 41e6;
lambda = 0.15;
bw = 50e3;
p_t = [50 5 .5];
g_t_db = 30;
a_r = 0;
g_r_db = 0;
l_0_db = 3;
t_r = 700;


% mag2db: 20 * log(x)
% pow2db: 10 * log(x)

% calculate free-space loss
fsl_db = mag2db(lambda / (4 * pi * d));

% calculate received signal power (A.70)
p_r_db = fsl_db + pow2db(p_t) + g_t_db + g_r_db - l_0_db;

% calculate noise level power
p_int_db = pow2db(k * t_r * bw);   % (A.72)
% p_int_db = pow2db(k * t_0) + pow2db(t_r / t_0) + pow2db(bw);
    % (A.73)

% calculate SNR at receiver output
snr_0_db = p_r_db - p_int_db


% calculate energy-per-bit-to-noise spectral density ratio
eb_n0_db = snr_0_db + bt_b_db;

% calculate bit error probability
p_e = qfunc(sqrt(2 * db2pow(eb_n0_db)))
```

4.4   Alternative modulation methods

The scripts below are used to investigate the properties and performances of
different modulation schemes.

Listing 6: ../scripts/satellite_ask.m

```matlab
% script setup
clear all
pkg load communications

%%% constants
% standard temperature
```

```
t_0 = 290;
% boltzmann constant
k = 1.38064852e-23;
% B * T_b
bt_b_db = pow2db(2);

%%% input parameters
d = 41e6;
lambda = 0.15;
bw = 50e3;
p_t = [50 5 .5];
g_t_db = 30;
a_r = 0;
g_r_db = 0;
l_0_db = 3;
t_r = 700;

% mag2db: 20 * log(x)
% pow2db: 10 * log(x)

% calculate free-space loss
fsl_db = mag2db(lambda / (4 * pi * d));

% calculate received signal power (A.70)
p_r_db = fsl_db + pow2db(p_t) + g_t_db + g_r_db - l_0_db;

% calculate noise level power
p_int_db = pow2db(k * t_r * bw);   % (A.72)
% p_int_db = pow2db(k * t_0) + pow2db(t_r / t_0) + pow2db(bw);
    % (A.73)

% calculate SNR at receiver output
snr_0_db = p_r_db - p_int_db;

% calculate energy-per-bit-to-noise spectral density ratio
eb_n0_db = snr_0_db + bt_b_db;

% calculate bit error probability
p_e = qfunc(sqrt(db2pow(eb_n0_db)))
```

Listing 7: ../scripts/satellite_fsk.m

```
% script setup
clear all
pkg load communications

%%% constants
% standard temperature
t_0 = 290;
```

```matlab
% boltzmann constant
k = 1.38064852e-23;
% B * T_b
bt_b_db = pow2db(2.5);

%%% input parameters
d = 41e6;
lambda = 0.15;
bw = 50e3;
p_t = [50 5 .5];
g_t_db = 30;
a_r = 0;
g_r_db = 0;
l_0_db = 3;
t_r = 700;

% mag2db: 20 * log(x)
% pow2db: 10 * log(x)

% calculate free-space loss
fsl_db = mag2db(lambda / (4 * pi * d));

% calculate received signal power (A.70)
p_r_db = fsl_db + pow2db(p_t) + g_t_db + g_r_db - l_0_db;

% calculate noise level power
p_int_db = pow2db(k * t_r * bw);   % (A.72)
% p_int_db = pow2db(k * t_0) + pow2db(t_r / t_0) + pow2db(bw);
    % (A.73)

% calculate SNR at receiver output
snr_0_db = p_r_db - p_int_db;

% calculate energy-per-bit-to-noise spectral density ratio
eb_n0_db = snr_0_db + bt_b_db;

% calculate bit error probability
p_e = qfunc(sqrt(db2pow(eb_n0_db)))
```