



**Instituto Tecnológico de Aeronáutica**

# **Aplicativo para cálculo de parâmetros de fibra óptica**

André Marcello Soto Riva Figueira

Daniel Prince Cerneiro

Dylan N. Sugimoto

Gabriel Adriano de Melo

São José dos Campos

01/07/2017

# Introdução

A proposta desse trabalho é fornecer um aplicativo web que a partir de alguns parâmetros sobre a fibra óptica retorna a distância máxima de um link feito por fibra óptica considerando o balanço de potência e o balanço de tempo de atraso e que tenha uma interface amigável com o usuário. De forma mais específica, o aplicativo resolve dois problemas correlacionados. O primeiro problema é a partir de um valor de potência de entrada em dB, um valor de atenuação em dB/km e uma sensibilidade do sensor em dB encontrar a distância máxima para que o link seja viável, que é conhecido por balanço de potência. Para resolver esse problema, considerou-se que a sensibilidade do sensor é menor do que a potência máxima de entrada, caso contrário, não seria possível atender à especificação para qualquer distância, pois a potência no sensor deve ser maior ou igual à sensibilidade do sensor para que o link seja viável. Assim, como a potência a cada ponto da fibra óptica decai com inclinação negativa com módulo igual à atenuação, para encontrar a distância máxima do link considerando apenas o efeito da atenuação, basta utilizar a equação 1.

$$d_{max} = \frac{P_a - S}{\alpha} \quad (1),$$

em  $d_{max}$  é a distância máxima em km,  $P_a$  é a potência de entrada em dB,  $S$  é a sensibilidade do sensor em dB e  $\alpha$  é a atenuação em dB/km.

O segundo problema é a partir do coeficiente de dispersão cromática, do coeficiente de dispersão modal, do tempo de subida do receptor, da largura espectral do laser e do tipo do modo da fibra óptica, encontrar a distância máxima para que o link tenha uma taxa de bits por segundo desejada, que é conhecido por balanço de tempo de atraso. Para resolver esse problema, começa-se calculando o tempo de bit a partir da taxa de bits por segundo, conforme equação 2.

$$T_{bit} = \frac{1}{T_x} \quad (2),$$

em que  $T_{bit}$  é o tempo de 1 bit em segundos e  $T_x$  é a taxa de transmissão em bits por segundo.

Admite-se que o tempo de atraso por dispersão aceitável seja em torno de um décimo do tempo de um bit (equação 3).

$$t_{disp} < 0.1 \cdot T_{bit} \quad (3),$$

em que  $t_{disp}$  é o tempo de atraso por dispersão em segundos e  $T_{bit}$  é o tempo de 1 bit em segundos.

Se a fibra óptica for do tipo multimodo, o tempo de dispersão cromática pode ser desprezado da equação 4 chegando-se à equação 5. E sabendo a equação 6, chega-se ao valor do comprimento máximo da fibra. Se a fibra óptica for do tipo monomodo, não há tempo de dispersão modal e o tempo de dispersão cromática é considerado no cálculo chegando-se a equação 7, e sabendo da equação 8, encontra-se a distância máxima do link.

$$t_{disp} = \sqrt{t_{rx}^2 + t_{modal}^2 + t_{cromatico}^2} \quad (4),$$

em que  $t_{disp}$  é o tempo de atraso por dispersão em segundos,  $t_{rx}$  é o tempo de subida,  $t_{modal}$  é o tempo de dispersão modal e  $t_{cromatico}$  é o tempo de dispersão cromática.

$$t_{modal}^2 < (0.1 \cdot T_{bit})^2 - t_{rx}^2 \quad (5),$$

em que  $t_{modal}$  é o tempo de dispersão modal,  $T_{bit}$  é o tempo de 1 bit em segundos e  $t_{rx}$  é o tempo de subida.

$$t_{modal} = L \cdot D_{modal} \quad (6),$$

em que  $t_{modal}$  é o tempo de dispersão modal,  $L$  é o comprimento da fibra óptica e  $D_{modal}$  é o coeficiente de dispersão modal.

$$t_{cromatico}^2 < (0.1 \cdot T_{bit})^2 - t_{rx}^2 \quad (7),$$

em que  $t_{cromatico}$  é o tempo de dispersão cromática,  $T_{bit}$  é o tempo de 1 bit em segundos e  $t_{rx}$  é o tempo de subida.

$$t_{cromatico} = L \cdot \Delta \lambda \cdot D_{cromatico} \quad (8),$$

em que  $t_{cromatico}$  é o tempo de dispersão cromática,  $L$  é o comprimento da fibra óptica,  $\Delta \lambda$  é a largura espectral do laser e  $D_{cromatico}$  é o coeficiente de dispersão cromática.

O tempo de subida do receptor por ser calculado multiplicando-se o inverso da banda de largura do receptor ( $B_{rx}$ ) por uma constante numérica, conforme equação 9.

$$t_{rx} = \frac{0.44}{B_{rx}} \quad (9),$$

em que  $t_{rx}$  é o tempo de subida do receptor e  $B_{rx}$  é a banda de largura do receptor.

Assim, a distância máxima para a fibra óptica de forma que o link seja viável é o mínimo entre as duas distâncias máximas calculadas.

## Descrição do Algoritmo

O código do arquivo “FiberOptical\_calculate\_parameters.js ” faz os cálculos do balanço de potência, do balanço de tempo de atraso, e da distância mínima do resultado encontrado para cada um dos balanços e também resolve o problema inverso do balanço de atraso, ou seja, ao invés de fornecer uma distância para uma dada taxa de bits por segundo, ele retorna uma taxa de bits dada uma distância. Essa última função foi necessário para plotar o gráfico de taxa de bits por segundo pela distância.

A primeira função do arquivo “FiberOptical\_calculate\_parameters.js” é a “datenuação” que faz o balanço de potência utilizando a equação 1 apresentada na introdução deste relatório. A segunda função é a “ddisp” que faz o balanço de tempo de atraso do sistema que implementou a lógica descrita para resolução desse problema explicada na introdução deste relatório por meio das equações 2 até 9, também apresentadas na introdução. A terceira função é a “dmax” que retorna o mínimo dos resultados encontrados pelas duas funções anteriormente citadas, caso ambas as funções tenham encontrado uma distância máxima para o seu respectivo problema, pois é possível dependendo das entradas que o link seja impossível para qualquer distância. Neste caso, a função “dmax” retorna o valor “NaN” (“not a number”) para indicar que não é possível realizar o link desejado para nenhuma distância. Por fim, tem-se a função “TxT” que resolve o problema inverso ao do balanço de tempo de atraso, e que utiliza as equações 2 até 9 da introdução.

# Utilização

## Conclusão

# Anexo

## Código do programa

### FiberOptical\_calculate\_parameters.js

```
1  /*
2  * To change this license header, choose License Headers in
Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6
7  /**
8   * @function [datenuacao] Calcula a distancia maxima devido
atenuacao
9   * @param {number} Pa potencia de entrada em dB
10  * @param {number} S sensibilidade do sensor em dB
11  * @param {number} alfa atenuacao em dB por km
12  * @returns distancia maxima em km
13  */
14  function datenuacao(Pa,S,alfa){
15      return (Pa-S)/alfa;
16  }
17
18  /**
19   * @function [Prr] Calcula a distancia maxima devido dispersao
20   * @param {number} Tx taxa de bits por segundo
21   * @param {number} Brx tempo de subida em MHz
22   * @constant {number} trx tempo de subida em ns
23   * @param {number} Dmodal dispersao modal em ns / km
24   * @param {number} Dcrom dispersao cromatica em ns / nm km
25   * @param {number} deltal largura espectral do laser em nm
26   * @param {boolean} singlemode verdadeiro se a fibra for
singlemode
27   * @returns distancia maxima em km
28   */
29  function ddisp(Tx,Brx,Dmodal,Dcrom,deltal,singlemode){
30      /*Calculo do tempo de 1 bit*/
31      Tbit = 1/Tx;
32      /*Calculo do tempo de atraso do sistema
33       * Admitindo no maximo um atraso de 0.1 do tempo do bit
34       * calculado em nanosegundos*/
35      tdisp = 0.1*Tbit*Math.pow(10,9);
36      /*Calculo do tempo de subida*/
37      trx = 0.44/Brx;
38      /*Transformando de microsegundo para nanosegundo*/
39      trx = trx*1000;
40      if (singlemode){
41
```



```

42         return Math.sqrt(Math.pow(tdisp,2) - Math.pow(trx,2))/
(deltal*Dcrom);
43
44     }
45     else{
46         return Math.sqrt(Math.pow(tdisp,2) -
Math.pow(trx,2))/Dmodal;
47     }
48
49 }
50
51
52 /**
53  * @function [dmax] Calcula a distancia maxima
54  * @param {number} Pa potencia de entrada em dB
55  * @param {number} S sensibilidade do sensor em dB
56  * @param {number} alfa atenuacao em dB por km
57  * @param {number} Tx taxa de bits por segundo
58  * @param {number} Brx tempo de subida em MHz
59  * @param {number} Dmodal dispersao modal em ns / km
60  * @param {number} Dcrom dispersao cromatica em ns / nm km
61  * @param {number} deltal largura espectral do laser em nm
62  * @param {boolean} singlemode verdadeiro se a fibra for
singlemode
63  * @returns distancia maxima em km
64  */
65 function dmax(Pa,S,alfa,Tx,Brx,Dmodal,Dcrom,deltal,singlemode)
{
66     /*calcula da distancia maxima
67     * considerando balanco de potencia*/
68     distA = datenuacao(Pa,S,alfa);
69     /*calcula da distancia maxima
70     * considerando balanco de tempo de atraso*/
71     distD = ddisp(Tx,Brx,Dmodal,Dcrom,deltal,singlemode);
72     if (distA > distD || isNaN(distD)){
73         return distD;
74     }
75     return distA;
76 }
77
78 /**
79  * @function [TxT] Calcula a taxa de trasmissao em Megabits
por segundo
80  * @param {number} d distancia em km
81  * @param {number} Brx tempo de subida em MHz
82  * @constant {number} trx tempo de subida em ns
83  * @constant {number} tdisp tempo de atraso por dispersao em
ns
84  * @param {number} Dmodal dispersao modal em ns / km
85  * @param {number} Dcrom dispersao cromatica em ns / nm km
86  * @param {number} deltal largura espectral do laser em nm
87  * @param {boolean} singlemode verdadeiro se a fibra for
singlemode

```

```

88  * @returns Taxa de trasmissao em Megabits por segundo
89  */
90  function TxT(d,Brx,Dmodal,Dcrom,deltal,singlemode){
91
92      /*Calculo do tempo de subida*/
93      trx = 0.44/Brx;
94      /*Transformando de microsegundo para nanosegundo*/
95      trx = trx*1000;
96      if (singlemode){
97
98          tdisp = d*deltal*Dcrom;
99          return 100/Math.sqrt(Math.pow(tdisp,2) +
Math.pow(trx,2));
100
101      }
102      else{
103
104          tdisp = d*Dmodal;
105          return 100/Math.sqrt(Math.pow(tdisp,2) +
Math.pow(trx,2));
106      }
107
108  }
109

```

## codigo.js

```
1 //UTILIZA A BIBLIOTECA math.js para o cálculo com números
complexos
2
3 /* global MathJax, Plotly */
4
5
6 function mudarEntradas() {
7     var singlemode = document.getElementsByName("singlemode")
[0].value;
8     var MMs = document.getElementsByClassName("mm");
9     var SMs = document.getElementsByClassName("sm");
10    if (singlemode) {
11        for (var i = MMs.length; i > 0; i--) {
12            MMs[i-1].style.display = 'none';
13        }
14        for (var i = SMs.length; i > 0; i--) {
15            SMs[i-1].style.display = '';
16        }
17    } else {
18        for (var i = MMs.length; i > 0; i--) {
19            MMs[i-1].style.display = '';
20        }
21        for (var i = SMs.length; i > 0; i--) {
22            SMs[i-1].style.display = 'none';
23        }
24    }
25 }
26
27
28
29 function calcular() {
30
31     var Pa = Number(document.getElementsByName("Pa")[0].value);
32     var S = Number(document.getElementsByName("S")[0].value);
33     var alfa = Number(document.getElementsByName("alfa")
[0].value);
34     var Tx = Number(document.getElementsByName("Tx")[0].value)
* Math.pow(10, 6);
35     var Brx = Number(document.getElementsByName("Brx")
[0].value);
36     var Dmodal = Number(document.getElementsByName("Dmodal")
[0].value);
37     var Dcrom = Number(document.getElementsByName("Dcrom")
[0].value)/1000;
38     var deltal = Number(document.getElementsByName("deltal")
[0].value);
39     var singlemode =
Boolean(document.getElementsByName("singlemode")[0].value);
40
```

```

41     lmax =
dmax(Pa,S,alfa,Tx,Brx,Dmodal,Dcrom,deltal,singlemode);
42
43     if (isNaN(lmax)) {
44         texto = "$ \\text{ N\u00e3o h\u00e1; nenhum comprimento para a
taxa de bits especificada. }$";
45     } else {
46         lmax = lmax.toFixed(2);
47         texto = "$ \\text{Comprimento m\u00e1ximo da Fibra } = " +
lmax + " km$";
48     }
49
50     document.getElementById("respostas").innerHTML = texto;
51     MathJax.Hub.Queue(["Typeset",MathJax.Hub,"respostas"]);
52
53     var comprimentos = [];
54     var taxas = [];
55
56     var comprimentoMax = datenuacao(Pa,S,alfa);
57     var comprimentoMin = comprimentoMax/100;
58     var passo = 100;
59
60     for (var i = 0; i < passo; i++) {
61         comprimentos[i] = comprimentoMin +
i/passo*(comprimentoMax-comprimentoMin);
62         taxas[i] =
TxT(comprimentos[i],Brx,Dmodal,Dcrom,deltal,singlemode);
63     }
64
65
66     var dados = [{
67         x: comprimentos,
68         y: taxas,
69         mode: 'lines',
70         line: {shape: 'spline'},
71         type: 'scatter'
72     }];
73
74     var estilo = {
75         title: 'An\u00e1lise de Taxa de Transmiss\u00e3o',
76         xaxis: {
77             title: 'Comprimento (Km) '
78         },
79         yaxis: {
80             title: 'Taxa de Transmiss\u00e3o (Mbits/s) '
81         }
82     };
83
84     Plotly.newPlot('grafico', dados, estilo);
85
86
87 }

```

