

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica
Departamento de Telemática

PC-SIMFO:
**Um Software para Simulação de Sistemas
de Comunicação por Fibra Óptica**

Autor: Patricia Maria Santos Lobão
Orientador: Prof. Dr. Edson Moschim†

Tese apresentada à Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas – FEE-UNICAMP, como parte dos requisitos exigidos para a obtenção do título de MESTRE EM ENGENHARIA ELÉTRICA.

Este exemplar, correspondendo à redação final da tese defendida por PATRICIA MARIA SANTOS LOBÃO e aprovada pela Comissão Julgadora em 29/07/1992.

Orientador

JULHO 1992.

TRADE	BC
COMPLAINT:	L7829
NO.	14303
DATE	215/92
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
AMOUNT	A\$ 20,000.00
DATE	09/09/92
CPD	

CM-00031298-1

A meus pais.
Para Adatao.

Agradecimentos

Este trabalho foi possível graças ao apoio e à colaboração de um elevado número de pessoas que, direta ou indiretamente, proporcionaram condições para o seu desenvolvimento. A todos quero expressar o meu agradecimento, destacando em especial:

- O Prof. Edson Moschim, pela orientação indispensável, e pela convivência agradável durante todo este tempo;
- Os amigos Tereza, Paulo, Sérgio e Menotti, entre tantos outros, pelas dicas valiosas e oportunas, e pelos momentos que passamos juntos;
- Os professores da FEE que ministraram os cursos que contribuíram na confecção deste trabalho;
- O CNPq e a FAEP, pelo apoio financeiro.

Sumário

O PC-SIMFO é um software dedicado ao projeto e análise, através de simulação, de sistemas de comunicação tendo como suporte de transmissão fibras ópticas. As entradas para a simulação são provenientes de uma biblioteca de modelos, que comporta os vários tipos de componentes optoeletrônicos disponíveis atualmente no mercado, e de uma base de dados com os valores dos principais parâmetros de cada um destes componentes, que podem ser alterados pelo usuário, possibilitando uma otimização sistêmica. Neste trabalho são apresentados os componentes básicos de um sistema de comunicação, os modelos matemáticos implementados e a organização geral e estrutura do software. São mostradas as facilidades disponíveis para o usuário, exemplificadas através de menus e gráficos.

Abstract

PC-SIMFO is a software for analysis and design of optical fiber communication systems, using simulation. The entries to the simulation are provided by a model library, which holds the transfer functions of the various optoelectronic components that are available; and from a database, which holds the values of the physical parameters of these components. These parameters can be modified by the user in order to optimize the system configuration. In this work we present the basic components of a communication system; the implemented mathematical models and the general organization and structure of the software. The facilities provided to the user are showed and exemplified by menus and graphics.

Conteúdo

Introdução Geral	1
1 Sistema de comunicação por fibra óptica	3
1.1 Introdução	3
1.2 Sistema de transmissão digital	3
1.3 Elementos básicos do modelo	4
1.3.1 Gerador de sinais	4
1.3.2 Codificador de linha	5
1.3.3 Fonte de Luz	5
1.3.4 Fibra Óptica	6
1.3.5 Fotodetector	8
1.3.6 Pré-amplificador	8
1.3.7 Filtro	9
1.3.8 Circuito de decisão	9
1.3.9 Decodificador	10
1.3.10 Ruído	10
1.4 Conclusão	10
Referências Bibliográficas	10
2 Estrutura do PC-SIMFO	12
2.1 Introdução	12
2.2 Conceitos básicos sobre o PC-SIMFO	13
2.2.1 Estrutura dos menus	14
2.2.2 Metodologia de Simulação	17
2.2.3 Programas e arquivos	18
2.3 Processamento e rotinas auxiliares	19
2.4 Conclusão	22
Referências Bibliográficas	22
3 Interfaces do PC-SIMFO	24
3.1 Introdução	24
3.2 Interface de entrada	24
3.2.1 Rotinas	24
3.3 Interface de saída	31
3.3.1 Gráficos	31
3.3.2 Diagrama de olho	33

3.3.3	Listagem de valores	34
3.4	Conclusão	34
	Referências Bibliográficas	34
4	Modelamento dos componentes	35
4.1	Introdução	35
4.2	Modelamento do transmissor	35
4.2.1	Diodos Emissores de Luz	36
4.2.2	Diodos laser	38
4.3	Modelamento do canal	39
4.3.1	Eficiência de acoplamento	41
4.3.2	Fator de atenuação	41
4.3.3	Resposta em frequência normalizada	43
4.4	Modelamento do receptor	44
4.4.1	Fotodiodo	46
4.4.2	Pré-amplificador	48
4.4.3	Filtro	51
4.5	Conclusão	55
	Referências Bibliográficas	55
5	Biblioteca de modelos	57
5.1	Introdução	57
5.2	Gerador de sinais	57
5.3	Codificador	58
5.4	LED	58
5.5	Laser	59
5.6	Fibra Óptica	61
5.7	Fotodiodo	63
5.8	Pré-amplificador	63
5.9	Filtro	64
5.10	Detector de Sinal	65
5.11	Medidor de erros	65
5.12	Ruído gaussiano	66
5.13	Conclusão	66
	Referências Bibliográficas	66
6	Aplicação do PC-SIMFO	68
6.1	Introdução	68
6.2	Análise de desempenho de um sistema de comunicação	68
6.2.1	Tecnologia de fibras ópticas de plástico	68
6.2.2	Tecnologia de fibras ópticas de sílica	73
6.3	Análise de desempenho de componentes	77
6.4	Conclusão	83
	Conclusão Geral	84

Apêndices	85
Modelo de sinal para potência emitida	85

Introdução Geral

A utilização de fibra óptica como suporte de transmissão em sistemas de comunicação apresenta uma série de vantagens em relação aos sistemas convencionais, dentre as quais imunidade eletromagnética, isolamento galvânica, grande banda passante, baixa atenuação, dimensão e peso reduzidos, além da abundância de matéria-prima. Assim, a tendência é de se substituir gradativamente os canais de comunicação de cabo metálico por fibras ópticas.

A grande variedade de componentes optoeletrônicos ¹ e de configurações de pré-amplificadores de baixo ruído possibilita várias opções na configuração de um sistema de comunicação por fibra óptica, tornando a tarefa de projetar e analisar um sistema consideravelmente complexa. Por outro lado, o desempenho de um sistema pode ser avaliado corretamente através de um modelo que englobe todas as características dos componentes básicos, possibilitando calcular parâmetros como taxa de erros, relação sinal/ruído, margem e perda de potência devido à atenuação e dispersão do canal, formas de ondas ópticas e elétricas e níveis de ruído na recepção.

Neste contexto, uma ferramenta interessante para avaliar a opção sistêmica mais adequada para uma dada aplicação é a análise de sistemas de comunicação com auxílio de computador. Esta tendência tem se generalizado a partir da proliferação de microcomputadores e estações de trabalho, que têm se tornado ferramentas indispensáveis na avaliação do desempenho de sistemas de comunicação.

A análise e o projeto de sistemas de comunicação assistidos por computador apresentam duas abordagens: na primeira, baseada em *cálculos analíticos*, o computador é utilizado para resolver equações matemáticas, substituindo números em fórmulas; na segunda, baseada em *simulação*, o computador é usado para simular um sinal que flui pelo sistema e a avaliação é baseada nas amostras do sinal transmitido. Este segundo enfoque torna o projeto mais flexível e permite a utilização de modelos tão detalhados quanto necessário, o que tem transformado a simulação computacional em uma ferramenta importante na concepção de sistemas de comunicação. O processo global de simulação inclui a construção, verificação e validação do modelo, bem como a verificação dos programas de computador associados e a validação dos dados de simulação.

Os softwares atualmente disponíveis para simulação caem em duas categorias: com topologia fixa e com topologia definida pelo usuário. Nos softwares que trabalham com topologia fixa, é apresentada a topologia do sistema a ser simulado ao usuário, que seleciona, através de menus, cada um dos componentes que participam na construção dos blocos que compõem o sistema. Nos softwares com topologia variável, o usuário especifica a topologia do sistema a ser simulado a partir de um diagrama de blocos, utilizando os módulos presentes em uma

¹Todo tipo de componente que gera, processa, conduz ou detecta fótons, tais como fontes de luz, multiplexadores e demultiplexadores de comprimento de onda, conectores, fibras ópticas, acopladores, derivadores, fotodiodos, etc.

biblioteca de modelos, e em seguida seleciona os componentes que representam cada um destes blocos. Em ambos os casos, a simulação é executada após a definição dos componentes do sistema.

O trabalho desenvolvido apresenta um software de simulação concebido dentro da filosofia de topologia fixa, funcionando como uma ferramenta auxiliar no projeto e na análise de sistemas de comunicação digital por fibra óptica com modulação por intensidade e detecção direta. O modelo adotado considera um sistema de comunicação ponto-a-ponto, formado por um transmissor, um canal e um receptor. A partir deste modelo geral podemos construir diferentes sistemas de comunicação variando os elementos que compõem a topologia. Uma vez definida a configuração do sistema, a simulação consiste dos seguintes passos: gerar amostras dos sinais que se deseja transmitir pelo sistema; executar operações de processamento do sinal no domínio do tempo e da frequência em cada um dos blocos que compõem o sistema; gerar e armazenar os valores amostrados do sinal transmitido nos vários pontos selecionados; analisar os valores amostrados obtidos no final da simulação e obter medidas de desempenho.

Esta dissertação é composta por seis capítulos, organizados de acordo com a descrição que se segue. No capítulo I são apresentados de maneira descritiva os componentes básicos de um sistema de comunicação por fibra óptica. No capítulo II é introduzido o software PC-SIMFO através da apresentação de seus objetivos, características e estrutura. No capítulo III são descritas as rotinas que implementam as interfaces com o usuário: menus e gráficos. Os modelos matemáticos usuais para caracterização de cada um dos componentes são apresentados no capítulo IV. No capítulo V é apresentada a biblioteca de modelos, que implementa os modelos matemáticos descritos no capítulo IV. Um exemplo de aplicação do software na análise de desempenho de uma transmissão digital é o tópico do capítulo VI. Finalmente é apresentada uma conclusão geral, onde são dadas sugestões para trabalhos futuros.

Capítulo 1

Sistema de comunicação por fibra óptica

Neste capítulo são apresentadas de maneira sucinta as propriedades dos componentes básicos que compõem um sistema de comunicação digital por fibra óptica. Estas propriedades serão utilizadas na implementação do simulador PC-SIMFO.

1.1 Introdução

De acordo com o tipo de ligação existente entre seus elementos, os sistemas de comunicação podem ser divididos em dois grupos: *sistemas ponto-a-ponto*, onde existe uma conexão individual entre um terminal transmissor e um terminal receptor; e *sistemas multiponto*, nos quais em uma mesma linha podem estar conectados diversos terminais. Na transmissão multiponto, onde as configurações mais comuns são o barramento, o anel e a estrela, pode-se ter uma estação enviando simultaneamente para várias outras através da mesma via ou várias estações tentando enviar uma informação ao mesmo tempo. Neste tipo de transmissão, um acoplador (passivo ou ativo) é utilizado para distribuir o sinal. A orientação do transporte da informação da origem ao destino e o gerenciamento do compartilhamento do canal são feitos através de protocolos de comunicação [1].

De maneira geral pode-se configurar qualquer tipo de sistema (ponto-a-ponto ou multiponto) através de uma célula padrão formada por um transmissor, um canal e um receptor.

1.2 Sistema de transmissão digital

Em um sistema de transmissão digital típico, formado por um transmissor, um canal e um receptor, o transmissor é composto por um codificador e uma fonte de luz associada com seu circuito de comando, o canal é formado por uma fibra óptica monomodo ou multimodo, e o receptor compreende um fotodetector, um pré-amplificador, um filtro, um circuito de extração de relógio, um amostrador/comparador e um decodificador [2][3].

Em uma transmissão digital binária, a entrada do transmissor é uma seqüência aleatória de 0's e 1's. No caso de sistemas por modulação de intensidade, a potência óptica emitida pela fonte de luz é proporcional ao sinal de entrada e é acoplada à fibra através de um conector. Este sinal óptico injetado se propaga na fibra, sofrendo atenuação e dispersão. No

receptor o sinal é detectado, amplificado e filtrado e na saída é amostrado e comparado com um limiar para a reconstrução da seqüência transmitida. A medida mais usual de avaliação de desempenho do sistema é a taxa de erros. O diagrama de blocos de um sistema de transmissão digital, utilizado como modelo para o desenvolvimento do software de simulação aqui proposto é mostrado na figura 1.1.

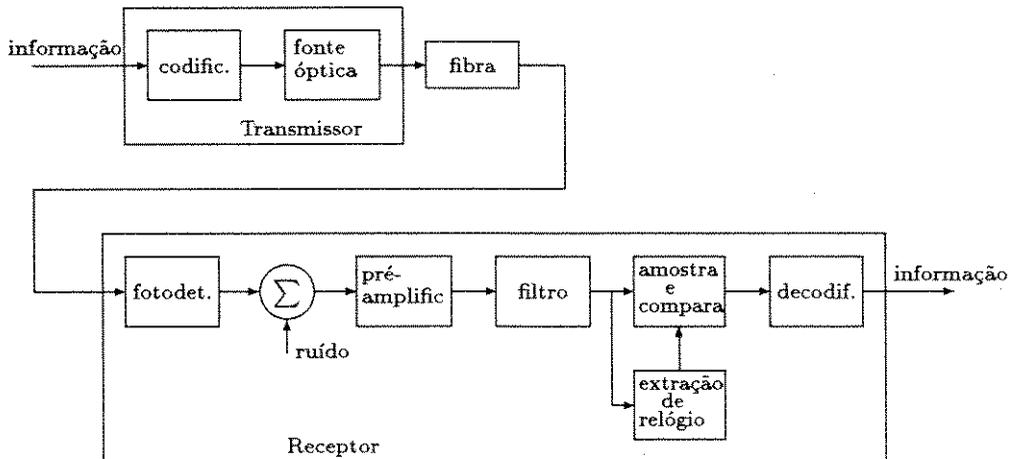


Figura 1.1: Sistema de comunicação digital por fibra óptica ponto-a-ponto.

No modelamento do sistema em estudo, as seguintes características são levadas em consideração: (a) a potência óptica emitida pela fonte de luz é modulada diretamente pela variação da amplitude da corrente injetada sobre ela (modulação em intensidade); (b) a potência óptica que chega no receptor é relacionada com a potência óptica emitida pela fonte de luz através da função de transferência da fibra óptica (a fibra apresenta uma resposta impulsiva em termos de potência óptica); (c) a potência óptica que chega no receptor é convertida diretamente em uma corrente eletrônica pelo fotodiodo (detecção direta) e (d) o sinal fotodetectado é pré-amplificado e filtrado, de forma a otimizar a relação sinal/ruído.

1.3 Elementos básicos do modelo

Além dos componentes básicos descritos anteriormente, um sistema de comunicação apresenta ainda dois componentes associados ao usuário do sistema: a fonte e o destino da informação. Apresentamos a seguir uma descrição básica dos elementos necessários para o modelamento do sistema descrito anteriormente.

1.3.1 Gerador de sinais

O gerador de sinais está associado basicamente com a fonte da informação e é descrito através da definição dos dados a serem transmitidos e da taxa de transmissão. O sinal na entrada do transmissor, representando a informação que será transmitida, corresponde a um conjunto de valores binários. Estes valores, que serão injetados no transmissor óptico, devem ser aleatórios, uma vez que a informação a ser transmitida é imprevisível. Uma seqüência randômica gerada no computador é, na realidade, pseudo-aleatória, já que é gerada a partir de um conjunto definido de instruções atuando sobre um conjunto finito de números. No

software desenvolvido estão disponíveis seqüências de três tipos: uma seqüência de 0s e 1s gerada aleatoriamente e duas seqüências de sinais determinísticos do tipo trem de pulsos e pulso isolado. Caso o usuário opte pela seqüência aleatória, é utilizada uma função interna do TURBOC de geração de números randômicos. São gerados inteiros entre 0 e 9 que são mapeados nos níveis 0 e 1 de acordo com um limiar definido pelo usuário [4]. Por exemplo, se o limiar escolhido for 5 todos os números randômicos gerados entre 0 e 4 serão mapeados em 0 e os demais em 1. Caso se deseje uma geração desbalanceada de zeros e uns, pode-se variar o limiar. No trem de pulsos é gerada uma seqüência constante de 1s e 0s, nesta ordem. O pulso isolado corresponde a um bit de valor 1 seguido de 0s.

1.3.2 Codificador de linha

A codificação de linha é usada para minimizar a ocorrência de longas seqüências de 0s e 1s, otimizando a transmissão em relação à sincronização na recepção. Os esquemas de codificação mais comumente usados são: Não Retorna a Zero (NRZ), Retorna a Zero (RZ) e Manchester [5].

O código mais simples é o NRZ, que codifica o sinal numa seqüência do tipo ON-OFF. A emissão toma a forma de um curto pulso de radiação durante cada período de um bit. No código RZ o sinal 1 é codificado como um pulso que dura o intervalo de tempo de metade de um bit, ou seja, o sinal permanece com valor 1 e, na metade do intervalo cai para zero. O sinal 0 é representado pela ausência de pulso. No código Manchester usa-se a codificação de fase, isto é, os bits não são reconhecidos pelo nível dos pulsos, mas sim pelas fases (nas transições). O bit 0 é representado por uma transição de descida no meio do intervalo e o bit 1 por uma transição de subida.

1.3.3 Fonte de Luz

As fontes de luz utilizadas como transmissores em sistemas de comunicação por fibra óptica são basicamente junções *pn* em semicondutores [6]. Quando a junção é polarizada diretamente, parte da energia fornecida ao dispositivo pela corrente elétrica é emitida em forma de luz, proporcional ao sinal na entrada do transmissor. A potência óptica emitida pela fonte de luz deve ser suficiente para sobrepor todas as perdas no sistema e produzir um sinal que possa ser detectado e amplificado no receptor.

Os **lasers** a semicondutor e os **LEDs** são as fontes de potência de luz mais apropriadas, dadas as suas características: configuração e tamanho compatíveis com as fibras; emissão de radiação nos comprimentos de onda de baixa atenuação das fibras; simplicidade de modulação direta até a faixa de GHz para o laser; facilidade de acoplamento com fibras através de conectores; largura espectral estreita (minimizando a dispersão); manutenção da potência de saída constante com o envelhecimento; baixo preço e alta confiabilidade.

Os lasers e LEDs apresentam qualidades particulares de operação e a escolha entre um ou outro está relacionada com o tipo de sistema desejado. Os parâmetros básicos de julgamento para o projeto de sistemas são: nível de potência de transmissão, sensibilidade à temperatura e ao envelhecimento, tempo de resposta, resposta em freqüência, linearidade na potência de transmissão, tempo de vida e largura espectral.

O laser é uma fonte apropriada para sistemas de alta velocidade devido ao rápido tempo de resposta e uma largura espectral estreita, podendo acoplar um alto nível de potência

em fibras ópticas e apresentando um tempo de vida em temperatura ambiente entre 10^5 a 10^6 horas de operação. A grande desvantagem é a diminuição da estabilidade da potência transmitida com a temperatura e com o envelhecimento.

Em sistemas com baixa velocidade ou modulação analógica de pequeno alcance, o LED apresenta-se como a fonte mais apropriada. Sua grande vantagem está na estabilidade da potência transmitida com a temperatura e com o envelhecimento durante a sua longa vida de operação (10^6 a 10^7 horas).

Na modelagem das fontes ópticas as características mais importantes são a potência óptica, o espectro de radiação e a forma de onda do sinal óptico na saída do transmissor, que depende da resposta em frequência do dispositivo.

- A **potência óptica** emitida pelo LED é, com uma boa aproximação, proporcional à corrente injetada, embora exista uma tendência à saturação na saída para altos níveis de potência, conforme a temperatura da junção aumenta. O comportamento do laser é mais complexo. Ele é construído de forma que, acima de uma corrente de limiar o mecanismo de geração de luz muda: a baixas correntes, a potência emitida é produzida da mesma maneira que em um LED, através de emissão espontânea. Acima do limiar, correspondendo à região de emissão estimulada, o laser atua como um oscilador.
- A **radiação** emitida pelo LED é incoerente e cobre um amplo espectro de comprimento de onda. No caso do laser, abaixo da corrente de limiar a radiação é produzida da mesma forma que no LED. Acima do limiar, porém, há uma mudança na característica da radiação emitida: ela se torna mais direcional, mais coerente e o espectro se torna mais estreito. Tanto a corrente de limiar quanto o espectro de radiação são sensíveis à temperatura e podem variar com as condições ambientais e altas potências.
- A **resposta em frequência** de uma fonte de luz relaciona a potência emitida com a corrente injetada modulada em frequência no dispositivo, definindo a banda passante. As equações que caracterizam o LED e o laser são apresentadas no capítulo 4.

1.3.4 Fibra Óptica

As fibras ópticas são guias de onda de forma cilíndrica fabricadas de materiais dielétricos e transparentes, cuja função é confinar e guiar energia eletromagnética na região do comprimento de onda de 0.4 a $1.6 \mu\text{m}$ [7]. O princípio de guiamento é a reflexão interna total, obtida por uma interface entre duas regiões de índice de refração diferentes, denominadas núcleo e casca. Os materiais básicos para a fabricação de fibras são os vidros (SiO_2) e os plásticos (polímeros). A tecnologia de fibras ópticas em plástico oferece um menor custo, entretanto, seu desempenho é inferior ao das fibras de sílica devido às grandes perdas.

As fibras atualmente disponíveis para uso como suporte de transmissão em sistemas de comunicação são de dois tipos: **multimodo**, que permitem a propagação de centenas de modos através de um núcleo com diâmetro da ordem de 50 a $1000 \mu\text{m}$; e **monomodo**, que possuem o diâmetro do núcleo pequeno (3 a $10 \mu\text{m}$) e permitem a propagação de um único modo. As fibras multimodo possuem ainda uma caracterização em relação ao perfil do índice de refração, podendo ser: **índice degrau**, que apresentam uma distribuição do índice de refração homogênea no núcleo, que é limitado por uma casca com índice de refração menor; e **índice gradual**, que apresentam uma distribuição do índice de refração não-homogênea

no núcleo e que decresce em sentido radial continuamente do núcleo até a casca e é constante na casca.

As principais características de transmissão das fibras ópticas são a abertura numérica, a atenuação e a dispersão. Estas características estão diretamente ligadas às propriedades do material que constitui a fibra.

- A **abertura numérica** de uma fibra dá a medida do ângulo de aceitação dos raios incidentes e, por consequência, a potência da fonte que pode ser acoplada na fibra para uma máxima transferência.
- A **atenuação**, que é devida à absorção e à difusão, dá a perda de potência do sinal emitido durante o percurso na fibra. A atenuação não depende somente do material do núcleo, por onde o sinal é transmitido, mas também da casca, já que no processo de reflexão interna os campos eletromagnéticos penetram na interface núcleo-casca e se estendem na casca. Se esta for de baixa qualidade irá contribuir na atenuação geral da fibra.
 - A **absorção** converte a energia eletromagnética guiada em calor (fótons \rightarrow fónons). Dois tipos de absorção são correntemente encontrados nas fibras ópticas: a *absorção intrínseca*, que é inerente ao material que constitui a fibra e portanto não pode ser eliminada, e a *absorção extrínseca*, que é devida às impurezas e pode ser minimizada em função do processo de fabricação da fibra. A absorção intrínseca aparece em função dos átomos que compõem o material da fibra. Na região do ultra-violeta ($\lambda < 0.6\mu m$) esta absorção é devido às transições eletrônicas e na região do infravermelho ($\lambda > 1.6\mu m$), assim como nos picos da região visível, é devido às vibrações harmônicas dos radicais CH, OH e NH. A absorção extrínseca ocorre devido à contaminação pelos componentes metálicos que têm uma influência na região de comprimento de onda visível como cobre, cromo, manganês e ferro. Em geral o nível destas contaminações é controlado no processo de fabricação e esta perda pode ser minimizada.
 - A **difusão**, ou espalhamento, converte a energia eletromagnética guiada em energia não-guiada. Dois mecanismos de difusão são importantes em uma fibra óptica: a difusão devido ao material (difusão intrínseca) e difusão devido a irregularidades na interface núcleo/casca (difusão extrínseca). A difusão intrínseca, cujo principal mecanismo é a difusão Rayleigh, é causada pela não-homogeneidade da fibra em pequena escala e à flutuação térmica do material.

As fibras ópticas apresentam uma perda suplementar independente do comprimento de onda, mas dependente do comprimento da fibra, chamada **perda do guia de onda**, que elimina os raios com ângulos próximos ao ângulo crítico. Esta perda ocorre devido à fronteira entre o núcleo e a casca da fibra e pode ser decomposta em duas partes: uma devido à absorção da interface (atenuação da interface) e outra devido às irregularidades da interface (difusão da interface).

- A **dispersão** é um mecanismo que limita a banda passante das fibras ópticas e é devido à dispersão modal, à dispersão cromática ou material e, no caso de fibras monomodo, à dispersão do guia de onda. A dispersão modal ocorre em fibras multimodo e é

devido à diferença nas velocidades de propagação dos modos guiados. A dispersão cromática, comum a todas as fibras, é devido à diferença de velocidade na propagação da energia eletromagnética para os diferentes comprimentos de onda emitidos pela fonte de radiação. Esta dispersão pode ser minimizada com a utilização de fontes de luz monocromáticas. A dispersão do guia de onda ocorre devido à variação da velocidade de grupo com o comprimento de onda para um modo particular.

1.3.5 Fotodetector

O fotodetector consiste de um fotodiodo e seus circuitos eletrônicos associados. Os fotodiodos são basicamente junções *pn* polarizadas reversamente em semicondutores que convertem a energia luminosa incidente sobre sua superfície em uma corrente eletrônica de forma proporcional à sua responsividade e, se for o caso, ao seu ganho de avalanche. Os dois tipos de detectores usados em sistemas de comunicação por fibras ópticas são: **PIN** (*positive-intrinsic-negative*), que não amplifica o sinal, e **APD** (*avalanche photodetector*), que produz um ganho sobre o sinal [8]. A escolha entre fotodiodos PIN ou APD é feita geralmente levando-se em conta custo x sensibilidade necessária. Para aplicações nas quais um sinal óptico de baixa potência é recebido um fotodiodo de avalanche é normalmente usado, dado que tem uma maior sensibilidade devido ao mecanismo interno de ganho (efeito de avalanche). Os principais parâmetros dos fotodiodos são a eficiência quântica e a velocidade de resposta.

- A **eficiência quântica** é definida como a taxa de potência óptica absorvida por colisões de ionização na região de depleção em relação à potência incidente total e está relacionada com a quantidade de pares elétron-lacuna gerados pela irradiação, diretamente proporcional ao volume do fotodiodo irradiado. A eficiência quântica pode ser interpretada como a probabilidade de que um fóton incidente irá produzir um par elétron-lacuna.
- A **velocidade de resposta** é definida como a taxa que o fotodiodo pode responder a mudanças na intensidade do sinal óptico de entrada e é a medida da taxa de transmissão máxima ou largura de banda do fotodetector. A velocidade de resposta está limitada ao tempo de trânsito dos pares elétron-lacuna fotogerados: quanto maior o volume irradiado, maior o tempo gasto no percurso destes pares e, por consequência, menor a velocidade de resposta do fotodiodo. Existe, assim, um compromisso entre a eficiência quântica e a velocidade de resposta dos fotodiodos.

1.3.6 Pré-amplificador

Na maioria dos casos o sinal na saída de um fotodiodo é fraco e geralmente requer amplificação de baixo ruído para um futuro processamento. Geralmente, o estágio de pré-amplificação apresenta um controle automático de ganho (AGC) para compensar as variações no nível de potência recebido. Assim, o circuito seguindo o fotodetector pré-amplifica e formata o sinal através de um filtro e o envia a um circuito de extração da informação [9]. O projeto do amplificador que segue o fotodiodo é de importância crítica, porque é no amplificador que as maiores fontes de ruído aparecem.

Duas configurações de pré-amplificadores são frequentemente usadas em receptores ópticos: **alta-impedância** e **transimpedância**. O amplificador de alta-impedância apresenta uma alta resistência de entrada para reduzir o efeito do ruído térmico. Esta estrutura, porém, apresenta uma resposta em frequência degradada, já que aumenta a impedância de baixa frequência. A saída do detector é efetivamente integrada em uma constante de tempo e deve ser restaurada por diferenciação, atenuando os componentes de baixa frequência do sinal e reduzindo a faixa dinâmica. A configuração de transimpedância supera a anterior através da utilização de um amplificador com realimentação negativa de baixo ruído. A maior vantagem desta configuração é a grande faixa dinâmica, resultante do mecanismo de atenuação diferente para os componentes de baixa frequência do sinal, que são amplificados pela malha fechada no dispositivo.

O transistor menos ruidoso é o transistor a efeito de campo, **FET**, que controla o fluxo de corrente com um campo elétrico produzido por uma voltagem aplicada na entrada do dispositivo. A alta-impedância de entrada, associada com seu baixo ruído e capacitância fazem com que o FET seja mais apropriado para amplificação de sinais fotodetectados. Contudo, as propriedades superiores do FET sobre os transistores **bipolares** são limitadas pela sua comparativamente baixa transcondutância, restringindo o ganho. Assim, em frequências acima de 25 MHz o transistor bipolar é mais adequado.

1.3.7 Filtro

A função do filtro é maximizar a relação sinal/ruído, preservando as características essenciais do sinal, isolando-o de interferências, ruídos e distorções. Um filtro pode ser **equalizante**, quando compensa a distorção do sinal devido a características do transmissor, do canal e do receptor, ou **não equalizante**, quando apenas reduz a banda do ruído [10].

O filtro equalizante é formado por um circuito de compensação do efeito integrador do transmissor, do efeito dispersivo do canal, do efeito integrador do receptor e por um estágio de filtragem para dar a forma desejada de sinal equalizado e eliminar o ruído fora da faixa útil [11]. Este estágio define a banda equivalente de ruído do receptor e o formato do pulso detectado.

Para pequenas dispersões no canal não existe diferença significativa no desempenho de receptores equalizantes ou não-equalizantes. Para altas dispersões, os não-equalizantes apresentam uma penalidade de potência média menor do que os equalizantes.

O tipo de filtro mais simples é o **butterworth** passa-baixa de n -ésima ordem, cujo circuito contém n elementos reativos (capacitores e indutores). O filtro do tipo **cosseno levantado** introduz mais ruído, porém é mais tolerante a rajadas e espalhamento do pulso.

1.3.8 Circuito de decisão

Em um receptor digital a filtragem é seguida por um componente responsável por regenerar o sinal recebido. Após a amplificação, a informação de relógio é extraída do sinal e é usada para comandar um circuito amostrador/comparador, que amostra a forma de onda em algum ponto durante cada período de bit e compara com um limiar previamente definido. Se a amplitude excede o limiar estabelecido, um 1 é regenerado, se não, um 0 é assumido. Quando ocorrem erros, a forma de onda regenerada difere da originalmente transmitida.

1.3.9 Decodificador

Os pulsos regenerados requerem decodificação a fim de recuperar a forma de onda original do sinal. O tipo de codificação determina, obviamente, a forma do decodificador, que deve fazer a operação inversa do codificador.

1.3.10 Ruído

As fontes de ruído do sistema são devidas principalmente ao ruído do fotodetector, resultante do processo de conversão fóton-elétron, e ao ruído térmico, associado com o circuito amplificador [11]. Em receptores que utilizam fotodiodos PIN ou fotodiodos APD operando a baixo ganho, o efeito do ruído balístico gerado pelo diodo no processo de fotodeteção é desprezível se comparado ao ruído térmico introduzido no estágio de amplificação. Neste caso o ruído será praticamente independente do sinal e sua variância será relacionada à variância do ruído térmico estacionário introduzido no pré-amplificador. Em receptores que utilizam fotodiodos APD, operando com alto ganho, o ruído balístico é realçado pelo mecanismo de avalanche e se torna comparável ou maior que o ruído térmico.

1.4 Conclusão

Neste capítulo foram apresentados de maneira descritiva os elementos básicos que compõem um sistema de transmissão digital por fibra óptica. O projeto de um sistema de comunicação envolve a análise de diversas configurações. O ponto inicial compreende geralmente algumas especificações descrevendo os objetivos do sistema e parâmetros de desempenho. Todos os componentes apresentam características que podem limitar o desempenho do sistema, portanto, é necessária uma compreensão de cada um deles antes que o projeto seja executado. No capítulo seguinte será apresentada a estrutura do simulador PC-SIMFO.

Referências Bibliográficas

- [1] I. P. Kaminow, "Photonic local networks", cap 26, *Optical fiber telecommunications II*, editado por S. Miller e I. P. Kaminow, John Wiley, 1988.
- [2] J. Gowar, *Optical communication systems*, Prentice-Hall, 1984.
- [3] G. Keiser, *Optical fiber communications*, McGraw-Hill, 1983.
- [4] V. N. Yarmolik e S. N. Demidenko, *Generation and application of pseudorandom sequences for random testing*, John Wiley, 1988.
- [5] D. J. Morris, *Pulse code formats for fiber optical data communication: Basic principles and application*, Marcel Dekker, Inc., 1987.
- [6] H. Kressel e J. K. Butler, *Semiconductor lasers and heterojunction LEDs*, Academic Press, 1977.
- [7] J. E. Midwinter, *Optical fibers for transmission*, Wiley-Interscience, 1979.

-
- [8] D. P. Schinke, R. G. Smith e A. R. Hartman, "Photodetectors", cap 3, *Semiconductor devices for optical communication - Topics in applied physics*, editado por H. Kressel, Springer-Verlag, 1980.
- [9] R. G. Smith e S. D. Personick, "Receiver design for optical fiber communication systems", cap 4, *Semiconductor devices for optical communication - Topics in applied physics*, editado por H. Kressel, Springer-Verlag, 1980.
- [10] A. B. Carlson, *Communication systems - An introduction to signals and noise in electrical communication*, McGraw-Hill, 1986.
- [11] S. D. Personick, "Receiver design for digital fiber optical communication systems I and II", *Bell Syst. Tech. J.*, vol 52, pp. 843-886 (1973).

Capítulo 2

Estrutura do PC-SIMFO

Neste capítulo é apresentada a estrutura do simulador PC-SIMFO. Cada um dos módulos que compõe o software é analisado e suas rotinas são descritas.

2.1 Introdução

O ambiente de simulação tem mudado rapidamente nos últimos anos. Até o início dos anos 70, uma simulação envolvia computadores de grande porte operando em *batch*, com programas normalmente escritos em FORTRAN. A partir do final dos anos 70 a tendência era de se utilizar mini e microcomputadores interativamente e linguagens de programação como PASCAL e C. A tendência atual é o uso de estações de trabalho e de PCs e as linguagens de programação C e LISP [1][2][3][4]. Para muitas aplicações, um PC tem poder de computação suficiente para permitir avaliação de projeto e análises precisas de sistemas de comunicação, oferecendo diversas vantagens, sendo a principal o baixo custo [5][6][7]. Uma outra vantagem é que, por dificilmente ser um equipamento compartilhado, o usuário tem completo domínio sobre o ambiente de simulação e longos tempos de processamento não se tornam um problema. Além disso, o uso de um recurso dedicado encoraja experimentação tanto do sistema em estudo como da própria metodologia de simulação.

Os softwares para análise e projeto de sistemas de comunicação baseados em simulação usam uma estrutura modular a fim de apresentar uma máxima flexibilidade. Os principais elementos presentes nestes softwares são: biblioteca de modelos, configurador de simulação, gerenciador de simulação e pós-processador [8]. A figura 2.1 apresenta a estrutura funcional de um típico software de simulação.

- A **biblioteca de modelos** contém os modelos matemáticos pré-programados dos componentes utilizados na configuração de um sistema de comunicação, tais como: geradores de sinal, codificadores/decodificadores, fontes de luz/detectores, fontes de ruído, pré-amplificadores, filtros, fibras, etc...
- O **configurador de simulação** é usado para construir o diagrama de blocos funcionais de acordo com a topologia desejada, utilizando os elementos disponíveis na biblioteca de modelos. Para propósitos de definição de topologia, um bloco é uma abstração que consiste de um nome que o identifica e de um modelo matemático que faz parte da biblioteca de modelos.

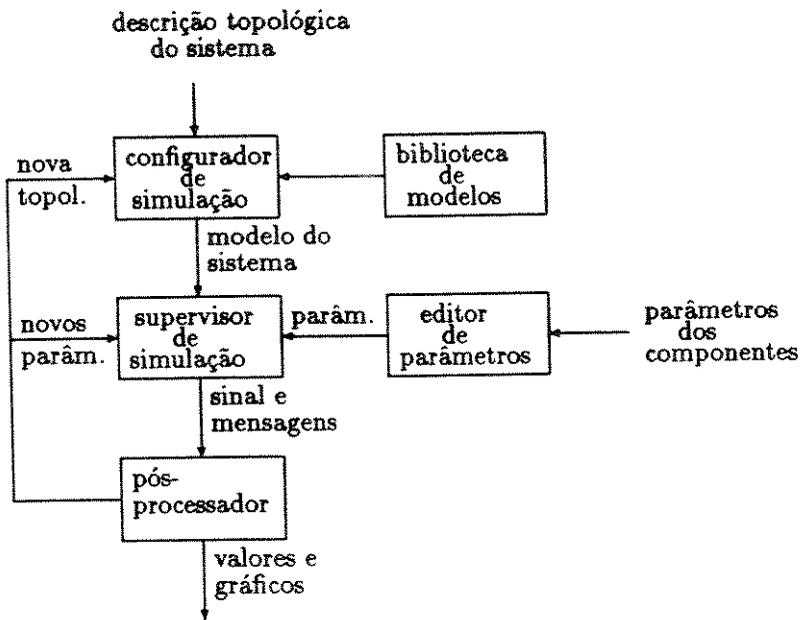


Figura 2.1: Diagrama funcional do software de simulação.

- O **gerenciador de simulação** é responsável por supervisionar a simulação e também por promover o interfaceamento entre as rotinas do software, gerando e armazenando amostras do sinal transmitido em vários pontos do sistema.
- O **pós-processador** é usado para analisar as amostras geradas a fim de obter informações sobre o desempenho do sistema. Se os objetivos não foram alcançados, então os valores dos parâmetros ou a topologia do sistema devem ser mudados e simulações repetidas até que um projeto adequado seja encontrado.

Dentro desta estrutura funcional aparecem outros componentes que, a despeito de sua condição auxiliar, desempenham funções importantes, como permitir a verificação e alteração dos parâmetros físicos dos componentes que são utilizados durante a simulação (*editor de parâmetros*), e traçar gráficos, facilitando a análise dos resultados.

2.2 Conceitos básicos sobre o PC-SIMFO

O PC-SIMFO - Simulador de Sistemas de Comunicação por Fibra Óptica é um software interativo, dirigido por menus, escrito com o objetivo de funcionar como uma ferramenta auxiliar no ensino e na pesquisa de sistemas de comunicação que utilizam fibra óptica como suporte de transmissão, podendo tanto ser usado na análise e projeto de sistemas como na análise e projeto de componentes optoeletrônicos isolados [9].

O ambiente PC-SIMFO fornece as seguintes ferramentas integradas: a construção da configuração do sistema a ser simulado através de blocos funcionais; a execução da simulação no domínio do tempo e da frequência; a análise do resultado da simulação, usando diferentes formas de processamento digital, a apresentação gráfica e a alteração de parâmetros objetivando uma otimização sistêmica.

Por empregar uma interface interativa não é necessário que o usuário tenha conhecimentos de programação, podendo gerar e processar dados, simular sistemas e calcular seu desempenho através de um conjunto de opções de menu, sem a necessidade de memorizar qualquer sintaxe específica.

O PC-SIMFO foi desenvolvido usando o compilador TURBOC versão 2.0 da Borland International. O programa consiste de diversas rotinas compiladas separadamente e posteriormente "linkadas", resultando no arquivo SIMFO.EXE. A linguagem C foi escolhida por sua portabilidade, por ser estruturada, permitir compilação separada de subrotinas utilizando alocação dinâmica de memória, estruturas sofisticadas de loops e operadores de baixo nível, além de possuir uma boa flexibilidade no controle de telas e gráficos. O programa pode ser executado em qualquer microcomputador PC compatível com sistema operacional DOS, um monitor CGA (Color Graphics Adapter) ou EGA (Enhanced Graphics Adapter). Para executar o programa, estando no DOS, basta digitar SIMFO. O sistema é carregado e a partir daí as opções são feitas pelo usuário através de menus.

2.2.1 Estrutura dos menus

O PC-SIMFO é construído como um conjunto de menus estruturados na forma de árvore. O menu principal contém os quatro maiores segmentos do sistema: *facilidades*, *componentes*, *simulação* e *análise*, além da opção de abandonar o sistema. Cada um destes segmentos apresenta diversos sub-menus que são mostrados de acordo com a seleção do usuário. A árvore completa de menus é apresentada na figura 2.2 e as principais opções são explicadas brevemente em seguida.

Facilidades

A opção *facilidades* apresenta algumas funções auxiliares, como: listar arquivos, traçar gráficos (que podem ou não ter sido gerados pelo sistema) e processar sinais através de FFT e FFT inversa. Dentro da opção de traçar gráficos o usuário pode também observar o diagrama de olho.

Componentes

A opção *componentes* é usada para a caracterização dos elementos que são utilizados na simulação. Através desta opção o usuário pode consultar e alterar as bases de dados que contêm as características físicas e geométricas de cada um dos componentes presente na biblioteca de modelos. Caso esta opção não seja utilizada a simulação é executada com valores pré-estabelecidos. Podem ser definidos parâmetros de cinco componentes, como mostrado na figura 2.3.

Simulação

A opção *simulação* pode ser executada para cada componente optoeletrônico isolado ou para todo o sistema. Em ambos os casos são utilizados os componentes presentes na biblioteca de modelos e os parâmetros existentes nas bases de dados, que representam as características físicas e geométricas dos elementos utilizados. Para a execução da simulação de um componente isolado o usuário deve selecionar uma seqüência de entrada, que pode ser do tipo

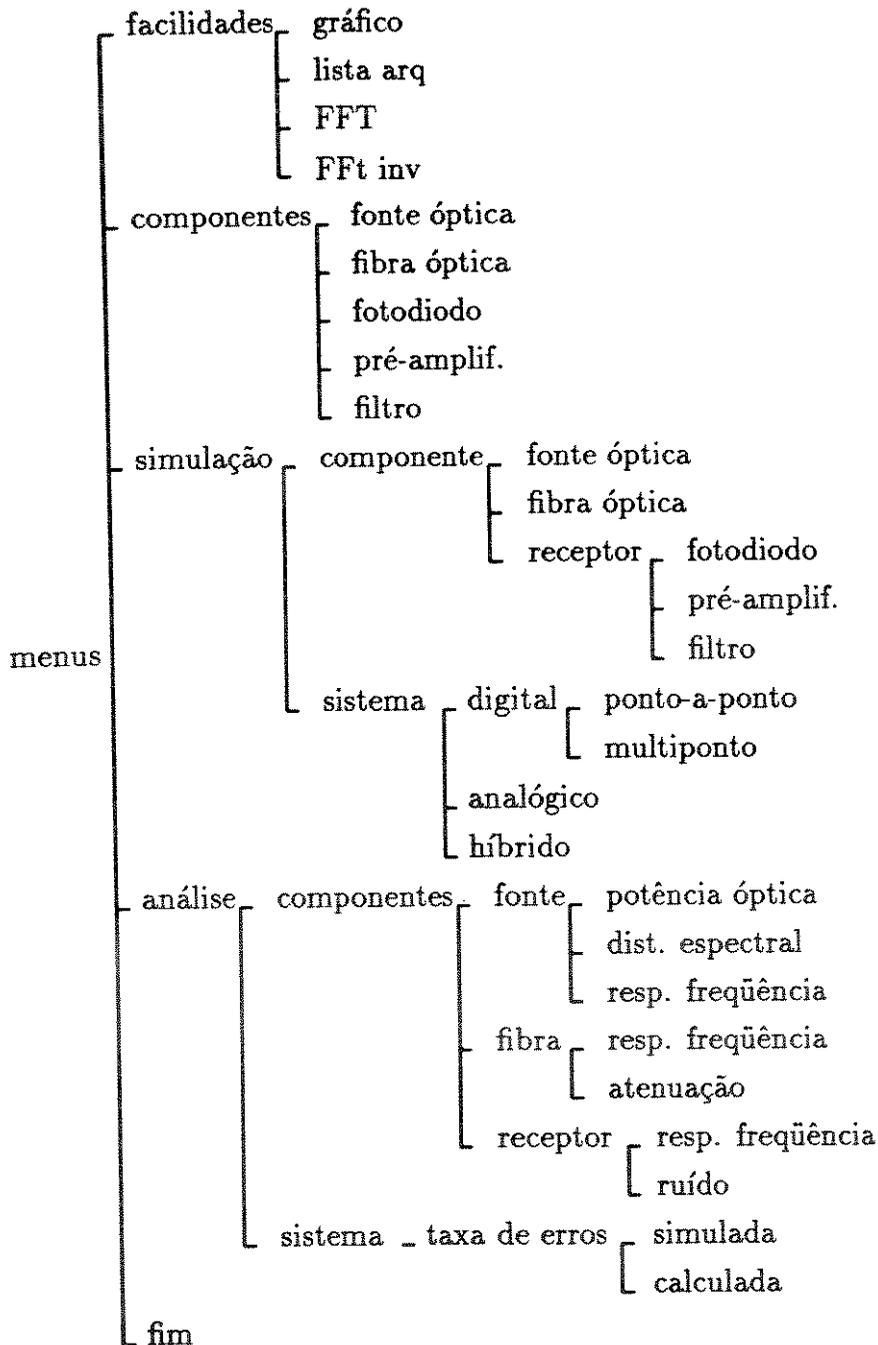


Figura 2.2: Estrutura dos menus

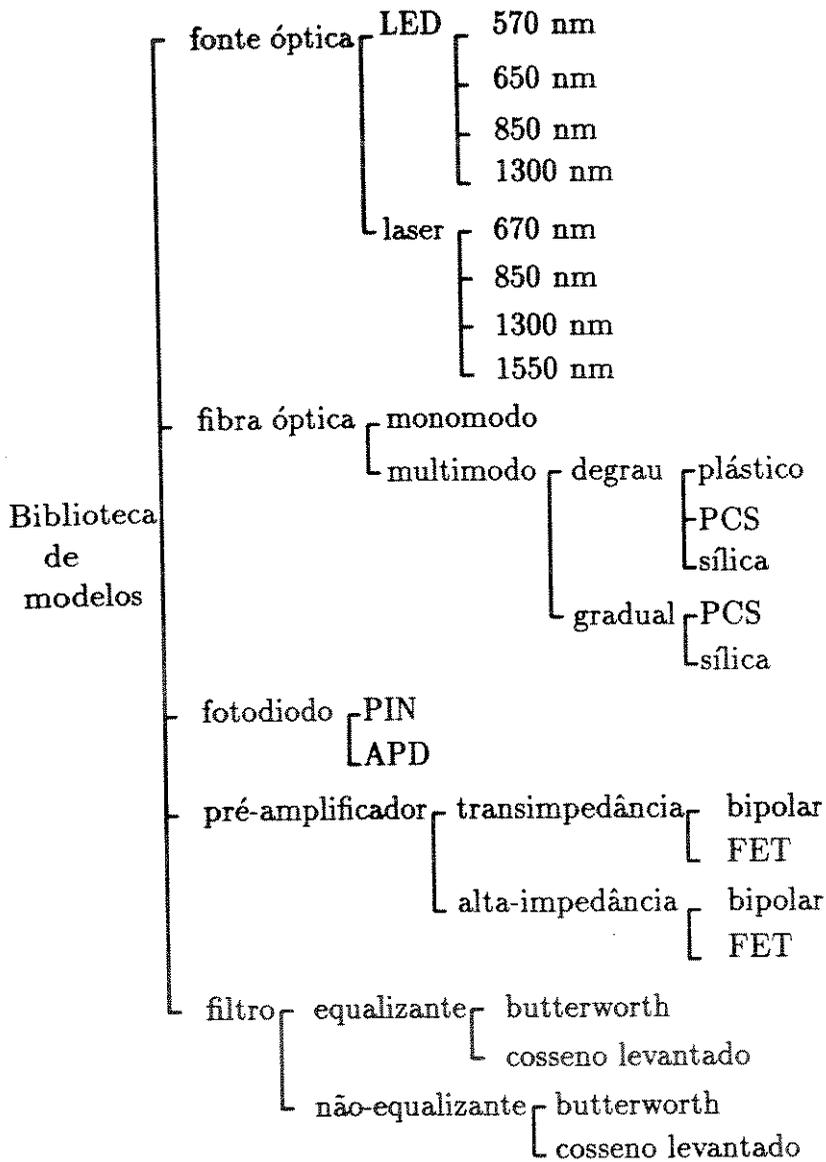


Figura 2.3: Componentes da biblioteca e modelos

aleatório, trem de pulsos ou pulso isolado. No caso da seqüência aleatória, deve selecionar também o número de amostras a serem processadas: 2048, 4096, 8192 ou 16384 ¹. Esta entrada, no domínio do tempo, será transformada para o domínio da freqüência e será multiplicada pela função de transferência do elemento sob análise, resultando em um sinal na saída. Este sinal pode ser observado no domínio do tempo ou da freqüência, de acordo com seleção do usuário. Após a execução da simulação pode-se ver o resultado gráfico ou simular um novo componente.

Na simulação de um sistema de comunicação pode-se optar entre sistema digital, analógico ou híbrido, numa configuração ponto-a-ponto ou multiponto, devendo-se selecionar cada um dos componentes do sistema presente na biblioteca de modelos. Para a simulação de um sistema digital ponto-a-ponto, por exemplo, deve-se selecionar o gerador, o codificador, a fonte de luz, a fibra óptica, o fotodiodo, o pré-amplificador e o filtro. No final da simulação pode-se observar os gráficos relativos à saída de cada um dos componentes.

Análise

Na opção *análise* estão disponíveis diversas opções de avaliação de desempenho, que variam de acordo com o componente sob análise e que correspondem a uma parte importante do projeto de sistemas de comunicação.

Na medida de desempenho de um sistema de comunicação digital, o critério mais significativo é a probabilidade de erro, que pode ser simulada, isto é, através da contagem da ocorrência de erros durante a simulação do sistema; ou calculada, através de uma equação que leva em conta a situação de pior caso.

Para a análise de componentes isolados temos: para a fonte óptica (LED ou laser), curvas de potência óptica, distribuição espectral e resposta em freqüência. Para a fibra, resposta em freqüência e curvas de atenuação e para o receptor a resposta em freqüência e ruído.

2.2.2 Metodologia de Simulação

A primeira das fases do processo de simulação consiste na definição dos parâmetros dos componentes do sistema. O usuário tem acesso às bases de dados que contêm as características físicas de cada um dos componentes do sistema sob análise. Dentro desta base de dados é possível consultar e alterar os valores de cada uma destas características.

Na segunda fase, que consiste na definição da configuração do sistema a ser simulado, são apresentados ao usuário os componentes disponíveis na biblioteca de modelos. A configuração do sistema é especificada a partir da seleção destes componentes. Podem ser selecionados para simulação componentes isolados ou um conjunto de componentes. O usuário tem a flexibilidade de trocar os parâmetros de projeto e mudar os componentes selecionados para

¹Salvo quando se quer analisar a probabilidade de erro com base nos sinais transmitidos e recebidos, o número de amostras é irrelevante, sendo preferível escolher o menor número, diminuindo assim o tempo de simulação. A título de exemplo, podemos verificar que, em um microcomputador do tipo PCxt, a simulação de um LED usando uma seqüência aleatória de 2048 amostras demora aproximadamente 135 segundos, assim divididos: 26s na geração dos dados, 29s na FFT dos dados gerados, 17s na função de transferência, 29s na resposta do sistema e 34s na FFT inversa, já computados os tempos de leitura/gravação, enquanto que a mesma simulação usando 8192 amostras demora 493 segundos, sendo 72s para geração, 111s para FFT, 67s para função de transferência, 114s para resposta do sistema e 129s para FFT inversa.

atingir os objetivos do sistema enquanto a opção não for confirmada. Depois que as opções são confirmadas é iniciada a simulação propriamente dita.

A simulação consiste das seguintes etapas: geração das amostras do sinal a ser transmitido; processamento das amostras do sinal no domínio do tempo e da frequência em cada um dos componentes do sistema; geração e armazenamento dos valores amostrados deste sinal nos vários pontos selecionados; e análise dos valores amostrados obtidos no final da simulação.

Durante a processo de simulação o usuário é informado do estado dos procedimentos de simulação e de quaisquer erros resultantes das entradas do usuário ou de acesso às bases de dados, que terminam o procedimento de simulação.

Na fase de análise o usuário tem como opções critérios de desempenho como taxa de erros, potência emitida em função da corrente gerada, resposta em frequência, distribuição espectral, etc. A estimativa da probabilidade de erro pode ser feita de duas formas: contagem dos eventos de erros ou através de um método analítico obtido para a situação de pior caso. Na contagem de eventos o software simula a passagem do sinal transmitido pelo sistema e a taxa de erros é avaliada comparando-se o sinal detectado na saída com o sinal transmitido. Taxas típicas de erros para sistemas por fibra óptica variam de 10^{-6} a 10^{-10} , dependendo da relação sinal/ruído no receptor.

A fase final permite mostrar graficamente e gerar cópia impressa do desempenho do sistema. Esta fase também opera no modo interativo, ou seja, o usuário informa ao sistema como quer a saída, nomes dos arquivos, tipos de variáveis, etc. O sistema oferece ao usuário a oportunidade de ver os resultados no domínio do tempo ou da frequência, além de apresentar diagrama de olho. Através dos gráficos o projetista pode também acompanhar a distorção do sinal em diferentes pontos do sistema, já que a caracterização mais direta de um sistema de comunicação por fibra óptica é através da potência do sinal óptico em pontos entre o transmissor e o receptor. Esta facilidade é importante no projeto do sistema, já que podem ser comparadas as diferenças entre os componentes e o usuário é assim capaz de determinar precisamente o benefício em relação ao custo de diferentes opções.

2.2.3 Programas e arquivos

O simulador PC-SIMFO é composto por diversas rotinas que executam tarefas específicas e que podem ser agrupadas de acordo com suas funções em quatro grandes subsistemas: interface de entrada, interface de saída, biblioteca de modelos e processamento e rotinas auxiliares, discutidos oportunamente. Estas subrotinas obtêm os dados necessários para sua execução de duas maneiras: alguns valores são passados como parâmetros, durante a chamada da função e outros são passados através de arquivos intermediários. O programa principal é responsável apenas por apresentar os menus de acordo com as seleções do usuário e ativar as subrotinas ordenadamente. Com esta abordagem, a maioria das pequenas modificações no software envolve mudanças em apenas uma ou poucas subrotinas. Além disso, a estruturação das rotinas facilita a localização e eliminação de erros de programação.

Fazem parte do sistema também os arquivos *global.h*, *funcoes.h* e *borda.h*, que, no momento da compilação, devem estar no mesmo subdiretório que estiverem os programas que formarão o SIMFO.EXE. O arquivo *global.h* contém as estruturas globais e as definições de todas as rotinas que compõem o sistema. O arquivo *borda.h* contém a função responsável por delimitar janelas para entrada de dados em pontos específicos da tela. O arquivo *funcoes.h*

contém algumas funções básicas que são utilizadas por várias rotinas do sistema: *le-inteiro*, *le-real*, *mens-erro* e *integral*, descritas na seção 2.3.

2.3 Processamento e rotinas auxiliares

Aqui são apresentadas as rotinas que realizam algum tipo de processamento sobre os dados utilizados e que servem como suporte na execução de outras rotinas do software.

FOUR

Calcula a transformada rápida de Fourier (FFT) de um conjunto de números complexos. Os valores estão em um arquivo cujo nome é passado como parâmetro para a rotina. O arquivo contém *nn* números complexos, onde *nn* deve ser uma potência de dois. O programa também calcula a transformada inversa, bastando para isso que se indique qual o cálculo desejado através de um parâmetro passado para a rotina. Ao final do cálculo os novos valores são armazenados em um arquivo para serem utilizados por outros programas.

O programa é chamado com dois parâmetros - *nome* e *sinal*. O parâmetro *nome* indica o arquivo de cujos dados será calculada a FFT ou inversa. O parâmetro *sinal* indica o tipo de operação desejada: se *sinal* for igual a 1 a operação escolhida é a FFT. Por outro lado, se *sinal* for igual a -1 é realizada a operação da inversa da FFT.

MULT

Faz a multiplicação de dois vetores de números complexos. Mais especificamente, multiplica os valores calculados da FFT que correspondem à corrente injetada (*sinal gerado*) com os valores correspondentes à função de transferência dos componentes optoeletrônicos, gerados em rotinas específicas. O resultado representa a transformada de Fourier da potência óptica emitida na saída do componente.

O programa é chamado com dois parâmetros - *nome1* e *nome2*, informando os nomes dos arquivos (complexos) que serão multiplicados.

LE-INTEIRO

Lê um valor numérico do teclado e verifica se é inteiro.

LE-REAL

Lê um valor numérico do teclado e verifica se é real.

MENS-ERRO

Responsável por apresentar ao usuário mensagens de erro que porventura ocorrerem durante a execução dos programas.

INTEGRAL

Calcula o valor da integral usando o método do trapézio a partir de um vetor de valores, passado como parâmetro.

TRANSCOM

Transforma um vetor de números reais em um vetor de complexos, através da inserção de zeros na parte imaginária de cada um dos números, preparando assim o vetor para ser utilizado na rotina *four*, para cálculo da FFT. O programa é chamado com um parâmetro - nome, que informa o arquivo que será transformado em complexo. O arquivo binário gerado é denominado **complexo.dat**.

ALT-LED

Permite a alteração dos parâmetros que serão utilizados no cálculo da função de transferência do LED. O programa é chamado com um parâmetro - opção, indicando o comprimento de onda, que pode ser 570 nm, 650 nm, 850 nm, 1300 nm e *outro*. Se a opção selecionada for *outro* é pedido o nome do arquivo que vai ser criado ou alterado. Caso contrário os dados são recuperados do arquivo **led.std**. Os parâmetros que são apresentados na tela e podem ou não ser alterados pelo usuário são: comprimento de onda de pico (nm), largura espectral (nm), tempo de recombinação radiativa (ns), tempo de recombinação não-radiativa (ns), eficiência de injeção, índice de refração do semicondutor e temperatura (K). Após a alteração dos parâmetros é criado um arquivo *ascii* contendo os valores. Se a opção escolhida tiver sido *outro* o arquivo criado terá o mesmo nome indicado pelo usuário, caso contrário será apresentado ao usuário o nome do arquivo gerado. Os valores típicos para cada comprimento de onda são preservados a despeito das alterações introduzidas pelo usuário.

ALT-LASER

Permite a alteração dos parâmetros do laser que serão utilizados no cálculo da sua função de transferência. O programa tem um parâmetro de entrada - opção, que indica o comprimento de onda de emissão do laser: 670 nm, 850 nm, 1300 nm e 1550 nm e *outro*. Se a opção selecionada for *outro* é pedido o nome do arquivo a ser criado ou alterado. Para qualquer outra opção são recuperados valores do arquivo **laser.std**, correspondendo aos parâmetros: comprimento de onda de pico (nm), corrente de limiar (mA), tempo de vida dos portadores minoritários (ns), tempo de vida dos fótons (ps), largura espectral (nm), tempo de recombinação radiativa (ns), tempo de recombinação não radiativa (ns), eficiência de injeção, índice de refração do semicondutor, fator de perdas, reflectância dos espelhos, dimensão (μm) e temperatura (K). Os novos valores são armazenados em um arquivo *ascii* cujo nome é informado ao usuário após a confirmação das alterações.

ALT-FIBRA

Através desta rotina são alterados os parâmetros que serão utilizados no cálculo da função de transferência da fibra. O programa tem um parâmetro de entrada - opção, que indica o tipo da fibra cujos valores serão alterados: monomodo, multimodo de grau de plástico, PCS ou sílica e multimodo gradual de PCS ou sílica e *outro*. Se a opção selecionada for *outro* é solicitado o nome do arquivo a ser criado ou alterado. Caso contrário, são recuperados do arquivo **fibra.std** os valores relativos aos parâmetros: produto distância x largura de banda (MHz), índice de refração do núcleo e índice de refração da casca. Após a alteração é informado ao usuário o nome do arquivo *ascii* que contém os novos valores.

ALT-FOTOD

Permite a alteração dos parâmetros utilizados no cálculo da conversão da potência óptica no fotodiodo. O parâmetro de entrada do programa - opção, indica o tipo de fotodiodo utilizado: PIN, APD ou outro. Caso a opção selecionada tenha sido outro é solicitado o nome do arquivo a ser criado ou alterado. Caso contrário, os valores são recuperados do arquivo *fotod.std* e correspondem a: capacitância (F), corrente de escuro não multiplicativa (A), corrente de escuro multiplicativa (A), eficiência quântica (0.3 a 0.9), taxa de ionização e resistência de polarização. Os novos valores são gravados em um arquivo *ascii* cujo nome é informado ao usuário após a confirmação das opções.

ALT-PREAMP

Através desta rotina podem ser alterados os valores dos parâmetros responsáveis pela amplificação do sinal e introdução do ruído térmico gerado no pré-amplificador. O parâmetro de entrada da rotina - opção, indica o tipo de pré-amplificador selecionado: transimpedância ou alta-impedância dos tipos bipolar e FET ou outro. Para cada um destes tipos podem-se alterar os valores da transcondutância do FET (S), resistência de entrada do transistor (Ω), resistência de contra-reação (Ω), capacitância de entrada do pré-amplificador (F), ganho de corrente do transistor e temperatura (K). Caso a opção escolhida tenha sido outro o usuário deve informar o nome do arquivo a ser criado ou alterado. Para qualquer outra opção estes parâmetros estão armazenados no arquivo *preamp.dat*. Após a confirmação das alterações é informado ao usuário o nome do arquivo *ascii* que contém os novos valores.

LE-PARAM

Permite a entrada, via teclado, de parâmetros necessários à execução da simulação. Os parâmetros pedidos dividem-se em dois grupos: o primeiro é formado por variáveis necessárias a qualquer simulação, como a taxa de transmissão de dados (Mbit/s) e o segundo corresponde a parâmetros específicos de um componente. Assim, as perguntas feitas ao usuário dependem dos componentes do sistema de comunicação selecionados. Em todos os casos, porém, se a opção de simulação for outro, isto é, o usuário quer simular um componente diferente dos oferecidos pelo software, é pedido o nome do arquivo onde estão gravados os valores típicos de determinados parâmetros do componente. A descrição destes parâmetros é feita nas rotinas *alt-led*, *alt-laser*, *alt-fibra*, *alt-foto* e *alt-prea* para os componentes LED, laser, fibra óptica, fotodiodo e pré-amplificador, respectivamente.

Na simulação de LED é pedida a corrente injetada (mA). No caso do laser é pedido o valor da corrente de pré-polarização como um percentual da corrente de limiar e a corrente injetada (mA). Para a simulação da fibra óptica é pedido o comprimento da fibra, em quilômetros. Na simulação do receptor é pedido o valor do ganho de avalanche para o fotodiodo APD, enquanto que para o fotodiodo PIN nenhuma pergunta específica é feita. É pedido também o ganho do pré-amplificador (10-100). Se o filtro for do tipo Butterworth é pedida a ordem do filtro (1 a 4). Caso o filtro seja do tipo Cosseno Levantado é pedido o fator de ocupação do intervalo (entre 0 e 1).

Quando a opção do usuário for a simulação do sistema completo são feitas perguntas relativas a todos os componentes selecionados.

Após ter recuperado todas as respostas do usuário, o programa gera um arquivo *ascii* denominado *valor.par* que será utilizado durante o processo de simulação.

SIM-SIST

Responsável por gerenciar a simulação de um sistema de comunicação formado por gerador, codificador, fonte óptica, fibra óptica, fotodiodo, pré-amplificador e filtro. A opção para cada um destes componentes é feita no programa *menusist.c* e passada como parâmetro para a rotina.

O programa ativa ordenadamente as diversas rotinas necessárias à simulação responsáveis por gerar o sinal de entrada, processar este sinal, levando-o ao domínio da frequência, gerar os valores correspondentes às funções de transferência da fonte, da fibra óptica, do pré-amplificador e do filtro selecionados e simular a passagem do sinal de entrada por cada um dos componentes do sistema.

Ao final da simulação o programa apresenta ao usuário os nomes dos arquivos gerados em diversos pontos, a abertura de olho em relação ao sinal na saída do sistema e a probabilidade de erro em função desta abertura.

SIM-COMP

Gerencia a simulação de um componente isolado de um sistema de comunicação. Este componente pode ser um gerador, um codificador, uma fonte óptica, uma fibra óptica ou um receptor, este último formado por um fotodiodo, um pré-amplificador e um filtro. A escolha do componente é feita no programa *menusist.c* e a opção é passada como parâmetro.

O programa é responsável por ativar as rotinas que executam os diversos passos da simulação: gerar valores representando o sinal de entrada, transportar este sinal para o domínio da frequência e simular sua passagem pelo componente escolhido.

Ao final é apresentado ao usuário o nome dos arquivos gerados que correspondem ao sinal de entrada, ao espectro deste sinal, à função de transferência do componente e à potência emitida. Além disso é apresentado também o valor da abertura de olho em relação ao sinal na saída do componente.

2.4 Conclusão

Neste capítulo foi apresentada a estrutura do simulador PC-SIMFO. Foram introduzidos os conceitos básicos sobre a utilização do software e a metodologia de simulação. As rotinas de processamento e as que servem como suporte na execução de outras rotinas foram descritas.

Referências Bibliográficas

- [1] M. A. Marsan et al., "Digital simulation of communication systems with TOPSIM III", *IEEE Journal on Selected Areas in Communications*, SAC-6 (1984).
- [2] D. G. Messerschmitt, "A tool for structured functional simulation", *IEEE Journal on Selected Areas in Communications*, vol 1, pp. 24-32 (1984).
- [3] D. G. Messerschmitt, "A transmission line modeling program written in C", *IEEE J. Select. Areas Commun.*, SAC-2, pp. 148-159 (1984).

-
- [4] J. W. Modestino e K. R. Matis, "Interactive simulation of digital communication systems", *IEEE Journal on Selected Areas in Communications*, **SAC-6** (1984).
 - [5] W. H. Tranter e C.R. Ryan, "Simulation of communication systems using personal computers", *IEEE Journal on Selected Areas in Communications*, **SAC-6**, pp. 13-23 (1988).
 - [6] S. V. Ahamed and V. B. Lawrence, "A PC based CAD environment for fiber optic simulation", in *Proc. Globecom 89*, pp. 147-153 (1989).
 - [7] Lobão P. M. S. e Moschim E., "PC-SIMFO: Simulation of optical fiber communication systems", aceito para apresentação no Bilcon'92, 1992.
 - [8] K. S. Shanmugam, "An update on software packages for simulation of communication systems (Links)", *IEEE Journal on Selected Areas in Communications*, **SAC-6**, pp. 5-12 (1988).
 - [9] Lobão P. M. S. e Moschim E., "PC-SIMFO: A Software for Analysis and Design of Optical Fiber Communication Systems", IV Canadian Conference on Electrical and Computer Engineering", 1991.

Capítulo 3

Interfaces do PC-SIMFO

Neste capítulo são apresentadas as interfaces de entrada e saída do simulador PC-SIMFO, representadas pelos menus de opções, os gráficos e as listagens de valores obtidos

3.1 Introdução

A principal preocupação dos softwares atuais é apresentar uma interface amigável com o usuário [1][2][3]. Assim, dentro do PC-SIMFO foi dada grande ênfase ao sistema de menus, que facilitam a comunicação do usuário com o sistema, diminuindo o risco de erros e tornando sua utilização agradável. Também com o objetivo de facilitar o trabalho do projetista de um sistema de comunicação, aumentando a sua compreensão do sistema sendo simulado e melhorando a visualização dos resultados obtidos, o PC-SIMFO oferece uma interface gráfica como opção de saída, além da possibilidade de visualização dos valores gerados e processados através da listagem de valores.

3.2 Interface de entrada

Quando o programa PC-SIMFO é iniciado, a tela de abertura, mostrada na Figura 3.1, é apresentada. Assim que qualquer tecla é pressionada, o programa apresenta o menu principal, no topo da tela, contendo cinco opções: facilidades, componentes, simulação, análise e fim. sendo que a primeira destas opções está no modo vídeo-reverso. Existem dois métodos para se fazer a seleção de qualquer menu: o primeiro é baseado no movimento da barra de vídeo-reverso através das setas apropriadas do teclado. Quando a barra estiver na opção desejada, pressiona-se <ENTER>. Um método alternativo (e mais rápido) de se fazer a seleção é pressionando-se a letra “chave” (primeira letra maiúscula da palavra) do item desejado. Por exemplo, para trazer o menu de simulação basta pressionar a letra “S”. Assim que uma opção é selecionada o programa entra no submenu correspondente. O usuário vai se deslocando através da árvore de menus até atingir o ponto desejado.

3.2.1 Rotinas

O programa *menusist* pode ser considerado como uma “rotina mestre”, já que coordena a ordem de funcionamento de todo o sistema, apresentando as opções ao usuário, recebendo sua

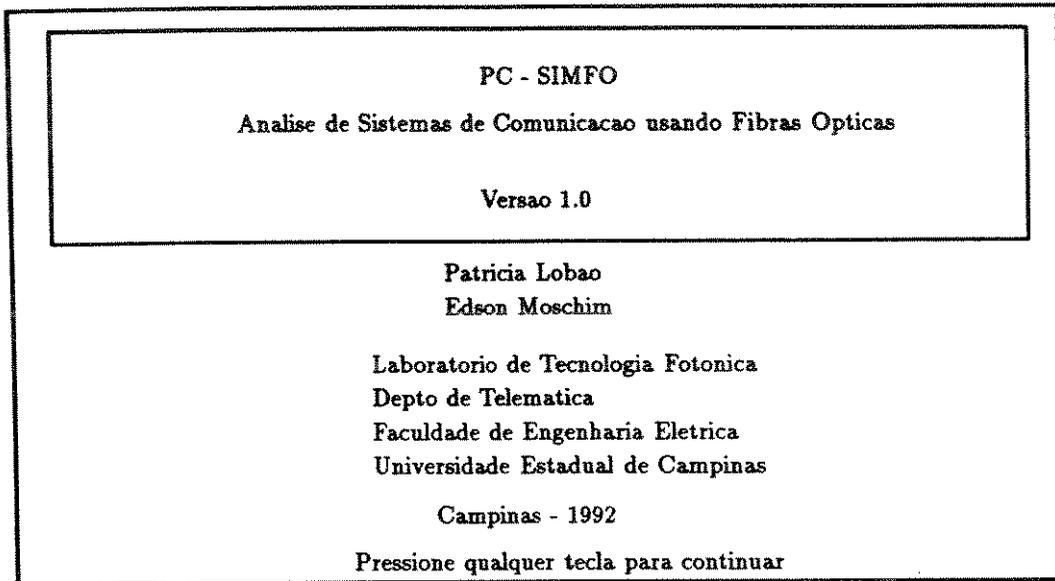


Figura 3.1: Tela inicial do PC-SIMFO

seleção e transferindo o controle para a rotina adequada. O *menusist* é composto de diversas subrotinas de tratamento de menus que são responsáveis, entre outras coisas, por salvar e restaurar o conteúdo do vídeo, criar, ativar e mostrar menus e criar bordas e molduras.

O módulo principal do programa é responsável por criar os menus, apresentar a tela de abertura e chamar a rotina de tratamento de menus. Os menus criados são guardados em um vetor denominado *frame*. A cada menu é atribuído um número de identificação serial correspondendo à sua posição dentro do vetor e que equivale a uma estrutura formada por nove campos. Os quatro primeiros correspondem à posição do menu na tela, linha e coluna, iniciais e finais. O quinto campo é um ponteiro para a posição onde será armazenada a parte do vídeo que irá ser coberta pelo menu. Em seguida aparece um campo que indica o tipo de borda do menu (simples, dupla ou sem borda). Os próximos dois campos informam as letras chave e o número de opções do menu. O último campo serve como uma indicação de estado, informando se o menu está ou não ativo no momento. A estrutura do vetor é definida como se segue:

```
struct menu-frame {
int inix,
    fimx,
    iniy,
    fimy;
unsigned char *p;
char **menu,
    *chaves;
int borda,
    cont,
    ativo;
} frame[MAX-FRAME];
```

O campo *p* representa um ponteiro que determina o endereço da memória de vídeo baseado no modo corrente do vídeo (CGA, EGA, etc). Sabendo o endereço do buffer de vídeo, a tarefa de salvar e restaurar a tela se torna simples: para salvar basta alocar um espaço de memória do tamanho apropriado e mover os bytes que aparecem no vídeo para este espaço e para restaurar a parte da tela que foi coberta pelo menu é necessário apenas mover os dados que estão armazenados para o buffer de vídeo.

As *chaves* são letras pré-determinadas (normalmente a primeira letra da palavra ou, na impossibilidade desta, uma letra qualquer que identifique a opção) que, quando pressionadas, fazem com que a opção correspondente seja automaticamente executada.

Um menu aparece quando é requisitado e desaparece quando não é mais necessário, restaurando a área que foi coberta enquanto ele estava presente. Para que a restauração seja feita é necessário que a imagem seja salva antes que o menu seja apresentado. Assim, para fazer o menu desaparecer a imagem salva é simplesmente copiada no buffer de vídeo. Para realizar estas operações é necessário um controle direto sobre o vídeo (monocromático, EGA e CGA são os monitores mais comuns), que pode ter diversos modos de operação, incluindo o de texto ou gráfico, de 40 ou 80 colunas. As rotinas de menu desenvolvidas são projetadas para serem usadas no modo texto de 80 colunas, o mais comum para as aplicações de propósito geral.

Os caracteres apresentados na tela são mantidos em uma RAM reservada ao vídeo, localizada nos endereços B0000000H, para vídeo monocromático, de dimensão 80x25, no modo texto, e B8000000H, para EGA e CGA de 80x25, também no modo texto. Este endereço corresponde à posição [0,0], isto é, o canto superior esquerdo da tela. Cada caracter apresentado requer dois bytes de memória de vídeo. O primeiro byte contém o caracter propriamente dito e o segundo seus atributos de tela (vídeo normal, vídeo-reverso, etc). Assim, para cada menu a ser apresentado é colocado no endereço apropriado da memória de vídeo cada caracter (com seu atributo) que irá aparecer na tela.

Para executar todas as funções necessárias para o tratamento de menus o programa *menusist* se utiliza de diversas subrotinas que são descritas a seguir:

ativa-menu

É responsável por ativar/desativar os menus de acordo com a opção do usuário, além de chamar as rotinas responsáveis pela execução das tarefas relacionadas à seleção feita pelo usuário. Para o tratamento dos menus se utiliza das subrotinas *exibe-menu* e *rec-video*. A partir das opções selecionadas pelo usuário pode também sair do sistema ou chamar as rotinas de gráfico, as referentes à alteração de parâmetros dos componentes fotônicos, as de simulação de componentes ou sistema, bem como as de análise.

exibe-menu

Exibe um menu e retorna a opção escolhida pelo usuário. Ao executar suas tarefas segue alguns passos, verificando se o menu já está ativo, mostrando o menu e lendo a resposta do usuário. Para isso utiliza-se das subrotinas *le-resp*, *salva-video*, *cria-borda* e *mostra-menu*.

cria-menu

É a primeira rotina a ser chamada (ainda no módulo principal), sendo responsável por construir o quadro do menu dentro do vetor de menus, analisando o tamanho e

número das opções, posição na tela, letras chave, tipo de borda, alocando memória dinamicamente para suportar o "buffer" de vídeo e detectando erros na quantidade de menus e dimensões do menu sendo criado.

cria-janela

Cria janelas para entrada de dados via teclado. É usada em momentos específicos, como por exemplo, a seleção de opções para o gráfico e a alteração das características físicas e geométricas dos componentes optoeletrônicos.

mostra-menu

É responsável pela exibição das opções do menu no local determinado pelo vetor de menus criado na rotina *cria-menu*, utilizando-se para isso do número do menu a ser exibido.

cria-borda

É responsável pelo desenho das bordas em torno dos menus e janelas.

le-resp

Permite a entrada da opção do usuário, fazendo o movimento da barra de vídeo-reverso de acordo com as teclas digitadas e recuperando a seleção feita pelo usuário. Esta seleção pode ser indicada de duas formas: pressionando-se a tecla <ENTER> quando o vídeo-reverso estiver na posição desejada ou digitando-se uma das letras chave.

highlight e normal

Coloca uma opção em vídeo-reverso e volta para vídeo normal, respectivamente.

explica-opcao e apaga-opcao

Mostra e apaga, respectivamente, uma breve explicação na última linha da tela sobre a opção do menu que está em vídeo-reverso.

moldura

Apresenta uma moldura (ou retângulo) em qualquer parte da tela, independente de ser ou não correspondente a um menu. A moldura pode ter borda simples ou dupla e é usada para desenhar os blocos que representam os componentes de um sistema de comunicação.

escreve-string e escreve-caracter

Rotinas auxiliares que apresentam mensagens ou caracteres na tela, respectivamente, com um atributo específico (vídeo-reverso ou normal).

escreve-vertical

Apresenta na tela uma cadeia de caracteres na vertical com um atributo específico (vídeo-reverso ou normal).

salva-vídeo

Salva a parte do vídeo aonde vai ser mostrado o próximo menu, no local indicado pelo campo "p" da estrutura do vetor de menus e limpa esta parte para a exibição do menu.

rec-vídeo

Recupera a parte do vídeo aonde foi exibido um menu recentemente desativado.

inic-video e video-mode

Verificam o tipo de vídeo disponível e estabelecem o endereço apropriada da memória RAM de vídeo para a correta exibição dos menus.

cursor

Responsável por ligar e desligar o cursor, com o objetivo de permitir uma melhor visualização dos menus e das janelas durante a utilização do software.

des-sistema

Apresenta no vídeo o desenho do diagrama de blocos funcionais, mostrado na Figura 3.2, que representa um sistema de comunicação por fibras ópticas, recuperando as opções do usuário para cada um dos componentes, dentre as opções:

- **gerador:** randômico, trem de pulsos e pulso;
- **codificador:** NRZ, RZ e Manchester;
- **fonte:** LED (570 nm, 650 nm, 850 nm e 1300nm) e laser (670 nm, 850 nm, 1300 nm e 1500 nm);
- **fibra:** monomodo, multimodo índice degrau (plástico, PCS e sílica) e multimodo índice gradual (PCS e sílica);
- **fotodiodo:** PIN, APD;
- **pré-amplificador:** transimpedância (bipolar e FET) e alta-impedância (bipolar e FET);
- **filtro:** butterworth (equalizante e não equalizante) e cosseno levantado (equalizante e não equalizante);

É perguntado também se o usuário deseja observar os resultados no domínio do tempo ou da frequência. Após a confirmação das opções são chamadas as rotinas responsáveis pela simulação de cada um dos componentes selecionados. Ao final da execução são informados os nomes dos arquivos gerados que correspondem ao sinal na saída de cada um dos componentes, como no exemplo da Figura 3.3.

des-componente

Apresenta no vídeo o desenho de um bloco representando o componente, sua entrada e sua saída, como mostrado na Figura 3.4. O usuário escolhe o tipo de entrada e o tipo de saída. As opções são:

- **entrada:** randômico, trem de pulsos e pulso;
- **saída:** resposta no tempo ou resposta em frequência.

Após a confirmação das opções é gerado o sinal de entrada de acordo com a seleção do usuário. O arquivo com os dados de entrada é convertido de real para complexo, é calculada a transformada de Fourier deste sinal (tem-se então o sinal de entrada no

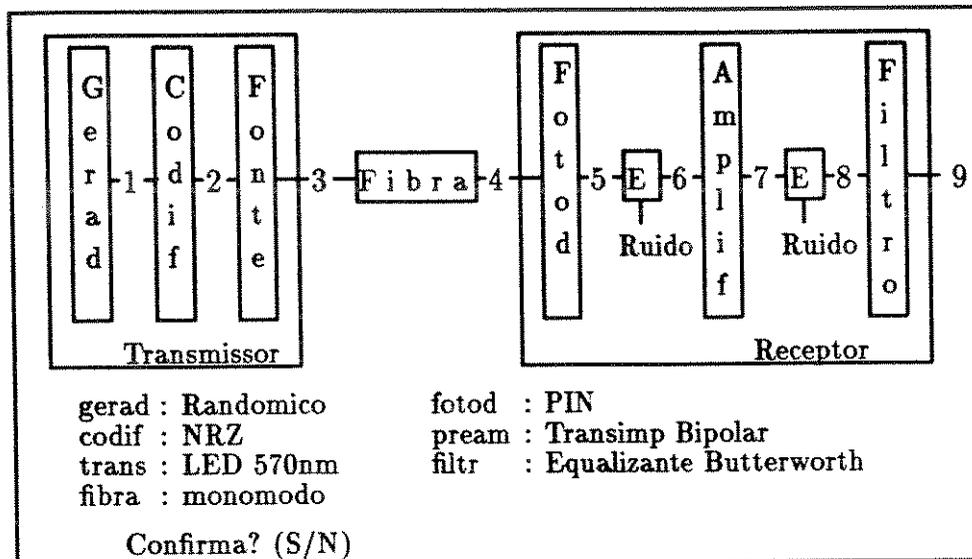


Figura 3.2: Tela para seleção de componentes

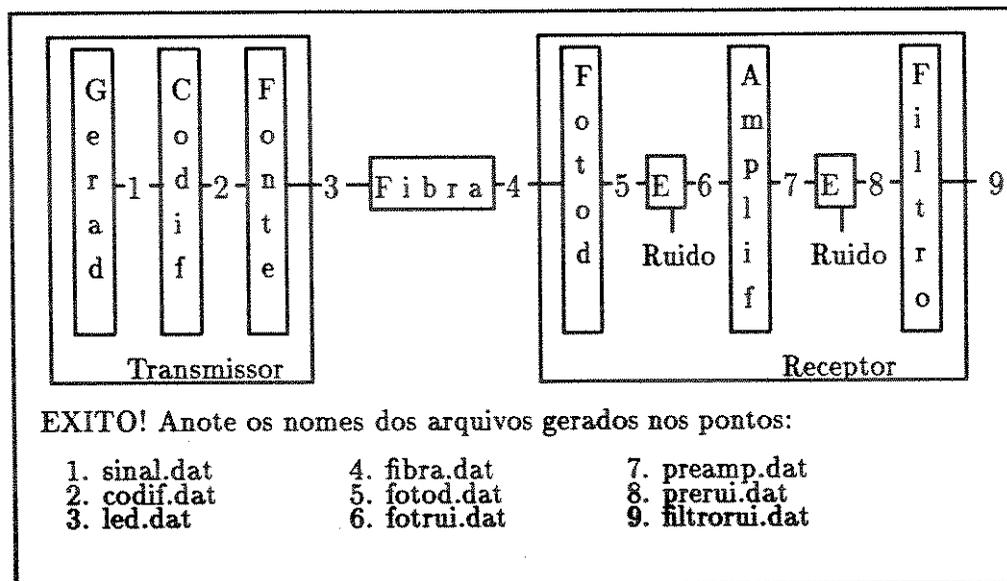


Figura 3.3: Tela ao final da simulação de um sistema de comunicação

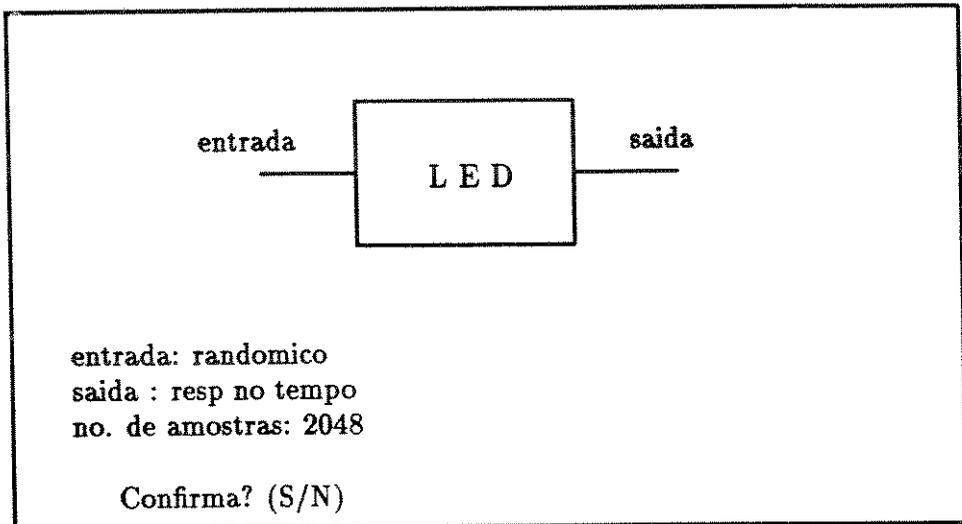


Figura 3.4: Opções para simulação de um componente isolado

domínio da frequência). Em seguida é chamada a rotina específica do componente escolhido que calcula a função de transferência deste. É feita a multiplicação do sinal de entrada em frequência com a função de transferência do componente, resultando na transformada de Fourier da potência óptica emitida (no domínio da frequência). Se o usuário tiver optado por uma saída no domínio do tempo é então calculada a transformada inversa da potência emitida. Caso a opção tenha sido por uma saída de diagrama de olho este é calculado fazendo-se a superposição de segmentos da forma de onda do sinal no intervalo de tempo de um bit. Em seguida são apresentados ao usuário os nomes dos arquivos criados, como no exemplo da figura 3.5. que devem ser anotados para posterior geração dos gráficos.

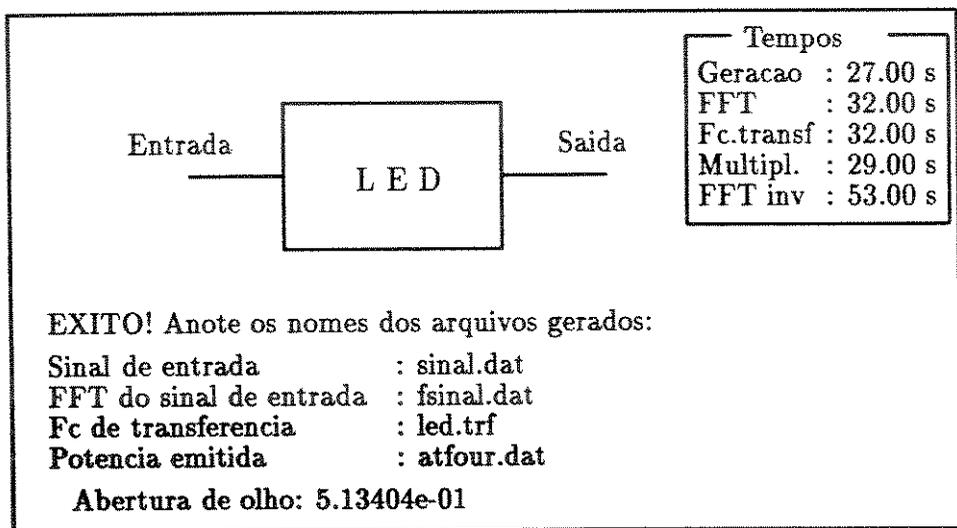


Figura 3.5: Tela ao final da simulação de um componente isolado

3.3 Interface de saída

O PC-SIMFO oferece ao usuário três opções de saída: gráficos, diagrama de olho e listagem dos valores.

3.3.1 Gráficos

Para traçar os gráficos selecionados pelo usuário, as rotinas se utilizam de dois arquivos de opções: *var.grf* e *opcao.grf*, além dos arquivos contendo os valores dos eixos x e y a serem colocados no gráfico.

O arquivo *var.grf* se torna disponível quando é selecionada a opção *variáveis* do submenu *gráfico*; neste momento o usuário pode consultar e/ou alterar os nomes dos arquivos cujos valores serão utilizados para traçar o gráfico. São permitidos nomes de três arquivos: um para o eixo x e dois para o eixo y, sendo que o único campo de preenchimento obrigatório é o da primeira variável do eixo y. Se não for indicado arquivo para o eixo x são gerados valores de 0 até o número de amostras da variável do eixo y que são colocadas em x. Se for escolhida uma segunda variável para y a curva correspondente do gráfico será tracejada, para diferenciar da linha contínua da primeira variável.

O arquivo *opção.grf* pode ser consultado e/ou alterado a partir da seleção da opção *opções* do submenu *gráfico*. Os valores contidos neste arquivo são usados para determinar características do gráfico a ser traçado:

- se as curvas são ou não digitais, isto é, o valor da variável do eixo y se mantém constante de x_i até x_{i+1} ;
- se os dados que serão colocados no gráfico são reais ou complexos (no caso de números complexos o valor a ser utilizado corresponde ao módulo dos números, ou seja, valor $= \sqrt{\text{real}^2 + \text{imaginaria}^2}$);
- se o gráfico é do tipo discreto ou contínuo;
- se os valores do eixo y são normalizados ou não;
- se as buscas de máximos e mínimos de x e y serão automáticas ou não (no caso de busca não automática devem ainda ser fornecidos os valores iniciais e finais de cada uma das variáveis);
- o título a ser colocado no gráfico (em duas linhas) e
- nomes dos eixos x e y.

As opções presentes nos arquivos *var.grf* e *opcao.grf* podem ser alteradas a qualquer momento pelo usuário, bastando para isso que ele siga a estrutura dos menus até a opção desejada.

Depois que as opções estão selecionadas e antes que um gráfico seja traçado, o programa reconhece o modo de vídeo sendo utilizado. Diversos modos de vídeo estão disponíveis no PC e os mais comuns estão mostrados na tabela 3.1. Em todos os modos o canto superior esquerdo é [0,0]. No programa desenvolvido foi utilizada uma função do TURBOC que

detecta automaticamente o modo de vídeo sendo usado e procede à inicialização, carregando o “driver” gráfico apropriado do disco, colocando assim o sistema no modo gráfico que ofereça maior resolução.

MODO	TIPO	DIMENSAO	ADAPTADORES
4	gráfico,4 cores	320x200	CGA/EGA
5	gráfico,4 tons cinza	320x200	CGA/EGA
6	gráfico,p/b	640x200	CGA/EGA

Tabela 3.1: Modos de tela para PC

O gráfico é traçado com base em linhas que ligam um ponto (x_i, y_i) ao seu sucessor (x_{i+1}, y_{i+1}) , de acordo com a ordem estabelecida no arquivo de valores dos eixos x e y, no caso do gráfico ser do tipo contínuo. Para gráfico discreto a linha é traçada de (x_i, y_i) até (x_i, y_{i+1}) . Depois que o gráfico é traçado, se o usuário desejar imprimí-lo deve pressionar a tecla **F1**, de acordo com a indicação que aparece no canto inferior esquerdo da tela.

O programa de gráficos é chamado pelo programa de menus com um parâmetro - opção, que informa se o usuário quer alterar os valores das variáveis, alterar as opções ou traçar o gráfico. Dependendo desta opção, são chamadas as subrotinas que irão verificar o “hardware” para gráfico, selecionar parâmetros e traçar as curvas. Estas subrotinas são apresentadas a seguir.

inic-graf

Detecta o tipo de vídeo sendo usado e o inicializa para a utilização no modo gráfico, indicando a ocorrência de erros, se for o caso.

buscamax e buscamin

Recuperam o maior e menor valores de y a serem plotados, respectivamente. Só são utilizados se a opção do usuário for de *busca automática de máximos*.

des-borda

Define os limites do gráfico.

curva

É responsável por traçar a curva do gráfico, respeitando as opções feitas pelo usuário. A primeira curva é traçada com uma linha sólida e a segunda, quando for o caso, com uma linha tracejada. A cada ponto traçado é verificado se a curva não excedeu os limites do gráfico delimitados pela rotina *des-borda*. Caso isto ocorra o traçado da curva é interrompido e é apresentada uma mensagem ao usuário.

rec-valor

Recupera os valores que serão utilizados para traçar a curva, dentro dos limites definidos pelo usuário, colocando-os em um vetor e retornando o número total de pontos recuperados. Se o arquivo for de números complexos o valor a ser utilizado para traçar o gráfico é colocado no vetor que corresponde ao módulo dos números. Se o arquivo for composto por números reais eles são utilizados diretamente.

ver-ex

Verifica a existência dos arquivos cujos nomes estão no arquivo *var.grf* e que representam os valores dos gráficos a serem gerados. Estes valores são recuperados através da chamada da subrotina *rec-valor*. No caso do eixo x, se não for indicado nenhum arquivo de valores são gerados números de 0 a n sequencialmente para traçar o gráfico, onde n é o número de pontos do eixo y.

prepara

Faz os ajustes finais antes que o gráfico seja traçado, como definir escala, que corresponde ao número de vezes que o gráfico é aumentado, dependendo do número de pontos a serem amostrados; desenhar os limites do gráfico; normalizar os valores do eixo y, escrever valores nos eixos x e y; título do gráfico e nomes dos eixos. As informações para a preparação do gráfico são obtidas do arquivo *opcao.grf*.

rec-opcao

Recupera os nomes das variáveis a serem utilizadas para traçar o gráfico e as opções que definem o tipo de gráfico a ser traçado e que serão utilizadas em outras subrotinas.

alt-variaveis

Apresenta ao usuário os nomes das variáveis dos eixos x e y e permite que sejam alteradas. É utilizada se a opção do usuário através dos menus seja a de alterar variáveis, ou seja, não é utilizada no momento de traçar o gráfico.

alt-opcoes

Apresenta ao usuário todas as opções que serão utilizadas para traçar o gráfico e permite que sejam alteradas. As opções disponíveis são: curva 1 e 2 digitais ou não (a opção é feita para cada uma das curvas), tipo de dados dos arquivos 1 e 2 (reais ou complexos), gráfico discreto ou contínuo, gráfico normalizado ou não, busca automática de máximos e mínimos e, se busca não for automática, valores máximos e mínimos, título do gráfico e nomes dos eixos x e y. Também não é utilizada no momento de traçar o gráfico.

le-string

Gerencia a entrada de dados do tipo string como, por exemplo, nomes dos arquivos, títulos dos gráficos e nomes dos eixos.

3.3.2 Diagrama de olho

O diagrama de olho é traçado a partir da superposição de segmentos da forma de onda do sinal no intervalo de tempo de um bit. Para uma melhor visualização, o diagrama de olho é apresentado segmentado em períodos correspondentes a três bits. A rotina responsável por traçar o diagrama de olho, denominada *diag-olho.c*, não tem parâmetros de entrada. O nome do arquivo cujos valores serão utilizados para traçar o diagrama de olho é solicitado ao usuário no momento da execução. Estes valores são recuperados e armazenados temporariamente em um vetor. No caso de números complexos é armazenado o valor do seu módulo.

Depois que o diagrama é apresentado, se o usuário desejar imprimí-lo, deve pressionar a tecla F1, de acordo com a indicação que aparece no canto inferior esquerdo da tela.

3.3.3 Listagem de valores

Outra opção de saída é a listagem dos valores gerados pelo software no decorrer da simulação, implementada pela rotina *lista-ar*, que não possui parâmetros de entrada. Esta rotina lista valores de arquivos gerados ou não pelo software, desde que estejam no formato binário. O usuário deve informar o nome do arquivo a ser listado, se deseja impressão total ou parcial, caso em que serão pedidos os valores iniciais e finais a serem mostrados, e se deseja ou não saída na impressora. Além disso deve informar se o arquivo é composto de números reais ou complexos. No caso de arquivo de complexos será listado o valor correspondente ao módulo dos valores. Quando a saída é direcionada para o vídeo o usuário pode observar todos os valores do arquivo contidos no intervalo selecionado através das teclas ↑, ↓, PgDn e PgUp, que permitem movimentação de retrocesso e avanço de uma linha e retrocesso e avanço de uma página, respectivamente. Ao pressionar a tecla ESC o usuário interrompe a listagem do arquivo e retorna ao menu principal.

3.4 Conclusão

Neste capítulo foram apresentadas as interfaces de entrada e saída do simulador PC-SIMFO. Estas interfaces com o usuário facilitam a utilização do software, tornando mais rápida a seleção de opções e melhorando a visualização dos resultados.

Referências Bibliográficas

- [1] Schildt H., *Linguagem C - Guia Prático e Interativo*, Mc Graw Hill, 1989.
- [2] Porter K. "A Graphics Toolbox for Turbo C", *Dr. Dobb's Journal*, nov/dez, 1989.
- [3] O. Alkin, *PC-DSP - 5^{1/4} IBM Version*, Prentice-Hall, 1990.

Capítulo 4

Modelamento dos componentes

Neste capítulo são apresentados os modelos matemáticos que caracterizam a relação de entrada/saída de cada um dos elementos que compõe os blocos funcionais do sistema discutido no capítulo 1.

4.1 Introdução

Para a avaliação do desempenho sistêmico, as relações funcionais de entrada e saída dos componentes do sistema devem ser modeladas por equações que mais se aproximam da verificação experimental. Uma vez que os modelos são especificados, a configuração do sistema é descrita pela interconexão destes componentes. A confiabilidade da simulação depende da qualidade da informação obtida por estes modelos. A seguir, apresentaremos os modelos que foram adotados na confecção do simulador [1][2][3].

4.2 Modelamento do transmissor

O transmissor óptico é constituído por um diodo eletroluminescente ou por um diodo laser e por seu circuito eletrônico de comando. A potência óptica emitida é modulada diretamente pela corrente injetada. Para a caracterização do transmissor, uma função de transferência pode ser escrita considerando somente os parâmetros da fonte de luz ¹. Neste caso, a relação entre a potência óptica emitida e a corrente injetada, no domínio do tempo, é dada por ²:

$$p_e(t) = i_d(t) * h_T(t) \quad (4.1)$$

onde:

$p_e(t)$ = potência óptica emitida (W)

$i_d(t)$ = corrente injetada no diodo (A)

$h_T(t)$ = resposta impulsiva do diodo (W/A)

¹Considera-se que o tempo de resposta do circuito de comando é mais rápido que o da fonte de luz. Esta suposição é válida para a aplicação do nosso simulador.

²O modelo de sinal que caracteriza a corrente injetada em função da do sinal na saída do gerador é apresentado no apêndice A.

* = operador de convolução

No domínio da frequência temos:

$$P_e(f) = I_d(f) \times H_T(f) \quad (4.2)$$

onde:

$P_e(f)$ = transformada de Fourier da potência óptica (W)

$I_d(f)$ = transformada de Fourier da corrente injetada (A)

$H_T(f)$ = função de transferência do diodo (W/A)

\times = operador de produto

A função de transferência do diodo, $H_T(f)$, pode ser decomposta em duas partes, de acordo com o modelo apresentado na figura 4.1:

$$H_T(f) = H_T(0)H_T^*(f) \quad (4.3)$$

onde:

$H_T(0)$ = eficiência quântica da fonte de luz (W/A)

$H_T^*(f)$ = resposta em frequência normalizada da fonte de luz

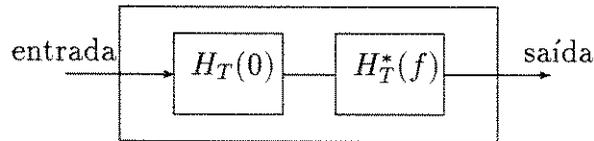


Figura 4.1: Modelo geral para fonte óptica

A eficiência quântica e a resposta em frequência, dependem do tipo de fonte de luz e são discutidas nas seções que se seguem.

4.2.1 Diodos Emissores de Luz

No caso dos LEDs a eficiência quântica é dada por [4]:

$$H_T(0) = \left(\frac{hc}{\lambda q}\right) \eta_{int} \eta_{ext} \eta_{inj} \quad (4.4)$$

onde:

η_{ext} = eficiência quântica externa

η_{inj} = eficiência de injeção de corrente

η_{int} = eficiência quântica interna

h = constante de Planck (6.6256×10^{-34} J.s)

c = velocidade da luz no vácuo (2.99793×10^8 m/s)

λ = comprimento de onda (m)

q = carga eletrônica (1.60218×10^{-19} C)

A eficiência quântica interna é dada por:

$$\eta_{int} = \frac{\tau_{nr}}{\tau_r + \tau_{nr}} \quad (4.5)$$

onde:

τ_{nr} = tempo de recombinação não-radiativa (s)

τ_r = tempo de recombinação radiativa (s)

A eficiência quântica externa é dada por [5]:

$$\eta_{ext} = [1 - (\frac{n_s - n_m}{n_s + n_m})^2][1 - \cos(\frac{n_m}{n_s})] \quad (4.6)$$

onde:

n_s = índice de refração do semiconductor

n_m = índice de refração do meio (ar = 1)

A resposta em frequência normalizada é dada por [6]:

$$H_T^*(f) = \frac{1}{1 + j(\frac{f}{f_c})}$$

$$f_c = \frac{1}{2\pi\tau_r} \quad (4.7)$$

onde:

f_c = frequência de corte (3dB óptico)

Outro parâmetro importante para o cálculo da análise de desempenho é a distribuição espectral, dada por [7]:

$$S(\lambda) = \frac{H_T(0)I_d}{\sigma_s\sqrt{2\pi}} \exp\left(\frac{-(\lambda - \lambda_p)^2}{2\sigma_s^2}\right) \quad (4.8)$$

onde:

$S(\lambda)$ = distribuição espectral de potência (W/m)

σ_s = largura espectral rms do LED (m)

λ_p = comprimento de onda de pico do LED (m)

4.2.2 Diodos laser

No caso dos diodos laser, o comportamento dinâmico depende do tipo do dispositivo. No nosso caso consideramos somente diodos laser multimodo. Para a eficiência quântica temos duas funções: uma abaixo da corrente de limiar, caracterizada por uma emissão espontânea e outra acima da corrente de limiar, caracterizada por uma emissão estimulada. Assim, na região de emissão estimulada, ou seja, $I_d > I_{th}$, temos [8]:

$$H_T(0) = \left(\frac{hc}{\lambda q}\right) \eta_{int} \eta_{ext} \left[\frac{I_d - I_{th}}{I_d}\right] \quad (4.9)$$

onde:

η_{int} = eficiência quântica interna do laser

η_{ext} = eficiência quântica externa do laser

I_d = corrente injetada (A)

I_{th} = corrente de limiar (A)

A eficiência quântica interna é dada por:

$$\eta_{int} = \frac{\tau_{nr}}{\tau_r + \tau_{nr}} \quad (4.10)$$

A eficiência quântica externa é dada por:

$$\eta_{ext} = \frac{\ln\left(\frac{1}{R_1}\right)}{\gamma l + \ln\left(\frac{1}{R_1}\right)} \quad (4.11)$$

onde:

R_1 = reflectância dos espelhos

γ = coeficiente de perdas (m^{-1})

l = dimensão do laser (m)

Para $I_d \leq I_{th}$, o laser está na região de emissão espontânea e tem comportamento similar ao do LED. Assim, a eficiência quântica é dada por:

$$H_T(0) = \left(\frac{hc}{\lambda q}\right) \eta_{int} \eta_{ext} \eta_{inj} \quad (4.12)$$

A resposta em frequência normalizada do laser é dada por [9]:

$$\begin{aligned} H_T^*(f) &= \frac{f_0^2}{f_0^2 - 4\pi^2 f^2 + j\beta 2\pi f} \\ f_0^2 &= \frac{(I_0 - I_{th})}{\tau_{sp}\tau_{ph}I_{th}} \\ \beta &= \frac{I_0}{\tau_{sp}I_{th}} \end{aligned} \quad (4.13)$$

onde:

I_{th} = corrente de limiar (A)

I_0 = corrente de pré-polarização (A)

τ_{sp} = tempo de recombinação dos portadores (s)

τ_{ph} = tempo de vida dos fótons (s)

β = frequência de amortecimento (Hz)

f_0 = frequência de ressonância (Hz)

Para o laser, o modelo da distribuição espectral é dado por [11]:

$$\begin{aligned} S(\lambda) &= \frac{H_T(0)I_d}{\sigma_s\sqrt{2\pi}} \exp\left(\frac{-(\lambda - \lambda_p)^2}{2\sigma_s^2}\right) \times \\ &\quad \sum_{k=-n}^n \frac{1}{\sigma'_s\sqrt{2\pi}} \exp\left(\frac{-(\lambda - k\lambda_p)^2}{2\sigma'^2_s}\right) \\ \sigma'_s &= \sigma_s/10 \end{aligned} \quad (4.14)$$

onde:

$S(\lambda)$ = distribuição espectral de potência (W/m)

σ_s = largura espectral rms do laser (m)

λ_p = comprimento de onda de pico do laser (m)

4.3 Modelamento do canal

No nosso modelo a fibra é considerada como um canal de transmissão linear em potência³, cuja função de transferência é caracterizada por uma eficiência de acoplamento, η_F , um fator de atenuação, $H_F(0)$, e um filtro passa-baixa $H_F^*(f)$. A figura 4.2 apresenta um esquema do modelo geral adotado para a fibra.

³Esta é uma consideração válida para os sistemas que abordamos neste trabalho [12].

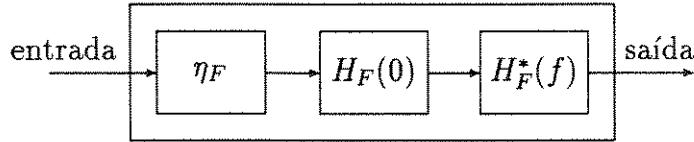


Figura 4.2: Modelo geral para fibra óptica

A relação entre as potências ópticas na entrada e na saída da fibra é dada por:

$$p_r(t) = p_e(t) * h_F(t) \quad (4.15)$$

onde:

$p_r(t)$ = potência óptica na saída da fibra (W)

$p_e(t)$ = potência óptica na entrada da fibra (W)

$h_F(t)$ = resposta impulsiva da fibra

No domínio da frequência temos:

$$P_r(f) = P_e(f)H_F(f) \quad (4.16)$$

onde:

$P_r(f)$ = transformada de Fourier da potência óptica na saída da fibra (W)

$P_e(f)$ = transformada de Fourier da potência óptica na entrada da fibra (W)

$H_F(f)$ = função de transferência da fibra

A função de transferência da fibra, $H_F(f)$, pode ser decomposta em três partes:

$$H_F(f) = \eta_F H_F(0) H_F^*(f) \quad (4.17)$$

onde:

η_F = eficiência de acoplamento

$H_F(0)$ = fator de atenuação

$H_F^*(f)$ = resposta em frequência normalizada da fibra

4.3.1 Eficiência de acoplamento

A eficiência de acoplamento é definida como a relação entre a potência acoplada na fibra e a potência emitida pela fonte e é dada por [13]:

$$\eta_F = \frac{P_a}{P_e} = 1 - (1 - AN^2)^{\frac{m+1}{2}} \quad (4.18)$$

onde:

AN = abertura numérica

m = fator dependente da configuração geométrica da fonte

A abertura numérica é definida por:

$$AN = \sqrt{n_n^2 - n_c^2} \quad (4.19)$$

onde:

n_n = índice de refração do núcleo

n_c = índice de refração da casca

4.3.2 Fator de atenuação

O fator de atenuação leva em conta todas as perdas discutidas no capítulo 1. De maneira geral, a perda total ou atenuação pode ser modelada como:

$$A(\lambda) = A_{abs}(\lambda) + A_R(\lambda) + A_{gd} \quad (4.20)$$

onde:

A_{abs} = perda devido à absorção (dB/km)

A_R = perda devido ao espalhamento Rayleigh (dB/km)

A_{gd} = perda devido ao guia de onda (dB/km)

A atenuação devido à absorção pode ser decomposta em três partes:

$$A_{abs}(\lambda) = A_{uv}(\lambda) + A_{ir}(\lambda) + A_{imp}(\lambda) \quad (4.21)$$

onde:

A_{uv} = absorção do ultra-violeta

A_{ir} = absorção do infra-vermelho

A_{imp} = absorção devido a impurezas

Os modelos para o cálculo destas perdas são dados por [14]:

$$A_{uv} = A \exp B/\lambda \quad (4.22)$$

$$A_{ir} = C \exp -D/\lambda \quad (4.23)$$

$$A_{imp} = \sum_{n=1}^k a_n \lambda^n \quad (4.24)$$

onde os parâmetros a_n , A , B , C e D dependem do tipo de fibra utilizada e são apresentados no capítulo 5.

O modelo para o cálculo da perda devido ao espalhamento Rayleigh é dado por:

$$A_R(\text{dB}/\text{km}) = K\lambda^{-4} \quad (4.25)$$

onde:

K = é o coeficiente de espalhamento ($\text{dB}/\text{km} \cdot \mu\text{m}^{-4}$)

A perda do guia de onda, que pode ser decomposta em duas partes: absorção e difusão, conforme apresentado na seção 1.3.4, pode ser ajustada por uma função do tipo [15]:

$$A_{gd} = [E + F \exp(-G \times 10^{-3}L)]A_g \quad (4.26)$$

onde:

L = comprimento da fibra (km)

A_g = atenuação da interface (dB/km)

E, F e G dependem da fibra utilizada e são apresentados no capítulo 5

A parte constante na equação representa a perda devido à absorção da interface caracterizada por uma constante de absorção que é independente do comprimento da fibra. A parte dependente do comprimento pode ser interpretada como a perda devido à difusão da interface.

Como visto acima, a atenuação total de uma fibra apresenta uma dependência do comprimento de onda, então, definimos um fator de atenuação médio que leva em conta a distribuição espectral da fonte de luz, ou seja:

$$H_F(0) = \frac{\int_{-\infty}^{+\infty} S(\lambda) 10^{-\frac{A(\lambda)L}{10}} d\lambda}{H_T(0)I_d} \quad (4.27)$$

onde:

$S(\lambda)$ = distribuição espectral de potência (W/m)

$A(\lambda)$ = atenuação da fibra (dB/km)

L = comprimento da fibra (km)

4.3.3 Resposta em frequência normalizada

O filtro passa-baixa que caracteriza a resposta em frequência da fibra pode ser modelado como [9]:

$$H_F^*(f) = \frac{1}{1 + j\sqrt{3}\frac{f}{B}} \quad (4.28)$$

onde:

B = largura de faixa de 3dB (Mhz)

A largura de faixa de 3dB da fibra óptica depende do tipo de fibra e é definida abaixo.

Fibra Multimodo

No caso de fibra multimodo de sílica e PCS, a largura de faixa depende da dispersão modal e da dispersão cromática e é dada por [10]:

$$B = \left[\sqrt{\frac{10^9}{B_0^2} l^{2\gamma} + \left(\frac{10^6 D \Delta \lambda}{0.44} \right)^2 l^2} \right]^{-1} \quad (4.29)$$

onde:

B_0 = produto largura de banda x distância devido à dispersão modal (Mhz.km)

l = comprimento da fibra (km)

γ = fator de concatenação (0.7 a 1)

D = dispersão cromática da fibra óptica (ps/nm.km)

$\Delta \lambda$ = largura espectral da fonte (nm)

O produto largura de banda x distância depende do perfil de índice de refração da fibra e é dado por:

Fibra degrau

$$B_0 = \frac{c}{AN^2} n_m \quad (4.30)$$

onde:

n_m = índice de refração do núcleo da fibra

AN = abertura numérica

Fibra gradual

$$B_0 = \frac{2c}{AN^4} n_m^3 \quad (4.31)$$

A dispersão cromática é dada por [3]:

$$D = 10^{15} \frac{S(\lambda - \lambda_0)}{c \lambda^2} \quad (4.32)$$

onde:

$S = 0.047$ (para sílica)

$c =$ velocidade da luz no vácuo

$\lambda_0 =$ comprimento de onda de mínima dispersão cromática (1312 nm para sílica)

$\lambda =$ comprimento de onda

No caso de fibra multimodo de plástico, a dispersão cromática é desprezível em relação à dispersão modal e a largura de faixa é dada por [15]:

$$B = \left[\sqrt{\frac{10^9}{B_0^2}} l^{2\gamma} \right]^{-1} \quad (4.33)$$

onde:

$B_0 =$ produto largura de banda x distância devido à dispersão modal (Mhz.km)

Fibra monomodo

No caso da fibra monomodo somente a dispersão cromática entra no cálculo da largura de faixa. Neste caso temos:

$$B = \left[\left(\frac{10^6 D \Delta \lambda}{0.44} \right)^2 l^2 \right]^{-1} \quad (4.34)$$

4.4 Modelamento do receptor

O receptor é formado por um fotodiodo, um pré-amplificador, um filtro, um circuito de decisão e um decodificador, como mostrado na figura 4.3. A função do fotodiodo consiste em converter a potência óptica que chega no receptor em um sinal elétrico do qual a informação transmitida pode ser extraída. O amplificador é responsável por aumentar a amplitude do sinal elétrico gerado pelo fotodiodo a níveis que permitam seu processamento. O filtro limita a largura de faixa e define a resposta em frequência do receptor, de forma a otimizar seu desempenho. O sinal amplificado e filtrado é amostrado e comparado a um limiar, de forma a decidir se a amostra corresponde a um nível lógico 0 ou a um nível lógico 1. Na saída do circuito de decisão o sinal é decodificado e a informação transmitida é recuperada. Nosso

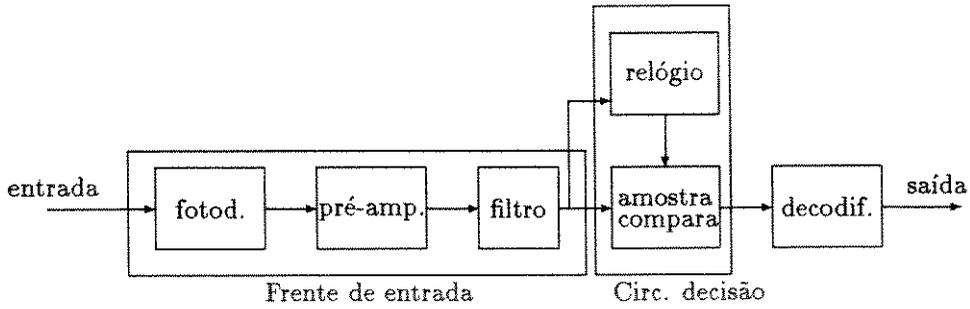


Figura 4.3: Diagrama em blocos do receptor óptico

critério de desempenho é a taxa de erros, que define o número de bits errados encontrado na decodificação de uma seqüência transmitida [16].

A configuração formada pelo fotodiodo, pelo pré-amplificador e pelo filtro é chamada de frente de entrada do receptor, e seu modelo elétrico, incluindo as fontes de ruído, é mostrado na figura 4.4. O fotodiodo com uma eficiência quântica η , é modelado por uma fonte de corrente em paralelo com uma capacitância C_d e uma resistência de polarização R_p , que gera uma corrente de ruído térmico $i_{R_p}(t)$. O pré-amplificador tem uma impedância de entrada representada pela combinação paralela de uma resistência R_a e uma capacitância C_a e apresenta duas fontes de ruído térmico: $i_a(t)$, devido à resistência de entrada R_a , e $e_a(t)$, devido ao canal do amplificador. Estas fontes de ruído são consideradas gaussianas e são descritas por suas densidades espectrais de ruído, S_{i_a} e S_{e_a} . A resistência de realimentação R_f pode ser introduzida intencionalmente ou pode ser infinita, dependendo da configuração do pré-amplificador. Associado a esta resistência tem-se a fonte de ruído térmico i_{R_f} .

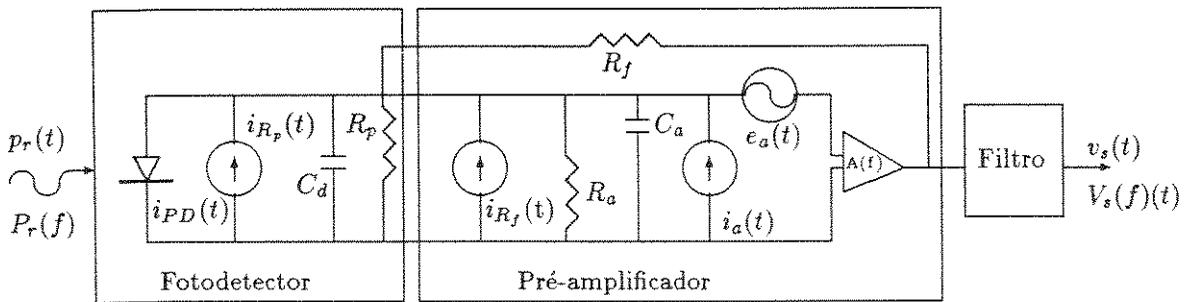


Figura 4.4: Modelo elétrico da frente de entrada do receptor óptico

No nosso modelo, somente o ruído gerado pela frente de entrada do receptor é levado em consideração. O valor da corrente ou voltagem ruidosa pode ser calculado à partir da seguinte expressão [17]:

$$\langle n^2(t) \rangle^{1/2} = \left[\int_0^\infty S(f) |H_{eq}(f)|^2 df \right]^{1/2} \quad (4.35)$$

onde:

$S(f)$ = densidade espectral unilateral de ruído

$H_{eq}(f)$ = faixa equivalente de ruído

4.4.1 Fotodiodo

A corrente gerada pelo fotodiodo, $i_{PD}(t)$, corresponde a um processo aleatório, cujos parâmetros estatísticos dependem do tipo de fotodiodo utilizado (PIN ou APD). Na saída do fotodiodo, a corrente pode ser separada em duas partes: o valor médio da fotocorrente, que corresponde ao sinal contendo a informação desejada, e o ruído introduzido pelo fotodiodo, como mostra a equação 4.36.

$$i_{PD}(t) = \langle i_{PD}(t) \rangle + n_{PD}(t) \quad (4.36)$$

onde:

$i_{PD}(t)$ = corrente gerada pelo fotodiodo (A)

$\langle i_{PD}(t) \rangle$ = valor médio da fotocorrente⁴ (A)

$n_{PD}(t)$ = corrente ruidosa gerada no fotodiodo (A)

A valor médio da fotocorrente é dado por:

$$\langle i_{PD}(t) \rangle = RGp_r(t) \quad (4.37)$$

onde:

R = responsividade do fotodiodo (A/W)

G = ganho médio de avalanche ($G = 1$ para fotodiodo PIN)

$p_r(t)$ = potência óptica recebida (W)

A responsividade do fotodiodo é dada por:

$$R = \frac{\lambda q}{hc} \eta \quad (4.38)$$

onde:

η = eficiência quântica do diodo

Ruído do fotodiodo

O ruído gerado no fotodiodo, $n_{PD}(t)$, é composto do ruído inerente ao sinal fotodetectado, chamado também ruído quântico, caracterizado por uma distribuição de Poisson não-estacionária; do ruído de avalanche, inerente aos fotodiodos de avalanche, caracterizado por uma distribuição de McIntyre-Personick não estacionária; e do ruído devido à corrente de escuro do fotodiodo, criada pela corrente inversa e radiações parasitas. A densidade espectral deste ruído, $S_{n_{PD}}$, depende do sinal recebido e é dada por:

$$S_{n_{PD}} = S_{i_B} + S_{R_p} \quad (4.39)$$

onde:

⁴média estatística

$S_{n_{PD}}$ = densidade espectral da corrente ruidosa total do fotodiodo (A^2/Hz)

S_{i_B} = densidade espectral da corrente de ruído interna do fotodiodo (A^2/Hz)

S_{R_p} = densidade espectral da corrente de ruído gerada pela resistência de polarização do fotodiodo (A^2/Hz)

A densidade espectral da corrente de ruído interna do fotodiodo é dada por:

$$S_{i_B} = 2q\{[\langle i_{PD}(t) \rangle + i_{obs_M}] \langle g^2 \rangle + i_{obs_{NM}}\} \quad (4.40)$$

onde:

i_{obs_M} = fotocorrente de escuro criada na região de ganho do fotodiodo (A)

$i_{obs_{NM}}$ = fotocorrente de escuro criada na superfície do fotodiodo (A)

g = ganho de avalanche

O valor quadrático médio do ganho de avalanche pode ser aproximado por [18]:

$$\begin{aligned} \langle g^2 \rangle &= G^2 F(G) \\ F(G) &= G[1 - (1 - k_{ef})(\frac{G-1}{G})^2] \end{aligned} \quad (4.41)$$

onde:

$F(G)$ = fator de excesso de ruído do fotodiodo

G = ganho de avalanche médio

k_{ef} = taxa de ionização

A densidade espectral da corrente de ruído do fotodiodo é dada então por:

$$S_{i_B} = 2q\{[\langle i_{PD}(t) \rangle + i_{obs_M}]G^2 F(G) + i_{obs_{NM}}\} \quad (4.42)$$

Associada a esta corrente, a resistência de polarização do fotodiodo gera uma corrente de ruído cuja densidade espectral é dada por:

$$S_{R_p} = \frac{4K_B T}{R_p} \quad (4.43)$$

onde:

R_p = resistência de polarização do fotodiodo (Ω)

K_B = constante de Boltzmann (1.38054×10^{-23} J/K)

T = temperatura (K)

4.4.2 Pré-amplificador

Na saída do pré-amplificador a voltagem gerada pode ser separada em duas partes: o valor médio da voltagem, que corresponde ao sinal contendo a informação desejada, e o ruído introduzido pela combinação fotodiodo + pré-amplificador, como mostra a equação 4.44.

$$v_{PA}(t) = \langle v_{PA}(t) \rangle + n_{PA}(t) \quad (4.44)$$

onde:

$\langle v_{PA}(t) \rangle =$ valor médio da voltagem (V)

$n_{PA}(t) =$ voltagem ruidosa gerada pelo fotodiodo + pré-amplificador (V)

O valor médio da voltagem que representa o sinal informação é dado por:

$$\langle v_{PA}(t) \rangle = \langle i_{PD}(t) \rangle * h_{PA}(t) \quad (4.45)$$

onde:

$h_{PA}(t) =$ resposta impulsiva do pré-amplificador (V/A)

No domínio da frequência temos:

$$V_{PA}(f) = \langle I_{PD}(f) \rangle H_{PA}(f) \quad (4.46)$$

onde:

$H_{PA}(f) =$ função de transferência do pré-amplificador

$I_{PD}(f) =$ transformada de Fourier da fotocorrente (A)

A função de transferência do pré-amplificador é dada por:

$$H_{PA}(f) = H_{PA}(0) H_{PA}^*(f) \quad (4.47)$$

onde:

$H_{PA}(0) =$ ganho de transimpedância do pré-amplificador

$H_{PA}^*(f) =$ função de transferência normalizada do pré-amplificador

Existem duas abordagens no projeto de pré-amplificadores para receptores ópticos: alta-impedância e transimpedância.

Pré-amplificador de alta-impedância

O pré-amplificador de alta-impedância apresenta função de transferência dada por [19]:

$$H_{PA}(f) = \frac{AR_T}{1 + j2\pi R_T C_T f} \quad (4.48)$$

onde:

A = ganho de voltagem do pré-amplificador

R_T = Resistência total na entrada do pré-amplificador (Ω)

C_T = Capacitância total na entrada do pré-amplificador (F)

A capacitância e a resistência totais na entrada do pré-amplificador são dadas por:

$$\begin{aligned} C_T &= C_d + C_a \\ R_T &= \frac{R_p R_a}{R_p + R_a} \end{aligned}$$

onde:

C_a = capacitância de entrada do pré-amplificador (F)

C_d = capacitância do fotodiodo (F)

R_a = Resistência na entrada do pré-amplificador (Ω)

R_p = Resistência de polarização do fotodiodo (Ω)

Pré-amplificador de transimpedância

O pré-amplificador de transimpedância apresenta uma função de transferência dada por [19]:

$$H_{PA}(f) = \frac{R_f}{1 + j2\pi R_f C_T f / A} \quad (4.49)$$

onde:

A = ganho de voltagem do pré-amplificador em malha aberta

R_f = Resistência de realimentação (Ω)

A resistência de realimentação é uma função da taxa de transmissão e o nosso trabalho utiliza o seguinte modelo:

$$R_f = \frac{A}{2\pi C_T \times 0.8T_b} \quad (4.50)$$

onde:

T_b = taxa de transmissão (bit/s)

Ruído do pré-amplificador

O ruído gerado pelo pré-amplificador é caracterizado por uma distribuição gaussiana e modelado como uma fonte de corrente ruidosa com densidade espectral S_{i_a} em paralelo com a entrada e uma fonte de voltagem ruidosa com densidade espectral S_{e_a} em série com a entrada, como mostra a figura 4.4. Estas densidades espectrais dependem do tipo de transistor de entrada do pré-amplificador: FET ou bipolar.

Transistor FET

Um típico transistor FET apresenta uma resistência de entrada R_a muito alta (geralmente maior que $10^6 \Omega$), assim, podemos definir $R_a = \infty$. A resistência total R_T , se reduz então ao valor da resistência de polarização R_p . A densidade espectral de corrente ruidosa S_{i_a} para o transistor FET é dada por [19]:

$$S_{i_a} = \frac{4K_B T}{R_a} \quad (4.51)$$

Dado que a resistência de entrada é muito alta, a densidade espectral da corrente ruidosa é desprezível. Assim, a fonte de ruído básica em um FET é o ruído térmico da resistência do canal de condução, que é caracterizado pela transcondutância g_m . A densidade espectral da voltagem de ruído, S_{e_a} , é dada por [19]:

$$S_{e_a} = \frac{2}{3} \frac{4K_B T}{g_m} \quad (4.52)$$

onde:

g_m = transcondutância do FET (S)

Transistor Bipolar

A resistência interna de entrada de um transistor bipolar, R_{in} , é dada por:

$$R_{in} = \frac{K_B T}{q i_B} \quad (4.53)$$

onde:

i_B = corrente de base do transistor (A)

Para um amplificador com transistor bipolar a resistência de entrada, R_a , é dada por uma combinação paralela da resistência de polarização do transistor e da resistência de entrada do transistor R_{in} . A densidade espectral da corrente de ruído, S_{i_a} , resulta do ruído balístico da corrente de base e é dada por:

$$S_{i_a} = 2q I_B = \frac{4K_B T}{R_{in}} \quad (4.54)$$

A densidade espectral da voltagem de ruído é dada por:

$$S_{e_a} = \frac{4K_B T}{g_m} \quad (4.55)$$

A transcondutância g_m é dada por:

$$g_m = \frac{\beta}{R_{in}} \quad (4.56)$$

onde:

β = ganho de corrente do transistor

Em um pré-amplificador de transimpedância, há ainda uma corrente ruidosa devido à resistência de realimentação R_f . A densidade espectral desta corrente é dada por:

$$S_{R_f} = \frac{4KbT}{R_f} \quad (4.57)$$

4.4.3 Filtro

Na saída do filtro, a voltagem gerada pode ser separada em duas componentes: o valor médio de voltagem, que corresponde ao sinal contendo a informação desejada, e o ruído introduzido pela frente de entrada do receptor, ou seja:

$$v_S(t) = \langle v_S(t) \rangle + n_S(t) \quad (4.58)$$

onde:

$\langle v_S(t) \rangle$ = valor médio da voltagem (V)

$n_S(t)$ = voltagem ruidosa (V)

O valor médio da voltagem é dado por:

$$\begin{aligned} \langle v_S(t) \rangle &= \langle v_{PA}(t) \rangle * h_R(t) \\ &= \langle i_{PD}(t) \rangle * h_{PA}(t) * h_R(t) \\ &= RGP_r(t) * h_{PA}(t) * h_R(t) \end{aligned} \quad (4.59)$$

onde:

$h_R(t)$ = resposta impulsiva do filtro

No domínio da frequência temos:

$$\begin{aligned} V_S(f) &= \langle V_{PA} \rangle H_R(f) \\ &= \langle I_{PD}(f) \rangle H_{PA}(f) H_R(f) \\ &= RGP_r(f) H_{PA}(f) H_R(f) \end{aligned} \quad (4.60)$$

onde:

$H_R(f)$ = função de transferência do filtro

A função de transferência do filtro depende do tipo de receptor utilizado.

Receptor equalizante

No nosso modelo a função de transferência do filtro equalizante é uma combinação dos estágios de equalização do transmissor, da fibra e do pré-amplificador e do estágio de formatação do sinal e eliminação do ruído, ou seja:

$$\begin{aligned} H_R(f) &= H_T(f)^{-1} H_F(f)^{-1} H_{PA}(f)^{-1} H_n(f) \\ H_T(f)^{-1} &= H_{eq_{trans}}(f) \\ H_F(f)^{-1} &= H_{eq_{fib}}(f) \\ H_{PA}(f)^{-1} &= H_{eq_{pre}}(f) \end{aligned} \quad (4.61)$$

onde:

$H_{eq_{trans}}$ = função de equalização do transmissor

$H_{eq_{fib}}$ = função de equalização da fibra

$H_{eq_{pre}}$ = função de equalização do pré-amplificador

H_n = função de formatação do sinal e eliminação do ruído

Receptor não equalizante

No nosso modelo a função de transferência do filtro não equalizante é somente uma combinação dos estágios de equalização do pré-amplificador e de formatação do sinal e eliminação do ruído, ou seja:

$$H_R(f) = H_{PA}(f)^{-1} H_n(f) \quad (4.62)$$

Filtro de formatação do sinal e filtragem do ruído

Os filtros que fazem a formatação do sinal e filtragem do ruído na saída do receptor são de dois tipos: Butterworth e cosseno levantado.

Filtro Butterworth

Para os filtros do tipo Butterworth de ordem n , onde n representa o número de elementos reativos (capacitores e indutores) do filtro, a função de transferência é dada por [17]:

$$H_n(f) = \frac{1}{P_n(jf/B)} \quad (4.63)$$

onde B é igual à largura de banda de 3dB e $P_n(jf/B)$ é um polinômio complexo cuja família é definida pela propriedade:

$$|P_n(jf/B)|^2 = 1 + (f/B)^{2n} \quad (4.64)$$

então temos:

$$|H_n(f)| = \frac{1}{\sqrt{1 + (f/B)^{2n}}} \quad (4.65)$$

A tabela 4.1 lista os polinômios de Butterworth para n de 1 a 4, usando a variável normalizada $p = jf/B$.

n	$P_n(p)$
1	$1 + p$
2	$1 + \sqrt{2}p + p^2$
3	$(1 + p)(1 + p + p^2)$
4	$(1 + 0.765p + p^2)(1 + 1.848p + p^2)$

Tabela 4.1: Polinômios de Butterworth

Filtro Cosseno levantado

Para os filtros do tipo cosseno levantado a função de transferência, $H_n(f)$, apresenta modelos dependentes de um fator de ocupação de faixa (α) e da frequência. Assim temos [16]:

Se $\alpha = 0$

$$H_n(f) = \begin{cases} 1 & \text{se } 0 < fT_b \leq \frac{1-\alpha}{2} \\ 0 & \text{caso contrário} \end{cases}$$

Se $\alpha = 1$

$$H_n(f) = \begin{cases} 0.5[1 - \sin(\frac{\pi fT_b}{\alpha} - \frac{\pi}{2\alpha})] & \text{se } \frac{1-\alpha}{2} \leq fT_b \leq \frac{1+\alpha}{2} \\ 0 & \text{caso contrário} \end{cases}$$

Se $0 < \alpha < 1$

$$H_n(f) = \begin{cases} 1 & \text{se } 0 < fT_b \leq \frac{1-\alpha}{2} \\ 0.5[1 - \sin(\frac{\pi fT_b}{\alpha} - \frac{\pi}{2\alpha})] & \text{se } \frac{1-\alpha}{2} \leq fT_b \leq \frac{1+\alpha}{2} \\ 0 & \text{caso contrário} \end{cases}$$

onde:

α = fator de ocupação do intervalo

T_b = taxa de transmissão (bits/s)

Ruído na saída do filtro

Podemos considerar o conjunto das fontes de ruído no receptor como uma fonte total de ruído equivalente na entrada do pré-amplificador cujo efeito na saída é idêntico ao efeito produzido por todas as fontes de ruído, e tendo densidade espectral de corrente ruidosa dada por:

$$\begin{aligned} S(f) &= S_A + f^2 S_B \\ S_A &= S_{i_B} + S_{R_p} + S_{R_f} + S_{i_a} + \frac{S_{e_a}}{R_T^2} \\ S_B &= 2\pi C_T^2 S_{e_a} \end{aligned} \quad (4.66)$$

onde:

S_{i_B} = densidade espectral da corrente de ruído do fotodiodo (A^2/Hz)

S_{R_p} = densidade espectral da corrente de ruído da resistência de polarização (A^2/Hz)

S_{R_f} = densidade espectral da corrente de ruído da resistência de realimentação (se o pré-amplificador possui realimentação) (A^2/Hz)

S_{i_a} = densidade espectral da corrente de ruído do pré-amplificador (A^2/Hz)

S_{e_a} = densidade espectral da voltagem de ruído do pré-amplificador (V^2/Hz)

O ruído total do sistema na saída do filtro receptor é dado por:

$$n_s(t) = v_B(t) + v_{th}(t) \quad (4.67)$$

onde:

$v_B(t)$ = tensão do ruído balístico

$v_{th}(t)$ = tensão do ruído térmico

A variância de $n(t)$, definida como a densidade espectral de potência de ruído total, é dada por:

$$\begin{aligned} \langle n^2(t) \rangle &= \langle [v_s(t) - \langle v_s(t) \rangle]^2 \rangle \\ &= \langle v_s^2(t) \rangle - \langle 2v_s(t) \langle v_s(t) \rangle \rangle + \langle v_s(t) \rangle^2 \\ &= \langle v_s^2(t) \rangle - \langle v_s(t) \rangle^2 \\ &= \langle v_B^2(t) \rangle + \langle v_{th}^2(t) \rangle \end{aligned} \quad (4.68)$$

onde

$\langle v_B^2(t) \rangle$ = variância do ruído balístico do fotodiodo

$\langle v_{th}^2(t) \rangle$ = variância do ruído térmico do pré-amplificador

A variância do ruído total na saída do filtro é definida por:

$$\langle n_S^2(t) \rangle \triangleq \int_{-\infty}^{+\infty} S(f) |H_{PA}(f)H_R(f)|^2 df \quad (4.69)$$

Levando a equação 4.66 em 4.69, temos:

$$\langle n_S^2(t) \rangle = S_A \int_{-\infty}^{+\infty} |H_{PA}H_R(f)|^2 df + S_B \int_{-\infty}^{+\infty} f^2 |H_{PA}H_R(f)|^2 df \quad (4.70)$$

4.5 Conclusão

Neste capítulo apresentamos uma modelagem dos componentes presentes em um sistema de comunicação por fibra óptica. O sistema foi dividido em três blocos: o transmissor, o canal e o receptor e a função de transferência de cada bloco foi escrita em função dos parâmetros físicos de cada um dos componentes. Estes modelos foram utilizados na construção da biblioteca de modelos, apresentada no capítulo 5.

Referências Bibliográficas

- [1] J. C. Cartledge, "Receiver sensitivity of optical fiber communication systems: the effects of channel characteristics and receiver filter design", *IEEE Trans. Commun.*, **COM-26**, pp. 1103-1109 (1978).
- [2] D. D. Duff, "Computer Aided Design of Digital Lightwave Systems", *IEEE Journal on Selected Areas in Communications*, **SAC-6**, pp.171-185 (1984).
- [3] A. F. Elrefaie et al., "Computer Simulation of Digital Lightwave Links", *IEEE Journal on Selected Areas in Communications*, **SAC-6**, pp. 94-105 (1988).
- [4] M. T. Abuelma'atti, "Nonlinear distortion properties of fibre-optic analogue transceivers", *Int. J. of Optoelectronics*, **vol 4**, pp. 163-172 (1989).
- [5] A. A. Bergh e P. J. Dean, *Light emitting diodes*, Claredon, London, 1976.
- [6] D. Gloge et al., "High-speed digital lightwave communication using LEDs and PIN photodiodes at 1.3 μm ", *Bell System Tech. J.*, **vol 59**, pp. 1365-1382 (1980).
- [7] J. Gimlett e N. Cheung, "Dispersion penalty analysis for LED/ single mode fiber transmission systems", *IEEE J. Lightwave Technol.*, **vol LT-4**, pp. 1381-1392 (1986).
- [8] J. Wilson e J.F.B. Hawkes, *Optoelectronics: An Introduction*, Prentice-Hall, 2nd edition, 1991.
- [9] D. J. Channin, "Optoelectronic Performance Issues in Fiber-Optic Communications", *RCA Review*, **vol. 46**, pp. 240-269 (1985).
- [10] A. Moncalvo e R. Pietroiusti, "Transmission numérique sur fibres optiques", *Journal des Télécommunications*, **vol 49-II**, pp. 84-92 (1982).

-
- [11] E. Moschim, comunicação particular.
- [12] S. D. Personick, "Baseband linearity and equalizing in digital fiber optical communication systems", *Bell Syst. Tech. J.*, vol 52, pp. 1175-1193 (1973).
- [13] M. K. Barnaski, "Coupling components for optical fiber waveguides", in *Fundamentals of optical fiber communications*, edited by M. K. Barnaski, Academic Press, 1976.
- [14] K. Inada, "A new method relating to optical fiber attenuation", *Opt. Comm.*, vol 19, pp. 437-440 (1976).
- [15] E. Moschim, "Contribution à la modélisation des composants photoniques et des systèmes de communication utilisant les fibres optiques en plastique comme support de transmission", thèse docteur en science physique, Université de Paris-Sud, Orsay, France, 1989.
- [16] S. D. Personick, "Receiver design for digital fiber optical communication systems I and II", *Bell Syst. Tech. J.*, vol 52, 843-886 (1973).
- [17] A. B. Carlson, *Communication systems - An introduction to signals and noise in electrical communication*, McGraw-Hill, 1986.
- [18] R. J. McInyre, "The distribution of gains in uniformly multiplying avalanche photodiodes theory", *IEEE Trans. Electron. Devices*, vol ED-19, pp. 703-721 (1972).
- [19] R. G. Smith e S. D. Personick, "Receiver design for optical fiber communication systems", in *Semiconductor devices for optical communication*, edited by H. Kressel, Springer-Verlag (1980).
- [20] H. Waldman, C. Bezzan e E. Moschim, "Cálculo geral de desempenho de receptores ópticos em sistemas de transmissão digital (aproximação gaussiana com a ocorrência da interferência entre símbolos devido à fibras dispersivas", Contrato FEC-UNICAMP/TELEBRÁS 033/80, Nota Técnica 7, 1983.

Capítulo 5

Biblioteca de modelos

Neste capítulo são apresentadas as rotinas que implementam os modelos de cada um dos blocos funcionais que compõe o sistema apresentado no capítulo 1.

5.1 Introdução

A biblioteca de modelos contém as rotinas que representam os modelos matemáticos dos componentes modelados no capítulo anterior. Os parâmetros físicos e geométricos de cada um destes componentes estão armazenados nas bases de dados, que podem ser modificadas para se obter uma otimização¹. As rotinas desenvolvidas implementam as funções de transferência destes componentes e ruídos, além de gerar os sinais que representam as entradas do sistema. Nas referências [1][2][4] são apresentados alguns resultados.

5.2 Gerador de sinais

O gerador de sinais, implementado pela rotina *geraentr.c*, é responsável por gerar um arquivo de dados que representa o sinal a ser transmitido. Os dados gerados podem ser de três tipos: *aleatório*, que corresponde a uma seqüência de uns e zeros gerada randomicamente; *trem de pulsos*, equivalendo a uma seqüência constante de uns e zeros, nesta ordem e *pulso*, que corresponde a um valor um seguido de zeros.

No caso da seqüência aleatória, o usuário define a probabilidade de que o número seja igual a zero, selecionando um valores entre 0 a 9. Se o limiar escolhido for zero, todos os números gerados serão iguais a um. Se for escolhido o limiar nove, 90% dos números gerados serão iguais a zero. O usuário seleciona também o número de amostras a serem geradas: 2048, 4096, 8192 e 16384, e a taxa de transmissão dos dados, em Mbit/s.

O programa é chamado com dois parâmetros: *opção*, que informa o tipo de sinal a ser gerado escolhido pelo usuário, e *namost*, que indica o número de amostras a serem geradas. A saída do gerador é um arquivo binário denominado *sinal.dat*, contendo o sinal gerado.

¹Estes dados foram obtidos na literatura atual.

5.3 Codificador

O codificador, implementado através da rotina *codifica.c*, é responsável por codificar os dados de entrada do sistema, que foram gerados através da rotina *geraentr*, em um dos códigos escolhidos pelo usuário dentre: NRZ, RZ e Manchester. O programa é chamado com dois parâmetros - *opção* e *nome*. A opção indica o padrão de codificação selecionado, enquanto o nome informa o arquivo cujos dados serão codificados. Cada valor a ser codificado corresponde a um conjunto de dez amostras. Assim, quando a codificação selecionada é a NRZ, as dez amostras permanecem com o mesmo valor, equivalendo ao sinal gerado zero ou um. No caso da codificação selecionada ser a RZ, o valor do bit é mantido durante as cinco primeiras amostras e as cinco últimas são gravadas como zero. No caso da codificação Manchester, o valor do bit é mantido nas cinco primeiras amostras e invertido nas cinco últimas. Neste caso, um bit 1 é gravado como cinco amostras 1 seguidas de cinco amostras 0, enquanto um bit 0 é codificado como cinco amostras 0 seguidas de cinco amostras 1. Estes dados codificados são gravados em um arquivo binário denominado *codif.dat*.

5.4 LED

Para a análise do LED estão disponíveis três opções: potência óptica, distribuição espectral e resposta em frequência, implementadas pelas rotinas *an-led.c*, *pot-led.c*, *desp-led.c* e *fc-led.c*. Nos três casos o usuário deve selecionar o comprimento de onda do diodo, dentre 570 nm, 650 nm, 850 nm, 1300 nm e *outro*. Se a opção escolhida for *outro* será perguntado o nome do arquivo que contém os parâmetros do LED. Para qualquer outra opção estes parâmetros já estão definidos no arquivo *led.std*.

Na primeira opção, implementada pela rotina *pot-led.c*, o usuário tem a sua disposição a curva da potência emitida em função da corrente, em Watts. A rotina tem apenas um parâmetro de entrada, *opção*, que indica o comprimento de onda de pico do LED, determinando os parâmetros recuperados do arquivo para o cálculo da eficiência quântica do diodo, dados pela tabela 5.1. Ao final do programa é gerado o arquivo *potw.dat*, correspondendo à potência emitida.

Para o cálculo da distribuição espectral, implementada pela rotina *desp-led.c*, é utilizada a equação 4.8. Os resultados mostram os valores da distribuição espectral de potência no intervalo de 100 nm em torno do comprimento de onda de pico do diodo. A rotina tem um parâmetro de entrada, *opção*, indicando o comprimento de onda de pico do LED, que é utilizado na recuperação dos parâmetros do arquivo *led.std*. Além destes parâmetros, é pedido ao usuário, no momento da execução da rotina, o valor da corrente fornecida na entrada do diodo. Os valores representando a distribuição espectral são gravados ao final da rotina no arquivo *dist.dat*.

A resposta em frequência, implementada pela rotina *fc-led.c*, gera valores correspondentes à função de transferência do LED, de acordo com os modelos matemáticos apresentados no capítulo 4. O programa é chamado com três parâmetros: *opção*, que indica o comprimento de onda do LED para o qual será calculada a função de transferência; *namost*, que indica o número de amostras do sinal que foram geradas e *tbits*, que indica a taxa de transmissão dos dados (Mbit/s). Ao final do programa é gerado um arquivo binário denominado *led.trf* contendo os valores correspondentes à função de transferência do led.

Os parâmetros recuperados do arquivo `led.std` que são utilizados no cálculo da função de transferência, da potência óptica e da distribuição espectral são: comprimento de onda de pico, λ_p ; largura espectral, σ_s ; tempo de recombinação radiativa, τ_r ; tempo de recombinação não-radiativa, τ_{nr} ; eficiência de injeção de corrente, η_{inj} ; índice de refração do semicondutor, n_s ; e temperatura, T . A tabela 5.1 apresenta estes parâmetros e seus valores típicos para os diversos comprimentos de onda.

	λ_p (nm)	σ_s (nm)	τ_r (ns)	τ_{nr} (ns)	η_{inj}	n_s	T(K)
570	567	26	5000	375	0.9	3.33	300
650	665	35	21	80	0.93	3.67	300
850	832	50	10	40	0.95	3.62	300
1300	1310	62	2.5	10	1	3.56	300

Tabela 5.1: *Parâmetros para análise do LED*

A rotina `an-led.c` é responsável por identificar a seleção do usuário, ativando a rotina adequada e preparando as opções para a rotina dedicada a traçar os gráficos correspondentes aos resultados obtidos. A rotina tem dois parâmetros de entrada: `opção1`, que indica o tipo de análise desejada, e `opção2`, que informa o comprimento de onda de pico do LED. Nenhum arquivo é gerado pela rotina.

5.5 Laser

Para a análise do laser estão disponíveis três opções: potência óptica, distribuição espectral e resposta em frequência, implementadas pelas rotinas `an-laser.c`, `pot-laser.c`, `desp-laser.c` e `fc-laser.c`. Em todos os casos, a seleção do comprimento de onda é feita através de menus, dentre as opções 670nm, 850nm, 1300nm, 1550nm e outro. Como no caso do LED, se a opção selecionada for `outro` é pedido o nome do arquivo que contém os parâmetros do laser. Para qualquer outra opção estes parâmetros já estão definidos no arquivo `laser.std`.

Na primeira opção, implementada pela rotina `pot-laser.c`, o usuário tem a sua disposição a curva da potência emitida em função da corrente, em Watts. No caso do laser, a função que define a eficiência quântica depende da corrente injetada, como mostrado pelas equações 4.9 a 4.12. A corrente de limiar, I_{th} , que define o modelo a ser adotado para os cálculos, depende do comprimento de onda de emissão do laser, λ , como dado pela tabela 5.2.

λ (nm)	I_{th} (mA)
670	50
850	35
1300	20
1550	10

Tabela 5.2: *Corrente de limiar*

A rotina tem um parâmetro de entrada, `opção`, que indica o comprimento de onda de pico do laser sendo analisado. Para o cálculo da eficiência abaixo da corrente de limiar são utilizados os valores do tempo de recombinação radiativa, τ_r , tempo de recombinação

não-radiativa, τ_{nr} , eficiência de injeção de corrente, η_{inj} , eficiência quântica interna, η_{int} e eficiência quântica externa, η_{ext} , dados pela tabela 5.3.

	λ_p (nm)	τ_r (ns)	τ_{nr} (ns)	η_{inj}	n_s	R_1	γ (m ⁻¹)	l (μ m)
670	665	21	80	0.93	3.67	0.32	3.5e3	300
850	832	10	40	0.95	3.62	0.32	3.5e3	300
1300	1310	2.5	10	1	3.56	0.32	3.5e3	300
1550	1550	0.5	5	1	3.52	0.32	3.5e3	300

Tabela 5.3: *Parâmetros para cálculo da eficiência quântica do laser*

Quando a corrente injetada supera a corrente de limiar, o cálculo da eficiência quântica leva em consideração um coeficiente de perdas, γ , reflectância dos espelhos, R_1 , e dimensão do laser, l , dados pela tabela 5.3, além dos valores das correntes injetada e de limiar. Ao final do programa é gerado o arquivo **potw.dat** que corresponde à potência emitida.

Para o cálculo da distribuição espectral, implementada pela rotina *desp-laser.c*, é utilizada a equação 4.14. O resultado mostra os modos gerados contidos na largura espectral da curva de ganho, que equivale a uma variação de 100nm em torno do comprimento de onda de pico do laser. A rotina tem um parâmetro de entrada, *opção*, indicando o comprimento de onda de pico do diodo, que será usado também na recuperação dos parâmetros no arquivo **laser.std**. No momento da execução da rotina é solicitado ao usuário o valor da corrente fornecida na entrada do diodo. Ao final é gerado um arquivo denominado **dist.dat**, contendo os valores equivalentes à distribuição espectral do laser.

A resposta em frequência do laser é implementada pela rotina *fc-laser.c*, chamada com três parâmetros: *opção*, que indica o comprimento de onda para o qual será calculada a função de transferência, *namost*, que indica o número de amostras geradas do sinal e *tbits*, que indica a taxa de transmissão dos dados (Mbit/s). Ao final da rotina é gerado o arquivo **laser.trf** que representa a função de transferência do laser.

Os parâmetros recuperados do arquivo **laser.std** que são utilizados no cálculo da função de transferência e da distribuição espectral, além dos já apresentados nas tabelas 5.2 e 5.3, são: tempo de vida dos portadores minoritários, τ_{sp} , tempo de vida dos fótons, τ_{ph} , largura espectral, σ_s , e temperatura, T , apresentados na tabela 5.4.

	τ_{sp} (ns)	τ_{ph} (ps)	σ_s (nm)	T (K)
670	15	6	10	300
850	10	2	4	300
1300	5	2	2	300
1550	3	2	1	300

Tabela 5.4: *Parâmetros para análise do laser*

A rotina *an-laser* é responsável por identificar a seleção do usuário, ativando a rotina apropriada e atribuindo valores às variáveis que identificam o gráfico a ser traçado. A rotina tem dois parâmetros de entrada: *opção1*, que indica o tipo de análise a ser realizada, e *opção2*, que indica o comprimento de onda do laser. Esta rotina não gera arquivo de saída.

5.6 Fibra Óptica

Para a análise da fibra estão disponíveis duas opções: resposta em frequência e curvas de atenuação, implementadas pelas rotinas *an-fibra.c*, *fc-fibra.c* e *aten-fibra.c*.

A fibra a ser simulada pode ser do tipo monomodo, multimodo degrau (de plástico, PCS ou sílica) e multimodo gradual (de PCS ou sílica) ou *outro*. Exceto quando a opção selecionada for *outro*, caso em que é pedido o nome do arquivo que contém os parâmetros da fibra, os valores dos parâmetros utilizados pelo rotina são obtidos a partir do arquivo *fibra.std* e correspondem ao índice de refração do núcleo, n_n e ao índice de refração da casca, n_c . A Tabela 5.5 apresenta os valores típicos destes parâmetros.

	n_n	n_c	k	A_g (dB/km)
monomodo	1.456	1.452	1.0	0.2e3
mult.degrau.plástico	1.4	1.5	2.75	40e3
mult.degrau PCS	1.456	1.4	2.2	2.2e3
mult.degrau sílica	1.456	1.449	1.6	1.6e3
mult.gradual PCS	1.456	1.4	2.0	2.0e3
mult.gradual sílica	1.456	1.449	1.2	0.7e3

Tabela 5.5: Parâmetros para cálculo da função de transferência da fibra

A resposta em frequência da fibra, implementada pela rotina *fc-fibra.c*, está relacionada com a abertura numérica, a atenuação e a dispersão, que dependem do tipo de fibra utilizado. A rotina é chamada com três parâmetros: *opção*, que indica o tipo de fibra a ser simulada, *namost*, que indica o número de amostras do sinal gerado e *tbits*, indicando a taxa de transmissão de dados. Na saída da rotina é gerado um arquivo denominado *fibra.trf* cujos valores representam a função de transferência da fibra.

As curvas de atenuação são calculadas através da rotina *aten-fibra.c*, que apresenta ao usuário as perdas devido à absorção e devido ao espalhamento Rayleigh, além da atenuação total, correspondendo aos arquivos gerados *abs.dat*, *ray.dat* e *aten.dat*. A rotina tem um parâmetro de entrada *opção*, que indica o tipo da fibra sendo analisada.

Os valores usados no cálculo da absorção por ultra-violeta e infra-vermelho, atenuação Rayleigh e perdas devido ao guia de onda são apresentados nas tabelas 5.5 e 5.6.

	A	B	C	D	E	F	G
plástico	4.61e-12	11.8	0	0	0.25	0.64	20
PCS	1e-4	4.63	7.81	44.48	0.15	0.35	30
sílica	3e-5	4.63	7.81	44.48	0.1	0.25	50

Tabela 5.6: Valores para cálculo de absorções

No modelamento da absorção devido a impurezas, dado pela equação 4.24, é utilizada uma aproximação polinomial, onde a_n assume valores que dependem do tipo de fibra utilizado. Para as fibras de plástico estes valores são apresentados na tabela 5.7. Os valores de a_n para as fibras de PCS e sílica são apresentados na tabela 5.8.

	$400 \leq \lambda \leq 600$ (nm)	$620 < \lambda \leq 700$ (nm)
	k = 7	k = 8
a1	1.030644135e7	6.247968759256442e10
a2	-1.277382022e8	-6.670322004235436e11
a3	6.573280270e8	3.049894015824350e12
a4	-1.797343234e9	-7.742136920624145e12
a5	2.753846451e9	1.178427480283088e13
a6	-2.241542024e9	-1.075511120783437e13
a7	7.572115388e8	5.449760609519782e12
a8	0	-1.182743053906084e12

	$700 < \lambda \leq 800$ (nm)	$\lambda > 800$ (nm)
	k = 8	k = 8
a1	-1.459621009385149e12	-5.004377532971162e12
a2	1.361942485997290e13	4.087129511501341e13
a3	-5.443398401373562e13	-1.429946938280045e14
a4	1.208031004206551e14	2.778172477200531e14
a5	-1.607704449981710e14	-3.237131595408129e14
a6	1.283082293560241e14	2.262162416884760e14
a7	-5.685902200972915e13	-8.778618948962345e13
a8	1.079286870103311e13	1.459364695007273e13

Tabela 5.7: Valores dos coeficientes a_n para fibra de plástico

	$600 \leq \lambda \leq 1300$ (nm)	$\lambda > 1300$ (nm)
	k = 8	k = 5
a1	-6.82383e3	-1.0268e4
a2	5.36820e4	0
a3	-1.78042e5	2.5606e4
a4	3.23312e5	-2.1690e4
a5	-3.47481e5	0.5114e4
a6	2.21155e5	
a7	-7.72144e4	
a8	1.14138e4	

Tabela 5.8: Valores dos coeficientes a_n para fibras de PCS e de sílica

A rotina *an-fibra.c* é responsável por identificar a seleção do usuário, ativando a rotina adequada e preparando as opções para que os gráficos sejam traçados. A rotina não gera arquivos e possui dois parâmetros de entrada: *opção1*, que indica o tipo de análise desejada, e *opção2*, que indica o tipo da fibra a ser analisada

5.7 Fotodiodo

A rotina *fc-fotod.c* gera valores que representam a corrente eletrônica na saída do fotodiodo, proporcional à potência óptica que chega no receptor e à responsividade do fotodiodo, de acordo com as equações 4.37 e 4.38. O sinal na saída do fotodiodo é distorcido pelo ruído balístico gerado dentro do componente, modelado como gaussiano e com variância dada pela equação 4.42. Os valores da capacitância do fotodiodo, C_d , corrente de escuro não multiplicativa, $I_{obs_{NM}}$, corrente de escuro multiplicativa, I_{obs_M} , eficiência quântica, η , taxa de ionização, k_{ef} e resistência de polarização, R_p , armazenados no arquivo *fotod.std*, são mostrados na tabela 5.9.

	C_d (F)	$I_{obs_{NM}}$ (A)	I_{obs_M} (A)	η	k_{ef}	R_p (Ω)
PIN	5e-12	1e-11	0	0.5	0.1	1e6
APD	1e-12	1e-12	1e-11	0.5	0.01	1e5

Tabela 5.9: Parâmetros para cálculo do sinal na saída do fotodiodo

A rotina é chamada com quatro parâmetros: *opção*, que indica o tipo de fotodiodo utilizado, *namost*, que informa o número de amostras do sinal geradas, *tbits*, que indica a taxa de transmissão de dados (Mbit/s) e *lambda*, que indica o comprimento de onda da fonte de luz.

Ao final da rotina são gerados dois arquivos com dados representando o sinal na saída do fotodiodo: *fotod.dat*, que representa o sinal sem a adição de ruído e *fotru.dat*, que corresponde ao sinal distorcido pela presença do ruído balístico. Além disso é gerado o arquivo *balist.dat* que representa o ruído gerado pelo fotodiodo.

O ruído balístico segue um processo de Poisson, porém resultados práticos e análises técnicas conhecidas na literatura indicam que a aproximação de se tomar o ruído balístico como sendo gaussiano apresenta resultados satisfatórios [5][6]. Assim, em nosso software modelamos o ruído balístico com distribuição gaussiana.

5.8 Pré-amplificador

O pré-amplificador é implementado pela rotina *fc-preamp.c*, que gera valores representando o sinal na saída do dispositivo. Este sinal corresponde ao sinal na saída do fotodiodo amplificado e somado ao ruído térmico gerado no próprio pré-amplificador, que apresenta uma distribuição gaussiana. O modelamento é feito considerando todas as fontes de ruído na entrada do pré-amplificador. Assim, o ruído que no componente real é gerado na saída do pré-amplificador deve ser compensado de forma a anular o efeito que produz no sinal pelo modelo. O modelamento desta compensação pode ser observado na equação 4.66, onde

o termo S_{e_a} , correspondente à densidade espectral da corrente de ruído na saída do pré-amplificador é dividido por R_T^2 , que representa as alterações sofridas pelo sinal devido aos componentes do pré-amplificador.

O ganho proporcionado ao sinal pelo pré-amplificador é fornecido pelo usuário no momento da simulação. Os outros valores utilizados no cálculo do ganho e do ruído, recuperados do arquivo **preamp.std** e apresentados na Tabela 5.10 são: transcondutância do FET, gm , resistência de entrada do transistor, R_a , resistência de contra-reação, R_f , capacitância de entrada do pré-amplificador, C_a , ganho de corrente do transistor, β e temperatura, T. Além disso é necessário também recuperar o valor da capacitância do fotodiodo, C_d , do arquivo de parâmetros do fotodiodo, **fotod.std**, cujos valores são mostrados na Tabela 5.9. Para os pré-amplificadores do tipo bipolar o valor da transcondutância, gm , não é lido do arquivo, mas calculado através da equação 4.56.

No caso dos pré-amplificadores de alta-impedância, a resistência de realimentação (R_f) é infinita, conseqüentemente o valor da densidade espectral da corrente de ruído da resistência de contra-reação é igual a zero.

A rotina é chamada com quatro parâmetros: *opção*, que indica o tipo de pré-amplificador selecionado, *opção2*, que define o tipo de fotodiodo utilizado, *tbits*, que indica a taxa de transmissão de dados (Mbit/s) e *namost*, que informa o número de amostras geradas do sinal. Ao final da rotina são gerados dois arquivos: **preamp.dat**, que corresponde ao sinal da saída do pré-amplificador com os ruídos térmico e balístico, e **térmico.dat**, que corresponde aos valores do ruído térmico gerado pelo pré-amplificador.

	$gm(S)$	$R_a(\Omega)$	$C_a(F)$	β	$i_B(A)$	T(K)
trans.bipolar	0	20e3	10e-12	50	5e-6	300
trans.FET	5e-3	-	5e-12	0	0	300
alta bipolar	0	20e3	10e-12	50	5e-6	300
alta FET	5e-3	-	5e-12	0	0	300

Tabela 5.10: Parâmetros para cálculo do sinal na saída do pré-amplificador

5.9 Filtro

Os valores que correspondem à função de transferência do filtro são gerados através da função *fc-filtro.c*. O usuário tem dois tipos de filtro a sua escolha: *Butterworth* e *Cosseno levantado*. Para cada um deles existe ainda a opção de *equalizante*, que compensa as distorções causadas pelo transmissor, pela fibra óptica e pelo pré-amplificador durante a transmissão do sinal; e *não-equalizante*, que compensa apenas a distorção introduzida pelo pré-amplificador. Os valores dos parâmetros necessários ao cálculo da função de transferência do filtro são fornecidos pelo usuário no momento da simulação. Para o filtro Butterworth é pedida a ordem do filtro (de 1 a 4) e para o filtro cosseno levantado é pedido o fator de ocupação do intervalo (entre 0 e 1).

No caso dos filtros do tipo equalizante são recuperados os valores correspondentes à função de transferência da fibra, no arquivo **fibra.trf**, do transmissor, no arquivo **transm.trf**, e do pré-amplificador, no arquivo **preamp.trf**, que são divididos pela função de transferência

do filtro. No caso dos filtros do tipo não equalizante a função de transferência do filtro é dividida apenas pelos valores da função do pré-amplificador.

A rotina é chamada com três parâmetros: *opção*, que informa o tipo de filtro selecionado, *namost*, que indica o número de amostras geradas do sinal e *tbits*, que indica a taxa de transmissão de dados (Mbit/s). Ao final da rotina é gerado um arquivo denominado **receptor.trf** contendo os valores correspondentes à função de transferência do filtro.

5.10 Detector de Sinal

O detector de sinal é implementado pela rotina *detecta.c*, que verifica o sinal na saída do filtro, recuperando uma amostra por período de bit e decidindo sobre o valor a ser regenerado (0 ou 1). A rotina é chamada quando o usuário seleciona a opção de simulação ou de análise, com quatro parâmetros: *nome*, que indica o nome do arquivo que contém os valores representando o sinal na saída do filtro, utilizado quando a rotina é acionada através da opção de simulação; *an-sinal* e *ani-sinal*, representando o endereço dos vetores que contêm as partes real e imaginária, respectivamente, dos valores que representam o sinal a ser processado, utilizados quando a rotina é ativada a partir da opção de análise; e *namost*, que informa o número de amostras do sinal geradas.

Quando a rotina é ativada a partir da opção de simulação, os valores que representam o sinal na saída do filtro, armazenados no arquivo **filtro.dat**, são recuperados e colocados em dois vetores: um contendo a parte real e outro a parte imaginária. O módulo destes valores é calculado e o valor que corresponde à amplitude do sinal no meio do intervalo de um bit é recuperado e colocado em outro vetor, denominado *detec*. São também recuperados os valores da maior e menor amplitudes do sinal, que servirão para determinar o limiar acima do qual o bit é regenerado como 1. Ao final do processamento é gerado um arquivo denominado **saida.dat** contendo o sinal regenerado na saída do circuito de decisão.

Quando a rotina é ativada a partir da opção de análise da taxa de erros do sistema, os valores que correspondem às partes real e imaginária do sinal na saída do filtro já estão armazenados em dois vetores, e o restante do procedimento é idêntico. Os valores regenerados ficam armazenados em um vetor denominado *an-sinal*.

5.11 Medidor de erros

O medidor de erros é implementado por duas rotinas *med-erro.c* e *compara.c*. A primeira é ativada quando a opção de análise da taxa de erros simulada do sistema é selecionada e a segunda é ativada durante a simulação do sistema.

A rotina *med-erro* é responsável por ativar os módulos que executam as funções de geração de sinais, FFT e funções de transferência de todos os componentes, além da rotina de detecção do sinal, antes de passar pelo módulo que faz a contagem de erros, propriamente dita. Nesta opção todos os valores correspondentes ao sinal nos diversos pontos do sistema são armazenados em vetores. Por questão de disponibilidade de memória os módulos são acionados diversas vezes até que o número de amostras selecionado pelo usuário (10^3 , 10^4 , 10^5 ou 10^6) seja processado. Esta rotina tem dois parâmetros de entrada: *namo*, que indica o número de amostras a serem geradas e *opção*, um vetor que informa quais os componentes selecionados do sistema de comunicação. Nenhum arquivo é gerado e ao final do processamento

a rotina informa o número de amostras e o total de erros encontrado.

A rotina *compara.c* é ativada quando a opção de simulação de sistema é selecionada. Esta rotina é responsável por comparar o arquivo que contém as amostras de entrada com o arquivo que contém o sinal regenerado na saída do sistema e computar os erros encontrados. A rotina tem três parâmetros de entrada: *nome1*, que indica o arquivo contendo os valores correspondentes ao sinal de entrada, *nome2*, que indica o arquivo contendo os valores correspondentes ao sinal regenerado na saída, e *namost*, informando o número de amostras geradas. Neste caso também nenhum arquivo é gerado e após o término do processamento a rotina informa o número de amostras e o número de erros encontrados.

5.12 Ruído gaussiano

A rotina que implementa o ruído gaussiano, *ruído-br.c*, gera ruído com distribuição gaussiana e desvio-padrão especificado, calculado através dos modelos de ruído apresentados no capítulo 4. Os valores gerados irão simular o ruído balístico, introduzido pelo fotodiodo, e o ruído térmico, pelo pré-amplificador.

A rotina é ativada com seis parâmetros: *namost*, que indica o número de amostras de ruído a serem geradas, *tbits*, que indica a taxa de transmissão de dados do sistema (Mbit/s), *desvio-padrão*, que especifica o desvio-padrão a ser utilizado na geração do ruído, *an-ruído*, vetor utilizado apenas quando a opção é de análise da taxa de erros, e *nome*, que corresponde ao nome do arquivo de ruído a ser gerado quando a opção é simulação.

O ruído é gerado através de uma função interna do TURBOC que gera números randômicos com distribuição uniforme. Estes valores são processados de forma a assumirem uma distribuição gaussiana e em seguida são armazenados em um vetor na memória ou gravados em um arquivo cujo nome foi definido em um dos parâmetros de entrada.

5.13 Conclusão

Neste capítulo foram apresentadas as rotinas que compõem a biblioteca de modelos do software, implementando os diversos componentes de um sistema de comunicação disponíveis para simulação. As rotinas foram apresentadas através da descrição de suas entradas, saídas e processamento sobre os dados.

Referências Bibliográficas

- [1] Lobão P. M. S. e Moschim E., "PC-SIMFO: A software for analysis and design of optical fiber communication systems", IV Canadian Conference on Electrical and Computer Engineering", 1991.
- [2] Lobão P. M. S. e Moschim E., "PC-SIMFO: Um software para simulação de sistemas de comunicação usando fibras ópticas", IX Simpósio Brasileiro de Telecomunicações, 1991.
- [3] Lobão P. M. S. e Moschim E., "PC-SIMFO: Simulador de sistemas de comunicação por fibras ópticas", aceito para publicação no X Simpósio Brasileiro de Telecomunicações, 1992.

-
- [4] Lobão P. M. S. e Moschim E., "PC-SIMFO: Simulation of optical fiber communication systems", aceito para apresentação no Bilcon'92, 1992.
- [5] D. P. Schinke, R. G. Smith e A. R. Hartman, "Photodetectors", cap 3, *Semiconductor devices for optical communication - Topics in applied physics*, editado por H. Kressel, Springer-Verlag, 1980.
- [6] H. Waldman, C. Bezzan e E. Moschim, "Cálculo geral de desempenho de receptores ópticos em sistemas de transmissão digital (aproximação gaussiana com a ocorrência da interferência entre símbolos devido à fibras dispersivas", Contrato FEC-UNICAMP/TELEBRÁS 033/80, Nota Técnica 7, 1983.

Capítulo 6

Aplicação do PC-SIMFO

Neste capítulo são apresentados exemplos de utilização do PC-SIMFO na simulação de algumas configurações sistêmicas. São descritas as características de cada uma das configurações e são mostrados gráficos representando o sinal em diversos pontos do sistema. São apresentadas também comparações entre o desempenho de diversos componentes optoeletrônicos.

6.1 Introdução

Uma das maiores vantagens do PC-SIMFO é permitir a simulação de diferentes configurações sistêmicas com grande simplicidade, possibilitando a comparação entre elas e auxiliando na escolha da mais adequada. O usuário tem a sua disposição também alguns parâmetros de desempenho que podem ser utilizados para analisar componentes individuais. Neste capítulo iremos apresentar alguns resultados obtidos na avaliação de algumas configurações e suas aplicações, assim como uma avaliação dos componentes existentes na biblioteca de modelos.

6.2 Análise de desempenho de um sistema de comunicação

O PC-SIMFO oferece a possibilidade de avaliação do desempenho de um sistema de comunicação utilizando os componentes atualmente disponíveis no mercado, através de simulação. Nesta seção apresentaremos a avaliação de duas tecnologias de fibras ópticas usadas para sistemas com capacidade de até 100 Mbits/s.

6.2.1 Tecnologia de fibras ópticas de plástico

Na escolha de um sistema de comunicação por fibra óptica, além das características de desempenho, uma consideração a ser levada em conta é o custo dos componentes. Atualmente, a opção mais barata compreende a utilização de um LED como transmissor óptico, uma fibra óptica de plástico como canal e um receptor formado por um fotodiodo do tipo PIN, um pré-amplificador do tipo transimpedância com transistor FET e um filtro butterworth do tipo não-equalizante, conforme especificado na tabela 6.1.

Configuração 1	
• Transmissor:	LED 570 nm
• Canal:	Fibra multimodo de plástico (FOP-PMMA)
• Receptor:	Fotodiodo PIN Pré-amplificador transimpedância com transistor FET Filtro tipo butterworth de ordem 2 não equalizante

Tabela 6.1: Configuração sistêmica de baixo custo para transmissão à curta distância

Na análise desta configuração estamos interessados em verificar qual é a máxima taxa de transmissão possível. A simulação foi executada repetidas vezes e a abertura de olho foi anotada. Considerando que o sistema admite uma abertura de olho de até 0.5 para manter uma probabilidade de erro razoável, a taxa de transmissão máxima para este tipo de fonte de luz é de 10 Kbit/s, como pode ser observado na figura 6.1, que apresenta a variação da abertura de olho na saída do transmissor em função da taxa de transmissão.

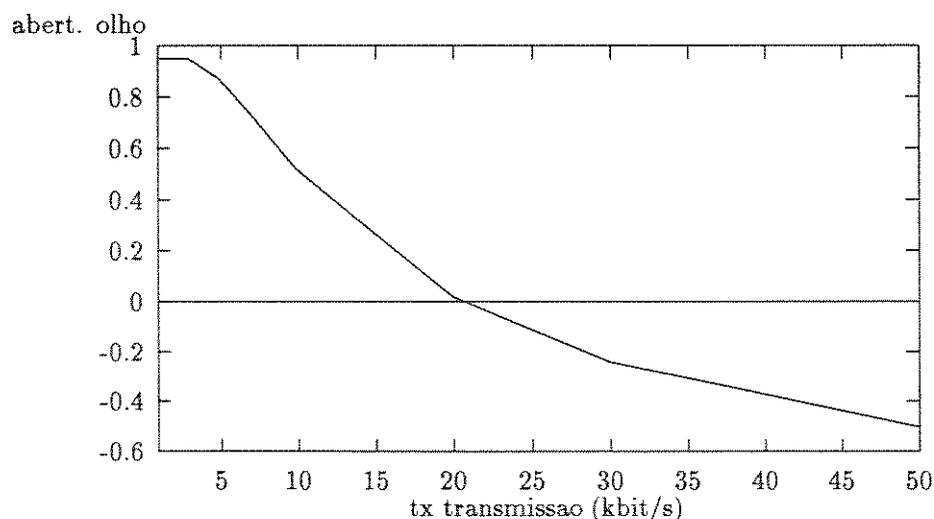
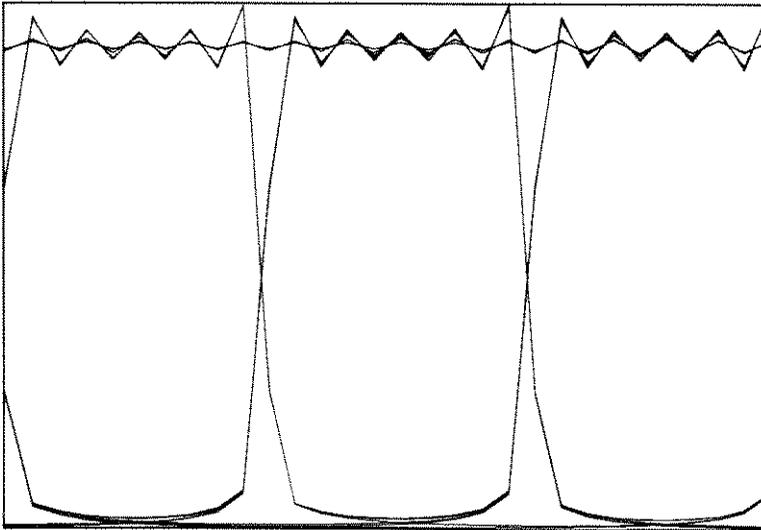
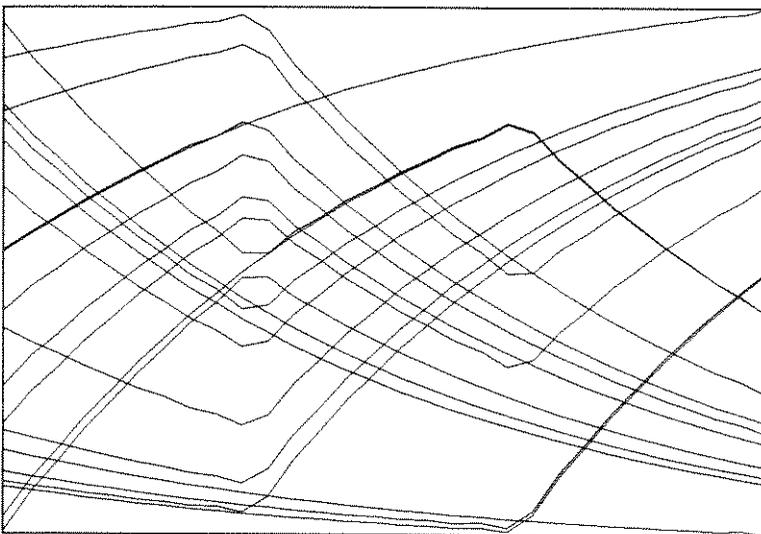


Figura 6.1: Variação da abertura de olho em função da taxa de bits

Como exemplo da degradação do sinal devido à resposta em frequência destes LEDs, a figura 6.2 permite uma visualização do diagrama de olho na saída do transmissor para as taxas de transmissão $T_b = 1$ Kbit/s e $T_b = 50$ Kbit/s.

Uma forma de aumentar a taxa de transmissão sem acarretar aumento significativo nos custos para este tipo de sistema, é fazer a substituição do LED 570nm por outro com comprimento de onda igual a 650 nm. A tabela 6.2 mostra a nova configuração do sistema.

(a) $T_b = 1 \text{ kbit/s}$ (b) $T_b = 50 \text{ kbit/s}$ **Figura 6.2:** Diagrama de olho para taxas de transmissão $T_b = 1 \text{ Kbit/s}$ e $T_b = 50 \text{ Kbit/s}$

Configuração 2

- Transmissor: LED 650 nm
- Canal: Fibra multimodo de plástico (FOP-PMMA)
- Receptor:
 - Fotodiodo PIN
 - Pré-amplificador transimpedância com transistor FET
 - Filtro tipo butterworth de ordem 2 não equalizante

Tabela 6.2: *Configuração sistêmica para LED 650 nm*

A simulação deste sistema, executada considerando uma corrente de pré-polarização de 5 mA, uma corrente de modulação de 50 mA, uma taxa de transmissão de 100 Kbit/s, usando uma fibra com comprimento de 250 m e um amplificador com ganho 10, pode ser acompanhada através da seqüência de gráficos da figura 6.3, onde podemos observar o sinal gerado, na saída do transmissor, na saída da fibra, na saída do fotodiodo, na saída do pré-amplificador e na saída do filtro receptor.

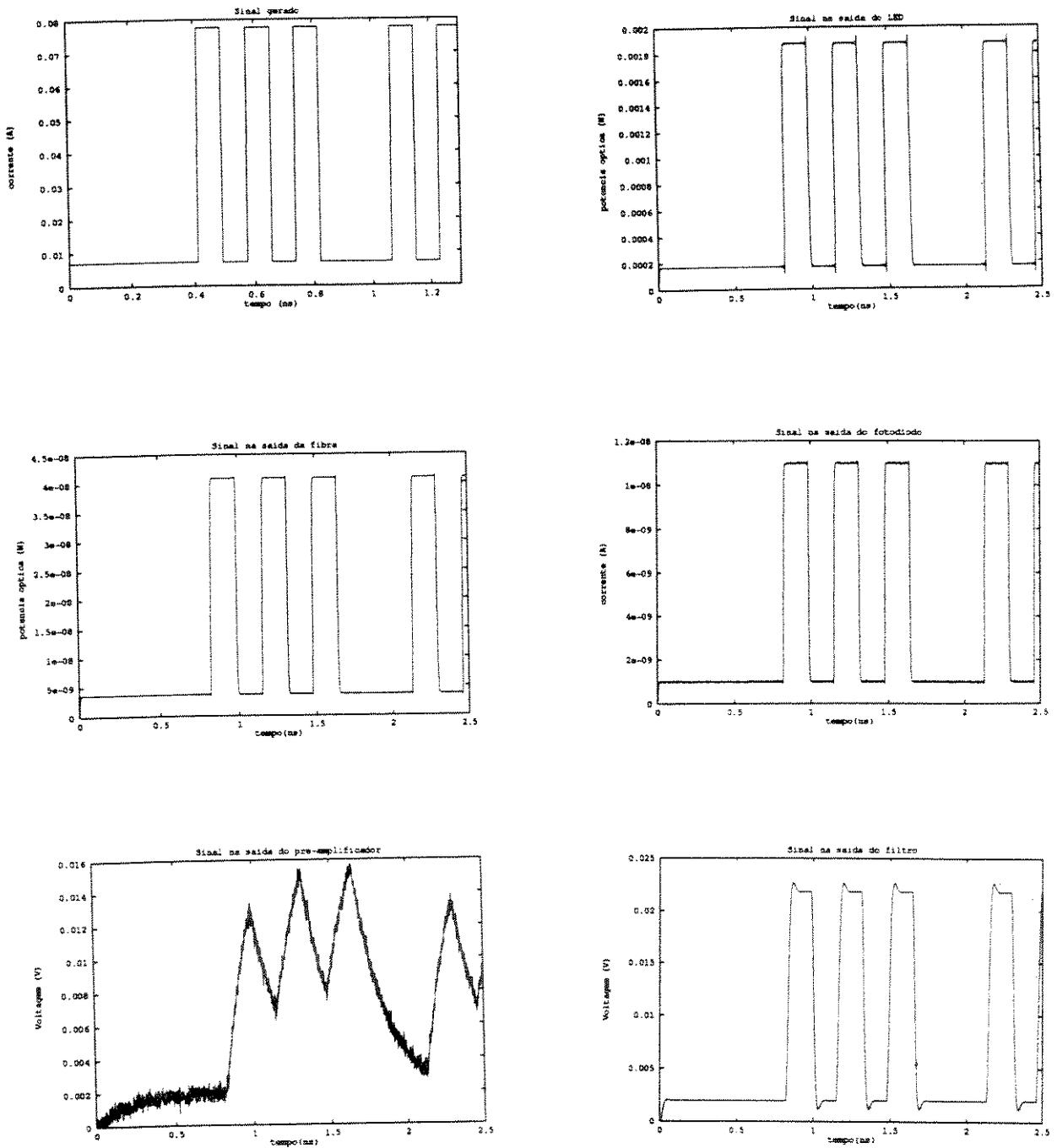


Figura 6.3: Sinal em diversos pontos do sistema com a configuração 2

6.2.2 Tecnologia de fibras ópticas de sílica

A utilização da tecnologia de fibra de plástico limita a taxa de transmissão e a distância entre os pontos. Podemos melhorar razoavelmente a qualidade do sistema substituindo a fibra óptica de plástico por uma fibra de sílica. A tabela 6.3 apresenta a configuração adotada para uma transmissão a curta distância usando a tecnologia de fibra óptica de sílica.

Configuração 3	
• Transmissor:	LED 850 nm
• Canal:	Fibra multimodo de índice degrau de sílica
• Receptor:	Fotodiodo PIN Pré-amplificador alta-impedância com transistor bipolar Filtro tipo butterworth de ordem 3 equalizante

Tabela 6.3: Configuração sistêmica para transmissão à curta distância usando tecnologia de fibra óptica de sílica

A figura 6.4 apresenta as formas de onda do sinal gerado, na saída do transmissor, na saída da fibra, na saída do fotodiodo, na saída do pré-amplificador e na saída do filtro receptor. A configuração foi testada considerando uma corrente de pré-polarização de 5 mA, uma corrente de modulação de 50 mA, uma taxa de transmissão de 1 Mbit/s, com uma fibra de 1 km comprimento e um amplificador com ganho 10.

A qualidade do sistema pode ser melhorada sensivelmente através de uma configuração que utiliza como transmissor um diodo laser e como canal uma fibra multimodo de índice gradual de sílica. Esta configuração permite uma transmissão na faixa de GHz a distâncias de dezenas de quilômetros. O laser multimodo com comprimento de onda de 850 nm apresenta um desempenho razoável a um baixo custo.

Configuração 4	
• Transmissor:	laser 850 nm
• Canal:	Fibra multimodo de índice gradual de sílica
• Receptor:	Fotodiodo APD Pré-amplificador alta-impedância com transistor bipolar Filtro tipo cosseno levantado equalizante com $\alpha = 0.5$

Tabela 6.4: Configuração sistêmica para transmissão à longa distância usando tecnologia de fibra óptica de sílica

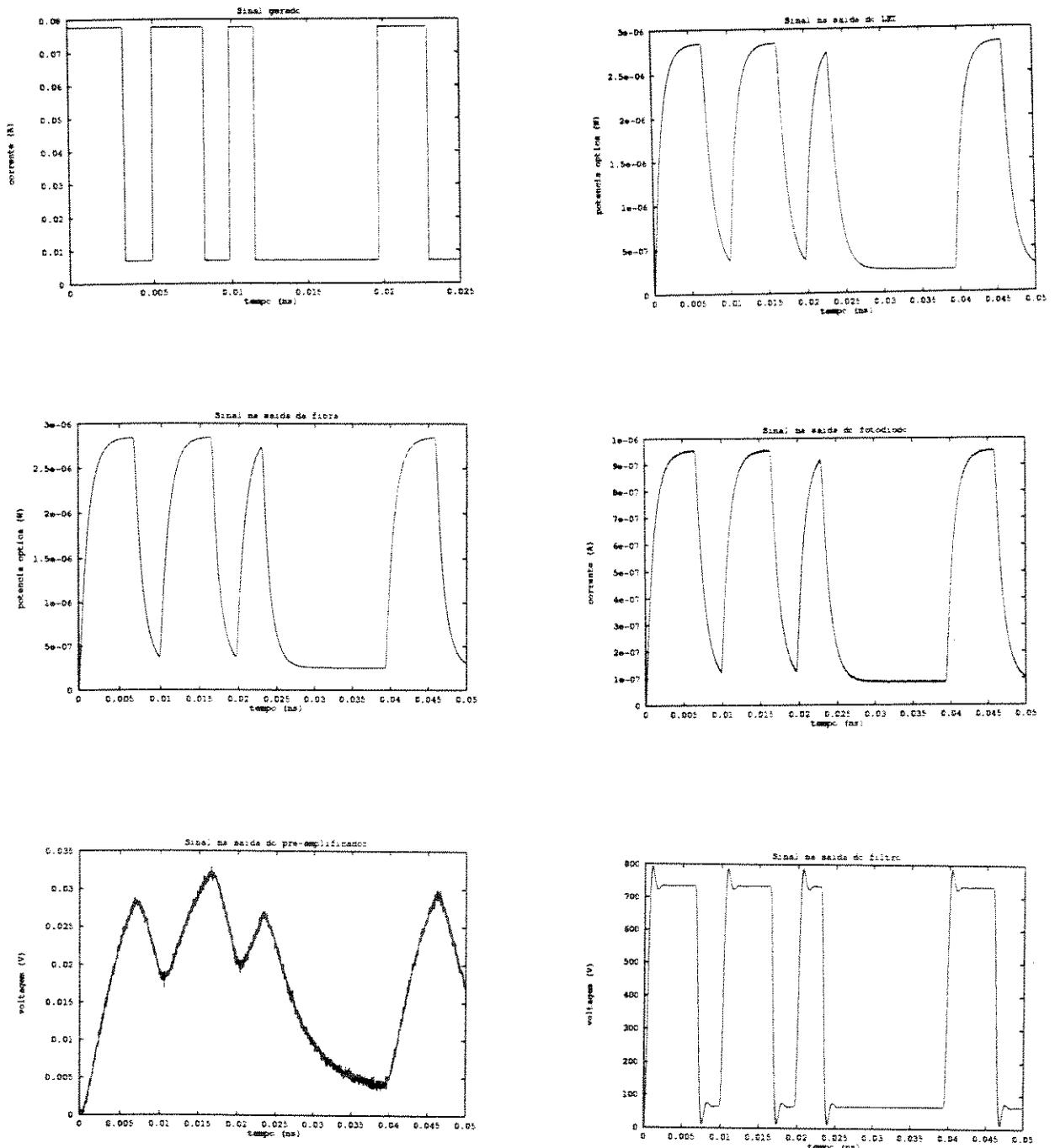


Figura 6.4: Sinal em diversos pontos do sistema com a configuração 3

Podemos observar, através da seqüência de gráficos da figura 6.5 que, mesmo aumentando a distância entre os pontos para 10 km, e aumentando a taxa de transmissão para 100 Mbit/s, esta configuração apresenta um desempenho maior que as anteriores. Os gráficos apresentados correspondem ao sinal gerado, modulado na saída do laser, na saída da fibra, na saída do fotodiodo, na saída do pré-amplificador e na saída do filtro receptor. A configuração foi testada utilizando uma corrente de pré-polarização equivalente à 20% acima da corrente de limiar, uma corrente de modulação de 25 mA, um fotodiodo com ganho de avalanche de 100 e um pré-amplificador com ganho 10.

Todos os componentes presentes na biblioteca de modelos podem ser combinados na forma desejada pelo usuário. Sendo muito grande o número de configurações possíveis, nos limitamos aqui a apresentar uma seleção de opções a título de exemplo.

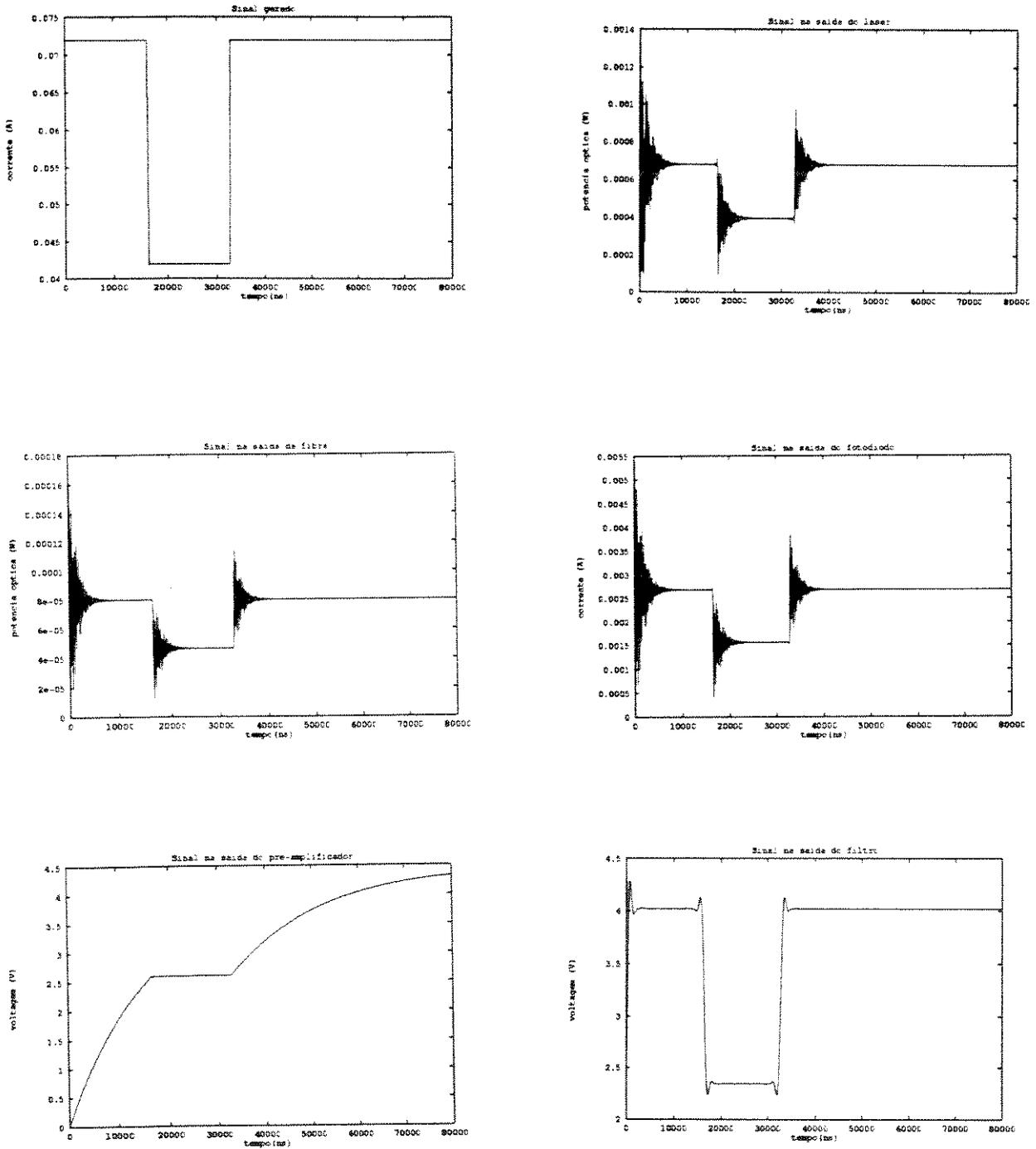


Figura 6.5: Sinal em diversos pontos do sistema com a configuração 4

6.3 Análise de desempenho de componentes

O PC-SIMFO oferece também a possibilidade de avaliação de componentes individuais, como mostraremos nesta seção. Neste caso, se o usuário quiser verificar o desempenho de um componente isolado, alterando parâmetros e projetando um novo componente, o sistema oferece uma série de opções para análise de desempenho.

Para o transmissor óptico estão disponíveis as opções de potência emitida, distribuição espectral e resposta em frequência. Estes parâmetros podem ser comparados entre os diversos tipos de LED ou laser, como mostrado a seguir.

Na figura 6.6 temos as curvas de potência emitida para os diversos tipos de LED.

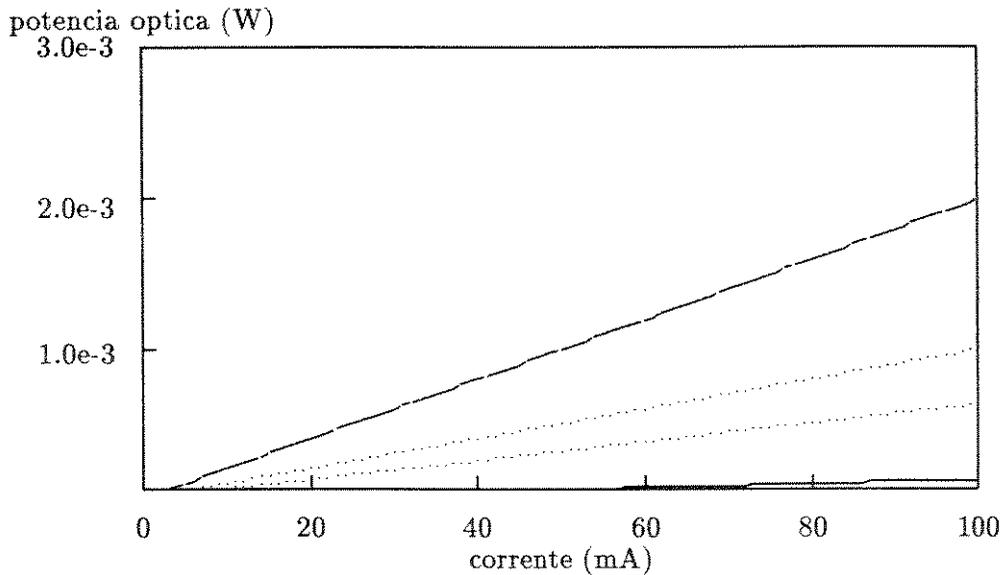


Figura 6.6: Potência emitida do LED

A figura 6.7 apresenta as curvas que representam a distribuição espectral para os diferentes tipos de LED.

A comparação entre as curvas de resposta em frequência pode ser feita através da observação das curvas apresentadas na figura 6.8.

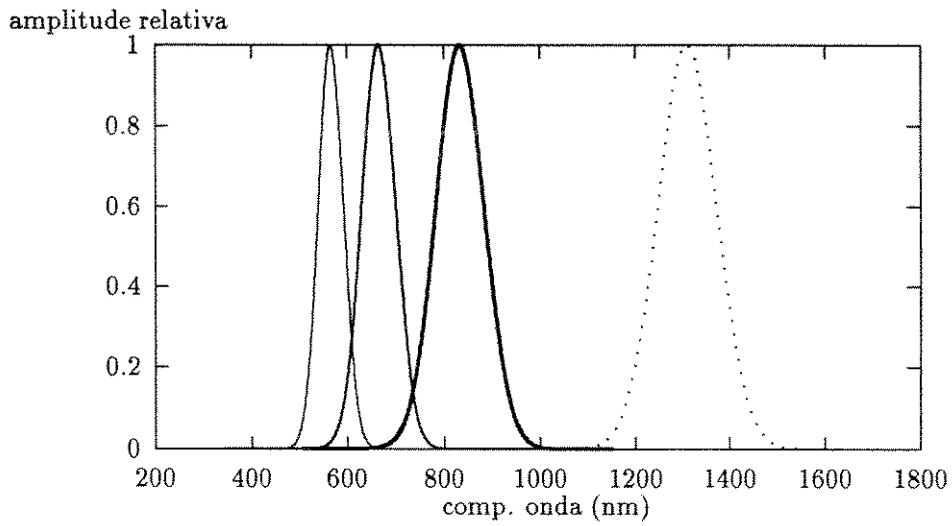


Figura 6.7: Distribuição espectral do LED

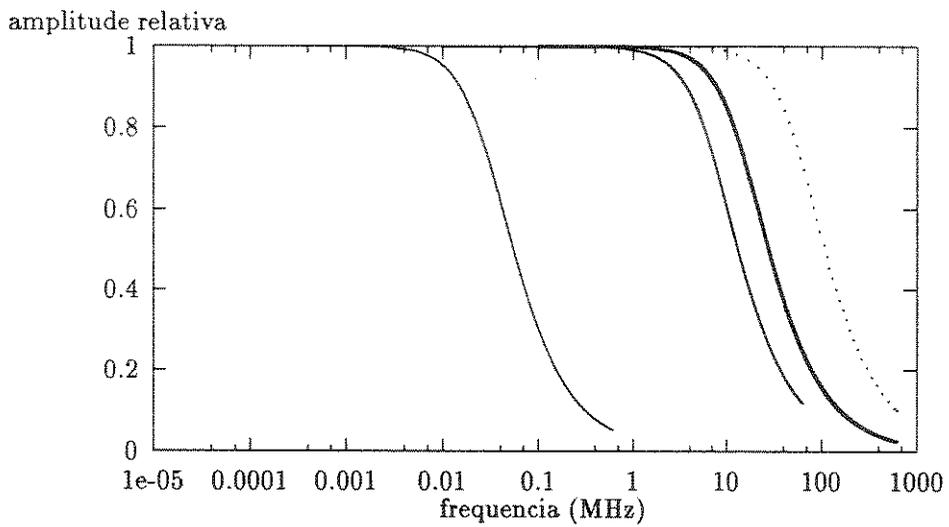


Figura 6.8: Resposta em frequência do LED

Na figura 6.9 são apresentadas as curvas de potência emitida em função da corrente injetada para os diferentes tipos de laser. O ponto onde ocorre a mudança na curva corresponde à corrente a partir da qual a emissão do laser deixa de ser espontânea e passa a ser estimulada, ou seja, a corrente de limiar. Este valor varia de acordo com o comprimento de onda do laser.

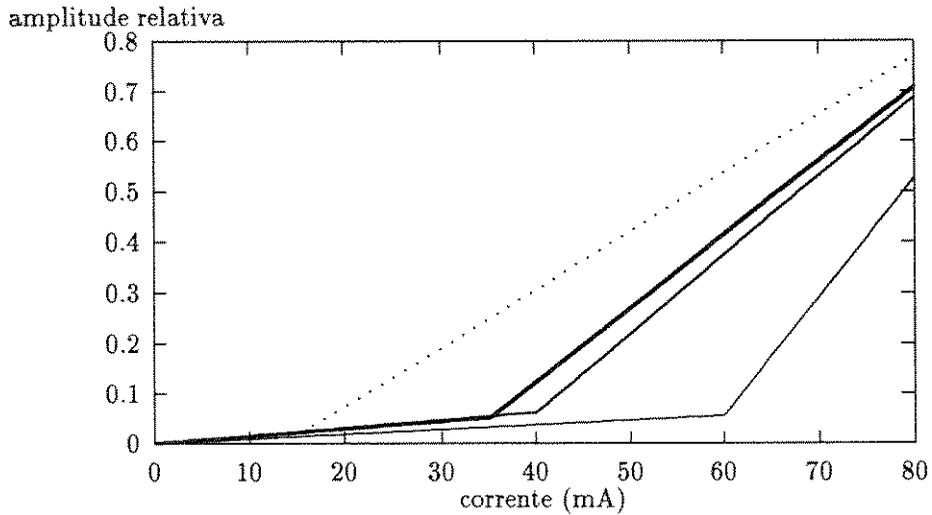


Figura 6.9: Potência emitida do laser

As curvas que representam a distribuição espectral para os diversos tipos de laser são mostradas na figura 6.10.

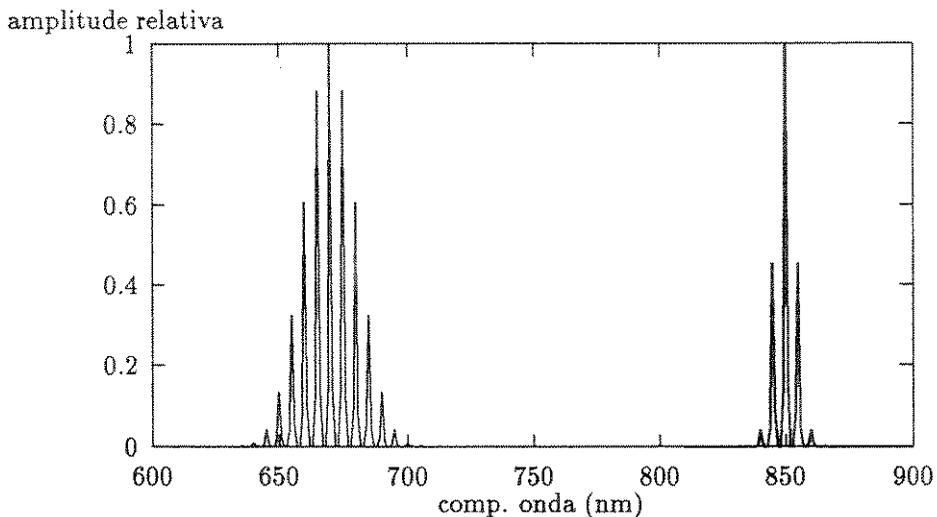


Figura 6.10: Distribuição espectral do laser com $\lambda = 670\text{nm}$ e 850 nm

A comparação entre as curvas de resposta em frequência pode ser feita através da observação das curvas apresentadas na figura 6.11.

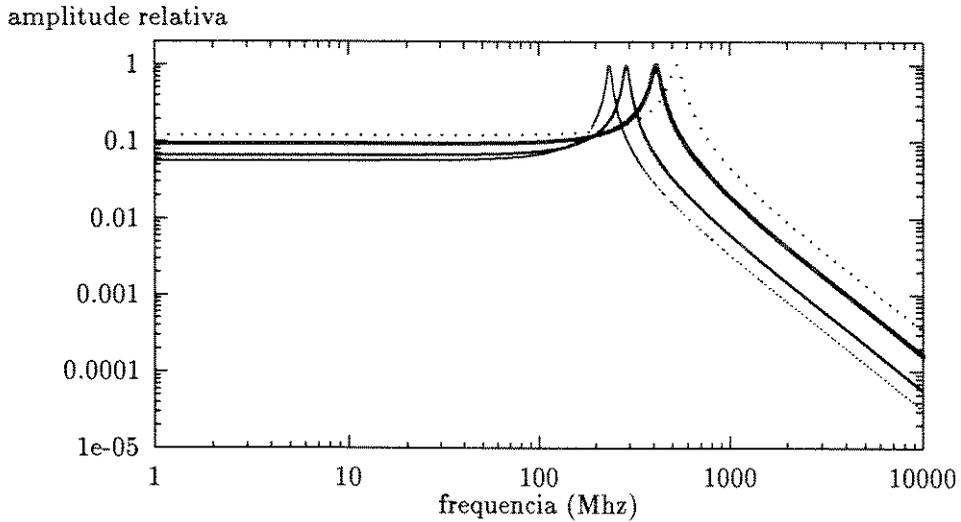


Figura 6.11: *Resposta em freqüência do laser*

Para a análise da fibra estão disponíveis as opções de resposta em freqüência e de curvas de atenuação. Na figura 6.12 temos as curvas de resposta em freqüência para os diversos tipos de fibra.

A figura 6.13 apresenta a curva de atenuação total da fibra óptica multimodo de índice degrau de plástico.

A análise do receptor pode ser feita através de dois parâmetros: resposta em freqüência e ruído. A resposta em freqüência do receptor depende fundamentalmente do filtro. Para filtros do tipo butterworth, a resposta depende da ordem do filtro e de sua banda passante. A figura 6.14 mostra as diferenças entre as respostas em freqüência para filtros butterworth de ordem 1, 2, 3 e 4, considerando a mesma banda passante.

Em filtros do tipo cosseno levantado, a resposta em freqüência depende do fator de ocupação do intervalo, que varia de 0 a 1. Na figura 6.15 são apresentadas as curvas de resposta para os filtros com fator de ocupação igual a 0, 0,5 e 1.

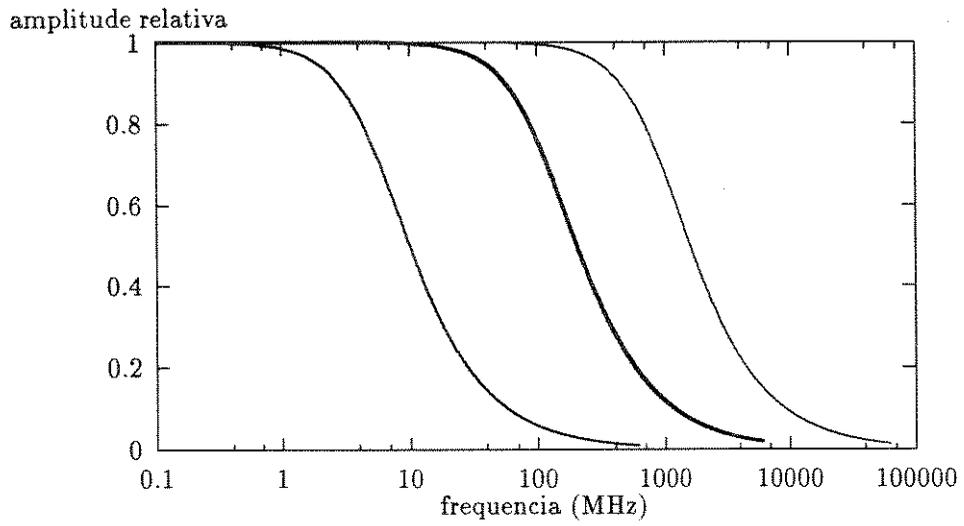


Figura 6.12: Resposta em frequência da fibra

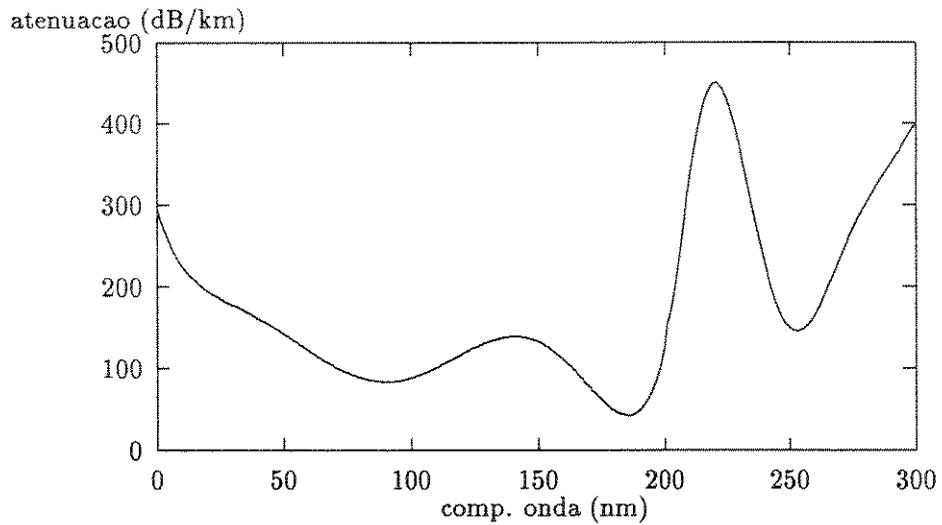


Figura 6.13: Atenuação total da fibra de plástico

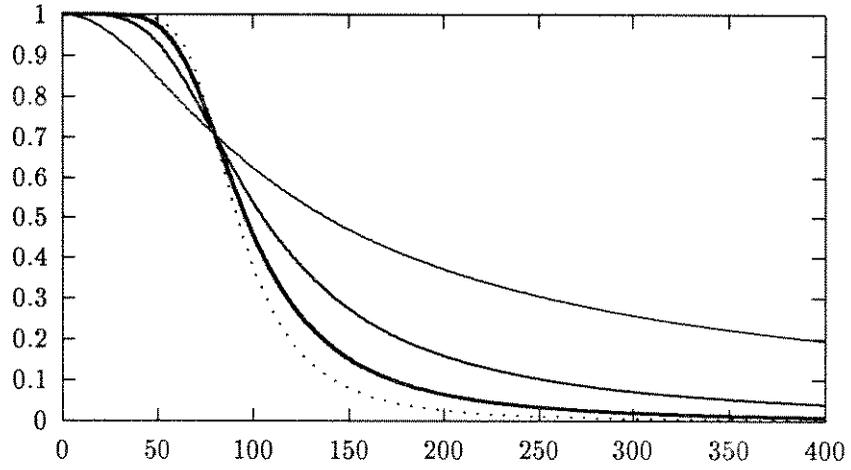


Figura 6.14: Resposta em frequência do filtro butterworth

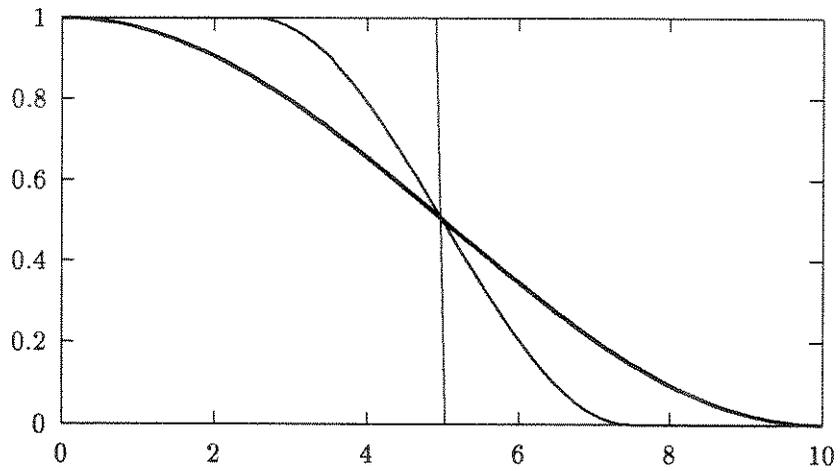


Figura 6.15: Resposta em frequência do filtro cosseno levantado

6.4 Conclusão

Neste capítulo foram apresentados alguns exemplos de utilização do software PC-SIMFO. Foram analisadas algumas configurações sistêmicas e alguns tipos de componentes optoeletrônicos. O objetivo deste capítulo foi mostrar a potencialidade do simulador. Diversas outras opções são possíveis, o que possibilita ao usuário uma análise ampla do sistema em estudo.

Conclusão Geral

Diversas vantagens aparecem quando escolhe-se a simulação como ferramenta de projeto e análise, especialmente em sistemas baseados em PCs. Considerações importantes são o custo e a acessibilidade do computador, a precisão na simulação, a velocidade de processamento e as capacidades da linguagem de simulação. O PC satisfaz vários destes pontos: é acessível a um grande número de pessoas, os custos de compra e operação são baixos, além de permitir que a simulação seja realizada a uma velocidade razoável.

A proliferação do uso de microcomputadores em simulação acarretou o aparecimento de diversos pacotes de software visando uma maior precisão na simulação. Nestes pacotes, para se avaliar o desempenho de uma transmissão, amostras de sinais são geradas e processadas através de modelos dos blocos funcionais do sistema, implementados na forma de rotinas com objetivos específicos. Independente da abordagem e da linguagem de programação utilizada, pode-se notar que o computador tem se tornado uma ferramenta útil e efetiva na simulação de sistemas de comunicação, permitindo ao projetista uma avaliação mais detalhada do desempenho do sistema sob análise, possibilitando um bom dimensionamento dos componentes do sistema projetado com menor esforço.

O crescente uso de simulação como um auxílio no projeto e análise de sistemas de comunicação é uma indicação da confiança nos resultados de simulação e a importância da simulação como uma técnica de solução de problemas. Neste contexto a simulação deve se tornar uma ferramenta de fácil utilização para usuários comuns de microcomputadores através de softwares amigáveis para todos os estágios de simulação, incluindo a definição, desenvolvimento e validação dos modelos de simulação.

Com o PC-SIMFO, a possibilidade de se testar diversas configurações em um curto intervalo de tempo torna claro ao usuário os custos e benefícios de utilizar diferentes componentes optoeletrônicos, facilitando a tarefa de projetar sistemas de comunicação com um baixo custo operacional. O usuário pode, dentro de um sistema de comunicação definido, mudar um ou vários componentes e observar as alterações (distorções) que ocorrem no sinal, encontrando assim a melhor opção. Por exemplo, em um sistema de curta distância pode-se observar que a troca de um laser por um LED, bem mais barato, não afeta de modo significativo o sinal transmitido. O usuário pode também fazer testes em um componente isolado, mudando alguns dos parâmetros físicos e criando um novo componente.

Dentro deste contexto foram apresentadas as características dos elementos necessários para a configuração de um sistema de comunicação usando fibras ópticas, através de seus parâmetros físicos e tecnológicos, assim como um modelamento de suas funções de transferência. Com base neste modelamento foi calculado o desempenho de um sistema de transmissão digital e feita uma análise de diversos componentes optoeletrônicos.

Nesta primeira versão do PC-SIMFO foram implementadas todas as funções referentes ao menu de definição de parâmetros (*componentes*) e as funções do menu de *simulação* dos sub-

menus *componentes* e *sistema digital ponto-a-ponto*. Porém toda a estrutura de menus já está pronta para que as outras opções possam ser integradas ao sistema sem maiores esforços além da programação das próprias rotinas. Isto porque, no desenvolvimento do PC-SIMFO, um dos princípios básicos foi fazer o sistema o mais modular possível, de modo a facilitar a expansão, isto é, a incorporação de novos módulos; e as modificações e melhoramentos nos módulos já implantados.

O PC-SIMFO se apresenta como uma ferramenta bastante atrativa no auxílio ao projeto e análise de sistemas de comunicação usando fibras ópticas. Cabe ressaltar, porém, que a implantação de um sistema de comunicação real requer estudos adicionais, incluindo estudos de protocolos, utilização de multiplexadores, acopladores e conectores, entre outros.

Esta linha de trabalho oferece diversas opções de continuação, tais como:

- O desenvolvimento de novos modelos, com componentes recentemente desenvolvidos à disposição no mercado, além da inclusão de conectores e lentes, etc;
- O desenvolvimento de novos modelos de análise dos componentes, como margem e perda de potência, perdas por conectores, etc;
- A simulação e análise de sistemas de comunicação multiponto, com várias configurações, tipo anel, estrela, barramento, etc;
- A simulação e análise de sistemas analógicos;
- A utilização de processamento paralelo visando melhorar o tempo de simulação;
- A inclusão de modelos diferentes tipos de ruído (no nosso caso usamos apenas o modelo gaussiano), aumentando a correspondência com os modelos de ruído dos fotodiodos de avalanche;
- A introdução de uma biblioteca de definições dos termos utilizados no software, na forma de um tutorial;
- A construção de uma versão do software para estações de trabalho.

Apêndice A

Modelo de sinal para potência emitida

Neste apêndice apresentamos o modelo de sinal no domínio do tempo que relaciona a corrente injetada com o sinal na saída do gerador.

No nosso caso a corrente injetada que modula a fonte de luz é dada por:

$$i_d(t) = I_{pre} + \sum_{k=-\infty}^{+\infty} a_k g(t + kT_s) \quad (\text{A.1})$$

onde:

I_{pre} = corrente de pré-polarização (A)

a_k = seqüência de dados na saída do gerador (0,1)

$g(t)$ = formato do pulso na saída do codificador

T_s = janela de tempo

No nosso caso consideramos $g(t)$ como um sinal retangular de amplitude i_m , dado por:

$$g(t) = \begin{cases} i_m & \text{se } 0 \leq t \leq T, \quad T \leq T_s \\ 0 & \text{caso contrário} \end{cases}$$

onde:

$i_m(t)$ = corrente de modulação (A)

T = tempo de ocupação do pulso na janela de tempo T_s ($T = T_s$ para código NRZ)

A corrente de pré-polarização é usual para melhorar o tempo de resposta das fontes de luz. Entretanto, ela introduz no receptor uma penalidade devido a uma taxa de extinção diferente de zero. A taxa de extinção é definida como a relação entre a potência emitida para valor mínimo da corrente (corrente de pré-polarização) e a potência emitida devido ao valor máximo da corrente (corrente de pré-polarização mais corrente de modulação). Uma sugestão para amenizar este efeito é utilizar uma corrente de modulação de 10 a 20 vezes maior que a corrente de pré-polarização.