SOFWARE PARA SIMULAÇÃO DE SISTEMAS OFDM (ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING) E MC - CDMA (MULTICARRIER CODE DIVISION MULTIPLE ACCESS)

Ricardo Luiz Moreira Paschoeto

PROJETO SUBMETIDO AO CORPO DOCENTE DO DEPARTAMENTO DE ENGENHARIA ELETRÔNICA E DA COMPUTAÇÃO DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO ELETRÔNICO.

_____
Ricardo Luiz Moreira Paschoeto
(Autor)

_____
Gelson Vieira Mendonça, Ph.D.
(Orientador)

_____
Marcello Luiz Rodrigues de Campos, Ph.D.
(Examinador)

_____
Wallace Alves Martins, M.Sc.
(Examinador)

DEL

Outubro de 2009

AGRADECIMENTO

Gostaria de agradecer a colaboração de todos os professores que contribuíram na conclusão da minha formação como engenheiro e sem os quais a realização deste projeto não teria sido possível.

Agradeço também ao Professor Gelson Vieira Mendonça, meu orientador, pela disponibilidade e boa vontade em me auxiliar nas dúvidas juntamente com o professor Paulo Sérgio Ramirez Diniz, ambos foram fundamentais para a conclusão do meu trabalho.

RESUMO

Este trabalho tem como objetivo desenvolver um sistema de simulação para a tecnologia MC-CDMA no *Downlink* em banda base, na forma de tempo discreto, visando à utilização do mesmo no meio acadêmico por alunos e professores das cadeiras relacionadas aos sistemas de comunicações e transmissão digitais.

Tal sistema foi desenvolvido em Matlab na forma de uma GUI (*Graphical User Interface*). A linguagem Matlab foi escolhida por apresentar funções e bibliotecas específicas para o projeto e construção do sistema, pela sua familiaridade no meio acadêmico e facilidade de programação, vindo a ser útil em futuras alterações do sistema, seja na melhoria ou na inclusão de novas funcionalidades.

ABSTRACT

This work aims develop a MC – CDMA technology system simulation for *Downlink* base band, in discrete time, aiming to use in academic environment by students and professors in disciplines related communications systems and digital transmission.

The system was developing in Matlab language script in GUI form (*Graphical User Interface*). The language script in Matlab was choice because it presentes libraries and procedures for system development, easy understanding and has academic familiarityand also will becames to be useful for future implementations, improving or adding new features.

Indíce de Figuras do Capítulo 3

Indíce de Tabelas do Capítulo 3

Indíce de Figuras do Capítulo 4

1. INTRODUÇÃO

Motivado pelo histórico de avanço tecnológico da transmissão digital, este trabalho foi desenvolvido a fim de analisar conceitos importantes como, modulação digital, técnicas de codificação para controle de erro, técnicas de espalhamento espectral, aplicados a transmissão MC-CDMA.

O trabalho não tem como objetivo ensinar, mas ajudar o aluno que apresenta um conhecimento básico da técnica de transmissão OFDM e MC-CDMA a se tornar apto a utilizar o software para simulações aperfeiçoando seu conhecimento nos sistemas referidos.

O trabalho é organizado da seguinte forma:

No segundo capítulo será apresentado um modelo matemático discreto de transmissão multiusuário, sendo assim será usado como base para construção do sistema, ou seja, o padrão de transmissão e a formação das portadoras serão modelados de acordo com este sistema.

No terceiro será explicado detalhadamente o fluxo e o processamento dos dados nos processos de transmissão OFDM e MC-CDMA, será apresentada as matrizes e vetores gerados em cada etapa do processo de transmissão e recepção desde a formação da mensagem binária passando por sua codificação, modulação, espalhamento, processamento na IFFT e transmissão pelo canal ruidoso e fading até todo processo de recuperação no receptor.

No quarto capítulo a demonstração do software na análise das tecnologias OFDM e MC-CDMA respectivamente, são apresentados exemplos para vários parâmetros de entrada e modos de operação de ambos os sistemas e os resultados gerados pelo sistema são ilustrados através de figuras das telas capturadas do sistema.

No quinto capítulo o software é discutido e analisado pela óptica do desenvolvedor, é apresentado e discutido um fluxograma do software e suas principais funções são apresentadas e explicadas separadamente e em alguns casos é discutido o modo de realizar uma nova implementação.

No sexto capítulo é apresentada a conclusão geral sobre a funcionalidade e utilidade do software.

2. Transmissão e Recepção Multiusuário

Considerando um sistema com K usuários, cada um com índice k e cada símbolo decorrente da modulação e espalhado por uma seqüência específica de espalhamento $s_{k,}$ sendo m o tempo discreto. Os símbolos de dados b_k [m] resultantes do código convolucional, do interleaving, com taxa de mapeamento R_s e taxa de código R_c, sendo assim a informação binária a ser modulada com comprimento R_sR_cM, onde M representa o tamanho do bloco de símbolos. Consideramos também a amplitude de cada usuário como sendo unitária e denotada por a_k (não há perda de percurso e assumindo controle perfeito de potência).

Após o espalhamento dos símbolos, para o usuário k, temos,

(2.1)

Neste vetor é aplicada a IFFT, a partir deste ponto a transmissão passa a seguir o mesmo mecanismo do sistema OFDM, é inserido prefixo cíclico de comprimento G ou zero, temos então a seqüência de transmissão do chip para o usuário k,

(2.2)

Onde,

$$\in [IDFT_N]i, l = e^{-j2\pi il/N},\ i,\ l\ \ \{0, ..., N - 1\},$$

e,

Matriz para inserção do prefixo cíclico, P = N + G, ou

Para inserção do prefixo zero

Substituindo (2.1) em (2.2),

(2.3)

Considerando canal com desvanecimento multipercurso $h[l]$ com comprimento L, expressamos o canal de forma vetorial como,

(2.4)

O sinal resultante será dado pela convolução discreta entre sinal e símbolo,

(2.5)

Para um canal ruidoso AWGN [1], [3], dado por z[n], temos,

(2.6)

Representando o canal por uma matriz *Toeplitz* triangular inferior,

E por uma matriz *Toeplitz* triangular superior,

Reescrevendo matricialmente,

(2.7)

Onde o segundo termo representa a interferência intersimbólica entre dois símbolos MC - CDMA consecutivos.

No receptor o prefixo cíclico é removido e a FFT é aplicada aos *N* chips restantes. A remoção do prefixo cíclico pode ser representada pela matriz,

Para o caso de prefixo zero,

A operação R_{CP} x_k [m], remove as *G* primeiras amostras do vetor x_k [m],

(2.8)

Porém,

(2.9)

Indicando que qualquer interferência intersimbólica entre dois símbolos consecutivos é completamente eliminada.

Podemos então formular o sinal recebido para um usuário *k*,

(2.10)

Sendo (2.11), chamada de matriz circulante do canal.

E ***g*** é a resposta em freqüência do canal, definida como a DFT da resposta impulsiva do canal.

A equação (2.10) descreve a essência matemática da transmissão multiusuário. A matriz do canal é circulante pela inserção do prefixo cíclico. Portanto, a operação para formar a matriz circulante forma uma matriz diagonal com a diagonal formada pelos autovalores da matriz resultante da convolução. A inversão de *diag* (g_k) é necessária para equalização do canal.

Podemos reescrever (2.10) usando (2.11),

(2.12)

Definindo a seqüência de espalhamento efetiva como,

(2.13)

E representando o sistema multiusuário por,

(2.14)

Onde a matriz de espalhamento efetiva é definida como,

e

(2.15)

Contém os símbolos de informação para K usuários com tempo discreto *m*.

Esta análise discreta será de fundamental importância neste trabalho, pois o sistema de simulação desenvolvido irá seguir este modelo matemático de transmissão no *downlink.*

3. Sistema de simulação MC - CDMA

O sistema de simulação MC – CDMA seguirá o modelo descrito no tópico anterior sendo então realizada no "downlink", em tempo discreto e banda base, a análise será realizada com os resultados gerados a partir da simulação *Monte Carlo*. A parametrização foi realizada nas principais técnicas, como codificação, modulação e espalhamento, inserção de prefixos, canal digital e AWGN e número de usuários. As principais funcionalidades oferecidas são listadas abaixo e serão explicadas separadamente:

- Escolha do número de usuários variável e o número de portadoras do sistema (64 a 1024);
- Codificação convolucional, opções para código convolucional:
  - k =1/2 – (k=3; [7 4], k=4; [17 11], k=5; [27 31], k=6; [57 65], k=7; [117 155], k=8; [237 345], k=9; [617 435])
  - k = 1/3 – (k=3; [7 5], k=4; [17 13 15], k=5; [37 33 25], k=6; [57 65 71], k=7; [117 127 155], k=8; [357 233 251])
- Opções para código em bloco (n, k):
  - (7,4), (15,11), (31,26);
- Escolha de técnicas de modulações binárias e M-árias e código de espalhamento (2, 4, 8, 16, 32);
- Modelos de canais AWGN e discreto e escolha do tipo de prefixo aplicado (zero ou cíclico);
- Resultados apresentados em gráficos como BER x SNR, espectro do sinal, constelações M – árias.

A figura (3.1) mostra uma visão geral do sistema.

Figura 3. - Sistema de simulação MC – CDMA.

3.1. Fluxo de dados

3.1.1. OFDM

- **Geração da mensagem binária**

Escolhendo o sistema com apenas 1 (um) usuário o mesmo se comportará como um sistema OFDM [1], [3], [5], [9], [10]. É gerada uma mensagem binária aleatória através da função matlab *randint* o resultado é um vetor com um tamanho de 16 (valor escolhido para se ter um valor mínimo fixo e como potência de 2) multiplicado pelo valor *k* do código de bloco pelo fator $log_2(M)$ da constelação *M-ária* e pelo fator *M* de modulação, dessa forma a mensagem terá um comprimento mínimo de 32 bits, para o caso de ausência de codificação (k = 1), e M = 2 (modulação BPSK) a tabela (3.1) mostra o vetor formado para o caso citado, os valores escolhidos servem também para manter uma compatibilidade entre o comprimento da mensagem e as operações de codificação e modulação.

Tabela 3. – Exemplo vetor 1 x 64.

0	1	2	3	...	60	61	62	63
0	1	1	0	...	0	1	1	1

- **Codificação e Modulação**

No processo de transmissão antes do vetor de mensagem binária ser modulado o mesmo poderá ser codificado por aplicação de códigos convolucionais e/ou de blocos [1], [3], [5], [9], então de acordo com os parâmetros *r* e (n,k_b), o vetor mensagem poderá ter seu tamanho aumentado por um fator de $[(1/r)\ x\ (n/k)]$, o sistema oferece para os códigos convolucionais opções de parametrização de *r* e k_c e para os códigos de bloco a parametrização do par k_b e *n*. Sendo assim o vetor mensagem poderá ter, caso seja codificado,

tamanhos entre 32 ($k = 1$ e $r = 1$) e 95232 ((sinal modulado 64-QAM) x 3 x (31/26)) quando os parâmetros máximos da modulação e codificação são escolhidos. O passo seguinte na transmissão envolve o "interleaving" [1] do vetor afetando apenas a disposição dos bits na mensagem.

O vetor gerado será modulado com as seguintes técnicas: BPSK, QPSK, 8-QAM, 16-QAM, 32-QAM e 64-QAM [1], [2], [3], [6]. A tabela (3.2) mostra a modulação BPSK, onde cada bit de mensagem irá formar um símbolo BPSK, considerando N o tamanho da mensagem gerada e codificada e ainda M o índice de modulação, o comprimento do vetor modulado será dado pela relação $N/log_2(M)$.

Tabela 3. – Vetor modulado BPSK.

0	1	2	3	4	5	6	...	N
-1	1	1	-1	1	-1	1	...	1

Para o caso QPSK cada $log_2(M)$ bits da mensagem formarão um símbolo QPSK com valores no plano complexo {±1,±j}, a tabela (3.3) mostra o resultado, na modulação QPSK temos um índice de modulação M = 4, então $log_2(M) = 2$, sendo este resultado um valor *par* esta constelação é conhecida como constelação *quadrada* [1], [3], [5].

Tabela 3. – Vetor modulado QPSK.

0	1	2	3	4	5	6	...	$N/log_2(M)$-1
-1+j	1-j	1-j	1+j	-1-j	-1-j	1+j	...	-1-j

Para a opção 8-QAM cada 3 bits da mensagem irão formar um símbolo 8-QAM no plano complexo {±3,±1,±j,±3j}, a tabela (3.4) mostra o vetor modulado, sendo esta uma constelação com $log_2(M) = 3$, um valor ímpar, ela é conhecida como constelação *cruzada* [1], [3], [5].

Tabela 3. – vetor modulado 8-QAM cruzada.

0	1	2	3	4	5	6	...	$N/log_2(M)$-1
-3+j	1-3j	3-3j	1+j	-1-j	-1-j	1+j	...	-1-3j

Conseqüentemente as constelações 16-QAM, 32-QAM e 64-QAM terão um comportamento similar tendo cada símbolo 4, 5 e 6 bit's respectivamente. As tabelas (3.5), (3.6) e (3.7) ilustram as respectivas modulações.

Tabela 3. – vetor modulado 16-QAM quadrada, {±3,±1,±j,±3*j}.

0	1	2	3	4	5	6	...	$N/log_2(M)-1$
-3+j	1-3j	3-3j	1+j	-1-j	-1-j	1+j	...	-1-3j

Tabela 3. – vetor modulado 32-QAM cruzada, {±5,±3,±1,±j,±3*j,±5*j}.

0	1	2	3	4	5	6	...	$N/log_2(M)-1$
-5+j	1-3j	3-5j	1+j	1+j	-1-5j	1+j	...	-1-5j

Tabela 3. – vetor modulado 64-QAM cruzada, {±7,±5,±3,±1,±j,±3*j,±5*j,±7*j}.

0	1	2	3	4	5	6	...	$N/log_2(M)-1$
-7+j	1-5j	3-5j	1+j	-1-7j	-1-7j	1+j	...	-1-5j

- **Espalhamento Espectral**

Caso seja marcada a opção de espalhamento espectral, o usuário poderá escolher a dimensão da matriz de *walsh – hadamard* [7], [9] que será gerada previamente. Como apenas um usuário transmite dados pelo canal o vetor de dados modulados será multiplicado pela primeira linha da matriz *walsh – hadamard* (um código de *walsh*), essa multiplicação vetorial irá gerar N_c *chips*, sendo N_c o comprimento do código de espalhamento escolhido, ao final irá se obter uma matriz formada por N_c *x* $\boldsymbol{N_s}$**,** sendo N_s o número de símbolos. A tabela (3.8) ilustra a matriz formada com modulação M - QAM.

Tabela 3. – Matriz de Chip's x Símbolos.

S C	0	1	2	3	4	5	6	...	$N/log_2(M)-1$
0	-7+j	1+5j	3-5j	1+j	-1-7j	1-7j	1+j	...	-1-5j
1	-7+j	1+5j	3-5j	1+j	-1-7j	1-7j	1+j	...	-1-5j
2	-7+j	1+5j	3-5j	1+j	-1-7j	1-7j	1+j	...	-1-5j
...	-7+j	1+5j	3-5j	1+j	-1-7j	1-7j	1+j	...	-1-5j
$N_c – 1$	-7+j	1+5j	3-5j	1+j	-1-7j	1-7j	1+j	...	-1-5j

- **IFFT**

O sistema usa como geração de multiportadoras a aplicação da IFFT sobre o sinal formado na modulação, cada símbolo dentro sinal irá modular uma portadora, o número de

portadoras é um parâmetro do sistema. Caso existam mais portadoras que o número de símbolos presentes em um sinal modulado, o sinal sofrerá adição de zeros até que seu comprimento seja compatível com o número de portadoras, sendo retirados os zeros antes do processamento na recepção.

No entanto se o número de portadoras disponíveis é menor que o número de símbolos no sinal modulado, o sinal será então dividido em blocos com número de símbolos compatíveis com o número de portadoras, cada bloco será transmitido individualmente e remontado para formação do sinal modulado. O número de blocos será determinado pela relação N/N_p, sendo N o número de símbolos no sinal e N_p o número de portadoras.

Tabela 3. – Vetor original de símbolos.

0	1	2	3	4	5	6	7	8	...	$N/log_2(M)$-1
-7+j	1-5j	3-5j	1+j	-1-7j	-1-7j	1+j	1+j	1+j	...	-1-5j

Tabela 3. – *Padding* com zeros para ajuste do vetor de símbolos para transmissão sobre as portadoras, caso $N_p > N_s$.

0	1	2	3	4	5	6	...	$N/log_2(M)$-1	0	...	0
-7+j	1-5j	3-5j	1+j	-1-7j	-1-7j	1+j	...	-1-5j	0	...	0

Tabela 3. – Exemplos de blocos de símbolos para transmissão.

0	1	2	3	4	5	6	7	8	...	$N/log_2(M)$-1
-7+j	1-5j	3-5j	1+j	-1-7j	-1-7j	1+j	1+j	1+j	...	-1-5j

- **Prefixo e Canal**

Outro parâmetro do sistema é a escolha de utilização de prefixo cíclico ou zero [1], [3], [5], [11]. Este processo é realizado pela multiplicação matricial do sinal vindo da IFFT com comprimento 1 x N_p e a matriz de inserção de prefixo, a matriz do prefixo é formada baseando-se na matriz apresentada no capítulo 2.(equação 2.2), o número de amostras inseridas é determinada pelo número P de *taps(coeficientes)* do canal digital, sendo o canal também um parâmetro do sistema. Com N_p o número de portadoras e sendo este valor também o número efetivo de símbolos transmitidos N, então $N = N_p$, sendo assim a matriz para inserção de prefixo cíclico terá dimensão $P \times N$ e para prefixo zero $N \times P$, sendo $P = N + G$, com G o número de amostras inseridas.

Para incluir os efeitos do canal são geradas matrizes *Toeplitz* com dimensão $P \times P$ para que sejam compatíveis com o vetor de símbolos com prefixo incluso.

Como exemplo, quando se tem um canal com *10 taps* e blocos com comprimento de *64*, a matriz para prefixo terá dimensão *64 x 74* e matrizes *Toeplitz* com dimensões *74 x 74.* O canal também poderá ser afetado por ruído AWGN caso esta opção seja escolhida.

- **Remoção do Prefixo Ciclico**

Após inserir o ruído AWGN o prefixo cíclico será removido pela matriz de remoção do prefixo gerada no mesmo processo da matriz de inserção. Esta matriz terá dimensão $N_p \, x \, P$, sendo assim a multiplicação matricial ($N_p \, x \, P$) x ($P \, x \, N_p$) = ($N_p \, x \, 1$) ou ($N_p \, x \, \boldsymbol{N_c}$) removendo então as amostras inseridas anteriormente, resultando em um vetor, caso os símbolos modulados não tenham sofrido espalhamento, ou uma matriz com as linhas formadas pelos símbolos e as colunas os chip´s gerados com o espalhamento.

- **FFT, Equalização e Desespalhamento**

Após o processo de remoção do prefixo será aplicada a matriz ou vetor a FFT para recuperação dos símbolos na freqüência discreta. O resultado gerado pela FFT será equalizado pelo processo que envolve a multiplicação matricial deste resultado pela matriz quadrada de equalização com dimensão $N_p \, x \, N_p$ sendo esta última uma matriz diagonal formada pelos autovalores da convolução do bloco transmitido com a matriz de canal.

Realizado esse processo é formada uma matriz ou vetor com os símbolos estimados, no caso de uma matriz com as colunas determinando o número de chip´s transmitidos, sendo necessário aplicar o desespalhamento para formar o vetor de símbolos estimados na transmissão. Este processo é realizado multiplicando a matriz pelo código de *walsh* gerado no transmissor, ($1 \, x \, N_c$) x ($N_c \, x \, N_p$) = ($1 \, x \, N_p$).

- **Demodulação e Decodificação**

O processo final para análise do desempenho do sistema envolve dois processos: demodulação e decodificação. A demodulação irá transformar a matriz de símbolos em uma matriz de bits com o número de bits por símbolo relacionado com o tipo de modulação empregada na transmissão, cada símbolo da matriz gerada no processo de demodulação formará $log_2(M)$ bits, sendo M o índice de modulação. Assim cada bloco de símbolos terá um comprimento $N \, x \, log_2(M)$, com N o comprimento do bloco de símbolos.

A decodificação por bloco e/ou *Viterbi* [1], [3], [5], [11] removerá a redundância inserida na mensagem para controle de erros, o vetor sofrerá uma redução de $(k_b/n) \, x(r)$,

decorrente da decodificação por bloco e convolucional, respectivamente, deinterleaving poderá ser aplicado caso o interleaving tenha sido realizado na transmissão.

Será finalmente realizada a contagem de erros comparando o vetor recebido com o transmitido. Este resultado será armazenado em um vetor de erros e apresentado de forma logarítmica em função da relação sinal/ruído (SNR).

3.1.2. MC – CDMA

- **Geração da mensagem binária**

Para que o sistema opere no padrão MC – CDMA basta que o número de usuários seja maior que 1, sendo assim o fluxo de dados sofrerá alterações em quantidade de processamento principalmente no espalhamento. A função de geração de bits de mensagem agora irá gerar uma matriz em que as linhas correspondem aos usuários e as colunas aos dados binários, a tabela (3.12) mostra a matriz gerada.

Tabela 3. – Matriz Usuários x Dados.

B U	0	1	2	3	4	5	6	...	$N/log_2(M)-1$
2	1	1	0	0	1	0	1	...	1
3	0	0	0	1	1	1	0	...	0
...	1	0	1	0	0	1	0	...	1
N	1	1	1	1	0	0	0	...	0

Sendo U variando 2 a 32 (com número de usuários como potência de 2). O número de bits por usuário segue o padrão OFDM apresentado anteriormente. A codificação e modulação seguem também o padrão do sistema OFDM, porém agora a função de codificação convolucional e/ou bloco irá realizar uma iteração nas linhas da matriz de bit's, referentes aos usuários, gerando uma nova matriz codificada assim como a função de modulação.

- **Espalhamento Espectral, IFFT, canal e Prefixos**

Para o sistema MC – CDMA em questão, a geração de portadoras é realizada pelo sistema OFDM, sendo assim após cada bloco de símbolos de cada usuário no sistema ser espalhado será somado "chip" a "chip". O resultado será então uma matriz resultante da soma das matrizes formadas pelo espalhamento do vetor de símbolos de cada usuário. Esta é a principal característica que diferenciará um sistema OFDM de um MC – CDMA no processo

de transmissão/recepção. O processamento da IFFT será como no OFDM com espectro espalhado assim como o processo de convolução com o canal e inserção/remoção de prefixos.

Na recepção o desespalhamento será realizado para cada linha da matriz referente a cada usuário, o resto do processo envolvendo a equalização, demodulação e decodificação que são análogos ao sistema OFDM. A contagem de erros agora será uma matriz com as colunas indexando os usuários do sistema e as linhas os erros contados no processo de transmissão.

4. Simulações

4.1. Painel de Usuários

Neste painel temos a opção de escolher o número de usuários que irão compartilhar o sistema.

Caso se escolha apenas 1 usuário este sistema será puramente OFDM, sendo assim poderá ter espalhamento espectral ou não, de acordo com a opção do usuário no painel de modulação. Se o sistema tiver 2 ou mais usuários, ele será MC – CDMA, e só poderá ser realizado com espalhamento espectral, pois esta é a característica de transmissão e detecção MC -CDMA.

A opção do número de portadoras irá determinar o número de símbolos enviados pelo sistema e cujos valores mínimos e máximos será de 64 e 1024 respectivamente, calculado após determinar o tamanho do bloco a ser transmitida, a figura (4.1) mostra este painel.

Figura 4. – Painel de usuários

4.2. Painel de Codificação

Este painel dá opções de códigos para controle de erros aos usuários, tanto Convolucionais [1][3][5] como de bloco de *hamming* [1][3][5], clicando na *check Box* correspondente será habilitada a *combo box* para escolha, dentre várias opções, dos parâmetros do código a ser utilizado no sistema. As opções apresentadas são de códigos com valores de taxa e comprimento otimizados. Outra opção será também a inserção de um interleaving de bloco, neste caso será habilitada esta opção caso uma codificação por bloco seja escolhida. Na figura (4.2) temos a visão deste painel.

Figura 4. – Painel de codificação

4.3. Painel de Modulação

Este painel apresenta as opções de modulação digital: BPSK, QPSK, 8-QAM, 16-QAM, 32-QAM, 64-QAM. Com isso o usuário poderá analisar seu sistema configurado com constelações pares e ímpares.

Outra opção é o comprimento da matriz de *walsh – hadamard* [7] usada no espalhamento espectral, cujos valores variam de 2 a 32. A figura (4.3) mostra o painel de modulação.

Figura 4. – Painel de Modulação

4.4. Painel de Canal

Este painel modela o canal, com as opções para canal ruidoso AWGN e/ou canal com desvanecimento. Para o canal digital com desvanecimento, os coeficientes dos atrasos serão inseridos pelo usuário dentro de colchetes. Neste painel também poderá ser escolhido o tipo de prefixo usado na transmissão, cíclico ou zero. A figura (4.4) mostra o painel de canal.

Figura 4. – Painel de canal

4.5. Apresentação dos Resultados

A apresentação dos resultados do desempenho do sistema configurado será apresentada nos gráficos da BER e do espectro localizados no painel frontal, outro gráfico, como as constelações poderá ser verificados ao ser “clicar” nos botões inferiores “Exibir Constelações”. Estes dados estarão disponíveis para análise após o botão “simular” ser pressionado. Para terminar o sistema, basta clicar no botão “Sair”, pode-se limpar o gráfico

com as curvas da BER clicando no botão “Reset”. As figuras (4.5) e (4.6) mostram os componentes descritos.

Figura 4. – Componentes para análise da simulação

Figura 4. – Constelações

4.6. Utilizando o Software

Exemplo 1: Para demonstrar o uso do software será realizada a simulação de dois sistemas, um OFDM e um MC-CDMA, onde serão analisados os resultados obtidos. Para o sistema OFDM vamos configurá-lo da seguinte forma:

- 64 portadoras;
- Modulação 64 – QAM;
- Sem espalhamento espectral;
- Canal AWGN
- Canal digital com coeficientes:

[0.2816 0.5622 0.2677]

- Prefixo Cíclico;

Primeiro vamos ao painel de Usuários e marcamos a primeira opção e o número de portadoras logo abaixo, ambas já estarão selecionadas por default. Na figura (4.7) ilustra o procedimento.

Figura 4. – Painel usuários do exemplo 1.

Não utilizaremos por enquanto nenhum código corretor, nem convolucional, bloco ou “interleaving” deixando então as opções do painel de codificação desmarcadas, figura (4.8).

Figura 4. – Painel Codificação exemplo 1.

Escolhemos agora as opções de modulação e espalhamento, no caso do espalhamento, como nosso sistema é OFDM, temos a opção de escolher com e sem espalhamento, caso o sistema seja MC-CDMA a opção com espalhamento torna-se obrigatória e uma mensagem de alerta é enviada ao usuário para que o mesmo escolha uma opção válida de acordo com seu sistema, a figura (4.9) ilustra o procedimento.

Figura 4. – Painel Modulação e Espalhamento do exemplo 1.

Por último, No painel de canal escolhemos as configurações de canal, AWGN e/ou "Fading" e ainda o tipo de prefixo, no exemplo em questão temos o canal AWGN e/ou "Fading" e prefixo cíclico, a figura (4.10) ilustra o procedimento.

Figura 4. – Painel Canal do exemplo 1.

Depois de configurado o programa apresentará um resumo do sistema como mostra a figura (4.11).

Figura 4. – Sistema Configurado do exemplo 1.

Após pressionar o botão simular o resultado será exibido como mostram as figuras (4.12) e (4.13). Este resultado condiz com o esperado para um sistema OFDM simples.

Figura 4. – BER x SNR e Espectro do exemplo 1.

Figura 4. – Constelações do exemplo 1.

Para o gráfico da BER o resultado de cada curva, no caso de várias simulações, terá um rótulo que mostrará primeiro o tipo de modulação, o fator de espalhamento, as codificações convolucional e por bloco respectivamente, todos separados por vírgula.

Exemplo 2: Agora para o mesmo sistema vamos aplicar um código de espalhamento Walsh de comprimento 32 e codificação convolucional com k = 1/2 e r = 9, codificação por bloco de *Hamming* com n = 7 e k = 4 e interleaving, a figura (4.14) ilustra as modificações.

Figura 4. – Nova Configuração do exemplo 2.

Como pode ser verificado pelo gráfico da BER ocorreu uma melhora significativa no desempenho do sistema, como era esperado. A figura (4.15) mostra os resultados.

Figura 4. – Resultados do exemplo 2.

Estes resultados mostraram-se de acordo com a teoria de transmissão OFDM.

Exemplo 3: Para o sistema MC-CDMA vamos configurá-lo primeiro da seguinte forma:

- 4 usuários;
- Número de bandas: 128;
- Modulação 64-QAM;
- Ruído AWGN;
- Espalhamento de comprimento 4;
- Modelo de canal:

  [0.2816 0.5622 0.2677 -0.3260 -0.4415 0.3102 0.0292 -0.2992 -0.2004]
- Prefixo Cíclico;

Abaixo, figura (4.16) tem as configurações realizadas:

Figura 4. – Sistema MC-CDMA configurado do exemplo 3.

Após a simulação temos os resultados abaixo, figura (4.17). Não foi aplicada nenhuma codificação e/ou interleaving para controle de erros.

Figura 4. – Resultados do exemplo 3.

Exemplo 4: Aplicando agora codificação convolucional, k=1/2 e r = 9, codificação por bloco, n = 7 e k = 4 e interleaving, figura (4.18), veremos que o sistema assim como no caso OFDM apresenta uma significativa melhora a partir de 22 dB, figura (4.19), a BER para os usuários se mostra idêntica, pois todos são submetidos à mesma taxa de espalhamento e modulação.

Figura 4. – Configuração da codificação do exemplo 4.

Figura 4. – Resultados comparativos do exemplo 4.

Aplicando agora um código convolucional com r = 1/3 e k = 8, temos o resultado apresentado na figura (4.20).

Figura 4. – Resultados com novo código convolucional do exemplo 4.

A última curva representa o sinal com a codificação aplicada, ele mostra um desempenho ainda melhor do sistema a partir de 17 dB.

A figura (4.21) mostra um sistema MC-CDMA com o número de usuários variando até sua carga máxima, que ocorre quando o número de usuários é igual ao fator de espalhamento, nesse caso variando 2 a 8 usuários com fator de espalhamento igual a 8, confirmando a teoria do sistema quando operando a plena carga apresenta uma BER satisfatória em presença de canal desvanecimento.

Figura 4. – MC-CDMA a plena carga em canal ruidoso e desvanecimento do exemplo 4.

Exemplo 5: Na figura 4.22 temos os resultados para as codificações convolucionais e em bloco aplicadas anteriormente, porém agora com prefixo zero.

Figura 4. – Resultados com aplicação de prefixo zero.

5. Estrutura Interna do Software

A figura (5.1) mostra as funções do sistema e o diagrama em blocos do software, no Apêndice A mostra a lista das funções do programa principal. À medida que as funções forem descritas será também discutida a inclusão de novas funcionalidades por alteração da função em questão ou inclusão de uma ou mais funções. Na figura (5.2) é mostrado um diagrama com o fluxo de dados dentro sistema.

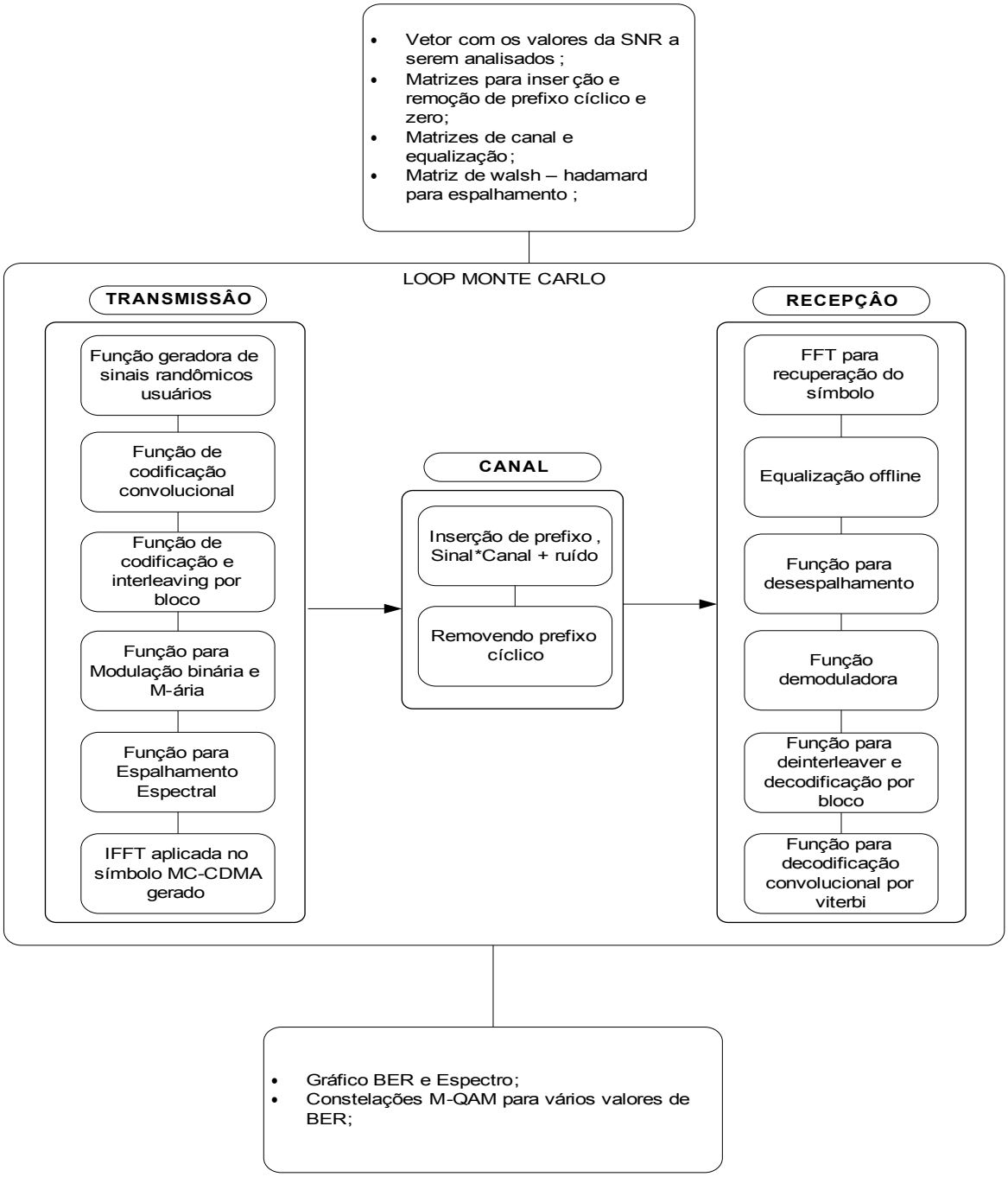

Figura 5. – Diagrama em blocos das funções do software.

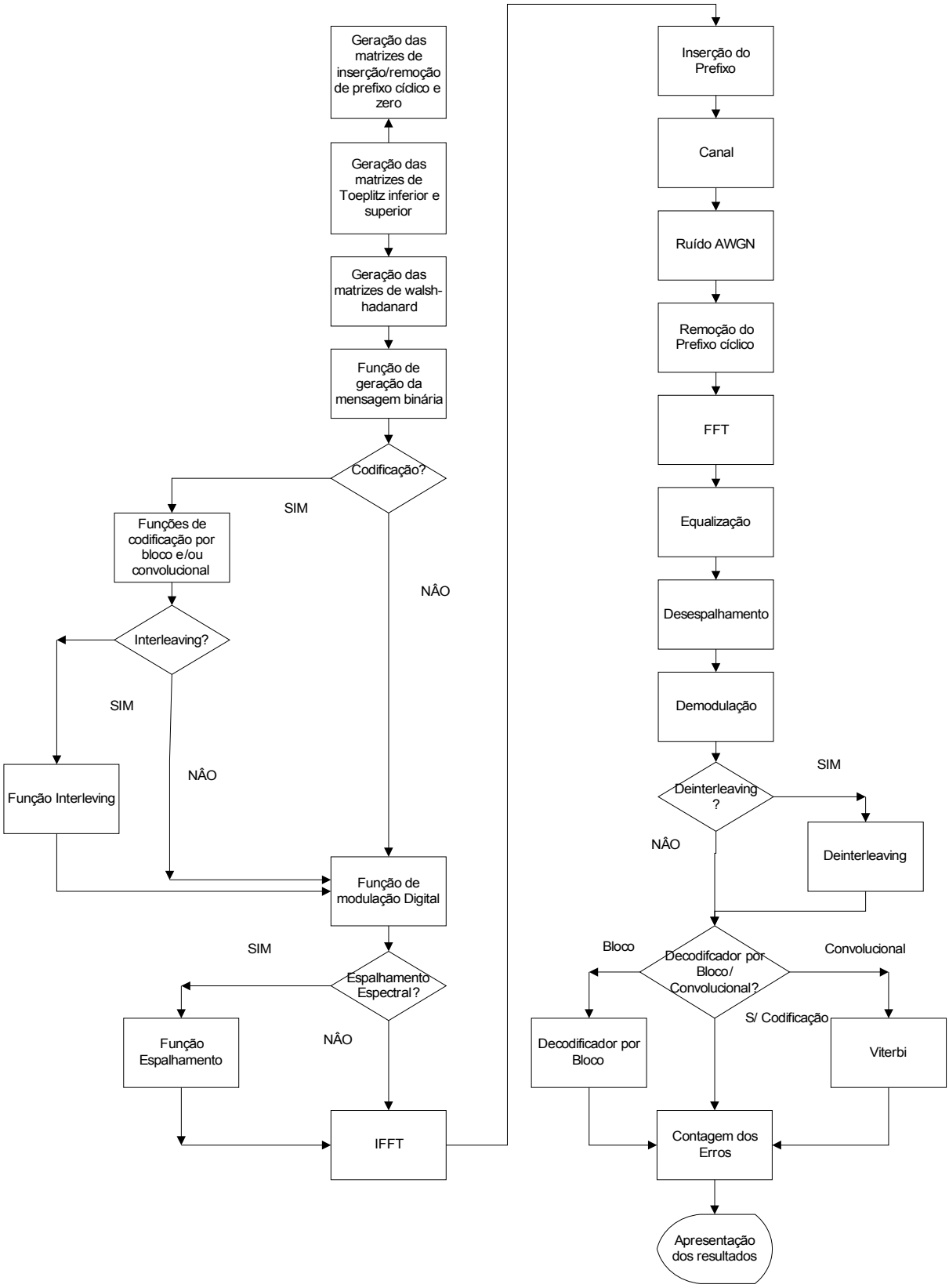

Figura 5. – Fluxo de Dados.

5.1. Principais funções

1. Função *simulação*

Esta é a função principal do sistema onde todas as outras funções são chamadas, esta função é o *callback* do botão simular na interface com o usuário, sendo assim ela recebe uma *struct* chamada *sistema* com todos os parâmetros escolhidos na interface tais como: **número de usuários, número de portadoras, taxa do código convolucional e o vetor em octal de realimentação, par (*k, n)* do código de bloco, "flag" para interleaving, índice de modulação, taxa de espalhamento, tipo de prefixo**. Esta função ainda retornará alguns valores importantes como: o vetor com a BER (*Bit Error Rate*), o vetor com o espectro do sinal, vetores com as constelações sem e com espalhamento espectral, todos esses dados serão processados dentro da interface para que possam ser exibidos em forma gráfica, isso será detalhado mais tarde neste trabalho. Abaixo temos o protótipo da função *Simulação.*

```
function [errosAcumulados, espectro, constelacao_ns, constelacao_ss] =
                                simulacao(sistema)
```

É importante citar que dentro dessa função alguns parâmetros importantes são declarados como, o vetor *SNR* que irá receber os valores da relação sinal ruído de 0 á 30 dB, este será indexado dentro do loop *Monte Carlo*, temos também o vetor se saída *errosAcumulados* que receberá os valores da BER, os parâmetros *h* e *h_Demod* relativos á modulação e demodulação *M-ária*, respectivamente, e a matriz *ss* que irá representar a matriz de *walsh - hadamard*.

2. Função *inserirPrefixo*

Esta função recebe como parâmetros o número de portadoras do sistema, o número de usuários e o número de *taps* do canal, ela tem como finalidade, como indica seu nome, a formação das matrizes de prefixos cíclicos e zero, bem como a matriz de remoção no caso do prefixo cíclico ou de ajuste no caso de prefixo zero. Ela retorna as matrizes de inserção e remoção/ajuste de prefixo. Caso não tenha sido selecionada a opção do canal digital o valor do *tap* será igual a -1 e não haverá inserção de prefixo sendo as matrizes de inserção/remoção iguais a matriz identidade *I*. Abaixo temos o protótipo da função.

```
function [matrizPrefixo, matrizRemoverPrefixo] =
     inserirPrefixo(portadoras, tipo, tap)
```

3. Função *canalDigital*

Esta função recebe como parâmetros o vetor do canal digital modelado pelo usuário e o número de portadoras, ela é responsável por gerar uma matriz de *Toeplitz* com os coeficientes do canal, gerando como saída esta matriz. Esta função só será chamada caso o comprimento do vetor do canal seja diferente de zero, caso contrário a matriz gerada será a matriz identidade. Abaixo temos o protótipo da função.

```
function [H0] = canalDigital(canal,portadoras)
```

4. Função *matrizEqualizacao*

Esta função recebe como parâmetros a matriz de remoção/ajuste de prefixo, a matriz de canal gerada pela função *canalDigital*, a matriz de inserção de prefixo e o número de portadoras do sistema. Dentro desta função ocorrerá uma seqüência de processos que se inicia com a formação da matriz circulante gerada pela multiplicação das matrizes de remoção/ajuste de prefixo, de canal e de inserção de prefixo, e terminando com a matriz de equalização que será o parâmetro de retorno desta função.

```
function [equalizacao] = matrizEqualizacao(RCP, H0, TCP, portadoras)
```

5. Função *hadamard*

Esta é uma função interna do Matlab que recebe como parâmetro o valor da dimensão da matriz de *hadamard* a ser gerada para espalhamento espectral, seu protótipo é mostrado abaixo.

```
function H = hadamard(n)
```

6. Função *geradorSinaisUsuarios*

Esta função irá formar a mensagem binária recebendo como parâmetro o número de usuários, o parâmetro *k* do código de bloco escolhido e o índice de modulação, seu valor de retorno será um vetor ou uma matriz de acordo com o parâmetro número de usuários.

```
function [matrizUsuarios] = geradorSinaisUsuarios (nUsuarios, kb,
                                                   nQAMusuario)
```

A função irá gerar um vetor ou matriz compatível com os processos de codificação e modulação escolhidos.

7. Função *codificadorConvUsuarios*

Esta função realizará a codificação convolucional da mensagem binária, recebendo como argumento o vetor ou matriz com as mensagens, o fator *k* e o vetor octal do código convolucional escolhido, a função então irá gerar a treliça para codificação.

```
function[vetorUsrCodificados] = codificadorConvUsuarios(vetorSinais, kc, p)
```

8. Função *codificadorBloco*

Esta função realizará a codificação por bloco de *Hamming* da mensagem, ela recebecomo parâmetro o vetor ou matriz de sinais, os fatores *(n, k)* do código de bloco escolhido.

```
function [matrizBlocoCod] = codificadorBloco(Sinais, n, kb)
```

Para inserção de mais tipos de códigos corretores como *Reed Solomon, Turbo* e mais opções tanto de códigos convolucionais e de bloco, a estrutra do sistema deverá sofrer modificações tanto na interface gráfica com ajuste para inserção dos novos códigos quanto na estrutura das funções. Uma opção seria criar uma função de codificação onde a mesma irá conter as funções de codificação por bloco, convolucional e as novas funções, assim a seleção do tipo de codificação ficaria a cargo desta função, isto também acarretaria um código mais limpo e legível. Caso o sistema sofra uma modificação drástica e venha a ser reprojetado utilizando o paradigma de orientação a objeto, uma classe especifica para codificação poderia ser criada e nela conteriam as funções de codificação especificadas.

Dentro da interface para inserção de mais opções tanto de códigos convolucionais quanto de blocos, basta para isso seguir o seguinte procedimento: na tela da interface basta "clicar" com o botão direito do "mouse" sobre uma das combos referentes aos códigos e selecionar a opção *property inspector-> String* e adicinar as novas opções.

9. Função *intlvr*

Esta função realiza o "interleaving" por bloco do sinal codificado e recebe como parâmetro o vetor codificado.

```
function[vetorUsrIntlvr] = intlvr(vetorCodificado)
```

10. Função *moduladorSinal*

Esta função irá modular o sinal mensagem, ela irá receber como parâmetro a estrutura *h*, o vetor sinal e o índice de modulação. Retornando assim um vetor com símbolos.

```
function[vetorModulado] = moduladorSinal(h, sinalIntlvr, M)
```

Para inserção de novos valores de modulação M- ária basta realizar o processo descrito anteriormente através da propriedade da combo *property inspector*. Para novas técnicas de modulação diferentes da M – ária será necessário alterar a estrutura tanto da inerface como das funções.

11. Função *equalizador*

Esta função irá equalizar o sinal distorcido pelo canal, ela receberá como parâmetro a matriz de equalização gerada anteriormente pela função de equalização e o sinal procedente da *fft*.

```
function [sinalEq] = equalizador(eq, simbolo)
```

12. Função *espalhamento*

Esta função irá utilizar a matriz de *hadamard* gerada anteriormente, matriz *ss*, para espalhar o sinal modulado. Ela recebe como parâmetro o sinal modulado, a matriz *ss* e o número de usuários.

```
function[simbolo] = espalhamento(sinais, wHadamard, nUsuarios)
```

Dentro da função *simulação* existem três loops, o primeiro é o loop que faz a indexação do vetor SNR, o segundo é o loop de *Monte Carlo*, o terceiro é loop para transmissão dos sucessivos blocos do símbolo mensagem original. Neste ponto do processo também temos as funções *ifft* e *fft*, ambas recebendo como parâmetro um vetor de símbolos e o número de portadoras, estas funções são nativas do Matlab.

```
ifft(sinaisModulados,sistema.Portadoras);

fft(sinalSP,sistema.Portadoras);
```

Alguns processos importantes podem ser citados, como as distorções do canal aplicadas ao sinal pela multiplicação do mesmo pela matriz de canal gerada anteriormente junto também com a inserção/remoção/ajuste dos prefixos pela multiplicação das respectivas matrizes geradas no processo inicial. Os vetores de saída da função como o espectro, e as constelações serão selecionados neste processo. As funções presentes no processo de recepção realizam o processo inverso das funções presentes na transmissão, estas são listadas com seus respectivos parâmetros: `function [desespalhamento] = desespalhamento(wHadamard, sinal, nUsuarios)`, `function [demodulado] = demodulador(hdemod, sinal, M)`, `function[vetorUsrDeIntlvr] = deintlvr(sinal)`, `function [decodificado] = decodificador(sinal, n, kb, nUsuarios)`, `function [decodificado] = decodificadorConv(sinal, kc , polinm, nUsuarios)`.

Após a execução da função simulação os valores retornados por ela serão tratados dentro da função *simMCCDMA*, para os vetores de símbolos representando as constelações geradas os mesmos serão salvos em arquivos para que possam ser processados em outra GUI e exibidos em gráficos, resultado do *callback* do botão “Exibir Constelações” na interface principal.

6. CONCLUSÃO

Com este trabalho procura-se apresentar uma ilustração das técnicas multiportadoras a fim de aumentar o interesse no meio acadêmico dos alunos que estarão entrando em contato com as técnicas de transmissão digital no decorrer do curso de engenharia eletrônica na UFRJ. São mostrados vários conceitos como, modulação, codificação e outras abordadas no decorrer do curso de engenharia.

Nos primeiros capítulos apesar de várias bibliografias abordarem o assunto, apresentou-se os sistemas de forma mais resumida e direcionada para introduzir conceitos essenciais ao entendimento do sistema desenvolvido, o mesmo apresentou bom resultado em termos de desempenho sendo esta a principal preocupação em relação à utilização de uma linguagem de *script* como o matlab, o sistema foi construído dentro de um projeto e compilado com o compilador interno do matlab. Apresentam-se também resultados condizentes com a teoria de transmissão multiportadora confirmados com os exemplos apresentados na seção anterior.

Algumas melhorias podem ser implementadas futuramente no sistema como:

- Escolha do número de bits ou símbolos para usuário;
- Escolha de modulação, espalhamento e codificação independente para cada usuário;
- Mais opções de códigos Convolucionais e de bloco, além de códigos como *Reed-Solomon*, *Turbo* e outros;
- Mais opções de constelações;
- Interleaving convolucional;
- Scrambling;
- Uma versão do sistema em Java ou C/C++ aproveitando a interoperabilidade entre o matlab e essas linguagens;

Estas são apenas algumas funcionalidades que podem ser incorporadas, porém outras poderão acontecer com o uso do sistema, podendo assim ser usado também futuramente para simulação de um sistema de transmissão *VSF-OFCDMA*, como citado no início deste trabalho um candidato para a próxima geração de telefonia (4 G), este sistema usa a técnica de espalhamento variável, incorporando as técnicas OFDMA e MC-CDMA.

Apêndice A

```
function [errosAcumulados, espectroSS, constelacao_ns, constelacao_ss] =
simulacao(sistema)

function [matrizPrefixo, matrizRemoverPrefixo] = inserirPrefixo(portadoras,
tipo, tap)

function [H0] = canalDigital(canal,portadoras)

function [equalizacao] = matrizEqualizacao(RCP, H0, TCP, portadoras)

function [matrizUsuarios] = geradorSinaisUsuarios (nUsuarios, kb,
nQAMusuario)

function[vetorUsrCodificados] = codificadorConvUsuarios(vetorSinais, kc, p)

function [matrizBlocoCod] = codificadorBloco(Sinais, n, kb)

function[vetorUsrIntlvr] = intlvr(vetorCodificado)

function[vetorModulado] = moduladorSinal(h, sinalIntlvr, M)

function[simbolo] = espalhamento(sinais, wHadamard, nUsuarios)

function [sinalEq] = equalizador(eq, simbolo)

function [desespalhamento] = desespalhamento(wHadamard, sinal, nUsuarios

function [demodulado] = demodulador(hdemod, sinal, M)

function[vetorUsrDeIntlvr] = deintlvr(sinal)

function [decodificado] = decodificador(sinal, n, kb, nUsuarios)

function [decodificado] = decodificadorConv(sinal, kc ,  polinm, nUsuarios)
function [erros] = contadorErros(sinalEnv, sinalGerado, nUsuarios)
```

REFERÊNCIAS

[1] HAYKIN, Simon Haykin. *Communication Systems* - 4ed, Bookman.

[2] HANZO, WEEB, KELLER L. Hanzo, W. T. Weeb, T. Keller "*Single- and Multi-Carrier Quadrature Amplitude Modulation:Principles and Applications for Personal Communications, WLANs and Broadcasting*", IEEE Press-John Wiley, p. 739, 2000.

[3] DINIZ, Paulo Sérgio Ramirez Diniz. *Princípios de comunicações sem fio*. Federal University of Rio de Janeiro, Rio de Janeiro (BR), notas de aula.

[4] F.C.C De Castro e M.C.F. De Castro, PUCRS, Faculdade de Engenharia – Departamento de Engenharia Elétrica da Pontifícia Universidade Católica do Rio Grande do Sul. *Comunicação digital*.

[5] SKLAR, Bernard Sklar. "*Digital communications. Fundamentals and Applications –Second Edition",* University of California, Los Angeles

[6] IDFT/DFT. Disponível em: <http://en.wikipedia.org/wiki/Discrete_Fourier_transform >. Acesso em 21 março de 2009.

[7] Marcio Henrique Doniak, ESTUDO DA TRANSFORMADA DE WALSH-HADAMARD APLICADA A TRANSMISSÃO OFDM, Universidade Federal de Santa Catarina, Programa de Pós - Graduação em Engenharia Elétrica, Florianópolis Setembro de 2006, tese para obtenção de grau em Mestre em Engenharia Elétrica.

[8] Códigos matlab, disponível em: http://www.mathworks.com/matlabcentral/fileexchange/ Acesso 30 março de 2009.

[9] MC-CDMA, DS-CDMA, OFDM e Spreading Codes, em: <http://wireless.per.nl/> Acesso 25 março de 2009.

[10] Thomas Zemen, OFDM Multi-User Comunication Over Time-Variant Channels, DISSERTATION, Dipl.-Ing, Oxford.

[11] Martins, Wallace Alves Martins, "*Transceptores em Bloco com Redundância Mínima*", Universidade Federal do Rio de Janeiro, Programa de Pós - Graduação em Engenharia Elétrica, Rio de Janeiro Março de 2009.