

Análise de Complexidade de Detectores Heurísticos DS/CDMA Implementados em Plataforma DSP

Complexity Analysis of Heuristic DS/CDMA Detectors Implemented in DSP Platform

Rafael de Oliveira Ribeiro¹; Alex Miyamoto Mussi¹; Taufik Abrão²;
Fernando Ciriaco Dias Neto³

Resumo

Este trabalho investiga questões de relativas à complexidade computacional de receptores multiusuários em sistemas de múltiplo acesso por divisão de código e sequência direta (DS/CDMA - *direct sequence/code division multiple access*) baseados em técnicas heurísticas de detecção e implementados a partir de plataforma de processamento digital de sinais (DSP - *digital signal processing*) comercial (). Os algoritmos de detecção multiusuários heurísticos implementados em banda-base foram baseados no princípio heurístico de busca local (LS - *local search*) e por recozimento simulado (SA - *simulated annealing*). São discutidas figuras de mérito relativas ao incremento da complexidade computacional em função do aumento da carga de processamento (neste caso, número de usuários ativos no sistema DS/CDMA), bem como apresentado brevemente a metodologia de projeto/implementação do receptor em banda-base utilizando plataforma DSP.

Palavras-chave: Detecção multiusuário. DSP. DS/CDMA. Algoritmos heurísticos. Complexidade computacional.

Abstract

This work investigates issues concerning the computational complexity of multiuser receivers in direct sequence code division multiple access (DS/CDMA) systems based on heuristic techniques and implemented on commercial platform for digital signal processing (DSP) (*Texas TMS320C6713*). The multiuser detection algorithms which were implemented in baseband are based on the heuristic local search (LS) and simulated annealing (SA) approaches. Figures of merit regarding to the computational complexity increasing, due to the system loading increment (number of active users) are discussed, and a brief methodology for the design/implementation of the baseband receiver using DSP platform is presented as well.

Key words: Multiuser detection. DSP, DS/CDMA. Heuristic algorithms. Computational complexity.

¹ Alunos de mestrado do Departamento de Engenharia Elétrica, Universidade Estadual de Londrina; rafael.rib87@gmail.com – alexmmussi@gmail.com.

² Docente do Departamento de Engenharia Elétrica da Universidade Estadual de Londrina - DEEL-UUEL; taufik@uel.br – www.uel.br/pessoal/taufik.

³ Docente do Departamento de Engenharia Elétrica da Faculdade Pitágoras – fernando@pitagoras.com.br.

Introdução

A busca por maior mobilidade aliado às características de altas taxas de transmissão, fizeram com que novas tecnologias de transmissão/recepção fossem desenvolvidas visando principalmente a otimização de tais sistemas.

O processo de otimização de algum recurso no sistema procura não só determinar as possíveis soluções, como também procura agregar uma maior agilidade nesse procedimento, sendo muitas vezes necessária uma busca extensa e complexa para a determinação das respostas de um determinado problema, como por exemplo, a definição de um máximo ou mínimo em uma determinada função com inúmeras variáveis.

Um aspecto fundamental de muitos problemas de otimização na área de telecomunicações é o fato destes serem dinâmicos, implicando que uma possível solução para um problema num determinado instante poderá não ser a melhor solução poucos segundos subsequentes. Abordagens analíticas ou algoritmos simples podem ser uma maneira para resolver um problema na área de telecomunicações que são dinâmicos, porém, muitas vezes esses problemas mudam rapidamente, fazendo-se necessárias modernas técnicas computacionais para uma busca mais eficiente de soluções, já que em muitos desses problemas o tempo ou o espaço de busca cresce exponencialmente com o aumento das variáveis de entrada (RIBEIRO, 2009).

Algoritmos heurísticos possibilitam diminuir a complexidade em problemas aplicados a sistemas de comunicação sem fio e simultaneamente proporcionam uma maior velocidade na obtenção de resultados quase-ótimos tornando, deste modo, o uso de algoritmos heurísticos extremamente interessante.

Em Rajagopal, Xu e Cavallaro, (1999) foi proposto algoritmos para a estimativa de canal e detecção multiusuário CDMA utilizando plataforma DSP. O detector pode ser implementado

usando um processador de ponto fixo, entretanto para estimação de canal uma plataforma de ponto flutuante teve de ser utilizada devido a um aumento da complexidade, tendo-se inclusive que utilizar de uma memória externa adicionalmente a memória já existente na plataforma DSP.

Em um sistema DS/CDMA, o sinal é recebido e detectado por um detector convencional constituído de um banco de filtro casado (MFB - *matched filter bank*) (DIAS NETO; ABRÃO; JESZENSKY, 2004). Uma vantagem dessa modulação é a redução da sensibilidade do receptor à interferência, onde tal vantagem deve-se ao fato de que o circuito de desespalhamento funciona como um circuito de espalhamento para qualquer sinal para o qual o ele não está casado (HAYKIN; MOHER, 2005).

Entretanto, este tipo de receptor é incapaz de recuperar o sinal completamente, i.e., livre de erros, independentemente se o canal tem ruído aditivo branco Gaussiano (AWGN - *additive white Gaussian noise*), multiplicativo plano ou multiplicativo seletivo em frequência, pois o sistema DS/CDMA tem sua capacidade limitada pela interferência de múltiplo acesso (MAI - *multiple access interference*) e pelo efeito perto-longe (NFR - *near-far ratio*) (DIAS NETO; ABRÃO; JESZENSKY, 2006).

O receptor convencional (CD - *conventional detector*) apresenta uma considerável redução no seu desempenho ao se aumentar o número de usuários ativos no sistema, i.e., o incremento no carregamento do sistema implica em um considerável aumento na interferência de múltiplo acesso, reduzindo assim o desempenho do sistema em termo de taxa de erro de bit, acarretando um redução na capacidade (*throughput*) do sistema, dada uma qualidade de serviço (QoS - *quality of service*) mínima a ser atendida para cada classe de usuário.

A solução para a redução da capacidade do sistema, devido ao incremento da interferência de múltiplo acesso em sistemas CDMA consiste em

lançar mão de vários métodos, que vão desde a utilização de métodos de diversidade⁵ até o uso de métodos de detecção mais eficientes capazes de utilizar a informação dos usuários interferentes para a melhoria da detecção da informação do usuário de interesse. A esse princípio, a literatura denomina detecção multiusuário, tendo sido proposto na década de 1980 por S. Verdú e sistematizado posteriormente em Verdú (1998).

O método de máxima verossimilhança consiste em estabelecer uma função de verossimilhança que expresse a probabilidade do evento observado em função do parâmetro que se deseja estimar (GUIMARÃES, 2005). Assim, a detecção de máxima verossimilhança consiste de um banco de filtros casados seguido de um detector de seqüência de máxima verossimilhança (ML - *maximum likelihood*), onde este detector ML produz uma seqüência de máxima verossimilhança, $\hat{\mathbf{b}} = [\hat{b}_1, \hat{b}_2, \hat{b}_3, \dots, \hat{b}_K]$, em relação à seqüência transmitida. Assim, é estimado o vetor $\hat{\mathbf{b}}$ de forma a maximizar a probabilidade da seqüência de dados ter sido transmitida dado que o sinal foi recebido. Esta probabilidade é denominada de Probabilidade Conjunta a Posteriori, $P_r(\mathbf{b} | r(t), \forall t)$, com a hipótese de todas as mensagens transmitidas serem equiprováveis (DIAS NETO, 2004).

Assim, o desempenho em termos de taxa de erro de bit (BER - *bit error rate*) do detector ML é ótimo, entretanto, este possui uma alta complexidade para a detecção, onde esta complexidade aumenta de forma exponencial com o aumento do número de usuários, inviabilizando a implementação do detector ML em DSP, pois uma busca exaustiva torna-se inviável quando o número de usuários cresce. A utilização de algoritmos heurísticos adaptados ao problema da detecção multiusuário (MUD - *multiuser detector/detection*), permite que se obtenha um desempenho igual ou muito próximo do detector de máxima verossimilhança

(ML), sem entretanto ter que utilizar uma grande complexidade computacional.

Modelo de Sistema

Em um sistema DS-CDMA com modulação BPSK (*binary phase-shift keying*) em canal AWGN, o sinal recebido, em banda base, é dado por:

$$r(t) = \sum_{k=1}^K A_k b_k q_k(t) + n(t) \quad (01)$$

onde K é o número de usuários, A_k é a amplitude do sinal transmitido do k -ésimo usuário, $b_k \in \{-1, +1\}$ é o bit de informação transmitido, q_k é a seqüência de espalhamento do k -ésimo usuário, $t \in [0, T_b]$ onde T_b é o período de bit e $n(t)$ representa o ruído AWGN.

O sinal à saída de um banco de filtros casados (CD) pode ser expresso como:

$$y_k = \int_0^{T_b} r(t) q_k(t) dt = A_k b_k c_k + \sum_{j \neq k} A_j b_j c_j \lambda_{k,j} + n_k \quad (02)$$

onde c_k indica o coeficiente complexo do canal para o k -ésimo usuário, n_k é o ruído AWGN filtrado para o k -ésimo usuário e $\lambda_{k,j}$ denota o k,j -ésimo elemento da matriz de correlação \mathbf{R} .

Em Verdú (1986) foi mostrado que uma solução ótima para recuperar os bits de informação da Equação (1) consiste em estimar a informação transmitida utilizando a saída de um detector de máxima verossimilhança:

$$\hat{\mathbf{b}} = \arg \left\{ \max_{\mathbf{b} \in \{\pm 1\}^K} [2\Re\{\mathbf{y}^T \mathbf{c}^H \mathbf{A} \mathbf{b}\} - \mathbf{b}^T \mathbf{c} \mathbf{A} \mathbf{R} \mathbf{A} \mathbf{c}^H \mathbf{b}] \right\} \quad (03)$$

Na implementação do sistema DS-CDMA, um banco de filtros casados como também o decisor

⁵ Espacial, frequência, micro e macro-diversidades, entre outras.

abrupto foram implementados em DSP, em conjunto com algoritmos heurísticos (LS e SA) a fim de otimizar a recepção do sinal transmitido.

Assim, a estratégia adotada para solucionar o problema MUD através de técnicas heurísticas, consiste na procura de uma melhor solução segundo uma função custo (dentro de um determinado espaço de busca), sendo no contexto MUD a função dada por:

$$F(\mathcal{G}) = 2\mathbf{y}^T \mathbf{A}\mathcal{G} - \mathcal{G}^T \mathbf{A} \mathbf{R} \mathbf{A} \mathcal{G} \quad (04)$$

onde \mathbf{y} é o vetor de saída do banco de filtros casados, \mathbf{A} é a matriz de amplitudes dos sinais dos usuários e \mathbf{R} é a matriz de correlação. A matriz diagonal dos coeficientes do canal da função de verossimilhança é $\mathbf{C}=1$, já que no trabalho o modelo do canal adotado é AWGN síncrono.

Ferramentas Computacionais

Os softwares utilizados no desenvolvimento do trabalho foram o MATLAB® da *Mathworks* e o *Code Composer Studio* (CCS), associado à plataforma de processamento digital de sinais (DSPTMS320C6713 da *Texas Instruments*) (CHASSAING, 2005). Tais ferramentas computacionais, juntamente com a plataforma DSP (*hardware*) foram empregadas na solução do problema da detecção MUD.

Diversas aplicações podem ser desenvolvidas com o uso do MATLAB®, incluindo processamento de imagens e sinais, comunicações, controle, teste, medição, modelagem e análise financeira, entre outros. Como possui diversas funcionalidades, pode-se integrar o seu código com outras linguagens e aplicações computacionais.

O CCS é um programa desenvolvido pela *Texas Instruments* para o desenvolvimento de programas e interface dos modelos de DSP que ela fabrica, inclui ferramentas para a geração de código, apresentando também capacidades gráficas e suporte a depuração dos dados em tempo real. Pode-se passar dados para

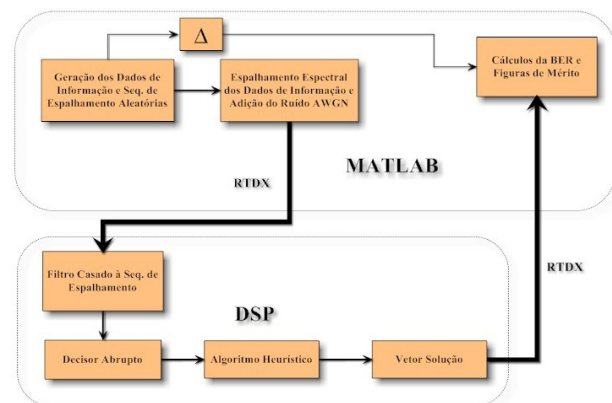
o DSP utilizando a troca em tempo real (RTDX - *Real-Time Data Exchange*) bem como a análise do que está sendo processado na plataforma (MUSSI, 2009).

O MATLAB® foi utilizado no trabalho para o desenvolvimento dos algoritmos, visualização, análise de dados bem como da utilização da ferramenta *Real-Time Workshop* (RTW) do DSP que gera e executa códigos em C, onde o código resultante pode ser utilizado em tempo real para muitas aplicações. Pode-se ajustar o código gerado utilizando-se de recursos internos de análise e/ou execução para interagir com o código fora do ambiente, como foi utilizado no trabalho para a troca de dados entre MATLAB® e o DSP.

Utilizando-se do recurso RTW conseguiu-se trabalhar com dois ambientes de processamento ao mesmo tempo ao empregar o RTDX para troca de dados entre o MATLAB® e o kit de processamento DSK6713 (CCS).

A Fig. (1) mostra o diagrama de blocos do sistema DS/CDMA implementado em DSP e MATLAB® com os algoritmos heurísticos LS e SA.

Figura 1. Diagrama de blocos do sistema DS/CDMA implementado



O RTDX forma dois canais para caminho de dados entre DSP e MATLAB®, onde os dados podem ser inseridos nesses canais de forma

assíncrona, ou seja, a qualquer momento.

O dispositivo de destino envia dados para o computador chamando funções da biblioteca RTDX localizada na plataforma de destino, chamada “RTDX *Target Library*”. Estas funções imediatamente armazenam os dados em um buffer de dados e então retornam. A “RTDX *Target Library*”, em seguida, transmite os dados do buffer para o computador, sem interferir no dispositivo de destino. O computador registra estes dados ou em um buffer de memória ou um arquivo de *log* RTDX, dependendo do modo de gravação no computador especificado na configuração do RTDX. Os dados gravados podem ser recuperados por qualquer aplicativo-cliente no computador que esteja interfaceado com o CCS via RTDX.

Do mesmo modo, um aplicativo-cliente no computador pode enviar dados para o dispositivo de destino. A biblioteca RTDX localizada no CCS, chamada “RTDX *Target Library*”, armazena no buffer todos os dados enviados para o dispositivo de destino. Se a “RTDX *Host Library*” receber um pedido de dados do dispositivo de destino e os dados no buffer do host forem suficientes para atender à solicitação, a “RTDX *Host Library*” envia os dados para o dispositivo de destino. Os dados são gravados para o local solicitado sem interferir com o dispositivo de destino. O *host* notifica o “RTDX *Host Library*” após a conclusão de operação.

A configuração do RTDX é realizada no software CCS através do menu *Tools*, podendo este procedimento ocorrer sem que o DSP esteja conectado.

Por padrão, a configuração inicial do RTDX apresenta 4 *buffers* contendo cada um 1024 *bytes*. Porém, pode-se alterar estes valores e os adequar ao projeto a ser implementado.

Após a configuração de todos parâmetros do RTDX, o projeto do detector foi compilado e o DSP conectado, podendo somente após essas etapas habilitar o RTDX (*Enable RTDX*) para realização

da troca de dados entre o MATLAB® e DSP. Durante a implementação dos algoritmos HEUR-MUD (*heuristic - MUD*) em DSP, foi utilizado os 4 *buffers* com 1024 bytes cada definidos por padrão.

Para não ter que compilar e carregar o projeto em cada simulação, criaram-se funções no MATLAB® para que esses processos fossem realizados automaticamente em cada simulação. Assim, conseguiu-se realizar diversas simulações consecutivamente sem ter que compilar/carregar o projeto no CCS, otimizando o tempo gasto nas simulações já que em cada uma dessas eram necessários que novos parâmetros fossem compilados e carregados no processador.

Descrição do Sistema

Para implementação do sistema DS/CDMA foi utilizado o software MATLAB® para implementar o transmissor bem como a adição do ruído (canal AWGN síncrono). Já o subsistema do Receptor Convencional (CD ou SUD) e do Heur-MuD utilizando o algoritmo de Busca Local 1-ótimo (1-opt LS - *1-optimum local search*) ou simulated annealing (SA) foi implementado em DSP. Este último subsistema (CD e HEUR-MUD) foi realizado a análise de complexidade computacional em função do número de ciclos de cada um.

Em Mussi, Ribeiro e Abrão, (2010) é descrito a metodologia utilizada para implementação de um detector multiusuário com abordagem heurística em DSP.

Detector Convencional

O Detector Convencional em canal AWGN síncrono faz com que o sinal à saída de um banco de filtros casados (MFB) esteja casado à sequência de espalhamento de cada usuário (ABRÃO; CIRIACO; JESZENSKY, 2004), sendo definida pela Equação (2).

O Detector Convencional estima a informação

baseado na polaridade dos sinais à saída do integrador (decisor abrupto), sendo dado por:

$$\hat{\mathbf{b}} = \text{sgn}[\mathbf{y}] \quad (05)$$

Detector Multiusuário baseado em Algoritmo Heurístico

O termo heurística provém do grego *heuriskein*, com significado de descobrir ou achar.

Goldbarg (GOLDBARG; LUNA, 2005) define heurística como um método de busca de soluções em que não há qualquer garantia de que tais respostas tenham sucesso.

Assim, a ferramenta heurística pode ser definida como uma técnica que procura a melhor solução utilizando-se de tecnologia computacional razoável e conseguindo garantir a otimização da solução encontrada ou uma solução viável muito próximo a esta.

Os algoritmos heurísticos utilizados neste trabalho foram o de busca local (LS) e o *simulated annealing* (SA). O algoritmo LS é um método de otimização que se caracteriza pela varredura em uma vizinhança pré-estabelecida do universo de busca (AARTS; LENSTRA, 2003). O algoritmo SA é uma técnica de otimização baseada no processo físico de recozimento, onde uma substância é aquecida a uma elevada temperatura, passando em seguida a ser resfriada progressivamente o que minimiza a sua probabilidade de distribuição de energia (HAUPT; HAUPT, 2004).

Os algoritmos procuram por uma solução ao redor de uma vizinhança composta por todas as possíveis soluções cuja distância de Hamming (KATAYAMA; NARIHISA, 2001), Equação (6), é igual a 1.

$$V(\mathcal{G}^m) = \left\{ \mathcal{G}_i \in \{-1, 1\}^K \text{ tal que } \|\mathcal{G}_i - \mathcal{G}_m\| = 1 \right\} \quad (06)$$

onde \mathcal{G}^m é o vetor-base; M_i é o número máximo de iterações e $m=1, \dots, M$ e $i=1, \dots, K$.

As implementações realizadas neste trabalho foram baseadas no Método de Simulação Monte Carlo (MCS - *Monte Carlo simulation*). As estimativas da quantidade de *bits* a serem transmitidos foram feitas considerando o número de erros estimados no *single-user bound* (SUB).

Complexidade Computacional

Para análise da complexidade computacional dos algoritmos LS e SA em função do número de ciclos realizados pelo DSP, foi utilizado a ferramenta *Profile* do CCS (TEXAS INSTRUMENTS INC., 2005).

Esta ferramenta possibilita monitorar alguns parâmetros do código em execução no processador DSP durante o processamento do código ou parte dele. Com isso, utilizou-se a opção “Cycles” do *Profile* que se refere aos ciclos realizados pelo DSP no processamento do código.

Primeiramente determinou-se a complexidade do detector convencional (apenas MFB e decisor abrupto), realizando vários processamentos em DSP com variação dos parâmetros de número de usuários K e ganho de processamento N , e com isso obtendo-se valores médios de ciclos entre 5 execuções da função.

Observa-se que os ciclos realizados pelo DSP durante a execução da função do CD são proporcionais aos valores K e N . Assim, com os dados obtidos dos ciclos do detector CD em função do fator de proporcionalidade $K \cdot N$, pode-se determinar uma função para linearizar tais dados através da ferramenta “*cftool*” do MATLAB®.

A linearização obtida do número de ciclos para o processamento do CD é dado por:

$$DSP_{CD}^{Cycles} = \frac{1658 + (1120 \cdot K \cdot N)}{T_b} \quad (07)$$

sendo DSP_{CD}^{Cycles} o número de ciclos de

processamento por segundo que o DSP executa para detectar um bit de todos os K usuários com ganho de processamento N , sendo que o ciclo de processamento deve ser realizado dentro de um período de informação detectada, T_b .

Com intuito de realizar a análise da complexidade computacional dos algoritmos LS 1-*opt* e SA, também foi utilizado a ferramenta *Profile* do CCS, onde a função-custo dada pela Equação (4) pode ser simplificada já que o canal adotado é AWGN síncrono.

Observando-se a Equação (4) nota-se que os termos $2\mathbf{y}^T\mathbf{A}$ e \mathbf{ARA} serão os mesmos para todas as iterações, independentemente dos vetores-candidatos. Assim, para reduzir a complexidade computacional durante os processamentos em DSP, calcularam-se estes termos separadamente, Equação (8) e Equação (9), fora do laço de iterações dos algoritmos.

$$F^{(1)} = \text{Re}\{2\mathbf{y}^T\mathbf{A}\} \quad (08)$$

$$F^{(2)} = \text{Re}\{\mathbf{ARA}\} \quad (09)$$

Com isso, em cada iteração os valores de $F^{(1)}$ e $F^{(2)}$ já estavam calculados. A Equação (10) representa a Função Custo de cada iteração.

$$F(\mathcal{G}) = F^{(1)}\mathcal{G} - \mathcal{G}^T F^{(2)}\mathcal{G} \quad (10)$$

Logo, podem-se realizar diversos processamentos em DSP variando-se os parâmetros K e N como realizado anteriormente.

Para obter o número de ciclos dos algoritmos, dividiu-se estes em ciclos da Função Custo Inicial e ciclos das Iterações, obtendo-se valores médios de ciclos entre 5 execuções de cada função.

Os ciclos da Função Custo Inicial foram obtidos analisando-se apenas os cálculos realizados que antecedem o laço das iterações, i.e., os cálculos dos termos $F^{(1)}$ e $F^{(2)}$ seguidos pelo cálculo da função-custo utilizando o vetor à saída do decisor

abrupto.

Para o cálculo dos ciclos da função Iterações foram realizadas iterações até que não ocorresse melhora na função-custo por duas iterações consecutivas.

Analisando-se os dados obtidos percebe-se que as duas funções (Função Custo Inicial e Iterações) dos algoritmos LS e SA dependem apenas do número de usuários, ou seja, o parâmetro de ganho de processamento N não influencia o número de ciclos destas duas funções. Esta tendência pode ser confirmada ao se verificar que o ganho de processamento não participa da execução dos algoritmos, pois, o seu vetor-inicial é dado pelo vetor à saída do decisor abrupto que é um sinal vindo do MFB (sinal desespalhado e com dimensão $K \times 1$).

Em seguida, realizou-se uma linearização para a Função Custo Inicial através de um *Fitting* do tipo *Cubic Polynomial* utilizando-se o MATLAB®. Para a função Iterações realizou-se também o mesmo procedimento.

As linearizações obtidas para os dois algoritmos (Função Custo Inicial e Iterações) são dadas por:

$$DSP_{\text{inc}}^{\text{LS}} = \frac{204,5K^3 + 581K^2 + 2140K + 1597}{T_i} \quad (11)$$

$$DSP_{\text{inc}}^{\text{SA}} = \frac{204,5K^3 + 581K^2 + 2140K + 1597}{T_i} \quad (12)$$

$$DSP_{\text{iter}}^{\text{LS}} = \frac{(52,9K^3 + 318,4K^2 + 564,6K + 393,1)M}{T_i} \quad (13)$$

$$DSP_{\text{iter}}^{\text{SA}} = \frac{(52,9K^3 + 316,9K^2 + 568,6K + 1962)M}{T_i} \quad (14)$$

Entretanto, as Equações (13) e (14) dependem também do número de iterações (M) que os algoritmos devem realizar para convergir para a melhor solução (este valor ocorre em média de $K/2$ iterações). Portanto, como $M = K/2$ estas equações podem ser simplificadas para:

$$DSP_{\mu}^{LS} = \frac{26,5K^4 + 159,2K^3 + 282,3K^2 + 196,6K}{T_i} \quad (15)$$

$$DSP_{\mu}^{SA} = \frac{26,5K^4 + 158,5K^3 + 284,3K^2 + 981K}{T_i} \quad (16)$$

A complexidade computacional dos algoritmos LS e AS será dada pela soma da Função Custo Inicial com a função Iterações, apresentada nas Equações (17) e (18).

$$DSP_{\mu}^{LS} = \frac{26,5K^4 + 364,2K^3 + 854,5K^2 + 2381,6K + 1544}{T_i} \quad (17)$$

$$DSP_{\mu}^{SA} = \frac{26,5K^4 + 363K^3 + 865,3K^2 + 3121K + 1597}{T_i} \quad (18)$$

A complexidade computacional para o receptor HEUR-MUD completo será dada pela soma dos ciclos do detector convencional com os ciclos do algoritmo heurístico, sendo estes apresentados nas Equações (19) e (20), respectivamente.

$$DSP_{\mu}^{LS} = \frac{1120KN + 26,5K^4 + 364,2K^3 + 854,5K^2 + 2381,6K + 3202}{T_i} \quad (19)$$

$$DSP_{\mu}^{SA} = \frac{1120KN + 26,5K^4 + 363K^3 + 865,3K^2 + 3121K + 3255}{T_i} \quad (20)$$

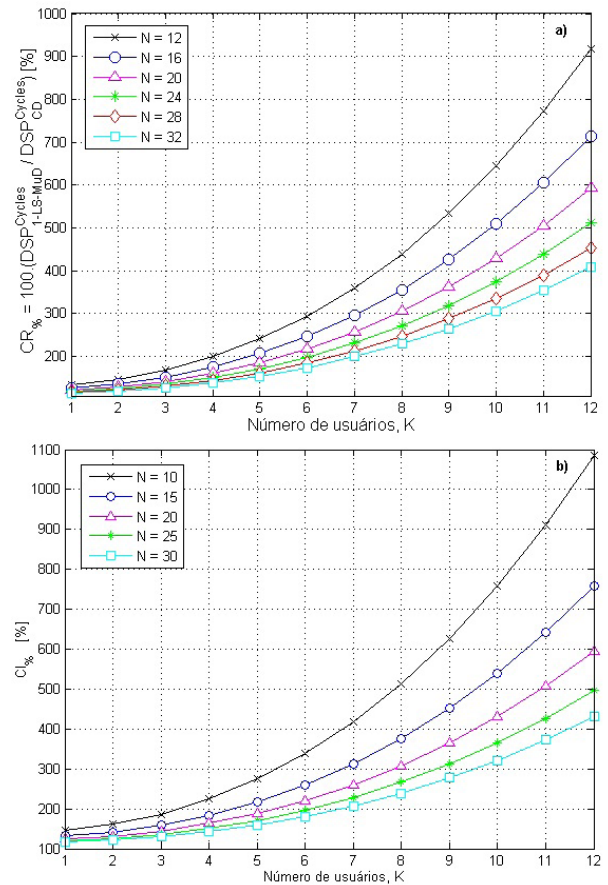
Outra análise realizada foi do incremento de complexidade percentual ($CI_{\%}$), onde este é dado pela razão do número de ciclos do receptor HEUR-MUD em relação ao detector CD.

$$CI_{\%} = \frac{DSP_{\mu}^{Cycles}^{Heur-MUD}}{DSP_{CD}^{Cycles}} \cdot 100 \quad [\%] \quad (21)$$

A Fig. (2) mostra o incremento de complexidade percentual do receptor LS-MUD e SA-MUD para diferentes ganhos de processamento na faixa $N \in [10;32]$. As complexidades para ambos os detectores multiusuários heurísticos apresentam o mesmo comportamento em relação ao número de usuários K e ganho de processamento N , resultando em valores bastante semelhantes de

$CI_{\%}$ para ambos os detectores com mesmo K e N . Observa-se que, como esperado, a complexidade cresce com o número de sinais (i.e.,) processados simultaneamente no DSP. No entanto, nota-se que o incremento de complexidade em relação ao receptor convencional (CD) apresenta tendência decrescente com o ganho de processamento N , indicando que o custo de processamento por chip relativo decresce com o incremento de N .

Figura 2. Gráfico do incremento de complexidade percentual $CI_{\%}$ do receptor heurístico: a) LS-MuD; b) SA-MuD.



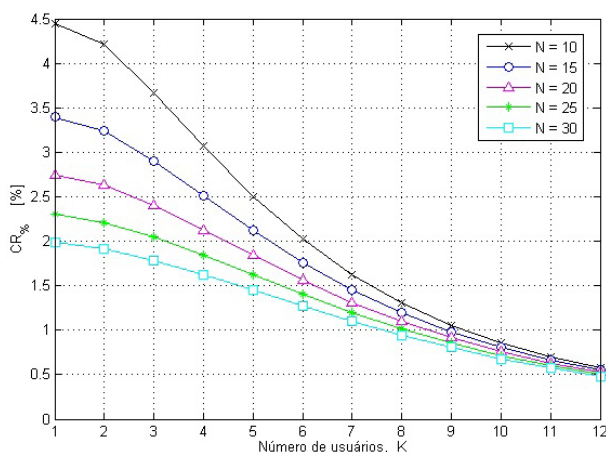
Comparação de Complexidade Computacional entre os Algoritmos LS e SA

Para uma melhor análise e comparação, realizou-se o estudo do incremento de complexidade relativa percentual ($CR_{\%}$), dado por:

$$CR_{\%} = \frac{DSP_{SA-MUD}^{Cycles}}{DSP_{LS-MUD}^{Cycles}} \cdot 100 \quad [\%] \quad (22)$$

A Fig. (3) apresenta este incremento de complexidade do receptor SA-MUD em relação ao receptor LS-MUD para ganhos de processamento $N = 10, 15, 20, 25$ e 30 . Pode-se observar que o algoritmo SA-MUD apresenta um pequeno incremento ($< 5\%$ para uma ampla faixa de ganho de processamento e número de usuários) na complexidade computacional quando comparado à implementação em DSP do algoritmo LS-MUD. Adicionalmente, pode-se inferir da Fig. (3) que esse pequeno incremento na complexidade tende a diminuir convergindo para o valor $CR_{\%} = 0,5\%$ quando o número de sinais de usuários processados simultaneamente, K , cresce. Nesta condição assintótica, as diferenças no ganho de processamento N apresentam pouca influência na complexidade relativa. Em resumo, utilizando-se a mesma metodologia de projeto, as implementações em DSP dos algoritmos LS-MUD e SA-MUD resultam em complexidades muito semelhantes.

Figura 3. Complexidade Computacional LS x SA.



Conclusão

A implementação dos algoritmos heurísticos LS e AS aplicados ao problema de detecção

MUD (canal AWGN síncrono) em plataforma DSP resultou em um desempenho muito próximo ao obtido com o detector ML, porém com uma reduzida complexidade computacional em relação ao detector ótimo. Tal redução de complexidade é proporcionada pelo espaço de busca reduzido inerente ao algoritmo heurístico.

Assim, as duas implementações em DSP para o detector multiusuário em banda-base utilizando os algoritmos heurísticos LS e SA mostraram-se viáveis, mesmo considerando aplicações em tempo real e um número de usuários máximo em torno de uma dezena. Foram implementados não somente a parte heurística, como também o banco de filtros casados e o decisor abrupto (detector convencional), resultando excepcional melhoria de desempenho com acréscimo marginal de complexidade em relação ao detector convencional.

Trabalhos futuros incluem a implementação e caracterização em DSP dos algoritmos HEUR-MUDS sujeitos a canais com desvanecimento uni e multi-percurso e com/sem linha de visada (*Rice e Rayleigh*, respectivamente), bem como a avaliação da degradação de desempenho quando ocorrem erros nas estimativas dos coeficientes de canal.

Referências

- AARTS, E.; LENSTRA, J. K. Local search in combinatorial optimization. *Princeton*, New Jersey: Princeton University Press, 2003.
- ABRÃO, T.; DIAS NETO, F. C.; JESZENSKY, P. J. E. Detecção multiusuário utilizando algoritmos heurísticos evolucionários e de busca local. *Semina: Ciências Exatas e Tecnológicas*, Londrina, v. 25, n. 2, p. 145-162, jul./dez. 2004.
- CHASSAING, R. *Digital signal processor and applications with the C6713 and C6416 DSK*. New Jersey: John Wiley & Sons, 2005.
- DIAS NETO, F. C. *Abordagens Heurísticas aplicadas à detecção multiusuário em sistemas DS/CDMA*. 2004. Trabalho de Conclusão de Curso de Graduação (Graduação em Engenharia Elétrica) - Universidade Estadual de Londrina, Londrina.

- DIAS NETO, F. C.; ABRÃO, T.; JESZENSKY, P. J. E. Algoritmos heurísticos evolucionários aplicados à detecção multiusuário DS-CDMA. In: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES, 21., 2004, Belém. *Anais...* Belem: SBT, 2004.
- _____. *Multirate multiuser DS/CDMA with genetic algorithm detection in multipath channels*. In: IEEE 9th INTERNATIONAL SYMPOSIUM ON SPREAD SPECTRUM TECHNIQUES AND APPLICATIONS, 2006, p. 128-132, Manaus, Brasil.
- GOLDBARG, M. C.; LUNA, H. P. L. *Otimização combinatória e programação linear*. 2. ed. New Jersey: Elsevier, 2005.
- GUIMARÃES, P. J. G. *Estimativa de parâmetros de canal para DS-CDMA baseada na decomposição em subespaços*. 2005. Dissertação (Mestrado em Engenharia) - Escola Politécnica da Universidade de São Paulo, São Paulo.
- HAUPT, R. L.; HAUPT, S. E. *Practical genetic algorithms*. New Jersey: John Wiley and Sons, 2004.
- HAYKIN, S.; MOHER, M. *Modern wireless communications*. New Jersey: Pearson Education, 2005.
- KATAYAMA, K.; NARIHISA, H. Performance of simulated annealing-based heuristic for unconstrained binary quadratic programming problem. *European Journal of Operational Research*, Amsterdam, v. 134, p. 103-119, 2001.
- MUSSI, A. M. *Implementação de subsistemas DS/CDMA utilizando plataforma DSP com abordagem heurística*. 2009. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) - Universidade Estadual de Londrina, Londrina.
- MUSSI, A. M.; RIBEIRO, R. O.; ABRÃO, T. Metodologia de projeto e implementação em DSP de algoritmos heurísticos de detecção multiusuário em sistemas CDMA. *Semina: Ciências Exatas e Tecnológicas*, Londrina, v. 31, n. 2, p. 165-180, jul./dez. 2010.
- RAJAGOPAL, S.; XU, G.; CAVALLARO, J. Implementation of channel estimation and multiuser detection algorithms for W-CDMA on digital signal processors. *Rice University*, Houston, n. 8, p. 259-270, Oct. 1999.
- RIBEIRO, R. O. *Aplicabilidade de heurísticas na otimização de problemas combinatórios em telecomunicações*. 2009. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) - Universidade Estadual de Londrina, Londrina.
- TEXAS INSTRUMENTS INC. TMS320C6000 *Code Composer Studio Tutorial*. Texas Instruments Inc, 2005.
- <http://focus.ti.com/general/docs/lit/getliterature.tsp?baseLiteratureNumber=spru301&track=no>
- VERDÚ, S. *Multiuser detection*. Cambridge: Cambridge University Press, 1998.
- _____. Optimum multiuser asymptotic efficiency. *IEEE Transactions on Communications*, New York, p. 890-897, 1986.
- Recebido em 23 Fevereiro 2011- Received on February 23, 2011.*
- Aceito em 25 Junho, 2011 - Accepted on June 25, 2011.*