

# Competição de Inovação do SBrT 2017- Equalização Adaptativa e Decodificação de Viterbi para transmissão HF Ionosférica visando aplicação no Programa RDS-Defesa

Juraci F. Galdino, Francisco C. Ribeiro Jr., Nilson Maciel P. Jr. e Elaine C. Marques  
Centro Tecnológico do Exército, Rio de Janeiro, Brasil  
E-mails: {jgaldino, nmaciell, melaine}@ctex.eb.br, fcarlos@cpqd.com.br

**Resumo** – Devido à grande extensão territorial, no Brasil as transmissões ionosféricas na faixa de HF são muito importantes, justificando assim investimentos em pesquisas, desenvolvimentos e inovações no sentido de prover comunicações cada vez mais eficazes e eficientes para essa aplicação. Em razão do alcance global, diversas normas foram desenvolvidas para padronizar essas comunicações, sendo uma das principais delas a norma militar MIL-STD-188-110C. Essa competição tem por objetivo propor estratégias de equalização adaptativa e/ou de decodificação de canal para receptores que seguem a norma MIL-STD-188-110C.

**Palavras-chaves** – MIL-STD-188-110C, HF, Forma de Onda, Equalização Adaptativa, Decodificação de Viterbi

## I. INTRODUÇÃO

Transmissões em canais de Alta Frequência (HF, da expressão em inglês *High Frequency*) ionosféricas são muito empregadas em todo o mundo, em que pese as baixas velocidades de transmissão, pois possibilitam o estabelecimento de enlaces digitais de longo alcance, de baixo custo e sem a necessidade de montagem de infraestrutura de comunicações, como as comunicações satelitais. Em razão da grande dimensão territorial e dos imensos vazios de infraestruturas de comunicações em áreas ermas, no Brasil as comunicações HF ionosféricas assumem a condição de assunto estratégico. Elas servem para prover uma rede de comunicação de contingência de abrangência nacional mantida pelo Exército Brasileiro, alcançar áreas inacessíveis às comunicações satelitais, bem como atender com meios de comunicações nos casos de desastres e catástrofes naturais e acidentados em áreas em que não há nenhum tipo de infraestrutura de comunicações disponível.

A fim de padronizar as comunicações HF ionosféricas, foram desenvolvidas diversas normas, dentre elas a MIL-STD-188-110C [1]. Amplamente utilizada, não apenas para aplicações militares, mas também para aplicações civis como as realizadas por rádio amadores, essa norma apresenta requisitos mínimos de desempenho (sob diversas condições de operação e de propagação de canal) e padrões técnicos necessários para garantir a interoperabilidade nas comunicações, bem como padroniza a estrutura do transmissor.

No entanto, ela não especifica alguns componentes importantes do receptor. Dessa forma, fica aberto o campo para a investigação de novas técnicas de recepção que permitam atingir o desempenho almejado com a menor complexidade possível. As principais questões em aberto giram em torno do desenvolvimento de eficientes estratégias de equalização adaptativa, técnicas de sincronização (fase, portadora, de quadro e de símbolo) e de decodificação de códigos corretores de erro convolucionais.

A competição de inovação ora apresentada tem por objetivo motivar a comunidade acadêmica nacional a se envolver em um tema estratégico para as comunicações sem fio, sobretudo para a Defesa e Segurança Nacional. Equipes de pesquisadores, alunos de pós-graduação e graduação poderão propor estratégias de equalização adaptativa e/ou de decodificação de canal para receptores de uma forma de onda especificada na norma MIL-STD-188-110C, submetendo artigo resumido ao Simpósio Brasileiro de Telecomunicações, seguindo procedimentos que serão apresentados na página do evento.

A avaliação das propostas será realizada com base na clareza e formalismo da proposta, bem como nos resultados obtidos tendo como figuras de mérito a taxa de erro de bit (BER – do termo em inglês *Bit Error Rate*) e/ou a complexidade computacional (memória e operações aritméticas). Os competidores deverão descrever as técnicas propostas e os resultados em um documento (artigo resumido) de até 3 páginas, com formato de 2 colunas, incluindo apresentação da proposta, resultados e referências bibliográficas.

A fim de simplificar e facilitar os trabalhos de pesquisa, aos competidores serão disponibilizados, na página do evento, *softwares* desenvolvidos pela equipe de P&D do Projeto RDS-Defesa lotada no Centro Tecnológico do Exército (CTEx). Serão fornecidos o transmissor e o canal implementados em MATLAB em código fechado. Também serão fornecidos o sinal amostrado, na taxa de símbolo, e outros blocos do receptor. Além disso, serão apresentadas curvas de desempenho em termos da taxa de erro de bit e erro quadrático médio (EQM) na saída de um equalizador (DFE) e a taxa de erro de bit na saída de um decodificador de Viterbi que implementa uma regra de decisão abrupta. Tais curvas de desempenho, particularmente as de BER, devem ser utilizadas pelos competidores para fim de comparação.

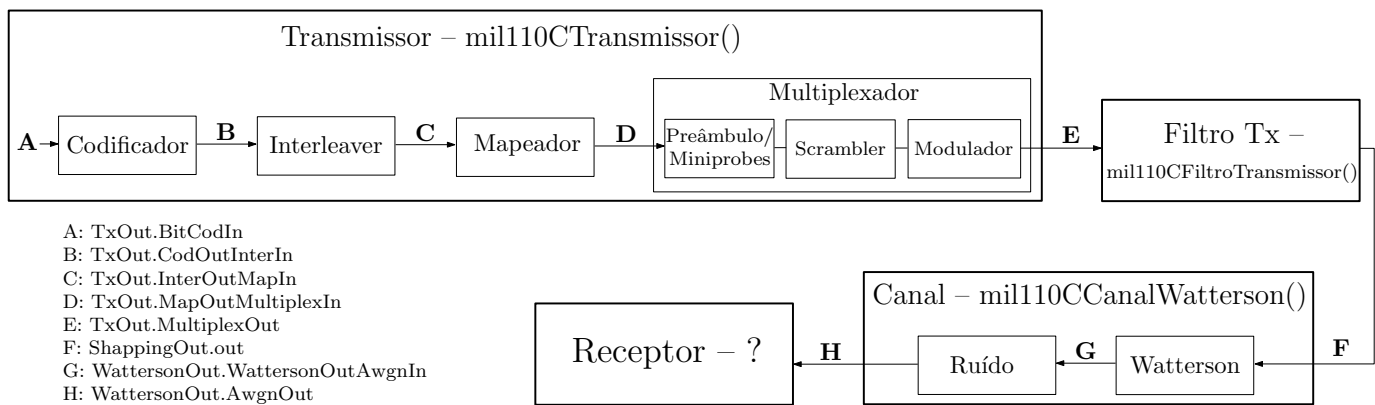


Figura 1. Diagrama de um transceptor compatível com a norma MIL-STD-188-110C.

O restante deste documento é organizado da seguinte forma: a Seção II apresenta características da Norma MIL-STD-188-110C, bem como descreve os *softwares* desenvolvidos pela equipe do Projeto RDS-Defesa e que serão disponibilizados aos competidores. A configuração de referência para a competição é descrita na Seção III. Informações adicionais sobre o desafio e considerações finais são apresentadas na Seção IV.

## II. TRANSECTOR MIL-STD-188-110C

A norma MIL-STD-188-110C apresenta, além de requisitos mínimos de desempenho, diretrizes técnicas para o projeto de transceptores militares operando na faixa de HF utilizando canais de 3 a 24kHz de largura de banda [1].

Em sua parte principal (mandatória), ela estabelece a estrutura de quadros e as técnicas de banda base a serem adotadas no transmissor em função da velocidade de transmissão da camada física, que deve ser escolhida dentre um dos seguintes valores: 75bps, 150bps, 300bps, 600bps, 1200bps, 2400bps e 4800bps. Em particular, um quadro é composto de pré-âmbulo e de sucessivos blocos de dados (*payload*) intercalados por sequências de treinamento (*miniprobos*). Além disso, registra-se que o quadro possui duração fixa de 0,6 ou 4,8 segundos, de acordo com o tipo de *interleaver* escolhido (curto ou longo, respectivamente).

A Figura 1 apresenta a estrutura geral do diagrama de blocos de um sistema de comunicações cujo transceptor segue a norma MIL-STD-188-110C. Nela são representados, com variados níveis de detalhamento, blocos representando o transmissor, o canal e o receptor. Em destaque na figura, três dos principais *softwares* que serão disponibilizados aos competidores: *mil110CTransmissor()*, *mil110CFiltroTransmissor()* e *mil110CCanalWatterson()*.

Cabe destacar que não é objetivo desta Seção discutir as técnicas de transmissão de um sistema que segue a norma MIL-STD-188-110C, nem modelagem dos canais e tampouco as técnicas de recepção. Para isso, o leitor interessado pode recorrer a própria norma e a literatura especializada no assunto. Busca-se tão somente apresentar, de forma sucinta, os programas que serão disponibilizados aos competidores, bem

como algumas informações importantes relacionadas ao tema da competição.

A norma MIL-STD-188-110C especifica detalhadamente as técnicas de transmissão e seus parâmetros, tais como o codificador de canal, a matriz de entrelaçamento (*interleaver*) e o modulador. A equipe do projeto RDS-Defesa, do Centro Tecnológico do Exército (CTEx), implementou a função *mil110CTransmissor()* que compreende parte do transmissor, conforme mostrado na Figura 1, cujo executável será franqueado aos competidores.

Nesse mesmo diapasão, destaca-se que os modelos de propagação para transmissão ionosférica em HF utilizados para validação e avaliação de desempenho dos sistemas de transmissão são bem conhecidos e difundidos na literatura [2]. A própria norma em comento, especificamente no Anexo E, discute o modelo a ser utilizado com as formas de onda especificadas em sua parte mandatória. No intuito de facilitar as atividades de pesquisa dos competidores, a equipe do RDS-Defesa do CTEx desenvolveu e disponibilizará a função *mil110CCanalWatterson()*, em código fechado, que simula os distúrbios do canal de HF ionosférico conforme preconizado na norma. Tal função foi implementada com base na técnica de simulação de Monte Carlo [3].

Em contrapartida, como de praxe nas normas de sistemas de comunicações, o projeto do receptor não é detalhado, ficando a cargo de projetistas a definição das técnicas que serão nele utilizadas, bem como em aberto o campo de investigação para a realização de pesquisa a fim de se desenvolver técnicas de recepção que proporcionem melhores características de desempenho segundo as mais variadas figuras de mérito.

No caso concreto ora em discussão, estão em aberto as técnicas de sincronização de portadora, de quadro e de símbolo; as estratégias de decodificação de canal, que podem considerar abordagens abruptas ou suaves, por exemplo; e, por se tratar de um sistema de comunicação com portadora única ou singela em canal duplamente seletivo (desvanecimento seletivo em frequência e variante no tempo), as técnicas de equalização. Ademais, os pesquisadores podem investigar ainda estratégias que realizam as tarefas de sincronização, equalização e decodificação isoladamente, com algum grau de cooperação ou até mesmo conjuntamente.

O fulcro da competição reside, portanto, em alguns elementos do receptor, os quais são discutidos na Seção Seção II-C. Adicionalmente, nas seções II-A e II-B, são discutidas as funções de implementação do transmissor (*mil110CTransmissor()* e *mil110CFiltroTransmissor()*) e do canal de propagação (*mil110CCanalWatterson()*), respectivamente.

#### A. Transmissor

Na Figura 1 são apresentados blocos da representação, em banda básica, de um transmissor compatível com a norma MIL-STD-188-110C. Os processamentos inerentes a esses blocos foram codificados em duas funções desenvolvidas pela equipe do RDS-Defesa: *mil110CTransmissor()* e *mil110CFiltroTransmissor()*.

A função *mil110CTransmissor()* engloba ou compreende as etapas de Codificação de Canal, *Interleaver*, Mapeamento e Multiplexador, conforme ilustrado na Figura 1. Ela recebe dois parâmetros de entrada: o primeiro, *mil110C*, é um objeto (*struct*) contendo todos os ajustes que orientam a formação dos quadros para a velocidade de transmissão e duração escolhidos. O segundo parâmetro, *varMil110C.nFrames*, é o número de quadros a serem formados. Ao ser executada, ela processa *streams* de dados em cada etapa da cadeia, que são armazenados em vetores. Ao final, ela retorna o objeto *TxOut*, que disponibiliza os vetores de saída de cada uma das etapas, conforme ilustrado na Figura 1.

O primeiro vetor presente no objeto *TxOut* é o *TxOut.BitCodIn*, correspondente ao *stream* de bits gerados internamente na função *mil110CTransmissor()* e que alimenta a entrada do codificador convolucional. Seu tamanho depende do número de quadros a serem formados, da velocidade de transmissão escolhida e da duração do quadro. No entanto, para a forma de onda de 2400 bps, ele conterá múltiplos de 1440 bits em uma duração de 0,6s (no caso de ter sido adotado o *interleaver* curto), ou múltiplos de 11520 bits na duração 4,8s (no caso de ter sido adotado o *interleaver* longo).

O processo de codificação é então feito de acordo com o item 5.3.2.3.3 da norma MIL-STD-188-110C: são usados dois polinômios geradores de comprimento igual a 7, que processam um bit por vez na entrada e geram dois bits na saída. Dependendo da velocidade de transmissão desejada, realiza-se ainda códigos de repetição, reduzindo assim a taxa do codificador de canal, mediante combinação do codificador convolucional com o codificador de repetição. Com isso, as taxas de codificação de canal empregadas pelas formas de onda da parte mandatária do padrão variam entre 1/2 e 1/8. A forma de onda de 2400 bps não emprega código de repetição e seu codificador de canal possui taxa igual a 1/2, que corresponde a taxa do codificador convolucional.

A saída do codificador produz o vetor *TxOut.CodOutInterIn*, que, como o nome sugere, alimenta a entrada da etapa de entrelaçamento (*interleaving*). Seu tamanho é igual ao do vetor *TxOut.BitCodIn* dividido pela taxa de codificação. No caso da forma de onda 2400 bps, para duração 0,6s, ele terá múltiplos de 2880 bits, sendo 2880 o tamanho de um bloco curto de *interleaving*; para duração 4,8s, ele terá múltiplos de 23040 bits,

comprimento de um bloco longo de *interleaving*. O processo de entrelaçamento é descrito nos itens 5.3.2.3.4 e 5.3.2.3.5 da norma. Em linhas gerais, ele consiste no carregamento das colunas de uma matriz de *interleaving* com os bits de entrada, seguindo uma ordem de inserção preestabelecida na norma. A recuperação da sequência correta dos bits no receptor é obtida mediante o descarregamento das linhas dessa matriz segundo uma ordem estabelecida na norma (processo inverso do *interleaver*).

O *interleaver* produz o vetor de saída *TxOut.InterOutMapIn*, de mesmo tamanho do vetor de entrada (*TxOut.CodOutInterIn*). A saída do *interleaver* segue para o processo de mapeamento, que codifica os bits nos símbolos de constelações PSK (*Phase-Shift Keying*) utilizando código de Gray. As constelações disponíveis (modulações 2, 4 e 8-PSK) são utilizadas de acordo com a velocidade de transmissão selecionada. A saída deste bloco é o vetor *TxOut.MapOutMultiplexIn*, formado por números inteiros que especificam os pontos das constelações. O tamanho desse vetor equivale ao da entrada dividido pelo número de bits por símbolo da constelação selecionada. No caso da forma de onda de 2400 bps, cuja modulação é 8-PSK, seu tamanho será um terço do vetor *TxOut.InterOutMapIn*. Assim sendo, considerando a duração de quadro 0,6s, o tamanho do vetor será de múltiplos de 960 símbolos, enquanto para a duração 4,8s será de múltiplos de 7680 símbolos.

Na etapa aqui denominada de multiplexação, o vetor *TxOut.MapOutMultiplexIn*, contendo os símbolos de dados (*payload*), passa por três processos: inserção de preâmbulo/miniprobos, embaralhamento (*scrambling*) e modulação, propriamente dita. A inserção de preâmbulo e miniprobos corresponde à formação da estrutura dos quadros de transmissão.

A estrutura de quadros padrão da norma MIL-STD-188-110C é composta de preâmbulo, *payload* e miniprobos, conforme ilustrado na Figura 2. Comumente utilizado no lado de recepção para auxiliar estratégias de sincronização (início de quadro e relógio), detecções ou estimativas de parâmetros de quaisquer técnicas eventualmente empregadas na camada física, o preâmbulo consiste de repetições de uma sequência básica de 15 tribits, que são símbolos inteiros variando entre 0 e 7, conforme mostrado na Figura 2. Alguns desses tribits possuem valores fixos; outros variam de acordo com a escolha da velocidade de transmissão e da duração dos quadros (neste caso os tribits  $D_1$  e  $D_2$ ) ou da posição da sequência básica dentro do preâmbulo (neste caso os tribits  $C_1$ ,  $C_2$  e  $C_3$ ). Cabe destacar que o número de repetições da sequência básica define o tamanho do preâmbulo; aquele número, por seu turno, é determinado pela duração do quadro. Para quadros de duração de 0,6s, a sequência básica é repetida 3 vezes, enquanto para quadros de duração 4,8s, ela é repetida 24 vezes. Mais detalhes sobre a formação do preâmbulo são encontrados no item 5.3.2.3.7.2 da norma MIL-STD-188-110C.

Convém destacar que o receptor utiliza os tribits  $D_1$  e  $D_2$  para auxiliar na identificação da forma de onda que está sendo recebida.

Os dados a serem transmitidos são inseridos nos *payloads*, entre os quais são encaixados os *miniprobos*. Estes contem

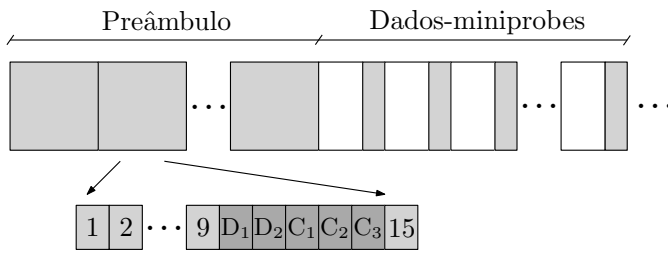


Figura 2. Estrutura de quadros da norma MIL-STD-188-110C.

seqüências padronizadas pela norma (seqüências de treinamento) que podem ser empregadas para facilitar o rastreo de parâmetros do receptor que variam com o tempo. A forma como *payloads* (dados) e *miniprobres* (seqüências de treinamento) são intercalados depende da velocidade de transmissão. Na forma de onda de 2400 bps, por exemplo, 16 símbolos de miniprobres são inseridos a cada 32 símbolos de dados; nesse caso, cada transmissão de um *payload* curto de 960 símbolos leva à formação de um quadro de dados-miniprobres com  $30 \times (32 + 16) = 1440$  símbolos, enquanto cada *payload* longo de 7680 símbolos é transmitido em um quadro de dados-miniprobres com  $240 \times (32 + 16) = 11520$  símbolos. A geração e inserção dos miniprobres é tratada em detalhes no item 5.3.2.3.7.1.2 do padrão.

Ainda na etapa de multiplexação, os quadros passam pelo processo de embaralhamento (*scrambling*), de acordo com o item 5.3.2.3.8.1 da norma MIL-STD-188-110C. Resumidamente, esse processo consiste em uma soma módulo 8 dos símbolos com uma seqüência de tribits definida pela norma e gerada por um registrador de deslocamento.

O embaralhamento confere características pseudoaleatórias às informações contidas nos blocos de *payload*, bem como às seqüências de treinamento, além de distribuir uniformemente os símbolos a serem transmitidos entre os números inteiros de 0 a 7, fazendo com que, no processo de modulação subsequente, todos os símbolos sejam mapeados em uma constelação 8-PSK, independentemente da velocidade de transmissão selecionada (ou seja, mesmo para uma forma de onda que emprega uma modulação BPSK, por exemplo, ela é efetivamente transmitida como 8-PSK, isso facilita sobretudo o projeto do receptor, particularmente as estimativas iniciais para a formação da estrutura de quadros a ser adotada no próprio receptor). Já o embaralhamento do quadro de preâmbulo ocorre separadamente, conforme descrito no item 5.3.2.3.8.2 da norma. Finalmente, os quadros embaralhados passam pelo modulador, que converte os símbolos reais inteiros em seus correspondentes valores complexos na constelação 8-PSK.

A saída da etapa de multiplexação é o vetor *TxOut.MultiplexOut*, que também é a saída final da função *mil110CTransmissor()*. Esse vetor contém os quadros MIL-STD-188-110C prontos para transmissão. É importante notar que o tamanho desse vetor não depende da velocidade de transmissão, mas apenas do número de quadros e da duração do quadro, esta definida pelo tamanho do *interleaver*. Dessa maneira, um número de bits equivalente a um bloco de

*interleaving* curto (longo), independentemente da taxa de bits, levará sempre à transmissão de um quadro de preâmbulo de 1440 (11520) símbolos e de um quadro de dados-miniprobres de 1440 (11520) símbolos. Como a taxa de sinalização do transmissor especificada pela norma é de 2400 baud, chega-se à duração de  $1440/2400 = 0.6s$  ( $11520/2400 = 4.8s$ ) por quadro, conforme esperado.

Considerando o sistema em banda básica, o último passo antes da transmissão dos quadros pelo canal é a transformação dos símbolos 8-PSK em pulsos limitados em banda. Para essa etapa de formatação de pulso, a norma MIL-STD-188-110C recomenda (não é obrigatória a adoção desse tipo de filtro) o uso do pulso raiz quadrada de cosseno levantado (*Square-Root Raised Cosine* - SRRC) com fator de *rolloff*  $\beta = 0.35$ , valor que garante confinamento de 99% da energia do sinal dentro da banda de interesse de 1500Hz (banda base). Os participantes da competição que decidirem seguir a recomendação da norma podem utilizar a função *mil110CFiltroTransmissor()* para a formatação de pulso. Ela recebe, como parâmetros de entrada, o vetor *TxOut.MultiplexOut* com os quadros e o objeto *varShapping*, que contém os parâmetros de ajuste do pulso SRRC: fator de *rolloff*, retardo e fator de superamostragem (*varShapping.oversamplingFactor*). O processamento da função consiste, basicamente, na superamostragem dos símbolos PSK seguida de convolução com o filtro SRRC. Como resultado, ela retorna o vetor resultante da convolução, *ShappingOut.out*, cujo comprimento corresponde ao do vetor de entrada multiplicado pelo fator de superamostragem, acrescido ainda das amostras de retardo introduzidas pelo filtro.

## B. Canal de propagação

Os efeitos inerentes à comunicação na faixa de HF, como dispersões no domínio do tempo e da frequência ocasionados pelas reflexões ionosféricas, são geralmente reproduzidos com o modelo proposto por Watterson em [2]. Esse modelo consiste em um filtro transversal com 2 componentes (raios) de mesma potência média, variantes no tempo e cujos espectros Doppler são iguais e possuem formato dado pela função densidade de probabilidade (*fdp*) de uma distribuição Gaussiana de média nula. A variância dessa função caracteriza a intensidade do efeito Doppler. Assim sendo, define-se uma família de modelos, em que um modelo em particular é especificado por dois parâmetros: o atraso temporal entre os raios e o espalhamento Doppler (variância da *fdp* Gaussiana). Cabe destacar que quanto maiores os valores desses dois parâmetros, mais difíceis tendem a ser as condições de propagação do canal e mais severo e complexo tende a ser o padrão de interferência interssimbólica no receptor. Três conjuntos de parâmetros bem difundidos na literatura [4] são contemplados na norma, os quais são comumente referidos como canal “Bom”, canal “Moderado” e canal “Ruim”. A Tabela I resume os parâmetros que definem esses três modelos.

O modelo de canal proposto na norma para avaliação de desempenho das técnicas de recepção é bastante complexo e impõe grandes desafios não apenas para realizar o ajuste de parâmetros do receptor, mas também para as atividades de pesquisa que visam a desenvolver novas técnicas de recepção.

Tabela I  
TIPOS DE CANAL.

| Tipo de Canal              | Bom   | Moderado | Ruim |
|----------------------------|-------|----------|------|
| Atraso entre os raios (ms) | 0,5ms | 1ms      | 2ms  |
| Espalhamento Doppler (Hz)  | 0,1Hz | 0,5Hz    | 1Hz  |

Cabe destacar que, além dos efeitos de desvanecimento variante no tempo e seletivo em frequências, requerendo o emprego de técnicas de equalização adaptativa sofisticadas, os parâmetros do modelo do canal que definem os perfis de espalhamento de atraso e do efeito Doppler variam de acordo com as características de propagação do canal. As pesquisas realizadas pela equipe do RDS-Defesa do CTEEx indicam que a própria quantidade de parâmetros e ajustes desses parâmetros podem ser diretamente influenciadas pelos perfis do modelo do canal (atrasos entre os raios e variância do espectro Doppler).

Para os parâmetros do modelo denominado de “Ruim”, embora a intensidade de espalhamento Doppler seja aparentemente pequena quando comparada com a de outras aplicações (comunicações móveis, por exemplo), ela é suficiente para causar efeito de desvanecimento rápido em razão da baixa taxa de símbolos adotada pela norma MIL-STD-188-110C em sua parte mandatória; na presença desse canal, o receptor precisa contar com algoritmos adaptativos com boa capacidade de rastreo.

Por outro lado, com os parâmetros do modelo estabelecidos para canal “Bom”, a resposta ao impulso do canal de comunicação varia mais lentamente, não requerendo do equalizador adaptações tão críticas nem a compensação de Interferências Interssimbólicas muito longas (*long span*). Em contrapartida, a dinâmica lenta do canal pode resultar em longos períodos de “apagamento” (vales profundos de desvanecimento - provocando, além da Interferência Entre Símbolos, uma baixa razão sinal ruído instantânea), que podem provocar longas rajadas ou surtos de erros na saída do equalizador, dificultando assim as condições de operação do codificador/decodificador de canal, requerendo elevadas capacidades de correção e/ou *interleavers* eficientes, com grande profundidade. Isso, além de tornar a decodificação de canal mais complexa, pode aumentar o atraso na recepção dos dados, um aspecto crítico, uma vez que o enlace de HF ionosférico possui, por natureza, uma latência elevada.

Há ainda outros distúrbios associados à transmissão ionosférica que não são abarcados no modelo aqui discutido. Por exemplo, sinais trafegando por esse canal sofrem intensa interferência de fontes de ruído gaussiano e não-gaussiano. Outro fenômeno conhecido é o efeito de dutos [5], em que sinais ionosféricos gerados em uma longínqua região do planeta podem alcançar outros continentes e interferir nas comunicações neles realizadas.

O simulador de canal disponibilizado aos participantes por meio da função *mil110CCanalWatterson()* reproduz o modelo previsto na norma. Ele foi implementado com base na técnica de Monte-Carlo, conforme detalhado em [3]. Além disso, essa função inclui também a possibilidade de inserção de ruído gaussiano branco aditivo (*Additive White Gaussian Noise - AWGN*), que emula fenômenos de natureza interna ao receptor

sobre o sinal (ruído térmico) e interferências de outros sistemas de transmissão. Seus principais parâmetros de entrada são o vetor de saída do formatador de pulso (*ShappingOut.out*) e o objeto *varWatterson*, que contém variáveis para seleção do submodelo Watterson (parâmetros que definem os espalhamentos nos domínios do tempo e da frequência) e a razão sinal-ruído média do enlace, dada por  $E_s/N_0$ , em que  $E_s$  é a energia média do símbolo e  $N_0$  define a densidade espectral de potência do ruído. É importante destacar que os efeitos de dutos e de interferências não gaussianas não foram incluídos no simulador.

O processamento da função *mil110CCanalWatterson()* é realizado em duas etapas, conforme ilustrado na Figura 1. Na primeira, geram-se as duas cópias de sinal distorcidas (multiplica o vetor *ShappingOut.out* pelas componentes de canal formadas com o método de Monte-Carlo), introduzem-se as amostras de retardo entre as duas cópias e as soma, produzindo o vetor intermediário *WattersonOut.WattersonOutAwgnIn*. Na segunda etapa, o AWGN é adicionado a esse vetor, dando forma ao vetor de saída final, *WattersonOut.AwgnOut*. Um detalhe importante sobre a primeira etapa é que as amostras de retardo são calculadas pelo produto  $AER \times 2400 \times varShapping.oversamplingFactor$ , em que AER representa os atrasos entre raios, cujos valores são apresentados na Tabela I. Nesse sentido, o fator de superamostragem ajustado no filtro de transmissão deve ser tal que o cálculo do retardo resulte em um número inteiro, para evitar erros na simulação desse importante parâmetro do canal. Para evitar esse tipo de erro, a função *mil110CCanalWatterson()* realiza uma checagem interna, a partir do objeto *varShapping* a ela fornecida. Após o processamento da referida função, a saída final *WattersonOut.AwgnOut* terá comprimento igual ao da entrada, acrescido das amostras de retardo.

### C. Receptor

O vetor *WattersonOut.AwgnOut* representa a versão superamostrada na entrada de um receptor de banda básica, de um sinal gerado por um transmissor MIL-STD-188-110C, transmitido através em um canal HF ionosférico e corrompido por ruído gaussiano. Assim sendo, este sinal está em condições de ser utilizado pelo receptor para recuperar a informação transmitida.

Em que pese o receptor não ser especificado pelo padrão MIL-STD-188-110C, essa norma estabelece requisitos mínimos de desempenho a serem alcançados no enlace, bem como define a estrutura da cadeia de recepção, em face da padronização do transmissor, requerendo assim a adoção de alguns processamentos inversos àqueles realizados pelo transmissor. Componentes que realizam esses processamentos inversos também são fornecidos, em código fechado, aos participantes da competição e são discutidos adiante.

Em sistemas de comunicação com portadora única em canal cuja resposta ao impulso é desconhecida (situação típica de sistemas sem fio), o início da cadeia de recepção é geralmente formada por um filtro de recepção, um amostrador e um equalizador para compensar os efeitos da interferência interssimbólica [6]. Na prática, há ainda um quarto componente e

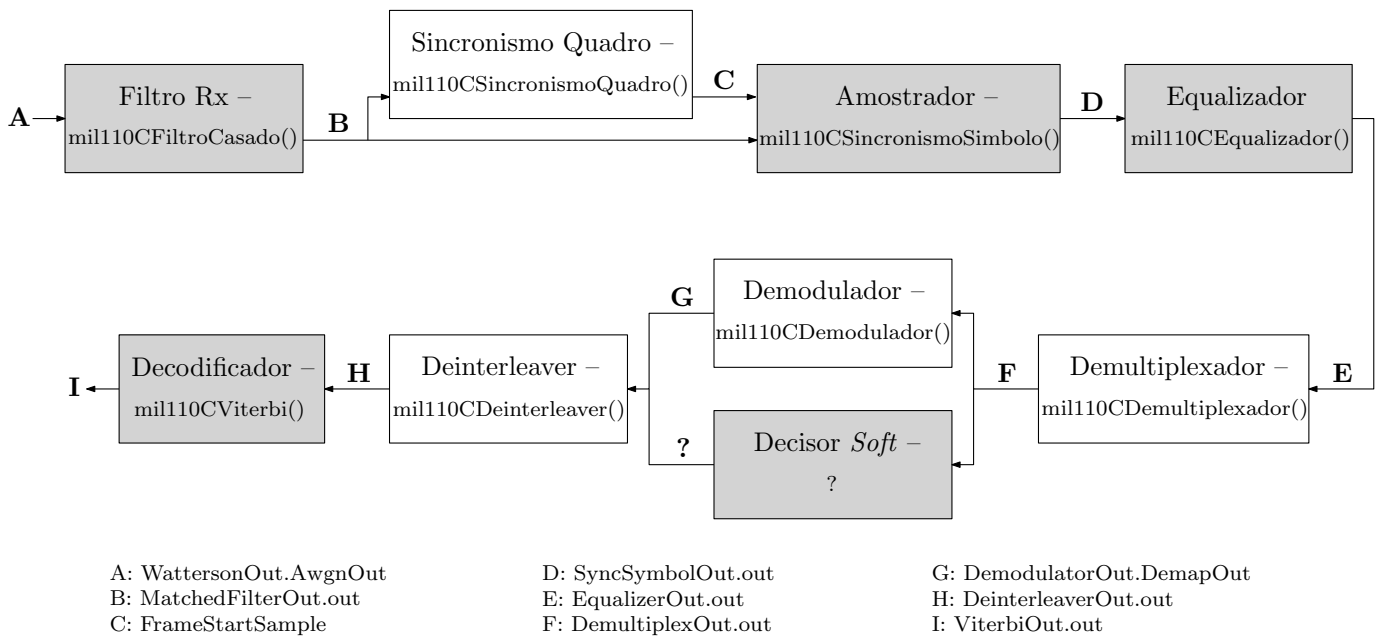


Figura 3. Diagrama sugerido para o projeto do receptor.

que é fundamental, o sincronizador de início de transmissão (ou de quadro), bem como o de identificação do instante ótimo de amostragem de sua cadência cujo papel é identificar não apenas em que instante, dentro do *stream* de amostras de entrada, começa uma transmissão útil, mas também qual amostra contém o instante ótimo de amostragem. Essas informações são fundamentais para conferir bom funcionamento ao restante da cadeia de recepção. Juntos, esses quatro componentes constituem o que é aqui denominado de subprocesso de detecção, pois seu objetivo precípua é detectar os símbolos a partir dos pulsos de sinal distorcidos pelo canal. Eles são ilustrados nos blocos da parte superior do diagrama da Figura 3.

Competidores interessados nesse tema podem abordar o problema de forma abrangente, isto é, tratá-lo de forma integrada, ou separadamente, focando em componentes específicos. Neste último caso, o componente mais importante do subprocesso de detecção é o equalizador. Dependendo da natureza do canal e de restrições de processamento no receptor, diferentes tipos de equalizadores podem ser implementados. Para canais com seletividade mais branda, isto é, que impõem pouca distorção em frequência ao sinal, equalizadores lineares convencionais podem proporcionar desempenho adequado e, como apresentam em geral baixa complexidade computacional, são excelentes opções. Já canais com presença de nulos espectrais e grandes espalhamentos nos domínios de tempo e da frequência, em geral, requerem soluções mais robustas (e complexas), a exemplo do equalizador com realimentação de decisão [7].

Estratégias de equalização adaptativas são indispensáveis para o estudo de caso em questão [8], porém os competidores devem atentar para proporem técnicas que propiciem um bom compromisso entre complexidade computacional e velocidade de rastreamento dos parâmetros do equalizador. Um aspecto importante na aplicação em comento é a necessidade de reduzir ao

máximo a complexidade do processamento digital de sinais, visto que, entre os principais óbices do desenvolvimento de rádios táticos portáteis, estão o consumo e o aquecimento. Soluções de baixo consumo e que requeiram baixa velocidade de processamento são desejáveis. Os equipamentos devem ser leves, possuir alcances elevados e ter grande autonomia, pois, quando em missão, é desejável que eles possam ser operados por um longo período de tempo, sem substituição de baterias.

Muitos trabalhos propõem estimar os parâmetros do equalizador a partir do sinal de observação na saída do amostrador, outros a partir da estimação da resposta ao impulso do canal [9], [10]. Neste caso, a ideia é explorar o fato de que a quantidade de parâmetros a ser estimada (coeficientes da resposta ao impulso do canal) tende a ser significativamente menor do que a quantidade de parâmetros do equalizador, podendo assim reduzir a complexidade computacional e, ao mesmo tempo, aumentar a velocidade de convergência. Um problema crítico dessa abordagem é a obtenção recursiva dos parâmetros do equalizador a partir de atualizações dos coeficientes da resposta ao impulso do canal. Finalmente, a esparsidade característica de alguns canais de propagação de HF motiva o desenvolvimento de algoritmos que processem apenas coeficientes principais de seus filtros, reduzindo a complexidade do receptor. Trabalhos nessa linha de pesquisa podem ser encontrados enfocando o problema de estimação de canal [11] e também de equalização direta [12], inclusive em HF [13].

Outras linhas de pesquisa lidam com o filtro de recepção, o amostrador e o equalizador de maneira integrada, com vista ao melhor compromisso possível entre desempenho próximo do receptor ótimo e complexidade computacional reduzida [6]. A estrutura de detecção mais recorrente na literatura é composta por um filtro casado ao pulso de transmissão, seguido de amostrador e equalizador, ambos operando na taxa de símbolo.

Embora essa seja uma opção mais simples, operar na taxa de símbolo significa, em geral, amostrar o sinal de entrada a uma taxa inferior à de *Nyquist*, produzindo distorções nos símbolos resultantes e levando a um desempenho de equalização sensível ao instante de amostragem [14], [15]. Uma forma de reduzir esse problema é implementar amostradores mais sofisticados, cujos algoritmos buscam o ponto ótimo de amostragem iterativamente [16], [17]. É possível, entretanto, eliminar esse problema com o uso de estruturas de detecção com superamostragem [14], [15]. Estruturas desse tipo consistem, geralmente, em um filtro passa-baixas seguido de amostrador e equalizador, ambos operando a taxas superiores à de símbolo; dessa forma, eles não somente eliminam a sensibilidade à amostragem como também se aproximam da implementação do receptor ótimo [18]. Em contrapartida, receptores trabalhando com superamostragem podem elevar a complexidade a níveis que não compensem o ganho de desempenho [9], [19].

Após a detecção dos símbolos, procede-se a fase de decodificação de canal, cujos componentes são representados na parte inferior da Figura 3. Para o caso de estudo em discussão, o campo de pesquisa é mais restrito, pois o codificador de canal e o *interleaver* empregados no transmissor são especificados pela norma. Os componentes demultiplexador, demodulador e *deinterleaver*, por exemplo, extraem o *payload* dos quadros, demodulam os símbolos e desentrelaçam os bits desfazendo processos realizados no transmissor, os quais são desenvolvidos de acordo com o preconizado na norma MIL-STD-188-110C.

Nesse sentido, o decodificador convolucional é o componente principal desse processo. Embora seu funcionamento também seja condicionado a elementos do transmissor, como polinômios geradores e tipo de terminação da treliça (truncado, terminado em zeros etc.), sua implementação pode variar conforme o algoritmo adotado para a montagem da treliça e com o tipo de dado processado pelo decodificador. O algoritmo mais difundido na literatura em decodificação convolucional, por oferecer melhor desempenho de detecção por máxima verossimilhança, é o algoritmo de Viterbi [20]. Como sua complexidade cresce exponencialmente com a memória (*constraint length*) do codificador, a aplicação dessa estrutura de decodificação pode ser inviável em alguns cenários, sobretudo quando a complexidade computacional é uma figura de mérito importante. Alternativas comuns são o algoritmo sequencial proposto por Fano em [21] e o algoritmo BCJR, que é baseado no critério de maximização *a posteriori* [22].

Cabe destacar que o decodificador pode receber decisões *hard* (binárias) tomadas pelo demodulador acerca dos bits transmitidos, ou pode receber informações *soft*, munindo-se assim de confiabilidades sobre essas decisões. A estratégia *hard* simplifica o projeto do decodificador, mas não oferece ganho significativo de desempenho em cenários de baixa SNR. Já a estratégia *soft* pode proporcionar desempenho bem superior, mas sua implementação passa pelo projeto de um decisor *soft* capaz de calcular a razão de verossimilhança (*Log-Likelihood Ratio* - LLR) de cada bit transmitido, o que é uma tarefa consideravelmente complexa [23]. Alguns trabalhos na área de decodificação *soft* focam no desenvolvimento de

algoritmos para simplificar o cálculo analítico da LLR [23], enquanto outros aproximam o cômputo da LLR explorando propriedades específicas das constelações [24], [25], [26].

Os dois subprocessos descritos nesta seção – subprocessos de detecção e de decodificação – constituem, portanto, as duas linhas de investigação sugeridas aos participantes da competição. Grupos que optarem pelo subprocesso de detecção poderão investigar um ou mais dos blocos em destaque na parte superior da Figura 3, enquanto aqueles que decidirem pelo subprocesso de decodificação poderão focar em um ou ambos os blocos destacados na parte inferior da figura. Nesse sentido, o papel dos códigos do receptor, do transmissor e do canal é formar uma cadeia de recepção de referência de desempenho (conforme será visto na Seção III) e oferecer estruturas básicas para ambos os subprocessos, de maneira que participantes que optarem por um subprocesso foquem em seu desenvolvimento e aproveitem a estrutura básica do outro.

Os componentes fornecidos para o receptor estão em códigos-fonte separados, cujos nomes são mostrados em diversos blocos da Figura 3. O filtro de transmissão, por exemplo, corresponde à função *mil110CFiltroCasado()* e, como o nome sugere, implementa um filtro casado ao pulso transmitido – no caso, o pulso SRRC. Como parâmetros de entrada, ela recebe o vetor de saída do simulador de canal, *WattersonOut.AwgnOut*, e o objeto *varShapping*, o mesmo utilizado no lado da transmissão para a formatação de pulso. A saída da função é o vetor *MatchedFilterOut.out*, contendo o sinal novamente confinado em 1500Hz (banda base) após eliminação do ruído fora da banda de passagem. Seu comprimento corresponde ao do vetor de entrada acrescido das amostras de retardo introduzidas pelo filtro.

O vetor *MatchedFilterOut.out* segue a entrada do sincronizador de quadro, implementado pela função *mil110CSincronismoQuadro()*. Por simplicidade, o sincronizador fornecido considera que sempre há transmissão: ele apenas identifica em que momento ela se inicia. Seu processamento consiste na correlação do vetor de entrada com uma parte do preâmbulo, a fim de identificar qual dos dois componentes de multipercurso do canal ionosférico é mais forte e, então, sincronizar o resto da cadeia de recepção em torno desse componente. A saída da função é um número inteiro, *FrameStartSample*, representando a posição detectada do primeiro símbolo no vetor *MatchedFilterOut.out*: quando o primeiro componente (raio) de canal é predominante, a posição detectada tende a ser em torno da amostra seguinte à soma dos retardos introduzidos pelos filtros de transmissão e recepção; quando o segundo raio é o predominante, a amostra detectada tende a ser aquela seguinte à soma dos atrasos dos filtros com o atraso entre os raios do canal. Na realidade, como se trata de uma correlação, pode ocorrer uma combinação convexa da posição dos dois componentes em cenários em que eles estiverem muito próximos (canal “Bom”), resultando em eventuais posições iniciais no meio dos dois raios.

Uma vez identificada a amostra de início da transmissão, ela é repassada, juntamente com o vetor *MatchedFilterOut.out*, ao amostrador, implementado pela função *mil110CSincronismoSimbolo()*. Ele simplesmente realiza a

amostragem do vetor de entrada na taxa de símbolo – 2400 baud – a partir do ponto entregue pelo sincronizador de quadro. A saída é o vetor *SyncSymbolOut.out*, com símbolos 8-PSK complexos. O comprimento desse vetor é igual ao do vetor de saída do transmissor (*TxOut.MultiplexOut*) acrescido de alguns símbolos no final, correspondentes ao espalhamento de atraso provocado pelo canal.

É válido ressaltar que um receptor adequado para ser utilizado com a norma MIL-STD-188-110C precisa identificar qual forma de onda (taxa de bits e duração do quadro) foi transmitida, a fim de configurar corretamente a cadeia de recepção. Isso seria tarefa de um componente adicional capaz de extrair essas informações constantes no preâmbulo – os tribits *D* mencionados na Seção II-A. Para efeito da competição, entretanto, essa funcionalidade não foi incluída nos programas disponibilizados, por simplicidade, uma vez que poucas configurações de transmissão serão objeto de estudo: forma de onda de 2400 bps com durações de quadro 0,6s (*interleaving* curto) e 4,8s (*interleaving* longo). Vale mencionar que, para repassar informações concernentes a essas configurações, a maioria das funções do transmissor e do receptor tem, como parâmetro de entrada, o objeto *mil110C*.

Após a amostragem, o vetor *SyncSymbolOut.out* segue para a entrada do equalizador, que, nos programas disponibilizados para os competidores, foi implementado pela função *mil110CEqualizador()*. A saída do equalizador é o vetor *EqualizerOut.out*, formado por símbolos 8-PSK, em notação complexa, que representam versões detectadas dos símbolos presentes na saída do transmissor (vetor *TxOut.MultiplexOut*).

Em seguida, o vetor *EqualizerOut.out* segue para o bloco aqui denominado de demultiplexador, implementado pela função *mil110CDemultiplexador()*. Diferentemente do multiplexador empregado no lado de transmissão e que reúne três tarefas principais, o demultiplexador realiza a extração do *payload* da estrutura de quadros e desembaralhamento dos dados contidos no *payload*, assim, desfazendo duas das tarefas realizadas no multiplexador. Isso é feito para deixar disponíveis, na saída do componente, os símbolos complexos não-demodulados, com os quais os competidores que optarem por investigar a decodificação podem desenvolver alguma técnica de decisão *soft* (não disponível no código). Esses símbolos de saída do demultiplexador estarão no vetor *DemultiplexOut.out*, que, em condições de canal ideal, será idêntico ao vetor auxiliar *TxOut.ComplexPSKSymbols*, disponibilizado na saída do transmissor especificamente para essa comparação.

A demodulação passa então a ser feita em um componente à parte, implementado pela função *mil110CDemodulador()*. Essa função, na realidade, realiza duas tarefas: a de demodulador, recuperando os símbolos PSK reais a partir dos símbolos complexos da entrada *DemultiplexOut.out*; e a de demapeador Gray, recuperando os bits representados pelos símbolos. A saída é o vetor *DemodulatorOut.DemapOut*, contendo decisões *hard* sobre os bits transmitidos e que, em condições de canal ideal, será idêntico ao vetor *TxOut.InterOutMapIn* no transmissor.

Os dois últimos componentes da cadeia de recepção fornecida são o desentrelaçador (*deinterleaver*) e o decodificador, representados, respectivamente, pelas funções

*mil110CDeinterleaver()* e *mil110CViterbi()*. Vale destacar que ambos podem processar tanto informações *hard*, provenientes do demodulador, quanto informações *soft*, advindas de um decisor. O desembaralhador simplesmente realiza, sobre o vetor de entrada, o processamento inverso àquele determinado na norma MIL-STD-188-110C para a etapa de embaralhamento. A saída é o vetor *DeinterleaverOut.out*, que, em condições ideais, será idêntico ao vetor *TxOut.CodOutInterIn*. Finalmente, o decodificador implementa o algoritmo de Viterbi para decodificar a informação de entrada e disponibilizar os bits na saída *ViterbiOut.out*, que idealmente será uma versão totalmente recuperada dos bits transmitidos (vetor *TxOut.BitCodIn*).

### III. CONFIGURAÇÕES DE REFERÊNCIA PARA A COMPETIÇÃO

Como referência da forma de onda *narrowband*, 3kHz<sup>1</sup>, implementou-se um padrão de comunicação de camada física de acordo com a parte principal da norma MIL-STD-188-110C [1] que oferece taxa de transmissão de 2400 bps, modulação 8-PSK e largura de banda de 3 kHz. O script *mil110CSimulador.m*, que compõe o pacote de códigos disponibilizado aos participantes, oferece um simulador desse padrão e utiliza as funções descritas na seção anterior. Para gerar os resultados mostrados nesta seção, ele foi utilizado com o *interleaver* configurado na opção "LONGO". O canal simulado foi o canal "Ruim" da Tabela I (canal CCIR *Poor* [1]).

Implementou-se um equalizador com realimentação de decisão (DFE - *Decision-Feedback Equalizer*) utilizando o algoritmo *Inverse QR-RLS* (IQR-RLS), que é uma variação numericamente estável do algoritmo *Recursive Least Squares* (RLS) [8], [19].

Um dos parâmetros de entrada da função que implementa o equalizador é o objeto *varEqualizer*, que carrega variáveis de ajuste do equalizador e do algoritmo adaptativo: número de coeficientes diretos e de realimentação, retardo de equalização e fator de esquecimento. O equalizador opera diferentemente em dois períodos: no período de treinamento, ele extrai mini-probes do vetor de entrada *TxOut.MultiplexOut* para treinar o algoritmo adaptativo; no período de decisão direta, ele usa símbolos detectados internamente como referência para o algoritmo. O parâmetro de entrada que orienta a alternância entre ambos os períodos no tempo correto é o vetor binário auxiliar *TxOut.ProbeInfo*, gerado pelo transmissor e cujo formato segue a estrutura de quadro apresentada na Figura 2.

A título de ilustração, foram usados 16 coeficientes no filtro avante do DFE e 5 coeficientes no de realimentação. O fator de esquecimento do algoritmo IQR-RLS foi 0,975. Finalmente, para a decodificação convolucional, utilizou-se o algoritmo de Viterbi com decisão abrupta.

A Fig. 4 mostra os resultados de BER na saída do equalizador ("Saída do DFE") e do decodificador ("Saída do

<sup>1</sup>A norma permite utilizar larguras de banda maiores, mas para efeito de simplificação e delimitação de escopo, optou-se nessa competição por escolher apenas uma forma de onda convencional de 3kHz de largura de banda. No âmbito do projeto RDS-Defesa, essa forma de onda é denominada de *narrowband*, em contraposição as formas de onda *wideband* que empregam larguras de banda maiores, de até 24kHz



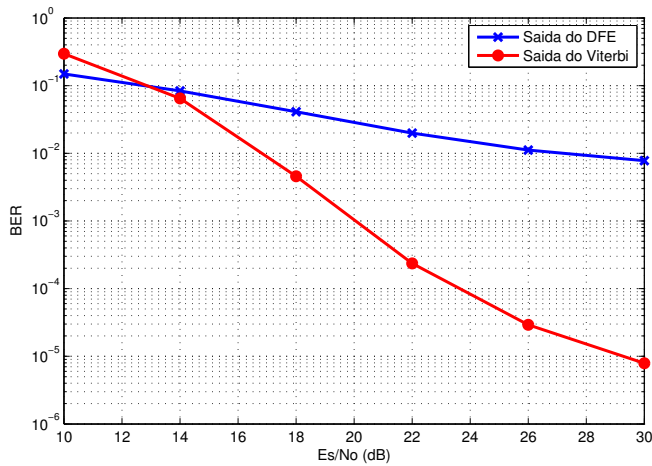


Figura 4. Curva de BER em função da  $E_s/N_0$  (dB).

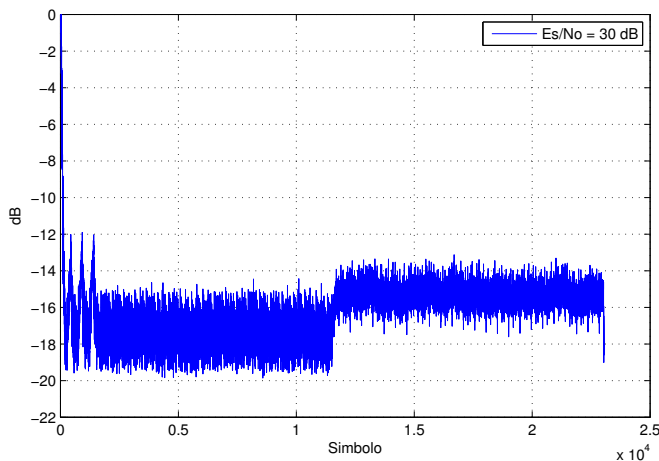


Figura 5. Erro quadrático médio (EQM).

Viterbi”) obtidos para uma faixa de razão sinal-ruído (*Signal-to-Noise Ratio* - SNR), aqui definida como  $E_s/N_0$ , em que  $E_s$  é energia média do símbolo e  $N_0$  é a densidade espectral de potência do ruído AWGN. Já a Fig. 5 apresenta o erro quadrático médio (EQM) na saída do equalizador para uma SNR específica (30 dB). Cada ponto da curva de BER (assim como a curva de EQM) é um resultado médio de 5000 transmissões independentes, cada uma destas de um bloco de *interleaver* longo (11520 bits). Seus valores são apresentados na Tabela II.

#### IV. DETALHAMENTO DO DESAFIO E CONSIDERAÇÕES FINAIS

A Estratégia Nacional de Defesa [27] preconiza maior integração entre as Instituições Científicas e Tecnológicas (ICT) civis e militares, indústria e universidade, bem como uma maior participação dos diversos setores civis da sociedade brasileira em assuntos de Defesa e de Segurança Nacional.

Há diversos assuntos estratégicos para a área de Defesa e Segurança Nacional que envolvem temas de pesquisas de

Tabela II  
BER.

| Es/No | Sem Codificação   | Com Codificação   |
|-------|-------------------|-------------------|
| 10    | 0.148774236111111 | 0.295769166666666 |
| 14    | 0.083499296875000 | 0.064853628472222 |
| 18    | 0.041083993055556 | 0.004573958333333 |
| 22    | 0.019910312500000 | 0.000236076388889 |
| 26    | 0.011178246527778 | 0.000029201388889 |
| 30    | 0.007777065972222 | 0.000007899305556 |

emprego dual, ou seja, que possibilitam aplicações e inovações voltadas tanto para setores civis quanto militares. Uma área em que isso ocorre com grande intensidade é o das Comunicações Sem Fio, amplamente utilizados nos sistemas de comunicações táticos e estratégicos das Forças Armadas, os quais estão repletos de questões em aberto que se assemelham aquelas investigadas em diversas áreas de pesquisas destinadas às comunicações civis.

Malgrado os competidores sejam incentivados a suplantarem os desempenhos fornecidos como referências, o objetivo principal de curto prazo desta competição é identificar grupos de pesquisa interessados em pesquisar assuntos voltados para a área de Defesa e de se integrar com especialistas das ICT militares na busca de solução de problemas genuinamente nacionais, bem como colaborar com a inovação no setor e com a Base Industrial de Defesa.

Em médio prazo, busca-se criar condições para se estabelecer uma “Rede Nacional de Pesquisa em Comunicação Estratégicas e Táticas” com os grupos de pesquisa que participarem da competição em comento, cujo fulcro será o de promover a Pesquisa, o Desenvolvimento e a Inovação autóctone em áreas sensíveis das comunicações táticas e estratégicas das Forças Armadas.

Essa rede terá como objetivos precípuos ampliar a cooperação entre academia e ICT militares; difundir na comunidade acadêmica temas estratégicos à área de Defesa e Segurança Nacional na área de comunicações sem fio (RDS, Rádio Cognitivo, Redes de Cognição, Cibernética operando nas faixas de HF, VHF e UHF); promover maior interação entre Academia - Governo (via ICT militares) - e Indústria. Propiciando assim melhores condições de aumentar a capacidade tecnológica instalada em nosso país no setor e criando melhores condições para a geração de inovação.

Para participar da competição os interessados poderão utilizar os programas que serão disponibilizados na página do evento e que implementam procedimentos de transmissão e recepção, além do canal de comunicação. Os programas são implementados em MATLAB em código fechado. Caberá aos competidores implementar propostas para o processamento de detecção, cujo elemento principal é o equalizador, ou propostas voltadas para o processamento de decodificação do código corretor de erros, e comparar o desempenho das estratégias propostas com os dados de desempenho de referência Fig. 4 e Tabela II. Tais resultados de desempenho podem ser reproduzidos com os programas disponibilizados.

Vale destacar que os competidores que apresentarem propostas de processamento de detecção devem também incluir, em seus resultados, curvas de desempenho na saída do decodificador. Resultados de pesquisas realizadas no bojo do

projeto RDS-Defesa comprovam que tão importante quanto as características de rastreamento dos parâmetros do equalizador e baixa taxa de erro em sua saída, são as características de correlação dos erros na saída do equalizador. Quanto mais correlacionados forem esses erros piores, em geral, serão os resultados de desempenho na saída do decodificador de canal, que em última instância é a figura de mérito mais importante de um sistema de transmissão [28], [13].

Por fim, cabe mencionar que apesar de apenas terem sido fornecidos resultados de desempenho para o Canal Ruim e *interleaver* longo, os competidores são incentivados a apresentarem também resultados de avaliação de desempenho de suas propostas para os demais tipos de canais e *interleaver* curto. As curvas de desempenho de referências para esses cenários podem ser obtidas com os próprios *softwares* fornecidos aos competidores.

#### REFERÊNCIAS

- [1] D. O. D Interface Standard, "Interoperability and Performance Standards for Data Modems: MIL-STD-188-110C," 2011.
- [2] C. Watterson, J. Juroshek, and W. D. Bensema, "Experimental Confirmation of an HF Channel Model," *IEEE Transactions on Communication Technology*, Dec. 1970.
- [3] A. G. Guimarães, E. L. Pinto, J. F. Galdino, and C. J. Silva, "Comparação de Desempenho de Simuladores de Canais com Desvanecimento Rápido - Parte I - Avaliação Numérica," *Anais do XV Simpósio Brasileiro de Telecomunicações*, pp. 426–430, 1997.
- [4] ITU, "HF Ionospheric Channel Simulators," *Recommendations and Reports of the CCIR*, vol. III, pp. 59–67.
- [5] K. Davies, *Ionospheric Radio*. The Institution of Engineering and Technology, 1990.
- [6] C. J. L. Pimentel, *Comunicação Digital*. Brasport, 2007.
- [7] D. George, R. Bowen, and J. Storey, "An Adaptive Decision Feedback Equalizer," *IEEE Transactions on Communication Technology*, 1971.
- [8] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, 3rd ed. Springer, 2008.
- [9] B. Farhang-Boroujeny, "Channel Equalization via Channel Identification: Algorithms and Simulation Results for Rapidly Fading HF Channels," *IEEE Transactions on Communications*, November 1996.
- [10] R. Otne, "Improved receivers for digital High Frequency communications: Iterative channel estimation, equalization, and decoding," Ph.D. dissertation, Norwegian University of Science and Technology, 2002.
- [11] S. F. Cotter and B. D. Rao, "Sparse Channel Estimation via Matching Pursuit with Application to Equalization," *IEEE Transactions on Communications*, Mar. 2002.
- [12] E. Vlachos, A. S. Lalos, and K. Berberidis, "Stochastic Gradient Pursuit for Adaptive Equalization of Sparse Multipath Channels," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, Mar. 2012.
- [13] F. C. R. Junior, E. C. Marques, N. M. de Paiva Junior, and J. F. Galdino, "Sparsity-aware Direct Decision-feedback Equalization of Ionospheric HF Channels," in *IEEE Military Communications Conference*, Oct. 2015.
- [14] R. D. Gitlin and S. B. Weinstein, "Fractionally spaced equalization: an improved digital transversal equalizer," *Journal Bell System Technical*, vol. 60, pp. 275–296, 1981.
- [15] S. Binggeli, "Fractionally Spaced Equalizer for a NLOS Receiver in the 60 GHz Band," Master's thesis, University of California, 2011.
- [16] A. Gardner, "BPSK/QPSK timing-error detector for sampled receivers," *IEEE Transactions on Communications*, vol. 34, no. 5, pp. 423–429, 1986.
- [17] W. Chung, "Adaptive blind timing recovery methods for MSE optimization," *Journal on Advances in Signal Processing*, 2012.
- [18] S. U. H. Qureshi, "Adaptive Equalization," *Proceedings of the IEEE*, vol. 73, no. 9, 1985.
- [19] F. C. R. Junior, E. C. Marques, N. M. de Paiva Junior, and J. F. Galdino, "Avaliação de Desempenho de Equalizadores DFE Adaptativos em Enlaces HF Ionosféricos que Empregam a Norma MIL-STD-188-110C," *Cadernos CPqD Tecnologia*, vol. 10, pp. 101–112, 2014.
- [20] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.
- [21] R. M. Fano, "A Heuristic Discussion of Probabilistic Decoding," *IEEE Transactions on Information Theory*, vol. 9, no. 2, pp. 64–73, May 1963.
- [22] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, no. 284–287, Mar. 1974.
- [23] C. Xu, D. Liang, S. Sugiura, S. X. Ng, and L. Hanzo, "Reduced-Complexity Soft-Decision Aided PSK Detection," in *IEEE Vehicular Technology Conference*, Sep. 2012.
- [24] M. K. Cheng, D. Divsalar, and S. Duy, "Structured Low-Density Parity-Check Codes with Bandwidth Efficient Modulation," *SPIE Proceedings: Wireless Sensing and Processing*, May 2009.
- [25] J. Su, Z. Lu, X. Yu, and C. Hu, "A Novel Low Complexity Soft-decision Demapper for QPSK 8PSK Demodulation of DVB-S2 Systems," in *International Conference of Electron Devices and Solid-State Circuits*, Nov. 2011.
- [26] A. Barré, E. Boutillon, N. Blas, and D. Diaz, "A polar-based demapper of 8PSK demodulation for DVB-S2 systems," in *IEEE Workshop on Signal Processing Systems*, Oct. 2013.
- [27] M. da Defesa, *Estratégia Nacional de Defesa*, 2008. [Online]. Available: [http://www.defesa.gov.br/arquivos/estado\\_e\\_defesa/END-PND\\_Optimized.pdf](http://www.defesa.gov.br/arquivos/estado_e_defesa/END-PND_Optimized.pdf)
- [28] A. H. G. Samary, "Técnicas de Combate à Propagação de Erros de Decisão em Equalizadores DFE," Master's thesis, Instituto Militar de Engenharia, 2006.