

Introdução ao Desenvolvimento de Rádios Definidos por Software para Aplicações de Defesa

Nilson Maciel de Paiva Junior *, Elaine Crespo Marques *, Fabrício Alves Barbosa da Silva *, Robson França de Moraes *, David Fernandes Cruz Moura * e Juraci F. Galdino*†

* Núcleo de Inovação e Pesquisa em Comunicações Aplicadas à Defesa, Divisão de Tecnologia da Informação, Centro Tecnológico do Exército

† Seção de Engenharia Elétrica, Instituto Militar de Engenharia

Resumo—O constante desenvolvimento de novas tecnologias empregadas nos sistemas de comunicações tem possibilitado o aumento do tráfego de informação e usuários assim como a diminuição dos custos. Nos últimos anos, a pesquisa e o desenvolvimento de Rádios Definidos por Software (RDS) vem substituindo as implementações tradicionais em *hardware* para implementações mais flexíveis e programadas em *software*. Este trabalho apresenta um estudo de caso do desenvolvimento de um rádio definido por software voltados para aplicações de defesa, com o emprego do padrão aberto de software SCA e da plataforma de *hardware* de baixo custo USRP na implementação de um transmissor e um receptor FM.

Palavras-Chave—Rádio Definido por Software, SCA, USRP, OSSIE

Abstract—The constant development of new technologies used in communications systems have enabled the increase of traffic information and users as well as reducing costs. In recent years, research and development for Software Defined Radio (SDR) has been replacing traditional deployments in hardware, bringing more flexible implementations by programming radio applications mainly in software. This paper presents a case study of the development of a defense-oriented software-defined radio, with the use of SCA open standard software platform and low-cost USRP hardware platform, with the implementation of a FM transmitter and receiver.

Keywords—Software Defined Radio, SCA, USRP, OSSIE

I. INTRODUÇÃO

No passado, o projeto de rádios militares foi totalmente orientado ao emprego de componentes eletrônicos dedicados. Em seguida, tem-se assistido ao aparecimento de rádios de software configuráveis, nos quais os usuários têm a oportunidade de escolher as formas de onda mais adequados para diferentes cenários de combate. Nos últimos anos, porém, o desenvolvimento de soluções tecnológicas em radiocomunicação foi submetido a uma enorme mudança de paradigma - o Rádio Definido por Software (RDS), tecnologia emergente em que os recursos anteriormente baseados em *hardware* tornou-se definidos por software e os usuários podem também introduzir novas aplicações no decorrer da utilização do equipamento rádio.

Este progresso é devido a várias melhorias em diversas áreas, como sistemas embarcados, conversores analógico-digitaes, transmissão digital, processamento digital de sinal,

antenas multi-banda, arquiteturas de *software* e, especialmente, na capacidade de execução de novos processadores de propósito geral. Com base nisso, os RDS antecipam como conseqüência importantes vantagens para o desenvolvimento de soluções sem fio em sistemas de comunicações militares. Entre as características imaginadas, podemos listar a interoperabilidade entre tropas de diferentes forças e nações, a portabilidade de formas de onda, e a possibilidade de atualização com os mais recentes avanços nas comunicações de rádio sem os requisitos de substituição de *hardware*. Além disso, o RDS é visualizado como a plataforma mais adequada para o desenvolvimento de rádios cognitivos.

À primeira vista, o modelo de alto nível funcional de um RDS é constituído por um subsistema de *front-end* de RF, que realiza seleção de canais, *down-conversion* para a banda básica, e o encaminhamento de dados para uma unidade de processamento baseado em *software*. Nessa unidade, o conjunto de dados digitais associado é apresentado a diversas camadas (por exemplo, módulos de enlace, rede, segurança) para executar tarefas de decodificação adequados à extração da informação desejada. Este processo é invertido no lado de transmissão, onde o sinal de entrada é codificado e modulado, de forma adequada para a transmissão da informação. Este sinal é então passado para o subsistema de RF para inserção no canal rádio.

Devido à multiplicidade de conceitos relacionados com o modelo funcional descrito no parágrafo anterior, vários esforços têm sido feito para a padronização de elementos-chave dentro da arquitetura RDS, proporcionando uma plataforma comum para o desenvolvimento de equipamentos rádio. Os padrões suportados podem ser proprietários ou da indústria, sendo desenvolvidos por meio de um processo de consenso - enquanto a primeira abordagem traz diferenciação do produto para os fabricantes, a última estratégia comoditiza a tecnologia, permitindo o suporte de terceiros na criação da plataforma de rádio para alcançar os objetivos de negócios específicos.

Uma das zonas mais típicas de padronização é o *framework* de aplicações, que proporciona um ambiente de operação de *software* comum, no qual o fornecedor disponibiliza de forma livre um conjunto de interfaces para instalar, configurar, controlar e liberar a operação de aplicações em uma plataforma SDR. Como exemplos de *frameworks* de aplicações relevantes para os sistemas SDR incluem-se o Open Mobile Alliance, o Fórum de Disponibilidade de Serviços e o Software Communications

Architecture (SCA), apoiado pelo Grupo de Trabalho do Fórum SDR SCA. Em especial, o padrão SCA foi originalmente proposto pelo programa Joint Tactical Radio System (JTRS) [1], que é um programa para o desenvolvimento de rádios para operações militares patrocinado pelo Departamento de Defesa dos EUA. Assim, o padrão SCA/JTRS está se tornando o padrão de fato para a construção de rádios militares; no entanto, o interesse na SCA vai além do domínio militar, uma vez que este padrão tem inspirado projetos acadêmicos e comerciais [2].

Assim, o objetivo deste trabalho é apresentar um estudo de caso do desenvolvimento de um rádio definido por *software*, no qual foram empregados o padrão aberto de software SCA e a plataforma de *hardware* de baixo custo USRP na implementação de um transmissor e um receptor FM. O restante deste trabalho está organizado como se segue: a Seção II apresenta o SCA. A seção III apresenta a ferramenta de desenvolvimento OSSIE, seguida da descrição da plataforma de *hardware* USRP na Seção IV. A implementação do transmissor e do receptor FM é apresentada na Seção V e os comentários finais acerca do trabalho se encontram na Seção VI.

II. SOFTWARE COMMUNICATIONS ARCHITECTURE

O padrão SCA [1] é uma infraestrutura de *software* aberta, desenvolvida pelo Departamento de Defesa dos EUA, através do programa JTRS (*Joint Tactical Radio System*), voltada para o desenvolvimento de rádios definidos por *software*. O padrão especifica os mecanismos para criar, implantar, gerenciar e interconectar aplicações rádio, baseadas em componentes, em plataformas distribuídas. O SCA foi proposto para aumentar consideravelmente a interoperabilidade de comunicação de sistemas rádio, bem como reduzir o tempo de implantação e custos de desenvolvimento de aplicações rádio.

No contexto SCA, uma aplicação rádio é denominada *forma de onda*, a qual é definida como o resultado de um conjunto de transformações realizadas com o objetivo de superar distúrbios, causados seja pela propagação em ambiente rádio, seja em função de possíveis ações de interferência inimigas. Tais transformações são realizadas no transmissor e aplicadas à informação veiculada através do canal rádio. Além disto, esta definição também contempla o conjunto correspondente de transformações no receptor para converter os sinais recebidos, recuperando a informação original.

Existe uma grande variedade de formas de onda típicas [3], [4], contemplando protocolos das camadas física, de enlace de dados e de rede. Tais formas de onda descrevem mecanismos de segurança na transmissão de dados, codificação de fonte (voz, imagem e compressão de vídeo), codificação de canal, com mecanismos de retransmissão, bem como de detecção e correção de erros, técnicas de modulação e demodulação, equalização adaptativa, controle automático de ganho, sincronização, filtragem e controle de acesso ao meio, dentre outras funcionalidades [5].

Assim, o SCA foi desenvolvido como uma arquitetura escalável, capaz de suportar plataformas de diferentes capacidades, indo de centros de comunicação fixos até dispositivos portáteis.

Esta escalabilidade, juntamente com padrão de implantação de formas de onda, melhora muito a portabilidade de aplicações de rádio. Um rádio SCA pode incluir elementos de processamento de diversos tipos, como DSPs, FPGAs e processadores de uso geral (GPP), dentro de um mesmo rádio.

A estrutura de *software* do padrão SCA define um ambiente operacional (AO) comum. O ambiente operacional prevê mecanismos comuns para implantar aplicações em diferentes plataformas com diferentes componentes de *hardware*, controladores de dispositivo ou mecanismos de transporte. O ambiente operacional também define interfaces para gerenciar e controlar as aplicações e seus componentes, sendo que estas interfaces são independentes de implementações particulares.

O Ambiente Operacional é dividido em quatro camadas, fornecendo níveis distintos de abstração:

- Sistema operacional em tempo real (RTOS)
- *Middleware* (baseado em CORBA)
- *Core Framework* (CF)
- Serviços CORBA

O SCA demanda um sistema operacional compatível com o padrão *Portable Operating System Interface* (POSIX). O SCA define um subconjunto das interfaces POSIX, chamado de *Application Environment Profile* (AEP), que limita os serviços do Sistema Operacional disponíveis para as aplicações rádio (formas de onda). Os componentes do *Core Framework*, por sua vez, têm acesso total ao sistema operacional. O padrão SCA também define o uso opcional de serviços CORBA de registro (*logging*), de eventos e de nomes, como especificados pelo *Object Management Group* (OMG) [6].

III. OSSIE

O OSSIE (*Open-Source SCA Implementation Embedded*) é uma ferramenta de desenvolvimento de código aberto baseado no padrão SCA usado para implementar rádios definidos por *software* [2], [7]. O projeto OSSIE é desenvolvido na Universidade Virginia Tech (VT) e fornece diversas ferramentas de projeto que permitem o desenvolvimento de formas de onda, componentes e aplicações baseadas na arquitetura SCA. Essas ferramentas ajudam a geração e depuração visual de aplicações rádio, incluindo a geração automática de código para facilitar a conformidade com o SCA [1]. A existência de tais funcionalidades reduz a necessidade de programação manual, bem como o risco de erros de programação. Toda a infraestrutura do *software*, bem como as ferramentas de desenvolvimento e a biblioteca de componentes já implementados podem ser obtidos em [7].

Segundo a equipe de desenvolvimento, os principais objetivos do projeto OSSIE são o ensino, a pesquisa, o desenvolvimento e rápida prototipagem de RDS. [7]. Como exemplo, pode-se citar exercícios de laboratório desenvolvidos pela VT e outras universidades, além de relatos de uso em cursos e mini-cursos ministrados pela VT na própria universidade e em diversos congressos [8].

Uma das principais vantagens da ferramenta consiste em sua integração com placas USRP [9], por meio de componentes

implementados que permitem controlá-las via USB e Ethernet. Dentre outras vantagens, pode-se citar:

- Fácil uso de ferramentas RDS, com facilidade de expansão para um novo nível de aplicação interativa, nas fases de projeto, testes e configuração;
- Desenvolvimento do RDS baseado em SCA;
- Baixo custo de desenvolvimento com rápida prototipagem; e
- Portabilidade para outras plataformas de desenvolvimento baseadas em SCA.

O desenvolvimento de componentes no OSSIE emprega as linguagens de programação Python ou C++. Além disso, dispõe de um ambiente de desenvolvimento do projeto denominado OEF (*Ossie Eclipse Feature*), o qual é integrado ao Eclipse [10] e opera com quatro elementos distintos [11], listados a seguir:

- Um *componente* é a unidade básica de execução dentro do OSSIE, como um filtro, modulador AM ou um módulo de controle automático de ganho
- Um *dispositivo* é uma abstração lógica de um dispositivo físico de *hardware* dentro do sistema, como uma placa de som, um processador GPP ou um dispositivo USRP.
- Um *nó* representa o computador ou o rádio, onde os componentes serão executados.
- Uma *forma de onda* descreve as transformações realizadas nos sinais transmitidos ou recebidos, bem como as interconexões implementadas entre os componentes, dispositivos e nós.

Cada componente do OSSIE possui entradas e saídas que podem ser associadas aos diversos tipos de fluxos de dados. Além disso, cada componente pode possuir diversos parâmetros que definem seu comportamento. O OSSIE já fornece alguns componentes para usos mais comuns como AGC (controle automático de ganho) e amplificador, mas a maioria dos componentes disponíveis têm propriedades reconfiguráveis. Os valores de propriedades de cada componente podem ser definidos como instâncias padrão em formas de onda específicas desenvolvidas utilizando o ambiente OEF, e podem ser configuradas de forma interativa em tempo de execução usando o *software WaveDash*. Além desses componentes já disponibilizados, o usuário pode criar novos componentes de acordo com sua necessidade.

Como exemplo de tais possibilidades de expansão, considere-se os passos necessários para a criação de um componente. Assim, para a criação do referido novo componente no OSSIE, deve-se criar um novo projeto de componente a partir da ferramenta apropriada de desenvolvimento, denominada *OSSIE Component Development* (OCD). O OCD é composto por 4 partes:

- *Description*, na qual o usuário escreve uma descrição do componente que está desenvolvendo.
- *Generation Options*, para o fornecimento de opções ao componente, como a escolha da linguagem para a implementação.
- *Ports*, permitindo a colocação das portas de entrada e de saída do componente, bem como a definição dos tipos de

porta (entrada ou saída) e de dado (por exemplo, real ou complexo).

- *Properties*, para definição das propriedades (parâmetros) do componente - por exemplo, se o valor é simples ou um agregado de dados, se o tipo de dado é *boolean*, *short*, *double*, *string*, o tipo de acesso aos dados (*readonly*, *readwrite*, *writeonly*) e o valor padrão do componente. Tais propriedades são variáveis que podem ser alteradas pelo usuário sem alterar o código do componente, inclusive em tempo de execução do componente.

Depois de se definir todas as portas e propriedades de um componente, existe a geração automática de arquivos XML para a implementação e o correto funcionamento do componente de acordo com o SCA. Por fim, para a implementação precisa das funcionalidades do componente criado, deve-se alterar o método *ProcessData()* na classe gerada que recebe o nome dado ao projeto do componente. A partir da conexão entre diferentes componentes, cria-se uma forma de onda. Tal criação pode ser realizada em um ambiente gráfico, o *OSSIE Waveform Development* (OWD), a qual permite ao usuário fazer a interconexão dos componentes para a criação de formas de onda ou mesmo a construção customizada de componentes, além de identificar todos os componentes e dispositivos utilizados. O OWD é composto por 3 partes:

- *Available Resources*, para apresentação dos componentes implementados, dispositivos e nós disponíveis no sistema.
- *Waveform*, a qual exibe os componentes que compõem a forma de onda e suas conexões.
- *Platform*, para veiculação dos nós e dispositivos que compõem a forma de onda a uma plataforma específica de *hardware*.

Por serem baseados no padrão SCA, os componentes desenvolvidos possuem características próprias do SCA de portabilidade e reuso de *software*, que são requisitos básicos desta especificação. Assim, um mesmo componente pode ser usado em diferentes formas de onda, tornando o seu desenvolvimento mais rápido e de fácil adaptação de código, como consequência de ser um projeto de código aberto. Os diversos componentes desenvolvidos e utilizados também possuem parâmetros para configurações, permitindo assim ao usuário adequar a aplicação à sua necessidade.

Além do OCD e OWD, o OSSIE possui uma ferramenta denominada ALF que permite a visualização da forma de onda e depuração. O ALF permite que o desenvolvedor ou o usuário possa instalar, executar, parar e desinstalar as aplicações de forma de onda, além da exibição por meio de um diagrama de blocos com a interconexão dos componentes. Além disto, são disponibilizados diversos *plugins* para exibição de funcionalidades adicionais, como a obtenção do espectro em frequência do sinal na saída de cada componente e os diagramas das constelações através dos gráficos IQ.

Outra ferramenta presente no OSSIE é o *WaveDash* (*Waveform Dashboard*) que fornece uma interface gráfica interativa personalizável oferecendo um controle fácil de formas de ondas implementadas. A interface gráfica disponível não só permite

que um usuário instale e execute as formas de onda, mas também a configuração das propriedades dos componentes, a escolha de propriedades para exibição e o tipo de controle a ser usado para cada propriedade exibida. Com isso, atribui-se foco ao desenvolvimento da forma de onda, componentes e propriedades, o que acelera o desenvolvimento, reduzindo erros de programação.

IV. USRP - UNIVERSAL SOFTWARE RADIO PERIPHERAL

O equipamento *Universal Software Radio Peripheral*, da empresa Ettus Research LLC, é uma placa de baixo custo projetada para pesquisa e desenvolvimento de protótipos de aplicações em rádios definidos por *software*, permitindo que desenvolvedores possam criar um rádio em *software* utilizando um computador pessoal convencional por meio de uma porta USB. O USRP tem um projeto aberto e flexível, com a divulgação gratuita não só dos circuitos eletrônicos utilizados, como de *softwares* de integração com ambientes de desenvolvimento rádio, como o GNURadio [12], Simulink [13] e OSSIE [7].

Os principais componentes da arquitetura do USRP são a antena, a placa-mãe e as placas-filhas. A antena de transmissão é responsável pela transmissão efetiva da informação através do canal rádio. De acordo com a necessidade do usuário, podem ser conectadas antenas em diferentes faixas de frequências (0 a 3 GHz) e estratégias de funcionamento, permitindo a utilização de múltiplas antenas para a transmissão e recepção de sinais. As placas-filhas implementam o estágio de rádio-frequência do equipamento de comunicações, a partir da execução das conversões de sinal entre bandas, indicação da potência de transmissão (de 1 a 200 mW), e determinação da frequência de operação dos canais de transmissão e recepção do rádio.

Já a placa-mãe dispõe de conversores ADC e DAC, processamento digital de sinais via FPGA, interfaces de comunicação com o computador, bem como interfaces de expansão para integração com outras placas. Como modelos de equipamentos, existem as versões embarcadas (E100 e E110), em barramento (USRP1 e B100) e em rede (N200 e N210), as quais variam em capacidade de amostragem dos conversores ADC e DAC, grau de flexibilidade para operação com múltiplas antenas, disponibilidade de canais de RF, largura de banda para processamento em banda básica, modelo de FPGA e, principalmente, quanto ao tipo de operação, o qual pode ocorrer de forma independente (com uma UCP embarcada) ou acoplada a um computador para execução das aplicações.

V. ESTUDO DE CASO - IMPLEMENTAÇÃO DE FORMA DE ONDA NO OSSIE

Diversas aplicações podem ser desenvolvidas com o OSSIE, em face de sua união com a USRP, a qual permite a criação de diferentes sistemas aplicáveis nas mais diversas áreas. A implementação de um sistema de comunicação no OSSIE é feita através da conexão de diversos *componentes* gerando a aplicação ou *forma de onda*. Como exemplo, apresenta-se nesta seção a implementação de um modulador em frequência (FM). Este componente possui apenas uma entrada e uma saída de dados.

Além disso, são inseridas duas variáveis para sua configuração: máximo desvio de frequência e taxa de amostragem. Com estas características, deseja-se criar um componente no OSSIE que represente este modulador.

No início da criação de um novo componente no OSSIE, emprega-se o *OSSIE Component Development (OCD)*. Já as adições de portas e propriedades aos componentes estão ilustradas a seguir nas Figuras 1(a) e 1(b), respectivamente.

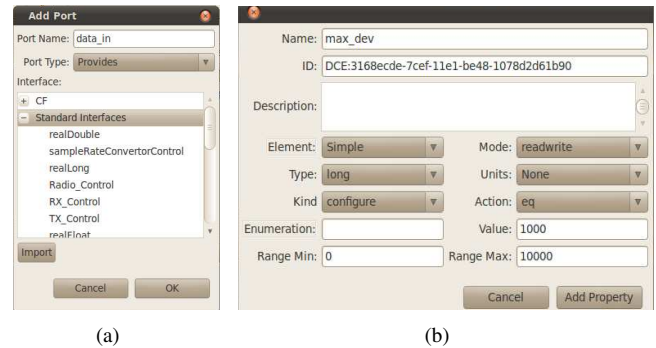


Fig. 1. (a) Adicionando uma porta a um componente. (b) Adicionando uma propriedade a um componente.

Depois de se definir todas as portas e propriedades de um componente, o menu *OSSIE* → *Generate Component* permite a geração do código básico do componente a ser implementado, a partir da alteração do método *ProcessData()* na classe gerada. Isso posto, os modelos de projeto adotados para o transmissor e o receptor FM estão apresentados nas Figuras 2 e 3, respectivamente. Como pode ser visto na Figura 2, o sinal de entrada é primeiramente amplificado e filtrado, para então sofrer a modulação em frequência. Em seguida, é feita uma interpolação, aumentando a taxa de amostragem do sinal, juntamente com uma nova filtragem. A última etapa realizada no GPP é um controle do nível de saída do sinal. Por fim, o sinal é transmitido, via USB, para a placa USRP que é responsável pela conversão digital-analógica e o deslocamento para a frequência de transmissão.

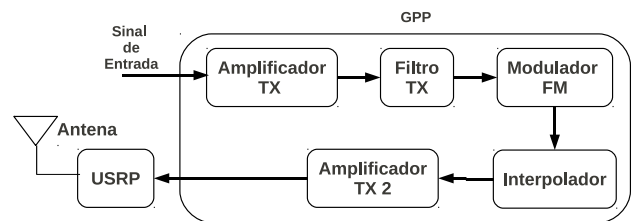


Fig. 2. Esquemático do transmissor FM implementado.

Já no receptor, o sinal é recebido pela antena conectada à USRP. Esta é responsável pelo deslocamento para a frequência compatível com o conversor A/D assim como a própria conversão analógica-digital. O sinal digitalizado é enviado, via USB, para o GPP que primeiramente amplifica-o. Em seguida é usado um decimador, diminuindo a taxa de amostragem do sinal, juntamente com um filtro. Depois o sinal passa por

um AGC (controle automático de ganho) para, então, sofrer a demodulação em frequência. Por fim, é realizada uma última filtragem do sinal.

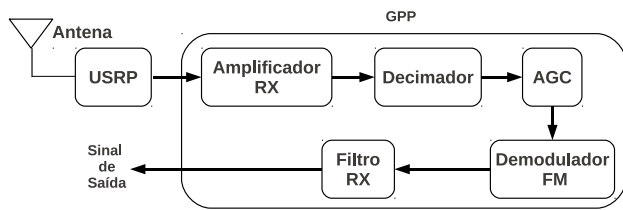


Fig. 3. Esquemático do receptor FM implementado.

Para a construção de uma forma de onda no OSSIE, deve-se seguir o menu *File* → *New* → *OSSIE Waveform*. Em seguida, escolhe-se o nome do projeto (neste exemplo chamaremos de TransceptorFM) e os arquivos do projeto são gerados com o botão *Finish*, com apresentação por meio do *OSSIE Waveform Development (OWD)*, no qual pode-se identificar todos os componentes e dispositivos utilizados. A comunicação entre o computador e a USRP, realizada via USB, é controlada pelo componente *Controlador_USRP*. A Figura 4 mostra a janela de configuração da USRP.

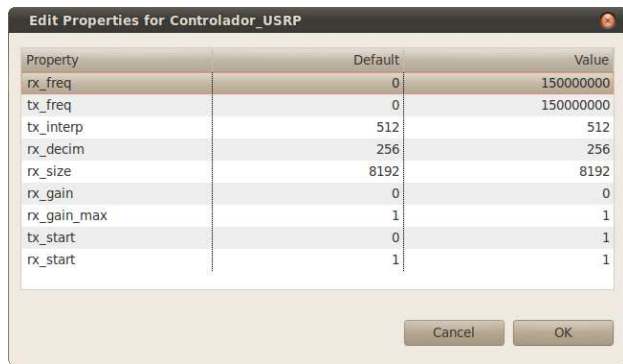


Fig. 4. Janela de configuração da USRP.

Por fim, como aplicação, realizou-se a transmissão com sucesso de um tom de 2,5kHz através do transceptor FM implementado para comunicação com um equipamento rádio de propriedade das Forças Armadas brasileiras. Foi utilizada uma taxa de amostragem de 25kHz e um fator 10 para o interpolador e o decimador. O máximo desvio de frequência adotado no modulador FM foi de 2kHz, com o emprego de filtros FIR com corte em 12kHz.

VI. CONCLUSÃO

O emprego de rádios definidos por *software* constituiu-se na tendência de momento no desenvolvimento de equipamentos rádio para aplicações militares, trazendo portabilidade de formas de onda e interoperabilidade entre diferentes tropas. Dentre as diversas soluções existentes, este trabalho mostrou a criação de um componente e de uma forma de onda em uma ferramenta compatível com a arquitetura SCA (OSSIE) em um ambiente

aberto de projetos de *hardware*, o USRP, a partir da implementação de um transceptor FM usando RDS. Tal ferramenta se mostra útil não só para o desenvolvimento de aplicações acadêmicas, como para a prototipação de produtos comerciais, dada sua simplicidade de uso e inspiração na arquitetura SCA, a qual tem se mostrado como o padrão de fato em aplicações militares.

Como trabalhos futuros, os autores pretendem desenvolver duas aplicações de forma de onda (transmissão analógica e digital) na faixa de VHF, como prova de conceito de um RDS de Defesa. Além disso, os passos seguintes contemplam a implementação de técnicas de transmissão adaptativa, bem como o estudo e avaliação de procedimentos de ataques e vulnerabilidades [14] no protótipo a ser desenvolvido.

AGRADECIMENTOS

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e à Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) pelo apoio financeiro prestado ao desenvolvimento deste trabalho.

REFERÊNCIAS

- [1] JTRS program. SCA - Software Communications Architecture Specification, v.4.0. <http://sca.jpeojtrs.mil/>, 2012.
- [2] C.R.A. Gonzalez, C.B. Dietrich, and J. H. Reed. Understanding the Software Communications Architecture. In *IEEE Communications Magazine*, volume 47, pages 50–57, sept 2009.
- [3] United States Department of Defense. Interoperability and Performance Standards for Data Modems - Interface Standard MIL-STD-188-110B, 2000.
- [4] North Atlantic Treaty Organization. Profile for High Frequency Radio Data Communication Edition 2 - STANAG 5066.
- [5] J. G. Proakis. Digital communications. In *Mc-Graw Hill International Editions*, volume 4th Edition, 2001.
- [6] Object Management Group. OMG web page. <http://www.omg.org/>, 2012.
- [7] Wireless@Virginia Tech. OSSIE project web page. <http://ossie.wireless.vt.edu/>, 2012.
- [8] VirginiaTech. Cognitive Radios & Networks. http://wireless.vt.edu/research/Cognitive_Radios_Networks/.
- [9] Matt Ettus. Universal software radio peripheral. <http://www.ettus.com>.
- [10] J. Snyder, S. H. Edwards, and C. B. Dietrich. An eclipse-based IDE for OSSIE waveforms and components. In *SDR Forum Tech. Conference, Washington, DC*, oct 2008.
- [11] M. Carrick, D. Cormier, C. Covington, C. Dietrich, J. Gaeddert, B. Hilburn, C. Phelps, S. Sayed, D. Seeralan, J. Snyder, and H. Volos. OSSIE 0.8.2 installation and user guide. In *Virginia Tech*, April 2011.
- [12] E. Blossom. Exploring GNU radio. <http://www.gnu.org/software/gnuradio/doc/exploring-gnuradio.html>, 2012.
- [13] The Mathworks Inc. Simulink. <http://www.mathworks.com>.
- [14] F.A.B. Silva, D.F.C. Moura, and J.F. Galdino. Classes of attacks for tactical software defined radios. In *International Journal of Embedded and Real-Time Communication Systems*, in press, jul 2012.