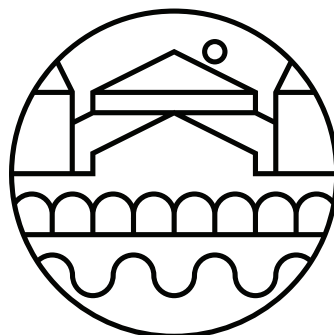




XXXV
SIMPÓSIO BRASILEIRO DE
REDES DE COMPUTADORES
E SISTEMAS DISTRIBUÍDOS
15 a 19 de maio de 2017
Belém - Pará

Anais V WoCCES 2017



X X X V

SIMPÓSIO BRASILEIRO DE
REDES DE COMPUTADORES
E SISTEMAS DISTRIBUÍDOS

15 a 19 de maio de 2017
Belém - Pará

Anais do V WoCCES 2017

Workshop of Communication in Critical Embedded Systems

Editora

Sociedade Brasileira de Computação (SBC)

Organização

Adriano Mauro Cansian – IBILCE – UNESP Rio Preto

Alex Sandro Roschildt Pinto – UFSC - Blumenau

Daniel Fernando Pigatto - UTFPR Curitiba

Kalinka Regina Lucas Jaquie Castelo Branco – ICMC – USP

Vandermi João da Silva - UFAM

Ronaldo Alves Ferreira (UFMS)

Realização

Sociedade Brasileira de Computação (SBC)

Universidade Federal do Pará (UFPA)

Laboratório Nacional de Redes de Computadores (LARC)

Copyright ©2017 da Sociedade Brasileira de Computação
Todos os direitos reservados

Capa: Catarina Nefertari (PCT-UFPA)

Produção Editorial: Denis Lima do Rosário (UFPA)

Cópias Adicionais:

Sociedade Brasileira de Computação (SBC)

Av. Bento Gonçalves, 9500- Setor 4 - Prédio 43.412 - Sala 219

Bairro Agronomia - CEP 91.509-900 - Porto Alegre - RS

Fone: (51) 3308-6835

E-mail: sbc@sbc.org.br

V Workshop of Communication in Critical Embedded Systems (5: 2017: Belém, Pa).

Anais / V Workshop of Communication in Critical Embedded Systems – WoCCES; organizado por Antônio Jorge Gomes Abelém, Eduardo Coelho Cerqueira, Ronaldo Alves Ferreira, Adriano Mauro Cansian, Alex Sandro Roschildt Pinto, Daniel Fernando Pigatto, Kalinka Regina Lucas Jaquie Castelo Branco, Vandermi João da Silva - Porto Alegre: SBC, 2017

84 p. il. 21 cm.

Vários autores

Inclui bibliografias

1. Redes de Computadores. 2. Sistemas Distribuídos. I. Abelém, Antônio Jorge Gomes II. Cerqueira, Eduardo Coelho III. Ferreira, Ronaldo Alves IV. Cansian, Adriano Mauro V. Pinto, Alex Sandro Roschildt VI. Pigatto, Daniel Fernando VII. Branco, Kalinka Regina Lucas Jaquie Castelo VIII. Silva, Vandermi João IX. Título.

Sociedade Brasileira da Computação

Presidência

Lisandro Zambenedetti Granville (UFRGS), Presidente

Thais Vasconcelos Batista (UFRN), Vice-Presidente

Diretorias

Renata de Matos Galante (UFGRS), Diretora Administrativa

Carlos André Guimarães Ferraz (UFPE), Diretor de Finanças

Antônio Jorge Gomes Abelém (UFPA), Diretor de Eventos e Comissões Especiais

Avelino Francisco Zorzo (PUC-RS), Diretor de Educação

José Viterbo Filho (UFF), Diretor de Publicações

Claudia Lage Rebello da Motta (UFRJ), Diretora de Planejamento e Programas Especiais

Marcelo Duduchi Feitosa (CEETEPS), Diretor de Secretarias Regionais

Eliana Almeida (UFAL), Diretora de Divulgação e Marketing

Roberto da Silva Bigonha (UFMG), Diretor de Relações Profissionais

Ricardo de Oliveira Anido (UNICAMP), Diretor de Competições Científicas

Raimundo José de Araújo Macêdo (UFBA), Diretor de Cooperação com Sociedades Científicas

Sérgio Castelo Branco Soares (UFPE), Diretor de Articulação com Empresas

Contato

Av. Bento Gonçalves, 9500

Setor 4 - Prédio 43.412 - Sala 219

Bairro Agronomia

91.509-900 – Porto Alegre RS

CNPJ: 29.532.264/0001-78

<http://www.sbrc.org.br>

Laboratório Nacional de Redes de Computadores (LARC)

Diretora do Conselho Técnico-Científico

Rossana Maria de C. Andrade (UFC)

Vice-Diretor do Conselho Técnico-Científico

Ronaldo Alves Ferreira (UFMS)

Diretor Executivo

Paulo André da Silva Gonçalves (UFPE)

Vice-Diretor Executivo

Elias P. Duarte Jr. (UFPR)

Membros Institucionais

SESU/MEC, INPE/MCT, UFRGS, UFMG, UFPE, UFCG (ex-UEPB Campus Campina Grande), UFRJ, USP, PUC-Rio, UNICAMP, LNCC, IME, UFSC, UTFPR, UFC, UFF, UFSCar, IFCE (CEFET-CE), UFRN, UFES, UFBA, UNIFACS, UECE, UFPR, UFPA, UFAM, UFABC, PUCPR, UFMS, UnB, PUC-RS, PUCMG, UNIRIO, UFS e UFU.

Contato

Universidade Federal de Pernambuco - UFPE

Centro de Informática - CIn

Av. Jornalista Anibal Fernandes, s/n

Cidade Universitária

50.740-560 - Recife - PE

<http://www.larc.org.br>

Organização do SBRC 2017

Coordenadores Gerais

Antônio Jorge Gomes Abelém (UFPA)
Eduardo Coelho Cerqueira (UFPA)

Coordenadores do Comitê de Programa

Edmundo Roberto Mauro Madeira (UNICAMP)
Michele Nogueira Lima (UFPR)

Coordenador de Palestras e Tutoriais

Edmundo Souza e Silva (UFRJ)

Coordenador de Painéis e Debates

Luciano Paschoal Gaspar (UFRGS)

Coordenadores de Minicursos

Heitor Soares Ramos (UFAL)
Stênio Flávio de Lacerda Fernandes (UFPE)

Coordenadora de Workshops

Ronaldo Alves Ferreira (UFMS)

Coordenador do Salão de Ferramentas

Fabio Luciano Verdi (UFSCar)

Comitê de Organização Local

Adailton Lima (UFPA)
Alessandra Natasha (CESUPA)
Davis Oliveria (SERPRO)
Denis Rosário (UFPA)
Elisangela Aguiar (SERPRO)
João Santana (UFRA)
Josivaldo Araújo (UFPA)
Marcos Seruffo (UFPA)
Paulo Henrique Bezerra (IFPA)
Rômulo Pinheiro (UNAMA)
Ronado Ferreira (META)
Thiêgo Nunes (IFPA)
Vagner Nascimento (UNAMA)

Comite Consultivo

Allan Edgard Silva Freitas (IFBA)
Antonio Alfredo Ferreira Loureiro (UFMG)
Christian Esteve Rothenberg (UNICAMP)
Fabiola Gonçalves Pereira Greve (UFBA)
Frank Augusto Siqueira (UFSC)
Jussara Marques de Almeida (UFMG)
Magnos Martinello (UFES)
Antonio Marinho Pilla Barcellos (UFRGS)
Moisés Renato Nunes Ribeiro (UFES)
Rossana Maria de Castro Andrade (UFC)

Mensagem dos Coordenadores Gerais

Sejam bem-vindos ao 35o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2017) e a acolhedora cidade das mangueiras - Belém / Pará.

Organizar uma edição do SBRC pela segunda vez no Norte do Brasil é um desafio e um privilégio por poder contribuir com a comunidade de Redes de Computadores e Sistemas Distribuídos do Brasil e do exterior. O SBRC se destaca como um importante celeiro para a discussão, troca de conhecimento e apresentação de trabalhos científicos de qualidade.

A programação do SBRC 2017 está diversificada e discute temas relevantes no cenário nacional e internacional. A contribuição da comunidade científica brasileira foi de fundamental importância para manter a qualidade técnica dos trabalhos e fortalecer a ciência, tecnologia e inovação no Brasil.

Após um cuidadoso processo de avaliação, foram selecionados 77 artigos completos organizados em 26 sessões técnicas e 10 ferramentas para apresentação durante o Salão de Ferramentas. Além disso, o evento contou com 3 palestras e 3 tutoriais proferidos por pesquisadores internacionalmente renomados, 3 painéis de discussões e debates, todos sobre temas super atuais, 6 minicursos envolvendo Big Data, sistemas de transportes inteligentes, rádios definidos por software, fiscalização e neutralidade da rede, mecanismos de autenticação e autorização para nuvens computacionais e comunicação por luz visível, bem como 10 workshops.

O prêmio “Destaque da SBRC” e uma série de homenagens foram prestadas para personalidades que contribuíram e contribuem com a área. O apoio incondicional da SBC, do LARC, do Comitê Consultivo da SBRC e da Comissão Especial de Redes de Computadores e Sistemas Distribuídos da SBC foram determinantes para o sucesso do evento. A realização do evento também contou com o importante apoio do Comitê Gestor da Internet no Brasil (CGI.br), do CNPq, da CAPES, do Parque de Ciência e Tecnologia Guamá, da Connecta Networking, da Dantec Telecom, da RNP e do Google. Nosso especial agradecimento à Universidade Federal do Pará (UFPA) e ao Instituto Federal do Pará (IFPA) pelo indispensável suporte à realização do evento.

Nosso agradecimento também para os competentes e incansáveis coordenadores do comitê do programa (Michele Nogueira/UFPA – Edmundo Madeira/UNICAMP), aos coordenadores dos minicursos (Stênio Fernandes/UFPE – Heitor Ramos/UFAL), ao coordenador dos workshops (Ronaldo Ferreira/UFMS), ao coordenador de painéis e debates (Luciano Gaspar/UFRGS), ao coordenador do Salão de Ferramentas (Fabio Verdi/UFSCar) e ao coordenador de palestras e tutoriais (Edmundo Souza e Silva/UFRJ). Destacamos o excelente trabalho do comitê de organização local coordenado por Denis do Rosário.

Por fim, desejamos a todos uma produtiva semana em Belém.

Antônio Abelém e Eduardo Cerqueira

Coordenadores Gerais do SBRC 2017

Mensagem do Coordenador de Workshops

É com grande prazer que os convido a prestigiar os workshops do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) nos dias 15, 16 e 19 de maio de 2017. Tradicionalmente, os workshops abrem e fecham a semana do SBRC e são responsáveis por atrair uma parcela expressiva de participantes para o Simpósio. Como coordenador de workshops, dividi com os coordenadores gerais do SBRC a nobre tarefa de selecionar os workshops que melhor representam a comunidade e que fortaleçam novas linhas de pesquisa ou mantenham em evidência linhas de pesquisa tradicionais.

Em resposta à chamada aberta de workshops, recebemos dez propostas de alta qualidade, das quais nove foram selecionadas. Além disso, mantivemos a longa colaboração com a RNP por meio da organização do WRNP, que já é uma tradição na segunda e terça-feira da semana do SBRC. Dentre as propostas aceitas, sete são reedições de workshops tradicionais do SBRC que já são considerados parte do circuito nacional de divulgação científica nas várias subáreas de Redes de Computadores e Sistemas Distribuídos, como o WGRS (Workshop de Gerência e Operação de Redes e Serviços), o WTF (Workshop de Testes e Tolerância a Falhas), o WCGA (Workshop em Clouds, Grids e Aplicações), o WP2P+ (Workshop de Redes P2P, Dinâmicas, Sociais e Orientadas a Conteúdo), o WPEIF (Workshop de Pesquisa Experimental da Internet do Futuro), o WoSiDA (Workshop de Sistemas Distribuídos Autônomicos) e o WoCCES (Workshop de Comunicação de Sistemas Embarcados Críticos). Como novidade, teremos dois novos workshops com programação diversificada e grande apelo social, o CoUrb (Workshop de Computação Urbana) e o WTICp/D (Workshop de TIC para Desenvolvimento).

Temos certeza que 2017 será mais um ano de sucesso para os workshops do SBRC pelo importante papel de agregação que eles exercem na comunidade científica de Redes de Computadores e Sistemas Distribuídos no Brasil.

Aproveitamos para agradecer o apoio recebido de diversos membros da comunidade e, em particular, a cada coordenador de workshop, pelo brilhante trabalho. Como coordenador dos workshops, agradeço imensamente o apoio recebido da Organização Geral do SBRC 2017.

Esperamos que vocês aproveitem não somente os workshops, mas também todo o SBRC e as inúmeras atrações de Belém.

Ronaldo Alves Ferreira

Coordenador de Workshops do SBRC 2017

Mensagem dos Coordenadores do V WoCCES 2017

O V Workshop of Communication in Critical Embedded Systems (WoCCES) em conjunto com o 35o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2017), em Belém, Pará, tem por objetivo atuar como um fórum para apresentações técnicas de pesquisas em andamento e atividades relevantes nas áreas de comunicação em sistemas embarcados críticos, congregando pesquisadores e profissionais que atuam ativamente nessas áreas. O workshop também procura estabelecer redes colaborativas multi-institucionais e grupos de competência técnico-científica, bem como fortalecer atividades em andamento.

Sendo assim, o Workshop se concentra em importantes inovações e avanços recentes na especificação, projeto, construção e utilização da comunicação em sistemas embarcados críticos, sendo que o objetivo do mesmo é o de reunir pesquisadores e profissionais da indústria e da academia e proporcionar-lhes uma oportunidade para se informar sobre os últimos desenvolvimentos, implantações, tendências tecnológicas e resultados de pesquisa, bem como iniciativas relacionadas com sistemas embarcados e suas aplicações em uma variedade de ambientes industriais.

A programação do V WoCCES inclui 8 artigos completos selecionados pelo Comitê de Programa e uma palestra convidada.

Gostaríamos de agradecer a todos os membros do Comitê de Programa pelo seu compromisso em retornar avaliações de qualidade em um prazo justo. Sem a pronta atuação de cada membro o WoCCES não teria acontecido. Finalmente, gostaríamos de agradecer aos autores pelas contribuições dadas ao Workshop.

Esperamos que a palestra e as apresentações dos trabalhos nas seções técnicas promovam discussões acaloradas entre os participantes do workshop, contribuindo assim para o avanço da área.

Adriano Mauro Cansian, Alex Sandro Roschildt Pinto, Daniel Fernando Pigatto,
Kalinka Regina Lucas Jaquie Castelo Branco, Vandermi da Silva

Coordenadores do V WoCCES 2017

Comitê de Programa

- Adão Boava (UFSC)
- Adriano Mauro Cansian (UNESP)
- Alex Sandro Roschildt Pinto (UFSC)
- Carlos Barros Montez (UFSC)
- Célia Leiko Ogawa Kawabata (IFSP)
- Daniel Fernando Pigatto (UTFPR)
- Edson dos Santos Moreira (USP)
- Ellen Francine Barbosa (USP)
- Fábio Dacêncio Pereira (UNIVEM)
- Fernando Santos Osório (USP)
- Gustavo Pessin (Vale)
- Horácio Antonio Fernandes de Oliveira (UFAM)
- Ivanovitch Medeiros Dantas da Silva (IMD)
- Jacir Luiz Bordim (UnB)
- João Cunha - Instituto Politécnico de Coimbra
- Kalinka Regina Lucas Jaquie Castelo Branco (USP)
- Luciana Martimiano (UEM)
- Luiz Henrique Castelo Branco (IFSP)
- Marco Vieira - Universidade de Coimbra
- Marcos Fagundes Caetano (UnB)
- Mariana Rodrigues (USP)
- Mario Antonio Ribeiro Dantas (UFSC)
- Mário Meireles Teixeira (UFMA)
- Natassya Barlate Floro da Silva (USP)
- Raimundo Barreto (UFAM)
- Rayner de Melo Pires (USP)
- Vandermi João da Silva (UFAM)

Sumário

Sessão Técnica 1	1
Metodologia para Localização de Ativos por Algoritmos de Fusão de Sensores para Rastreamento Autônomo	2
Igor Devulsky Prata (IME), Andressa da Silva Siqueira (IME), Paulo Fernando Ferreira Rosa (IME) e Anderson Fernandes Pereira dos Santos (IME)	
Reducing energy consumption on internal communications of unmanned vehicles with HAMSTER's Navigation Phases	12
Eduardo Aramizu (USP), Daniel Fernando Pigatto (UFPR) e Kalinka Regina Lucas Jaquie Castelo Branco (USP)	
I-Jack: Sistema Vestível Inteligente para Coleta e Avaliação de Dados Fisiológicos	25
Elpidio Holanda Júnior (UFAM), Lucas Farias da Cruz (UFAM) e Gustavo Luiz Pinhei (UFAM)	
Implementação de um Computador KeyFlow 10 Gbps para Redes Definidas por Software	34
Matheus de Souza (INATEL), Natan de Oliveira (INATEL), Antônio M. Alberti (INATEL) e Moisés R. N. Ribeiro (UFES)	
Sessão Técnica 2	44
Estimativa de Posição 3D de Obstáculos para VANTs	45
Rodolfo Barros Chiamonte (USP) e Kalinka Regina Lucas Jaquie Castelo Branco (USP)	
Concepção de Uma Plataforma de VANT de Baixo Custo do Tipo Quadricóptero para Uso em Pesquisas	56
Ricardo M. Bernardo (UFES), Luis C. B. da Silva (CEFET-RJ) e Paulo F. F. Rosa (IME)	
Sessão Técnica 3	66
Physical Web: Avaliação de Técnicas de Estimativa de Localização de Dispositivos BLE em Ambientes Internos	67
Guilherme David Branco (UnB), Samuel Vinicius Vieira Pala (UnB) e Jacir Luiz Bordim (UnB)	
Uso da Rede MQTT para a Manipulação de Sistemas Robóticos suportados por Realidade Virtual	77
Leandro R. Mattioli (UFU), Daniel S. Souza (UFU), Márcio J. Cunha (UFU) e Alexandre Cardoso (UFU)	

**V Workshop de Comunicação em Sistemas
Embarcados Críticos (WoCCES)
SBRC 2017
Sessão Técnica 1**

Metodologia para Localização de Ativos por Algoritmos de Fusão de Sensores para Rastreamento Autônomo

Igor Devulsky Prata, Andressa da Silva Siqueira,
Paulo Fernando Ferreira Rosa, Anderson Fernandes Pereira dos Santos

¹Seção de Engenharia de Computação – Instituto Militar de Engenharia (IME)
Praça Gen. Tibúrcio, 80 – Praia Vermelha, 22290-270 – Rio de Janeiro – RJ – Brasil

{igorprata, rpaulo, anderson}@ime.eb.br, siq.andressa@gmail.com

Abstract. *This article proposes a technique using different radio standards to improve positioning estimation of a RF source, through application of sensor fusion algorithms. The positioning estimation of a target by only one type of sensor presents considerable inaccuracies, that can be accentuated depending on the environment. Several studies suggest that localization challenges should not be solved using a single sensory approach, but rather using multiple indicators. Our proposed solution consists of practical application of wardriving techniques using well-known network protocols, such as WIFI (IEEE 802.11) and bluetooth, embedded on autonomous navigation robotic systems, with potential to support other standards if necessary.*

Resumo. *Este artigo propõe uma técnica que utiliza rádios em diferentes padrões com a finalidade de melhorar a estimativa de posicionamento de uma fonte emissora, através da aplicação de algoritmos de fusão de sensores. A estimativa de posicionamento do alvo por um único tipo de sensor apresenta consideráveis imprecisões que podem ser acentuadas dependendo do meio. Diversos trabalhos sugerem que a localização não deve ser tratada por uma abordagem sensorial única, que compreenda todos os desafios da área, mas sim pelo uso de múltiplos indicadores. A solução consiste em aplicações de técnicas de wardriving utilizando protocolos de rede bem difundidos, WIFI (IEEE 802.11) e bluetooth, mas com o potencial de suportar outros padrões, com a finalidade de embarcar em sistemas autônomos de navegação.*

1. Introdução

Aplicações sensíveis ao contexto, [Sapiezynski et al. 2015], são tendências impulsionadas pelas redes sociais, pela queda dos custos de produção de eletrônicos e pela facilidade de acesso à computadores móveis. Nessas aplicações, soluções para localização incluem estimativas geradas por um ou mais sensores. Suas coordenadas podem ser obtidas de fontes conhecidas como satélites, torres de comunicação de telefonia celular ou pontos de acesso a redes sem fio (*access points*). Produtos eletrônicos de consumo, como *smartphones*, possuem diversos recursos embarcados específicos para localização. Para estimativa de posicionamento são adotados receptores de GPS (*global positioning system*), sensores inerciais (IMUs), bússolas digitais e barômetros, além de tecnologias não dedicadas, como rádios WIFI (no padrão IEEE 802.11). Tal aparato tecnológico se encontra de maneira ubíqua em hardwares pessoais de uso diário.

Este trabalho apresenta uma alternativa de baixo custo para localização automatizada de alvos transmissores de sinal de radiofrequência compatíveis com os padrões WIFI e *bluetooth* de maneira precisa e dinâmica, empregando técnicas de *wardriving* e fusão de sensores de modo a oferecer maior redundância e taxa de acerto em suas estimativas.

Originalmente definido como a técnica de busca por redes WIFI através de veículos automotivos equipados com sensores [Thomas King 2006], atualmente o termo *wardriving* é utilizado para localização de pontos de acesso a redes sem fio por qualquer meio de transporte. [Tsui et al. 2010] acrescenta que o deslocamento do agente influencia na qualidade do mapeamento de pontos de acesso, em relação a velocidade e ao percurso.

O resgate de vítimas em áreas de desastre é uma atividade de risco às equipes de busca. Além das dificuldades no resgate, a localização de uma vítima é dificultada por causa da imprecisão de sua localização [Hawkinson et al. 2012] e [Amendolare et al. 2008]. Veículos aéreos não tripulados (VANT) já são empregados por equipes de emergência no Brasil, mas estes aparelhos ainda são utilizados como dispositivos de observação remota e sem rotinas automatizadas de busca e localização [Gomes 2016]. *Smartphones* tem a funcionalidade consideravelmente limitada em regiões sem infraestrutura de telefonia estabelecida, impossibilitando à vítima realizar ligações de emergência. Porém, esses dispositivos possuem transmissores de radiofrequência capazes de emitir sinais por algumas centenas de metros e passíveis de serem usados como localizadores (*beacons*) de emergência.

Um VANT, operando em busca e localização (como exemplificado na figura 1), deve ser autônomo mesmo fora da área de comunicação com a estação de controle, mantendo a capacidade em detectar sinais de radiofrequência dentro do seu raio, e estimar a origem de transmissão. O sistema desenvolvido será executável em hardwares embarcados com restrições computacionais (como processamento, quantidade de memória e energia), e não deve depender de bases de dados e outras fontes de informação externas.

Em ambientes não estruturados, a estimativa precisa da localização de um emissor de radiofrequência necessita da captura de várias amostras de sinais [Zhang et al. 2011]. Poucas verificações de sinais podem levar a um cálculo incorreto ao determinar um ponto. A utilização de múltiplos sensores em paralelo melhora a precisão e oferece uma taxa de redundância útil à estimativa [Lin et al. 2010]. Entretanto, há que se otimizar o consumo de energia do aparelho emissor sem que haja sacrifício à taxa de sinais necessária para o algoritmo garantir precisão na localização. Para tal, é necessário determinar o limiar ideal entre frequência e intensidade dos sinais e consumo de energia. Esse consumo está associado ao padrão de rádio utilizado: hardwares WIFI necessitam de mais energia para operação do que a maioria das classes *bluetooth*, que, em contrapartida, possuem alcance de transmissão menor.

Outros desafios associados à análise de registros obtidos por coletas de sinais envolvem flutuações das amostras recebidas entre diferentes *scans* e os fenômenos associados à propagação de ondas de radiofrequência como dispersão, atenuação, reflexão, refração, difração, distorção e ruído.

O conteúdo desse artigo está estruturado na seguinte sequência: a segunda seção apresenta a descrição do problema de estimativa de localização. Na seção 3, são apresentados alguns trabalhos na área. Na seção 4, é apresentada a estrutura geral da solução

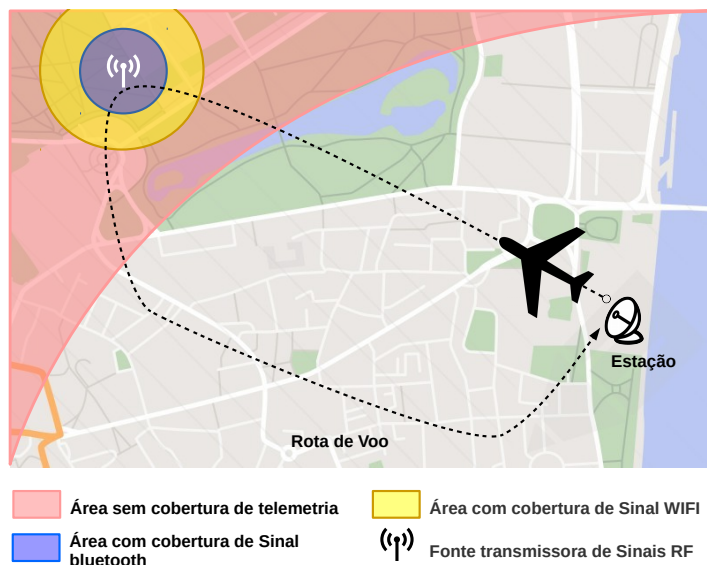


Figura 1. Uma rota de voo autônoma com região sem sinal e comunicação entre o VANT e a estação de controle.

proposta e as contribuições esperadas. O *framework*, hardwares e a plataforma de voo escolhidos são apresentados na seção 5. Na seção 6, são apresentados os resultados preliminares, seguida das conclusões e próximas etapas para alcançar as metas.

2. Descrição do problema

Dado um transmissor de radiofrequência em múltiplos protocolos, determinar a sua localização de origem baseado nas amostras coletadas por diferentes sensores (WIFI (W) e *bluetooth* (B)) em diferentes instantes, por um mesmo agente.

Os sensores adotados são capazes de determinar a distância relativa da fonte emissora, em função da intensidade de sinal (I) capturado e o posicionamento no instante da coleta de cada amostra (α). Registrando uma série temporal, cada amostra (α_i) é composta por uma tupla de valores, que conta com a localização da aeronave (p_i) no instante (t_i) e as informações de Intensidade de sinal (I_i^W) e (I_i^B), medidas de cada rádio bem como informações do respectivo erro associado (e_i^W) e (e_i^B), conforme equação 1.

$$\langle p_i, t_i, I_i^W, e_i^W, I_i^B, e_i^B \rangle_{\alpha_i} \quad (1)$$

Devido à características físicas de emissão de sinais, conclui-se que os erros de estimação desses são diretamente proporcionais à distância entre o ponto (P) e a origem de transmissão de radiofrequência. Para otimizar a solução de localização, espera-se a minimização dos erros (e) referentes aos dados de cada sensor, ou seja:

$$\min e_i^j, \text{ onde } j \in \{WIFI, bluetooth\} \quad (2)$$

Para tratar diferentes fontes de dados armazenadas em uma série temporal, o algoritmo de fusão de sensores adotado será o Filtro de Kalman Estendido (EKF).

3. Revisão de Literatura e Estado da Arte

Diversos algoritmos como FILA [Wu et al. 2012], SpotFi [Kotaru et al. 2015], o filtro Alpha [Prieto et al. 2012] e o RMCL [Zhu et al. 2013], vêm sendo propostos como técnicas de localização com o objetivo de mapear as redes WIFI. Poucos trabalhos referenciam aplicações práticas para localização de ativos em ambientes externos e fora de zonas urbanas, utilizando estimativas baseadas em intensidade de sinal radiofrequência. [Weber et al. 2010] explicam como mapeamentos baseados em *wardriving* são mais eficientes quando aplicado em cidades, devido à quantidade de emissores no padrão WIFI.

A localização precisa de ativos através de soluções de baixo custo computacional é o tema de trabalhos em desenvolvimento. Algumas propostas como o GLANSER [Hawkinson et al. 2012] buscam estimar a posição de bombeiros em ambientes *indoor*, dentro de edificações em chamas, onde o sinal de sistemas de localização por satélite são ineficientes. Técnicas acessíveis para mapeamento de pontos de acesso WIFI, como o *wardriving*, são aplicadas de diferentes maneiras promovendo estimativa de localização mais veloz e precisa em ambientes *indoor* e *outdoor*, como uma alternativa viável aos sistemas de localização por satélite [Haeberlen et al. 2004].

Para um VANT em rota de navegação autônoma, a detecção de ondas de radiofrequência e a determinação da origem do seu emissor podem ser implementadas de maneira simplificada. Segundo [Yu et al. 2016], essa abordagem é uma alternativa a soluções que empregam câmeras, que dependem de técnicas de visão computacional que podem gerar custo de processamento proibitivo em hardwares embarcados. Essas soluções comumente recorrem a computadores remotos para resolver o problema de localização de um robô [Dunkley et al. 2014]. [von Stumberg et al. 2016] apresenta um sistema de navegação autônomo de baixo custo computacional, para ser embarcado em hardwares dentro das restrições de peso, consumo de energia e desempenho.

A eficiência energética em sistemas de localização é abordado pelos autores do *Blue-fi* [Ananthanarayanan and Stoica 2009], uma solução desenvolvida para monitorar o ambiente através da interface *bluetooth* e da triangularização de torres GSM, habilitando as funções de WIFI apenas quando necessário. [Cheng and Hsiu 2015] desenvolvem um algoritmo de programação dinâmica para otimizar consumo de energia através de agendamentos para utilização de sensores na estimativa de localização. Em [Neishaboori and Harras 2013], para contornar o problema de eficiência energética, os autores abordam a medição por aglomeração de alvos, se baseando em padrões de deslocamento, e não no comportamento de um indivíduo. A tabela 1 mostra que interfaces de rede sem fio de padrões diferentes podem ser consideradas complementares em fatores como *throughput*, alcance e consumo de energia [Rahmati and Zhong 2007].

Tabela 1. Comparação de largura de banda e consumo médio de energia de diferentes estados entre dispositivos GSM, WIFI e bluetooth

	Banda de dados	Transferência (J/MB)	Idle(W)	Procura(W)	Alcance(m)
GSM (3G)	Menos de 1Mbps	100	-	-	500
WIFI	11-54 Mbps	5	0.77	1.29	100
Bluetooth	723kbps	0.1	0.01	0.12	10

4. Metodologia Proposta

Este trabalho apresenta como contribuição uma metodologia de baixo custo para localização automatizada de alvos transmissores de sinal de radiofrequência compatível com os padrões IEEE 802.11 e *bluetooth*, empregando fusão de sensores. A solução desenvolvida será embarcada em um VANT em missões de voo autônomo, buscando maior redundância e taxa de acerto nas estimativas de posicionamento da aeronave.

Ao percorrer uma rota de voo, a aeronave busca estimar a origem de emissores de radiofrequência através de sinais coletados em tempo real. Levando em consideração diferenças entre cada padrão de rádio, estão sendo desenvolvidas bibliotecas de coleta especializadas para cada sensor empregado. As bibliotecas armazenam seus resultados em base de dados local, que são consumidos por sistemas especializados em estimar a distância relativa em intervalos de tempo. Em paralelo, o VANT mantém informações de navegação por GPS caso seja necessário averiguar a posição absoluta de um alvo.

Para essa proposta não há a necessidade de transmitir pacotes de dados entre a aeronave e o ativo emissor, ou vice-versa. Toda a informação necessária para os algoritmos de estimativa é obtida pela força de sinal recebida (*received signal strength indication*, RSSI) e outros atributos referentes a qualidade da transmissão sem fio.

Partindo da hipótese que a técnica de *wardriving* pode ser empregada como recurso para a localização automatizada e de que a análise de sinais de radiofrequência emitidos por antenas *bluetooth* podem gerar maior precisão e/ou velocidade na estimativa, as contribuições esperadas para o presente trabalho são:

1. Determinar as variáveis relevantes para localização autônoma de ativos;
2. Colaborar com outras linhas de pesquisa em localização por radiofrequência;
3. Comparação de métodos existentes de localização por *wardriving*; e
4. Desenvolver um *framework opensource* escalar, capaz de comportar mais sensores e técnicas, acoplados por bibliotecas auxiliares.

5. Arquitetura Proposta

O relacionamento entre os componentes do *framework* proposto é descrito na figura 2. A primeira etapa coleta dados utilizando rádios de diferentes padrões. O resultado de cada amostra é registrado em base com o *timestamp* exato para depois ser consumido pelas ferramentas de cálculo de distância. Essas estimativas, específicas para cada tecnologia, também são registradas com o mesmo *timestamp* das amostras para então serem utilizadas pelo algoritmo de fusão de sensores, que oferece um resultado final de posicionamento relativo à aeronave. Coordenadas absolutas de voo, obtidas em paralelo pelo GPS, também podem ser utilizadas para gerar uma saída com o geoposicionamento da fonte emissora.

Os componentes responsáveis pela captura de dados de redes WIFI e *bluetooth* são as bibliotecas WIFIList e BTList, respectivamente. Toda informação obtida é persistida em banco de dados, contendo uma base histórica dos registros de voo. A modelagem do banco tem foco na retenção das informações geradas pelas bibliotecas e na manutenção do sincronismo entre os dados capturados e as localizações estimadas.

Os valores capturados e armazenados são usados nos cálculos dos algoritmos de localização: valor do RSSI, qualidade do sinal e frequência de transmissão. Para a

determinação da identidade de uma fonte são utilizados o endereço físico do hardware (*MAC Address*), tipo de criptografia adotada, nome da rede e modo de interface.

As bibliotecas de captura e a biblioteca que implementa o algoritmo de fusão de sensores estão sendo desenvolvidas em python. Dessa forma, o *framework* será modularizado por bibliotecas proporcionando uma arquitetura de baixo acoplamento e escalar, para futuramente suportar novas tecnologias, caso necessário.

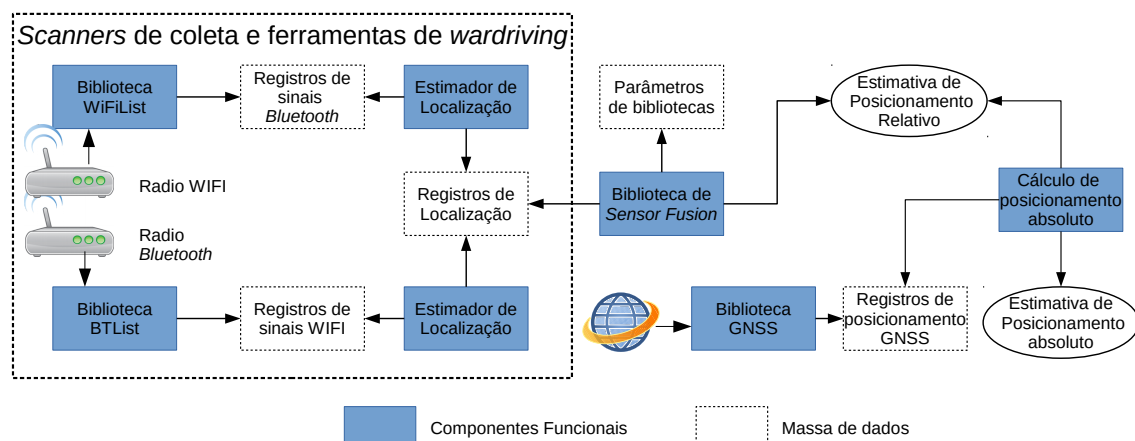


Figura 2. *Framework* proposto para a metodologia de localização

A prototipação de toda a infraestrutura que será embarcada no VANT está sendo realizada junto ao desenvolvimento do *framework* e serão feitos testes em cada uma das suas bibliotecas separadamente, antes do voo de validação.

5.1. Sistema Embarcado

Está sendo adotada a versão mais nova do Raspberry Pi 3 Model B (revisão 1.2). Esse modelo foi escolhido por incorporar um chip Broadcom BCM43438 que provê rádios para redes sem fio nos padrões IEEE 802.11n (2.4GHz), *bluetooth Low Energy* e *bluetooth 4.1*, dispensando o uso de periféricos no Barramento Serial Universal (*Universal Serial Bus*, USB). Essa placa tem 85,6mm de comprimento por 56mm de largura e 21mm de espessura, consumindo 1000mA/h, no máximo. Sua CPU possui 4 núcleos a 1,2 GHz (ARMv8 64-bits), além de suportar cartões de memória microSD e possuir uma interface de propósito geral (*General Purpose Input/Output*- GPIO) [RaspberryPi 2016].

Dentre os sistemas operacionais compatíveis com o hardware, foi adotado o Raspbian, um sistema Linux-ARM baseado na distribuição Debian. Sua escolha se deve ao grande suporte da comunidade de usuários e por já possuir módulos compilados em seu repositório para integração com os sistemas de voo *opensource* adotados no projeto.

5.2. Unidade Inercial de Movimento

Para o controle de voo, foi adotada a placa Navio2 da fabricante EMLID. Esse hardware possui duas Unidades de Medida Inercial (*Inertial Measurement Unit* - IMUs) com acelerômetro, giroscópio e magnetômetro de três eixos, Barômetro e GPS. Suporta Modulação por Largura e por Posição do Pulso (*Pulse Width Modulation* – PWM e *Pulse-Position Modulation* - PPM), e extensão de comunicação serial (*Universal Asynchronous Receiver/Transmitter* - UART). Suporta os padrões de barramento *Inter-Integrated Circuit*

(I2C) e *Analog-to-Digital Converter* (ADC). A placa tem dimensões de 55mm de comprimento por 65mm de largura e peso de 23g que suporta o piloto automático ArduPilot.

O processo de acoplamento do hardware de piloto automático com a placa-mãe representa a integração completa do sistema computacional embarcado. A configuração desse agente de controle exige a instalação de espaçadores no lado superior do Raspberry, parafusos e cabeçalho de extensão da porta GPIO entre as placas bem como a confecção de um gabinete personalizado para embarcar a solução na aeronave [Avkhimena 2015].

5.3. Plataforma de voo

Para os ensaios de voo, optou-se por um VANT do tipo asa rotatória de pequeno porte (ilustrado na figura 3) por ser capaz de comportar a carga necessária para customizações e por permitir controles precisos de altitude e velocidade. O aeromodelo adotado, utiliza a estrutura e propulsores do fabricante DJI, modelo F450. Possui 90cm de diâmetro e 282g sem os acessórios e um compartimento para eletrônica capaz de suportar até 1200g de carga extra. Suporta baterias LiPo do tipo 3S e 4S, permitindo uma autonomia de até 15 minutos de voo. Esse *frame* possui furações extras para customizações necessárias para fixação do gabinete com o sistema computacional embarcado.



Figura 3. VANT utilizado nos testes de voo, com carga paga de 1.2 kg.

6. Experimentos Preliminares

Nos primeiros testes de coleta de dados, foi verificada a aplicabilidade da biblioteca WIFIList na detecção de redes WIFI pelo hardware embarcado na placa Raspberry Pi 3 em execuções de coleta de sinais como utilizados em técnicas de *wardriving*.

Para a realização desse experimento foram utilizadas redes já instaladas e configuradas do Instituto Militar de Engenharia, não sendo possível obter todas as informações sobre os *access points* como, localização exata e versão do padrão IEEE 802.11.

Para validar o teste da biblioteca WIFIList, que consistia em *scanear* as redes WIFI a partir de um determinado ponto fixo, Foi utilizado somente o Raspberry Pi 3.

A biblioteca WIFIList foi executada 50 vezes por dia em cinco dias diferentes. A tabela 1 exhibe a média de redes encontradas e o tempo de execução, enquanto a tabela 2 exhibe a frequência de quantas vezes uma determinada rede foi detectada e a média dos valores de RSSI e da qualidade de sinal, parâmetro indexado de 0 a 70 pontos.

Tabela 2. Média do tempo total de execução do algoritmo por dia e média da quantidade de rede encontrada.

Data	Tempo de execução (s)	Nº de redes encontradas
03/05/2016	1,31	10
04/05/2016	1,24	10
05/05/2016	0,35	05
06/05/2016	0,30	04
09/05/2016	0,29	04

Tabela 3. Frequência de detecção de uma rede durante a execução, média do Valor de RSSI e da qualidade de sinal (0-70 pontos).

Identificador de rede	Detecções	Média do RSSI (dBm)	Média da qualidade do sinal
1	208	-28	69
2	125	-61	49
3	125	-59	51
4	124	-67	43
5	125	-62	48
6	111	-78	32
7	66	-67	15
8	106	-83	27
9	105	-77	33
10	70	-57	26
11	62	-66	16
12	10	-24	4
13	3	-46	10
14	46	-32	23
15	20	-23	5

As redes identificadas como 1 e 2 estavam no mesmo ambiente *indoor* que o sensor WIFI, a 30 cm e a aproximadamente 1 m de distância respectivamente, sendo a rede 1 a única com uma linha de visão direta para o módulo WIFI. Já os outros *access point* estavam distribuídos pela edificação.

Analisando os resultados, é possível perceber que os pontos de acesso encontrados com maior frequência foram os que possuem o menor módulo de RSSI e melhor qualidade de sinal. Isso comprova que quanto menor o valor absoluto do RSSI e quanto melhor a qualidade do sinal, mais perto o *Access Point* se encontra do módulo WIFI.

7. Conclusão

Equipar veículos não tripulados com a capacidade autônoma de navegação fortalece a sua aplicação em ambientes onde a assistência humana seja inviável ou onde espera-se um certo nível de automação. Tarefas de busca, pousos em *docking* para ancoragem e recarga ou para rotinas de navegação do tipo “*follow me*”, onde o veículo segue um usuário em movimento, são exemplos de problemas onde a metodologia é aplicável.

A metodologia proposta pode melhorar a localização de um ponto relativo ao VANT sem depender de fontes de dados ou recursos computacionais externos. A

utilização de múltiplos sensores de radiofrequência, além de oferecer redundância e precisão, possibilita a distribuição de suas antenas melhorando a detecção de redes e evitando interferências provocadas por outros componentes.

O *framework* proposto na metodologia é escalável. Pela popularidade e disponibilidade em dispositivos embarcados, na pesquisa foram adotados os padrões WIFI (IEEE 802.11) e *bluetooth*. Esses radiotransmissores podem ser diferenciados por assinaturas baseadas no endereço MAC, sendo boas escolhas para identificação de fontes emissoras. Porém, com o hardware necessário e novas bibliotecas de captura apropriadamente implementadas, outros sensores poderão ser integrados.

Como trabalhos futuros, serão realizados novos testes utilizando a biblioteca WIFIList com diferentes disposições de antena e rádios, independentes do hardware adotado. Também será implementado o suporte a registros de parâmetros *bluetooth*. Ensaio embarcados no VANT serão executados para garantir a viabilidade em voo autônomo, de modo que o algoritmo de fusão de sensores aproveite uma granularidade de informações.

8. Agradecimentos

Esse projeto foi parcialmente financiado pelo Programa Demanda Social/Capes e pelo Edital 21-2011 Pró-Estratégia, Capes.

Referências

- Amendolare, V., Cyganski, D., Duckworth, R. J., Makarov, S., Coyne, J., Daempfling, H., and Woodacre, B. (2008). Wpi precision personnel locator system: Inertial navigation supplementation. In *2008 IEEE/ION Position, Location and Navigation Symposium*.
- Ananthanarayanan, G. and Stoica, I. (2009). Blue-fi: Enhancing wi-fi performance using bluetooth signals. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, number 1 in MobiSys '09, NY, USA. ACM.
- Avkhimena, M. (2015). Navio2. <https://emlid.com/introducing-navio2>. 2017-03-23.
- Cheng, C.-C. and Hsiu, P.-C. (2015). Extend your journey: Considering signal strength and fluctuation in location-based applications. *IEEE/ACM Trans. Netw.*, 23(2).
- Dunkley, O., Engel, J., Sturm, J., and Cremers, D. (2014). Visual-inertial navigation for a camera-equipped 25g nano-quadrotor. In *IROS2014 Aerial Open Source Robotics*.
- Gomes, H. S. (2016). Drones são usados por bombeiros de são paulo para combater incêndios. 2016-11-28.
- Haerberlen, A., Flannery, E., Ladd, A. M., Rudys, A., Wallach, D. S., and Kavraki, L. E. (2004). Practical robust localization over large-scale 802.11 wireless networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, number 1 in MobiCom '04, NY, USA. ACM.
- Hawkinson, W., Samanant, P., McCroskey, R., Ingvalson, R., Kulkarni, A., Haas, L., and English, B. (2012). Glanser: Geospatial location, accountability, and navigation system for emergency responders - system concept and performance assessment. In *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*.
- Kotaru, M., Joshi, K., Bharadia, D., and Katti, S. (2015). Spotfi: Decimeter level localization using wifi. *SIGCOMM Comput. Commun. Rev.*, 45(4).

- Lin, K., Kansal, A., Lymberopoulos, D., and Zhao, F. (2010). Energy-accuracy trade-off for continuous mobile device location. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10. ACM.
- Neishaboori, A. and Harras, K. (2013). Energy saving strategies in wifi indoor localization. In *Proceedings of the 16th ACM International Conference on Modeling, Analysis; Simulation of Wireless and Mobile Systems*, MSWiM '13, NY, USA. ACM.
- Prieto, J., Mazuelas, S., Bahillo, A., Fernandez, P., Lorenzo, R. M., and Abril, E. J. (2012). Adaptive data fusion for wireless localization in harsh environments. *IEEE Transactions on Signal Processing*, 60(4).
- Rahmati, A. and Zhong, L. (2007). Context-for-wireless: Context-sensitive energy-efficient wireless data transfer. In *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, MobiSys '07, NY, USA. ACM.
- RaspberryPi (2016). Specification. <https://www.raspberrypi.org>. 2017-03-23.
- Sapiezynski, P., Gatej, R., Mislove, A., and Lehmann, S. (2015). Opportunities and challenges in crowdsourced wardriving. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, number 1 in IMC '15, NY, USA. ACM.
- Thomas King, Stephan Kopf, W. E. (2006). A location system based on sensor fusion: Research areas and software architecture. *Proc. of 2nd GI/ITG KuVS Fachgespräch "Ortsbezogene Anwendungen und Dienste"*, Stuttgart, Germany, 1.
- Tsui, A. W. T., Lin, W. C., Chen, W. J., Huang, P., and Chu, H. H. (2010). Accuracy performance analysis between war driving and war walking in metropolitan wi-fi localization. *IEEE Transactions on Mobile Computing*, 9(11).
- von Stumberg, L., Usenko, V., Engel, J., Stückler, J., and Cremers, D. (2016). Autonomous exploration with a low-cost quadcopter using semi-dense monocular SLAM. In *arXiv:1609.07835*.
- Weber, M., Liu, L., Jones, K., Covington, M. J., Nachman, L., and Pesti, P. (2010). On map matching of wireless positioning data: A selective look-ahead approach. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, number 1 in GIS '10, NY, USA. ACM.
- Wu, K., Xiao, J., Yi, Y., Gao, M., and Ni, L. M. (2012). Fila: Fine-grained indoor localization. In *2012 Proceedings IEEE INFOCOM*.
- Yu, E., Xiong, X., and Zhou, X. (2016). Automating 3d wireless measurements with drones. In *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*, WiNTECH '16. ACM.
- Zhang, Z., Zhou, X., Zhang, W., Zhang, Y., Wang, G., Zhao, B. Y., and Zheng, H. (2011). I am the antenna: Accurate outdoor ap location using smartphones. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, number 1 in MobiCom '11, NY, USA. ACM.
- Zhu, H., Zhong, X., Yu, Q., and Wan, Y. (2013). A localization algorithm for mobile wireless sensor networks. In *2013 Third International Conference on Intelligent System Design and Engineering Applications*.

Reducing energy consumption on internal communications of unmanned vehicles with HAMSTER's Navigation Phases

Eduardo Aramizu¹, Daniel Fernando Pigatto²,
Kalinka Regina Lucas Jaquie Castelo Branco¹

¹Instituto de Ciências Matemáticas e de Computação –
Universidade de São Paulo (USP)
13.566-590 – São Carlos – SP – Brazil

²Universidade Tecnológica Federal do Paraná – Câmpus Curitiba
80.230-901 – Curitiba – PR – Brazil

duaramizu@gmail.com, danielpigatto@gmail.com, kalinka@icmc.usp.br

Abstract. *Unmanned vehicles have been creatively explored on a wide range of applications. Towards the efficient operation of either unmanned vehicles and systems they integrate, communications improvements associated with security, safety and energy efficiency are highly important research topics. Moreover, it is highly important to provide energy efficient approaches that do not compromise the performance, but take advantage on reducing energy consumption as much as possible. This paper presents Navigation Phases platform that helps improving energy efficiency on unmanned vehicles. The simulated and field results demonstrate the applicability of Navigation Phases and elucidate how related concepts such as fly by wireless can also face benefits.*

1. Introduction

A brief definition for Unmanned Vehicle (UV) is a vehicle without a person on board. Also seen as uncrewed vehicles, they can be remotely or autonomously controlled, usually applied in a wide range of environmental sensing activities, high risk areas monitoring, driving assistance, monitoring activities and much more. There is massive research interest on this field as smart drones, cars and boats take place, get cheaper, easier to control and even more integrated to everyday situations. However, one of the biggest concerns in these vehicles is the energy consumption.

The HeAlthy, Mobility and Security-based data communication archiTEctuRe (HAMSTER) is an architecture that provides well-defined ways of implementing communications in unmanned vehicles and systems [Pigatto et al. 2016b]. HAMSTER is divided into three main versions according to the most common types of UV: aerial (Flying HAMSTER), ground (Running HAMSTER) and aquatic (Swimming HAMSTER). Moreover, four special elements are defined: i) a platform intended to control security and safety aspects under all architecture versions, named SPHERE; ii) an index to evaluate node criticality within a network, named NCI; iii) a platform that aims at mobility aspects, named NIMBLE; and iv) a platform for the provision of efforts towards energy efficiency, named Navigation Phases (NP).

This paper provides experiments and results on the reduction of energy consumption using Navigation Phases platform. Section 2 presents the Navigation Phases platform from HAMSTER architecture; section 3 presents the methodology for experiments execution; section 4 presents simulations; section 5 presents results from field experiments; and section 6 presents the conclusions.

2. Navigation Phases platform

A navigation phase is a very well defined UV operation stage where it is attributed at least an ON/OFF state and different transmission rate permissions for each single module. The most common classification splits modules into two categories [Pigatto et al. 2016a]: “mission-specific” and “main” modules. For instance, a UAV would have at least three main phases: takeoff, cruise and landing. Each phase would prioritise different modules over others, considering their individual demand for operation.

A general classification of UV navigation phases is presented in Table 1. There are five main phases and an emergency one. Main phases will usually follow a predictable order. On the other hand, emergency phases can be started at any time to treat adverse conditions. Each navigation phase defines which category of nodes is allowed to be working at that specific situation.

Table 1. General navigation phases for unmanned vehicles.

Navigation phases		Sub-navigation phases		Active modules
1	Start-up	1.1	Modules health checking	All
		1.2	Energy supply verification	All
		1.3	Authentication	All
2	Operation initialisation	2.1	Operation start	Main
		2.2	Moving to the target field	Main
3	Mission execution	3.1	Positioning on the field	Main
		3.2	Performing mission	All
4	Shutdown	4.1	Moving back home	Main
		4.2	Preparing to stop operation	Main
		4.3	Turning off vehicle	Main
5	Post-operation	5.1	Modules health checking	All
		5.2	Mission data acquisition	Mission-specific
E	Emergency situations	E.1	Operation abortion and home returning	Main
		E.2	Operation abortion and vehicle turning off	Main
		E.3	Data self-destruction (wipe data)	Mission-specific
		E.4	Stabilising (after non-predicted disturbances)	Main

The **Start-up** phase is dedicated for several preoperational tasks, e.g. modules health checking, energy supply verification and authentication. Probably, the transmission rates allowed for each node will be similar, considering that performed tasks are not differently implemented for adversary classes of nodes. That is not the case of **Operation initialisation** phase, which must prioritise UV main modules over mission-specific ones to guarantee that the UV will start operating and will successfully move to the target field. This can be considered a safety-critical operation that should meet restrict time requirements.

Next, a natural phase is the **Mission execution**. Unmanned vehicles will usually be designed mainly (often exclusively) for that purpose. Therefore, the

prioritisation of mission modules is an approach that might take place. Although very important, mission-specific nodes cannot operate by themselves; thus, all nodes should work in this phase with different transfer rates. Depending on the module, transfer rates may vary. From mission execution, the UV will likely move on to a **Shutdown** phase. The operation is similar to the Operation initialisation in criticality terms.

Finally, the UV must be put in a **Post-operation** phase. This stage can have different operation patterns. For instance, while in mission data acquisition sub-phase, distinct transfer rates will be allowed to each mission-specific module depending on the importance or amounts of data they have stored throughout the operation.

Emergency situations phase concentrates adversary and unpredictable situations that may be experienced by the system. For instance, if batteries are in critically low levels, the UV just collided, or a possible unauthorised entity is trying to steal sensitive information from the UV, an emergency situation might be started.

Figure 2 presents the interaction between NP Agent and central NP Manager.

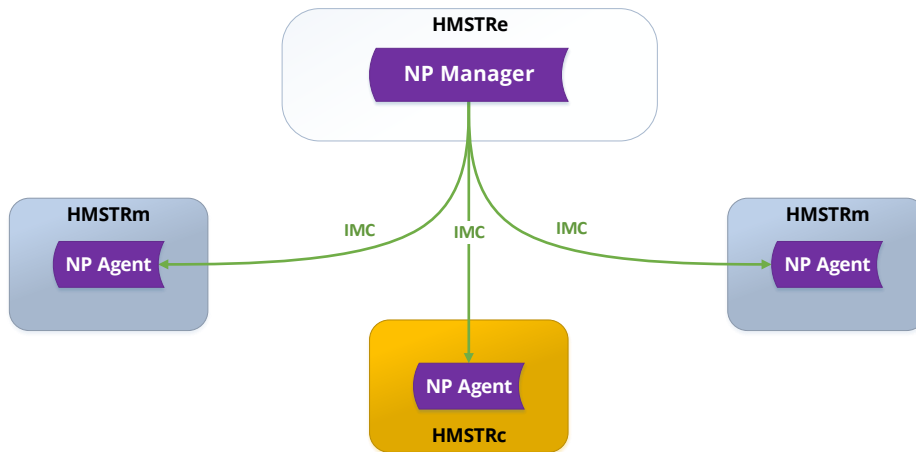


Figure 1. Navigation Phases interaction: NP Agent implements the current phase on each module (HMSTRm) and cluster of modules (HMSTRc); the information about the current phase is provided by NP Manager that runs on the entity (HMSTRe) e.g. a UV. The internal machine communications is represented by IMC [Pigatto et al. 2016b].

3. Material and methods

Navigation Phases concept was adapted to a Unmanned Aircraft Systems (UAS) and is presented in Table 2. This table will be used as reference for both simulated and field experiments.

Pre-flight phase is dedicated to several inspections. It is composed of only one sub-phase, which performs authentication and monitors “health” and energy. The transmission rate and the size of the exchanged messages are the same for all modules.

Table 2. Navigation phases applied to UAS.

Navigation phases	Navigation Sub-phases	ID	Description	Active modules	Identifier
1	Pre-flight	1.1	Modules health, energy and authentication checking	All nodes	ALL
		2.1	Taxiing	Main nodes only	MAIN
2	Departure and climb	2.2	Taking-off	Main nodes only	MAIN
		2.3	Climbing	Main nodes only	MAIN
		3.1	Stabilising from climbing	Main nodes only	MAIN
3	Cruise	3.2	Heading to the destination	All nodes	ALL
		3.3	Performing mission	All nodes	ALL
		3.4	Preparing to descent	Main nodes only	ALL
		4.1	Descending	Main nodes only	MAIN
4	Descent and approach	4.2	Landing	Main nodes only	MAIN
		4.3	Taxiing	Main nodes only	MAIN
		5.1	Modules health, energy and authentication checking	All nodes	ALL
5	Post-flight	5.2	Mission data manipulation	Mission nodes only	MISSION
		E.1	Returning to the Ground Control Station	Main nodes only	MAIN
E	Emergencies	E.2	Landing ASAP	Main nodes only	MAIN
		E.3	Starting self-destruction (wipe data)	Mission nodes only	MISSION
		E.4	Stabilising (after non predicted movements)	Main nodes only	MAIN

Departure and climbing phase occurs when the UAV is moving on the ground, taking off and stabilising in the air. At this phase, the UAV works exclusively with modules classified as Main since it is a critical phase.

Cruise phase is usually the longest flight. The UAV reaches a specific altitude (which can vary during the phase), stabilises, go to the destination (e.g., mission execution site), performs the mission and prepares for the next phase. At this stage, there is a greater variation about which nodes will be active in each sub-phase.

Descent and approach phase is the period when the UAV starts to descent, landing and then moving on the ground. Only modules classified as Main are allowed to exchange messages at this stage.

Post-flight phase the first phase checking is performed again and the acquisition and manipulation of mission data also takes place. The frequency and the amount of data exchanged at this stage varies for each module.

And finally, **Emergencies** phase includes various abnormal situations, such as power outages, flight difficulties, adverse weather conditions, unexpected obstacles, security attacks etc. For each case, a procedure is triggered trying to circumvent the problematic situation, save/delete sensitive data and prevent the UAV to cause some kind of injury.

4. Simulation of Navigation Phases

First, simulated results were performed for an initial behaviour analysis. A set of 30 modules were classified into Main and Mission-specific categories. Main modules are essential for the aircraft operation and must be working at almost all the navigation phases. Mission-specific modules can be turned off at various stages of operation, reducing energy consumption. Figure 4 demonstrates the modules positioning on simulated experiments.

Considering all the sensors that usually can be found in a UAV, an empiric list was created determining different network usage and power consumption for each node. Figure 4 presents some different situations where the energy consumption can be reduced. The blue line represents the energy consumption of all nodes during

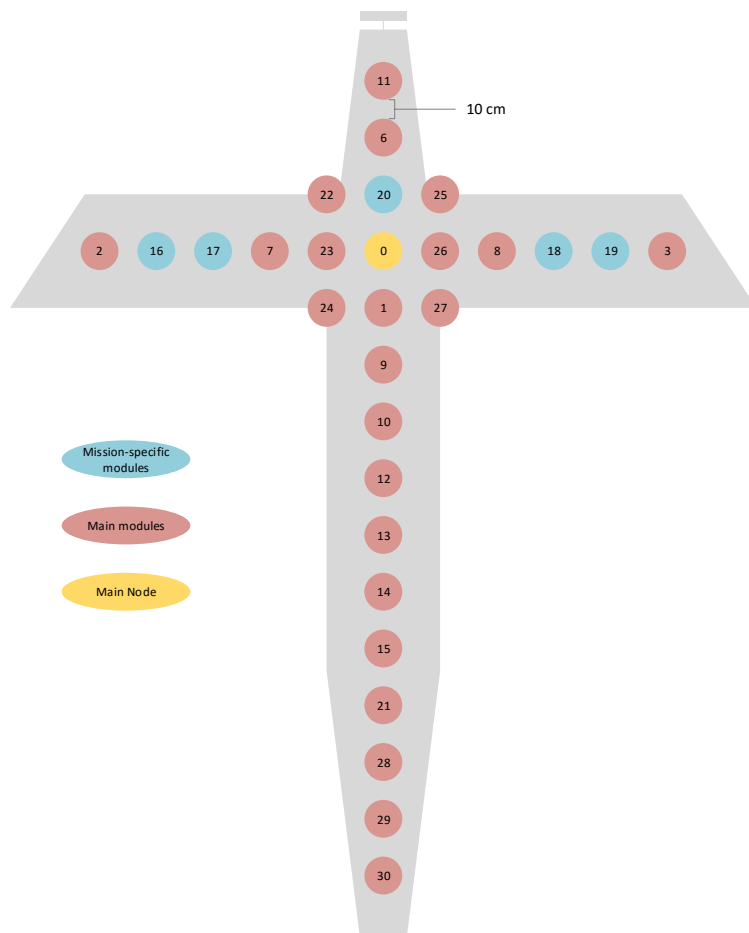


Figure 2. Nodes distributed on the inner side of UAV.

Navigation Phase 1.1. This phase consists on authentication, which is performed before taking off, thus the energy consumption is pretty much the same for every node. Similarly, the red line shows the consumption when all nodes are active and also illustrates the peak in nodes N_{16} to N_{19} , the Mission-specific ones. These nodes were defined as cameras and their energy consumption is bigger. The peak on node N_{21} is a specific case where the position of a node is determinant for its performance and energy consumption. Finally, the grey and yellow lines are either showing the energy consumption of Main nodes and Mission-specific nodes only, respectively.

Figure 4 shows the mean backoff for the same case aforementioned. A binary exponential backoff or truncated binary exponential backoff refers to an algorithm used to space out repeated retransmissions of the same block of data, often as part of network congestion avoidance. In these experiments, the technique took place when high amounts of data were transferred.

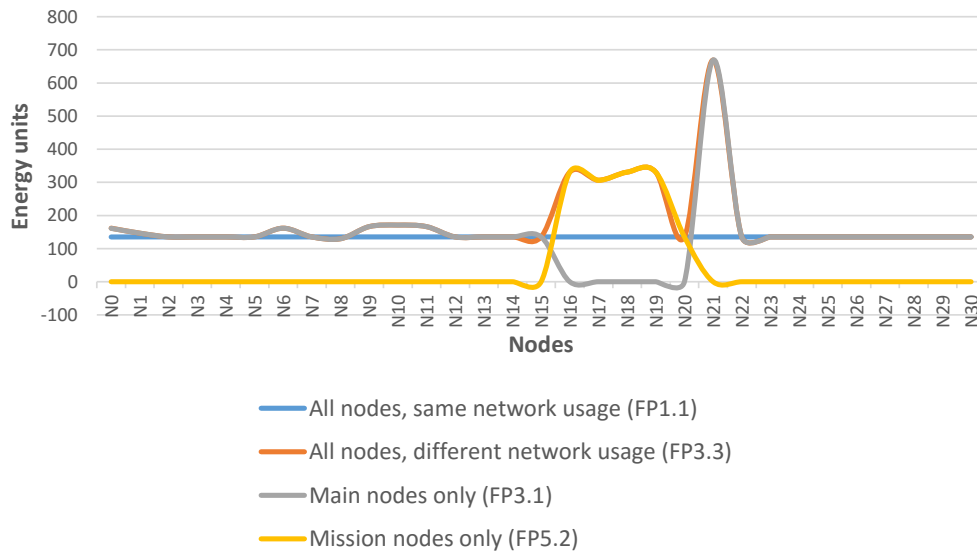


Figure 3. Energy consumption of 31 nodes in fly by wireless communications in four different situations. The Flight Phases identified in each situation can be seen in Table 2.

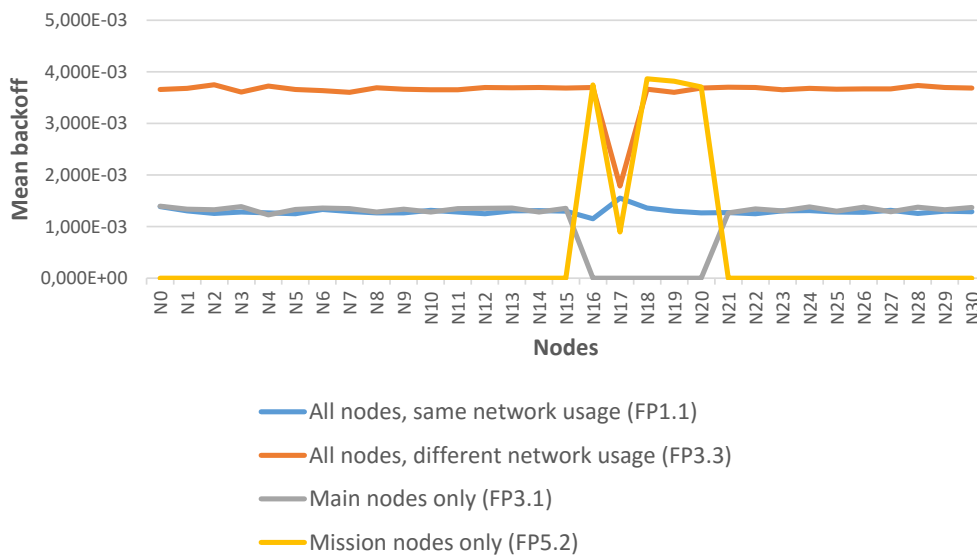


Figure 4. Mean backoff of 31 nodes in four different situations. The Navigation Phases identified in each situation can be seen in Table 2.

5. Field experiments on Navigation Phases

Although simulations may provide a more controlled environment and the possibility of replicating experiments as much as needed, experiments on real prototypes can highlight new issues that a simulator might not consider. Thus, a set of experiments was run using Arduino Leonardo and XBee boards.

Arduino board is equipped with an ATmega32u4 micro controller and has 20 digital input/output pins, a 16 MHz crystal oscillator, a micro USB connection, a power jack, one ICSP (In-Circuit Serial Programming) header and a reset button [Arduino 2016] (see figure 5). In this case study, Leonardo was used exclusively to control the XBee regarding data processing and to determine the cycles that XBee would be active or inactive. Concerning serial communication, Leonardo has an advantage when compared to other Arduino boards, since it has 2 serial ports. Thus, for these experiments, a serial port was used to exchange data with the computer and the other to communicate with XBee. Tests were easily monitored and controlled by the computer.

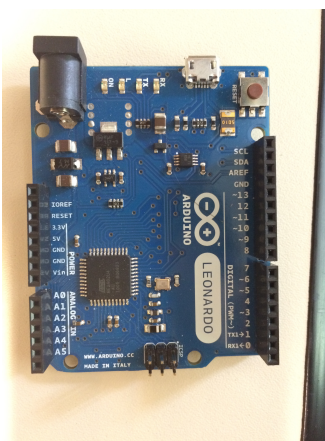


Figure 5. Arduino Leonardo photograph.

A specific expansion shield was used to connect XBee to Arduino Leonardo (see Figure 6). It enables communication between Arduino Leonardo pins and the XBee ports.

The XBees used in this case study were XBee S1 and XBee-PRO S1, both implementing IEEE 802.15.4 standard. Thus, they are suitable for projects with low-cost and low-power requirements. Differences between these XBee models consist basically on the transmitted signal power and the sensitivity to received signals. A comparison is shown in Table 3 [Sparkfun 2016].

The antennas are another important difference between XBee S1 and XBee-PRO S1. The available antennas are wire, printed circuit board (PCB) and Reverse-Polarity Sub-Miniature version A (RPSMA), which can be seen in Figure 5. The PCB antenna has a more directional propagation, having worse or better performance depending on the relative position of the transmitter and receiver. On the other hand, the wire and RPSMA antennas have a more multi-directional propagation [Digi 2016].



Figure 6. Expansion shield to connect XBee to Arduino Leonardo photograph.

Table 3. Specifications of the modules XBee and XBee-PRO.

Specification	XBee	XBee-PRO
Indoor range	up to 30 m	up to 60 m
Outdoor range (line-of-sight)	up to 90 m	up to 750 m
Transmit power	1 mW (0 dBm)	10 mW (10 dBm)
Transmission rate	250 kbps	250 kbps
Receiver sensitivity	-92 dBm	-100 dBm
Supply voltage	2,8-3,4 V	2,8-3,4 V
Transmit current (typical)	45 mA (at 3.3 V)	antenna RPSMA 180 mA
Idle/Receive current (typical)	50 mA (at 3.3 V)	55 mA (at 3.3 V)
Power-down current	< 10 μ A	< 10 μ A

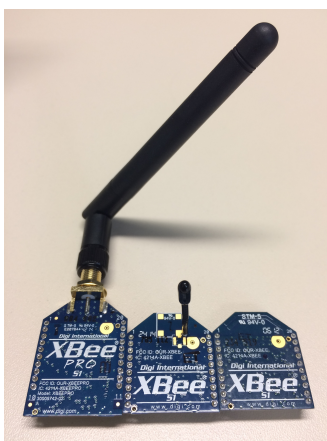


Figure 7. From left to right: RPSMA, wire and PCB antennas.

XBee was chosen due to evidence that it is suitable for fly by wireless applications [Amini et al. 2007, Dawson et al. 2008, Ossa-Gomez et al. 2011, Ossa-Gómez et al. 2011]. An important aspect analysed was the power consumption, considering that battery is a limited resource in embedded systems. In [Stankunas et al. 2011], a comparison between ZigBee, Bluetooth and WLAN proto-

cols was carried out (see Table 4). [Stankunas et al. 2011] summarised the observed characteristics, such as the superior performance of ZigBee protocol when compared to others regarding energy consumption.

Table 4. Bluetooth, WLAN and ZigBee specifications.

Name	Range (m)	Network Topology	Transmission Rate (kbps)	Power (mW)	Bandwidth (MHz)	Module dimension (cm)	Estimated battery time
Bluetooth	1-100	Ad hoc, point-to-point, star	2400	100	2400	31x16x2.2	days-months
WLAN	300	Mesh, ad hoc, star	11000	100	2400	10x10x1	days
ZigBee	up to 400	Mesh, ad hoc, star	250	30	2400, 868, 915	28x18x2	6 months-2 years

According to Dementyev2013, an important aspect is the cyclic sleep provided by XBee, which adds the possibility to activate or deactivate wireless communication modules as needed. Among ZigBee (which is implemented by XBee), Bluetooth and ANT, only ZigBee has cyclic sleep function implemented. In addition to the facility to use cyclic sleep, the communication between nodes can also be quite simple in XBee.

Regarding power consumption measurement, it was necessary to find a way of decoupling the XBee from the expansion shield to measure the current from XBee ports independently from the Arduino Leonardo board. The decoupled scenario can be seen in Figure 8.

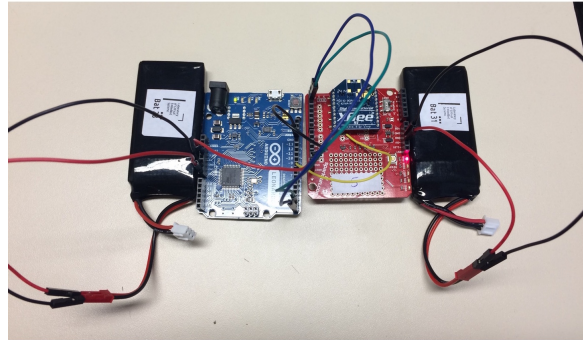


Figure 8. XBee isolated from Arduino Leonardo for current measurement.

To monitor Xbee voltage drop during the experiment, the configuration is shown in Figure 9a. A schematic of the circuit used to perform the voltage measurement can be seen in Figure 9b. In this schematic, the black board on the top left is the XBee.

Table 5 presents smaller set of Navigation Phases for real experiments. It describes the message size that each node must send and the time (in seconds) of each phase. Message sizes are shown in Table 6.

The Arduino program is divided in two stages: the setup that runs only once when the microcontroller is turned on and the loop stage that runs indefinitely afterwards. These specific programs for each node were recorded in the Arduino

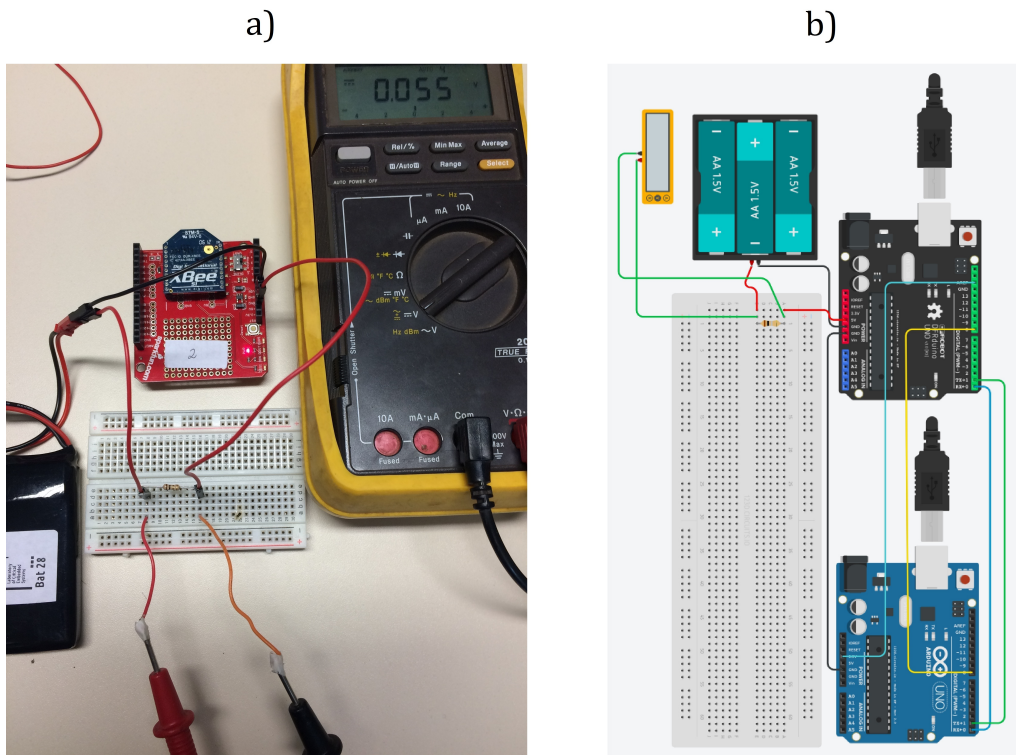


Figure 9. (a) Setup of the circuit used to measure the voltage drop; (b) Schematic of the circuit used to measure the voltage drop.

Table 5. Definition of a mission that contemplates different Navigation Phases. An individual packet size was defined for each and different time durations for each phase.

Navigation Phases	Main nodes			Mission Nodes		Duration (s)
	N1	N2	N3	N4	N5	
1.1	1	1	1	1	1	40
2.1	2	2	2	0	0	30
2.2	4	4	4	0	0	10
2.3	4	4	4	0	0	6
3.1	4	4	4	0	0	2
3.2	4	4	3	5	5	30
3.3	2	2	2	5	5	60
3.4	4	4	4	0	0	20
4.1	4	4	4	0	0	20
4.2	4	4	4	0	0	6
4.3	2	2	2	0	0	30
5.1	2	2	2	2	2	40
5.2	3	3	3	5	5	30

Leonardo boards, defining the behaviour of each XBee. The communication between Arduino Leonardo and XBee was performed by a serial communication. The setup stage is used to configure XBee, to set the serial communication between Arduino and the computer, to define the interruption parameters and to define Arduino pins behaviour as inputs or outputs.

Table 6. Messages size in bytes.

Reference	Size in bytes
Off (0)	0
Low (1)	4
Medium-low (2)	8
Medium (3)	16
Medium-high (4)	32
High (5)	64

The coordinator node and all the XBee boards were set with the same network ID. Moreover, the coordinator was defined as the network coordinator and its transmission mode was set as broadcast, because the only message sent by the coordinator is the command to begin the experiment.

The routine performed by the coordinator node is to receive all packets sent by all the other nodes and count how many packets have been received from each one. Main Nodes (1, 2 and 3) were configured similarly, sending unicast messages to the coordinator. In contrast, nodes 4 and 5 remain inactive during some phases. This is achieved thanks to the cyclic sleep feature that is implemented by *TimerOne.h* library for Arduino. All nodes start to operate at the same time as soon as the coordinator commands. The difference lies in the routine that each node performs.

Messages were sent with a frequency of 2 Hz by defining a delay of 500 ms. A total of 6 experiment replications were carried out. In each replication, nodes 1, 2 and 3 sent 648 messages each, while nodes 4 and 5 sent 400 messages.

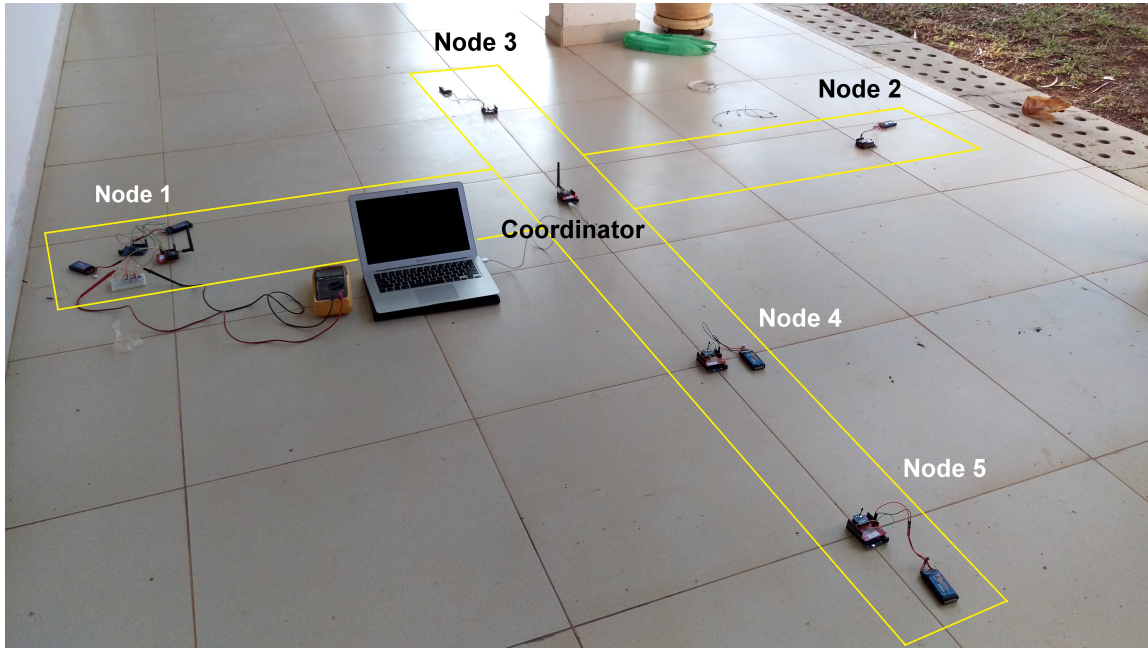


Figure 10. Experimental setup following similar dimensions of a UAV.

Figure 10 identifies each node. The coordinator and node 1 were equipped with a XBee-PRO S1 and RPSMA antennas; nodes 2 and 3 with a XBee S1 and

PCB antennas; and nodes 4 and 5 with XBee S1 and wire antennas. Samples were collected every 4 seconds, totalling 81 measures for each node.

The average power consumed by each node is shown in Figure 11. One can say that, in general, the coordinator node and node 1 spend more energy, which is expected because both are equipped with an XBee-Pro S1. In addition, generally node 1 had higher energy consumption than the coordinator. Nodes 4 and 5 performed similarly to node 3 in energy consumption. Node 2 had a smaller consumption than node 3, even with the fact that both nodes were equipped with the same XBee board and antenna.

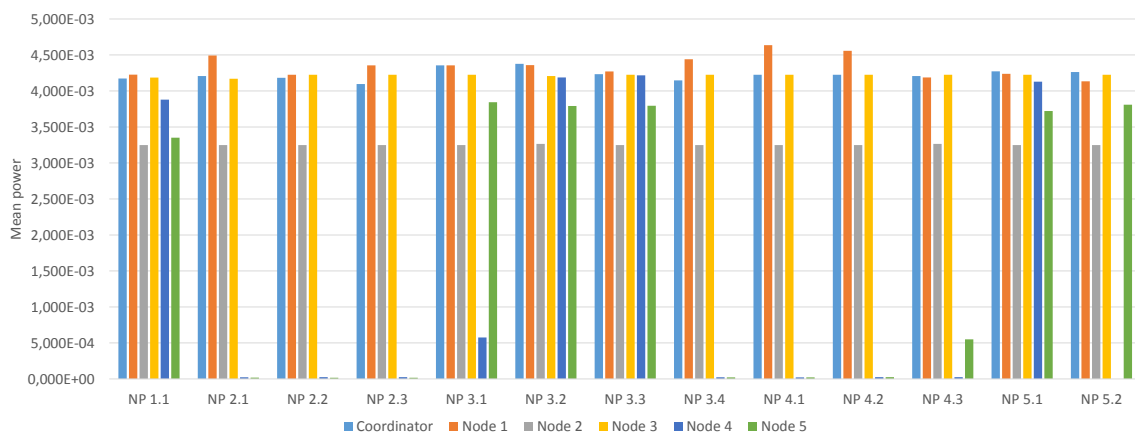


Figure 11. Average power in each Navigation Phase.

Messages size had no relevant influence on energy. On the other hand, inactive XBee periods could be clearly seen and measured. Nodes 4 and 5 were inactive during 39% of the total time. Therefore, if nodes had remained active, both would spend approximately 93% more energy.

Figure 12 shows the percentage of messages received by the coordinator in each experiment. It was calculated based on the quantity of messages sent by nodes. Analysing the data packets received by the coordinator, it can be concluded that about 50% of the packets were lost. Moreover, node 3 had fewer packet loss (30%) due to its position closer to the coordinator.

6. Conclusions

This paper presented results related to fly by wireless with Navigation Phases platform provided by HAMSTER. First, simulated experiments were discussed which proved that it is possible to explore scenarios with an elevated quantity of nodes. However, experiments on a real prototype were also carried out to identify how different XBee boards, antennas and the possibility of inactivate idle nodes interferes on energy consumption.

References

- Amini, R., Gill, E., and Gaydadjiev, G. (2007). The challenges of intra-spacecraft wireless data interfacing. *International Astronautical Federation - 58th International Astronautical Congress 2007*, 6(January):4020–4025.

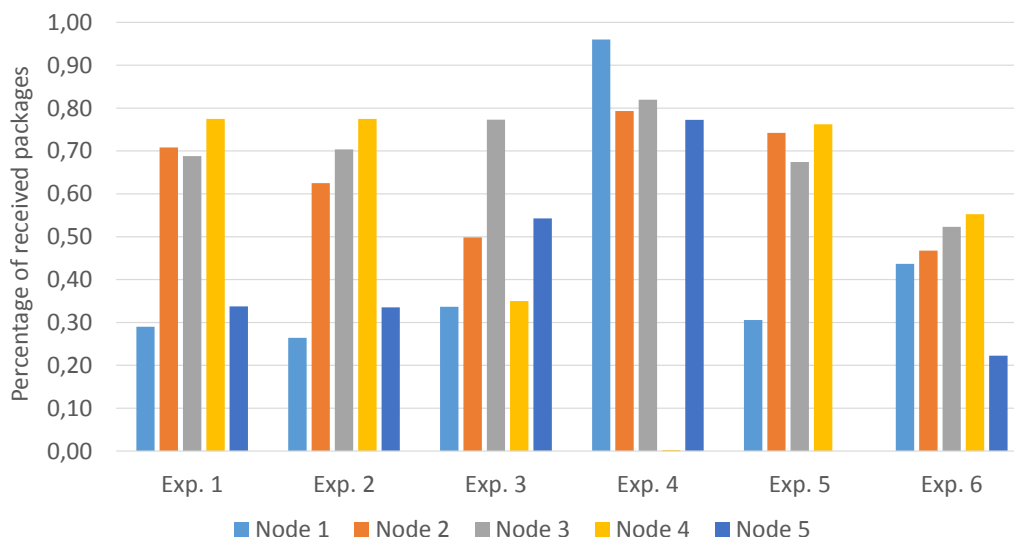


Figure 12. Percentage of received packets by the coordinator in each experiment.

Arduino (2016). Arduino Board Leonardo. <https://www.arduino.cc/en/Main/ArduinoBoardLeonardo>.

Dawson, J. F., Hope, D. C., Panitz, M., and Christopoulos, C. (2008). WIRELESS NETWORKS IN VEHICLES. In *Electromagnetic Propagation in Structures and Buildings, 2008 IET Seminar*. IEE.

Digi (2016). Wireless Mesh Networking RF Module. http://ftp1.digi.com/support/images/XST-AN019a_XBeeAntennas.pdf.

Ossa-Gomez, C., Moarref, M., and Rodrigues, L. (2011). Design, construction and fly-by-wireless control of an autonomous Quadrotor helicopter. In *2011 4th Annual Caneus Fly by Wireless Workshop*, volume 224, pages 1–4. IEEE.

Ossa-Gómez, C., Moarref, M., and Rodrigues, L. (2011). Design, construction and fly-by-wireless control of an autonomous Quadrotor Helicopter. *4th Annual Caneus Fly-By-Wireless Workshop, FBW 11*, pages 79–82.

Pigatto, D., De Castro, A., Branco, K., and Martin, T. (2016a). Aplicação de Fuzzy para a redução do consumo de energia de módulos internos em veículos aéreos não tripulados. In *2016 8th Euro American Conference on Telematics and Information Systems, EATIS 2016*.

Pigatto, D. F., Gonçalves, L., Roberto, G. F., Rodrigues Filho, J. F., Floro da Silva, N. B., Pinto, A. R., and Lucas Jaquie Castelo Branco, K. R. (2016b). The HAMSTER Data Communication Architecture for Unmanned Aerial, Ground and Aquatic Systems. *Journal of Intelligent & Robotic Systems*, pages 1–19.

Sparkfun (2016). Xbee RF Module datasheet. <https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf>.

Stankunas, J., Rudinskas, D., and Lasauskas, E. (2011). Experimental Research of Wireless Sensor Network Application in Aviation. *Electronics and Electrical Engineering*, 111(5):41–44.

I-Jack: Sistema Vestível Inteligente para Coleta e Avaliação de Dados Fisiológicos

Elpidio Holanda Júnior¹, Lucas Farias da Cruz¹, Gustavo Luiz Pinheiro da Silva¹, Vandermi João da Silva¹, Raimundo da Silva Barreto²

¹ Universidade Federal do Amazonas (UFAM) – Campus de Itacoatiara
Amazonas – AM– Brasil

² Universidade Federal do Amazonas (UFAM) – ICOMP
Amazonas – AM– Brasil

(elpidio.ufam, lucasfariasdacruz, gustavo.eng18)@gmail.com,
rbarreto@icompu.ufam.edu.br, vandermi@ufam.edu.br

Resumo. *Profissionais de educação física enfrentam alguns problemas para coletar dados fisiológicos dos praticantes de atividades físicas. Neste caso é necessário que praticantes utilizem dispositivos capazes de realizar a medição desses dados. Neste contexto este artigo propõe o uso da tecnologia Wearable, sistema vestível, utilizando sensores conectados a uma jaqueta denominada I-Jack, capaz de monitorar dados fisiológicos tais como: temperatura corporal, frequência cardíaca, atividades e stress do praticante ao longo da execução dos exercícios físicos, com o objetivo de auxiliar profissionais de educação física na recomendação de atividades mais adequadas para cada tipo de usuário.*

1. Introdução

Segundo pesquisas realizadas pelo Instituto Brasileiro de Geografia e Estatísticas (IBGE, 2014), a cada cem brasileiros, quarenta e seis não praticam atividades físicas suficientes. Este índice mostra que a quantidade de pessoas que se tornam sedentárias é grande, levando a população a ter mais chances de contrair doenças.

A Pesquisa Nacional de Saúde (PNS) recomenda a prática de atividades físicas por pelo menos cento e cinquenta minutos na semana (Ministério da Saúde, 2015). Isso mostra que praticar atividades físicas em média trinta minutos por dia durante a semana, garante uma qualidade de vida melhor.

Para se ter um maior controle da saúde ao realizar estas atividades, necessita-se que ocorra o acompanhamento periódico dos parâmetros fisiológicos dos praticantes, ao desenvolverem certo tipo de exercícios físicos, não só como medida preventiva, mas também visando um melhor aproveitamento dos exercícios.

Com o surgimento da computação em nuvem, que é a capacidade de armazenamento de dados na internet, em conjunto com a evolução dos sistemas embarcados, há a possibilidade de se desenvolver soluções para resolver tarefas do cotidiano, assim pode-se ter um sistema móvel capaz de capturar dados e torná-los acessíveis a qualquer pessoa no mundo (Coutinho, Souza, Gomes & Souza, 2013).

No cenário tecnológico, o conceito de coisas conectadas está em pleno desenvolvimento, e atualmente é possível encontrar diversos dispositivos com esta

tecnologia no mercado. Tais dispositivos podem ser usados para interagir diretamente com a rede mundial de computadores (Web). Além disso, IoT do termo em inglês *Internet of Things*, desponta como uma dessas tecnologias para facilitar a troca de dados e serviços entre dispositivos conectados. IoT propõe uma infraestrutura de Tecnologia da Informação, (TI) em que objetos trocam informações na rede de forma segura e facilitada (Rolf HW, Romana W., 2010). Estes ambientes tecnológicos permitem que qualquer dispositivo eletroeletrônico possa receber um endereço de Internet Protocol (IP) e a partir daí se comunicar com a Web, permitindo o controle remoto do ambiente conectado.

Por consequente, este trabalho apresenta um sistema vestível utilizando sensores conectados a uma jaqueta denominada I-Jack, capaz de capturar dados fisiológicos tais como: temperatura corporal, frequência cardíaca e atividades do indivíduo ao longo das suas atividades físicas. Este estudo tem como propósito auxiliar profissionais de educação física no monitoramento dos dados capturados da jaqueta, após a captura dos dados, os mesmos serão enviados a um servidor na nuvem, onde as informações coletadas terão disponibilidade, para que possam ser acessadas de qualquer lugar utilizando sistemas computacionais com acesso a internet.

2. Trabalhos Relacionados

No trabalho de Watthanawisuth *et al.* (2010), foi desenvolvido um sistema de monitoração de dados fisiológicos portátil sem fio, o sistema utiliza oxímetros de pulso para coleta de dados de frequência cardíaca e saturação de oxigênio no sangue dos usuários, com monitoramento em tempo real baseado em rede sem fio ZigBee. Uma placa micro controladora é usada para o recebimento e transferência dos dados para um servidor que tem a função de armazenar esses dados. O trabalho apresentado diferencia-se deste, no quesito da visualização dos dados que está centralizada em um computador, pois neste trabalho, os dados poderão ser consultados não somente por um computador central, mas também por outros dispositivos móveis que possuam acesso à internet.

Em Zulkifli *et al.* (2012), foi apresentado um sistema que utiliza a topologia de rede denominada mesh (rede em malha), com intuito de permitir que médicos e enfermeiros realizem o acompanhamento de seus pacientes simultaneamente de qualquer lugar, sem a necessidade de que a equipe médica esteja próxima ao paciente.

No trabalho de Zulkifli, foram utilizados dois oxímetros de pulso, responsáveis por coletar os batimentos cardíacos dos pacientes. Para processar os dados foram utilizadas placas micro controladoras Arduino Nano e para enviar os dados do sensor para o servidor, foram utilizados dispositivos de rede sem fio XBee, do padrão 802.15.4 (ZigBee), conectados as placas Arduino. O objetivo foi obter dados de frequência cardíaca, enviá-los a um servidor equipado com outro dispositivo XBee que recebe os dados para serem processados e por fim apresentados na tela por meio de uma interface gráfica.

A diferença do trabalho de Zulkifli com este trabalho é a visualização dos dados coletados pelos sensores, pois no trabalho de Zulkifli está condicionada apenas a um servidor não permitindo assim a visualização dos dados via dispositivo móvel. Neste trabalho, os dados não serão apresentados em um servidor e sim disponibilizados pelo servidor para serem apresentados tanto em um computador quanto em dispositivos móveis.

O trabalho de Kini *et. al.* (2015) descreve uma rede de sensores sem fio que tem como principal objetivo realizar o monitoramento de dados fisiológicos como: batimentos cardíacos, temperatura corporal e a oxigenação sanguínea de forma totalmente remota, fornecendo assim uma resposta em tempo real para a equipe médica responsável, com o objetivo de alertá-los quando ocorrer algum tipo de alteração nos padrões de dados dos seus pacientes, por exemplo, em caso de irregularidades como alta temperatura que poderia significar um caso de febre.

O trabalho de Kini difere-se deste trabalho no quesito de utilização de rede local, pois este trabalho tem como objetivo integrar a computação em nuvem para disponibilizar os dados por meio de internet, assim permitindo a visualização dos dados remotamente.

No trabalho de Ghanavati *et. al.* (2016), o monitoramento de dados fisiológicos em tempo real resulta em grandes quantidades de informações, assim exigindo uma alternativa para o armazenamento e análise desses dados. Com isso, os autores propuseram uma estrutura baseada em nuvem denominada WBAN (*wireless body area networks*) para a solução deste problema.

De acordo com os autores, é necessário à integração de WBANs com a tecnologia em Nuvem para o gerenciamento e armazenamento dos dados de forma eficaz, tornando-se assim importante a utilização de tecnologia móvel e da computação em nuvem como prestação de serviços aos pacientes.

A principal diferença do trabalho de Ghanavati para este estudo, é que o autor em sua pesquisa utilizou dados fisiológicos de pacientes enfermos, o mesmo propõe oferecer os dados coletados pelo sistema como um serviço ao paciente. Neste presente estudo foram coletados dados fisiológicos de praticantes de atividades físicas por meio de um sistema vestível (*wearables*) e que posteriormente enviou estes dados para um servidor em nuvem com a finalidade de auxiliar profissionais de educação física nas recomendações de melhores práticas de atividades físicas.

A Tabela 1 apresenta um comparativo de funcionalidades e tecnologias de redes utilizadas nos trabalhos relacionados e neste trabalho.

Tabela1. Comparação entre os trabalhos relacionados e a proposta deste trabalho

Trabalhos	Tecnologia de Rede	Armazenamento de Dados	Acesso Remoto aos Dados
Watthanawisuth <i>et al.</i> (2010)	XBee	Não	Não
Zulkifli (2012)	XBee	Sim	Não
Kini (2015)	Redes Sub-1 Ghz	Não	Não
Ghanavati (2016)	WBAN- Tecnologia em Nuvem	Sim	Sim
Este Trabalho	WBAN- Tecnologia em Nuvem	Sim	Sim

3. Metodologia Adotada

Esta sessão apresenta a metodologia utilizada para o desenvolvimento deste trabalho que é composta de três etapas, sendo elas:

Etapa 1: Definição de tecnologias a serem utilizadas para o desenvolvimento deste trabalho, como por exemplo definição de sensores, sistema embarcado e desenvolvimento de arquitetura de comunicação;

Etapa 2: Construção do protótipo denominado I-Jack, desenvolvimento de software para a coleta de dados e montagem da rede de comunicação do sistema;

Etapa 3: Teste do protótipo em ambiente de laboratório.

4. Arquitetura

A arquitetura desenvolvida para este trabalho é apresentada na Figura 1 e tem como objetivo descrever o funcionamento do sistema em geral.

O sistema vestível, denominado I-Jack, provê o monitoramento de dados fisiológicos do usuário praticante de atividades físicas. Com isso, os profissionais da área de educação física responsáveis pelo praticante, poderão monitorar os dados fisiológicos de seus alunos no decorrer das atividades de forma remota.

Esta arquitetura foi projetada em três módulos, um módulo Ambiente Vestível, com sensores e dispositivos embarcados em uma jaqueta inteligente, um módulo de Processamento de Dados, onde está presente o servidor, denominado concentrador e armazenamento de dados na nuvem e o terceiro módulo, os Sistemas Computacionais com aplicativos disponíveis para *tablets*, *smartphones* e computadores.

O primeiro módulo é composto por sensores embarcados na vestimenta. Esses sensores são responsáveis por coletar os dados fisiológicos do praticante de atividade física, e estão conectados ao módulo concentrador que enviam os dados para o servidor.

No módulo concentrador, um protótipo que estará preso na cintura do indivíduo, composto por um minicomputador Raspberry Pi 3 alimentado por bateria, é o responsável por identificar e processar os dados vindos dos sensores, armazená-los em uma base de dados e posteriormente enviá-los para o servidor na nuvem.

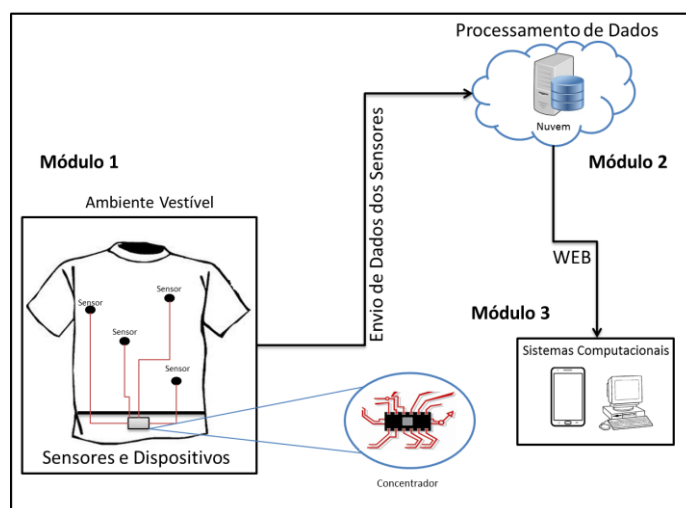


Figura 1. Arquitetura do Sistema I-Jack.

No terceiro módulo, são usados sistemas computacionais embarcados em *smartphone*, *tablets* e computadores, que serão utilizados para a visualização dos dados fisiológicos disponibilizados pelo servidor na nuvem. É nessa fase que os usuários profissionais da área de educação físicas terão acesso aos dados gerados pelos sensores fisiológicos, a fim de fazer boas recomendações de atividades para que o usuário faça melhor aproveitamento dos exercícios.

5. Resultados

Foi desenvolvido um protótipo de sistema vestível para coletar dados fisiológicos do usuário. Este protótipo é composto por dois sensores de temperatura do modelo MLX-90614, que possuem uma leitura não invasiva por meio de led infravermelho, permitindo realizar a leitura de temperatura corporal apenas apontando o sensor para a pele do indivíduo.

Outro sensor que compõe o sistema é o sensor de batimentos cardíacos do modelo Pulse Sensor, responsável por coletar os dados de batimentos cardíacos. Seguindo referência do próprio fabricante, escolhemos realizar a leitura dos batimentos cardíacos através do lóbulo da orelha, por possuir muitos vasos sanguíneos.

Além desses dois sensores, foi adicionado ao sistema, um dispositivo acelerômetro que está localizado na região do peito, com o objetivo de coletar informações do posicionamento corporal durante a prática da atividade física.

Pensando em aprimoramentos futuros, pode-se utilizar um dispositivo como esse para gerar uma base de dados com posições do corpo e aplicar tecnologias como aprendizagem de máquina e reconhecimento de padrões na base de dados para identificar, por exemplo, se o indivíduo está ereto ou deitado ou suas posturas durante determinados exercícios e assim obter mais informações para que o profissional de educação física possa realizar uma melhor avaliação de seus alunos e posteriormente recomendar correções.

Estes sensores e o dispositivo acelerômetro estão conectados ao módulo concentrador, que possui um servidor desenvolvido em um minicomputador Raspberry Pi 3, responsável por receber os dados coletados, processá-los e armazená-los em base de dados e em seguida, por meio de internet, enviar esse dados para um servidor na nuvem.

No servidor na nuvem, os dados são recebidos, armazenados e disponibilizados para visualização no terceiro módulo, que são os sistemas computacionais, compostos por computadores, *notebooks*, *tablets* e *smartphones*. A Figura 2 apresenta o protótipo de *hardware* e *software* desenvolvidos.

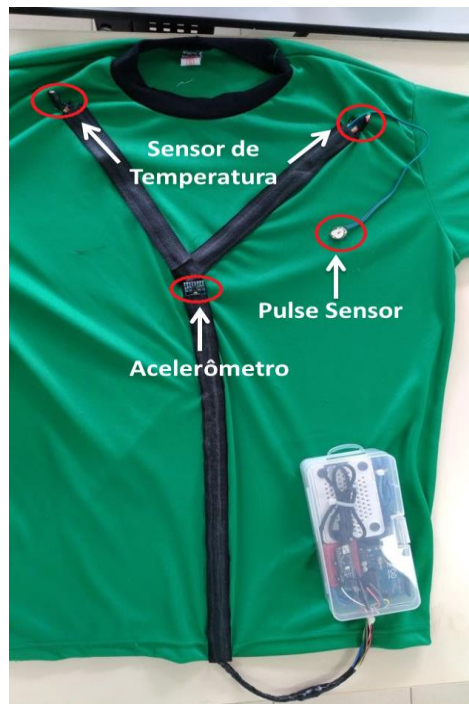


Figura 2. Protótipo do projeto.

Foram coletados dados através dos sensores conectados à vestimenta como: temperatura corporal, batimentos cardíacos e movimentos, para a realização dos testes iniciais foram coletados dados de uma mesma pessoa em atividade física, após a obtenção dos dados fisiológicos, o dispositivo concentrador realizou o envio desses dados para a nuvem. Foi utilizada a plataforma *ThingSpeak* para armazenar e plotar gráficos coletados dos sensores. A Figura 3 apresenta os dados do sensor de temperatura do ambiente de testes variando entre 16 e 26 graus.

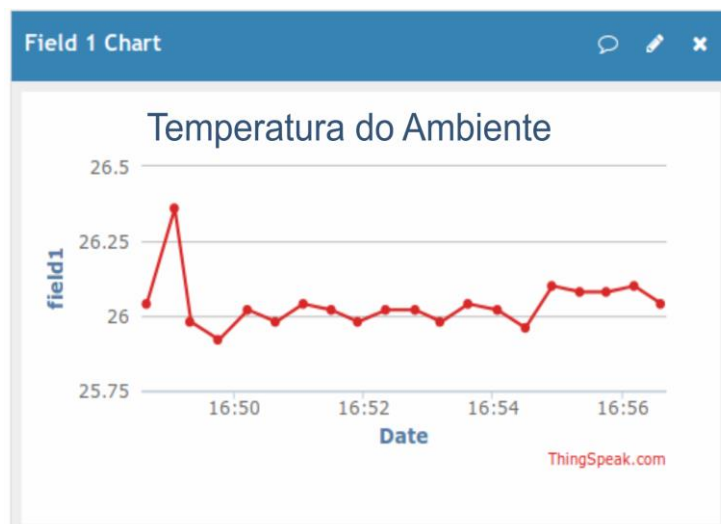


Figura 3. Apresentação dos dados coletados de temperatura.

Após coletados os dados do I-Jack, foi realizada uma análise estatística da média dos resultados, que visa descrever e entender os dados fisiológicos do sistema. Foram utilizados os dados da temperatura corporal de um termômetro digital convencional com o sensor de temperatura da vestimenta, e do sensor de batimentos

cardíacos da vestimenta com um dispositivo *smartwatch* apresentado no trabalho de (Cruz, L. F.*et.Al.*,2016).

A Figura 4 mostra o protótipo do I-Jack usando o Arduino, sensores e uma *Raspberry* fixados na cintura do usuário por meio de uma bolsa. Nesse protótipo inicial, foi necessário acondicionar os dispositivos em uma bolsa maior, entretanto, com a evolução da tecnologia, pretende-se embarcar o dispositivo coletor em um hardware menor.

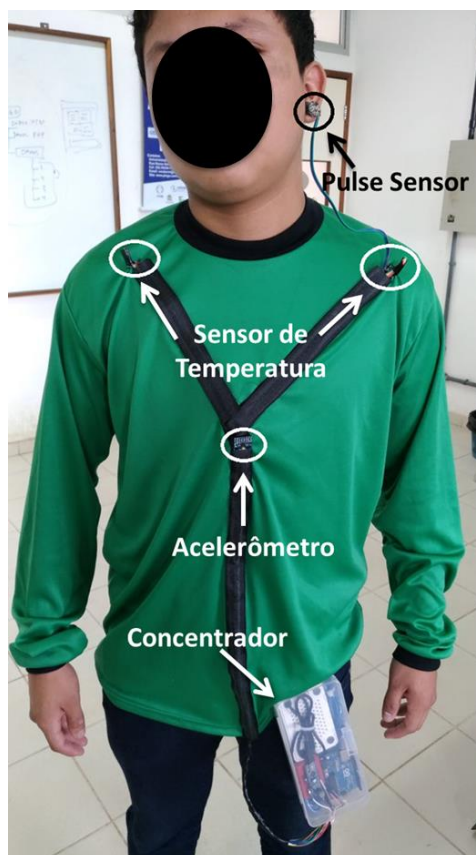


Figura 4. Protótipo I-Jack e placas micro controladoras.

A Figura 5 mostra o gráfico comparativo dos dados obtidos dos sensores do sistema vestível e dos dispositivos convencionais, nessa comparação, o protótipo do sistema vestível obteve resultados satisfatórios podendo ser observados pequenas variações sem muita significância nos dados obtidos.

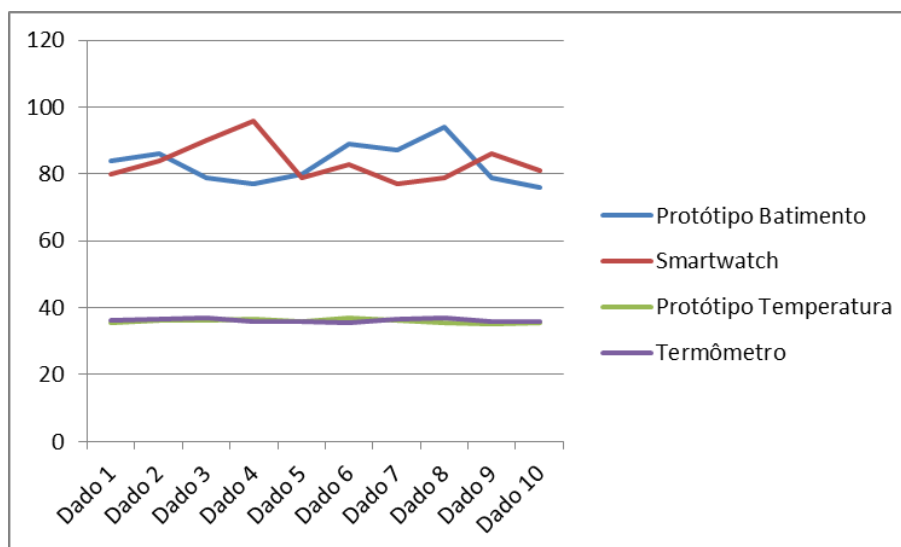


Figura 5. Gráfico comparativo do protótipo com dispositivos convencionais

A Tabela 2 mostra os dados comparativos dos valores obtidos dos sensores da vestimenta e pelos dispositivos convencionais encontrados no mercado. O sistema vestível proposto obteve uma taxa de acerto inicial de 70 %, quando comparado com o *smartwatch* indicando que a cada dez batimentos sete são iguais ao do dispositivo.

Com relação à comparação entre o sensor de temperatura convencional, encontrado no mercado e o utilizado neste trabalho, não houve variações significativas para serem mencionadas.

Tabela 2. Dados estatísticos descritivos entre o protótipo deste trabalho e dispositivos convencionais

Sistema	Temperatura	Batimentos Cardíacos
Protótipo do Trabalho	36,1	83,1
Dispositivos Convencionais	36,2	83,5

5. Considerações Finais

Neste projeto foi apresentado um sistema vestível de baixo custo para o monitoramento de dados fisiológicos de praticantes de atividades físicas, utilizando sensores, microcontroladores e rede de sensores.

O sistema foi desenvolvido com base numa arquitetura cliente servidor, com acesso a rede local de sensores da vestimenta e a internet, o sistema desenvolvido é um protótipo para o desenvolvimento de um sistema maior, com mais sensores integrados, acesso remoto aos dados fisiológicos e com recomendações de boas práticas de atividades físicas por parte da equipe de educação físicas responsáveis pelo usuário do sistema vestível que está sendo desenvolvido. O próximo passo do desenvolvimento do sistema vestível será a construção de um aplicativo mobile para o acesso dos dados remotamente via internet com recomendação de melhores práticas.

Agradecemos ao Instituto de Ciências Exatas e Tecnologia da Universidade Federal do Amazonas, ICET-UFAM, laboratório 303 Pixgo, pelo apoio no desenvolvimento deste projeto através do Parque Tecnológico (PCTIS).

6. Referências Bibliográficas

- Coutinho, E. F., F. R. C. Souza, D. G. Gomes, e J. N. Souza. “Elasticidade em Computação na Nuvem: Uma Abordagem Sistemática.” *31º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*. 2013.
- Cruz, L. F., Bernardon, C., Santos, E. V. M. S., Simões, W. C. S. S., Silva, V. J., 2016 “Um Sistema para monitoramento de sinais fisiológicos baseado em hardware de baixo custo com acesso via WEB” *Simpósio Brasileiro de Redes de Computadores – Workshop de Comunicação de Sistemas Embarcados Críticos SBRC – WoCCES*.
- Ghanavati S., Abawajy J., Izadi D. “An alternative sensor Cloud architecture for vital signs monitoring.” *Neural Networks (IJCNN), 2016 International Joint Conference on IEEE*. 2016. 39.
- IBGE. *IBGE diz que 46% dos brasileiros não fazem atividade física suficiente*. 2014. <http://www.redebrasilatual.com.br/saude/2014/12/ibge-diz-que-46-dos-brasileiros-nao-fazem-ativi> (acesso em 03 de Janeiro de 2017).
- Kini, V., C, Patil, S, Bahadkar, A, Sreedharam, e A. Kshirsagar. “Low Power Wireless Health Monitoring System.” *Advances in Computing, Communications and Informatics (ICACCI)*. International Conference on IEEE, 2015. 980 – 986.
- Ministério da Saúde. Os benefícios da atividade física para hipertensos. 2015. <http://promocaodasaude.saude.gov.br/promocaodasaude/assuntos/incentivo-a-atividade-fisica/noticias/os-beneficios-da-atividade-fis> (acesso em 03 de Janeiro de 2017).
- Pereira, M. G. “Revisões sistemáticas da literatura : passos para sua elaboração.” 183–184. 2014.
- Rolf HW, Romana W. *Internet of Things Legal Perspectives*. Berlin:Springer.2010; 1(1):1-22.
- Watthanawisuth, N.; Lomas, T.; Wisitsoraat, A.; Tuantranont, A. (2010). “Wireless Wearable Pulse Oximeter for Health Monitoring using ZigBee Wireless Sensor Network”. *Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON)*, 2010 International Conference on, IEEE, pages 575 – 579.
- Zulkifli, N.S.A., F.K.C, Harun, e N.S. Azahar. “Centralized Heart Rate Monitoring Telemetry System Using ZigBee Wireless Sensor Network” *Biomedical and Health Informatics (BHI)*.” *IEEE-EMBS International Conference on IEEE*. 2012. 265 – 268.

Implementação de um Comutador KeyFlow 10 Gbps para Redes Definidas por Software

Matheus de Souza¹, Natan de Oliveira¹, Antônio M. Alberti¹ e Moisés R. N. Ribeiro²

¹ICT Lab., Instituto Nacional de Telecomunicações (INATEL),
Caixa Postal 05, Santa Rita do Sapucaí, Brasil.

²Laboratório de Telecomunicações (LABTEL), Departamento de Eng. Elétrica,
Universidade Federal do Espírito Santo, Av. Fernando Ferrari, 514,
Goiabeiras, Vitória, Brasil.

{matheus.souza,alberti}@inatel.br, natanelias@get.inatel.br
moises@ele.ufes.br

Abstract. *This article aims at presenting the design and implementation, using the NetFPGA-10G platform, of a stateless switch based on global labels (KeyFlow) for Software-Defined Networking (SDN) capable of operating at 10 Gbps. The validation of the prototype is also discussed, from the qualitative point of view, for verifying the correct forwarding of packages generated and received externally by a network acceleration card (NAC).*

Resumo. *Este artigo tem como objetivo apresentar a concepção e a implementação, na plataforma NetFPGA-10G, de um comutador baseado em rótulos globais e sem estados (KeyFlow) para Redes Definidas por Software (SDN) capaz de operar a 10 Gbps. A validação do protótipo também é discutida, do ponto de vista qualitativo, verificando o correto encaminhamento de pacotes gerados e recebidos externamente por uma placa aceleradora de rede.*

1. Introdução

Quando se fala em redes de computadores no contexto atual, é possível observar que uma boa parte das aplicações é limitada pela infraestrutura da rede. Essa infraestrutura é constituída por equipamentos proprietários e de alto custo, cujas arquiteturas possuem circuitos integrados dedicados ao processamento dos pacotes, dificultando, assim, a inovação originada da oferta de novos serviços que dependem de inovações na infraestrutura das redes subjacentes.

A falta de abertura e flexibilidade nos equipamentos de rede como *switches* e roteadores, os quais possuem uma camada de software fechada, traz como consequência um ambiente de rede que não consegue dar o suporte ideal para determinadas aplicações devido ao fato dos protocolos atuais terem sido desenvolvidos para atender principalmente demandas convencionais das redes locais, tornando assim bastante problemática a evolução das arquiteturas e a inovação originada da oferta de novos serviços [Rothenberg et al. 2010].

Nesse contexto, surgiu o conceito de Redes Definidas por Software (SDN - *Software Defined Networking*), que propõe a separação do plano de dados do plano de controle. O plano de controle é responsável por definir as rotas dos fluxos de pacotes dentro da

rede. O plano de dados tem a função de realizar o encaminhamento do fluxo de pacotes de acordo com regras definidas pelo plano de controle. Ao realizar essa separação, se torna necessário alterar somente a camada de controle para modificar as regras de definição do fluxo de pacotes, pois os equipamentos de rede podem compartilhar o mesmo plano de controle. Com isso é possível programar uma determinada rede de acordo com uma aplicação desejada. Além disso, com a introdução da SDN torna-se possível realizar testes e medições em redes em operação, fazendo com que os recursos disponíveis sejam melhor aproveitados [McKeown et al.].

O padrão mais difundido para SDN é o OpenFlow [Stallings 2013][Lara et al. 2014][Braun and Menth 2014]. Esse protocolo estabelece a interface de comunicação entre o controlador e os comutadores controlados, bem como o formato da tabela de encaminhamento. Apesar de bem difundido, o OpenFlow pode enfrentar problemas de escalabilidade devido à necessidade de manutenção de estado dos fluxos ativos nas tabelas de encaminhamento dos comutadores e a necessidade de comunicação com o controlador a cada novo fluxo. Uma proposta de complemento ao OpenFlow é o KeyFlow, cuja contribuição é o encaminhamento de pacotes sem a necessidade de consulta à tabelas de roteamento. O uso do KeyFlow apresenta uma melhora no desempenho da SDN em termos de latência, variabilidade na entrega de quadros e tempo de reconfiguração da rede [Martinello et al. 2014], além de redução do consumo de energia [Cercós et al. 2014]

Este artigo apresenta pela primeira vez a implementação de um comutador de rede utilizando o protocolo KeyFlow para a taxa de 10 Gbps, bem como testes de validação do hardware desenvolvido. A fim de contextualização, na Seção 2 é apresentada uma revisão de SDN e OpenFlow. Na Seção 3 é apresentado o modelo KeyFlow de SDN. Na Seção 4 são apresentados detalhes técnicos da implementação do comutador. Na Seção 5 é descrito o cenário de testes de validação. E, finalmente, na Seção 6 são apresentadas as conclusões do trabalho.

2. SDN - *Software Defined Networking*

O principal objetivo da arquitetura de uma SDN é realizar a separação física do plano de controle e do plano de encaminhamento de dados. O modelo atual trabalha com esses dois planos acoplados no próprio equipamento de rede. Ao realizar essa separação, os equipamentos terão como função única o encaminhamento de pacotes e a rede pode ser programada de acordo com uma aplicação específica, eliminando qualquer tipo de restrição relacionada com os protocolos de equipamentos proprietários [McKeown et al.].

A Figura 1 mostra uma visão geral da arquitetura de uma Rede Definida por Software. Nesse caso, os equipamentos de rede são vistos como uma camada única pelo controlador, que tem como função a tomada de decisão de encaminhamento de acordo com as informações contidas no cabeçalho de cada pacote. A camada de Aplicação se comunica com a camada de controle através de API's (*Application Programming Interfaces*) e enxerga a rede como um único *switch* lógico. No caso da comunicação entre o plano de controle e o plano de dados se torna necessário existir um protocolo padrão que possa levar controles entre os planos. Atualmente o protocolo mais utilizado para essa comunicação é o OpenFlow.

A infraestrutura de uma SDN tem como foco principal o encaminhamento dos

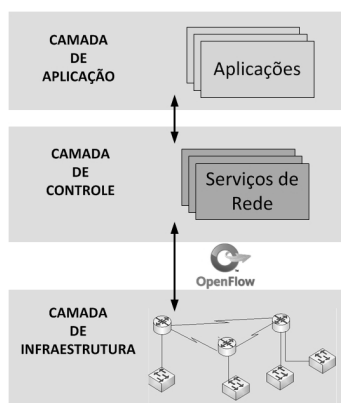


Figura 1. Arquitetura de uma Rede Definida por Software.

pacotes pelo plano de dados e o recebimento desses de acordo com as rotas dos fluxos de pacotes definidas pelo plano de controle. Sendo assim, o custo dos equipamentos utilizados no desenvolvimento pode ser reduzido se comparado aos equipamentos com software proprietários convencionais, tendo em vista que a estrutura do hardware empregado é bem mais simples.

2.1. OpenFlow

O protocolo OpenFlow foi desenvolvido na universidade de Stanford no ano de 2008 com o principal objetivo de atender a demanda de novas arquiteturas e protocolos de rede, que entram no contexto da necessidade de abertura e flexibilidade dos equipamentos comerciais já existentes [Open Networking Foundation 2012].

O OpenFlow estabelece a interface de comunicação entre o controlador e os comutadores controlados, bem como o formato da tabela de fluxos. Ao receber um pacote, o comutador realiza uma consulta em sua tabela que contém as regras de encaminhamento. Caso não haja nenhuma regra estabelecida para um determinado pacote, seu cabeçalho é enviado ao controlador para processamento e definição de uma nova regra. Uma vez que os controladores são responsáveis por definir as regras de fluxo, a complexidade dos comutadores é reduzida. O OpenFlow é o padrão mais difundido de SDN, no entanto, uma de suas principais limitações está relacionada com a escalabilidade da rede, tendo em vista que a tabela de fluxos tende a aumentar de acordo com o crescimento da rede e do número de fluxos, além da necessidade de manutenção de estado dos fluxos ativos nas tabelas de encaminhamento dos comutadores e da necessidade de comunicação com o controlador a cada nova regra desconhecida. Com isso, se torna necessário buscar alternativas que evitem a abordagem de regras em tabelas de fluxos [Martinello et al. 2014].

3. KeyFlow

O KeyFlow pode ser definido como um método de encaminhamento de pacotes dentro de uma SDN que substitui a tabela de regras pela operação de módulo (resto da divisão entre dois números inteiros) para determinar a porta de encaminhamento de um pacote. Este método traz como benefícios a melhora no desempenho da rede em termos de latência, variabilidade na entrega de quadros e tempo de reconfiguração [Martinello et al. 2014].

O método de encaminhamento no qual se baseia o KeyFlow consiste na criação de topologias *overlay* em redes OpenFlow através da utilização do Esquema de Informação

de Chave (KIS - *Key Information System*), proposto originalmente para redes ópticas [Wessing et al. 2002]. As chaves locais funcionam como identificadores de cada comutador na rede. Para que as chaves criadas sejam válidas, é necessário que, em cada caminho da rede, elas sejam definidas por números primos entre si.

Cada pacote que será encaminhado dentro da rede possui uma informação própria contida no cabeçalho chamada de rótulo. A relação entre as chaves locais e os rótulos se dá pela operação de módulo. Nesse caso, é realizada a divisão do rótulo do pacote pela chave local de cada comutador. Essa operação tem como principal objetivo a determinação da porta de saída do fluxo de pacotes que chegam a cada comutador da rede.

Os rótulos são gerados a partir do Teorema Chinês do Resto (TCR) [Kak 1985]. Para cada caminho ou circuito virtual, são definidos dois vetores. O primeiro vetor é determinado a partir das chaves que determinam o caminho desejado. O segundo vetor é determinado através do número da porta de saída de cada um dos comutadores da rede.

3.1. Teorema Chinês do Resto

A função do TCR é encontrar um rótulo para o pacote dada a informação do caminho que ele deve seguir na rede, que é definido pelo próprio controlador. Após o rótulo ser calculado, ele é enviado como resposta ao comutador de borda, que irá incorporá-lo ao cabeçalho do pacote. Com este rótulo, é possível fazer o processo inverso, ou seja, obter o caminho a ser seguido pelo pacote em cada comutador da rede.

Suponha que o caminho que o pacote deva percorrer contenha N nós. Com isso, definimos o vetor $J = (J_1, J_2, J_3, \dots, J_n)$ e $K = (K_1, K_2, K_3, \dots, K_n)$ com os inteiros que representam a chave de cada comutador no caminho desejado e a identificação das portas de saída dos comutadores correspondentes, respectivamente.

Queremos obter um inteiro R tal que permita recuperar o vetor K . O sistema admite solução única somente se as chaves dos comutadores forem primas entre si, ou seja, $mdc(J_i, J_j) = 1$, para $i \neq j$. Nesse contexto, pode-se definir o seguinte sistema de congruência:

$$\begin{cases} R \equiv K_1 \pmod{J_1} \\ R \equiv K_2 \pmod{J_2} \\ \dots \\ R \equiv K_n \pmod{J_n}. \end{cases}$$

Um algoritmo para obter o rótulo R através do TCR pode ser encontrado em [Cercós et al. 2014].

3.2. Rede KeyFlow

A Figura 2 mostra um exemplo de uma arquitetura de rede KeyFlow integrada a uma estrutura SDN. Neste exemplo, um pacote que sai do nó 43 com destino ao nó 31 atravessando os nós intermediários definidos por $J_i = (7, 10, 13)$ tem as portas de saída definidas por $K_i = (1, 4, 4)$. O caminho em estudo está definido pelas setas pontilhadas.

Para que o valor do rótulo do pacote seja definido, uma solicitação é realizada ao controlador, que o calcula com base no caminho definido até o destino. Logo após,

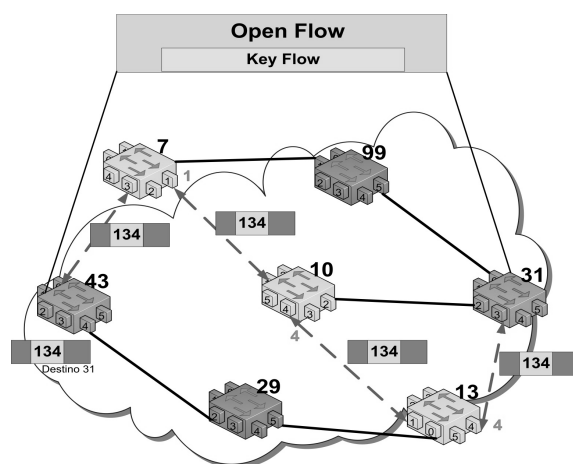


Figura 2. Arquitetura de uma rede KeyFlow integrada a uma SDN.

o controlador envia essa informação aos nós de borda, que operam como comutadores OpenFlow pois devem modificar o cabeçalho do pacote inserindo o rótulo recebido, por exemplo, no campo de Vlan ID. Os nós centrais são comutadores KeyFlow que realizam o encaminhamento dos pacotes sem consulta em tabela de fluxos como no exemplo da Figura 2 no qual o rótulo do pacote é definido com o valor 134. Após a chegada ao nó 7, é realizada a operação de módulo, sendo que $134 \bmod 7 = 1$ determina que a porta 1 seja a de saída. A operação se repete nos nós 10 e 13: $134 \bmod 10 = 4$, $134 \bmod 13 = 4$, determinando assim as respectivas portas de saída dos nós.

4. Implementação do Protótipo

A seguir serão abordados detalhes técnicos do desenvolvimento do protótipo do comutador KeyFlow sobre a plataforma NetFPGA-10G.

4.1. NetFPGA-10G

A NetFPGA-10G é uma das plataformas para prototipagem de hardware e software livre desenvolvidas pelo grupo NetFPGA das universidades de Stanford e Cambridge. O projeto foi criado para fornecer plataformas de pesquisa e ensino que possibilitam o desenvolvimento em hardwares de aplicações em rede com alto desempenho. As placas acompanham um material de apoio composto de projetos de referência documentados que servem como ponto de partida para novos projetos [Gibb et al. 2008].

Os projetos de referência que acompanham a placa NetFPGA-10G seguem a estrutura lógica padrão (“*standard pipeline*”) mostrada na Figura 3. Cada um dos blocos mostrados é composto por blocos lógicos funcionais (*IPcores*) agrupados em estruturas modulares com interfaces bem definidas. Ao desenvolver um novo projeto, é comum inserir um novo bloco lógico na estrutura padrão ou modificar um dos existentes. O presente projeto teve como referência o comutador OpenFlow desenvolvido por Tatsuya Yabe [Yabe]. Modificações foram realizadas para retirar os módulos de busca em tabela e modificações de cabeçalhos próprios da lógica OpenFlow, além de acrescentar o módulo de encaminhamento desenvolvido segundo a lógica KeyFlow, chegando na estrutura ilustrada na Figura 4.

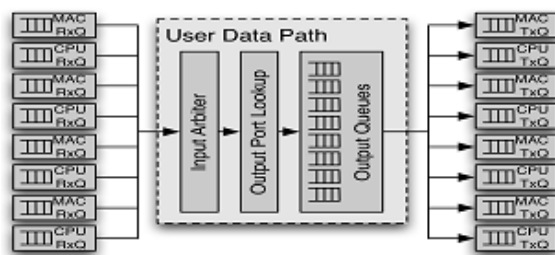


Figura 3. Standard pipeline para NetFPGA-10G.

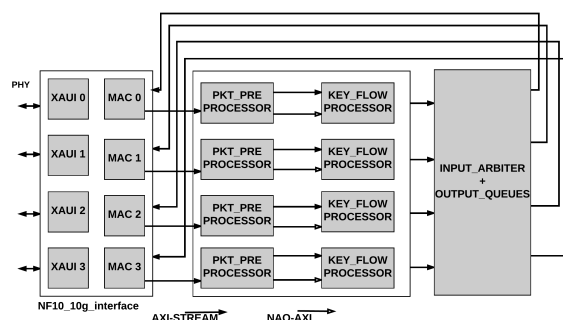


Figura 4. Diagrama em blocos da estrutura implementada.

4.2. Interface AXI

Para manter a interoperabilidade e o reuso de *IPcores*, os desenvolvedores que utilizam as placas NetFPGA são encorajados a adotarem as mesmas interfaces nos blocos criados. As interfaces adotadas para a comunicação entre os blocos da NetFPGA-10G são versões refinadas pela Xilinx dos protocolos AXI4-Lite e AXI4-Stream (AXI - *Advanced eXtensible Interface*), desenvolvidos pela ARM, como parte da família AMBA (*Advanced Microcontroller Bus Architecture*), para interconexão de blocos em *SoC's*. O protocolo AXI4-Lite é utilizado para comunicações com requisitos de memória mapeada, simples e de baixa vazão; e o protocolo AXI4-Stream, para fluxo de dados de alta vazão. A Tabela 1 mostra o comprimento e a função dos campos do barramento AXI4-Stream que é utilizado nos blocos descritos adiante.

Tabela 1. Campos do barramento AXI4-Stream.

Campo	Comprimento em bits	Função
ACLK	1	Clock global
ARESETN	1	Reset global
TVALID	1	Utilizado pelo <i>master</i> para sinalizar início de transferência
TDATA	64	Dados transferidos
TREADY	1	Utilizado pelo <i>slave</i> para sinalizar que aceita iniciar transferência.
TSTRB	8	Sinaliza dados inválidos.
TLAST	1	Sinaliza fim de transferência.
TUSER	128	Metadados da informação transferida.

4.3. Blocos Funcionais

A seguir são apresentados os blocos descritos em Verilog que constituem a implementação realizada.

4.3.1. NF10_10g_interface

A função desse bloco é fazer a interface com o *transceiver* e executar a lógica da sub-camada MAC do padrão Ethernet 10Gbps. Esse bloco é parte do material de referência e é composto pelos *IPcores* da Xilinx, Xaui (Xilinx 10 Gigabit Attachment Unit Interface) e 10GMAC (10 Gigabit Ethernet Media Access Controller) e por um adaptador para o barramento AXI4-Stream. Os sinais provenientes do meio físico são capturados e decodificados pelo *transceiver*. O *transceiver* e o FPGA se comunicam através de um barramento XAUI. No interior do FPGA, os dados que chegam pela interface XAUI são convertidos para o padrão XGMII pelo *IPcore* XAUI e enviados para o *IPcore* 10GMAC no qual ocorre os tratamentos da sub camada de controle de acesso ao meio. Por fim, os dados são convertidos para o formato AXI-Stream padrão. Os pacotes enviados seguem o mesmo fluxo, mas da direção oposta. As interfaces XAUI e XGMII são definidas no padrão Ethernet 10Gbps (IEEE 802.3-2012).

4.3.2. Pkt_preprocessor

Esse bloco é originário do projeto de referência e foi desenvolvido para analisar os cabeçalhos dos pacotes que chegam e organizar os campos extraídos em um formato próprio para a busca na tabela OpenFlow. No presente projeto, é utilizado desse bloco somente a capacidade de analisar cabeçalhos IEEE 802.1Q. No contexto do KeyFlow, o espaço do cabeçalho destinado ao campo “Vlan ID” no padrão IEEE 802.1Q, é utilizado para armazenar o KeyFlow ID (chave que determina a rota do pacote na rede). No comutador KeyFlow, a função do *pkt_preprocessor* passou a ser o elemento que extrai o KeyFlow ID de cada pacote.

4.3.3. KeyFlow_processor

O *KeyFlow_processor* é o bloco principal do projeto e tem como funcionalidade executar a operação de resto da divisão inteira entre o valor do KeyFlow ID (rótulo do pacote) e a chave local do comutador e daí determinar a porta de encaminhamento para cada pacote. Na Figura 5 é possível observar a constituição interna do bloco. Durante o desenvolvimento, foi utilizado o simulador ISim fornecido pela Xilinx. O ISim permite analisar o comportamento dos sinais ao longo do tempo, como mostrado na Figura 6 que será utilizada na explicação do fluxo de operação do bloco.

Os pacotes que chegam são armazenados na fila “*pkt_fifo*” e os correspondentes metadados (campo TUSER do barramento AXI4-Stream) são armazenados na fila “*mdata_fifo_in*”. Assim que o bloco *pkt_preprocessor* termina de extrair o KeyFlow ID, o sinal “*valid*” vai a nível lógico “1” (ponto “A” da Figura 6). Quando isto ocorre, um valor válido é colhido no registro “*kf_id*” para a divisão. O sinal “*valid*” é atrasado, em

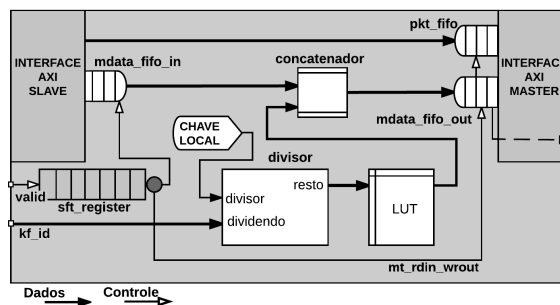


Figura 5. Constituição do bloco KeyFlow Processor.

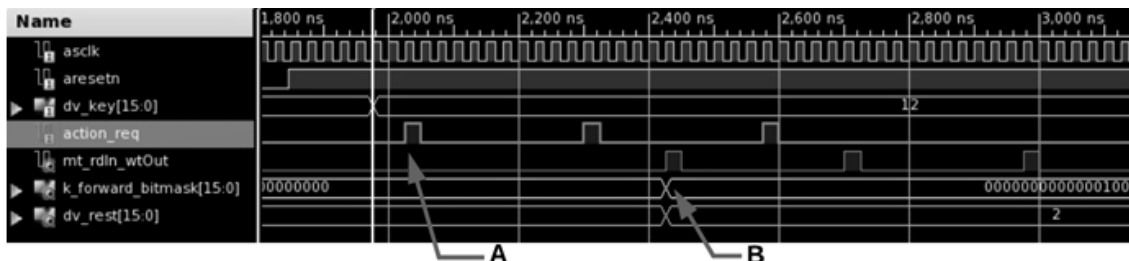


Figura 6. Curvas obtidas por simulação.

“sft_register”, por dezesseis pulsos de *clock* devido à latência do divisor. A divisão é executada por um *IPcore* fornecido pela Xilinx com capacidade de iniciar uma divisão por ciclo de *clock*. O divisor opera com paralelismo (*pipeline*) e sua latência (tempo entre o início e o fim de uma divisão) é de M ciclos de *clock* para divisões de restos inteiros, sendo M o comprimento em bits do valor do dividendo. Neste caso, $M = 16$. Para manter a vazão, o divisor fica constantemente habilitado, no entanto, o valor em sua saída só é válido quando a saída do “sft_register” está em nível lógico “1”. Assim que o resto da divisão é obtido, uma parte do metadado é alterada, nos blocos “LUT” e “concatenador”, para a máscara de bits correspondente à porta de saída do pacote (Ponto “B” da Figura 6). A saída do “sft_register” está diretamente ligada ao habilitador de leitura da fila “mdata_fifo_in” e ao habilitador de escrita da fila “mdata_fifo_out”, ou seja, depois de alterado, o metadado é transferido da fila de entrada para a fila de saída se tornando acessível ao bloco subsequente. O sinal que indica que a fila “mdata_fifo_out” não está vazia é ligado ao sinal “TVALID” da interface AXI4-Stream *Master*. Portanto, quando o processo descrito é concluído o próximo bloco é sinalizado e poderá ocorrer a transferência simultânea do pacote e seu metadado.

4.3.4. Input arbiter + output queues

Encapsula dois blocos da estrutura padrão proposta para a NetFPGA-10G. Esse bloco contém quatro interfaces de entrada (AXI4-Stream *Slave*) e quatro interfaces de saída (AXI4-Stream *Master*). Sua primeira função é determinar qual das interfaces de entrada será atendida. A escolha da interface a ser atendida é feita por um simples algoritmo de varredura circular (*round-robin*), ou seja, uma interface com pacotes na fila é atendida a cada vez. Outra função desse bloco é direcionar o fluxo de pacotes de cada interface para uma porta pré-determinada. O direcionamento da porta de saída é feito pela verificação



(a) Conexão da aceleradora com comutador KeyFlow.

```

15 0.000001504 f0:0d:f0:0d:f0:0d 64:d1:0d:53:0f:00
16 0.000002464 f0:0d:f0:0d:f0:0d 64:d1:0d:53:0f:00
17 0.000002504 f0:0d:f0:0d:f0:0d 64:d1:0d:53:0f:00
18 0.000001560 f0:0d:f0:0d:f0:0d 64:d1:0d:53:0f:00
19 0.000002496 f0:0d:f0:0d:f0:0d 64:d1:0d:53:0f:00
20 0.000002504 f0:0d:f0:0d:f0:0d 64:d1:0d:53:0f:00
21 0.000001576 f0:0d:f0:0d:f0:0d 64:d1:0d:53:0f:00
22 0.000002512 f0:0d:f0:0d:f0:0d 64:d1:0d:53:0f:00

```

```

Frame 8: 145 bytes on wire (1160 bits), 145 bytes captured (1160 bits) on interface 0
Ethernet II, Src: f0:0d:f0:0d:f0:0d, (f0:0d:f0:0d:f0:0d), Dest: 64:d1:0d:53:0f:00
802.1Q Virtual LAN, PVID: 0, CFI: 0, ID: 14
[Malformed Packet: LLC]

```

(b) Pacotes capturados na porta 4.

Figura 7. Testes e resultados.

dos bits 16 ao 23 do metadado de cada pacote. A porta, ou as portas, para as quais o pacote é direcionado segue a codificação *one-hot encoded* mostrada na Tabela 2. Cada bit em nível lógico “1” direciona o pacote para a porta correspondente. As portas MACx são portas físicas da placa e as portas CPUx são portas virtuais associadas à interface PCIE (não utilizada neste projeto).

Tabela 2. Máscara de bits para porta de saída.

Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
MAC0	CPU0	MAC1	CPU1	MAC2	CPU2	MAC3	CPU3

5. Validação

O protótipo foi submetido a testes de caráter qualitativo para verificar sua operabilidade. Durante os testes, foi utilizada uma aceleradora (Napatech modelo NT20E2-PTP) conectada ao comutador de modo a formar um circuito fechado, como mostrado na Figura 7(a). As funções da aceleradora foram de replicar um arquivo de captura injetando pacotes a 10 Gbps em uma das portas do comutador e capturar o fluxo de saída em outra porta. O arquivo replicado continha pacotes idênticos entre si exceto pelo valor do rótulo KeyFlow que podia conter os valores 11, 12, 13 ou 14. A Figura 7(b) mostra um dos registros obtidos nos testes. Nesse caso, ao conectar a porta de captura da aceleradora na porta 4 do comutador, constatou-se que somente os pacotes com o rótulo Keyflow igual a 14 eram encaminhados para aquela interface. Este era o resultado esperado tendo em vista que o valor da chave local do comutador no momento dos testes era 10 e que $14 \bmod 10 = 4$. Resultados equivalentes foram observados nas outras 3 portas do comutador, validando a implementação.

6. Conclusões

Este trabalho apresentou a primeira implementação de um comutador de rede utilizando o protocolo KeyFlow capaz de operar em 10 Gbps. Foram apresentados conceitos fundamentais envolvidos em uma rede definida por software, bem como sua expansão através da lógica de encaminhamento com rótulos globais e encaminhadores sem estados do KeyFlow. O hardware desenvolvido foi validado através de testes com fonte e destino externo à NetFPGA. A implementação aqui apresentada pode servir de referência para o desenvolvimento de comutadores com vazão maior e mais capacidade de comutação usando formas alternativas de encaminhamento.

7. Agradecimentos

Este trabalho foi parcialmente financiado pela Finep, com recursos do Funttel, contrato No 01.14.0231.00, sob o projeto Centro de Referência em Radiocomunicações (CRR) do Instituto Nacional de Telecomunicações – Inatel, Brasil. Este trabalho também conta com o suporte do projeto Horizon 2020 da União Européia para pesquisa, desenvolvimento tecnológico e demonstração sob no. 688941 (FUTEBOL), assim como do Ministério Brasileiro da Ciência, Tecnologia e Inovação (MCTI) por meio da RNP e do CTIC. Agradecemos também ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e Fundação de Amparo à Pesquisa e Inovação do Espírito Santo (FAPES).

Referências

- Braun, W. and Menth, M. (2014). Software-defined networking using openflow: Protocols, applications and architectural design choices. *Future Internet: Open Access Journal*, pages 302–336.
- Cercós, S. S., Oliveira, R., Vitoi, R., Martinello, M., Ribeiro, M., Fagertun, A. M., and Monroy, I. T. (2014). Tackling openflow power hog in core networks with keyflow. *Electronics Letters*, 50(24):1847–1849.
- Gibb, G., Lockwood, J. W., Naous, J., Hartke, P., and McKeown, N. (2008). Netfpga: An open platform for teaching how to build gigabit-rate network switches and routers. *IEEE Transactions on Education*, 51:364–369.
- Kak, S. (1985). Computational aspects of the Āryabhata. *Indian Journal Of History Of Science*, pages 62–71.
- Lara, A., Kolasani, A., and Ramamurthy, B. (2014). Network innovation using openflow: A survey. *IEEE Communications Surveys & Tutorials*, pages 493–512.
- Martinello, M., Ribeiro, M. R., and de Oliveira, R. E. Z. (2014). Keyflow: a prototype for evolving sdn toward core network fabrics. *IEEE Network*, 7:12–19.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. Openflow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38.
- Open Networking Foundation (2012). Software-defined networking: The new norm for networks. <https://www.opennetworking.org>.
- Rothenberg, C. E., Nascimento, M. R., Salvador, M. R., and Magalhães, M. F. (2010). Openflow e redes definidas por software: um novo paradigma de controle e inovação em redes de pacotes. *Cad. CPqD Tecnologia*, 7:1–6.
- Stallings, W. (2013). Software-defined networks and openflow. *The Internet Protocol Journal*, 38:2–14.
- Wessing, H., Christiansen, H., and Fjelde, T. (2002). Novel scheme for packet forwarding without header modifications in optical networks. *Journal of Lightwave Technology*, 20:1277–1283.
- Yabe, T. Openflow implementation on netfpga-10g, design document. <https://docs.google.com/document/d/1ZwHXQZocKwQls6Ted8VZO8h9MjBtu9WxV2fAY44eOgE/edit>.

**V Workshop de Comunicação em Sistemas
Embarcados Críticos (WoCCES)
SBRC 2017
Sessão Técnica 2**

Estimativa de Posição 3D de Obstáculos para VANTs

Rodolfo Barros Chiaramonte^{1,2}, Kalinka Regina Lucas Jaquie Castelo Branco¹

¹Institute of Mathematics and Computing Sciences (ICMC)
University of São Paulo (USP),
São Carlos, São Paulo, Brazil

²Computing and Information Systems Research Lab (COMPSI)
Eurípides University of Marília (UNIVEM)
Marília, São Paulo, Brazil

rodolfo@icmc.usp.br, kalinca@icmc.usp.br

Abstract. *Autonomous vehicles are important to execute different types of missions, reducing risks to humans and running missions in an efficient manner. In this context, unmanned aerial vehicles are increasingly used in surveillance, reconnaissance, rescue and other. One of the characteristics of these vehicles is to perform as missions in an autonomous manner, without the intervention of human operators. Hence, there must be ways to detect obstacles that can cause a collision and then carry out the necessary deviation. At the time of evasive maneuver it is important to know the position of the obstacle in order to decide the best deviation approach. This paper proposes an extension of a method to estimate obstacle positions from images of a single camera and analyzes the results obtained through simulations.*

Resumo. *Veículos autônomos são importantes para executar diferentes tipos de missões, reduzindo os riscos para os seres humanos e executando missões de forma eficiente. Neste contexto, veículos aéreos não tripulados são cada vez mais utilizados em vigilância, reconhecimento, salvamento e outros. Uma das características destes veículos é realizar missões de forma autônoma, sem a intervenção de operadores humanos. Assim, deve haver maneiras de detectar os obstáculos que podem causar uma colisão e, em seguida, realizar o desvio necessário. No momento da manobra evasiva é importante saber a posição do obstáculo para decidir a melhor abordagem de desvio. Este artigo propõe uma extensão de um método para estimar posições de obstáculos a partir de imagens de uma única câmera e analisa os resultados obtidos através de simulações.*

1. Introdução

Veículos autônomos são importantes para a execução de missões dos mais variados tipos, reduzindo riscos aos seres humanos e executando as missões de uma maneira mais eficiente. Neste contexto existem os veículos aéreos não tripulados que são cada vez mais utilizados em missões de vigilância, reconhecimento, resgate, entre outros [Moon and Prasad 2011].

Uma das características destes veículos é realizar as missões de maneira autônoma, sem a intervenção de operadores humanos. Com isto, durante a realização

de uma missão, estes veículos estão vulneráveis a ameaças que podem ser obstáculos fixos (paredes, construções, vegetação, entre outros) ou móveis (outros veículos, pássaros, entre outros).

Para realizar o desvio do obstáculo é necessário conhecer a distância ou a posição do obstáculo, podem ser destacados vários trabalhos com este objetivo, utilizando de diferentes abordagens. O trabalho proposto por [Majumder et al. 2015] utiliza câmeras estéreo e explora o conceito de paralaxe para obter uma estimativa de distância. Outros trabalhos buscam conhecer a posição tridimensional do obstáculo; estes trabalhos geralmente utilizam métodos baseados em SLAM (*Simultaneous Location and Mapping*) [Esrafilian and Taghirad 2016, Mur-Artal et al. 2015].

Considerando que nem sempre é possível embarcar câmeras estéreo em aeronaves autônomas de pequeno porte, [Saha et al. 2014] propõem um método para cálculo da posição do obstáculo baseado em imagens obtidas em diferentes instantes de tempo, conhecendo a posição exata da aeronave em cada um dos instantes de tempo. Em seu trabalho, eles utilizam a IMU (*Inertial Measurement Unit*) da aeronave em conjunto com o filtro de Kalman estendido para obter a posição da aeronave. No entanto, os autores consideraram que sempre a aeronave está em nível, o que nem sempre é uma verdade.

Neste artigo é tratado o cálculo de posição de obstáculos estáticos baseados em imagens provenientes de uma única câmera. São considerados no cálculo da posição do obstáculo a posição corrente e a direção da câmera da aeronave. Desta forma, o trabalho apresentado por [Saha et al. 2014] foi estendido para considerar movimentação nos ângulos de *yaw* e *pitch* permitindo assim um avanço no estado da arte e uma contribuição importante para a área.

2. Método proposto

O artigo [Saha et al. 2014] descreve como encontrar a posição de um obstáculo utilizando como base a formula de projeção (1) baseada em proporcionalidade do triangulo, onde: P_I é um ponto na imagem na posição x_I, y_I ; f é a distância focal da câmera (em pixels) obtida através da calibração, $\overrightarrow{CP_o}$ é o um vetor formado entre a posição da câmera (C) e o obstáculo (P_o) e os vetores $\overrightarrow{n_x}$, $\overrightarrow{n_y}$ e $\overrightarrow{n_z}$ indicam a orientação da câmera.

$$P_I(x_I, y_I) = \frac{f}{\overrightarrow{n_x} \cdot \overrightarrow{CP_o}} (-\overrightarrow{n_y} \cdot \overrightarrow{CP_o}, \overrightarrow{n_z} \cdot \overrightarrow{CP_o}) \quad (1)$$

[Saha et al. 2014] assumem que o eixo z da câmera ($\overrightarrow{n_z}$) está alinhado com o eixo z do mundo real e considera a rotação apenas em torno do eixo z (*yaw*). Para considerar também a rotação em torno do eixo y (*pitch*) é necessário estender as equações (2) e (3) apresentadas por [Saha et al. 2014]. Para isto, será considerada a representação ilustrada na figura 1.

Para obter os vetores de orientação da câmera ($\overrightarrow{n_x}$, $\overrightarrow{n_y}$ e $\overrightarrow{n_z}$), considerando os ângulos *yaw* e *pitch*, foram utilizadas as matrizes de rotação $R_z(\textit{yaw})$ e $R_y(\textit{pitch})$ como seguem:

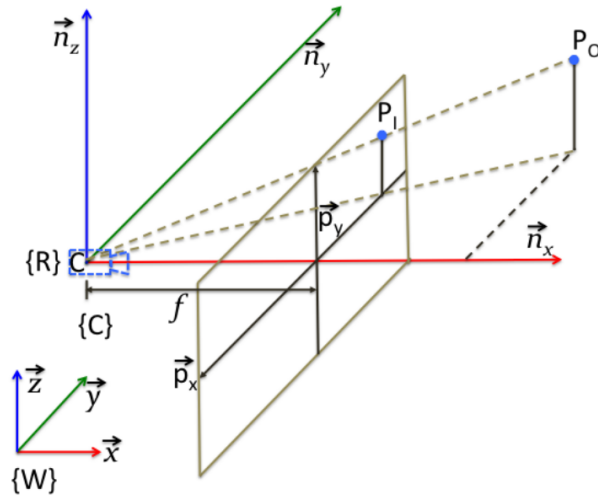


Figura 1. Representação considerada para o modelo matemático [Saha et al. 2014].

$$R_z(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}$$

$$R_z(\gamma) R_y(\beta) = \begin{bmatrix} \cos(\gamma)\cos(\beta) & -\sin(\gamma) & \cos(\gamma)\sin(\beta) \\ \sin(\gamma)\cos(\beta) & \cos(\gamma) & \sin(\gamma)\sin(\beta) \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}$$

$$\vec{n}_x = (\cos(\gamma)\cos(\beta), -\sin(\gamma), \cos(\gamma)\sin(\beta)) \quad (2)$$

$$\vec{n}_y = (\sin(\gamma)\cos(\beta), \cos(\gamma), \sin(\gamma)\sin(\beta)) \quad (3)$$

$$\vec{n}_z = (-\sin(\beta), 0, \cos(\beta)) \quad (4)$$

Onde $\gamma = \text{yaw}$ e $\beta = \text{pitch}$. Para simplificar, considera-se:

$$a = \cos(\text{yaw})$$

$$b = \sin(\text{yaw})$$

$$c = \cos(\text{pitch})$$

$$d = \sin(\text{pitch})$$

Considerando que um frame foi capturado no instante de tempo t , a partir de 1, obtem-se:

$$x_{I_t} = \frac{-b_t c_t (x_O - x_t) - a_t (y_O - y_t) - b_t d_t (z_O - z_t)}{a_t c_t (x_O - x_t) - b_t (y_O - y_t) + a_t d_t (z_O - z_t)} f_x \quad (5)$$

$$y_{I_t} = \frac{-d_t (x_O - x_t) + c_t (z_O - z_t)}{a_t c_t (x_O - x_t) - b_t (y_O - y_t) + a_t d_t (z_O - z_t)} f_y \quad (6)$$

Seguindo o método proposto por [Saha et al. 2014], dividindo x_{I_t} por y_{I_t} obtem-se:

$$\begin{aligned} x_O(-x_{I_t} f_y d_t + y_{I_t} f_x b_t c_t) + y_O y_{I_t} f_x a_t + z_O(x_{I_t} f_y c_t + y_{I_t} f_x b_t d_t) \\ = x_{I_t}(-f_y d_t x_t + f_y c_t z_t) + y_{I_t}(f_x b_t c_t x_t + f_x a_t y_t) + y_{I_t} f_x b_t d_t z_t \end{aligned} \quad (7)$$

Para a imagem em um tempo $t=1$

$$\begin{aligned} x_O(-x_{I_1} f_y d_1 + y_{I_1} f_x b_1 c_1) + y_O y_{I_1} f_x a_1 + z_O(x_{I_1} f_y c_1 + y_{I_1} f_x b_1 d_1) \\ = x_{I_1}(-f_y d_1 x_1 + f_y c_1 z_1) + y_{I_1}(f_x b_1 c_1 x_1 + f_x a_1 y_1) + y_{I_1} f_x b_1 d_1 z_1 \end{aligned} \quad (8)$$

Para a imagem em um tempo $t=2$

$$\begin{aligned} x_O(-x_{I_2} f_y d_2 + y_{I_2} f_x b_2 c_2) + y_O y_{I_2} f_x a_2 + z_O(x_{I_2} f_y c_2 + y_{I_2} f_x b_2 d_2) \\ = x_{I_2}(-f_y d_2 x_2 + f_y c_2 z_2) + y_{I_2}(f_x b_2 c_2 x_2 + f_x a_2 y_2) + y_{I_2} f_x b_2 d_2 z_2 \end{aligned} \quad (9)$$

No método proposto por [Saha et al. 2014], os últimos termos do sistema de equações são gerados a partir da divisão dos dois lados de 6 por f_y .

$$\frac{y_{I_t}}{f_y} = \frac{-d_t (x_O - x_t) + c_t (z_O - z_t)}{a_t c_t (x_O - x_t) - b_t (y_O - y_t) + a_t d_t (z_O - z_t)}$$

E, mais uma vez, considerando imagens no tempo $t=1$ e $t=2$, obtem-se:

$$\frac{y_{I_1}}{f_y} = \frac{-d_1 (x_O - x_1) + c_1 (z_O - z_1)}{a_1 c_1 (x_O - x_1) - b_1 (y_O - y_1) + a_1 d_1 (z_O - z_1)}$$

$$\frac{y_{I_1}}{f_y} [a_1 c_1 (x_O - x_1) - b_1 (y_O - y_1) + a_1 d_1 (z_O - z_1)] = -d_1 (x_O - x_1) + c_1 (z_O - z_1) \quad (10)$$

$$\frac{y_{I_2}}{f_y} = \frac{-d_2 (x_O - x_2) + c_2 (z_O - z_2)}{a_2 c_2 (x_O - x_2) - b_2 (y_O - y_2) + a_2 d_2 (z_O - z_2)}$$

$$\frac{y_{I_2}}{f_y} [a_2 c_2 (x_O - x_2) - b_2 (y_O - y_2) + a_2 d_2 (z_O - z_2)] = -d_2 (x_O - x_2) + c_2 (z_O - z_2) \quad (11)$$

E subtraindo 10 de 11:

$$\begin{aligned} & \frac{1}{f_y} [y_{I_1} \{a_1 c_1 (x_O - x_1) - b_1 (y_O - y_1) + a_1 d_1 (z_O - z_1)\} \\ & - y_{I_2} \{a_2 c_2 (x_O - x_2) - b_2 (y_O - y_2) + a_2 d_2 (z_O - z_2)\}] = \\ & -d_1 (x_O - x_1) + c_1 (z_O - z_1) - [-d_2 (x_O - x_2) + c_2 (z_O - z_2)] \end{aligned}$$

$$\begin{aligned} y_{I_1} (a_1 c_1 x_O - a_1 c_1 x_1 - b_1 y_O + b_1 y_1 + a_1 d_1 z_O - a_1 d_1 z_1) - y_{I_2} (a_2 c_2 x_O - a_2 c_2 x_2 - b_2 y_O + b_2 y_2 \\ + a_2 d_2 z_O - a_2 d_2 z_2) = f_y (-d_1 x_O + d_1 x_1 + c_1 z_O - c_1 z_1 + d_2 x_O - d_2 x_2 - c_2 z_O + c_2 z_2) \end{aligned}$$

$$\begin{aligned} x_O (y_{I_1} a_1 c_1 - y_{I_2} a_2 c_2 + f_y d_1 - f_y d_2) + y_O (-y_{I_1} b_1 + y_{I_2} b_2) \\ + z_O (y_{I_1} a_1 d_1 - y_{I_2} a_2 d_2 - f_y c_1 + f_y c_2) = y_{I_1} (a_1 c_1 x_1 - b_1 y_1 + a_1 d_1 z_1) \\ + y_{I_2} (-a_2 c_2 x_2 + b_2 y_2 - a_2 d_2 z_2) + f_y (d_1 x_1 - c_1 z_1 - d_2 x_2 + c_2 z_2) \end{aligned} \quad (12)$$

Conhecendo a posição da câmera nestes momentos $C_1 = (x_1, y_1, z_1)$ e $C_2 = (x_2, y_2, z_2)$ e gerando os valores de a, b, c, d a partir dos ângulos de *yaw* e *pitch* é possível montar um sistema de equações e encontrar os valores de posição do objeto $P_O = (x_O, y_O, z_O)$.

Para isto, a partir de 8 e 9 assume-se:

$$e_1 = -x_{I_1} f_y d_1 + y_{I_1} f_x b_1 c_1$$

$$e_2 = y_{I_1} f_x a_1$$

$$e_3 = x_{I_1} f_y c_1 + y_{I_1} f_x b_1 d_1$$

$$e_4 = -x_{I_2} f_y d_2 + y_{I_2} f_x b_2 c_2$$

$$e_5 = y_{I_2} f_x a_2$$

$$e_6 = x_{I_2} f_y c_2 + y_{I_2} f_x b_2 d_2$$

$$B_1 = x_{I_1} (-f_y d_1 x_1 + f_y c_1 z_1) + y_{I_1} (f_x b_1 c_1 x_1 + f_x a_1 y_1) + y_{I_1} f_x b_1 d_1 z_1$$

$$B_2 = x_{I_2} (-f_y d_2 x_2 + f_y c_2 z_2) + y_{I_2} (f_x b_2 c_2 x_2 + f_x a_2 y_2) + y_{I_2} f_x b_2 d_2 z_2$$

A partir de 12:

$$e_7 = y_{I_1} a_1 c_1 - y_{I_2} a_2 c_2 + f_y d_1 - f_y d_2$$

$$e_8 = -y_{I_1}b_1 + y_{I_2}b_2$$

$$e_9 = y_{I_1}a_1d_1 - y_{I_2}a_2d_2 - f_y c_1 + f_y c_2$$

$$B_3 = y_{I_1}(a_1c_1x_1 - b_1y_1 + a_1d_1z_1) + y_{I_2}(-a_2c_2x_2 + b_2y_2 - a_2d_2z_2) + f_y(d_1x_1 - c_1z_1 - d_2x_2 + c_2z_2)$$

Com isto, similarmente ao proposto por [Saha et al. 2014], é possível encontrar a posição do objeto ao resolver o sistema de equações.

$$A_x = B$$

onde,

$$A = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix}, x = \begin{bmatrix} x_O \\ y_O \\ z_O \end{bmatrix}, B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}$$

3. Resultados

Para avaliar o funcionamento das equações, foram geradas simulações com posições aleatórias para a câmera e ângulos de *yaw* e *pitch*, mantendo o obstáculo estático. Foram realizadas simulações com o obstáculo posicionado a 10m, 20m, 30m, 40m e 50m da câmera para avaliar se o erro do cálculo aumenta drasticamente com o aumento da distância do objeto.

Para testar a influência da variação dos ângulos de *yaw* e *pitch* da câmera no cálculo da posição do objeto foram considerados casos com faixas de variação de 0,5° para estes ângulos, como apresentado na tabela 1.

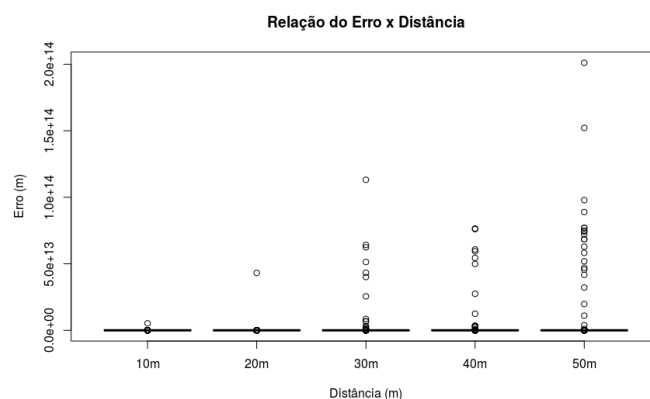
Como descrito anteriormente, o método proposto por [Saha et al. 2014] não considera variações no ângulo de *pitch*. Com isto, não é possível comparar diretamente os dois métodos quando é introduzida uma variação neste valor (a partir da linha 5 da tabela 1). Como o método proposto neste artigo é uma extensão matemática do método apresentado por [Saha et al. 2014], a comparação se torna possível quando o valor de *pitch* é igual a 0° (zero), desse modo os dois métodos passam a ser equivalentes. Cabe salientar que os resultados são exatamente os mesmos no método original e na extensão do mesmo quando o valor do *pitch* é igual a 0°.

Tabela 1. Casos escolhidos

<i>Yaw</i>	<i>Pitch</i>
0°	0°
0° a 0,5°	0°
0,5° a 1,0°	0°
1,0° a 1,5°	0°
0°	0° a 0,5°
0° a 0,5°	0° a 0,5°
0,5° a 1,0°	0° a 0,5°
1,0° a 1,5°	0° a 0,5°
0°	0,5° a 1,0°
0° a 0,5°	0,5° a 1,0°
0,5° a 1,0°	0,5° a 1,0°
1,0° a 1,5°	0,5° a 1,0°
0°	1,0° a 1,5°
0° a 0,5°	1,0° a 1,5°
0,5° a 1,0°	1,0° a 1,5°
1,0° a 1,5°	1,0° a 1,5°

Para cada caso especificado anteriormente foram geradas variações aleatórias de posição para x, y e z dentro de faixas de 1cm para avaliar também o impacto da variação de posição da câmera no cálculo da posição do objeto. Foram avaliadas as faixas de 1cm a 2cm, 2cm a 3cm, ..., 49cm a 50cm.

Considerando os casos propostos, cada variação de parâmetro foi avaliada por simulação com a geração de 100 amostras e o respectivo cálculo da posição estimada do objeto. Após o cálculo da posição estimada do objeto foi possível verificar o erro obtido em cada uma das situações.

**Figura 2. Erro em relação à distância**

Nas figuras 2 e 3 são apresentados os valores do erro em relação a distância. Pode-se perceber, que quanto maior a distância da câmera ao objeto, maiores são os valores dos *outliers* e maior o erro gerado.

Na figura 3 são apresentados os mesmos dados da figura 2, apenas com a exclusão dos *outliers* na visualização do gráfico, sendo que eles continuaram a ser considerados para a geração dos quartis. Interessante notar, que o 3º quartil do erro foi menor que 1m para distâncias de 10m do obstáculo e menor que 5m para distâncias de 50m do obstáculo. A mediana do erro para o caso de 10m de distância foi 0,1003995m e para 50m de distância foi 2,1569715m.

Tabela 2. Tabela quartis (relação à distância)

Distância	Mínimo	1º Quartil	Mediana	3º Quartil	Máximo
10m	0.000345	0.0457180	0.1003995	0.2018530	5.236508e+12
20m	0.001463	0.1882185	0.3987545	0.8237138	4.314262e+13
30m	0.002571	0.3394560	0.7660290	1.6724333	1.131454e+14
40m	0.003622	0.5907107	1.3358880	2.9799325	7.653952e+13
50m	0.003261	0.9488295	2.1569715	4.8007538	2.010945e+14

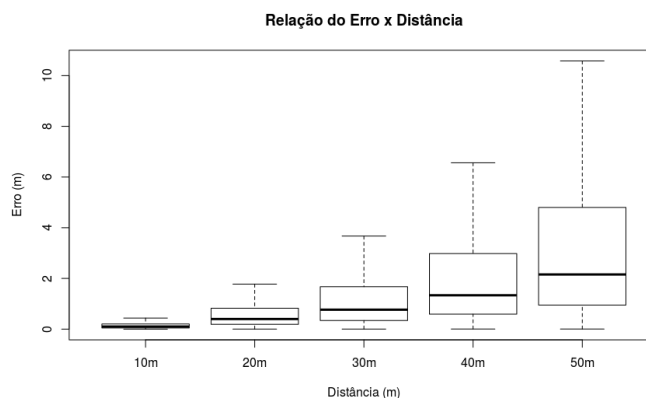


Figura 3. Erro em relação à distância sem *outliers*

No entanto, pode-se perceber pela figura 4 que os maiores *outliers* ocorrem quando o ângulo de *pitch* não tem variação. Vale notar que, como na proposta realizada por [Saha et al. 2014] não existe variação de *pitch*, apenas os resultados com *pitch*=0º poderiam ser obtidos. No entanto, a presente proposta apresenta uma grande possibilidade de melhoria nos resultados apenas por considerar o ângulo de *pitch*.

Apesar dos *outliers* gerados quando *pitch*=0º, é possível notar que eles pouco influenciaram na mediana e nos quartis conforme o gráfico apresentado na figura 5. Vale ressaltar que os *outliers* apenas não foram plotados, mas foram considerados no cálculo dos quartis.

Pela avaliação realizada com a variação do *yaw*, pode-se perceber que ele teve pouca influência nos resultados apresentando *outliers* mais densos quando *yaw*=0º como pode ser observado na figura 6. Como ocorreu com o *pitch*, pode-se notar que a variação do *yaw* também quase não gerou influência nos quartis calculados (figura 7).

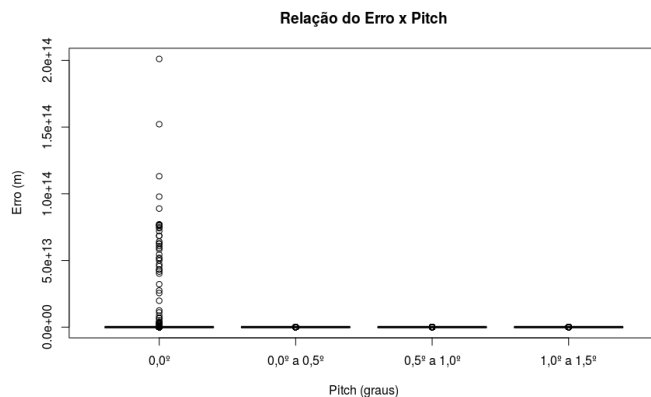


Figura 4. Erro em relação ao Pitch

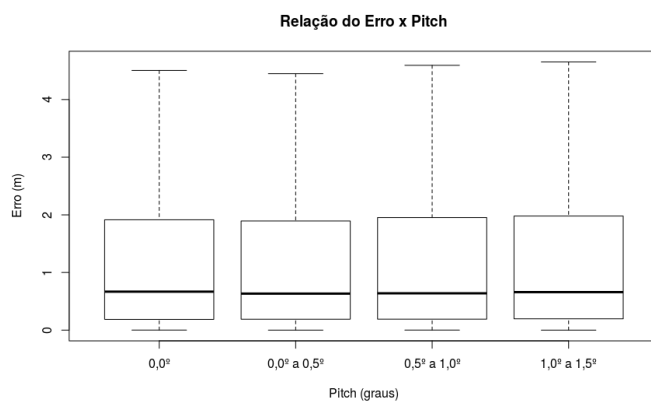


Figura 5. Erro em relação ao Pitch sem outliers

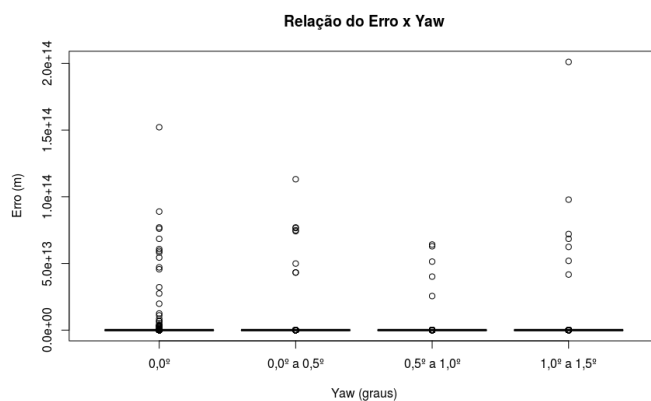


Figura 6. Gráfico yaw

Outro parâmetro que foi avaliado foi em relação à movimentação da câmera entre dois *frames* consecutivos. A figura 8 apresenta a distribuição dos *outliers* nas faixas de variação. Pode-se perceber que o número de *outliers* e seus valores são maiores quando a variação de posição da câmera é pequeno. Em especial com valores menores que 5cm.

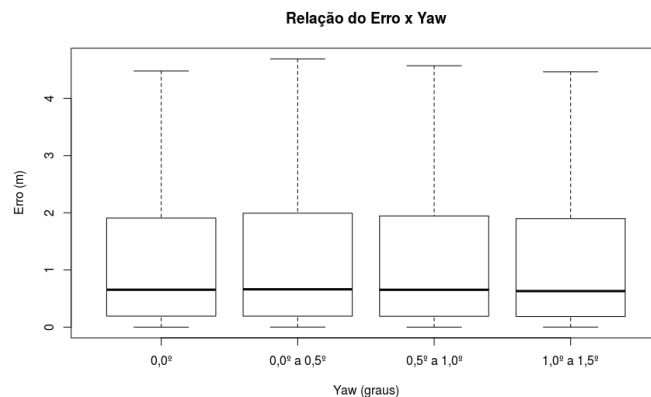


Figura 7. Gráfico yaw-sem-outliers

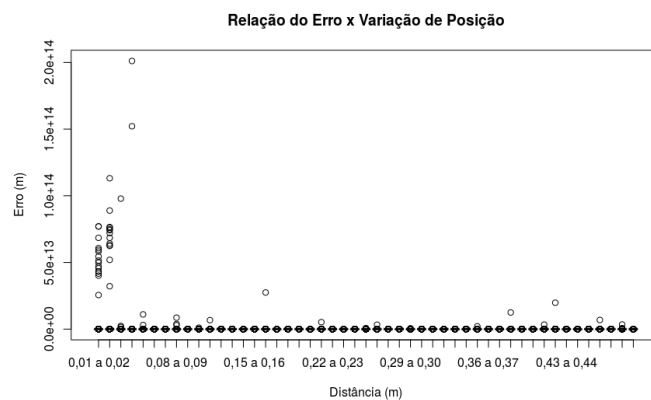


Figura 8. Gráfico Variação de posição

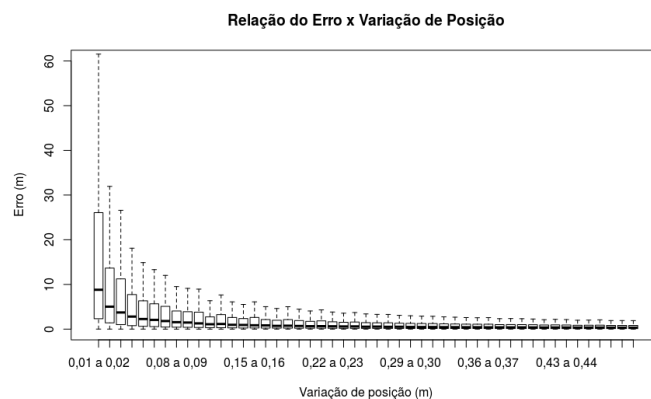


Figura 9. Gráfico Variação de posição-sem-outliers

De forma similar ao que o corre com a distância entre a câmera e o objeto, a variação de posição também tem impacto significativo no erro, como pode ser observado na figura 9. É possível verificar que pequenas variações de posição na câmera geram

um maior erro de cálculo da posição do objeto e também que o erro tende a estabilizar conforme a variação de posição da câmera aumenta.

4. Conclusões e trabalhos futuros

Os resultados obtidos mostram que é viável a utilização do método para o cálculo da posição de um obstáculo. A inclusão da informação do ângulo de *pitch* contribui para a eliminação de grandes valores de *outliers*. No entanto, estes valores de *outliers* ocorrem principalmente quando o deslocamento da aeronave é pequeno, sugerindo que a utilização do método em aeronaves em movimento deve gerar resultados melhores do que quando a aeronave paira no ar.

Outra constatação interessante é que conforme o obstáculo se aproxima da aeronave, a precisão do cálculo de sua posição aumenta significativamente. Sendo que quando o obstáculo está a 30m de distância o erro de cálculo da posição do objeto é menor do que 4m.

Como trabalho futuro pode-se estender o sistema de cálculo da posição do obstáculos para considerar também o ângulo de *roll*, o que deve ajudar ainda mais na redução de *outliers*. Outra melhoria possível seria a aplicação de métodos de filtragem próprios para a eliminação de *outliers* e diminuição do erro de cálculo.

Os métodos aqui propostos podem ser integrados à algoritmos de fluxo óptico ou de detecção e descrição de pontos de interesse para a execução em tempo real.

Referências

- Esrafilian, O. and Taghirad, H. D. (2016). Autonomous flight and obstacle avoidance of a quadrotor by monocular SLAM. In *2016 4th International Conference on Robotics and Mechatronics (ICROM)*, pages 240–245.
- Majumder, S., Shankar, R., and Prasad, M. S. (2015). Obstacle size and proximity detection using stereo images for agile aerial robots. In *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 437–442. IEEE.
- Moon, J. and Prasad, J. V. R. (2011). Minimum-time approach to obstacle avoidance constrained by envelope protection for autonomous UAVs. *Mechatronics*, 21(5):861–875.
- Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163.
- Saha, S., Natraj, A., and Waharte, S. (2014). A real-time monocular vision-based frontal obstacle detection and avoidance for low cost UAVs in GPS denied environment. In *2014 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology*, pages 189–195. IEEE.

Concepção de Uma Plataforma de VANT de Baixo Custo do Tipo Quadricóptero para Uso em Pesquisas

Ricardo M. Bernardo^{1,3}, Luis C. B. da Silva^{2,3}, Paulo F. F. Rosa³

¹ Instituto Federal do Espírito Santo (IFES)
Cachoeiro de Itapemirim – ES – Brasil

²Centro Federal de Educação Tecnológica Celso Suckow Fonseca (CEFET)
Friburgo – RJ – Brasil

³Instituto Militar de Engenharia (IME)
Rio de Janeiro – RJ – Brasil

ricardo.maroquio@ifes.edu.br, luis.silva@cefet-rj.br, rpaulo@ime.eb.br

Abstract. *Several problems that require remote sensing and consequent processing of the sensed data are likely to be tackled with the use of unmanned aerial vehicles (UAVs). However, the ready-to-use platforms available nowadays have a high acquisition and maintenance cost and have little flexibility for modifications. To overcome this issue, this article presents the detailed process of designing a robust, extensible, low-cost UAV platform built with “off-the-shelf” components, including basic sensors and sufficient flight autonomy for use in a variety of academic applications.*

Resumo. *Diversos problemas que demandam sensoriamento de regiões remotas e o consequente processamento dos dados sensorizados são passíveis de serem atacados com o uso de veículos aéreos não tripulados (VANTs). Entretanto, as plataformas prontas para uso disponíveis atualmente possuem custo elevado de aquisição e manutenção, além de terem pouca flexibilidade para modificações. Para superar esta questão, este artigo apresenta o processo de concepção detalhado de uma plataforma de VANT robusta, extensível e de baixo custo, construída com componentes “de prateleira” facilmente substituíveis, incluindo sensores básicos e com autonomia de voo suficiente para uso em diversas aplicações acadêmicas.*

1. Introdução

São inúmeras as aplicações que demandam aquisição de dados sensorizados em áreas remotas e seu imediato processamento para a devida extração de alguma informação útil sobre a área em questão e sobre os objetos presentes nessa área. Os veículos aéreos não tripulados (VANTs) são plataformas robóticas móveis, popularmente conhecidos como drones, cujo uso vem se tornando cada vez mais comum nessas situações, tanto para fins civis quanto para fins militares. O uso militar pode ser interessante em uma vasta gama de tarefas, incluindo vigilância e patrulhamento, busca e resgate, mapeamento, abastecimento e comunicação. O uso civil de alguns tipos de VANTs vem ganhando cada vez mais espaço. As aplicações civis para VANTs incluem entrega de pacotes, aquisição de imagens, entretenimento e agricultura [Valavanis and Vachtsevanos 2014].

Vários fabricantes produzem VANTs “prontos para voar” (RTF, do inglês *ready to fly*) passíveis de serem usados como plataforma de pesquisa. Como vantagem, um VANT RTF permite que o usuário/pesquisador dedique-se à camada de software de mais alto nível, uma vez que o equipamento (hardware e software básico) já vem calibrado e testado. Entretanto, os VANTs RTF apresentam algumas desvantagens relevantes para uso em pesquisas, incluindo alto custo de aquisição e de manutenção e pouca flexibilidade para modificações, por usar componentes específicos para ele. A Tabela 1 apresenta características de algumas plataformas de VANT de pequeno porte do tipo quadricóptero RTF disponíveis no mercado, usando dados obtidos em [Graphiq 2017]. As plataformas apresentadas foram escolhidas de acordo com características usualmente presentes em VANTs de pesquisa, a saber: autonomia de voo mínima de 15 minutos, peso máximo de 3kg, carga extra (*payload*) mínima de 300g e preço máximo de US\$ 3.500,00.

Tabela 1. Características e preços de VANTs RTF do tipo quadricóptero.

Veículo	Envergadura	Autonomia Máxima	Velocidade Máxima	Peso (g)	Payload (g)	Preço (US\$)
DJI Matrice 100	650mm	40min	79km/h	2355	1000	3300,00
MicroDrones MD4-1000	1030mm	88min	54km/h	2650	1200	2000,00
TurboAce Matrix G	1110mm	25min	NI	2170	1882	2770,00
AscTec Pelican	650mm	20min	58km/h	1000	650	NI
3DR Solo	460mm	25min	88km/h	1500	420	1000,00

Com a popularização da robótica e o consequente aumento da produção, o preço dos componentes vem caindo, viabilizando cada vez mais a construção de VANTs de baixo custo. Além disso, os componentes eletrônicos estão cada vez menores, o que também favorece a construção de VANTs mais leves e com maior autonomia de voo. Entretanto, a escolha de uma configuração adequada para se construir um VANT não é uma tarefa trivial, principalmente pela diversidade existente, e isso faz com que muitos acabem optando por uma plataforma RTF.

Com a ajuda da computação, tarefas de mais alto nível podem ser executadas pelos VANTs, sendo que tais tarefas dependem bastante dos recursos tecnológicos embarcados. Quando os dados a serem processados não são tão complexos, um computador embarcado pode realizar o trabalho e oferecer um tempo de resposta menor e uma independência maior. Vale ressaltar que, por se tratar de um veículo cujo peso é fator determinante para a autonomia de voo, os recursos embarcados devem ser ponderados de acordo com cada aplicação e nem sempre serão suficientes para que o processamento ocorra localmente. Essas restrições podem ser superadas com o uso de mais de um veículo de forma colaborativa ou usando processamento em uma estação de maior poder computacional, como ocorre, por exemplo, no trabalho de [Minaeian et al. 2016].

Diante do exposto, este artigo apresenta o processo de concepção de um VANT do tipo quadricóptero, de baixo custo e com aparato tecnológico básico para a realização de diferentes tipos de pesquisas. A seção 2 apresenta uma breve revisão da literatura relacionada. Na seção 3, uma visão geral sobre VANT é apresentada. A seção 4 descreve a plataforma de VANT proposta. A seção 5 apresenta dados numéricos referentes ao desempenho do VANT proposto, obtidos por simulação. Por fim, são apresentadas as conclusões e sugestões para trabalhos futuros.

2. Revisão de Literatura

Diversos trabalhos de pesquisa fazem uso de VANTs como plataforma base para se extrair dados de ambientes remotos, processá-los e, em alguns casos, atuar de alguma forma nestes ambientes. Em boa parte dessas pesquisas, os VANTs são construídos pelas pessoas envolvidas. Este é o caso, por exemplo, dos trabalhos de [Castillo et al. 2003, Zhou 2010], cujos VANTs são do tipo, respectivamente, quadricóptero e asa fixa.

O trabalho de [Bouabdallah et al. 2004] foi um dos pioneiros na concepção de uma plataforma de VANT de baixo custo do tipo quadricóptero sem finalidade específica. Os autores desenvolveram um micro-quadricóptero, incluindo o projeto mecânico, a modelagem dinâmica, o sensoriamento e o sistema de controle. Por outro lado, com o propósito mais voltado ao ensino, [Devaud et al. 2012] descreve a construção de um VANT do tipo quadricóptero de baixo custo por uma equipe de cinco estudantes, que tinham a missão de criar uma plataforma modular para uso diversificado, com orçamento limitado a € 450,00 em um período máximo de 60 dias (~500h). Foi construído um quadricóptero com estrutura em alumínio (braços e suportes centrais) de 610mm de diâmetro, com peso de 1.4kg e com autonomia de voo de pelo menos 15 minutos usando uma bateria LiPo de 4500mAh.

O trabalho de [Davis et al. 2013] apresenta o processo de desenvolvimento de um quadricóptero de baixo custo para experimentos com enxames de VANTs. Os autores construíram um quadricóptero de fibra de carbono com custo inferior a US\$ 500,00 usando a aviônica do Ardupilot APM 2.0 e componentes eletrônicos de prateleira. As considerações de projeto estabeleceram uma autonomia mínima de 10 minutos e o peso máximo de 500g, para uso principal em ambientes fechados. O limite de peso de 500g foi estabelecido como forma de aumentar a segurança de uso e reduzir as perturbações quando a formação do enxame forçar uma proximidade muito grande entre os veículos.

Um componente fundamental para o bom funcionamento de um quadricóptero é a controladora de voo, considerada o “cérebro” do VANT. No trabalho de [Devaud et al. 2012], os autores optaram por desenvolver sua própria controladora de voo. Já no trabalho de [Davis et al. 2013], os autores optaram pela Ardupilot APM 2.0, uma controladora pronta para uso com alta disponibilidade e muitos usuários. A importância deste componente fez com que os autores de [Yang et al. 2016] analisassem as opções comerciais atualmente disponíveis e realizassem uma comparação entre as mais conhecidas.

A primeira controladora de voo analisada em [Yang et al. 2016] foi a *Pixhawk*, um projeto *open hardware* desenvolvido em colaboração com a 3D Robotics. Como vantagens, foram citadas a grande comunidade de usuários e as possibilidades de customização. Por outro lado, tanto o software quanto o hardware estão suscetíveis a pequenos problemas, por conta da baixa frequência de atualização do produto. Várias controladoras da fabricante DJI também foram analisadas, a saber, *A2*, *Wookong-M*, *Naza-M V2* e *Naza-M Lite*. Por serem hardwares proprietários, os autores não conseguiram obter detalhes dos componentes utilizados em cada controladora, mas concluíram que todas oferecem grande confiabilidade e robustez, bem como uma interface humano-computador amigável e uma documentação bem elaborada. Contudo, as controladoras da DJI são caras e não podem ser modificadas. Outra controladora analisada foi a *Navio2*, da Emlid. A *Navio2* funci-

ona acoplada ao minicomputador *Raspberry Pi*¹, versão 2 ou 3. Isso significa que, além de integrar todos os sensores necessários à navegação, o poder computacional disponível permite realizar missões de mais alto nível, favorecendo seu uso em projetos voltados para pesquisa e educação. Os autores analisaram também a *Snapdragon Autopilot*, da Qualcomm, que possui um poder computacional muito maior do que as demais, permitindo a realização de tarefas mais complexas. Entretanto, a *Snapdragon Autopilot* ainda não foi oficialmente lançada. A *OcPoC Autopilot*, da Aerotenna, oferece uma plataforma robusta baseada em FPGA, o que lhe permite ter o melhor desempenho de controle entre as comparadas. Entretanto, também é uma plataforma de alto custo e baixo poder computacional extra. Por fim, os autores analisaram a *Trinity*, da AscTec, que possui redundâncias que a torna a mais robusta entre as comparadas. Contudo, a AscTec ainda não recebeu autorização para vendê-la. A Tabela 2, traduzida e adaptada de [Yang et al. 2016], resume o comparativo realizado pelos autores.

Tabela 2. Comparativo das diferentes controladoras disponíveis no mercado.

Recursos	Pixhawk	DJI A2	Navio2	Snapdragon	OcPoC	Trinity
Tipos de VANTs	Avião, Helicóptero, Multicóptero	Multicóptero	Avião, Helicóptero, Multicóptero	Avião, Helicóptero, Multicóptero	Avião, Helicóptero, Multicóptero	Multicóptero
Modos de voo	Auto, Manual, Assistido	Auto, Manual, Assistido	Auto, Manual, Assistido	Auto, Manual, Assistido	Auto, Manual, Assistido	Auto, Manual, Assistido
Processador	Cortex-M4, Cortex-M3	-	Quad-core ARMv8 (RPi3)	Quad-core 2.26GHz Krait + Hexagon DSP	Dual-core ARM A9 + FPGA Logic	-
Memória	256KB RAM, 2MB Flash	-	1GB RAM (RPi3)	2GB LPDDR3, 32GB Flash	512MB DDR3, 128MB Flash	-
SO	NuttX	-	Linux	Linux + QuRT	Linux	-
IMU	2 x 6DOF, 1 x Magnet.	-	2 x 9DOF	1 x 9DOF	2 x 9DOF	3 x 9DOF
Interfaces	SPI, I2C, CAN, PWM, ADC	CAN	UART, I2C, ADC, PWM	USB, UART, I2C	CAN, I2C, SPI, USB, Ethernet	-
Redundância	IMU	-	IMU	Não	IMU	Sistema
Preço (US\$)	199,00	579,00	168,00	675,00	599,00	-

As opções de controladora de voo apresentadas na Tabela 2 mostram a diversidade de configurações disponíveis. As principais siglas e acrônimos presentes nessa tabela são descritas na Seção 3.

3. Visão Geral de um VANT

Um VANT é um robô móvel aéreo que possui sensores intrínsecos e extrínsecos para navegação e percepção, assim como atuadores para movimentação e interação. Em relação ao tipo, os VANTs mais comuns são os aviões (asa fixa), helicópteros (rotores) e os multicópteros (multirotores). Nos aviões, a flutuação é obtida pelo contato das asas com o ar em alta velocidade. Nos helicópteros e multicópteros, a flutuação depende exclusivamente dos propulsores. O trabalho apresentado em [Korchenko and Illyash 2013] classifica detalhadamente os VANTs sob 16 diferentes critérios. Em suma, as classificações para VANTs do tipo quadricóptero, semelhantes aos apresentados neste artigo, podem ser

¹Preço de US\$ 35,00 não incluso no valor da *Navio2* apresentado na tabela.

definidas como: um veículo tático (curto alcance de voo), do tipo multicóptero (flutuação baseada em rotores), com pouso e decolagem vertical (V-TOL, do inglês *Vertical Take-off and Landing*) e que pode ter diferentes tamanhos, a saber: nano (até 25g), micro (até 500g) e mini (até 20kg). Em [Valavanis and Vachtsevanos 2014], o tamanho micro é definido para VANTs entre 1kg e 5kg, portanto, neste caso, os quadricópteros mencionados até aqui são do tipo micro-VANT. A Figura 1 apresenta uma visão em alto nível da arquitetura básica de um VANT.

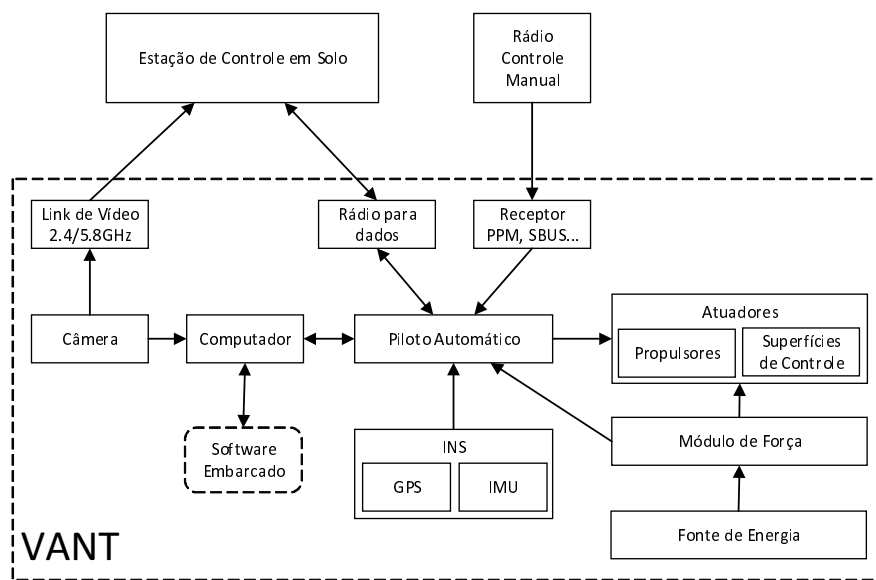


Figura 1. Esquema em alto nível de um VANT e seus componentes.

Na Figura 1, o bloco “Piloto Automático” representa a controladora de voo, que obtém dados de um sensor de navegação inercial (INS, do inglês *Inertial Navigation System*) para conhecer o grau de inclinação e de aceleração nos três eixos espaciais obtidos da Unidade de Medição Inercial (IMU, do inglês *Inertial Measurement Unit*), bem como a posição global usando um sistema global de navegação por satélite (GNSS, do inglês *Global Navigation Satellite System*). Esses sistemas são compostos basicamente por sensor GPS², acelerômetro, girômetro, magnetômetro e barômetro, sendo que estes quatro últimos compõem a IMU. O GNSS e o INS combinados fornecem a posição e a orientação do VANT no espaço a partir de um sistema de referência global fixo. Outros sensores que podem auxiliar o sistema de navegação são os sensores de distância, como sonares e medidores de distância por infravermelho ou laser. A câmera (RGB, infravermelha ou térmica) é um sensor que pode ter múltiplas funções em um VANT, pois permite o uso de técnicas de visão computacional para extrair informações de alto nível das imagens captadas do ambiente.

Os componentes de comunicação de um VANT são fundamentais para transmissão/recepção de dados de uma estação de controle em solo. Toda a comunicação é feita por ondas de rádio. Para transmissão de dados de telemetria, usa-se a frequência de 433MHz ou 915MHz. Para vídeos, usa-se 2.4GHz ou 5.8GHz. Para rádio controle, normalmente usa-se também a frequência de 2.4GHz. Portanto, diferentes transmissores e

²Atualmente, os sistemas de satélite disponíveis são o GPS (americano), o GLONASS (russo), o GALILEO (europeu) e o COMPASS (chinês).

receptores de rádio podem ser necessários em um VANT [Drozd 2015]. Por fim, para que tudo isso funcione, é necessária uma fonte de energia. A fonte mais comum é a bateria LiPo (polímeros de lítio), principalmente em VANTs do tipo quadricóptero. Outros tipos de veículos, usualmente aqueles de maior porte, podem fazer uso de combustíveis.

4. O VANT Proposto

Como já mencionado, o VANT proposto é do tipo quadricóptero. A escolha desse tipo de veículo se deu, principalmente, pelo fato de que ele possui um mecanismo de controle simples, capacidade de decolar e pousar na vertical, capacidade de pairar e abundância de componentes disponíveis no mercado, características essas relevantes para projetos de pesquisa envolvendo sensoriamento de áreas remotas de curta distância e com baixo orçamento. A Figura 2 apresenta uma visão geral do quadricóptero.

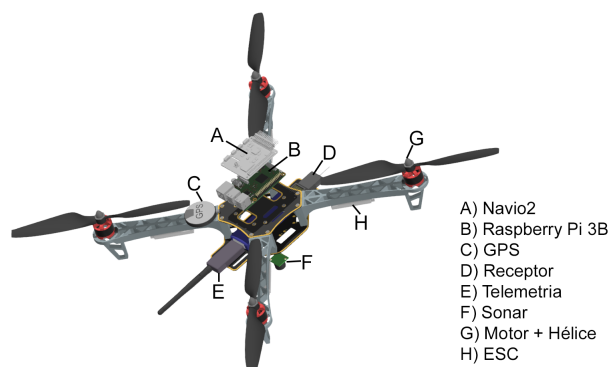


Figura 2. Esquema em alto nível do VANT proposto e de seus componentes.

Na Figura 2, o componente A corresponde à controladora de voo Navio2, cujos detalhes são apresentados em [Yang et al. 2016]. Os fatores mais relevantes para a escolha da Navio2 foram a robustez, o bom suporte e a associação com o Raspberry Pi 3 (RPi3 - componente B), que possui poder computacional relevante, com processador ARMv8 64bit 1.2GHz de quatro núcleos e 1GB de RAM, além de conector WiFi 802.11n e bluetooth 4.1 BLE, favorecendo a comunicação com outros dispositivos. O RPi3 é pequeno e possui baixo custo, além de grande comunidade de usuários, boa documentação e ampla aceitação na área acadêmica. Seu poder computacional alcança 462,07 MFLOPS, sendo que cerca de 10% é usado pelo software da controladora de voo. O kit da controladora Navio2 já vem com o sensor e antena GPS (componente C) compatíveis.

O *frame*, estrutura de suporte dos componentes, é um DJI Flamewheel F450, que possui 450mm de envergadura (ou diâmetro). A escolha se deu pela sua alta disponibilidade, baixo custo e boa resistência a impactos. Os motores usados são do tipo 2312 da DJI (G) com 960kv e que, combinados com as hélices de 9.4 polegadas também da DJI, são capazes de erguer de 350g a 400g cada motor usando-se apenas 50% de sua capacidade. Os motores são ligados aos ESCs (controladores eletrônicos de velocidade, componente H) que por sua vez são ligados diretamente à bateria através de um mecanismo de distribuição de energia presente no *frame*. Os motores, ESCs e hélices compõem o kit E310 da DJI. O componente D é o receptor do rádio controle, que deve ter ao menos 4 canais, para controlar a propulsão, a guinada, a rolagem e a arfagem. Neste caso, o receptor

usado foi um FrSky X8R, que possui conexão compatível com a Navio2 (SBUS) e tem 16 canais, permitindo controlar outros dispositivos que estejam embarcados. Os dados de telemetria do veículo são enviados a uma estação de controle em solo (ECS) através do componente E, que é um rádio transmissor/receptor de 433MHz da 3DR específico para telemetria via porta UART ou USB.

Um componente opcional que vale ser mencionado é a câmera. A Raspberry Camera V2 conecta-se diretamente ao RPi3, possui um sensor de 8MP e oferece os modos de vídeo 1080p a 30fps, 720p a 60fps e VGA a 90fps. Para estabilizar a câmera, um gimbal de dois eixos (rolagem e arfagem) pode ser construído com dois servos ligados diretamente à Navio2, que faz o controle de estabilização de forma automática. Outro sensor opcional que vale mencionar é o sonar, capaz de auxiliar na medição mais precisa de altitudes inferiores a 7 metros. O sonar usado foi o MaxBotix I2CXL-MaxSonar-EZ0, um sonar de alto desempenho e custo acessível. A Tabela 3 apresenta os componentes e seus custos, bem como os custos de componentes opcionais que podem ser úteis ao quadricóptero.

Tabela 3. Componentes básicos (a) e opcionais (b) usados na montagem.

Componentes básicos	Preço (US\$)	Componentes opcionais	Preço (US\$)
DJI Flamewheel F450	32,00	MaxBotix I2CXL-MaxSonar-EZ0	45,00
DJI E310 (motores + ESCs + hélices)	159,00	Raspberry Camera V2	25,00
Receptor FrSky X8R	37,00	Servo TowerPro MG90S x 2	10,00
Rádio p/ telemetria 3DR 433MHz	49,00	Rádio controle Taranis XD9 Plus	200,00
Kit Navio2 + Raspberry Pi 3 Model B	244,00	Bateria Multistar 3S 5200mAh	32,00
Total	521,00	Total	312,00

(a)

(b)

A bateria e o rádio controle foram dispostos como opcionais porque podem ser oriundos de projetos preexistentes. O custo básico do quadricóptero ficou em US\$ 521,00. O custo total, incluindo os opcionais, é de US\$ 833,00. É possível encontrar motores, ESCs, receptores de rádio, rádio-controles e receptores de telemetria similares aos apresentados a um custo mais baixo. Ainda, é possível adquirir o kit da Navio2 de US\$ 209,00 por US\$ 143,00 cadastrando-se como estudante/pesquisador no site do fabricante.

Uma estação de controle em solo (ECS) pode ser usada para comunicação direta com o VANT. Dois softwares de ECS populares são o APM Planner e o Mission Planner. O primeiro tem a vantagem de executar nas plataformas Windows, Mac e Linux, enquanto o segundo executa somente no Windows. Em contrapartida, o Mission Planner possui uma interface mais sofisticada e melhor documentada. Entre as principais funções do software da ECS, estão a visualização da posição do VANT em um mapa, o envio de comandos de atuação, o planejamento de missões e a visualização de dados de telemetria. A plataforma proposta permite realizar voos controlados tanto pela ECS quanto pelo rádio controle. A ECS permite a criação de missões em que o VANT realiza tarefas autônomas, incluindo decolagem, passagem por *waypoints* e pouso. Além disso, é possível controlar o VANT enviando-o comandos via protocolo MAVLink através de um console na ECS. Apesar de não ser item obrigatório, a ECS oferece recursos relevantes para se trabalhar com VANTs.

5. Resultados Obtidos

Os detalhes comportamentais do VANT foram obtidos através de simulação usando o software eCalc Multicopter³. O simulador leva em consideração um coeficiente de arrasto padrão para quadricópteros. Os principais parâmetros de entrada do simulador são: peso (sem bateria, 1035g), número de rotores (4), diâmetro do *frame* (450mm), ESC com 20A de corrente contínua e máxima e resistência de 0.01Ohm, motor DJI 2312-960, hélice DJI com 9.4pol de diâmetro, 5pol de *pitch* e 2 pás. A Figura 3 apresenta gráficos de autonomia, alcance e velocidade.

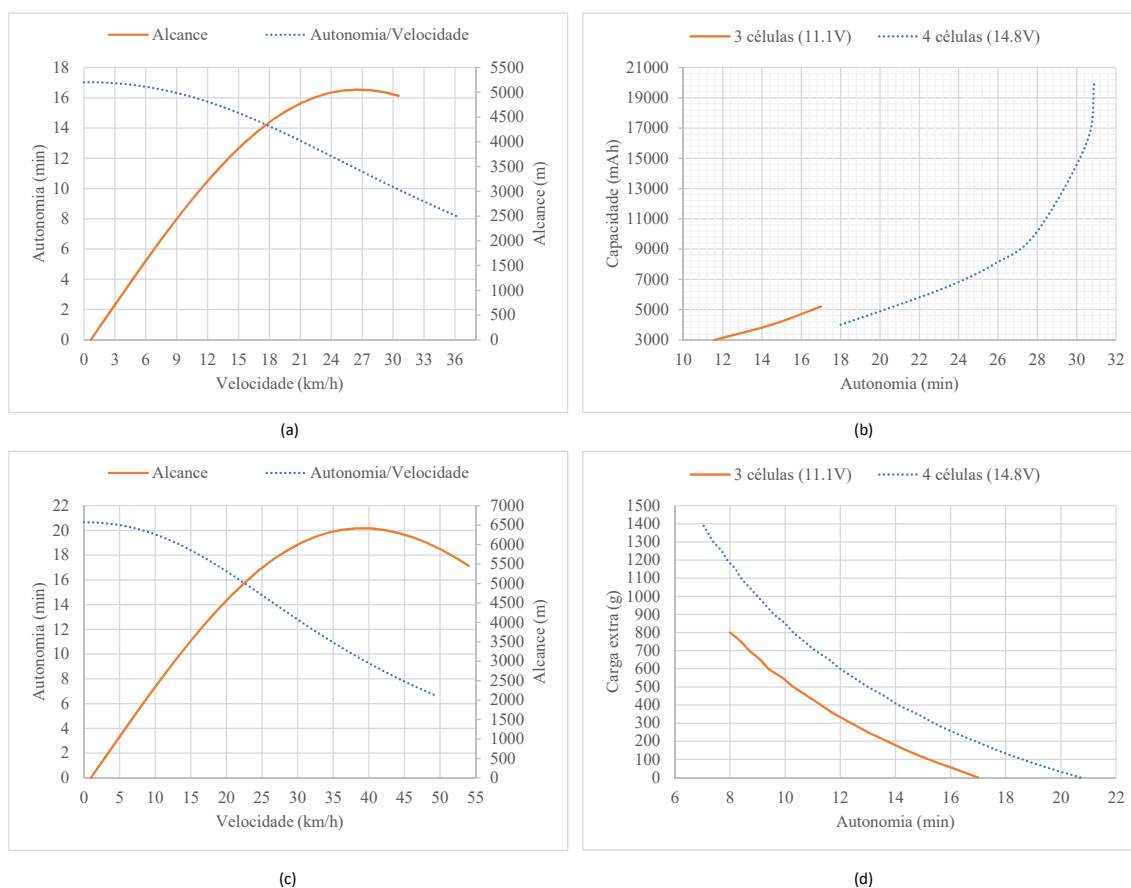


Figura 3. Autonomia e alcance em dada velocidade, usando bateria de 3 células (a) e 4 células (c). (b) Variação da autonomia de acordo com a capacidade da bateria. (d) impacto do peso extra na autonomia, usando bateria de 5200mAh.

Na Figura 3 (a) e (b), o tempo máximo de voo ocorre com o VANT em velocidade nula (pairando), sendo 17min e 20,7min, respectivamente. O alcance máximo em (a) é de 5053m, obtido quando em velocidade constante de 30,17m/s, enquanto em (c) é de 6418m, obtido quando em velocidade constante de 31,27m/s. Em (b) é possível visualizar que capacidades de bateria superiores a 17000mAh não impactam significativamente na autonomia, por conta do peso da bateria e do VANT somados se aproximarem da capacidade máxima de empuxo do VANT. Por fim, em (d), é possível visualizar o impacto do peso de componentes extras embarcados no VANT. Assim, havendo necessidade de um componente adicional, basta saber seu peso para se estimar a autonomia.

³Disponível em <http://www.ecalc.com>

Apesar do software de simulação possuir um erro médio de $\pm 15\%$, informado pelo fabricante, em experimentos reais com usando bateria de 3 células com 5200mAh, obteve-se autonomia de 17,2min, diferindo em apenas 0,01% do valor de 17min calculado pelo simulador. A plataforma concebida possui mais de 10 modos de voo, incluindo os modos autônomo (por *waypoints* georreferenciados), estabilizado, altitude fixa, orientação relativa e retorno autônomo ao ponto de decolagem.

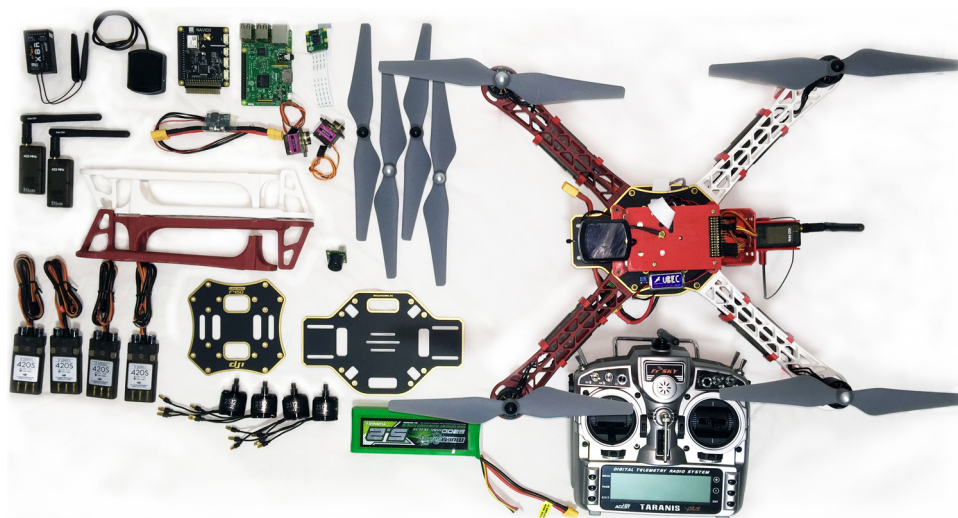


Figura 4. Componentes necessários e um dos cinco quadricópteros montados usando as definições propostas neste trabalho.

Na Figura 4, pode-se notar que o quadricóptero possui diversas peças plásticas de cor vermelha. Tais peças foram projetadas e impressas em uma impressora 3D em plástico ABS. Como pode ser visto, elas são usadas para se realizar uma montagem mais organizada, mas não são de uso obrigatório. Os arquivos para impressão 3D em formato STL podem ser obtidos em <https://goo.gl/KmiQlo>, incluindo os arquivos para montagem do gimbal para estabilização da câmera, que também foi usado no projeto. Um vídeo com os passos da montagem está disponível em <https://goo.gl/bWmqBN>.

6. Conclusão

Este artigo apresentou uma proposta de VANT do tipo quadricóptero com uso voltado para pesquisas, com peso entre 1,2kg e 2,6kg, autonomia de 11,6min a 30,9min e velocidade máxima de 36km/h a 53km/h, de acordo com a bateria utilizada. A plataforma proposta vem sendo usada com sucesso em 2 trabalhos de doutorado, um relacionado à cobertura de áreas usando múltiplos VANTs e outro que aborda a classificação de entidades terrestres móveis. Ambos os trabalhos fazem uso de voo autônomo por *waypoints* georreferenciados. A construção de uma plataforma de VANT como a apresentada ajuda a diminuir custos e a adquirir conhecimentos importantes para a manutenção posterior da plataforma, uma vez que os componentes passam a ser conhecidos de forma mais detalhada pela equipe de pesquisa. O uso de uma configuração predefinida pode poupar meses de trabalho e evita montagens com gargalos. Como trabalho futuro, vale mencionar a criação de um ambiente de simulação visual que incorpore o modelo dinâmico do VANT proposto e que permita a programação de algoritmos diretamente no simulador. Ainda como trabalho futuro, vale mencionar a construção de um gimbal com motores do

tipo *brushless* que, diferentemente do gimbal com servos, é capaz de estabilizar a câmera sem vibração, evitando borramento e computação extra para pré-processamento das imagens capturadas. Outro projeto, já em andamento, aborda a construção de um VANT de baixo custo usando componentes “de prateleira” e folhas e tubos de fibra de carbono, que tenha autonomia superior a 50 minutos, capacidade de carga extra (*payload*) de pelo menos 1kg e peso total inferior a 3kg, permitindo novas possibilidades de uso.

Agradecimentos

Os autores agradecem ao Edital Universal 2016 do CNPq e ao Edital CAPES/Pró-Estratégia.

Referências

- Bouabdallah, S., Murrieri, P., and Siegwart, R. (2004). Design and control of an indoor micro quadrotor. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 5, pages 4393–4398. IEEE.
- Castillo, P., Dzul, A., and Lozano, R. (2003). Real-time stabilization and tracking of a four rotor mini-robotcraft. In *IEEE European Control Conference (ECC)*, pages 3123–3128. IEEE.
- Davis, E., Nizette, B. E., and Yu, C. (2013). Development of a low cost quadrotor platform for swarm experiments. In *32nd IEEE Chinese Control Conference (CCC)*, pages 7072–7077. IEEE.
- Devaud, J.-B., Najko, S., Le Nahédic, P., Maussire, C., Zante, E., and Marzat, J. (2012). Full design of a low-cost quadrotor uav by student team. In *IEEE International Conference on System Engineering and Technology (ICSET)*, pages 1–6. IEEE.
- Drozd, A. L. (2015). Spectrum-secure communications for autonomous uas/uav platforms. In *IEEE Military Communications Conference*. IEEE.
- Graphiq (2017). Specout tech research engine website. <http://www.specout.com>. Online. Acesso em 15 mar. de 2017.
- Korchenko, A. and Illyash, O. (2013). The generalized classification of unmanned air vehicles. In *2nd IEEE International Conference on Actual Problems of Unmanned Air Vehicles Developments Proceedings (APUAVD)*, pages 28–34. IEEE.
- Minaeian, S., Liu, J., and Son, Y.-J. (2016). Vision-based target detection and localization via a team of cooperative uav and ugvs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(7):1005–1016.
- Valavanis, K. P. and Vachtsevanos, G. J. (2014). *Handbook of unmanned aerial vehicles*. Springer Publishing Company, Incorporated.
- Yang, Z., Lin, F., and Chen, B. M. (2016). Survey of autopilot for multi-rotor unmanned aerial vehicles. In *42nd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pages 6122–6127. IEEE.
- Zhou, G. (2010). Geo-referencing of video flow from small low-cost civilian uav. *IEEE Transactions on automation science and engineering*, 7(1):156–166.

**V Workshop de Comunicação em Sistemas
Embarcados Críticos (WoCCES)
SBRC 2017
Sessão Técnica 3**

Physical Web: Avaliação de Técnicas de Estimativa de Localização de Dispositivos BLE em Ambientes Internos

Guilherme David Branco, Samuel Vinicius Vieira Pala, Jacir Luiz Bordim

¹Departamento de Ciência da Computação, Universidade de Brasília

Asa Norte, 70910-900, Brasília-DF, Brasil

E-mail{gdbranco,samuelpala}@gmail.com, bordim@unb.br

Abstract. *Physical Web consists in an enabling technology aiming to provide seamless interactions with physical objects and locations, to achieve such a goal the devices are expected to provide acceptable location information. This paper investigates the location information accuracy provided by a BLE device, for this purpose, we have evaluated location accuracy using renowned Received Signal Strength Indicator (RSSI) location-estimation techniques. Besides that, another factor that needs to be accounted for relies in the seamless interactions, needing fast filtering methods to provide a better user experience. Experimental results shows that the BLE beacons used are capable of providing reasonable distance estimation up to 0,17 meters of error, while waiting around 10 through 15 seconds.*

Resumo. *Physical Web consiste em uma tecnologia que visa fornecer interações contínuas com objetos físicos e suas respectivas localizações, para atingir o seu objetivo, esses dispositivos devem ser capazes de fornecer informações de localização a um nível aceitável. Este estudo investiga a precisão de informações de localização fornecida por um dispositivo BLE, para isso, avaliou-se sua precisão de localização usando os mecanismos de estimativa de distância baseadas no Indicador de Potência do Sinal (RSSI), além disto, outro fator que deve ser considerado consiste na experiência de uso do sistema, busca-se melhor tempo de resposta para o usuário por meio de técnicas de filtragem rápidas, resultando em esperas pequenas, fazendo com que esta experiência ocorra com maior fluidez. Os resultados experimentais mostram estimativas com erro médio de 0,17m com uma espera entre 10 a 15 segundos.*

1. Introdução

Physical Web (PW) compreende um conceito de tecnologia que visa fornecer interações contínuas com objetos físicos e suas respectivas localizações [Google]. Utiliza sinais transmitidos por pequenos dispositivos, chamados de *beacons*, os quais funcionam como um “farol” enviando sinais para que receptores próximos possam identificá-los. Para empresas, escolas, lojas e outros estabelecimentos, a tecnologia configura uma forma eficiente de comunicação com pessoas que estejam na proximidade do *beacon* emissor e que possam ter interesse em seus anúncios. Por exemplo, uma empresa pode enviar propagandas direcionadas para clientes potenciais conforme ilustrado na Figura 1.

Recentemente, a Google disponibilizou um conjunto de protocolos em formato aberto com o intuito de difundir o uso da PW. Para tal, a Google disponibilizou o protocolo Eddystone-URL, o qual define os parâmetros e os formatos para acesso às

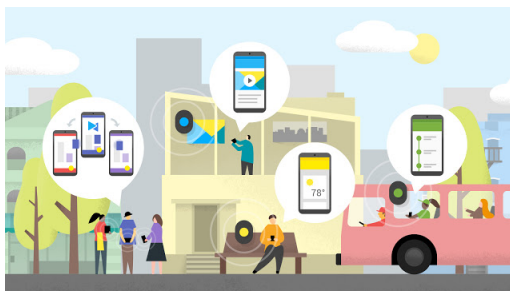


Figure 1. Autor Google, Exemplo de aplicação da PW.

informações disponibilizadas pelo *beacon* por meio da tecnologia *Bluetooth Low Energy* (BLE) [Google 2016]. O BLE foi criado com o intuito de proporcionar melhor autonomia energética aos dispositivos [Tanenbaum and Wheterall 2011]. Esta tecnologia tornou possível a utilização de pequenas baterias de 3V que possibilitam manter os dispositivos BLE ativos por períodos bastante longos se comparados com tecnologias anteriores [Lindh 2016]. O BLE opera em quatro modos distintos: periférico, central, difusão e observador. Na PW, o primeiro e o terceiro modo, que se referem a anunciantes, constituem os principais modos de operação [Townsend et al. 2014]. Os dispositivos móveis escutam periodicamente o sinal transmitido pelos *beacons* via conexão BLE, este sinal consiste de pequenos pacotes de dados que identificam cada *beacon* e se encontram padronizados por protocolos de encapsulamento, tais como o iBeacon da *Apple* e Eddystone da *Google*.

Para este estudo, o principal emprego da *Physical Web* encontra-se na identificação e localização dos objetos anunciados. Por exemplo, ao receber um anúncio em um centro comercial ou área de lazer, deseja-se poder distinguir entre os anunciantes e seus produtos. Isto deve ser feito de maneira fluída ao usuário, com rapidez na convergência de estimativa de distância de forma a permitir identificar e filtrar os objetos por sua proximidade. Os principais desafios para a identificação de produtos e anunciantes recaem na necessidade de estimar a distância do receptor e do *beacon*. Realiza-se a estimativa por meio do Indicador de Potência do Sinal (RSSI), entretanto, este pode variar significativamente, especialmente em ambientes fechados [Chowdhury et al. 2015]. Este trabalho tem como foco verificar a acurácia das técnicas de estimativa de distância para dispositivos BLE que utilizam o RSSI. Além disso, avalia-se o tempo e o quantitativo de leituras necessárias para se obter uma estimativa de distância aproximada. Os resultados experimentais mostraram estimativas com erro médio de 0,17m e um tempo de convergência entre 10 a 15 segundos.

Para este estudo, apresenta 3 seções principais: a Seção 2 mostra uma breve revisão dos modelos de propagação e técnicas de filtragem. A Seção 3, apresenta o cenário de experimentos e a Seção 4, os resultados dos experimentos laboratoriais sobre a estimativa de distância e melhoria da fluidez de utilização por meio da calibragem do tempo de espera da convergência das estimativas. Por fim, a Seção 5 apresenta a conclusão.

2. Mecanismos de Localização

Devido a natureza simples dos dispositivos BLE, os mecanismos de localização empregados se resumem na utilização do RSSI como forma de estimativa de distância e posicionamento. Esta seção apresenta uma breve revisão dos modelos de propagação mais

utilizados na literatura bem como os filtros comumente utilizado para amenizar flutuações nas leituras realizadas.

2.1. Modelos de Propagação

Dentre os modelos de propagação existentes, destacam-se o *Log Distance Path Loss Model* (LDPLM) e *Rappaport's Path Loss Model* (RPLM) [Galván-Tejada et al. 2013, Wang et al. 2013, Dalce et al. 2011].

O LDPLM, consiste em um modelo padrão para desvanecimento de potência de sinal para antenas sem controle de direção ou conhecimento de ganho da antena. Este modelo se mostra amplamente utilizado em ambientes internos e se define pela seguinte equação:

$$RSSI = RSSI_0 - 10 * n * \log_{10}(d), \quad (1)$$

onde $RSSI$ se refere ao valor estimado para RSSI, $RSSI_0$ o valor para o RSSI a 1m de distância do transmissor, n coeficiente de perda de percurso, para ambientes livres $n = 2$, e d a distância. Para calcular a distância basta isolar d :

$$d = 10^{\frac{RSSI_0 - RSSI}{10 * n}}. \quad (2)$$

O modelo de propagação de Rapaport consiste uma evolução ao modelo de propagação LDPLM e visa incluir o fenômeno *Slow-fading*, que ocorre devido a obstruções no caminho [Vargas 2016]. O termo X , assim como n , coeficiente de perda de percurso, objetiva dar melhor controle ao modelo, em geral, utiliza-se a variância como valor para o termo X , conforme definido abaixo:

$$RSSI = RSSI_0 - 10 * n * \log_{10}(d) + X. \quad (3)$$

Para calcular a distância basta isolar d , assim como no modelo anterior:

$$d = 10^{\frac{RSSI_0 - RSSI + X}{10 * n}}. \quad (4)$$

2.2. Filtros

Devido a inconsistência entre as leituras no dispositivo receptor causadas por fatores externos, como sombreamento, multipercurso, desvanecimento do sinal entre outros, necessita-se utilizar técnicas de filtragem, que objetivam atenuar as flutuações do sinal. Dentre as técnicas existentes, destacam-se Média Móvel (MM), Filtro de Kalman (K) [Welch and Bishop 2001] e Filtro Duplo (do Inglês, *Double Filter* (DF)) [Viswanathan and Srinivasan 2015].

A MM utiliza o conceito de média simples somado à movimentação de uma janela de dados, assim ajustando o tamanho da janela pode-se fazer uma média de acordo com a chegada de valores. Portanto:

$$M = \frac{\sum a_i}{k}, \quad (5)$$

onde k representa o tamanho da janela, e i vai de 1 a k .

O filtro de Kalman se baseia em um estimador de estados que presume o valor de uma variável a partir de entradas com ruído, expressado por um algoritmo que se fundamenta no histórico das entradas. Além disso, o filtro de Kalman assume um modelo linear,

portanto os estados e as estimativas devem ser transformações lineares. Define-se o estado x_t como uma combinação do estado anterior x_{t-1} dado uma matriz de transformação A , um controle u_t junto também de uma matriz de transformação B e o ruído processado, proveniente do próprio sistema E . Seguiu-se a ideia [Bulten 2015], de desconsiderar a variável de controle u e utilizar A como matriz identidade, tornando o modelo mais simples:

$$x_t = A_t x_{t-1} + B_t u_t + E_t \approx x_{t-1} + E_t. \quad (6)$$

Faz-se necessária a definição de como um estado X resulta em uma medição z .

$$Z_t = C_t x_t + \delta_t, \quad (7)$$

onde C representa a matriz de transformação e δ o ruído. A etapa de atualização do filtro ocorre pela predição, da seguinte forma:

$$\bar{\mu}_t = \mu_{t-1} \quad (8)$$

$$\bar{\Sigma}_t = \Sigma_{t-1} + R_t. \quad (9)$$

Note que μ denota a predição e X o estado real, já Σ define a certeza da predição e R o ruído causado pelo sistema, pode-se assumir que devido ao ambiente estático o ruído esperado procede, quase exclusivamente, do ambiente. Em ambiente dinâmico, onde há o movimento do receptor, deve-se aumentar o valor desta variável. A partir da etapa de predição computa-se o ganho de Kalman:

$$K_t = \bar{\Sigma}_t (\bar{\Sigma}_t Q_t)^{-1}. \quad (10)$$

Onde Q representa o ruído da informação. Para modelagem sobre RSSI, a variância simboliza o valor de influência. Assim, o ganho juntamente a Q serve como peso para avaliar a certeza da predição contra a certeza da medição. O que torna a etapa de atualização completa por:

$$\mu_t = \bar{\mu}_t + K_t (z_t - \bar{\mu}_t) \quad (11)$$

$$\Sigma_t = \bar{\Sigma}_t - (K_t \bar{\Sigma}_t), \quad (12)$$

desta forma, quanto maior o ganho K mais a medição se torna integrada ao sistema, e quanto menor este ganho mais a predição se torna integrada.

O *Double Filter* caracteriza-se pelo uso de duas variáveis de controle A e B [Viswanathan and Srinivasan 2015]. A variável A controla o ganho do sinal, sendo um valor de 0,5 a 1, simbolizando as porcentagens de ganho, enquanto B representa um valor de controle para o erro propagado no sistema, variando de 0,05 a 0,5, conforme definido abaixo:

$$S_n = (1 - A_n) S_{n-1} + A_n (E(S_n) + (RSSI_n - E(S_n)) B_n) \quad (13)$$

Onde S_n simboliza o novo valor do sinal filtrado, S_{n-1} , o valor anterior do sinal filtrado e $E(S_n)$, o valor de erro do sistema até o momento calculado através de uma média simples dos valores de RSSI filtrados de 0 a $n - 1$. A e B constituem vetores atualizados por:

$$Sel((A_n > A_B) \& \& (B_n > B_b)) := A_n = W_n A_{n-1}, B_n = W_n B_{n-1} \quad (14)$$

Onde A_B , B_B e W_n representam valores de limitação, dados como 0,7, 0,05 e 0,9, respectivamente, de acordo com o estudo.

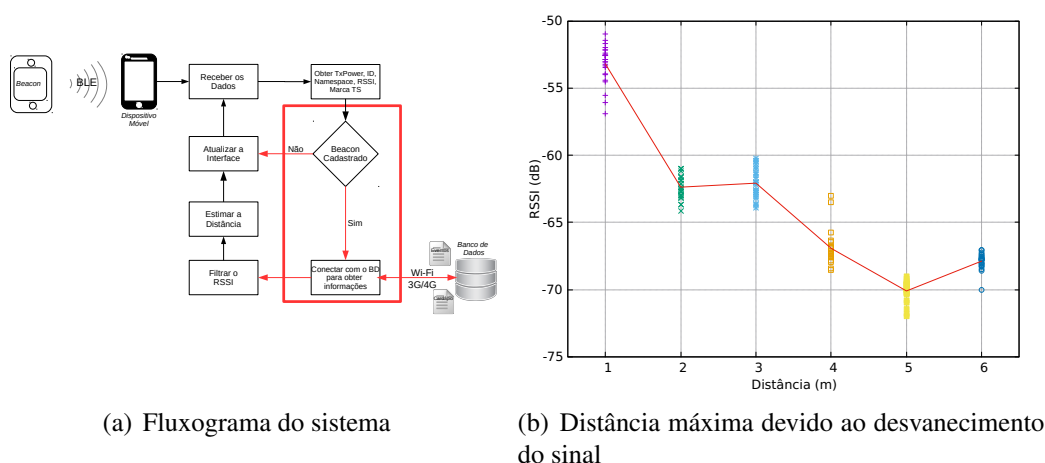


Figure 2. Distância Máxima e Fluxograma

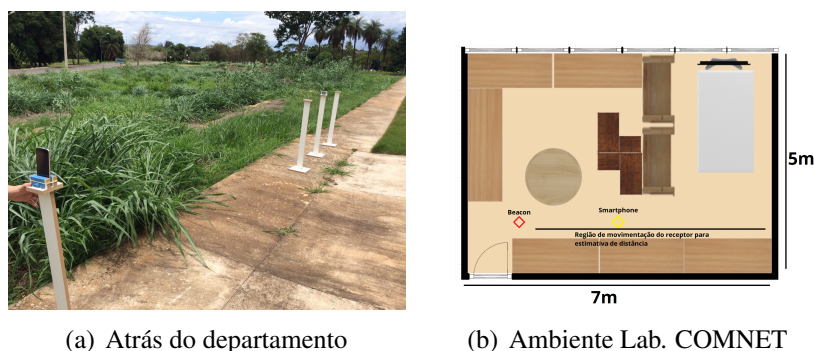
3. Ambiente Experimental

Esta seção apresenta a metodologia utilizada para a coleta de dados, bem como o *hardware* e o *software* necessários para obter-se às informações desejadas. Para avaliação da estimativa de distância com base no RSSI dos dispositivos *beacon* BLE, escolheu-se dispositivos que oferecessem suporte às tecnologias necessárias ao estudo. Ressalta-se que durante este estudo ainda não existiam *beacons* oferecidos no mercado nacional, assim, optou-se pela importação de dois modelos de *beacons*: o Ion e Particle, ambos da empresa KST [KST]. Como dispositivo receptor, utilizou-se um *smartphone* que atendesse as especificações quanto à versão *Bluetooth* (BLE) e acesso à última API *Android* compatível com o Eddystone [Google 2016], Lenovo Moto-Z XT1635-02. Para a coleta das informações, desenvolveu-se uma aplicação para o *Android* que permitisse obter as informações do *beacon* bem como estimar a potência do sinal (RSSI).

O *beacon* envia mensagens encapsuladas com protocolo Eddystone por meio do sinal BLE, capturado pelo dispositivo móvel na interface *Android*, passando pelos módulos *parser*, filtragem e estimativa de distância e finalizando com a atualização da interface, como ilustrado na Figura 2(a).

- **Parser:** A informação passa por um *parser* onde aplica-se uma máscara de *bits*, para a extração das informações necessárias, a *Identification* e a *Namespace*. Já, o RSSI e o *Power* possuem métodos para sua obtenção por meio da API *Android*.
- **Filtragem:** O RSSI passa por filtros com o intuito de reduzir o ruído do sinal.
- **Estimativa de distância:** A partir da filtragem, calcula-se a distância com base no modelo RPLM, pois mostrou-se melhor em relação aos demais.

Os testes iniciais foram projetados com base nos problemas experienciados nos fundamentos e nos artigos abordados nas seções anteriores. Em todo os experimentos seguiu-se um padrão, sendo realizadas ao menos 25 leituras, e os que utilizam distância, começa-se em 1m e termina-se em 6m. Os valores de 6 metros em diante não se revelaram tão úteis quanto valores anteriores a esta marca, como demonstrado na Figura 2(b), devido a natureza dos dispositivos transmissores e receptores, que para manter um consumo energético menor, seu *hardware*, no caso, as antenas, possuem menor capacidade de operação, não distinguindo eficientemente as variações que ocorrem após os 5m,



(a) Atrás do departamento

(b) Ambiente Lab. COMNET

Figure 3. Experimento de desvanecimento de RSSI e estimativa de distância

resultando em maior erro após esse limite. Ainda na Figura 2(b), pode-se observar um padrão onde há um desvanecimento da intensidade do sinal à medida que se aumenta a distância. Fato que não ocorre do quinto metro em diante, limitando a estimativa útil até os 5m. As opções de configuração de tempo para envio de anúncios não surtiram efeitos, posto que a mudança do intervalo de transmissão de 1s para 100ms não gerou leituras proporcionais, mantendo-se em torno dos 1,2s por leitura. Isso pode ocorrer devido a falha no *hardware* dos *beacons* ou à falta de opção de configuração da taxa de leitura na API *Android* para BLE. Duas áreas para os testes foram selecionadas, ambas na Universidade de Brasília (UnB), uma com bom campo de visão, situada atrás do departamento de Ciência da Computação e outra no laboratório COMNET. As Figuras 3(a) e 3(b) denotam os locais dos testes.

4. Resultados

Nesta seção, apresenta-se os resultados referentes à estimativa de distância, utilizando-se das técnicas explicadas na seção anterior. Em particular, esta seção visa mensurar a acurácia da estimativa e o tempo de convergência, a fim de serem constatados os valores que possibilitem o desenvolvimento de aplicações baseadas em estimativa de distância a partir do RSSI com a tecnologia BLE.

4.1. Flutuação da Intensidade do Sinal

O indicador de potência de sinal, RSSI, apresentou grande variação ao longo do experimento, mesmo em ambiente estático, sem movimento da fonte ou do receptor. Para um *beacon* colocado a um metro e aumentando a distância a cada metro do receptor, sendo realizadas 25 leituras por metro, pôde-se observar variância média de 7,43dB para o modelo Ion. Na Figura 4, observa-se a oscilação entre as leituras, o que evidencia a necessidade de técnicas de filtragem.

O valor de RSSI foi utilizado como forma de estimativa de distância, portanto, a filtragem tornou-se necessária devido a variação no sinal, causada por efeitos físicos como reflexo do sinal, multipercursos, sombreamento, desvanecimento da intensidade entre outros.

Com relação a filtragem pela Média Móvel, tornou-se possível obter a média de acordo com as leituras, o tamanho adotado para a janela foi de 25, pois necessita-se de ao menos 20 para garantir confiabilidade a informação. Nota-se diminuição da variância, porém a convergência para novos valores se torna mais lenta.

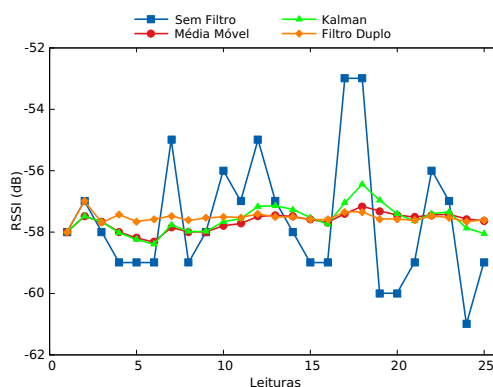


Figure 4. RSSI ao longo do tempo observado a 1m de distância

Table 1. Comparação entre os modelos LDPLM e RPLM

RSSI (dB)	LDPLM (m)	RPLM (m)	Distância Real (m)
-57,64	0,99	1,11	1,00
-64,20	1,80	2,00	2,00
-69,40	2,88	3,21	3,00
-70,96	3,31	3,69	4,00
-75,04	4,79	5,33	5,00
Erro Médio (m)	0,24	0,19	0,00

Já, para o Filtro de Kalman, como explicado anteriormente, notou-se menor discrepância entre os valores de leitura, o que resultou em melhor precisão no recebimento de informação referente ao RSSI e por consequência na estimativa de distância.

Por fim, implementou-se uma versão do filtro duplo, porém os resultados não se mostraram interessantes em ambientes dinâmicos quando comparados com ambientes estáticos, pois o filtro em questão depende dos primeiros valores para convergir a predição, além disso se mostra resistente a variações bruscas no sinal, raramente divergindo de sua predição inicial.

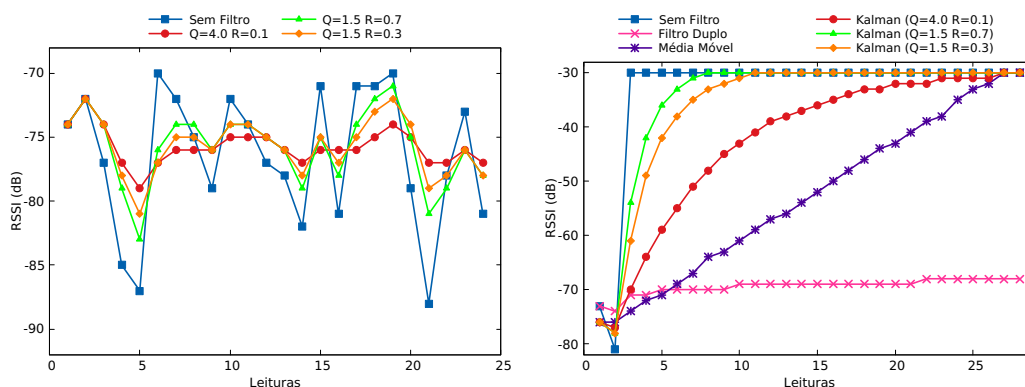
Os resultados dos três filtros podem ser constatados na Figura 4 e nota-se grande vantagem em relação à variação não filtrada. A média móvel se apresenta melhor que o filtro de Kalman, porém manifesta um problema semelhante ao filtro Duplo, no qual se faz necessário maior número de leituras para convergir a um novo valor de RSSI, piorando a precisão em ambientes dinâmicos.

4.2. Calibração dos Modelos de Propagação

Para a calibração do modelo LDPLM e RPLM, necessitou-se encontrar o valor de $RSSI_0$, para tal realizou-se leituras e obteve-se a média, sendo esta de $-57,70\text{dB}$ no modelo Ion e $-53,20\text{dB}$ no modelo Particle. Já para o valor X , utilizou-se a média das variâncias para o filtro de Kalman, obtendo-se $1,22\text{dB}$. Por fim, para o coeficiente de perda do sinal n , obteve-se, empiricamente, o valor de $2,55$. O cálculo da distância para estes parâmetros encontram-se na Tabela 1, o modelo de Friis não se comportaria bem para os ambientes internos.

4.3. Calibração da Convergência de RSSI

Um problema enfrentado pelo usuário, ao utilizar aplicações que contenham técnicas de filtragem, ocorre pela demora na convergência dos valores de RSSI, posto



(a) Variância no filtro Kalman devido a mudança de valores iniciais (b) Quantidade de leituras para convergência nos filtros

Figure 5. Variância e convergência no filtro de Kalman

que se esse tempo for muito grande, a pessoa perderá o interesse no *software* e recorrerá a outros métodos para a localização do objeto físico.

Modificando os valores iniciais do filtro de Kalman, pôde-se chegar, rapidamente, à convergência, porém isto causa efeitos colaterais na variância como consta na Figura 5(a). Portanto, optou-se por utilizar uma convergência em equilíbrio com o seu efeito colateral, para que fosse possível obter um novo valor de convergência, sem afetar de forma desfavorável a estimativa de distância, sendo estes valores $Q = 1,5$ e $R = 0,7$, como expressado na Figura 5(b). Os valores convergiram por volta de 15 leituras, contra mais de 25 do modelo inicial, onde cada leitura ocorreu em torno de 1,2s.

O filtro Duplo e a Média Móvel possuem grande rigidez na convergência, por isso garantem alta precisão na estimativa, porém voltam para o problema da espera do usuário. O filtro Duplo não reagiu conforme o estudo original [Viswanathan and Srinivasan 2015] mesmo modificando os valores de controle, mas se manteve bom em ambientes estáticos. Já, a Média Móvel necessita de uma espera do tamanho de sua janela em questão de quantidade de leituras.

4.4. Estimativa de Distância

A partir dos resultados com filtragem sobre o valor do RSSI, pôde-se estimar a distância de maneira mais confiável devido a baixa flutuação de valores e melhor convergência. Para tal, utilizou-se a estimação pelo método de *Rappaport* de desvanecimento do sinal a partir do isolamento da variável de distância como explicado previamente.

O melhor erro médio foi de 0,12m para o Filtro Duplo no modelo Ion de *beacon*, valor aceitável quando comparado a erros encontrados na literatura. Porém, o Filtro Duplo, assim como a Média Móvel possuem alta rigidez quanto a convergência para novos valores. A partir do erro de estimativa da distância, calculou-se a taxa de melhora em relação ao teste Sem Filtro (SF), onde a Média Móvel obteve 18%, Kalman 11% e o Filtro Duplo 58%.

Considerando somente a Tabela 2, pode-se inferir que o valor sem filtragem foi apropriado, porém, vale lembrar que os valores apresentados passaram por uma média aritmética devido a variação de seus índices, o que faz com que valores sem filtragem se

Table 2. Estimativa de distância - Ion

Distância Real (m)	Sem Filtro	Média Móvel	Kalman	Filtro Duplo
1,00	1,11	1,11	1,10	1,10
2,00	2,00	1,83	1,90	1,74
3,00	3,21	3,06	3,08	2,81
4,00	3,69	3,73	3,70	4,00
5,00	5,33	5,21	5,30	4,92
Erro Médio (m)	0,19	0,16	0,17	0,12

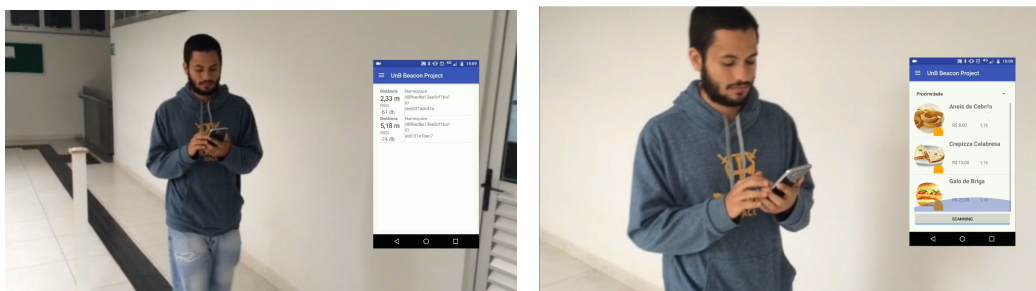
Table 3. Variação e desvio padrão para cada metro e filtragem

Distância (m)	Ion							
	Variância				Desvio Padrão			
	SF	MM	K	DF	SF	MM	K	DF
1,00	4,24	0,08	0,20	0,03	2,06	0,29	0,44	0,17
2,00	12,50	0,57	1,96	0,07	3,54	0,75	1,40	0,26
3,00	8,67	2,67	3,04	0,25	2,94	1,63	1,74	0,42
4,00	5,37	0,15	0,29	0,38	2,32	0,39	0,54	0,62
5,00	6,37	0,34	0,61	0,05	2,52	0,58	0,78	0,22
Média	7,43	0,76	1,22	0,15	2,68	0,73	0,98	0,34

comportem de maneira similar a Média Móvel, a variação se torna evidente na Tabela 3, obteve-se média da variância de 7,43db num cenário sem filtragem.

O filtro de Kalman se mostrou inferior em relação aos outros, mas com erro ainda satisfatório e, além disso, possui bom controle sobre a flexibilidade da filtragem, podendo assim se tornar mais rígido com valores de estimativa semelhantes a seus concorrentes. Ao se considerar a movimentação do usuário, mostrou-se com melhor convergência, resultando em maior confiabilidade da estimativa, pois os demais não acompanharam os novos valores de RSSI, aumentando os seus erros gradativamente ao movimento.

Realizou-se um experimento real em que o usuário recebe informações sobre produtos de um *Food Truck* fictício com base na proximidade ao *beacon*. O resultado obtido está em consonância com o apresentado nas seções anteriores. As Figuras 6(a) e 6(b) ilustram o sistema do caso de uso onde o usuário recebe informações do cardápio em conformidade com a sua proximidade do *beacon*.



(a) Encontrando o sinal e estimativa de distância (b) Identificando o sinal e mostrando os produtos de um *Food Truck* em proximidade

Figure 6. Sistema recebendo informações sobre os produtos de um *Food Truck*

5. Conclusão

Com os resultados aqui apresentados, obtendo-se erro médio de 0,17m para a estimativa de distância e em torno de 9 leituras para convergência, argumenta-se ser a intensidade do sinal uma boa opção para localização para ambientes internos, com acurácia relativamente alta, condizendo, pois, com outros estudos já desenvolvidos nesse cenário. Como trabalho futuro, sugere-se a utilização de outras funcionalidades da API *Android*, como giroscópio e acelerômetro, para modificar os valores de controle dos filtros em tempo real de acordo com a movimentação. Assim, após se obter uma estimativa satisfatória e o usuário permanecer parado, pode-se aumentar a precisão filtrando-se os valores de maneira mais rígida, e aumentando-se a flexibilidade quando o usuário se mover.

References

- Bulten, W. (2015). Kalman filters explained: Removing noise from rssi signals. <https://wouterbulten.nl/blog/tech/kalman-filters-explained-removing-noise-from-rssi-signals/>, Acessado em junho 2016.
- Chowdhury, T., Rahman, M. M., Parvez, S.-A., Alam, A., Basher, A., Alam, A., and Rizwan, S. (2015). A multi-step approach for rssi-based distance estimation using smartphones. *IEEE*.
- Dalce, R., Val, T., and Bossche, A. V. D. (2011). Comparison of indoor localization systems based on wireless communications. *Wireless Engineering and Technology*.
- Galván-Tejada, C. E., Carrasco-Jiménez, J. C., and Brenna, R. F. (2013). Bluetooth-wifi based combined positioning algorithm, implementation and experimental evaluation. *The 2013 Iberoamerican Conference on Electronics Engineering and Computer Science*.
- Google. Physical web. <https://github.com/google/physical-web>, acessado em julho 2016.
- Google (2016). Eddystone format. <https://developers.google.com/beacons/eddystone>, Acessado em março 2016.
- KST. Kst. <https://kstechnologies.com/>, Acessado em março 2016.
- Lindh, J. (2016). Bluetooth low energy beacons application report. *Texas Instruments*.
- Tanenbaum and Wheterall (2011). *Computer Networks*. Pearson.
- Townsend, K., Cufi, C., Akiba, and Davidson, R. (2014). *Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power Networking*. O'REILLY.
- Vargas, M. H. (2016). Indoor navigation using bluetooth low energy beacons. *Turkey University*.
- Viswanathan, S. and Srinivasan, S. (2015). Improved path loss prediction model for short range indoor positioning using bluetooth low energy. *IEEE*.
- Wang, Y., Yang, X., Zhao, Y., Liu, Y., and Cuthbert, L. (2013). Bluetooth positioning using rssi and triangulation methods. *IEEE Consumer Communications and Networking Conference(CCNC)*, pages 837–842.
- Welch, G. and Bishop, G. (2001). An introduction to the kalman filter. *University of North Carolina at Chapel Hill*.

Uso da Rede MQTT para a Manipulação de Sistemas Robóticos suportados por Realidade Virtual

Leandro R. Mattioli¹, Daniel S. Souza¹, Márcio J. Cunha¹, Alexandre Cardoso¹

¹Faculdade de Engenharia Elétrica
Universidade Federal de Uberlândia (UFU) – Uberlândia – MG – Brasil

{leandro.mattioli, danielss, mjcunha, alexandre}@ufu.br

Abstract. *Publisher/subscriber architectures allow loose coupling between the communicating parts, keeping as dependency only the transmitted messages format. This feature is quite relevant for the development of robotic systems, since they risk premature obsolescence for not being able to keep up with new technologies and algorithms. In this scenario, the MQTT protocol presents itself as a good alternative, as seen by the growing adoption in applications concerning Internet of Things. The present work evaluates the performance of the MQTT network for the control of a virtual robotic arm. Preliminary tests indicate that the real-time rendering requirements are not compromised by the adoption of this architecture.*

Resumo. *As arquiteturas do tipo publicador/subscritor permitem o desacoplamento entre as partes comunicantes, deixando como dependência apenas o formato das mensagens transmitidas. Esta característica é bastante relevante para o desenvolvimento de sistemas robóticos, sob pena de obsolescência prematura de tais sistemas por não acompanhar novas tecnologias e algoritmos. Neste cenário, o protocolo MQTT se destaca pela crescente utilização em aplicações que envolvem Internet das Coisas. O presente trabalho avalia o desempenho desta rede para o controle das juntas de um braço robótico virtual. Testes preliminares apontaram que os requisitos de renderização em tempo real não são comprometidos pela adoção desta arquitetura.*

1. Introdução

Dentre o conjunto de requisitos esperados para frameworks de sistemas robóticos, seja para fins de simulação, teleoperação ou execução de missões, destaca-se a modularidade [Limsoonthrakul et al. 2009] [Casse et al. 2009] [Poppa and Zimmer 2012]. Tal princípio, quando não atendido, pode resultar na rápida obsolescência do sistema, uma vez que as novas tecnologias para sensores, atuadores, além daquelas direcionadas para interação humano-computador, não podem ser incorporadas em soluções monolíticas.

Nesse sentido, o protocolo MQTT e as arquiteturas publicador/subscritor em geral, provêm significativa flexibilidade no desenvolvimento destes módulos, notavelmente devido a: (i) transparência, considerando que o robô não necessita conhecer a natureza dos módulos que com ele interagem e (ii) evolutividade, uma vez que novos módulos podem ser agregados dinamicamente ao sistema e (iii) comutatividade, sabendo que, analogamente ao caso do robô, os outros módulos não precisam diferenciar se trata-se de um robô real ou virtual.

Assim, este trabalho trata da aplicação do protocolo MQTT no controle das juntas de um manipulador serial concebido em um ambiente virtual não imersivo.

2. MQTT

O protocolo foi criado pela IBM no final da década de 90 com o foco em sistemas de supervisão e coleta de dados do tipo SCADA (*Supervisory Control and Data Acquisition*) [Lampkin et al. 2012] [Aziz 2014]. Um fato importante para se destacar é a crescente utilização desse protocolo em aplicações que envolvem Internet das Coisas (IoT – *Internet of Things*), devido a sua definição de qualidade de serviço, especificação para o uso com segurança, simples implementação, além de garantir o uso moderado de banda, embora não seja considerado um protocolo bem elaborado, como o AMQP (*Advanced Message Queuing Protocol*).

O padrão adotado no MQTT para troca de mensagens é o *publish/subscriber* (publicador/subscritor). Neste tipo de padrão, se um determinado nó da rede precisa receber uma informação, ele faz a subscrição dessa informação, enviando essa requisição para o elemento responsável de gerenciar as mensagens de publicação e subscrição contida na rede. Na rede MQTT, esse elemento é o *broker*, responsável pela intermediação contida no processo de comunicação. Os outros nós na rede que desejam publicar suas informações enviam mensagens de publicação também para o *broker*.

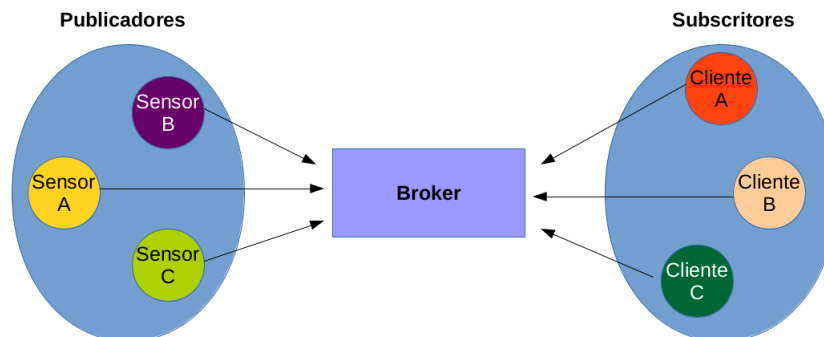


Figura 1. Visão do padrão MQTT (Adaptado de ALMEIDA et. al, 2016)

Segundo [Cunha et al. 2016], o *broker* pode representar um item de fragilidade na rede, no momento em que a comunicação é centralizada. Porém, ele permite o desacoplamento entre as partes comunicantes, algo que não é possível em modelos de comunicação do tipo cliente/servidor.

As mensagens no MQTT são definidas por meio de tópicos (*topics*). A ideia do conceito do tópico é de lembrar uma URL, tendo os seus níveis separados por barras (“/”), uma vez que, os elementos da rede podem enviar vários tópicos para o *broker* e os subscritores podem escolher quais os tópicos eles podem subscrever.

Pensando em uma situação de um robô industrial, pode-se ter um tópico em que seja indicado que o robô chamado r_1 já alcançou um determinado objetivo (*set point*):

robot/r1/busy

Outro exemplo é um tópico para acessar o nome de um objeto com o qual o robô está interagindo:

```
robot/r1/grabbedobj
```

No MQTT o símbolo + possui uma função de “curinga”, o que faz que qualquer valor seja aceito naquele nível do tópico. Por exemplo, para assinar as informações do dicionário de informações que estão disponíveis somente para acesso:

```
robot/r1/+/get
```

Outra notação importante é o símbolo #, que indica todas as informações que estão abaixo de um determinado nível serão acessadas. Por exemplo, pode-se assinar todas as informações do robô que podem ser acessadas e editadas:

```
robot/r1/vars/set/#
```

O tipo de conexão feita do cliente (publicador/subscritor) MQTT com o *broker* é feita via TCP, contendo informações de login, onde são definidos os usuários e senhas, criptografia (SSL/TLS), além de dispositivos que utilizam de outros meios físicos, como por exemplo, aqueles que utilizam links de comunicação serial. Além disso, todo o processo de conexão estabelece também um nível de qualidade de serviço (QoS – *Quality of Service*), que pode ser utilizado de acordo com a necessidade da aplicação. Esse tipo de característica indica qual a relação que existe entre os elementos comunicantes da rede. No MQTT existem 3 níveis de qualidade de serviço, o QoS 0 que é o que possui o menor esforço, e que possui características semelhantes do protocolo UDP, onde não são exigidas confirmações no momento em que a mensagem é entregue. O outro tipo é o QoS 1, onde já se tem um tipo de confirmação de entrega de mensagem, porém, várias mensagens iguais são geradas, mas apenas uma terá o reconhecimento de chegada. E por fim, o QoS 2 que garante que a mensagem seja entregue exatamente uma vez.

2.1. Objetivos

Este trabalho objetiva avaliar quantitativamente a proposta de adoção do protocolo MQTT para a interação com um robô virtual. Nesse sentido, deve ser verificado se a adição de uma arquitetura do tipo publicador/subscritor pode comprometer a renderização em tempo real, imprescindível em sistemas de Realidade Virtual.

3. Trabalhos Correlatos

O uso de arquiteturas modulares baseadas no padrão publicador/subscritor já foi avaliado em sistemas de robótica móvel, com foco particular à integração de sensores para georreferenciamento [Limsoonthrakul et al. 2009]. No entanto, tais sistemas foram avaliados apenas no sistema de fila de mensagens POSIX, além de incorporar compartilhamento de dados por meio de um *blackboard*.

Para veículos aéreos autônomos, a solução AEROSTACK [Sanchez-Lopez et al. 2016] foi aplicada com sucesso em simulações de missões de procura e resgate.

Ainda no campo de robótica móvel, o ambiente Webots [Lancheros et al. 2016] foi usado na simulação de um sistema robótico modular. Mesmo permitindo o uso de

plug-ins codificados em C++, a ferramenta carece do nível de flexibilidade do MQTT: os plug-ins não podem ser espalhados na rede, de forma transparente ao robô.

Para o desenvolvimento específico das interfaces gráficas de sistemas robóticos, um *framework* modular tratado em [Poppa and Zimmer 2012]. Observa-se, para o caso da presente proposta, que é possível incorporar à arquitetura baseada em MQTT tópicos específicos para as ações da interface gráfica do ambiente virtual. A título de exemplo, pode-se imaginar o controle do nível de transparência da malha geométrica do robô, a fim de visualizar objetos que, de outra forma, ficariam invisíveis devido à oclusão.

Soluções baseadas em redes industriais fieldbus também já foram analisadas, como o URB (*Universal Robot Bus*) [Saranli et al. 2011].

4. Robô Virtual

O robô usado nesta pesquisa é composto por servomotores de tamanho padrão (MG996R), bem como um conjunto de peças comercialmente disponíveis como rolamentos, suportes e a garra.

Para construir o modelo geométrico do robô virtual, as partes individuais foram modeladas dentro do ambiente Blender [Blender], à exceção do servomotor, para o qual um modelo gratuito disponível na internet foi usado. O manipulador como um todo foi concebido dentro do ambiente Unity 3D [Unity], por meio de translações, rotações e agrupamentos em conformidade com a configuração mecânica desejada. Além disso, os pontos de pivot foram relocados para ficar em conformidade com as posições dos eixos de rotação. Finalmente, a hierarquia de objetos foi convenientemente ajustada de modo que, quando uma dada rotação é aplicada a um eixo, todos os objetos filhos são automaticamente rotacionados pela mesma quantidade, em harmonia com o comportamento de manipuladores seriais.

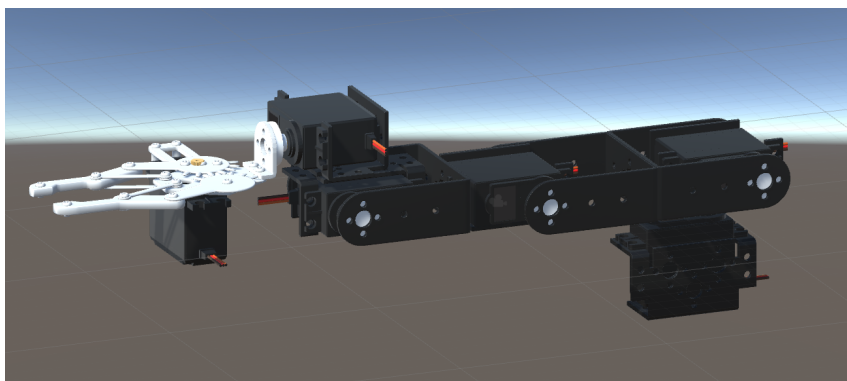


Figura 2. Modelo Geométrico do Robô Virtual

Para fins de conferência, o robô foi manipulado por um *joystick* sem o uso do protocolo MQTT, possibilitando o ajuste da noção de ângulo nulo para cada junta.

5. Método para Teste da Solução

Para avaliar o desempenho da rede MQTT, foi desenvolvido um sistema composto por apenas um publicador e um subscritor: o robô virtual atualiza seu vetor de juntas conforme

os dados publicados periodicamente por um módulo que captura o estado dos eixos de um *joystick* (Figura 3). O *broker* usado foi o Mosquitto [Mosquitto], pacote de código aberto para esta finalidade.

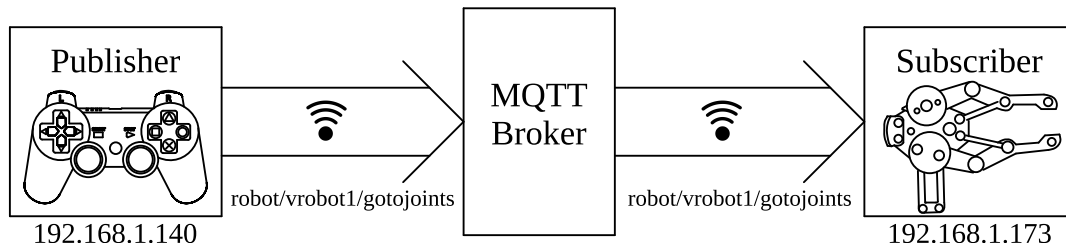


Figura 3. Sistema para Teste de Desempenho da Rede

A mensagem presente no tópico *robot/vrobot1/gotojoints* conta com um campo adicional, usado para armazenar o valor de um contador de pacotes, incrementado exclusivamente pelo publicador após cada publicação.

Imediatamente antes de enviar os valores provenientes do *joystick* para o *broker*, o publicador adiciona, em um arquivo do tipo CSV (*Comma-Separated Values*), uma linha contendo (i) o valor atual do contador de pacotes e (ii) o *timestamp* em milissegundos. O formato CSV foi usado pela simplicidade de leitura e escrita.

O subscritor, por sua vez, acrescenta em outro arquivo uma linha no mesmo formato, contemplando (i) o número do pacote, tal como recebido do *broker* e (ii) o *timestamp* local em milissegundos.

Para reduzir discrepâncias devido a uma sincronização de relógios precária, ambas estações têm seu relógio ajustado por um mesmo servidor NTP [NTP].

Os relatórios gerados permitem, a princípio, duas análises quantitativas: a taxa de transmissão (a quantidade de pacotes MQTT por unidade de tempo) e a latência da rede (variação temporal entre os instantes registrados pelo publicador e pelo subscritor, para cada identificador de pacote).

Para um dado conjunto de linhas de qualquer um dos relatórios CSV gerados, a taxa de transmissão é dada por:

$$PTR = \frac{t_{Final} - t_{Inicial}}{id_{Final} - id_{Inicial}}, \quad (1)$$

onde *PTR* é a taxa de transmissão dos pacotes, *t* é o *timestamp* em milissegundos e *id* é o identificador de pacote.

A latência de cada pacote numerado, em contrapartida, é medida com:

$$\Delta t(k) = t_{Subscriber}(k) - t_{Publisher}(k), \quad (2)$$

onde $t_{Publisher}$ é o instante de tempo registrado antes do envio do *k*-ésimo pacote e $t_{Subscriber}$ é o *timestamp* registrado após a atualização do robô com os dados deste mesmo pacote.

6. Resultados

Foram gerados 1584 pacotes, enviados num período de aproximadamente 26 ms . Tanto o publicador como o subscriber registraram taxas de transmissão de pacotes similares, especificamente $PTR_{Publisher} = 16.6747$ ms e $PTR_{Subscriber} = 16.6525$ ms.

A taxa obtida, de aproximadamente 60 atualizações por segundo, vai de acordo com o tradicionalmente aceito para simulações realísticas de movimento. No entanto, trabalha-se com a hipótese da própria *engine* de renderização ter comprometido a taxa máxima de transmissão, uma vez que os eventos do *joystick* e o processamento do novo vetor de juntas é feito segundo o laço de eventos da própria *engine*.

A característica linear da transmissão dos pacotes (Figura 4) sugere que, nas circunstâncias adotadas, a rede não sofreu sobrecarga.

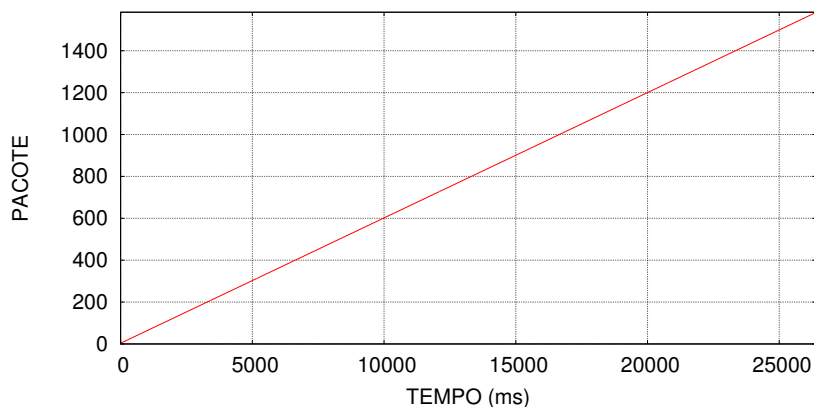


Figura 4. Identificadores de pacotes no tempo

A sincronização via NTP pela Internet, neste caso, se mostrou ineficaz para as medidas de latência, tendo em vista que, para alguns pacotes, obteve-se $t_{Subscriber} \leq t_{Publisher}$. Assim, considerou-se apenas o módulo da latência calculada, resultando na média $\overline{\Delta t} = 15.469$ ms, com desvio padrão $\sigma = 10.73$.

O comportamento da latência no tempo é apresentado na Figura 5.

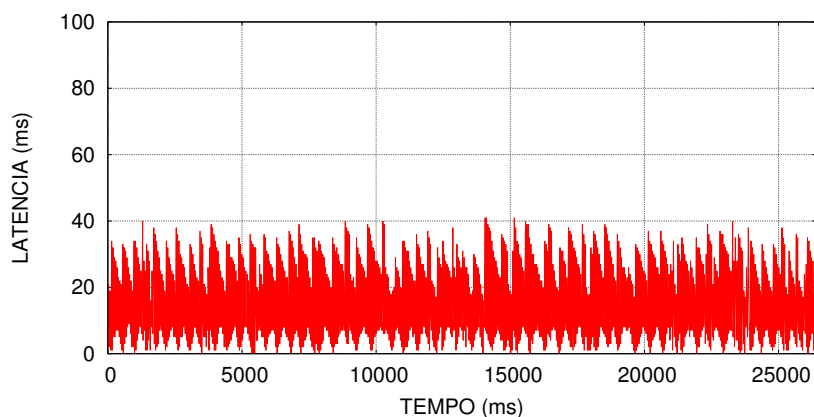


Figura 5. Latência no tempo

7. Conclusão

Os testes preliminares parecem ir de acordo à aplicação do protocolo MQTT para a comunicação entre módulos de um sistema de robótica suportado por técnicas de Realidade Virtual. Não foi observado nenhum atraso na renderização do mecanismo, e a fluidez obtida no controle pelo *joystick* foi equiparável à da solução local (dispositivo diretamente integrado ao ambiente virtual).

Trabalhos futuros incluem a utilização de mecanismos mais sofisticados de sincronização de tempo, tais como aqueles baseados em sinais de GPS (*Global Positioning System*), possibilitando uma medida mais precisa da latência da rede. Além disso, é pertinente a avaliação do desempenho da solução com outros módulos, em particular, após a integração com o robô real. Finalmente, uma análise comparativa com outros protocolos de rede é desejável, avaliando o desempenho em várias condições de rede, como latência e perda de pacotes.

Referências

- Aziz, B. (2014). A formal model and analysis of the mq telemetry transport protocol. In *2014 Ninth International Conference on Availability, Reliability and Security*, pages 59–68.
- Blender. Blender – Free and Open 3D Creation Software. <https://www.blender.org/>. Acessado em: 03/04/2016.
- Casse, K. D., Koay, K. L., Ho, W. C., and Dautenhahn, K. (2009). Reducing the cost of robotics software: Samgar, a generic modular robotic software communication architecture. In *2009 International Conference on Advanced Robotics*, pages 1–6.
- Cunha, M. J., Almeida, M. B., Fernandes, R. F., and Carrijo, R. S. (2016). Proposal for an IoT architecture in industrial processes. In *2016 12th IEEE International Conference on Industry Applications (INDUSCON)*, pages 1–7.
- Lampkin, V., Leong, W., Olivera, L., Rawat, S., Subrahmanyam, N., Xiang, R., Kallas, G., Krishna, N., Fassmann, S., Keen, M., et al. (2012). *Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry*. IBM redbooks. IBM Redbooks.
- Lancheros, P. N., Sanabria, L. B., and Castillo, R. A. (2016). Simulation of modular robotic system mecabot in caterpillar and snake configurations using webots software. In *2016 IEEE Colombian Conference on Robotics and Automation (CCRA)*, pages 1–6.
- Limsoonthrakul, S., Dailey, M. N., Srisupundit, M., Tongphu, S., and Parnichkun, M. (2009). A modular system architecture for autonomous robots based on blackboard and publish-subscribe mechanisms. In *2008 IEEE International Conference on Robotics and Biomimetics*, pages 633–638.
- Mosquitto. Mosquitto – An Open Source MQTT v3.1/v3.1.1 Broker. <https://mosquitto.org/>. Acessado em: 03/04/2016.
- NTP. NTP: The Network Time Protocol. <http://www.ntp.org/>. Acessado em: 03/04/2016.
- Poppa, F. and Zimmer, U. (2012). Robotui - a software architecture for modular robotics user interface frameworks. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2571–2576.

- Sanchez-Lopez, J. L., Fernández, R. A. S., Bavle, H., Sampedro, C., Molina, M., Pestana, J., and Campoy, P. (2016). Aerostack: An architecture and open-source software framework for aerial robotics. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 332–341.
- Saranli, U., Avci, A., and Ozturk, M. C. (2011). A modular real-time fieldbus architecture for mobile robotic platforms. *IEEE Transactions on Instrumentation and Measurement*, 60(3):916–927.
- Unity. Unity – Game Engine. <https://unity3d.com/>. Acessado em: 03/04/2016.

Realização



Apoio Fomento



Apoio Institucional



Patrocinador Diamante



Patrocinador Ouro



Patrocinador Bronze

