



DINCON 2017

CONFERÊNCIA BRASILEIRA DE DINÂMICA, CONTROLE E APLICAÇÕES

30 de outubro a 01 de novembro de 2017 – São José do Rio Preto/SP

Sistema de Reconhecimento de Atividades Humanas implementado no processador Nios II

Willian de Assis Pedrobon Ferreira¹

Departamento de Engenharia Elétrica, UNESP, Ilha Solteira, SP

Melisa de Jesús Barrera Durango²

Departamento de Ingeniería Eléctrica, Universidad de Antioquia, Medellín, Colombia

Alexandre Araujo Amaral de Almeida³

Departamento de Engenharia Elétrica, UNESP, Ilha Solteira, SP

Alexandre César Rodrigues da Silva⁴

Departamento de Engenharia Elétrica, UNESP, Ilha Solteira, SP

Resumo. No reconhecimento de atividades humanas, a aplicação de sensores e algoritmos de classificação permitem identificar atividades corporais realizadas em diversas situações. Na abordagem *online*, as classificações são obtidas em tempo de execução, possibilitando o monitoramento constante e até mesmo remoto. Neste trabalho, o Sistema de Reconhecimento de Atividades Humanas (SIRAH), que utiliza dados de um acelerômetro localizado na cintura do usuário e uma Rede Neural Artificial para classificar atividades humanas, foi implementado no processador Nios II da Altera, gerando um sistema embarcado que realiza classificações durante o desenvolvimento das atividades analisadas.

Palavras-chave. Reconhecimento de atividades humanas, Acelerômetro, Redes neurais artificiais, Sistemas embarcados, FPGA

1 Introdução

No passado, a automação se restringia a processos financeiramente custosos, como aplicações industriais e militares. Contudo, a evolução tecnológica tem proporcionado a obtenção de dispositivos eletrônicos mais acessíveis, o que possibilitou a inserção de sistemas autônomos no cotidiano das pessoas. O aumento no interesse de pesquisadores

¹willianferreira51@gmail.com

²melisa.barrera@udea.edu.co

³aamaralalmeida@gmail.com

⁴acrsilva@dee.feis.unesp.br

e maior rentabilidade econômica também tornaram mais frequente o uso de entidades computacionais na rotina do ser humano [9].

Avanços na tecnologia da informação são estudados na Inteligência Ambiental, ou AmI (*Ambient Intelligence*), cujo objetivo é desenvolver ambientes capazes de detectar e interagir às necessidades dos usuários, de modo transparente e não intrusivo [6]. A AmI originou-se principalmente da computação ubíqua [11], associada a outras áreas, como redes de sensores e inteligência artificial (IA). Ultimamente, a Internet das Coisas, ou IoF (*Internet of Things*) também é utilizada na implementação de ambientes inteligentes [7].

Vários serviços são beneficiados aplicando-se técnicas inteligentes de monitoramento, como a segurança, a saúde e os sistemas de manufatura [5]. Outra área importante é o reconhecimento de atividades humanas, ou HAR (*Human Activity Recognition*), pois possibilita analisar o comportamento diário de uma pessoa e identificar situações emergenciais que necessitam de intervenção [12].

Apresenta-se, nesse trabalho, a implementação o Sistema de Reconhecimento de Atividades Humanas (SIRAH) embarcado no processador Nios II da Altera, para realizar classificações *online*. Nessa abordagem, o algoritmo de classificação é executado no próprio dispositivo dedicado, diferentemente da primeira versão proposta por Durango [2], que opera no software MATLAB.

Na Seção 2 aplicações de AmI e HAR são exemplificadas, e o SIRAH é apresentado. Na Seção 3 o desenvolvimento da versão embarcada do SIRAH é detalhada. Os resultados das avaliações são discutidas na Seção 4 e, na Seção 5, apresentam-se conclusões.

2 Revisão da literatura

A detecção de atividades corporais baseia-se na análise das características extraídas dos sinais provenientes de sensores e na aplicação de algoritmos de classificação [1]. Ha e Choi [3] utilizaram uma rede neural convolucional para classificar doze atividades diárias a partir dos dados gerados por acelerômetros, giroscópios e sensores de eletrocardiograma. Tran e Phan [10] classificaram atividades humanas em um *smartphone* aplicando-se o algoritmo de aprendizado de máquina SVM (*Support Vector Machine*), que recebe dados dos sensores do próprio dispositivo.

Joaquinito e Sarmiento [4] utilizaram um FPGA (*Field Programmable Gate Array*) no processamento de sinais da temperatura corporal e dos batimentos cardíacos, e transmitiram os resultados para um *smartphone* via *Bluetooth*. Tang et al. [8] empregaram dispositivos portáteis e sensores de contexto para auxiliar autistas no desenvolvimento de atividades diárias, e também no relacionamento com outras pessoas.

Apresentam-se a seguir as características gerais do SIRAH.

2.1 SIRAH: Sistema de Reconhecimento de Atividades Humanas

O SIRAH foi desenvolvido para monitor atividades humanas realizadas diariamente. O sensor utilizado foi um acelerômetro de três eixos localizado na cintura, cujos dados são armazenados em um cartão SD. Um *Zigbee* é responsável pela transmissão de dados, e as comunicações desses componentes são gerenciadas por um microcontrolador Atmega 328.

O sistema classifica sete atividades, divididas entre estáticas (em pé, deitado e sentado) e dinâmicas (caminhar, correr, sentar e levantar), e foi projetado para analisar o movimento de um único indivíduo por vez. Logo, todas as etapas da classificação *offline* apresentadas na Figura 1 devem ser executadas exclusivamente para cada indivíduo monitorado.

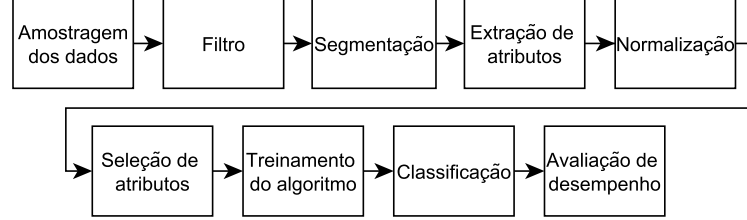


Figura 1: Etapas da classificação *offline* do SIRAH.

Durante a amostragem de dados, cada atividade é executada por 6 minutos e os valores das acelerações são armazenadas no cartão SD. A sensibilidade do acelerômetro foi ajustada em $\pm 8g$, a uma taxa de amostragem de 40 Hz. Nas Figuras 2(a) e 2(b) são exemplificadas, respectivamente, as acelerações das atividades em pé e correndo. As demais etapas foram implementadas no software MATLAB.

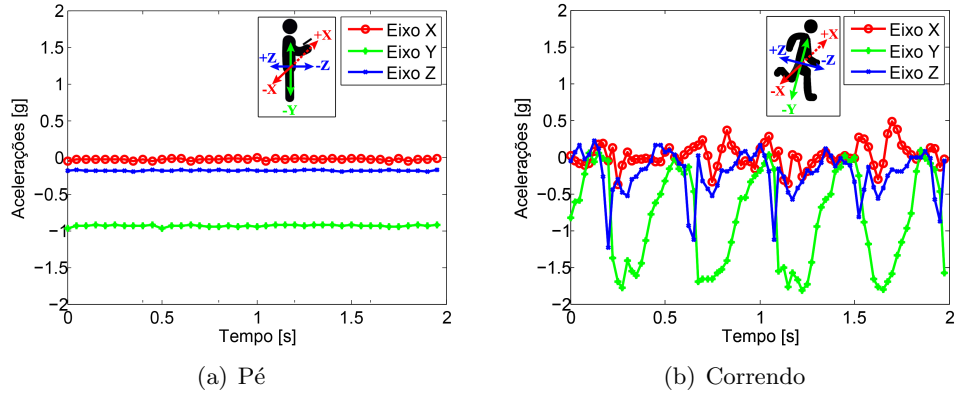


Figura 2: Exemplos de acelerações das atividades estudadas

Utilizou-se um filtro passa baixa de média móvel com três coeficientes para minimizar possíveis ruídos ocorridos na amostragem. Na segmentação, os sinais das acelerações são divididos em janelas adjacentes de tempo com duração de 2 segundos, obtendo-se 180 exemplos de cada atividade, que após a extração de atributos, são utilizados no treinamento da RNA e na avaliação de desempenho.

Foram considerados atributos no domínio do tempo: média, desvio padrão, variância, variação, SMA (*Signal Magnitude Area*), assimetria, curtose, correlação e excentricidade, e no domínio da frequência: energia média, entropia espectral e magnitude máxima. Com exceção da SMA, que gera um atributo para os três eixos do acelerômetro, os demais geram três valores, um para cada eixo, totalizando em 34 atributos, que são normalizados

para que tenham média nula e variância unitária, gerando uma convergência mais rápida no algoritmo de aprendizagem.

As classificações são efetuadas por uma RNA do tipo *feedforward*, com uma camada oculta. O número de neurônios da camada de entrada e da camada oculta é igual à quantidade de atributos considerados e na camada de saída há sete neurônios que, individualmente, representam a probabilidade da classificação pertencer a cada atividade considerada. O treinamento da RNA é realizado pelo algoritmo *backpropagation* implementado no MATLAB.

Nas avaliações, 70% dos exemplos foram utilizados no treinamento e 30% na validação, que são separados de forma aleatória. Três conjuntos de atributos foram verificados: 1º) média, desvio padrão, assimetria, curtose e excentricidade, que alcançaram 95,1% de exatidão; 2º) todos os 34 atributos descritos anteriormente, garantindo 92,8% de acerto; 3º) utilizou-se a técnica de seleção de atributos PCA (*Principal Component Analysis*), obtendo-se 95,7% de acerto considerando 22 atributos.

Dessa forma, o sistema oferece boa exatidão nas classificações *offline*. Para incrementar a aplicabilidade, apresenta-se a seguir a versão embarcada do SIRAH.

3 Implementação do SIRAH em sistema embarcado

A plataforma utilizada foi o processador Nios II da Altera, que pode ser configurado em vários modelos de FPGAs dessa fabricante. Esse dispositivo executa aplicações descritas nas linguagem de alto nível C e C++. Neste trabalho, optou-se pelo uso da linguagem C.

Na versão embarcada os atributos considerados foram os da primeira validação feita por Durango [2], ou seja, média, desvio padrão, assimetria, curtose e excentricidade. Esse conjunto foi escolhido devido aos bons resultados obtidos na abordagem *offline*, e também por empregar apenas atributos no domínio do tempo, tornando o desenvolvimento mais rápido e atrativo para se avaliar a primeira implementação *online* do sistema.

Os componentes de hardware utilizados na configuração do processador Nios II estão apresentados na Figura 3, em que *cpu_0* é o processador Nios II, *jtag_uart_0* é a comunicação entre um microcomputador e a placa DE2-115 que permite a gravação do sistema descrito no FPGA, *sdram_0* é a memória RAM, *uart_0* é a comunicação serial utilizada na leitura das acelerações enviadas pelo *Zigbee* durante a classificação *online* e o *performance_counter_0* foi utilizado nas avaliações de consumo de tempo.

Name	Description
☒ cpu_0	Nios II Processor
instruction_master	Avalon Memory Mapped Master
data_master	Avalon Memory Mapped Master
jtag_debug_module	Avalon Memory Mapped Slave
☒ jtag_uart_0	JTAG UART
avalon_jtag_slave	Avalon Memory Mapped Slave

Name	Description
☒ sdram_0	SDRAM Controller
s1	Avalon Memory Mapped Slave
☒ uart_0	UART (RS-232 Serial Port)
s1	Avalon Memory Mapped Slave
☒ performance_counter_0	Performance Counter Unit
control_slave	Avalon Memory Mapped Slave

Figura 3: Componentes de hardware do sistema embarcado.

A exatidão da classificação *offline* da implementação em C foi verificada em um microcomputador e no sistema descrito no FPGA. No processador Nios II também foi avaliado

o treinamento da RNA e a classificação *online*.

4 Resultados e discussões

Para autenticar a funcionalidade da codificação em C, o mesmo algoritmo da classificação *offline* processado no MATLAB foi compilado no GCC. Durante a execução, geram-se números aleatórios que são utilizados na inicialização dos pesos da RNA e nos sorteios da validação. Os valores criados pelo MATLAB foram utilizados na implementação em C, garantindo a execução do algoritmo com os mesmos parâmetros nas duas descrições. Nesse experimento, ambas versões obtiveram uma taxa de acerto de 94,71%, o que comprovou o desempenho da implementação em C.

Constatou-se que a extração dos atributos de 180 exemplos de cada atividade para realizar o treinamento da RNA, no processador Nios II, consumiu em torno de 240 minutos. Para aprimorar o tempo de execução, diminuiu-se a quantidade de exemplos.

A codificação em C foi executada por duas mil vezes para cada quantidade de exemplos analisada, sempre na proporção de 70% para o treinamento e 30% para a validação. A exatidão das classificações decrementou à medida que menos exemplos de treinamento foram utilizados, como pode ser visualizado na Tabela 1. Com 180 exemplos de cada atividade (126 para o treinamento e 54 para a validação), a exatidão obtida foi de 94,23%, diminuindo para 92,78% empregando-se 22 modelos de cada atividade.

Tabela 1: Avaliação do treinamento da RNA diminuindo-se a quantidade de exemplos.

Número de exemplos de treinamento	Número de exemplos de validação	Média da exatidão das classificações
126	54	94,23%
63	27	93,09%
32	13	93,05%
15	7	92,78%

Utilizando-se 30 exemplos de cada atividade, o treinamento da RNA no Nios II obteve a exatidão de 92,81% e consumiu aproximadamente 40 minutos. A implementação do processador é feita em linguagem de descrição de hardware, porém a arquitetura que executa o software em C não é gerada para aplicações específicas, tornando operações como o acesso à memória RAM e cálculos com ponto flutuante pouco otimizadas. Para obter um tempo de operação viável, manteve-se o treinamento da RNA no microcomputador e no sistema embarcado foi implementado apenas o algoritmo de classificação.

As etapas da classificação *online* executada no processador Nios II estão apresentadas na Figura 4. Os pesos da RNA, as médias e os desvios padrões da normalização obtidos no treinamento efetuado em um microcomputador são definidos como constantes no algoritmo em C. Durante a amostragem executada por 2 segundos, as informações do acelerômetro, localizado na cintura do usuário, são transmitidas ao FPGA por transceptores *Zigbee*.

Para executar o pré-processamento e a classificação de uma janela são necessários,

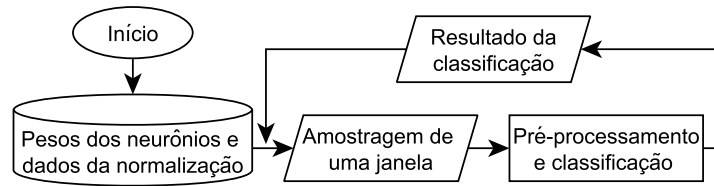


Figura 4: Fluxograma da classificação *online* executada no processador Nios II.

em média, 673 milissegundos. Nesse intervalo o processamento fica restrito à classificação, pois o sistema configurado executa apenas instruções sequenciais. O resultado é informado no microcomputador no qual o FPGA está conectado.

5 Conclusões

Com a implementação inicial do SIRAH, gerou-se um sistema capaz de classificar atividades corporais com até 95,7% de acerto. Porém, a execução no MATLAB não o torna totalmente aplicável em situações que se deseja obter classificações imediatas, pois o foco dessa versão é realizar o treinamento da RNA.

O desenvolvimento da versão embarcada, apresentada nesse trabalho, contribuiu para a geração de um classificador que opera no modo *online*, sendo essa uma característica essencial de sistemas aplicados em HAR, possibilitando o monitoramento remoto como, por exemplo, no acompanhamento de idosos ou pessoas com necessidades especiais.

Verificou-se que o treinamento da RNA no processador Nios II não é eficiente pelo alto consumo de tempo. A melhor alternativa é continuar o treinamento no modo *offline*, podendo ser efetuada no MATLAB ou com a codificação em C compilada no GCC, pois oferecem a mesma exatidão.

Na abordagem *online*, o algoritmo de classificação consome 673 milissegundos, que é um intervalo de tempo viável. Porém, o processamento sequencial não torna a amostragem das acelerações contínua, podendo afetar a exatidão das classificações. Para solucionar essa restrição, pretende-se implementar o algoritmo de classificação *online* descrito na linguagem de descrição de hardware VHDL, em FPGA, possibilitando o processamento paralelo e a obtenção de classificações em tempo real sem a perda de amostras.

Agradecimentos

Os autores agradecem ao Programa de Pós-Graduação em Engenharia Elétrica da Faculdade de Engenharia de Ilha Solteira por possibilitar o desenvolvimento da pesquisa, à CAPES e ao CNPq pelo processo n° 309193/2015-0.

Referências

- [1] L. Cheng, Y. Guan, K. Zhu and Y. Li, Recognition of Human Activities using Machine Learning Methods with Wearable Sensors, Annual Computing and Communication

- Workshop and Conference, vol. 7, 1–7, (2017), DOI: 10.1109/CCWC.2017.7868369.
- [2] M. J. B. Durango, A. C. R. da Silva e N. J. Londoño, SIRAH: Sistema de reconhecimento de actividades humanas, XXI Congresso Brasileiro de Automática - CBA2016, (2016), to appear.
 - [3] S. Ha and S. Choi, Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors, International Joint Conference on Neural Networks, vol. 1, 381–388, (2016), DOI: 10.1109/IJCNN.2016.7727224.
 - [4] R. Joaquineto and H. Sarmiento, A wireless biosignal measurement system using a SoC FPGA and Bluetooth Low Energy, International Conference on Consumer Electronics, vol. 6, (2016), 36–40, DOI: 10.1109/ICCE-Berlin.2016.7684711.
 - [5] R. Obukata, T. Oda and L. Barolli, Design of an Ambient Intelligence Testbed for Improving Quality of Life, International Conference on Advanced Information Networking and Applications Workshops, vol. 30, (2016), DOI: 10.1109/WAINA.2016.148.
 - [6] S. Resendes, P. Carreira e A. C. Santos, Conflict detection and resolution in home and building automation systems: a literature review, Journal of Ambient Intelligence and Humanized Computing, vol. 5, 699–715, (2014) DOI: 10.1007/s12652-013-0184-9.
 - [7] S. Samarah, M. GH. AL. Zamil, A. F. Aleroud, M. Rawashdeh, M. F. Alhamid and A. Alamri, An Efficient Activity Recognition Framework: Toward Privacy-Sensitive Health Data Sensing, IEEE Access, vol. 5, 3848–3859, (2017), DOI: 10.1109/ACCESS.2017.2685531.
 - [8] Z. Tang, J. Guo, S. Miao, S. Acharya and J. H. Feng, Ambient Intelligence Based Context-Aware Assistive System to Improve Independence for People with Autism Spectrum Disorder, Hawaii International Conference on System Sciences, vol. 49, 3339–3348, (2016), DOI: 10.1109/HICSS.2016.417.
 - [9] G. M. Tosch, L. B. Campos e C. E. Cugnasca, Home automation networks: a survey, Computer Standards & Interfaces, vol. 50, 42–54, (2017), DOI: 10.1016/j.csi.2016.08.008.
 - [10] D. N. Tran and D. D. Phan, Human Activities Recognition in Android Smartphone Using Support Vector Machine, International Conference on Intelligent Systems, Modelling and Simulation, vol. 7, 64–68, (2016), DOI: 10.1109/ISMS.2016.51.
 - [11] M. Weiser, The computer for the 21st century, Scientific american, vol. 265, 94–104, (1991), to appear.
 - [12] E. Zdravevski, P. Lameski, V. Trajkovik, A. Kulakov, I. Chorbev, R. Goleva, N. Pombo and N. Garcia, Improving Activity Recognition Accuracy in Ambient-Assisted Living, Systems by Automated Feature Engineering, IEEE Access, vol. 5, 5262–5280, (2017), DOI:10.1109/ACCESS.2017.2684913.