



UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA

Autorizada pelo Decreto Federal nº 77.496 de 27/04/76

Recredenciamento pelo Decreto nº 17.228 de 25/11/2016



PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO

COORDENAÇÃO DE INICIAÇÃO CIENTÍFICA

XXIII SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UEFS SEMANA NACIONAL DE CIENTÍFICA E TECNOLÓGICA - 2019

CONTROLE DE SERVOMOTOR USANDO SENSORES MIOELÉTRICOS

Gabriel Gomes¹; Thiago Alberto Barbosa²;

1. Bolsista PIBIC/CNPq, Graduando em Engenharia de Computação, Universidade Estadual de Feira de Santana, e-mail: ggomes1015@gmail.com
2. Orientador, Departamento de Tecnologia, Universidade Estadual de Feira de Santana, e-mail: thiago.barbosa@ecomp.uefs.br

PALAVRAS-CHAVE: Robótica; Mioelétrico; Servomotor.

INTRODUÇÃO

No decorrer da história, a automação industrial substituiu a mão de obra humana por máquinas com a capacidade de realizar os mesmos trabalhos só que com mais rapidez, aumentando assim a produção das fábricas. Com toda essa capacitação eles se tornaram aptos a realizar tarefas que seriam incapazes para seres humanos. Para evitar que trabalhadores sejam submetidos a tarefas perigosas ou até mesmo impossíveis de serem realizadas por eles, que exigem força e precisão superior, os robôs passaram a ser usados em seus lugares, Craig (2005).

A teleoperação consiste no controle de robôs à distância por um operador, que é responsável por fornecer julgamento na execução de cada tarefa feita pelo robô. Exemplos de robôs teleoperados são os robôs que realizam operações em áreas confinadas e altamente radioativas, robôs de exploração espacial e robôs que realizam procedimentos médicos, Nof (1985).

O sinal eletromiográfico (EMG) é um sinal biomédico proveniente da atividade elétrica produzida pelo músculo ao contrair. Esse sinal é usado como mecanismo para controlar próteses denominadas próteses mioelétricas, Díaz *et al.* (2013).

Para este trabalho, o objetivo foi desenvolver um sistema capaz de controlar os movimentos de um servomotor, que é amplamente utilizado em robótica, por meio de gestos que são feitos com a mão por um operador humano e que são identificados usando um bracelete com sensores mioelétricos. O sistema foi projetado usando o *Robot Operating System* (ROS), que é responsável por obter os gestos lidos por meio de uma braceleira de sensores eletromiográficos, Myo, e os mapear em comandos para controlar os movimentos de um servomotor.

MATERIAL E MÉTODOS OU METODOLOGIA (ou equivalente)

1. Robot Operating System (ROS)

O ROS é um *framework* bastante utilizado para escrever *software* em robótica. Ele possui uma coleção de ferramentas e bibliotecas que ajudam a simplificar a tarefa de criar comportamentos para robôs complexos e robustos em uma grande variedade de plataformas robóticas, Quigley *et al.* (2015).

Segundo Fernández *et al.* (2015), o ROS cria uma rede onde todos os processos estão conectados. Qualquer nó no sistema pode acessar essa rede, interagir com outros nós, ver as informações que eles estão enviando e transmitir dados para a rede. Uma visão dos conceitos usados no ROS:

- Nós: são onde a computação é feita. Para ter um processo que possa interagir com outros nós é preciso criar um nó com esse processo para depois conectá-lo na rede ROS.
- O Mestre: fornece o registro de nomes e o serviço de consulta para o restante dos nós. Ele também é responsável por configurar a conexão entre os nós, sem ele no sistema não é possível se comunicar com nós, serviços, mensagens e outros.
- Mensagens: os nós se comunicam com os outros nós através de mensagens. As mensagens contêm dados que enviam informações para outros nós.
- Tópicos: cada mensagem possui um nome para ser transmitida na rede ROS. Quando um nó está enviando dados na verdade ele está publicando um tópico. Para que um nó possa receber dados de um tópico basta ele assinar este tópico.

2. Myo Armband

A Myo, Figura 1, é uma braçadeira que usa sensores EMG proprietários para medir a atividade elétrica nos músculos do usuário e detectar cinco gestos feitos por sua mão. Ela possui uma Unidade de Medida Inercial (IMU) de nove eixos que também detecta movimento, orientação e rotação do antebraço.



Figura 1: Braçadeira Myo e adaptador *Bluetooth*.

O resultados obtidos com a leitura dos sensores da Myo são transferidos através de uma conexão *Bluetooth Smart* para dispositivos compatíveis. A Myo também transmite os

dados brutos de EMG e IMU para desenvolvedores possam usar em seus projetos e aplicativos.

3. Servomotor

O servomotor, é capaz de girar seu eixo para uma posição específica, esta posição está em algum lugar ao longo de 180 graus em qualquer direção a partir do ponto de referência. Portanto, em vez de girar 360 graus possíveis, o eixo do motor é limitado apenas a metade disso, 180 graus. A quantidade da volta dentro da faixa de zero a 180 graus é especificada por um sinal eletrônico. O sinal é produzido como uma série de pulsos, quando o pulso chega, o eixo do motor gira e, quando não chega pulso, ele para. A quantidade que o eixo do motor gira quando o pulso chega é determinada pela duração do pulso, quanto maior o pulso, maior o ângulo de giro, Mataric (2007).

4. Placa Lynxmotion SSC-32

É um servo-controlador que possui alta resolução para movimentos precisos e extremamente suaves. O controle de deslocamento pode ser feito de resposta imediata, velocidade controlada, movimento temporizado ou até mesmo uma combinação. A placa possui 32 canais que podem ser usados para controle dos servos.

RESULTADOS E/OU DISCUSSÃO (ou Análise e discussão dos resultados)

O projeto foi desenvolvido usando o ROS Melodic Morenia instalado no Linux Ubuntu 18.04 LTS Bionic Beaver.

Para desenvolvimento do projeto foi construída uma arquitetura que permite realizar a leitura dos gestos feitos com a mão por um operador humano, por meio de uma braçadeira Myo, e a partir disto fazer o controle de movimentação de um servomotor que está conectado em uma Placa Lynxmotion SSC-32, Figura 2. Essa arquitetura foi feita por meio do ROS, sendo que foram usados três nós, um para reconhecer os gestos da Myo e os transmitir, outro para obter os gestos que foram transmitidos e os mapear em movimentos que devem ser executados pelo terceiro nó, o nó da Placa SSC-32 que controla os servomotores.

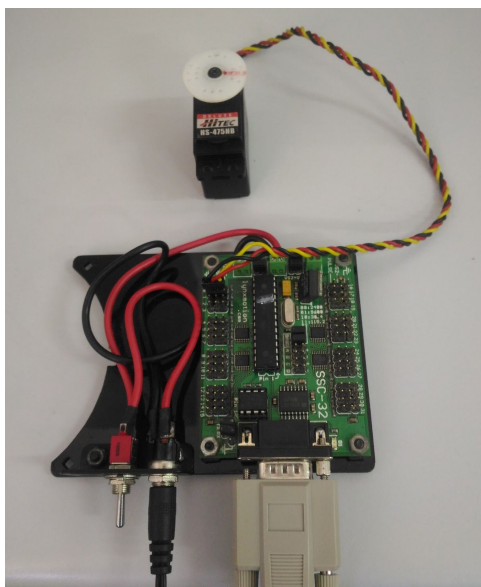


Figura 2: Placa SSC-32 com um servomotor conectado.

Para realizar a conexão com a Placa SSC-32 foi utilizado um nó ROS, feito em C++, encontrado na própria comunidade ROS, que por meio dele foi possível enviar comandos para a placa de uma forma mais abstrata, passando apenas o número do canal que o servo está conectado, a taxa de transmissão e o comando de controle de deslocamento, que foi citado anteriormente. Neste sistema foi usado o comando de controle de deslocamento de resposta imediata, que passa para a placa a posição que o servomotor deve se deslocar, foram mapeadas apenas duas posições para as quais o servomotor pode se mover, que são os dois extremos, zero e 180 graus.

Os desenvolvedores da braçadeira não fornecem suporte para distribuições Linux, mas na comunidade ROS foi encontrado um projeto, feito em Python chamado *myo-raw*, que consegue identificar os gestos realizados com a mão e publicá-los na rede ROS, contudo, é necessário fazer uma calibração toda vez que a braçadeira é colocada. A calibração consiste em armazenar valores dos sensores mioelétricos quando uma determinada posição de mão está sendo realizada, para que depois possa ser feita uma classificação com os valores salvos e se possa identificar que posição de mão está sendo feita no momento. Quanto maior o número de amostras armazenadas maior será a precisão para identificar o gesto.

CONSIDERAÇÕES FINAIS (ou Conclusão)

No projeto, foi possível testar o controle de um servomotor usando a Myo, visto que a Placa SSC-32 pode realizar o controle de 32 servomotores o projeto pode ser expandido para que se possa controlar mais de um servomotor. O projeto foi feito mapeando apenas dois gestos para valores de deslocamento definidos, uma possível melhoria para ser feita é aumentar a quantidade de gestos reconhecidos para uma maior quantidade de valores de deslocamento.

REFERÊNCIAS

- QUIGLEY, M., GERKEY, B., SMART, W. D. 2015. *Programming Robots with ROS: A Practical Introduction to the Robot Operating System*. O'Reilly Media. p.3
- FERNÁNDEZ, E. et al. 2015. *Learning ROS for Robotics Programming*. Packt Publishing, 2nd edition. p.33-34.
- MATARIC, M.J. 2007. *The robotics primer*. Intelligent Robotics and Autonomous Agents series. MIT. p.37
- CRAIG, J. J. 2005. *Introduction to Robotics Mechanics and Control*. Pearson Education. 3 ed.
- NOF, S. Y. 1985. *Handbook of Industrial Robotics*. John Wiley Sons Inc.
- DÍAZ, F. J. D. O., DUQUE, L. H., GODOI, T. S. M. 2013. Prótese Mioelétrica Controlada por Redes Neurais. 3(2):2-3
- How does the Myo armband work. Disponível em:
<<https://support.getmyo.com/hc/en-us/articles/202532376-How-does-the-Myo-armband-work->>
. Acesso em: 06 ago. 2019.