

## PROJETO E IMPLEMENTAÇÃO DE UM BRAÇO ROBÓTICO COM TECNOLOGIA HAPTIC

**JÔNATAS LEMUEL BISPO ZOTTIS<sup>1</sup>; LUCIANO ANACKER LESTON<sup>2</sup>**

<sup>1</sup>*Universidade Federal de Pelotas – lemueltra@gmail.com*

<sup>2</sup>*Universidade Federal de Pelotas – luleston@gmail.com*

### 1. INTRODUÇÃO

À medida que os computadores se tornam cada vez mais predominantes, a capacidade de comunicação entre humanos e computadores se torna mais importante. Embora muito possa ser transmitido visualmente, os humanos têm acesso a outros sentidos, os quais podem ser utilizados para comunicar informações e fornecer ‘feedback’ a um usuário humano (SKORINA et al., 2008).

Cada vez mais se tem procurado encontrar formas “diferentes” de interagir com robôs, com o objetivo de tornar a interface homem – máquina mais intuitiva. Este é um dos grandes motivos do uso de tecnologia HAPTIC nestes sistemas. Como descreve HUSSEIN et al. (2014), o uso de sistemas hapticos possibilita ao operador interagir com um robô sem a necessidade de ser um especialista em robótica, ao contrário, o operador pode focar apenas na definição correta da tarefa robótica.

Haptics é um termo derivado do verbo grego “haptethai”, significa tocar, referindo-se ao ato de detecção e manipulação através do toque. A palavra haptic também é apontada como a ciência do toque, a qual se dedica a estudar e simular a pressão, textura, vibração e outras sensações biológicas relacionadas com o toque (MOURATO, N. R. S., 2010).

As aplicações da tecnologia haptic vêm se espalhando rapidamente em diversas áreas, como a indústria de videogames, artes e criação, manufatura, robótica, educação e treinamento, bem como no campo da medicina (TALASAZ et al., 2010).

Em algumas dessas novas aplicações, sistemas baseados em tecnologia haptic podem ter um papel fundamental e, em muitos casos, eles são o ponto de partida para a evolução da robótica, principalmente nos campos cujo feedback de força do robô é uma questão importante para o sucesso da execução das tarefas. No entanto, a maioria dos dispositivos hapticos “sofisticados” é relativamente cara (HUSSEIN et al., 2014).

Desta forma, este trabalho tem por objetivo o desenvolvimento de um sistema mestre-escravo com tecnologia haptic, baseando-se especialmente no controle de realimentação de força de um braço robótico de 4 graus de liberdade, utilizando somente ferramentas ‘open source’. Capacitando a qualquer operador controlar o robô e, ao mesmo tempo, receber o feedback das forças exercidas no robô.

### 2. METODOLOGIA

Para desenvolver este trabalho foram utilizadas uma ampla variedade de ferramentas para gerenciar e facilitar a implementação do projeto, dentre essas, o ROS (*robot operating system*), o qual é um sistema operacional de código aberto para robôs. Cabe salientar que o ROS não é apenas um sistema operacional no sentido tradicional de gerenciamento e agendamento de processos, ele possui também uma camada de comunicação estruturada acima dos sistemas operacionais (QUIGLEY et al., 2009). Esta ferramenta possui vários outros

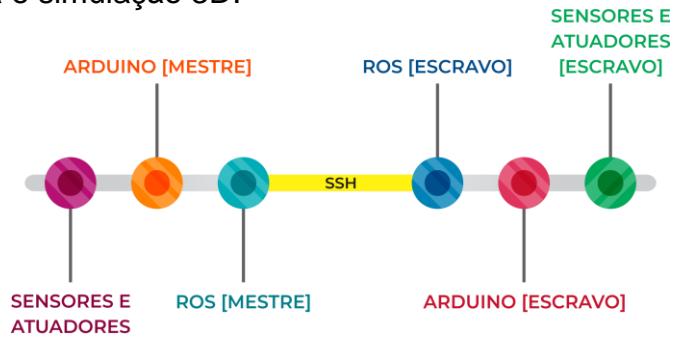
sistemas capazes de facilitar o desenvolvimento de sistemas robóticos, também se relacionando com vários outros programas e 'frameworks' de robótica existentes. Além disso, o ROS possui também ferramentas para modelagem cinemática de robôs, suporta diversas linguagens de programação e se comunica com seus diferentes processos e 'hosts' através de redes de topologia P2P ('peer to peer').

A plataforma utilizada para o desenvolvimento do presente trabalho foi o Arduino Mega 2560. O Arduino Mega é uma placa com microcontrolador ATMEGA2560. Ele possui 54 pinos de entradas/saídas digitais, 16 entradas analógicas, conexão USB, um oscilador de cristal de 16 MHz, 256 KB de memória flash, botão de reset e uma entrada de alimentação. Ele também possui várias facilidades para a comunicação com um computador, com outro Arduino ou com outro microcontrolador de uma arquitetura diferente.

Neste projeto também foram utilizados 12 servo motores CF Sunbird SG90 (6 motores para o sistema mestre e 6 motores para o sistema escravo), os quais possuem tensão de alimentação de 5V e torque de 2,5 Kg/cm. Além disso, também foram utilizadas 6 placas ponte h l298n para o acionamento destes motores e 10 unidades do circuito integrado de sensor de corrente ACS712 (5 para o sistema mestre e 5 para o sistema escravo) ), estas disponibilizam 100 mV/A na sua saída, podendo medir a corrente máxima de 20A.

Para realizar a comunicação entre os sistemas mestre-escravo se utilizou em conjunto com o ROS o protocolo de comunicação SSH. O protocolo SSH (também conhecido como ‘Secure Shell’) é um protocolo para login remoto seguro de um computador para outro. Ele fornece várias opções de autenticação e protege a segurança e a integridade das comunicações com criptografia. É uma alternativa segura aos protocolos de login não protegidos, como telnet e rlogin, e métodos de transferência de arquivos inseguros como FTP (SSH, 2019).

A Figura 1 abaixo apresenta o diagrama do sistema de comunicação do trabalho, com os Arduinos Mega atuando nos motores e se comunicando com os sensores e o ROS. A comunicação entre os hosts ROS é realizada através de uma rede com o protocolo SSH. No ROS são feitas todas as operações de controle, cinemática e simulação 3D.



**Figura 1 - Diagrama do sistema de comunicação**

As estruturas mecânicas do projeto foram projetadas nos softwares de modelagem 3D Blender e de desenho vetorial Inkscape e cortadas à laser em pequenas peças de MDF de 3mm e 6mm. Como as estruturas físicas dos sistemas mestre e escravo são iguais, desenvolveu-se apenas um modelo de estrutura mecânica para este trabalho o qual foi reproduzido duas vezes. Na Figura 2a, apresenta-se a estrutura mecânica do robô já montada e com todos os motores devidamente posicionados e fixados. Na Figura 2b, pode-se observar um

pequeno diagrama do robô, na qual são identificados os 4 graus de liberdade e as dimensões físicas do robô em milímetros.

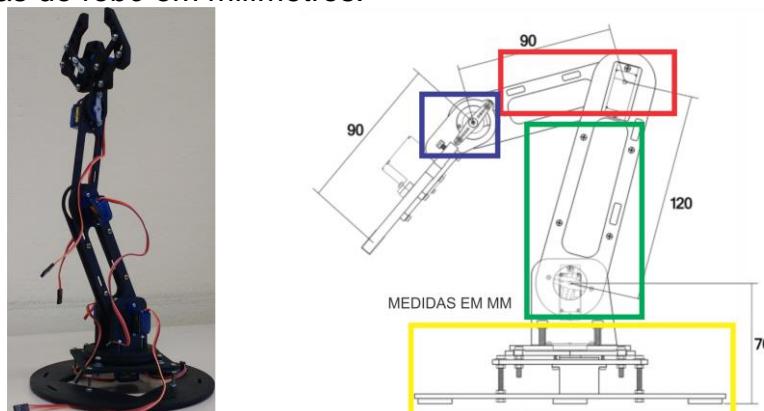


Figura 2 - a) Estrutura física do robô; b) Diagrama da estrutura física robô

Neste trabalho foram utilizadas duas versões diferentes de malha de controle para os motores. Na Figura 3a, pode-se observar a malha de controle quando o fluxo de informação sai do sistema mestre em direção ao sistema escravo, controlando a posição dos motores do sistema escravo. Para isso, utilizou-se o controle PD, a qual é uma das alternativas dentre os controladores mais utilizados em sistemas de controle industriais. Isto se deve ao fato de possuir um número reduzido de parâmetros a serem ajustados (BASILIO; MATOS, 2002).

Na Figura 3b, observa-se a malha de controle PD quando o fluxo de informação sai do sistema escravo em direção ao sistema mestre, controlando a corrente dos motores do sistema mestre (etapa de controle da tecnologia haptic).

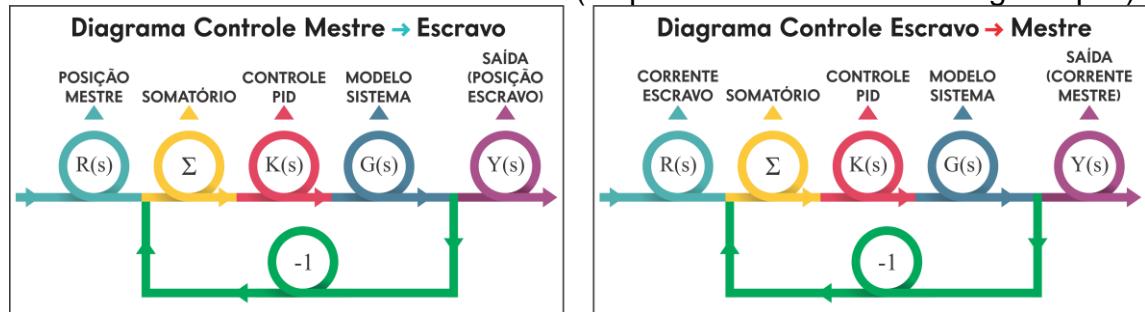


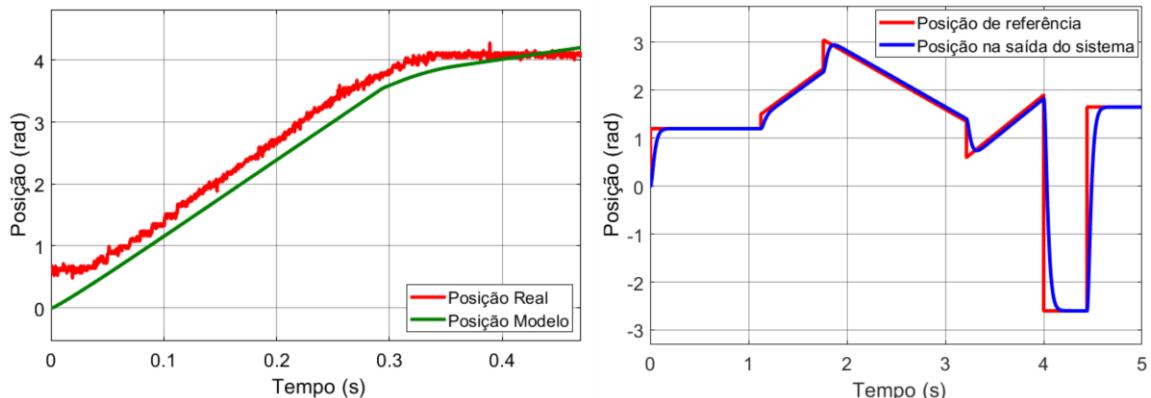
Figura 3 - a) Diagrama de controle de posição (escravo); b) Diagrama de controle de corrente (mestre)

### 3. RESULTADOS E DISCUSSÃO

Utilizando algumas medições de tensões, correntes e posição do eixo do motor realizadas com o auxílio de um osciloscópio estimou-se o modelo matemático do motor. As curvas comparadas do modelo matemático obtido e do motor real, validando esta estimativa, são apresentadas na Figura 4a.

Com os resultados conquistados na etapa de modelagem matemática do motor, realizou-se o controle PD da posição desse motor. Os primeiros resultados foram obtidos em simulações com um sistema de controle contínuo. Logo após, discretizou-se este sistema, sob uma taxa de amostragem de 20 ms, assim como, o reajuste dos ganhos do controlador PD. As curvas do sistema de controle discretizado da posição do motor podem ser observadas na Figura 4b.

Comclui-se também as etapas de testes de comunicação SSH entre hosts ROS, a etapa de modelagem 3D do robô, a etapa de modelagem cinemática e a etapa de controle do modelo 3D do robô utilizando ROS juntamente com o software de simulação Gazebo.



**Figura 4 - a) Curvas da posição real do motor e das obtidas no modelo matemático; b) Curvas da posição do motor com controle de posição discreto**

#### 4. CONCLUSÕES

Grande parte dos objetivos propostos neste trabalho foram concluídos com êxito. Utilizando ROS, microcontrolador Arduino, sensores e motores foi possível realizar várias etapas do projeto. Como trabalho futuro, dever-se-á realizar a união de todos os sistemas resultantes das etapas já finalizadas em um único sistema robótico (mestre-escravo). Concluindo-se assim, todos os objetivos propostos inicialmente neste trabalho.

#### 5. REFERÊNCIAS BIBLIOGRÁFICAS

Erik SKORINA, E. H. et al. A Soft Robotic Wearable Wrist Device for Kinesthetic Haptic Feedback. **Frontiers in Robotics and AI**, Trieste, volume 5, artigo 83, 2018.

HUSSEIN, M. A. et al. Haptic Control Development of Robotic Arm. **International Journal of Control, Automation and Systems**, San Jose, volume 3, número 3, 2014.

MOURATO, N. R. S. **Interaction with Real Environments: an Approach Based on Haptic Systems**. 2010. Dissertação (Mestrado em Engenharia Mecânica) - Faculdade de Ciências e Tecnologia, Universidade de Coimbra.

TALASAZ, A. et al. Haptics-enabled Teleoperation for Robot-assisted Tumor Localization. **IEEE International Conference on Robotics and Automation**, Anchorage, pp.534-5345, 2010.

QUIGLEY, M. et al. ROS: an open-source Robot Operating System. In: **ICRA WORKSHOP ON OPEN SOURCE SOFTWARE**, Kobe, 2009.

SSH. **SSH Protocol**. Acessado em 14 set. 2019. Online. Disponível em: <https://www.ssh.com/ssh/protocol/>.

BASILIO, J. C.; MATOS, S. R. Design of PI and PID Controllers With Transient Performance Specification. **IEEE Transactions on Education**, volume 45, número 4, 2002.