

## **Sistema de comunicação entre neuronavegador e robô neurocirúrgico baseado em ROS**

**José Carlos Andrade do Nascimento**

**Prof. Dr. Glauco A. P. Caurin, Paulo H. Polegato**

Escola de Engenharia de São Carlos / Universidade de São Paulo

[josecarlos.101@usp.br](mailto:josecarlos.101@usp.br), [gcaurin@sc.usp.br](mailto:gcaurin@sc.usp.br), [paulopolegato@usp.br](mailto:paulopolegato@usp.br)

### **Objetivos**

Dado o procedimento médico SEEG (estéreo-eletroencefalografia) [1] no contexto da robótica aplicada a ambientes cirúrgicos, o projeto objetivou o desenvolvimento da comunicação do braço robótico *KUKA iiwa 14* com o *software open source 3D Slicer*, usado para pesquisas médicas auxiliadas pela tecnologia [1], visando transformá-lo em um neuronavegador com uma arquitetura em ROS (Robot Operating System) e comunicação TCP. Além disso, o projeto foi responsável por aplicar métodos de correção fiducial com o intuito de relacionar as bases das coordenadas do espaço virtual do *3D Slicer* com o espaço físico do robô.

### **Métodos e Procedimentos**

O projeto se passou no laboratório AeroTECH, localizado no campus 2 da USP de São Carlos, e faz parte do Projeto Cirurgia, que conta com diversos projetos relacionados que buscam, em conjunto, realizar uma simulação do procedimento com o braço robótico do laboratório.

Quanto à comunicação, a linguagem escolhida foi Python 3, devido às suas aplicações do paradigma orientado a objetos. Foram desenvolvidos módulos individuais que são importados por outras partes do projeto e, principalmente, pela interface desenvolvida

pelo laboratório como um *add-on* para o *3D Slicer*, sendo facilmente integrável com a arquitetura do laboratório. Quanto às tecnologias utilizadas, foi usado a biblioteca *pyigtl*, desenvolvida por Andras Lasso [3], um dos principais desenvolvedores do *software* em questão. Esta biblioteca é responsável por abstrair a transmissão de dados médicos por dispositivos na rede local, como matrizes, pontos, imagens e afins, usando o protocolo *OpenIGT* [4], padronizado para a troca de dados entre dispositivos em ambientes médicos. Sua implementação foi simples e permitiu que partes separadas da interface se comuniquem sem perda de informação.

Na etapa do correção, é necessário a coleta de 3 pontos: duas tâmporas e um náculo, responsáveis por fixar uma base no sistema de coordenadas correspondentes. Isso deve ser feito, como é instruído na interface desenvolvida pelo laboratório, tanto no programa quanto no espaço físico. Desse modo, a aplicação desenvolvida faz a coleta desses pontos e realiza o processo de *registration*, que, através do método ICP (*Iterative Closest Point*) [5] gera uma matriz de transformação homogênea que, ao ser usada para transformar um ponto qualquer na base do *software*, gera seu correspondente no espaço real.

Já na etapa de operação, o usuário seleciona dois pontos da reconstrução no *software*, sendo estes o ponto de entrada e o ponto alvo,

que representam a posição e orientação do eletrodo. Esses dados são então transmitidos para o nó *commander* em ROS que é responsável por intermediar a comunicação com o robô.

A arquitetura geral deste projeto consiste no diagrama apresentado na figura [1].

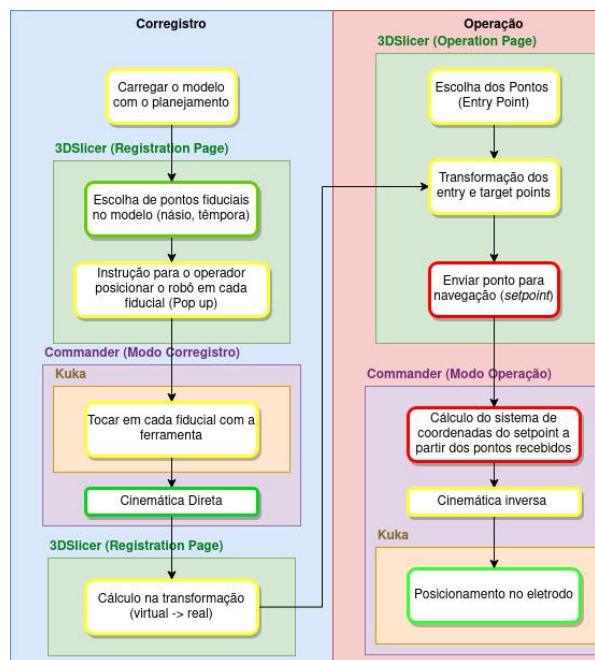


Figura 1 - Diagrama de funcionamento da etapa de corregistro e operação.

## Resultados

A implementação da comunicação na arquitetura do laboratório gerou resultados positivos e nos permitiu realizar testes para validar o projeto como um todo, concretizando etapas individualmente. Entretanto, por ser uma biblioteca recente e com um único desenvolvedor, o *pyigtl* ainda conta com alguns problemas de estabilidade de conexão, o que não pode acontecer em um ambiente cirúrgico. Logo, um dos próximos passos do laboratório é o *neurolink*, uma frente voltada para o desenvolvimento de uma rede *open source*

customizada e própria do laboratório, que atenda às suas necessidades específicas.

Quanto ao processo de corregistro, foi possível validar seu funcionamento, fisicamente e com o cálculo de seu erro através do TRE e FRE [6], contando com uma precisão positiva. Entretanto, devido à falta de ferramentas necessárias para seu teste, não foi possível realizar nenhum teste de precisão real com o braço robótico. Logo, uma das próximas etapas quanto ao corregistro seria realizar testes de precisão com a produção de novas ferramentas para o laboratório.

## Conclusões

Dado os testes realizados com a participação de integrantes do HCFM (Hospital das Clínicas da Faculdade de Medicina), pode-se concluir que o Projeto Cirurgia tem um potencial promissor, uma vez que já ocorreu uma simulação das três etapas do processo: planejamento, corregistro e operação. Embora haja partes a serem repensadas e novas frentes a serem criadas, o projeto apresentou resultados com projeções positivas.

## Referências

- [1] CENTENO, R. S. et al. Estereoeletroencefalografia na era da cirurgia guiada por imagem. pt. *Journal of Epilepsy and Clinical Neurophysiology*, scielo, v. 15, p. 178–183, dez. 2009. ISSN 1676- 2649. DOI: <https://doi.org/10.1590/S1676-2649200900040008>
- [2] 3D Slicer image computing platform. 2023. Disponível em: [www.slicer.org](http://www.slicer.org)
- [3] ANDRAS Lasso github page, one of the 3D Slicers core developer. 2023. <https://github.com/lassoan>
- [4] OPENIGTLINK protocol. 2023. Disponível em: <http://openigtlink.org/>
- [5] XIANG GAO, T. Z. - Introduction to Visual SLAM From Theory to Practice. [S.I.: s.n.], 2021.

[6] REGISTRATION Errors - TRE and FRE with python examples. 2023. Disponível em: <[http://insightsoftwareconsortium.github.io/SimplerTK-Notebooks/Python\\_html/68\\_Registration\\_Errors.html](http://insightsoftwareconsortium.github.io/SimplerTK-Notebooks/Python_html/68_Registration_Errors.html)>. (Acessado em: 9 de set de 2023).