

## SOFTWARE DE CONTROLE PARA ROBÔ DE TELEPRESENÇA

**Kevim Iochims**

Acadêmico do curso de Engenharia de Controle e automação - Universidade Federal de Santa Maria

k.iochims@gmail.com

**Ana E. M. Ramos**

Acadêmico do curso de Engenharia de Controle e automação - Universidade Federal de Santa Maria

anaemr@gmail.com

**Rodrigo da Silva Guerra**

Professor/Pesquisador do curso de Engenharia de Controle e automação - Universidade Federal de Santa Maria

rodrigo.guerra@ufsm.br

***Resumo.** Este documento relata o desenvolvimento de um software de controle para um robô de telepresença. O software proposto tem por finalidade permitir o controle dos movimentos de um robô de telepresença através de uma rede sem fio entre computadores. O robô de telepresença é dotado de um computador, display, câmera, microfone, caixas de som, motores e rodas. Este software permitirá o controle do deslocamento do robô remotamente, assim como movimentar a posição vertical da câmera e outras funções acessórias.*

***Palavras-chave:** Software. Telepresença. Robótica.*

### 1. INTRODUÇÃO

Um robô de telepresença é um equipamento que simula a presença de uma pessoa em um determinado ambiente. Basicamente trata-se de um sistema de videoconferência (áudio e vídeo) móvel, ou seja, o robô possui a capacidade de se movimentar livremente através de comandos enviados remotamente [1]. A tecnologia de videoconferência hoje é acessível à grande parte da população, sendo as ligações cada vez mais simples e corriqueiras, mesmo a longas distâncias. Existem vários softwares

que permitem estas ligações como Skype, Google Hangouts, Liphone e Jitsi [2].

O desenvolvimento de uma aplicação que permita a mobilidade do robô a partir de comandos remotos é menos comum. Embora já sejam comercializadas algumas versões de robôs de telepresença, como os Anybots [3], e a UFSM possua uma versão em atividades experimentais, existem poucos softwares gratuitos e de código aberto que permitam a utilização no controle de outros robôs a partir da mudança de alguns parâmetros. Uma notável exceção é o software do robô Oculus [4].

Este trabalho aborda o desenvolvimento de um software projetado com o intuito de permitir que o usuário controle os movimentos do robô de telepresença de qualquer lugar geográfico, desde que ambos estejam conectados à internet.

A necessidade de um aplicativo de controle aliado ao sistema de videoconferência é vital para a existência da telepresença. É muito importante que o robô possa se movimentar e que faça sua presença ser sentida, pois este é o aspecto que o diferencia de uma simples ligação de videoconferência habitual.

Dentre as dificuldades de se fazer uma conexão remotamente através da internet destacam-se os problemas de largura de

banda (velocidade da internet) e cobertura Wi-Fi. Softwares de videoconferência geralmente necessitam de uma grande largura de banda para enviar e receber áudio/vídeo, o que dificulta o funcionamento onde não há uma conexão rápida com a internet.

A cobertura da rede sem fio também restringe a aplicação do robô. Para a transmissão contínua de áudio/vídeo em tempo real é estritamente necessária uma intensidade de sinal razoável de forma a evitar possíveis travamentos. Além disso, a área de cobertura da rede sem fio equivale à área onde é possível controlar o robô remotamente, uma vez que sem conexão com a rede o robô não pode ser movimentado pelo usuário, e o mesmo se torna um computador imóvel que não cumpre seu propósito.

## **2. DESENVOLVIMENTO**

Esta seção descreve os objetivos, a metodologia utilizada e apresenta o software proposto.

### **2.1 Objetivos**

A partir da participação do autor em projetos ligados ao robô de telepresença da UFSM, e da necessidade de se obter uma forma de controlar o dispositivo remotamente para a utilização do mesmo em testes, foi proposto o desenvolvimento de um software que suprisse estas necessidades.

Esta aplicação aliada com algum software de videoconferência qualquer, permite que o robô de telepresença seja controlado remotamente por um usuário e que sua presença seja sentida não só pela videoconferência, mas também pela sua liberdade de movimentação no ambiente em que estiver.

Os objetivos, portanto, são: criar um software que estabeleça uma conexão entre o usuário e o robô, de forma que este proporcione o controle das rodas para movimentar o robô e da câmera para que

possa visualizar melhor o ambiente em que se encontra; analisar os resultados obtidos com sua utilização.

### **2.2 Referencial Teórico e Metodologia**

O desenvolvimento do software de controle para o robô de telepresença para suprir a necessidade de estabelecer um elo entre o usuário e o robô se deu conforme as necessidades teóricas e, posteriormente, conforme as demandas obtidas na prática.

Teoricamente o software deveria ser simples e funcional, capaz de estabelecer uma conexão segura via internet entre os dois computadores (usuário e robô) e proporcionar a troca de informações (comandos) que resultassem na movimentação e obtenção de informações gerais, como a carga das baterias do robô.

Porém, com um estudo prévio sobre os locais onde o robô seria utilizado com mais frequência em atividades de projetos, foi possível notar que a largura de banda e a velocidade de conexão com a internet não proporcionavam a transmissão em tempo real de áudio/vídeo.

### **2.3 Software Proposto**

Para a execução do software em questão foi utilizada a linguagem de programação Python por ser uma linguagem de alto nível, orientada a objetos e que permite a portabilidade para diferentes sistemas operacionais.

Esta aplicação se divide em dois programas. Um deles estará sendo executado no computador do usuário e o outro computador embarcado no robô. Estes programas se conectam entre si através de Sockets, utilizando o protocolo TCP/IP, que estabelece uma conexão segura entre as duas partes, e permite a troca de dados entre as duas aplicações (tráfego de dados bidirecional).

A linguagem Python oferece um módulo que implementa a conexão via Sockets, o que facilita a programação. Além disso, a

utilização deste módulo propicia a conexão tanto via internet quanto via rede local, o que atende as especificações propostas.

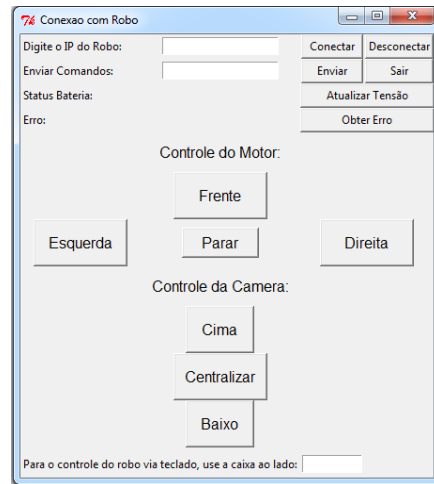
Além do requisito de se proporcionar uma conexão entre ambas as aplicações via rede wireless, o programa executado no computador embarcado no robô deve receber comandos do computador do usuário, interpretá-lo e agir de forma correta.

Os comandos de movimentação (frente, direita e esquerda) são traduzidos em velocidade nos motores que giram as rodas do robô. Para executar esta interface entre o programa e o robô é utilizada uma conexão USB/Serial, utilizando-se outro módulo que o Python oferece chamado PySerial. Este módulo torna possível o envio de comandos para uma placa controladora, que por sua vez faz os motores girarem. A configuração dos parâmetros de comunicação é feita de forma simples e objetiva. Em poucas linhas de código é possível passar os parâmetros e “abrir” uma conexão com um dispositivo.

O ajuste do ângulo de visão da câmera é feito de maneira semelhante ao da movimentação do robô. O usuário remoto envia um comando, que é processado pelo software do robô. Este, por sua vez, através de conexão USB/Serial, envia o comando para uma placa microcontroladora que ajusta a posição do eixo de um servomotor, onde está a câmera.

Outras funções acessórias foram criadas, como a de obter o nível de tensão da bateria conectada ao motor que controla o movimento do robô e uma função para obtenção de possíveis erros na parte de movimentação (placa controladora e motores).

Para o desenvolvimento de uma Interface Gráfica do Usuário (GUI) foi utilizado o módulo Tkinter por sua simplicidade e eficiência. Este módulo permite a criação de uma janela gráfica acessível aos mais diversos usuários, que geralmente não são habituados a utilizar uma janela de comandos. A figura 1 mostra a interface gráfica do usuário do software controlador desenvolvido.



(Figura 1 – Interface Gráfica)

É possível notar que para estabelecer a conexão basta conhecer o endereço IP do computador embarcado no robô. Após a conexão, diversas opções podem ser selecionadas com um simples clique do mouse. Os botões Frente, Parar, Esquerda e Direita enviam comando de movimento para o robô. Os comandos Cima, Centralizar e Baixo controlam a câmera situada no topo do robô para vencer a limitação do ângulo de visão.

Para evitar rápidas arrancadas foi criado um sistema de aceleração para o movimento. Este controle consiste em um incremento na velocidade do robô cada vez que um botão é pressionado repetidamente em um curto intervalo de tempo. Assim é possível obter arrancadas mais suaves e se movimentar curtas distâncias em baixas velocidades.

Assim, basta clicar em um dos botões que o robô se movimenta na direção desejada, regula o ângulo da câmera ou para. Outras opções já abordadas anteriormente e implementadas na interface gráfica são a obtenção da tensão das baterias e possíveis erros. Há também um campo para enviar comandos através de texto, o que não é muito comum.

Outro aspecto importante é a caixa de texto no final, utilizada para o controle via teclado, onde teclas são configuradas para fazer o papel dos botões descritos anteriormente. Logo, é possível controlar o robô tanto por teclado quanto por mouse.

Pensando em possíveis falhas na conexão, devido a possíveis quedas da rede Wi-Fi utilizada, por exemplo, o robô foi dotado de um sistema que cessa os seus movimentos após um Timeout (perda de conexão ou não recebimento de dados) de 2 segundos. Assim, evitam-se movimentos indesejados e possíveis acidentes devido a problemas técnicos.

### 3. CONSIDERAÇÕES FINAIS

No projeto mais recente, foi possível notar a facilidade que as crianças internadas no Hospital Universitário de Santa Maria tiveram em se adaptar ao software. Com apenas algumas instruções e um pequeno tempo de reconhecimento, as crianças foram capazes de movimentar o robô pelo ambiente com nenhuma dificuldade.

A aceleração do robô foi um aspecto importante quando era necessário fazer manobras em pequenos espaços, uma vez que se torna mais fácil em baixas velocidades. Nestes casos de manobra, o movimento vertical da câmera também foi muito conveniente, prevenindo possíveis colisões com objetos de pequena altura que não são vistos quando a câmera está em sua posição original.

A comunicação entre os computadores e entre os dispositivos do robô (motor, servomotor e placas controladoras) se deu de forma robusta com a utilização do software. Além disso, em eventuais perdas de conexão, o robô possui um sistema que cessa seus movimentos, o que foi importante em casos de perda de sinal wireless para evitar movimentos indevidos.

### 4. REFERÊNCIAS

[1] TecMundo, “Robô de tele presença faz muito mais do que projetar sua imagem”, disponível em: <http://www.tecmundo.com.br/robotica/1703-2-robo-de-tele-presenca-faz-muito-mais-do-queprojetar-sua-imagem.htm/>, acesso em 11 de junho de 2014.

[2] TANG, J. C.; WEI, C.; KAWAI, R. Social telepresence bakeoff: Skype group video calling, google+ hangouts, and Microsoft avatar Kinect. In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion, p. 37–40, 2012.

[3] BLACKWELL, T. Anybots. Disponível em: <https://www.anybots.com/>. Acesso em junho de 2014.

[4] XAXXON, Oculus, Surveillance and Telepresence Robot Laptop Conversion Kit. Disponível em: <http://www.xaxxon.com/>. Acesso em junho de 2014.