UTBot@Home 2017 Team Description Paper

Uso do robô Pioneer LX para tarefas domésticas

Danilo Henrique Santos[1,2], **Piatan Palar**[1,2]

João Alberto Fabro[1,2], **André Schneider de Oliveira**[1,2]

[1] UTFPR – Universidade Tecnológica Federal do Paraná - Curitiba (www.utfpr.edu.br)
LASER – Laboratório Avançado de Sistemas Embarcados e Robótica
(laser.dainf.ct.utfpr.edu.br)

[2]PPGCA - Programa de Pós-Graduação em Computação Aplicada (ppgca.dainf.ct.utfpr.edu.br)

fabro@utfpr.edu.br, andreoliveira@utfpr.edu.br
danilo.santos@ifsp.edu.br, kpiatan@gmail.com

Abstract. *This TDP describes the hardware and software implemented to allow a Pioneeer LX robot to execute domestic tasks related to the Robocup@Home initiative. The robot uses the information provided by its sensors to execute navigation tasks such as environment mapping and auto-localization, and the "follow-me" task. The sensors are the Microsoft Kinect and a high resolution Laser scan. It also executes some other Robocup@Home tasks, such as voice recognition, and object manipulation (using a robotic arm), and has an innovative human-robot interaction interface through facial expressions based on simulated emotions. The robot is programmed using ROS (Robot Operating System), and the decision making is programmed using the LIDA (Learning Intelligent Distribution Agent) Framework.*

Resumo. *Este artigo descreve o hardware e software que possibilitam a um robô industrial model Pionner LX executar tarefas domésticas relacionadas com a iniciativa Robocup@Home. O robô usa as informações fornecidas pelos seus sensores para executar tarefas de navegação, tais como mapeamento do ambiente e auto-localização, e a tarefa "siga-me". Os sensores utilizados são o Microsoft Kinect e um scanner a Laser de alta resolução. Ele também executa algumas outras tarefas da Robocup@Home, como o reconhecimento de voz, manipulação de objetos por meio de um braço robótico, e possui uma inovadora interface humano-robô através da geração de expressões faciais com base na simulação de emoções. O robô é programado com o sistema operacional de robôs ROS (Robot Operating System) e a tomada de decisão é feita através do framework LIDA (Learning Intelligent Distribution Agent).*

1. Introdução

A Robocup@Home é uma competição que visa fomentar o desenvolvimento de robôs que venham a ajudar no ambiente doméstico, sendo a maior competição para robôs de serviço do planeta [1] [2]. Afim de executar as tarefas domésticas, são desenvolvidos, dentre

muitos outros campos da robótica, a interação homem-robô, a navegação, o mapeamento de ambientes dinâmicos e a visão computacional [1].

Para a execução destas tarefas, está sendo programado um robô industrial modelo Pioneer LX, com o ROS (*Robot Operating System*), e um conjunto de bibliotecas de sofware e ferramentas que ajudam a desenvolver aplicações para robótica [3] e o framework LIDA (*Learning Intelligent Distribution Agent*), um framework baseado no modelo conceitual de cognição humana, que por sua vez implementa a Teoria do Workspace Global (GWT - *Global Workspace Theory).* O sistema de controle do robô é responsável por adquirir e interpretar as leituras dos diversos sensores utilizados no robô e, a partir desses dados e do cenário que se está inserido, tomar decisões para cumprir suas tarefas.

Nesse documento, são apresentados os resultados parciais e expectativas de projetos em desenvolvimento para o robô. A seção 2 apresenta uma descrição do hardware do robô, incluindo suas especificações e características. O software já desenvolvido, e o que ainda está em desenvolvimento, são descritos na seção 3. Conclusões e perspectivas futuras para o desenvolvimento do robô são apresentadas na seção 4.

2. Descrição do Hardware

2.1. Especificações do robô

O robô utilizado de base para o projeto é o Pioneer LX, que é uma plataforma de pesquisas baseado no robô industrial Adept Lynx. Esse robô de 60 kg pode carregar até 60 kg de carga e se deslocar à velocidade máxima de 1.8 m/s, além de possuir uma autonomia de 13 horas. As dimensões do robô Pioneer LX são: 50 cm de largura, 70 cm de profundidade e 45 cm de altura. Um computador com processador Dual Core 1.8 GHz, memória RAM de 2GB DDR3 e Wireless Ethernet estão inclusos neste robô. A montagem final inclui os sensores apresentador no próximo tópico, assim como um tablet Genesis TAD modelo GT-7305, com resolução de 1280x720, que é utilizado para exibir faces simulando as emoções atuais do robô. A figura 1 representa o robô montado.

2.2. Suporte

Em sua superfície se encontra um suporte construído em alumínio (como observado na Figura 1), com dois níveis e placas de alumínio de $90cm^2$. As placas estão a 16cm de distância uma da outra, e o primeiro nível está a 14,5cm do robô, deixando o robô com um total de 59cm de altura.

2.3. Sensores

Alguns sensores também estão embutidos no robô: um sensor laser rangefinder de 270 graus, sensores ultrassônicos frontal e traseiro e um sensor de choque frontal [1]. Além destes sensores, foi incorporada uma base para um sensor de profundidade Kinect. Com este sensor é possível criar uma imagem de profundidade, posicionar suas câmeras verticalmente, rastrear a silhueta de uma pessoa, entre outras funções [5]. Para estabelecer a comunição entre o sensor Kinect e o ROS foi usado o projeto open source Openni Kinect[6] que já acompanha os drivers do Kinect e torna a comunicação entre o sensor e o ROS bastante simples. Os dados gerados pelo Kinect e pelo sensor laser que são publicados em tópicos no ROS são peça chave para o funcionamento da maioria dos algoritmos

usados nesse projeto, visto que é a partir da interpretação dessas imagens que o robô toma grande parte de suas decisões.

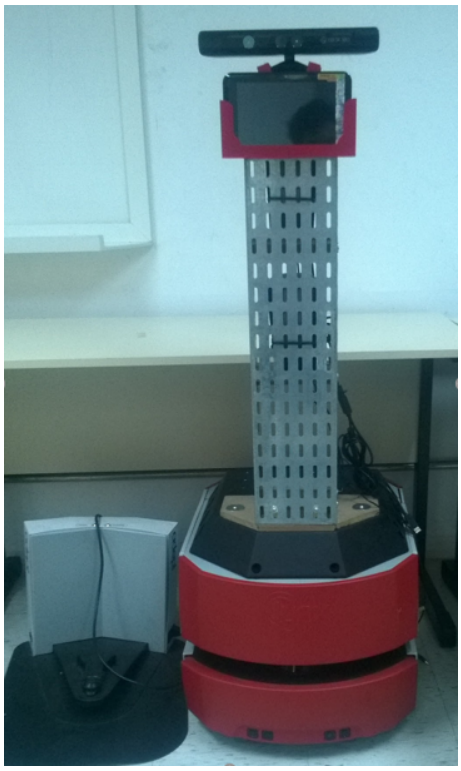

Figura 1. Pioneer LX com suporte e dispositivos extras e sua estação de recarga)

3. Software

3.1. ROS e comunicação

Sobre o robô será acoplado um notebook com a versão Indigo do ROS, sobre o sistema operacional Ubuntu 14.04 LTS. O ROS é um sistema que possui diversas ferramentas, bibliotecas e convenções que simplificam tarefas complexas e robustas para robôs [3]. Nesse sistema, é necessário executar um núcleo, o roscore, que funciona como um servidor que gerencia tópicos e nós. Tópicos são barramentos de dados nomeados nos quais nós trocam mensagens, sendo que, em um mesmo tópico, vários nós podem estar inscritos (recebendo dados) ou publicando (enviando dados). Nós são processos computacionais, aplicações propriamente ditas, e se baseiam nos valores de alguns tópicos específicos para gerar novos dados e publicar no mesmo ou em outro tópico.

O robô possui uma cinemática calculada e os encoders integrados, ele se comunica por um cabo serial-usb com o notebook por onde publica e recebe dados através do ROS. Esse robô está configurado para ser acessado por protocolo *secure shell*(SSH) a partir de outros computadores, podendo assim ser monitorado a distância por uma rede sem fio. Além disso, o acesso SSH juntamente com outro computador que contenha o ROS e a configuração do ROS para trabalhar com computadores múltiplos [7] possibilita que parte do processamento seja feito em um outro computador, sendo possível dividir o processamento apenas acoplando outros notebooks ao robô, se necessário.

3.2. Navegação e SLAM (*Simultaneous Localization and Mapping*, Localização e Mapeamento simultâneos)

Usando os dados dos *encoders* e corrigindo a posição atual estimada do robô usando de pacotes de localização autônoma baseados nas leituras do sensor laser e do Kinect, consegue-se calcular uma posição relativa aproximada do robô. O robô observa o ambiente afim de criar um modelo (mapa) do ambiente. São então realizadas medições através dessas observações para identificar onde o robô está e são definidos *landmarks* (pontos de referência) que ajudam o robô a se localizar dentro do mapa. Qualquer erro no mapa irá propagar erros na localização, mas, felizmente, os erros de estimativas de localização se correlacionam. Esses erros de correlação são corrigidos por meio do sistema EKF (*Extended Kalman Filter*) [8], que consegue armazenar a correlação entre todos os erros de estimativa e torna a localização dependente apenas da incerteza inicial de localização do robô.

Já foi implementado no robô o uso do SLAM (Localização e Mapeamento Simultâneos) [9], uma técnica utilizada por robôs e veículos autônomos para construir um mapa de um ambiente ao mesmo tempo que se localiza. Para realizar tal feito, o SLAM tem uma série de tarefas: extração de pontos de referência, associação de dados, estimação de estado, atualização de estado e atualização de ponto de referência. Para tal, dentre outras funções, ele usa do EKF. Essa implementação se deu usando funções já prontas do ROS, das quais os parâmetros foram ajustados para que a navegação tivesse bom desempenho.

3.3. Seguidor de Pessoas com Kinect

O algoritmo utilizado no robô *turtlebot* e disponível no pacote turtlebot_apps do ROS, foi adaptado para funcionar como seguidor de pessoas e cumprir esta tarefa da Robocup@Home. Estuda-se, até o prazo, integrar o nó openni_tracker com a navegação autônoma para se obter um resultado melhor do que o conseguido com o pacote turtlebot_apps [10].

3.4. Reconhecimento e síntese de voz

Para a síntese de voz, está sendo utilizado um pacote do ROS chamado *espeak_ros*. Este pacote, que reproduz vozes masculinas e femininas, em inglês e português, funciona executando um nó ROS que se subscreve ao tópico *speak_line*, e realiza a produção da voz com base nos parâmetros especificados. A configuração destes parâmetros da fala é realizada através do pacote *dynamic_reconfigure*, que permite alterar parâmetros de nós ROS sem precisar reiniciá-los. A configuração dos parâmetros de fala engloba diversas características de voz, tais como: velocidade da fala, volume, tom (mais grave ou mais agudo), espaço entre as palavras, idade e variação da entonação. Através da alteração destes parâmetros, estuda-se demonstrar emoções no robô de acordo com a situação em que ele se encontra e em conformidade com as faces exibidas no seu "rosto" (que é um tablet). A figura 3 demonstra o resultado da alteração destes parâmetros, com a face apresentando uma emoção simulada de "alegria". O reconhecimento de voz será realizado com o pacote PocketSphinx [15].

3.5. Mapeamento e visualização 3D

A Robocup@Home também contempla algumas provas extras, sendo que vai ser apresentado um projeto de mapeamento e visualização 3D, baseado no pacote octomap [13]

do ROS. Usando de dados do Kinect e da odometria. Esse pacote consegue criar uma visualização 3D baseada na ocupação ou não de voxels (subespaços), conforme pode ser visto na figura 2.

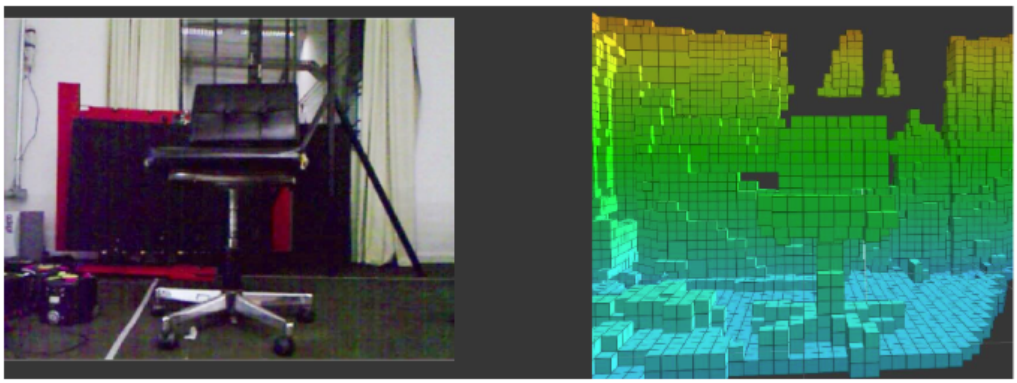

Figura 2. Visualização 3D baseada no pacote octomap – visão do Kinect a esquerda e visualiação 3D a direita.

3.6. Reconhecimento e manipulação de objetos

Estuda-se a utilização de um pacote de reconhecimento de objetos, ainda em fase de desenvolvimento, que utiliza os pacote OpenNI Kinect [6] para coleta de dados do Kinect e um algoritmo que processa e analisa os dados captados. Assim como a manipulação destes objetos reconhecidos por meio de um braço robótico em desenvolvimento.

3.7. Simulação de Emoções

Emoções são demonstradas no robô através de faces de acordo com a situação presente, para uma melhor visualização e interação entre robô e usuário. As emoções foram baseadas na roda de emoções de Plutchik [14]. Um tablet rodando o aplicativo ROS Image Viewer exibe estas faces subscrevendo-se a um tópico ROS. A figura 3 ilustra uma emoção sendo exibida pelo robô através do tablet:

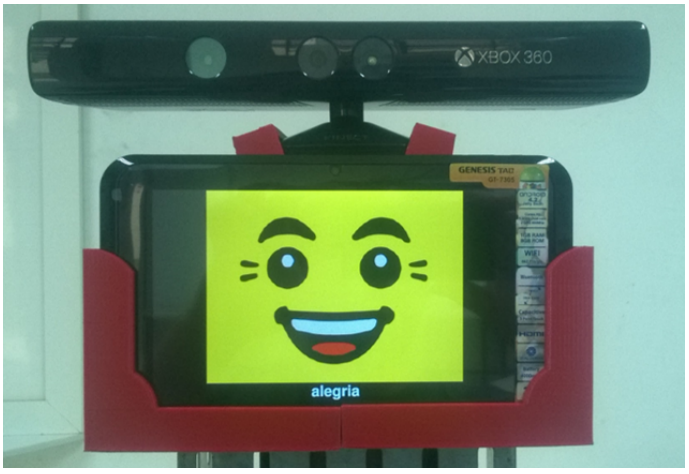

Figura 3. Emoção sendo demonstrada no *tablet*.

4. Conclusões e Trabalhos Futuros

Com os trabalhos correntes já é possível realizar diversas das provas da categoria @Home com êxito, sendo que se espera, até a data da competição, conseguir ainda mais resultados satisfatórios. Entre as tarefas a serem desenvolvidas encontram-se o reconhecimento mais acurado dos comandos de voz, e a integração entre os mapas utilizados para a navegação (SLAM) e para a visualização 3D do ambiente (octomap).

5. Agradecimentos

Os autores agradecem a Universidade Tecnológica Federal do Paraná e a Fundação Parque Tecnológico de Itaipu pelo apoio dado a este projeto.

6. Referências

1. Site oficial Robocup@Home. Disponível em: http://www.robocupathome.org/. Acesso: 10 de Junho, 2017.
2. D. HOLZ, J.R. del SOLAR, K. SUGIURA, S. WACHSMUTH, "On RoboCup@Home – past, present and future of a scientific competition for service robots". Disponível em: http://goo.gl/fi7V51. Acesso: 11 de Julho, 2016.
3. About ROS – Página oficial. Disponível em: http://www.ros.org/about-ros/. Acesso em: 11 de Junho, 2017.
4. ADEPT, Pioneer 3-AT Specifications . Disponível em: http://www.mobilerobots.com/Libraries/Downloads/Pioneer3AT-P3AT-RevA.sflb.ashx. Acesso: 11 de Junho, 2017.
5. MICROSOFT, Develop Network: Kinect Sensor. Disponível em: http://msdn.microsoft.com/en-us/library/ hh438998.aspx. Acesso: 12 de Junho, 2017.
6. V. RABAUD, T. FOOTE, Openni_kinect ROS. Disponível em: http://wiki.ros.org/openni_kinect. Acesso: 12 de Junho, 2017.
7. ROS Multiple Machines. Disponível em: http://wiki.ros.org/ROS/Tutorials/MultipleMachines. Acesso em: 12 de Junho, 2017.
8. M. I. RIBEIRO, "Kalman and Extended Kalman Filters: Concept, Derivation and Properties". Disponível em: http://users.isr.ist.utl.pt/ mir/pub/kalman.pdf. Acesso: 12 de Junho, 2017.
9. S. RIISGAARD, M. R. BLAS, "SLAM for Dummies". Disponível em: http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam_blas_repo.pdf. Acesso em: 13 de Junho, 2017.
10. D. STONIER, T. FOOTE, M. WISE, turtlebot_apps ROS wiki page. Disponível em: http://wiki.ros.org/turtlebot_apps. Acesso em: 14 de Junho, 2017.
11. V. SEIB, R. MEMMESHEIMER, D. PAULUS. A ROS-based System for an Autonomous Service Robot. Disponível em: https://svn.uni-koblenz.de/vseib/homer_ros_packages/. Acesso em: 14 de Junho, 2016.
12. A. HENDRIX, B. GASSEND, sound_play ROS wiki page. Disponível em: http://wiki.ros.org/sound_play. Acesso em: 14 de Junho, 2017.
13. A. HORNUNG, K. M. WURM, A. HORNUNG, octomap ROS wiki page. Disponível em: http://wiki.ros.org/octomap. Acesso em: 14 de Junho, 2017.
14. R. PLUTCHIK; H. KELLERMAN (Ed.). Emotion: theory, research and experience. New York: Academic press, 1986.
15. PocketSphinks. Disponível em: http://wiki.ros.org/pocketsphinx. Acesso em: 19 de Junho, 2017.