

UTBot@Home 2016 Team Description Paper

Uso do robô Pioneer 3-AT para tarefas domésticas

João Alberto Fabro^{1,2}, André Schneider de Oliveira^{1,2}
Vinicius Amilgar Brenner¹, Geovana Franco¹, Gabriel V. M. de Santana¹

¹ UTFPR – Universidade Tecnológica Federal do Paraná, Brazil (www.utfpr.edu.br)
LASER – Laboratório Avançado de Sistemas Embarcados e Robótica
(<http://laser.dainf.ct.utfpr.edu.br>)

²PPGCA - Graduate Program on Applied Computing

fabro@utfpr.edu.br, andreoliveira@utfpr.edu.br
vini_brenner@hotmail.com, geovana.franco01@gmail.com, gabriell.vieira@hotmail.com

Abstract. *This TDP describes the hardware and software implemented to allow a Pioneer 3-AT robot to execute domestic tasks related to the Robocup@Home initiative. The robot uses the information provided by the sensor to execute navigation tasks such as environment mapping and auto-localization, and the “follow-me” task. The sensor it uses for so is the Microsoft Kinect. It also executes some other Robocup@Home tasks, like the voice related ones. The robot uses ROS (Robot Operating System) and a lot of its provided packages to accomplish its tasks.*

Resumo. *Este TDP descreve o hardware e software implementados para permitir ao robô Pioneer 3-AT para executar tarefas domésticas relacionadas com a iniciativa Robocup@Home. O robô usa as informações fornecidas pelo sensor para executar tarefas de navegação, tais como mapeamento do ambiente e auto-localização, e a tarefa “siga-me”. O sensor que utiliza para isso é a Microsoft Kinect. Ele também executa algumas outras tarefas da Robocup@Home, como o reconhecimento de voz. O robô usa ROS (Robot Operating System) e diversos de seus pacotes disponíveis para cumprir as suas funções.*

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]. A fim de executar as tarefas domésticas, são desenvolvidos, dentre muitos outros campos da robótica, a interação homem-robô, a navegação e o mapeamento de ambientes dinâmicos e a visão computacional [1].

Para a execução das tarefas configurou-se um robô Pioneer 3-AT com o ROS (*Robot Operating System*), um conjunto de bibliotecas de software e ferramentas que ajudam a desenvolver aplicações para robótica [3]. 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 móvel utilizado para construir o UTBot@Home é do modelo Pioneer 3-AT [4]. Como pode ser visto na Figura 1, é um robô de 4 rodas de borracha, conectadas duas a duas por correias, que executa movimentos de rotação por derrapagem. Possui uma carcaça com 1.6mm de alumínio, pesa 12kg e mede 497mm de comprimento, 508mm de largura e 277mm de altura (Figura 2). Seus motores possibilitam uma velocidade para frente ou para trás de até 0,7m/s e uma velocidade de rotação de até 140°/s. É alimentado por até três baterias ao mesmo tempo (12V, 7.2Ah cada), possibilitando de 2 a 4 horas de autonomia.



Figura 1. Robô Pioneer P3AT, suporte para os sensores (nível intermediário) e notebook (nível superior)

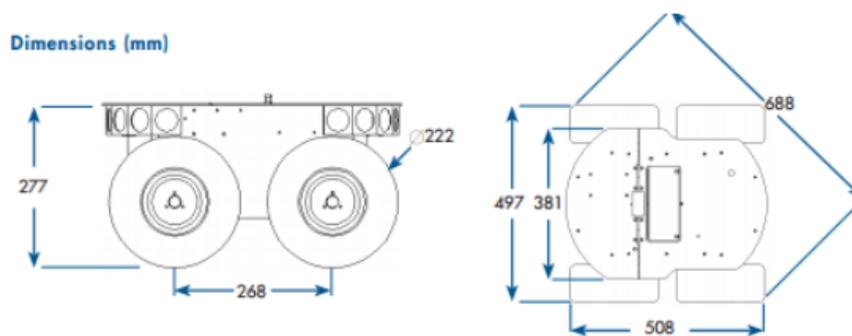


Figura 2. Dimensões do robô

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

A versão da base móvel inicial que o time possuía veio com poucos sensores de fábrica, sendo que a maioria dos sensores presentes no robô atualmente foram instalados no próprio laboratório. Os únicos sensores já presentes no robô são os *encoders* nas rodas, que permitem inferir a velocidade e posição do robô em relação a referência inicial.

Os robô atualmente conta com um acelerômetro, GPS e giroscópio, microfone e caixas de som, e um sensor de profundidade: o 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 comunicação 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 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.

3. Software

3.1. ROS e comunicação

Acoplado ao robô está 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 que computam, 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 notebook está configurado para ser acessado por SSH a partir de outros computadores, podendo assim ser controlado 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 computadores externos ao robô.

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 na leitura do Kinect, consegue-se calcular uma posição relativa aproximada do robô. O robô observa o ambiente a partir do Kinect, a fim 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 o reconhecimento de voz estuda-se a utilização do pacote *homer_speech* [11]. Esse pacote contém uma adaptação para o ROS de dados reconhecidos pelo aplicativo instalado em um *smartphone* Android (*homer_speech.apk*). Até o momento atual, ainda encontra-se em desenvolvimento a integração completa entre o pacote de reconhecimento de voz e os comandos do robô.

Quanto a síntese de voz é utilizado outro pacote do ROS: o *sound_play* [12]. Esse pacote recebe uma sequência de caracteres e sintetiza a fala em inglês. O pacote não

requer muita configuração e funciona de forma satisfatório para a maioria das palavras que se utilizaram. Após estudos de comportamento do pacote foi possível se adquirir um domínio do pacote e fazer com que a interação com o usuário ocorra de forma adequada.

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 3.

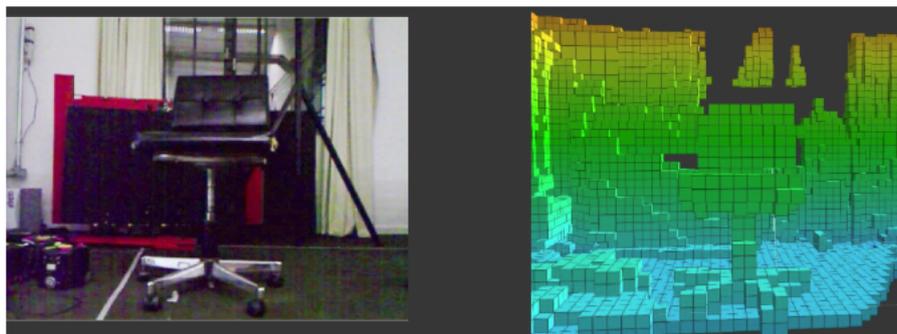


Figura 3. Visualização 3D baseada no pacote octomap – visão do Kinect a esquerda e visualização 3D a direita.

3.6. Reconhecimento de objetos

Estuda-se a utilização de um pacote de reconhecimento de objetos, ainda em fase de desenvolvimento, que utiliza os pacotes OpenNI Kinect [6] para coleta de dados do Kinect e um algoritmo que processa e analisa os dados capturados.

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 julho, 2015.
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, 2015.
3. About ROS – Página oficial. Disponível em: <http://www.ros.org/about-ros/>. Acesso em: 11 de junho, 2016.
4. ADEPT, Pioneer 3-AT Specifications . Disponível em: <http://www.mobilerobots.com/Libraries/Downloads/Pioneer3AT-P3AT-RevA.sflb.ashx>. Acesso: 11 de junho, 2016.
5. MICROSOFT, Develop Network: Kinect Sensor. Disponível em: <http://msdn.microsoft.com/en-us/library/hh438998.aspx>. Acesso: 12 de junho, 2016.
6. V. RABAUDE, T. FOOTE, openni_kinect ROS wiki page. Disponível em: http://wiki.ros.org/openni_kinect. Acesso: 12 de junho, 2016.
7. ROS Multiple Machines wiki page. Disponível em: <http://wiki.ros.org/ROS/Tutorials/MultipleMachines>. Acesso em: 12 de junho, 2016.
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, 2016.
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/1aslambblas_repo.pdf. Acesso em: 13 de junho, 2016.
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, 2016.
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, 2016.
13. A. HORNUNG, K. M. WURM, A. HORNUNG, octomap ROS wiki page. Disponível em: <http://wiki.ros.org/octomap>. Acesso em: 14 de Junho, 2016.