

UTBot@Home 2014 Team Description Paper - Uso do robô Pioneer 3-AT com sensores Kinect, acelerômetro e giroscópio em tarefas de navegação doméstica.

Rodrigo Longhi Guimarães , André Schneider de Oliveira, João Alberto Fabro e
Thiago Becker

UTFPR – Universidade Tecnológica Federal do Paraná, Brazil

LASER – Laboratório Avançado de Sistemas Embarcados e Robótica

rolongui@yahoo.com.br, andre@dainf.ct.utfpr.edu.br, fabro@utfpr.edu.br, beckerthiago@gmail.com,

Abstract. This TDP describes the hardware and software implemented to allow a Pioneer 3-AT robot to execute navigation tasks related to the Robocup@Home initiative. The robot was connected to a set of sensors (Kinect, accelerometer and gyroscope), and a software stack, based on ROS (*Robot Operating System*), that uses the information provided by the sensors to execute navigation tasks such as environment mapping, auto-localization, and the “follow-me” task.

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]. A fim de executar as tarefas domésticas, são desenvolvidos, dentre muitos outros campos da robótica, a interação homem-robô, a navegação e mapeamento de ambientes dinâmicos e a visão computacional. Uma das tarefas desta categoria é a “Follow Me”, na qual o robô tem que seguir uma pessoa num ambiente onde há muitas outras pessoas, entrar em um elevador com a pessoa que está seguindo, sendo que, durante a prova, é permitido que pessoas passem entre o robô e a pessoa que ele está seguindo e, em certos pontos, a pessoa que o robô está seguindo se esconde, fazendo com que o robô tenha que readquirir sua posição[2].

Para controlar o robô é utilizado 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 do 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 já obtidos e expectativas de projetos em desenvolvimento para o robô. A seção 2 apresenta uma descrição do hardware do robô, incluindo suas especificações, as características dos ajustes realizados (suporte para notebook e sensores, interface entre o notebook e os sensores embarcados, e também com o sensor Kinect). 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 (12V, 7.2Ah cada), possibilitando de 2 a 4 horas de autonomia.



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

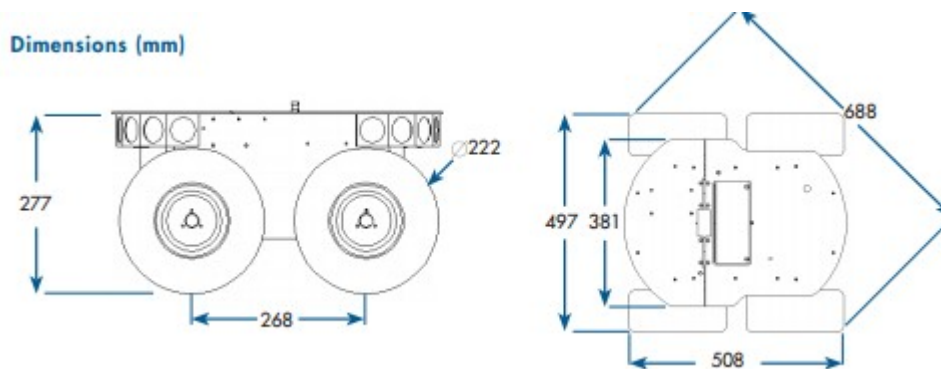


Fig. 2. Dimensões do robô

2.2 Suporte

Para suportar o notebook e os sensores do robô foi construído um suporte em alumínio (como pode ser visto na Figura 1) com dois níveis, com placas de alumínio de 30cm por 30cm, sendo que o primeiro nível está a 14,5cm do robô e o segundo a 30,5cm, fazendo com que o robô tenha uma altura total de aproximadamente 59cm. As placas são sustentadas por barras roscadas de 7mm de diâmetro, presas com roscas auto-travantes e com barras de alumínio em volta.

2.3 Sensores

A versão do robô que o time possui veio com bem 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, que usamos para saber como o robô se movimentou dentro do ambiente, são encoders nas rodas. Os encoders funcionam gerando um certo número fixo de pulsos por volta da roda e publicando esses dados em um tópico do ROS, sendo que para tal consideram que o robô sempre se locomove usando a parte central da roda e que não há deslizamento. Como essas suposições usadas no cálculo da posição relativa do

robô acabam tornando o resultado final impreciso, são necessários outros sensores para confirmar a posição do robô.

Além dos encoders, foram instalados mais três sensores usados no cálculo da posição do robô: um acelerômetro, um giroscópio e um GPS. O primeiro é capaz de medir a aceleração linear dos robôs nos três eixos cartesianos: x , y e z ; É possível ler de seus registradores um número inteiro de 16 bits que, após convertidas as unidades de forma correta, apresenta a velocidade em m/s^2 nos três eixos. De forma bastante semelhante, o giroscópio realiza a leitura da velocidade angular de *roll*, *pitch* e *yaw* (balanceio, empinamento e cabeceio). Por fim, o GPS obtém a latitude e longitude (além de algumas outras informações que não se aplicam ao escopo do projeto) através de informações provenientes de satélites e repassa essas informações ao sistema de controle do robô. Para a leitura desses sensores foi usado um circuito micro-processador Arduino Mega, que lê as informações do acelerômetro e do giroscópio por meio do protocolo de comunicação SPI (*Serial Peripheral Interface*) e as informações do GPS por meio do protocolo serial padrão. As informações dos sensores são interpretadas e publicadas no ROS através de uma biblioteca *open-source* chamada *rosserial-arduino*, que estabelece uma comunicação entre o Arduino e o notebook, onde está sendo executado o ROS.

Um sensor do tipo *sonar* também foi instalado na parte da frente do robô, sendo que o pulso gerado por ele varia com a distância, de forma que o Arduino interpreta o tamanho desse pulso e o converte em um valor de estimativa de distância em metros. Foi feito um programa teste usando o valor desse sensor, fazendo com que o robô pare quando chegar perto demais de um obstáculo.

Por fim, o robô também conta com um sensor de visão computacional: o Kinect da empresa Microsoft. Esse sensor é capaz de criar uma imagem em profundidade, posicionar suas câmeras verticalmente, rastrear a silhueta de uma pessoa, identificar comandos de voz, 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

Foi instalado em um netbook que fica sobre o robô a versão *Fuerte* do ROS, sobre o sistema operacional Ubuntu 12.04 LTS. O ROS é uma estrutura (*framework*), que traz uma grande coleção de ferramentas, bibliotecas e convenções que visam simplificar a tarefa de criar aplicações complexas e robustas para robôs[3]. Nesse sistema, é necessário executar um núcleo (*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 já possui a cinemática calculada e os encoders integrados, sendo que por um cabo serial-usb ele se comunica com o notebook e publica e recebe dados por meio do ROS. Esse notebook está configurado para ser acessado por login remoto (SSH) a partir de outros computadores, podendo assim ser controlado a distância por uma rede sem fio. Além disso, o acesso remoto juntamente com outro computador que contenha o ROS e a configuração do ROS para trabalhar com computadores múltiplos[7] possibilita que o processamento pesado 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 uma correção propiciada por cálculos a partir dos dados do acelerômetro e do giroscópio é possível conseguir uma posição relativa bastante precisa. Usando do GPS,

pode-se conseguir uma posição absoluta, mas além de ela não ser muito precisa (tem cerca de 3m de erro), é bastante difícil usar esse sensor em ambientes fechados, pois ele não consegue se comunicar com o satélite. Dessa forma, o método usado para a navegação no projeto é interpretar os dados de imagem provindos do Kinect.

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.

3.3 Seguidor de Pessoas com Kinect

No intuito de cumprir a tarefa proposta da Robocup@Home, nos basearemos no trabalho de Garratt Gallagher [10], que, usando dos mesmo drivers que estamos usando para o Kinect (OpenNI) e também da biblioteca NITE, que têm funções de rastreamento de esqueleto a partir da imagem capturada pelo Kinect, desenvolveu um código que faz o robô seguir uma pessoa.

4 Conclusão

A visão computacional e o algoritmo de SLAM ajudarão o robô a se localizar e locomover, sendo que, junto com um algoritmo baseado no projeto de Garratt Gallagher, será possível rastrear pessoas e segui-las, a fim de cumprir a tarefa proposta da Robocup@Home.

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: 11 de agosto, 2014.
- [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 agosto, 2014.
- [3] About ROS – Página oficial. Disponível em: <http://www.ros.org/about-ros/>. Acesso em: 11 de agosto, 2014.
- [4] ADEPT, Pioneer 3-AT Specifications. Disponível em: <http://www.mobilerobots.com/Libraries/Downloads/Pioneer3AT-P3AT-RevA.sflb.ashx>. Acesso: 11 de agosto, 2014.
- [5] MICROSOFT, Develop Network: Kinect Sensor. Disponível em: <http://msdn.microsoft.com/en-us/library/hh438998.aspx>. Acesso: 12 de agosto, 2014.
- [6] V. RABAUD, T. FOOTE, *openni_kinect* ROS wiki page. Disponível em: http://wiki.ros.org/openni_kinect. Acesso: 12 de agosto, 2014.
- [7] ROS Multiple Machines wiki page. Disponível em: <http://wiki.ros.org/ROS/Tutorials/MultipleMachines>. Acesso em: 12 de agosto, 2014.
- [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 agosto, 2014
- [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/1aslamb_blas_repo.pdf. Acesso em: 13 de agosto, 2014.
- [10] G. GALLAGHER, Kinect-based Person Follower ROS news page. Disponível em: <http://www.ros.org/news/2010/12/kinect-based-person-follower.html>. Acesso em: 13 de agosto, 2014.