**Executando a Pilha de Navegação no robô P3AT**

**PIONEER - 3AT**

**HARDWARE:**

O robô do modelo Pionner-3At é um robô de 4 rodas de borracha, conectadas 2 a 2 por correias que executam o movimento de rotação por derrapagem. Possui uma carcaça de 1,6mm de alumínio, pesa 12 kg e mede 497mm de comprimento, 508mm de largura e 207mm de altura. Seus motores possibilitam uma velocidade para frente ou para trás de até até 0,7m/s e uma velocidade de rotação de até 140º/s. É alimentado por até três baterias ao mesmo tempo(12V, 7.2Ah cada), possibilitando de 2 a 4 horas de autonomia.

Possui sensores de odometria, giroscópio, óptico, além de suporte para sensores e para notebook em sua parte superior.

**SOFTWARE**

ROS e comunicação -

Foi instalado em um computador acoplado ao robô a versão Kinetic do ROS, sobre o sistema operacional Ubuntu 16.04.4 LTS. O ROS é uma estrutura (framework), que traz uma coleção de ferramentas, bibliotecas e convenções que visam simplificar a tarefa de criar aplicações complexas e robustas para robôs. 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 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 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 possibilita que o processamento pesado seja feito em computadores externos ao robô.

ROS e RosAria -

Para comunicar e controlar o robô vamos utilizar a biblioteca RosAria, que é mantida e desenvolvida pela Pioneer para os seus robôs.

É necessário construir o nó RosAria e executá-lo.

vide: <http://roboticadecampo.blogspot.com/2016/06/primeiro-experimento.html?m=1>

O RosAria deve ser instalado para que informações da base do robô e controle de velocidade e aceleração sejam implementadas por meio de um nó RosAria , que publica tópicos fornecendo dados recebidos do controlador embarcado do robô pela ARIA e define a velocidade, aceleração e outros comandos desejados no ARIA quando novos comandos são recebidos de tópicos de comando. O ARIA pode controlar dinamicamente a velocidade, o rumo, o rumo relativo e outros parâmetros de movimento do seu robô, seja por meio de comandos simples de baixo nível ou por meio de sua infraestrutura de Ações de alto nível. O ARIA também recebe estimativas de posição odométrica, leituras de sonar e todos os outros dados operacionais atuais enviados pela plataforma do robô. Assim, o ROSARIA utiliza do pacote ARIA para se comunicar com o robô e determinar os dados dos sensores.

Um ponto importante é que sempre devemos clonar os pacotes para dentro da pasta src e rodar o catkin\_make na pasta acima.

ROS e Stack Navigation -

A navegação tem o objetivo de criar um mapa do ambiente para que o robô possa obter uma orientação de sua trajetória.

Na realização desse caminho o robô pode se chocar com qualquer objeto que esteja inserido no ambiente. Então é preciso de um planejamento de rotas que permita que o robô faça o seu caminho sem que colida em algum objeto (BERG et al., 2008; LATOMBE, 2004). Esse problema foi caracterizado por J. T. Schwartz e M. Sharir, em 1983, como “O Problema do Movimento de Pianos”, e generalizado por J. Reif como o “Problema do Móvel Generalizado”.

Numa preocupação de trazer ganhos na geração de rotas para que um Robô possa percorrer, com melhor desempenho em sua autonomia, este trabalho tem como proposta implementar a Pilha de Navegação (Navigation Stack) como algoritmo de navegação no Robô de esteira, tendo, este, um modelo para a aplicação de navegação inteligente que proporciona a autonomia de máquinas.

O Navigation Stack é um algoritmo de navegação que coleta informações de odometria (método utilizado para estimar a posição de um robô), fluxos de sensores e posição do objeto e gera comandos de velocidade seguros que são enviados para uma base móvel. O primeiro passo deste algoritmo é mapear o ambiente em que o robô se encontra para traçar uma rota segura, sem que haja colisão com obstáculos, logo em seguida, esse caminho é desenhado através de pontos para que o robô possa seguir de acordo com a trajetória descrita pelo método de navegação.

O navegador deverá ser implementado em uma placa Raspberry Pi (RPI), um Single Board Computer de baixo custo tendo como sistema operacional o ROS, específico para robôs, que está integrado ao robô esteira com a função de realizar operações de alto nível e processar as informações trocadas com um microcontrolador.

Assim, será possível inserir o robô em um espaço conhecido e realizar diversas tarefas com ele indicando apenas por onde deve percorrer, o tornando relativamente autônomo.

Uma pilha de navegação 2D que coleta informações de odometria, fluxos de sensores e uma pose de meta e gera comandos de velocidade segura que são enviados para uma base móvel.

Controlando a Stack Navigation -

A pilha de navegação é rodada baseada no ROS (sistema operacional para robôs), assim, é necessário fazer a instalação do ROS e seus periféricos que lhe darão estrutura para executar as tarefas da navigation stack.

A estrutura do ROS é desenvolvida através de nós que se comunicam uns com os outros para realizar as tarefas propostas.

vide:<https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&prev=search&rurl=translate.google.com&sl=en&sp=nmt4&u=http://wiki.ros.org/ROS/Tutorials&xid=17259,15700023,15700186,15700190,15700256,15700259&usg=ALkJrhjDwBA2FYJaOJOjVzkN9X2Nt0vGOw>

A configuração da máquina deve ser feita para que haja a compatibilidade entre o robô e o ROS. Neste caso, foi instalado o Kinetic para o P3AT. Ainda, é essencial que se use a API TF [tf é um pacote que permite ao usuário acompanhar vários quadros de coordenadas ao longo do tempo], como uma thread.

A pilha de navegação capta os sensores do robô, sendo o de odometria(exigindo a importância de utilizar o TF) e o sonar. Além disso, é recomendável que se utilize um mapa (gmapping - map-server) construído a partir de dados de varredura de transformação.

. As informações da base do robô e controle de velocidade e aceleração são implementadas por meio de um nó RosAria, que publica tópicos fornecendo dados recebidos do controlador embarcado do robô pela ARIA e define velocidade, aceleração e outros comandos desejados no ARIA quando novos comandos são recebidos de tópicos de comando. O ARIA pode controlar dinamicamente a velocidade, o rumo, o rumo relativo e outros parâmetros de movimento do seu robô, seja por meio de comandos simples de baixo nível ou por meio de sua infraestrutura de Ações de alto nível. O ARIA também recebe estimativas de posição odométrica, leituras de sonar e todos os outros dados operacionais atuais enviados pela plataforma do robô. Assim, o ROSARIA utiliza do pacote ARIA para se comunicar com o robô e determinar os dados dos sensores. Em seguida, é necessário a instalação e configuração da Pilha de Navegação.

Observações:  
ttyS0 - é o dispositivo para a primeira porta serial UART para LINUX  
ttyUSB0 é o dispositivo para o primeiro conversor serial USB para LINUX.

O próprio RosAria já possui características específicas de publicação de tópicos, assim, não é necessário instalar a pilha do zero, como no tutorial do ROS.

Com a instalação da Pilha de Navegação feita ($ sudo apt-get install ros-kinetic-navigation), deve-se implementar o conjunto de arquivos que compilem a fim de executar a pilha, ativando, assim, os sensores a serem utilizados, neste caso, sem o laser.

Execução e controle da pilha de navegação passo a passo:

ssh p3at02@192.168.1.37

Roscore: roscore

RosAria: rosrun rosaria RosAria \_port:=/dev/ttyS0

$ sudo chmod 777 -R /dev/ttyUSB0

$ rosrun rosaria RosAria

Navigation Stack: roslaunch p3at\_2dnav p3at-teste.launch

$ rviz

[WhatsApp Image 2019-06-05 at 11.46.44.jpeg](https://trello-attachments.s3.amazonaws.com/5ba94c760a258554ed448a2b/5cba28d0b83f5066cef6835f/686695c0fb79e51bce808784286b3aec/WhatsApp_Image_2019-06-05_at_11.46.44.jpeg)

importante deixar como master - ROS\_MARTER\_URI

Desta forma, foi necessário abrir o Rviz para executar e visualizar o mapa da pilha de navegação.