

Robô Recepcionista Pioneer P5-DX

Pioneer 5

Pedro Isidro - 67220, Diogo Silva - 75136, João Pedrosa - 79833

Resumo—Abstract (summary of the work)

Index Terms—Pioneer P5-DX, ROS, autónomo.

I. INTRODUÇÃO

Os requisitos para do nosso projecto exigiam ter um robô Pioneer P3-DX capaz de executar tarefas modeladas por uma máquina de estados finitos. Essas tarefas incluem o robô ser capaz de mapear, auto-localiza-se e navegar o seu ambiente e deslocar-se para diferentes localizações indicadas através de interação com utilizadores.

A nossa motivação foi ter um robô recepcionista capaz de receber visitantes num edifício, e.g. um bloco de escritórios. O robô seria capaz de mapear o edifício *a priori* para futura utilização, de receber um conjunto de coordenadas já conhecidas (e.g. escritórios de certas pessoas) associadas a palavras-chave, de interagir com utilizadores através de síntese e reconhecimento de voz e, por fim, de guiar os mesmos aos seus destinos.

Durante a implementação do projecto simplificámos algumas destas funcionalidades, nomeadamente o mapeamento e o reconhecimento de voz. O mapeamento foi executado apenas no quinto piso da Torre Norte do IST. O reconhecimento de voz foi restringido a um pequeno dicionário para reduzir a complexidade e a necessidade de adaptar modelos acústicos e treinamento.

December 6, 2013

II. ALGORITMOS E IMPLEMENTAÇÃO

A. Mapeamento

To do the Mapping of 5 floor, we use the gmapping.

Hokuyo node:

The Hokuyo node obtain the data of Hokuyo connected to the computer and publishe in topic scan.

Gmapping:

The gmapping get the information in topics tf and scan. The topic tf transforms necessary to relate frames for laser, base, and odometry. The topic scan transforms Laser scans to create the map from. The gmapping give the topic map for creating the map.

Initial Map creating in gmapping

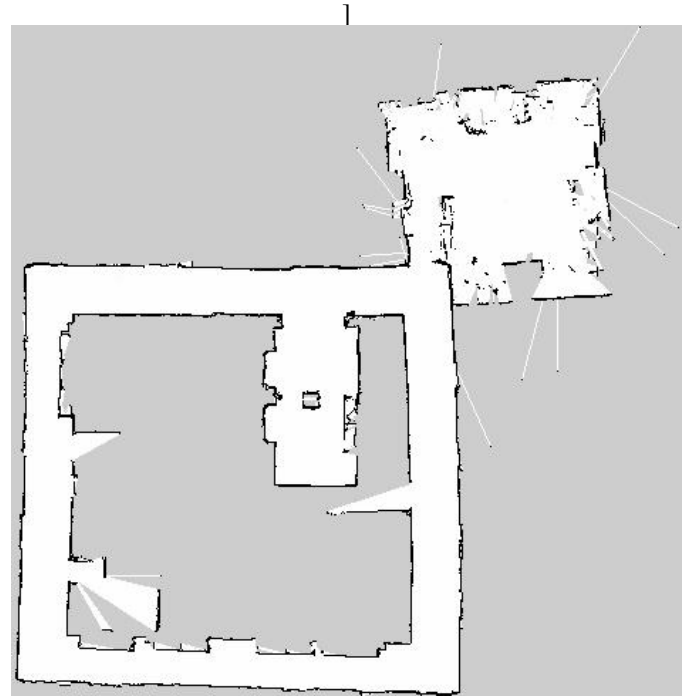


Figura 1. Initial Map

B. Localização

C. Navegação

For doing the Navigation the used commands given by the user

Odometry:

The pose have tree parameters: x , y , θ

Algorithm:

We use Rosaria to obtain odometry information for doing the localization in the map, the odometry information obtain in topic pose published by Rosaria. Another thing we use Rosaria is for reading the sonar, give by the topic sonar.

D. Execução do Plano Coordenado

In order to coordinate the robot's actions, an abstract representation of the task to be carried out is needed. The receptionist robot is a sequential system, *i.e.*, it performs one action at a time – either moving to a target location or interacting with a user to acquire one. For such a system,

a State Machine (SM), or Finite State Automaton (FSA) is usually used to specify the control flow through the system.

A SM is implemented using the *smach* package. The task of the robot is divided into three states:

- INITIAL
- TO_GOAL
- GET_GOAL

E. Interação com o utilizador

A interação com utilizadores foi feita com recurso a reconhecimento e síntese de voz. Para o primeiro, foram testadas duas soluções com características muito distintas. Para o último, foi utilizado um sintetizador de som com capacidade de sintetizar voz a partir de um texto recebido.

1) *Reconhecimento de voz*: Para compreender como é que o reconhecimento de voz é realizado é necessário introduzir alguns conceitos importantes. A unidade básica da fala é entendida como um fone. Contudo, as características acústicas correspondentes a um fone variam conforme o contexto em que esse fone aparece, a pessoa, etc. Devido a estas variações utilizam-se subestados dentro de um fone para melhorar o reconhecimento. Usa-se, também, o contexto em que os fones aparecem, o que se irá traduzir num problema de procura do contexto que melhor se aproxima dos dados recebidos. Os fones constróem sílabas. Dependendo das condições de fala, a mesma sílaba corresponde a diferentes fones. Palavras restringem consideravelmente a quantidade de fones que se têm de comparar e quanto menor for o dicionário mais rápido será o reconhecimento.

Uma das soluções testadas foi a livreria Pocketsphinx da plataforma Sphinx. Esta livreria, disponível num pacote do ROS com o mesmo nome, está optimizada para portabilidade que é exactamente o que pretendemos na implementação num robô. A plataforma Sphinx utiliza algoritmos extensivamente utilizados em investigação baseados em *Hidden Markov Models* (HMM). Os principais componentes do processo de reconhecimento de voz estão representados na Figura 2.

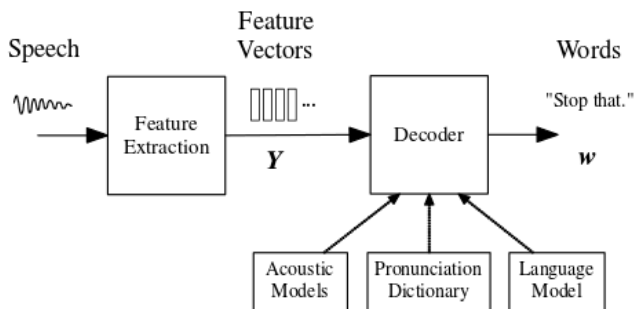


Figura 2. Principais componentes do processo de reconhecimento de fala.

O audio recebido pelo microfone é convertido em numa sequência de vectores acústicos $Y_{1:T} = y_1, \dots, y_T$, num processo denominado de extração de *features*. Os números dos vectores são baseados nas propriedades acústicas da fração correspondente da sequência. Depois desta extração, o decodificador tenta encontrar a sequência de palavras $w_{1:L} =$

w_1, \dots, w_L que é mais provável ter estado na origem de Y , i.e.,

$$w = \operatorname{argmax}\{P(w|Y)\} \quad (1)$$

No entanto, $P(w|Y)$ é difícil de modelar directamente, pelo que se utiliza a regra de Bayes para transformar 1 no equivalente:

$$w = \operatorname{argmax}\{P(Y|w)P(w)\} \quad (2)$$

A complexidade da implementação dos algoritmos nesta plataforma é algo que não foi explorado na implementação do projecto, mas segue o processo descrito nos parágrafos anteriores.

O reconhecimento é feito a partir de uma amostra que é separada em diferentes partes por silêncios ou pausas e depois tenta-se identificar o que é dito em cada uma dessas partes comparando com todas as combinações de palavras. O resultado é a melhor combinação possível. Esta comparação tem em conta três conceitos importantes. O primeiro é o de *features* - a fala é dividida em pequenas partes (frações de segundo) e de cada parte é extradido um conjunto de números designado por vector de *features*. O segundo é o modelo utilizado - objecto matemático que codifica propriedades da fala, normalmente baseado no espectro. Por último, é o processo de procura em si, que pode ser mais ou menos simplificado para devolver resultados em tempos com utilização prática. A forma como o vector de *features* é calculado, o quão o modelo utilizado se aproxima da realidade e o uso ou não de simplificações ou optimizações do processo de procura vão influenciar a qualidade do reconhecimento de fala.

Uma das soluções testadas para reconhecimento de voz foi um *wrapper* do *Google Speech API*. Esta solução tem a grande vantagem de nos dar acesso a uma plataforma poderosa de reconhecimento de voz.

2) *Síntese de voz*: O uso desta ferramenta tem duas grandes vantagens e uma grande desvantagem. As vantagens é que não precisa de ser treinada e é independente da pessoa, i.e., funciona em qualquer pessoa. A desvantagem é que é muito lenta pois a amostra de som tem de se enviada para os servidores da Google que irão depois retornar uma resposta com o que foi dito e com um grau de certeza.

Como temos o grau de certeza, podemos usar isso estabelecer um */threshold/* para o qual o robô obedece de imediato a uma acção ou repete o que pensa que recebeu e pergunta se foi realmente isto que foi dito. Um simples 'Yes' ou 'No', à partida, é algo que será sempre reconhecido e reduz acções incorrec

F. Visualização

The project was developed on the Robot Operating System framework, which is open-source and contains a myriad of different off-the-shelf packages ready for use.

III. RESULTADOS

IV. CONCLUSÃO

APÊNDICE A

TITLE OF APPENDIX A

Appendix A text goes here. desired:

REFERÊNCIAS

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.