

NAO - II



MAJOR PROJECT – SSAD

ADVISOR : Dr. Kamalakar Karlapalem

MENTOR : Raghudeep

SSAD MENTOR : Manish Jha

GROUP : Group20

MEMBERS :

Harshit Daga (200801028)

Aman Saxena (200801010)

TASK ALLOCATED TO TEAM

- We were supposed to calculate the error in the localisation in the NAO robot .. In general words .. Say we assume a map given in terms of rectangular coordinates .. And we assume that the robot is standing at some (x,y) after some given operation say moving from some other position.. But actually it had not reached that point because of the errors in the motor and the movement of the robot .. So we were supposed to calculate that θ, h, k to move so that it could reach the desired position

ABOUT NAO

- ❖ *Nao is an autonomous, programmable and medium-sized humanoid robot, developed by the French company Aldebaran Robotics.*
- ❖ *Its is designed for entertainment purposes , and will be able to interact with its owner, with evolving behaviors and functionalities. Additionally, the user will be able to teach Nao new behaviors using a computer with Wi-Fi connectivity. The behaviours creating software is designed to fit any users level: from graphical block editing for beginners to code for more skilled users. The possible behaviors are limited only by our imagination !*

❖ Nao is based on a Linux platform and scripted with Urbi, an easy-to-learn programming language, with the option of a graphic interface for beginners or code commands for experts.

❖ On August 15, 2007, Nao replaced the robot dog Aibo by Sony as the standard platform for the Robocup ("Robot Soccer World Cup"), an international robotics competition

SPECIFICATION

N

Technical Specification :

Height : 58cm

Weight : 4.3kg

Autonomy : 45min

Degree of freedom : 21 to 25

CPU : x86 AMD GEODE

500MHz

Built-in OS: Linux

Compatible : Windows,

MacOS, Linux

Programming languages :

C++, C, Python,

Urbi

Vision : Two CMOS 640 x 480

cameras

Connectivity : Ethernet, Wi-Fi

- All versions feature an inertial sensor and 4 ultrasound captors that provide stability and positioning within space to NAO.
- Nao also features a powerful multimedia system (4 microphones, 2 hi-fi speakers, 2 CMOS cameras) for text-to-speech synthesis, sound localization or facial and shape recognition amongst various other abilities.
- The package includes a dedicated programming software, Aldebaran Choregraphe and Nao is compatible with Microsoft Robotics Studio, Cyberbotics' Webots and Gostai Urbi Studio.

WHAT WE DID ??

Our Task was divided into 2 parts :

First Part was making the C/C++ Code.

Second Part was putting the code on NAOqi.

First Task :

C/C++ Code

In this task we made a C/C++ code that satisfied the required conditions.

Were given

1. The matrix.
2. The standing position of the robot.
3. Number and the co-ordinates of the obstacles.(consider obstacles as stationary)
4. Theta , h , k as an input taken by the user so generate a virtual real map which have been got by sensor readings if working on robot.

We made this code in C and this is the code that can be used for any general robot for the purpose of localisation.

Second Task :

NAOqi Part : We put the C code modules on the NAOqi so that the general code that we made for any robot is now compatible with the simulator on which NAO robot works thus we register our modules in NAOqi for any other person to use it .

SAID AND DID

SAID

- We broke it into two modules : error generation and graphical interphase .
- In error generation part , we will be calculating error on the basis of censored readings we will be getting .
- Graphics generation using glut/gl library.

DID WITH PROBLEMS AND REASON OF NOT DOING THAT

- As we were not able to take the sensor readings from the robot (reason mentioned in next few slides) we just made a C/C++ code general for all the robots that then can be changed according to the robot simulator. But the task was still completed if seen in coding terms.

PROBLEMS FACED THAT STOPPED ROBOT MOVEMENT

- Task was given to make the robot move to the point that will be specified in the input as the standing position of the robot as because of the error the robot is not at that point. So we were supposed to take the sensor readings from robot's eye and then using these sensor readings and from the ideal map that we know we use geometry to calculate the (θ, h, k) that the robot should move in order to reach the desired point. But for the sensor reading can be obtained from 2 sensors provided in the robot
 - One is a set of **Infrared Sensors** located in its eyes
 - Second is a 4 **Ultrasonic Sensors** located on its chest.

Problem :

But since there is no API developed till now to get the reading of the infrared sensors from the NAO robot memory so we were not able to use these sensor for the purpose of getting the actual obstacle position from the place the robot is standing . And if we use the chest sensor then in that case robot need to rotate and this will introduce a great amount of error in our calculation.

Solution :

As we can't introduce more error by taking the sensor reading from the robot's chest as it needs to be rotated so we just made a C/C++ code and NAOqi part considering the map and ideal situation given as input and then generating the virtual map that will be seen as per by the robot using the mathematics. And thus we got the 2 set of points as ideal map points and the points as seen by the robot and then performing the localisation algorithm on these set we calculated the θ , h , k the robot needs to move in order to reach the correct position on the map ...

WHAT WE LEARNED AND IDEAS THAT WE HAVE

Learn ed/Overvied :

**The localization
algorithm*

**Coding part of
Robotics.*

**About NAO.*

Ideas :

**Virtual Map*

Generation .

**Usage of previous data*

.

Learned

- 1.) Localization Algorithm : How the error in the robots location can be corrected by using the maths
- 2.) Coding part of Robotic : We got an overview of some what how the robots is coded and how these code are used to control the robot .
- 3.) About NAO : We got to know about (overview) of the working of the NAO ... how this robot is used to play a game like football ...how it get the idea of the obstacles etc ...

WHAT WE LEARNED AND IDEAS THAT WE HAVE

Learn ed/Overvied :

- *The localization algorithm*
- *Coding part of Robotics.*
- *About NAO.*

Ideas :

- *Virtual Map Generation .*
- *Usage of previous data .*

Ideas:

- 1.) Virtual Map Generation : In the localisation algorithm we generated a virtual map as we were not having the sensor reading. In apart from what was specified in the algorithm to calculate the virtual map as seen actually by the robot we just figured out another way of generating the set of points and what we felt was that that points we more close to what actually we can get from the sensor reading (as we were provided the θ , h , k as in the input).
- 2.) Usage of Previous Data : Since we were not able to work on the robot its self but still what we thought was this ... We thought to move the robot step by step to its correct position but since after every step we need to get the sensor reading again and it will take time to get the readings so we decided that since after every step we are improving the error so if we begin with a particular number of readings and after every step we decrease this number then we can reduce some time (and it is sure that we need to every time take the sensor reading because of various factors in place and cant use the stored one ...)



Thank You

-Harshit Daga

-Aman Saxena