

Autonomous Rover Controlled by NAO

Ramkumar A, Sasidaran K, Yogesh M, Sureshkumar R



Abstract: In this project, a new approach of a Humanoid robot controlled mobile platform is presented. The robot we are using is Nao by Aldebaran Robotics. The main reason for choosing Nao is for its flexibility and versatility. Nao's wide range of movements allows it to perform a steering mechanism in a real world. The Driving behavior for Nao is done by using NAOqi python SDK and through using Raspberry Pi controller interface. We use the HC-SR04 ultrasonic sensor for obstacle range detection in the navigation system to avoid the obstacle in real time. The system architecture and algorithms used in each stage are described in this project.

Keywords: Nao humanoid Robot, Navigation, Raspberry Pi, Ultrasonic sensor, NAOqi.

I. INTRODUCTION

NAO, a humanoid robot which is used by academic institutions, healthcare industries and research laboratories. It is originally developed by Aldebaran robotics and later acquired by Softbank Group in 2015. The robot can be programmed to walk, speak, and respond to queries and recognizing the faces [1].

It is made up of a distinct composition of hardware and software. Naoqi, the operating system of NAO which controls motors, sensors and Softwares. Such composition of technologies provides NAO the skill to detect its surroundings. Using the operating system, NAO is able to process what it has detected and activated programmed responses [7], [8].

A. Nao Features

NAO with 25 degrees of freedom and a human like structure that make him to adapt to the environment around it. Inertial sensor provides balanced support and helps NAO to know whether it is standing or lying down.

The tactile sensor in head, hands, feet, as well as ultrasonic sensor, provides valuable input about the environment [5].

NAO can respond to queries with the help of microphones and loudspeakers around its head. NAO gets full level of

support and integration in both English and French but it can also support other languages with certain limitations.

NAO can classify objects and shapes with the help of two cameras attached to the head.

For the first time the robot needs to be configured for internet connection and later it can connect to internet autonomously using ethernet port and Wi-Fi. There are few limitations for using artificial intelligence algorithms but NAO can replicate human behavior.

B. Naoqi-Nao OS:

Naoqi, the operating system of NAO which runs on it and helps to control the actuators and sensors. The NAO robot is programmed using framework called Naoqi Framework. The framework provides the basic requirements of robot-like resource, parallelism, events and synchronization. It also supports in uniform communication among different modules like motion, video and audio, standard programming and information sharing.

Naoqi is a Linux distribution based on Gentoo, especially developed for Aldebaran robot needs.

The framework can be developed in cross platform like Mac or Windows and it uses cross language with indistinguishable API for both python and C++.

Inbuild API can be accessed from both C++ as well as from python language. Most of the time behaviors are developed in python and services in C++.

II. NAVIGATION OF MOBILE ROBOT

A. Autonomous Mobile Platform:

Autonomous mobile robot with self-navigation without any human intervention are the major robotic research area in recent times. Such platforms provide many supports without human interventions in taking humans to guided tour, unknown environment mapping, product delivery, defense and healthcare [1]. The designed Mobile robot platform is shown in figure 1.

The intention of autonomous navigation mobile platform is to reach a destination point in an environment from a predefined starting point. The platform also needs to avoid different obstacles in the path and successfully reaches the specific end point. The obstacles to be avoided includes scatted objects, barriers and bumps during navigation in different terrains [4].

The two principal issues in mobile robot navigation are localization and path planning. Localization determines the current position and orientation of the robot with respect to its environment. The objects around the robot should be recognized by it. It needs to perceive the items that are identified as obstacles. Vision system, Sonar, laser range finder and GPS are used to find range in localization.

Manuscript published on 30 September 2019.

* Correspondence Author (s)

Ramkumar A, Department of Mechatronics Engineering, Kumaraguru College of Technology, Coimbatore, Tamilnadu, India. (Email: ramkumar.a.mce@kct.ac.in)

Sasidaran K, Department of Mechatronics Engineering, Kumaraguru College of Technology, Coimbatore, Tamilnadu, India. (Email: sasidaran.14mc@kct.ac.in)

Yogesh M, Department of Mechatronics Engineering, Kumaraguru College of Technology, Coimbatore, Tamilnadu, India. (Email: yogesh.14mc@kct.ac.in)

Sureshkumar R, Department of Mechatronics Engineering, Kumaraguru College of Technology, Coimbatore, Tamilnadu, India. (Email: sureshkumar.14mc@kct.ac.in)

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Group of cameras are also required for range detection in larger perspective of environment. In case of path planning, robot needs to determine collision free path from its initial position to the final position. [2]



Figure 1. Mobile Robot Platform Design

MOBILE ROBOTS NAVIGATION:

In the mobile robot navigation, environment can be dynamic rather than static objects in a fixed map. In such cases, sensors predict the environment based on the information collected and modify the motion of the robot to any new changes [3]. Usually sensor network helps to satisfy the above requirement. Information from the network are combined together to solve navigation problem is called as sensor fusion [6], [10]. The mobile platform with NAO is shown in figure 2. The three important processes of navigation are Map learning, Localization and path planning. Map learning is the process of recording data received from exploration. Localization is the process of estimating current position in the map. Path planning helps to find correct path to reach to the destination position [4].



Figure 2. NAO in Mobile Platform for Navigation

III. CONTROL ARCHITECTURE & RESULTS

A. Communication between nao and rover

There are many ways to communicate the Nao with the outside world. The first one is to connect the Nao with the Arduino using the USB port available. The sonar value from Nao is send to Arduino and with that it can control the motor

and stepper based on that value. But this way of connecting Nao and Arduino is difficult and it has no proper documentation or examples to do that. The next way is to install the Naoqi python SDK in raspberry pi. This Python SDK will allow user to connect with the Nao and also it enable us to control, send and receive data from Nao to outside world. This SDK is available for Linux, Windows and Mac. To use the sonar value from Nao to control the motor GPIO pins are needed. The Linux system does not have that feature but Raspberry pi [12] does. Even though we have GPIO pins in Raspberry pi the Naoqi python SDK is not available for pi because of its Arm architecture. But to compile the Naoqi library in Raspberry pi has no proper documentation so, compiling the library in an unsupported architecture will completely destroy the board and make it unusable. Architecture of NAO robot communication is shown in figure 3.

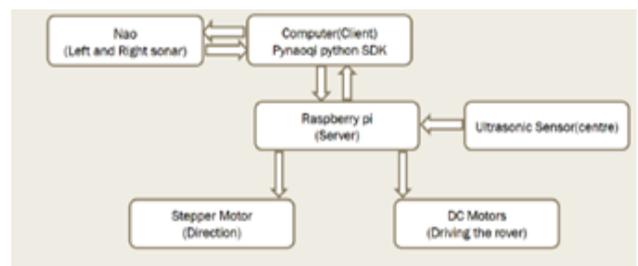


Figure 3. Nao Communication

B. Python Socket

Two nodes in a network can be commincate to each other using socket programming. One node (socket) attends on a specific port at an IP and other node reaches out to establish connection. Server acts as listener socket while client reaches out to the server. Web browsing works on the above process. In simpler terms there is a server and a client. The client process flow chart is shown in figure 5.

The socket in python [11] is very simple it uses IPv4 to connect to the node. When all devices are connected in the same network the data can be encoded and send it to the another or same device and decoded in the another or same computer to use it. Generally, the encode used is UTF-8. The data can be send or streamed over the network as bytes. The server Process flow chart is shown in figure 6.

Naoqi python SDK was running in ubuntu computer client. So, by using this SDK we can take the sonar value from Nao and send it to Raspberry pi via Socket in python. For this Nao, computer and Raspberry pi are connected in the same Wi-Fi network. The computer can be act as a client between the Nao and Raspberry pi. The pi can act as server. The data can be stored and processed before it can send to Raspberry pi. The computer can do all the calculation leaving Nao and Raspberry pi do less computation while running continuously. Because of this the Nao and Raspberry pi work more efficiently with less time delay for each data bytes to send and receive over thenetwork.

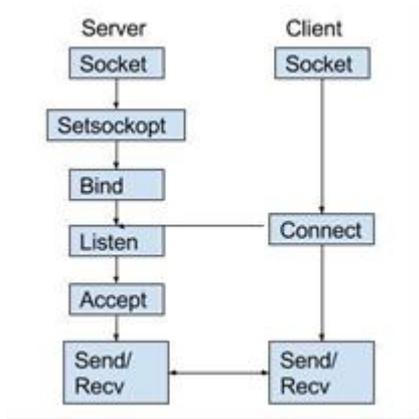


Figure 4. Client Server

C. Programming Using Python and PYNAOQI SDK

Pynaoqi is the python library file which is a software development kit used for controlling the Nao with python. Pynaoqi SDK comes with lot of inbuilt functions and methods which allow the user to access the Nao easily using python programming language. This SDK is available only for python

2.7 version. The SDK is free for download from Aldebaran community website. Installation of this library file is done by exporting the path of the files with the python 2.7 executable path. This will automatically add the library files with the python core library files. Now the functions in this SDK can be imported like normal python import.

D. Obstacle Avoidance

The processed data should be in the form command that can be send to server from client. The command should be turn left, turn right, go straight and obstacle. These commands are encoded into bytes and send to server. The turn left command works only when left sonar value is greaterthan 1 meter and right sonar value is less than 1 meter. The turn_right command works only when left sonar value is less than 1 meter and right sonar value is greater than 1 meter. The go_straight command works only when both left and right sonar values are greater than 1 meter. The obstacle command works only when both left and right sonar values are less than 1 meter. These commands are working in the basis of sonar value. So, "if and elif" condition is used to get these commands. In a time only one if condition executes and remaining are terminated. So, this means only one command issend from client to server in a time that is either one of these four commands

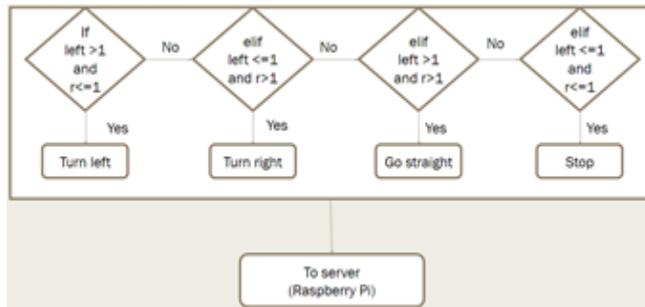


Figure 5. Client Process

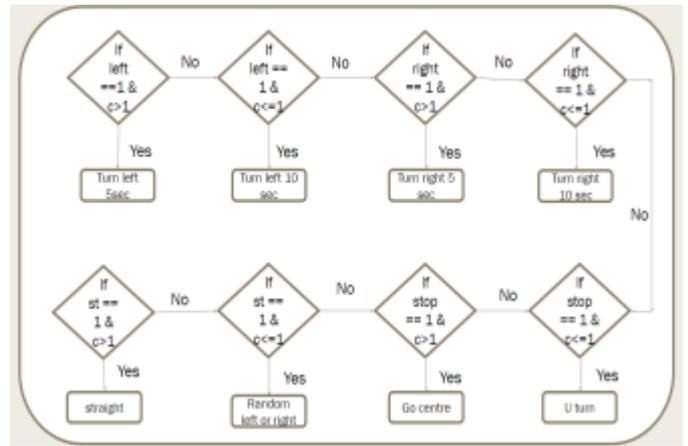


Figure 6. Server Process

E. Programmed Output

The server, Raspberry pi [9], [12] listens for a connection and the client collects the sonar value from the Nao and binds with server to transmit and receive the data. The transmitted data is then processed in the server itself and response signal is sent to client. The output from client and server communication is shown in figure 7.

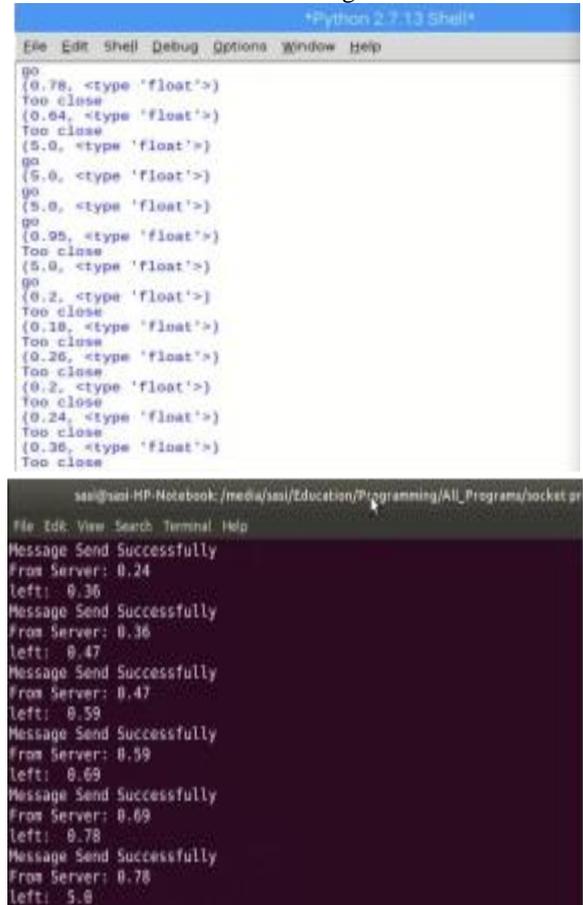


Figure 7. Client and Server Communication

The output from the server changes according to the sonar value of the Nao. The result from the server shows that the Nao and Raspberry pi communicates with each other using Python Socket and also both systems send and receive data between each other is clearly identified.

IV. CONCLUSIONS

The Nao robot, a humanoid robot has its capabilities of replicating a few tasks of a human. It got sensors and motors for motion, vision and voice. In this project, a system integration between external controller Raspberry pi with humanoid robot NAO through Python socket wireless interface to make the humanoid controlled navigation mobile platform with obstacle avoidance ability.

REFERENCES

1. Ilmi Mohd Ariffin, Ahmad Ismat Hakam Mohamed Rasidi, Hanafiah Yussof, Zulkifli Mohamed, Mohd Azfar Miskam, Adam Tan Mohd Amin, Abdul Rahman Omar, "Sensor Based Mobile Navigation Using Humanoid Robot Nao", 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS 2015).
2. Ilmi Mohd Ariffin, Azhar Baharuddin, Anderson Cyril Atien, Hanafiah Yussof, "Real-Time Obstacle Avoidance for Humanoid-Controlled Mobile Platform Navigation", 2016 IEEE International Symposium on Robotics and Intelligent Sensors, IRIS 2016, 17-20 December 2016, nTokyo, Japan.
3. Avier Minguez, Luis Montano, "Sensor-based robot motion generation in unknown, dynamic and troublesome scenarios", Instituto de Investigación en Ingeniería de Aragón, Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, Spain, Robotics and Autonomous Systems 52 (2005) 290-311.
4. Jean-Arcady Meyer, David Filliat, "Map-based navigation in mobile robots: A review of map-learning and path-planning strategies", Cognitive Systems Research 4 (2003) 283-317.
5. Gerard Gibbs, Huamin Jia, Ifran Madani, "Obstacle Detection with Ultrasonic Sensors and Signal Analysis Metrics", International conference of Air Transport - INAIR 2017
6. Reda Boukezzoula, Didier Coquin, Thanh-Long Nguyen, Stéphane Perrin, "Multi-sensor information fusion: Combination of fuzzy systems and evidence theory approaches in colour recognition for the NAO humanoid robot", Laboratoire d'Informatique, Systèmes, Traitement de l'Information et de la Connaissance LISTIC, Université Savoie, Mont Blanc, BP. 80439-Annecy-le-vieux, 74944 Annecy, Cedex, France, Robotics and Autonomous Systems 100 (2018) 302-316.
7. Amir Massah B, Arman Sharifi K, Yaser Salehinia, Farid Najafi, "An Open Loop Walking on Different Slopes for NAO Humanoid Robot", International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).
8. Noel Maalouf, Imad H. Elhadj, Elie Shammass, Daniel Asmar, "Humanoid push recovery using sensory reweighting", Robotics and Autonomous Systems 94 (2017)
9. Mallikarjun Anandhalli, Vishwanath P. Baligar, "A novel approach in real-time vehicle detection and tracking using Raspberry Pi", Department of Computer Science and Engineering, BVB College of Engineering and Technology, Hubballi 580031, India Received 25 October 2016; revised 15 May 2017; accepted 10 June 2017.
10. Aziza M. Zaki, Osama Arafa, Sanaa Amer, "Microcontroller-based mobile robot positioning and obstacle avoidance", Journal of Electrical Systems and Information Technology 1 (2014) 58-71.
11. Nagaraj Balakrishnan, Reshmi S., and R. Arunkumar. "Smart real time rescue system for fishermen." Pak J Biotechnol 15, no. 1 (2018): 73-75.
12. Premalatha J, Shanthi Vengadeshwari R, Veerendha A, "Optimization of transmission tower using Genetic Algorithm", International Journal of Civil Engineering and Technology Volume 8, Issue 9, September 2017, pp. 654-662.
13. Shivappriya S N, Dhivyaprabha R, Kalaiselvi A, Alagumeenakshi M, "Telemedicine Approach for Patient Monitoring System using IOT", Research Journal of Engineering and Technology, 8(3): July-September 2017.