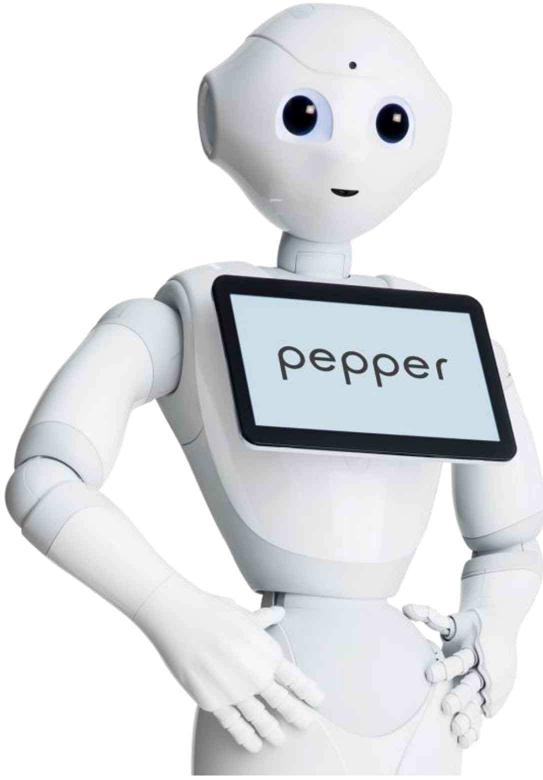
Teleoperation on Pepper robot



Name: Wei Tan
Date: 28/02/2020

Table of Contents

1. Introduction

The aim of this project is to develop a teleoperation framework for the Pepper humanoid robot (http://wiki.ros.org/pepper), which allows easy and intuitive control of the robot by having it imitating behaviors of its human operator.

The teleoperation framework will contain three modules:

1. Recorder: record speech of a human operator using a tie-clip microphone, while capturing whole body movement of her using a Microsoft Azure Kinect camera (https://docs.microsoft.com/en-gb/azure/Kinect-dk/). The recorded movement and speech will be synchronized to the same timeline;
2. Translator: translate the recorded movement into a sequence of positions of Pepper's joints, and extract lexical content and prosodic features of the recorded speech;
3. Imitator: reconstruct the speech using voice of the robot, while commencing the joint position sequence to reproduce the body movement of the human operator in accordance to the synchronized speech-movement time alignment.

2. Design framework

The framework involves the integration of software and hardware. The hardware includes external depth camera and USB microphone for operating with Pepper. The software provides the ability to fully remotely control the robot in real time, enabling voice-based guidance, including gesture-based remote control of arm motion control. Using the above modules, the Pepper robot can maintain voice-based communication with the user, can replicate the movements of the arm, and can walk and rotate in all directions. The framework map is shown in Figure 1.

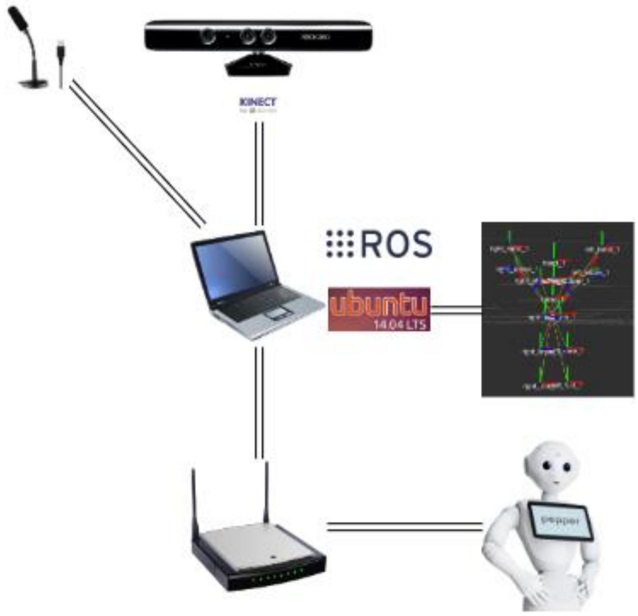


Figure 1 Framework Map

2.1 Hardware requirement

- Depth camera Kinect V1
- Ubuntu 14 Computer
- Pepper robot.

2.2 Software requirement

- Python 2.7
- NAOqi
- Choregraphe 2.5
- ROS
- OpenNi
- Openni tracker
- NITE

3. Design Approaches and Methods

We describe three basic components of our teleoperation framework: Gesture-based remote control, speech recognition and path plan; and explain how teleoperation framework was implemented.

3.1 ROS for Pepper

The robot uses a ROS network as a platform of communication between the various camera and USB microphone and the on-board computer (ROS master). We use Ubuntu 14 OS with ROS indigo installed on Dell computer.

3.2 Microsoft Kinect with Pepper

Microsoft Kinect is attached to the computer. Imitating full body human gesture requires a 3D positioning camera like Microsoft Kinect and a gesture recognition system such as Kinect SDK or deep neural network based OpenPose. The imitating controller orientation estimation of human joints into a series of robot joint angles–trajectory by using openni_tracker. These generated gesture trajectories are then used to control humanoids directly, indirectly after editing, or with a model learned from machine learning.

3.3 ROS with NAOqi

We present our efforts to integrate ROS Indigo in the Pepper robot with NAOqi 2.4.3. Pepper is controlled by an executable, NAOqi, which is started automatically when the operating system NAOqi OS is started. The NAOqi executable comes with a list of core modules and a public API. NAOqi Motion and Naoqi Audio are the one used on Pepper.

3.4 Gesture Simulation

Choreographe and ROS rviz are the simulation tools for our visualisation. We can use Choreographe to test the gesture teleoperation without a Pepper robot. However, the speech recognition can not be tested on Choreographe. ROS rviz is used for visualising the skeleton track of the human body moving throught the camera vision. There two tools can help us to adjust the movement and gesture of Pepper in order to increase the accuracy.

3.5 Speech Recognition

We combine NAOqi's speech recognition with the Google Cloud Speech platform to improve accuracy and allow for general speech input. Users connect to the service and send streaming audio, and the service returns transcription results in real-time, along with a confidence level. Dialogflow consists of a deep neural network that has been trained on a large amount and variety of speech from Google users, and is able to recognize general speech.

3.6 Path Plan

To navigate an environment optimally its map is required. Pepper achieves this using the on board kinect sensor and the gmapping ROS package. The package uses Simultaneous Localisation and Mapping Algorithm (SLAM) to generate the map. Localisation: As the size of the robot increases the drift in the wheels also increases.

4. Project progress

Figure 2 shows the progress of work completion from December to February.

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Prototype 1**					
1.1	Explore relevant paper	Wei	3/12/19	15/12/19	12	100%
1.2.	Framework design	Wei	15/12/19	22/12/19	7	100%
1.2.1	Testing on Mac OS	Wei	22/12/19	02/01/20	7	100%
1.2.2	Redesign on Ubuntu OS	Wei	02/01/20	17/01/20	7	100%
1.3	Develop the gesture motion	Wei	17/01/20	31/01/20	7	100%
1.4	Testing the right arm gesture	Wei	31/01/20	07/02/20	7	100%
1.5	Develop both the arms gesture	Wei	07/02/19	14/02/20	7	100%
1.6	Synchronising both the arms gesture	Wei	14/02/20	20/02/20	7	100%
1.7	Develop Speech Recognition	Wei	20/02/20	28/02/20	7	100%

Figure 2 Work Completion

The code is on github: https://github.com/davidtw999/pepperTeleoperation

5. Further progress

The following part can be developed in future:

5.1 Adjust the accuracy for the teleoperation (arms and should)

5.2 Implement the head and other body parts for the teleoperation

5.3 Improve the Speech Recognition synchronisation.

5.4 Design the path plan

6. Reference

https://www.groundai.com/project/estimating-emotional-intensity-from-body-poses-for-human-robot-interaction/1
https://people.eecs.berkeley.edu/~sojoudi/Tian_Semaphore_Case.pdf
https://msr-peng.github.io/portfolio/projects/skeleton_tracking/
http://www.cs.cmu.edu/~mmv/papers/17arxiv-pepper.pdf
http://www.cs.cmu.edu/~mmv/papers/18aamas-pepper.pdf