Presented at the DLSU Research Congress 2017
De La Salle University, Manila, Philippines
June 20 to 22, 2017

Investigating the Navigation Performance and Sensing Requirements of a Mobile Robot Platform to Access a Passenger Elevator

Mark Anthony A. Arevalo[1], Hiroki M. Asaba[2], and Juan Paolo J. Nablo[3]
[1]*Undergraduate*
[2]*Undergraduate*
[3]*Undergraduate*
Clement Y. Ong: clem.ong@delasalle.ph

Abstract: This research aims to have system capable of navigating itself into the elevator and to the desired floor. The system will make use of ultrasonic sensors and Infrared transmitters and receivers for localization, and a navigation algorithm. Based on the tests conducted in a real simulated elevator environment and actual elevator environment, the mobile robot platform was able to enter the elevator and exit at the correct floor with a maximum position error of (-46.3cm,-17cm) and maximum orientation error of -17 degrees. A benchmark test was conducted to quantify odometry errors of differential robots to indicate that the significant position errors are due to non-systematic errors of the TurtleBot 1 differential drive robot. The findings indicate that the position errors are mostly non-systematic, since the wheelbase and wheel diameter factors are 0.998 and 1.008, respectively.

Keywords: TurtleBot; ROS; Floor-to-floor-navigation; Elevator; Multi-floor-navigation;

I. Introduction

The use of mobile robots in different applications is no longer an idea for future generations but is currently done in today's society as stated in numerous articles (Myers, 2015; Popkin, 2014; Chesire, 2015; Allain, 2015; Baya & Wood, 2015). Through the use of mobile platforms, cost and labor would be reduced. In addition, the robot would be able to complete certain tasks more consistently compared to humans (The Technic Gear, 2014). Robots can be found in offices, schools, hospitals, and residences and they could also assist humans in doing everyday tasks. Today, there are different types of robots that are now being implemented for indoor use. However, some commercial and research implementations have certain limitations or restructuring to the environment. These implementations may require significant retrofitting to the environment to aid the mobile platform in its localization and navigation. On the other hand, some academic researches develop their mobile robot platform to operate having a large degree of automation and intelligence in exchange for less retrofitting to the environment. Instances of commercial and research implementations include Savioke's relay robot that operates a Wi-Fi modified elevator (Savioke, n.d.), McAree's Pioneer LX Platform that installed large visual aids to determine its current floor (McAree et al., 2015), and Abdulla's H20 robot that made use of a powerful sensor module for localization (Abdulla, Lui, Stoll, & Thurow, 2015). However, increasing the degree of intelligence such as adding a large array of different sensors would only result in a higher required amount of computing power and resources. Is it possible to develop a mobile robot that uses simpler, direct-sensing transducers, localization aids, and a static map, resulting in a platform that is reliable, practical and robust?

II. System Overview

The mobile robotic platform, TurtleBot, is programmed to traverse across different floors with the use of an elevator, a predefined map, and localization aids to accomplish its task. The system would be developed for De La Salle University's Br. Andrew Gonzalez Hall with the use of a manned elevator to assist in pressing necessary buttons for the robot in accessing multiple floors, an implemented response of stopping due to obstructions, and the assumption of the IR localization signal is received by the system. The Netbook, also known as the "brain" of the system, would be gathering environment data from various sensors which are embedded on a microcontroller. Combining sensor data and odometric values from the robotic platform and application of the navigation algorithm would be done through software. The robotic platform then results to performing what is defined in the source code.

A. Mobile Robotic Platform Hardware

The mobile robotic platform consists of the TurtleBot 1 with Kinect, notebook laptop with Ubuntu 14.04 and Robot Operating System (ROS), Arduino Uno R3, four SR04 ultrasonic sensors, and an Infrared (IR) receiver module (which came with IR localization aid). The dimensions of the iRobot Create, the TurtleBot 1's base, are illustrated in Fig. 1, and the setup of the Mobile Robotic Platform is illustrated in Fig. 2, and 3. It is as follows: the notebook laptop is mounted above the TurtleBot. The Arduino Uno R3 with SR04 ultrasonic sensors and an 8 pin IR receiver module, and the TurtleBot is connected directly to the laptop via USB.

For additional sensors, four SR04 ultrasonic sensors and an IR receiver module is connected to an Arduino UNO R3 board. The TRG pins of the ultrasonic sensors are connected to pin 2 of the Arduino board, while to A3 the ECHO pin of the front sensor connected to analog pin A2, right sensor, left sensor to A4, and back sensor to A5. The 8 channel infrared module is connected to pins 4 to 11 from the lowest to the highest bits.

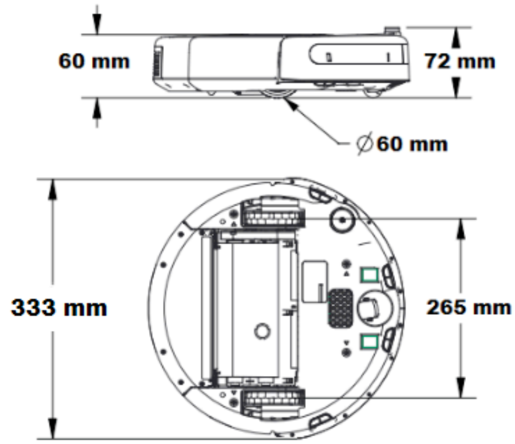

Fig. 1. TurtleBot Platform

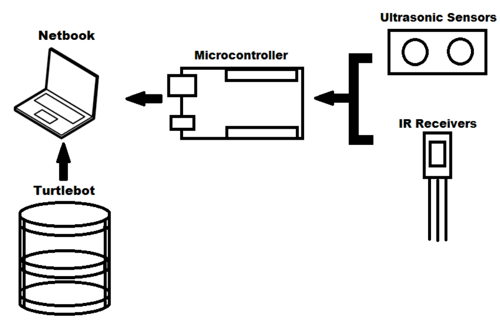

Fig. 2. TurtleBot with Additional Hardware

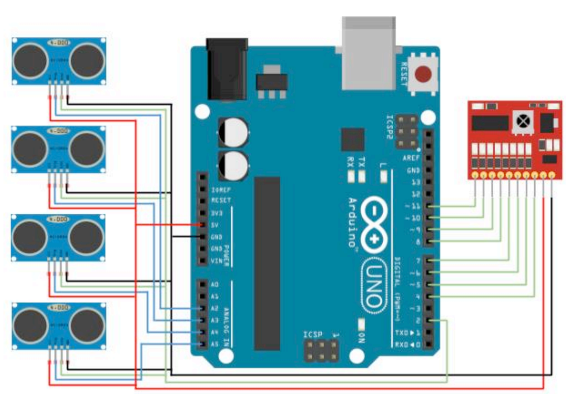

Fig. 3. Schematic for Additional Hardware

B. Environment

As shown in Fig. 4, the environment consists of the TurtleBot 85cm away from the elevator doors. Said elevator has an area of 180cm by 180cm. The elevator doors are 8cm thick. Moreover, there is a 4cm gap between the elevator floor and the hallway.

C. Navigation

The basic idea on the flow of floor-to-floor navigation is shown in Fig. 5. It starts off by having the robot position itself 85cm facing the elevator doors. The notebook resting on the TurtleBot is used to communicate to nearby people to aid it in pressing elevator buttons.

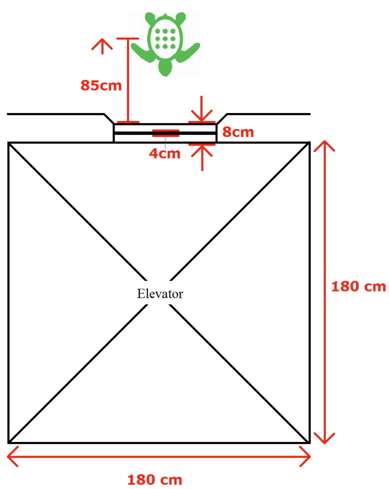

Fig. 4. Environment Layout

The first task of the TurtleBot is to rotate 180 degrees to enter the elevator backwards. This enables it to go through the gap between elevator and hallway floors. The robot determines if the elevator doors are open or closed, with the use of its back ultrasonic sensors. When the elevator doors open, a distance greater than 150cm (allowance to see if there is room for the robot) is returned by the back ultrasonic sensor to determine if the doors are actually open. Once determined, the robot will move backward to enter the elevator.

As it continues to move, it frequently checks its ultrasonic sensors to determine if there is an obstruction in its path. The robot is programmed to move 233cm which gives 40cm allowance from the back wall of the elevator (85cm + 8cm + 180cm – 40cm). Afterwards, the robot rotates 180 degrees. But if an obstruction is detected before travelling 233cm, the robot stops and waits for the obstruction

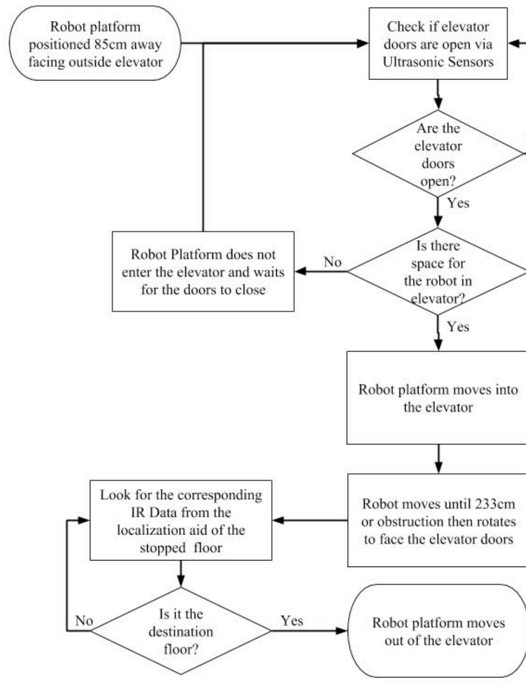

Fig. 5. Multi-Floor Navigation

to move. If after a certain time the obstruction does not move, the robot proceeds to rotate 180 degrees.

Every time the doors open, the robot will read IR signals with the use of the IR receivers that will aid it in determining the correct floor to exit. Once the correct IR signal is received, the robot leaves the elevator by moving how far it entered backwards. And, lastly, it rotates 180 degrees.

The robot will be able to determine obstructions in its path by the use of its front and back ultrasonic sensors (depending on which direction it is moving). The robot will know that there is an obstruction once the ultrasonic sensors return a distance of 40cm or below. If it does not detect an obstruction, the robot should just continue in its movement. However, if there is an obstruction, then the robot will stop and the remaining distance for movement is computed. The robot will only continue moving once it does not sense the obstruction anymore, then moves based on the remaining distance. Fig. 6 shows this process.

D. Localization Aids

For this system, a unique IR signal for each floor is pulsed at 2 Hz and is transmitted from the ceilings after the elevator doors. An IR remote is used to transmit IR signals. It is received by the 8 channel module which is connected to the Arduino.

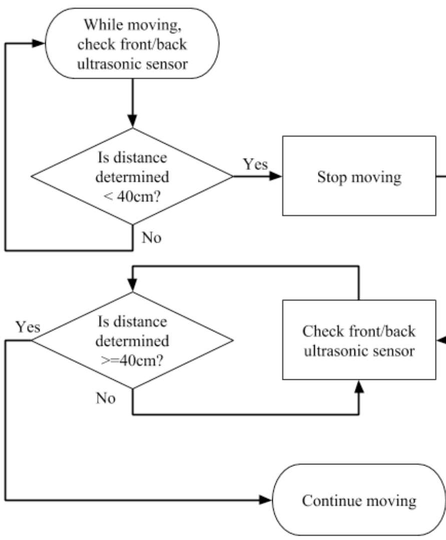

Fig. 6. Obstruction Response

E. Odometry

For the TurtleBot's odometry calibration, Bouchier mentions that it is recommended to calibrate the TurtleBot before running any navigation based applications (Borenstein & Feng, 1996). The TurtleBot has its own calibration process known as the "Create Odometry and Gyro Calibration. Further adjustments were done to the robot's linear and gyro scale correction to optimize its movement and rotation.

III. Results

The TurtleBot is attached with four markers (front, back, right, and left) to help align the robot properly. Tape was placed on the floor perpendicularly to help align the TurtleBot. For the Move and Elevator Tests, each would require two perpendicular lines of tape: the starting point, and end point. On the other hand, the Rotate Test would only need one set of perpendicular lines. The setup for Move, Rotate, and Elevator Tests is shown in Fig. 7.

To record the end position and orientation of the robot, graphing paper is utilized to determine the position of the markers. Once the position of each marker was recorded, another graphing paper was placed in the center. This is used to obtain the midpoint by drawing lines from the marker to the back marker, and from the left marker to the right marker. The end position and orientation are measured with respect to the expected end position and orientation without drift. Fig. 8 displays the actual set up of the end point.

A. Elevator Entrance Test

The TurtleBot was tested to enter one of the elevators at the Andrew Building. Forward movement to enter the elevator would cause the swivel to get stuck in the gap. However, the TurtleBot can enter and exit the elevator in reverse. The TurtleBot was tested to enter the elevator in reverse with different speeds and with and without a 2.6Kg load (weight of notebook).

The robot is able to enter and exit the elevator at speeds from 0.3m/s to 0.5m/s. At the speeds of 0.2m/s and below, the rear caster wheel falls within the gap of the elevator and the hallway.

B. Move Results:

Initially, the TurtleBot was aligned facing the end point. TurtleBot was programmed to go 1, 2 and 3 meters far. In addition, the linear scale correction of the TurtleBot was modified to get closer results. There were 5 trials for each distance with the speed based on the results of the Elevator Entrance Test. The final position and orientation of the TurtleBot was recorded to determine its drift from the ideal final position.

As distance increases, with a constant speed, starting position and orientation, the robot platform's drift becomes larger. Shown in Fig. 9 are the points where the robot stopped its forward movement after a specific distance while Table I shows the actual movement drift error. The origin represents the ideal ending position of the robot. Kinematic imperfections of the mobile robot cause these errors. These properties include wheel diameter, wheelbase, and other systematic and non-systematic errors. Due to this, the robot's drift to the right becomes more noticeable as distance increases while moving in a straight line.

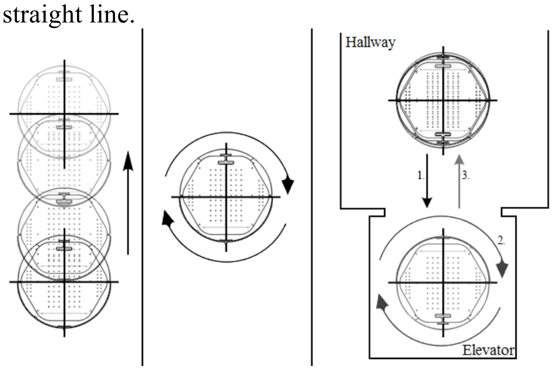

Fig. 7. Diagram for the Starting and End Position for Move, Rotate, and Elevator Tests

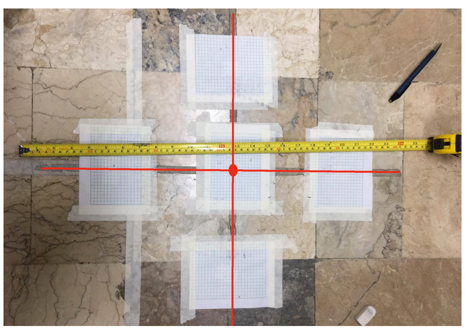

Fig. 8. Image of the End Point

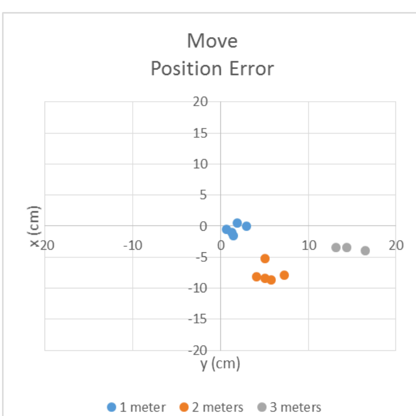

Fig. 9. Movement Position Drift/Error

C. *Rotate Results*

Similar to the Move test, this test involved aligning the TurtleBot at a starting position and orientation and recording the starting position and orientation. The TurtleBot was programmed to rotate 90 and 180 degrees in clockwise and counter-clockwise directions at a speed of 0.3m/s.

Fig. 10 illustrates the robot's drift from its starting position when rotating at 90 and 180 degrees in clockwise and counter clockwise directions. This is supported by data at Table II. When programmed to rotate 90 degrees clockwise, the robot rotates at an average of 87.2 degrees; and when programmed to do 180 degrees clockwise, the robot rotates at an average of 177.6 degrees. The difference of the average robot's actual rotation compared to its ideal rotation is higher when the robot is rotating counterclockwise. The robot rotates at an average of 81.35 degrees when programmed to rotate 90 degrees counterclockwise, and rotates at an average of 173.8 degrees when programmed to rotate 180 degrees counterclockwise. Both clockwise and counterclockwise runs had under-rotated, with the latter under-rotated more, while also slightly drifting from its initial position. Moreover during these tests, the robot tends to randomly over rotate or under-rotate by a huge amount which may be caused by systematic errors such as IMU or motor error. These outliers were not taken in consideration during recording tests.

D. *Elevator Test Results*

This test involves conducting and programming the navigation algorithm of the system. The TurtleBot would have a fixed starting position and orientation facing a simulated elevator environment as shown in Fig. 11. The resulting end position and orientation would be recorded.

Position and orientation information was recorded after the robot platform enters and exits the elevator. Moreover, ultrasonic signals were used in determining if the elevator doors are open and IR signals for determination of the correct floor.

The tests done in the simulated environment have the results placed in Tables III and IV which show position and orientation errors from the expected positions inside and outside the elevator. Based on the results, the system has a maximum position and orientation error of (-40.9cm,-6.7cm) and -17 degrees, respectively, before it exits the elevator. Moreover, errors for the final position and orientation outside the elevator has the maximum of (-35.8cm,-7.5cm) and -18.5 degrees.

While Table V shows the actual test results conducted using the elevators of Andrew Building. The allowable final position and orientation error has the maximum of (-46.3cm,-17cm) and -17 degrees, respectively.

During the real environment tests, the TurtleBot had two runs that were close to the expected end position (test 1 and 3). The other tests on the other hand had under-rotated during its 180 degree turns. This resulted to a drift from the expected end position. Moreover, by exceeding the maximum position and orientation errors, it is a certainty that the robot would hit the wall, preventing its exit. Lessening this error would increase its chances of exiting the elevator.

TABLE I. MOVE TEST RESULTS, 0.3M/S

Distance	X (cm)	Y (cm)
1 meter	1.5	-1.5
	1.3	-1
	1.9	0.5
	3	0
	0.7	-0.525
2 meters	4.1	-8.15
	5.775	-8.7
	7.25	-7.95
	5.05	-5.25
	5.1	-8.4
3 meters	13.1	-3.2
	16.5	-3.9
	14.8	-3.4

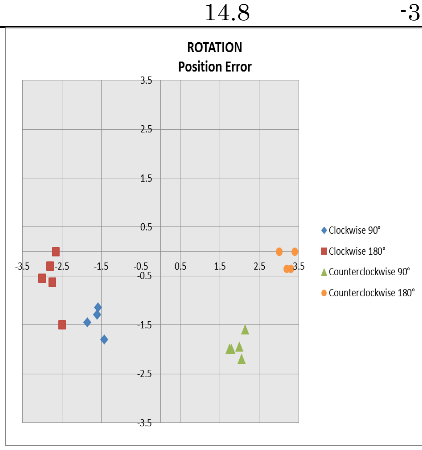

Fig. 10. Rotation Test, Position Error

TABLE II. ROTATE TEST RESULTS, 0.3M/S

Direction and angle	X(cm)	Y(cm)	θ
Clockwise 90°	-1.6	-1.3	88°
	-1.58	-1.15	86.5°
	-1.42	-1.8	86.5°
	-1.6	-1.3	88°
	-1.85	-1.45	87°
Clockwise 180°	-2.65	0	173.5°
	-2.5	-1.5	180°
	-2.75	-0.625	178°
	-2.8	-0.3	178.5°
	-3	-0.55	178°
Counterclockwise 90°	1.75	-2	82°
	2.15	-1.6	80.5°
	1.8	-2	83°
	2.05	-2.2	81°
	2	-1.95	80.25°
Counterclockwise 180°	3.3	-0.35	173°
	3	0	174.5°
	3.4	0	173°
	3	0	174.5°
	3.2	-0.35	174°

TABLE III. POSITION AND ORIENTATION ERROR FROM EXPECTED POSITION INSIDE ELEVATOR (SIMULATED ENVIRONMENT)

test #	X(cm)	Y(cm)	θ
1	-29.25	-4	-15°
2	-40.9	-6.7	-17°
3	-25.6	-5.7	-16°
4	-33.25	-5.45	-15°
5	-32.5	-4.2	-16°

TABLE IV. POSITION AND ORIENTATION ERROR FROM EXPECTED POSITION OUTSIDE ELEVATOR (SIMULATED ENVIRONMENT)

test #	X(cm)	Y(cm)	Θ
1	-31	-5.3	-14°
2	-31.65	-6.4	-18°
3	-35.8	-7.5	-18.5°
4	-31.45	-5.1	-16°
5	-31.5	-3.4	-15°

TABLE V. POSITION AND ORIENTATION ERROR FROM EXPECTED POSITION OUTSIDE ELEVATOR (ANDREW BUILDING)

test #	X(cm)	Y(cm)	Θ
1	-2.7	-0.5	-2°
2	-45.1	-11	-15°
3	-2.05	-1.7	-2.5°
4	-11.7	-2.5	-17°
5	-46.3	-17	-11°

Fig. 11. Simulated Environment

E. University of Michigan Benchmark (UMBmark) Test Results

Systematic and non-systematic errors are the cause of the mobile platform's position and orientation drift. The two most notorious systematic sources are unequal wheel diameters and the uncertainty about the effective wheelbase. These will be denoted by E_d and E_b, respectively. While non-systematic errors are due to wheel slippage, uneven floor, obstacles, etc.

Based on the document written by Borenstein and Feng (1996) and the additional information provided by The Technicgear (n.d.), the UMBmark method measures odometry errors in mobile robots and focuses on differential-drive vehicles. The UMBmark procedure introduces a method of measuring systematic and non-systematic errors. It is not possible to discern these errors based solely on the previous data recorded.

The results of the TurtleBot's UMBmark test are shown in Fig. 12. The clockwise end points are clustered more closely together compared to the counterclockwise end points.

From the recorded results, the odometric accuracy for system errors, $E_{max\ syst}$, which is the largest possible odometry error, calculates to 132.0478 cm. The values of E_b and E_d are equal to 0.997893 and 1.007625. Given E_b and E_d are practically 1.0, systematic errors are proven to be small. The significant position errors are thus non-systematic in nature.

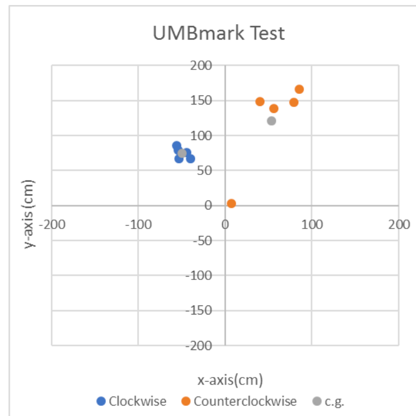

Fig. 12. UMBmark Test Results

IV. CONCLUSION

Due to the problem of the rotation, where it randomly rotates to some value, further improvement to the system's rotate should be done. Once precise, additional calibration to the robot would also increase the accuracy and would help improve the robot's movement. As seen in the data recorded there is drift while the mobile robot does its movement and rotation. Though the TurtleBot was able to perform its task during simulated and real environment in most test runs, the system would need additional sensors, such as a compass or ultrasonic sensors, and modification in software for its implementation to compensate for non-systematic errors.

BIBLIOGRAPHY

Abdulla, A. A., Lui, H., Stoll, N., & Thurow, K. (2015). Multi-floor Navigation Method for Mobile Robot. *Instrumentation and Measurement Technology Conference (I2MTC).* Pisa.

Allain, R. (2015, January 7). *The Robotification of Society is Coming.* (Wired) Retrieved February 22, 2017, from https://www.wired.com/2015/01/robotification-society-coming/

Baya, V., & Wood, L. (2015). *Service robots: The next big productivity platform.* (PWC) Retrieved February 22, 2017, from http://www.pwc.com/us/en/technology-forecast/2015/robotics/features/service-robots-big-productivity-platform.html

Bouchier, P. (2015, April 6). *Create Odometry and Gyro Calibration.* (ROS) Retrieved February 23, 2017, from http://wiki.ros.org/turtlebot_calibration/Tutorials/Calibrate%20Odometry%20and%20Gyro

Chesire, T. (2015, August 31). *Robot Revolution: The Future Is Happening Now.* (Sky News) Retrieved February 26, 2017, from http://news.sky.com/story/robot-revolution-the-future-is-happening-now-10128792

Ginii, G., & March, A. (2002). Indoor Robot Navigation with Single Camera Vision. *DEI.* Milano.

Klingbeil, E., Carpenter , B., Russakovsky, O., & Ng, A. Y. (2010). Autonomous operation of novel elevators for robot navigation. *IEEE International Conference on Robotics and Automation.* California.

Myers, R. (2015, October 3). *How humans and robots coexist in our office.* (VentureBeat) Retrieved February 26, 2017, from http://venturebeat.com/2015/10/03/how-humans-and-robots-coexist-in-our-office/

Popkin, H. A. (2014, July 6). *Robots in the Office May Not Be Far Off. But Will They Be Safe?* (NBC News) Retrieved February 24, 2017, from http://www.nbcnews.com/tech/innovation/robots-office-may-not-be-far-will-they-be-safe-n146611