Flinders
UNIVERSITY

Engineering thesis projects - 2019

**Project Title:** Intelligent system of a wheelchair with a Kinect camera and Arduino board.

A thesis Submitted to the College of Science and Engineering in partial fulfilment of the requirements for the degree of Master of Engineering (Electronics) at Flinders University - Adelaide Australia

**Supervisor:** Dr Nasser Asgari

**October 2019**

**Student name:** MAZEN AHMED SHUWAYALN

**ID NO.**2099478

**FAN NO.** SHUW0003

## Declaration:

I certify that this work does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

**Signature**: 

**Student:** MAZEN AHMED SHUWAYALN

**Date:** 20/10/2019

## Acknowledgments:

I would like to acknowledge my thanks and appreciation to my supervisor Dr. Nasser Asgari for his unwavering support, guidance and insight not only during this research project, but throughout studying my master's degree.

I would like to thank Mr Jonathan Wheere for his guidance and support in the field of robotics during studying Robotics Operation Systems, which has profound impact on continuation of research in this project.

My thanks and appreciations also to my family and friends for their continued support.

## Abstract:

This final report is about the smart wheelchair system, designed for the achievement of the three main objectives of the project. The three main goals for this project were to ensure that the newly designed smart wheelchair system with the Arduino board and Kinect camera is capable of identifying the door tags, identifying objects in its path, and track the purposes of these identified objects. A thorough literature review was conducted in this project to discover relevant data and information about the three main objectives of this smart wheelchair system project. The wheelchair system was designed with the integration of Microsoft Kinect camera and the Arduino board that not only ensured identification and finding of objects and obstacles in the path of the wheelchair but also helped in tracking objects and identifying doors from their door tags.

## Table of Contents:


Declaration:....................1
Acknowledgments: ....................2
Abstract:....................3
List of Figures:....................5
Chapter 1: Introduction:....................6
- Background: ....................7
- Project Aim:....................9

Chapter 2: Theory and Methodology: ....................10
- Wheelchair system with voice recognition:....................10
- Kinect based wheelchair control system: ....................12
- Features of Detecting Objects:....................12
- Arduino Board wheelchair system:....................13
- Autonomous Navigation of the Intelligent wheelchair: ....................20
- Homography Matrix:....................21

Chapter 3: Experimental & Calculation Results:....................23
1- Finding the Object:....................23
2- Tracking the Object:....................25
3- Finding the Doors:....................27

Chapter 4: Discussion:....................33
Chapter 5: Limitations:....................39
Chapter 6: Conclusion: ....................40
References:....................41
Appendix A:....................44
Appendix B: ....................49

## List of Figures:

**Figure 1 Kinect Camera attached wheelchair with Arduino Board.......................................... 8**
**Figure 2 The diagram of NavChair (Simpson & Levine, 1997). .............................................. 10**
**Figure 3 The conception of Kinect-based powered wheelchair control system (Theodoridis, et al., 2013) ........................................................................................................................ 12**
**Figure 4 block diagram of wheelchair with kinect camera (Yatnalli, et al., 2012). ................. 12**
**Figure 5 The Block Diagram of the Project (Lodhi, et al., 2016). ........................................... 13**
**Figure 6 wheelchair diagram (Wei, et al., 2012)..................................................................... 20**
**Figure 7 Maps of the experimental environment (Cavanini, et al., 2014).............................. 20**
**Figure 8 Homography Matrix ................................................................................................. 21**
**Figure 9 dx= h31 and dy= h32 values regarding to the object position in camera frame ........ 21**
**Figure 10 Connection Arduino board with computer and joystick ........................................ 34**
**Figure 11 Interface board and Arduino connection............................................................... 35**
**Figure 12 Image recognition Kinect camera.......................................................................... 35**
**Figure 13 Object tracking process smart wheelchair system ................................................ 36**
**Figure 14 Object tracking based on the position of the objects............................................. 37**
**Figure 15 Wheelchair identifying rooms by door tag ............................................................ 38**

## Chapter 1: Introduction:

In recent years, research and usage of mobility aids have increased. For example, the electric scooters of electric wheelchairs that are specially made for the people suffering from specific disabilities. However, the majority of the people due to some other limitations are not able to control the maneuvering of the electric wheelchair or scooter on their own(Purwanto, et al., 2009). This illustrates the requirement of a smart wheelchair system that is capable of detecting and avoiding obstacles in its path automatically to avoid collisions and keep the disabled patient riding the wheelchair safe from any accidents. People that are suffering from the sensory, motor, or cognitive impairment, whether it is because of a disease or impairment, rely on electric or power wheelchairs for fulfilling their mobility requirements. As some people that are suffering from any kind of disability are incapable of utilizing a traditional joystick for navigating their electric-powered wheelchairs, they then mostly use the alternate control systems. These alternate systems include the chin joysticks, head joysticks, spin-n-puff, along with the thought control system technology. In a lot of cases, the electric-powered wheelchair users have a lot of difficulty carrying out regular mobility tasks and might benefit a lot from a smart or navigation automated system wheelchair(Nguyen & Jo, 2012). Aside from just mobility, the disabled people are also heavily reliant on the people that are providing care to them for tasks like drinking and eating, communicating with others and handling items, particularly in large gatherings and groups. For accommodating the population of people that found it difficult to or even operate the electric-powered wheelchair several investigators and developers have considered technologies that were initially developed to serve mobile robots for the development of the smart wheelchair system. The intelligent wheelchair system typically contains either the regular power-based wheelchair technology on which the collection of sensors and a computer are integrated or the mobile robot based technology on which a wheelchair seat is attached. The transition of the wheelchair system, which cooperates with the user, is considered at least as significant as it forms the regular to powered system wheelchairs arguably even more significant as this might be the paradigmatic shift as compared to just the technological shift(Shibata & Murakami, 2012). With the consideration of the previous research work and development in this field, it is clear that a considerable effort and stride has been made in the development and research of both powers based wheelchair systems and smart wheelchair systems to improve upon the assistive technology in general for helping the people suffering from disabilities.

A lot of success has been reported regarding the application of research and development in innovative autonomous smart wheelchair systems. But in the last decade, little effort has been made to intensively and extensively review the possibility of an intelligent wheelchair system and provide statistical significance of a comprehensive, high-quality overview regarding the investigation trends in smart wheelchair system(Kondori, et al., 2014). For these reasons, this report is aimed at providing the significance and importance of an intelligent wheelchair system, which is capable of detecting any obstacles in its path and navigating without making any collisions. Literature was conducted for looking at the possibilities and applications of a smart wheelchair system to develop a wheelchair system that is capable of detecting doors and other obstacles in its path through a Kinect camera with the image recognition technology.

- Background:

This proposed wheelchair system would also be capable of tracking objects in its way, thanks to the sensors installed and the Arduino board. The proposed smart wheelchair system is a power-based wheelchair system with the integration of the sensors, Kinect camera, Arduino board, computer, and the controller to control the movement of the chair. The objective of the project is to create a wheelchair that is capable of navigating as an electric wheelchair, but it is capable of detecting and avoiding obstacles in its path thanks to its image and objects recognition technology and the automatic capability of avoiding doors and other obstacles in its way(Carrino, et al., 2012). This type of system reduces the physical burden on the disabled individual because it does not require any sort of high physical activity for either calling the wheelchair from its parking location or moving the wheelchair when the disabled individual is sitting in the wheelchair. The integration of the Microsoft Kinect technology in the power wheelchair system makes it possible to develop a system that is capable of recognizing the hand gestures and movements of the disabled person but also develop a suitable route for the mobility of the wheelchair. The future work that is proposed by Yatnalli et al. (2012) in the application of Microsoft Kinect in power wheelchair system is to focus on developing a wheelchair system that is capable of moving towards the disabled person or to a specified location without following the line.


*Figure 1 Kinect Camera attached wheelchair with Arduino Board*

The project is regarding the construction of the intelligent system of wheelchair integrated with the Arduino board alongside the Microsoft Kinect camera. Considering the limitations that were faced in the previous project of the smart system of wheelchair, which was that the system was unable to identify or detect objects or doors, track objects and identify doors from their door tags. These new project goals are to overcome these identified limitations(Milella & Cicirelli, 2010). That is why the main three targets of the new project are mitigation of the limitations of the previous project. The first objective of this project is to ensure the wheelchair system is capable in the identification of objects and doors in its path through image recognition technology via the integrated Kinect camera. The second objective is to ensure the new system is also capable of object tracking. The third objective is of ensuring that this system is capable of locating rooms through the identification of their room tag number (Chang, et al., 2013). Development of the project is done to assist elderly people and individuals with disabilities in effective and independent mobility. The wheelchair system being developed is integrated with the Arduino board and Microsoft Kinect camera, making it capable of identifying objects or doors and traveling from one location to another by avoiding collision with different objects in the path (Simpson & Levine, 1997). This final report provides valuable data and information about how these three objectives of the project were achieved(Takahashi & Matsuo, 2011). The robotic or electric wheelchairs, with the introduction of the locomotion controls have enhanced the standard wheelchair technology significantly. The electronic or robotic wheelchairs have eased the life of disabled people, particularly those who suffer from severe impairments by improving their range of mobility. The application of the robotics technology and speech recognition technology in the electronic wheelchair systems is proposed as a revolutionary system that will provide a lot of benefits for those disabled people who cannot use their legs or hands (Wang & Chaing, 2012). The application of voice recognition in an electronic or robotic wheelchair system will provide disabled people that have difficulties in using their hands for controlling wheelchair movement with the system of voice recognition so that they can control the movement of the wheelchair through their voice rather than using their hands for controlling the flow of wheelchair system. The authors Khare et al. (2017) propose a wheelchair system, which can not only be controlled by a joystick but also through voice commands with the application of the speech recognition technology. The wheelchair system will be managed through voice by speaking the name of the direction like for example left, right, forward, backward, etc. and to stop the wheelchair movement the user has only to say stop. In the hardware development of this wheelchair system, the researcher used the HM2007 voice recognition module that is capable of correlating commands for performing speech processing and giving the results to the Arduino that is programmed further with the respective locomotion commands (Khare, et al., 2017). With the increase of the aging population, it is becoming more and more difficult, particularly for elderly disabled individuals to move around in their wheelchair with the use of their hands. It is vital to develop a wheelchair system that not only provides good mobility potential to disabled people but also considers the improvement of the living quality by ensuring that the wheelchair system has the quality of giving functional mobility without the use of hands all the time(Tian, et al., 2009). That is why it is essential to consider a wheelchair system that provides versatility with minimum physical effort and movement to help in the improvement of quality of life for disabled and elderly individuals.

- Project Aim:

Considering previous project limitations of a wheelchair system not being able to find doors and obstacles, this thesis aims to provide valuable data and information regarding how the wheelchair will be capable of detecting obstacles like doors in its path.

1. Using the image recognition technology in the wheelchair system to make it capable of identifying doors and other obstacles with the utilization of the Kinect camera.
2. Furthermore, the wheelchair will be capable of tracking objects thanks to the sensors installed and the Arduino board.
3. With its image recognition technology, the wheelchair system will be able to identify the room number tags or sample.

# Chapter 2: Theory and Methodology:

- Wheelchair system with voice recognition:

NavChair that is recognized as a wheelchair assistive navigation system was constructed with the purpose of reducing cognition alongside physical requirements of developing powered wheelchair systems(Simpson & Levine, 1997). The wheelchair was considered like the adaptive control shared system and control was division between wheelchair itself, and the operator. This system is adaptive as power was divided between wheelchair system and wheelchair operator, varying in accordance with the latest requirements of this project(Simpson & Levine, 1997). These systems have the ability of control allocation automatically between wheelchair and operator, which illustrates outcomes through performance evaluation of NavChair automatic system of adaption through experiment with able-bodied individuals who utilized the speech recognition system of wheelchair system to steer and navigate through several transitions between different operating modes of the wheelchair (Simpson & Levine, 1997). Mechanism of automatic adaption of NavChair system presented a suitable solution to achieve NavChair project requirements(Tamura, et al., 2010). This adaptive system mainly assures the design and planning criteria of the project that was specified through making accurate decisions regarding adaptation, maintenance of rider comfort, and minimize the utilization of computational resources(Nishimori, et al., 2007).

Image removed due to copyright restriction.

*Figure 2 The diagram of NavChair (Simpson & Levine, 1997).*

Furthermore, a lot of tests were performed with this system, which gave similar results for sufficient NavChair mobility, especially when NavChair got operated by the user through joystick controls. The particular adaptive approach was also applied to other different assistive technologies (Simpson & Levine, 1997). The newly concluded tests showed Bayesian network capabilities of making decisions adaptive for one-switch column/row accessibility computer system. According to Khare, et al., (2017) everywhere in the world the application of voice recognition can be found which is making the lives of people more effective and easier. Some examples of voice recognition applications being used globally are Amazon’s Alexa, Google Home, and Apple’s Siri, among others. Furthermore, a lot of applications available with latest smart phones allow the feature of voice recognition, where the user can call someone by just speaking their name or can type and send a text message by speaking(Khare, et al., 2017). People who want to save their time whether it is using speech recognition software like Siri on their phone for information or using a smart home device to control the electronic devices in their home through speech recognition consider the speech recognition technology in the 21st century as a necessity (Seki, et al., 2009). Instead of operating a system through buttons or typing a keyboard, speech recognition technology provides a much more convenient way of controlling a system. The application of the speech recognition technology in electronic or robotic wheelchairs is considered a necessity to assist disabled individuals in easily maneuvering from one location to another. According to Khare et al. (2017) there are approximately two critical components in the development of the voice-controlled or speech recognition wheelchair system; one is the hardware design of the wheelchair system, and the other is the control. In both areas of the system, a substantial amount of work is required to be done for developing a system that is smart and effective in its operation. The performance of the robot can be increased with the application of excellent hardware design and often has the capability of making each of the other fundamental issues much more comfortable to address(Silva, et al., 2013). In the electric wheelchair system, the motored wheels are used for traversing through everyday terrain. The motors of these wheels are supported by a high power supply to ensure fast and responsible mobility of the wheelchair. These motors are connected with the relays and the LC293D IC is connected with the Arduino board. The L293D is considered as the general motor driver or the IC of motor driver that ensures the DC motor to move in directions. This system mainly operates in accordance with the H-bridge system concept, which is the circuit that allows the flow of voltage in either direction. In the method proposed by Khare, et al., (2017) the voice recognition module is connected with the RX pin of the Arduino board for receiving the coded data stream from the module specifying the corresponding command for the direction. The control of the locomotion is ensured with the utilization of L293D and relays that control the motors. For the control problem of the voice recognition wheelchair system, there are three main components. The first is the recording of commands, the second is the recognition of authorities, and the third is the locomotion control(Stenberg, et al., 2016). First, it is essential to record the commands or voice instructions in the system before making it available for use. Each voice instruction in this system has been allotted a maximum length of approximately 1300ms that ensures the majority of the commands can be recorded in the order. Once the recording is started, the recording process cannot be stopped until all five-voice instructions are finished recording for one group.

Furthermore, once the user has started recording the voice commands, the previous content of voice commands in the system is erased. Moreover, in the recording stage of this system, the module does not reply to any other serial commands. The recording of the powers in this system is done through serial communication with the PC with the utilization of the access port with the baud rate of 9600 and data bit of eight. The voice recognition wheelchair system developed by Khare, et al., (2017) is capable of recognizing five, voice instructions of a particular group at the same time. Meaning the system can have approximately 15 voice commands in three groups.

- ## Kinect based wheelchair control system:

Wheelchairs control system that is Kinect based is made for helping and assisting disabled people in improving their independent mobility and living quality. The control system Kinect based wheelchair mainly uses low-cost, infrared coordination from the CMOS sensors to conduce the precise indoor positioning of a wheelchair along with controlling the wheelchair moving system. Realization of this system is mainly done to evaluate its operations(Theodoridis, et al., 2013). In accordance with performance outcomes gained by researcher Chang et al., (2013) in their experiments, a wheelchair system is particularly displacing an offset at the path plane, which at 20cmis limited, and towards a disabled individual, at 10cmit is limited. In accordance with designed architecture outcomes of wheelchair system developed by Chang et al. (2013), the design is classified among three parts; i.e. gesture recognition capabilities, wheelchair controls and indoor positioning(Wei, et al., 2012).

Image removed due to copyright restriction.

*Figure 3 The conception of Kinect-based powered wheelchair control system (Theodoridis, et al., 2013)*

- ## Features of Detecting Objects:

In a lot of situations, teenagers and children have accessibility of fewer leisure activities as compared to the able-bodied individuals of the same age. The author Yatnalli, et al. (2012) developed a wheelchair system that is Kinect based with the integration of embedded system techniques, gesture recognition, ZIGBEE communication, along with the indoor position using image processing and image recognition IR techniques. The proposed system recognizes the hand gestures and movements of the disabled person for calling the wheelchair and moving the wheelchair when he requires (Yatnalli, et al., 2012). The functionality of this system works following the hand movements and gestures,

Image removed due to copyright restriction.

*Figure 4 block diagram of wheelchair with kinect camera (Yatnalli, et al., 2012).*

and once the server is integrated with the interface of Kinect camera it recognizes the gesture i.e., wheelchair call gesture, it mainly does inquiry of wheelchair through the interface of ZIGBEE to move towards a disabled person.

Furthermore, the server is capable of simultaneously planning the route of the wheelchair from its current position to the disabled person, with the help of LED infrared rays, which help direct the wheelchair to the disabled person smoothly. Moreover, when the wheelchair system is in motion, it follows the line of the infrared lights towards the disabled individual(Taher, et al., 2014). When the wheelchair reaches the disabled person, he can use the touch panel of the wheelchair system to determine the location where he wants the wheelchair to traverse when he sits on the wheelchair. Also, with the help of the hand gestures and the panel, the disabled individual can also direct the wheelchair towards its parking location.

- Arduino Board wheelchair system:

Image removed due to copyright restriction.

*Figure 5 The Block Diagram of the Project (Lodhi, et al., 2016).*

Arduino board control wheelchair system with the Bluetooth module consists of two components mainly, the hardware component of the wheelchair and software component(Lodhi, et al., 2016). The architecture of hardware component of an intelligent wheelchair system is based upon the Arduino board control along with the integration of Bluetooth module consisting of an embedded system, that is based upon an Uno Arduino board alongside a motor driver, Android phone integrated with a Bluetooth module. This system Bluetooth module gives a communication medium between the user and the wheelchair, that helps in controlling the wheelchair through the app on Android phone by giving voice commands (Lodhi, et al., 2016). Specific commands are spoken by the user and the Arduino voice control BT and voice application AMR that is integrated into the smart phone analyzes the command and sends the digital information after analysis to the motor driver of the wheelchair(Lodhi, et al., 2016). Bluetooth module gets the command from the user and then it converts this command into a language that is understandable for system's microcontroller for executing this specific user command accurately(Assante & Fornaro, 2015). Like for example, the disabled person gives the command of wheelchair parking, the Arduino Uno board receives the command and it converts or translates this information to the digital code of wheelchair parking and delivers it to the motor driver of the wheelchair, which is responsible for the wheelchair movements(Cavanini, et al., 2014). If the disabled person gives the command of "Go" then the system of this wheelchair is programmed of understanding that command as an indication to move the wheelchair in forward direction. Similarly when a disabled person says the command of "Back" this wheelchair system is programmed of understanding that command as the request of moving in backward direction, and in the same manner the system understands and executes the "Right" command as a request of turning the wheelchair in right direction and the "Left" command as a request of turning the wheelchair in left direction or rotate it in left direction. Finally, the command "Stop" is considered by the system to either park or stop the wheelchair movement(Lodhi, et al., 2016).

The purpose of designing the system was mainly to ensure timesaving, cost saving, and energy saving for a disabled individual using the wheelchair. Ultrasonic sensors of the system are designed to help the detection of obstacles or objects that arrive in the way or path of this wheelchair. In doing so, it makes the system capable of not only detecting those obstacles but also avoiding them while moving. According to Mirin et al. (2018), because of the increased percentage of physically disabled and elderly people, wheelchairs are considered as the best assistive devices to help these people in enhancing their physical mobility. However, the traditional wheelchairs had some limitations like bulkiness, flexibility, and limited functionality(Hashimoto, et al., 2009). There are now latest technologies that allow mitigation of these limitations by providing the disabled people the ability to control the movement of the wheelchair through hand gestures, voice recognition and joystick controls(Mirin, et al., 2018). These wheelchairs are recognized as smart wheelchair systems that are mainly developed with the aim of helping the disabled and elderly people to move from one location to another with minimum effort and complete independence. Latest smart wheelchair systems can be controlled through an Android app that can be installed in the smart phone by the user to control the movement of the wheelchair. It basically contains two controlled modes, the first mode is known as the touch mode, and the second mode is known as the voice recognition mode. In the voice recognition mode, the user can control the movement of the wheelchair by giving voice inputs and commands through the smart phone app. The smart phone app of the wheelchair system converts the voice commands of the user into the string of data, and this string of data is sent through the Bluetooth module of the phone to the wheelchair(Achic, et al., 2016). Lastly, it is delivered to Arduino Uno. Next, the Arduino Uno board on the wheelchair decodes the message and processes it. The motor driver on the wheelchair directs the wheelchair according to the given command of the user. This implies that when the user says the "go" command the wheelchair system recognizes this as an instruction to move forward, while the word "reverse" from the user is interpreted by the wheelchair system as the command to move in the backward direction. Similarly the command "left" instructs the wheelchair to turn left, while the right command instructs it to turn "right" by the system. In contrast, using the second mode, the user can control the movement of the wheelchair by selecting the desired location and direction on the Android smart phone app. The command given by the user on the smart phone app is forwarded to the Arduino Uno board of the wheelchair through the Bluetooth module. The powers granted by the user on the smart phone app are converted by the Bluetooth in the binary format, and they are delivered to the Arduino Uno board of the wheelchair. The Arduino Uno board on the wheelchair reads and executes the commands, and lastly, it sends the digital values to the wheelchair's motor driver device(Faria, et al., 2014). This motor driver device directs the wheelchair in accordance with the user command. This means that on the smart phone app when the user selects the option of "go" the wheelchair moves forward, when the user selects the option of "reverse" the wheelchair move backward, when he chooses "left" the wheelchair turns left, and when the user selects the "right" option the wheelchair turns right.

This type of smart wheelchair system provides a lot of facilities to disabled and elderly people by allowing them the options of controlling the movement of the wheelchair through the touch input on the smart phone app or using their voice. Another idea is of a smart wheelchair system that follows the eye movements of the user. These are known as 'eye-controlled systems' that enable the movement of the wheelchair depending upon the user's movement of eyeballs. In this system, the camera gets mounted upon the wheelchair. The camera then notes the eye movement of the user and moves the wheelchair in a specific direction, decided by the user's eye movements. Based upon the location detected through eye movements, possible motion direction is discovered, and the command is delivered through the Arduino Uno board on the wheelchair to the motor control device of the wheelchair system. A micro control system is also installed, which enables the wheelchair to be controlled via motion of the head. This system comprises of both mechanical and electrical components. The accelerometer in the system is used for collecting the data of head motion. Results of digital system are connected with the mechanical actuator, as they are utilized for positioning the joystick of the wheelchair basing upon the commands of the user. The novel algorithm processes the sensors data and is run within a microcontroller(Jang & Choi, 2014). Head movements of the user are translated in the position of the wheelchair's joystick. Another example of a voice-controlled wheelchair system, which uses the voice commands of the user as the main interface of the wheelchair system, is also present. The recognition parser grammar-based, which is named "Julian," is utilized in the system. It is the open-source application that is created by the Kyoto University along with the Nara technology and science institute. The voice instructions in this voice-controlled wheelchair system consist of 9 reactive commands along with 5 verification commands. The reaction commands have approximately 5general reaction commands and the short moving 4 reaction commands. The wheelchair system is basically based upon the electronic, commercial wheelchair system developed by the Nissin NEO-PI Medical Industries. This system basically consists of a microphone headset and a laptop PC. The user has the choice of controlling the system either through the laptop buttons or through their voice using the microphone headset(Lin, et al., 2010). The control signals in this system are sent to the PIC from laptop and PIC generates the motor control signals for driving the wheelchair. The voice recognition module along with the smart wheelchair have similar control movements with the project developed by Mirin et al., (2018) but the only difference is that this system recognizes the commands of the user through the microphone. Whereas, recent research shows that the smart wheelchair systems with the voice recognition capabilities help in upgrading the capabilities through navigation maps and the present location of the user with GSM and GPRS system.

The author Mirin, et al. (2018) developed a smart wheelchair system for improving the mobility of disabled and elderly people and helping them in getting from one place to another independently. For this system, Mirin et al. (2018) developed an Android app and integrated into the smart phone for the purpose of controlling the system. The author developed the Android implementation of the order using the MIT app inventor platform. The app inventor MIT platform is the application for transforming the complex language of the coding, which is text-based into the form of visual building block. The commands are recognized as a block, which specifies the action to get performed on a phone. Some of the authorities require 1 or probably more inputs for ultimately determining their response. According to Mirin, et al., (2018) the best part of utilizing the Android platform for the app is that it is the open-source electronic/digital platform that is capable of reading inputs and converting them into the output. The Arduino Uno is much less expensive in comparison to other different micro-controllers as this is capable of running on Linux, Macintosh OSX and Microsoft operating systems. Considering the hardware design of Mirin, et al., (2018) smart wheelchair system, it consisted of six main components, two scooter DC motors, two motor drivers, a microcontroller, a Bluetooth module, software, and the power supply. Both the engines got mounted on both wheels as control box was placed just among them as the 24 volts power supply was integrated in the front of control box (Tamura, et al., 2010). The control box consists of the Arduino Uno board alongside the MD30C motor driver along with the Bluetooth module. Arduino Uno board is utilized for controlling the motor driver. But only the digital output pin of the motor driver is used. The command that is provided by the user in this system is transmitted towards the Arduino Uno through Bluetooth connection. The controls are first converted by the Bluetooth module offered by the disabled person in the binary format, and then it sends them towards Arduino Uno. Then the Arduino Uno board reads and executes the commands provided by the user, and then it sends the digital values to the motor driver device of the wheelchair system. Then the motor driver device of the system is responsible for driving the wheelchair in accordance with the specific direction command provided by the user.

The Arduino Uno board is the microcontroller that is based upon the technology ATmega328. In project of Mirin, et al., (2018) the Arduino Uno microcontroller board architecture is AVR, and its operating voltage is approximately 5 volts. Furthermore, the Bluetooth HC-05 module in the system is mainly integrated for the purpose of the transparent setup of serial wireless connection. This unit has approximately four pins, which are the GND, 5V, RX, and TX. The GND and 5V pins are basically utilized for power purposes, and the RX and TX pins are being used mainly for the purpose of the serial interface. The pin configuration in the system of 6 to 9 is utilized for controlling the motor driver. Moreover, the pins six and seven are in connection with the motor driver one, and the pins eight and nine are in connection with the motor driver two. Whereas, the pin configuration of pins ten and eleven are utilized for the serial interfacing with the RX and TX. The data is decoded and processed by the Arduino Uno board that is delivered by the Bluetooth module. Then after the processing and decoding of data, the Arduino Uno board passes or transfers the data to the motor drivers(Jang & Choi, 2014). The motor drivers are responsible for directing the wheelchair in accordance with the command provided by the user either as a voice command or through the touch panel of the Android app. The power supply, which is attached to this wheelchair system, supplies the electrical power to the Arduino Uno board, Bluetooth module, motor, and the motor driver. So basically in this system, the user is capable of controlling the movement of the wheelchair by giving commands just by tapping i.e., forward, backward, left or right on the Android app of this system. Both modes the voice recognition or the touch panel can be used by the disabled person to control the mobility of this wheelchair system. In the voice recognition mode, the user is required to turn the Bluetooth module on, in the smart phone device and then choose the wheelchair Bluetooth module to pair or connect the smart phone with the wheelchair system. When the Bluetooth of the smart phone is connected with the wheelchair, the user can start giving commands, and both the modes either voice recognition or the touch panel are connected with Bluetooth to the wheelchair so the user can use both modes at the same time to give commands to the wheelchair(Tamura, et al., 2012). Therefore, through the Bluetooth module, the system transmits the user commands to the Arduino Uno board. The user commands are first converted by the Bluetooth module to the binary format whether it is a voice command or a command from the touch panel of the smart phone, then it transmits that command to the Arduino Uno board. The Arduino board reads and executes the command, and finally, it sends the digital information towards the motor device driver of the system. Then the motor driver installed in the wheelchair system directs the wheelchair in accordance with the user command, which is four possible direction-commands either backward, forward, right or left.

For testing the effectiveness of this particular system Mirin, et al., (2018) performed a few experiments with the wheelchair for analyzing the consistency of the system in executing both the touch panel input and the voice recognition input commands to the system. In two different areas, the first experiment with the operation was carried out i.e., noisy cities and the quiet regions. Whereas, in the second experiment, ten different speakers were used in different environments. Each speaker in the analysis gave nine various commands along with the rate the system was capable of responding accurately to those commands provided the favorable results for the system. Each test was almost repeated ten times, and the consequences of each test were averaged to get a conclusive result regarding the performance of this wheelchair system. The purpose of these experiments was mainly for identification of consistency of voice recognition capabilities of the system in different areas, which were embedded in the smart phone app of the system. The percentage of flexibility for each command provided by the user was calculated in each test(Halawani, et al., 2012). The first five tests were carried out in the quiet environment, and all the four possible powers of forwarding, backward, right and left were given in a random order to the system with the addition of some wrong words basically a repetition of two wrong command words in the order of every five repeated command words. The system was able to recognize thirty-six out of forty-five commands in the voice recognition mode of the operation, which provided an approximate percentage of consistency of 80% in the quiet environment for this system. Whereas, the experiment, which was conducted in the noisy environment, had the same format of commands provided to the system as in the previous research with the quiet atmosphere. The results of this experiment illustrate that the system was capable of understanding and executing twenty-five out of forty-five commands given by the user in the voice recognition feature, which provided the approximate consistency percentage of 55.6% in the noisy environment for this system (Tian & Xu, 2009). Considering the outcomes of experiments both in the silent ambiance and the noisy environment in the voice recognition mode of the operation illustrates that the system performed better in a calm background as compared to the noisy environment as the percentage of consistency of the system was higher in understanding and executing the user commands in a silent environment as compared to a noisy environment. Which totally shows that the application of the voice recognition mode in the noisy environment is less practical as compared to the quiet atmosphere. The last experiment was with the ten random speakers through which nine different commands were given in the voice recognition mode of the system both in the calm environment and the noisy environment. The order of these tests was to provide three wrong commands in every ten speakers repeated words.

The results of this experiment of Mirin et al., (2018) system showed that value of percentage of consistency could get higher for this system if the commands are given to the system in good pronunciation and the commands are spoken at the moderate or low pace. Furthermore, the results of the experiments also provided that there was not any inconsistency or inefficiency in the performance of the system, whether the speaker was either female or male. Moreover, the results of each experiment in the noisy environment illustrated that the values of percentage of consistency of this system could get higher if good pronunciation is used by the user and if the experiment gets conducted in the quiet environment to avoid any disturbance and distortion in the deliverance of commands, and also it was noted that those commands that were spoken in a moderate or slower pace were understood at a higher consistency by the system as compared to those commands which were spoken at a faster pace. The conclusion that was drawn by Mirin, et al., (2018) from these tests and experiments with the system was that although the system showed some efficiency and accuracy in understanding the user commands in a noisy environment it is clear that the performance of this system's voice recognition mode is significantly higher when it is used in a silent environment. The developer of the wheelchair system Mirin, et al., (2018) also performed some tests with the touch mode of the system. After the proper implementation of the wheelchair system and its connection with the smart phone app through the Bluetooth module, the functionality of the touch model was tested. According to the findings through the experiments with the touch mode of the wheelchair system, it was clear that the maneuvering of the wheelchair using the touch panel of a smart phone app in touch mode showed excellent efficiency, accuracy, and functionality in all direction commands. Therefore, the recommended mode by Mirin, et al., (2018) for this system is the touch mode of the system because it provided more accurate and efficient results as compared to the voice recognition mode because the silent or noisy environments do not affect this mode. For analyzing the system efficiency, these experiments were mainly conducted with the unload condition along with four different sizes of individuals that traveled similar distances using the wheelchair(Maruno, et al., 2011). According to the results, the time taken by wheelchair system to reach the user's desired direction depending on the weight of the person sitting on the wheelchair as the weight on the wheelchair increased the time consumed by the wheelchair in moving from one destination to another also increased. The findings of Mirin et al., (2018) showed that a load of approximately more than 50 kg on the wheelchair illustrated higher consumption of time in reaching a destination as compared to loads under 50 kg. This wheelchair system proposed by Mirin, et al., (2018) according to the experiment results showed that it is an effective wheelchair system and can be used by disabled and elderly people to easily move from one place to another independently with minimum effort.

- ## Autonomous Navigation of the Intelligent wheelchair:

Image removed due to copyright restriction.

*Figure 7 Maps of the experimental environment (Cavanini, et al., 2014).*

Autonomous or assisted navigation intelligent wheelchair systems are developed for increasing the efficiency and effectiveness of mobility of the disabled person(Mittal & Goyal, 2014). Intelligent wheelchair system with the computer system Linux based along with infrared sensors and Kinect camera RGB-D is proposed by (Chang, et al., 2013) in their research. This intelligent wheelchair system is constructed and designed with the integration of obstacle avoidance technology, wall following automatically, and passage identification capabilities. This intelligent wheelchair autonomous navigation system was regarded as NavChair, the was initially came into development in Michigan University, United States from 1993 to the year of 2002 through assistance from high-quality computer systems along with the ultrasonic range finder series to ensure that this system is capable to find and detect obstacles in its path. This system also had wheel encoders of odometry by the assistance of a joystick panel(Cavanini, et al., 2014). Individuals, who can benefit from the NavChair system, are those who find it difficult to move from one place to another without assistance or those that have quadriplegia or quadriparesis resulting from the injuries in spinal cord, neuromuscular disease or cerebral palsy and people that are suffering through perceptual impairments (Mittal & Goyal, 2014). With the consideration of NavChair system analysis in terms of its performance, it has demonstrated the capabilities of an efficient system aim towards giving independent mobility to disabled individuals that are incapable to independently control a manual or powered wheelchair (Wei, et al., 2012). NavChair intelligent wheelchair is designed for allowing various operating levels, which range from simple obstacle-avoidance to total autonomous wheelchair navigation (Assante & Fornaro, 2015). The requirement is also of integrating additional environmental sensors in intelligent wheelchair system. The current design of NavChair has small number of sensors integrated on each side, but there are not any sensors present at the back of the chair. That is why, the objective of final NavChair

Image removed due to copyright restriction.

*Figure 6 wheelchair diagram (Wei, et al., 2012).*

design was to not just integrate a power module, Joystick module, wheelchair motors, NavChair control software, and wheel counters but also integrated the sonar sensors in NavChair.

NavChair project got its name from the intelligent wheelchair MIT project; this was started initially in the year of 2005, as the project got designed to get entirely controlled by speech recognition. The wheelchair system works similarly to the autonomous robot system as an individual using this wheelchair system commands the wheelchair through speech supervising the movements and controlling the target location(Simpson & Levine, 1997). To get to the target destination, this intelligent, autonomous wheelchair system requires typically the complete map or address of the target region or connection to GPS (Mittal & Goyal, 2014). The NavChair system is not only integrated with speech recognition technology but it also has the Kinect camera enabling the wheelchair in detecting and avoiding obstacles in its path that might hinder its mobility(Theodoridis, et al., 2013). Integration of sensors in NavChair allows this system to devise the latest path for navigation every time it is reassuring to avoid any sort of accidents or collisions.

- Homography Matrix:


$$s\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

*Figure 8 Homography Matrix*

*Figure 9 dx= h31 and dy= h32 values regarding to the object position in camera frame*

The homography matrix is the 3x3 matrix, but it has eight DoF's (degrees of freedom) as this matrix is estimated up to a scale. The homography matrix is computed among the images that are shot from the same camera but in different angles. The homography associated with coordinates of pixels of images that are taken i.e., if x = Mx when the homography matrix is implemented on all pixels a new image is generated which is known as the wrapped version of the original image. The conditions of the homography matrix include, the two images that are being analysed can be considered related by the homography if each image is viewing the same plane at different angles. Furthermore, the important thing is to ensure that both images are shot from one camera, however obviously at different angles and the camera must be rotated from its centre of projection. The important thing to note is the fact that the relationship of homography is independent of the structure of a scene. The relationship is not dependent on what the cameras are looking at as the relationship holds regardless of the fact what is seen in the images. If the rotation R and the calibration K are known, then the value of Homography M can be directly computed as $x' = KRK^{-1}x$. The application of this homography to one image provides the image that one would get if the cameras were rotated by R. Inverting of M for getting $M^{-1}$ is similar to as applying the inverse rotation $R^{-1}$. But if there are two rotated images but do not have any idea about the rotation, then the homography can be computed by considering the given set of correspondences, pixels in the left image which equals the right image(Ning, et al., 2011). Then the homography equations are written down which must relate the correspondences and the computation of the homography is done using the same method as utilized for the computation of fundamental matrix or for the computation of projection matrix. It is required to compute the eigenvector, which is associated with the smallest eigen-value of A $A^T$ matrix. Things that are important in computing homography matrix are; if the translation is null, the epi-polar geometry fails to hold, and in this case, one can only get image rotation. Depth for any point cannot be computed for which correspondences are present, and the homography matrix can be computed from the known rotation and camera calibration or the correspondences between two images. If only correspondences are used for computation, then one can make a mosaic from the rotating images. The mosaic can be made actually whenever there is a homography relationship i.e. when rotating or looking at a plane or having a higher resolution camera from the single point of view.

## Chapter 3: Experimental & Calculation Results:

Appendix A and B of this report illustrates the calculation results that were achieved in this project i.e., the relevant calculations and code to accomplish the wheelchair system capable of finding the objects in its path, tracking the objects and identifying doors by their door tags. The three primary methods that were followed for the planning, development and testing of the Kinect camera and Arduino board based smart wheelchair system are as follows.

### 1- Finding the Object:

The first method of this project was to ensure that smart wheelchair is able to identify doors along with objects in its path using the image recognition technology and with the help of the integrated Kinect camera in the wheelchair. For this purpose, various literature articles were reviewed as illustrated in the previous section of this report to find the application of a wheelchair system that is capable of identifying objects through image recognition with the help of Microsoft Kinect camera technology. Primary parts of the wheelchair system have a power wheelchair board, touch panel, CMOS sensors, Kinect camera, and the PC system. The interface of CMOS sensor, Kinect camera, touch panel, and the PC is integrated within the wheelchair system (Chang, et al., 2013). The PC system is connected to the Microsoft Kinect interface, which mainly handles and controls the system's gesture recognition capabilities making it possible to identify location of disabled or elder individual and also planning the path of movement of the wheelchair from the current location of the wheelchair towards the disabled individual(Chang, et al., 2013). After the server completes the detection of disabled person position who is commanding the wheelchair system through gestures, PC server of wheelchair does planning regarding the location and path, then it lights the LED infrared light according to a route planned along with directing wheelchair movement towards disabled individual(Chang, et al., 2013). This action occurs when shot image by CMOS sensor with Kinect camera is faced towards a ceiling. Mainly CMOS sensor captures images of the roofs that have the LED infrared-lighted (moving route) displayed periodically (Chang, et al., 2013). Kinect system does processing of images that were captured and ensures the wheelchair system follows the specified moving path directed by PC server in connection with the wheelchair. Disabled individuals using this system have the requirement of just giving gestures to control the wheelchair movements. The mission is achieved through recognition of different gestures and actions of a disabled individual with the use of a Kinect camera. The system has three main movements or gestures, which are recognizable. These are the call gesture, parking gesture for wheelchair and regular gesture of movement. This system particularly focuses upon the movements of the limb of a disabled individual for acknowledging the particular gestures (Chang, et al., 2013). For example, this system is capable of recognizing the wheelchair call gesture when a disabled person raises a single hand up above his shoulder. Also, when a person raises both arms above the shoulder, the system then recognizes this gesture as a parking wheelchair gesture in its system(Chang, et al., 2013).

All other gestures like the movement gestures of left, right, forward and backward are regarded as general gestures for controlling wheelchair movements when the disabled person is on the wheelchair. Kinect system based wheelchair system project was developed to ensure a power wheelchair system that is capable of improving the living quality of the disabled people(Liu, et al., 2011). This project aimed to create a system that is not only capable of recognizing the hand movement and gestures of the disabled individual but also enable him to control the wheelchair through specific hand gesture and movements. Furthermore, this wheelchair system was developed to have the capability of automatically moving to and from one location to another just by reading the hand movements and gestures of the disabled person. This wheelchair system also had a touch panel to move the wheelchair only in case the system fails to recognize the particular hand gestures and movements. This system is mainly divided into three components: Tx-Rx operation, controlling and gesture recognition of the wheelchair system. The primary parts of the wheelchair system are mobile application, power wheelchair system prototype, a PC server, Microsoft Kinect, the interface of ZigBee, and IR sensors. Both ZigBee interface and the touchpad interface are installed in this wheelchair system(Saadatzi, et al., 2013). The gesture recognition capabilities of the system are handled by the PC server with the Microsoft Kinect interface and mainly identify the location of the disabled individual and plan the moving path from the current position of the power wheelchair towards the disabled person. As PC server that is in connection with the Microsoft Kinect interface detects the request of the wheelchair based on the hand gestures and movement. It then decides the movement path of a wheelchair based on certain algorithms, after which the infrared system of the wheelchair activates, and the wheelchair moves towards the disabled person following the route planned by the PC server. This wheelchair system is built on the technology of line-following robots that follows the path and parks near to the patient.

Furthermore, the patient can also use the smart phone application of the system for controlling the navigation of the wheelchair towards his desired location and then he can ensure that he leaves the wheelchair on the line to move towards its original parking location with the use of a specific hand gesture or movement that is sensed by the Kinect camera attached on the wheelchair. The main idea for the development of this system was to help a disabled person that has difficulty in physical movement to control the wheelchair through their hand movements and gestures. Furthermore, the wheelchair system not only is capable of reading and responding to hand gestures and movements but also can be controlled by speech recognition with the integration of speech recognition technology(Rehman, et al., 2009). Also, the author proposed that in future works the range of the gestures that can be understood by the system could be increased so that if the system fails to read one call gesture from the disabled individual, he can use an alternative gesture. Lastly, Yatnalli et al., (2012) proposed that there are some other areas, which can be, improved in future works, i.e., the size of a wheelchair, the Microsoft Kinect camera cost along with the overall cost of the wheelchair system.  The vision algorithms are used for the detection of the object details in this wheelchair system. The find-object package has multiple features, which are used in this project. It has the capability of detecting the features of an object by using OpenCV that supports descriptors and detectors like FAST, STAR, SIFT, BRISK, SURF and GoodFeaturesToTRack.

## 2- Tracking the Object:

The second methodology in the planning, development, and testing of this smart wheelchair system project was to ensure that the system is capable of tracking objects especially if the order is successful in the identification of the object in terms of its features and position. For this purpose, various journal articles were sourced to find numerous applications of smart wheelchair system in which objects are tracked using the image recognition technique and Kinect camera. According to Tomari et al. (2012), in past years, several different methods have been introduced for the development of smart wheelchair systems for accommodating disabled people. The development trend of the intelligent wheelchair system can be classified into three main areas; the improvements in mechanics of assistive technology, the improvements in the user-machine physical interface, and the improvements in shared control among the machine and the user (Tomari, et al., 2012). The critical aspects of an intelligent wheelchair system are to give the users independent mobility especially those that are suffering from severe impairments and are not capable of controlling the wheelchair employing the standard joystick or control system. This might be because of several different reasons, for example, cognitive impairment, cerebral palsy, or severe fatigue(Fukuda, et al., 2009). According to previous research people that suffer from mobility issues tend to get more anxious and depressed as compared to able-bodied and healthy persons. Therefore the recovering of their mobility might help in the significant improvement of their mental health and quality of living. Generally, developed smart wheelchair systems and platforms largely depend upon the profile of its user i.e., disabilities and abilities, and there is no single resolution that is currently present which is suitable for all types of users. The patients who are suffering from motor impairments, for example, spinal cord injuries regularly lack in muscle controlling and worse case scenarios are unable to command the movements of their legs or hands. For aiding the mobility of these patients the actions and cues, which are constructed from the head, i.e., brain, voice, bite, gaze, or tongue can be used as input devices in a smart wheelchair system for all types of injury levels. The meaning of word "possible" in this context is that if levels of the damage are low the patients can utilize their hands or head for generating the input commands of a wheelchair system but as the level of damage increases the inputs sourced from the leader has to be considered as the only solution. The voice-activated or recognition navigation of a smart wheelchair system requires a quiet ambiance and might not be regarded as suitable for use in noisy and busy environments(Niitsuma, et al., 2010). The consideration of the brain wave for controlling the machine movement is highly complicated and challenging to achieve technology, particularly for a smart wheelchair system.

For this, a wheelchair system must be capable of reading electroencephalography EEG signal patterns for classifying and grouping them into intended actions. Furthermore, the user is required to have the right concentration and emotional control for adequate control. This is not only a burden for the system but also the user. However, this system might be considered for completely paralyzed patients. The author Tomari et al. (2012) proposes a semi-autonomous wheelchair control technology with the integration of the multiple-input interface for aiding the mobility of disabled people suffering from severe mobility impairments. Since the disabled person using this system can provide limited controlled command in the short duration of time, PC server or computer linked with the system takes on responsibilities of navigation along with avoiding any possible threats of accidents and collisions as the disabled person only has the responsibility for heading or leading wheelchair in the specified and desired locations(Njah & Jallouli, 2013). With the application of this type of setup, dangerous and critical scenarios can get tackled effectively and at the same time, disabled person can feel still responsible for driving and controlling the mobility of the wheelchair. The system proposed by Tomari et al. (2012) is implemented on the electronic wheelchair i.e., TT-Joy by Matsunaga corporation. This wheelchair system is integrated with the switch along with sensors of four types i.e., the standard Logicool webcam, the RGB camera of Microsoft Kinect, the unit sensor VN-100 for inertial measurements by Vectornav and range finder laser UTM-04LX by Electric Machinery Hokuyo.

In this system, the first human-computer interaction HCI switch is an input device. Its nature is momentary type and single, which is responsible for triggering the several different maneuvering modulations like for example manual, semi-auto or stops depending upon how long a user keeps the system on. This can get realized through different mediums like detecting of the different facial parts like eye shaking or blinking, the button switch or voice. The physiological features might impose a lot of burden on the disabled person when the person is required to frequently provide a command, particularly when the user is navigating in a limited area or space. That is why a simple bite like switch button is used in the system(Takahashi & Seki, 2009). The second input HCI device of this system is the webcam, which is used by the system for steering the wheelchair to its manual modulation.

Furthermore, it also gives the regular direction of movement when the wheelchair system is facing obstacles in a semi-autonomous way. For the sensor component of this system, the association of Kinect and laser is utilized for the detection of barriers along with dangerous areas in the vicinity of this wheelchair system(Kobayashi & Nakagawa, 2015). Before the utilization of each sensor, the calibration gets performed for ensuring the data reliability that is gathered. For this purpose, Tomari et al. (2012) utilized the standard procedures of stereo calibration for a linear calibration model and Kinect from tailoring information of laser. IMU sensor regarding the latest state of a wheelchair according to world coordinates, provides the information, and it is employed mainly here for the correction of heading orientation. This wheelchair system has the capability of recognizing the objects by their distinct edges and contrasting colors. The object_3d.lauch is used by the system for processing the object information. Each object is given a special object ID that is used by the system for tracking each object.

### 3- Finding the Doors:

The third main functionality of this wheelchair system is the capability of finding and identifying doors from the door tag. The aim of this functionality is to enable the wheelchair system to detect a door tag and display the information of the door tag to the user on the computer screen. Similarly, for the achievement of this functionality, various journal articles were sourced in relation to this information to find essential data regarding the application of this particular functionality. The output provided by the sensor gets calibrated, and the issues get compensated with the utilization of the extended Kalmar filter. That is the reason why the production of this system provides reliable six DOF readings. During the fulltime operation of this system, the HCI inputs are not used directly for continuously controlling the wheelchair; instead, the user with the manual mode uses them for heading a wheelchair in the intended direction. This system has a computer through the controller (semi-autonomous) which takes over navigation responsibilities along with low-level controls while the user of the system is only responsible for determining the path and general travel locations(Yukawa, et al., 2010). This causes lower user involvement in the operations of the system and assurance of higher comfort of the user. While maneuvering if a user decides to deviate or stop moving towards the specified path, he or she is always capable of interrupting by issuing the manual command or stay with the use of input switch. In the manual mode of the system, the gaze directions are used for steering the wheelchair right or left until the desired user direction is discovered. The gaze directions can be issued by the user several times by repeating the same process over and over again until the user is satisfied with the location. When this operation is completed, there are two possibilities; the first possibility is that the user already reaches his desired destination, and the second is that the user wishes to travel towards the specified location. If the disabled person is faced with the second possibility, he is required to issue a semi-autonomous command. The computer or PC server will set up the current orientation of the wheelchair as the latest travel location and resume its navigation along with avoiding obstacles in the path with the help of the sensors. When the semi-autonomous control system of the wheelchair is activated the wheelchair moves towards the specified or desired direction of the user avoiding objects and refining orientation when it is required. Due to the sequence of these operations, the final position of the wheelchair might get diverse a bit from its target or intended location(Stamps & Hamam, 2010). In these scenarios, the user has the capability of manually steering or repositioning wheelchair until it reaches its desired location. The reason why this system is tuned in this manner is that unlike the joystick-controlled systems, any sort of alternative medium offers few distinctive commands, which sometimes have errors in the process. That is why solely hinging on such type of medium might lead towards an inadequate motion command.

Furthermore, this type of medium needs more attention from users and consequently considering a long-time performance this might become difficult for the user and might discourage the quality of life and enjoyment of surroundings when maneuvering with the system. Therefore, with the integration of the manual controls along with the automatic controls of the wheelchair provides more freedom to users. The user can partly move the wheelchair or control its movements by giving the command to set the latest goal directions as they wish naturally and easily while ensuring the reduction of the navigation difficulty for the longer-time operation.

The joystick conventional system is well renowned for its functionality in effectively controlling a wheelchair. That is why when the users are incapable of using the regular capabilities of the system; the best alternative solution must be the imitation of the joystick operation. This must get easily adapted by users and help in minimization of workload. In the proposed system by Tomari et al., (2012) the gaze direction is used to command leftward or rightward direction, as it has the ability to reflect the behavior of human beings, as they change orientation while walking or driving. For getting the gaze data, the face API software is used by the system, which is capable of supplying relevant information on a real-time basis. The system only takes into account the yaw angle, as it can sufficiently identify the direction of the current gaze. For getting an insight into how the head gazing behaves naturally while the wheelchair is driven by the user, the system makes use of recorded values of angles made when the user directs the wheelchair with the help of the joystick. Based upon the results, the yaw angle values are empirically classified into action to determine the three possible effects of moving forward, turning left, and turning right. For the turn actions, the higher amount of yaw angle concerning the predefined thresholds along with the sharper turning curves is generated. However, only when the user operates the button switch, are the gaze commands executed(Rossen, et al., 2012). The setup helps distinguish the actions of the user, such as steering the wheelchair or looking around, by the controller. Different functions are triggered by the switch button, based on how the button is pressed. When the system receives a cycle of momentary pattern, the semi-auto or stops, the command is executed depending upon the current state i.e., when the current state of the wheelchair is completed the system automatically performs semi-auto state and vice versa. When the system continuously receives a high signal, it enters into the full user control mode until the profound message is issued by the user. Every time the user decides to exit from the manual mode, the system stops and waits for the next command from the user. The top priority in the order is the safety of the user, and that is why in all the operating modes of the system the safety map is supplied for ensuring that the motion that is generated is free from a collision. In the semi-auto way of the operation, the safety map guides the computer for creating an optimal action while seeking goal direction and avoiding the obstacles in the path of the wheelchair. In the manual mode of the system, the safety map helps the user to brake when the given command is subjected to collision.

The reasoning for the development of the safety map is evaluated by the researchers Tomari et al. (2012), who used the combination of the Microsoft Kinect camera and the laser sensor. The laser sensors are utilized for perceiving the solid-state objects or obstacles placement, and the Kinect camera is utilized for the detection of any sort of obstacles that are not uniform in the dimension of their shape i.e., a chair or a table. In this system, both the Kinect camera sensor and the laser sensors complement each other in the sense that the laser system has a wide FOV, but it is unable to portray the surroundings precisely. On the other hand, the Microsoft Kinect camera is a smaller FOV and is capable of recovering 3D information. The Kinect camera on the wheelchair is mounted and fully calibrated at approximately 1.3 meters high from the ground plane. Since the Microsoft Kinect camera is utilized by the system for the detection of obstacles, the camera is tilted in the downward direction to ensure that most likely the ground information is partly visible to the camera. For pruning out such information, this wheelchair system relies on the height map. When the Kinect camera is facing forward, the ground plane can be easily determined using the simple series of height thresholds, which is the actual height where the Kinect camera is installed in the wheelchair. But when the Kinect camera is tilted in the downward direction, this assumption will not work as the farther the ground plane is from the camera, the smaller will be the threshold of height. For overcoming this issue Tomari, et al., (2012) modeled the ground plane as "$\Delta: Z = \alpha X + \beta Y + \gamma$" where the variables X, Y, and Z represent the location of 3D points in the world coordinates and the α, β and γ represents the constant parameters representing the 3D plane of the ground. With the selection of the non-collinear points, the estimation of parameters is done belonging to the floor and with the performance of the least square fitting methodology. As the laser and Microsoft Kinect sensors in the system are located at different axes, the designers Tomari et al. (2012) calibrated both data with the minimization of the reading error in each of the FOV angles. The calibration process makes use of Kinect axis mainly as a reference. The researchers Tomari et al. (2012) empirically discovered that the linear transformation model is enough for the attainment of higher precision of the fitted data with the lower computational complexity. The safety map ensures that both data for the laser and the Kinect are aligned well and the Kinect system is successful in the detection of the table, which is placed on the right side of the wheelchair, which the laser system is unable to sense correctly. According to the findings of the experiment conducted by Tomari et al. (2012) with this wheelchair system, shows that the ability of Kinect camera in the successful detection of the table as compared to the laser illustrate the importance of why sensor integration of Kinect and laser is important.

The researchers Tomari et al. (2012) believe that with more information, the detail safety map of the local surroundings can be mapped by this wheelchair system. Furthermore, with the acquisition of a safety map, which is more accurate, can ensure the reduction in the risk of collisions, mainly when the obstacles in irregular shapes exist in cluttered environments. The researchers Tomari et al. (2012) conducted the first experiment with this wheelchair system for evaluating the ability of the system for responding to the commands of the users through the designed HCI of the system. The user drove the wheelchair on three locations that were predefined in the lab environment, and which were cluttered through the use of semi-auto and full manual controls of the wheelchair. Considering the comparison, the user was asked to control the wheelchair through the means of the standard joystick. According to the findings, with the utilization of all the interfaces, the user was able to reach all the checkpoints and was successful in the completion of the maneuvering tasks in the experiment without engaging into any collisions. According to the findings regarding the completion time of jobs, the joystick system was the fastest, and both the semi-autonomous and manual modes required approximately double of the joystick case time. This might have happened due to the fact that unlike joystick both the semi-auto and manual medium needed a bit more time for reinitiating the commands of the user and re-orientate towards the desired direction or location of the user. Considering the distance of travel, there are no significant differences generally among all the mediums, but the semi-auto medium of the system illustrated the shorter length of journey as compared to the other two i.e., joystick and manual medium. Which is the indicator that the user is capable of reaching his or her destination goal with the utilization of the HCI proposed in almost the same travel distance as with the usage of the joystick medium? Furthermore, this also illustrates semi-auto mode reliability as an alternative interface for the user to joystick control. According to Tomari et al. (2012), one way of measuring the performance of the system is utilizing the complexity of operating a wheelchair. This can be measured by counting the time, which the user requires to complete the interaction with the order for the completion of a particular task. According to the experimental results of this system, the semi-auto mode requires a lesser amount of effort from the user at the functional end, as compared to the manual medium. This is because in the semi-auto mode the user is required to interact with the system only when he wishes to change the direction of travel.

From testing the semi-auto mode of this wheelchair system Tomari et al. (2012) discovered that in this mode, the user is only required to issue as low as one-third of the commands issues in a manual method of the system. Furthermore, along with the assurance of less workload on the user, the semi-auto mode still provides the user with freedom of changing the path of the wheelchair in a natural and straightforward manner. That is why the semi-auto mode is considered as the better option, provided that the old-fashioned way of the system is easy to use. In the second, experiment Tomari et al. (2012) investigated the safety map feasibility, i.e., whether the map is capable of preventing the wheelchair from engaging into collisions particularly in the manual mode, and is capable of steering away from the obstacles in the semi-auto mode. As described earlier, the integration of the laser and Kinect sensors is beneficial in the production of a safety map. According to the results, the laser is unable to clearly or correctly detect the whole table of obstacle shape in the path of the wheelchair, and according to the resultant POD, it illustrates a motion towards the table that could be dangerous to the user. But with the integration of the Kinect system, the POD is capable of blocking any movements towards the desk and provides motion in the left direction, which helps steer clear. Similarly, in another situation in the second experiment, the combined sensor data of the laser and Kinect was able to successfully prevent the wheelchair from colliding with the obstacle iron rack. These examples illustrate the importance of the versatility in the fusion of multiple sensors for visualizing the surrounding environment more appropriately as compared to just relying upon the single perception tool. For the evaluation of the safety map in the experiment, the user was tasked to steer the wheelchair through the environments and intentionally issue a command to the system that might subject the wheelchair to the collision. In the manual mode, the driving signal which was sent to the wheelchair system by the user usually was followed by the user gaze direction i.e., when the user looked in the left direction the wheelchair also turned in the left path. But the system did not execute the commands of the user i.e., and the motor command was equal to zero as it was not safe and might have caused an accident. That is why the wheelchair system remained stationary to avoid any collision. Similarly later in the experiment, the user commanded the wheelchair to turn in the leftwards direction and since the steering command of the wheelchair did not show any possibility of the collision the system correctly responded to the authority of the user and executed the command. This type of verification process is carried out continuously in this order as long as the user is keeping the wheelchair in the manual mode. The response of the wheelchair system might be different if it is in the semi-auto mode as the head direction is valid only when the wheelchair is facing an obstacle i.e., an object that is blocking the goal destination of the wheelchair.

In the semi-auto mode of the wheelchair, the direction of avoiding the obstacle is also followed closely by the gaze direction of the user. When the wheelchair system faces an obstacle, the user of the wheelchair in the experiment gave the right direction suggestion, but as this direction was dangerous and might have caused a collision the planner of the system rejected the user command and instead steered in the left direction to maintain the safety. Once the wheelchair in this experiment reached the appropriate space, and there was no chance of collision, it corrected its path towards the user's desired destination to enter the goal location following the correct way. The results of this experiment illustrate the semi-auto controller of the wheelchair system benefits for providing help to the user of the wheelchair by passing through or avoiding obstacles automatically and also safe while seeking the desired location or goal location. Besides that, when it is appropriate, the planner is capable of moving the wheelchair by following the preference of the user while avoiding the obstacles in the path. For the real-world implementation of this system, the execution time is critical because the system is required to respond as soon as it detects any threat or obstacle in its path. For this system, the vision part needs approximately 83ms for the accomplishment of the task of 3D data mapping. With the integration of the IMU sensor along with the laser range finder, the total execution time of the system increased as the laser requires approximately 25ms for the completion of a cycle of scanning and the IMU requires approximately 5ms of acquisition time. In conclusion, the execution time of the system was approximately 113ms, which is about eight frames per second. Considering the current application of this wheelchair system, this processing speed can be considered enough for this wheelchair system to receive, analyze, evaluate, and execute the commands of the user effectively and promptly. Therefore, the proposed wheelchair system of Tomari et al. (2012) according to the experimental results provides a smart wheelchair system, which is capable of catering to the needs of the disabled person particularly those that are suffering from severe motor disabilities. The proposed system has as described earlier two control methods or modes the semi-auto mode and the manual mode. This freedom of choice is provided to the user to freely change the patterns according to their comfort and liking by using the manual switch operation of the wheelchair. Furthermore, with the incorporation of the safety map capabilities, this wheelchair system is capable of avoiding obstacles in its path to prevent collisions in both manual and semi-auto modes, and hence it ensures the reduction of the burden on the user of continuously controlling the wheelchair manually and monitoring the surroundings to avoid obstacles while maneuvering. The operational experiments of this system conducted by Tomari et al. (2012) in the actual cluttered environments illustrated that this wheelchair system is capable of maneuvering efficiently and effectively without engaging in any collisions. In future work, Tomari et al. (2012) provide evidence that the system with more participants can help in gaining the generality of the reliability of the overall system. Moreover, the cognitive complexity of the users can be measured with this system with the introduction of an extra amount of workload during navigation in future experiments.

## Chapter 4: Discussion:

After conducting a thorough literature review analysis related to the three main objectives of this project, the following are the experimental results that were achieved during the planning, development, and testing of this smart wheelchair system. The following image illustrates the functionality of this wheelchair system image recognition technology and how it uses the Microsoft Kinect camera for finding and tracking objects in the path of the wheelchair system.

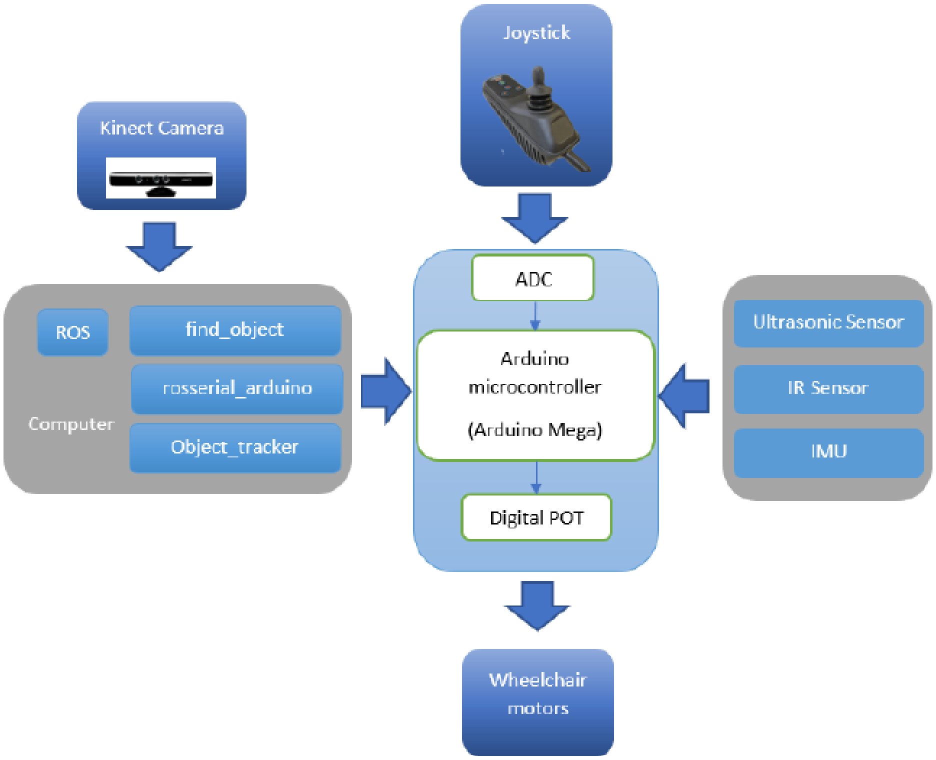



*Figure 10 Connection Arduino board with computer and joystick*

The above-illustrated image shows the whole connection Arduino board of this smart wheelchair system in connection with the equipment and the joystick module. The Arduino Uno board, as illustrated in the figure above, takes inputs from the joystick module of the wheelchair, computer and the Ultrasonic sensors, IR sensors and IMU attached with the wheelchair. The Arduino board analyses these inputs and converts them into digital signals that are transferred to the motors of the wheelchair, which then reads those commands. The wheelchair motors are responsible for executing the user commands for not only moving the wheelchair from one place to another but also avoiding objects and obstacles in the path of the wheelchair that is detected and identified by the sensors or Microsoft Kinect camera of the wheelchair.

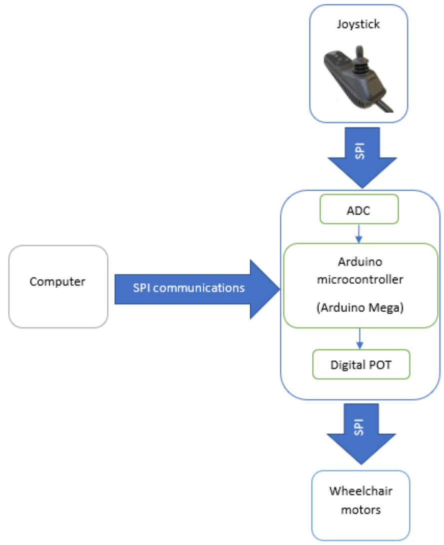



*Figure 11 Interface board and Arduino connection*

The above image illustrates the interface board and the Arduino connection of this smart wheelchair system. It is a simple SPI communication board where the user provides the input to the system through the computer or joystick module attached with the wheelchair, this information is then translated and analysed by the Arduino Uno board of the wheelchair system and it is decoded and transferred to the motor controller in the form of a digital output to ensure that the motor drivers understand the correct user input and execute the command effectively by moving the wheelchair in the direction specified by the user through the joystick module or computer.

1. **Finding the Objects and tracking the objects:**

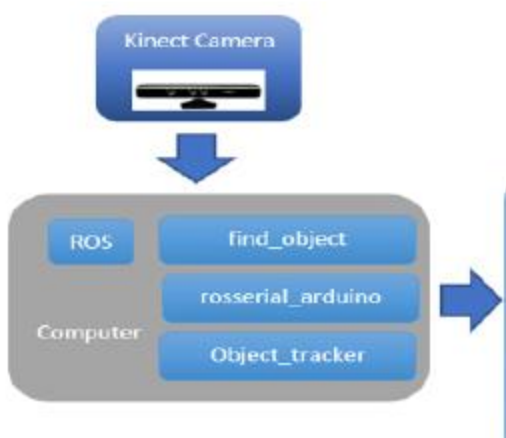



*Figure 12 Image recognition Kinect camera*

The figure illustrated above shows how this wheelchair system not only detects and identifies objects in the path of the wheelchair but also tracks those objects through the ROS module in the computer software. The following figure shows the flowchart of the smart wheelchair object finding and tracking system. Firstly, the Kinect camera attached to the wheelchair captures the image of the object in the path of the wheelchair. This image is then transferred to the computer system connected to the wheelchair system for analysing the particular model. The "find object package" of the ROS system analyses the image feature of the detected object, publishes the homography parameters and tf_listener. If the object is identified successfully by the system in terms of its position and features, then the system finds the door tag or tracks the purpose. If the door tag is identified, then the system displays the user door tag information, and the objective is followed to allow the user to stay away from the obstacle in the path of the wheelchair.

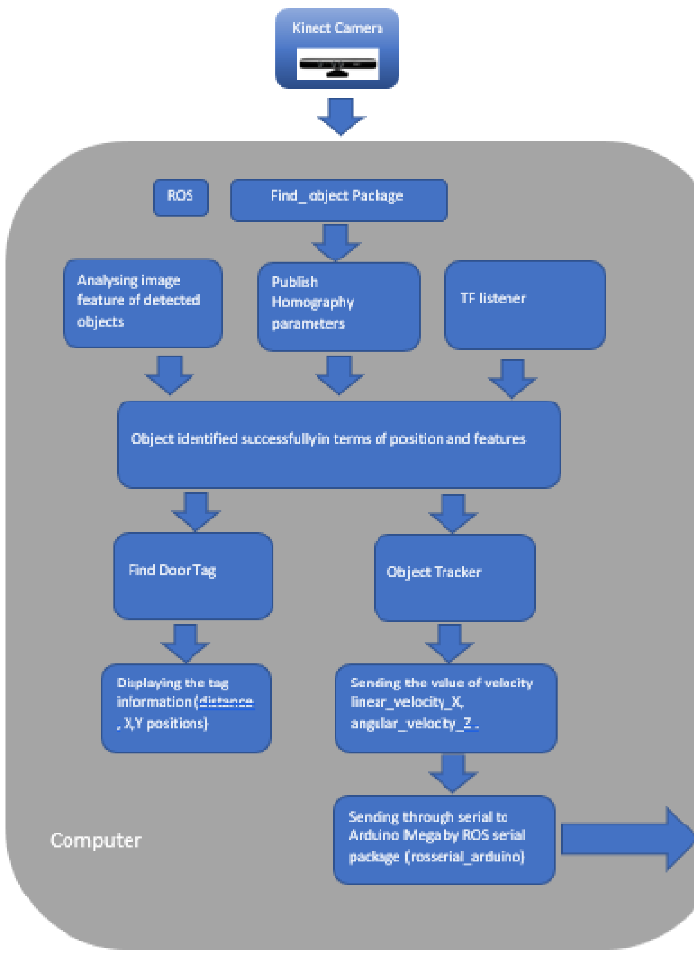



*Figure 13 Object tracking process smart wheelchair system*

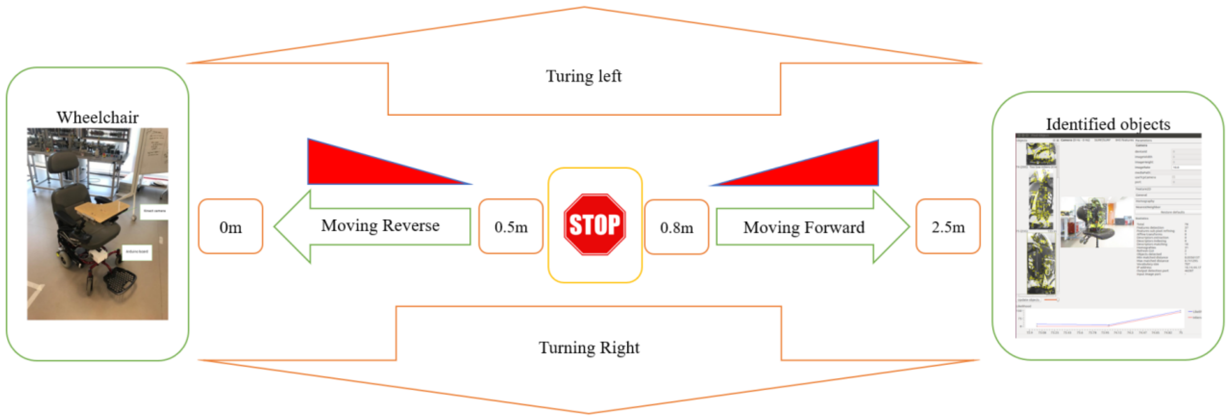



*Figure 14 Object tracking based on the position of the objects*

**2. Identifying objects from door tag:**

The following image illustrates how this wheelchair system is capable of identifying the doors from their door tag. When the wheelchair system detects the door card through the Microsoft Kinect camera attached with the wheelchair, it sends the relevant information of the door tag to the computer screen of the user as shown in figure-3 below. The door tag information, after going through the proper process and analysis, as illustrated in figure-2, appears on the user's computer screen. This information contains the door tag number, the distance between the camera and the door tag along with information regarding the position of the door tag i.e., x-plane position or y-plane position.

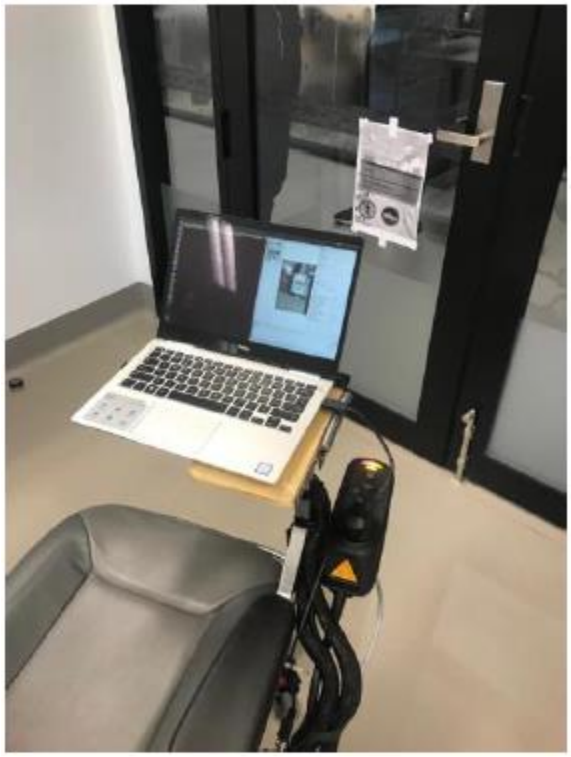


*Figure 15 Wheelchair identifying rooms by door tag*

The unique serial number or the tracking number of an object is delivered to the Arduino Mega by ROS following package. The calculations in the Appendix a section of this report illustrates how the software of this smart wheelchair system runs the serial communication in the computer system and the Arduino board of this smart wheelchair system. This controlled and effective communication between the Kinect camera, Arduino board and the computer system attached with the wheelchair system allows sufficient image identification, recognition, and tracking as illustrated by the execution of the code in the appendix section of this report. The three main objectives of this project as identified earlier were to ensure the fact that this smart wheelchair system is not only capable of detecting and finding doors in the path of the wheelchair but also be able to track objects and identify entries from their door tag. The experimental and testing results of this system show that this smart wheelchair project was able to achieve these objectives as the developed wheelchair system not only identified and tracked objects or doors in the path of the wheelchair thanks to the Microsoft Kinect camera attached with the system, but also this system was able to identify entries from their door tags and was able to display the door tag information on the computer screen of the user driving the wheelchair.

## Chapter 5: Limitations:

The only limitation of this project is the scale and practical application of this project. As this project was performed for experimental purposes only and the purpose of this project was to provide a prototype of an intelligent wheelchair system with Kinect camera and Arduino board. For the practical application of this project it will be important to consider the use of this wheelchair system in a real-world situation. Furthermore, the testing of this wheelchair system was completed in a closed indoor environment, which might have limited the overall testing of this wheelchair system. In future works it is recommended that this wheelchair system must be tested in both indoor and outdoor environments to provide the fully conclusive results regarding the overall performance of this system in a real-world situation. Moreover, the Microsoft Kinect system has got a lot of complaints from users regarding its difficulty in recognizing motion and gestures. If a better system is available than Kinect which have greater response times and gesture recognition timing then that system could be preferred over Kinect.

## Chapter 6: Conclusion:

The primary purpose of the project is of achieving the three main goals of the plan of the smart wheelchair system integrated with the Arduino board and Kinect camera. The first objective was to ensure that wheelchair is capable in identifying and finding the obstacles or objects in its path. The second goal was ensuring that this system is able to track objects, while the third objective was to ensure that this system is capable of identifying doors from their door tags. For this purpose, a thorough literature review analysis was conducted to find previous research, projects, and relevant journal articles related to these three objectives of the plan for the smart wheelchair system. After gathering pertinent information through the literature review, the design and prototype of the intelligent wheelchair system with the Arduino board and Kinect camera was planned, designed, constructed, and tested. The experimental results of this project illustrate that the newly designed smart wheelchair system was capable of achieving its three main objects. The Kinect camera used with the wheelchair system was not only helpful in identifying and finding objects in the path of the wheelchair but also helped in tracking purposes and identifying doors from their door tags. The user of the wheelchair controlled the movement of the wheelchair with the help of the joystick module, the sensors attached, and the Kinect camera attached to the computer system. These worked as automatic identifiers and trackers of objects in the path of the wheelchair to ensure avoidance of accidents and reduction of the collision of the wheelchair with obstacles in its way.

## References:


Achic, F., Montero, J., Penaloza, C. & Cuellar, F., 2016. *Hybrid BCI system to operate an electric wheelchair and a robotic arm for navigation and manipulation tasks.* Shanghai, IEEE.

Assante, D. & Fornaro, C., 2015. *Involving graduating engineers in applying a commercial brain-computer interface to motorized wheelchair driving.* Tallinn, IEEE.

Carrino, F. et al., 2012. *A self-paced BCI system to control an electric wheelchair: Evaluation of a commercial, low-cost EEG device.* Manaus, IEEE.

Cavanini, L. et al., 2014. *SLAM-based autonomous wheelchair navigation system for AAL scenarios.* Senigallia, IEEE.

Chang, C. L., Chen, C. C., Chen, C. Y. & Lin, B. S., 2013. *Kinect-based Powered Wheelchair Control System.* Bangkok, IEEE.

Chang, Y. J., Han, W. Y. & Tsai, Y. C., 2013. A Kinect-based upper limb rehabilitation system to assist people with cerebral palsy. *Research in Development Disabilities,* 34(11), pp. 3654-3659.

Faria, B., Reis, L. & Lau, N., 2014. A Survey on Intelligent Wheelchair Prototypes and Simulators. *New Perspectives in Information Systems and Technologies, Volume 1,* 275(1), pp. 545-557.

Fukuda, Y., Kurihara, Y., Kobayashi, K. & Watanabe, K., 2009. *Development of electric wheelchair interface based on laser pointer.* Fukuoka, IEEE.

Halawani, A., Rahman, S. U., Li, H. & Anani, A., 2012. Active vision for controlling an electric wheelchair. *Intelligent Service Robotics,* 5(2), p. 89–98.

Hashimoto, M., Takahashi, K. & Shimada, M., 2009. *Wheelchair control using an EOG- and EMG-based gesture interface.* Singapore, IEEE.

Jang, G. & Choi, Y., 2014. *EMG-based continuous control method for electric wheelchair.* Chicago, IEEE.

Khare, V., Gupta, S. & Meena, K., 2017. Voice Controlled Wheelchair. *International Journal of Electronics, Electrical and Computational System,* 6(4), pp. 23-27.

Kobayashi, N. & Nakagawa, M., 2015. *BCI-based control of electric wheelchair.* Osaka, IEEE.

Kondori, F., Yousefi, S., Liu, L. & Li, H., 2014. *Head operated electric wheelchair.* San Diego, IEEE.

Lin, J. S., Chen, K. C. & Yang, W. C., 2010. *EEG and eye-blinking signals through a Brain-Computer Interface based control for electric wheelchairs with wireless scheme.* Gyeongju, IEEE.

Liu, G., Yao, M., Zhang, L. & Zhang, C., 2011. *Fuzzy Controller for Obstacle Avoidance in Electric Wheelchair with Ultrasonic Sensors.* Kota Kinabalu, IEEE.

Lodhi, D. K. et al., 2016. Smart Electronic Wheelchair Using Arduino and Bluetooth Module. *International Journal of Computer Science and Mobile Computing,* 5(5), p. 433 – 438.

Maruno, Y., Okajima, H., Zengin, A. & Matsunaga, N., 2011. *Direct yaw moment control of front drive type electric wheelchair with free-casters.* Gyeonggi-do, IEEE.

Milella, A. & Cicirelli, G., 2010. *Mechatronic Systems: Applications.* s.l.:BoD – Books on Demand.

Mirin, S., Annuar, K. & Yook, C. P., 2018. Smart wheelchair using android smartphone for physically disabled people. *International Journal of Engineering and Technology,* 7(2), pp. 453-457.

Mittal, R. & Goyal, D., 2014. Autonomous Navigation of Smart Wheelchair using Kinect Camera. *International Journal of Engineering and Technical Research,* 2(2), pp. 357-358.

Nguyen, Q. X. & Jo, S., 2012. Electric wheelchair control using head pose free eye-gaze tracker. *Electronics Letters ,* 48(13), pp. 750 - 752.

Niitsuma, M., Ochi, T., Yamaguchi, M. & Hashimoto, H., 2010. *Interaction between a user and a smart electric wheelchair in Intelligent Space.* Nagoya, IEEE.

Ning, B. et al., 2011. Human Brain Control of Electric Wheelchair with Eye-Blink Electrooculogram Signal. *International Conference on Intelligent Robotics and Applications,* 7506(1), pp. 2-27.

Nishimori, M., Saitoh, T. & Konishi, R., 2007. *Voice Controlled Intelligent Wheelchair.* Takamatsu, IEEE.

Njah, M. & Jallouli, M., 2013. *Wheelchair obstacle avoidance based on fuzzy controller and ultrasonic sensors.* Sousse, IEEE.

Purwanto, D., Mardiyanto, R. & Arai, K., 2009. Electric wheelchair control with gaze direction and eye blinking. *Artificial Life and Robotics,* 7(1), pp. 14-39.

Rehman, S., Raytchev, B., Yoda, I. & Liu, L., 2009. *Vibrotactile rendering of head gestures for controlling electric wheelchair.* San Antonio, IEEE.

Rossen, C., Sorenson, B., Jochumsen, B. & Wind, G., 2012. Everyday life for users of electric wheelchairs – a qualitative interview study. *Disability and Rehabilitation: Assistive Technology,* 7(5), pp. 399-407.

Saadatzi, M., Poshtan, J., Saadatzi, M. & Tafazzoli, F., 2013. Novel system identification method and multi-objective-optimal multivariable disturbance observer for electric wheelchair. *ISA Transactions,* 52(1), pp. 129-139.

Seki, H., Ishihara, K. & Tadakuma, S., 2009. Novel Regenerative Braking Control of Electric Power-Assisted Wheelchair for Safety Downhill Road Driving. *IEEE Transactions on Industrial Electronics,* 56(5), pp. 1393 - 1400.

Shibata, T. & Murakami, T., 2012. Power-Assist Control of Pushing Task by Repulsive Compliance Control in Electric Wheelchair. *IEEE Transactions on Industrial Electronics,* 59(1), pp. 511 - 520.

Silva, A. N. et al., 2013. *Virtual electric wheelchair controlled by electromyographic signals.* Rio de Janerio, IEEE.

Simpson, R. C. & Levine, S. P., 1997. *Adaptive shared control of a smart wheelchair operated by voice control.* Grenoble, IEEE.

Stamps, K. & Hamam, Y., 2010. Towards Inexpensive BCI Control for Wheelchair Navigation in the Enabled Environment – A Hardware Survey. *International Conference on Brain Informatics,* 2(1), pp. 336-345.

Stenberg, G., Henje, C., Levi, R. & Lindstorm, M., 2016. Living with an electric wheelchair – the user perspective. *Disability and Rehabilitation: Assistive Technology,* 11(5), pp. 385-394.

Taher, F. B., Amor, N. & Jallouli, M., 2014. *EEG control of an electric wheelchair for disabled persons.* Sousse, IEEE.

Takahashi, Y. & Matsuo, S., 2011. *Running experiments of electric wheelchair powered by natural energies.* Gdansk, IEEE.

Takahashi, Y. & Seki, H., 2009. *Fuzzy logic based regenerative braking control system of electric wheelchair for senior citizen.* Kyoto, IEEE.

Tamura, H. et al., 2010. A Study of the Electric Wheelchair Hands-Free Safety Control System Using the Surface-Electromygram of Facial Muscles. *International Conference on Intelligent Robotics and Applications,* 6425(1), pp. 97-104.

Tamura, H., Manabe, T., Tanno, K. & Fuse, Y., 2010. *The electric wheelchair control system using surface-electromygram of facial muscles.* Kobe, IEEE.

Tamura, H. et al., 2012. Development of the electric wheelchair hands-free semi-automatic control system using the surface-electromyogram of facial muscles. *Artificial Life and Robotics,* 17(2), p. 300–305.

Theodoridis, T., Hu, H., Maier, K. M. & Gu, D., 2013. Kinect Enabled Monte Carlo Localisation for a Robotic Wheelchair. *Frontiers of Intelligent Autonomous Systems,* 466(1), pp. 17-27.

Tian, Z. & Xu, W., 2009. *Electric Wheelchair Controller Based on Parameter Self-Adjusting Fuzzy PID.* Wuhan, IEEE.

Tian, Z., Yang, J. & Zhang, J., 2009. *Location-based Services Applied to an Electric Wheelchair Based on the GPS and GSM Networks.* Wuhan, IEEE.

Tomari, R., Kobayashi, Y. & Kuno, Y., 2012. Development of Smart Wheelchair System for a User with Severe Motor Impairment. *Procedia Engineering,* 41(1), p. 538 – 546.

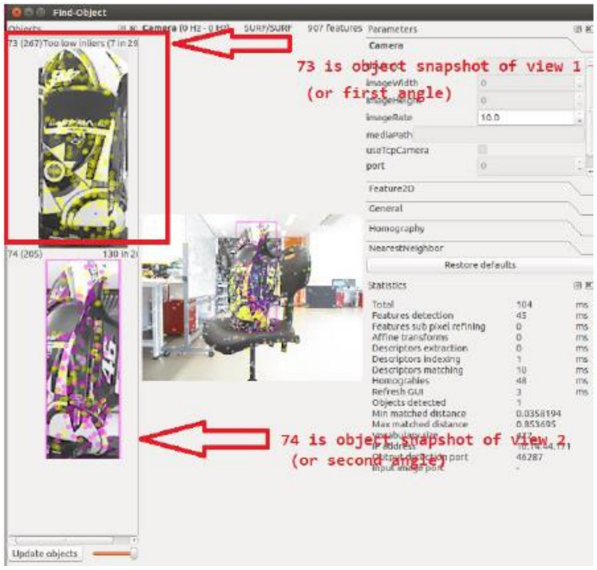
Wang, F.-C. & Chaing, Y. S., 2012. Design and control of a PEMFC powered electric wheelchair. *International Journal of Hydrogen Energy,* 37(15), pp. 11299-11307.

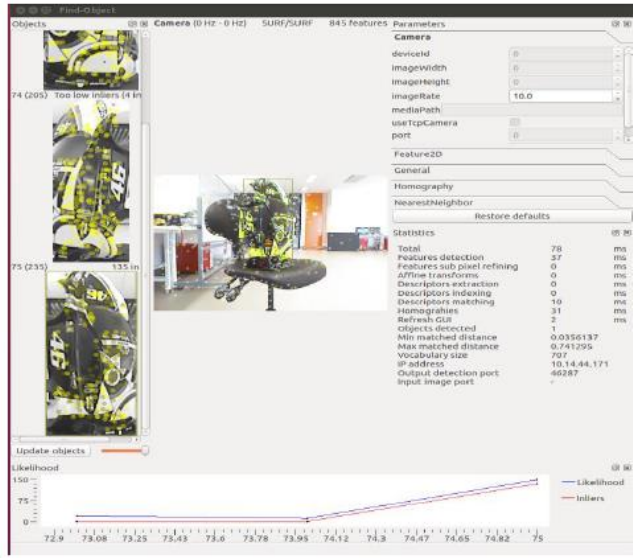
Wei, Z., Chen, W. & Wang, J., 2012. 3D Semantic Map-Based Shared Control for Smart Wheelchair. *International Conference on Intelligent Robotics and Applications,* 7507(1), pp. 41-51.

Yatnalli, V. et al., 2012. *GUESTURE BASED WHEEL CHAIR CONTROL USING KINECT CAMERA,* s.l.: GM INSTITUTE OF TECHNOLOGY, DAVANGERE.

Yukawa, T., Nakata, N., Obinata, G. & Makino, T., 2010. *Assistance system for bedridden patients to reduce the burden of nursing care (first report — Development of a multifunctional electric wheelchair, portable bath, lift, and mobile robot with portable toilet).* Sendai, IEEE.

## Appendix A:



object_tracker.cpp

```
#include <ros/ros.h>     // header files
#include "iostream"
#include <std_msgs/Float32MultiArray.h>
#include <geometry_msgs/Twist.h>
#include "geometry_msgs/Vector3.h"
#include <std_msgs/String.h>
#include <opencv2/opencv.hpp>
#include "math.h"
#include <tf/transform_listener.h>

#define follow_object1 73  // object snapshot of view 1 (or first angle)
#define follow_object2 74  // second view
#define follow_object3 75  // 3rd
#define follow_object4 76  // 4th
#define follow_object5 77  // 5th
#define follow_object6 78  // 6th
#define Turn_Left 79
#define Turn_Right 80
#define Moving_Forward 81

int id = 0;
ros::Publisher pub;
geometry_msgs::Twist vel;
int camera_center = 320;           // cameraWidth from the left 0 to right 640
float max_ang_velocity = 0.7;      // maximum angular velocity
float min_ang_velocity = 0.5;      // minimum angular velocity
float ang_vel = 0;                 // inital angular velocity is zero
```

```
float max_distance = 2.2;           // maximum distance
float min_distance = 0.9;           // minimum distance
float stop_dis = 0.6;               // stop distance
float max_linear_vel = 1.5;         // maximum linear velocity
float min_linear_vel = 0.3;         // minimum linear velocity
//float average_dist = 0;
float distance = 0 ;

tf::TransformListener *listener;//to use listener as pointer
tf::StampedTransform poseCam; // position of the object based on camera position

void objectCallback(const std_msgs::Float32MultiArrayPtr &object) // callback function getti
ng messages and the data of recognized objects
{
   if (object->data.size() > 0)
   {
      id = object->data[0]; // get object ID from array

      float objectWidth = object->data[1]; // get objectWidth from array
      float objectHeight = object->data[2]; // get objectHeigh from array
      float x_position;       // variable for calculating object central position in x axis
      float speed_coefficient = (float)camera_center / max_ang_velocity / 4; //calculate spe
ed coefficient

      // Find the corners OpenCV
      cv::Mat cvHomography(3, 3, CV_32F);
      std::vector<cv::Point2f> inPts, outPts;
      switch (id)
      {
      case Turn_Left:  // if get Turn_Left sign
         vel.linear.x = 0.0;
         vel.angular.z = 0.5; //Turn Left the robot
         break;
      case Turn_Right: // if get Turn_Right sign
         vel.linear.x = 0.0;
         vel.angular.z = -0.5; //Turn Righ the robot
         break;
      case Moving_Forward: // if get Forward sign
         vel.linear.x = 0.5; //Move the robot Forward
         vel.angular.z = 0.0;
         break;
      case follow_object1:
      case follow_object2:
      case follow_object3:
      case follow_object4:
```

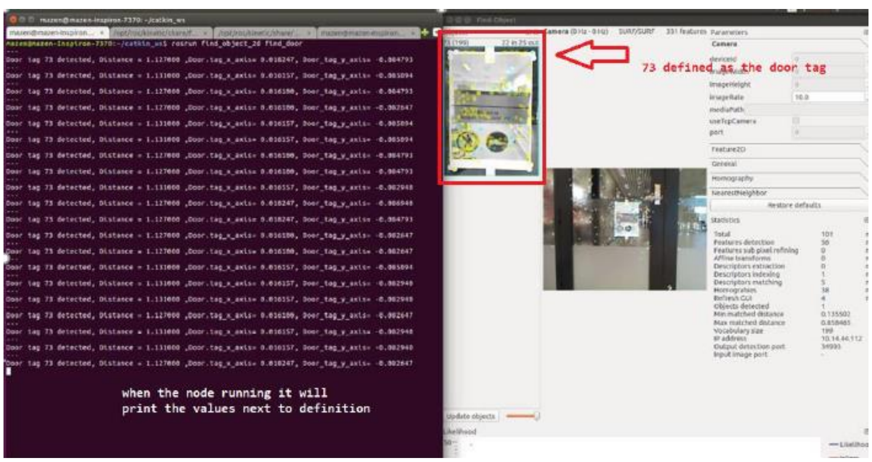
```
      case follow_object5:
      case follow_object6:
         cvHomography.at<float>(0, 0) = object-
>data[3]; // get homography matrix variable from array
         cvHomography.at<float>(1, 0) = object->data[4];
         cvHomography.at<float>(2, 0) = object->data[5];
         cvHomography.at<float>(0, 1) = object->data[6];
         cvHomography.at<float>(1, 1) = object->data[7];
         cvHomography.at<float>(2, 1) = object->data[8];
         cvHomography.at<float>(0, 2) = object->data[9];
         cvHomography.at<float>(1, 2) = object->data[10];
         cvHomography.at<float>(2, 2) = object->data[11];

         inPts.push_back(cv::Point2f(0, 0));               //Defining corners of input plane
         inPts.push_back(cv::Point2f(objectWidth, 0));
         inPts.push_back(cv::Point2f(0, objectHeight));
         inPts.push_back(cv::Point2f(objectWidth, objectHeight));
         cv::perspectiveTransform(inPts, outPts, cvHomography);      // Calculating perspect
ive transformation

         x_position  = (int)(outPts.at(0).x + outPts.at(1).x + outPts.at(2).x + outPts.at(3)
.x) / 4;  // From the corners calculate the central of the object

         ang_vel = -
(x_position - camera_center) / speed_coefficient; //  calculating angular velocity value pro
portional from the position of object
    try{
         switch (id)
         {
         case follow_object1:
         listener-
>lookupTransform("camera_rgb_optical_frame", "object_73",ros::Time(0), poseCam);
         break; // from tf listener get the position between the camera and object_73
         case follow_object2:
         listener-
>lookupTransform("camera_rgb_optical_frame", "object_74",ros::Time(0), poseCam);
         break; // from tf listener get the position between the camera and object
         case follow_object3:
         listener-
>lookupTransform("camera_rgb_optical_frame", "object_75",ros::Time(0), poseCam);
         break;  // from tf listener get the position between the camera and object
         case follow_object4:
         listener-
>lookupTransform("camera_rgb_optical_frame", "object_75",ros::Time(0), poseCam);
         break; // from tf listener get the position between the camera and object
```

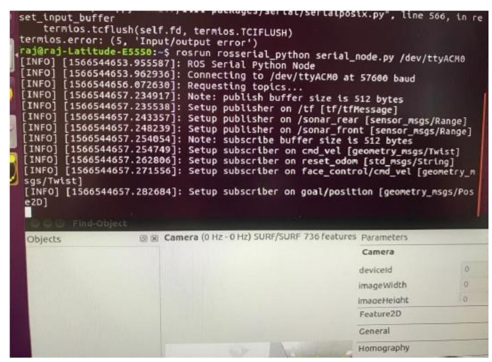
```
        case follow_object5:
        listener-
>lookupTransform("camera_rgb_optical_frame", "object_75",ros::Time(0), poseCam);
        break; // from tf listener get the position between the camera and object
        case follow_object6:
        listener-
>lookupTransform("camera_rgb_optical_frame", "object_75",ros::Time(0), poseCam);
        break; // from tf listener get the position between the camera and object
        case follow_object7:
        listener-
>lookupTransform("camera_rgb_optical_frame", "object_75",ros::Time(0), poseCam);
        break; // from tf listener get the position between the camera and object
        }}
        catch (tf::TransformException &ex) {
        ROS_ERROR("%s",ex.what());
        ros::Duration(0).sleep();
        //continue;
        }
   distance = poseCam.getOrigin().z(); //get the position between the camera and object in z
 axis which is distance between them

        if (ang_vel >= -(min_ang_velocity / 2) && ang_vel <= (min_ang_velocity / 2))
        {
           vel.angular.z = 0;
        }
        else if (ang_vel >= max_ang_velocity)
        {
         vel.angular.z = max_ang_velocity; //Turning Left the robot
       }
       else if (ang_vel <= -max_ang_velocity)
       {
          vel.angular.z = -max_ang_velocity; //Turning Righ the robot
       }
       else
       {
         vel.angular.z = ang_vel;
       }


       if (distance < min_distance && distance >= stop_dis) // if the distance less than mi
nimum distance and the distance higher than or equal stop_distance
        {
           vel.linear.x = 0; // stop the robot (set velocity linear 0)
        }
        else if (distance < stop_dis ) // if the distance higher than stop_distance
```

```
         {
            vel.linear.x = (-max_linear_vel/max_distance)*distance;
//((0.3 - min_linear_vel)/(0.2-
distance))* min_linear_vel; // move the robot backwards depend on how close it's increase li
near velocity
         }

         else if (distance >= min_distance && distance <= max_distance ) //if the distance h
igher than or equal minimum distance and the distance less than or equal maximum distance
         {
            vel.linear.x = (max_linear_vel/max_distance)*distance; //((max_linear_vel-
min_linear_vel)/(max_distance-
distance))* min_linear_vel ;// move the robot forwards depend on how far the object increase
 linear velocity
         }
         else
         {
            vel.linear.x = 0;
         }

         break;
      default: // other object
         vel.linear.x = 0.0;
         vel.angular.z = 0.0;
      }
      pub.publish(vel); //publish the value of set_velocity
   }
   else
   {
      // No object detected
      vel.linear.x = 0;
      vel.angular.z = 0;
      pub.publish(vel);
   }
}

int main(int argc, char **argv)
{
   std_msgs::String s;
   std::string str;
   str.clear();
   str.append("");
   //std::to_string(3);
   s.data = str;
   ros::init(argc, argv, "object_tracker");
```

```
    ros::NodeHandle nnh;
    listener = new tf::TransformListener; // receiving tf transformations
    ros::Subscriber sub = nnh.subscribe("/objects", 5000, objectCallback); // subscribe objec
ts topic
    ros::Rate loop_rate(500000);
    pub = nnh.advertise<geometry_msgs::Twist>("cmd_vel",50);//publish to cmd_vel as geometry_
msgs::Twist whitch will send to arduio
    vel.linear.x = 0;       // inital velocity is zero
    vel.linear.y = 0;
    vel.linear.z = 0;
    vel.angular.x = 0;
    vel.angular.y = 0;
    vel.angular.z = 0;
    while (ros::ok())
    {
        ros::spinOnce();
        loop_rate.sleep();
    }
}
```

## Appendix B:



find_door.cpp

```
#include <ros/ros.h>
#include "iostream"
#include <std_msgs/Float32MultiArray.h>
#include <geometry_msgs/Twist.h>
#include "geometry_msgs/Vector3.h"
#include <std_msgs/String.h>
```

```
#include <opencv2/opencv.hpp>
#include "math.h"
#include <tf/transform_listener.h>

#define door_tag 73

int id = 0;
ros::Publisher chatter_pub;
std_msgs::String msg;
int camera_center = 320;            // cameraWidth from the left 0 to right 640

float distance = 0 ;
float Door_tag_x_axis = 0 ;
float Door_tag_y_axis = 0 ;

tf::TransformListener *listener;//to use listener as pointer
tf::StampedTransform poseCam; // position of the object based on camera position

void objectCallback(const std_msgs::Float32MultiArrayPtr &object) // callback function getti
ng messages and the data of recognized objects
{
   printf("---\n");
   if (object->data.size() > 0)
   {
      id = object->data[0]; // get object ID from array

      // Find the corners OpenCV
      cv::Mat cvHomography(3, 3, CV_32F);
      std::vector<cv::Point2f> inPts, outPts;
      switch (id)
      {

      case door_tag:  // if get Turn_Left sign

    try{
         listener-
>lookupTransform("camera_rgb_optical_frame", "object_73",ros::Time(0), poseCam);   // from t
f listener get the position btween the camera and object
         }
         catch (tf::TransformException &ex) {
         ROS_ERROR("%s",ex.what());
         ros::Duration(1.0).sleep();
         //continue;
         }
```

```
    distance = poseCam.getOrigin().z(); //get the position between the camera and object in
z axis which is distance between them
    Door_tag_x_axis= poseCam.getOrigin().x(); //get the position between the camera and obje
ct in x axis which is distance between them
    Door_tag_y_axis= poseCam.getOrigin().y();  // //get the position between the camera and
object in y axis which is distance between them
                  printf("Door tag %d detected, Distance = %f ,Door.tag_x_axis= %f, Door_tag
_y_axis= %f  \n",
               id, distance ,Door_tag_x_axis,Door_tag_y_axis); // print the values next to d
efinition
         break;

      case door_right: // if get Turn_Right sign
    try{
         listener-
>lookupTransform("camera_rgb_optical_frame", "object_73",ros::Time(0), poseCam);   // from t
f listener get the position btween the camera and object
         }
         catch (tf::TransformException &ex) {
         ROS_ERROR("%s",ex.what());
         ros::Duration(1.0).sleep();
         //continue;
         }
    distance = poseCam.getOrigin().z(); //get the position between the camera and object in
z axis which is distance between them
         break;
       case door_left:

    try{
         listener->lookupTransform("camera_rgb_optical_frame", "object_73",
                               ros::Time(0), poseCam);   // from tf listener get the positio
n btween the camera and object
         }
         catch (tf::TransformException &ex) {
         ROS_ERROR("%s",ex.what());
         ros::Duration(1.0).sleep();
         //continue;
         }
   distance = poseCam.getOrigin().z(); //get the position between the camera and object in z
 axis which is distance between them

         chatter_pub.publish(msg); //publish the value of set_velocity
         break;
      default: // other object
 printf("No Door detected.\n");
```

```
      }}

   else
   {
      // No object detected
    printf("No Door detected.\n");
      chatter_pub.publish(msg);
   }
}

int main(int argc, char **argv)
{
   std_msgs::String s;
   std::string str;
   str.clear();
   str.append("");
   //std::to_string(3);
   s.data = str;
   ros::init(argc, argv, "find_door");
   ros::NodeHandle nnh;
   listener = new tf::TransformListener; // receiving tf transformations
   ros::Subscriber sub = nnh.subscribe("/objects", 1, objectCallback); // subscribe objects
topic
   ros::Rate loop_rate(10);
   chatter_pub = nnh.advertise<std_msgs::String>("chatter", 1000);

   while (ros::ok())
   {
      ros::spinOnce();
      loop_rate.sleep();
   }
}
```