Follow Me: A Personal Robotic Companion System for the Elderly

Rong Wang, Huiguo Zhang, Cyril Leung

Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, Nanyang Technological University, Singapore

Wang.rong@ntu.edu.sg, hgzhang@ntu.edu.sg, CLeung@ntu.edu.sg

Abstract

Aging population and empty-nest problem have become a significant issue worldwide. In order to overcome loneliness and promote an active living for the elderly, an elderly companion robot is a great solution in the future. The robot using Kinect sensor is able to intelligently detect correct user and to follow automatically and robustly. To complement the automatic mode, the robot car can also be controlled by voice commands. For better interaction and smart companion, the robot is able to talk and give suggestions by using text-to-speech algorithm. It is also designed for both outdoor and indoor environment and to assist elderly to carry miscellaneous belongings.

Keyword: Elderly Companion, Robot, Kinect, Aging, Follow me, Voice Recognition, speech generation.

I. Introduction

With the improvement of healthcare, aging is becoming a significant issue. The number of older persons aged 60 years or above was 841 million in 2013 all over the world, which was four times higher than the 202 million that lived in 1950. The older population will almost triple by 2050, when it is expected to surpass the two billion mark.

The elderly are usually physically and emotionally weak. What's worse, there are a high volume of empty-nest elderly. Loneliness is a potential killer that leads to mental disease. To this concern, technology is a good solution to complement this problem. An intelligent robot is thus developed to company elderly, interact and assist them in daily life.

Based on research, currently there are very few robots are specially designed for elderly and they easily cost US$10,000 and above which is unaffordable to elderly. To this concern, a toy car of approximately 1 metre long costing less than US$200 was selected as the platform. It is adaptive to both indoor and outdoor environment. Some papers highlight the advantage of Kinect when tracking the user [1]. In this project, Kinect sensor is selected because it integrates both depth and voice sensor [2-4]. And its IR camera which simulates the user skeleton helps with fast tracking [5-6]. Based on the Kinect microphone and SDK, voice can be collected and processed [7-8]. Lastly, for good human-robot interaction [9], the robot not only needs to understand elderly's commands but also needs to speak out and interact with elderly [10]. This robot system uses Microsoft Text-to-Speech API to generate speech to alert in certain danger situations.

The paper is organized as follows: Section II gives details of hardware components of the robot system and how these components are implemented. Section III focuses on software development. It gives details on how the Kinect tracks users and robot follows accordingly. It also elaborates the working principle of voice recognition and text-to-speech application. In Section IV, it suggests the future possible directions and improvement. Lastly, Section V leads to conclusion.

II. Hardware Implementation

1. Hardware Components

A. Toy Car –Body and Limbs

In Figure 1, a toy car of 1.5m long, 0.8m wide and approximately 0.7 high was selected as robot platform. It functions as body of the system and its four big wheels functions as limbs.

Figure1: Robot car platform

B. Motor Driver – Muscle

One front motor controling left and right direction is powered by front motor driver and two back motors controling forwards / backwards are powered by another back motor driver. In this project, two back motors are electrically parallel connected. Thus the two wheels move in the same direction with same speed all the time. In order for capability of carrying weitht, motor drivers with maximum loading of 10A is selected. It enables the robot to move swiftly without overheating functioning as muscles of the system.

C. Kinect – Eyes and Ears

A first-generation Kinect is used as eyes and ears of the system. The depth sensor integrated in the Kinect is able to get the distance of the target user. The Kinect IR carema is used to catch the skeleton of the user. The microphone array is able to collect sound input from surrounding.

D. Controller and Converter – Brain

A portable laptop of Windows system is used as the controller of the system. In order to transmit commands from the laptop to the robot system, and to read feedback information from the robot system, a small portable USB converter shown in Figure 2 is connected to the laptop. It is able to generate PWM signal, read/ write input and output signals.

Figure 2: Portable USB converter

E. Encoder – Nerve System

In order to control the system accurately, a close loop with feedback information is required. An encoder is used to record the left / right turning angle of the robot. This non-contacting magnetic encoder provides high accuracy.

Figure 3: Encoder to record turning angle

2. Hardware Connection

Hardware connection is as shown in Figure 4. Firstly, Kinect is mounted on top of the robot car and Kinect sends data inputs to the laptop which sits on the robot car. Secondly, controller processes data inputs and feedbacks. Thirdly, controller sends commands to USB converter. Fourthly, USB converter sends instructions to motor driver 1 to generate front/back motion. At the same time, it sends instructions to motor driver 2 to generate left/right motion. Lastly, an encoder is mounted on the shaft of front wheels to generate feedback information and give back to controller to be processed further.

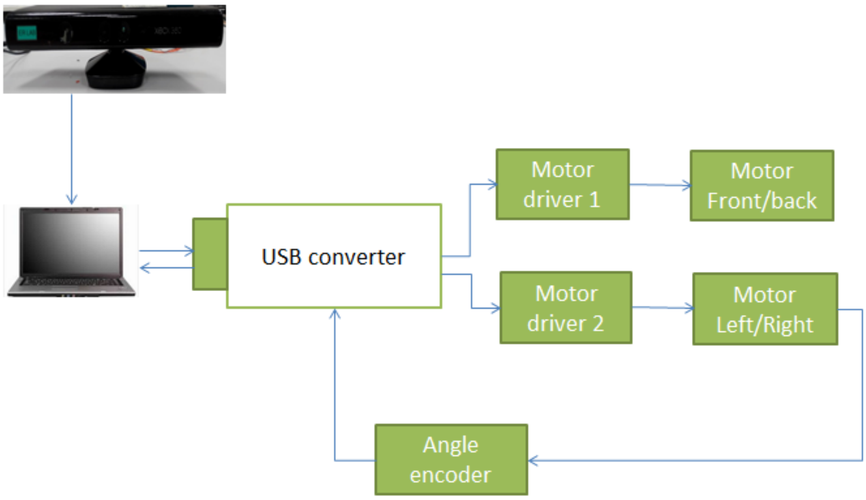

Figure 4: Flow diagram of system connection

III. Software Implementation

The project is written in Visual Studio C++ program language.

System can be divided into four functions, namely target detection function, user following function, voice control function and speech generation function.

1. Target Detection Function

1.1 Detection Algorithm

The target detection function is developed based on "Kinect for Windows Developer Toolkit v1.8"—Kinect Explorer – D2D.

A flow chart is as illustrated in Figure 5. When the program runs, it searches middle person. Once it finds the target user confidently, assign ID to the first user. It will get the user's position and follow accordingly. If more than one user existing, it will assign ID accordingly (up to 6 users). In this project, it will track and get data of the first user. Only if the first user is lost, it will track another user. It makes sure that the robot car will follow the only user. Any strangers or distraction will not take effect.

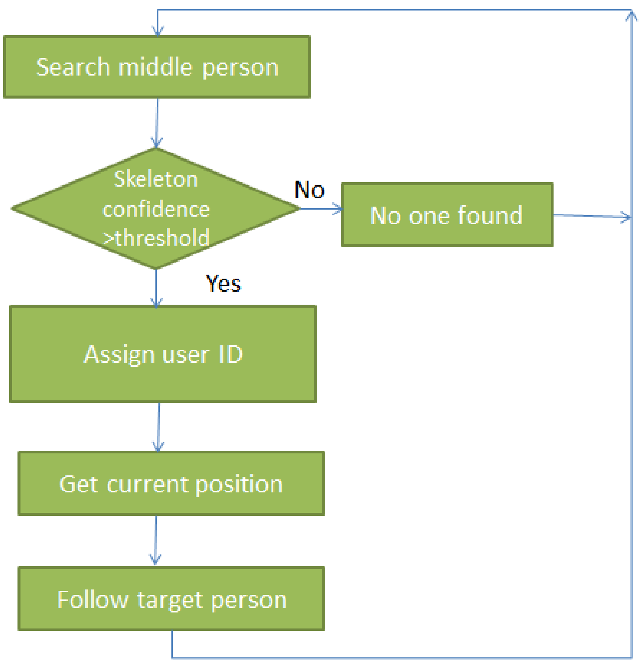

Figure 5: A flow chart to illustrate the process of target detection.

1.2 Improve Detection Accuracy [11]

1.2.1 Skeleton Detection Confidence

As mentioned previously, when the program finds a targeted person, it is not sufficient to tell if it is a human. According to experiements, false detections like mistreating a door, wall or a chair as a human. In Figure 6, in case like target person is too near, and only part of the body is detected, skeleton pattern is very disordered. In this situation, it is not confident to tell the exact position of the user. Then program should not execute.

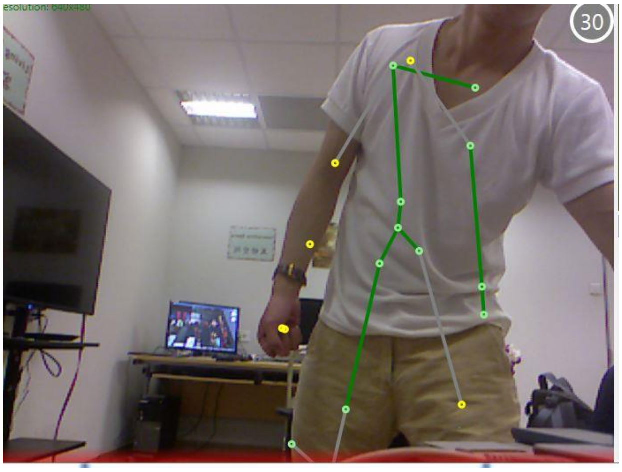

Figure 6: When user is too near, skeletion disordered.

In order to improve the accuracy, skeleton detection confidence is introduced.

There are in total 20 joints for a user to form 1 skeleton pattern. They are listed in the table1 and Figure 7.

Joint 1	Hip center
Joint 2	Spine
Joint 3	Shoulder center
Joint 4	Head
Joint 5	Shoulder left
Joint 6	Elbow left
Joint 7	Wrist left
Joint 8	Hand left
Joint 9	Shoulder right
Joint 10	Elbow right
Joint 11	Wrist right
Joint 12	Hand right
Joint 13	Hip left
Joint 14	Knee left
Joint 15	Ankle left

Joint 16	Foot left
Joint 17	Hip right
Joint 18	Knee right
Joint 19	Ankle right
Joint 20	Foot right

Table 1: Skeleton joints representation

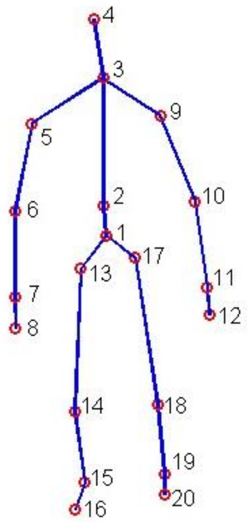

Figure 7: Skeleton joints representation

For each joint, there are 3 statuses to illustrate the confidence level.

Tracked – it is confident to tell the joint is found

Inferred – it is not confident to tell the joint is found (may or may not exist)

Not tracked – the joint is not found

Count the number of tracked joints to represent the confident level. The higher number, the more confident it is. There is another condition that the algorithm can be further improved. In this project, as when the user is close to the Kinect, joints of lower body is usually unable to be captured, which causes the confident level low. But in this case, the user does exist and program should execute. Thus count the number of upper body joints will be more accurate. Joints 14- 20 of the whole body will be omitted.

1.2.2 Noise Rejection

As there is always noise and disturbance when detecting skeleton, the following example occurs frequently.

Number of confident upper body joints
14, 14, 14, 14,14,14,14,14,7,14,14,14,14

As can be seen, when a sudden noise as circled occurs, it will tell the program that the user is lost. The robot companion stops immediately and speeds up again right after. This will cause an unsmooth action and do harm to motors.

To solve this problem, the author developed an algorithm to get rid of this noise.

1) Take 22 data sample (data in roughly 1 second).
2) Set threshold (below threshold meaning user is lost)
3) Count the number of data below threshold
4) Only consider user is lost if the number of data below threshold appears more than **4 times** in 22 data sample.
5) Otherwise, consider the user is still tracked

Number of confident upper body joints
14, 14, 5, 14,14,8,14,14,7,14,14,14,14,14,10,14,14,14,11 14,14,14

In this case above, set threshold as 12. There are more than 4 times lost detections within one sample range. It is considered as user lost.

Number of confident upper body joints
14, 14, 5 14,14,8,14,14,7,14,14,14,14,14,14,14,14,14,14,14,14,14

In this case above, set threshold as 12. There are less than 4 times lost detections within one sample range. It is considered as user in track.

1.2.3 False Detection Rejection

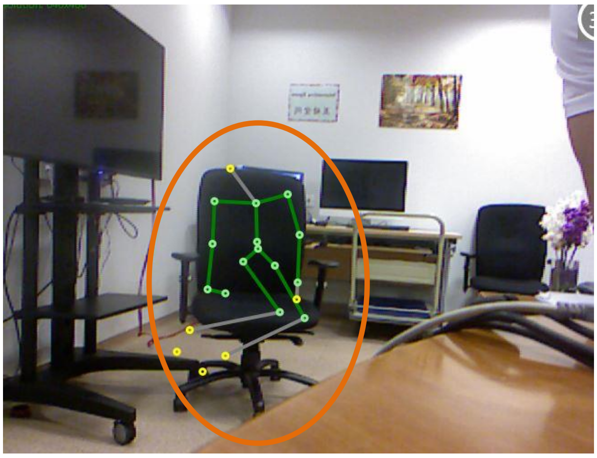

Figure 8: Kinect mistreats a chair as a user

In the case shown on Figure 8, the Kinect mistreats a chair as a user. And its skeleton confidence level is still quite high. Therefore algorithm discussed above is not sufficient to reject this false detection.

A new algorithm is developed to reject false detection. Since within a very short period of time, the user hardly moves body too much, the joint position in such short period is changing very little. According to experiment data, when false detection happens, for example Kinect mistreats a chair as a user; the joint position is jumping unreasonably.

Therefore, variance of position is a good parameter to judge the joint moving fluctuation.

$$Sn^2 = \frac{1}{N}\sum_{i=1}^{N}[(xi - xave)^2 + (yi - yave)^2 + (zi - zave)^2]$$

N: data sample

x_{ave}: x-axis data average

y_{ave}: y-xis data average

z_{ave}: z-xis data average

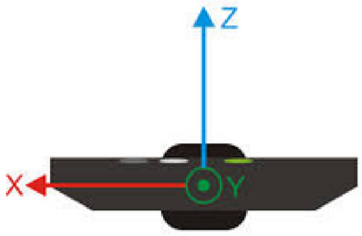

Figure 9: Axis of position based on Kinect

The axis if the joint is set based on Kinect as shown in Figure 9. Take the average of x, y, z axis respectively and compute the variance according to the formula above.

If the variance is above threshold, it means big data violation. It is considered as false detection. In this case, the robot should not follow. By doing so, it is able to reject situations like mistreating some obstacles as a human.

2. User Following Function

2.1 Operation Mode Selection

There are two modes to control the robot as shown in Figure 10. If target person is found, it will go to auto control mode. The robot will follow the elderly tightly. If the target person is not found or lose contact, it will go to voice control mode. The elderly could use voice command to call the robot back.

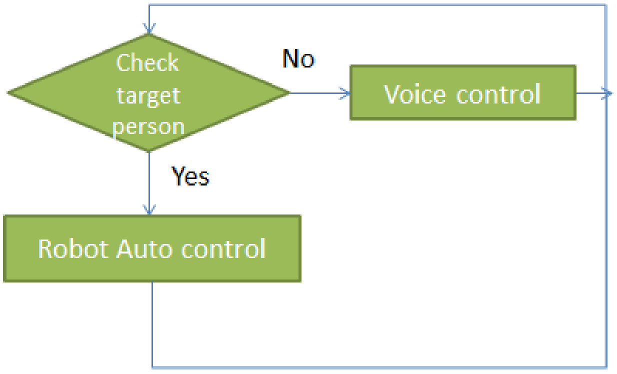

Figure 10: Flow chart of mode selection

2.2 Auto Control Mode

In auto control mode, the control action is based on the position of the user. Distance data is able to be read from the Kinect depth sensor. The further the user from the user, the faster the robot catches up.

Front / back speed calculation

If distance away from Kinect less than minimum margin, it will stop.

If distance away from Kinect greater than maximum margin, it will run at full speed.

If distance within minimum and maximum margin, it will adjust its speed linearly according to distance.

In this project, it is divided into a scale of 10 for simplicity. Thus the speed varies from 0-100%. For example, 1.5-1.8m speed is 20% of full speed. A plot of speed with reference to distance is shown in Figure 11.

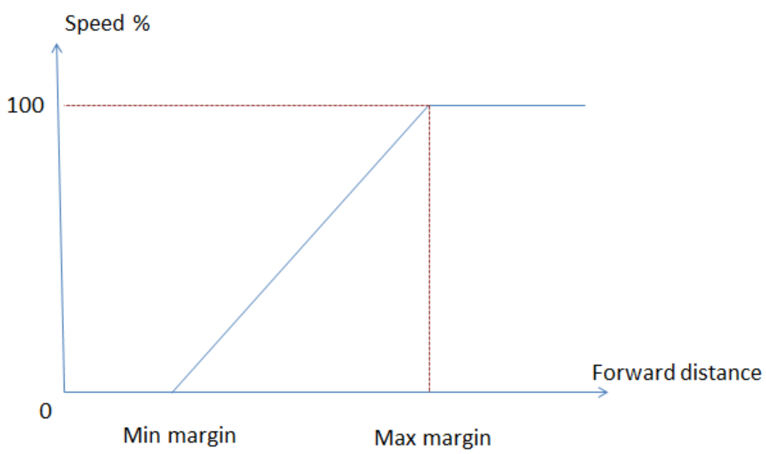

Figure 11: Speed with reference to distance plot

Left / Right direction and speed determination

In left / right direction, in order to keep the robot car and user in line, PID control logic is implemented here. In this project, the angle difference between direction of the user and direction of the robot car does not require extremely precise. If too precise, the robot is keeping turning all the time, which is not as expected. Only when the angle difference is big enough, then the robot adjusts its direction. Therefore, proportional control (P) is adopted because it is fastest.

As shown in Figure 12, when output angle read from encoder differs from required angle, the angle difference will be processed by the controller and a proportional gain will be generated to compensate the difference.

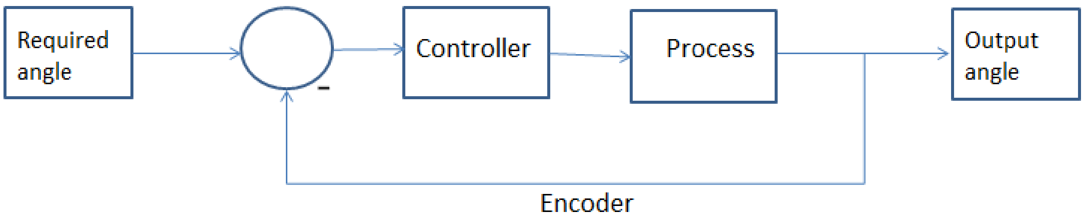

Figure 12: Feedback loop of left / right turning angle

1) Calculate user angle measured by Kinect shown in Figure 13.

User angle Ө = arc tan (X distance / Z distance);

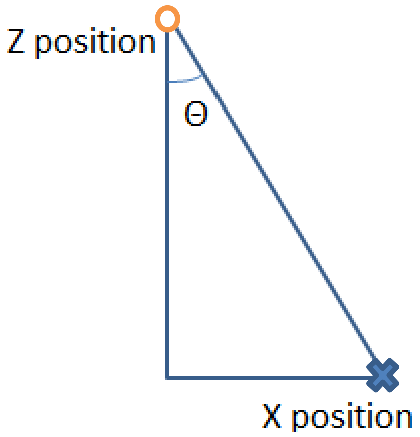

Figure 13: Determine user angle Ө

2) Measure robot car turning angle recorded by encoder

3) Determine turning direction according to the two angles.

Case 1: User is on the right hand side of robot car (Figure 14). It turns right.

Case 2: User is on the left hand side of robot car (Figure 14). It turns left.

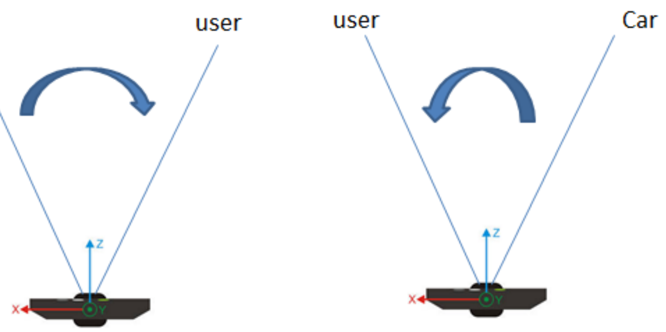

Figure 14: Case 1 Case 2

Case 3: Angle between robot car and user is less than 5 degrees (margin), it remains as previous state (Figure 15).

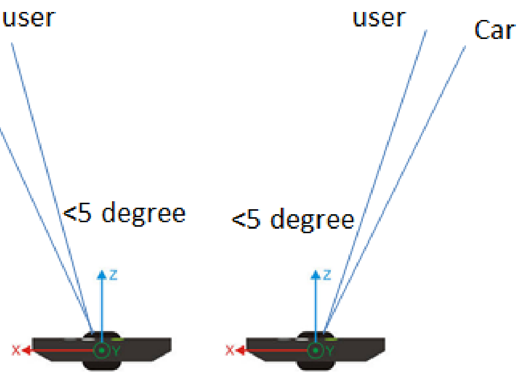

Figure 15: Case 3

Case 4: If the robot companion is moving towards right, but user has a motion towards left, then it stops moving right and wait for next data input (Figure 16).

Case 5: If the robot companion is moving towards right, but user has a motion towards left, then it stops moving right and wait for next data input (Figure 16).

Both case 4 and case 5 are used as prediction to prevent over turn and follow tightly.

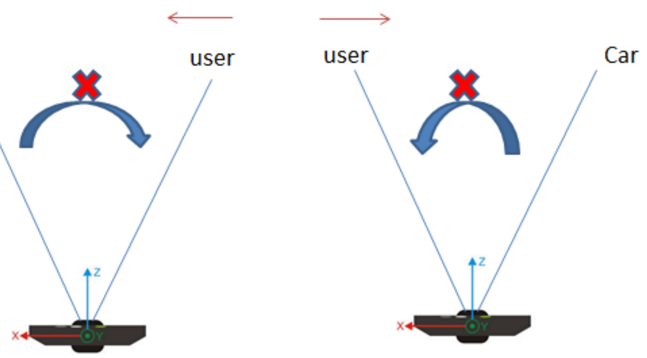

Figure 16: Case 4 Case 5

4) Determine the speed magnitude

To follow tightly and prevent over turn, a speed calculation algorithm is developed to estimate the left right moving speed (magnitude and direction) using first derivative.

$$\text{Speed} = \frac{\text{current position} - \text{previous position}}{\text{time}}$$

The magnitude of speed is divided into a scale of 4 for simplicity. 4 means full speed and 1 means low speed.

3. Voice Control

This program is based on Kinect SDK 1.8v, “speech basic”.

3.1 Voice Commands

When a target person is not found and in extreme case when the lighting is too strong, Kinect is unable to detect the user and it goes to voice control mode as shown in Figure 17. In voice control mode, the robot companion will stop and wait for voice command. After capturing the correct speech, it will do the action accordingly.

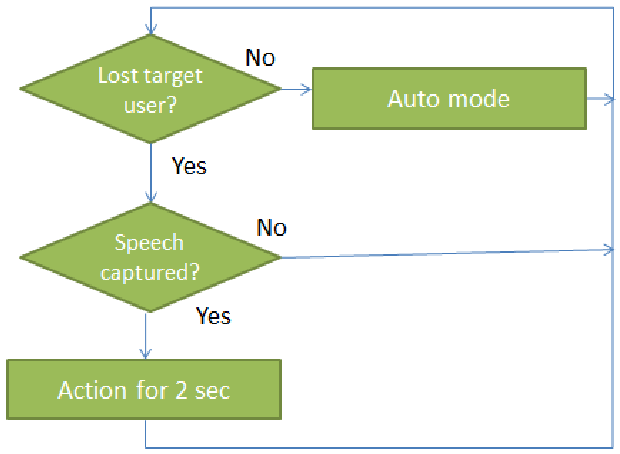

Figure 17: A flowchart showing when speech recognition mode is on

In this project, four Basic Commands are created for voice control:

Forward --- move forward for 2 seconds with low speed

Back --- move backward for 2 seconds with low speed

Right --- move forward and right for 2 seconds with low speed

Left --- move forward and left for 2 seconds with low speed

3.2 Improve Command Recognition

For higher accuracy purpose, a grammar file is created in order to catch more than one keyword of commands, such that it has higher capability of understanding.

Any of the keywords “Forward”, “Straight” or “Forwards” will lead to execute “Forward” command.

Any of the keywords “Backwards”, “Backward” or “Back” will lead to execute “Back” command.

Any of the keywords “Left”, or “Turn Left” will lead to execute “Left” command.

Any of the keywords “Right”, “Turn Right” or “Back” will lead to execute “Right” command.

3.3 Improve Accent Recognition

As different user has different accent, Asian users’ accent unfortunately leads to low recognition confidence. Therefore training needs to be down to improve accuracy.

Under Windows control panel, there is an option called “Speech Recognition - Train your computer to better understand you”**.** You will be asked to read the text from 15-30 phrases and it will take approximately 5 minutes to complete. More training can be done for the system to understand your accent better.

4. Speech Generation

Speech generation function enables the robot companion to communicate with the elderly. It can simulate human’s voice and speak out through laptop speaker.

The technology called TTS (Text-to-Speech) is adopted. As Windows system is used, a Windows embedded speech TTS SDK is used.

This SDK enables user to select voice (male or female), pitch (low or high) according to preference.

There are two conditions in this project that will actived speech generation function. First condition is when target user is lost. A speech of “I can’t find you” will be generated. The second condition is when speed exceeds a recommended speed. A speech of “You are too fast” will be generated.

In the cases where there is false detection or a correction has been done by user, too many voice reminders might be redundant. Therefore an algorithm is implemented: only when conditions happen and last for at least 2 seconds, a voice reminder is activated. If the condition remains after first reminder, the same reminder will be activated again after a period of time (i.e. 8 seconds) according to user’s preference.

IV. Future Development

Future Hardware Development

Due to the hardware limitation, the front wheels could only be controlled to turn left and right, while the back wheels can be driven forwards and backwards. This results in the robot companion to turn with a very big radius. It is unable to pivot on origin. In the case of narrow situation, it is hard to perform ultimately without touching obstacles.

To overcome these limitations, it is planned to implement the whole system to a motorized wheelchair [12-13] in the future, which is very robust and elderly friendly. It is able to turn with small radius and the elderly may even sit on the wheelchair when tired.

In addition, a 2-generation Kinect will be adopted. It will provide higher accuracy and get rid of some lighting problems.

Future Software Development

In terms of software development, apart from existing functions, a lot more function will be added. “Elderly Agent” is an intelligent approach. It will remember elderly’s daily routine, and thus remind those with memory deterioration. It will also to able to have conservation with elderly just like talking to a friend. In addition, it will be connected to server, once the system detects elderly’s abnormal actions or strange behaviours like falling down, it will send message to guardians through server.

Lastly, a user recognition function will be developed to recognize the owner. It makes the robot more loyal to owner and more personalized.

V. Conclusion

In this paper, an elderly companion robot is built based on Kinect sensor. With effort of Kinect detection accuracy improvement, the robot system is able to intelligently follow the user. Voice control function provides another option for user to control the robot apart from auto mode when Kinect loses target user. A big room can be improved for voice recognition to make it more robust and understandable. In addition, speech generation function makes the robot system more entertaining. It sets a solid foundation for future human-robot interaction development. It can foresee that elderly-friendly intelligent robot will play a big role in the background of fast growing aging population.

VI. Acknowledgement

This research is supported by the National Research Foundation, Prime Minister’s Office, Singapore under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office.

References

[1] J. L. Azevedo, C. Cruz, J. Cunha, M. B. Cunha, N. Lau, C. Martins, A. J. R. Neves, E. Pedrosa, A. Pereira, A. J. S. Teixeira, M. Antunes, “CAMBADA@Home 2011 Team Description Paper,” in Transverse Activity on Intelligent Robotics IEETA/DETI – University of Aveiro 3810-193 Aveiro, Portugal.

[2] Weibo Song , Xianjiu Guo, Fengjiao Jiang, Song Yang , Guoxing Jiang, Yunfeng Shi, “Teleoperation Humanoid Robot Control System Based on Kinect Sensor,” Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on, vol. 2, 2012, p. 264-267.

[3] Sang-Hyuk Lee1 and Seung-Hyun Oh, “A Kinect Sensor based Windows Control Interface,” in International Journal of Control and Automation, vol. 7, no. 3, 2014, pp. 113–124.

[4] Machida. E, Meifen Cao ; Murao, T. ,Hashimoto, H, “Human motion tracking of mobile robot with Kinect 3D sensor” in SICE Annual Conference (SICE), 2012 Proceedings of, 2012, pp. 2207 - 2211.

[5] Abhishek Kar, “Skeletal Tracking using Microsoft Kinect,” Department of Computer Science and Engineering, IIT Kanpur, 2010.

[6] Livingston, M.A, Sebastian, J, Zhuming Ai, Decker, J.W, “Performance measurements for the Microsoft Kinect skeleton,” in Virtual Reality Short Papers and Posters (VRW), 2012 IEEE, 2012, pp. 1087-8270.

[7] J. Norberto Pires, “Robot-by-voice: experiments on commanding an industrial robot using the human voice,” in Industrial Robot: An International Journal, vol. 32, issue 6, 1973.

[8] Christopher Kawatsu, Jiaxing Li, C.J. Chung, “Development of a Fall Detection System with Microsoft Kinect,” in International Journal of Advanced Robotic Systems, vol. 208 ,2013, pp. 626-630.

[9] Liying Cheng, Qi Sun , Han Su, Yang Cong , “Design and implementation of human-robot interactive demonstration system based on Kinect,” in Control and Decision Conference (CCDC), 2012 24th Chinese, 2012, pp. 971-975.

[10] Xuedong Huang, Acero. A, Adcock. J, Hsiao-Wuen Hon, Goldsmith. J, Jingsong Liu, Plumpe. M, “Whistler: a trainable text-to-speech system,” in Spoken Language, 1996. ICSLP 96. Proceedings, Fourth International Conference on, vol. 4, 1996, pp. 2387 - 2390.

[11] Nguyen, C.V, Izadi, S, Lovell, D, “Modeling Kinect Sensor Noise for Improved 3D Reconstruction and Tracking,” in 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on, 2012, pp. 524 - 530.

[12] Mark G. Greenbaum, MD, Sylvia Fernandes, MD, and Stanley F. Wainapel, MD “Use of a motorized wheelchair in conjunction with a guide dog for the legally blind and physically disabled” in Archives of Physical Medicine and Rehabilitation, vol. 79, issue 2, 1998, pp. 216-217.

[13] Alan L. Breed and Irene Ibler, “The Motorized Wheelchair: New Freedom, New Responsibility and New Problems” in Developmental Medicine & Child Neurology, vol. 24, issue 4, 1982, pp 366-371.