The Digital Scanning Collision Avoidance Device with Risk Assessment

Malik Haddad[1], Jamal Shamieh[2], David Sanders[3], Amir Gharavi[4], Martin Langner[5], Giles Tewkesbury[3] and Mohamed Hassan-Sayed[6]

[1] Northeastern University London, London United Kingdom
[2] America University of Madaba, Madaba, Jordan
[3] University of Portsmouth, Portsmouth, United Kingdom
[4] University College London (UCL), London, United Kingdom
[5] Chailey Heritage Foundation, North Chailey, United Kingdom
[6] University of Southampton, Southampton, United Kingdom
`malik.haddad@nulondon.ac.uk`

Abstract. This paper presents a new approach to digitising the Scanning Collision Avoidance Device (SCAD) which is used to detect obstacles in front of a powered wheelchair. The new approach replaced the SCAD electronic circuit with a Raspberry Pi. Python programming language is used to create a program to analyse readings from the SCAD ultrasonic transducer. The program is installed on the Raspberry Pi. An op–amp isolation circuit was inserted between SCAD and Raspberry Pi. The Raspberry Pi analysed readings received from the SCAD ultrasonic transducer. The program used the readings to identify distances and locations of obstacles in the surroundings. User input switches were connected to the Raspberry Pi and converted inputs from switches into commands controlling the wheelchair motors. The program incorporated two driving modes: Stop and Avoid. A User Interface was created to allow users to select the desired driving mode and operate the new system. Stop mode will stop the user from driving in the direction of the detected obstacle. Avoid mode will avoid the obstacles in the wheelchair surroundings by driving in the opposite direction. The Python program captured the users' desired direction and intelligently translated these commands to a safe driving direction that avoided obstacles in the wheelchair surroundings. If the user did not select a driving mode, the program would disable the wheelchair motors and a message will appear asking the user to select a driving mode before driving. This provided a safety feature that prevented the system from being accidentally operated.

Keywords: Collision Avoidance, Disabled, Risk Assessment, Python, Steer, Wheelchair.

1 Introduction

The work presented in this paper is part of a broader research conducted by the authors [1] aiming to improve mobility and enhance the quality of life of disabled powered wheelchair users by increasing their self-confidence and self-reliance.

Langner [2] created a Scanning Collision Avoidance Device (SCAD) that used a single rotating ultrasonic transducer to detect obstacles in the wheelchair surroundings. The SCAD had two parts: a SCAD head and a Control box as shown in Figure 1. The SCAD head had a single ultrasonic transducer mounted on a stepper motor and an electronic circuit used to operate the stepper motor. The Control box had an electronic circuit used to operate the SCAD. The work presented in this paper aims to insert an electronic circuit and a microcomputer to digitise the SCAD.

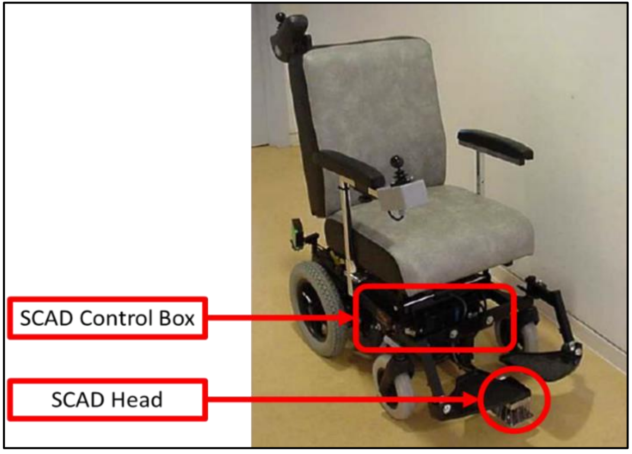


Fig. 1. SCAD mounted onto a powered wheelchair [2].

The idea of the SCAD was to use a single rotating ultrasonic transducer sending ultrasonic pulses through stepped periods. The distance from a detected obstacle was determined by measuring the time of flight required by a pulse to be sent and reflected to the receiver [2].

An ultrasonic transducer was mounted on a 15º per step stepper motor. An optical method was used to identify the start and end points of the scan. The stepper motor rotated 180º forward sweep and then returned to the start position. SCAD scanned the 180º area in front of the wheelchair on the forward sweep and then again on the backward sweep. The area in front of the wheelchair was divided into six sectors: Extreme Left, Mid Left, Front Left, Front Right, Mid Right and Extreme Right. The area scanned by the SCAD and sector division is shown in Figure 2.

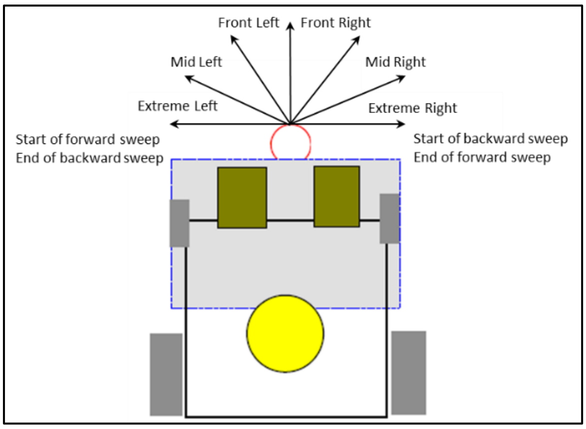


Fig. 2. The area scanned by the SCAD and sector division [2].

The motor was stepped to the middle of the sector and the transducer sent a 50 KHz pulse chirp, then a blanking pulse was sent to clear the transmission. Then, the transducer switched to listening mode and started listening to the echo of the transmitted pulse. If an object was present in a sector, an echo pulse would reflect to the transducer before stepping to the next sector. This cycle was repeated six times for a complete forward sweep cycle and again for a complete backward sweep cycle [2]. When an echo was received, the distance and location of the obstacle were identified.

Using a stepper motor to rotate the transducer proved that scanning was an efficient and compact alternative to a multiple-transducer array [2].

2 The New Approach

A new approach to digitising the SCAD is presented. An electronic circuit was created and inserted between the SCAD head and the SCAD Control box. The electronic circuit used a voltage divider circuit, an op-amp isolation circuit and a Raspberry Pi. Python programming language was used to create a program that analysed readings from the SCAD ultrasonic transducer. Readings were used to identify the distances and locations of obstacles in the wheelchair surroundings. Figure 3. shows a prototype of the new approach installed onto a powered platform.

Fig. 3. A prototype of the new approach installed onto a powered wheelchair.

User input switches were connected to the Raspberry Pi. User input switches are shown in Figure 4. The Red switch was used to drive the wheelchair backwards, the Yellow switch was used to drive the wheelchair to the left and the Blue switch was used to drive the wheelchair to the right. User-desired direction was acquired using a system similar to the intelligent systems in [3,4].

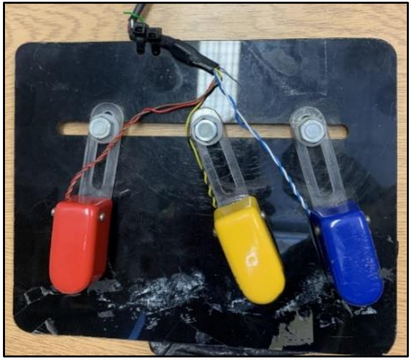


Fig. 4. User switches used to control a powered wheelchair.

The program intelligently mixed the users' desired direction and the readings from the ultrasonic transducer to provide a collision-free route for the wheelchair. A simple flow chart showing the operation of the new system is shown in Figure 5.

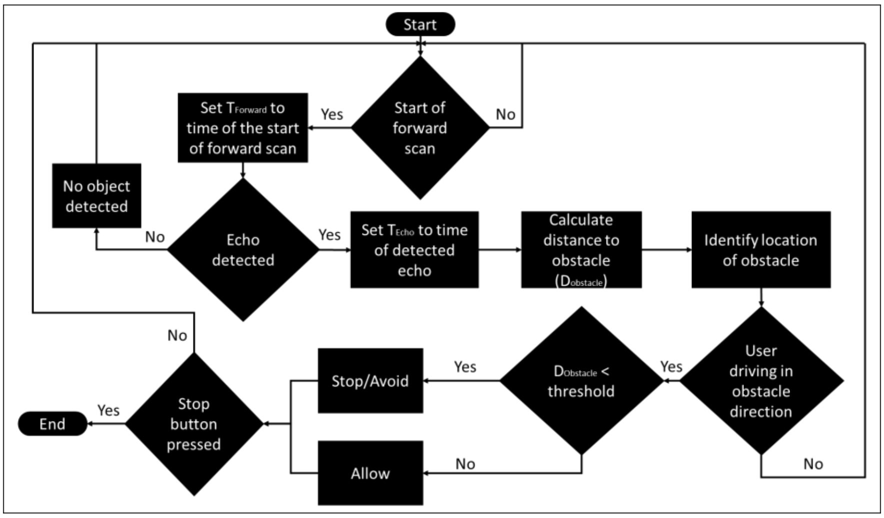


Fig. 5. A simple flow chart showing the operation of the new approach.

The SCAD operated with a 5 Volts power supply, however, that voltage would have damaged the inputs of the Raspberry Pi. A high-impedance voltage divider was used to reduce the 5 Volts used in the SCAD to 3.3 Volts compatible with the Raspberry Pi input voltage. An Op-Amp isolation circuit was used to provide isolation between the SCAD electronic circuit and the Raspberry Pi. Three inputs were used from the SCAD:

Start Scan: This input identified the start of the sweep cycle of the stepper motor. A falling edge on this input identified a forward sweep and a rising edge identified a backward sweep.

Blanking: This pin identified post-transmitted ringing pulse damping. A falling edge of this pin identified the end of transmission.

Echo: This pin was used to receive echo reflected from detected obstacles. A rising edge on this pin identified that an echo was received.

Figure 6. shows the three inputs from the SCAD, voltage divider and the Op-Amp isolation circuits, and outputs. The outputs were used as inputs to the Raspberry Pi.

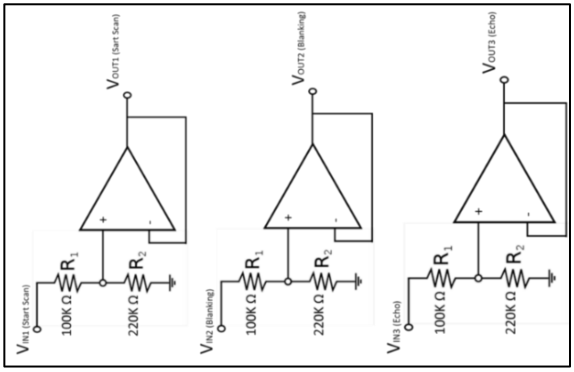


Fig. 6. A schematic diagram showing the inputs from the SCAD, the voltage divider and Op-Amp isolation circuits and the outputs [5].

During the backward sweep ghost echoes were received. These ghost echoes were from pulses transmitted during the previous forward sweep. The new approach considered the forward sweep to detect obstacles in the wheelchair surroundings. That eliminated the problem caused by ghost echoes received from previous transmissions. Inputs to the Raspberry Pi were used to identify the start of the forward sweep cycle, end of transmission, blanking pulse, and if an echo was received. The Python program identified the start time of the forward sweep cycle (T_{Forward}), the end of transmission time (T_X) and the time if an echo was received (T_{Echo}).

Time of Flight (ToF) represented the time needed for the transmission pulse to be sent and reflected from an obstacle. The program calculated ToF using Equation 1. Time to an obstacle (T_{Obstacle}) was calculated using Equation 2. ToF and T_{Obstacle} were in milliseconds (msec.).

$$ToF=T_{Echo}-T_X \tag{1}$$

$$T_{Obstacle}=0.5*ToF \tag{2}$$

The program calculated the distance to an obstacle using Equation 3 which assumed the speed of sound is 343 m/s. The distance to an obstacle was calculated in centimetres (cm).

$$D = [34300*(T_{Obstacle}]/1000 \tag{3}$$

If an echo was received, the program calculated the duration required from the start of the sweep till the time the echo was received using Equation 4.

$$T_{Forward\ Duration} = T_{Echo}-T_{Forward} \tag{4}$$

The program used $T_{\text{Forward Duration}}$ to identify the corresponding sector in which the obstacle was located. Table 1 shows the $T_{\text{Forward Duration}}$ and the corresponding sectors.

Table 1. $T_{\text{Forward Duration}}$ in milliseconds (msec).and corresponding sector.

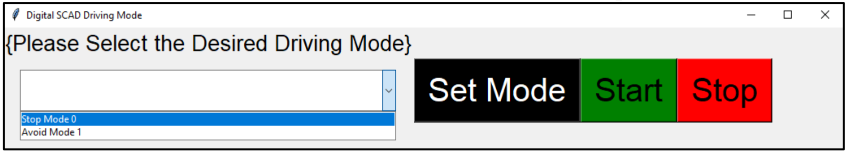
$T_{\text{Forward Duration}}$ (msec.)	Sector
$1 < T_{\text{Forward Duration}} < 20$	Extreme Right
$21 < T_{\text{Forward Duration}} < 40$	Mid Right
$41 < T_{\text{Forward Duration}} < 60$	Front Right
$61 < T_{\text{Forward Duration}} < 80$	Front Left
$81 < T_{\text{Forward Duration}} < 100$	Mid Left
$101 < T_{\text{Forward Duration}} < 120$	Extreme Left

The program incorporated two driving modes:

The Stop mode: This mode would detect obstacles in the wheelchair's surroundings and prevent the wheelchair from moving in the direction of an obstacle if the obstacle was closer than a predefined threshold. The predefined threshold could be digitally adjusted in the Python program.

The Avoid mode: This mode would detect obstacles in the wheelchair's surroundings and prevent the wheelchair from driving in the direction of an obstacle if the obstacle was closer than a predefined threshold and would adjust the wheelchair route by driving it in an opposite direction to the detected obstacle if there were no other obstacles in the opposite direction. Once the detected obstacle was at a greater distance than a predefined threshold, the wheelchair would be able to drive it in the desired direction. If there were other obstacles in the opposite direction closer than the predefined threshold, the wheelchair would drive backwards for two seconds to clear its path. The predefined threshold could be adjusted digitally in the Python program.

A User Interface (UI) was created to control the new approach. The User Interface is shown in Figure 7. The UI had a drop-down menu, Set Mode, Start and Stop buttons. The drop-down menu was used to allow the user to select the desired driving mode. The set Mode button was used to apply the desired driving mode. If a user pressed the Set Mode button or Start button without selecting a desired driving mode, the user would not be able to control the wheelchair and a message would be displayed requesting the user to select a driving mode before starting to drive. When a mode was selected and the Start button was pressed, the Python program would start analysing readings from the ultrasonic transducer and intelligently mixing them with the user's desired direction supplied by the user input device [3,4]. The approach provided a collision-free driving direction for the powered wheelchair. The Stop Button was used to stop the user from controlling the wheelchair. For safety reasons the user would not be able to control the wheelchair unless the Start Button was pressed.



{Please Select the Desired Driving Mode}

Fig.7. User Interface used to control the new approach.

3 Testing

Testing was conducted using small plastic cones to create a path for the powered wheelchair. The plastic cones marked the edges of the path and were used as obstacles. Figure 8. shows the setting used to test the new approach. Both driving modes were tested, in both modes, the new approach successfully detected the cones in both driving modes. Using the stop mode, the new approach prevented the wheelchair from knocking the cones off. Using the Avoid mode, the new approach successfully drove the wheelchair away from the cones and provided a collision-free route for the wheelchair.

Fig. 8. The setting used to test the new approach.

Figure 9. shows the wheelchair detecting the cones on the right. Using the Stop mode, if the user wanted to drive the wheelchair to the right, the new approach would not allow that. In this case, the user drove to the left and avoided the cones. Using the Avoid mode, if the user desired to drive the wheelchair to the right, the new approach would not allow that and would drive the wheelchair to the left to avoid the cones until the cones were farther than the threshold distance, only then, would it allow the user to drive to the right.

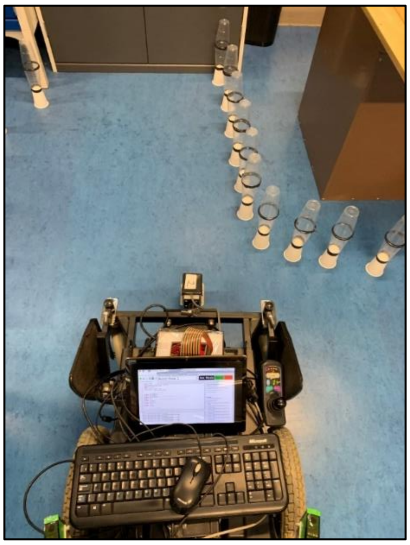


Fig. 9. The wheelchair detecting cones to the right.

Figure 10. shows the wheelchair detecting the cones on the left. Using the Stop mode, if the user wanted to drive the wheelchair to the left, the new approach would not allow that. In this case, the user drove to the right and avoided the cones. Using the Avoid mode, if the user desired to drive the wheelchair to the left, the new approach would not allow that and would drive the wheelchair to the right to avoid the cones until the cones were farther than the threshold distance, only then would it allow the user to drive to the left.

Fig. 10. The wheelchair detecting cones to the left.

4 Risk Assessment and Mitigation

4.1 Risk Anatomy

There is no single definition of risk. Risk is classically defined as uncertainty concerning the occurrence of a loss [6]. Loss is a major concern to everybody especially when it comes to the health and well-being of powered wheelchair users. If a powered wheelchair user is confronted with an obstacle in their way, depending on the nature of the obstacle, the consequence could be anything from null to a major loss caused by an injury.

For a risk to manifest itself, two components are essential [6]:

1. The hazard: This is a condition that creates or increases the frequency or severity of loss. Such a condition could be the existence of obstacles in the wheelchair path.
2. The peril: This is the cause of loss, which is the impact resulting from the hazard.

More than three thousand years ago, Aristotle said: “if it can happen, it will”.

4.2 Types of Risk

Risks have been classified mainly into the following categories [6]:

1. Pure and speculative risk
2. Diversifiable risk and non-diversifiable risk
3. Enterprise risk
4. Systemic risk

The risk confronted by a wheelchair user is a pure risk which means either “loss” (injury) or “no loss” (no injury). The authors argue that humanity cannot afford impediments to wheelchair users due to financial, emotional, and practical considerations.

4.3 Risk Mitigation

Besides other financial techniques, Risk can be mitigated and controlled in four ways [6,7]:

- Avoidance: This is achieved by totally avoiding the risk.
- Loss prevention: This is achieved by reducing the probability of risk manifesting itself.
- Loss reduction: This is achieved by reducing the impact of the risk.
- Retention: This is achieved by doing nothing to mitigate the risk and consequently accepting it.

The direct implication is to mitigate wheelchair impact risks using an intelligent system similar to the Intelligent-SCAD setup presented in this paper.

5 Conclusions

This paper presented a new approach to digitising the Scanning Collision Avoidance Device created by Langner [2]. The new approach was more responsive than the system created by [2] and could interface any input device used to supply users’ desired direction.

The new approach incorporated two driving modes: Stop mode and Avoid mode. The new approach successfully detected obstacles in the wheelchair surroundings and identified the distance to the obstacles and their location. Stop mode prevented the user from driving the wheelchair in the direction of an obstacle direction if the obstacle was closer than a predefined safety distance. Avoid mode prevented the user from driving the wheelchair in the direction of an obstacle if the obstacle was closer than a predefined safety distance and drove the wheelchair in an opposite direction if no obstacles were detected in the opposite direction until the obstacle was farther than the predefined safety distance. If there were several obstacles in all directions surrounding the wheelchair, the new approach would drive the wheelchair backwards for two seconds to clear the wheelchair’s path.

Different collision avoidance algorithms could be installed onto the Raspberry Pi to improve wheelchair driving such as computer vision [8] and deep learning [9]. The authors are currently using different machine-learning classification algorithms.

The new approach improved mobility and enhanced the quality of life of powered wheelchair drivers and reduced the need and cost of carers.

Future work will include using a combination of computer vision [10], deep learning [11] and other machine learning algorithms for Intelligent user inputs, Human-Machine Interaction [12] and collision avoidance.

References

1. Sanders, D., Gegov, A., 2018. Using artificial intelligence to share control of a powered-wheelchair between a wheelchair user and an intelligent sensor system, EPSRC Project, 2019.
2. Langner, M., 2012. Effort Reduction and Collision Avoidance for Powered Wheelchairs: SCAD Assistive Mobility System, Ph.D. dissertation, University of Portsmouth.
3. Haddad, M., Sanders, D., Langner, M., Bausch, N., Thabet, M., Gegov, A., Tewkesbury, G. and Ikwan, F. (2021) Intelligent Control of the Steering for a Powered Wheelchair Using a Microcomputer. In: Arai K., Kapoor S., Bhatia R. (eds) Intelligent Systems and Applications. IntelliSys 2020. Advances in Intelligent Systems and Computing, vol 1252. Springer, Cham.
4. Haddad, M., Sanders, D., Ikwan, F., Thabet, M., Langner, M. and Gegov, A. Intelligent HMI and Control for Steering a Powered Wheelchair Using a Raspberry Pi Microcomputer, 2020 IEEE 10th International Conference on Intelligent Systems (IS), 2020, pp. 223-228.
5. Texas Instruments, Isolated Voltage-Measurement Circuit With ±250-mV Input. Availabe at: https://www.ti.com › lit › pdf › sbaa350, accessed on 22nd October 2023.
6. Rejda et al., Principles of Risk Management and Insurance. pp.22-27, 2014, 5th ed.
7. Hubbard, D., The Failure of Risk Management: Why It's Broken and How to Fix It. 2009, John Wiley & Sons. p. 46.
8. Haddad, M., Sanders, D., Tewkesbury, G., Langner, M., Fakhraddin Muhyaddin, S. and Ibrahim, M.: Computer-vision algorithms to steer powered wheelchairs. Journal of Physics: Conference Series. (2021).
9. Haddad, M.J. and Sanders, D.A., 2020. Deep Learning Architecture to Assist With Steering a Powered Wheelchair. IEEE Transactions on Neural Systems and Rehabilitation Engineering: a Publication of the IEEE Engineering in Medicine and Biology Society, 28(12), pp.2987-2994.
10. Haddad, M., Sanders, D., Langner, M. and Tewkesbury, G., 2022. One shot learning approach to identify drivers. In Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 1 (pp. 622-629). Springer International Publishing.
11. Haddad, M., Sanders, D., Tewkesbury, G., Langner, M. and Keeble, W. A novel collision avoidance approach for powered wheelchair steering using deep learning. Journal of Physics: Conference Series. (2021).
12. Sanders D, Haddad M, Tewkesbury G, Bausch N, Rogers I and Huang Y (2020) Analysis of Reaction Times and Time-Delays Introduced into an Intelligent HCI for a Smart Wheelchair. In 2020 IEEE 10th International Conference on Intelligent Systems (IS), pp 217-222. IEEE.