Intelligent User Interface to Control a Powered Wheelchair using Infrared Sensors

Malik Haddad[1], David Sanders[1], Giles Tewkesbury[1] and Martin Langner[1,2]

[1] University of Portsmouth, Portsmouth, PO1 3DJ, UK
[2] Chailey Heritage Foundation, North Chailey. BN8 4EF, UK
malik.haddad@port.ac.uk

Abstract. This paper presents a new system to steer a powered wheelchair using a Sharp IR sensor and a Raspberry Pi. Interviews with occupational therapists, helpers and carers at Chailey Heritage Foundation/School revealed that clicking noises generated from closing switch contact used to operate powered wheelchairs disturbed the attention and reduced the focus of young wheelchair users having cognitive or physical disability. Also switches often slipped away and became unreachable. The new system replaced lever-switches used to steer powered wheelchairs by an electronic circuit. The circuit consisted of a Sharp IR sensor, Analogue to Digital converter, relays, and a Raspberry Pi. The sharp IR sensor detected movement in its range and the Raspberry Pi interpreted the data and generated commands to steer a powered wheelchair. Two modes were used to overcome the problem of sensors slipping from position: Click to Calibrate and Auto-Calibrate. A technical User Interface was created to modify sensitivity, user and detection settings. Practical testing showed the system behaved satisfactorily. It detected users' voluntary movements and used them to steer a powered wheelchair and overcome the problem of switches slipping from position. Clinical trials will be conducted at Chailey Heritage Foundation.

Keywords: Disabled, Infrared Sensor, User Interface, Wheelchair.

I. Introduction

This paper presents a new system to steer a powered wheelchair using a Sharp IR sensor, a Raspberry Pi and a set of relays. The work described here is part of broader research carried by the authors at Chailey Heritage Foundation and the University of Portsmouth funded by the Engineering and Physical Sciences Council (EPSRC) [1]. The main aims of this research are to use Artificial Intelligence (AI) techniques to increase mobility and improve the quality of life of disabled powered wheelchair users by improving their self-reliance and confidence.

Recent studies showed that around one sixth of the world population were diagnosed with some sort of disability with 2-4% of them diagnosed with mobility problems [2]. Due to recent medical advancements, population ageing and the spread of long term health problems these figures were increasing [2,3] and consequently disabled individuals usually had lower quality of life [4].

People with disability often used powered mobility for daily activities [5] and researchers have aimed to improve powered mobility using: sensor systems [6], ultrasonics [7] Multiple Criteria Decision Making and ultrasonic sensors [8-11], Expert Systems [12,13], Rule Based Systems [14], microcomputers and Human Machine Interfaces [15,16], Deep Learning [2,17] and camera modules [18-21].

Wheelchair users often controlled their speed and direction with a joystick. If a user lacked the coordination to effectively use a joystick or if they could not use their hands or fingers then other input devices could be used (lever-switch, foot control, sip tubes/puff switches, or head or chin controllers, etc.). Of these lever switches were often used to control powered wheelchairs. A lever-switch was a type of electrical component used to control current to the electrical motors. Fig. 1 shows lever switches used to control a powered wheelchair.

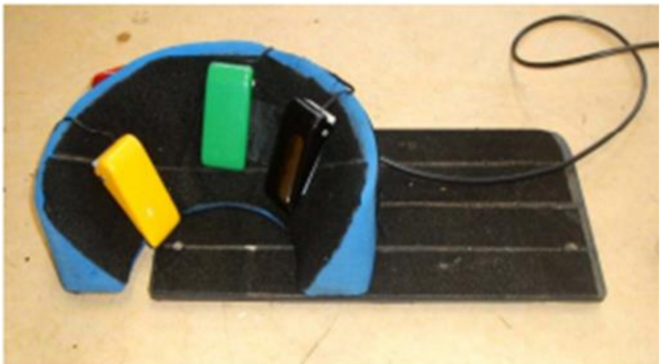

Fig. 1. Lever switches used to steer a powered wheelchair [7].

Several switches were often used to control a wheelchair. Wheelchair controllers have normally been open-loop. Powered wheelchair users indicated their preferred direction by pressing the switch responsible for driving a wheelchair in that direction. Pressing the switch closed contacts and connected a circuit which triggered a specific relay and allowed current to flow to activate the motor responsible for driving the wheelchair in the desired direction.

Closing switch contacts made clicking sounds. Recent interviews carried out by the authors with occupational therapists, helpers and carers at Chailey Heritage Foundation/School revealed that the clicking noise caused discomfort, disturbed attention and reduced the focus of some young wheelchair users having cognitive or physical disability. Also switches often slipped away and became unreachable because of wheelchair movement or because of the terrain. Helper intervention was often required to reposition a switch. That often disrupted driving sessions and user engagement.

A new system is presented in this paper to replace the lever switches and overcome the problem of switches slipping from position. The next Section presents the new system. Section 3 presents some results and discussion. Conclusions and future work are presented within Section 4.

II. The New System

The work presented in this paper replaced the lever switches used to operate a powered wheelchair with a new system that did not generate clicking sounds and that overcame the problem of repositioning. A replacement system was created using a Sharp IR range sensor (SHARP GP2Y0A41SK0F) and a Raspberry Pi. The prototype system is shown in Fig. 2.

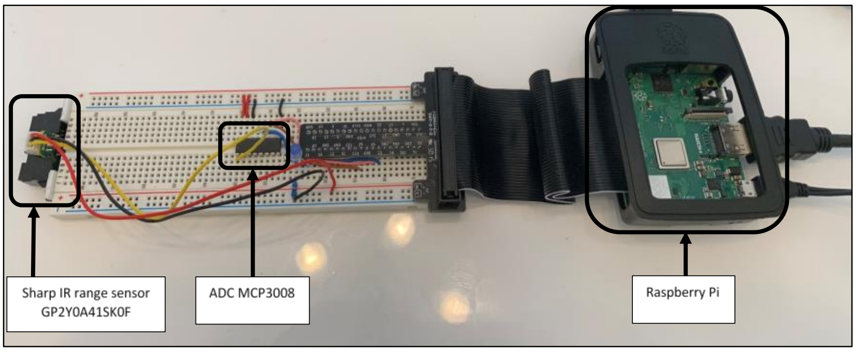

Fig. 2. Prototype of the new system to replace lever-switches.

The Sharp IR range sensor was suitable to detect objects in ranges from 4cm to 30cm and required 5Volts DC and ground connections. The Sharp sensor communicated with the Raspberry Pi via Special Peripheral Interface (SPI). A program was created using Python programming language as shown in Fig. 3. To control the function of the range sensor, a simple User Interface (UI) was created containing three buttons: Start, Stop and Calibrate. The UI is shown in Fig. 4. Users with cognitive disability often used the UI. The simple UI matched the capabilities of users with disability. The new system had a straightforward operation for steering powered wheelchairs and provided a suitable match between desired commands and user capabilities [22].

```
from subprocess import call
from tkinter import ttk
from tkinter import *
from tkinter import messagebox
import sys
import threading
import tkinter
import time
import Adafruit_GPIO.SPI as SPI
import Adafruit_MCP3008
import RPi.GPIO as GPIO

# for safety set condition for while loop
false
global X
X = False
#Set pin confguration
BlueLED = 20
CLK = 18
MISO = 23
MOSI = 24
CS = 25
#use adafruit library using pin configuration
mcp = Adafruit_MCP3008.MCP3008(clk=CLK,
cs=CS, miso=MISO, mosi=MOSI)
#use set mode to assign led
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(BlueLED,GPIO.OUT)

def SetXTrue():
    # for safty keep wheelchair off
    GPIO.output(BlueLED, GPIO.LOW)
    #Calculate initial reference distance
    v = (mcp.read_adc(0) / 1023.0) * 3.3
    RefDist = 16.2537 * v**4 - 129.893 * v**3
+ 382.268 * v**2 - 512.611 * v + 301.439
    global X
    X = True
    while X==True:
        root.update()
    # for safty keep wheelchair off
        GPIO.output(BlueLED, GPIO.LOW)
    # calculate current distance
        v = (mcp.read_adc(0) / 1023.0) * 3.3
        dist = 16.2537 * v**4 - 129.893 *
v**3 + 382.268 * v**2 - 512.611 * v + 301.439

    #condition used if movement was detected
    #Threshold can be modified in the if
statement
        if dist < (RefDist-2):
            GPIO.output(BlueLED, GPIO.LOW)
            GPIO.output(BlueLED, GPIO.HIGH)

    #condition used if no movement was
detected
    #Threshold can be modified in the if
statement
        if dist > (RefDist-2):
            GPIO.output(BlueLED, GPIO.LOW)
    #delay
        time.sleep(0.5)
    GPIO.output(BlueLED, GPIO.LOW)

def SetXFalse():
    global X
    X = False
    return (X)

def Calibrate():
    global X
    X = False
    SetXTrue()

#set up tkinter box
root = tkinter.Tk()
root.title("IR Sensor")
root.geometry("1000x600")

#Start sensor by clicking a button
User=tkinter.Button(root, text="Start", fg =
"black", bg="green", font=('comicsans', 50),
command=(SetXTrue))
User.grid(row = 15, column=10)

#Stop sensor by clicking a button
User=tkinter.Button(root, text="Stop", fg =
"black", bg="red", font=('comicsans', 50),
command=(SetXFalse))
User.grid(row = 15, column=15)

#Calibrate Refrence Distance by clicking a
button
User=tkinter.Button(root, text="Calibrate",
fg = "white", bg="black", font=('comicsans',
50), command=(Calibrate))
User.grid(row = 40, column=12)
```

Fig. 3. Python3 program used to the create User Interface and control the Sharp IR range sensor.

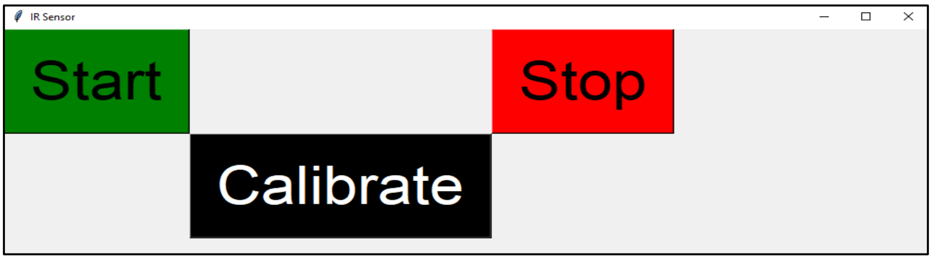

Fig. 4. Simple user interface to control the function of the Sharp IR range sensor.

The program was installed on to the Raspberry Pi. The Start button was used to activate the sensor. For safety reasons the sensor would not control the function of the wheelchair unless the Start Button was pressed. If the Start Button was pressed, the sensor would detect the nearest object in its range and the Raspberry Pi would calculate the distance between them. This distance was used as a reference to check for movement.

If the sensor detected an object in its range, the sensor would send analogue data to Channel 0 of the ADC. The ADC would converted the analogue data to digital data that could be interpreted by the Raspberry Pi. Using sensor datasheet, a fourth order polynomial approximation was applied to find the distance in (cm) between the sensor and a detected object [23]. The distance between the sensor and a detected object was calculated using Equation 1.

$$D = 16.2537v^4 - 129.893v^3 + 382.268v^2 - 512.611v + 301.439 \qquad (1)$$

Where D represented the distance between the sensor and the detected object and v represented the digital value generated by ADC.

If a sensor detected an object in its range, The Raspberry Pi would calculate the distance between the sensor and that object. The Python program would set that distance as a reference distance. If the object moved, the sensor would detect that movement and the Raspberry Pi would calculate the new distance between the sensor and the object, the distance moved and the direction of the movement.

A threshold value was set to identify how much movement was required to activate the wheelchair. A user would have a specific threshold value assigned to him. That threshold value depended on the amount of voluntary movement the user could generate based on their level of functionality and type of disability. If the object moved towards the sensor and the distance moved was greater than the threshold value, the Raspberry Pi would assign a high logic value to a designated pin identified by the program. The high logic value was used to activate the wheelchair motor. If the object moved away from the sensor and the distance moved was greater than the threshold value, the Raspberry Pi would assign a low logic value to the designated pin to deactivate the motor. The Stop Button was pressed.to stop the sensor from controlling the wheelchair.

The Calibrate Button was used to overcome the problem of undesired slippage of the sensor. For example, if the sensor slipped towards the user and the wheelchair was undesirably activated or the voluntary movement produced by the user was no longer in the range of the sensor. In such cases, helpers could press the Calibrate Button to trigger the sensor to search for the nearest object and the Raspberry Pi would calculate a new reference distance to detect movement from the sensor's new position.

Auto-Calibrate mode was added to the new system. In this mode, the sensor automatically calculated the reference distance without the need to press the Calibrate Button. The new mode did not require a UI since it operated during the boot-up sequence of the Raspberry Pi.

III. Testing

A student at Chailey Heritage School was considered as a case study. The student could produce voluntary movement with their head and used that movement to control a powered wheelchair. Head switches were used to transform that movement to steering direction of a powered wheelchair. The voluntary movement was used to steer a powered wheelchair left or right.

The head switches were replaced by two Sharp IR sensors. The sensors were placed at the same physical location as the head switches. The sensors were used to detect the movement of the student's head as shown in Fig. 5.

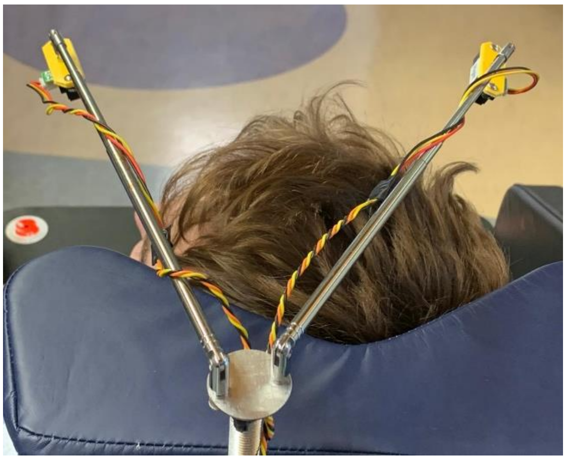

Fig. 5. IR sensors used to operate a powered wheelchair.

A sensor was activated when the student moved their head towards it. The new system presented in this paper transformed the voluntary movement to steering directions for the powered wheelchair.

A new technical UI was created to allow helpers input user settings, advanced parameters were used to accurately translate users' desires and filter out unwanted movement, see Fig. 6.

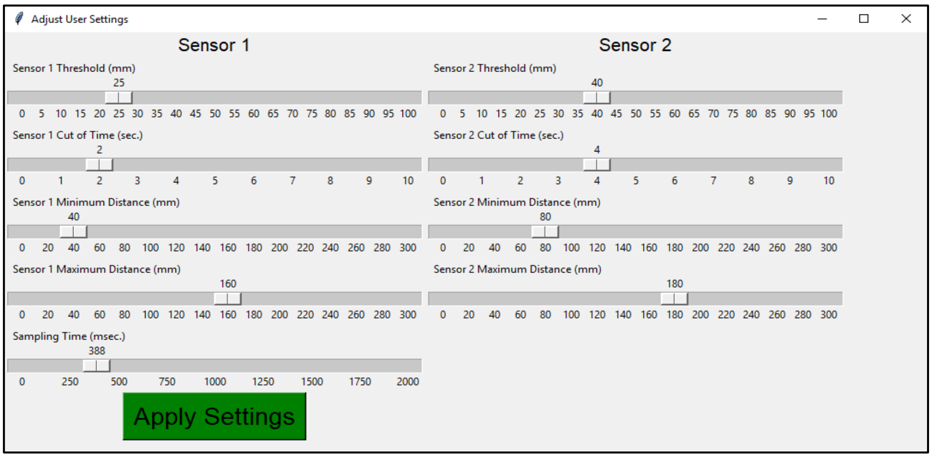

Fig. 6. Technical User Interface to input user settings.

The new technical UI allowed helpers to input and modify different parameters using track-bars as shown in Fig. 6. The following parameters were considered:

- Threshold represented the minimum detection distance of the sensor. The larger the Threshold value the less sensitive the sensor would be to movements of the user's head.
- Cut off Time allowed the new system to operate in two different modes: Switch Mode and Time Delay Mode. Setting the Cut off Time track-bar to 0 would trigger the system to operate in Switch Mode, where the system would use the sensors as switches. If an object was detected, a specific relay was triggered on and would remain triggered until the object was no longer detected. Setting the Cut off Time track-bar to any value other than 0 allowed the system to operate in Time Delay Mode where if an object was detected in a sensor range, a specific relay was activated depending on the value specified by the track-bar in seconds, then the relay would be switched off if no object was detected within sensor range.
- Minimum distance represented the minimum cut off distance, where closer objects were not detected.
- Maximum distance represented the maximum cut off distance, where objects beyond it were not detected.
- Sampling Time allowed powered wheelchair users drive their wheelchairs safely across uneven ground. If the drive terrain was unsettled, the sensors could provide readings that did not represent the users' desires because of the vibrations, head movement, or momentary sensor movements. The new system took two readings for an object detected in its range with the Sampling Time in between. That allowed the new system to overcome the problem of momentary sensor movement and head movement due to uneven driving terrain.

Detection range was calculated by subtracting Minimum distance from Maximum distance. If helpers input Maximum distance smaller than the Minimum distance, an error message would appear asking the helper to adjust the sensor position according to the input distances as shown in Fig. 7.

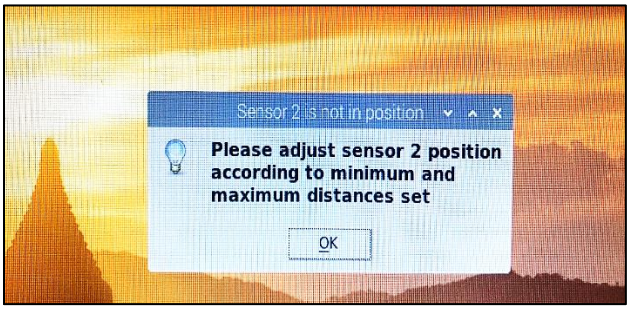

Fig. 7. Error message asking Helpers to adjust sensor position.

Fig. 8 shows the prototype of the new system connected to a touchscreen used to input users' settings.

Fig. 8. Prototype of the new system showing a touch screen to input user settings.

Once Driver Settings were input and the Apply Settings Button was pressed, the program stored all the settings in a CSV file. The CSV file could be used later to install driver settings during boot-up. The Auto-Calibrate Mode would read the CSV file and install driver's settings during boot-up.

IV. Conclusions and Future Work

A new system was created to replace lever switches used to operate powered wheelchairs. The new system eliminated the clicking noise generated from closing switch contacts and the problem of switches becoming unreachable to users due to powered wheelchair movement or uneven terrain. This was achieved by introducing two functioning modes to overcome the problem: Click-to-Calibrate and Auto-Calibrate. The two modes improved disabled powered wheelchair users' mobility.

The new system provided a friendly and simple User Interface and required less effort to control a powered wheelchair. To reduce cost, the authors will upload the schematic diagram and the Python program to an open access platform. Disabled wheelchair users will be able to download free of charge. The new system provided a faster response time and required less hardware than the lever-switches. Clinical trials will be conducted to investigate the effectiveness of the new system and improvements will be made based on users, helpers and carers feedback.

Future work will consider creating a new UI to incorporate multiple Users [15,24]. Users could be digitally added to the new system. A specific threshold value will be assigned for each user based on their needs, level of functionality and type of disability. Future work will also consider using Artificial Neural Networks to identify which user is operating the new system and how they are performing.

Acknowledgment

This research was supported by an EPSRC EP/S005927/1 project titled "Using artificial intelligence to share control of a powered-wheelchair between a wheelchair user and an intelligent sensor system". Investigators: Sanders, DA and Gegov, AE. Senior Researchers Haddad MJ and Langner MC.

References

1. Sanders D, Gegov A.: Using artificial intelligence to share control of a powered-wheelchair between a wheelchair user and an intelligent sensor system. EPSRC Project, 2019 – 2022. (2018).
2. Haddad, M.J., Sanders, D.A.: Deep Learning architecture to assist with steering a powered wheelchair. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 28(12), 2987-2994 (2020).
3. Krops, L.A., Hols, D.H., Folkertsma, N., Dijkstra, P.U., Geertzen, J.H. and Dekker, R.: Requirements on a community-based intervention for stimulating physical activity in physically disabled people: a focus group study amongst experts. Disability and Rehabilitation, 40(20), 2400-2407 (2018).
4. Bos, I., Wynia, K., Almansa, J., Drost, G., Kremer, B. and Kuks, J.: The prevalence and severity of disease-related disabilities and their impact on quality of life in neuromuscular diseases. Disability and rehabilitation, 41(14), 1676-1681 (2019).
5. Frank, A.O., De Souza, L.H.: Clinical features of children and adults with a muscular dystrophy using powered indoor/outdoor wheelchairs: disease features, comorbidities and complications of disability. Disability and rehabilitation, 40(9), 1007-1013 (2018).
6. Sanders D.A, Langner M, Tewkesbury G E. Improving wheelchair-driving using a sensor system to control wheelchair-veer and variable-switches as an alternative to digital-switches or joysticks. Industrial Robot: An international journal, 32(2), 157-167 (2010).
7. Langner, M.: Effort Reduction and Collision Avoidance for Powered Wheelchairs: SCAD Assistive Mobility System, Doctoral dissertation, University of Portsmouth (2012).
   Sanders, D.A.: Using self-reliance factors to decide how to share control between human powered wheelchair drivers and ultrasonic sensors. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 25(8), 1221-1229 (2016).
8. Haddad, M., Sanders. D., Gegov. A., Hassan. M., Huang, Y., Al-Mosawi, M.: Combining multiple criteria decision making with vector manipulation to decide on the direction for a powered-wheelchair. In: Arai K, Kapoor S, Bhatia R, (eds.) Intelligent Systems and Applications. IntelliSys 2020. Advances in Intelligent Systems and Computing, vol. 1037, pp. 680-693. Springer, Cham (2019).
9. Haddad, M., Sanders, D., Langner, M., Ikwan, F., Tewkesbury, G., Gegov, A.: Steering Direction for a Powered-Wheelchair Using the Analytical Hierarchy Process. In: Proceedings of the 2020 IEEE 10th International Conference on Intelligent Systems (IS), pp. 229-234. IEEE, Bulgaria (2020).
10. Haddad, M., Sanders, D., Thabet, M., Gegov, A., Ikwan, F., Omoarebun, P., Tewkesbury, P., Hassan, M.: Use of the Analytical Hierarchy Process to Determine the Steering Direction for a Powered Wheelchair. In: Arai K, Kapoor S, Bhatia R, (eds.). Intelligent Systems and Applications. IntelliSys 2020. Advances in Intelligent Systems and Computing, vol. 1252, pp. 617-630. Springer, Cham (2021).
11. Haddad, M.J., Sanders D A.: Selecting a Best Compromise Direction for a Powered Wheelchair Using PROMETHEE. IEEE Transactions on Neural Systems and Rehabilitation Engineering 27(2), 228-235 (2019).
12. Sanders, D.A., Bausch, N.: Improving steering of a powered wheelchair using an expert system to interpret hand tremor. In International Conference on Intelligent Robotics and Applications, pp. 460-471. Springer, Cham (2015).

13. Sanders, D.A., Haddad, M., Tewkesbury, G.E., Thabet, M., Omoarebun, P., Barker, T.: Simple expert system for intelligent control and HCI for a wheelchair fitted with ultrasonic sensors. In 2020 IEEE 10th International Conference on Intelligent Systems (IS), pp. 211-216. IEEE, Bulgaria (2020).
14. Sanders, D.A., Gegov A, Haddad M, Ikwan F, Wiltshire D, Tan Y C.: Rule-based Expert System to decide on direction and speed of a powered wheelchair. In: Arai K, Kapoor S, Bhatia R, (eds.). Intelligent Systems and Applications. IntelliSys 2019. Advances in Intelligent Systems and Computing, vol. 868, pp. 822-832. Springer, Cham (2019).
15. Haddad, M., Sanders, D., Ikwan, F., Thabet, M., Langner, M., Gegov, A.: Intelligent HMI and control for steering a powered wheelchair using a Raspberry Pi microcomputer. In 2020 IEEE 10th International Conference on Intelligent Systems (IS), pp. 223-228. IEEE, Bulgaria (2020).
16. Haddad, M., Sanders, D., Langner, M., Bausch, N., Thabet, M., Gegov, A., Tewkesbury, G., Ikwan, F.: 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, pp. 594-603. Springer, Cham (2021).
17. Haddad, M., Sanders, D., Tewkesbury, G., Langner, M.: A Novel Collision Avoidance System for Steering a Powered Wheelchair Using Deep Learning Architecture. In: 24th IEEE International Conference on Intelligent Transportation - ITSC2021. *Submitted*. IEEE, USA (2021).
18. Haddad, M., Sanders, D., Tewkesbury, G., Langner, M.: Novel Approach for Digitizing the Scanning Collision Avoidance Device Detection Range. In: 32nd IEEE Intelligent Vehicles Symposium, 2021. *Submitted*. IEEE, Japan (2021).
19. Haddad, M., Sanders, D., Langner, M., Tewkesbury, G.: Novel Approach to Steer a Powered Wheelchair Using Image Processing Algorithm and Raspberry Pi. In: 32nd IEEE Intelligent Vehicles Symposium, 2021. *Submitted*. IEEE, Japan (2021).
20. Haddad, M., Sanders, D., Tewkesbury, G., Langner, M.: Using Open Source Computer Vision Algorithms to Drive a Powered Wheelchair. In: 24th IEEE International Conference on Intelligent Transportation - ITSC2021. *Submitted*. IEEE, USA (2021).
21. Haddad, M., Sanders, D., Langner, M., Tewkesbury, G.: Steering a powered wheelchair using a camera module and Python Imaging Library. In: 24th IEEE International Conference on Intelligent Transportation - ITSC2021. *Submitted*. IEEE, USA (2021).
22. Lewis, C.: Simplicity in cognitive assistive technology: a framework and agenda for research. Universal Access in the Information Society, 5(4), 351-361 (2007).
23. Spark Fun Homepage, https://www.sparkfun.com/products/8958#comment-4f864d34ce395f9161000000, last accessed 2020/11/16.
24. Haddad, M., Sanders, D., Langner, M., Tewkesbury, G.: One Shot Learning Approach to Identify Drivers. In: SAI Intelligent Systems Conference, IntelliSys. *Accepted and in Press*. Netherlands (2021).