Chatzidimitriadis, Sotirios, Oprea, Paul, Gillham, Michael and Sirlantzis, Konstantinos (2017) *Evaluation of 3D obstacle avoidance algorithm for smart powered wheelchairs.* In: 2017 Seventh International Conference on Emerging Security Technologies (EST). IEEE, pp. 157-162. ISBN 978-1-5386-4019-

Evaluation of 3D obstacle avoidance algorithm for smart Powered Wheelchairs

Sotirios Chatzidimitriadis
University of Kent, Canterbury, UK and
School of Electrical and Computer Engineering
Aristotle University of Thessaloniki
Thessaloniki, Greece
Email: sotichat@ece.auth.gr

Paul Oprea, Michael Gillham and Konstantinos Sirlantzis
University of Kent
Canterbury, UK
Email: P.Oprea@kent.ac.uk, M.Gillham@kent.ac.uk,
K.Sirlantzis@kent.ac.uk

***Abstract*—This research investigates the feasibility for the development of a novel 3D collision avoidance system for smart powered wheelchairs operating in a cluttered setting by using a scenario generated in a simulated environment using the Robot Operating System development framework. We constructed an innovative interface with a commercially available powered wheelchair system in order to extract joystick data to provide the input for interacting with the simulation. By integrating with a standard PWC control system the user can operate the PWC joystick with the model responding in real-time. The wheelchair model was equipped with a Kinect depth sensor segmented into three layers, two representing the upper body and torso, and a third layer fused with a LIDAR for the leg section. When using the assisted driving algorithm there was a 91.7% reduction in collisions and the course completion rate was 100% compared to 87.5% when not using the algorithm.**

I. Introduction

Driving assistance and semi-autonomous mobile robotic systems need to be able to detect obstacles and then take some appropriate action. There are two major issues for developing a safe system which is sufficiently robust for use with onboard human pilots, the first is to be able to detect all obstacles and the second is to have an algorithm which allows the platform to maneuver in highly cluttered human environments.

Previous research has utilized a range of sensors; LIDAR or laser ranging is limited to a thin 2D slice which limits the ability of obstacle detection [1] others have used ultrasonic ranging [2] with a wider slice but still 2D, and some have used stereo cameras to detect obstacles and provide mapping [3]; however, although 3D it is limited to a narrow angular width. Even though sensor data fusion and the use of multiple sensor arrays could provide a more robust obstacle detection system there still remains the unmet need for a suitable collision avoidance model.

Many smart wheelchairs collision avoidance systems [4] rely upon the potential field concept [5] or they are derived from The Virtual Force Field (VFF) method which led to the Vector Field Histogram (VFH) method [6]. However other research has found the method difficult to tune [7], they found another method, the Dynamic Window Approach (DWA) [8], was easier to adjust and a better representation of the platform dynamics. However they found this method was not well suited to real-time application due to the high computation overhead, instead they developed a novel hybrid approach. Research at the University of Seville took a similar tactic by modifying the DWA method in a shared dynamic control [9].

There are several reasons for developing powered wheelchair (PWC) simulators, the main reason has been to provide a means to train and better acquaint users prior to the issue of a device [10]. However by integrating our Robot Operating System (ROS) [11] based simulator with a standard PWC control system, using our previously developed hardware [12], [13], we are able to provide a simulation which can not only be used to train the user but also provide a means of evaluating assistive technologies and to provide data for setting up and adjusting standard PWC control system parameters, particularly as users abilities can change over time.

II. Development of 3D Collision Avoidance

The powered wheelchair kinematic model can be described as a unicycle [14] with a tank like differential drive where both drive wheels are on the same axle yet independently driven; the platform reference frame is shown in Fig. 1 and the platform dynamic constraints expressed mathematically by:

$$\dot{x} = \frac{v_{right} + v_{left}}{2} = v_{body} \tag{1}$$

$$\dot{\theta} = \omega_{body} \tag{2}$$

$$v_{mr} = v_{body} + \frac{W\omega_{body}}{2} \tag{3}$$

$$v_{ml} = v_{body} - \frac{W\omega_{body}}{2} \tag{4}$$

Where:
W = The distance between the two rear drive wheels
$v_{right,left}$ = The velocity of the rear drive wheels
v_{ml}, v_{mr} = Motor drive outputs

The Dynamic Localized Adjustable Force Field (DLAFF) employs the concept of an active window/frame containing a nonlinear adjustable force field [15] which is elliptically shaped to provide a better mathematical relationship between

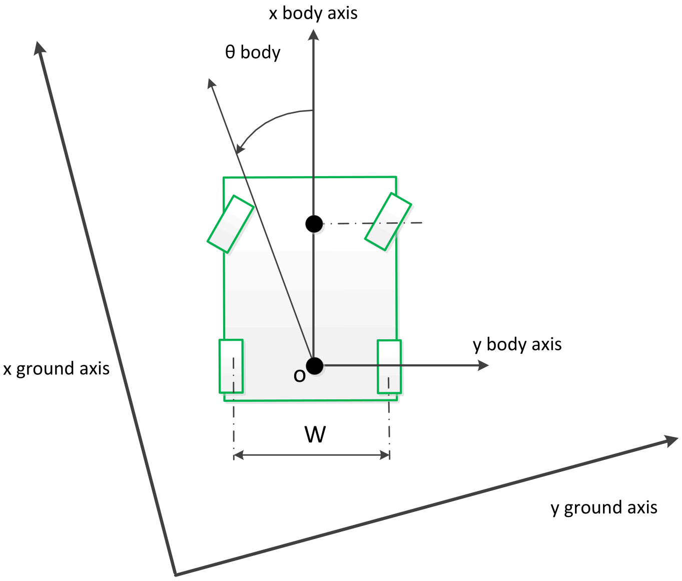

Fig. 1. Powered wheelchair frame of reference

the repulsive force and the kinematic of the platform. This repulsive force field behaves according to Eqn.5 in a manner which allows the platform to maneuver in a highly cluttered environment with minimal free space by adjustment of the k term in the equation. In the DLAFF method the inner ellipse provides a zone in which the physical boundary of the platform is fully contained and the outer ellipse provides the furthest extent of the repulsive field, where the repulsive force is determined by Eqn. 5 along the vector P-r (Fig. 2) to the nearest obstacle in each quadrant, where that repulsive force acts to damp the motor drive outputs given by Eqns. 3 and 4.

$$F = 1 - \frac{1}{e^{((R-p)/k)}} \tag{5}$$

One foci (f1 in Fig. 2) of the inner ellipse is the origin of the platform reference frame marked o in Fig. 1 and Fig. 2, and the a dimension of the ellipse lies on the x body axis with the b of the ellipse on the y body axis, which is also the drive wheel axis. The outer ellipse is free to expand outward from the inner ellipse, along the body axes, or contract back down to the inner ellipse in order to adjust the collision avoidance to take into account the velocity of the platform when reacting to the environment.

We can extending the DLAFF method to act in a more 3D manner which better represents the platform shape profile shown in Fig. 3. We propose to represent the upper layer using one ellipse which is shaped according to the head and upper torso, the middle layer shaped to encompass the arms and the arm rests of the platform. The lower layer acts to provide the inner ellipse safety zone around the legs and feet, and base of the platform.

The advantage of structuring into distinct layers, each one separated from the others at the points, at which the shape of the vehicle changes will be to allow intricate maneuvers to take place. For example; in the case of the wheelchair being driven up to a desk, or table, would mean the algorithm would allow the users legs to pass under the desk, since no obstacle would be detected in the respective layer, but would stop before the users body touched the desk, because the desk itself would be recognized as an obstacle in the “body” layer.

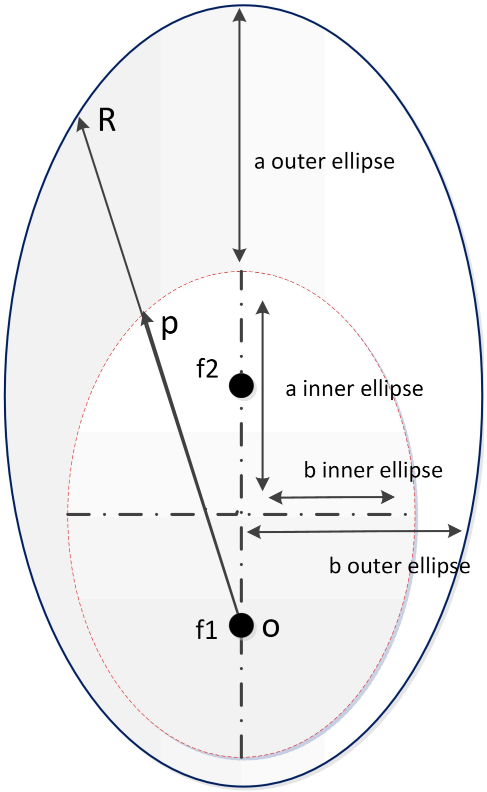

Fig. 2. DLAFF collision avoidance ellipse

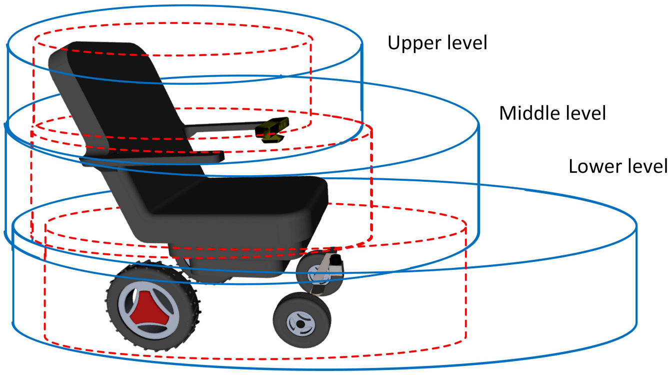

Fig. 3. 3D avoidance layers

The DLAFF method utilizes the closest object in each of two quadrants, which two depends on the translation and rotation, to determine the damping force applied to each drive motor. We can extend this to the three layers by simply determining the nearest object and which layer it belongs to; where we are only concerned if an object has entered the outer boundary of one of the layers in the quadrants related to the current platform motion. We therefore end up with the two closest obstacles to the actual body of the model and we can proceed with calculating the necessary dampening for each wheel as in the original method.

The general equation of the ellipse is given by:

$$\frac{x^2}{a} + \frac{y^2}{b} = 1 \tag{6}$$

And its eccentricity by:

$$e = \sqrt{1 - \frac{b^2}{a^2}} \quad \text{where} \quad 0 \leq e \leq 1 \tag{7}$$

If we use polar coordinates with the origin at f1, which is the body frame of reference o as shown in Fig. 2 where the angular coordinate $\theta = 0$ is aligned to the semi-major axis in x positive body axis direction such that $y = 0$ then the equation is:

$$r(\theta) = \frac{a(1 - e^2)}{1 - e\cos\theta} \tag{8}$$

The idea is that we can set a desired value for the inner ellipse (of any layer) at 0°. We called that value $l(r(0°) = 1)$. As shown in Fig. 2, that value is how far the inner ellipse can be extended from the focus $f1$ to its edge. For each of the layers we have created, we consider a to be equal to half of the length of the chair's shape at that layer and we can also assign a value to the ellipse (the inner one) calculated by (8) at zero degrees, call it l. That value will be the maximum range at which we want the inner ellipse to be extended, it is adjustable and changes for each layer. Therefore, by having the values of a and $r(0°)$ it is now possible to calculate b as follows:

$$l \equiv r(0°) \Rightarrow l = \frac{a(1 - e^2)}{1 - e\cos(0)} \Rightarrow e = \frac{l}{a} - 1 \tag{9}$$

From equation (2) we know that:

$$\begin{aligned} e = \sqrt{1 - \frac{b^2}{a^2}} &\overset{(4)}{\Rightarrow} \frac{l}{a} - 1 = \sqrt{1 - \frac{b^2}{a^2}} \\ \Rightarrow l^2 - a^2 - \frac{2l}{a} + 1 &= 1 - \frac{b^2}{a^2} \Rightarrow b^2 = 2la - l^2 \\ \Rightarrow \qquad\qquad & b = \sqrt{2al - l^2} \end{aligned} \tag{10}$$

Concerning the rate at which the ellipses grow according to the speed and turn of the wheelchair, we use two coefficients respectively, again a pair for each layer. With the help of those coefficients we calculate the biases as following:

$$speed_{bias} = speed_{actual} \cdot speed_{coeff} \tag{11}$$

$$turn_{bias} = turn_{actual} \cdot turn_{coeff} \tag{12}$$

Those biases are then added to the a and b of the ellipses that belong to the layer at which the obstacle was detected.

$$a_{adjusted} = a + speed_{bias} \tag{13}$$

$$b_{adjusted} = b + turn_{bias} \tag{14}$$

From these new values, we can now calculate the value of the ellipses (inner and outer) at the angle the obstacle was detected from (8) (i.e calculate the distance of p and R from $f1$ in Fig. 2). For this, we also need the value of the eccentricity that corresponds to those ellipses. Since that quantity is under a square root, we should take into account this limitation.

We consider the speed coefficient to be equal to the inverse of the friction (also adjustable), so we only need to tune the turn coefficient in a way that it does not conflict with the aforementioned calculation. So:

$$\begin{aligned} & 1 - \frac{b^2_{adjusted}}{a^2_{adjusted}} \geq 0 \Rightarrow \frac{b^2_{adjusted}}{a^2_{adjusted}} \leq 1 \\ & \Rightarrow \quad b^2_{adjusted} - a^2_{adjusted} \leq 0 \\ & \overset{(13),(14)}{\Rightarrow} (b + turn_{bias})^2 - (a + speed_{bias})^2 \leq 0 \\ & \Rightarrow (b + turn_{bias} - a - speed_{bias})(b + turn_{bias} + \\ & a + speed_{bias}) \leq 0 \\ & \Rightarrow b + turn_{bias} - a - speed_{bias} \leq 0 \\ & \Rightarrow a - b \geq turn_{bias} - speed_{bias} \end{aligned} \tag{15}$$

Let:

$$\begin{aligned} & A = turn_{bias} - speed_{bias} \overset{(11),(12)}{\Rightarrow} \\ & A = turn_{actual} \cdot turn_{coeff} - speed_{actual} \cdot speed_{coeff} \end{aligned} \tag{16}$$

The speed and turn take values in [-1, 1] in the simulation environment. Since we only care about the magnitudes and not the direction and since the coefficients take values in [0, 1], we choose to remap the absolute values of the speed and turn in [1, 10]. Hence, the maximum value of A will appear when $turn_{actual} = 10$ and $speed_{actual} = 1$. In this case:

$$A_{max} = 10 \cdot turn_{coeff} - speed_{coeff} \tag{17}$$

Also, we consider the turn coefficient to be a function of the speed coefficient, such as:

$$turn_{coeff} = \frac{speed_{coeff}}{x} \tag{18}$$

From (15) and (17) we get:

$$\begin{aligned} a - b &\geq 10 \cdot turn_{coeff} - speed_{coeff} \\ \Rightarrow a - b &\geq \frac{10 \cdot speed_{coeff}}{x} - speed_{coeff} \\ \overset{k=a-b}{\Rightarrow} x(k + speec_{coeff}) &\geq 10 \cdot speed_{coeff} \\ x &\geq \frac{10 \cdot speed_{coeff}}{k + speed_{coeff}} \end{aligned} \tag{19}$$

Any value for x that obeys relation (19) is acceptable. From that we can adjust it by trial and error according to our model's behavior (or later on by a machine learning algorithm along with other parameters).

III. Implementation of the Simulation

In the case of 3D collision avoidance a camera sensor is used to detect the obstacles. By getting the x, y and z coordinates (the origin $O(0, 0, 0)$ is considered to be the sensor itself) of each point in the image, it is possible to determine the respective polar coordinates and heights of those points. According to its height, each point belongs to one of the three layers that have already been defined. Again, working with the right and the left quadrant of the frontal view, the nearest obstacle in each one is detected, but this time for each layer as well.

For the sake of this simulation, a camera sensor is needed. The type of sensor that is used to collect the needed information regarding the obstacles in the 3D environment is the Microsoft Kinect sensor (emulated in the Gazebo simulator), which can provide us with a depth map of the captured image. From that, it is possible to infer the position of the nearest obstacles, their distances from the sensor (and thus the distance from any desired point of the wheelchair, through basic geometry) and their height, which is necessary to classify the obstacles in one of the aforementioned layers.

The input for the simulation is provided through a mock setup using the manufacturers own system (Fig. 5). The setup consists of a joystick controller (a DX2-REM550/551 Advanced Joystick Remote model) provided by Dynamic Controls, a proprietary module allowing for communication with the controller, and a USB-to-TTL Serial interface connected to a Linux machine running ROS Jade and the Gazebo simulation environment.

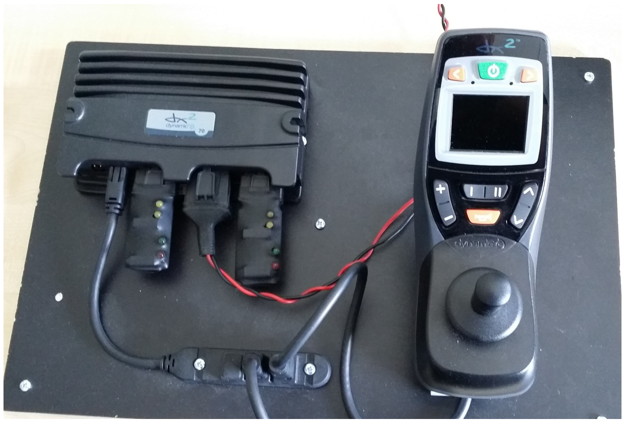

Fig. 5. Hardware Joystick Controller

The system is a hardware-in-the-loop (HIL) simulation which incorporates a real joystick controller and provides data to the hardware motor loads as well as to the simulated environment. Robot Operating System (ROS) is the robotic software framework used for development and integration of the hardware, collision avoidance system and the simulated environment. The computation graph in Fig. 4 illustrates these processes.

The simulated environment implemented in Gazebo presents a maze with obstacles placed in configurations of varying degrees of difficulty (Fig. 6).

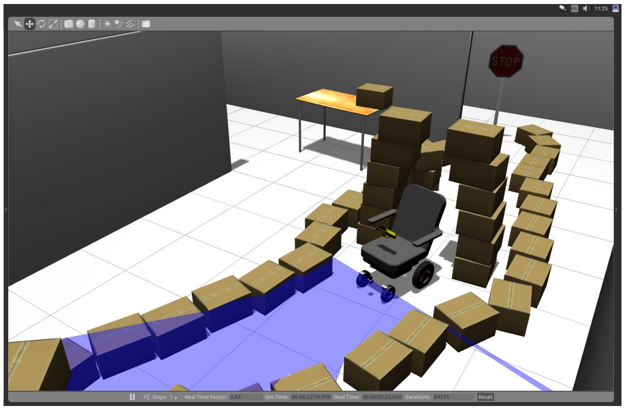

Fig. 6. Simulated wheelchair scenario in Gazebo

The 3D wheelchair model used in the simulation has been adapted from Argallabs smart wheelchair project [16]. The original model developed by Abhishek Patil [17] has been modified from a front-wheel drive to a rear-wheel drive wheelchair model using the Universal Robotic Description Format, a ROS based XML file format. The dimensions of the wheelchair have been modified to reflect the specifications of our in-house real-world wheelchair.

Visualizing the behaviour of the system is provided through RViz, a ROS module that allows monitoring of sensor information. Fig. 7 presents the online feedback from the simulation. The transparent elliptical cylinders represent the different layers being considered with their respective dimensions, and the intersection points of the closest obstacles with the edges of the inner and outer ellipse.

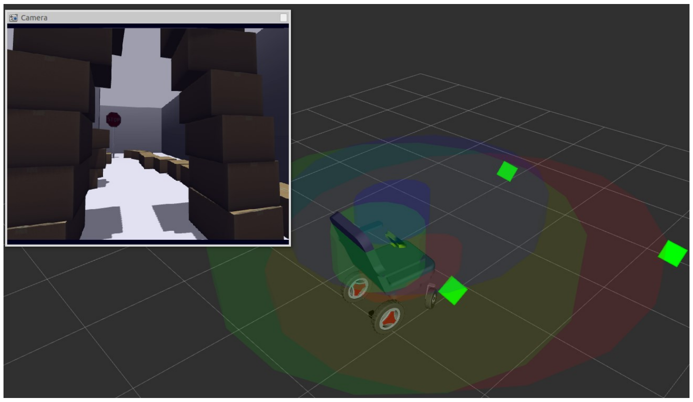

Fig. 7. Visualization of the 3 layers (head=blue, body=green, legs=red) in RViz

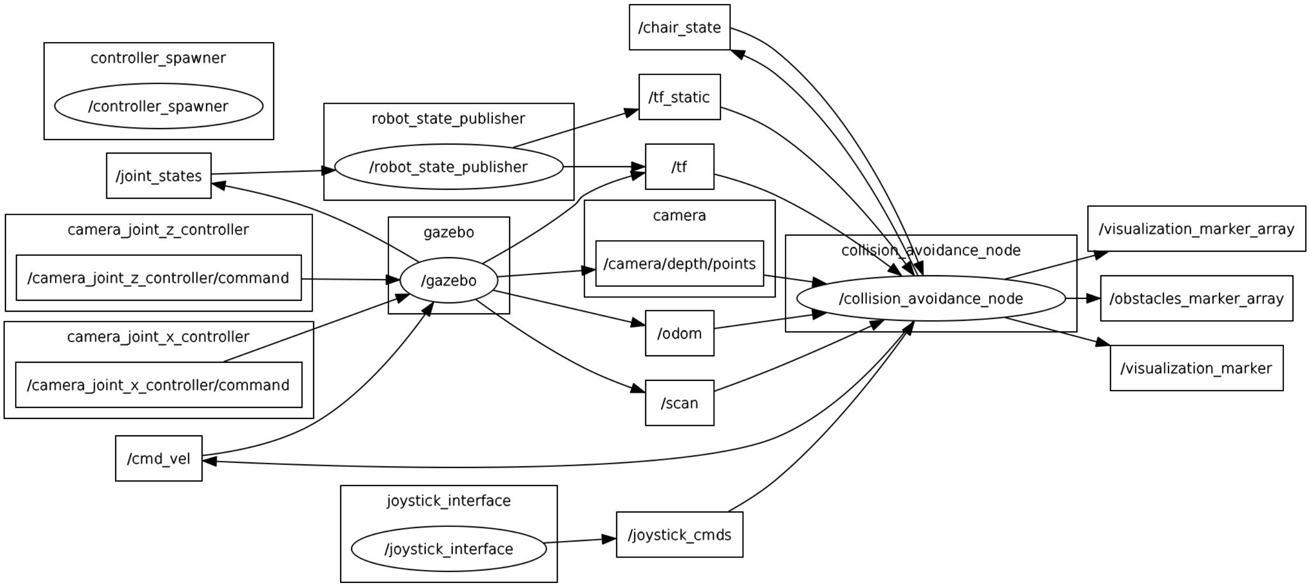

Fig. 4. ROS Computation Graph

IV. Results

Using the joystick controller along with the visual feedback from the simulated environment and the first-person visual feedback from the camera sensor, users drive the wheelchair without enabling the collision avoidance system, and, afterwards with the collision avoidance enabled. The recorded information from these experiments is presented in Table I. The users testing our system experienced a significant reduction in the number of collisions with the environment when they were assisted by the 3D collision avoidance algorithm, although the course times were slightly longer the average by an average of 42.8% the number of collisions were reduced by 91.7% and the number of failed to complete course reduced to zero. All collisions that occurred when the collision avoidance was activated happened outside of the sensor angular range, where the rear and side of the platform collided with obstacles after the front had passed by.

V. Conclusions

We have shown a method of simulating a cluttered environment for evaluating 3D PWC collision avoidance algorithms and this development has integrated a standard PWC control system into a real-time simulated environment, such that, as the user operates the PWC joystick the data from the system bus is then taken immediately into the simulation and the model responds accordingly.

When we evaluated our adaption of the 2D collision avoidance to 3D it was clear that the participants testing our system had a significantly reduced number of collisions with the environment and a zero failed to complete. Although there was a modest increase in course duration a future extension to the algorithm would be to reduce the difference in time by predicting the chair trajectory and experimenting with different sensors and configurations.

TABLE I
Participant Evaluation of 3D Collision Avoidance Simulation

ID	PWC experience	Run	Assistance(A)		Non-Assisted(NA)		% $\frac{t_A - t_{NA}}{t_{NA}}$
			Time(t)	*Collision No.*	*Time(t)*	*Collision No.*	
A	Proficient	1	52.4	3	32.9	1	59.4
		2	39.6	0	N/C	1	-
		3	41.3	0	36.9	3	11.9
		4	48.0	0	30.0	0	59.7
		mean/σ	45.3/6.0	0.8/1.5	25.0/16.9	1.3/1.3	43.7/22.5
B	Novice	1	42.3	0	27.97	3	51.1
		2	38.5	0	34.0	2	13.5
		3	41.6	0	29.9	3	43.2
		4	43.7	0	27.1	2	61.2
		mean/σ	41.5/2.2	0/0	29.7/3.0	2.5/0.6	42.2/17.8
C	Novice	1	46.9	0	65.1	4	-27.9
		2	40.9	0	N/C	2	-
		3	47.1	0	N/C	3	-
		4	41.1	0	29.7	2	38.2
		mean/σ	44.0/3.5	0/0	47.4/25.0	2.8/1.0	5.2/33.0
D	Novice	1	42.1	0	27.7	3	52.0
		2	40.6	0	28.1	1	44.4
		3	41.6	0	21.1	1	97.3
		4	39.8	0	23.5	1	69.8
		mean/σ	41.1/1.0	0/0	25.1/3.4	1.5/1	65.9/20.4
E	Novice	1	45.1	0	26.6	0	69.5
		2	44.8	0	30.5	2	46.8
		3	46.5	0	32.1	1	44.9
		4	46.7	0	27.6	1	69.4
		mean/σ	45.8/1.0	0/0	29.2/2.6	1/0.8	57.7/11.8
F	Novice	1	44.8	1	37.5	4	19.5
		2	45.3	0	32.2	2	40.9
		3	41.1	0	37.2	2	10.4
		4	44.2	0	35.7	4	23.8
		mean/σ	43.8/1.9	0.3/0.5	35.6/2.4	3/1.2	23.7/11.1

[*] N/C = Not completed

Spatial awareness inadequacies, which caused the three collision when the assistance was on, could be partially addressed by employing additional sensors such as 3D LIDARs or other multi-modal sensor arrays; however this may come at the cost of increased memory and computational requirements. Therefore careful consideration must be undertaken to develop suitable sensors to provide full environmental coverage to ensure a collision free path.

Additionally this development could be used to expand the state-of-the-art of the PWC simulator development to one which is able to mount a users standard PWC to a tiling platform. The 3D collision avoidance work also needs to be extended to detect drop kerbs and slopes.

Acknowledgment

This work was supported by the Assistive Devices for empowering disAbled People through robotic Technologies (ADAPT) project. ADAPT was selected for funding by the INTERREG VA France (Channel) England Programme which is co-financed by the European Regional Development Fund (ERDF). The European Regional Development Fund (ERDF) is one of the main financial instruments of the European Unions (EU) cohesion policy.

A Wellcome Foundation funded project; A Synergetic Adaptive non-intrusive Navigation Assistance System for empowering the disabled, elderly and infirm powered wheelchair users (SANAS).

References

[1] T. Rofer, C. Mandel and T. Laue. Controlling an automated wheelchair via joystick/head-joystick supported by smart driving assistance. Presented at *Rehabilitation Robotics*, 2009. ICORR 2009. IEEE International Conference On. 2009.

[2] D. A. Sanders and N. Bausch. Improving steering of a powered wheelchair using an expert system to interpret hand tremor. Presented at International Conference on Intelligent Robotics and Applications. 2015.

[3] M. Bailey et al,“Development of vision-based navigation for a robotic wheelchair,” in *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference* On, 2007, pp. 951-957.

[4] C. Urdiales et al, “A new multi-criteria optimization strategy for shared control in wheelchair assisted navigation,” *Autonomous Robots*, vol. 30, pp. 179-197, 2011.

[5] O. Khatib, “Real-Time Obstacle Avoidance for Manipulators and Mobile Robots,” *The International Journal of Robotics Research*, vol. 5, pp. 90-98, March 01, 1986.

[6] J. Borenstein and Y. Koren, “The vector field histogram-fast obstacle avoidance for mobile robots,” Robotics and Automation, IEEE Transactions On, vol. 7, pp. 278-288, 1991.

[7] H. Soh and Y. Demiris, “Towards early mobility independence: an intelligent paediatric wheelchair with case studies,” 2014.

[8] D. Fox, W. Burgard and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics & Automation Magazine*, vol. 4, pp. 23-33, 1997.

[9] P. Inigo-Blasco et al, “The shared control dynamic window approach for non-holonomic semi-autonomous robots,” in ISR/Robotik 2014; 41st *International Symposium on Robotics; Proceedings Of*, 2014, pp. 1-6.

[10] T. Pithon et al. Wheelchair simulators: A review. *Technology and Disability* 21(1, 2), pp. 1-10. 2009.

[11] M. Quigley et al. ROS: An open-source robot operating system. Presented at ICRA Workshop on Open Source Software. 2009.

[12] V. Canoz et al. Embedded hardware for closing the gap between research and industry in the assistive powered wheelchair market. Presented at 2016 IEEE/SICE International Symposium on System Integration (SII). 2016, . DOI: 10.1109/SII.2016.7843983.

[13] M. Henderson et al, “Powered wheelchair platform for assistive technology development,” in 2014 *Fifth International Conference on Emerging Security Technologies (EST)*, 2014, pp. 52-56.

[14] P. F. Muir and C. P. Neuman, “Kinematic modeling of wheeled mobile robots,” J. Robot. Syst, vol. 4, pp. 281-340, 1987.

[15] M. Gillham and G. Howells. A dynamic localized adjustable force field method for real-time assistive non-holonomic mobile robotics. *International Journal of Advanced Robotic Systems* 12(10), pp. 147. 2015.

[16] M. Derry and B. Argall. Automated doorway detection for assistive shared-control wheelchairs. Presented at Robotics and Automation (ICRA), 2013 IEEE International Conference On. 2013.

[17] *Nuric Wheelchair Model*. Abhishek Patil (Dec 9 2016). https://github.com/patilnabhi/nuric_wheelchair_model_02 Accessed 5th May 2017