COMPARATIVE STUDY BETWEEN A VOICE COMMAND INTERFACE AND A CHIN JOYSTICK FOR CONTROLLING A MOTORIZED WHEELCHAIR

Alexandre Aragão Souza Coelho[1], Patrícia Soares Micheletto[2] and Walter de Britto Vidal Filho[1]
[1]Department of Mechanical Engineering, University of Brasília, Brasília, Brazil.
aragao__souza@hotmail.com, wbritto@unb.br
[2]Federal University of Santa Catarina, Florianópolis, Brazil
pa.220@hotmail.com

ABSTRACT

Motorized wheelchair control interface technologies have been developed to enhance the quality of life for individuals with severe motor impairments. While various types of interfaces have been created over time, invasive systems remain the most commonly used today. This research focuses on wheelchair users who lack movement in both their upper and lower limbs. The objective of this article is to contribute to the advancement of motorized wheelchair control interfaces for quadriplegic individuals. To identify key parameters for the development of improved interfaces, a comparative study was conducted analyzing two specific technologies: the chin joystick and voice control. These two interfaces were designed and assembled for testing; the chin joystick, which is invasive, is widely used, while the voice control interface is non-invasive and less commonly adopted. A software simulator was implemented to assess factors such as ease of use, learning curve, and user adaptation to the interfaces. A systematic comparison method was developed and employed, as discussed in this article. The results, including test data and analyses of the advantages and disadvantages of each interface, are presented in graphical form. The study concluded that, despite being non-invasive and easier to learn, the voice command interface performed less effectively in the developed tests compared to the invasive chin joystick interface.

KEYWORDS: *Control interface, wheelchair, voice control, chin joystick*

I. INTRODUCTION

Assistive technologies have played a crucial role in enhancing the quality of life for individuals with motor disabilities. According to the Brazilian Institute of Geography and Statistics (IBGE), 6.5% of the Brazilian population aged two years and older have physical disabilities affecting their upper and/or lower limbs, accounting for approximately 13.8 million people. The significant proportion of individuals with physical disabilities in the country underscores the importance of developing accessibility technologies and policies. [11].

The American Spinal Cord Injury Association (ASIA) developed a classification system for assessing patients' neurological levels based on the location and severity of their injuries. According to this system, patients classified as NN C1-C4 are those who can perform limited or unrestricted movements using only their neck and head. For this group, assistive technologies have been developed and refined to enable greater freedom of movement. Technologies incorporating interactive interfaces, such as speech recognition and neck movement, can be adapted to suit the user’s abilities. However, when implementing user interfaces, it is essential not only to consider technological feasibility but also to prioritize usability and efficiency. The goal is to design interfaces that provide greater comfort, are intuitive, and require minimal effort to use. This article presents a comparative study of two interfaces, aiming to establish parameters for improving motorized wheelchair control for individuals with quadriplegia.

This article is organized as follows: The first section provides the introduction, outlining the context and objectives of the research. The second section presents a literature review, detailing the historical development of the primary interfaces currently in use, as well as relevant studies on the topic. The third and fourth sections describe the design and assembly of the two interfaces that will be compared. The fifth section introduces the simulator developed to test and evaluate the interfaces. The sixth section outlines the methodology used for interface evaluation. In the seventh section, the results are presented and discussed. The article concludes with the final section summarizing the findings, followed by the references.

II. LITERATURE REVIEW

Motorized wheelchairs have existed for many years, with the earliest patent traced to North America (US1348568A), dated 1919. This model featured two large motorized wheels and two smaller wheels, one at the back and one at the front (Fig. 1), with direction controlled by a lever. However, this device was not designed to assist individuals with disabilities, but rather as an electric propulsion vehicle, and is therefore not considered the first motorized wheelchair for people with disabilities. In 1948, a patent (US2495573A) was granted for a device incorporating an electric motor into a conventional wheelchair specifically for use by individuals with motor impairments. This model also used levers for control (Fig. 2), with a traction wheel positioned behind the chair to provide propulsion and steer the direction. [7].

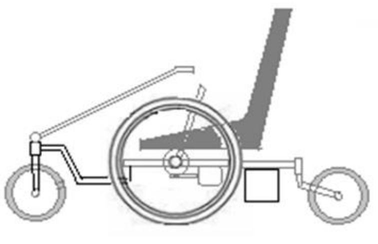


Figure 1. Motorized chair developed in 1919

Figure 2. Motorized chair adapted in 1948

The directional control of motorized wheelchairs shifted from levers to joysticks in the 1950s, a development pioneered by George Klein [2]. Klein introduced a joystick with electrical contacts that controlled the activation of two motors on either side of the chair. For many years, joysticks or buttons remained the primary method for controlling motorized wheelchairs. However, for individuals with upper limb impairments, such as those with tetraplegia, the most commonly used control system is the sip-and-puff interface (Fig. 3a). This system typically features one or two ducts. With a single duct, users can issue four commands by either sipping or puffing air—strong or weak puffs, and strong or weak sips—enabling forward and backward movement as well as left and right turns. Several studies have examined this interface, including [8] and [12]. In [14], a dual-duct sip-and-puff interface was compared to a chin-operated joystick (Fig. 3). However, most sip-and-puff interfaces do not provide precise speed control. In [14] a speed ramp was implemented via software.

Figure 3. Commercial interfaces: (a) Sip-and-puff (b) headrest (c) chin joystick [15]

Another widely used commercial interface is the headrest interface (Fig. 3b), which typically offers three control commands. Several studies have explored the use of head movements to control motorized wheelchairs. One of the earliest studies conducted in Brazil employed mechanical inclinometers with encoders to detect the tilt of the user's head in the desired direction of movement [10]. Over time, the use of accelerometers to measure head tilt has been incorporated, representing an advancement in the evolution of head-tilt interfaces. [9].

Some studies have focused on capturing other user movements, such as eye movements. The two primary techniques for this are Electrooculography (EOG) and Video-Oculography (VOG). EOG uses electrodes placed around the eyes to detect electrical signals from the muscles responsible for eye movement, allowing the determination of eye direction [3]. VOG, on the other hand, employs a camera to track and analyze eye movements to determine their direction [4]. A well-researched non-invasive interface is voice command control, in which the user speaks commands that are captured by a microphone, processed, and translated into motor activation signals [5].

Despite the wide variety of interfaces developed, market research indicates that the most commonly used are the sip-and-puff interface (Fig. 3a) and the chin joystick (Fig. 3c). After evaluating the available interfaces, the voice command interface was selected for this study due to its non-invasive nature. It was then compared with the widely used and commercially available chin joystick interface.

III. VOICE COMMAND INTERFACE

The voice command interface enables users to control the system by speaking specific phrases. Generally, this interface consists of three stages: an audio acquisition stage, a processing stage, and a stage for transmitting the identified command to the main system. Research on speech recognition has focused on utilizing neural network systems to recognize commands from any user without requiring individual adaptation of the system [13].

During the audio acquisition stage, a microphone captures environmental sounds. This can be executed in two modes: continuously, where the interface is always listening, or conditionally, where it activates upon receiving a specific command. Numerous programming libraries are available, implementing techniques that range from neural networks to fuzzy logic.

Once the audio is acquired, it undergoes interpretation in the processing step, where segments of the captured audio are compared with predefined valid expressions to extract commands or relevant information. This step can accommodate both complete sentences and keywords as valid inputs, with considerable variation depending on the implementation. Ultimately, the identified command must be transmitted to the active system connected to the interface to execute its functions, such as controlling the movement of a wheelchair.

The voice command interface presented in this article utilizes an Android smartphone for all the described steps. The device's microphone is employed to capture audio, while Google's “Speech to Text” service (an audio-to-text converter), available on Android smartphones, is used to convert the captured audio into text and compare it with keywords corresponding to valid commands. If a command is recognized, the data is sent via Bluetooth to an Arduino equipped with a Bluetooth module (Fig. 4). The platform used for programming the smartphone was App Inventor [1], maintained by the Massachusetts Institute of Technology (MIT).

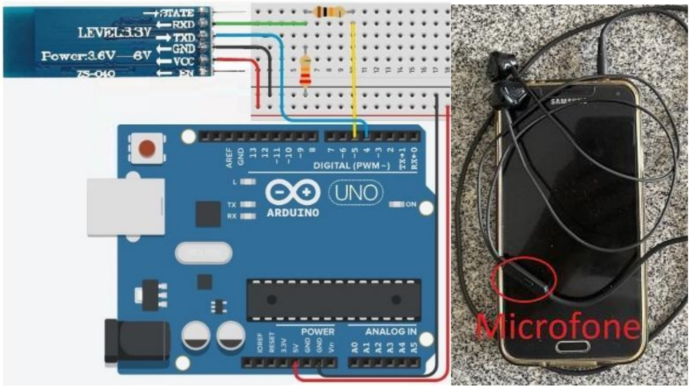


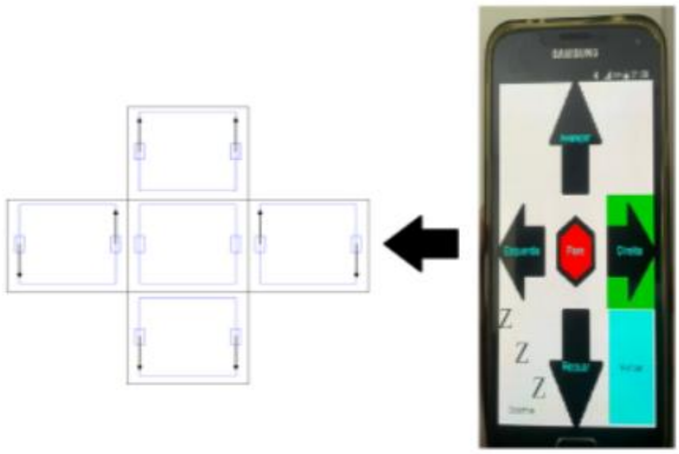
Figure 4. Commercial Bluetooth assembly (on the left) and cell phone used (on the right).

The developed system recognizes seven commands: five related to movement and two for usage management. Each command is interrupted only upon receiving a new command, which can complicate interface usability as the wheelchair's speed increases. Additionally, due to the high latency of this interface, it is unsuitable for short movements or situations that require rapid responses. The five movement commands—Forward, Backward, Left, Right, and Stop—and their corresponding effects are illustrated in Fig. 5. The remaining commands, Sleep and Aurora, are used to deactivate and activate normal operation, respectively, allowing the wheelchair user to converse without the risk of unintentionally triggering movement. For ease of use, the program displays only the recognized expressions on the screen and highlights the last command sent with a green glow, as shown in Fig. 5. To enhance the software's accuracy, the system not only accepts words similar to those designated as valid but also incorporates a comparison mechanism that searches for unique letter sequences—referred to as variables—that are not repeated in other commands. This approach helps maintain a low false-positive rate during normal usage, allowing the system to tolerate slight variations in user pronunciation and the inherent inaccuracies of speech recognition technology. The variables and their associated commands are presented in Table 1.

Table 1. Interface commands.

Command	Avançar (forward)	Pare (stop)	Direita (right)	Esquerda (left)	Recuar (backward)	Dorme (sleep)	Aurora
Variable	ava	par	dir	uer	rec	dor	aur

In Figure 5(a), the motor drives on the wheelchair are depicted alongside the corresponding commands displayed on the cellphone screen in Figure 5(b). When the user pronounces "forward," the system captures the sound through the cellphone's microphone, converts it into text, and the developed software compares it with the list of variables (Table 1). Once a match is found, a signal is sent to the control board (Arduino), which activates the two front motors at a fixed speed, as illustrated in Fig. 5(a). To prevent abrupt starting, an acceleration ramp must be implemented. When the voice command "left" is recognized, a signal is transmitted to the drive system, which operates one motor in the forward direction and the other in reverse, facilitating rotation around the central axis of the wheelchair. The user must issue the command "stop" to halt the rotation in the desired direction. A similar process occurs when the voice command "right" is recognized, resulting in a turn to the right. The rotation speed must be maintained at a low level to allow adequate time for the "stop" command to be recognized and for a signal to be sent to the controller to cease movement.



(a) (b)

Figure 5. a) Wheel movement b) Mobile phone screen with movements

IV. CHIN JOYSTICK INTERFACE

The joystick interface enables users to control the system by moving their head. This device features a lever positioned on the chin of the wheelchair user, allowing the user to tilt the joystick handle to indicate the desired direction. This design provides intuitive directional control, with the angle of inclination directly proportional to the desired speed in that direction.

The interface system is constructed from a metal frame that securely attaches the joystick to its tip, bringing it into contact with the user's chin. For the electronic assembly, the system utilizes a controller based on the Arduino Uno board, along with the joystick, as illustrated in Figure 6 below.

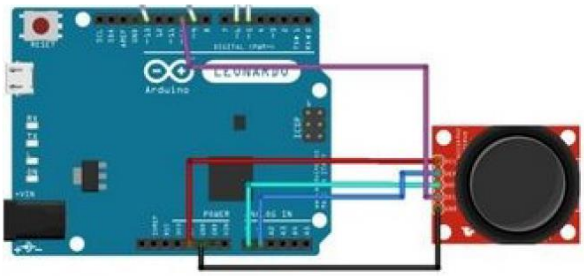


Figure 6. Joystick circuit.

The joystick features five pinouts: two for power supply (5V and ground), two for directional identification (X and Y axes), and one pin for click identification. It is equipped with two potentiometers, one for each axis, which convert the joystick's inclination into voltage. Consequently, using the data from the X and Y axes received by the Arduino, it is possible to determine the desired direction and speed.

The two potentiometers provide signals that are read by the Arduino's analog ports and converted into integer values ranging from 0 to 1023. These values are then interpreted to reference valid movements, as illustrated in Fig. 7. When the X axis is in the central position and the Y axis is tilted forward, both motors—one on each side of the chair—are driven forward at the same speed. If the X axis is tilted to the left while the Y axis is forward, the left motor operates at half the speed of the right motor, resulting in an arc trajectory to the left. A similar arc to the right occurs when the X axis is fully tilted to the right while the Y axis remains forward. When the inclinations of the X and Y axes are close to zero relative to the central position—specifically, when the acquired signal falls between 400 and 799—the motors are turned off. Due to the presence of reducers, the motors remain locked even on slightly inclined surfaces. When the Y axis is in the central position and the X axis is fully tilted to the left, the motors are activated to rotate the chair on its central axis to the left. Conversely, a full tilt to the right of the X axis while the Y axis remains in the central position results in a rightward rotation. When the Y axis is tilted backward, with an acquired signal between 0 and 399, and the X axis is in the central position, the motors are activated in reverse at the same speed. If the Y axis is tilted backward and the X axis is tilted to the left, the motors are also driven in reverse, with the left motor operating at half the speed of the right motor, resulting in a leftward arc.

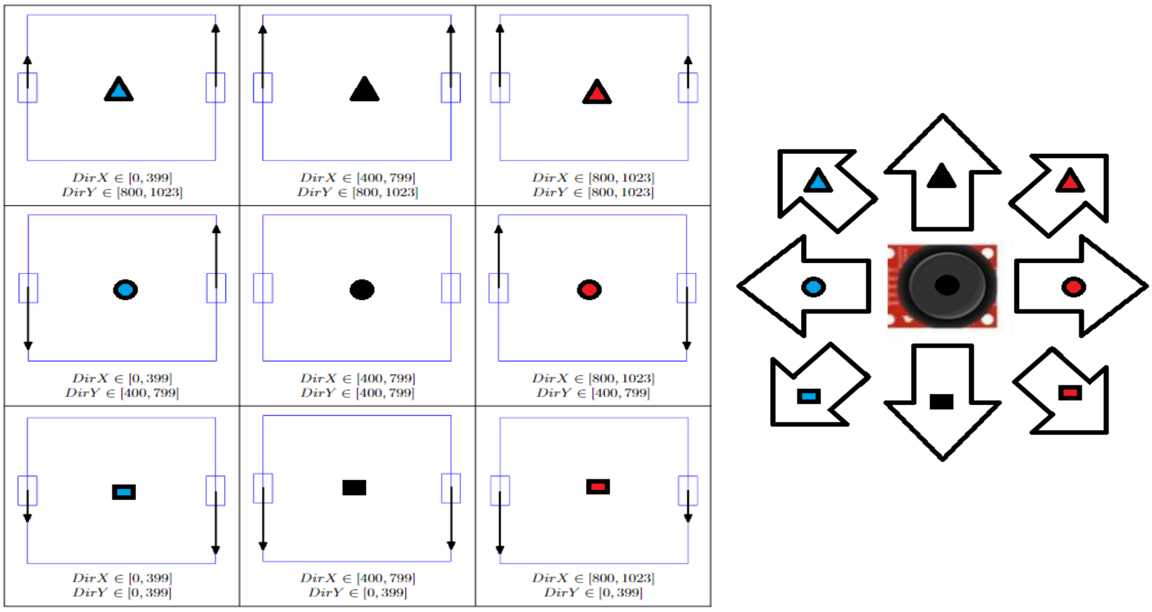


Figure 7. Relationship between joystick movement direction and simulated movement.

To improve user experience, an anatomically designed model was developed to replace the original joystick handle, as shown in Fig. 8. This new model enhances movement control by preventing the user's chin from slipping off the original structure, thereby reducing the risk of unintentional movement loss. The model was designed using SketchUp software and printed on a 3D printer using flexible filament to ensure user comfort and minimize the risk of injury.

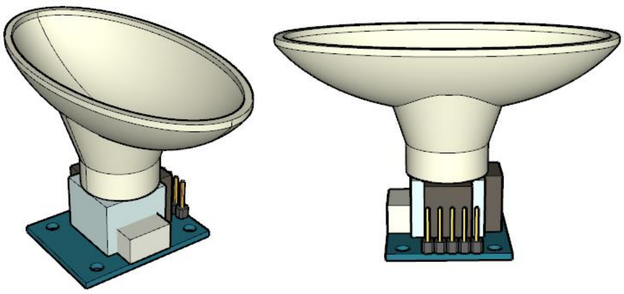


Figure 8. Model to support the chin.

V. VOICE COMMAND INTERFACE

The simulator was developed using MATLAB 2015a software, operating on the Windows 10 platform, and functions in conjunction with an Arduino Uno, which must be detected by the software for the program to execute correctly. Upon executing the main file (SimulatorUI.m), a window opens, allowing the user to select the simulation option (Area 1 of Fig. 9). The user then commands the program to generate code for Arduino (Area 2 of Fig. 9), which is tailored to the interface chosen (Area 4 of Fig. 9). This code is subsequently sent to the board using the Arduino IDE. The generated code comprises a ".ino" file for the Arduino IDE and a ".h" file containing the program, both located within the 'Simulador_Pro_Artigo_V1' folder.

The wheelchair representation is derived from the positions of the wheel centers, with both the maze and the initial position of the chair being fixed. The simulator must then establish a connection with the properly configured Arduino (Area 3 of Fig. 9) and, utilizing a master-slave architecture, request the command currently determined by the interface. Additionally, the communication channel must be specified in the simulator (Area 5 of Fig. 9).

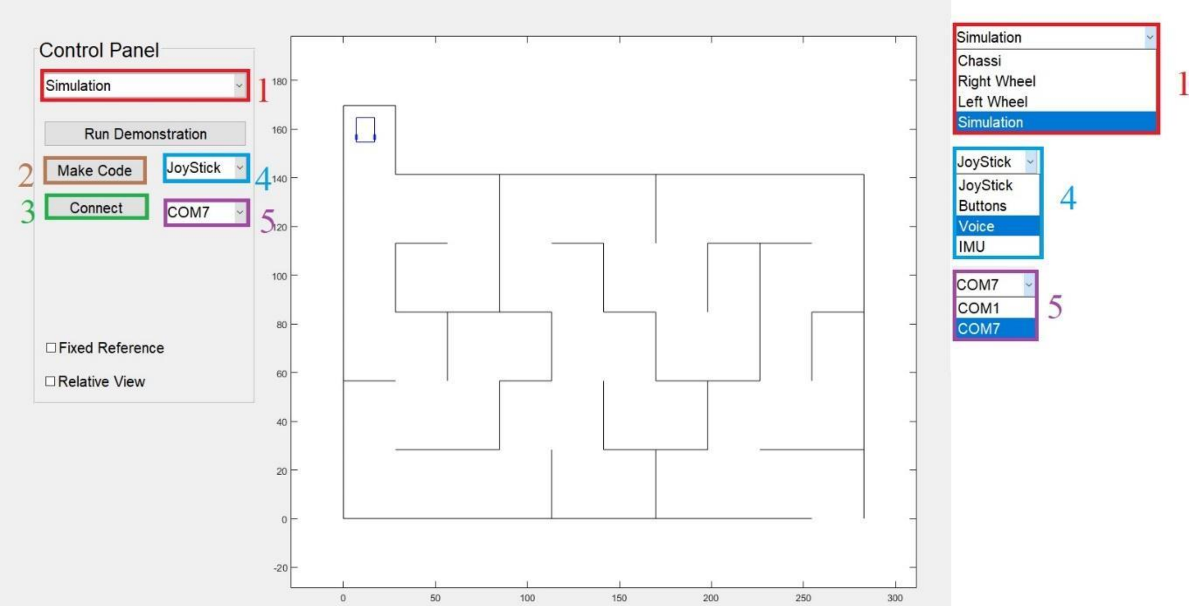


Figure 9. Simulation screen

The response is provided in a 'Read: E D' format, where E and D are numerical values representing the displacement of each wheel. The program subsequently simulates this displacement by adding the calculated values to the current positions of each wheel, provided both wheels exhibit non-zero movements in the same direction. Following this, the simulation incorporates a rotation of one wheel around the other: first, the right wheel rotates around the left wheel, and then the left wheel rotates around the newly established position of the right wheel, thereby completing the remaining linear displacement, as illustrated in Figure 10.

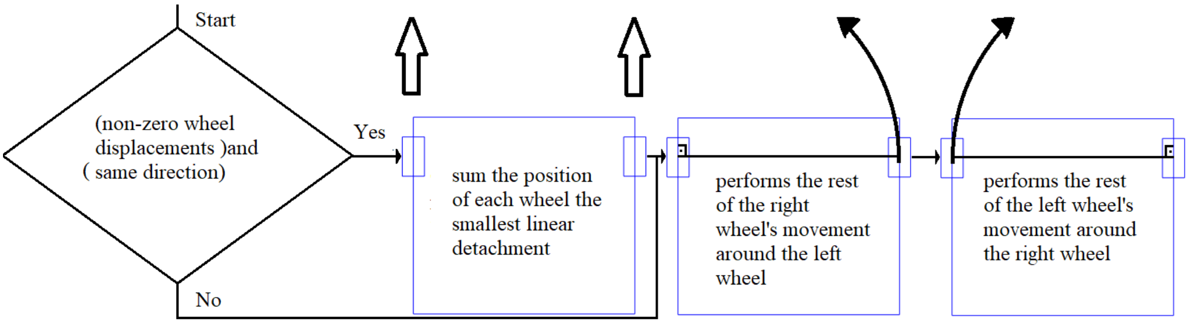


Figure 10. Simulated wheelchair movement algorithm

Communication between the microcontroller and the MATLAB software is facilitated through the use of the 'serial' object, along with the fprintf and fscanf functions. This approach allows for channel manipulation that is analogous to writing to files.

VI. Interface Evaluation Methodology

The voice command interface is among the least invasive and most comfortable options for wheelchair users; however, it is infrequently utilized in the motorized wheelchair market compared to other simpler and more invasive interfaces. The chosen evaluation methodology aims to elucidate this discrepancy. To conduct this evaluation and obtain parameters for the design of new interfaces, the voice command interface was compared with the chin joystick interface, which is the most commonly used commercially.

The developed method focused on assessing the following parameters: ease of use, task efficiency, ease of learning, and user adaptation. The systematic approach is divided into two analyses: quantitative and qualitative. Non-wheelchair user volunteers were selected and divided into two groups, with each group testing a different interface. A single group was not employed to test both interfaces due to potential fatigue, which could influence the outcomes of subsequent tests.

To facilitate the analysis, volunteers navigated a simulator to guide a wheelchair through a maze. Each group used one interface during this task, with standardized speed and time metrics recorded for each volunteer as they traversed the maze. Prior to the actual testing, each group underwent a familiarization stage followed by a learning stage.

During the familiarization stage, an instructor demonstrated the functionality of the interface and allowed the users to practice movements within the simulator to enhance their understanding and comfort with its operation, correlating commands with the software's visual cues. Once users confirmed their comprehension, the learning stage commenced, wherein participants navigated the maze twice to simulate real-world usage of the interface. In these initial stages, the time taken by each user was recorded, and during the learning stage, the number of errors was also documented. Errors were defined as collisions of the wheelchair with the maze walls or deviations from the intended path.

In the actual test, volunteers were instructed to navigate the maze in the opposite direction, with time taken and the number of errors once again measured. At the conclusion of the testing, participants completed a questionnaire to gather qualitative parameters related to their opinions and experiences.

VII. RESULTS AND DISCUSSION

The longest time recorded for navigating the maze with the chin joystick was shorter than the shortest time recorded for the same task using the voice command interface. Additionally, the voice command interface exhibited higher error rates both before and after the learning stage, as illustrated in Tables 2 and 3.

Regarding adaptation time, it is evident that the average adaptation time for chin joystick users is greater than that for voice command users. This suggests that, upon initial interaction, the voice command interface appears to be more intuitive for users, allowing them to feel more comfortable while continuing to use it. This observation was corroborated by responses from the questionnaire, in which participants rated their perceived learning difficulty on a scale from 0 to 10, with 10 indicating the highest level of difficulty.

Table 2. Results for chin joystick interface.

Users	Adaptation time	Average learning time	Average of Faults	Average Effective time	Average of Faults
1	3 min	3 min 19 sec	4.5	2 min 18 sec	1,5
2	1 min 22 sec	3 min 28 sec	1.5	3 min 20 sec	0,5
3	3 min 36 sec	2 min 59 sec	5	2 min 35 sec	1,5
4	2 min	3 min 40 sec	3	3 min 17 segc	0,5
Total Average	2 min 30 sec	3 min 21	3,5	3 min 24	1,25
Standard Deviation	51,9 sec	27,3 sec	1,8	29,4 sec	1,2

Table 3. Results for voice command interface.

Users	Adaptation time	Average learning time	Average of Faults	Average Effective time	Average of Faults
5	1 min 14 sec	9 min 58 sec	22	4 min 14 sec	9
6	1 min 16 sec	6 min 19 sec	16	6 min 05 sec	15

7	1 min 27 sec	8 min 37,5 sec	10	6 min 30 sec	6
8	1 min 22 sec	10 min 47,5 sec	17	7 min 11,5 sec	10,5
Total Average	1 min 19,8 sec	8 min 55,5 sec	16,3	6 min 0,1 sec	10,1
Standard Deviation	10,2 sec	2 min 28,1 sec	5,8	1 min 35,5 sec	4,0

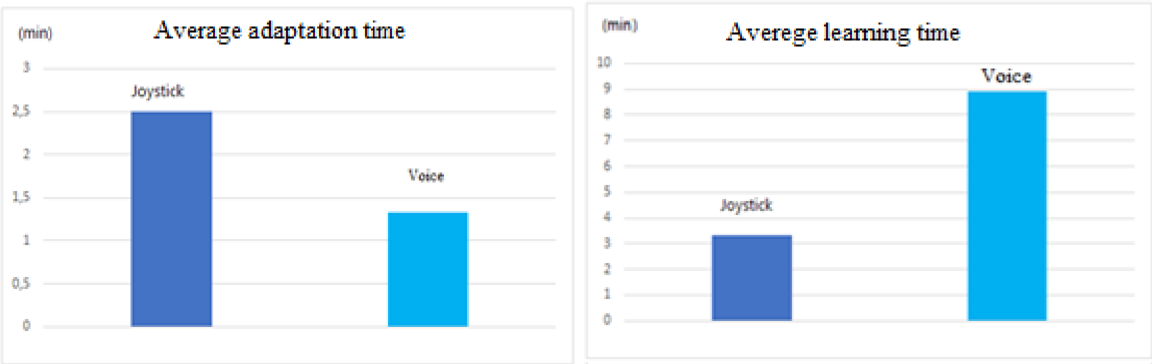


Figure 11. Graphs of average familiarization and learning time.

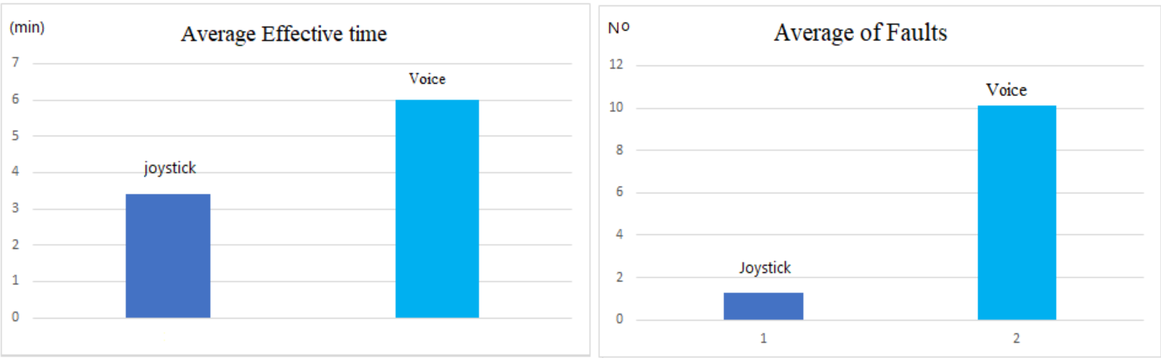


Figure 12. Comparative graph of the effective test

The average score assigned to the voice command interface was 0.25, while the chin joystick received an average score of 4. However, it is noted that delays in command execution pose challenges for users of the voice command interface, a difficulty not observed with the chin joystick, which demonstrated a significantly lower average learning time in comparison. This suggests that while the voice command interface may be more challenging to learn, it ultimately becomes easier to use. This observation is further supported by questionnaire responses, which yielded average scores of 6.75 for the voice command and 4 for the joystick.

Moreover, when comparing the average learning time with effective task completion time, an improvement in the use of the voice command interface is evident, decreasing from 8.91 minutes to 6 minutes, which represents a 32.7% enhancement. Nevertheless, additional usage tests are warranted to further assess the extent of user performance improvement with the voice command interface.

In terms of average failures during use, the voice command interface exhibited a higher standard deviation compared to the joystick, indicating that some users may find the voice command interface more intuitive. Despite the chin joystick interface demonstrating superior performance in terms of speed and consistency, it also resulted in greater discomfort for users, as indicated by questionnaire responses that produced average scores of 2 for the voice command interface and 6.25 for the chin joystick.

All volunteers using the voice command interface reported difficulties stemming from delays and the occurrence of false negatives. Additionally, two volunteers expressed discomfort with the need to speak loudly during the test, and one noted experiencing dryness in their throat afterward. A significant portion of the time spent by volunteers navigating the route with the voice command interface was attributed to error corrections and attempts to rotate the wheelchair in the desired direction. These tasks were rendered challenging due to inconsistent accuracy, characterized by multiple errors occurring within short time intervals, alongside the high latency observed.

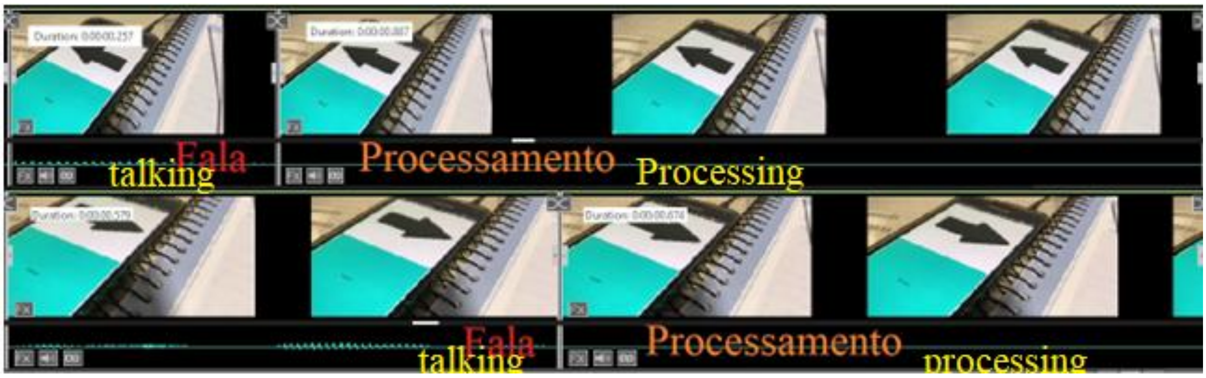


Figure 13. Measuring latency by processing recorded video.

To investigate latency, measurements were conducted (see Fig. 13) through video recordings capturing the speaking and processing times associated with the commands "Stop" (upper track in Fig. 13) and "Left" (lower track in Fig. 13). These measurements highlight a notable disadvantage of the voice command interface in comparison to other interfaces, specifically regarding the time required for verbalization. Even if processing were instantaneous, a delay would still exist between the user's decision to perform an action and the execution of that action, attributable to the time needed for pronunciation. Although this time can be adjusted by altering keywords, such modifications may render the interface less intuitive and, in stressful situations, potentially complicate usability.

Table 4. Latency segmented by cause.

Displays the delay in milliseconds and as a percentage of the measurement latency.

Command	**Measures (ms / %)**	**Minimum**	**Maximum**	**Medium**
Avançar (forward)	Latency	1073 (100%)	1933 (100%)	1378 (100%)
	Talking	514 (28%)	716 (57%)	626 (43%)
	Processing	500 (47%)	1355 (72%)	752 (53%)
Pare (stop)	Latency	1018 (100%)	1225 (100%)	1158 (100%)
	Talking	305 (26%)	380 (32%)	340 (29%)
	Processing	696 (68%)	906 (74%)	818 (71%)

Given that the primary drawback of the voice command interface is high latency, reducing processing delays could significantly enhance the performance of wheelchair users. As indicated in Table 4, even if the processing time were negligible, latency would still persist during the speaking duration, which constituted between 26% and 57% of the total latency, depending on the command. While altering keywords might shorten speaking time, it could also diminish the interface's usability by complicating the association between terms and their corresponding actions.

One potential solution to mitigate this issue is to combine the voice interface with another interface that does not incur significant delays. Since the only time-sensitive function is stopping—both to execute rotations to the desired extent and to halt forward or backward movements—implementing a faster command for this function is crucial.

Another approach to enhance the viability of the voice interface would involve modifying the program to recognize distance and angle values as part of the commands or to employ fixed values for each command. Integrating movement values into the commands could lead to increased latency; moreover, it might inconvenience users if success rates were low. For instance, if fixed movement values are utilized, such as in the case of a left rotation, issuing the command "Left" would result in the chair turning at a predetermined angle and subsequently stopping. If the movement increments are minimal, numerous commands would be required to achieve larger rotations. Conversely, if the increments are substantial, executing precise movements could become challenging.

Finally, the most immediate strategy to address latency would be to significantly reduce the rotation and translation speeds, thereby allowing latency-induced errors to be minimized. To further safeguard against accidental collisions arising from delays in executing the stop command, presence sensors should be installed at the front and rear of the wheelchair.

VIII. CONCLUSIONS

Two control interfaces for motorized wheelchairs were developed and compared: one utilizing a chin joystick and the other employing voice command via a cell phone. To evaluate and compare the performance of these interfaces, a simulator was created to observe the interactions of a group of volunteers in a controlled environment. Additionally, questionnaires were administered to qualitatively analyze the participants' experiences during the testing.

The results indicated that the chin joystick, despite being perceived as somewhat uncomfortable, demonstrated superior performance, characterized by lower error rates and reduced time required to navigate the predetermined route. In contrast, the voice command interface exhibited high latency, frequently necessitating rework to correct errors. Several solutions were proposed to mitigate latency issues and enhance the viability of this interface.

The chin joystick interface's prevalence in the market can be attributed to its greater efficiency in task execution and lower associated costs. When analyzing the design requirements for command interfaces, it was anticipated that ergonomics and ease of learning would emerge as the most critical parameters. Although these factors are indeed significant, the efficacy of the interface was determined to be more critical in the testing outcomes. While the joystick interface may present a steeper learning curve, its efficiency in use resulted in higher user satisfaction.

Conversely, the difficulties encountered in interpreting commands through the voice interface contributed to a sense of insecurity among users, leading to greater rejection of this option. Cost also emerged as a vital design requirement that should not be overlooked.

REFERENCES

[1]. MIT, (2022), *Appinventor* from: https://appinventor.mit.edu.
[2]. Burden, G.,(2021). *George J. Klein, Canada´s own Thomas Edison*, from: http://lifeasahuman.com/2011/arts-culture/history/george-j-klein-canadas-own-thomas-edison/
[3]. Barea, R. et al, (2003). “Electro-oculographic guidance of a wheelchair using eye movements codification.” *The international Journal of robotics Research*.
[4]. Cavalieri, D.C., (2007), *Interface homem-máquina para cadeira de rodas baseada no movimento ocular e piscada de olhos*. Master Thesis. Universidade Federal do Espírito Santo. Brazil.
[5]. Chiele, M.R., Zerbetto, A., (2010). “Desenvolvimento de uma cadeira de rodas controlada por voz”. In *XIII Congresso Brasileiro de Automática*. CBA, Brazil.
[6]. Duston, K., (1919). Electrically-propelled vehicle. US patent. 1348568A.
[7]. Duke S., (1948). Motor attachment for wheel chairs. US patent. 2495573A.
[8]. Ferreira, C.L.L., (2008), Interface de sopro e sucção para controle de cadeira de rodas. Master Thesis, Universidade Estadual de Londrina, Brazil.
[9]. Fusco, D.A., 2010. Acionamento de uma cadeira de rodas através de um acelerômetro bi-axial como inclinômetro. Undergraduate project. Universidade do Rio Grande do Sul, Brazil.
[10]. Getschko, N., Tannuri, E.A., Debenest, P.C., 2004,. “Desenvolvimento de um sensor de orientação angular inercial de baixo custo aplicado ao controle de uma cadeira de rodas elétrica.” *Revista Engenharia*, FAAP, nº 44, p23-32
[11]. Janone, L. e Almeida, P., 2021. *Brasil tem mais de 17 milhões de pessoas com deficiência, segundo IBGE*.from:https://www.cnnbrasil.com.br/noticias/brasil-tem-mais-de-17-milhoes-de-pessoas-com-deficiencia-segundo-ibge/.
[12]. Leôncio Júnior, A.P.L. et al., 2014, “Acionamento de uma cadeira de rodas pela percepção da intensidade de sopro e sucção, com motores de indução”. *XXIV Congresso Brasileiro de Engenharia*

Biomédica. Uberlândia.

[13]. Neto, J.A.O., Castro, M.A.A., Felix, L.B., 2010. "Reconhecimento de comandos de voz para o acionamento de cadeira de rodas", *XVIII Congresso Brasileiro de Automática*, Bonito-MS.

[14]. Pinto, F.F., 2016. *Interfaces de controle de cadeira de rodas para pessoas com tetraplegia*. Trabalho de Graduação em Engenharia de Controle e Automação, Faculdade de Tecnologia, Universidade de Brasília, Brasília, Brazil.

[15]. Trigueiros, P. e Ribeiro, F., 2012. "Vision-based Hand WheelChair Control". *12th International Conference on Autonomous Robot Systems and Competitions*, Guimarães, Portugal, pp 39-43.

Authors

Alexandre Aragão Souza Coelho - Graduated with a degree in Control and Automation Engineering from the University of Brasília, Brazil, in 2022. Enrolled in the master's program in Mechatronic Systems at the University of Brasília in 2023.

Patrícia Soares Micheletto – Undergraduate in Control and Automation Engineering in the University of Santa Catarina, Brazil

Walter de Britto Vidal Filho – Graduated in Mechanical Engineering in Federal University of Pernambuco, Brazil (1995), Master degree in Mechanical Engineering in PUC/RJ, Brazil (1997). Phd degree in Engineering with emphasis on mechatronics in University of São Paulo, Brazil (2003). Currently serving as a professor at University of Brasília.

.