Faculdade de Engenharia da Universidade do Porto

FEUP

Multimodal Interface for an Intelligent Wheelchair

Márcio Miguel Couto de Sousa

Project dissertation made in the scope of the
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major Automação

Supervisor: Prof. Dr. Luis Paulo Reis
Co-Supervisor: Prof. Rodrigo Braga

July 2008

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

MIEEC - MESTRADO INTEGRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES **2007/2008**

A Dissertação intitulada

"Interface Multimodal para Cadeira de Rodas Inteligente"

foi aprovada em provas realizadas em 16/Julho/2008

o júri

Presidente Professor Doutor José Manuel Martins Ferreira
Professor Associado da Faculdade de Engenharia da Universidade do Porto

Professor Doutor Luís Filipe de Seabra Lopes
Professor Auxiliar da Universidade de Aveiro

Professor Doutor Luis Paulo Gonçalves dos Reis
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.

Autor - Marcio Miguel Couto de Sousa

Márcio Miguel Couto de Sousa

Faculdade de Engenharia da Universidade do Porto

For my parents and Raquel

“In the face of uncertainty, the one thing we can be sure of is the value of certainty.”

Gregory Maguire, “Son of a Witch”

Abstract

With the rising concern about the needs of people with physical disabilities and with the aging of the population there is a major concern of creating electronic devices that may improve the life of the physically handicapped. The scientific progress in Artificial Intelligence, Computation, Multimedia and Robotics has given a window of opportunity to develop new solutions to aid the elderly and the handicapped persons, and improve their life quality.

One of these new solutions passes through the adaptation of electric wheelchairs so that they may have environment perception, more intelligent capabilities and may provide more adequate Human – Machine Interaction.

This dissertation explains the design of a multimodal interface for an intelligent wheelchair, describing its architecture, development and integration in the Intellwheels project in development at LIACC – Artificial Intelligence and Computer Science Laboratory of the University of Porto.

The main objective of the work was to create a user-friendly interface that allows the connection of the already developed input and control modules, enabling the wheelchair control through flexible input sequences of distinct types of inputs (voice, facial expressions, head movements, keyboard and, joystick).

The system created is capable of storing flexible associations, defined by the user, of input's sequences and corresponding output commands.

To attest the system efficiency and the wheelchair control achieved through the multimodal interface several tests were conducted. Tests were performed where the wheelchair was controlled using sequences of the available inputs in different kinds of environments (noise in the background, obstacles, etc.).

Resumo

Com o aumento da preocupação para com as pessoas com deficiências físicas e com o envelhecimento da população existe uma grande preocupação no desenvolvimento de sistemas electrónicos que possam melhorar a vida dos deficientes físicos. O progresso científico em Inteligência Artificial, Computação, Multimédia e Robótica trouxeram uma janela de oportunidade para desenvolver novas soluções para ajudar os idosos e as pessoas deficientes, e melhorar as suas vidas.

Uma destas novas soluções passa pela adaptação de cadeiras de rodas eléctricas para que tenham percepção do ambiente, maiores capacidades de inteligência e possam providenciar uma adequada interacção Homem-Máquina.

Esta dissertação explica o design de um interface multimodal para uma cadeira de rodas inteligente, descrevendo o desenvolvimento da sua arquitectura e a sua integração no projecto Intellwheels em desenvolvimento no LIACC - Laboratório de Inteligência Artificial e Ciência de Computadores da Universidade do Porto.

O principal objectivo deste trabalho foi de criar um interface amigável para o utilizador que permita a ligação dos já desenvolvidos módulos de entrada e controlo, e o controlo da cadeira de rodas através de sequências flexíveis de distintos tipos de entrada (voz, expressões faciais, movimentos da cabeça, teclado e, joystick).

O sistema criado é capaz de guardar associações flexíveis, definidas pelo utilizador, de sequências de entrada e os correspondentes comandos de saída.

Para comprovar a eficiência do sistema e o controlo da cadeira de rodas obtido através da interface multimodal desenvolvida foram conduzidos diversos testes. Foram realizadas experiências onde a cadeira de rodas foi controlada com as entradas disponíveis em diferentes tipos de ambiente (ruído de fundo, obstáculos, etc.).

Acknowledgements

I would like to be thankful my supervisor, Prof. Luis P. Reis, and also my co-supervisor, Prof. Rodrigo Braga, for all the support and guidance given to me in this project.

I'm also thankful to LIACC for the excellent work conditions and to FEUP for providing the means to make this project.

I also want to be thankful to my beloved and dear Raquel, for all the support, patience, love and sacrifice, given to me in this journey.
And finally I want to express my gratitude for my parents, for all the thrust and counselling and for always believing in me.

Index

Figure list

Table list

Abbreviations

List of abbreviations (alphabetically ordered)

DEEC	*Departamento de Engenharia Electrotécnica e de Computadores*
FEUP	*Faculdade de Engenharia da Universidade do Porto*
IP	Internet Protocol
LIACC	*Laboratório de Inteligência Artificial e Ciência de Computadores da Universidade do Porto*
HCI	Human Computer Interaction
IW	Intelligent Wheelchair
MMI	Multimodal Interface
OCR	Optical Character Recognition
OOP	Oriented Object Programming
TTS	Text to Speech
WIMP	Windows, Icons, Menus and Pointing Devices

Chapter 1

Introduction

The last decade has provided a large set of improvements within all sectors of our society, which have offered new opportunities and conditions that originated an increase of the population life expectancy.

With the aging of the population new problems arise, from health problems to physical disabilities is necessary to develop methods to ease and help this population group.

Principally focusing in the physical disabilities, one major solution is the use of wheelchairs to aid the mobility. But the traditional wheelchairs need to be manually propelled and, if the occupant has difficulties realizing this task he will be dependent of another individual to move him.

The use of an electrical wheelchair allows the user to control the chair movement through a joystick without external help, increasing the independence of the user, but regarding the user mobility needs it can be difficult to control the wheelchair only with the provided joystick.

In order to enhance and help the user with the control, intelligent wheelchairs are being researched, which main objective is to create a functional wheelchair with a pleasing construction. These types of wheelchairs provide mechanisms that have perception of the world where the chair is moving and provide improved control methods.

One solution to aid the control is the use of a multimodal interface which allows the use of several input methods to control the wheelchair, instead of the traditional joystick. This interface also permits that individuals that couldn't control the wheelchair through a joystick now can control through another input method.

This interface can provide an interaction between the wheelchair environment and the input method, so that at any instance the input information can be analyzed and checked if it's reliable, to assure the user safety.

1.1 - Motivation

Like it has been mentioned there is an increasing aging of the population. Combining this aging with the population sectors that have special mobility needs, it becomes necessary to develop methods to aid them in their daily life.

Aiming to help on this task becomes the obligation of computer scientist in the areas of artificial intelligence and robotics. One of the motivations for conducting this work was thus to create a system that can help and give a more independent life to any person with special mobility needs.

Beyond what has been said, personally, the biggest motivation for developing this project arises from the fact that my father is dependent from a traditional wheelchair to move, due to legs palsy. Because of this I see the several difficulties that any wheelchair dependent person has, and can say that with this project it's possible to minimize the control difficulties.

1.2 - Objectives

The main objective of this work is to develop a MultiModal Interface (MMI) to be integrated in the Intelligent Wheelchairs (IW) of the Intellwheels project under development at LIACC. This MMI must be able to:

- Provide user friendly characteristics;
- Allow the control of the IW through user-defined input sequences that may be associated with the desired output high-level commands;
- Design new input modules that can help the users in the control of the IW. These modules can implement voice, video or other type of commands.
- Allow the connection of the input modules developed (voice, facial expressions, joystick, keyboard and head movements);
- Show environment characteristics to the user.

The fulfilment of these objectives will lead to a very flexible user interface completely controlled by the user.

1.3 - Summary of contents

This thesis will be divided in six chapters, where this is the first chapter, and the reader can have a global insight over the developed work.

The second chapter will approach multimodal interfaces, and its state of art.

In the third and fourth chapter the Intellwheels project, where this interface is inserted, and the developed work will be described, respectively.

The fifth chapter presents the tests conducted and the results achieved.

Finally, the sixth chapter gives the final conclusions over this work, showing also perspectives of future work.

Chapter 2

Multimodal Interfaces

Generically an interface is an element that establishes a frontier between two entities. When an interface is used to assist in the Human-Computer Interaction it is called a user interface, being able to be graphical or command line based.

The traditional graphical user interfaces are based in the WIMP (Window, Icon, Menu, and Pointing device) paradigm, which uses the mouse and keyboard as physical input devices to interact with the interface, for example to access any information or accomplish any needed task.

An evolution to this paradigm and a way to create a more natural interaction with the user is the establishment of a multimodal interaction. This interaction contemplates a broader range of input devices such as video, voice, pen, etc, and so these interfaces are called Multimodal Interfaces.

A Multimodal Interface [1] "processes two or more user input modes – such as speech, pen, touch, manual gestures, gaze, and head and body movements – in a coordinated manner with multimedia system output. They are a new class of interfaces that aim to recognize naturally occurring forms of human language or behaviour, and that incorporate one or more recognition-based technologies (e.g., speech, pen, vision)".

In this chapter, first will be given a short overview of previous works in this field, and following will be approached several aspects that must be considered in the development of this type of projects.

2.1 – Multimodal Interfaces Projects

The improvements in computing have brought new developments to the interfaces design, and with this the computer graphical interfaces available to a wider range of users.

One new type of interface is the MMI, which are represented, in table 1, several works in this field.

Table 2.1 - MMI projects

Name	Inputs	Description
QuickSet [2]	Pen, voice	Interaction with distributed applications : agent simulation, map based tasks, etc.
MATCH [3]	Pen, speech	Mobile MMI to subway and restaurant information in New York.
Multimodal Interface for an Interactive Simulated Vascular Reconstruction System [4]	Hand gestures, speech, manipulation of virtual 3D objects	Simulation of a vascular reconstruction procedure and visualization of the results.
Video based interface to textual information for the visually impaired [5]	Video, optical character recognition (OCR), text-to-speech (TTS)	Access of textual information present in not "standard" sources, such as magazines or newspapers, for the visually impaired
A Multimodal 3D Healthcare Communication System [6]	Gesture recognition	Communication system between patients and doctors through gestures

2.2 – General aspects

Considering the purpose of this work the main aspects to consider should be the adaptability to users, usability and safety. These factors are determinative in a MMI design, where all subjective characteristics, like user satisfaction and cognitive learning, for example, and user interaction depend on them.

2.2.1 – Adaptability to the users

One of the main aspects of this project is the ability to be usable and understandable by any person, independently from his informatics knowledge and cognition.

This is important because it is necessary that the interface can be accessible to any person.

2.2.2 – Usability

The main factor in the interface is the multimodal interaction. This multimodality brings several improvements to the interface, since a wider range of output control options to a complementarily between inputs.

The output control is achieved by the combination of several inputs, only being limited by the total number of inputs. As the interaction between the inputs can differ depending on the environment, this multimodality achieves a complementarily that when any input become less recognizable, it can be compensated by another, but this must be done being in mind the interface accessibility [6].

2.2.3 - Accessibility

Having in account the project enclosure, the multimodality must enable the access to any user, despite his deficiency.

This shows the MMI accessibility importance, so that if a user as any deficiency that suppress the use of one input, there is another that compensates this handicap.

2.3 – Functionality

Since this is a Multimodal Interface, it's necessary that this project allows a transparent and intuitive control of the Wheelchair and also a flawless input interaction. This is achieved by the understanding of the user and inputs interaction.

2.3.1 Inputs interaction

The inputs interaction is one of the key points of a multimodal interface, since it will be this interaction that will produce the desired output to the user.

It's necessary that exist a support for integrating any kind of inputs like: video, speech, handwriting, gestures, etc, but also this support must contemplate a robust processing of the inputs to fully recognize the user intentions.

2.3.2 User interaction

The user interaction is another key point of a multimodal interface, if not the most important, so that the user can have and pleasant experience with the interface.

It's necessary to consistently verify the disposition of every component of the interface so that the visual information and content can be easily access. And also assure an intuitive interaction with the system, regarding the information about the available actions and how the user can interact with them.

Other factor is the interface output, which is divided in two parts: the execution of the inputs interpretation result and all the information given to the user about the system state.

2.4 - Human-Computer Interaction

In the past years the advances in computing have created several new fields of study. One of them is the Human-Computer Interaction (HCI).

HCI focuses not only in the interface design but also in the established interaction, trying to make the user interaction to the interface as natural as possible.

In the design of any interface is necessary to have a concern in the graphical disposition and on the interface content, so that the user can execute any task supported by the interface can be intuitive for the user the outcome of any action he makes [7]. For example if the interface represents a "Start and Stop button" application, the buttons must be distinguishable and since its nature, intuitively, the user knows the what each button does, so it is necessary that their function don't differ from their visual – Start button stops and Stop button starts.

2.5 - Conclusions

This chapter presented a brief overview of the concept of multi-modal interface and some of the projects in development around this concept in the world.

The multi-modal interface concept will be applied in the Intellwheels project by developing a truly flexible multimodal interface that lets the user define which sequences of distinct inputs correspond to which of the available high-level output commands.

Chapter 3

Intellwheels Project

This Multimodal Interface is inserted in the Intellwheels Project, which main objective is to provide an intelligent wheelchair framework to aid any person with special mobility needs.

This project encloses the prototype of an intelligent wheelchair, since its hardware structure to all software needed to provide a flawless control and experience to the user.

In this chapter it will be described the hardware and software design and also the full project architecture, with special focus in the multimodal interface.

3.1 Hardware

The present hardware structure is constituted by two electrical wheelchairs prototypes.

The first wheelchair is based in the Powertec model from Sunrise [8], and has the following characteristics:

- Independent rear traction, provided by two electrical motors;
- Two frontal free wheels;
- Two 12V/45Ah batteries;
- Power module;
- Control joystick.

The second wheelchair is based in the Evolution model from Vassilli [9], and has the following characteristics:

- Independent rear traction, provided by two 160W electrical motors;
- Two frontal free wheels;
- Two 40A batteries:
- Power module;
- Control joystick.

These wheelchairs are shown in figure 3.1.

Figure 3.1 - Picture of both wheelchairs

The first wheelchair served as a development prototype for the hardware architecture. This architecture was created with the objective of being flexible and generic, so that it doesn't imply considerable modifications in the wheelchair structure [10].

In figure 3.2 can be seen the hardware architecture of the wheelchairs, showing the connection of the several components.

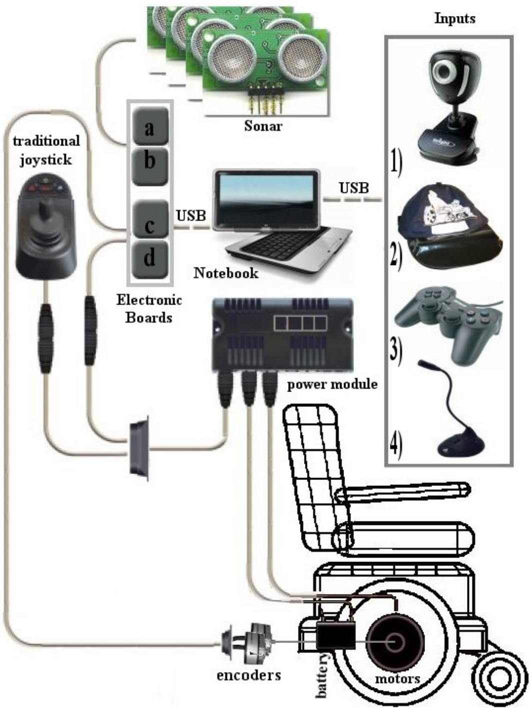


Figure 3.2 - Intellwheels wheelchair hardware architecture

3.1.1 User inputs

To enable a multimodal control of the wheelchair it's necessary to provide several inputs to user. It's also essential that these devices can map a broad kind of input methods, so that given any type of movement needs the user always has a way to control the wheelchair.

With that in mind were implemented the following input devices:

- USB Joystick;
- Microphone;
- Wiimote;
- Keyboard;
- Mouse;
- Video camera.

With this is possible to control the wheelchair using several types of inputs, from head movements (using the Wii command as input) to facial expressions (using the camera) [11] [12] [13].

3.1.2 Other hardware

Apart from the wheelchairs and user inputs are also being used sensoring devices like: encoders, for the odometry calculation, and sonars, for obstacle detection. There is also the necessary hardware interface to the encoders and sensors.

One final, and important hardware device, is the laptop HP Pavillion tx1270EP, which is used to run all the developed software.

3.2 Software

The software architecture being implemented is described in figure 3.3

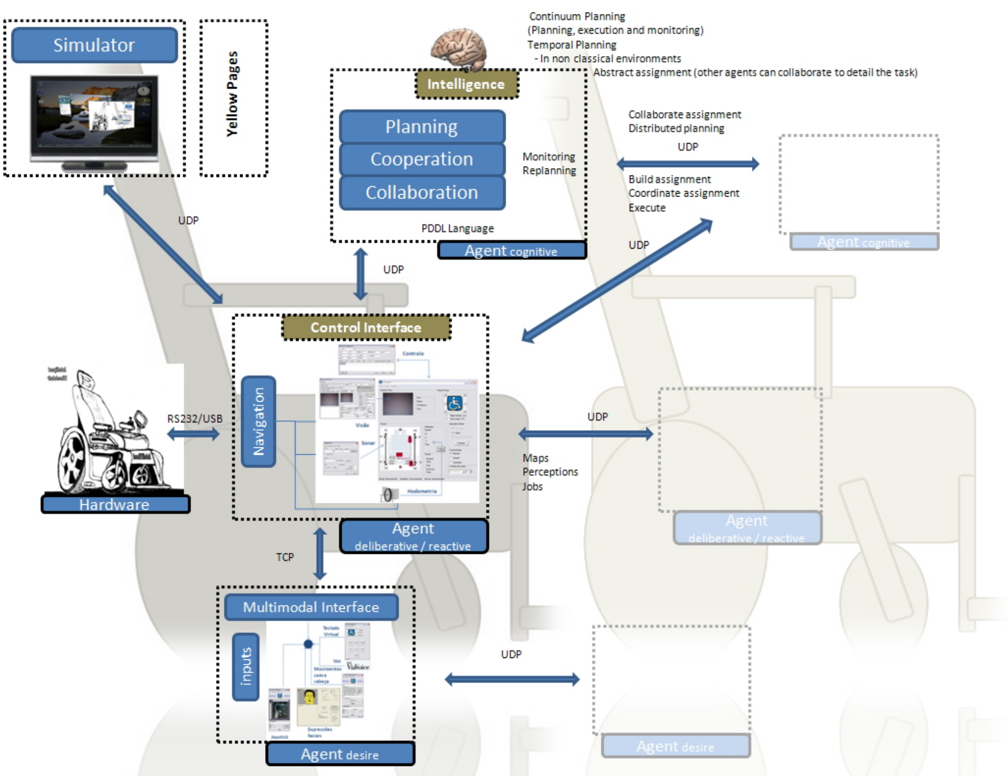


Figure 3.3 - Intellwheels project software architecture [10]

In this architecture the wheelchair control and hardware interface will be located in the Control Interface.

The Simulator is used to simulate the behaviour of a real wheelchair and test control routines. This module can also be used to enable an augmented reality mode, which generates an interaction between the real and virtual world. This simulator is being developed by Pedro Malheiro, with the name of Intelligent Wheelchair Simulation.

The augmented reality interaction allows tests with real wheelchairs inserted in a virtual world or vice versa, which in turn allows a broader range of tests before implementation in real situations, and reduced costs since the agents can be simulated [14] [15].

The high level planning structure is located at the “Intelligence” module, this module is responsible for the task planning for any high level action, like going to a defined place in a map, that the user wants to execute.

Focusing in the multimodal interface, this will interact with the Control Interface through a TCP socket connection, where the Control will inform the MMI of the available actions and state of any pending planning.

The user will interact with the MMI which provides the connection, also through a TCP socket, of several independent input modules that are used to create sequences in order to access the several control actions assign by the Control Interface

3.2.1 MMI

The MMI is the project interface to the user. In this interface is shown, in a graphical way, information about the actions, provided by the Control, and input modules, such as kind, name or type of action or input, respectively. It also shows the defined input sequences, for the actions execution.

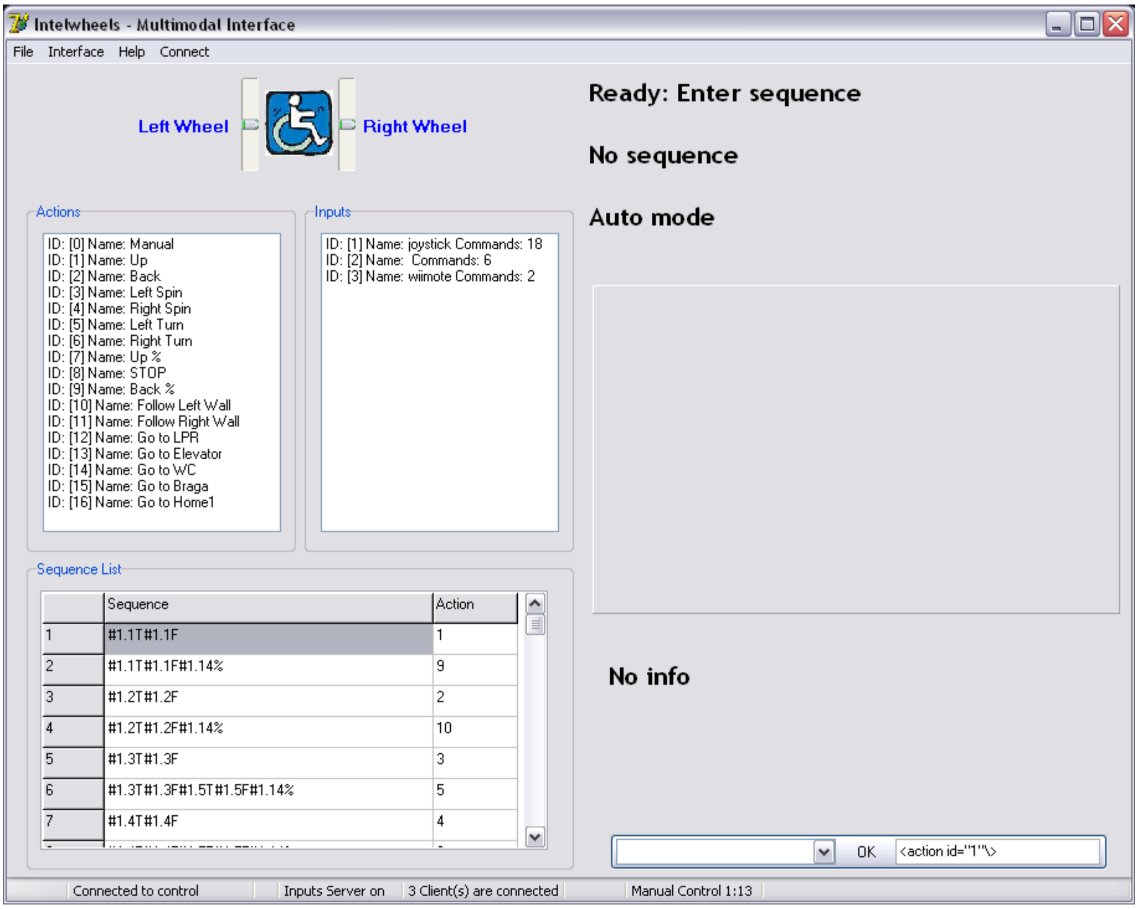


Figure 3.4 - Multimodal interface

3.2.2 Joystick module

The joystick module works as a driver to establish a connection between an USB joystick and the MMI. This module was adapted from [10], and it gets the information of the available buttons and analog sticks.

The aspect of the joystick module is shown in figure 3.5

Figure 3.5 - Joystick module

3.2.3 Voice module

To enable the voice interaction was necessary to implement speech recognition software. The presented solution takes advantage of the IBM Via Voice [16] capabilities using the navigation macros, which allows the user interaction with any software through, previously recorded, voice commands.

In this case was created a simple application, presented in figure 3.6, with six commands, and then train the macro to click in the correspondent button.

The use of Via Voice has a disadvantage since it needs the voice module window to be active so that the voice command macro can be perceived.

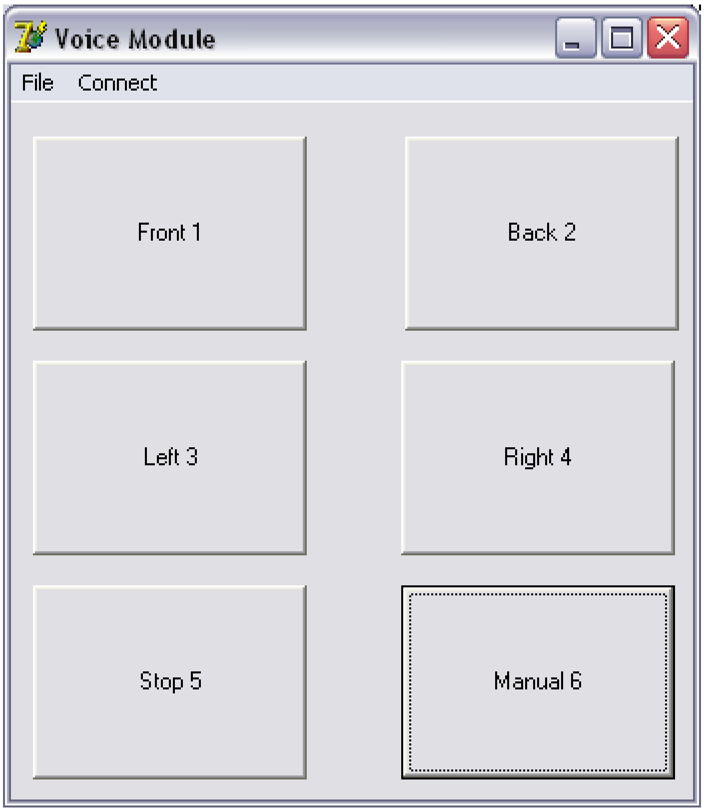


Figure 3.6 - Project voice module

3.2.4 Head movement module

One more time, to assure the integration of the already developed inputs, the head movement module, presented in figure 3.7, was adapted to communicate with the multimodal interface.

The head movement module takes advantage from the Wiimote accelerometer, where it reads its values and transforms in a position type value, for pointer control, or in a percentage speed value to control the wheelchair.

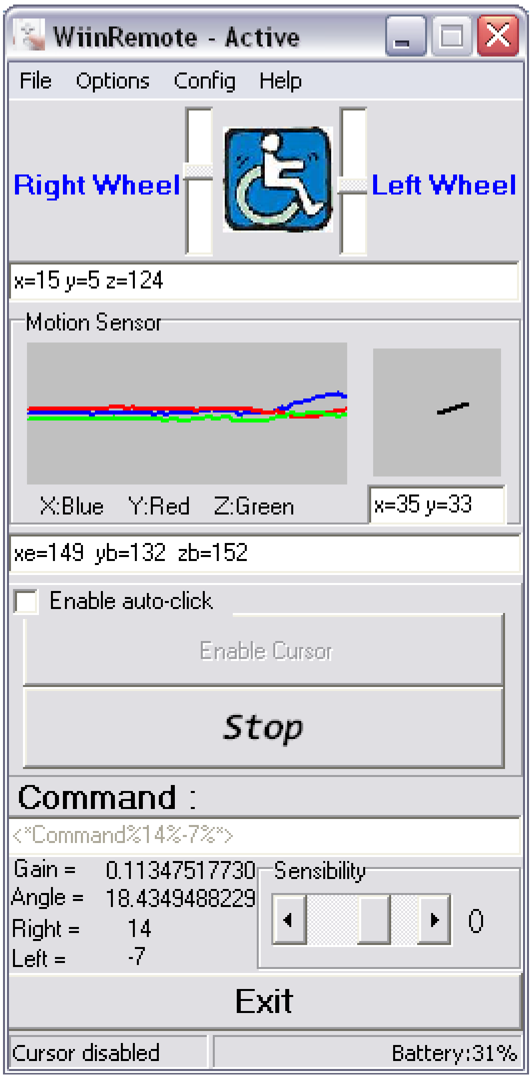


Figure 3.7 - Head movement module

3.3 Conclusions

This chapter approached the Intellwheels project, describing the hardware and software architecture, as long its components.

There was also been given a larger emphasis to the MMI, explaining in more detail its function and its modules.

In chapter 4 it will be presented the developed work to design the MMI, as long all the considerations take to establish a multimodal interaction.

Chapter 4

Multimodal Interface

This chapter first presents the architecture implemented for the multimodal interface developed. Following there is a presentation of the proposed interaction models for its three components: multimodal interface, control interface and input modules.

Secondly focusing in the user interaction the creation of input sequences is explained, demonstrating its format and the algorithms used for its analysis.

Finally the interface design is approached, as well as all other conceptual considerations being made during this project.

4.1 - Multimodal Interaction

Firstly, in order to enable the integration between all the components, it is necessary to establish an architecture, for them to communicate.

This section presents the system architecture, and all the taken measures to enable this integration.

4.1.1 Architecture

Since the wheelchair control platform is a distinct agent from the multimodal interface, it is necessary that the multimodal interface could interact with the already developed work [10]. Then with that intention was created a data structure and information processing methods for the components interaction.

The system architecture, illustrated in figure 4.1, is a zoom in on the main architecture shown in chapter 3, figure3.3. In this figure is possible to see the information exchanged between all the agents involved.

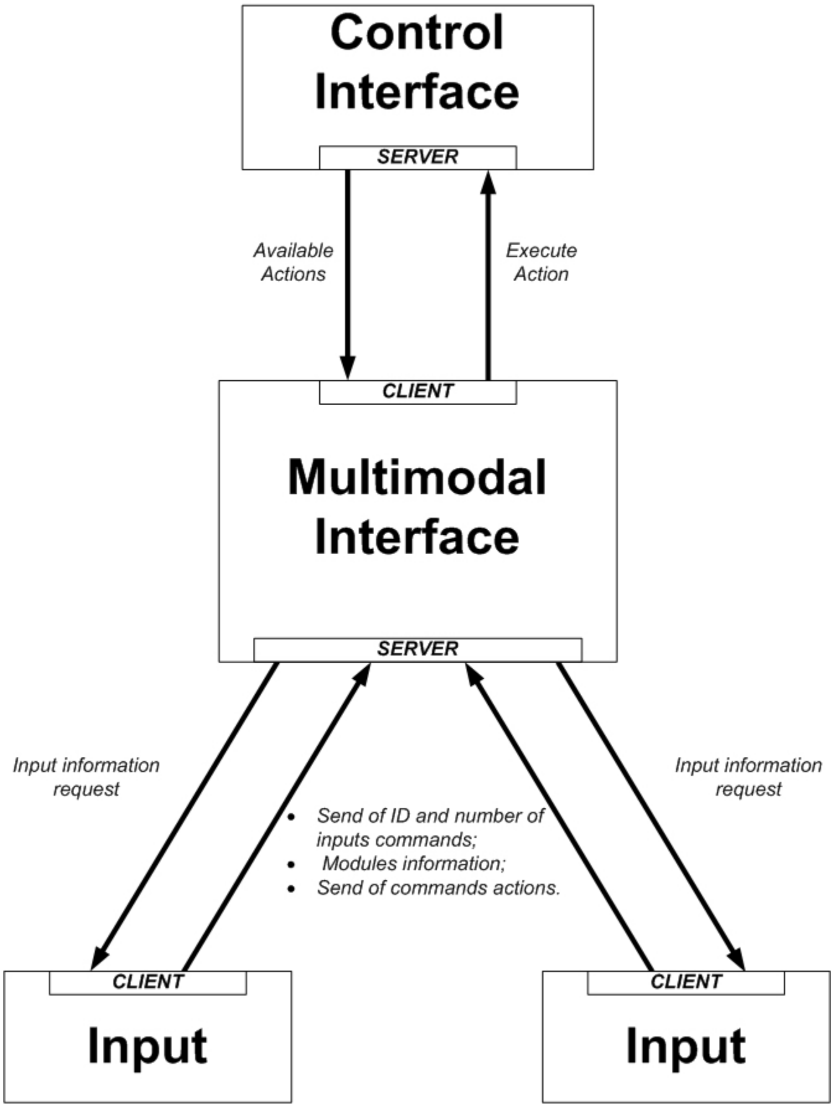


Figure 4.1 - MMI architecture

The control interface acts as a communications server to the multimodal interface, as well as the multimodal interface acts as a communications server to the input modules.

Since the communications are totally established by the used Delphi components, as soon as the multimodal interface connects to the control, the control sends the information about the available actions. For the input modules, as soon as one of them connects to the interface, firstly it sends its id and, number of module commands. Secondly upon the receiving of a request from the interface, the input module sends the description of all the commands.

4.1.2 – Interface information processing

The interface information processing is divided in two logical parts: the server side and the client side processing. This division is derived from the need of the MMI to act as the control connection client, but as the inputs connection server.

The MMI information processing is described in the data flow diagrams of figures 4.2 and 4.3, for the client side and server side, respectively.

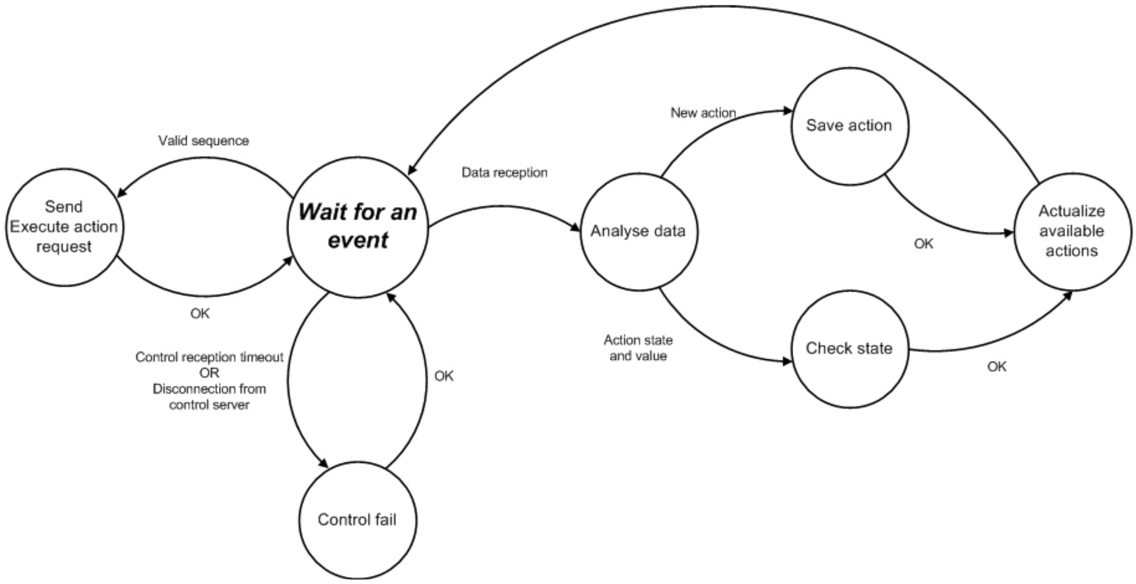


Figure 4.2 - Interface client side data flow

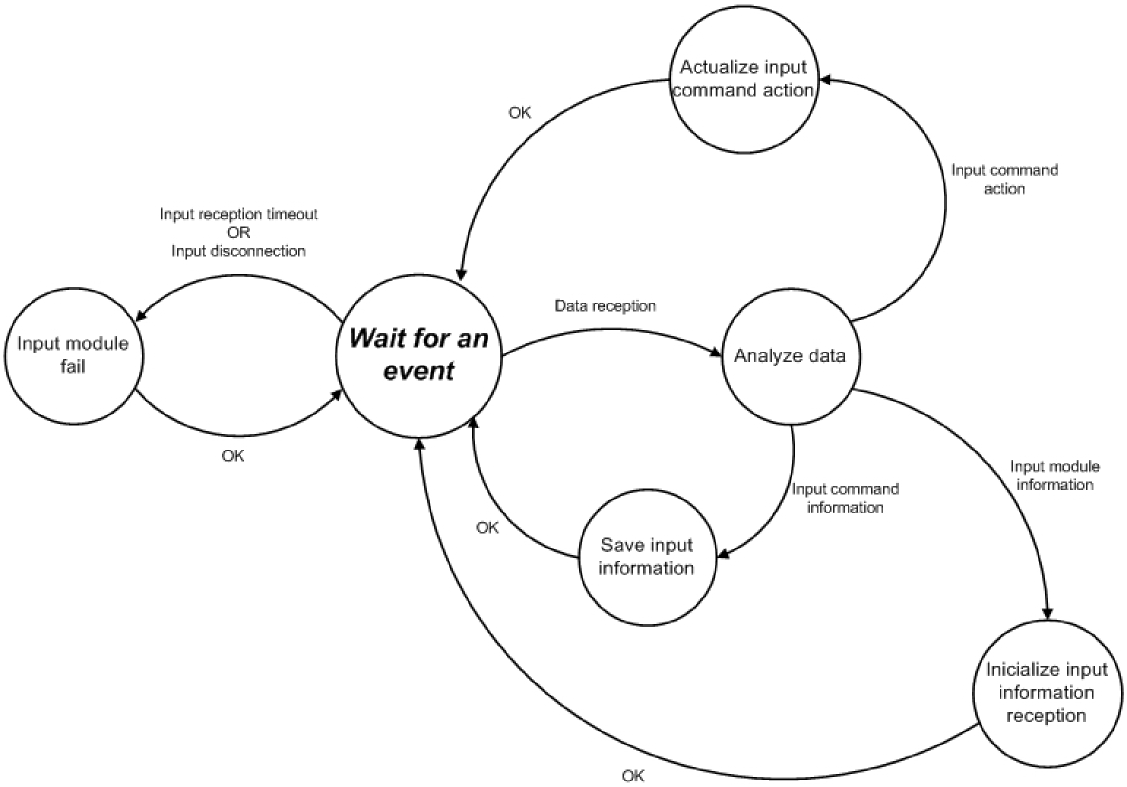


Figure 4.3 - Interface server side data flow

For these models two data structures were created. One for storing the control actions information and other for storing the input modules commands information. It remains to state that they are common to the respective applications.

The inputs structure is composed of six fields, which are:

- Num: the internal number of the command;
- Name: the name of the command;
- Kind: this defines the name of the input module;
- State: for a button, this represents if its pressed – "True" – or if it was released – "False" –;
- Value 1 and value 2: these fields serve for transmitting the analog values of a command, for example the analog stick of the joystick;

For a digital command, a button for example, the value fields will return a "n/a" string, being the same analogously applicable to an analog command, it returns the state field with a "n/a" string.

For the actions structure there will be five fields described below:

- Name: the name of the action;
- Kind: the kind of action, for example movement;

- State: the availability for executing a action, returns “True” for a available action, or “False” if not available;
- Value: informs the interface about the execution of an action, returns “ON” if the action is under execution, or “OFF” when it stops its execution;
- Data: this field serves as an information about the level type of the action, being its options in table 4.1;

Data	*Type name*	*Sent Parameters*
0	Stop action	0
1	Manual action	2
2	Mid-Level action	1
3	High-Level action	0

Table 4.1 - Action list: Data field properties

For the client side, it will only receive two kinds of information, depending if the control is refreshing its action states and values, or if the control is sending a new action. These actions are distinguishable by their tag names; “cmd_state” in the first and “cmd”, in the second case. The full transmitted messages are the following:

<cmd id=”” name=”” kind=”” state=”” value=”” data=””\>

<cmd_state id=”” state=””\>

When the interface initially connects to the control, the control will send to the interface the available actions list.

Upon the receiving of one of these messages the interface, respectively, refreshes its action list or creates a new action with the received information.

The server side of the interface has a similar behaviour as the client side. In this case the interface can receive three kinds of messages: “input_info”, that informs the interface about the input module characteristics; “input” and “module”, that function in a block and notify about the commands characteristics; and “input_action, that represents a command action event. These messages are shown below:

<input_info id=”” mods=””\>

<input id=””>

<module num=”” name=”” kind=””\>

…

<module num=”” name=”” kind=””\>

<input\>

<input_action num=”” state=””\>

<input_action num=”” value1=”” value2=””\>

The first message is send to the multimodal interface when the input module connects to it, but the second message's part is only sent after the interface accepts the connection and makes a request, through a *<input_data_req\>* message ; being this process as a registration of the input module in the interface.

Upon the receiving of the third message, the interface processes it as a fragment of an input sequence. This process is described in section 4.2.

If the interface processes a valid sequence then, the correspondent execute action request is sent to the control. This request is composed by the correspondent action id, and an extra field depending on the type of action, defined by the data type of the action.

For high level actions, such as go to a determined place, or, a stop wheelchair action, no extra field is sent, since the control does all the planning or only needs to stop the wheelchair, respectively. But, if the user wants to move in a particular direction, with a predetermined speed, then the interface transmits the execution request as a mid level action, and thus the request is composed by the action id and a value field to specify the selected speed.

One particularity to this method is found if the user wants to manually control the wheelchair. In this case the interface executes the normal request but, after that, any interaction with the defined manual control is interpreted as a movement order for the wheelchair, and the interface now sends the speed value, for each wheel, to the control.

Below are the three messages that can be sent to the control interface, depending on the action type:

- High level or Stop *<action id=””\>*;
- Mid level *<action id=”” value=””\>*;
- Manual mode *<manual value1=”” value2=””\>*;

To prevent an action overlapping, the control only terminates the manual mode upon receiving a stop command from the multimodal interface.

Both the client and server sides have a security mechanism, in case of the failure of any connection, which informs the user about the error. This mechanism also sends a Stop action to the control, if the failure is in any of the input modules.

Finally figure 4.4 shows the MMI global data flow.

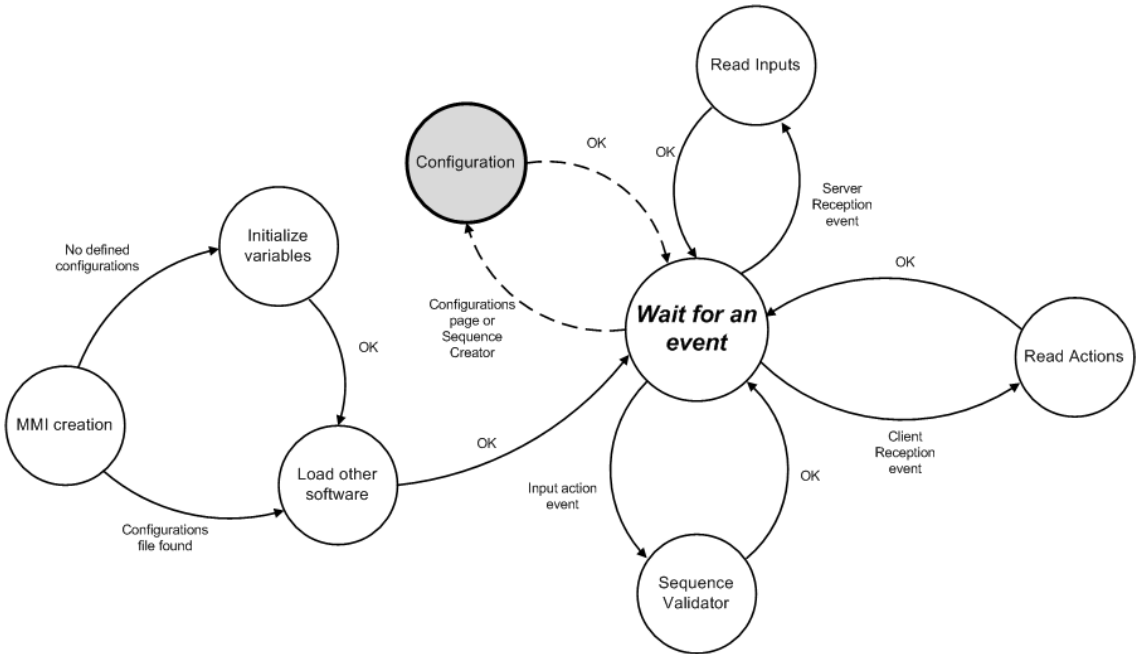


Figure 4.4 - MMI global data flow

The “Read Inputs” and “Read Actions” blocks respectively represent the server and client side data flows that were already explained in this subsection.

The “Sequence Validator” block contains all sequences analysis algorithm, which is explain in section 4.2.

The data flow also shows the MMI loading process, in which first searches for the proper configuration file. If the file doesn’t exist the configurations page is shown to the user, in order to fill the correspondent fields.

The last step before the MMI can be usable is to load all the external software necessary for the Intellwheels system functioning.

4.1.3 – Control information processing

To allow the interaction between the MMI and the Control interface [10] it was necessary to adapt the latter, so they can be compatible.

For the interaction to function correctly it is necessary that the control processes the available actions with the proper data structure, as shown in section 4.1.2. After the connection of the MMI, it needs to send the action list.

Like it was said in section 4.1.2, the control must send the action’s states within a time interval, so that the MMI can display to the user an actualized state of the control actions.

The availability of a new action, must be transmitted to the MMI, so the user can have knowledge about this new control option.

Upon the reception of an “action” message the control executes the user request. In case of receiving a “manual” message the control applies to the wheels the correspondent speed.

Figure 4.5 shows the data flow of the control server, which summarizes this subsection.

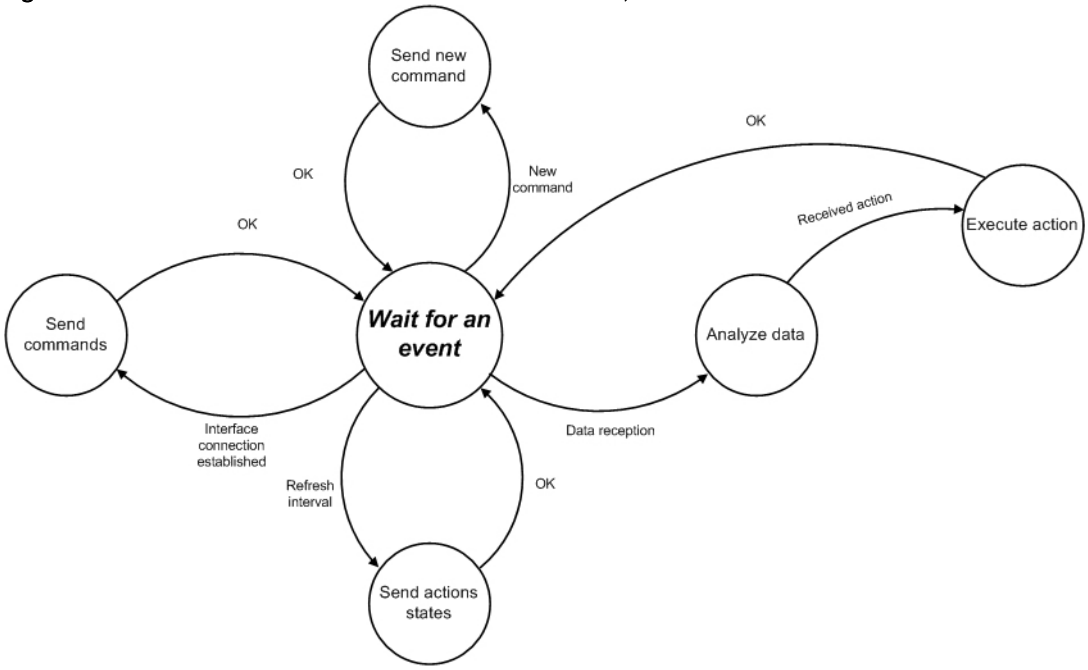


Figure 4.5 - Control data flow

As a security, like it was implemented in the MMI, the control must stop all elapsing actions if there’s a failure in any of its connections: simulator, MMI.

4.1.4 – Input module information processing

Finally the last module contains the information processing in the input modules, which can be seen if figure 4.6.

These modules, upon the establishment of the connection to the MMI, send to it their characterizing information: the id, and the number of commands. Then it waits for the request, from the MMI, to send the detailed command information.

When a command event action occurs, for example pressing a button, this event is sent to the MMI in an “input_action” message.

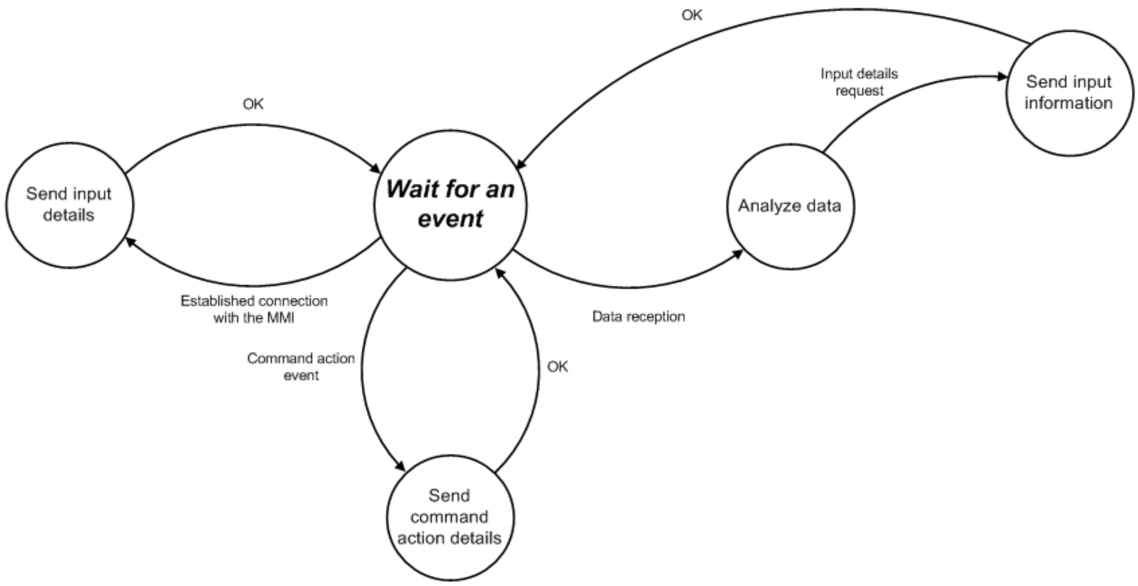


Figure 4.6 - Input modules data flow

4.2 – Input Sequences

The input sequences represent how the user interacts with the interface or being more precise how the user controls the wheelchair. These sequences are created through the combination of two or more input commands actions.

In this subsection the sequence creation and analysis are described.

4.2.1 – Sequence format

Independently from the input module kind, or if the command is digital or analog, the associated event has two common identifiers: the module id, and the command number.

To standardize the inputs events representation was defined that if the command is digital, then its state is "True" – T – when the button is pressed, or "False" – F - when the button is released.

For an analog command the state field is not used, but the input module will return the values of the axis, which are converted to a percentual value and used to directly control the wheelchair, or are "processed" and define a fixed speed value.

This fixed speed value is achieved by logically "dividing" the cursor area in one of the analog axis of the joystick. Due to the short length of the stick it's only possible to divide in four areas without losing precision, being the division shown in figure 4.7. The variation assumes increments of 25% per zone, from A to D.

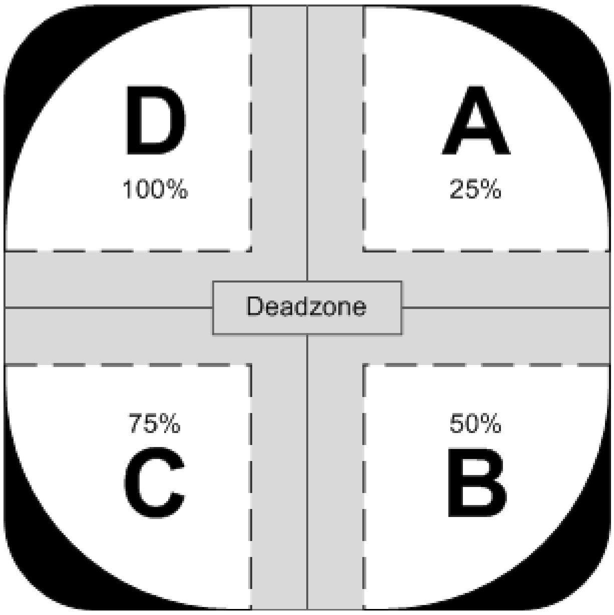


Figure 4.7 – Fixed speed values division zones

One imposition for this method is that the MMI must previously know what input module command acts as the manual control, in order to, when in manual mode, only accept analog values from one stick, in case that exists more than one stick.

To simplify the sequence creation method, it was imposed a maximum number of six input command actions (fragments) to generate a sequence. Also, a minimum number of one input was imposed.

At the reception of an input command action in the MMI, the information is stored in a sequence fragment with the following format:

#<input_module_id>.<command_number><state>

The state field can be composed by one of three possibilities: "T" or "F", in case of a button, or "%" in case of one of the four pre established values (A, B, C or D).

The sequence entrance is limited by the already referred maximum number of fragments, or at any instance by the detection of an existent or nonexistent sequence. That is, for each fragment received the developed algorithm will actualize the composing sequence and, will search in the sequence list for the same occurrence. The search returns one of three available options:

- The occurrence is unique, and therefore the composed sequence can be immediately analysed;
- There are more occurrences of the same sequence fragment;
- The occurrence doesn't exist in the list, meaning that the user is entering a not valid sequence and therefore the process is stopped;

In case of the search indicates more occurrences of the current sequence fragment the algorithm waits for more input event actions. With this is possible to evaluate if the user is still entering the sequence or, if during a pre established time interval none input action is received, if the sequence is already completed.

The use of this process turned the sequence input method more reactive to the user by providing an almost instant response to unique or wrong sequences, allowing a more effective control.

4.2.2 – Sequence analysis

Like it was shown in figure 4.4, when the MMI receives an input event action the “Sequence Validator” processes that information. This block is a summary of this subsection, which can be seen in figure 4.8.

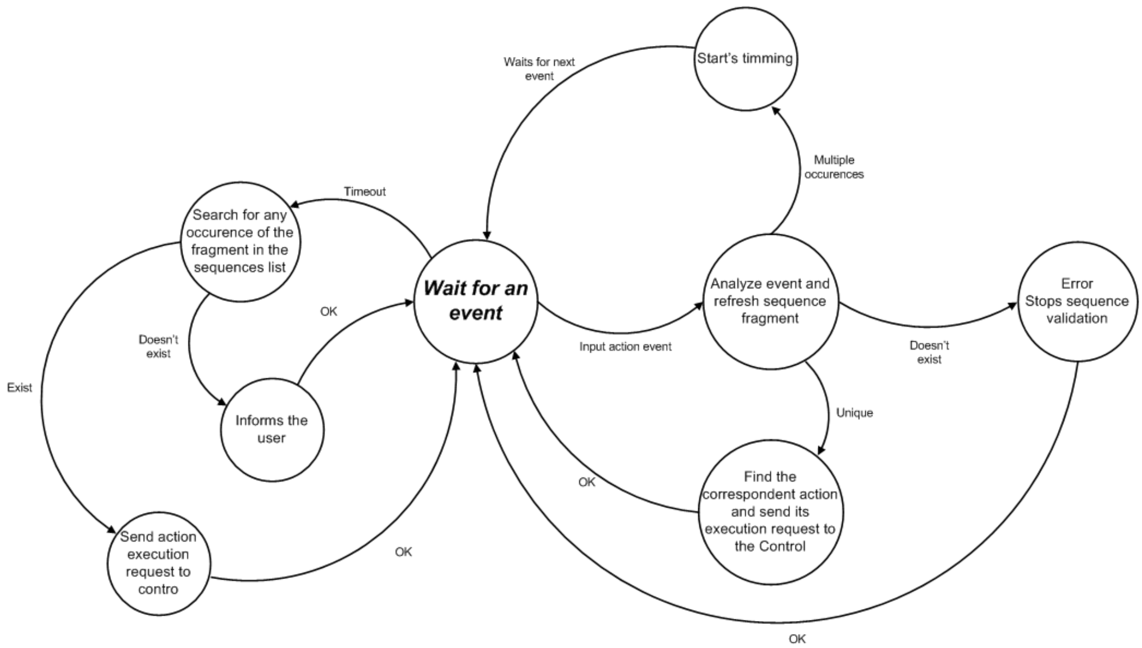


Figure 4.8 - Sequence analysis data flow

The sequence analysis is initiated when the system detects that the user has finished the insertion of an inputs sequence, and its objective is to search for any occurrence of the inserted sequence in the defined sequences list.

The analysis is implemented using a Binary Search algorithm [17]. This algorithm has the advantage of reducing the search time in large arrays, but requires that the search array must be sorted. It returns the position of the occurrence in the array if there is a match or -1 if it doesn’t find any match.

The sequences and actions list is stored in two arrays, one for each type, being the correspondence between them at the array position.

When the search algorithm gives a result different from -1, the MMI prepares for sending the action execution request for the control. This preparation checks for any additional parameter to send using the action data field.

Due to the sorted array restriction, when the user inserts a new sequence in the list, the list is ordered by a QuickSort algorithm [18]. This algorithm was chosen because of its efficient ordering, although this is achieved in arrays with twenty or more elements, in this case for a flexible control, where the user can assign several sequences, easily this “target” is surpassed.

4.3 – Interface Components

The interface components are all the interface visible components, since the simple buttons to images, menus and textual information.

For the MMI to be simple it only contains the following components, where some are shown in the figures 4.9 to 4.12:

- List of available actions, figure 4.9;
- Summary of the inputs connected, figure 4.9;
- Input modules and control connections status;
- Input sequence graphical information, figure 4.10;
- Sequence’s list, figure 4.12;
- Sequence’s analysis result, figure 4.10;
- Wheels speed information, figure 4.11;
- Menus for programming the interface options and adding more sequences.

All these components, except the input sequence information that shows the information in a graphical way, show the information in a textual way.

The following image’s shows the above referred components.

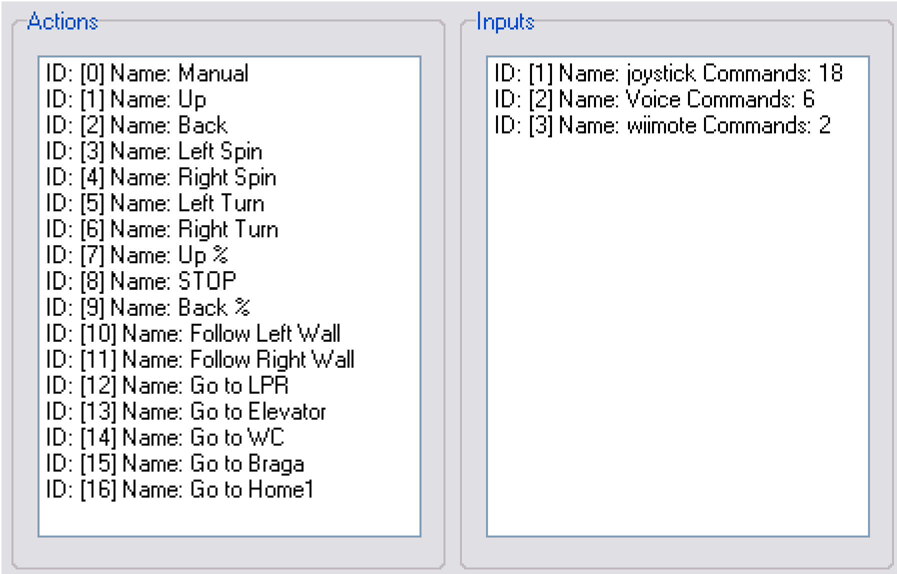


Figure 4.9 - Actions and input modules lists

In figure 4.10 can be seen the textual information, given to user, about the availability to insert a new sequence, the sequence analysis status, the actual control mode and any information transmitted by the control to the interface.

It can also be seen the input sequence graphical information.

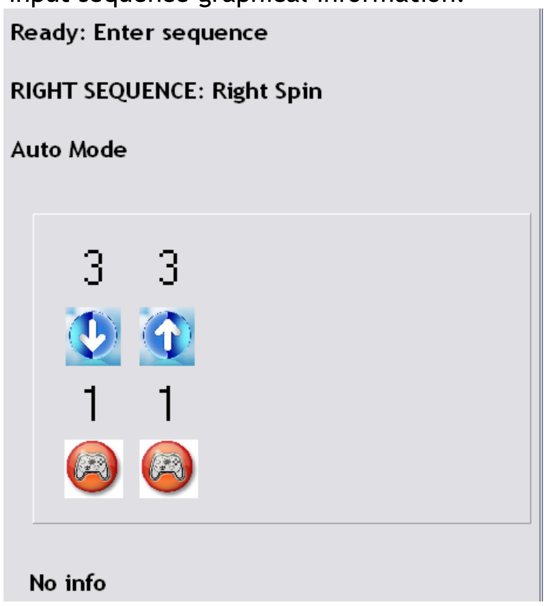
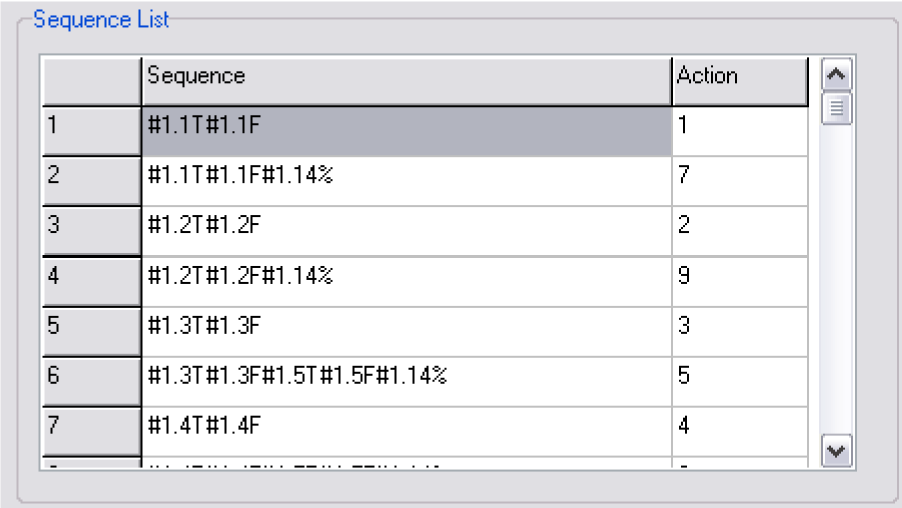


Figure 4.10 - Textual messages and input sequence graphical information

Figure 4.11 - Wheels speed information


Sequence List

	Sequence	Action
1	#1.1T#1.1F	1
2	#1.1T#1.1F#1.14%	7
3	#1.2T#1.2F	2
4	#1.2T#1.2F#1.14%	9
5	#1.3T#1.3F	3
6	#1.3T#1.3F#1.5T#1.5F#1.14%	5
7	#1.4T#1.4F	4

Figure 4.12 - Sequence's list

The menu options are:

- File **à** Exit;
- Interface **à** Configuration;
- Interface **à** Sequence Creator;
- Interface **à** About.

As the name says the first option exits the software, the Configuration shows the configuration menu, the Sequence Creator shows the menu for inserting a new sequence, and the last displays the project information.

In the Configuration menu, figure 4.13, it is possible to change the connection settings (IP's, Ports), the manual control and the insert sequence timing.

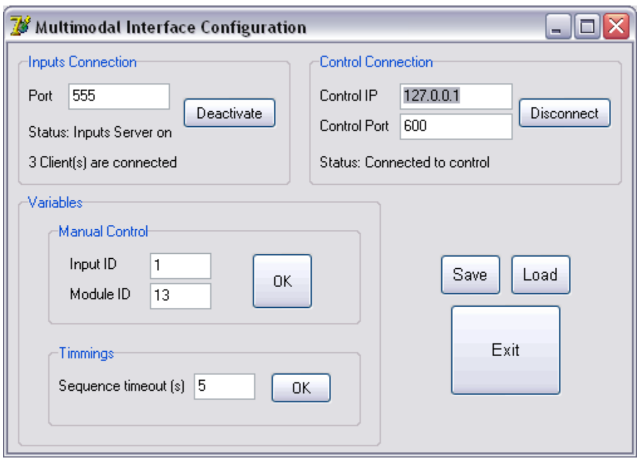


Figure 4.13 - Configuration form

In the Sequence Creator, figure 4.14, menu the user can see all the already configured sequences and their actions correspondence, as long as detailed information about the input modules and actions. This menu is used for defining new input sequences.

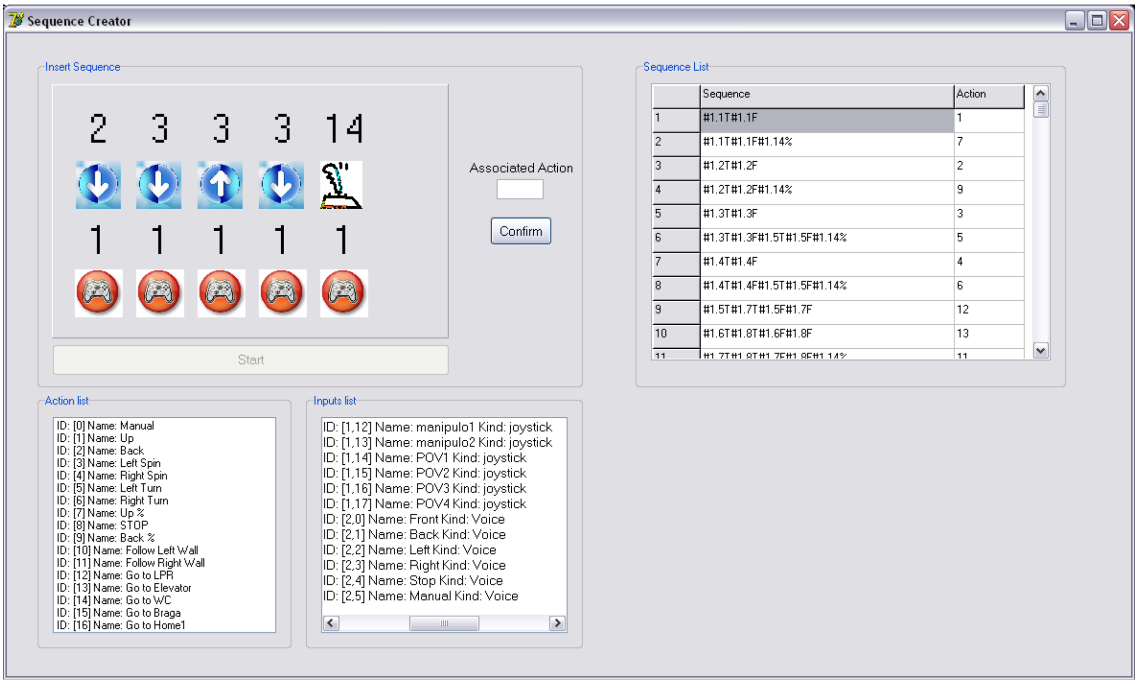


Figure 4.14 - Sequence creator form

Like it was seen in the figures, the sequence insertion is complemented by a graphical correspondence. This correspondence is taken in the format:

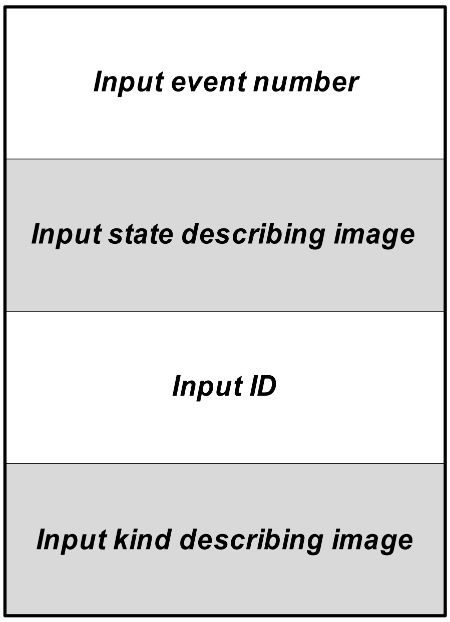


Figure 4.15 - Sequence graphical correspondence format

For the images fields the pictures in table 4.2 were chosen.

Table 4.2 - Pictures table

<table>
<tr><th colspan="12">Input kind</th></tr>
<tr><th colspan="4">Joystick</th><th colspan="4">Voice</th><th colspan="4">Head movement</th></tr>
<tr><td colspan="4"></td><td colspan="4"></td><td colspan="4"></td></tr>
<tr><th colspan="12">Input state</th></tr>
<tr><th colspan="3">Button up</th><th colspan="3">Button down</th><th colspan="3">Variable value</th><th colspan="3">Voice</th></tr>
<tr><td colspan="3"></td><td colspan="3"></td><td colspan="3"></td><td colspan="3"></td></tr>
</table>

4.4 - Conclusion

This chapter described the work developed from the architecture implemented for the MMI to the corresponding changes in the control and input modules.

It was also presented the sequence format and its analysis, and the visual aspect of the interface. This interface enables the complete configuration of the wheelchair command in a very flexible manner.

The next chapter will show the tests conducted to the implemented multimodal interface and their results.

Chapter 5

Tests and Results

This chapter describes the experiments made to evaluate the developed multimodal interface. These experiments were performed in different conditions to confirm the inputs interaction.

In order to attest the MMI integration in the Intellwheels project, the tests were made using the Intellwheels simulator.

It will be presented the results obtained by controlling the wheelchair using only one type of input.

The simulated experiments were made in a virtual environment that recalls the I120 lab at DEEC, FEUP.

5.1 Joystick Control

The joystick is the traditional input method for the electric wheelchairs. In these tests the implemented joystick input module was used in a simulated and in a real environment. For comparing purposes, in the real environment, it was also used the wheelchair built-in joystick.

These tests were performed by controlling the Wheelchair in manual mode, exclusively with the joystick.

The simulated tests firstly were made in an empty room, where the wheelchair starts from the middle and tries to go around the room, finishing in the same position. Secondly the course was the same but now it includes several obstacles that the wheelchair needs to deviate.

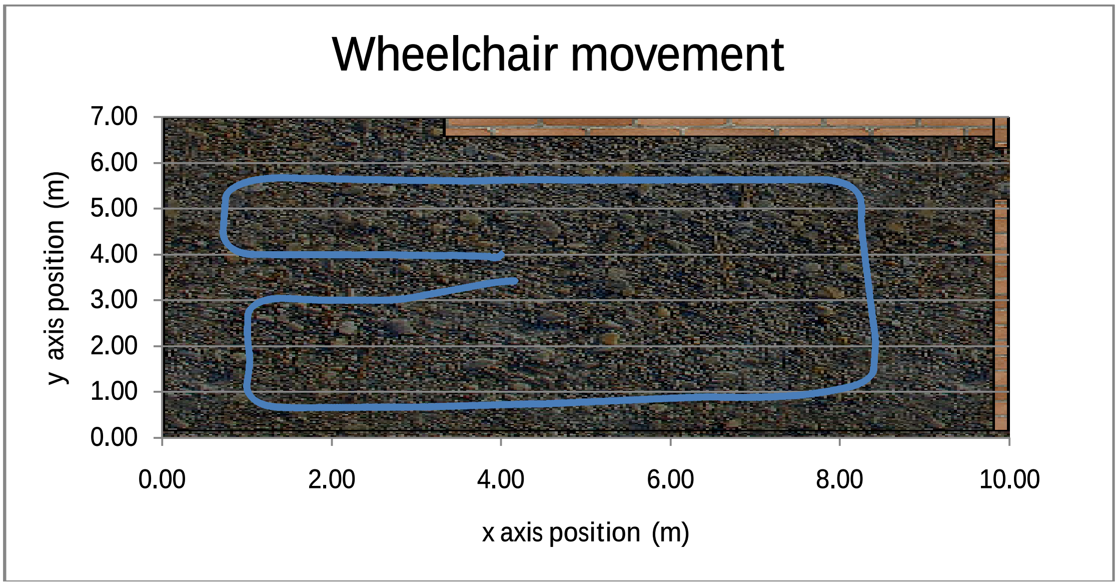


Figure 5.1 - Wheelchair movement in an empty room, with joystick control in a simulated environment

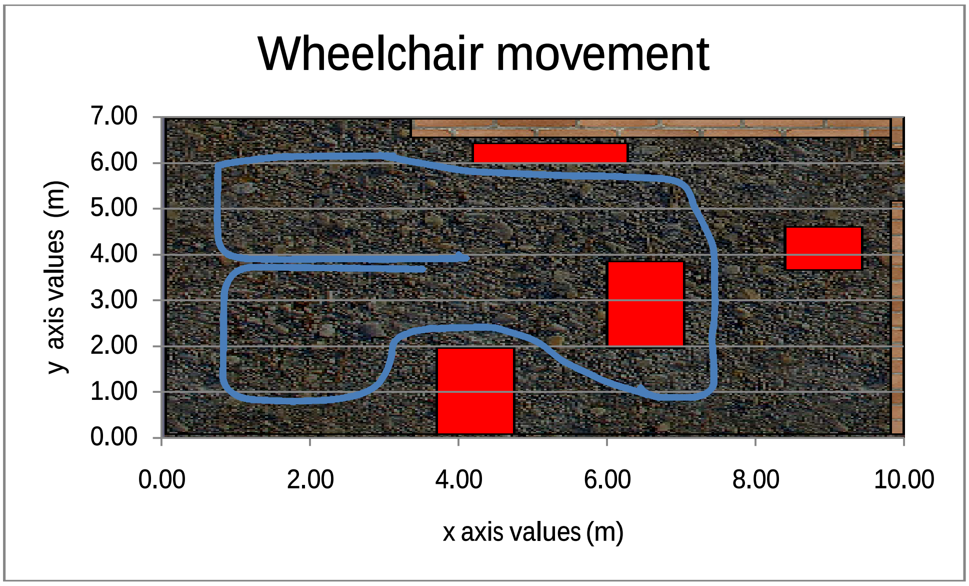


Figure 5.2 - Wheelchair movement in a room with obstacles, with joystick control in a simulated environment

With these results is possible to see the wheelchair movement through the room, being the input method able to drive the wheelchair in the predetermined course without any remarking problem.

5.2 Voice Control

The voice input module experiments were executed using the same environment settings as the joystick. The only difference resides in an extra experiment that introduces some background noise.

For the control were defined five commands (front, back, left, right and, stop) to control the wheelchair via voice commands.

The experiments results are presented in figures 5.3 to 5.6

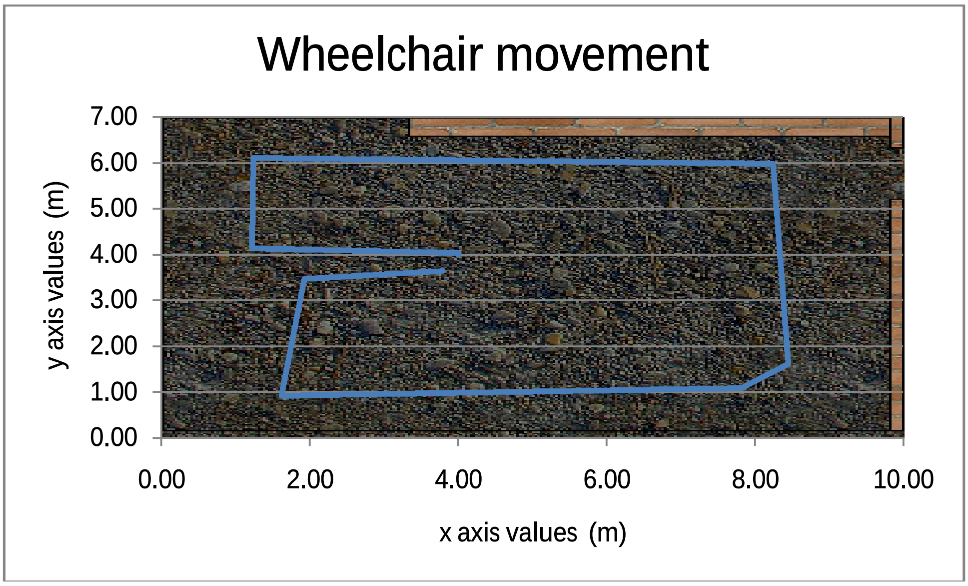


Figure 5.3 - Voice control in an empty room, without background noise

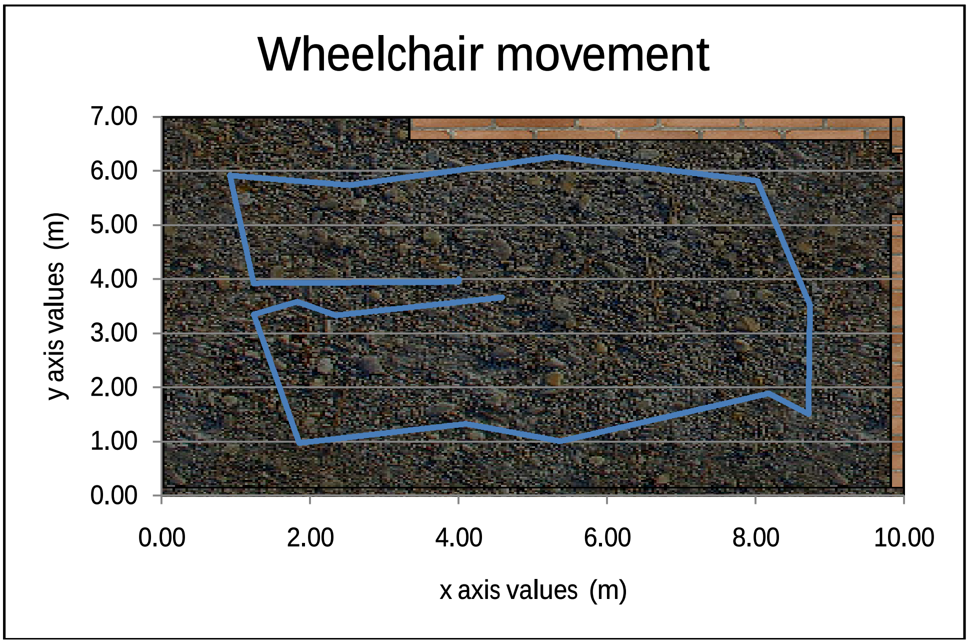


Figure 5.4 - Voice control in an empty room, with background noise

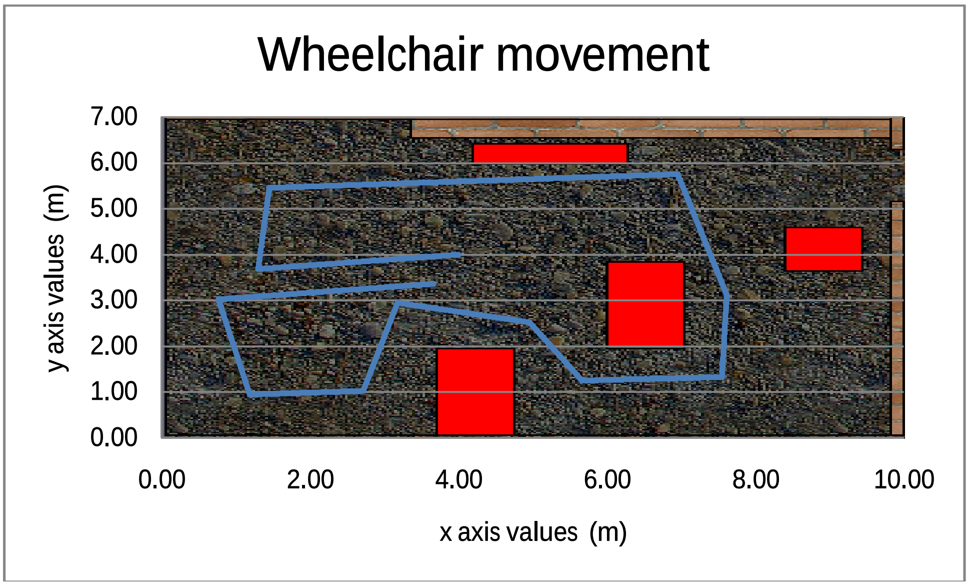


Figure 5.5 - Voice control in a room with obstacles, without background noise

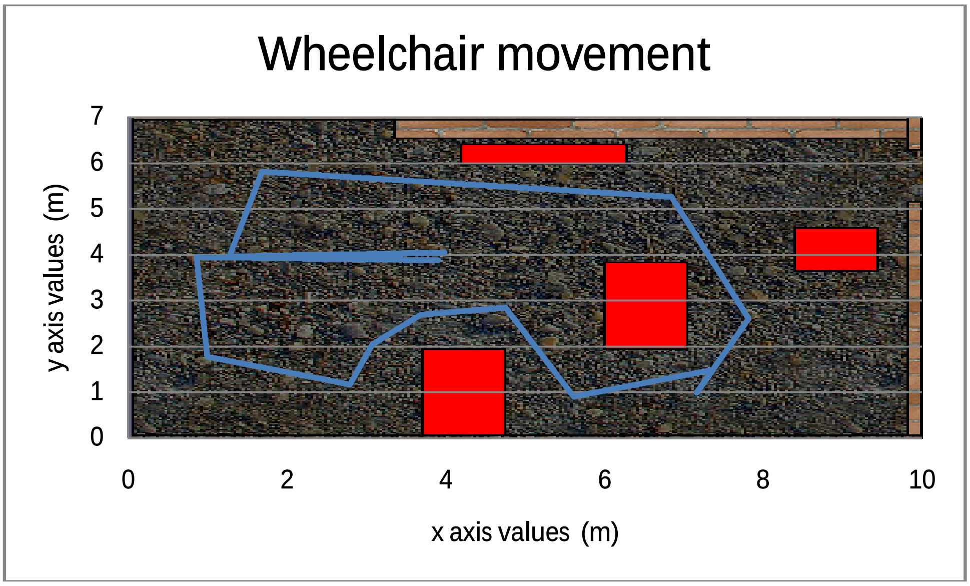


Figure 5.6 - Voice control in a room with obstacles, with background noise

After these experiments it's possible to see that the implemented voice input method, for directly control the wheelchair, is a preferably input for open areas without obstacles. This is due to the delay in the response of the speech recognition software, which in emergency situations can become dangerous.

Another aspect of the experiments is the sensibility of the speech recognition software with background noise. During the background noise experiments the microphone was approximately at 30 cm from the user, and the noise source was a radio playing music with low volume. In these conditions it was necessary to repeat the voice commands several times, which has increased the experiment time.

5.3 Head Movements

Once again this experiment uses the same settings as the joystick module experiment, but changing the manual control to the head movement module.

For the tests be comparable it was necessary to adapt the module sensibility to a high value (16 – 17), so that the achieved speed could reach the same values as the joystick or voice commands.

The results are shown in figures 5.7 and 5.8.

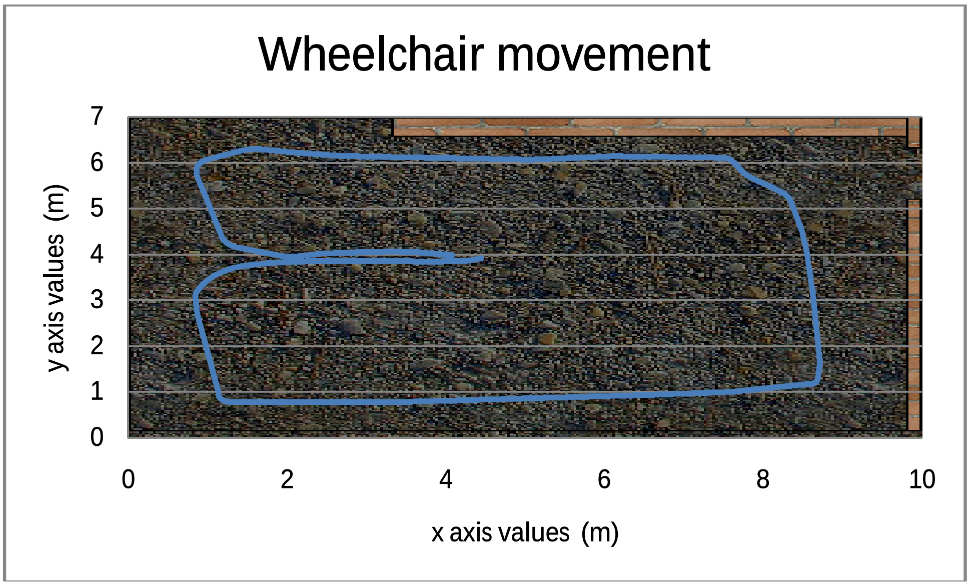


Figure 5.7 - Head movement control in a empty room

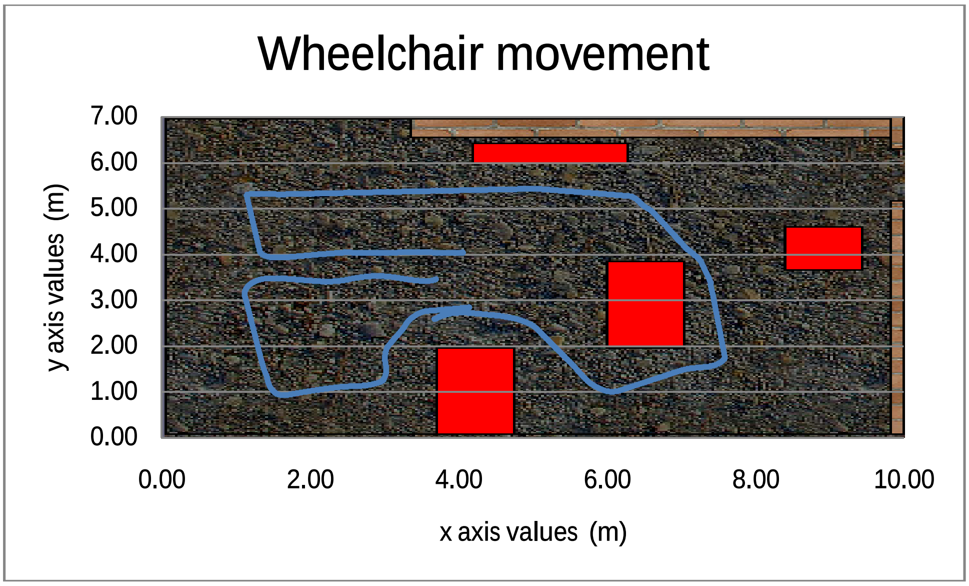


Figure 5.8 - Head movement control in a room with obstacles

The results show that it is possible to drive the wheelchair just using head movements with good performance. Again, this movement method is preferred for open environments without obstacles. However the method is completely capable of manoeuvring the wheelchair in a crowded room, performing precise movement tasks.

5.4 Conclusions

The set of tests performed has pointed out several differences between the wheelchair control input methods.

The best results have been achieved by the joystick, being this the traditional input for an electric wheelchair, for which is an outcome of its design and functionality.

The head movement module results are also very satisfactory, where in situations of difficult passing or manoeuvre it's possible to adjust its sensibility and therefore have a more precise control over the wheelchair.

Although these tests were made without any source of distraction, controlling the wheelchair with the head movement module, with a high sensibility, and with several sources of distraction, is possible to affirm that the wheelchair will not perform an accurate action.

The voice module due to the speech recognition software sensibility to background noises, which originate some errors in the voice commands detection, it is not proper to be used in situations where an accurate control is needed.

Although the experiments were individual for each input module, it is possible to conclude that they can interact with each other, and so compensate each other flaws.

Chapter 6

Conclusions and Future work

This thesis aimed at designing a multimodal interface for an intelligent wheelchair. It described its architecture, main functionalities and the integration in the Intellwheels project in development at LIACC. The developed multimodal interface showed to be very flexible enabling the user to define distinct types of command sequences and associate them to the available high-level outputs.

To attest the system efficiency and the wheelchair control through the developed multimodal interface several tests were conducted, where the wheelchair was controlled with the available inputs (joystick, voice, head movements and several inputs) in different kinds of environments (noise in the background, obstacles, etc.). The results achieved enabled to confirm the multimodal interface capabilities, except for the voice module, which proved not to be precise when there is noise in the background.

Although this flaw, the Via Voice software brings an additional component to the system, which is the computer control with the voice, through pre defined voice commands, widening this project application areas.

Some future directions for this project development are obvious and concern the development of the yet missing input modules. One missing feature is a robust facial expressions recognizing module, in order to create an ever more multimodal experience to the user.

Another aspect that should be observed is the speech recognizing software, since it is an extra application that interacts with the voice module, and has some recognizing flaws in noisy backgrounds. It should be searched an alternative that must be more robust allowing the same features, being totally integrated in the software.

With the recent developments in “mind-controlled” devices, a input method that could implement this type of control could break any physical disability barrier for controlling the wheelchair.

With the intention of making the MMI more user friendly, a text to speech output and some kind of virtual user assistant could be implemented. These elements would function as an integration process, with the interface, for the user.

Although several features are still missing and an extensive set of tests could be performed to the developed MMI, the final conclusion for the project is that the implemented multimodal interface is a very solid base for introducing all the improvements needed.

References

[1] S. Oviatt, "Multimodal Interfaces," in *Handbook of Human-Computer Interaction*, 2002, pp. 286-304.

[2] Philip R. Cohen, Michael Johnston, David McGee and Sharon Oviatt and Jay Pittman and Ira Smith and Liang Chen and Josh Clow, "QuickSet: multimodal interaction for distributed applications," in *Proceedings of the Fifth ACM international Conference on Multimedia*, Seattle, Washington, United State, 1997, pp. 31-40.

[3] M. Johnston, S. Bangalore, and G. Vasireddy, "MATCH: multimodal access to city help," in *IEEE Workshop on Automatic Speech Recognition and Understanding.*, 2001, pp. 256-259.

[4] E. V. Zudilova, P. M. A. Sloot, and R. G. Belleman, "A multi-modal interface for an interactive simulated vascular reconstruction system," in *Proceedings of Fourth IEEE International Conference on Multimodal Interfaces*, 2002, pp. 313-318.

[5] A. Zandifar and A. Chahine, "A Video Based Interface to Textual Information for the Visually Impaired," in *Proceedings of the 4th IEEE international Conference on Multimodal interfaces*, 2002, pp. 325-330.

[6] C. Keskin, K. Balci, O. Aran, B. Sankur, and L. Akarun, "A Multimodal 3D Healthcare Communication System," in *3DTV Conference*, 7-9 May 2007, pp. 1-4.

[7] S. Oviatt, "User-centered modeling and evaluation of multimodal interfaces," *Proceedings of the IEEE*, vol. 91, pp. 1457-1468, 2003.

[8] J. Preece, *Human-Computer Interaction*. Addison-Wesley, 1994.

[9] Sunrise. Sunrise Medical. [Consulted on June 2008]. http://www.sunrisemedical.com/

[10] Vassili. Vassilli. [Consulted on June 2008]. http://www.vassilli.it/

[11] R. M. A. Braga, M. R. Petry, A. P. Moreira, and L. P. Reis, "Intellwheels: A Development PlatForm for intelligent wheelchairs for disabled people," in *5th International Conference on Informatics in Control, Automation and Robotics*, Funchal, Madeira, Portugal, 2008, pp. 115-121.

[12] P. M. Faria, R. A. M. Braga, E. Valgôde, and L. P. Reis, "Interface framework to drive an intelligent wheelchair using facial expressions," in *IEEE International Symposium on Industrial Electronics*, 2007, pp. 1791-1796.

[13] P. M. Faria, R. A. M. Braga, E. Valgôde, and L. P. Reis, "Platform to Drive an Intelligent Wheelchair using Facial Expressions," in *Proceedings 9th International Conference on Enterprise Information Systems - Human-Computer Interaction*, Funchal, Madeira, 2007, pp. 164-169.

[14] B. Martins, E. Valgôde, P. Faria, and L. P. Reis, "Multimedia Interface with an Intelligent Wheelchair," in *Proc. of CompImage 2006 –Computational Modelling of Objects Represented in Images: Fundamentals Methods and Applications*, Coimbra, Portugal, 2006, pp. 267-274.

[15] A. M. R. Braga, M. R. Petry, L. P. Reis, and E. Oliveira, "Multi-Level Control of an Intelligent Wheelchair in a Hospital Environment using a Cyber-Mouse Simulation System," in *5th International Conference on Informatics in Control, Automation and Robotics*, Funchal, Madeira, Portugal, 2008, pp. 179-182.

[16] R. A. M. Braga, R. J. F. Rossetti, L. P. Reis, and E. C. Oliveira, "Applying multi-agent systems to simulate dynamic control in flexible manufacturing scenarios," in *Agent-Based Modeling and Simulation Symposium, 19th European Meeting on Cybernetics and Systems Research*, vol. II, Vienna, 2008, pp. 488-493.

[17] IBM. Via Voice. [Consulted on June 2008]. http://www-306.ibm.com/software/pervasive/embedded_viavoice/

[18] Wikipedia. Binary Search. [Consulted on June 2008]. http://en.wikipedia.org/wiki/Binary_search

[19] Wikipedia. Quick Sort. [Consulted on June 2008]. http://en.wikipedia.org/wiki/Quick_sort

[20] CodeGear. CodeGear. [Consulted on June 2008]. http://www.codegear.com/

[21] W3C. Multimodal Interaction Activity Group. [Consulted on June 2008]. http://www.w3.org/2002/mmi/

[22] Wikipedia. Multimodal Interaction. [Consulted on June 2008]. http://en.wikipedia.org/wiki/Multimodal_interaction