



Vrije Universiteit Brussel

FACULTY OF ENGINEERING



Development of a Graphical User Interface for a Rehabilitation Exoskeleton

Laura De Rijcke

Promoter: prof. dr. ir. Dirk Lefebber
Co-promoter: prof. dr. ir. Bram Vanderborght
Supervisors: ir. Branko Brackx
ir. Victor Grosu

Thesis submitted in partial fulfilment of the requirements for the Master's degree in:

Master of Science in Electro-Mechanical Engineering

Academic year 2012-2013



Acknowledgment

In the first place, I want to thank my thesis promoter professor dr. ir. Dirk Lefeber. He guided me through my investigation and was always willing to help me. I would like to thank the thesis president and my co-promoter professor dr. ir. Bram Vanderborght for all the support and advice on my master thesis. I am very grateful to my supervisors: ir. Branko Brackx and ir. Victor Grosu. Their opinion really helped me in making progress. Their expertise and helpfulness is really remarkable. Finally I would thank Doctor Piet Mortelé and the other physiotherapists of the department of 'physical medicine locomotory rehabilitation' at the Heilig Hart Hospital of Roeselare. They were really helpful for discussing the physiotherapeutic aspects of a rehabilitation therapy.

Abstract

Thesis title: Development of a Graphical User Interface for a Rehabilitation Exoskeleton

Author: Laura De Rijcke

Master's degree in: Science in Electro-mechanical Engineering

Academic year: 2012-2013

Keywords: Graphical User Interface, Rehabilitation Robot, Trajectory Generation

Gait impairment has a highly negative influence in the health-related quality of life. In order to help patients with gait impairment a gait rehabilitation robot was developed at the Vrije Universiteit Brussel, namely ALTACRO. In this context there is investigated about the development of a graphical user interface for such device.

A graphical user interface is very important for a rehabilitation device. The existence of such interface improves the user friendliness and the safety of the device. It facilitates the control over the device and the communication between the user and the robot.

The first part of the thesis examines existing graphical user interfaces of rehabilitation devices. These interfaces are compared with each other and the main options of each interface are listed up. Thereafter is investigated on performance indicators. These indicators are parameters which give an indication about the performance of the patient during a rehabilitation session. Several performance indicators are compared to see which one gives the most accurate results used in combination with a gait rehabilitation device. Finally, different programming environments are explored to see which is the most suitable for programming a graphical user interface for a rehabilitation robot.

In the second part of the thesis the requirements for a graphical user interface are explored. The interface is needed for three different types of users: engineers, physiotherapists and patients. Each of these user groups has specific requirements for the contents of the interface. The requirements of the different user groups are investigated and listed up. For the groups of the physiotherapists and the patients this is done with the cooperation of a physiotherapist.

In the third part of this master thesis a trajectory generation tool is developed. The trajectories are generated based on several user inputs specific for a gait rehabilitation therapy, such as the step length, the step height and the amplitude of the hip movement. These generated trajectories should also be kinematically and physiologically correct. In a next step of the thesis the influence of the different gait parameters on the generated joint angles is explored.

In the final part of this master thesis the gathered knowledge is used to make a graphical user interface for ALTACRO. The interface focuses mainly on the needs of the engineers. A trajectory generation tool is developed for this interface.

Résumé

Titre du mémoire: Développement d'une Interface Graphique pour un Exosquelette de Réhabilitation.

Auteur: Laura De Rijcke

Master degré en: Ingénieur civil électromécanique

Année académique: 2012-2013

Mots clés: Interface Graphique, Robot de Réhabilitation, Génération de Trajectoire

Une anomalie de la démarche peut avoir une forte influence négative sur la qualité de vie d'une personne. Afin d'aider des patients avec une anomalie de la démarche un robot de réhabilitation, ALTACRO, a été développé à la Vrije Universiteit Brussel. Dans ce contexte ce mémoire traite du développement d'une interface graphique pour un appareil de réhabilitation.

Une interface graphique est très importante pour un appareil de réhabilitation. La présence d'une interface de ce type améliore la convivialité et la sécurité de l'appareil. Elle facilite le contrôle de l'appareil et la communication entre le robot et l'utilisateur.

La première partie de ce mémoire enquête sur les interfaces graphiques d'appareils de réhabilitation existants. Ces interfaces sont comparées et les options principales sont extraites de chaque interface. Ensuite plusieurs indicateurs de performance sont comparés. Ces indicateurs sont des variables qui donnent une indication de la performance du patient durant la session de réhabilitation. Ils sont comparés afin de voir quel indicateur donne le résultat le plus précis en combinaison avec un appareil de réhabilitation. Plusieurs environnements de programmation sont explorés, afin de voir lequel est le plus approprié pour réaliser une interface graphique pour un robot de réhabilitation.

Dans la deuxième partie de ce mémoire les exigences pour une interface graphique sont examinées. L'interface est nécessaire pour trois types d'utilisateurs: les ingénieurs, les kinésithérapeutes et les patients. Chaque groupe d'utilisateurs a des exigences particulières concernant le contenu de l'interface. Les exigences de chaque groupe sont examinées et énumérés. Pour les exigences des kinésithérapeutes et des patients cette analyse a été faite avec l'aide de kinésithérapeutes.

Dans la troisième partie de ce mémoire un outil de génération de trajectoire est développé. Ces trajectoires sont générées en fonction de plusieurs données d'entrée spécifique pour la réhabilitation de la démarche, comme la longueur du pas, la hauteur du pas et l'amplitude du mouvement de la hanche. Ces trajectoires générées doivent être cinématiquement et physiologiquement correctes. L'influence des différents paramètres de la démarche sur les trajectoires générés est ensuite explorée.

Dans la dernière partie de ce mémoire les connaissances acquises sont appliquées afin de réaliser une interface graphique pour ALTACRO. Cette interface est axée sur les besoins des ingénieurs. Un programme pour générer des trajectoires est également implémenté dans cette interface.

Samenvatting

Thesis titel: Ontwikkeling van een Grafische Gebruikersinterface voor een Rehabilitatie Exoskelet.

Auteur: Laura De Rijcke

Master graad in: Science in de Ingenieurswetenschappen: Werktuigkunde-Electrotechniek

Academisch jaar: 2012-2013

Sleutelwoorden: Grafische Gebruikersinterface, Rehabilitatie Robot, Traject Generatie

Een probleem in het stappatroon kan een sterke negatieve invloed hebben op de levenskwaliteit van een persoon. Om mensen met een dergelijk probleem te helpen werd aan de Vrije Universiteit Brussel een stap rehabilitatie robot ontwikkeld, namelijk ALTACRO. In het kader van dit onderzoek wordt in deze thesis de ontwikkeling van een grafische gebruikersinterface voor een rehabilitatie toestel onderzocht.

Een grafische gebruikersinterface is zeer belangrijk voor een rehabilitatie toestel. De aanwezigheid van een dergelijke interface verbetert de gebruiksvriendelijkheid en de veiligheid van een robot. Het vergemakkelijkt de controle over het toestel en de communicatie tussen de gebruiker en de robot.

Het eerste deel van deze master thesis onderzoekt grafische gebruikersinterfaces van bestaande rehabilitatie toestellen. Deze interfaces worden vergeleken en de belangrijkste opties van iedere interface worden opgelijst. Vervolgens worden er meerdere prestatie indicatoren vergeleken. Deze prestatie indicatoren zijn variabelen die een indicatie geven over de prestatie van de patiënt gedurende de rehabilitatie sessie. De vergelijking wordt gebruikt om te zien welke indicator de meest accurate resultaten geeft indien het gebruikt wordt in combinatie met een stap rehabilitatie toestel. Ten slotte worden verschillende programmeer omgevingen vergeleken om te zien welke het meest geschikt is om een grafische gebruikersinterface voor een rehabilitatierobot te maken.

In het tweede deel van deze thesis worden de eisen waar een grafische gebruikersinterface aan moet voldoen onderzocht. De interface is nodig voor drie verschillende gebruikersgroepen: de ingenieurs, de kinesisten en de patiënten. Deze gebruikersgroepen hebben elk specifieke eisen betreffende de inhoud van de gebruikersinterface. De noden van iedere gebruikersgroep worden onderzocht en opgelijst. Voor de eisen van de kinesisten en van de patiënten werd dit gedaan in samenwerking met kinesisten.

In het derde deel van deze thesis wordt een traject generatie programma ontwikkeld. Deze trajecten worden gegenereerd op basis van meerdere inputs, specifiek aan een stap rehabilitatie therapie, zoals de stap lengte en hoogte en de amplitude van de heupbeweging. Deze gegenereerde trajecten moeten kinematisch en fysiologisch juist zijn. In het vervolg van deze thesis wordt de invloed van de verschillende stapparameters op de gegenereerde trajecten onderzocht.

In het laatste deel van deze master thesis wordt de verzamelde kennis gebruikt om een grafische gebruikersinterface voor ALTACRO te maken. Deze interface focust zich voornamelijk op de ingenieursnoden. Er wordt eveneens een traject generatie gedeelte in deze interface geïmplementeerd.

Contents

Acknowledgment.....	iii
Abstract.....	iv
Résumé.....	v
Samenvatting.....	vi
Contents.....	vii
List of abbreviations.....	ix
List of figures.....	x
List of tables.....	xii
1. Introduction.....	1
1.1. Motivation.....	1
1.2. State of the art.....	2
1.2.1. Existing GUI's of rehabilitation devices.....	2
1.2.2. Performance indicators.....	11
1.2.3. Programming environments.....	15
2. Specifications of a GUI.....	17
2.1. Contents of a GUI: interview with a physiotherapist.....	17
2.1.1. Important factors for the rehabilitation process.....	17
2.1.2. Data to display and/or save.....	18
2.1.3. Patient feedback methods.....	18
2.1.4. Some possible extra settings.....	19
2.2. Requirements of a GUI classified by user type.....	19
2.2.1. The different users.....	19
2.2.2. The engineers.....	20
2.2.3. The physiotherapist.....	20
2.2.4. The patient.....	21
3. Trajectory generation.....	22
3.1. Gait phases.....	22
3.2. Requirements.....	23
3.3. Trajectory generation based on previously measured joint angles.....	25
3.4. Trajectory generation independent of previously measured joint angles.....	26
<i>Development of a graphical user interface for a rehabilitation exoskeleton</i>	vii

3.5.	Influence of the input parameters on the generated joint angles.....	28
3.6.	Conclusion.....	31
4.	Practical implementation: realization of a GUI for ALTACRO.....	33
4.1.	ALTACRO: conventions	33
4.1.1.	Robot DOF's.....	34
4.1.2.	Robot frame definitions	34
4.1.3.	Winter Conventions	36
4.1.4.	MACCEPA conventions.....	37
4.1.5.	Link between the different conventions.....	39
4.2.	Programming of the interface.....	39
4.2.1.	The main structure.....	39
4.2.2.	The detailed code	41
4.3.	Operation of the interface	45
4.3.1.	Structure of the GUI.....	45
4.3.2.	How to use the GUI?	49
5.	Conclusions.....	54
6.	Future work	55
7.	Sources	56
	Attachments.....	60
	Attachment 1: Enlargements of Figure 2 and Figure 3 representing the settings of the GUI of Lokomat.....	60
	Attachment 2: Comparison between programming environments for making GUI's.....	62
	Attachment 3: Matlab code for the trajectory generation	64

List of abbreviations

ALTACRO	Automated Locomotion Training using an Actuated Compliant Robotic Orthosis
BWS	Body Weight Support
CP	Control Parameter
CS	Coordinate System
DOF	Degree Of Freedom
GUI	Graphical User Interface
HHR	Heilig Hart Hospital Roeselare
HR	Heart Rate
MACCEPA	Mechanically Adjustable Compliance and Controllable Equilibrium Position Actuator
MAS	Motor Assessment Scale
MRMI	Modified Rivermead Mobility Index
PCI	Physiological Cost Index
RAGT	Robotic Assisted Gait Training
RTS	Real Time Signal
THBI	Total Heart Beat Index
VB	Visual Basics
VUB	Vrije Universiteit Brussel

List of figures

FIGURE 1: PHOTO OF THE LOKOMAT DEVICE [8]	3
FIGURE 2: SCREENSHOT OF THE PATIENT SETTINGS TAB OF THE LOKOMAT [8].....	5
FIGURE 3: SCREENSHOT OF THE TRAINING SETTINGS OF THE LOKOMAT [8].....	5
FIGURE 4: OPTOGAIT [9], [10]	6
FIGURE 5: FEEDBACK SYSTEM OF OPTOGAIT WITH ABOVE THE CAMERA IMAGES, IN THE MIDDLE THE GRAPHICAL FEEDBACK AND ON THE BOTTOM THE NUMERICAL VALUES [9].	8
FIGURE 6: PABLO [5]	9
FIGURE 7: SCHEMATIC HEART RATE PLOTTED AGAINST TIME DURING EXERCISE [13]	13
FIGURE 8: THE DIFFERENT PHASES IN A GAIT CYCLE [23]	23
FIGURE 9: VERTICAL MOVEMENT OF THE TRUNK (UPPER CURVE), KNEE FLEXION (MIDDLE CURVE) AND PELVIC OBLIQUITY (BOTTOM CURVE) ROTATION ANGLES (IN DEGREE) FOR ONE GAIT CYCLE [24]	24
FIGURE 10: PELVIC ROTATION (IN DEGREE) FOR ONE GAIT CYCLE [25].....	25
FIGURE 11: GENERATED TRAJECTORY FOR THE HEEL FOR A STEP LENGTH OF 50 CM AND A FOOT CLEARANCE OF 10 CM	27
FIGURE 12: GENERATED TRAJECTORY FOR THE TOES FOR A STEP LENGTH OF 50 CM	27
FIGURE 13: THE ALTACRO REHABILITATION DEVICE [30]	33
FIGURE 14: ROBOT ANGLES: FRONT VIEW.....	35
FIGURE 15: ROBOT ANGLES: SIDE VIEW.....	36
FIGURE 16: WINTER ANGLE DEFINITIONS.....	37
FIGURE 17: MAIN STRUCTURE OF THE PROGRAMMING OF THE INTERFACE	40
FIGURE 18: PROGRAM STRUCTURE: THE ORANGE ARROWS CORRESPOND TO ACTIONS WHICH SEND THE EXECUTION TO THE CASE STRUCTURE. THE BLACK ARROWS CORRESPOND TO THE FLOW IN THE SECOND WHILE LOOP.	44
FIGURE 20: ILLUSTRATION OF THE DIFFERENT TAB LEVELS IN THE GUI.....	45
FIGURE 19: OVERVIEW OF THE STRUCTURE OF THE GUI	45
FIGURE 21: 'TRAJECTORIES' SUB TAB OF THE GUI	47
FIGURE 22: 'REAL TIME MEASUREMENTS' SUB TAB OF THE GUI.....	48

FIGURE 23: FLOWCHART FOR A CORRECT USE OF THE GUI: IN YELLOW THE TAB NAMES, IN BLUE THE MANDATORY ACTIONS AND IN SALMON THE OPTIONAL ACTIONS. 49

FIGURE 24: STARTING OF THE PROGRAM: THE ARROW IS INDICATING THE 'RUN' BUTTON IN LABVIEW 50

FIGURE 25: PATIENT TAB OF THE GUI: BUTTONS TO SELECT AN EXISTING PATIENT FILE 51

FIGURE 26: PATIENT TAB OF THE GUI: BUTTONS TO SELECT A NEW PATIENT FILE..... 52

FIGURE 27: HEADER OF THE GUI WITH THE TCP/IP CONNECTION BUTTONS HIGHLIGHTED..... 52

FIGURE 28: SCREENSHOT OF THE PATIENT SETTINGS TAB OF THE LOKOMAT [8]..... 60

FIGURE 29: SCREENSHOT OF THE TRAINING SETTINGS OF THE LOKOMAT [8]..... 61

List of tables

TABLE 1: GUI SETTINGS AND POSSIBILITIES OF THE LOKOMAT [7]	3
TABLE 2: GUI PROPERTIES OF OPTOGAIT [8]	7
TABLE 3: GUI PROPERTIES OF PABLO [10].....	9
TABLE 4: COMPARISON BETWEEN THE OPTIONS OF THE DIFFERENT GUI'S.....	10
TABLE 5: COMPARISON OF THE PERFORMANCE INDICATORS	14
TABLE 6: INPUTS OF THE GENERATED TRAJECTORIES.....	28
TABLE 7: EXTREMES OF THE GENERATED JOINT ANGLES	29
TABLE 8: INFLUENCE OF THE STEP HEIGHT ON THE JOINT ANGLES	29
TABLE 9: INFLUENCE OF THE HEEL OFF HEIGHT ON THE JOINT ANGLES.....	30
TABLE 10: INFLUENCE OF THE AMPLITUDE OF THE PELVIC OBLIQUITY SINUS ON THE JOINT ANGLES.....	30
TABLE 11: INFLUENCE OF THE AMPLITUDE OF THE PELVIC ROTATION SINUS ON THE JOINT ANGLES.....	30
TABLE 12: INFLUENCE OF THE AMPLITUDE OF THE TRUNK SINUS ON THE JOINT ANGLES.....	31
TABLE 13: MACCEPA ANGLE DEFINITIONS	38
TABLE 14: LINK BETWEEN THE ANGLE DEFINITIONS.	39
TABLE 15: COMPARISON BETWEEN DIFFERENT PROGRAMMING ENVIRONMENTS.....	62

1. Introduction

1.1. Motivation

Gait impairment has a major impact on health-related quality of life. The inability or reduced ability to walk affects a person's possibility to perform activities of daily living, induces physical deconditioning and puts strain on his/her psychological and psychosocial well-being. Rehabilitation of gait is essential for promoting recovery and improving the quality of life. The physical therapy involved varies significantly depending on the cause and nature of the gait impairment. Prevalent causes of gait impairment are neurological disorders and injuries, such as stroke, spinal cord injury, multiple sclerosis and Parkinson's disease. The burden of these disorders on the worldwide public health and the related demands for health resources are mostly underestimated and are assumed to keep rising in the next years [1].

Gait rehabilitation sessions are very exhausting for a physiotherapist. They really have to do a lot of efforts to move the legs of the patient during the rehabilitation session. In order to facilitate the work of the physiotherapists and to help patients suffering from gait impairment with their rehabilitation, a five year concerted research action project ALTACRO (Automated Locomotion Training using an Actuated Compliant Robotic Orthosis), has been started in 2008 at the VUB (Vrije Universiteit Brussel). The research is mainly focused on four aspects of robot-assisted gait training [2]:

- Active ankle assistance
- Balance and load distribution
- Functional, three dimensional gait kinematics
- Physical human robot interaction

The focus of this work is on the design and realization of a Graphical user interface (GUI) of ALTACRO. A GUI is very important for medical robots. This has several reasons [3]:

- First, a medical robot is usually a complicated system that may include precise mechanical parts, advanced electronic circuits, an automatic control system and sophisticated computer software.
- Second, the users of medical robots may not have any expertise in engineering.
- Third, medical robots have special requirements on operation and working environments.

With the integration of a well-designed GUI the handling of the robot is simplified and the requirements are fulfilled in a safe way. During the therapy the physiotherapist is the decision maker of the evolution of the session and of the movements that the patient has to accomplish. The GUI has to provide him the available choices, in a readable way, to perform this operation well. By making a well-designed GUI, the possibilities for erroneous user actions would be reduced to a minimum.

Also the feedback to the patient about their performance and motivating exercises are key aspects of the successful deployment of robotic systems within routine clinical use. A GUI can provide this feedback as it facilitates interaction between the stroke patient and the robotic system during the treatment session [4]. By training the motoric, sensory and cognitive deficits in a playful manner, the motivation and

attentiveness of the patient, which is thus encouraged, facilitates the repetitive exercising. The focus on the exercise is also reinforced by the software and patients relearn lost (or partially lost) abilities [5].

Care has to be taken that the devices, which are implemented to motivate the patient, do not distract him. Some pediatric rehabilitation centers that use robotic assisted gait training (RAGT) try to boost the motivation of patients by showing DVDs or by playing music. Such strategies may distract the patients (especially children) from the actual therapy, causing them to become less active in the Lokomat [6].

It is clear that the GUI must provide a readable way of feedback for the patient. This feedback has to be given in an optimistic way, to encourage the patient to do his exercises correctly. It can be done in different ways: e.g. graphs of the performance, results of a game, a smiley if the exercise was been done well, etc.

Another possibility is linking the exercise to a virtual reality (VR) environment [6]. VR techniques make it possible to directly interlink the patients' motor performances during the gait training with actions in a virtual world, such as in a computer game. If the VR games are adequately adapted to children's needs, it provides motivation and yet keeps the focus on the actual gait training. It should be pointed out that the social interaction between the therapist and the participant undoubtedly plays a crucial role, especially for patients. Thus, the use of VR during rehabilitation therapy should not replace the physical therapist, but rather provide an additional means of enhancing training efficiency [6].

If the exercise is done in a playful manner, care has to be taken that the game does not become tiresome after having played it a lot. Attention has to be paid to the fact that the patients will use the revalidation device for a long time. So the patient interface has to encourage the patient to evolve during the complete revalidation period without becoming tired of it.

A GUI is also important for the engineers who are developing the robot and who are working with it. They need the GUI to control the robot and to check the data coming from the robot. This is needed as well during the development phase as during the debugging phase and the maintenance of the device.

1.2. State of the art

1.2.1. Existing GUI's of rehabilitation devices

Firstly a look is taken on graphical user interfaces (GUI's) of other commercial available gait rehabilitation robots. At the moment little is written over GUI's of gait rehabilitation robots, so it is useful to also take a closer look to GUI's of other devices, such as GUI's of arm rehabilitation devices. Many of the concepts of these GUI's can be compared to those of gait rehabilitation devices.

1.2.1.1. Lokomat

The Lokomat was built by a Swiss company, named Hocoma [7]. It is, at the moment, one of the most used and most performing gait rehabilitation robots. In Belgium there are only two hospitals that use them currently: the 'Heilig Hart' hospital at Roeselare and the Jessa hospital at Herk-de stad (Hasselt).

The Lokomat system is an electrically driven gait orthosis consisting of a hip support and two gait orthoses. The gait orthoses are each equipped with a hip and a knee joint drive. The Lokomat system is

mounted via a parallelogram on a swivel door. It is operated in combination with a treadmill and body weight support system. The Lokomat system is controlled via a PC [8]. See Figure 1 for an image of the Lokomat gait rehabilitation device.

The different settings and possibilities of the Lokomat software are listed in Table 1. They are extracted from the user manual of the Lokomat robot [8].



Figure 1: Photo of the Lokomat device [8]

Table 1: GUI settings and possibilities of the Lokomat [7]

	Possibilities	
Patient settings	Patient identification	
	Saving of patient data	
	Possibility of data transfer to text file	
Parameters to be set by the physiotherapist	Sliders for:	Speed
		Unloading (%)
		Hip angles
		Knee angles
		Guidance force of the robot on the patient (by reducing this parameter the patient can walk more freely)
	Patient coefficient: gives the synchronization between the Lokomat and the treadmill. If the physiotherapist sees that the patient's foot remains too long on the treadmill or if his foot slips backwards at the end of the stance phase, the physiotherapist can change this coefficient.	

Possible displays for the patient	Line graphics for hips & knees with as reference the unloaded Lokomat for:	Values of the joints during the stance phase
		Values of the joints during the swing phase
	Smiley's: focus on only one leg, joint or phase. The more the patient supports his own movement, the broader the smile.	
	Thermometer: focus on one joint/side	General performance: this is the average of all the biofeedback values for the last steps (see also further on the paragraph on the methods for the performance calculation)
Actual performance: the performance during the current step for the selected gait characteristics		
Display for physiotherapist	Graphs with data of the joints of the hips and the knees	Stance phase
		Swing phase
Possibility to put on a separate monitor:	Torque of Lokomat drive motors	
	The average of measured forces weighted with gait cycle phase (in biofeedback units)(see also paragraph "biofeedback system")	
	Two values per joint (swing & stance phase)	
Global	Preprogrammed training programs	
	Walking speed calculated as a function of the leg lengths	
	Possibility to save additional notes	
	Sensitivity adjustment: tolerated forces and variations from the gait pattern	

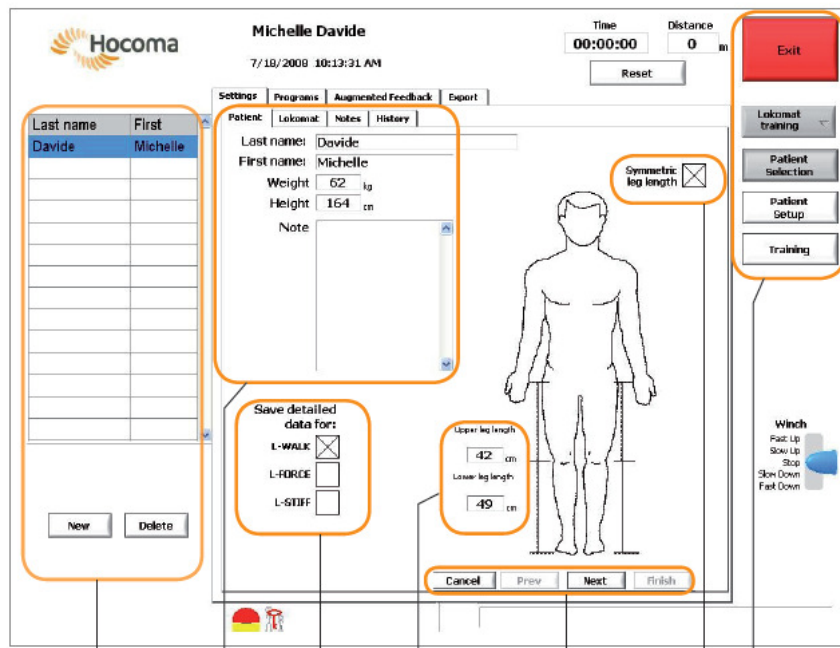


Figure 2: Screenshot of the patient settings tab of the Lokomat [8]

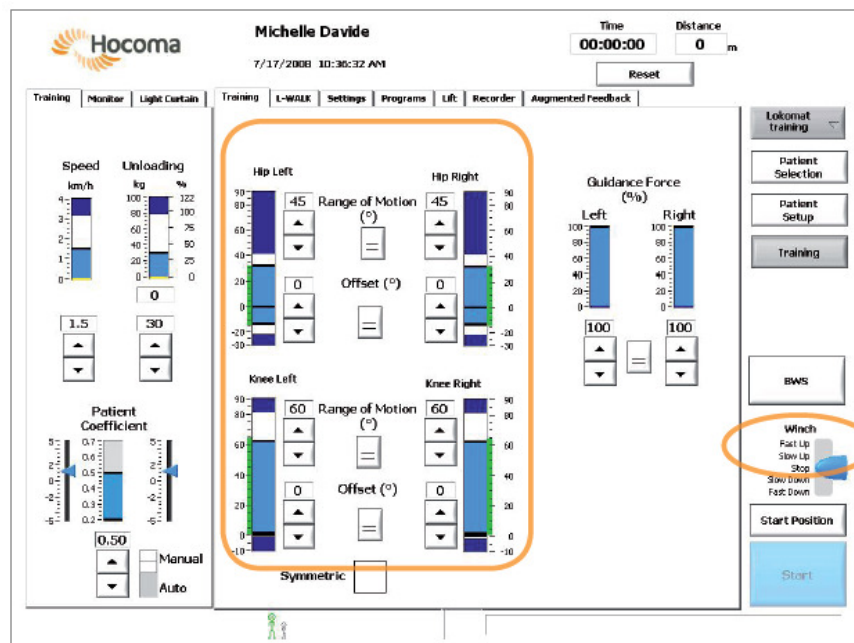


Figure 3: Screenshot of the training settings of the Lokomat [8]

A screenshot of the patient settings tab, with the patient identification settings, is displayed in Figure 2. A screenshot of the training settings for the Lokomat is given in Figure 3. The physiotherapist has to

impose the treadmill speed and the unloading percentage (later also called the body weight support percentage, BWS) at the left hand side of the screen.

The patient coefficient (at the left lower side of Figure 3) is a synchronization parameter between the Lokomat robot and the treadmill. It has to be changed if the patient's foot remains too long on the treadmill or if his foot slips backward at the end of the stance phase. In the middle of the 'training settings' screen, the range of motion and the offset of the hip and the knee angles can be changed. It is possible to give different values for the left and the right hip and knee angles. The last parameter that can be changed here is the guidance force of the robot that acts on the patient. By reducing this parameter the patient can walk more freely. A larger image of Figure 2 and of Figure 3 is given in attachment 1.

1.2.1.2. OptoGait

OptoGait is not really a rehabilitation device; it is made as a training instrument for athletes. OptoGait is a system for optical step detection constituted of a transmitting and a receiving bar. Each one contains 96 LEDs communicating on an infrared (visible) frequency with the same number of LEDs on the opposite bar. Once positioned on the floor or on the treadmill, the system detects the interruptions of the communication between the bars, caused by the patient's movement, and calculates the duration between two interruptions and the position of the feet [9]. The device is represented in Figure 4.



Figure 4: Optogait [9], [10]

It can be strange for the lecturer that this device is also taken in this state of the art. But this is also a device for gait training and it is interesting to take a closer look to the software. It gives a good idea how to display the different gait data in a readable way. It also gives some ideas of the way to give feedback to the user of the device. The possibilities of the software are listed up in Table 2. In Figure 5 a screenshot of the user interface of the Optogait device is shown. The different feedback options (camera images, graphical feedback and numerical values of the performance) are represented.

Table 2: GUI properties of OptoGait [8]

	Possibilities
Patient settings	Person identification
	Saving of personal data and notes
	Possibility to import and export data to other programs and formats (for example: xml, excel)
Displays for the user	Three kinds of feedback: Numerical, graphical or with a video (from a webcam) (see Figure 5)
Possibilities of analysis of the data after the test	Images of the video are automatically synchronized with the numerical/graphical data
	Speed of video reproduction can be changed
	Two or more tests can be compared
Global	Pre-defined tests or tests created by the user
	All the test data is saved (numerical, graphical and the video)
	Generates automatically a report after saving of the session data
	Gives a gait report with average values, standard deviation and variability coefficient for both legs or an extended report with all numerical and graphical data.

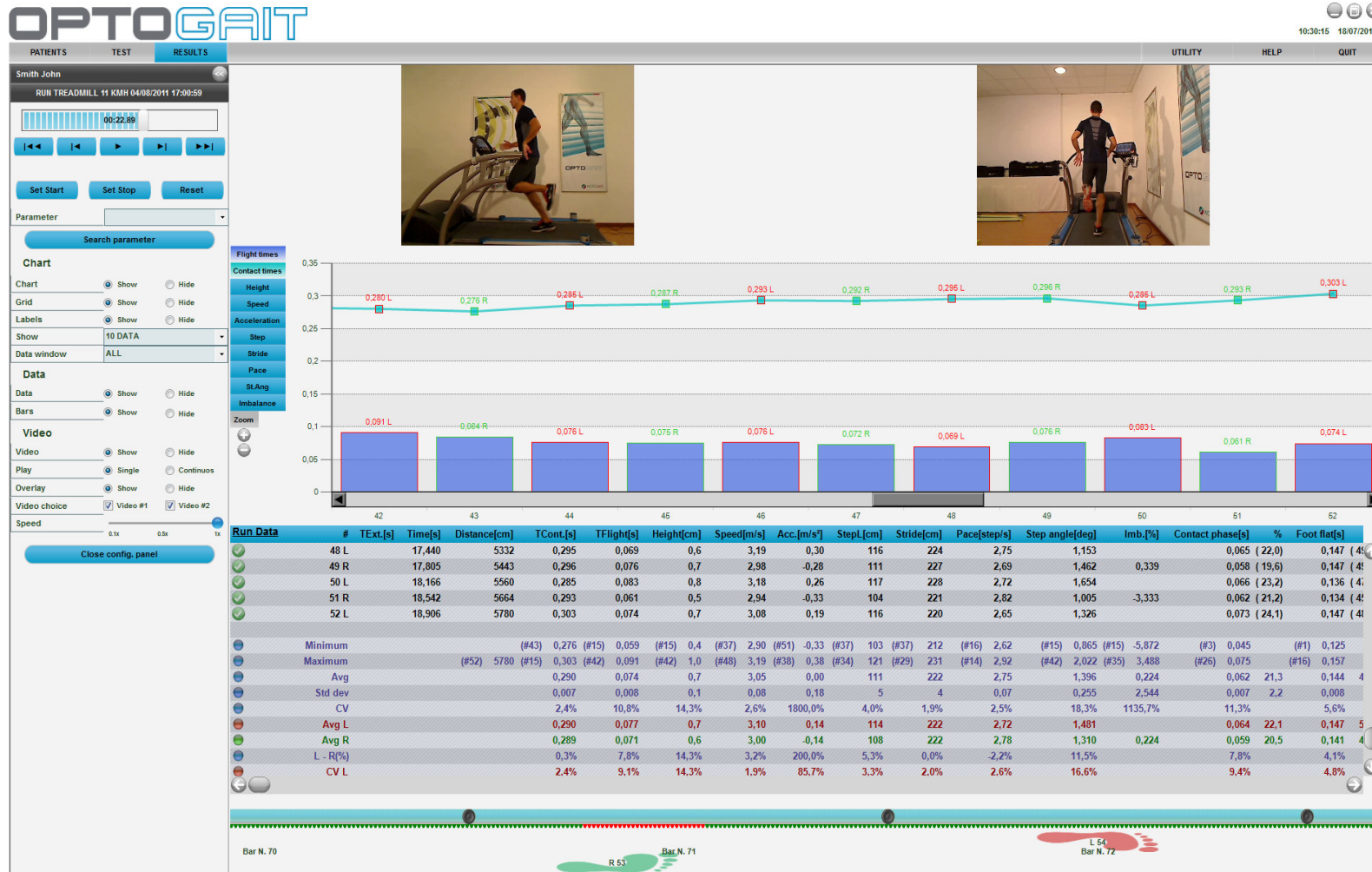


Figure 5: Feedback system of Optogait with above the camera images, in the middle the graphical feedback and on the bottom the numerical values [9].

1.2.1.3. Pablo [5]

Pablo differs from the previous rehabilitation devices as it is a hand and arm rehabilitation device, and not a gait rehabilitation device. However this device is considered in this state of the art as it has a lot of similarities with a gait rehabilitation device; so it could also be interesting to study the possibilities of the software of a device such as this one.



Figure 6: Pablo [5]

The main advantage of the software of Pablo is that there are different therapy games with different difficulties levels included in the software. These games ensure a maximum attention and motivation of the patient. The different levels of each game give the possibility to adjust the game to the patient's ability. As there are different games available the patient will not be so fast tired from his therapy as if there was only one single game available. The different GUI possibilities of Pablo are listed in table 3.

Table 3: GUI properties of Pablo [10]

	Possibilities
Patient settings	Patient identification
	Saving of patient data
	Possibility to export the report in pdf or excel
Displays for the user	Eight different interactive therapy games => results of the game are based on the chosen control settings (force or movement)
	Different difficulties in the game
Analysis possibilities	Report is made with graphics of the elbow, wrist and shoulder movements and with the different grip forces

	Possibility of calculating a force control index: compare the patient force with the force of a healthy person.
Global	Comparison between the right and the left extremity is possible
	Measure and evaluate the cylindrical grip force and the pincers grip
	Calculate the movement of the shoulder elbow and wrist
	All measurements are saved with date, time and length of the therapy
	Preprogrammed movements which has to be performed

1.2.1.4. Summary of the existing devices

The possibilities of the previous explained devices and the differences between the different devices are summarized in Table 4 (An “X” means that this option is present, a ‘/’ means that this option is not available). Only the most important parameters and differences are listed in order to have a clear overview.

Table 4: Comparison between the options of the different GUI's

Rehabilitation device	Lokomat	Pablo (Tyromotion systems)	OptoGait
Patient tab			
Patient identification	X	X	X
(Automatic) saving of patient data	X	X	X
Possibility of data transfer to a pdf file, a text file or an excel file	X	X	X
Way to adapt parameters	sliders	sliders	sliders
Feedback methods			
List of numerical values of parameters	/	/	X
Graphical feedback (graphic)	X	X	X
Other	smiley or thermometer	game results	video taken by a webcam
Training method			
Preprogrammed programs	X	games	X
Manual set program	X	/	X

By examining Table 4, one can see that these GUI's have a lot of common properties. From this comparison, a list of essential options for a useful GUI can be extracted:

- Patient identification (name, date of birth, leg parameters, height, weight, results of previous sessions)
- The possibility to export the results to a printable file (pdf, excel, ...)
- Not only a graphical feedback, but also a more 'funny'/readable kind of feedback (a game, or icon, or video)
- Need to have preprogrammed training programs that can be followed

These demands for a GUI will be integrated when a GUI for the rehabilitation robot ALTACRO will be made in chapter 5.

1.2.2. Performance indicators

To give feedback to the patient it is useful to have an indicator of his global performance during the rehabilitation session. It is also useful for the physiotherapists and for the objectiveness of the session evaluation to have an indicator about the patient's performance. There already exist multiple kinds of such indicators. In the next paragraph the utility of these indicators (sometimes also called bio-indicators) will be studied.

1.2.2.1. Modified Rivermead Mobility Index (MRMI)

The Modified Rivermead Mobility Index (MRMI) is a measure of disability related to bodily mobility. It demonstrates the patient's ability to move his own body. It does not measure the effective use of a wheelchair or the mobility when aided by someone else. It was developed for patients who had suffered a head injury or stroke at the Rivermead Rehabilitation Centre in Oxford England [11].

The mobility of the person is evaluated by 15 mobility-related questions as for example: "Can the patient move from bed to chair and back without any help?" or "Can the patient sit on the edge of the bed without holding on for 10 seconds". Points are attributed to the responses of the questions, with a score from 0 to 5 for each question. The global mobility of the patient is evaluated by his obtained total score. Note that these questions are based on the tools needed for moving.

1.2.2.2. Motor Assessment Scale (MAS)

The Motor Assessment Scale MAS is a measure of motor impairment and mobility in stroke patients. The MAS uses a seven-point ordinal scale to measure five mobility-related activities that are similar to the MRMI activities: rolling from supine to side-lying, rising from supine to a sitting position, balanced sitting, standing up from a sitting position and walking. Three additional items measure the impairment and function of the upper limb [12].

The way to evaluate the performance of the patient with the MAS is nearly the same as with the MRMI. Although the specific questions and the way to attribute the points are a little bit different, the principle of the method is to give points based on the result of questions. These questions are, as for the MRMI, based on the ability of the patient to move with or without external tools.

1.2.2.3. *Indirect Calorimetry*

Indirect Calorimetry is based on the assumption that all energy-releasing reactions in the body ultimately depend on oxygen uptake. The most common method of measuring oxygen uptake during exercise is by open-loop spirometry when exhaled air is sampled and analyzed for its oxygen content. Recent developments have led to lightweight, portable telemetric devices that are capable of performing breath-by-breath oxygen and carbon dioxide analysis. Oxygen uptake is expressed either in unit time (\dot{V}_{O_2} , in ml/(kg* min)) or in unit distance (oxygen cost, in ml/(kg*m)), which can be considered as a measure of metabolic efficiency [13].

The problem with \dot{V}_{O_2} measurements is that they are cumbersome to conduct, the instrumentation is expensive for a routine laboratory, and the measurements require trained personnel [14]. Also for the patient the session would become less comfortable which may affect his obtained results.

1.2.2.4. *Physiological Cost Index (PCI)*

The Physiological Cost Index (PCI) is calculated in beats per meter, by a combination of the heart rate (HR) of the patient and his walking speed. It is calculated as:

$$PCI = \frac{\text{Working HR} - \text{Resting HR}}{\text{Walking speed (meters/minute)}}$$

PCI can be used:

- to measure changes in locomotor efficiency over time
- to measure changes as a result of the use of different orthotic or prosthetic devices
- as an indicator of the handicap when compared with matched normative data [15]

The calculation of this indicator can easily be made by a rehabilitation device as it needs only the walking speed of the patient and the HR of the patient, which can easily be measured.

A negative point for this method is that the method works best at the preferred walking speed of the user. Since the patients are relearning to walk there would not be an important difference in some consecutive sessions as their walking speed would remain nearly the same.

The heart rate must also achieve a steady state. In unimpaired subjects, this state occurs when the cardiovascular system has adapted to the new physiologic demands, which occurs about the third minute of exercise. In a population with gait impairments, the effort of walking may be significantly higher and not be considered submaximal. Analysis was made with children with cerebral palsy and it was discovered that, in 9% of their subjects, the heart rate continued to rise during the walking trials. If a nonsteady-state heart rate is used as an alternative, the repeatability may be compromised [13].

1.2.2.5. *Total Heart Beat Index (THBI)*

To represent more accurately the total energy consumed during an activity, Total Heart Beat Index THBI investigates heart rate behavior throughout a period of exercise by finding the total number of heartbeats that occurred during the period. This is done with continuous heart rate monitoring, which is

now readily available with the development of portable heart rate monitors. This is illustrated in Figure 7 where:

- Area 1: The extra heartbeats required during the exercise
- Areas 1-2: The total number of heartbeats during the exercise including the basal level
- Area 3: The extra heartbeats that occur during the recovery phase
- Areas 3-4: The total heartbeats occurring during recovery [13]

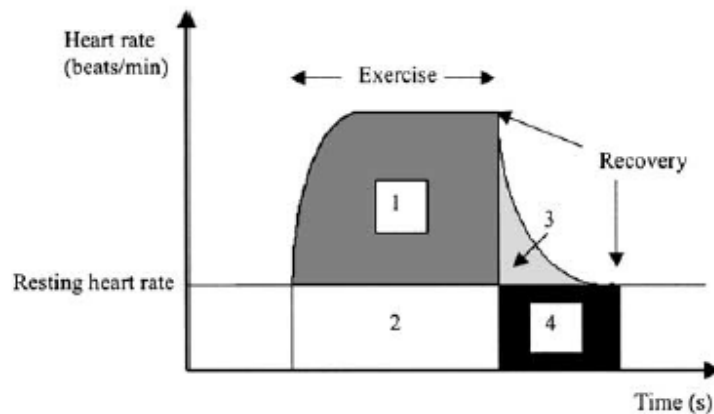


Figure 7: Schematic heart rate plotted against time during exercise [13]

An index based on the number of heartbeats provides a measure of a person's total energy consumption and this measure is independent of whether the activity is steady or non steady state. By including the recovery period, the repayment of the oxygen debt that occurs at the onset of exercise is included.

The THBI is easy to calculate and requires equipment that is readily available, comfortable to wear, and noninvasive. Repeatability statistics found the THBI to be comparable to oxygen cost (indirect calorimetry) and better than the PCI. The THBI is sensitive to change in workload, with a profile similar to that of indirect calorimetry [13].

1.2.2.6. L-walk

The L-walk, is the system used by the Lokomat device. This system measures the torque of the Lokomat's drive motors. If the patient actively moves with the programmed gait pattern, the power required from the drives is small. The less active the patient is, or the more awkward the movements he makes (e.g. due to spasticity), the greater the drive force required to move his limbs according to the preprogrammed pattern. The values displayed on the Lokomat are the average values for the forces measured within the Lokomat drives weighted according to the gait cycle phase. Therefore, the values displayed are strongly correlated with the force or torque values. However, they are not displayed in Newtons or Newton meters, but in biofeedback¹ units. Two values are calculated for each driven joint; one for the swing phase, the other for the stance phase. The calculation method is designed in such a

¹ Biofeedback is a way for having information on physiological functions by measuring the activity of these functions. This can be done with brain waves, muscle tone or HR.

way that therapeutically desirable movements increase the values (e.g. actively swinging the leg forward during the swing phase (hip swing), actively extending and flexing the knee during the swing phase and actively extending the knee during the stance phase) [8].

Although this is a relatively efficient way of feedback one has to note that this indicator works best for a guidance force of 100% of the device. If the guidance force is less than 80% the changes of the biofeedback values would be much smaller. Note that it is better for the rehabilitation if the guidance force is lower. A lower guidance force means that the patient is more active.

1.2.2.7. Conclusion

A problem with the first two performance indicators (MRMI and MAS) is that they are calculated as a function of the tools that the patient needs to move. The tool that the patient will use is always a treadmill, so the global performance of a patient will nearly remain the same. Therefore these indicators are not so useful in this case and another way to measure the performance has to be found.

From the previous paragraphs it is clear that a performance indicator for a rehabilitation device as ALTACRO has to be based on biofeedback parameters instead of on a questionnaire. This performance indicator may be based on e.g. velocity measurements combined with HR measurements, as with the PCI, oxygen consumption rates, HR measurements, or based on the torque provided from the drive motors as for the L-walk.

In table 5 a comparison is made between the different performance indicators. Taking the fact into account that a questionnaire based performance indicator is not suitable to evaluate a performance with a gait rehabilitation device, an L-walk based method seems to be the most appropriated method in this case.

Table 5: Comparison of the performance indicators

Performance indicators	Legend of table 5				
	Questionnaire based	Instrumentation cost	Comfortable for the patient	Requirement of trained personnel	Repeatable results
MRMI	x	++	++	/	++
MAS	x	++	++	/	++
Indirect calorimetry	/	--	--	--	+
PCI	/	-	-	-	+-
THBI	/	-	-	-	+
L-walk (torque measurement of actuators)	/	++	++	/	++

1.2.3. Programming environments

The programming language/environment that is chosen to make the graphical user interface (GUI) will have a major importance. Not only for the ease of programming the GUI, but also for the possibilities and the lay-out of the final GUI. It is useful to examine first different programming environments before choosing one. Otherwise it is possible that the programming task has to be restart several times due to the fact of touching the limits of the programming environment. Hereafter the most used/meaningful environments for making a GUI are exposed. Other possibilities than the ones explained, can be find in attachment 2 with their pro's and contra's.

1.2.3.1. QT

Qt is an object-oriented GUI toolkit which allows programmers to choose between the Motif and the Windows look and feel. It implements its own widgets (user interface elements). Almost all classes and member functions in Qt are documented. The documentation is available in HTML, postscript, text and as manual pages [16]. Qt has a high portability and flexibility. It is also very fast and is available for free. Another advantage of QT is its mechanism with signals and slots, which eliminates the need for 'callbacks'², and provides a type-safe way to send arguments of any type. These concepts make it much easier to control the communication between objects in a flexible and maintainable manner than it is with a fragile callback style. The larger applications get, the more important this advantage of QT becomes [17]. It makes QT a very powerful component programming system.

1.2.3.2. Visual Basics

Visual Basics is an event-driven programming language developed by Microsoft to make applications for Microsoft and to make GUI's. The fact that it is a programming language made by Microsoft, makes it very difficult to export programs made with Visual Basics to other operating systems.

The biggest advantage of VB is for sure its ease of use for most programmers. VB is very straight forward for making GUI's as it has a GUI-based development tool with graphical aspects incorporated. It also offers an extremely rapid development tool for applications, compared with other programs. Another advantage of VB is that it is widely used, so there are a lot of tutorials, examples, and online forums that can be consulted in case of problems.

A disadvantage of VB is that it uses a lot of memory, as well disk space for the initial installation as in order to function efficiently after installation. This makes that VB is not recommended for making programs that use a lot of processing time [18].

1.2.3.3. Matlab

Matlab is an object-oriented, fourth generation programming language developed by MathWorks. This programming language is especially used for numerical computing. It allows performing numerical calculations in an easy way and the visualization of results, without the need for complicated and time consuming programming. It allows users to accurately solve problems, produce graphics easily and

² A 'callback' is a pointer to a function which may be executed in the first function. For example if you push the 'close' button you want that the 'close' function is executed wherever you are in the program. These 'callbacks' are not type-safe. It is never sure that the processing function calls the 'callback' with the correct arguments. [16]

produce code efficiently [19]. This makes Matlab an efficient program to do all kind of mathematical operations and for displaying graphics in an easy way.

The main disadvantage of Matlab is that it uses a large amount of memory so it can become very slow. It uses also the maximum CPU time that Windows allows it to have which makes real-time applications very complicated [20]. Another disadvantage is that Matlab is not so optimal for making GUI's. The programming of it is rather complex. Each time a control is placed, the programmer has to manually compute his location. For moving a control the programmer has to recompute all the object positions, instead of only drag and drops it, as is possible in the other programming environments which are presented here. A last disadvantage of Matlab is that it is almost not possible to make a stand alone executable, so Matlab always has to be installed. Matlab is a very expensive program to buy for the customers.

1.2.3.4. LabView

LabView is a visual programming environment. It is also one of the most used languages for making industrial instrumentation-automation programs. LabView is especially recommended in real-time applications as it provides as well data acquisition tools, data analysis tools as data visualization tools. LabView works with dataflow programming, so each node is activated as soon as data is available at all of its inputs [21]. LabView programs can also easily be connected with other programs, such as with Microsoft Excel, Matlab or Mathematica.

It is possible to create distributable EXE's and DLL's that runs with the (freely available) LabView run-time engine. LabView programs can be executed on multiple operation systems, such as Windows, Mac OS, Linux and RTOS. In addition of the previous advantages, a LabView tool exists which permits the user to make lightweight, web-based applications. These web-based applications serve as GUI, which allows the user to control systems from a web browser.

Another benefit of LabView is the extensive support for accessing instrumentation hardware. The provided driver interface saves program development time [22]. LabView is linked to National Instruments hardware which makes it easy to interact between the hardware and the software.

1.2.3.5. Conclusion

Although programs like Qt and VB provide a lot of advantages for the ease of programming a GUI, LabView will be used for making the GUI. The main reason is that a GUI for a rehabilitation device has to provide real time results during the complete training session. LabView has proven itself to be robust for programming real-time applications. LabView is also well-designed for making easily connection with the instrumentation hardware which is important when making a real-time GUI.

Matlab is very easy and robust for performing mathematical calculations. As Matlab is especially designed for doing calculations it would be easier to use Matlab instead of LabView for eventually complicate calculations, so Matlab would be used if some pre/post-session calculations have to be made. It is also easy to move data from LabView to Matlab or from Matlab to LabView, so these two programs are a good combination for making a GUI for a gait rehabilitation device.

2. Specifications of a GUI

2.1. Contents of a GUI: interview with a physiotherapist

To understand clearly the needs and the wishes of the physiotherapists, which will work a lot with ALTACRO, an interview was made with Doctor Piet Mortelé. Doctor Mortelé is the head of the department of 'physical medicine locomotory rehabilitation' at the Heilig Hart Hospital of Roeselare (HHR). He is specialized in active gait rehabilitation, and in his service they have been working with the Lokomat device for approximately five years. The interview was made in the offices of the HHR on the 26th of November 2012. To have more objective results some other physiotherapists of the department were called during the interview to also have their point of view in the different subjects. In what follows a summary of the main results will be made.

2.1.1. Important factors for the rehabilitation process

In the first part of the interview the most important factors in a rehabilitation session were examined. These are the factors which counts the most for the physiotherapist to evaluate the quality of the exercise that the patient carried out. The two main factors are:

- The body weight support (BWS)
- The quality of the carried gait phases

2.1.1.1. *The body weight support (BWS)*

The most important factor for a physiotherapist is the percentage of body weight support (BWS) that the patient is carrying himself. In the beginning of the rehabilitation process the patient is not able to support his own weight. This factor gives a good indication to the physiotherapist how far the patient is evolved in his rehabilitation process. This BWS can be done by a harness or by the stiffness of an exoskeleton robot, but it is mandatory for the rehabilitation that it is present. The measurement of the BWS is easy if it is done by a harness, but becomes much more complicated if it is done by the stiffness of the robot.

For this reason it was also examined if it should be possible to replace the BWS by another parameter, with a kind of a performance indicator for example. A lot of performance indicators are developed until now (such as the modified Rivermead mobility index (MRMI) [11] or the Motor Assessment Scale (MAS) [12]), but none of them are really indicated to be used with a device as ALTACRO or Lokomat. The reasons that they are not indicated is that their calculations are related to the tools that the patient needs to walk (such as crutches for example), so you will always get similar results for the indicator as the device remains the same.

Another possibility could be to use a kind of bio-indicator. A bio-indicator is an indicator that takes for example the effort, the force or the energy consumed by the patient into account to evaluate the performance. More information about bio-indicators is given in paragraph '1.2.2 performance indicators'. These bio-indicators give a good view on the performance of the patient and some of these bio-indicators are easier to measure than the BWS.

2.1.1.2. The quality of the carried gait phases

A second element that is necessary to see during the session is the quality of the performed gait phases. With the currently available devices (such as the Lokomat) this is mainly evaluated by the therapist who looks at the movements of the patient. Unfortunately this is not a very objective indicator. The quality can also be evaluated by the software by giving the difference between the imposed trajectory and the really performed trajectory or with the L-Walk function of the Lokomat (see the paragraph '1.2.2.6. performance indicators'). This quality measurement is important because, during the therapy, the aim is to reduce the BWS, but this may not be done at the expense of the quality of the gait trajectory.

All the other parameters that can eventually be displayed during a session (such as the hip, ankle and knee angles and torques) may be interesting as they give the physiotherapist the ability to see if the patient is doing his exercises well, but they are not mandatory for Doctor Mortelé.

2.1.2. Data to display and/or save

To be able to evaluate the progress of a patient some data of the session have to be saved. These data are:

- The personal data of the patient: his name, weight, length, leg dimensions, eventually an asymmetry of the legs
- The percentage of BWS that the patient carried
- The velocity at which the patient walked: The mean velocity of treadmill walking during a session is 2km/hour.

The time of walking may eventually be displayed. Normally this is a normalized time of 45 minutes for each session. In one week a patient may have two or three sessions. For the patient's first sessions the walking time may be less than 45 minutes if the patient has not yet the endurance to complete an entire session.

The other data, such as graphics with the data of the joints, the torques in the joints or the data of the gait are not very important to be saved. They are only subject of interest during the session itself.

2.1.3. Patient feedback methods

A last important point is that it would be nice to have some direct feedback methods for the patient. This is because, although there is always a physiotherapist in the environment of the patient, this physiotherapist could not always be attentive to all the movements of the patient. In this way it would be useful if there is some signal if the patient is deviating a lot from the desired gait trajectory. This signal can be a visual signal, such as a led which starts blinking, or an auditory feedback, for example the device says the patient that he has to lift his left ankle a little bit more.

By doing so care must be taken to not overuse this kind of feedback. If a led is blinking almost all the time to say that the movement is not executed well enough, the patient would become dispirited and after a time the physiotherapist would not be aware anymore of the led. The same happens if the auditory feedback is exorbitant: both the patient as the physiotherapist would be tired to hear the

device all the time. So by using immediate visual or auditory feedback an equilibrium must be found between too much and too less feedback.

2.1.4. Some possible extra settings

As Dr. Mortelé has a lot of experience with the Lokomat device, a lot of time the discussion was referred to the Lokomat. Starting from the Lokomat device, it was analyzed if some extra settings, which are currently not available in the Lokomat, would be useful to have in a gait rehabilitation device.

A first possibility is to exaggerate the movements of the gait during the rehabilitation session. For Dr. Mortelé this would be a very bad thing to do. With the gait rehabilitation it is the aim to relearn patients how to walk. If an exaggerated movement is applied to the patient during the rehabilitation process, the patient would learn to always walk with this exaggerated movement. So applying an exaggerated movement would probably have a counter-productive effect on the patient.

A second possibility was the implementation of three dimensional gait kinematics of the hip. For Dr. Mortelé this will be a nice improvement for a gait rehabilitation device. By implementing a three dimensional guidance of the hip the patient can relearn a much more natural gait pattern than if the guidance is only in two dimensions. In this way it would be interesting to display the desired and the real trajectory imposed to the hip. It would also be interesting to display the difference between the left and the right hip displacement, especially for patients suffering from hemiplegia. Care has to be taken that the imposed trajectory is physiologically correct. If this is not the case a three dimensional hip guidance can engender more problems than before.

A third possibility is the implementation of active ankle assistance. This seemed also to be a good improvement for a gait rehabilitation device. For the current rehabilitation processes the ankle is mostly not taken into account. At the end of the rehabilitation this is often solved by implementing an orthosis. It should be a nice improvement if this active ankle assistance is combined with the gait rehabilitation.

2.2. Requirements of a GUI classified by user type

This chapter describes the required options of a graphical user interface (GUI) of a rehabilitation robot. Firstly the different users (namely the engineers, physiotherapists and the patients) are examined. Then the specific needs and requirements of each type of users are developed. Out of these needs the different options that need to be implemented in the GUI are extracted.

2.2.1. The different users

As stated before there are three different kind of people that uses the GUI:

- The engineers who develops the robot
- The physiotherapist who supervises the patient
- The patient who has to use the device for his/her rehabilitation

These three kinds of users all have specific needs. They use the GUI for different things so it is possible that different GUI's need to be made or that a single GUI has different tabs and/or screens to fulfill in everybody's needs.

2.2.2. The engineers

These are the people who will develop the rehabilitation device. They will mainly need an interface to test the behavior of the rehabilitation device. This is as well needed during the elaboration of the device, as later on, for maintenance of the device. They need a user interface which represents all the technical data in a readable way. This technical data consists of multiple parts.

- A communication part: This part is handling the settings which are needed to establish a correct communication between the device and the user interface. By making this part clear and easily accessible eventual communication problems can be quickly detected.
- A trajectory part: This part is handling and/or generating the trajectory's which are imposed on the joints of the rehabilitation device. If the trajectories are generated by the interface, they must be kinematically correct and physiologically possible.
- A technical input part: This part is handling the technical inputs which are sent to the device. These are mainly the control parameters; e.g. the variables and the gains used in the control loops of the device.
- A feedback part: This part of the interface should represent the data which is coming from the device in a readable way. These data are for example the current and the torque in the different joints of the device.
- An error part: This part should alert the user if there is some malfunctioning of the device, such as communication errors, heating of the device, etc.

2.2.3. The physiotherapist

The physiotherapist needs an interface for different purposes:

- He must be able to easily and safely control the rehabilitation device. So he needs some kinematically and physiologically correct trajectories that can be sent to the device. For the user friendliness, the trajectories that will be sent to the device must be easily adaptable on some parameters that are important for the therapy, such as the step height and step length for a gait rehabilitation robot. He must also be able to control the robot on other session parameters, such as the treadmill speed and the body weight support level (if applicable for the device).
- Secondly, for the objectiveness of the session, the physiotherapist needs an objective way for evaluating the patient's performance. Ideally this evaluation is made out of two parts. The first part is an evaluation of the real time effectuated movements, such as a visualization of the difference between the desired and the real effectuated trajectory. This part gives an indication about the quality of the effectuated movements. The second part gives an indication of how far the patient is advanced in his rehabilitation therapy, such as with a bio-indicator. If it is the user interface which gives a session evaluation, the patient's session results would be independent of the present therapist.
- As a third point, for the user friendliness, it should be possible to save the patient data together with the session results. In this way it is easy for the physiotherapist to see the evolution of the patient's rehabilitation process session after session. Also the used session parameters, such as the step length, the step height and the amplitude of the hip movement for a gait rehabilitation robot, may be saved in the same file.

2.2.4. The patient

For an optimal rehabilitation of the patient, he must execute his exercises as actively as possible. It has been reported by Brüttsch [6] that the feedback of a GUI can be as positive for the evolution of a rehabilitation therapy as the feedback of a physiotherapist. By combining the feedback of a physiotherapist with that of a GUI the evolution of the rehabilitation therapy is even better than by using only one of both [6].

Another reason for the importance of the user interface for the patient is that, if the exercise is done as a game, the attentiveness and the motivation of the patient can be improved. If the patient is paying more attention to and motivation for his exercises, he will put more energy in the execution of them and have a better result.

3. Trajectory generation

A robot has many different joints. A trajectory has to be imposed on these joints to let the robot move. For a rotational joint for example the trajectory contains the successive angles that the joint must occupy. The generation of the trajectories is very important for a gait rehabilitation robot. The desired robot trajectories must:

- a. be synchronized with the treadmill speed
- b. be kinematically compatible with the patients dimensions
- c. have a kinematically correct gait pattern
- d. be similar to a human gait cycle
- e. be safe for the patient

This patient safety is mandatory, especially with exoskeleton robots, as the patient is in a certain way imprisoned in the robot and he cannot escape if the robot is executing wrong movements. So the gait rehabilitation device must follow trajectories that are physiologically compatible with the human gait cycle. If the trajectories are not physiologically compatible the robot may hurt the patient. This should absolutely be avoided. In this chapter a method will be developed for generating kinematically correct and physiologically compatible trajectories for a gait rehabilitation robot.

3.1. Gait phases

Before starting the actual trajectory generation the different phases in a gait cycle must be examined. For the purpose of trajectory generation the gait cycle is divided in four phases, according to Perry [23]. These different phases are visualized in Figure 8:

- Phase 1: A first double support phase:
 - The right foot is going from the initial heel contact to a flat foot on the ground
 - The left foot is going from a flat foot to the end of the stance phase, with the heel off the ground (the toes are still on the ground)
- Phase 2: A single support phase:
 - The right foot is in the stance phase: foot flat on the ground
 - The left foot is in the swing phase
- Phase 3: A second double support phase:
 - The right foot is going from a flat foot to the end of the stance phase, with the heel off the ground (the toes are still on the ground)
 - The left foot is going from the initial heel contact to a flat foot on the ground
- Phase 4: A second single support phase:
 - The right foot is in the swing phase
 - The left foot is in the stance phase

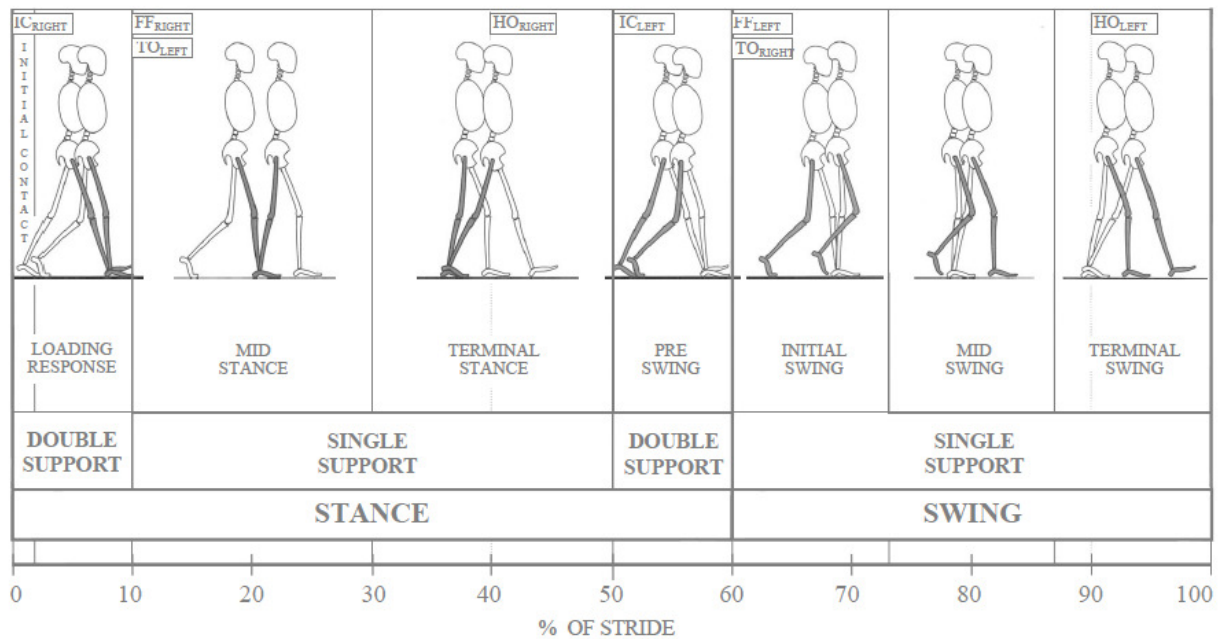


Figure 8: The different phases in a gait cycle [23]

3.2. Requirements

Beyond the fact that the generated trajectories must be kinematically correct (they must comply with the patients dimensions and with the constraints given by the presence of the ground/treadmill), they must also satisfy requirements coming from the physiological compatibility of the trajectory and from the wishes of the physiotherapist during the session. These requirements are:

- Requirements from the physiotherapist:
 - To impose a desired step length.
 - To impose a desired step height (also called the foot clearance).
 - To choose the walking speed of the patient. The walking speed will influence the ratio between the stance time to the swing time. This ratio decreases with an increase in walking speed [2].
- Requirements for the compatibility between the robot movements and the patient's physiology:
 - For the vertical movement of the trunk [24]:
 - The amplitude of the displacement is related to the walking speed: from 3 cm peak to peak for slow walking (± 0.8 m/sec) up to 8 cm peak to peak for fast walking (± 2.2 m/sec). For a mean walking speed (± 1.4 m/sec) the peak to peak amplitude of the displacement is typically 4 to 5 cm.
 - The displacement can be approximated by a double frequent sinus for one walking cycle. This double frequent sinus is a simple sinus which executes two sinus cycles during one gait cycle. The sinus starts at its mean value during the initial contact of the heel as is visualized in Figure 9.

- For the pelvic obliquity in the frontal plane [25]:
 - The pelvic obliquity during one gait cycle can be approximated by two superimposed sinus as illustrated in Figure 9.
 - At normal walking speed the mean amplitude of the pelvic obliquity is 7.72° peak to peak with a standard deviation of 2.26° .
 - The amplitude of the sinus changes as a function of the walking speed within a range from 6° to 20° [24].
- For the pelvic rotation (in the transverse plane):
 - The pelvic rotation during one gait cycle can be approximated by a simple sinus.
 - The sinus is starting at its maximum amplitude during the initial contact of the heel. This can be visualized in Figure 10.
 - The mean amplitude of the sinus is 10.40° peak to peak, with a standard deviation of 3.22° , as reported by Whittle [25]. The sinus is illustrated in Figure 10. The mean amplitude of the sinus is represented by the full line, the standard deviation is represented by the dotted lines.

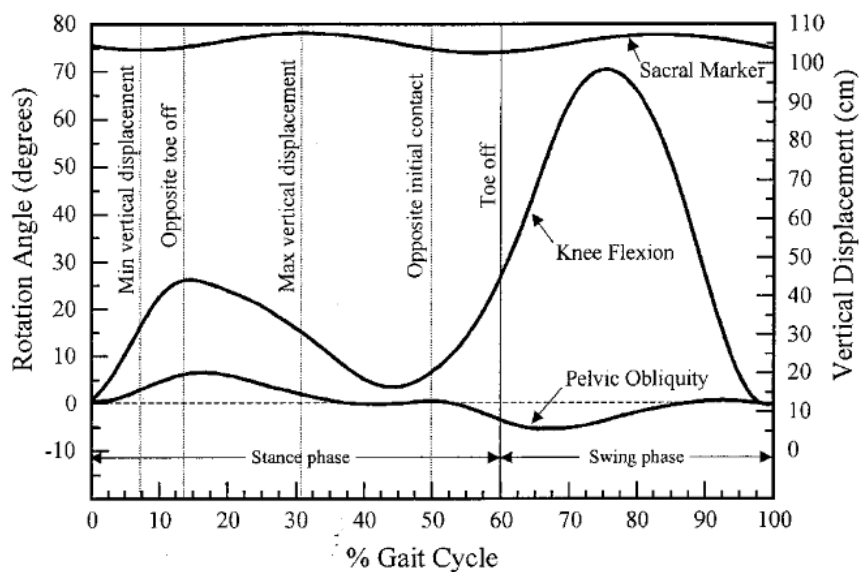


Figure 9: Vertical movement of the trunk (upper curve), knee flexion (middle curve) and pelvic obliquity (bottom curve) rotation angles (in degree) for one gait cycle [24]

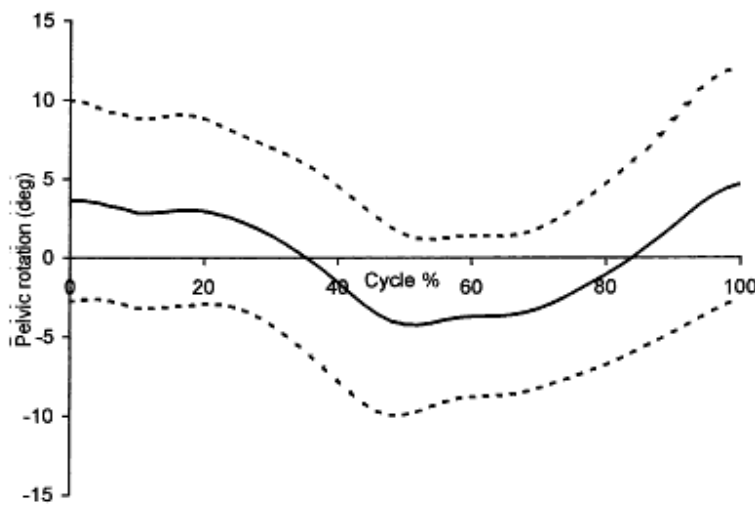


Figure 10: Pelvic rotation (in degree) for one gait cycle [25]

3.3. Trajectory generation based on previously measured joint angles

David Winter [26] investigated two dimensional human gait trajectories. For this he collected data about the two dimensional gait cycle. He measured the hip angle in the sagittal plane, the knee and the ankle angles in a normal human gait cycle. In a first approach to generate trajectories for a gait rehabilitation robot it seemed straightforward to start from this data. The angles measured by Winter are recorded independently of the step height and the step length. The idea was to start from this Winter data to make trajectories in two dimensions and expand further to three dimensional trajectories by implementing also the pelvic obliquity, the pelvic rotation and the movement of the trunk. The method in two dimensions is as follows:

- In two dimensions the pelvis is reduced to a single point.
- The Winter data is used to calculate the hip height during the stance phase.
- As the pelvis is reduced to a single point and one leg is in the stance phase when the other leg is in the swing phase, the calculated hip trajectory is also used for the leg in the swing phase.
- A trajectory is designed for the heel of the foot in the swing phase. For this a polynomial is generated with as constraints the required step height and the step length.
- The generated heel trajectory and the calculated hip trajectory are used as inputs. By using inverse kinematics the required hip and knee joint angles are calculated.

Different problems arose during the implementation of this procedure:

- The angles measured by Winter are recorded independently of the patient dimensions and the step length: When the desired step length was larger than the one used for the Winter data, it was kinematically impossible to follow the desired heel trajectory (with the given leg length the heel could not attain the desired trajectory).

- The Winter data is independent of the step length: Discontinuities arose for going from the stance phase to the swing phase as the stance phase was based on the Winter data and the swing phase on the desired step length.
- During the swing phase a trajectory was only imposed on the hips and on the heel. The ankle angle was given by the Winter data. Due to this combination the toes were sometimes asked to go through the ground level.

To solve these problems the procedure was extended as follows:

- The calculation of the hip height is done in the same way as before: the Winter data is used to calculate the height of the hip of the leg in the stance phase. This hip height is used, by calculating the inverse kinematics, to calculate the joint angles of the leg in the swing phase.
- A trajectory that covers the four phases of the gait cycle is imposed on the heel. This instead of only imposing a trajectory on the heel during the swing phase as before.
- Another trajectory is imposed on the toes. This trajectory covers also the complete gait cycle.

With these improvements the discontinuities were eliminated. Also the problem of the toes which passes through the ground level was solved. The problem of following the desired heel trajectories for larger steps remained. Due to this problem it was decided to discard the Winter data and to generate all the trajectories completely based on the physiological and kinematical constraints.

3.4. Trajectory generation independent of previously measured joint angles

First a two dimensional gait trajectory is generated. As before a trajectory is generated for the heel and for the toes. These trajectories are generated according to the four different phases in a gait cycle explained in section 3.1. The constraints for the heel and the toes in each of these phases are:

- In phase 1:
 - The heel is in contact with the ground
 - The toes go from their highest point (given as the place of the toes at the initial heel contact) to zero (ground level). This by taking the position of the heel and the length of the foot into account.
- In phase 2:
 - Both the heel and the toes are in contact with the ground
- In phase 3:
 - The heel is going from the ground level to the heel off position. The heel off height is defined as a variable and can be changed by the physiotherapist.
 - The toes remain in contact with the ground.
- In phase 4:
 - The heel is following a polynomial generated with respect to the desired step length and step height.

- The toes go linearly in the vertical (z) direction from the ground to their highest point (given as the place of the toes at the initial heel contact). Their longitudinal (x) position is defined by the trajectory of the heel, the length of the foot and the vertical position of the toes

An example of a generated heel trajectory can be visualized in Figure 11. A toe trajectory is given in Figure 12. Both trajectories are generated for a step length of 50 cm and a step height of 10 cm. The heel off height was 5 cm. Note that the heel is taken as the rear end at the bottom of the foot. The actual ankle joint is 8 to 10 cm above this point. The longitudinal (x) position is changing when the foot is on the ground because the figure is taken as if it was observed from beside the treadmill.

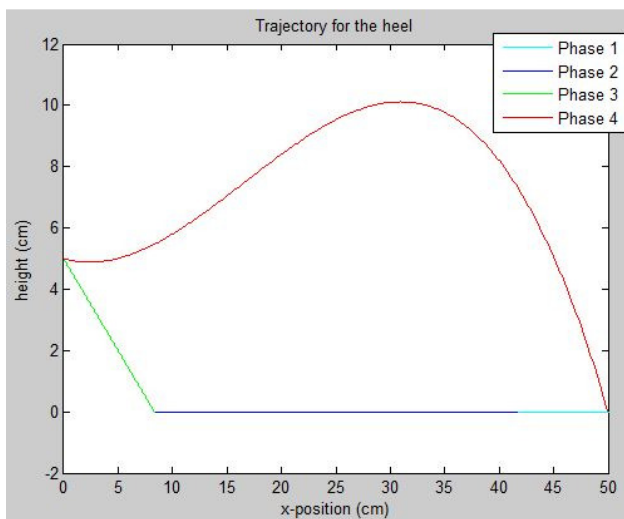


Figure 11: Generated trajectory for the heel for a step length of 50 cm and a foot clearance of 10 cm

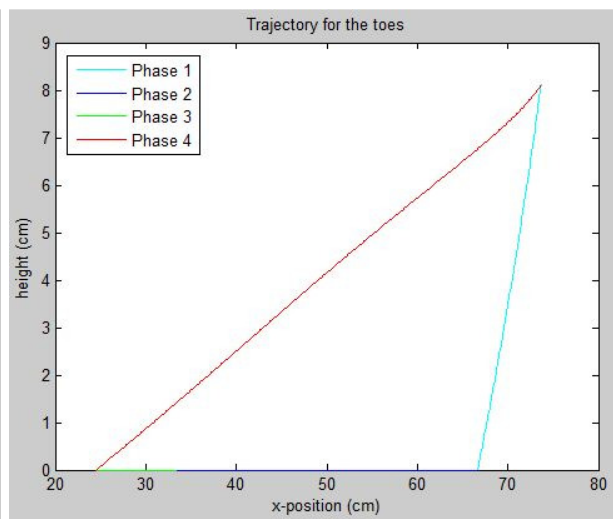


Figure 12: Generated trajectory for the toes for a step length of 50 cm

In the two dimensional trajectory generation the hip is represented by a fixed single point. The hip position is horizontally taken in the middle of the step length. The hip height is calculated as a function of the step length and the length of the leg. When the hip is taken as a fixed point, the heel is the furthest from the hip during the initial contact. The hip height at the initial contact can be calculated as:

$$H_{hip} = \sqrt{(L_{upper\ leg} + L_{lower\ leg})^2 - \left(\frac{Step\ length}{2}\right)^2}$$

With the hip height and the trajectories for the heel and the ankle as inputs, the hip, knee and ankle angles are calculated by using inverse kinematics. The exact kinematic formulations can be found in Attachment 3 (in the Matlab function: 'invers').

In this two dimensional trajectory generation the desired trajectories can perfectly be followed. There are no more problems of discontinuities or toes which pass through the ground. This trajectory generation tool now has to be extended to a three dimensional generation tool in order to have physiologically more compatible trajectories.

In the three dimensional trajectory generation the pelvis is represented by three points: one for the middle of the pelvis, one for the left extremity and one for the right extremity of the pelvis. On both pelvis extremities a trajectory is imposed in accordance with the physiological constraints:

A double frequent sinus with as standard amplitude 4.5 cm is imposed on the middle pelvis point for the vertical movement of the trunk. The amplitude is chosen as the mean displacement amplitude for a mean walking speed (see section 3.2). This amplitude can be changed by the physiotherapist. The offset from the ground (zero position) is the same height as the hip height calculated for the two dimensional trajectory generation.

For both extremity points this double frequent sinus is combined with a superposed sinus. This superposed sinus has a standard peak to peak amplitude of 8° in the z direction for the generation of the pelvic obliquity in the frontal plane. A simple sinus with standard peak to peak amplitude of 10° is imposed in the x direction for the transverse obliquity. The amplitudes of the different sinuses are chosen as the mean values during normal walking as given in the physiological requirements in section 3.2. They are taken as reference values but can be changed manually by the physiotherapist in order to adapt the gait pattern for different therapeutic sessions and for different walking speeds.

The angles of the two dimensional hip joint, the knee joint and the ankle joint are calculated by using inverse kinematics. The inputs of this inverse kinematics are the pelvis extremity trajectories and the heel and ankle trajectories. The exact kinematic formulas can be found in Attachment 3.

3.5. Influence of the input parameters on the generated joint angles

To verify the behavior of the previously developed trajectory generation tool a stick model is generated in Matlab. Firstly the reference values for the parameters, as found in section 3.2 are put in the program. This is done for three different step lengths: 47 cm, 60 cm and 78 cm. These step lengths are chosen as the smallest, mean and largest step length in a normal human gait cycle of people between 10 and 79 years, based on the results of Öberg [27]. The remaining input parameters of the simulation are given in Table 6.

Table 6: Inputs of the generated trajectories

Length upper leg	40 cm
Length lower leg	45 cm
Length feet	25 cm
Width hips	30 cm
Step height	10 cm
Heel off height	5 cm
Amplitude of the pelvic obliquity sinus	8°
Amplitude of the pelvic rotation sinus	10°
Amplitude of the trunk sinus	4.5 cm

The extremes of the generated joint angles are given in Table 7:

Table 7: Extremes of the generated joint angles

Step length (cm)	47	60	78
Min. hip joint angle (°)	-3.6	-2.5	-4.7
Max. hip joint angle (°)	36.7	42.0	50.4
Min. knee joint angle/knee extension (°)	0	0	6.6
Max. knee joint angle/knee flexion (°)	61.4	66.8	76.1
Min. ankle joint angle (°)	-0.5	-0.5	-0.5
Max. ankle joint angle (°)	27.7	33.2	41.4

Oberg [28] reported that the knee angles are maximum 68° during a normal gait cycle. This value is relatively close to the ones found for a short and a medium step length with the trajectory generation in table 7. The value of a large step length is a bit out of range. Oberg [28] reported that the maximum hip angle on a normal walking speed is around 47°. This value is also in the same range (cfr. Table 7), as the values found with the trajectory generation method. The ankle angle flexion range is around 30° as reported by Astephen [29]. The maximum ankle flexion is around 20° to 25°. The maximum ankle angle is acceptable for a short step length but has to be improved for higher step lengths.

The reason that some values are out of range for larger step lengths is that the other parameters, such as the amplitude of the sinus, are kept constant. It was already stated in section 3.2 that these parameters vary with the walking speed. As the step length increases for an increasing walking speed the other inputs should also be changed.

The influence of the other input parameters on the generated joint angles is now explored. The step length is always taken 60 cm. For each table there is only one parameter changing. The values of the other parameters are taken according to table 6.

Table 8: Influence of the step height on the joint angles

Step height (cm)	5	10	15
Min. hip joint angle (°)	-2.5	-2.5	-2.5
Max. hip joint angle (°)	34.1	42.0	49.1
Min. knee joint angle/knee extension (°)	0	0	0
Max. knee joint angle/knee flexion (°)	58.4	66.8	76.8
Min. ankle joint angle (°)	-0.5	-0.5	-0.5
Max. ankle joint angle (°)	33.5	33.2	33.0

The first parameter that is examined is the desired step height. As can be seen in Table 8 the minimum joint angles are independent of this parameter. The maximum hip and knee angles increase for an increasing step height. The maximum joint angle decreases for an increasing step height.

Table 9: Influence of the heel off height on the joint angles

Heel off height (cm)	2	5	10
Min. hip joint angle (°)	-2.1	-2.5	-2.5
Max. hip joint angle (°)	42.4	42.0	41.4
Min. knee joint angle/knee extension (°)	0	0	0
Max. knee joint angle/knee flexion (°)	65.2	66.8	71.0
Min. ankle joint angle (°)	-0.5	-0.5	-0.5
Max. ankle joint angle (°)	35.0	33.2	29.3

The second parameter that is examined is the heel off height. An increase of the heel off angle will decrease the maximum hip joint angle and the maximum ankle joint angle. By comparing table 8 and table 9 it is clear that the heel off height has much more importance for the maximum ankle joint angle than the step height. On the contrary the step height has much more effect on the maximal hip and knee angles. Notable (Table 9) is that the minimal hip joint angle is decreasing much for very low heel off heights. The maximal hip joint angle is decreasing very slowly for an increasing heel off height.

Table 10: Influence of the amplitude of the pelvic obliquity sinus on the joint angles

Amplitude of the pelvic obliquity sinus (°)	2	8	20
Min. hip joint angle (°)	-0.3	-2.5	-8.6
Max. hip joint angle (°)	44.6	42.0	40.7
Min. knee joint angle/knee extension (°)	0	0	0
Max. knee joint angle/knee flexion (°)	69.5	66.8	67.1
Min. ankle joint angle (°)	-0.5	-0.5	-0.5
Max. ankle joint angle (°)	30.5	33.2	38.3

The influence of the amplitude of the sinus of the pelvic obliquity is opposite to the influence of the step height. As can be seen in Table 10 the maximal hip and knee angles are decreasing for increasing obliquity amplitude. The maximal ankle joint angle is increasing for increasing obliquity amplitude.

Table 11: Influence of the amplitude of the pelvic rotation sinus on the joint angles

Amplitude of the pelvic rotation sinus (°)	5	10	20
Min. hip joint angle (°)	-1.6	-2.5	-4.1
Max. hip joint angle (°)	42.2	42.0	41.5
Min. knee joint angle/knee extension (°)	0	0	11.1
Max. knee joint angle/knee flexion (°)	66.8	66.8	66.8
Min. ankle joint angle (°)	-0.6	-0.5	-0.5
Max. ankle joint angle (°)	33.4	33.2	32.5

The influence of the amplitude of the sinus of the pelvic rotation on the joint angles is very small. For increasing amplitude of the pelvic rotation sinus, the maximum hip and ankle joint angles are slightly

decreasing. The maximum knee joint angle is approximately remaining the same with increasing rotation amplitude.

Table 12: Influence of the amplitude of the trunk sinus on the joint angles

Amplitude of the trunk sinus (cm)	2	4.5	10
Min. hip joint angle (°)	-1.0	-2.5	-7.9
Max. hip joint angle (°)	43.9	42.0	39.8
Min. knee joint angle/knee extension (°)	0	0	0
Max. knee joint angle/knee flexion (°)	68.7	66.8	66.5
Min. ankle joint angle (°)	-0.5	-0.5	-0.5
Max. ankle joint angle (°)	31.2	33.2	37.5

Lastly from table 12 it can be seen that the influence of the amplitude of the sinus on the trunk movement is almost the same as the influence of the amplitude of the sinus on the pelvic obliquity. The maximum knee and joint angles are decreasing for increasing amplitude of the sinus. The maximum ankle joint angle is increasing for increasing amplitude of the trunk sinus.

Before sending the generated trajectories to the rehabilitation robot a correction should eventually be made on the ankle angles, as they may not exceed 25°. This can be improved by changing several parameters: The heel off height, the step height and/or the amplitude of the sinus on the pelvic rotation can be increased, or the amplitude of the pelvic obliquity and/or the trunk sinus can be decreased. This can also be improved by changing the trajectory imposed on the toes. The trajectory imposed on the toes can especially be changed during the swing phase. At the moment it is assumed that the height is varying linearly between ground and the maximum height during the initial heel contact phase and that the horizontal position is changing according to the height position of the toes and the heel position. This linear assumption can be changed into other constraints.

3.6. Conclusion

The trajectory generation produces good trajectories for the hip and knee joints. The generated joint angles are in the same range as the values measured during a human gait cycle. The generated joint angles for the ankle joint are often too high. They can be improved by changing the values for the heel off height, the step length and/or height and/or the amplitude of the sinus.

The step length and the step height are the parameters with the most influence on the generated hip and knee joint angles. Note that the hip and the knee joint angles are both increasing for an increasing step length and step height. On the contrary the hip joint angles are decreasing and the knee joint angles are increasing for an increasing heel off height. So the hip and the knee joint angles are not always correlated in the same way for the different gait parameters.

The step length, the heel off height and the amplitude of the pelvic obliquity and the trunk sinus are the parameters with the most influence on the ankle joint angle. Especially the step length has a major role for the calculation of the ankle joint angle.

Before sending the data to a rehabilitation robot the generated trajectories have to be visualized in order to check the correct behavior of the complete trajectory. If the angles are not in the physiologically compatible ranges the physiotherapist has to change the input parameters.

4. Practical implementation: realization of a GUI for ALTACRO

In this chapter the previously found specifications for a graphical user interface for a rehabilitation robot will be implemented. A GUI for ALTACRO will be implemented. In this work the interface will especially be concentrated on the needs of the engineers who are working with the device. In a future work this interface can be extended with a rehabilitation session part, with performance feedback.

4.1. ALTACRO: conventions

ALTACRO (Automated Locomotion Training using an Actuated Compliant Robotic Orthosis) is a project started in 2008 at the VUB in order to help patients suffering from gait impairment. A picture of the rehabilitation device is given in Figure 13. For the project a rehabilitation device was made. The research was mainly focused on four aspects of robot-assisted gait training [2]:

- Active ankle assistance
- Balance and load distribution
- Functional, three dimensional gait kinematics
- Physical human robot interaction

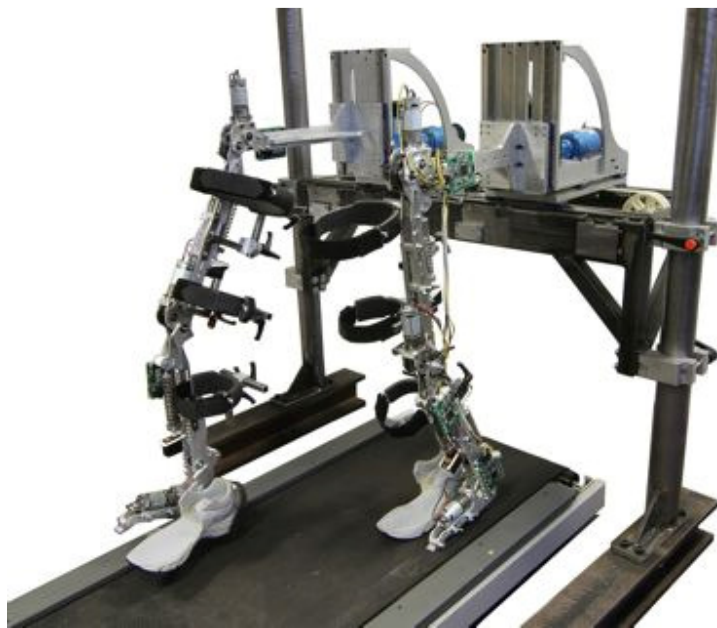


Figure 13: The ALTACRO rehabilitation device [30]

Before handling the robot, the different degree's of freedom (DOF) of ALTACRO are explained and some conventions of the reference frame are to be made. Three angle definitions are used with ALTACRO:

- The angle definitions in the coordinate systems (CS) linked to the joints and the DOF's of the actual robot (also defined further on as 'robot angles')
- The angle definitions as used for the Winter data
- The angle definitions inherent to the actuators (MACCEPA's)

These three different CS's are often used when handling the robot. So it is necessary to define these CS's and to define the relations between them.

4.1.1. Robot DOF's

The structure of ALTACRO is made of an exoskeleton and a support system which are both actuated. The robot has 8 DOF's for the exoskeleton structure and 5 DOF's for the support system. The DOF's of the exoskeleton are, for both legs: the hip, knee and ankle flexion/extension and the ab/adduction of the hip. The DOF's of the support system are: the lateral, sagittal and vertical translation of the pelvis, the internal/external rotation and the pelvic obliquity [30].

4.1.2. Robot frame definitions

The coordinate systems (CS) defined with the robot angles, will be the most used coordinate systems. It is also the most complete set of definitions.

The CS's are defined as follow (Figure 14 and Figure 15):

- CS₀: Base CS connected to the outside (fixed) world. It is in the middle of the complete robot structure and connected to the ground.
- CS₁: CS connected to the middle of the complete robot structure. It is obtained by a vertical translation of the robot over Δ_{z1} .
- CS₂: CS connected to the middle of the pelvis. It is obtained by a horizontal translation from CS₁ over Δ_{y2} .
- CS₃: CS connected to the left end of the pelvis³. This CS is obtained by a vertical translation over Δ_{z3} . By setting for example a positive translation to the left side and a negative translation to the right side of the pelvis, the pelvic obliquity is obtained.
- CS₄: CS connected to the cylindrical joint at the left end of the pelvis. This CS is obtained by a horizontal translation over Δ_{x4} from CS₃. By combination of a horizontal translation on the left and on the right side of the pelvis the obliquity in the transverse plane is created.
- CS₅: CS connected to the left (cylindrical) hip joint. The CS is obtained by a rotation of θ_5 around x_4 . It represents the ab/adduction of the left leg.
- CS₆: CS connected to the left upper leg link. It is obtained by a rotation of θ_6 around y_5 . It represents the rotation between the upper leg and the upper body.
- CS₇: CS connected to the left lower leg link. It is obtained by a rotation of θ_7 around y_6 and represents the rotation between the upper leg and the lower leg.
- CS₈: CS connected to the left foot link. It is obtained by a rotation of θ_8 around y_8 . It represents the rotation between the foot and the lower leg.

³ Keep attention: the robot DOF are not exactly the same as the physiological DOF, so the robot 'left/right end of the pelvis' is not the same as the physiological 'hip'. In ALTACRO there's an extra bar between the end of the pelvis and the hip joint, with a cylindrical joint connected to the bar. This extra cylindrical joint is creating the ab/adduction (see Figure 15).

Note that the coordinate systems CS_3 until CS_8 are here defined as the coordinate systems of the left leg of ALTACRO. The definitions of the CS's of the right leg are analog as indicated by an accent at the end of the name. So they are defined CS_3' until CS_8' .

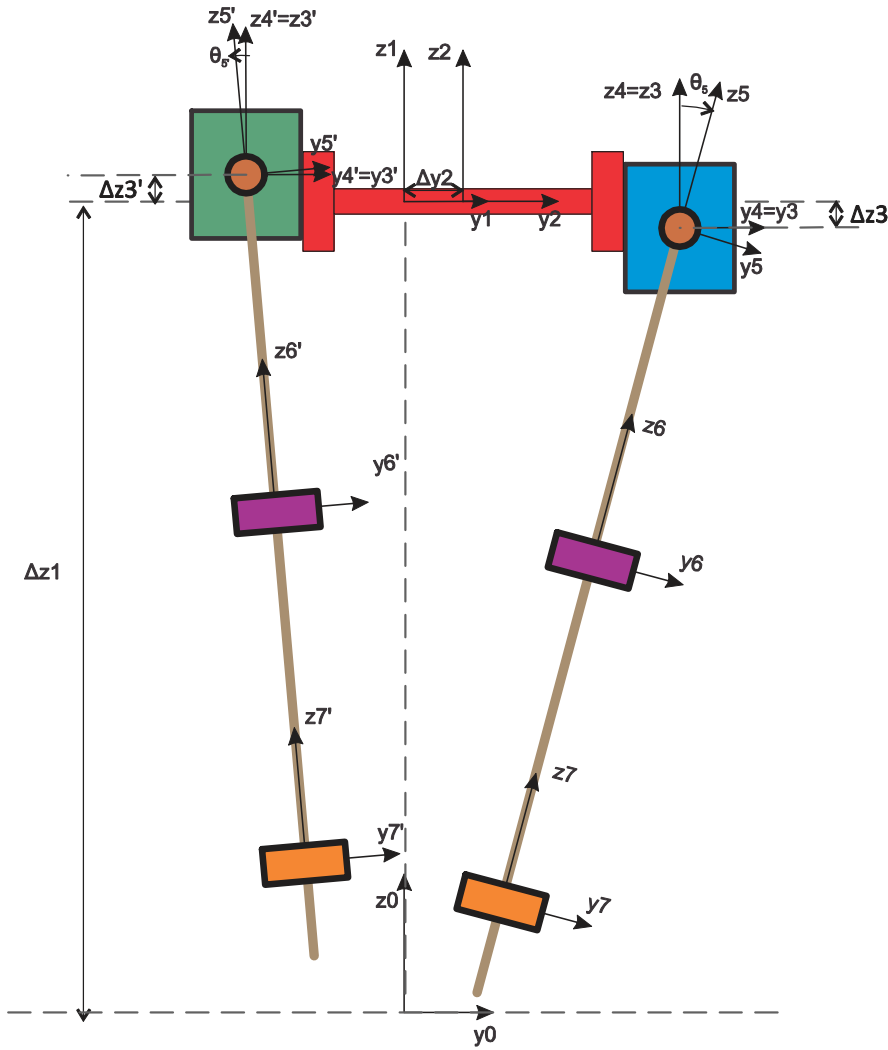


Figure 14: Robot angles: front view

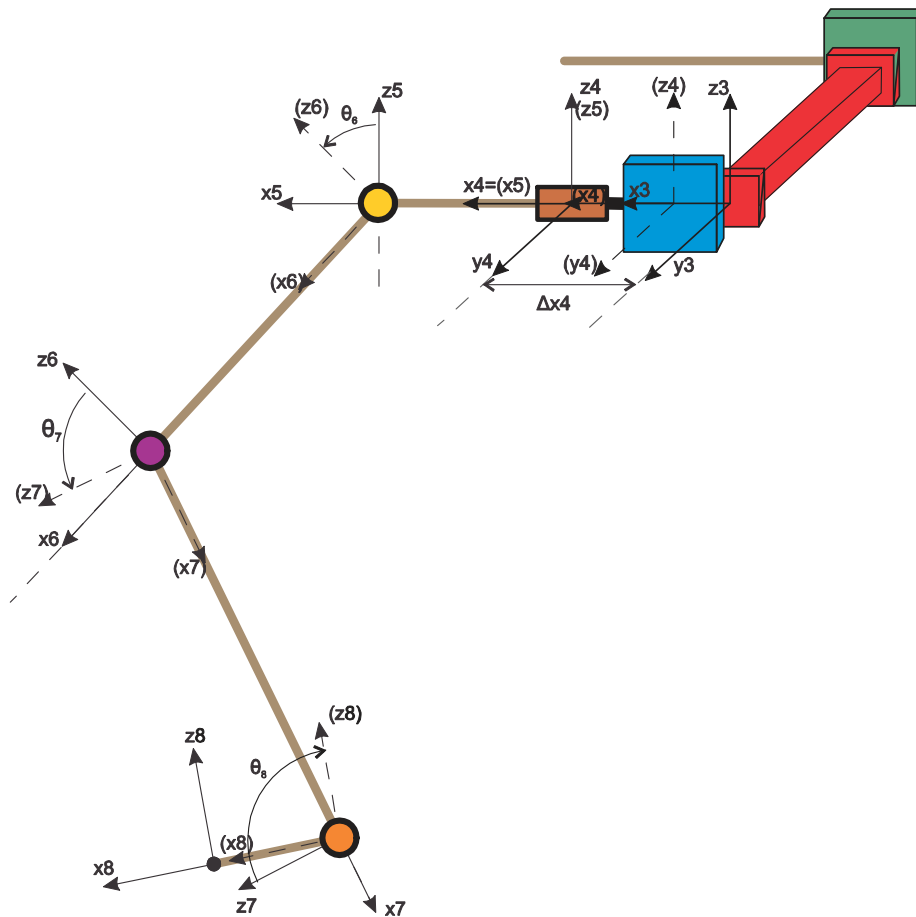


Figure 15: Robot angles: side view

4.1.3. Winter Conventions

The data used to generate reference trajectories of the hip, knee and ankle joint angles is based on the data from Winter [26]. To be able to use this data it is necessary to firstly recall the used conventions. Later on these coordinate systems will be transformed to the correspondent robot angles.

The Winter data defines only three angles (Figure 16):

- θ_1 : the angle between the vertical (the upper body) and the upper leg; measured between the negative z-axis and the upper leg.
- θ_2 : the angle between the extension of the upper leg and the lower leg.
- θ_3 : the angle between the perpendicular on the lower leg and the foot as illustrated in Figure 16.

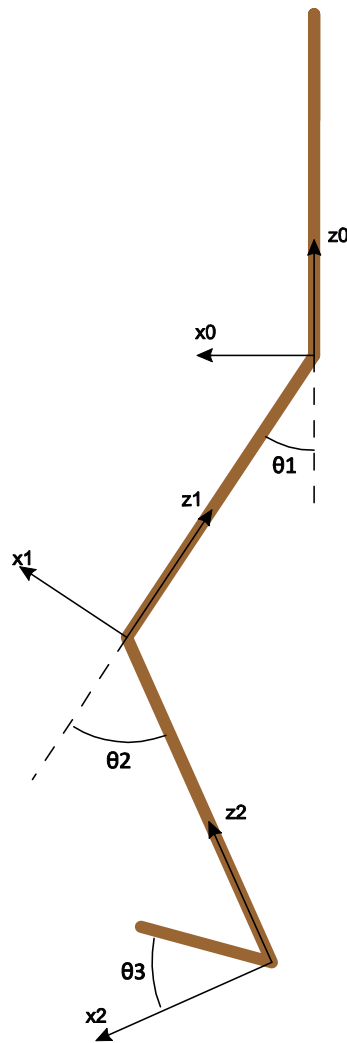


Figure 16: Winter angle definitions

4.1.4. MACCEPA conventions

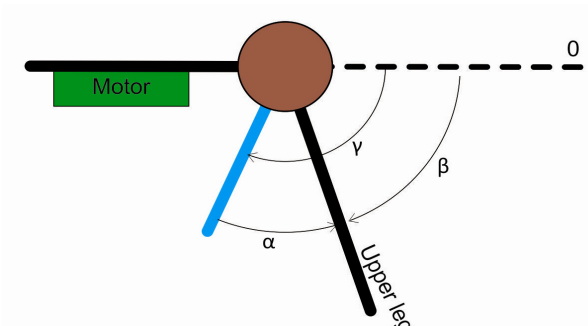
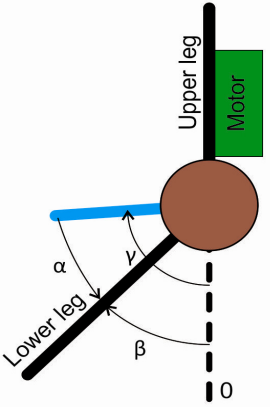
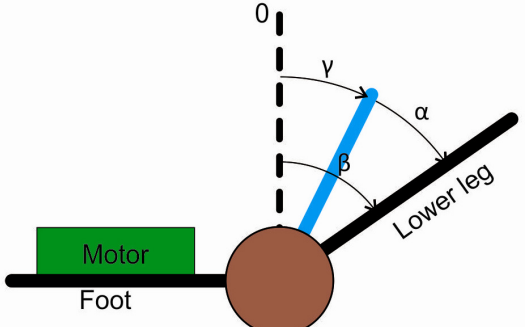
To ensure the variable stiffness of the actuators of the robot, the joints are equipped with MACCEPA actuators (Mechanically Adjustable Compliance and Controllable Equilibrium Position Actuators) [31]. A MACCEPA is a compliant actuator made of 3 bodies (namely two links and a lever arm) which are pivoting around a common rotation axis as can be seen in Table 13. The angle φ between one link and the lever arm is set by a traditional position controlled actuator. A spring is attached between the lever arm and the other link. The angle between these two bodies is called α . When α is zero the actuator is in its equilibrium position. When α differs from zero the force due to the elongation of the spring will generate a torque that tends to line up the second link with the lever arm [31].

As in each joint the (relative) placements of the MACCEPA actuator and of the motor are a little bit different, the MACCEPA angles are defined as follows (see also Table 13):

$$\alpha = \beta - \gamma$$

- for the hip joint:
 - α : angle between the lever arm and the upper leg
 - φ : angle between the prolongation of the upper body and the lever arm
- for the knee joint:
 - α : angle between the lever arm and the lower leg
 - φ : angle between the prolongation of the upper leg and the lever arm
- for the ankle joint:
 - α : angle between the lever arm and the lower leg
 - φ : angle between the perpendicular of the foot and the lever arm

Table 13: MACCEPA angle definitions

<p>Hip</p>	
<p>Knee</p>	
<p>Ankle</p>	

 MACCEPA lever arm

4.1.5. Link between the different conventions

In order to work properly with the different coordinate systems and angle definitions, they should be linked one to each other. This is done in table 14. The MACCEPA angles are called β (corresponding to the previous section), the Winter angles are indicated with a subscript 'w' and the robot angles have a subscript 'r'.

Table 14: Link between the angle definitions.

	MACCEPA-WINTER	WINTER-robot angles	Robot angles-MACCEPA
Hip	$\beta = -\theta_{1,w}$	$\theta_{1,w} = \theta_{6,r} - \frac{\pi}{2}$	$-\beta = \theta_{6,r} - \frac{\pi}{2}$
Knee	$\beta = \theta_{2,w}$	$\theta_{2,w} = \theta_{7,r}$	$\beta = \theta_{7,r}$
Ankle	$\beta = -\theta_{3,w}$	$\theta_{3,w} + \frac{\pi}{2} = -\theta_{8,r}$	$\beta + \frac{\pi}{2} = \theta_{8,r}$

4.2. Programming of the interface

The programming of the interface is done in LabView as justified in section 1.2.3.

LabView is a graphical dataflow programming language. The code is made out of graphical blocks which are connected one to another with 'wires'. As soon as all the input data (given by the wires) is available a block can be executed. This may be the case for multiple blocks at the same time which makes LabView capable of parallel coding. By the use of multi-processing and multi-threading hardware of the built-in scheduler the code is multiplexed. Some special structures are foreseen in LabView in the case that a strict sequential structure is needed for a program [32].

LabView subroutines are called virtual instruments (VI's) and can also be represented by graphical blocks. A regular use of these VI blocks can make a large program much more maintainable. The use of these VI blocks enables the programmer to reuse the same VI multiple times.

4.2.1. The main structure

The main structure of the program must meet the next requirements:

- The program has to be initialized: the graphs have to be cleaned, the variables have to be initialized to their default values and reference control values and trajectories have to be loaded.
- The program has to keep running until the 'STOP' button is pressed.
- If a 'special event' occurs, like the opening of the TCP/IP connection, the corresponding code has to be executed and the program has to be restart with the new parameters.

To satisfy these requirements the effective program is principally made out of a while loop, preceded by a sequential structure (see figure 17). The sequential structure ensures the initialization of the program. The 'while loop' contains a 'case structure' and a second 'while loop' and will run during the entire duration of the session. The second 'while loop' is the actual program and the 'case structure' handles the 'special events'.

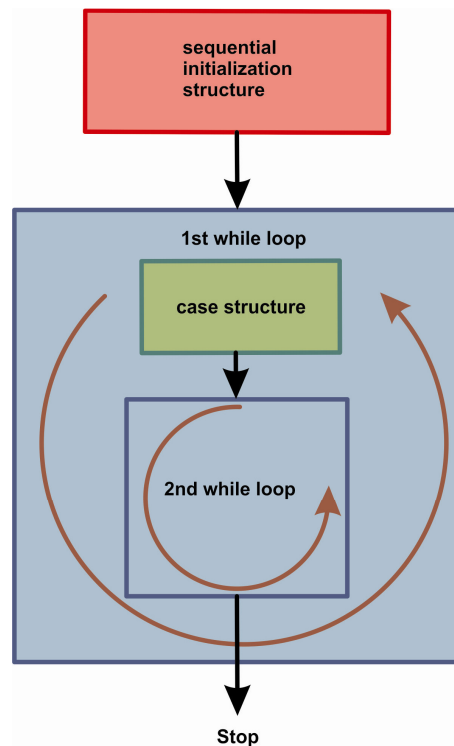


Figure 17: Main structure of the programming of the interface

More in detail: the sequential structure sets all the patient variables (dimensions, personal data ...) to zero, cleans the graphs, loads the last saved control parameters for the robot controllers and initializes the lists with the available trajectories for the ankle, hip and knee nodes. With the aim of facilitating and accelerating the use of the program for the engineers during the test phase of the robot, values of a fictive standard patient are loaded during the startup of the program. In this way the engineers can immediately test new trajectories or new control parameters without the need of selecting firstly some patient dimensions.

After passing this initialization phase the program is entering a first while loop. This 'while loop' runs non-stop during the execution of the program. It can only be stopped by pushing the 'stop' button. This first 'while loop' contains a 'case structure' and a second 'while loop' (See Figure 17).

The 'case structure' contains actions which influence and set many different values that are used in different places in the program. It is used to load new global parameters such as a file with new control parameters, a new trajectory file and a new patient file or to open the TCP/IP connection. In this way it is also sure that the GUI program restarts the main program loop with the new values.

The second 'while loop' (which is nested in the first one) contains the main code of the GUI program. It runs continuously and can only be stopped by the 'STOP' button or by one of the actions of the previously explained 'case structure'. If the 'STOP' button was pressed the complete program would be stopped; if one of the described actions of the case structure was asked, this action is executed and the 2nd while loop restarts.

So, to summarize the overall structure (Figure 17): The second 'while loop' and the 'case' structure are nested in an overall (first) 'while loop'. The second 'while loop' is the main part of the program. When an event, handled by the 'case' structure, occurs the second 'while loop' is left and the program goes back to the first 'while loop'. Here it can enter the 'case' structure. The code of the case structure is executed and the program goes again to the second 'while loop'. If the 'STOP' button is pressed the program leaves both 'while loops' and the complete program is stopped.

4.2.2. The detailed code

LabView is a graphical programming environment which is capable of parallel coding. For this reason the main program, situated in the second while loop, is programmed in separated blocks. Each block has a dedicated task and is coherent with the structure of the interface as defined in section 2.2. The communication between the different blocks is done with 'property nodes'. This is a special structure in LabView which replaces the wires to pass the value of a variable from one location to another in the program. The use of these 'property nodes' instead of wires makes a large program more readable. A disadvantage is that it is more difficult to see the flow of the variables, so during the programming this ambiguity has to be taken into account.

The programming structure/logic is illustrated in Figure 18. In this figure the structure is illustrated as a sequential flowchart structure. Although the code is in reality programmed in separated code blocks, this is done to easily illustrate the dependencies between the different parts of the program. A second reason for this is to make the structure more comprehensible for the reader. For the same reasons the flowchart goes through when the Booleans are false. Only if the Booleans are true, extra code blocks (the later explained 'event structures'), are executed and byways have to be followed in the flowchart. In Figure 18 the black arrows correspond to the flow in the second while loop. The orange arrows correspond to actions which will sent the execution flow of the program to the case structure in the first while loop.

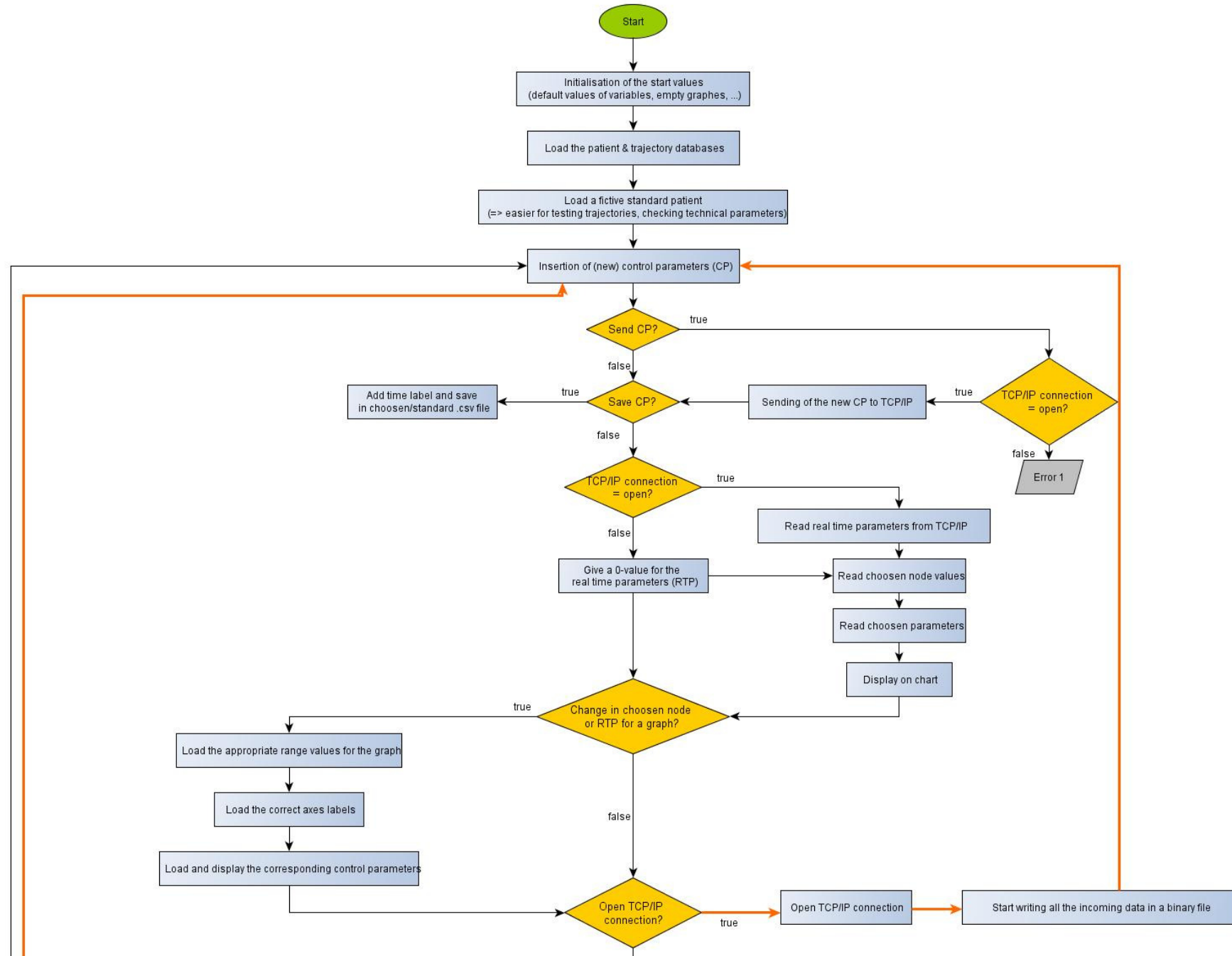
To speed up the execution of the code, event structures are used. An event structure is a block of code that is preceded by a boolean. The appended code block is only executed if this boolean has the desired value (true or false). In this way only the part of the code that is needed for the action that the user is asking for, is executed. Only the code which is susceptible to be used several times in the program, such as the code which manage the variables for the desired nodes, is read in every repetition of the second while loop.

The most important code blocks, programmed in the second while loop, are:

- A block that sends data (control parameters or trajectories) over the TCP/IP connection: It first checks if the connection is open and it gives an error if this is not the case. If the connection is open it sends the data over the TCP/IP and it checks if an acknowledgment of the arrival of the data is received. It checks also whether the same quantity of trajectories has arrived as were sent.
- A block that reads all the data coming over the TCP/IP connection. This block is active as long as the TCP/IP connection is open.

- A block that handles the representation of the real time signals: It checks the selected parameters and nodes and displays the desired data. It also changes the range of the graphs and the displayed units as a function of the chosen parameters and the type of the chosen node (if it is a translational joint or a rotational joint).
- A block that is handling the information related to the patient data: This block saves the entered patient data in a (new) file or fills the variables for the patient data with the data out of an existing file.
- A block that handles the simulation of the trajectory with a stick model: It loads the angles for the different nodes as well as the patient dimensions and takes them as input. From this data it generates a stick model such that the movement of the robot during the execution of the trajectories can be visualized.

Practical implementation: realization of a GUI for ALTACRO



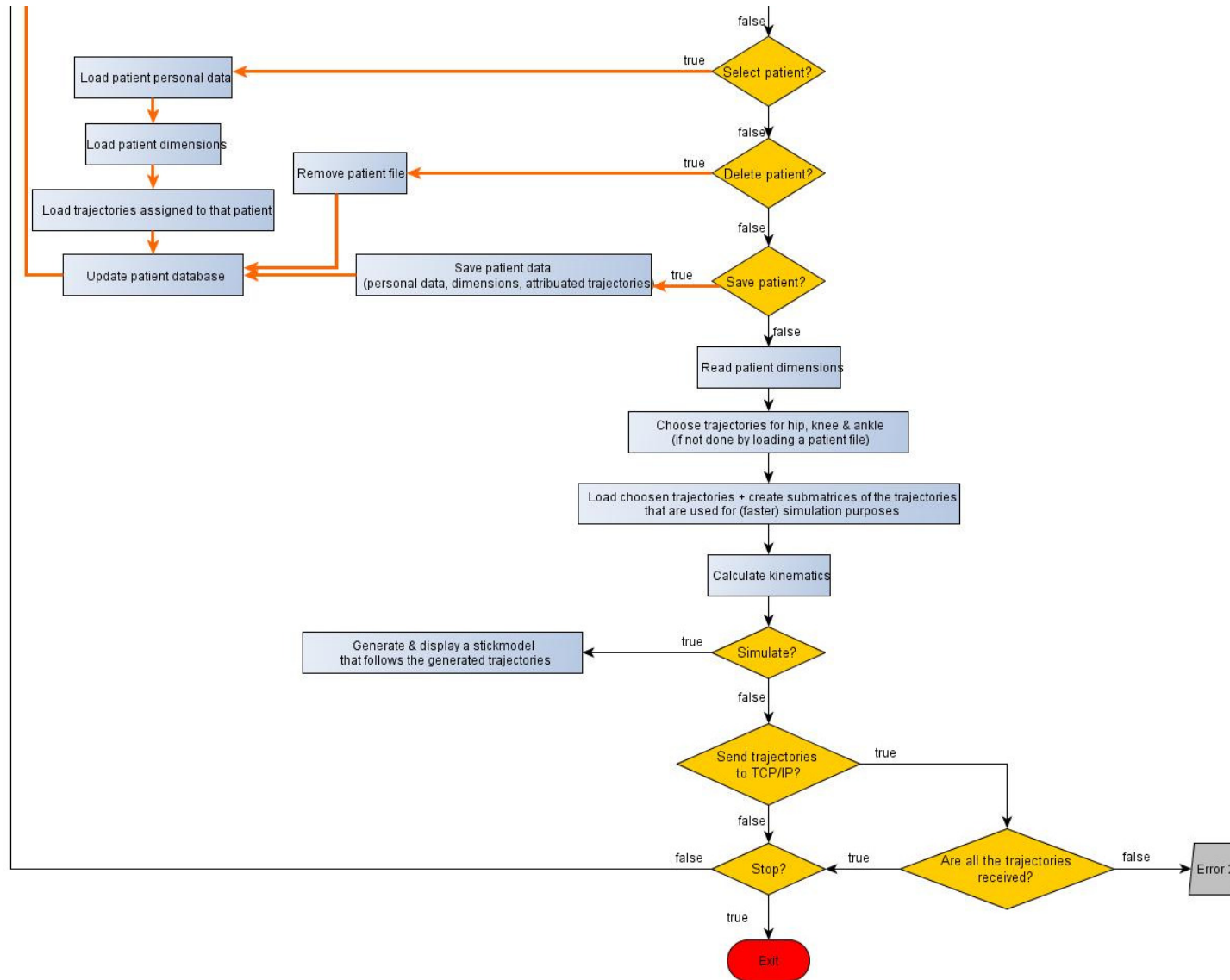


Figure 18: Program structure: The orange arrows correspond to actions which send the execution to the case structure. The black arrows correspond to the flow in the second while loop.

4.3. Operation of the interface

In this part the functioning of the interface will be explained. In a first section the structure of the graphical user interface (GUI) will be detailed. In a second section a startup sequence for the GUI will be given.

4.3.1. Structure of the GUI

The GUI is constituted of one main tab level containing three tabs: the ‘technical specifications tab’, the ‘patient tab’ and the ‘session tab’. The first tab, the ‘technical specifications tab’, is further divided in a second tab level. This structure (together with the main contents of each tab) is illustrated in a flowchart in Figure 19 and will be detailed in the next sections.

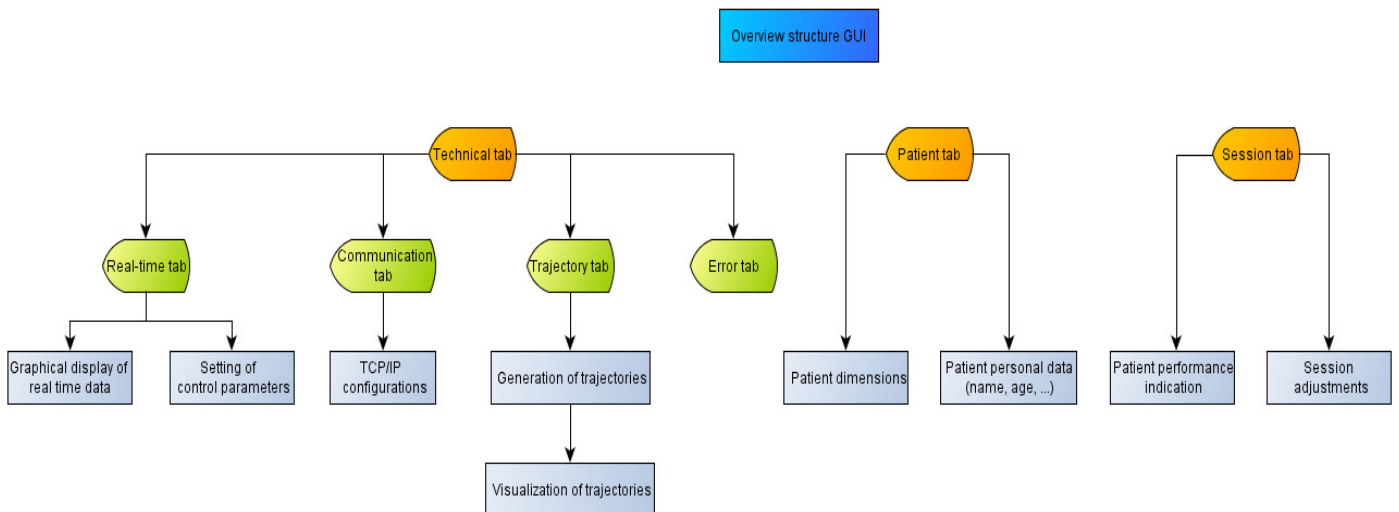


Figure 19: Overview of the structure of the GUI

4.3.1.1. Upper tab level

The GUI is made out of 2 tab levels (see Figure 20); one main tab level and a sub tab level which is only accessible from the ‘technical specifications’ main tab. The upper/main tab level contains:

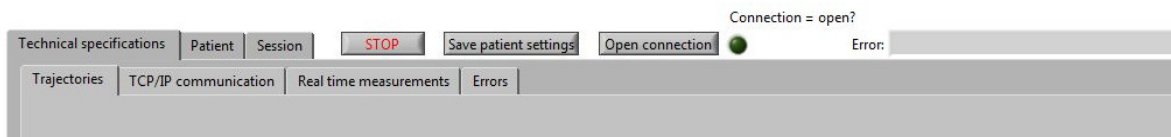


Figure 20: Illustration of the different tab levels in the GUI

- 3 main tabs:
 - ‘Technical specifications’: This is the main tab for the engineers who are working with ALTACRO. It gives access to all the inputs and outputs of the robot, such as the settings for the trajectories, the control parameters, the real time sensor readings and the TCP/IP communication settings.

- 'Patient': This tab contains information related to the patient, such as his personal data and his dimensions.
- 'Session': This tab is planned to contain information about the session itself. With a graph that gives an indication about the performance of the patient (e.g. graphical display of the real effectuated trajectory and the asked trajectory) and some variables for the session adjustments (such as the treadmill speed). This session information will be stored in the same file as the patient data.
- The 'STOP' button: Stops the robot and the program. This button is placed in the main window so that it is accessible from anywhere in the program. In this way it is easy to stop the robot in the case that something went wrong.
- A 'save patient settings' button: This button is accessible from each tab, so that you can always easily save the latest changes in the settings for the patient/session (all the specific session options, such as the used trajectories, are saved in the patient file so that they can easily be re-used for the next session).
- An 'open connection' button: This button opens the TCP/IP connection. If the connection is opened the led 'connection = open?' lights up.
- An error line: displays the errors.

4.3.1.2. Technical tab

The 'Technical specifications' main tab contains a second tab line with 4 sub tabs:

- 'Trajectories' (Figure 21): This tab is for the joint trajectories. The user (physiotherapist or engineer) can generate or load the reference trajectories for the hip, knee and ankle joints. To generate trajectories different gait parameters (such as the step height and length and the amplitudes of the hips) can be chosen. If the 'save patient settings' button is pushed after the selection of the trajectories the program associates these trajectories with the previously opened patient file (see the section 'startup sequence'). In this way there is no need to redo this step each time a new session is started with the same patient.
It is also possible to simulate and visualize the imposed trajectories in this tab before sending them to the real robot. For this the user first has to choose a simulation speed. If the user does not select a simulation speed, a default medium speed is selected. The simulation speed is defined as the time delay (in ms) the program waits before plotting a new step of the simulation, so the higher the chosen value, the slower the simulation. The simulation speed is completely independent of the treadmill speed. The simulation is started by pushing the 'Simulate' button. If the user pushes one of the 'view' buttons ('Top view', 'Front view' or 'Side view'), the corresponding view is also projected during the simulation of the trajectory. By pushing the 'Show only projection' button, the 3D-stick model is not visible anymore and only the chosen projections are displayed during the simulation.

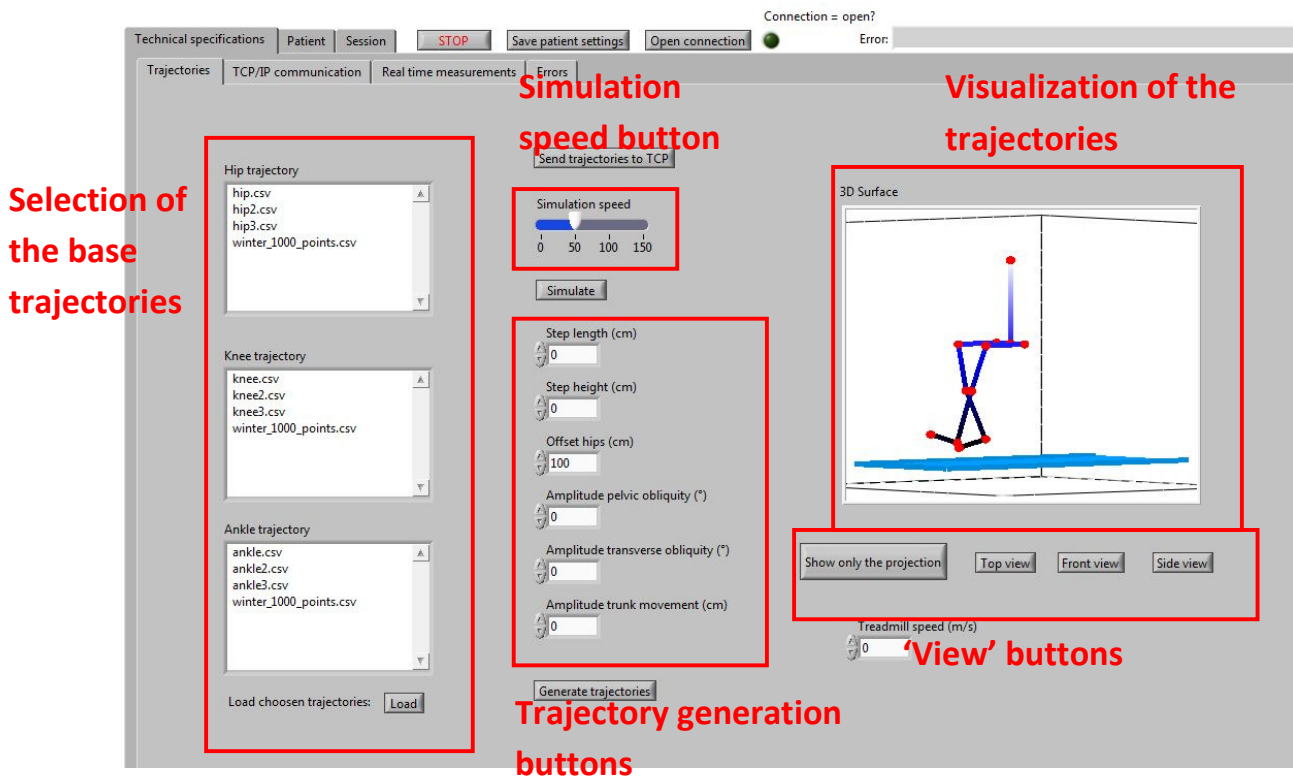


Figure 21: 'Trajectories' sub tab of the GUI

- 'TCP/IP communication': The settings for the TCP/IP communication are gathered in this tab. The user (physiotherapist) doesn't need to go to this tab as the settings remains constant for each session. Only if something changes in the communication protocol (e.g. if another computer port is used) an engineer needs this tab.
- 'Real time measurements' (Figure 22): This tab contains the returned 'real time' signals (RTS) as well as the control parameters to tune the robot. The control parameters are the parameters that manage the low level controllers of the robot. When passing with the mouse over the control parameters the names of the control parameters as well as their units are displayed. The control parameters may be saved in two different ways. If the user employs the 'save to default' button, the next time you run the program this (new) control parameters will be used automatically, independently of the chosen patient. If the 'save under a new name' button is used, the user needs to choose a new path name and a back-up file is made. The next time the program is used, the old values will be loaded as default values. The 'real time' signals are the signals that are collected, coming from the robot, such as the speed and the position (angular or translational) of each joint. The data (as well the control parameters as the real time signals) are displayed according to the chosen node, so that the interface is not overloaded with data. It is possible to display the data of maximum 2 different nodes at the same time. There are 3 graphs in this tab, so that the user can display three different RTS of the same node at the same time. If the user chooses to display the data of two different nodes, the RTS of both

nodes are displayed in the same graph so that it is easy for the therapist to compare the values of both nodes. The program automatically sets ranges for each graph. This range is defined by the type of the node that is selected (a rotational or a translational) and by the RTS that is selected. This range can also be changed manually. It is possible to freeze the graph on every moment. Once the graph is frozen it is also possible to zoom on the graph.

- 'Errors: In this tab error codes are logged and explained more in detail.

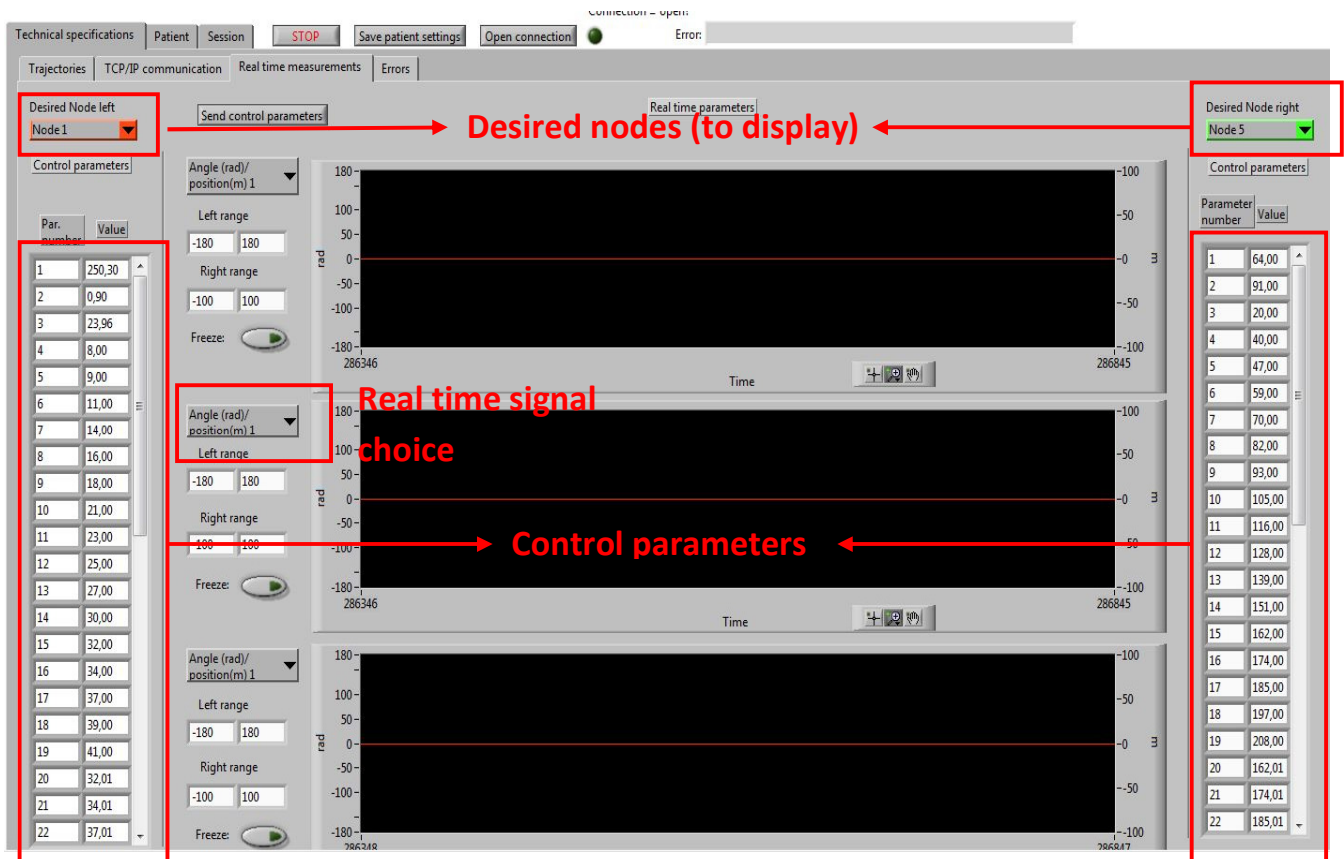


Figure 22: 'Real time measurements' sub tab of the GUI

4.3.1.3. Patient tab

The 'patient' tab (see further on Figure 25 & 26) contains the information directly related to the patients. These are mainly his personal data and his dimensions. This tab contains the possibility to open an existing patient file or to create a new one. If an existing patient file is opened all the boxes with the personal data and the dimensions of the patient are automatically filled in; if a new file is created the user needs to fill in all the boxes and push the 'Save patient settings' button. The file is then added to the patient database with as pathname the surname + the name of the patient, which makes it easy to find the file back during the next trainings session.

4.3.2. How to use the GUI?

Although the interface is made as clear as possible, for the ease of use a correct startup sequence will be given.

4.3.2.1. Startup sequence

The steps that are now explained for a correct startup of the GUI are summarized in the flowchart in Figure 23.

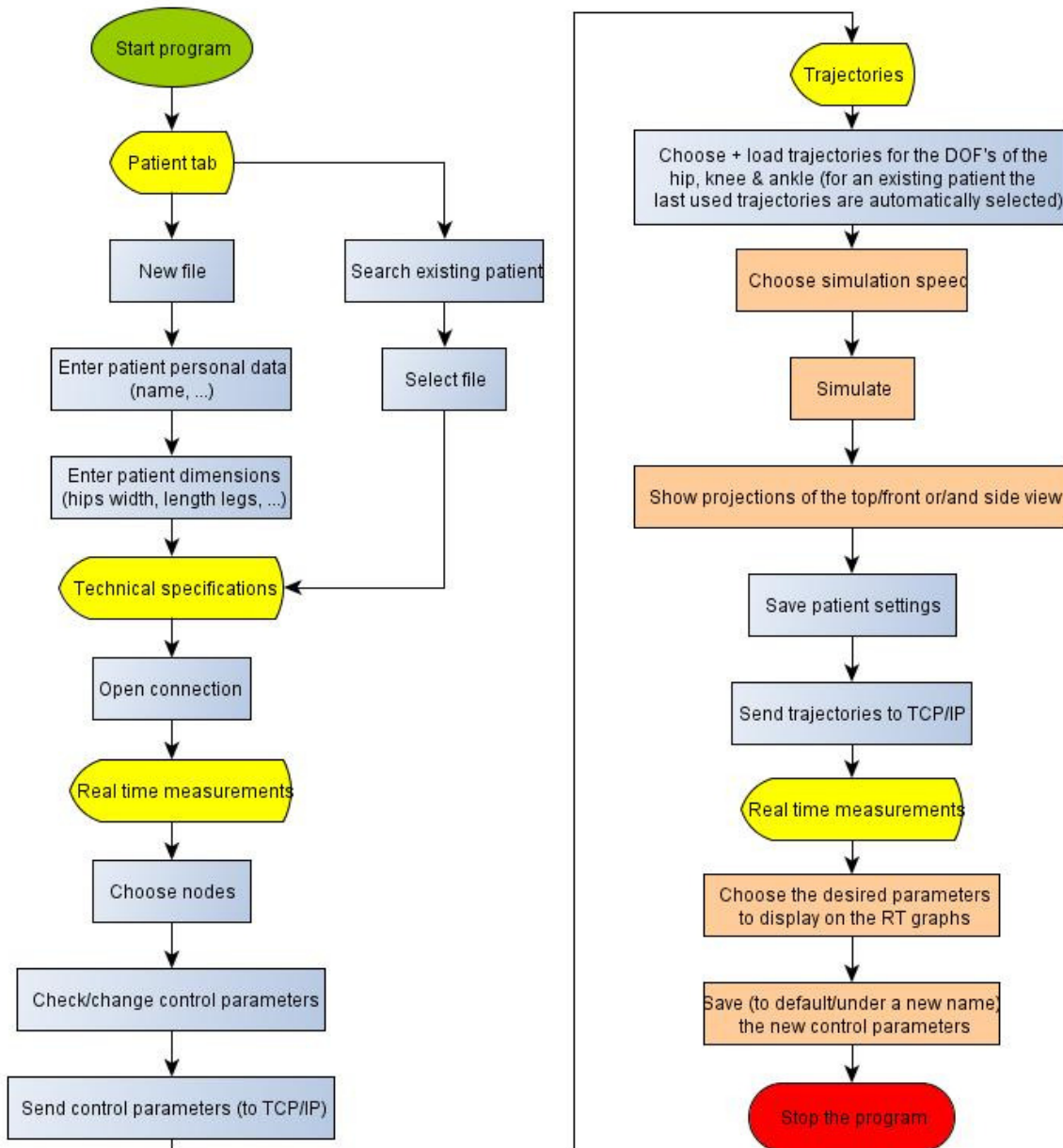


Figure 23: Flowchart for a correct use of the GUI: in yellow the tab names, in blue the mandatory actions and in salmon the optional actions.

1. Turn on the robot
2. Start the program. This is done by pushing the 'RUN' button in LabView (Figure 24)

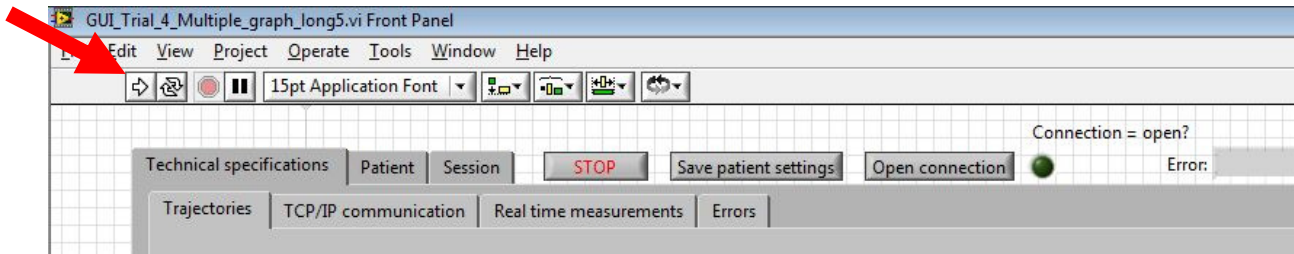


Figure 24: Starting of the program: the arrow is indicating the 'RUN' button in LabView

3. Open the Patient tab (Figure 25 & 26). Choose an existing patient file or add a new patient file.
 - a. For an existing file (Figure 25):
 - Push the 'Search existing patient' button
 - Choose the right patient in the patient database
 - Push the 'Select' button
 - All the previously added settings are automatically loaded

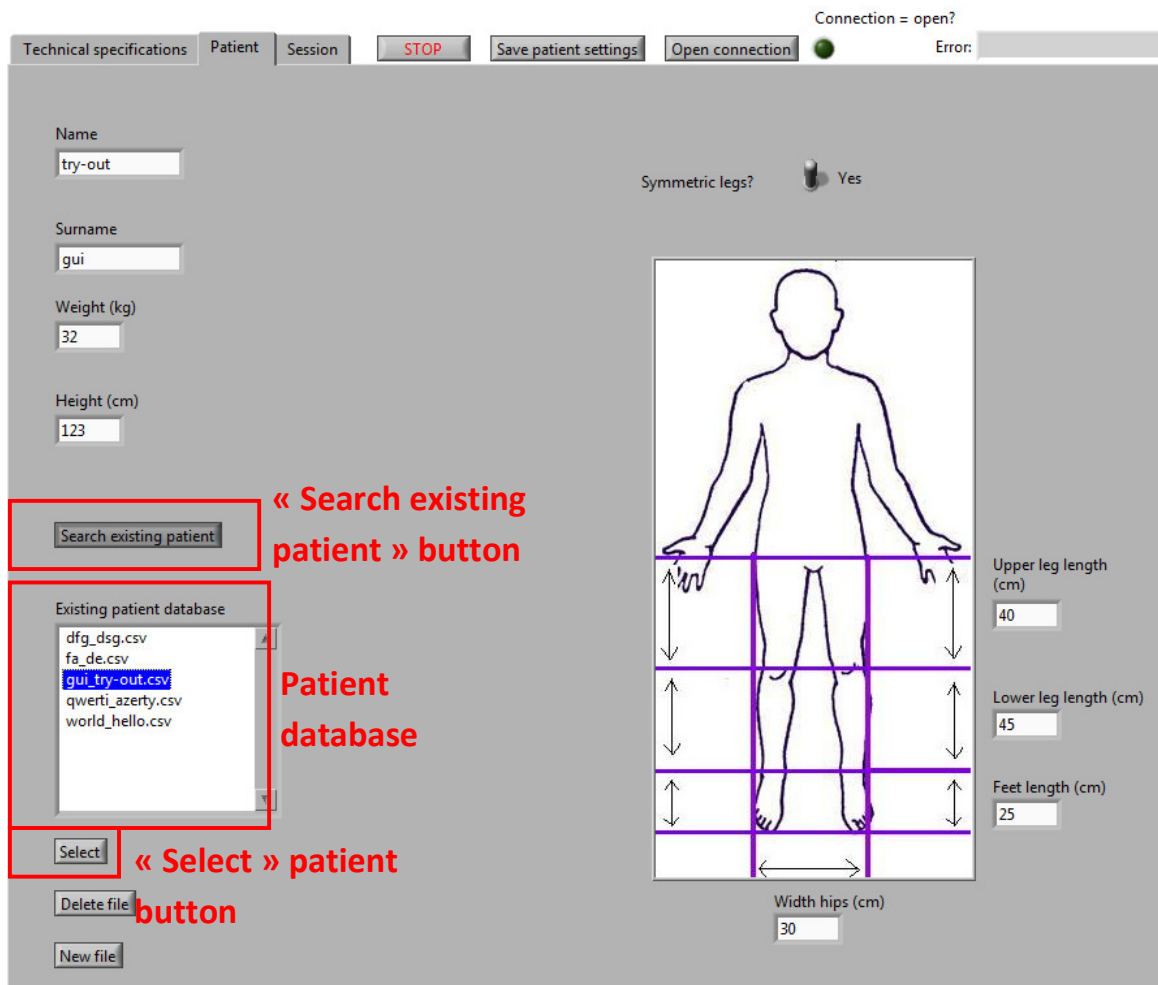


Figure 25: Patient tab of the GUI: buttons to select an existing patient file

- b. For a new patient file (Figure 26):
- Enter the patient data and patient dimensions into the boxes. Note that there is the possibility to give different lengths for the left and the right side of the patient. This is for in the (rare) case that the patient does not have symmetric legs and is done by toggling the switch of 'Symmetric legs?'. This switch is automatically put on 'symmetric legs' during the creation of a new patient file.
 - Push the 'Save patient settings' button

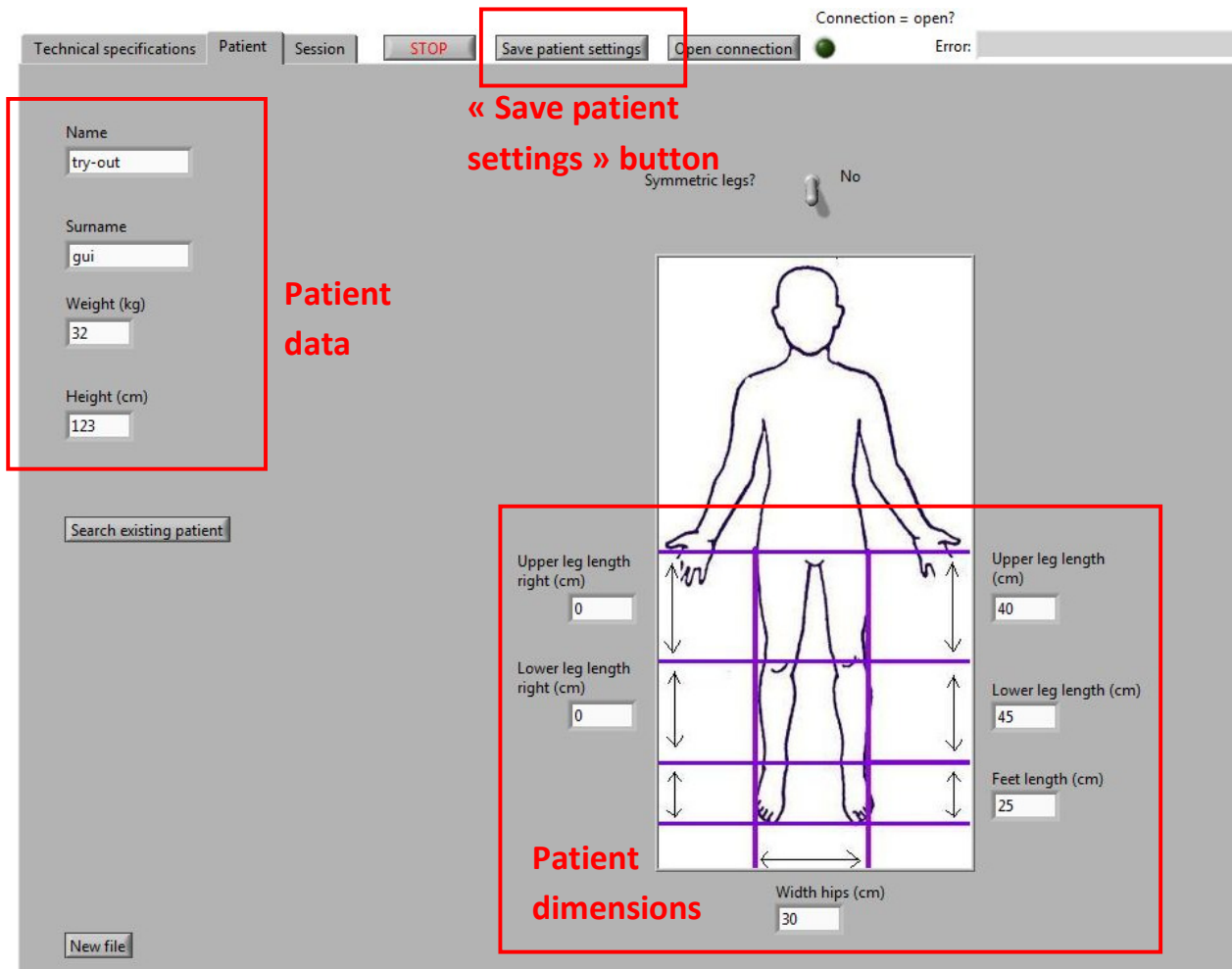


Figure 26: Patient tab of the GUI: buttons to select a new patient file

4. Open the TCP/IP connection (Figure 27):
 - a. Push the 'Open connection' button
 - b. Verify that the 'Connection = open?' led is lit

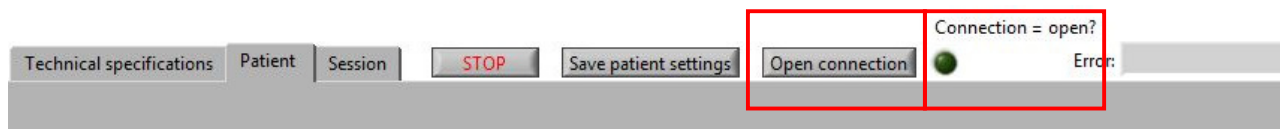


Figure 27: Header of the GUI with the TCP/IP connection buttons highlighted

5. Open the 'Technical specifications' tab.
 - a. Open the 'Real time measurements' sub tab
 - Check and/or change the control parameters for the different nodes
 - Push the 'send control parameters'
 - b. Open the 'Trajectories' sub tab

- *Note: the next 5 steps should not be executed if an existing patient file was chosen. The trajectories, associated to the last run of this patient file, are automatically loaded in the program. The trajectories are not sent automatically to the robot as the patient file is chosen before the TCP/IP connection is opened.*
 - Choose the desired trajectories for the hip, knee and ankle joints
 - Push the 'Load' button to load the chosen trajectories in the program
 - Choose the simulation speed
 - Push the 'Simulation' button to verify the behavior of the chosen trajectories
 - Push the 'Save patient settings' button to associate this trajectories to the patient file
 - Push the 'Send trajectories to TCP' button to send the trajectories to the robot.
- c. Open the 'Real time measurements' sub tab
- Choose the nodes to inspect
 - Choose the real time data to display on the graphs
 - Push the 'Save' (save to default or save under a new name) button to save the new control parameters.

5. Conclusions

In this thesis the development of a graphical user interface for a rehabilitation exoskeleton was investigated. For this in the first parts of this thesis the requirements for a graphical user interface were investigated. It was found that the GUI must fulfill the needs of the engineers, the physiotherapists and the patients. For the engineers the GUI must contain options to check and change the control parameters of the robot, check the data coming from the robot, generate trajectories for the robot and manage the communication. These generated trajectories represent the sequence of desired positions of the joints of the robot. The requirements of the physiotherapists and the patient have been developed in cooperation with physiotherapists. The GUI must enable the physiotherapist to easily and safely control the robot. For this the physiotherapist must be able to generate physiologically and kinematically correct trajectories based on several therapy specific parameters, such as the step height and length. The GUI must also contain options to save patient data and session data. Lastly it is advisable that the GUI contains a bio-indicator which gives an indication of the patient's performance. For the patient the GUI must give feedback about his session results.

In the second part of the thesis a trajectory generation tool was developed. The generated trajectories must satisfy kinematically imposed constraints as well as physiologically imposed constraints. The trajectories were firstly developed in two dimensions and were then extended to three dimensions. In two dimensions it was tried to base these generated trajectories on the data recorded by Winter [26]. This method was not feasible if the step length and the step height must be possible to change. The generated trajectories, as well in two dimensions as in three dimensions, must finally be completely based on the physiological constraints and the gait parameters given by the user inputs. The generated trajectories have then been evaluated by comparing them to measured data. By using right combinations for the input parameter values the trajectory generation is working well.

In the next part of the thesis a GUI is implemented for a gait rehabilitation robot, ALTACRO. It was chosen to focus this interface on the needs of the engineers. The interface was programmed in LabView and the trajectory generation was programmed in Matlab. By programming the interface in LabView, the GUI could easily and quickly interact with the robot. In this way the data coming from the robot joints could be viewed almost in real time. The interface satisfied to all the engineers needs: it could change and check the control parameters, gather and visualize the joint data in an organized way, manage the communication and control the trajectory generation.

The goal of this thesis was to investigate the development of a graphical user interface for a rehabilitation exoskeleton. This has been done by exploring the requirements of a GUI, specific for a rehabilitation device. The GUI has been extended with a trajectory generation tool to generate physiologically safe and kinematically correct trajectories. Further on the GUI has been build, with a focus on the engineers needs. The GUI satisfied the stated requirements and the trajectory tool worked well.

6. Future work

In a future work a feedback method may be developed for the patient. This will improve his attentiveness during the therapy and will give him immediate feedback on his session results. An interactive game can be developed for this purpose. A bio-indicator may also be developed to calculate the global performance of the patient during a gait rehabilitation process. The importance of the movement of each joint in a gait cycle for the calculation of this bio-indicator can be investigated. In this way the global performance indicator can be calculated based on the torque needed in the actuator of each joint.

For the trajectory generation it may be useful to investigate on the correlation between the different user inputs (step length, step height, walking speed, amplitudes of the hip movements). It has already been stated that the different gait parameters (step length, step height and the amplitude of the hip movements) are dependent off the walking speed, but the exact correlation factor has not yet been developed. This factor can be implemented in the trajectory development tool to increase the user friendliness of the device and to increase the robustness of the physiological compatibility of the trajectory generation tool. It is possible that the option to change the different input parameters manually may still be available as the gait cycle also changes dependent on the individual.

In a future work investigation can be done for the further development of a graphical user interface for the physiotherapist and for the patient. For the physiotherapist the effectuated trajectories may be visualized and compared to the desired trajectories. For the patient a feedback method may be developed, as stated before.

In a future work a procedure for going from a sitting position to a standing position and back may be developed. This will facilitate the start-up and the end of a rehabilitation session. At the moment, in the beginning of the session, the patient has to be put in a standing position in the rehabilitation robot. This is difficult for the physiotherapist if the patient is in the beginning of his rehabilitation process, especially when the patient is heavy. By implementing a stand-up and a sit-down procedure a wheelchair can be used to move the patient to and from the therapy. The patient can be attached in the exoskeleton when he is sitting in the wheelchair and the rehabilitation device will help him to stand-up. This can be difficult to implement for ALTACRO but is useful to take into consideration when developing a new gait rehabilitation device.

7. Sources

- [1] W. Organization, *Neurological Disorders: public health challenges*, 2006.
- [2] P. Beyl, «Design and control of a knee exoskeleton powered by pleated pneumatic artificial muscles for robot-assisted gait rehabilitation,» 2012.
- [3] B. Fei, W. S. Ng, S. Chauhan et C. K. Kwok, «The safety issues of medical robotics,» *Reliability engineering & system safety*, vol. 73, pp. 183-192, 2001.
- [4] S. Kemna, P. Culmer, A. Jackson, S. Makower, J. Gallagher, R. Holt, F. Cnossen, J. Cozens, M. Levesley et B. Bhakta, «Developing a user interface for the iPAM stroke rehabilitation system,» chez *Rehabilitation Robotics, 2009. ICORR 2009. IEEE International Conference on*, 2009.
- [5] «Pablo system,» [Online]. Available: <http://www.tyromotion.com/fr/produits/pablo/apercu.html>. [Accessed 12 2012].
- [6] K. Brüttsch, T. Schuler, A. Koenig, L. Zimmerli, S. Méryllat, L. Lünenburger, R. Riener, L. Jäncke et A. Meyer-Heim, «Research Influence of virtual reality soccer game on walking performance in robotic assisted gait training for children,» *Journal of NeuroEngineering and Rehabilitation*, vol. 7, p. 15, 2010.
- [7] «Lokomat - enhanced functional locomotion therapy,» [Online]. Available: <http://www.hocoma.com/products/lokomat>. [Accessed 12 2012].
- [8] *Lokomat System User Manual*, 2009.
- [9] «Optogait,» [Online]. Available: <http://www.optogait.com>. [Accessed 12 2012].
- [10] Noraxon, «Clinical gait analysis and gait therapy solutions,» [Online]. Available: http://www.noraxon.com/products/instruments/complete_gait_analysis.php. [Accessed 2012].
- [11] F. Collen, D. Wade, G. Robb et C. Bradshaw, «The Rivermead mobility index: a further development of the Rivermead motor assessment,» *Disability & Rehabilitation*, vol. 13, n° 12, pp. 50-54, 1991.
- [12] L. Johnson et J. Selfe, «Measurement of mobility stroke: a comparison of the Modified Rivermead Mobility Index and the Motor Assessment Scale,» *Physiotherapy*, vol. 90, n° 13, pp. 132-138, September 2004.
- [13] V. Hood, M. Granat, D. Maxwell et J. Hasler, «A new method of using heart rate to represent energy expenditure: the Total Heart Beat Index,» *Archives of physical medicine and rehabilitation*, vol. 83, n° 19, pp. 1266-1273, 2002.

- [14] M. Ijzerman, G. Baardman, M. van't Hof, H. Boom, H. Hermens et P. Veltink, «Validity and reproducibility of crutch force and heart rate measurements to assess energy expenditure of paraplegic gait,» *Archives of physical medicine and rehabilitation*, vol. 80, n° 19, pp. 1017-1023, 1999.
- [15] M. J. Bailey et C. M. Ratcliffe, «Reliability of Physiological Cost Index measurements in Walking Normal subjects using steady-state, non steady-state and post-exercise heart rate recording,» *Physiotherapy*, vol. 81, n° 110, pp. 618-623, October 1995.
- [16] E. Eng, *Qt GUI Toolkit*, vol. 31, 1996.
- [17] B. Rempt et D. Mertz, «Qt and PyQt,» Februari 2003. [Online]. Available: <http://www.ibm.com/developerworks/linux/library/1-qt/>. [Accessed 12 2012].
- [18] A. Writing, «The advantages & disadvantages of Visual Basic,» [Online]. Available: http://www.ehow.com/list_6148486_adavantages-disadvantages-visual-basic.html. [Accessed 12 2012].
- [19] D. Eyre, «Matlab basics and a little beyond,» august 1998. [Online]. Available: <http://www.math.utah.edu/~eyre/computing/matlab-intro>. [Accessed 2012].
- [20] R. Tabone, «Matlab Introduction,» [Online]. Available: <http://www.yorku.ca/jdc/Matlab>. [Accessed 12 2012].
- [21] N. instruments, «Advantages of Using LabView in Academic Research,» [Online]. Available: <http://www.ni.com/white-paper/8534>. [Accessed 12 2012].
- [22] «Advantages/Disadvantages,» [Online]. Available: <http://www.labviewportal.eu/en/introduction/advantages-disadvantages>. [Accessed 12 2012].
- [23] J. Perry, «Gait analysis Normal and pathological function.,» SLACK Incorporated, 1992.
- [24] S. A. Gard et D. S. Childress, «What determines the vertical displacement of the body during normal walking?,» *JPO: Journal of Prosthetics and Orthotics*, vol. 13, n° 13, pp. 64-67, 2001.
- [25] M. W. Whittle et D. Levine, «Three-dimensional relationships between the movements of the pelvis and lumbar spine during normal gait,» *Human Movement Science*, vol. 18, n° 15, pp. 681-692, 1999.
- [26] D. A. Winter, «Foot trajectory in human gait: a precise and multifactorial motor control task,» *Physical Therapy*, vol. 72, n° 11, pp. 45-53, 1992.

- [27] T. Öberg, A. Karsznia et K. Öberg, «Basic gait parameters: reference data for normal subjects, 10-79 years of age,» *Journal of rehabilitation research and development*, vol. 30, pp. 210-210, 1993.
- [28] T. Oberg, A. Karsznia et K. Oberg, «Joint angle parameters in gait: reference data for normal subjects, 10-79 years of age.,» *Journal of rehabilitation research and development*, vol. 31, n° 13, p. 199, 1994.
- [29] J. L. Astephen, K. J. Deluzio, G. E. Caldwell et M. J. Dunbar, «Biomechanical changes at the hip, knee, and ankle joints during gait are associated with knee osteoarthritis severity,» *Journal of Orthopaedic Research*, vol. 26, n° 13, pp. 332-341, 2008.
- [30] "The ALTACRO Prototype," VUB, [Online]. Available: altacro.vub.ac.be. [Accessed 01 06 2013].
- [31] R. van Ham, B. Vanderborght, M. van Damme, B. Verrelst et D. Lefeber, «MACCEPA, the mechanically adjustable compliance and controllable equilibrium position actuator: Design and implementation in a biped robot,» *Robotics and Autonomous Systems*, vol. 55, n° 110, pp. 761-768, October 2007.
- [32] N. Instruments, «NI LabView - Improving the Productivity of engineers and scientists,» [Online]. Available: <http://www.ni.com/labview/>. [Accessed 07 05 2013].
- [33] «Qt,» [Online]. Available: <http://qt.digia.com/Product>. [Accessed 11 2012].
- [34] A. Gauld, «GUI Programming with Tkinter,» [Online]. Available: <http://www.freenetpages.co.uk/hp/alan.gauld/tutgui.htm>. [Accessed 2012].
- [35] D. Cuthbert, «GUI Programming in Python,» April 2012. [Online]. Available: <http://wiki.python.org/moin/GuiProgramming>. [Accessed 12 2012].
- [36] D. A. Vogel, «Software user interface, Requirements for medical devices,» *Intertech engineering associates*, August 2007.
- [37] C. Stanger, C. Anglin, W. Harwin et D. Romilly, «Devices for assisting manipulation: a summary of user task priorities,» *Rehabilitation Engineering, IEEE Transactions on*, vol. 2, n° 14, pp. 256-265, 1994.
- [38] J. Rose, J. Medeiros et R. Parker, «ENERGY COST INDEX AS AN ESTIMATE OF ENERGY EXPENDITURE OF CEREBRAL-PALSIED CHILDREN DURING ASSISTED AMBULATION,» *Developmental Medicine & Child Neurology*, vol. 27, n° 14, pp. 485-490, 1985.
- [39] S. Jezernik, G. Colombo, T. Keller, H. Frueh et M. Morari, «Robotic orthosis Lokomat: a rehabilitation and research tool,» *Neuromodulation: Technology at the Neural Interface*, vol. 6, n° 12, pp. 108-115, 2003.

- [40] R. Huq, E. Lu, R. Wang et A. Mihailidis, «Development of a portable robot and graphical user interface for haptic rehabilitation exercise,» chez *Biomedical Robotics and Biomechatronics (BioRob)*, 2012 4th IEEE RAS & EMBS International Conference on, 2012.
- [41] t. GmbH, *Instruction ManualPablo System Hand arm rehabilitation*, 2011.
- [42] «GUIs and which one to use,» [Online]. Available: <http://www.scriptol.com/programming/gui.php>. [Accessed 12 2012].
- [43] G. Colombo, M. Joerg, R. Schreier et V. Dietz, «Treadmill training of paraplegic patients using a rabotic orthosis,» *Journal of rehabilitation research and development*, vol. 37, n° 16, pp. 693-700, November/December 2000.
- [44] V. T. Inman, H. D. Eberhart et others, «The major determinants in normal and pathological gait,» *The Journal of Bone & Joint Surgery*, vol. 35, n° 13, pp. 543-558, 1953.
- [45] A. Leung, A. Mak et J. Evans, «Biomechanical gait evaluation of the immediate effect of orthotic treatment for flexible flat foot,» *Prosthetics and orthotics international*, vol. 22, n° 11, pp. 25-34, 1998.

Attachments

Attachment 1: Enlargements of Figure 2 and Figure 3 representing the settings of the GUI of Lokomat

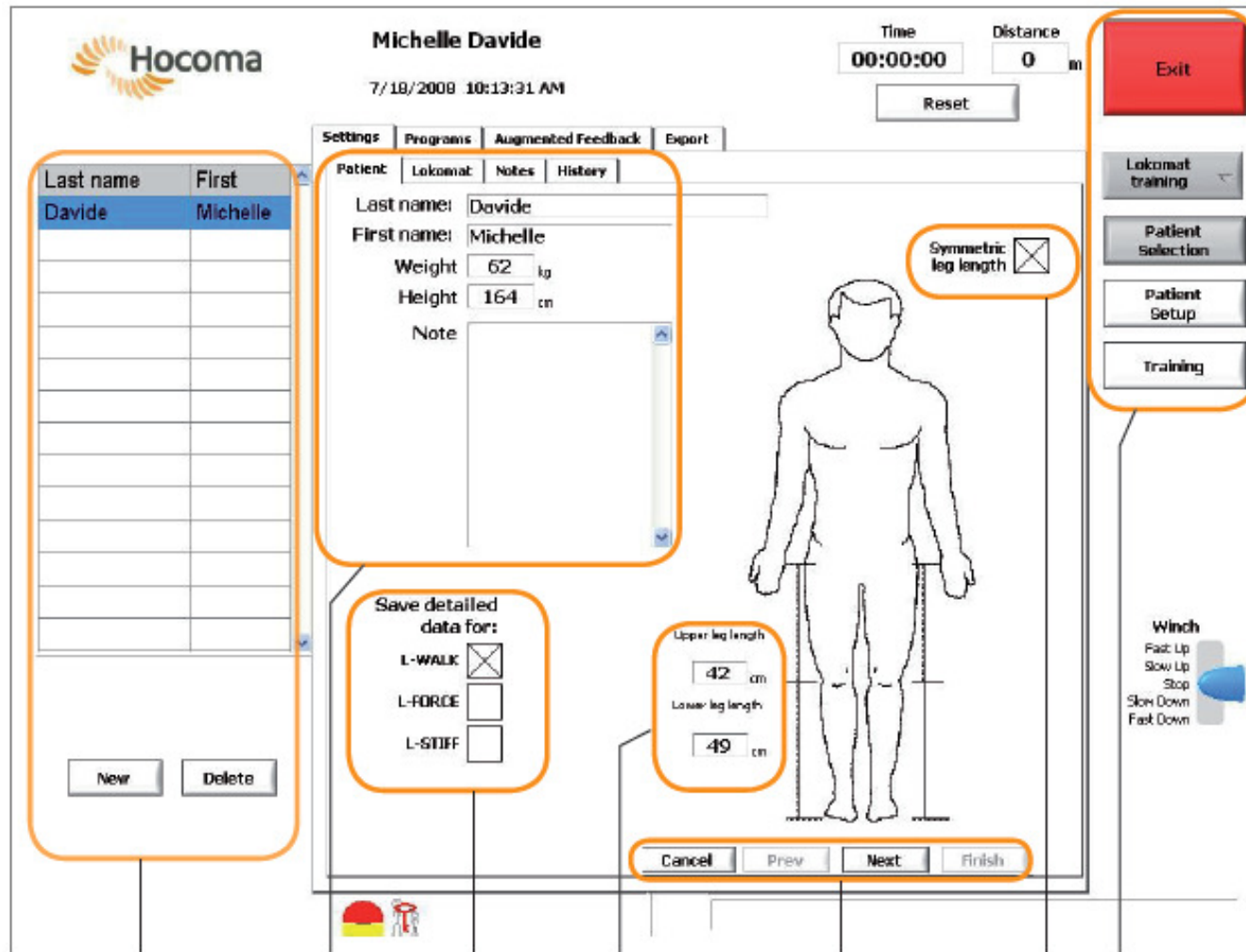


Figure 28: Screenshot of the patient settings tab of the Lokomat [8]

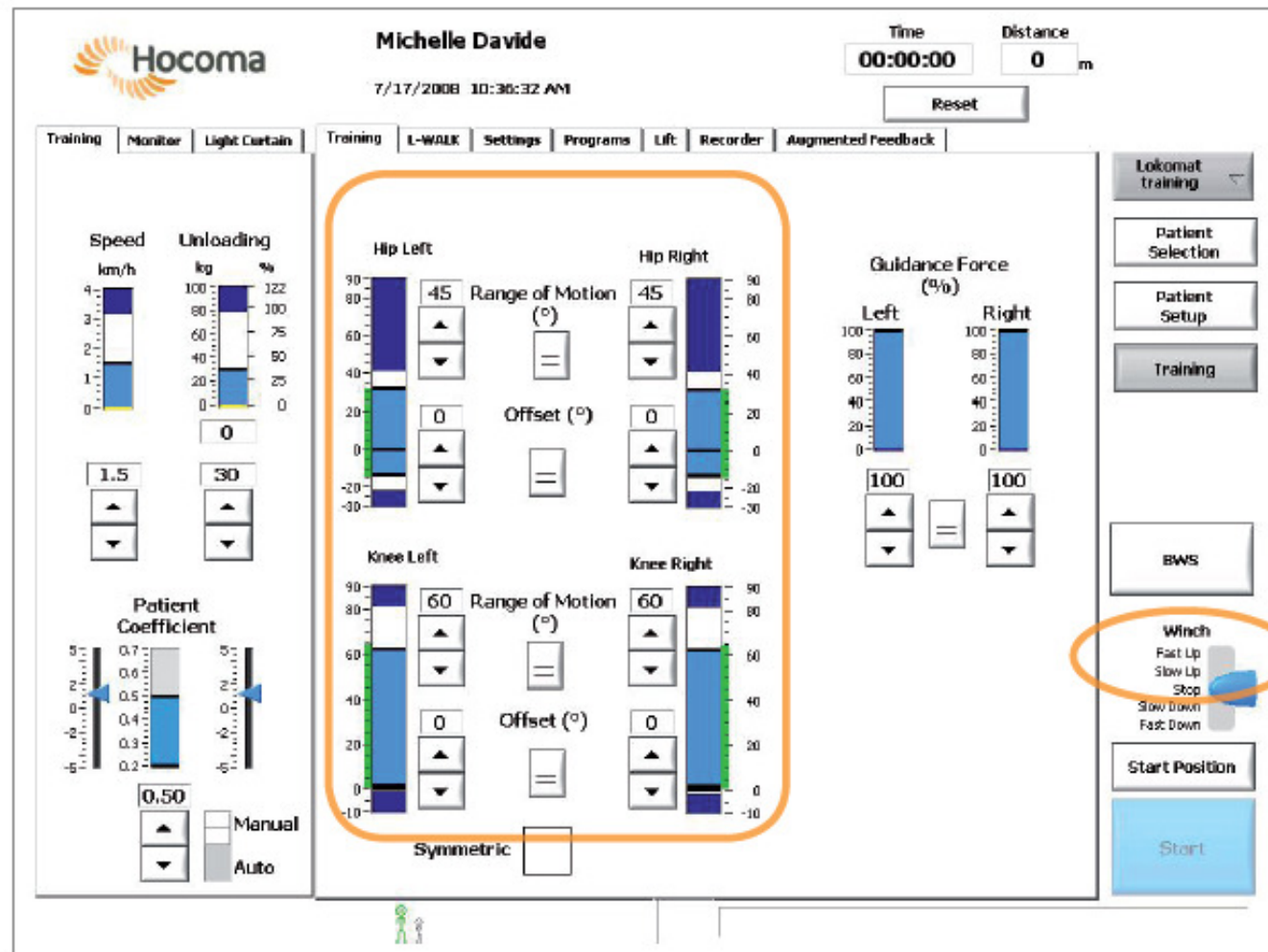


Figure 29: Screenshot of the training settings of the Lokomat [8]

Attachment 2: Comparison between programming environments for making GUI's

Table 15: Comparison between different programming environments

	Advantages	Disadvantages	GUI programming toolkit
Qt [33], [17], [16]	High portability	High size of the program	<i>QtSDK</i>
	High flexibility	Requires vast amounts of disk space for compilation	
	Openness		
	Mechanism called signals/slots => much easier to control communications between objects in a flexible and maintainable manner than it is with a fragile callback style [17]		
	Full API.		
	Fast.		
	OpenGL and WebKit components.		
	QML language of interfaces.		
	Available for free		
	Several programming languages possible.		
	Allows 3D animations.		
	QtCreator visual development interface.		
	Used on (some) mobile.		
	Good documentation.		
Java	Complete API for the desktop or the Web.	Rather complex to use but JavaFX.	Swing
	JavaFX language of interfaces.	Not especially fast.	
	Several development tools (see resources).		
	Portability.		
	Google GWT framework for the Web (converts Java to JavaScript).		
	Access to OpenGL with jogl and Swing Canvas object.		

WPF	Nothing to download to the user.	Works only on Windows. Ports to other systems are incomplete.	
	Language of interface XAML.	Rather slow in execution.	
	Silverlight is a lightweight version for Windows Phone 7.		
	Easy programming with Visual Studio.		
	Visual development tools for different languages. Silverlight plugin for Eclipse.		
Tcl (language)/ Tk (toolkit) [34], [35]	Ease of use.	Tcl is a dynamic language, and therefore is slow.	Tk/Tkinter
	Different programming languages available if needed.	Tk is designed for Tcl.	
	Portable.	Fewer built-in widgets	
	Most commonly used for Python	Rather simplistic layout abilities	
	Runs on any system where Python is installed		
FTLK	Portable (Windows, Linux, Mac and others).	Less comprehensive than previous alternatives.	
	Lightweight.	Restricted to the C programming language.	
	Supports OpenGL.		
	FLUID editor with a visual interface.		
GTK	Available on all operating systems	Less convenient than Qt	
		More difficult to program	
WxWidget	Different programming languages are possible	Lack of flexibility	

Attachment 3: Matlab code for the trajectory generation

```

%all the dimensions will be given in cm in this program

%initialisation of the program
clear all
close all
clc
%import of the Winter angles for hip, knee & ankle
ankle=importdata('Winter_NormalSpeed_Ankle_Angle_Periodic_1000p.mat');

%initialisation of the patient/session dimensions
L_feet=25;
L_lower_leg=45;
L_upper_leg=40;
L_upper_body=45;
L_hips=25;

Step_length=50;
Step_height=10;
Heel_off=5;
V_in_contact=10000;
V_heel_off=5;
percentage_swing=40;% phase1
percentage_ic=10; %phase2
percentage_mid_stance=40; %phase3
percentage_terminal_stance=100-percentage_swing-percentage_ic-
percentage_mid_stance; %phase4

amp_obl=L_hips/2*sin(8/180*pi); % this is the peak to peak amplitude of the
sine imposed on the pelvic obliquity in degree
amp_forw=L_hips/2*sin(10/180*pi); % peak to peak amplitude of the sine for the
pelvic rotation
amp_trunk=3.5;
hip_neutral_factor=2;

l=1000;%length(knee);
number_of_points_single_support=percentage_swing/100*l;
number_of_points_double_support=(l-2*number_of_points_single_support)/2;

% Names: Traj_1: Trajectory of the foot in the swing phase => completely
% removed from ground
% Traj_2: Heel is on the ground, the toes are going to the ground
% Traj_3: The foot is completely flat (Heel & toes on the ground)
% Traj_4: Heel is going from the ground to Heel_off, the toes are staying
% on the ground

%generation of the trajectory during the swing phase
[traj_1_x traj_1_z x_heel
z_heel]=traj_gen(Step_length,Step_height,Heel_off,V_in_contact,V_heel_off,numb
er_of_points_single_support);

%generation of the trajectory of the heel during the stance phase
% Generation of the trajectory of the heel during the initial contact (Traj_2)
p2=(percentage_mid_stance+percentage_terminal_stance)/(percentage_mid_stance+p
ercentage_terminal_stance+percentage_ic);

```

```

traj_2_x=Step_length:((p2*Step_length-Step_length)/(percentage_ic/100*1-
1):(p2*Step_length);
traj_2_z=0*traj_2_x;

% Generation of the trajectory during the mid stance (Traj_3)
p3=(percentage_terminal_stance)/(percentage_mid_stance+percentage_terminal_sta
nce+percentage_ic);
traj_3_x=(p2*Step_length):((p3-p2)*Step_length/(percentage_mid_stance/100*1-
1):(p3*Step_length);
traj_3_z=0*traj_3_x;

% Generation of the trajectory during the terminal stance + pre-swing
% (Traj_4)
p4=(percentage_mid_stance+percentage_terminal_stance)/(percentage_mid_stance+p
ercentage_terminal_stance+percentage_ic);
traj_4_x=(p3*Step_length):- (p3*Step_length/(percentage_terminal_stance/100*1-
1)):0;
traj_4_z=(traj_1_z(1)-traj_3_z(end))/(traj_1_x(1)-traj_3_x(end))*(traj_4_x-
traj_3_x(end))+traj_3_z(end);

ankle_stance=ankle(1:(1/2)); % take only the 40% of values of the single
support phase out of the Winter data
ankle_swing=ankle((1/2):end);

%Generation of a mean hip height + a superposed double sine
amp_up_work=amp_obl/2;
amp_forw_work=amp_forw/2;
x_sine=pi:4*pi/(1-1):5*pi;
z_sine=sin(x_sine);
z_neutral=sqrt((L_upper_leg+L_lower_leg)^2-
(Step_length/2)^2)+hip_neutral_factor
z_gen_left=z_neutral+amp_up_work*z_sine;
z_mean_hip=z_gen_left;
z_obliq_left=generator_sine(1);
z_obliq_left=z_obliq_left+z_mean_hip;

% the left foot starts with the IC (phase 2) and the hip at his furthest point
(in front), the right foot starts with phase4
z_forw=pi/2:2*pi/(1-1):5/2*pi;
x_forw=amp_forw_work*sin(z_forw);
x_trans_obliq_left=Step_length/2+x_forw;

percent=(percentage_ic+percentage_mid_stance)/100;
move_factor=percent*1;
x_trans_obliq_right=[x_trans_obliq_left(move_factor+1:end)
x_trans_obliq_left(1:move_factor)];
z_obliq_right=[z_obliq_left(move_factor+1:end) z_obliq_left(1:move_factor)];
x_mean_hip=x_sine-x_sine+Step_length/2;

% calculations of phase 1
%theta_ankle=[ankle_stance ankle_stance];
percent=(100-percentage_swing)/100;
move_factor=percent*1;
z_gen_phase1=z_obliq_left;
x_gen_phase1=x_trans_obliq_left;
z_gen_phase1=[z_gen_phase1(move_factor+1:end) z_gen_phase1(1:move_factor)];

```

```

z_mean_phase1=[z_mean_hip(move_factor+1:end) z_mean_hip(1:move_factor)];
x_gen_phase1=[x_gen_phase1(move_factor+1:end) x_gen_phase1(1:move_factor)];
[ theta_hip_1 theta_knee_1 theta_ankle_1 m1] = invers( traj_1_x, traj_1_z,
x_gen_phase1,z_gen_phase1,L_upper_leg, L_lower_leg,L_upper_body,L_feet,
Step_length, ankle_swing, 'phase1',traj_4_x, traj_4_x,0 );
[x_phase_1
z_phase_1]=coordinates_gen_inv(L_upper_body,L_upper_leg,L_lower_leg,L_feet,the
ta_hip_1, theta_knee_1,
theta_ankle_1, Step_length, z_gen_phase1, z_mean_phase1, x_gen_phase1, x_gen_phase1
);

% calculations of phase 2

x_inter_2=x_phase_1(6,end):- (x_phase_1(6,end)-
(traj_2_x(end)+L_feet))/(length(traj_2_x)-1):(traj_2_x(end)+L_feet);
line_toes_2=((0-z_phase_1(6,end))/(traj_2_x(end)+L_feet-
x_phase_1(6,end)))*(x_inter_2-x_phase_1(6,end))+z_phase_1(6,end);
x_toes_2=traj_2_x+sqrt(L_feet^2-(line_toes_2-traj_2_z).^2);

% creation of a trajectory for the toes
x_3_end=traj_3_x(end)+sqrt(L_feet^2-traj_3_z(end)^2);
x_inter_3=x_toes_2(end):- ((x_toes_2(end)-x_3_end)/(length(traj_3_x)-
1)):x_3_end;
line_toes_3=0*(x_inter_3-x_toes_2(end))+line_toes_2(end);
x_toes_3=traj_3_x+sqrt(L_feet^2-(line_toes_3-traj_3_z).^2);

x_end_4=traj_1_x(1)+L_feet*cos(asin(Heel_off/L_feet));
x_inter_4=x_toes_3(end):- (x_toes_3(end)-x_end_4)/(length(traj_4_x)-1):x_end_4;
line_toes_4=0*(x_inter_4-traj_3_x(end))+line_toes_3(end);
x_toes_4=traj_4_x+sqrt(L_feet^2-(line_toes_4-traj_4_z).^2);
x_inter_1=x_toes_4(end):((x_toes_2(1)-x_toes_4(end))/(length(traj_1_x)-
1)):x_toes_2(1);
spl=spline([x_toes_4(end) x_toes_2(1)], [line_toes_4(end) line_toes_2(1)]);
line_toes_1=ppval(spl,x_inter_1);
x_toes_1=traj_1_x+sqrt(L_feet^2-(line_toes_1-traj_1_z).^2);

x_inter_2=x_toes_1(end):- (x_toes_1(end)-
(traj_2_x(end)+L_feet))/(length(traj_2_x)-1):(traj_2_x(end)+L_feet);
line_toes_2=((0-line_toes_1(end))/(traj_2_x(end)+L_feet-
x_toes_1(end)))*(x_inter_2-x_toes_1(end))+line_toes_1(end);
x_toes_2=traj_2_x+sqrt(L_feet^2-(line_toes_2-traj_2_z).^2);

traj_ankle_x_l=[traj_2_x traj_3_x traj_4_x traj_1_x ];
traj_ankle_z_l=[traj_2_z traj_3_z traj_4_z traj_1_z ];
traj_toes_x_l=[ x_toes_2 x_toes_3 x_toes_4 x_toes_1];
traj_toes_z_l=[ line_toes_2 line_toes_3 line_toes_4 line_toes_1];

traj_ankle_x_r=[ traj_4_x traj_1_x traj_2_x traj_3_x];
traj_ankle_z_r=[ traj_4_z traj_1_z traj_2_z traj_3_z];
traj_toes_x_r=[ x_toes_4 x_toes_1 x_toes_2 x_toes_3];
traj_toes_z_r=[ line_toes_4 line_toes_1 line_toes_2 line_toes_3];

[ theta_hip_1 theta_knee_1 theta_ankle_1 m] = invers( traj_ankle_x_l,
traj_ankle_z_l, x_trans_obliq_left,z_obliq_left,L_upper_leg,
L_lower_leg,L_upper_body,L_feet, Step_length,
ankle_stance, 'phase2',traj_toes_z_l, traj_toes_x_l,0 );

```

```

[x_left
z_left]=coordinates_gen_inv(L_upper_body,L_upper_leg,L_lower_leg,L_feet,theta_
hip_l, theta_knee_l,
theta_ankle_l,Step_length,z_obliq_left,z_mean_hip,x_trans_obliq_left,x_mean_hi
p);

[ theta_hip_r theta_knee_r theta_ankle_r m] = invers( traj_ankle_x_r,
traj_ankle_z_r, x_trans_obliq_right,z_obliq_right,L_upper_leg,
L_lower_leg,L_upper_body,L_feet, Step_length,
ankle_stance, 'phase2',traj_toes_z_r,traj_toes_x_r,0 );
[x_right
z_right]=coordinates_gen_inv(L_upper_body,L_upper_leg,L_lower_leg,L_feet,theta
_hip_r, theta_knee_r,
theta_ankle_r,Step_length,z_obliq_right,z_mean_hip,x_trans_obliq_right,x_mean_
hip);

x_global=zeros(11,length(x_left));
x_global(1,:)=x_left(6,:);
x_global(2,:)=x_left(5,:);
x_global(3,:)=x_left(4,:);
x_global(4,:)=x_left(3,:);
x_global(5,:)=(x_left(3,:)+x_right(3,:))/2;
x_global(6,:)=x_left(1,:);
x_global(7,:)=x_global(5,:);
x_global(8,:)=x_right(3,:);
x_global(9,:)=x_right(4,:);
x_global(10,:)=x_right(5,:);
x_global(11,:)=x_right(6,:);

z_global=zeros(11,length(z_left));
z_global(1,:)=z_left(6,:);
z_global(2,:)=z_left(5,:);
z_global(3,:)=z_left(4,:);
z_global(4,:)=z_left(3,:);
z_global(5,:)=(z_left(3,:)+z_right(3,:))/2;
z_global(6,:)=z_global(5,:)+L_upper_body;
z_global(7,:)=z_global(5,:);
z_global(8,:)=z_right(3,:);
z_global(9,:)=z_right(4,:);
z_global(10,:)=z_right(5,:);
z_global(11,:)=z_right(6,:);

figure
for i=1:(length(x_left))

    %plot(x_left(:,i),z_left(:,i),'b-*',x_right(:,i),z_right(:,i),'r-o');%
    plot(x_global(:,i),z_global(:,i),'b-*');
    axis([-10 120 -10 120],'square');
    pause(0.001);
end

% as control for the trajectories
figure;
plot(traj_ankle_x_l,traj_ankle_z_l,'r',x_global(2,:),z_global(2,),'b');
title('trajectory of the ankle-left');
legend('desired trajectory', 'real trajectory');

```

```

dev=sqrt((traj_ankle_x_1-x_global(2,:)).^2+(traj_ankle_z_1-z_global(2,:)).^2);

figure;
plot(traj_toes_x_1,traj_toes_z_1,'r',x_global(1,:),z_global(1:,:),'b');
title('trajectory of the toes-left');
legend('desired trajectory', 'real trajectory');

function [traj_x traj_z x_heel
y_heel]=traj_gen(Step_length,Step_heigth,Heel_off,V_in_contact,V_heel_off,numb
er_of_points)

x_heel=[0 (Step_length)];%1+0.05)*
y_heel=[Heel_off 0];

%Generation of the ankle trajectory by making a cubic spline between the
%Heel off and the initial contact
xx_1=0:x_heel(2)/(number_of_points-1):(x_heel(2));
spl=spline(x_heel,[V_heel_off y_heel -V_in_contact]);
pp=ppval(spl,xx_1);
[m i]=max(pp);

%reshaping of the Generated trajectory in order to have the desired foot
%lift
xx_2=0:Step_length/(number_of_points-1):Step_length;
lin=(y_heel(2)-Heel_off)/(Step_length-0)*(xx_2-0)+Heel_off;
pp=pp-lin;
pp=pp/(m-lin(i))*(Step_heigth-lin(i));
pp=pp+lin;

traj_x=[xx_1];
traj_z=[pp];
end

function [theta_hip theta_knee theta_ankle m] = invers( traj_x, traj_z,
x,z,L_upper_leg, L_lower_leg,L_upper_body,L_feet, Step_length,
ankle_stance,phase,z_toes,x_toes,m_p )%x_sol z_sol
% calculation of the invers kinematics for the hip, knee and ankle joints
%calculation of the new knee angles with the invers kinematics
% x and z are the given coordinates of the hip extremity's

for i=1:length(traj_x)
c2(i)=((traj_z(i)-z(i)).^2+(traj_x(i)-x(i)).^2-L_upper_leg^2-
L_lower_leg^2)/(2*L_upper_leg*L_lower_leg);
end
[m1 in]=max(c2);
m=max(m1,m_p);
for i=1:length(traj_x)
%c2(i)=c2(i)/(m);
s2(i)=sqrt(1-c2(i).^2);
theta_knee(i)=atan2(s2(i),c2(i));
end
%initialisation of zero matrices for the calculation of the angles with
the inverse kinematics in the swing phase
theta_hip=zeros(1,length(traj_x));

```



```

c1=zeros(1,length(c2));
s1=zeros(1,length(c2));

%calculation of the new hip angles
if phase=='phase1'
    for i=1:length(c2)
        if theta_knee(i)<0
            theta_knee(i)=-theta_knee(i);
        end

c1(i)=(1/(L_upper_leg^2+L_lower_leg^2+2*L_upper_leg*L_lower_leg*c2(i)))*((L_upper_leg+L_lower_leg*c2(i))*(z(i)-traj_z(i))-L_lower_leg*s2(i)*(traj_x(i)-x(i)));

s1(i)=(1/(L_upper_leg^2+L_lower_leg^2+2*L_upper_leg*L_lower_leg*c2(i)))*(L_lower_leg*s2(i)*(z(i)-traj_z(i))+(L_upper_leg+L_lower_leg*c2(i))*(traj_x(i)-x(i)));

        theta_hip(i)=atan2(s1(i),c1(i));

    end
    theta_ankle=ankle_stance;
elseif phase=='phase2'
    for i=1:length(c2)
        if theta_knee(i)<0
            theta_knee(i)=-theta_knee(i);
        end

c1(i)=(1/(L_upper_leg^2+L_lower_leg^2+2*L_upper_leg*L_lower_leg*c2(i)))*((L_upper_leg+L_lower_leg*c2(i))*(z(i)-traj_z(i))-L_lower_leg*s2(i)*(traj_x(i)-x(i)));

s1(i)=(1/(L_upper_leg^2+L_lower_leg^2+2*L_upper_leg*L_lower_leg*c2(i)))*(L_lower_leg*s2(i)*(z(i)-traj_z(i))+(L_upper_leg+L_lower_leg*c2(i))*(traj_x(i)-x(i)));

        theta_hip(i)=atan2(s1(i),c1(i));

        c_kah(i)=(x_toes(i)-x(i)-L_upper_leg*sin(theta_hip(i))-L_lower_leg*sin(theta_hip(i)-theta_knee(i)))/L_feet;
        s_kah(i)=(z(i)-z_toes(i)-L_upper_leg*cos(theta_hip(i))-L_lower_leg*cos(theta_hip(i)-theta_knee(i)))/L_feet;
        theta_kah(i)=atan2(s_kah(i),c_kah(i));
        theta_ankle(i)=theta_knee(i)-theta_hip(i)-theta_kah(i);
    end
end
end

function[x
z]=coordinates_gen_inv(L_upper_body,L_upper_leg,L_lower_leg,L_feet,hip, knee,
ankle,Step_length,z_extremity_hip,z_mean_hip,x_extremity_hip,x_mean_hip)
x=zeros(5,length(hip));
z=zeros(5,length(hip));

for i=1:length(hip)

    x(1,i)=x_mean_hip(i);
    x(2,i)=x(1,i);

```

```
x(3,i)=x_extremity_hip(i);
x(4,i)=x(3,i)+L_upper_leg*sin(hip(i));
x(5,i)=x(4,i)+L_lower_leg*sin(hip(i)-knee(i));
x(6,i)=x(5,i)+L_feet*cos(knee(i)-ankle(i)-hip(i));

z(1,i)=z_mean_hip(i)+L_upper_body;
z(2,i)=z_mean_hip(i);
z(3,i)=z_extremity_hip(i);
z(4,i)=z(3,i)-L_upper_leg*cos(hip(i));
z(5,i)=z(4,i)-L_lower_leg*cos(hip(i)-knee(i));
z(6,i)=z(5,i)-L_feet*sin(knee(i)-ankle(i)-hip(i));
end
end
```