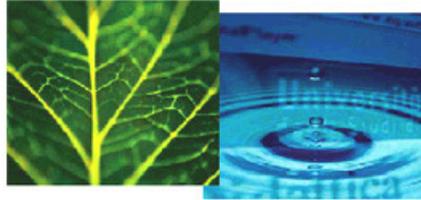


PhD Dissertation

---



**International Doctorate School in Information and  
Communication Technologies**

DISI - University of Trento

**A ROBOTIC WALKING ASSISTANT FOR LOCALISATION AND  
GUIDANCE OF OLDER ADULTS IN LARGE PUBLIC SPACES**

Federico Moro

Advisor:

Prof. Luigi Palopoli

Università degli Studi di Trento

---

April 2015



# Abstract

*Ageing is often associated with reduced mobility which is the consequence of a combination of physical, sensory and cognitive degrading. Reduced mobility may weaken older adults' confidence in getting out alone and traveling autonomously in large spaces. We have developed a robotic walking assistant, that compensates for sensory and cognitive impairments and supports the user's navigation across complex spaces. The device is a walker with cognitive abilities that we named c-Walker, and it is built around a common walker for elderly people. We show the difficulties that arise when building a robotic platform, focusing on hardware and software architecture for the basic functionalities and integration of high level software components. We developed an Extended Kalman Filter in such a way that we are able to select a configuration of sensors that meets our requirements of cost, accuracy, and robustness. We describe the technological and scientific foundations for different guidance systems, and their implementation in the device. Some of them are "active" meaning that the system is allowed to "force a turn" in a specified direction. The other ones are "passive" meaning that they merely produce directions that the user is supposed to follow on her own will. We show a comparison of the different guidance systems together with the results of experiments with a group of volunteers.*

## **Keywords**

[Assisted living, software architecture, localisation, passive guidance]



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Objectives . . . . .	2
1.3	Related work . . . . .	3
1.3.1	Ambient assisted living . . . . .	3
1.3.2	Software architecture . . . . .	3
1.3.3	Localisation . . . . .	4
1.3.4	Guidance systems . . . . .	6
1.4	Structure of the Thesis . . . . .	7
<b>2</b>	<b>System architecture</b>	<b>9</b>
2.1	Mechatronic level . . . . .	10
2.2	Functional modules . . . . .	14
<b>3</b>	<b>Localisation</b>	<b>19</b>
3.1	System configuration . . . . .	19
3.2	Components . . . . .	19
3.2.1	Wheel encoders . . . . .	20
3.2.2	Visual Odometry . . . . .	21
3.2.3	RFID tags . . . . .	22
3.2.4	Model based visual localisation . . . . .	22
3.3	Position tracking techniques . . . . .	24
3.3.1	Kalman filter framework . . . . .	24
3.3.2	Modalities . . . . .	26
3.4	Evaluation . . . . .	27
3.4.1	Environmental setup . . . . .	27
3.4.2	Description of experiments . . . . .	30
3.4.3	Ground truth and error metrics . . . . .	30
3.4.4	Experimental results . . . . .	31

<b>4</b>	<b>Guidance</b>	<b>45</b>
4.1	Guidance mechanisms . . . . .	45
4.1.1	Bracelets . . . . .	45
4.1.2	Audio interface . . . . .	47
4.1.3	Mechanical Steering . . . . .	48
4.2	Guidance algorithms . . . . .	49
4.2.1	Haptic and Acoustic algorithms . . . . .	50
4.2.2	Mechanical system: steering . . . . .	54
4.3	Implementation . . . . .	56
4.4	Experimental results . . . . .	58
4.4.1	Study 1 . . . . .	58
4.4.2	Study 2 . . . . .	67
<b>5</b>	<b>Steering by brakes</b>	<b>71</b>
5.1	Problem Formulation . . . . .	71
5.1.1	Half Cart Model . . . . .	71
5.1.2	Vehicle Dynamic Model . . . . .	72
5.1.3	Dynamic Path Following Problem . . . . .	73
5.1.4	Problem Formulation . . . . .	74
5.2	Solution . . . . .	74
5.2.1	Hybrid Solution to the Path Following Problem . . . . .	74
5.2.2	Braking System . . . . .	75
5.3	Simulations and Experiments . . . . .	77
5.3.1	Simulation Results . . . . .	77
5.3.2	Experimental Results . . . . .	77
<b>6</b>	<b>Conclusion</b>	<b>83</b>
	<b>Bibliography</b>	<b>89</b>

# Chapter 1

## Introduction

### 1.1 Motivation

Ageing is often associated with reduced mobility which is the consequence of a combination of physical, sensory and cognitive degrading.

Reduced mobility may weaken older adults' confidence in getting out alone and traveling autonomously in large spaces. Other factors have an adverse effect on mobility, the most obvious being physical impairments, loss or reduction of visual and auditory ability and of the key function of balance.

Less recognised but as important is the decline of cognitive abilities such as timely reaction to external stimuli, sense of direction, peripheral vision and navigation skills [81]. Cognitive problems like these are difficult to recognise and with a very few counter-strategies of proven effect. The afflicted gradually perceives such places as shopping malls, airports or train stations as unfamiliar and intimidating and starts to withdraw [45].

A growing body of research [82] suggests that a reduced out-of-home mobility can have widespread, detrimental effects for older adults and ultimately accelerate the process of ageing. Adults for whom mobility is a problem certainly experience a reduction in the quality of their social life. They have fewer choices in terms of where and when they can shop, and they have been found to have problems in maintaining a balanced diet.

Reduced mobility has other several serious consequences including an increase in the probability of falls and other physical problems, such as diabetes or articular diseases. Staying at home, people lose essential opportunities for socialisation and may worsen the quality of their nutrition. The result is a self-reinforcing loop that exacerbates the problems of ageing and accelerates physical and cognitive decline [13].

Several studies reveal that physical exercise ameliorates the general conditions of older adults, by increasing their physical strength and reducing the occurrence of falls [34, 42]. The use of robotic platforms to support navigation is commonly believed as an effective

strategy to offset the negative trend toward a reduced mobility of older adults and to spur them toward a sustained level of physical activity.

In the context of different research initiatives (the *DALi* project<sup>1</sup> and the *ACANTO* project<sup>2</sup>) we have developed a robotic walking assistant, that compensates for sensory and cognitive impairments and supports the user's navigation across complex spaces. The device is a walker with cognitive abilities that we named *c-Walker*.

## 1.2 Objectives

The development of a device of this type, being it a research project or a business product, requires a careful methodology. Applications and algorithms need to be developed in isolation and in parallel. The possible need to extend the functionalities asks for a software architecture able to adapt to different services. We need to be able to define a run-time configuration that is able to exploit new stream of data as soon as they are available, and keep a basic functionality when they are not. Moreover, communication among the software components should be independent from the their allocation in the computing units, since we make use of a distributed architecture. Customization is our objective, not only at the software level, but also at the hardware level where we need to be able to add or remove new sensors or actuators without specific attention at the communication and software level. We would like to reduce the procedure to a simple connection (plug and play) and have the software automatically handle the new stream of data.

The device has to be able to localise itself without help of the user. Recalling the above notion of customization, we want to be able to select different sets of sensors and fuse the data in order to obtain the location. This objective comes from potentially different requirements in terms of costs, complexity, performance, and robustness of the localisation functionality.

Finally, we want to be able to use localisation information in order to guide the user. We selected different types of actuators: we have actuators that act directly on the kinematic of the walker, and actuators that provide signals to the user. We need to be able to understand how each actuator can be best exploited, which one is preferred from users, and what kind of performance we can expect.

---

<sup>1</sup><http://www.ict-dali.eu>

<sup>2</sup><http://www.ict-acanto.eu>

## 1.3 Related work

### 1.3.1 Ambient assisted living

Robotic walkers have gained an undisputed popularity in the research community on ambient assisted living [46, 76, 51]. Closely related to DALi is the Assistants for SAfe Mobility (ASSAM) project [2], where a system consisting of a set of modular navigation assistants deployed on several devices encourages physical exercise. ASSAM has a major focus on the seamless transition from indoors to outdoors, while DALi specifically considers large indoors environment. Also, the behaviour of people in the surroundings is not considered in ASSAM.

The iWalkActive project [38] is more directly focused on the development of an active walker that facilitates and supports physical exercise, both indoors and outdoors. E-NO-FALLS [23] is another project focused on the development of a smart walker. In this case the main emphasis is on preventing falls.

Although of interest, these aspects are complementary with those of DALi, whose main focus is on fighting cognitive decline.

### 1.3.2 Software architecture

A cyber physical system (CPS) is, in the common lingo, a device or a system where the computation units are deeply interconnected with the physical system they control [49, 50]. In this sense, the *c-Walker* is very close to this definition since the kinematics of the walker and the effects on it generated by the user behavior affect the input data stream, and therefore the computational load on the electronic components.

CPSs are usually characterized by a heterogeneous and distributed architecture and, more frequently, have the ability to share information and services with other CPSs disseminated in the environment setting the basis for an “internet of things” [3].

Another characteristic is the number and complexity of control functions and their interconnection which is supported by a variety of sophisticated sensing devices and the related perception algorithms. The overall complexity is due to the need for a high degree of autonomy, and for reconfiguration and adaptation capabilities which provide robustness to changing and unanticipated environment conditions.

The integration of this complex network of modules calls for a middleware solution striking a good tradeoff between conflicting needs such as: modularity, architecture independence, re-use, easy access to the limited hardware resources and real-time constraints.

There are different middleware solutions that are compatible with Linux-based systems and that support the most used network protocols.

A first one is Open Data Distribution Service (OpenDDS) [61], which is an open source C++ implementation of the Object Management Group [62] Data Distribution Service (DDS). DDS is a type of Message Oriented Middleware (MOM) that supports a data-centric publish and subscribe style of communications. It comes from the experience of the CORBA community and offers a high level of abstraction.

With a special care for real-time performance, Open Real-Time Ethernet (ORTE) [77] is an open source implementation of Real-Time Publish-Subscribe (RTPS) communication protocol. Timing and reliability are taken under control because of the use of the UDP protocol. The widespread of the TCP/IP stack over different systems and architectures guarantees a good portability.

Other middleware solutions explicitly developed for robot applications are ROS [67] and OROCOS [9]. ROS has become a rich repository of algorithms and software modules developed by the research community. OROCOS was primarily created to address control tasks in industrial environments. Both seem to be more suitable for applications that can rely on a powerful computing architecture, not always available in autonomous robots unless a cloud infrastructure is present. One of the main characteristic is the possibility to interface different components at deploy-time.

ZeroMQ [35] implements a publish-subscribe paradigm to support concurrent programming over socket connections. It is lightweight and very suitable for embedded architectures, and is freely available from its website [89].

### 1.3.3 Localisation

Given that the literature on indoor localisation is vast and spans many different disciplines, a comprehensive review of dominant technologies is given in [79]. [26] aims at providing the reader with a review of the recent advances in wireless indoor localisation techniques.

Distance measurements can rely on ultra-sound sensors [88], laser scanners [52], radio signal strength intensity (RSSI) [85], time-of-arrival (ToA) measurements of Radio Frequency (RF) signals [32] or RF identification (RFIDs) readers [56]. Cameras [72] are frequently used for positioning systems as well, and the approach of SLAM combines mapping and self-localisation in a natural way.

RF solutions are widespread and can rely on IEEE 802.11 [25], Ultra-Wideband (UWB) [20], ZigBee [72], Bluetooth [54], or a combination thereof. However, except for the case of UWB (which can be very accurate, but also very power-hungry when the time of arrival of pulses is measured), in most cases the accuracy of RF-based solution is quite limited. Performance can be improved through finger printing [43]. In this case, prior to localisation, an off-line radio scene analysis is performed to extract radio fingerprints, i.e., features of the radio signal measured at predefined points in the environ-

ment. Unfortunately, the solutions based on fixed anchors (even the most accurate) can be severely affected by lack of line-of-sight (LOS) conditions caused by fixed or moving obstacles. This problem could be mitigated by using suitable inertial measurement units (IMU) [14, 18]. Nonetheless, in this case accuracy tends to degrade indefinitely due to the accumulation of the uncertainty contributions of the various sensors employed. In addition, the initial position and orientation are not observable with inertial techniques. For all the reasons above, nowadays it is recognized that the best approach for high-performance and scalable indoor navigation should rely on both local inertial techniques and absolute positioning solutions, properly combined through data fusion algorithms [18, 53].

Since we deal with the localisation of a wheeled device, a powerful resource for positioning is offered by odometry. However, as customary of dead reckoning techniques, odometry-based localisation suffers from unbounded uncertainty growth and lack of initial observability. While position and orientation errors generally increase with a rate depending on both odometer resolution and accuracy, estimation results can be considerably improved by fusing gyroscope and encoder data on the basis of their respective uncertainties in different conditions of motion [21]. Unfortunately, also in this case there are no guarantees to keep the overall position uncertainty bounded. Moreover, the initial state of the system is still unobservable. To tackle these problems, an additional absolute localisation technique is certainly needed. Absolute position values can be obtained from a set of passive RFID tags. In fact, they are inexpensive, can be stuck on the floor at known locations, and, even if they have a quite limited range (in the order of a few tens of cm or less), they can be easily detected, regardless of the number of people and obstacles in the environment. Three good solutions of this kind are described in [65, 11] and [10]. In [65] a fine-grained grid of passive RFID tags is used for robot navigation and trajectory reconstruction. No other sensors are employed. In [11] a similar approach is adopted, but an additional vision system is used to recognize the color patches placed on the top of different robots.

Finally, in [10] a similar grid of RFID tags is used along with a set of ultrasonic sensors installed on the front side of a robot for position refinement through data fusion. A common characteristic of the solutions mentioned above is high accuracy, which however is paid in terms of RFID grid granularity. In fact, in all cases the grids of tags are very dense (with distances between about 0.3 m and 0.5 m), which is costly and impractical in very large environments. Moreover, the fixed external cameras in [11] pose privacy and scalability issues, while the on-board ultrasonic sensors (which refine position in the presence of fixed obstacles) could lead to unpredictable results in densely populated environments.

In [47] a smart walker instrumented with encoders, a compass and an RFID reader

corrects the odometry-based position by reading mats of RFID tags placed in strategic points of corridors (i.e. where people are supposed to come across with a high probability).

#### 1.3.4 Guidance systems

The robot wheelchair proposed in [80] offers guidance assistance in such a way that decisions come from the contribution of both the user and the machine. The shared control, instead of a conventional switch from robot to user mode, is a collaborative control. For each situation, the commands from robot and user are weighted according to the respective experience and ability leading to a combined action.

Other projects make use of walkers to provide the user with services such as physical support and obstacle avoidance. In [15], the walker can work in *manual mode* where the control of the robot is left to the user and only voice messages are used to provide instructions. A shared control operates in *automatic mode* when obstacle avoidance is needed and user intention is overridden acting on the front wheels. The *park mode* is used when the walker needs to maintain a certain position and sustain the user.

The use of force sensors can help in understanding user intentions, as in [83] where the front wheels are used to modify orientation angle in case of concerns about the ease and safety of the user motion. In [16] a omnidirectional mobile base makes possible to change the center of rotation to accommodate user intended motion.

The *JAIST active robotic walker* [51] (JaRoW) constantly uses infrared sensors to detect lower limb movement of the user. In this way, the walker autonomously adjusts direction and velocity to the user's walking behavior.

The passive walker proposed by Hirata [36] takes safety a step further. The device is a standard walker, with two caster wheels and a pair of electromagnetic brakes mounted on fixed rear wheels, which is essentially the same configuration that we consider in this work. The authors propose a guidance solution using differential braking, which is inspired to many stability control systems for cars [66]. By suitably modulating the braking torque applied to each wheel, the walker is steered toward a desired path. While this choice poses severe limitations on the force and torque applicable to the cart, it has the considerable advantage of limiting the complexity of the hardware, with considerable savings in the cost of the device and in its mass. The same principle has been further developed by Saida et al. [71] achieving richer kinematic behaviours.

In [44], differential flatness is used to determine time-varying braking gains in order to achieve a control law that is both passive and dissipative. One of the main advantages of this approach is the reduced computational complexity of the system dynamic equations and brake gain constraints.

The work in [27] proposes a control algorithm based on the solution of an optimisa-

tion problem which minimises the braking torque. The paper considers a virtual tunnel enclosing the path. When the user is in the middle of the corridor, the system intervenes sporadically, while it becomes increasingly aggressive when the user is close to the border. Two potential limitations of this strategy are its frequent corrections (annoying for some of the users) and its reliance on real-time measurements of the torques applied to the walker, which are difficult without expensive sensors.

Haptic interfaces can be used in robotics applications as a mean of communication between robot and user. Examples could be teleoperation of vehicles for surveillance or exploration of remote or dangerous areas where haptic interfaces provide feedbacks on sense of motion and the feeling of presence [1], or rescue activities where the robot helps the user to move in environments where visual feedbacks are no longer available [31]. In the latter application the robot provides information on its position and direction to the user in order to help him to follow the robot. Guidance assistance can be provided by giving feedback on the matching between the trajectory followed by the user and the planned trajectory. In [75, 73], a bracelet provides a warning signal when a large deviation with respect to the planned trajectory is detected. In [24] a belt with eight tactors is used to provide direction information to the user in order to complete a waypoint navigation plan.

Acoustic guidance can be achieved recreating sounds coming from precise locations. Therefore, reproduction of 3-D sound signals can be used to give directional aids to the user. The main method to render 3-D sound is based on the Head Related Transfer Function (HRTF) which changes and needs to be determined for each individual [7]. It represents the ears response for a given direction of the incoming sound. Other approaches are based on the modeling of the sound propagation. In the modeling process, attenuation of the sound is taken in account by means of the Interaural Level Difference (ILD) which considers the presence of the listener head. Instead, Interaural Time Difference (ITD) considers the distance between ears and sound source [8]. These filtering processes are computationally demanding, requiring implementations suitable for execution on embedded platforms [68, 69].

## 1.4 Structure of the Thesis

We have already show the motivations behind the work presented in this thesis. The work covers some architectural objectives in common to all the components of the entire DALi project, that are described in Cha. 2.

Cha. 3 continues the description of how integration is performed at the architectural level, but also shows some functional implementation of some modules. In particular, we

address the localisation system of the *c-Walker*.

We present a comparison of different guidance solutions in Cha. 4 together with an experimental session in the field with users.

Cha. 5 is the proposal of a new guidance system. We show a possible use of the back wheels brakes for the implementation of a low cost guidance technique.

We finally conclude with Cha. 6.

## Chapter 2

# System architecture

In order to master the complexity of the system, the development of the *c-Walker* prototype has been organised based on a clear separation of concerns between the different components. This idea was implemented through a clear partitioning of the different functionalities and of the mechatronic components used to implement them.

Each component has been accurately specified during the design phase. This way it was possible to develop it in isolation and its final integration into the complete prototype was significantly simplified. The *c-Walker* is seen as a three layers structure where the subsystems are shown in Fig. 2.4:

- The physical subsystem,
- The mechatronic subsystem,
- The cognitive subsystem

The physical subsystem consists of the walker frame used as a basis for the implementation of the *c-Walker*. The key requirement for the choice of the walker frame was the possibility to easily integrate and possibly change the electronic and electro-mechanical components used for on-board sensing and actuation. This can be important in case some new application requirements arise during the experimental phase.

Other requirements are the cost and the commercial availability of the basic prototype. This is a key requirement in view of the potential commercial exploitation. The requirement is equally important for the ability to build multiple copies of the device for research purposes. Moreover, during the development, the prototype has to go under an intense testing phase which implies the transportation of the prototype in different locations. Therefore, transportability is another important requirement.

Our final choice was to take an off-the-shelf device (the Mercury rollator produced by

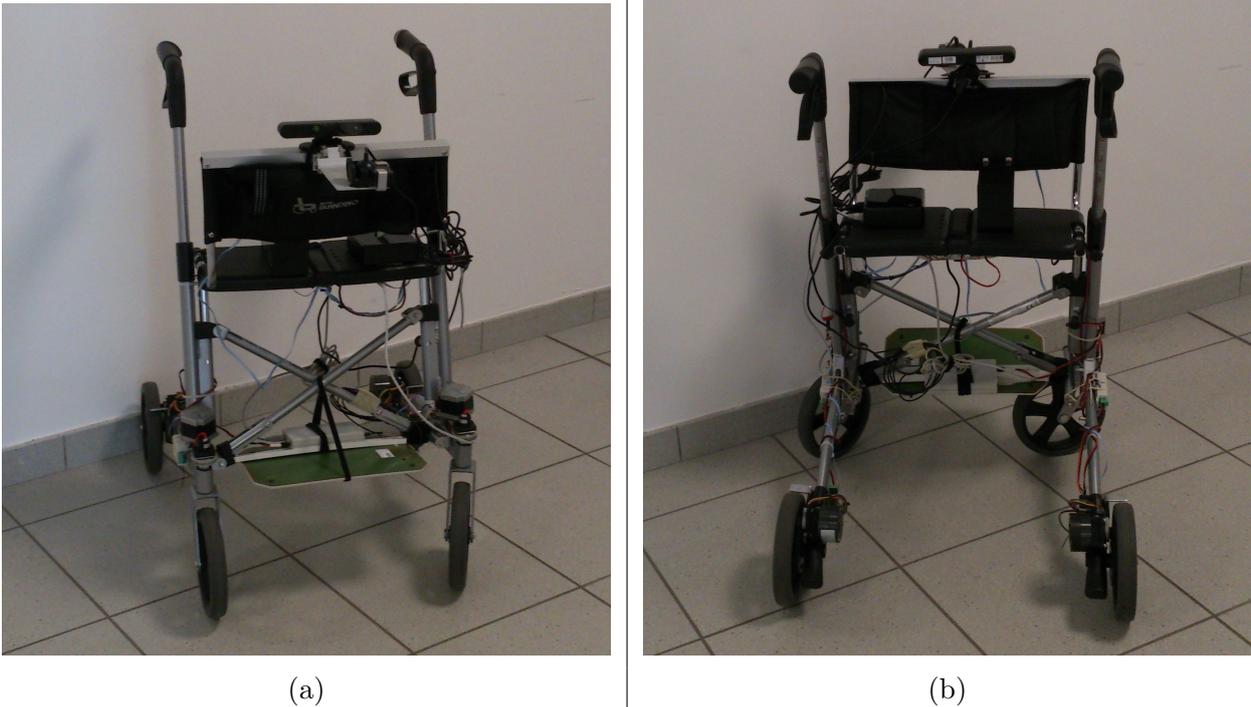


Figure 2.1: Views of the *c-Walker* with all the equipment (a) *c-Walker* seen from the front side, (b) *c-Walker* seen from the back side.

Nuova Blandino) which is not equipped with any device, but meets most of the requirements and has a structure which makes possible to add mechatronic components.

Fig. 2.1 shows two different views of the *c-Walker* equipped with all the devices described in Sec. 2.2.

## 2.1 Mechatronic level

The mechatronic subsystem is a set of sensors and actuators strictly related with the body frame of the walker. It is organised around a Control Area Network (CAN bus). At the moment, six nodes are present in the network: one node per wheel, an Inertial Measurement Unit (IMU), and a Beaglebone. The Beaglebone is responsible for the execution of the software component which periodically gathers data from the sensors of the mechatronic subsystem and provides the data to the “upper” level.

The Beaglebone, besides a CAN interface, has also an Ethernet port. This makes the board also part of the computing cluster where the cognitive subsystem is executed. At the architectural level, the Beaglebone works mainly as a bridge between the two different networks (Ethernet and CAN) and, therefore, as a software interface between the mechatronic and cognitive subsystems. The use of a CAN bus network makes the

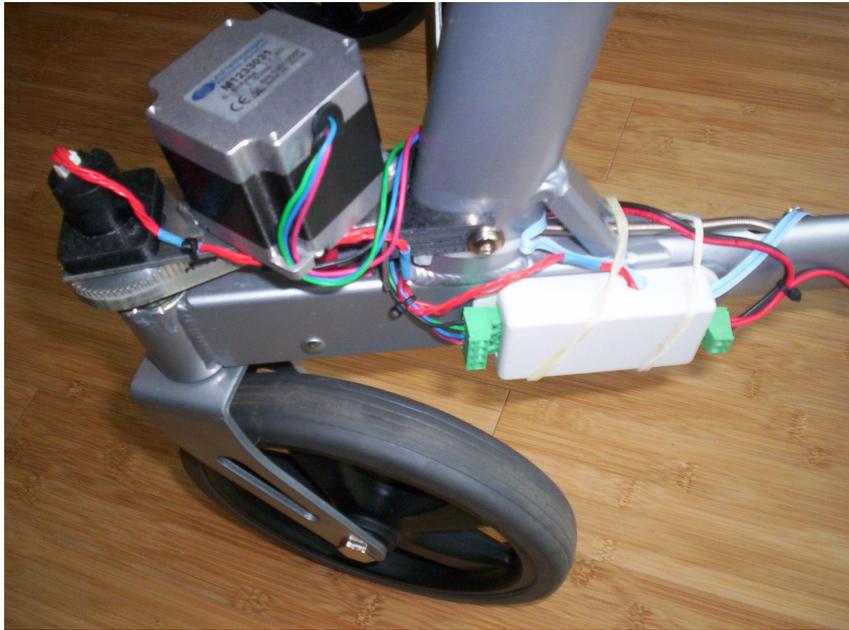


Figure 2.2: Front wheel: CAN bus node, motor and absolute encoder are visible.



Figure 2.3: Back wheel: CAN bus node, brake and gear of the incremental encoder are visible.

subsystem flexible because it is easier to add or remove a node. The only thing to take care of is to make sure that the Beaglebone knows about the existence of a sensor and knows how to recognize and interpret its messages. Removal of a sensor will result only in the absence of messages from the sensor without requiring specific care at the Beaglebone side.

The cabling required for the CAN bus can also be used for the power supply wires. Besides the IMU, which is powered directly from the 5 V of the CAN bus, all the other nodes (Beaglebone included) receive a 12 V input, which is also the main input power of the system. Every node has its own regulator circuit which takes care of providing the sensors, actuators and processing units in the node with the appropriate voltage.

The nodes mounted on the wheels have a microcontroller to handle messages and to interface with the sensors and actuators. The nodes on the back wheels have an incremental encoder and a electric brake, while the nodes on the front wheels have an absolute encoder and a motor.

In Fig. 2.3, it is possible to see the gear of the encoder, the brake and the node. Incremental encoders are responsible for counting the impulses produced by the rotation of the wheels. The *c-Walker* kinematic model is equivalent to the one of the unicycle with the back wheels being the two wheels considered in the model. Encoders data can be used to determine linear and angular velocity of the walker frame and estimate its location. The brakes mounted on the back wheels are electromagnetic devices which can be actuated governing the operational current. The CAN bus node receives the set point for the corresponding brake and periodically shares its current status. The brakes are used in one of the guidance systems that could be exploited by the cognitive subsystem.

The front wheels are connected to a swivel that allows the free wheels to mechanically adapt to the direction of the walker. We mounted two motors to the joints which allow us to force independently the rotation of the wheels. The motor, together with the absolute encoder and the relative CAN bus node, is visible in Fig. 2.2. Each one of the two nodes can receive commands which require a rotation specifying direction, velocity and quantity of movement. The motor also provides the number of steps of the shaft during actuation. Wheel and motor may not move in accordance due to some friction issues, therefore on the joint there is also an absolute encoder which allows us to track the actual orientation of the wheel. Motors on the front wheels have been mounted in order to implement a passive guidance system alternative to the brakes guidance system. The possibility to use the motors in a complementary system with the brakes is something yet to be explored.

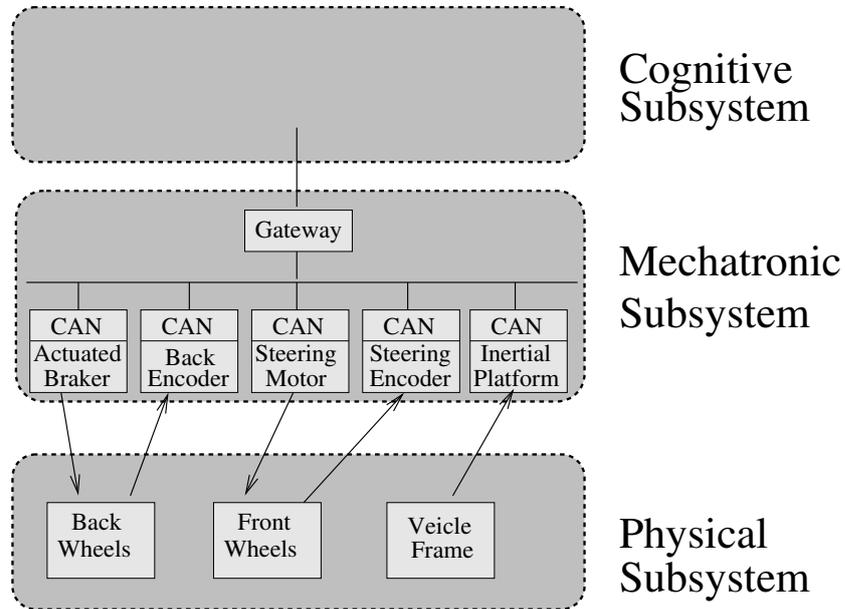


Figure 2.4: Conceptual architecture of the *c-Walker*.

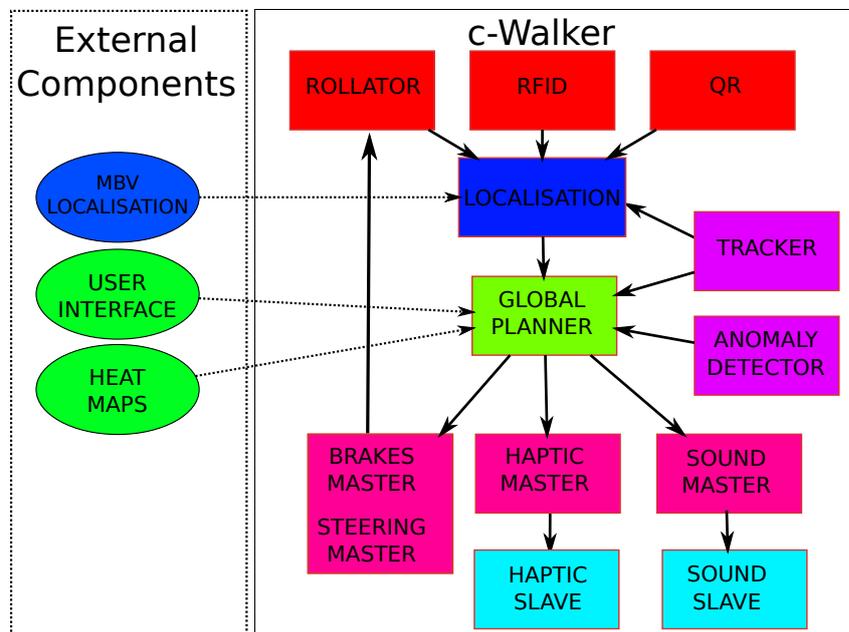


Figure 2.5: Functional diagram of the system where *c-Walker*'s internal modules and external modules are shown.

## 2.2 Functional modules

The cognitive subsystem encompasses all the abilities required to sense the environment, to decide a plan and to see its correct execution in interaction with the assisted person (e.g., deciding the set point for the guidance system implemented in the mechatronic layer, and more generally through the different HMI solutions). Together with the cognitive subsystem goes the most of the algorithmic components of the *c-Walker*.

In this subsystem, all the issues about the computational hardware and software architecture are addressed. We decided to adopt a cluster of computing units connected in a Ethernet network. Some of these units are directly connected to the devices used in the system: sensors and actuators. The software components are allocated to a computing unit in such a way that data flow from sensors is optimized. The communication between the different software components is managed with a Publish/Subscribe communication middleware. We selected ZeroMQ as messaging system, since it allows for easy allocation of software components in different hardware modules. In particular, the Publish/Subscribe method makes easy to establish communication between two components that start execution in an asynchronous fashion. If a particular data output stream is needed by more modules, multiple subscriptions to the same publisher are performed without requiring software modification. The API is an abstraction of the TCP/IP socket communication system and, besides simplifying the code, it allows to implement the communication between two modules regardless of them being executed on the same platform or in different units.

A wireless communication allows to exchange information also with the external components. Fig. 2.5 does not show details of the cognitive subsystem since it is too complex to be described in a single box, but it gives an overview of the interactions between the majority of the components.

The mechatronic subsystem, represented with the *Rollator* component, is a fundamental source of information for the cognitive subsystem, since it provides movements and position updates of the *c-Walker*. It also provides an alternative solution for system corrective manipulation by actuating directly the mechanic, and therefore dynamic, of the *c-Walker* itself. The objective of the *Localisation* module is to estimate in real-time the position of the *c-Walker*. Such information is represented as X and Y coordinates with respect to a known reference frame, and orientation expressed as the angle between the X axes of the *c-Walker* and the reference frame, with the X axis of the *c-Walker* corresponding to the forward direction. The algorithm is an Extended Kalman Filter (EKF) that relies on multi-sensor data fusion [57]. The available sensors are the IMU, back wheels encoders, a Radio-frequency identification (RFID) reader, a camera for Quick

Response (QR) code detection, and a Kinect.

Encoders and IMU can be used to extract relative motion information of the *c-Walker*, while RFID and QR provide global information about respectively position in terms of X and Y coordinates, and orientation. Both components provide this information by detecting/recognising the relative sensor object: RFID tag or QR code. In particular, the QR code recognition is performed with an open source library *zbar* which, given an image, determines if a QR code is present, and returns the image coordinates of the four corners. The list of four corners is returned always in the same order and therefore it is possible to have a global idea of the orientation of the code. Provided that the orientation of the camera is known, it is possible to apply a roto-translation to the four points in order to extract a more precise orientation of the mark [28]. The library also returns the Id of the code which is stored in the code itself. For both RFID and QR the position of the tags/marks is known.

Images captured with the Kinect are used by two modules. The *Tracker*, besides tracking people in the surroundings, is able to provide relative motion information analyzing two consecutive frames. The Model-Based Visual Localisation (*MVL*) is a cloud service that receives an image and returns a global position of the *c-Walker*. The server already contains a model of the environment built with a series of pictures. The received frame is matched with the model in order to reconstruct the position of the camera, and therefore the vehicle. The *Global Planner* is responsible to produce an optimal path to visit all of the requested points of interest from a specified initial position. The planner manages the persistent data related to the map of the environment and its working copy. All modules that require information about the map and/or modify the knowledge of the environment are required to communicate with the planner (e.g. heat maps, anomaly detector). *Global Planner* consists of three main components: 1) the graph constructor, which generates a data structure to represent the map of the environment and to create a persistent storage of the map, 2) the graph manager, which manages the working copy of the data structure and handles communication with external modules and update the local state of the map, 3) the planner, which produces the optimal path based on the current working copy of the map [17]. Generated plans are given as input to a guidance system. We have different solutions based on active or passive approaches. Active approaches are based on mechanics components like front steering wheels and back braking wheels. While, other approaches that do not have an active interaction with the *c-Walker* frame, are considered passive and rely on stimuli signals that indicate the user the appropriate direction. In this category we have systems that use an haptic interface and an audio interface. The terminology active/passive is not to be confused with the one that defines where the motive force comes from. In this sense, our solutions are always passive guidance systems

since it is the user who provides it.

Our requirement analysis and our business case have converged on the support of alternative configurations. For example, the following three configurations could be proposed as an effective response to clearly identify market targets and user requirements.

- **Configuration 1:** is the basic setup all other configurations build upon. The configuration is intended for normally-abled users who simply ask for navigation support. The main two functionalities are the *Localisation* and the *Global Planner*. The user is allowed to enter her/his destination and the system produces a motion plan that accounts for distances to be covered, density of people along the way and preferences of the user (e.g., the user may wish to always have a policeman within easy reach). While the user moves, the system tracks her/his position and produces (on request) a large basis of potentially useful information on the touch screen. It is worth observing that the two functionalities are independent and rely on different hardware devices. Therefore, each subsystem can be subject to changes in the algorithm and hardware without impacting the other one.
- **Configuration 2:** introduces a “lightweight” guidance support for users in need of an higher level of assistance. This support is implemented by haptic devices and/or audio interfaces. The guidance support comes along with a more complete version of the *Global Planner* module, which acquires information of the surroundings and modifies the plan to enforce safety requirements for the user (Local Planner).
- **Configuration 3:** adds a more aggressive guidance system (the so-called mechanical guidance). This configuration is utilised when the user has a low level of autonomy and requires a very accurate assistance. The architecture modularity allows us to create different configurations of the running system. This is possible by selecting only the required modules of the functional diagram in Fig. 2.5.

The *Localisation* module itself is an example of configurability. In fact, selection of different hardware components can create different systems because of the diversity of sensor information utilized in the data fusion. The choice of the selected hardware is motivated by a possible trade-off between system complexity, cost and efficiency. Cha. 3 will develop this issue more in depth.

At the moment, the majority of the software components of the cognitive subsystem are executed in an industrial PC which is very compact and provides enough computational performance. Therefore, the computational infrastructure is composed mainly by the PC and the Beaglebone. Experiences in the field showed that a more distributed architecture could be needed with configurations that require intensive use of great part

of the components here described. Different computing units allow for the isolation of functionalities and guarantee the required computational power. In particular, capturing simultaneously from multiple vision devices is expensive from the resources point of view.



# Chapter 3

## Localisation

### 3.1 System configuration

The main purpose of the localisation module is to continuously track the position of the *c-Walker* inside the environment, thus enabling efficient planning and guidance algorithms. With location we refer to the position in the map (represented as X and Y coordinates in cartesian plan) and also the orientation.

The requirements on the localisation module can be summarized as follows:

- No, or very low cost, instrumentation or alteration of the environment thus resulting in little to no maintenance.
- Positioning accuracy of at least one meter (in diameter) plus high angular resolution.
- Very high responsiveness as it is required for dynamic short term planning ( 1s) as well precision motion control ( 0.1s)
- A positioning system that can solely work on the walker whenever external connectivity is restricted (“survival mode”).

The module is not on mainstream wireless localisation techniques (WiFi, Bluetooth, etc.) because some of these restrictions. While we are open to incorporate wireless localisation techniques as additional cue whenever available, this was beyond the scope of our project since we think they will advance on a general application level.

### 3.2 Components

The localisation module can be made of different software components, each one extracting information from a specific sensor. Based on the nature of the sensor the complexity of

Modality	Incremental (I) Absolute (A)	Instrumented environment	Processing: on-board (L) cloud service (CS)	Computational load @ mobile platform
Wheel encoders	I		L	→
Visual odometry	I		L	↑
RFID	A	X	L	→
Model-based Visual Localisation	A	X	CS	→

Table 3.1: Overview on Modalities.

the data processing changes, as well as the type of information extracted: incremental information or global information. Incremental information are used to determine the movement the *c-Walker* has performed starting from a known location, and to derive the new configuration. Global information is the actual location of the *c-Walker*, or part of its configuration (i.e., only orientation).

Tab. 3.1 shows a list of components used in this particular study. In the table different features are described together with the ones mentioned above. In particular, some components also require instrumentation of the environment. The Model-based Visual Localisation (MVL) also requires a wireless connection in order to exploit a cloud service.

What follows is a description of the different components, even though not all the available sensors used in the project are considered. The selected ones should cover all the features listed in Tab. 3.1. We also describe some localisation techniques that can be built using different sensors and their main characteristics. Some of the components could be used in isolation to localise the *c-Walker*, but is the combination of some of them that can achieve better performance.

### 3.2.1 Wheel encoders

Dead reckoning based on encoders represents the basic component for the presented solution to the localisation problem. For the system at hand, we have an encoder for each of the two back wheels, thus allowing measurements of both angular and linear displacements of the vehicle.

As widely recognized in the literature, dead reckoning solutions are prone to uncertainty accumulations over time, mainly due to modeling inaccuracies and measurement noises on the adopted sensor. Besides the intrinsic uncertainty of the encoders, the un-

avoidable approximations of the mechanical components, mainly of the wheels radii and wheel axle length, lead to a systematic error. Moreover, the underlying assumptions for the adopted encoder-based localisation are: i) motion constrained on a plane surface; ii) slippage avoidance of the wheels with respect to the ground and iii) pure rolling motion. It is evident that the use of the sole encoders makes impossible to determine when these assumptions are violated.

Nevertheless, the simplicity of the encoder data opens to high rate sampling and therefore to a frequent update of the position estimates. As a consequence, the encoder data are used in the prediction step of a Kalman filter, which is based on the unicycle-like model. The choice of the Kalman filter reduces the computational load of the on-board computing system due to its iterative nature.

### 3.2.2 Visual Odometry

*This software component has been developed by Foundation for Research & Technology - Hellas (FORTH)*

FORTH's implemented solution [64] provides simultaneous localisation mapping and moving object tracking (SLAMMOT) in order to support the short range path planning module of the *c-Walker*. A direct by-product of the method is visual odometry which can be used as an additional cue to be fused with the rest of the localisation modalities built into the *c-Walker*.

FORTH's method uses RGB-D input acquired by the front facing sensor to build and maintain a point cloud that represents the immediate environment around the user. It uses a sparse 3D point cloud created for each frame and employs Particle Swarm Optimization (PSO) to fit it on a dense model that is appropriately built and maintained.

On each frame, a number of points are selected from a superset produced by a feature detector using the RGB input of the sensor. These points are then filtered and only the ones with reliable depth values are kept. The point cloud produced by the filtered points is then tested against the dense 3D point cloud of the model. The objective function which gives the matching score between the input and the model, projects the points as they would be seen from the camera and compares the generated depth maps of the dense model and the filtered features.

In order to find the change in the position and orientation of the platform in the world, multiple hypotheses are generated and evaluated using Particle Swarm Optimization (PSO) [41]. PSO is an evolutionary algorithm that achieves optimization based on the collective behavior of a set of particles that evolve in runs called generations. The rules that govern the behavior of particles emulate "social interaction". A population of particles is essentially a set of points in the parameter space of the objective function

to be optimized. Canonical PSO, the simplest of PSO variants, has several attractive properties. More specifically, it only depends on very few parameters, does not assume knowledge of the derivatives of an objective function and requires a relatively low number of objective function evaluations. Moreover each generation of each particle does not depend on the other particles. This allows for efficient implementations that compute all the particles of each generation in parallel.

The resulting estimation produced by PSO is the relative movement of the platform in the reference frame of the model. This estimation can be used in order to update the model of the environment as well as to provide the additional visual odometry cue to the walker's localisation module.

### 3.2.3 RFID tags

Radio-frequency identification (RFID) is used to obtain global position information. The environment is instrumented with a series of tags with known ID and known location. An antenna is placed on the vehicle that keeps emitting signals. When the vehicle is in the proximity of a tag (usually an area of few centimeters), the emitted signal powers the tag and allows it to send a message containing its ID. Upon ID reception, a search is performed on a lookup table and location information is extracted. This global information is fused with the available localisation estimate.

As for the encoders, data acquisition consists on message reception from a device. The data rate is subject to the available signal emission frequency of the device. Data is injected in a Kalman filter in the form of measurement update. Again, due to the nature of the Kalman filter, the data fusion process is not computationally heavy for the mobile platform.

As mentioned before, this subsystem asks for the instrumentation of the environment. This opens the problem of finding the minimal number of tags to place in order to cover the motion area in such a way that enough updates are performed and, hence, the desired target localisation accuracy is met.

### 3.2.4 Model based visual localisation

*This software component has been developed by Siemens*

In many scenarios, the environment, in which localisation is performed, is not subject to rapid changes in appearance, but more or less static over time. In such cases, MVL is a powerful alternative to the global localisation methods described so far. Fig. 3.1 roughly illustrates the idea of MVL. First, a 3D Model of the environment of interest is created by capturing a multitude of highly overlapping images from different viewpoints. Using

the concept of Structure-from-Motion (SfM), point correspondences are found between images, from which the relative orientation between the cameras can be computed. These point correspondences can then be triangulated to form a point cloud in 3D space, which is aligned with the metric coordinate system of the real-world setting. The resulting point cloud, together with the camera positions and orientations, serves as a synthetic model of the environment. We used a slightly modified version of the incremental SfM system described in [37]. With this system, we were able to create models from hundreds of images in the order of a few minutes. Note that this process only needs to be carried out once for each environment, as long as its appearance roughly stays the same.

In order to perform localisation, SIFT features are extracted in the camera image of interest. As in the model creation stage, point correspondences are searched by matching these features with the ones detected in the images used for capturing the model. Since matching with typically hundreds of images is both time-consuming and, because of non-overlapping image content in most images, unnecessary, a Vocabulary-Tree was used to find the most similar images for matching, as described in [60].

Some of the matching feature points in the model images have 3D points associated with them, yielding 2D-3D point correspondences between the image to be localised and the point cloud. These correspondences are used to solve the Absolute Pose Problem, requiring at least three such correspondences [33]. The problem is usually highly over-determined, since much more correspondences can typically be found. Therefore, we solved the problem using a RANSAC loop followed by an optimization over all inliers. It is highly important to identify and eliminate wrong pose estimates, which can occur due to the presence of wrong feature matches. The number of such matches can be significantly high, but they are typically not consistent with the same solution. Therefore, we defined a threshold of RANSAC inliers required for a valid solution. If the number of inliers is above this threshold, but below a second, higher one, we additionally require the inlier-outlier ratio to be above a certain value.

In the specific scenario of using a wheeled walker for localisation, we can also identify wrong pose estimates if the height of the resulting camera position does not accord with the fixed height on the walker, or if the viewing angle contains invalid pitch or roll components. Since it should be possible to combine the localisation result with pose estimates from other modalities using an Extended Kalman Filter, the covariance of the pose must be estimated. Assuming Gaussian noise with variance  $\sigma$  corrupting the feature locations, the covariance matrix  $\Sigma$  is computed as

$$\Sigma = \left( \frac{1}{2} H \right)^{-1} \quad (3.1)$$



Figure 3.1: Model based Visual Localization Pipeline. Multiple overlapping Images are used to create a synthetic model of the environment. In the localization stage, 2D-3D correspondences are found to compute the camera pose within this model.

, with  $\mathbf{H}$  being the Hessian of the reprojection error  $\mathbf{E}$ . The reprojection error function is defined as

$$E = \sum_{i=1}^N \|PX_i - \bar{x}_i\|^2 \quad (3.2)$$

, where  $N$  is the number of correspondences,  $X_i$  the  $i$ -th 3D point,  $P$  the associated camera projection matrix  $\bar{x}_i$  and its corresponding 2D feature location.

### 3.3 Position tracking techniques

#### 3.3.1 Kalman filter framework

The tracking of the position is performed by means of a Kalman filter based on the kinematic unicycle-like model. Since the RGB-D camera adopted for the Visual Odometry and the RFID antenna used for global position measures are mounted on the same position in front of the walker, the Cartesian coordinates of such a frontal point of the walker, namely  $X$  and  $Y$ , and the orientation of the vehicle  $\theta$  are chosen as the system state. Therefore, the kinematic of the frontal point and the orientation are expressed as:

$$\dot{\mathbf{Q}} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cos \theta - L\omega \sin \theta \\ v \sin \theta + L\omega \cos \theta \\ \omega \end{bmatrix} \quad (3.3)$$

where  $v$  and  $\omega$  are the linear and angular velocities, and  $L$  is the distance between the front and the back wheels. The nonlinearity of the system is addressed using point-wise linearization and discretization, as customary in the Extend Kalman Filter (EKF) approach here adopted. As a consequence, the inputs  $v$  and  $\omega$  of the system, after discretization, become the linear and angular displacement of the vehicle, which are directly observed with the encoders.

An alternative way to realize the prediction step is by integration of the relative displacements observed with Visual Odometry. Since the nature of both the encoders and the Visual Odometry is the same, we use the same system description for both these incremental approaches, which, in turn, simplifies the fusion of the two methods for the prediction step. It is worthwhile to note that the combination of encoders and Visual Odometry strengthen the prediction robustness to: i) typical encoders nuisances, as aforementioned; ii) when the scene is not sufficiently rich for the Visual Odometry; iii) in case of occasional hardware failures.

In [29], two different techniques to fuse data coming from different sensors have been presented: one consists on a fusion in the measurement space which is then used in a standard Kalman filter, the second one is the fusion of the estimation returned from separated updates, each one based on the different sensors available. In the second case, the fused predictions are used in the update step of a Kalman filter.

We follow this latter approach: two predictions, based on encoders and Visual Odometry, are independently carried out, while the other sensor updates are performed in the update step of the EKF as customary. It has to be noted that the sensors used in prediction come with radically different rates, as the Visual Odometry is related to the camera sampling rate (usually at 30 Hz), while the encoders can be sampled with much higher rate (e.g., hundreds of Hz). Therefore, after the computation of the two parallel prediction tracks, the fusion, which is based on the weight of the two sensors [29], is performed upon the reception of the Visual Odometry estimates. After the fusion, the two tracks are kept independent. The solution here proposed is necessarily suboptimal due to the intrinsic nonlinearity of the system.

As previously described, data coming from the sensors that provide absolute measurements are used in the update step of the EKF. Based on the nature of the sensor, it is possible to update the whole state or only part of it. Moreover, while some of the global readings can be available on a time-based fashion, e.g., the model-based visual localisa-

tion, some other are purely event-based, e.g., the RFID tags. Due to this characteristic, the approach followed is the fusion directly in the measurement space for the update step. Notice that the structure of the EKF thus defined represents a framework that makes possible the integration of additional very different sensors.

A further problem to be addressed is the presence of outliers coming from the model-based visual localisation whenever different portions of the environment are visually highly similar or in presence of scenes with a port level of distinctiveness. To increase the resiliency to outliers of the presented estimation scheme, we chose the Median Absolute Deviation (MAD) criterion [70]. To this end, we first compute the median and the standard deviation of the correct measurements  $z_c$  collected during the experimental trials by visual inspection. Next, during the execution of the algorithm, the outlier is detected each time the following measure

$$f = \frac{|z - \text{median}(z_c)|}{\text{std}(z_c)} \quad (3.4)$$

exceed 3, as explained in [70]. The approach thus described proves to be very effective in terms of outliers removal for the experiments at hand.

### 3.3.2 Modalities

In the process of selecting a technique, there are different issues to keep in consideration which are also motivation to the study described in this chapter.

Encoders, although a relative low cost in terms of hardware, may require a particular effort from the engineering point of view. The mounting on the wheels may not be straight forward and may require the study of a gearing system. Moreover, a precise measurement of the vehicle characteristics such as wheel radius and axle length is needed in case of porting to a different walker or system.

The Visual Odometry, instead, has a lower cost (represented by the cost of the Kinect) and the porting is performed by simply placing the sensor on the vehicle.

The two components are alternatives for two different techniques that rely on Visual Odometry or encoders for the prediction step in EKF. The use of both of them can be considered in favour of the reliability of the system. In fact, each one of the sensors can compensate for failures of the other, as mentioned in the EKF description.

Global location measures are therefore provided from RFID or MVL. The first requires an accurate instrumentation of the environment, but for very large scenarios a complete instrumentation could not be affordable. The latter guarantees a better coverage given that a wireless connection is guaranteed. However, it is probable to have a wireless coverage in a public spaces and this could avoid efforts of the instrumentation procedure

of the RFID system. A visit on-site, although, is needed to reconstruct the model in the environment. Furthermore, while the RFID is robust to changes in the scenario, the MVL is sensitive to insertion or removal of elements of the environment.

The experimentation was aiming at gathering data in order to test the performance of a set of techniques. We also defined as objective a successful integration of the different components that can allow for the creation of different techniques that are in some way independent from the potential improvement, in isolation, of a single component. A first analysis is focused on solely incremental solutions, while a second analysis considers the following techniques:

1. **WE + RFID:** encoders and RFID. Mid cost due to the engineering and the instrumentation of the environment. No need for a cloud infrastructure.
2. **WE + RFID + VO:** encoders, RFID and Visual Odometry. Mid-high cost due to the addition of a sensor.
3. **WE + MVL:** encoders and Model-based Visual Localisation. Mid cost, does not need any particular instrumentation. An infrastructure is needed for the cloud service.
4. **VO + MVL:** Visual Odometry and Model-based Visual Localisation. Low cost and use of a single sensor.
5. **TOT:** encoders, RFID, Visual Odoemtry and Model-based Visual Localisation. High cost since it requires all the components.

The MVL is also studied in isolation.

## 3.4 Evaluation

### 3.4.1 Environmental setup

All experiments have been conducted in two rooms linked by a corridor. See Fig. 3.2 for the room layout. The interior of the rooms has been prepared to reflect a shopping mall like appearance by placement of large paper cubes with custom texture applied. Additionally textures have been applied to pin board stands in room 2. Sample textures can be seen in Fig. 3.2. The rest of the room interior consisted of furniture like office equipment (desks, chairs), a wardrobe, posters, etc. as well as some special equipment since those rooms are laboratories (pipes, electric control cabinet). The corridor did stay rather plain, and hence lacks any visual features. In each room we did place nine equally distributed RFID tags.

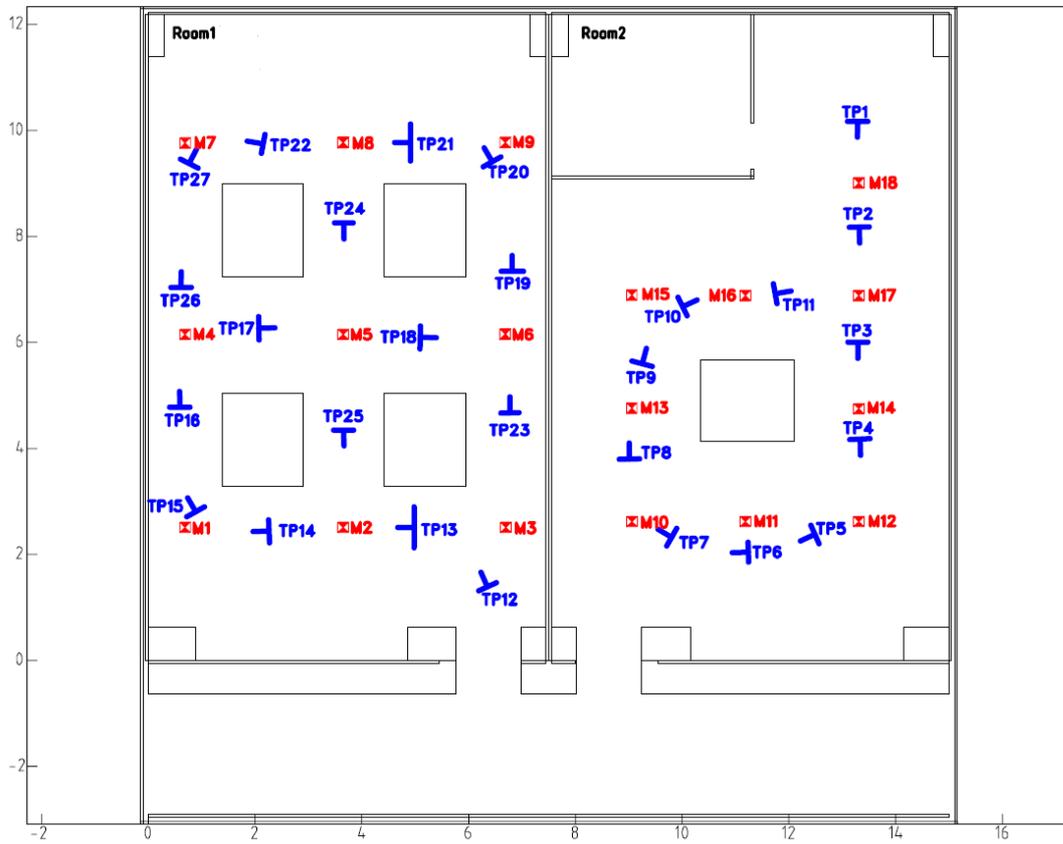


Figure 3.2: Room layout with placement of RFID tags (red) and placed waypoints (blue).

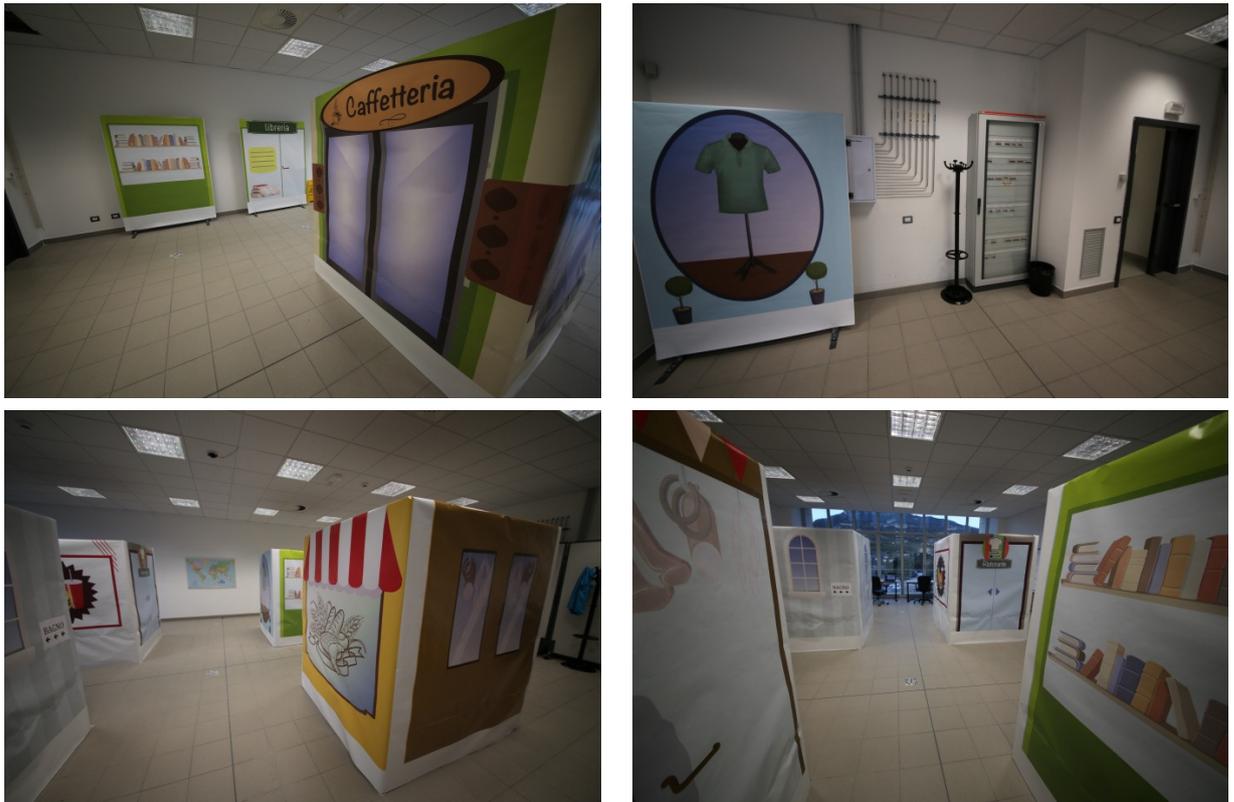


Figure 3.3: Impressions of the experimental setup.

Experiment	Description	Waypoints traversed	Room	Options
A	P-shape	1-11, 2 1	2	LR/HR EM/CR
B	J-Shape with reverse	1-7, 7-1	2	LR/HR
C	S-Shape horizontal	12-22	1	LR/HR EM/CR
D	S-Shape vertical	12,23,19- 21,24,25,14- 16,26,27	1	LR/HR
E	D and B connected	22-12, 7-1	Both	LR/HR

Table 3.2: Description of individual experiments conducted.

### 3.4.2 Description of experiments

#### Description of Options:

- LR. Standard resolution of Asus Xtion camera sensor (i.e. 640x480) in favor of higher frame rates (up to 25 fps).
- HR. Higher resolution of Asus Xtion camera sensor (i.e. 1280x1024) but with drop in framerate to x fps.
- EM. No or very sporadic appearance of people in the scene.
- CR. Simulates frequent appearance of up to 3 people in front of the camera.

### 3.4.3 Ground truth and error metrics

For reasonable precise re-positioning of the walker at the location of different waypoints we did implement the following configuration. Two laser pointing devices, actually laser range meters but we did only use their laser beam, have been mounted on the inside of the walker's rear wheel axis on each side respectively. As the walker drives two laser beams are projected onto the ground continuously. When reaching a desired waypoint position we did place perforated stickers (two for each walker position hence we can determine position and orientation) on the ground and did label them.

After placing all waypoint stickers we did manually measure them with respect to three RFID tags which are visible from that waypoint and afterwards triangulated their

positions. We believe this configuration allows for a precision in ground truth as well as re-positioning on the 1-2 centimeter level as well as around one degree in orientation. This uncertainty has to be kept in mind during the experimental results as some of the modalities will get close to the uncertainty present in the ground truth.

**Pose:** Pose in our case of an Ackermann steering geometry consists of an x and y position as well as a planar orientation in a 3 DOF-configuration.

**Relative Pose Error (RPE):** RPE measures the reconstructed relative transformations between nearby poses to the actual relative transformations (ground truth). In our case the ground truth poses are quantized to subsequent waypoints hence we estimate the RPE with respect to the waypoint spacing. RPE is well-suited for measuring the drift of a dead reckoning system, for example drift per waypoint.

**Self Localisation Error (SLE):** A self-localisation algorithm, fed with different sensor data streams collected in the same environment, is used to localise the platform within the map. The precision of such localisation is evaluated by comparing it with the actual pose of the platform (ground truth).

#### 3.4.4 Experimental results

MVL in contrast to other approaches does not always yield a positive localisation response to an arbitrary environmental query. Hence an additional metrics applies, which we call “Self Localization coverage” (SLC) which we define as the percentage of query positions at which MVL did respond with a positive result. For our experimental setup we determined this value as 72% across all experiments conducted for the given waypoints. A qualitative indication on the SLC at arbitrary positions can be observed in the section of qualitative results.

**RPE:** Tab. 3.4.4 shows the performance of the predictions made with encoders and Visual Odometry without measure updates. The RPE measures the accumulated error from a waypoint to another, where the distance is 2 meters. It is possible to notice a better performance of the encoders. This is due to the fact that the scene in some locations in the scenario was not sufficiently rich for the Visual Odometry.

While the average error on orientation is low for both sensors, the Visual Odoemtry has a higher deviation.

We can see that the fusion of the two sensors compensate for Visual Odoemtry failures since the performance of the fusion is comparable to the one of the encoders in isolation.

**SLE:** Tab. 3.4.4 and Tab. 3.4.4 show the performance of the techniques that make use of measure updates, respectively for position error and orientation error.

As for the RPE, also in the SLE fusion with encoders and Visual Odometry has similar performance to the encoders (see the first two columns showing the techniques with RFID

Exp	WE				VO				WE + VO			
	Pos (cm)		Ori (°)		Pos (cm)		Ori (°)		Pos (cm)		Ori (°)	
	$\mu$ RPE	$\sigma$ RPE	$\mu$ RPE	$\sigma$ RPE	$\mu$ RPE	$\sigma$ RPE	$\mu$ RPE	$\sigma$ RPE	$\mu$ RPE	$\sigma$ RPE	$\mu$ RPE	$\sigma$ RPE
<b>A</b>												
LR/EM	3.5	1.8	-0.2	1.2	31.7	21.2	1.8	7.3	3.6	2.0	-0.2	1.1
LR/CR	3.7	1.1	-0.5	1.9	41.9	23.8	-1.5	13.0	3.6	1.2	-0.5	1.9
HR/CR	3.2	1.9	-0.4	1.1	29.6	15.9	-0.3	5.2	3.3	2.0	-0.4	1.1
<b>B</b>												
LR	3.9	1.9	-0.4	1.3	29.7	15.3	0.6	9.5	4.1	1.9	-0.4	1.3
HR	3.7	1.4	-0.3	1.4	55.5	111.9	11.1	33.3	4.2	2.0	-0.2	1.2
<b>C</b>												
LR/EM	9.0	6.7	-0.2	2.0	39.8	15.5	-0.1	14.2	9.1	7.2	-0.2	2.0
LR/CR	12.4	7.7	0.1	2.9	53.6	35.4	-1.5	22.0	12.3	8.5	0.1	2.9
HR/EM	11.0	7.7	0.1	1.3	40.0	23.7	-3.1	13.6	11.0	7.9	0.1	1.3
HR/CR	10.9	7.9	-0.0	2.1	38.3	10.8	1.3	10.9	10.7	8.4	-0.0	2.1
<b>D</b>												
LR	6.9	4.2	-0.1	2.0	50.2	24.6	-2.8	14.8	6.9	4.4	-0.1	2.0
HR	6.2	3.5	-0.1	1.9	45.0	25.1	-1.3	11.3	6.3	3.6	-0.1	1.9
<b>E</b>												
LR	8.8	7.6	0.0	1.8	50.4	43.0	0.5	11.2	8.8	7.7	0.0	1.8
HR	9.7	8.3	0.0	1.9	64.3	126.2	-5.8	21.6	9.7	8.3	0.0	1.9
<b>Tot</b>	<b>7.1</b>	<b>6.3</b>	<b>-0.1</b>	<b>1.7</b>	<b>44.7</b>	<b>56.2</b>	<b>-0.2</b>	<b>16.1</b>	<b>7.2</b>	<b>6.4</b>	<b>-0.1</b>	<b>1.7</b>

Table 3.3: Relative Pose Errors (RPE) of the incremental modalities

Exp	WE + RFID		WE + RFID + VO		MVL		WE + MVL		VO + MVL		TOT	
	$\mu$ SLE	$\sigma$ SLE	$\mu$ SLE	$\sigma$ SLE	$\mu$ SLE	$\sigma$ SLE	$\mu$ SLE	$\sigma$ SLE	$\mu$ SLE	$\sigma$ SLE	$\mu$ SLE	$\sigma$ SLE
<b>A</b>												
LR/EM	8.1	4.9	7.5	6.2	4.0	1.7	1.7	1.1	4.6	8.9	1.6	1.2
LR/CR	6.6	4.9	7.8	5.5	4.2	1.4	1.7	1.1	8.3	11.2	1.8	0.8
HR/CR	3.1	2.5	5.3	4.1	3.6	2.6	1.9	1.5	4.1	5.6	1.9	1.3
<b>B</b>												
LR	3.6	3.2	4.1	2.9	3.6	2.6	1.3	1.5	5.2	10.6	1.3	1.5
HR	4.1	2.5	3.7	4.0	3.7	1.3	2.0	1.1	7.7	15.1	1.9	1.1
<b>C</b>												
LR/EM	9.9	6.6	10.8	7.7	2.6	1.0	2.3	1.0	9.0	18.0	2.1	1.0
LR/CR	3.2	1.9	2.9	1.9	8.7	7.8	2.2	1.2	8.6	14.6	2.1	1.3
HR/EM	5.1	4.2	3.8	2.3	9.1	9.2	2.1	1.3	5.1	9.2	2.0	1.3
HR/CR	5.7	4.2	5.0	4.9	5.1	5.3	2.3	1.2	8.9	16.3	2.2	1.3
<b>D</b>												
LR	11.7	6.3	10.0	6.1	2.4	0.7	1.7	0.9	3.7	4.0	1.6	1.0
HR	7.1	5.0	6.8	4.8	3.3	3.5	1.8	2.0	2.7	4.1	1.8	2.0
<b>E</b>												
LR	7.9	6.9	7.3	6.8	6.4	5.7	3.4	3.1	6.2	6.8	3.3	3.1
HR	4.8	3.5	5.0	3.5	5.7	5.9	2.3	1.3	7.3	11.4	2.2	1.3
<b>Tot</b>	<b>6.2</b>	<b>2.7</b>	<b>6.2</b>	<b>2.5</b>	<b>4.8</b>	<b>4.8</b>	<b>2.0</b>	<b>0.5</b>	<b>6.3</b>	<b>2.2</b>	<b>2.0</b>	<b>0.5</b>

Table 3.4: Self Localisation Errors (SLE) - Position in cm

Exp	WE + RFID		WE + RFID + VO		MVL		WE + MVL		VO + MVL		TOT	
	$\mu$ SLE	$\sigma$ SLE	$\mu$ SLE	$\sigma$ SLE	$\mu$ SLE	$\sigma$ SLE	$\mu$ SLE	$\sigma$ SLE	$\mu$ SLE	$\sigma$ SLE	$\mu$ SLE	$\sigma$ SLE
<b>A</b>												
LR/EM	2.2	1.8	2.3	1.8	0.6	0.5	0.6	1.3	1.9	5.9	0.5	1.3
LR/CR	2.6	2.2	3.0	2.6	0.9	0.7	0.9	1.4	2.7	7.5	0.9	1.4
HR/CR	2.0	1.4	2.5	1.8	0.8	0.7	0.9	1.4	1.6	8.9	0.8	1.4
<b>B</b>												
LR	1.3	0.9	1.6	1.1	0.6	0.6	1.5	1.3	-1.6	5.8	1.4	1.3
HR	1.4	1.6	2.9	3.2	0.6	0.6	1.5	1.9	-3.8	7.9	1.4	1.9
<b>C</b>												
LR/EM	4.4	3.5	4.7	4.0	1.2	1.1	-0.1	5.6	-4.4	9.4	-0.1	5.6
LR/CR	8.9	4.8	9.2	5.0	2.4	1.4	-0.3	5.8	-3.1	6.1	-0.3	5.7
HR/EM	4.9	4.5	5.0	4.2	0.6	0.6	0.2	6.2	-2.3	6.8	0.2	6.2
HR/CR	4.2	3.7	4.4	4.0	1.2	1.4	0.8	7.0	-2.8	6.1	0.7	7.0
<b>D</b>												
LR	3.4	1.7	3.0	1.6	0.9	0.4	-0.2	2.3	3.6	22.1	-0.3	2.2
HR	3.8	2.2	3.4	2.2	0.8	0.6	-0.9	2.3	-5.6	7.3	-0.7	2.8
<b>E</b>												
LR	4.7	3.8	5.3	4.0	1.3	0.7	0.3	4.6	-0.4	8.4	0.3	4.6
HR	7.5	7.0	7.9	6.8	1.2	0.8	0.4	4.5	-4.5	9.8	0.3	4.8
<b>Tot</b>	<b>4.0</b>	<b>2.3</b>	<b>4.3</b>	<b>2.2</b>	<b>1.0</b>	<b>0.5</b>	<b>2.5</b>	<b>1.3</b>	<b>5.4</b>	<b>2.1</b>	<b>2.5</b>	<b>1.3</b>

Table 3.5: Self Localisation Errors (SLE) - Orientation in degree

updates).

The use of MVL increases the performance of the prediction with encoders. Adding to the encoders and MVL the use of Visual Odometry and RFID does not show any particular benefit. Moreover, MVL in isolation seems to maintain a more accurate estimation of the orientation. We need to point out that MVL is used at a full rate, which may not be possible in real applications since it requires an intense communication with the cloud infrastructure. In the next session we show the evolution of performance using different rates of MVL.

Visual Odoemtry can benefit from MVL too. This technique, due to its simplicity can be used for a configuration in which the user does not need a precise guidance service, but just directional indications.

**MVL rates:** We performed data analysis usign different rates of the MVL. This study targets the technique that uses encoders and MVL, the technique that uses Visual Odoemtry and MVL, and the total technique.

The experiment is divided in two subsets represented by the two possible frame rates utilized with the Kinect. The considered rates are: every frame, every 1, 10, 50, 100 seconds. We can see that the evolution of the performance is monotonic, with performance worsening with lower rates as expected.

The results for encoders and MVL for position and orientation is shown in Fig. 3.4 and Fig. 3.5. The results for Visual Odoemtry and MVL are shown in Fig. 3.6 and Fig. 3.7. Finally, the performance of the technique using all sensors is shown in Fig. 3.8 and Fig. 3.9.

**Qualitative Results:** In this section we present a selection of qualitative results in continuous operation mode, i.e. each modality processes the raw data at maximum data rate. For MVL we process each frame individually since we have no processing “memory” and thus are independent on the frame rate being processed but still want to show the stability of individual queries over time. All other modalities process at the frame rate at which they actually run on the mobile platform which results in potential frame data skipping to stay real-time. For qualitative comparison, we also determine a ground truth (GT) trajectory which is computed with a specifically tailored Kalman smoother. This technique is the combination of two steps [12]. First, we perform a forward recursion using encoders, MVL updates and viewpoints timestamps. This part is a normal Kalman filter without any knowledge of future data. Finally, a backward recursion (smoothing) determines the trajectory considering the totality of the system evolution.

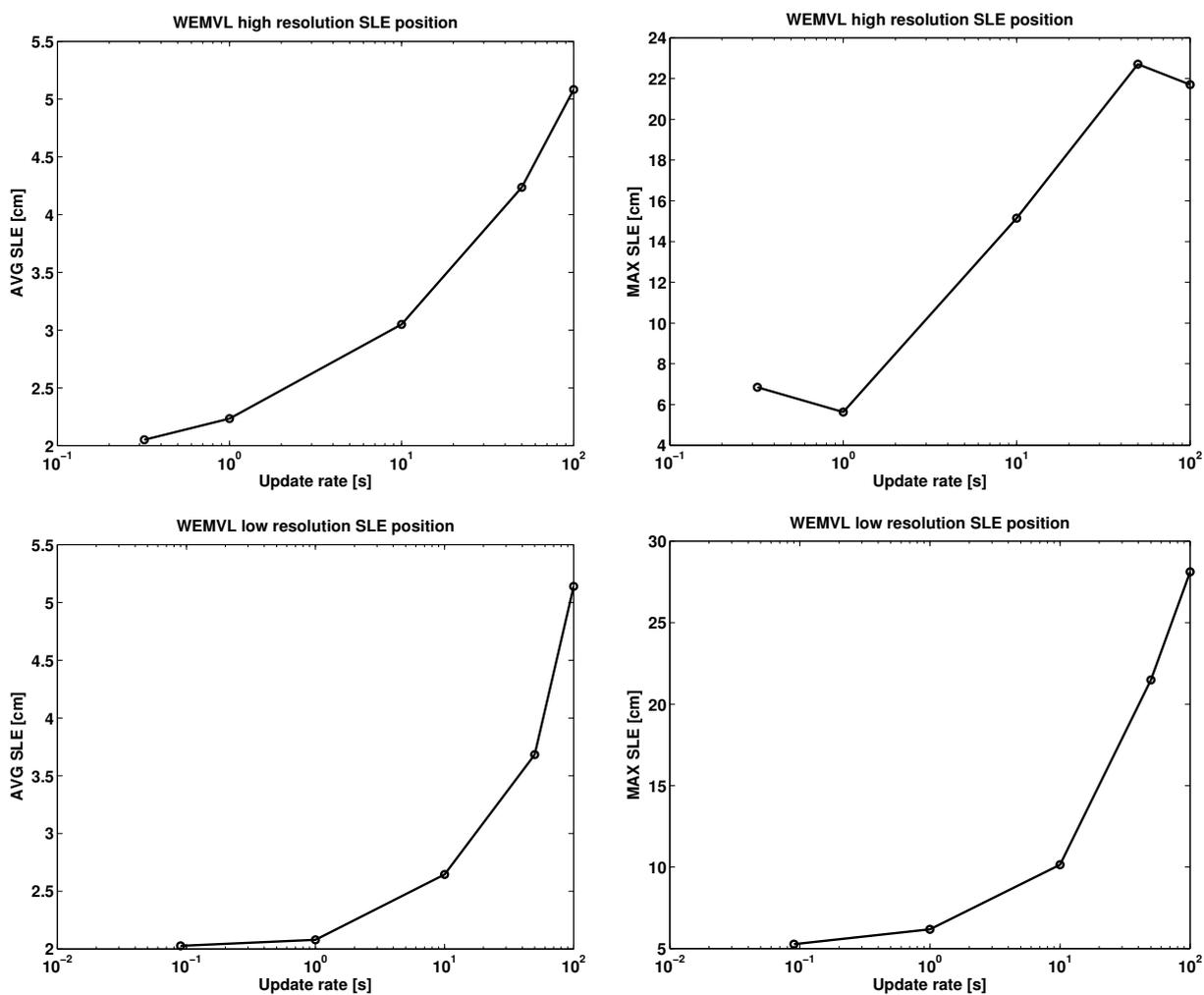


Figure 3.4: Average and maximum error on position: localisation system with encoders and MVL.

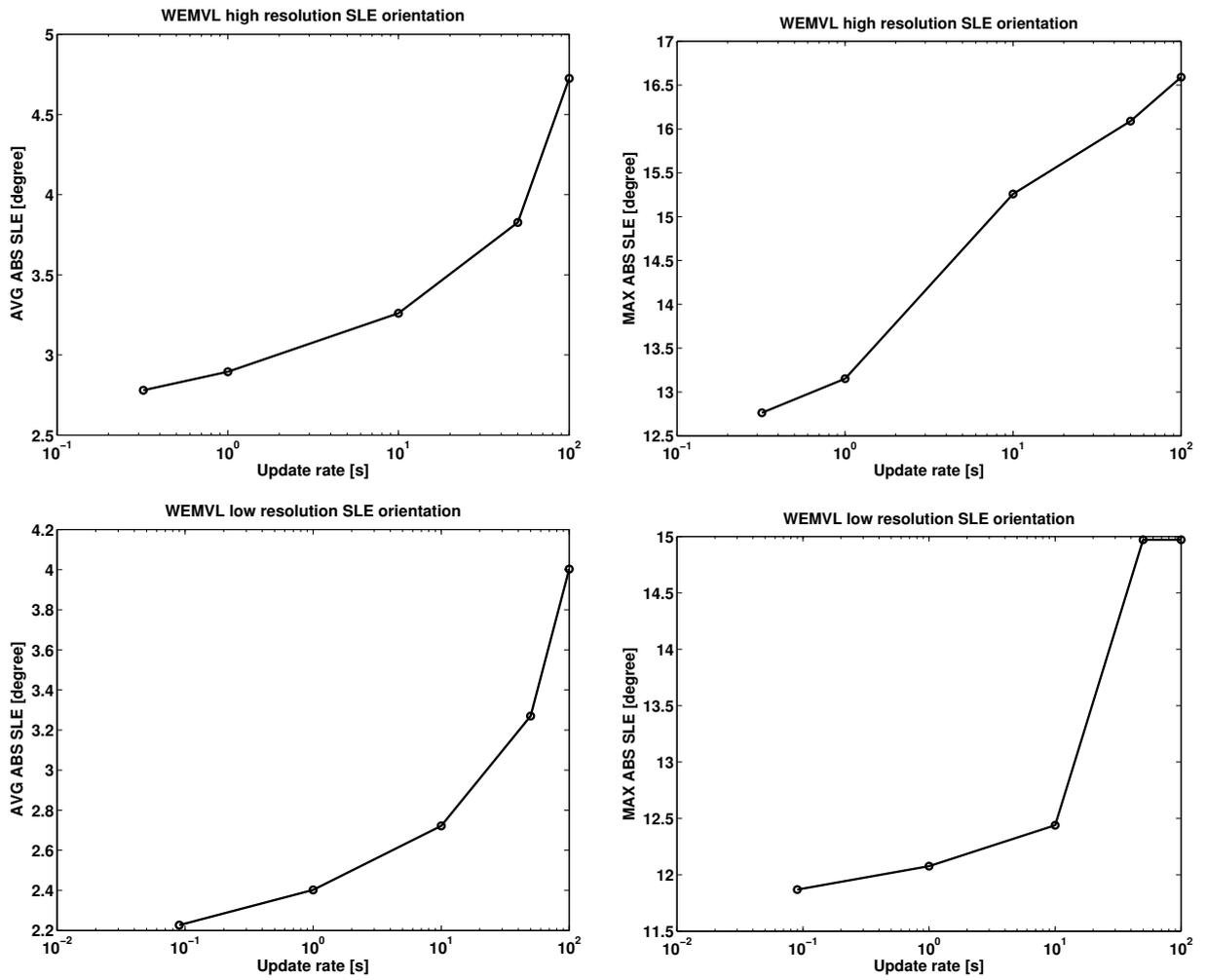


Figure 3.5: Average and maximum absolute error on orientation: localisation system with encoders and MVL.

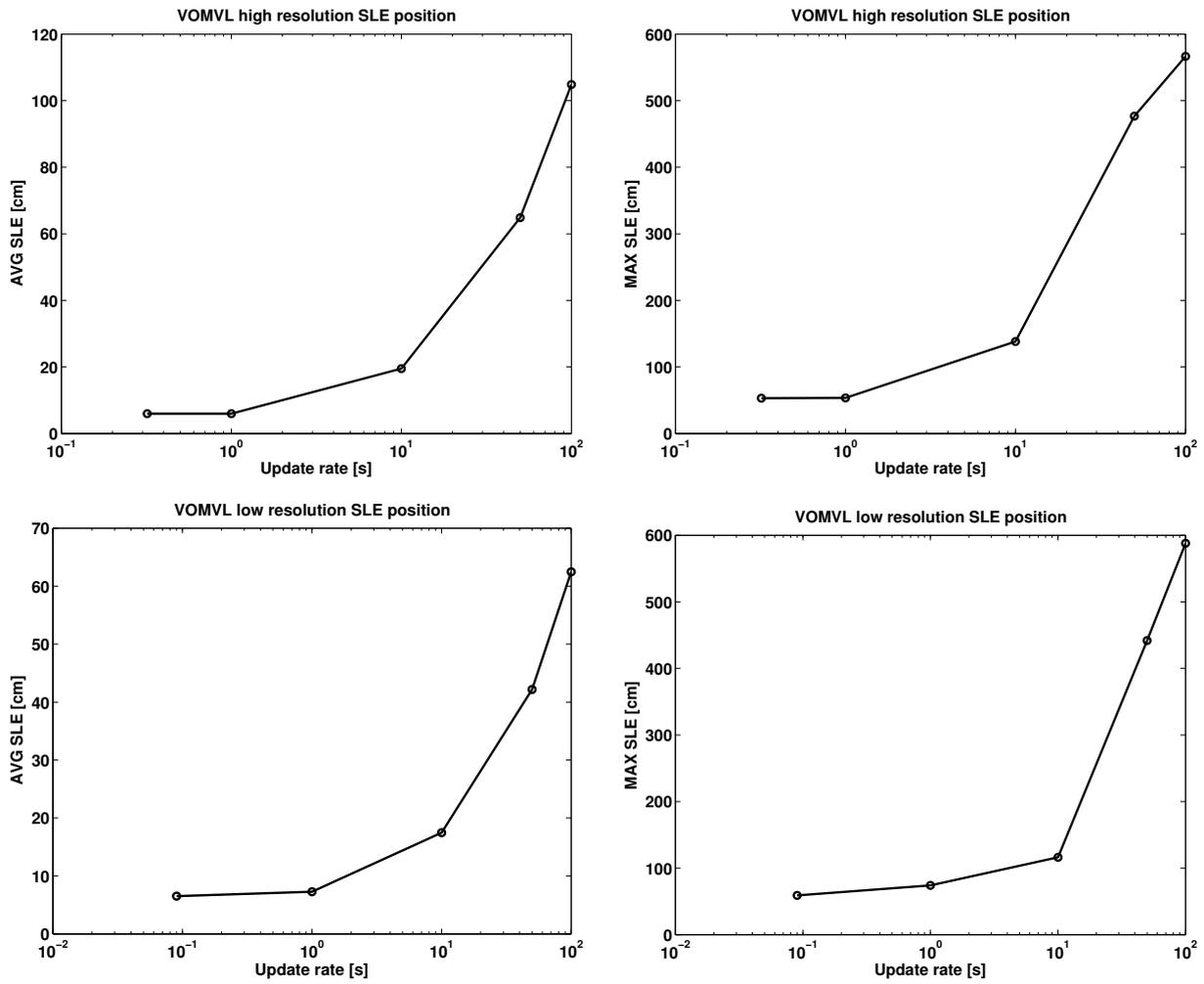


Figure 3.6: Average and maximum error on position: localisation system with Visual Odometry and MVL.

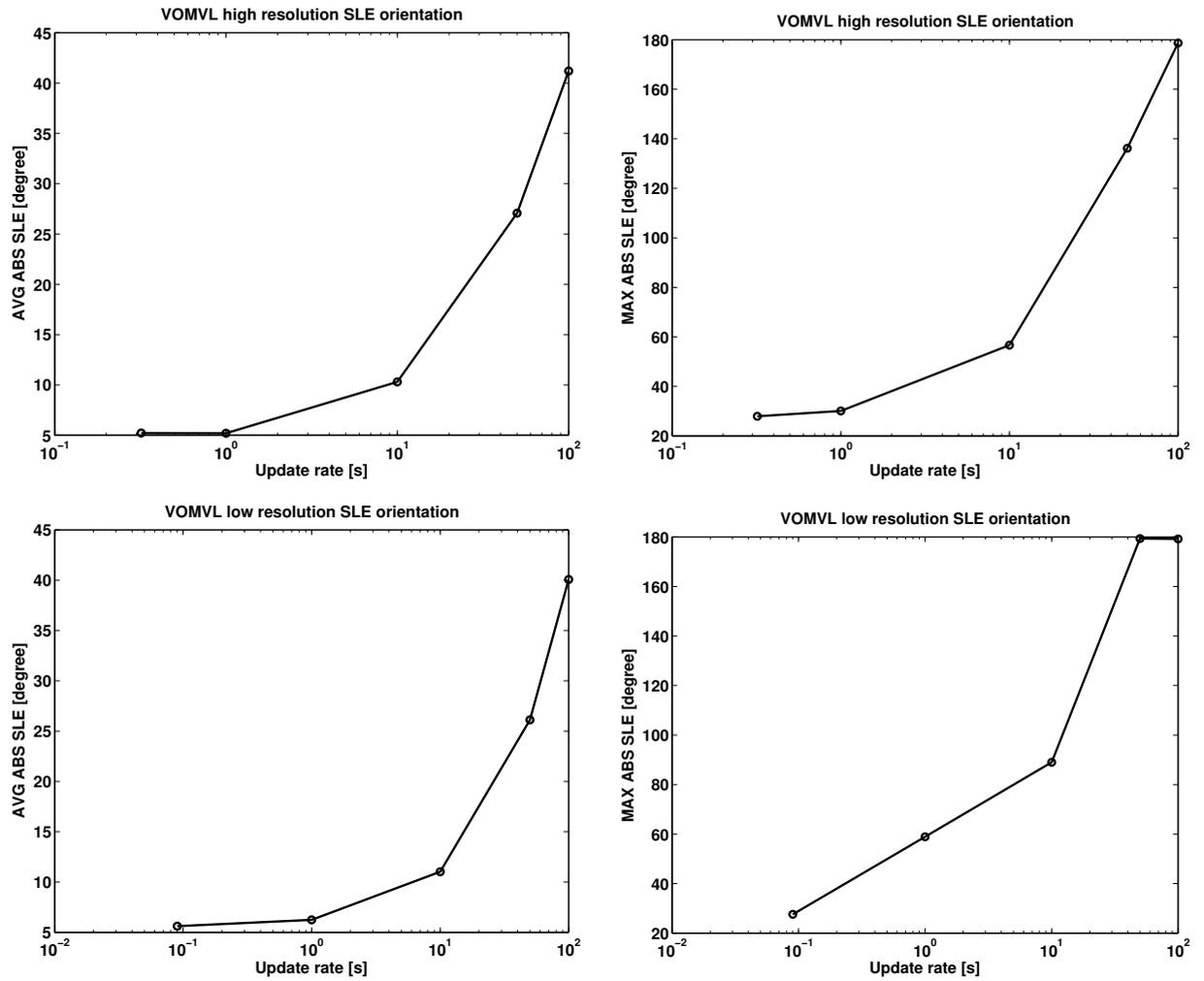


Figure 3.7: Average and maximum absolute error on orientation: localisation system with Visual odometry and MVL.

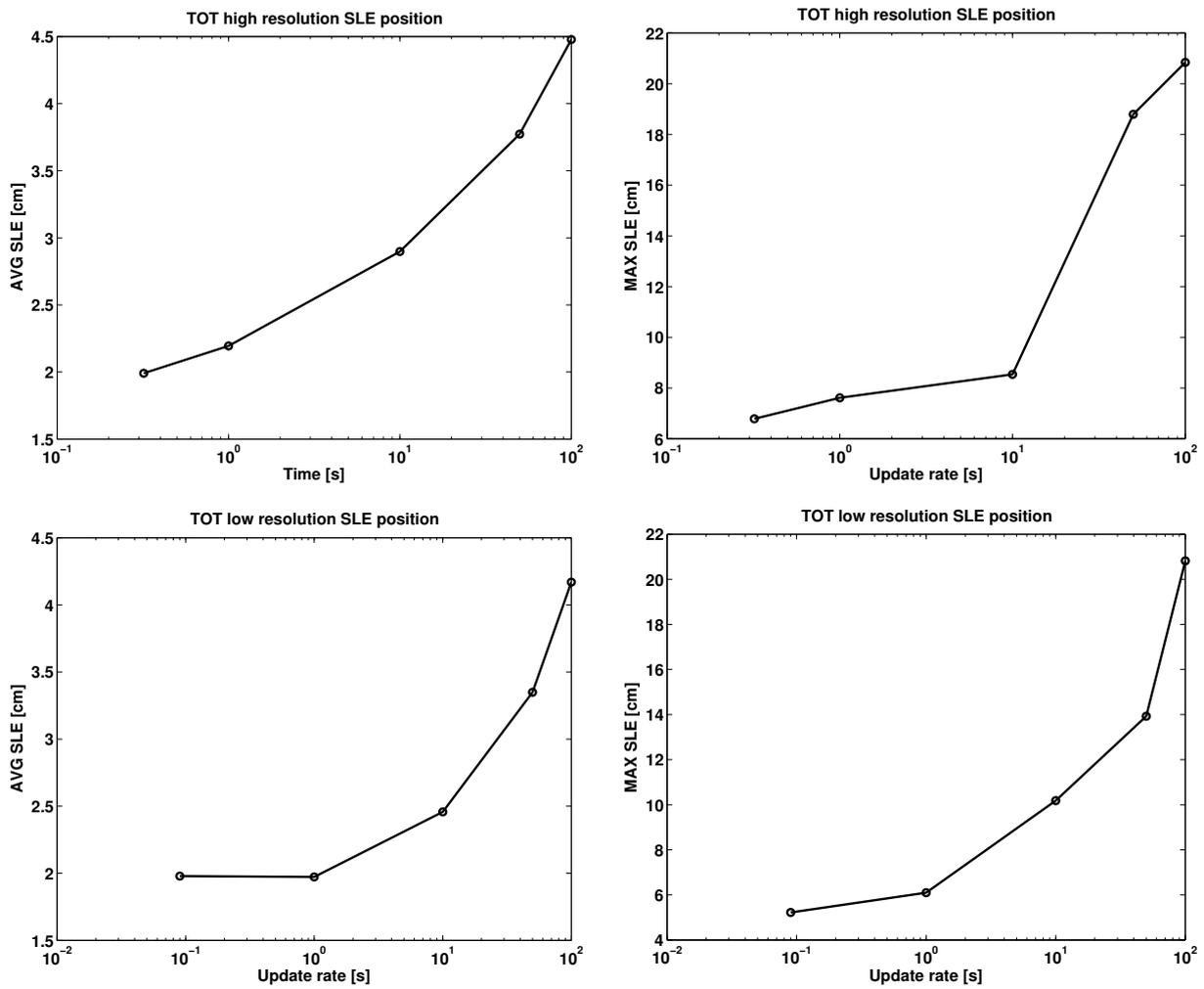


Figure 3.8: Average and maximum error on position: localisation system with all the components.

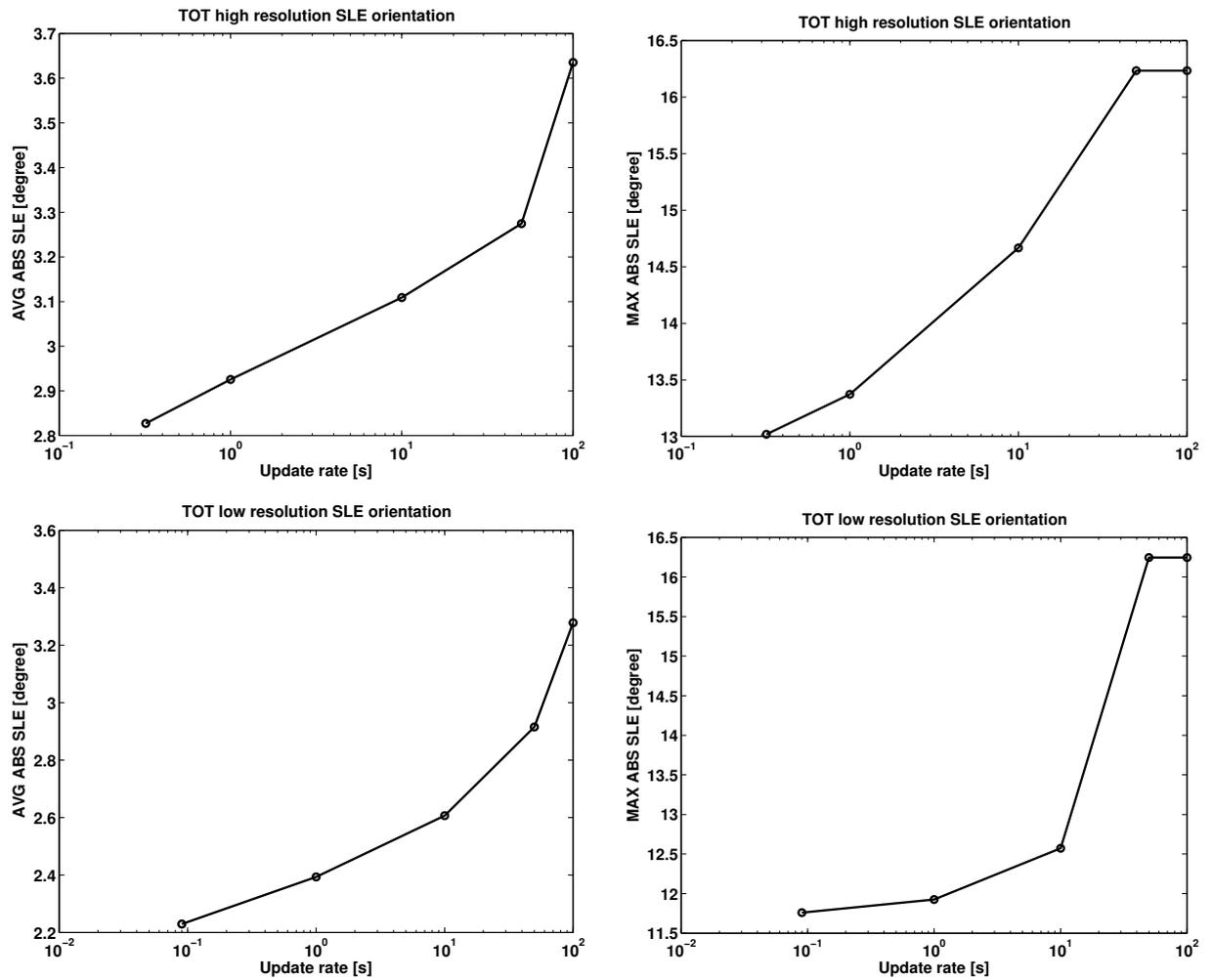


Figure 3.9: Average and maximum absolute error on orientation: localisation system with all the components.

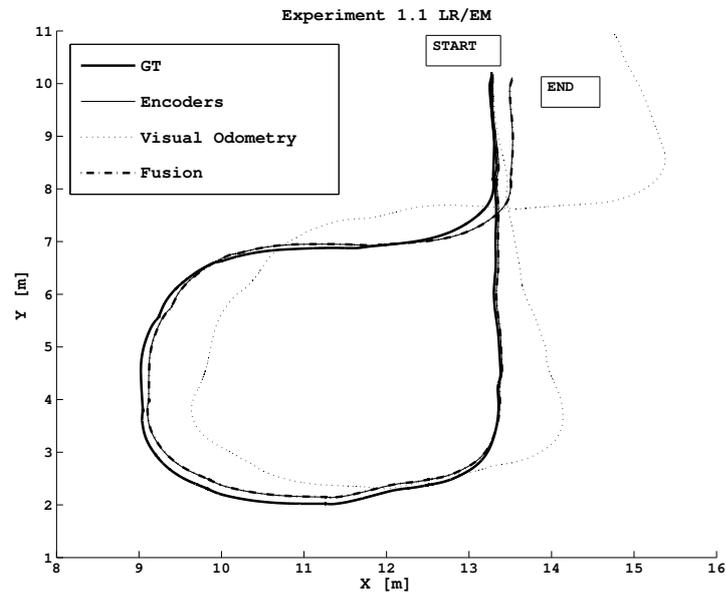


Figure 3.10: Trajectory reconstruction using only incremental sensors and fusion of them.

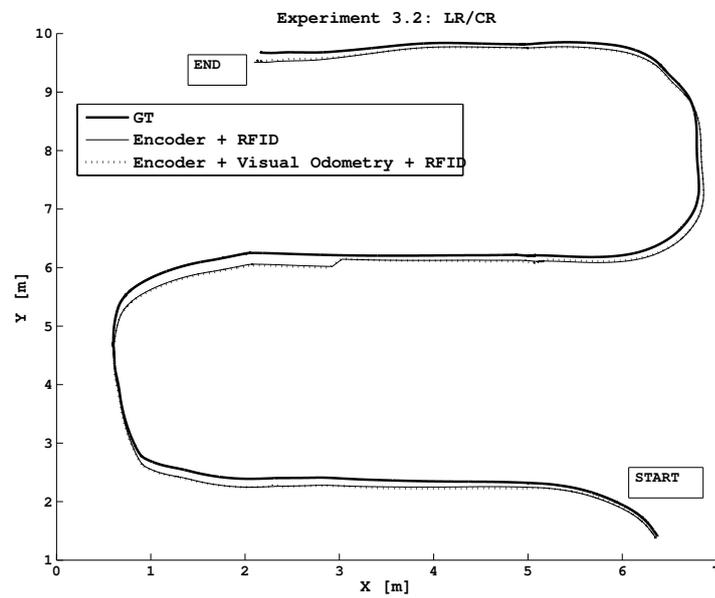


Figure 3.11: Trajectory reconstruction using encoders or fusion of incremental sensors as prediction and RFID for updates.

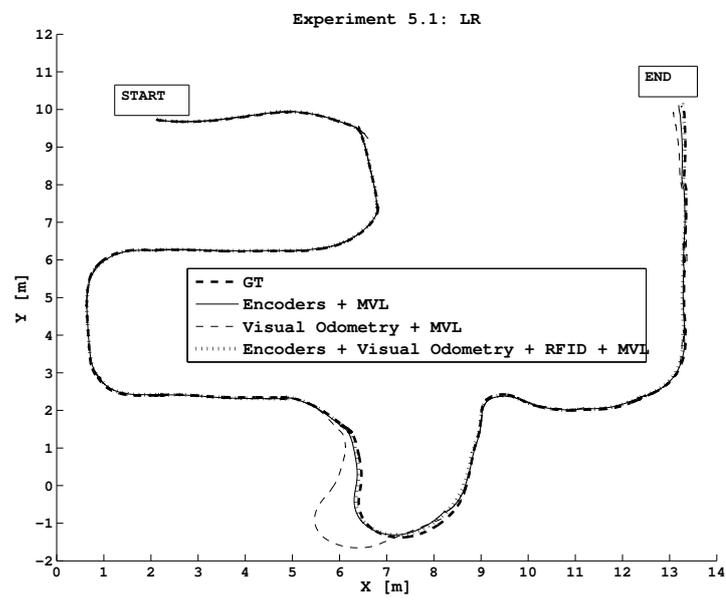


Figure 3.12: Trajectory reconstruction using incremental sensors updated with MVL, and the fusion prediction updated with both MVL and RFID.



# Chapter 4

## Guidance

As already stated, the *c-Walker* aims at providing guidance assistance to people with different disabilities or difficulties. Not every person needs the same type of assistance: some may need basic indications, some may need full support. For this reason, it is important to understand the kind of performance and user response we can obtain from different actuators and guidance algorithms. What follows is a comparison of three guidance mechanisms (acoustic, haptic, and mechanical) utilized in four different guidance algorithms.

### 4.1 Guidance mechanisms

In this section, we describe the three main mechanisms that can be used as “actuators” to suggest or to force changes in the direction of motion.

#### 4.1.1 Bracelets

*This software component has been developed by University of Siena.*

Haptic guidance is implemented through a tactile stimulation that takes the form of a vibration. A device able to transmit haptic signals through vibrations is said “vibrotactile”.

Vibration is best transmitted on hairy skin because of skin thickness and nerve depth, and it is best detected in bony areas. Wrists and spine are generally the preferred choice for detecting vibrations, with arms immediately following. Our application is particularly challenging for two reasons: I. the person receiving the signal is an older adult, II. the signal is transmitted while the user moves. Movement is known to affect adversely the detection rate and the response time of lower body sites ([39]). As regards the perception of tactile stimuli by older adults, [30] present studies on the effects of aging in the sense of

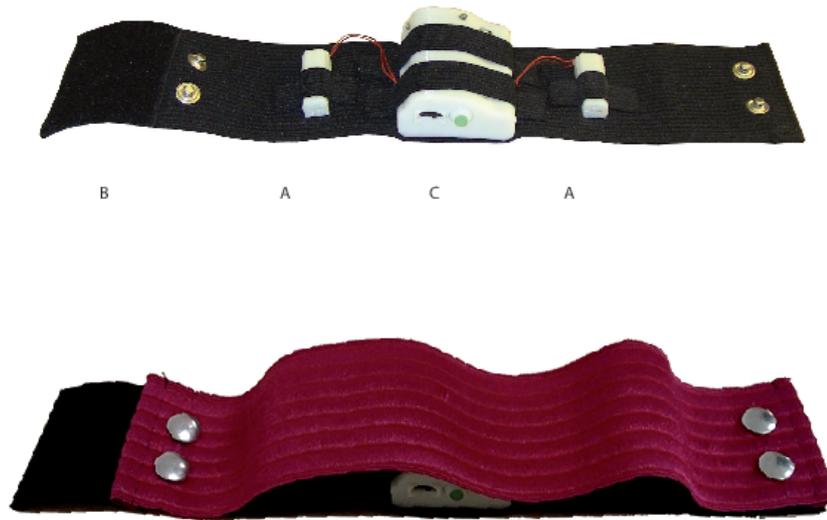


Figure 4.1: The vibrotactile bracelet equipped with two vibrating motors (A) attached to an elastic wristband (B). The Li-Ion battery and the Arduino board are in (C).

touch, which revealed that detection thresholds for several vibration intensities are higher in older subjects in the age class 65+.

Bearing in mind these facts, we designed a wearable haptic bracelet in which two cylindrical vibro-motors generate vibratory signals to warn the user (Fig. 4.1). The subject wears one vibrotactile bracelet on each arm in order to maximize the stimuli separation while keeping the discrimination process as intuitive as possible. In particular, vibration of the left wristband suggests the participant to turn left, and vice versa. On each bracelet the distance between the two motors is about 80 mm. In two-point discrimination, the minimal distance between two stimuli to be differentiated is about 35 mm on the fore-arms and there is no evidence for differences among the left and right sides of the body, according to [84]. In order to reduce the *aftereffect* problem typical of continuous stimuli and to preserve users' ability to localize vibration, we selected a pulsed vibrational signal with frequency 280 Hz and amplitude of 0.6 g, instead of a continuous one. In particular, when a bracelet is engaged its two vibrating motors alternatively vibrates for 0.2 s. The choice of using two vibrating motors instead of one was the effect of a pilot study in which a group of older adults tested both options and declared their preference for the choice of two motors. The choice of frequency and amplitude of the vibrations was another outcome of this study (see [74]).

From a technical point of view, the communication is realized with an RN42 Bluetooth module connected to an Arduino mini pro 3.3 V with a baud rate of 9600. From the technological point of view, two Precision Microdrives 303-100 Pico Vibe 3.2 mm vibration motors were placed into two fabric pockets on the external surface of the bracelet (the width of the wristband is about 60 mm), with shafts aligned with the elbow bone. The motors have a vibration frequency range of 100 Hz-280 Hz (the maximal sensitivity is achieved around 200 Hz-300 Hz), lag time of 21 ms, rise time of 32 ms and stop time of 35 ms. The application of a different voltage allows us to control frequency and amplitude of the vibration, to determine the best choice. The bracelet weighs about 80g, guarantees about 4 hours of battery life with one motor always turned on and can be easily worn underneath the shirt.

### 4.1.2 Audio interface

The acoustic interface communicates to the user the direction to take by transmitting synthetic signals through a headphone. For instance, when the system aims to suggest a left turn to the user, it reproduces a sound that is perceived by the user as coming from a point on her left aligned with the direction she is supposed to take. This is possible thanks to the application of the binaural theory.

The software module that generates this sound is called **Audio Slave** and it receives from a master the spatial coordinates  $(S_x, S_y)$  of the point that is required to be the source of the sound. The audio slave converts the cartesian coordinates into a pair  $(r, \theta)$  of relative polar coordinates, in which  $r$  represent the distance between the virtual sound source and the centre of the listener's head, and  $\theta$  represent the azimuthal angle. The pair  $(r, \theta)$  univocally identifies the position of the sound source on the horizontal plane.  $\theta$  takes on the value 0 when the source is in front of the user, positive angles identify positions on the right hand side, and negative values of  $\theta$  identify positions on the left of the listener. The guidance signal is a white noise with duration 50 ms, which is repeated every 150 ms. The binaural processing algorithm has been used to implement two different versions of the guidance interface:

- Left/Right Guidance;
- Binaural Guidance.

Using the Left/Right Guidance Interface, the system reproduces only virtual sources placed at  $\theta = 90^\circ$  or at  $\theta = -90^\circ$  to suggest a right turn or a left turn, in the same way as the haptic interface. With the Binaural Guidance Interface, a virtual sound source is allowed to be in any position. The resulting suggestion is not merely for a turn, but

it specifies a finer grained information on the exact direction. In this case, to ensure the correct displacement of the virtual sound relative to the user head orientation, an Inertial Measurement Unit (IMU) monitors the listener's head position with respect to the *c-Walker*.

Both the interface implementations are based on the same sound rendering engine which is based on the physics of sound waves. Each of the sound samples is delayed, and attenuated according to the principles of sound wave propagation. The binaural effect is obtained by proper filters that reproduce the presence of the listener's head and consider the ears displacement. However, the guidance interface is meant to generate recommendations on the direction to follow; therefore, stimuli have been processed without reverberation. As a consequence, users will perceive the sounds as intracranial, since the absence of reverb makes it difficult to externalize virtual sound stimuli. In both the audio interface implementations, the sound signals at the two channels are delayed, and attenuated according to the relative position of the virtual sound source and the listener's ears in the virtual space.

### 4.1.3 Mechanical Steering

The mechanical system based on steering uses the front caster wheels to suggest the user which direction to follow. The positioning of the wheels causes the *c-Walker* to perform a smooth turn manoeuvre without any particular intervention from the user and therefore is considered as an active guidance. That is, the user provides only the necessary energy to push the vehicle forward.

More in depth, the *c-Walker* is endowed with two caster wheels in front of the device, which are connected to a swivel that enables them to move freely around their axis. Taking advantage of this feature, we applied stepper motors to the joints to change the direction of the wheels by a specified amount. The presence of non-idealities (e.g., friction and slippage of the gears) can possibly introduce a deviation between the desired rotation angle and the actual one. Therefore, we need a position control scheme operating with real-time measurements of the current angular position of the wheels. Such measurements are collected by an encoder that is mounted on the same joint as the stepper motor. The connection between wheel and motor is through a gear system such that a complete turn of the wheel is associated with 4 turns of the motor. Every complete turn of the motor is 400 steps. The stepper motor and the encoder are controlled by a small computing node that is interfaced to the rest of the system through a CAN bus. The motor, together with the absolute encoder and the relative CAN bus node, is visible in Fig. 2.2.

With a fixed periodicity, the node samples the encoder and broadcasts the sensor reading through the bus. The node can also receive a CAN message coming from other

Message structure	Broadcast message	Actuation message
<b>ID</b>	0x180 + NodeID	0x200 + NodeID
<b>len</b>	8	8
<b>D0</b>	ABS LSB	DEG LSB
<b>D1</b>	ABS MSB	DEG MSB
<b>D2</b>	HOLD STATE	SPEED LSB
<b>D3</b>	0x00	SPEED MSB
<b>D4</b>	MOT POS LSB	DIR
<b>D5</b>	...	HOLD
<b>D6</b>	...	0x00
<b>D7</b>	MOT POS MSB	SET ABS OFFSET

Table 4.1: The structure of the periodic CAN messages used in the communication with the front wheels.

computing devices that requires a rotation of the wheels specifying the number of degree of rotation and the angular velocity (deg/s). The values are automatically converted in steps and used in a PID control loop that moves the wheel to the specified angular position. A different message is associated with the *hold state*, meaning that the motor remains in the current position. A special CAN packet specifies the amount of current used in stepper motor, with high values corresponding to larger torques (both during motion and in the hold state) and to a stiffer behaviour. Small values of the current are associated with softer actions that the user can easily override forcing her way through. Tab 4.1 describes the structure of the CAN messages.

## 4.2 Guidance algorithms

The guidance algorithms relies on an accurate estimate of the position of the *c-Walker* with respect to the planned path. Since the latter is generated internally by a module of the *c-Walker* (see [17]), only the knowledge on the position  $Q = [x \ y]^T$  and of the orientation  $\theta$  expressed in some known reference frame is needed. This problem, known in the literature as *localisation problem*, is solved in the *c-Walker* using the solutions proposed in [58, 57].

With this information it is possible to determine the Frenet-Serret point  $F_a$ , that is a point on the path representing the intersection between the projection of the vehicle and a segment that is perpendicular to it and tangent to the path, as in Fig. 4.2 (a). We define as  $y_d$  and  $\theta_d$  respectively the distance along the projection of the vehicle to

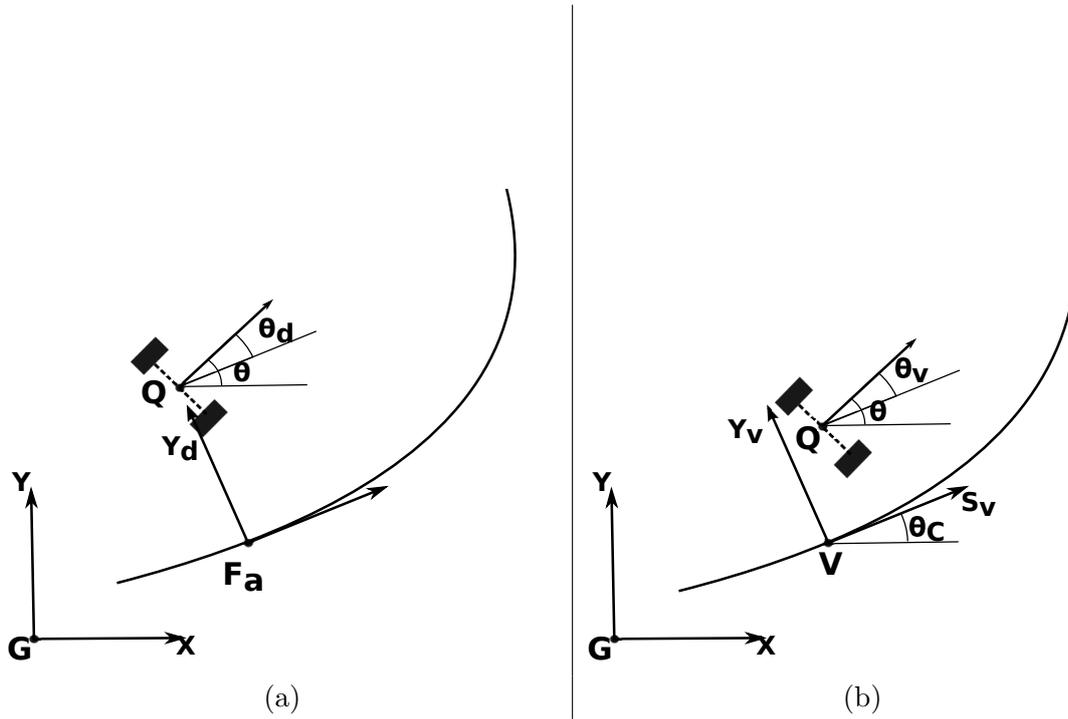


Figure 4.2: Graphic representation of the localisation of the *c-Walker* with respect to the path: (a) Frenet-Serret reference, (b) Virtual vehicle reference. In the Frenet-Serret reference frame, the vehicle always lies on the  $y_d$  axis.

$F_a$  and the difference between the orientation of the *c-Walker* and the orientation of the tangent to the path in the projection point. All the proposed guidance algorithms use this information to compute the specific “actuation”.

We observe that the objective of the guidance algorithms is not the perfect path following of the planned trajectory. In fact, such an objective would be very restrictive for the user and perceived as too authoritative and intrusive. In order to give the user the feeling of being in control of the platform, she is allowed an error (in both, position and orientation) throughout the execution of the path that is kept lower than a desired performance threshold. Therefore, the path can be considered as the centre line of a virtual corridor in which the user can move freely.

#### 4.2.1 Haptic and Acoustic algorithms

The haptic and acoustic guidance algorithms generate a quantised control action, which can be described with an alphabet of three control symbols: a) turn right; b) turn left; c) go straight. This is a good compromise between accuracy and cognitive load for the interpretation of signals.

The symbol to be suggested to the user is determined by the desired turning towards the path. A straightforward way to compute such a quantity is to determine the angular velocity an autonomous robotic unicycle-like vehicle would follow in order to solve the path following problem. To this end, we have designed a very simple control Lyapunov function which ensures a controlled solution to the path following in the case of straight lines acting only on the vehicle angular velocity and irrespective of the forward velocity of the vehicle. Such a controller works also for curved paths if we are only interested on the sign of the desired angular velocity.

To see this, consider the kinematic model of the unicycle (which is an accurate kinematic model of the *c-Walker*)

$$\begin{aligned}\dot{x}_d &= \cos(\theta_d)v, \\ \dot{y}_d &= \sin(\theta_d)v, \\ \dot{\theta}_d &= \omega,\end{aligned}\tag{4.1}$$

where  $v \neq 0$  is the forward velocity and  $\omega$  its angular velocity.  $y_d$  and  $\theta_d$  are the quantities defined in the previous section, while  $x_d$  is the longitudinal coordinate of the vehicle that, in the Frenet-Serret reference frame is identically zero by definition. It has to be noted that  $(x_d, y_d)$  are then the cartesian coordinates, in the Frenet-Serret reference frame, of the midpoint of the rear wheels axle. In light of model (4.1) and remembering that  $x_d$  does not play any role for path following, we can set up the following control Lyapunov function

$$V_1 = \frac{k_y y_d^2 + k_\theta \theta_d^2}{2},\tag{4.2}$$

which is positive definite in the space of interest, i.e.,  $(y_d, \theta_d)$ , and has as time derivative

$$\dot{V}_1 = k_y y_d \sin(\theta) v + k_\theta \theta \omega,\tag{4.3}$$

where  $k_y > 0$  and  $k_\theta > 0$  are tuning constants. Imposing  $\omega$  equals to the following desired angular velocity

$$\omega_d = -q_\theta \theta_d - \frac{k_y}{k_\theta} y_d \frac{\sin(\theta_d)}{\theta_d} v,\tag{4.4}$$

with  $q_\theta > 0$  additional degree of freedom, the time derivative in (4.3) is negative semidefinite; using La Salle and Krasowskii principles, asymptotic stability of the equilibrium point  $(y_d, \theta_d) = (0, 0)$  can therefore be established, with the *c-Walker* steadily moving toward the path.

As a consequence, the sign of  $\omega$  rules the direction of switching: a) if  $\omega > t_\omega$  then the user has to turn left; b) if  $\omega < -t_\omega$  then the user has to turn right; c) if  $\omega \in [-t_\omega, t_\omega]$  then the user has to go straight.  $t_\omega$  is a design threshold used to be traded between the user comfort and the authority of the control action.

In order to implement the idea of the virtual corridor around the path and to increase the user comfort, the actuation takes place only when  $V_1$  in (4.2) is greater than a certain  $V_1^{max}$ , which is defined as in (4.2) when  $y_d = y_h$  is half the width of the corridor, and  $\theta_d = \theta_h$  defines half of the amplitude of a cone centered on the corridor orientation in which the *c-Walker* heading is allowed. For the haptic, acoustic algorithms the parameters that define the corridor are the same, that are  $y_h = 0.3$  m and  $\theta_h = 0.52$  rad. Similarly, the constants  $q_\theta$ ,  $k_y$  and  $k_\theta$  are fixed to the same values for both haptic guidance and acoustic guidance. However, they are changing according to the *c-Walker* actual position: when the position is outside the corridor,  $k_y = 1$  and  $k_\theta = 0.1$ , so that the controller is more active to steer the vehicle inside the corridor; when, instead, the *c-Walker* is within the corridor boundaries,  $k_y = 0.1$  and  $k_\theta = 1$  in order to highly enforce the current orientation tangent to the path. The position of the *c-Walker* inside the corridor is determined by simply checking if  $y_d \leq y_h$ .

Finally, to take into account the corridor, the  $\omega = \alpha\omega_d$ , where  $\alpha$  is a time varying parameter related to the corridor, i.e.,

$$\alpha = \min\left(1, \frac{V_1}{V_1^{max}}\right). \quad (4.5)$$

The turning rule related to the sign of  $\omega$  is then applied as previously described.

### Acoustic source computation

For the acoustic guidance system, the sound source position has to be properly identified. To this end, let us define the circle centered in the vehicle position  $Q$  and having radius  $ds$  ( $ds = 1.2m$  in the experiments). Let us further define  $dp$  as the segment joining the origin of the Frenet-Serret reference frame  $F_a$  with the intersection point  $P$  between the circle and the tangent to the circle in the origin of  $F_a$ . (see Fig. 4.3 for reference). If multiple solutions exist, the one being in the forward direction of the walker is considered. If only a solution exists, i.e.,  $y_d = |ds|$ , then  $P$  coincides with the origin of  $F_a$ . Finally, no solution exists if  $y_d \geq |ds|$ , therefore  $P$  lies on the segment that connects  $Q$  to  $F_a$ .

We define  $S$  as the point closer to  $P$  and lying on the path. If the *c-Walker* is close to a straight component of the path or at a distance greater than  $ds$ ,  $S = P$ . Otherwise,  $S$  is computed as the projection of  $P$  on the path.  $S$  is the desired sound source with respect to a fixed reference frame, which has to be transformed in the *c-Walker* reference coordinate systems by

$$S_{cw} = \begin{bmatrix} s_{x_{cw}} \\ s_{y_{cw}} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} (S - Q).$$

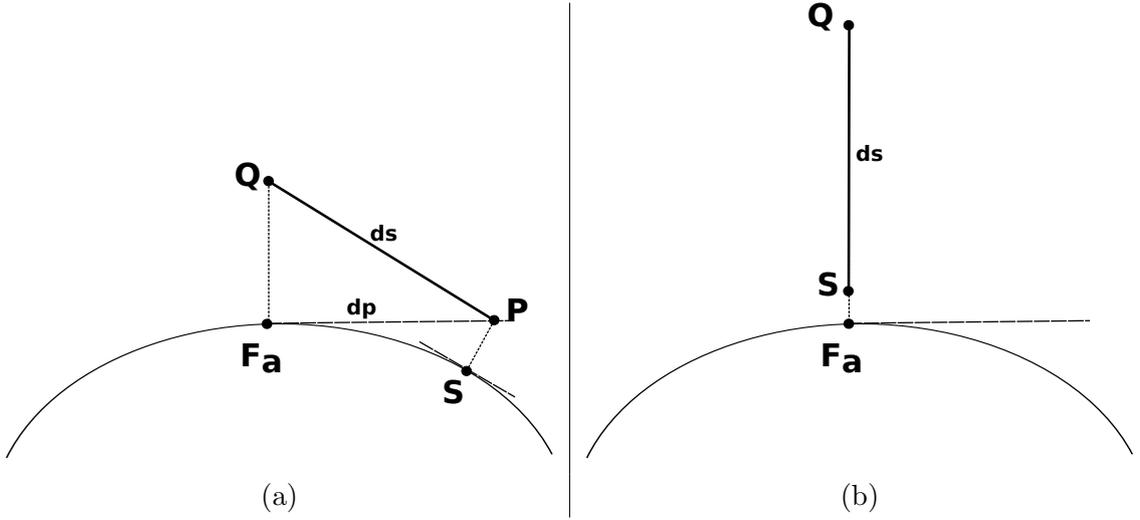


Figure 4.3: Graphic representation of the acoustic signal source: (a) *c-Walker* close to the path and sound projected ahead, (b) *c-Walker* too distant from the path and sound projected toward the Frenet-Serret point.

With the choice just illustrated, if the distance from the path is greater than  $y_d \geq |ds|$ , the target is pushed to the planned path following the shortest possible route.

### Actuation

**Haptic:** The bracelets are actuated according to the direction to follow. There are two choices of actuation: the first considers the value of  $\omega$  as discussed above, while the second considers the value of  $s_{y_{cw}}$  as computed in Sec. 4.2.1. In both cases, the sign determines the direction of turning.

**Left/Right** Three cones, having the vertices in  $Q$ , are defined:  $\mathcal{L}$ ,  $\mathcal{R}$  and  $\mathcal{S}$ . The cones divide the semicircle in front of the vehicle and with center in  $Q$  in three equal sectors. If  $S_{cw} \in \mathcal{L}$ , the user has to turn left; if  $S_{cw} \in \mathcal{R}$ , the user has to turn right; if  $S_{cw} \in \mathcal{S}$ , the user has to go straight. Positions behind the user are transformed in turn left or right depending on the position of  $S_{cw}$ . Using this taxonomy and the value of  $\alpha$  in (4.5), the slave application determines whether the sound has to be played or not and from which position.

**Binaural** The binaural algorithm fully exploits the reference coordinates  $S_{cw}$  using a finer granularity of positions than the Left/Right acoustic guidance. The number of cones is now equal to 7, with three equally spaced cones on the right and on the left the forward direction of the trolley. Each cone has a characteristic angle  $\beta_i$ , that is the one that equally splits the cone from the cart perspective. The described mechanism has the role of discretising the possible sound directions, since the human auditory system does not

have the sensibility to distinguish a finer partition. Again, positions behind the user are treated as in the Left/Right approach in order to avoid front/back confusion that commonly affects binaural sound recognition [6]. As a result of this quantisation, the new position of the sound source is  $S_s$ . By defining with  $\theta_i$  the user's head orientation measured with the IMU placed on top of the headphone, the final sound source  $S_p$  is computed as

$$S_p = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\sin(\theta_i) & \cos(\theta_i) \end{bmatrix} S_s.$$

#### 4.2.2 Mechanical system: steering

The rationale of the steering wheels controller is in the nature of the kinematic model. With respect to the model adopted in (4.1), which represents a unicycle-like vehicle model with differential drive on the back wheels, controlling the *c-Walker* using the steering wheels implies a different dynamic for the orientation rate, that becomes

$$\dot{\theta}_d = \frac{\tan(\phi)}{L} v, \quad (4.6)$$

where  $\phi$  is the steering angle. Since the steering angle is generated through the actuation of the front caster wheels, it is directly controlled in position by means of the stepper motors. Moreover,  $\phi \in [-\pi, \pi)$  and, hence, there is no theoretical limit on the value of  $\dot{\theta}_d$ . Nonetheless, there exists a singular point when  $v = 0$ , which can be ruled out because in such a case the path following does not have any sense even for the model in (4.1). This condition implies that acting on the steering wheels do not allow a turn on the spot.

As a consequence, it is possible to select any feasible path following controller conceived for the unicycle to solve the problem at hand. The controller adopted is the one proposed by [78], which is flexible (indeed, there are tuning parameters for the approaching angle to the path), and can be extended to include dynamic effects and uncertain parameters. The adopted controller, which is an extension of the one presented in [55], is based on the idea of a virtual target travelling on the path. Its adaptation to our context is discussed below.

Let  $V$  be the coordinates of the Virtual vehicle. The objective is to make the *c-Walker* perfectly track the Virtual vehicle. The position of the walker  $Q$  can be expressed in a global frame  $\mathbf{G}$  with  ${}^{\mathbf{G}}Q = |Q \ 0|^T = |x \ y \ 0|^T$ . Alternatively, the point can be expressed in the frame  $\mathbf{V}$ , which coincides with the point  $V$ , with  ${}^{\mathbf{V}}Q = |s_v \ y_v \ 0|^T$ .

Expressing with  $\theta_c$  the orientation in the global frame of the tangent to the path in  $V$ , with  $s$  the curvilinear abscissa of  $V$  along the path, and with  $c(s)$  the curvature of the path in that point, we have  $\omega_c = \dot{\theta}_c = c(s)\dot{s}$ . It is now possible to express the new

kinematic equations of the *c-Walker* w.r.t. the new frame  $V$  starting from (4.1). Indeed, starting from the rotation matrix relating  $\mathbf{G}$  to  $\mathbf{V}$ , i.e.,

$${}^{\mathbf{V}}R_{\mathbf{G}} = \begin{bmatrix} \cos(\theta_c) & \sin(\theta_c) & 0 \\ -\sin(\theta_c) & \cos(\theta_c) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

we first derive the velocity of the *c-Walker*:

$$\begin{aligned} \mathbf{G}\dot{Q} &= \mathbf{G}\dot{V} + \mathbf{G}R_{\mathbf{V}}{}^{\mathbf{V}}\dot{Q} + \mathbf{G}R_{\mathbf{V}}(\omega_c \times \mathbf{V}Q), \\ {}^{\mathbf{V}}R_{\mathbf{G}}\mathbf{G}\dot{Q} &= \mathbf{V}\dot{V} + \mathbf{V}\dot{Q} + (\omega_c \times \mathbf{V}Q), \\ {}^{\mathbf{V}}R_{\mathbf{G}} \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix} &= \begin{bmatrix} \dot{s} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{s}_v \\ \dot{y}_v \\ 0 \end{bmatrix} + \begin{bmatrix} -c(s)\dot{s}y_v \\ c(s)\dot{s}s_v \\ 0 \end{bmatrix}, \end{aligned}$$

and, then, we can express them as

$$\begin{bmatrix} \dot{s}_v \\ \dot{y}_v \\ \dot{\theta}_v \end{bmatrix} = \begin{bmatrix} -\dot{s}(1 - c(s)l) + v \cos(\theta_v) \\ -c(s)\dot{s}s_v + v \sin(\theta_v) \\ \omega - c(s)\dot{s} \end{bmatrix} \quad (4.7)$$

where  $\theta_v = \theta_d - \theta_c$ . Using Lyapunov techniques, it is possible to define the following set of control laws (see [78] for further details)

$$\begin{aligned} \dot{s} &= v \cos(\theta_v) + K_2 s_v, \\ \delta(y_v, v) &= -K_\delta \tanh(y_v v), \\ \dot{\theta}_v &= \dot{\delta}(y_v, v) - K_4 K_1 y_v v \frac{\sin(\theta_v) - \sin(\delta(y_v, v))}{\theta_v - \delta(y_v, v)} \\ &\quad - K_3 (\theta_v - \delta(y_v, v)), \end{aligned}$$

where  $\dot{s}$  represents the progression of the Virtual vehicle on the path,  $\delta(y_v, v)$  is the angle of approach of the vehicle with respect to the path (that can be tuned as necessary), while the  $K_i$  are tuning constants. With this choice,  $\dot{\theta}_v + c(s)\dot{s}$  is the angular velocity reference  $\dot{\theta}_v$  of the *c-Walker*, which can be generated by solving with respect to  $\phi$  the equation (4.6).

It has to be noted that  $\phi$  is the reference of the wheel if the half-car model is adopted. In order to transform the reference  $\phi$  to a reference for the left and right wheel, the constraint imposed by the Ackerman geometry are imposed. Finally, in order to implement the idea of the corridor previously presented, the actual steering angle imposed to the wheels considers the reference computed as described in this section as a reference  $\phi_d$ , which is

used in combination with the actual orientation  $\phi_a$ . More precisely, using the value of  $\alpha$  in (4.5), the commanded orientation of the steering wheel is given by  $\phi = \alpha\phi_d + (1 - \alpha)\phi_a$ . The value of the threshold in this case is increased to  $\theta_h = 1.62$  rad.

### 4.3 Implementation

The *c-Walker* hardware and software architecture has been designed for an easy integration of heterogeneous components, possibly developed by different teams (see Cha. 2).

The core modules of the architecture are shown in Fig. 4.4. The scheme is more specific than the one presented in Fig. 2.5 since it is focused on the guidance subsystem.

The **Planner** decides the plan to be followed based on: 1. the requests and the preferences of the user, 2. the map of the environment, 3. the presence of obstacles and crowded areas along the way. While the *c-Walker* is moving, it collects information from the environment and the planned path can be updated to avoid obstacles or safety risks.

The Localisation module integrates information from several sources (encoders, Inertial platform, cameras, RFID reader) to produce an updated information on the estimated position of the vehicle in the environment with a few centimetres position. A mechatronic subsystem encapsulates all the modules that are used to read and process sensor data from the encoders and from the inertial sensors. Additionally, the mechatronic module contains all the logic required to send command to the actuators (e.g., the motors on the caster wheels). The mechatronic system is reached through a CAN bus.

These core modules can be interconnected with other modules to implement the different guidance solutions discussed above, as shown in Fig. 4.4.

It was described in Cha. 2 that the different components are interconnected using a publisher/subscriber middleware, whereby a component can publish messages that are broadcast through all the different level of networking (CAN bus for mechatronic components, ethernet for high level sensors and computing nodes) in the *c-Walker*. This is a key enabler for the adoption of a truly component-based paradigm, in which the different guidance systems can be obtained by simply tuning on some of the modules and allowing them to publish messages or subscribe to messages. Three different configurations are schematically shown in Fig. 4.5.

Fig. 4.5 shows how the three guidance systems we present in this chapter interact with the *c-Walker*.

The scheme on the top of the figure refers to mechanical guidance. The Planner periodically publishes updated plans (i.e., the coordinates of the next points to reach). This information is subscribed to by a path follower that implements the algorithm presented in Section 4.2.2. This component decides a direction for the wheel that is transmitted

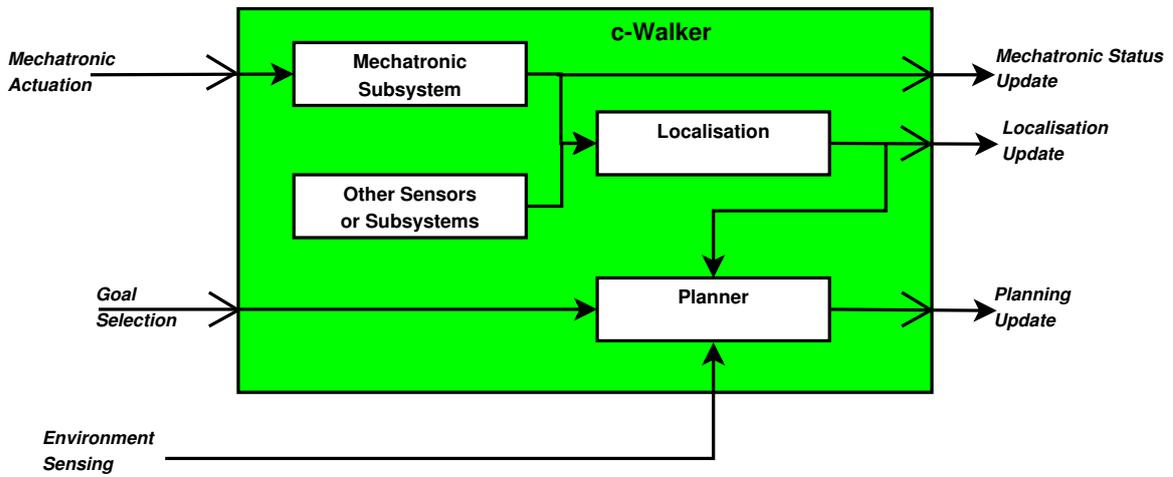


Figure 4.4: The block-scheme of the *c-Walker* architecture with its core components: mechatronic subsystem, localisation subsystem and planner.

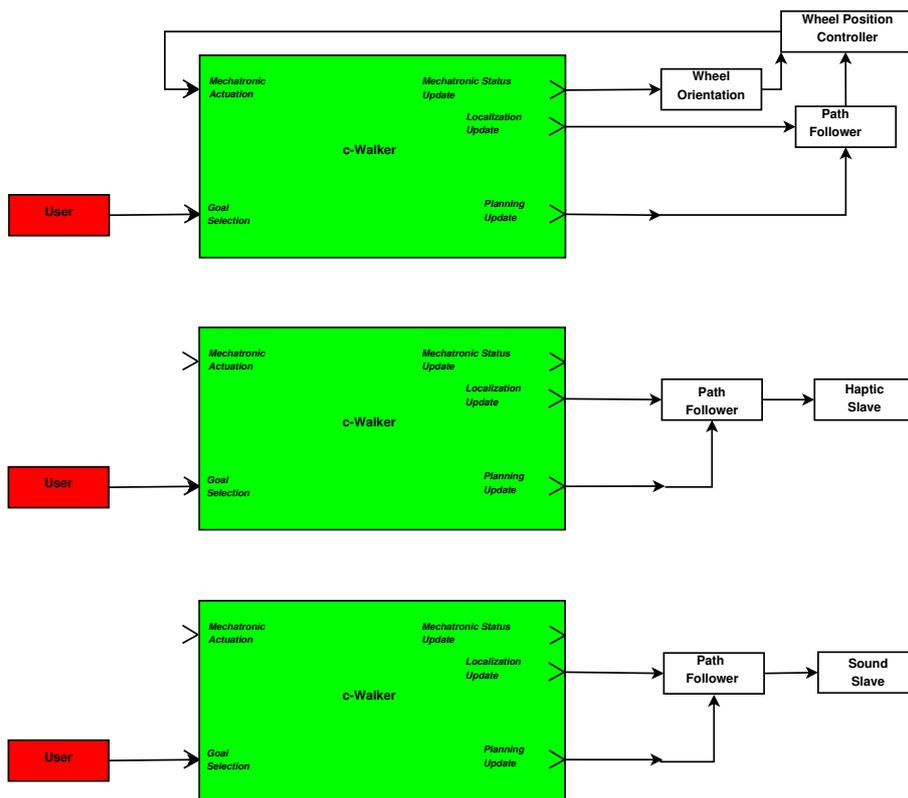


Figure 4.5: The block-scheme of the three guidance systems. The three solutions interact with the *c-Walker* by means of the same software interface, but then are diversified in the way they perform the actuation.

to a `Wheel Poisition Controller` using the Publish/Subscribe middleware. This component also receives real-time information on the current orientation of the wheel and decides the actuation to set the direction to the desired position. The schemes on the middle and on the bottom apply, respectively, to haptic and acoustic (and binaural) guidance. In this case the `Path Follower Components` implements the algorithms discussed in Sec. 4.2.1 and transmits its input either the `Haptic Slave` (see Sec. 4.1.1) or the `Audio Slave` (see Sec. 4.1.2).

## 4.4 Experimental results

### 4.4.1 Study 1

A formative evaluation was designed to compare and contrast the performance of the different guidance systems. Since the preliminary state of user research in this field [87, 86], the main focus of the evaluation was on system performance, rather than on the user experience. The study had two concurrent objectives: to develop a controlled experimental methodology to support system comparisons and to provide practical information to re-design. In line with an ethical application of the inclusive design process [40], at this early stage of the methodological verification process of an evaluation protocol, we involved a sample of University students.

#### Participants

Thirteen participants (6 females, mean age 30 years old, ranging from 26 to 39) took part in the evaluation. They were all students or employees of the University of Trento and gave informed consent prior to inclusion in the study.

#### Design

The study applied a within-subjects design with Guidance (4) and Path (3) as experimental factors. All participants used the four guidance systems (acoustic, haptic, mechanical, binaural) in three different paths. The order of the system conditions was counterbalanced across participants.

#### Apparatus

The experimental apparatus used in the experiment is a prototype of the *c-Walker* shown in Figure 2.1.

The experiments reported here were organised as multiple repetitions of relatively short trajectories. We believe that the adoption of this paradigm produces results comparable to “fewer” repetitions of longer trajectories, in a more controllable and repeatable way. This simplifies the localisation problem. Indeed, the mere use of relative localisation provides acceptable accuracy with an accumulated error below 5cm, when the system operates for a small time (e.g., smaller than 50m) [57]. Therefore the activation of absolute positioning systems which would entail some instrumentation in the environment (e.g., by deploying RFID tags in known positions) was not needed.

The short term **Planner** in the *c-Walker* is reactive: it collects real-time information in the environment and uses it to plan safe courses that avoid collisions with other people or dangerous areas [17]. In this context, we could disable this feature since the experiments took place in free space, without any dynamic obstacles along the way. The planner was configured to generate three different virtual paths (60 centimetres wide and 10 meters long): straight (I), C shaped (C) and S shaped (S). The width of the virtual corridor was above 30 centimetres to the left and to the right of centre of the *c-Walker*. The C path was a quarter of the circumference of a circle with a radius of 6.37 meters. The S path comprised three arches of a circumference with a radius of 4.78 meters. The first and the third arches were  $\frac{1}{12}$ , while the one in the centre was  $\frac{1}{6}$  of the whole circumference. The second arch was bent in the opposite direction compared to the other two. In total there were 6 path variations, two symmetric paths for each shape.

Finally, the **Path Follower** component implements the guidance algorithms described in Section 4.2. The concrete implementation was adapted to the different guidance algorithms. For mechanical guidance, the component decides a direction for the wheel that is transmitted to a **Wheel Poisition Controller**. This component also receives real-time information on the current orientation of the wheel and decides the actuation to set the direction to the desired position. For the haptic and the acoustic (and binaural) guidance, the **Path Follower** implements the algorithms discussed in Section 4.2.1 and transmits its input either to the **Haptic Slave** or to the **Audio Slave**, as detailed in Section 4.1.1 and Section 4.1.2 respectively.

## Procedure

The evaluation was run in a large empty room of the University building by two experimenters: a psychologist who interacted with the participants and a computer scientist who controlled the equipment. At the beginning of the study, participants were provided with the instructions in relation to each guidance system. It was explained that they had to follow the instruction of the *c-Walker*: while they were on the correct trajectory there would be no system intervention. Otherwise, each system would have acted in dif-

ferent ways. The mechanical system would have turned the front wheels modifying its direction onto the right path. In this case, participants could not force the walker and might only follow the suggested trajectory. At the end of the mechanical correction, the participants were given back the control of the walker. For the haptic/acoustic guidance, a vibration/sound (either on the left or right arm/ear) would have indicated the side of the correction necessary to regain the path. It was stressed that under these conditions there was no information indicating the turn intensity. Finally, the binaural guidance would have provided a sound indicating the direction and (the amount of the correction).

Participants were told to be careful in following the instructions to avoid bouncing from one side to the other of the virtual corridor. It was also suggested that whenever they felt like zigzagging, the actual trajectory might be likely in the middle.

Before each trial, the appropriate device was put on the participant (i.e., headphones or haptic bracelets). Only in the case of the binaural system, participants were given a brief training to make them experience the spatial information of the sounds. The starting position of each trial varied among the four corners of a rectangular virtual area (about 12 x 4 meters). The *c-Walker* was positioned by the experimenter with a variable orientation according to the shape of the path to be followed. Specifically, at the beginning of each I trial, the walker was turned 10 degrees either to the left or to the right of the expected trajectory. At the beginning of each C and S trials, the walker was located in the right direction to be followed. Participant started walking after a signal of the experimenter and repeated 10 randomised paths for each guidance system.

At the end of each system evaluation, participants were invited to answer 4 questions, addressing ease of use, self-confidence in route keeping, acceptability of the interface in public spaces and an overall evaluation on a 10 points scale (10=positive). Moreover, participants were invited to provide comments or suggestions. The evaluation lasted around 90 minutes, at the end participants were thanked and paid 10 euros.

### Data analysis

Performance was analysed considering four dependent variables. A measure of error was operationalized as deviation from the optimal trajectory and calculated using the distance of the orthogonal projection between the actual and the optimal trajectory. We collected a sample of 100 measurement (about one value every 10 centimetres along the curvilinear abscissa of the path) that were then averaged. *Time* was measured between the start of participant's movement and the moment the participant reached the intended end of the path. *Length* measured the distance walked by the participant, whereas *speed* corresponded to the ratio between the length and the time.

For each participant and guidance system, we averaged an index scores for the four S,

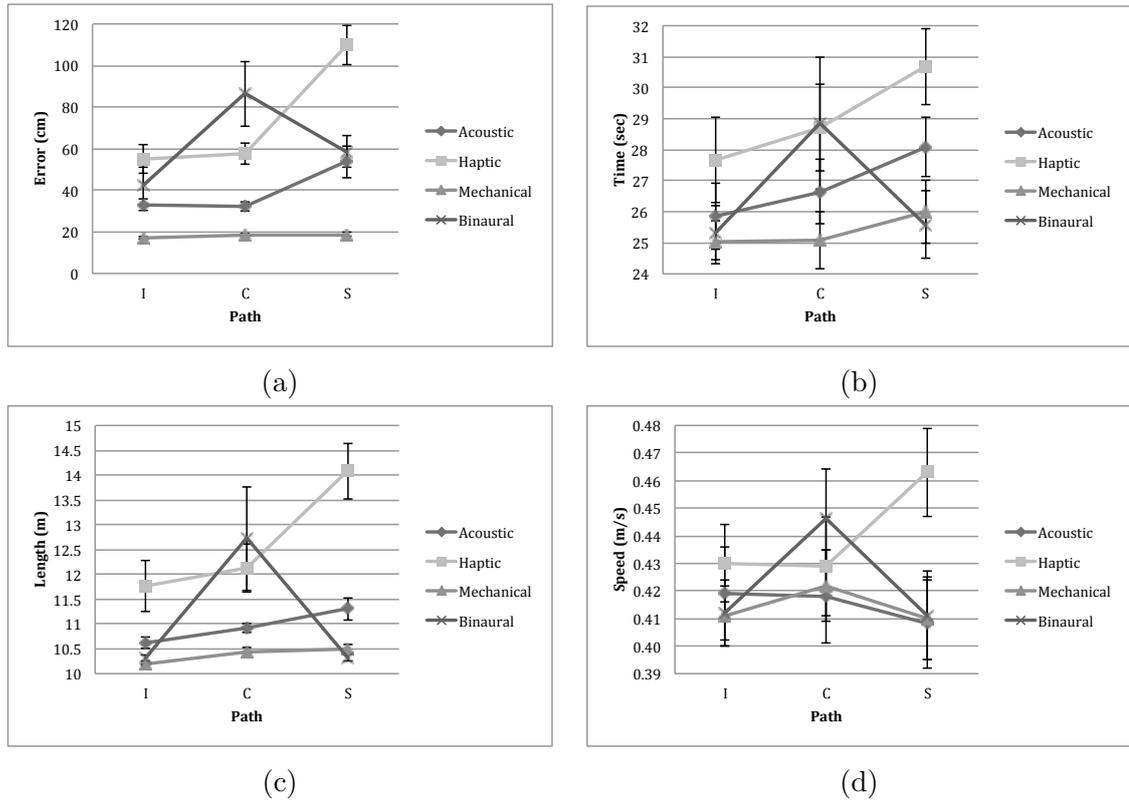


Figure 4.6: Different metrics as function of Guidance and Path: (a) average error (cm), (b) average time (s), (c) average length (m), (d) average speed (m/s).

the four C and the two I paths. Data analysis was performed employing the analysis of variance (ANOVA) with repeated measures on the factors ‘Guidance’ and ‘Path’. Post-hoc pairwise comparisons corrected with Bonferroni for multiple comparisons (two tails) were also computed.

## Results

**Error** Descriptive statistics of error are reported in Fig. 4.6 (a) as a function of Guidance and Path. The ANOVA highlighted a significant effect for Guidance  $F(3, 36) = 27.4$ ,  $p < .01$ , Path  $F(2, 24) = 17.3$ ,  $p < .01$  and for the interaction  $F(6, 72) = 10.3$ ,  $p < .01$ . Post-hoc pairwise comparison (Tab. 4.2) indicated that the mechanical guidance differed significantly from all the others ( $p < .01$ ) being the most precise. Moreover, the acoustic guidance was significantly different from the haptic ( $p < .01$ ). Post-hoc comparisons indicated that the I path was significantly easier from the other two ( $p < .01$ ).

In the mechanical guidance condition, the error was not affected by the path and showed very low variability among participants. On the contrary, for all other conditions

Guidance	I	C	S	Average	Pairwise comparison
Acoustic	33.0	32.2	53.7	<b>39.6</b>	I vs. S, $p = .053$ C vs. S, $p < .05$
Haptic	55.0	57.4	109.7	<b>74.0</b>	I vs. C and S, $p < .01$
Mechanical	17.1	18.3	18.8	<b>18.1</b>	n.s.
Binaural	42.1	86.3	58.7	<b>62.4</b>	I vs. C, $p < .05$
Average	<b>36.8</b>	<b>48.6</b>	<b>60.2</b>		
Pairwise comparison	M vs. A and H, $p < .01$ A vs. H, $p = .07$	A vs. H and M, $p < .01$ A vs. B, $p < .05$ M vs. H and B, $p < .01$	M vs. all, $p < .01$ , H vs. A and B, $p < .01$		

Table 4.2: Average error (cm) for each experimental condition and significant post-hoc pairwise comparisons.

there was an effect of Path on the magnitude of the error. Mostly for the haptic, but also for the acoustic guidance, the S path had the highest error. Interestingly, for the binaural guidance, the highest error emerged with the C path. Fig. 4.7 shows some qualitative results of the experiments.

**Time** The ANOVA on time showed a significant effect for Guidance  $F(3, 36) = 3.98$ ,  $p < .05$ , Path  $F(2, 24) = 7.54$ ,  $p < .01$  and for the interaction  $F(6, 72) = 2.89$ ,  $p < .05$ . Fig. 4.6 (b) shows the average time in relation to both Guidance and Path. Post-hoc pairwise comparison showed that the mechanical guidance system was significantly faster than the haptic ( $p < .05$ ), and that the I path differed significantly from the S path ( $p < .05$ ). Walking time was independent of Path for the mechanical and the binaural guidance. Conversely, the S path was performed significantly slower than the I path for both the acoustic and the haptic guidance. The average time and the results of the post-hoc pairwise comparisons are summarized in Tab. 4.3.

**Length** Only two participants (both in the S path and the binaural guidance condition) walked less than the optimal path length. The ANOVA showed a significant effect for Guidance  $F(3, 36) = 15.1$ ,  $p < .01$ , Path  $F(2, 24) = 9.1$ ,  $p < .01$  and for the interaction

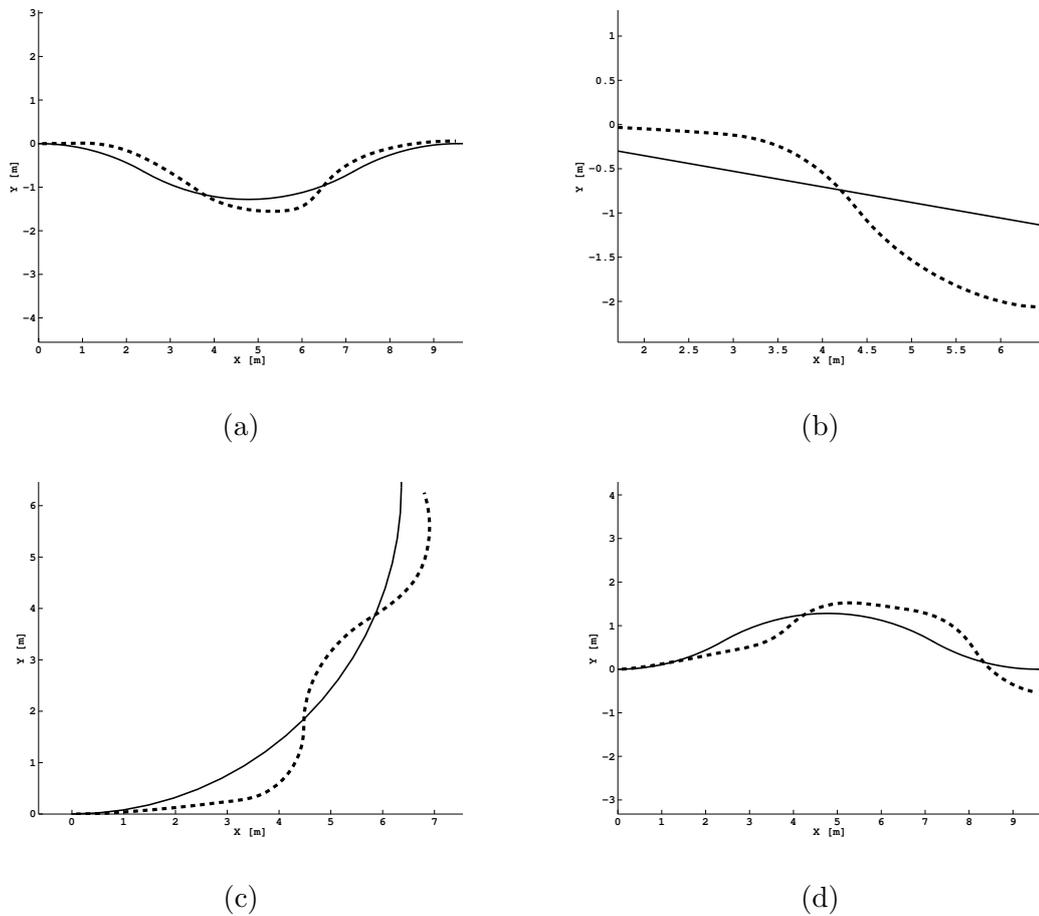


Figure 4.7: Some examples of qualitative results for different guidance algorithms: (a) mechanical, (b) haptic, (c) audio, (d) binaural. The trajectory of the user is the dash line.

$F(6, 72) = 6.1, p < .01$ . Post-hoc comparisons indicated that the haptic guidance differed significantly from all the others ( $p < .01$  mechanical and acoustic and  $p < .05$  binaural). Moreover, the mechanical guidance differed significantly from the acoustic ( $p < .01$ ). The I path differed significantly from the C ( $p < .01$ ) and S ( $p < .05$ ) paths (Tab. 4.4). The haptic guidance showed the worst result in the S path. For the mechanical condition, the performance was different between the I and S path. For the binaural condition there was no effect of Path. Fig. 4.6 (c) shows the average length in relation to both Guidance and Path.

**Speed** The analysis of variance on speed reported only a significant interaction between Guidance and Path  $F(6, 72) = 3.05, p < .01$ . Fig. 4.6 (d) reports the average time as a function of experimental conditions. The average speed and the results of the post-hoc pairwise comparisons are summarized in Tab. 4.5. Participants were particularly fast

Guidance	I	C	S	Average	Pairwise comparison
Acoustic	25.8	26.7	28.1	<b>26.9</b>	I vs. S, $p < .01$
Haptic	27.7	28.7	30.7	<b>29.0</b>	I vs. S, $p < .05$
Mechanical	25.0	25.1	26.0	<b>25.4</b>	n.s.
Binaural	25.3	28.9	25.6	<b>26.6</b>	n.s.
Average	<b>26.0</b>	<b>27.3</b>	<b>27.6</b>		
Pairwise comparison	n.s.	n.s.	H vs. M and B, $p < .05$		

Table 4.3: Average time (sec) for each experimental condition and significant post-hoc pairwise comparisons.

Guidance	I	C	S	Average	Pairwise comparison
Acoustic	10.62	10.92	11.3	<b>10.9</b>	I vs. C, $p < .05$ I vs. S, $p < .01$
Haptic	11.76	12.12	14.08	<b>12.7</b>	S vs. I, $p < .01$ S vs. C, $p < .05$
Mechanical	10.18	10.44	10.5	<b>10.4</b>	I vs. S, $p < .01$
Binaural	10.31	12.72	10.32	<b>11.1</b>	n.s.
Average	<b>10.7</b>	<b>11.6</b>	<b>11.6</b>		
Pairwise comparison	M vs. A, $p < .05$	M vs. H, $p < .05$ M vs. A $p = .07$	all the comparisons $p < .01$ except M vs. A, $p < .05$ and M vs. B not significant		

Table 4.4: Average travelled length (m) for each experimental condition and significant post-hoc pairwise comparisons.

Guidance	I	C	S	Average	Pairwise comparison
Acoustic	0.42	0.42	0.41	<b>0.42</b>	n.s.
Haptic	0.43	0.43	0.46	<b>0.44</b>	I vs. S, $p < .01$
Mechanical	0.41	0.42	0.41	<b>0.41</b>	n.s.
Binaural	0.41	0.45	0.41	<b>0.42</b>	n.s.
Average	<b>0.42</b>	<b>0.43</b>	<b>0.42</b>		
Pairwise comparison	n.s.	n.s.	A vs. H, $p < .05$ M vs. H, $p = .07$		

Table 4.5: Average speed (m/sec) for each experimental condition and significant post-hoc pairwise comparisons.

walking the S path.

### Questionnaire

Participants scores to the the four questionnaire items were normalized for each participant in relation to the highest score provided among all the answers. The ANOVA indicates that the mechanical guidance is perceived as easier to use with no other significant differences among the other systems. The same results have emerged in relation to the confidence to maintain the correct trajectory. Concerning the acceptability to use the guidance systems in public spaces, the mechanical guidance was again the preferred one in relation to both the acoustic and binaural while no difference has emerged in relation to the haptic. Finally, participants liked the mechanical guidance the most in relation to both haptic and acoustic systems, while no difference has emerged in relation to the binaural one.

Participants spontaneously commented that the mechanical system was easy to follow and required little attention. However, some of them complained that it might be perceived as coercive and risky due to possible errors in route planning. Other people worried about the dangerous effect of a quick turn of the wheels mostly for older users. Participants reported a general dislike about wearing headphones mostly because they might miss important environmental sounds and because of the look. Most of the participants agreed that the binaural condition required more attention than all the other systems. Participants however appreciated that it was something new, interesting and provided a constant feedback on the position. Most of them preferred the binaural system to

the acoustic one because it provided more information, yet some reported a difficulty in discriminating the direction the sound was coming.

Most of the participants reported to prefer the haptic guidance system to the acoustic, as easier and less intrusive. In relation to the both guidance condition, participants complained about the poverty of the left and right instructions and the lack of a modulation. Some participants suggested possible ways to increase communication richness, such as, for the acoustic system, different volume indicating the magnitude; verbal feedback; different tones in relation to the angle. For the haptic system comments included modulating the frequency of the vibration in relation to the magnitude of the correction. Some participants reported a kind of annoyance for the haptic stimulation but only for the first minutes of use.

### Discussion

The aim of this study was to gather quantitative and qualitative information in relation to the evaluation of four different guidance systems. To this aim participants had the opportunity to navigate non-visible paths (i.e., virtual corridors) using four the different guidance systems. To maintain the correct trajectory participants could only rely on the instructions provided by the *c-Walker* and, after using each system, they were asked to provide feedback.

As expected, in terms of performance, the mechanical guidance was the most precise. Although an error emerged because of the freedom left to participants, the results show the consistency of the deviation along the different paths, a low variability among the participants and a slight difference in relation to the shape of the paths. The results of the questionnaire further support quantitative data showing that, on average, participants liked the mechanical guidance the most in relation to easiness, confidence in maintaining the trajectory, acceptability and overall judgment. The only concern for some users was that it might be perceived as coercive and risky due to possible errors in route planning. In fact, the mechanical guidance was active in the sense that participants had to passively follow the trajectory imposed by the walker. Differently, in the other three guidance systems, the participants were actively driving based on the interpretation of the provided instructions. In the acoustic guidance, there were only left and right sounds while in the binaural guidance, the sound was modulated by modifying the binaural difference between the two ears. Although more informative, in terms of quantifying the angle of the suggested trajectory, the binaural guidance system emerged to be worse than the acoustic system in the C path. However, it is likely that with adequate training the performance with the binaural system could improve a lot. The results of the questionnaire suggest that both the systems using headphones were not very acceptable because of the possibility to

miss environmental sounds and because of the look. Moreover, the binaural system was reported to require more attention than the acoustic one, although no difference emerged in terms of confidence in maintaining the correct trajectory. Overall, the binaural guidance was appreciated because it was something new and provided detailed information. Indeed, most of the participants' suggestions related to the acoustic and haptic guidance systems were addressed at codifying the instructions in terms of the angle of the correction.

Significant performance differences emerged between the haptic and the acoustic guidance, which could in part be explained by the natural tendency to respond faster to auditory stimuli rather than to tactile stimuli, and in part by the different algorithm employed in the evaluation. This issue is addressed in the second study here presented. Looking at participants performance however it is evident that, independent of the communication channel, the dichotomous nature of the stimulation (left-right) tended to stimulate long left and right corrections leading to zigzagging. One participant explicitly mentioned this feeling while commenting on the haptic guidance. In terms of user experience, the haptic guidance was perceived as more acceptable than the acoustic and the binaural systems, and no different from the mechanical one. Indeed, most of the participants commented that the haptic bracelets could be hidden and did not interfere with the environmental acoustic information.

#### 4.4.2 Study 2

This evaluation study was designed to clarify the differences emerged in study 1 between the haptic and the acoustic guidance. To this aim both input devices (bracelets and headphones) were interfaced to the same guidance algorithm and tested following the same experimental protocol as study 1, except that participants were required to test only the acoustic and the haptic guidance systems. Moreover, the haptic guidance system was modified using the acoustic guidance algorithm. In this way, we could test directly the effect of the interface. Ten participants (2 females, mean age 30 years old range 24-35) took part in the study.

#### Results

Descriptive statistics of error are reported in Fig. 4.8 as a function of Guidance and Path. The ANOVA showed a significant effect for the factor Path  $F(2,18) = 11.0$ ,  $p < .01$  but not Guidance. The post-hoc pairwise comparison showed that the S path differed significantly from the other two ( $p < .05$ ) confirming its higher complexity.

The ANOVA on time, length, and speed returned the same trend of results: a main effect only for Path. The analysis of the questionnaire confirmed a preference for the

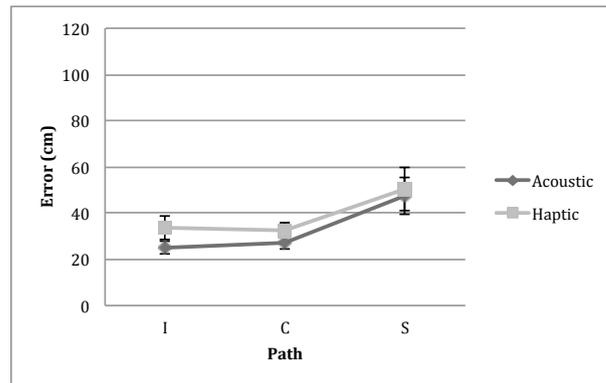


Figure 4.8: Average error (cm) as a function of Guidance and Path.

acceptability of the haptic guidance in public spaces. Finally, a between-study analysis of variance comparing the performance of participants using the sound system in study 1 and study 2 returned no significant differences due to study, path, or their interaction.

### Discussion

The study indicates that the haptic and acoustic interfaces do not differ in terms of performance, and that the results of study 1 may be attributed entirely to the different algorithms tested. Furthermore, they confirm a preference for the haptic guidance but only regarding its social acceptability in public spaces. Furthermore, the similarities in both performance and user-experience of the acoustic guidance in the two studies is an indicator of the strong reliability and external validity of the evaluation protocol.

Tab. 4.6 propose a ranking of the 4 guidance systems, combining empirical observations, measurements and participants' comments in both studies. The best guidance was no doubt the mechanical one, followed by the haptic, acoustic and binaural systems. The evaluations highlighted new challenges for the sociotechnical design of future guidance system. In particular a major issue emerged with regards to the acceptability of the practical requirement of wearing headphones. The binaural system was perceived as a promising solution which captured the user attention. However, more work is needed in order to improve the communication of the directional information.

	<b>Performance</b>	<b>Easiness</b>	<b>Confidence</b>	<b>Acceptability</b>	<b>Total</b>
<b>Acoustic</b>	++	++	++	+	7
<b>Haptic</b>	++	++	++	+++	9
<b>Mechanical</b>	++++	+++	+++	+++	13
<b>Binaural</b>	+	+	++	+	5

Table 4.6: Summary of the results.



# Chapter 5

## Steering by brakes

### 5.1 Problem Formulation

#### 5.1.1 Half Cart Model

The *c-Walker* is driven in the environment using the thrust generated by the user, who remains in charge of the motion along the trajectory. The thrust is applied through the grips of the walker and it is a combination of forces coming from the arm muscles with the user weight. In the proposed solution, the following simplifying assumptions hold.

**Assumption 1** *The half-car model is considered. Hence, the left and right force inputs are analysed separately.*

**Assumption 2** *The viscous friction on the wheels is assumed sufficiently high to avoid rotations around the trolley Center of Gravity (CoG) when the right or left forces are applied in isolation.*

**Assumption 3** *The forces applied on the handles that are parallel to the wheel axles do not generate motions, i.e., the trolley does not slide along the lateral direction. Generally speaking, the four wheels roll without slippage with respect to the plane of motion.*

**Assumption 4** *The wheels always remain in contact with the plane of motion, hence no roll or pitch motions are considered.*

In view of Assumption 2, when only the right (left) force is applied, the cart rotates around the rear left (right) wheel. Assumption 3 is the most difficult to verify. In practice, it requires to use wheels with inflated inner tubes.

With reference to Fig. 4.2 (a), let  $\langle G \rangle = \{O, X, Y, Z\}$  be a fixed right-handed reference frame, whose plane  $\Pi = X \times Y$  is the plane of motion of the cart,  $Z$  points outwards the

plane  $\Pi$  and  $O$  is the origin of the reference frame. In light of the previous assumptions, the dynamic of the rear wheels can be expressed as

$$\tau_{(\cdot)} = a_1(\tau_{(\cdot)}^b - b_w\omega_{(\cdot)} + \tau_{(\cdot)}^h), \quad (5.1)$$

where  $a_1 = J_w/(r^2m_w + J_w)$ , with  $m_w$  the mass of the wheel,  $J_w$  its moment of inertia and  $r$  is the wheel radius, which are supposed equal for all the wheels, as customary. By considering the notation  $M_{(\cdot)}$  to refer to the quantity  $M$  of the left or right side of the trolley,  $\tau_{(\cdot)}^h$  comprises: a) the rolling resistance  $\tau_{(\cdot)}^a = \xi_{(\cdot)}^b c_v/r$ , with  $c_v$  being the coefficient of rolling friction and  $\xi_{(\cdot)}^b$  being the reaction force of the ground on the back wheel; b) the torque  $w_{(\cdot)}^b r$ , where  $w_{(\cdot)}^b$  is the acceleration of the wheel rotation axis imposed by the user and by the system inertia, and transferred to the wheel hubs by the mechanical structure. Finally,  $\tau_{(\cdot)}^b$  is the braking torque and  $b_w\omega_{(\cdot)}$  the viscous friction around the wheel tube, with  $\omega_{(\cdot)}$  the angular velocity of the wheel.

In order to verify the pure rolling Assumption 3, the following equation must hold

$$\begin{aligned} -\frac{\mu_s \xi_{(\cdot)}^b (r^2 m_w + J_w) + w_{(\cdot)}^b J_w}{r m_w} &\leq \tau_{(\cdot)}^b + \tau_{(\cdot)}^a - b_w \omega_{(\cdot)} \\ &\leq \frac{\mu_s \xi_{(\cdot)}^b (r^2 m_w + J_w) - w_{(\cdot)}^b J_w}{r m_w}. \end{aligned} \quad (5.2)$$

where  $\mu_s$  is the friction coefficient of the contact point between the wheel and the ground.

Since  $w_{(\cdot)}^b$  is a known function of the user thrust and assuming that for safety reasons we need to halt the vehicle from time to time, the maximum braking action is required to satisfy  $\tau_{(\cdot)}^s > |w_{(\cdot)}^b r|$  (from (5.1)) and constraint (5.2).

### 5.1.2 Vehicle Dynamic Model

Let  $\mathbf{x} = [x, y, \theta]^T \in \mathbb{R}^2 \times S$  be the kinematic configuration of the cart, where  $(x, y)$  are the coordinates of the mid-point (as Q in Cha. 4) of the rear wheels axle in  $\Pi$  and  $\theta$  is the orientation of the vehicle w.r.t. the  $X$  axis (see Fig. 4.2 (a)). The dynamic model of the *c-Walker* can be assimilated to a unicycle

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \cos(\theta) \\ v \sin(\theta) \\ \omega \\ \frac{F}{m} \\ \frac{N}{J} \end{bmatrix} \quad (5.3)$$

where  $v$  is the forward velocity of the vehicle and  $\omega$  its angular velocity.  $F$  is the external force acting on the vehicle along the direction of motion,  $N$  is the external torque about

the  $Z$ -axis,  $m$  and  $J$  are the mass and moment of inertia of the cart. Recalling that the torques independently applied to the right and left wheel are given by (5.1), we have the invertible linear relations

$$F = \frac{\tau_r + \tau_l}{r} \quad \text{and} \quad N = \frac{(\tau_r - \tau_l)d}{2r}. \quad (5.4)$$

where  $d$  is the axle length .

### 5.1.3 Dynamic Path Following Problem

We require the walker to converge to our planned path defined in  $\Pi$ , which we will assume to be smooth (i.e., with a well defined tangent on each point) and with a known curvature. The planned path is typically composed of straight segments and circular arcs connected with clothoids.

In this setting, a standard way of solving the problem is by a dynamic path following algorithm. In order to design a control law of this kind, we first express the vehicle generalised coordinates  $\mathbf{x}$  in terms of a Frenet-Serret frame moving along the path. Denote by  $s$  the curvilinear abscissa along the path, by  $y_d$  the distance between the origin of the Frenet-Serret frame and the reference point of the  $c$ -Walker along the  $Y$ -axis of the Frenet-Serret frame and by  $\theta_d$  the angle between the  $X$ -axis of the  $c$ -Walker and the  $X$ -axis of the Frenet-Serret frame (see Fig 4.2 (a)). Define  $\tilde{\theta} = \theta - \theta_d$ , then the vehicle dynamics (5.3) can be rewritten as (see [55]):

$$\begin{aligned} \dot{s} &= \frac{v \cos(\tilde{\theta})}{1 - c(s)y_d}, \\ \dot{y}_d &= v \sin(\tilde{\theta}), \\ \dot{\tilde{\theta}} &= \omega - \frac{c(s)v \cos(\tilde{\theta})}{1 - c(s)y_d}, \\ \dot{v} &= \frac{F}{m}, \\ \dot{\tilde{\omega}} &= \frac{N}{J} - c(s)\ddot{s} - c(s)' \left( \frac{v \cos(\tilde{\theta})}{1 - c(s)y_d} \right)^2, \end{aligned} \quad (5.5)$$

where  $\tilde{\omega} = \dot{\tilde{\theta}}$ . The path curvature is defined as  $c(s) = d\theta_d(s)/ds$ , while  $c'(s) = dc(s)/ds$ . This model is commonplace in the literature [22, 48, 78, 63] and we adopted it for this work. The dynamic path following problem requires to design a control law  $F(t)$  and  $N(t)$  for the system in (5.5) such that

$$\lim_{t \rightarrow +\infty} y_d(t) = 0 \quad \text{and} \quad \lim_{t \rightarrow +\infty} \tilde{\theta}(t) = 0. \quad (5.6)$$

### 5.1.4 Problem Formulation

The objective is not to achieve an asymptotic convergence to a path. Rather, we require the user to always be within a safety corridor. Inside the corridor the user has to be able to move freely. In addition, we require that she makes constant progress along the path (as long as she pushes forward the walker). Such requirements lead us to the definition of the *loose following problem*, which can be stated in terms of the following definitions:

$$\exists \bar{t} \text{ s.t. } |y_d(t)| \leq \bar{y}_d, |\tilde{\theta}(t)| \leq \bar{\theta}, \forall t > \bar{t}. \quad (5.7)$$

The first requirement encodes the idea of the safety corridor, while the second one ensures that the vehicle is always oriented in the direction of the path, thus leading to a finite completion time if the forward velocity is non-null. In control theoretical terms, we require controlled invariance [5] of the subset of the state-space defined in (5.7).

## 5.2 Solution

Our solution to the problem formulated in (5.7) is based on the algorithm for path following proposed by Bicchi et. al. [4]. We first summarise the key points of this algorithm (for the reader convenience) and then we move to the description of our particular solution.

### 5.2.1 Hybrid Solution to the Path Following Problem

In [4] a hybrid feedback controller able to solve the path following problem in the sense of (5.7) for unicycle-like vehicles with bounded curvature radius has been presented. Strictly speaking, the solution proposed solves the problem of driving a Dubin's car to a generic path assuming a maximum curvature for the path. The authors show that their hybrid controller is asymptotically stable with respect to the path-related coordinates  $(\tilde{y}_d, \tilde{\theta})$ , where  $\tilde{y}_d = y_d/R$  and  $R$  is the fixed maximum turning radius of the vehicle. The controller automaton is compounded of three discrete states, i.e., *Go Straight*, *Turn Right* and *Turn Left*. The state space  $(\tilde{y}_d, \tilde{\theta})$  is suitably partitioned into a set of non-overlapping regions. In each region only one of the three states is active. With respect to the kinematic subsystem of (5.5), each state defines the control input  $\omega$  as

$$\begin{cases} \omega = 0 & \text{if } Go \text{ Straight} \\ \omega = -\frac{v}{R} & \text{if } Turn \text{ Right} \\ \omega = \frac{v}{R} & \text{if } Turn \text{ Left} \end{cases}$$

assuming the forward input  $v > 0$  is known.

### 5.2.2 Braking System

For the problem at hand, the available input variables are the independent braking forces  $\tau_l^b$  and  $\tau_r^b$ , acting on the left and right wheels respectively. Similarly to [36, 71], the braking forces in (5.1) and (5.2) are represented by

$$\tau_{(\cdot)}^b = b_{(\cdot)}\omega_{(\cdot)} - \delta_{\omega_{(\cdot)}}c_{(\cdot)}\tau_{(\cdot)}^h, \quad (5.8)$$

where  $b_{\max} \geq b_{(\cdot)} \geq 0$  is the friction coefficient of the electromagnetic brakes, which is a function of the commanded current, while  $\delta_{\omega_{(\cdot)}}$  is the classic Kronecher  $\delta$ -function, which is equal to one in zero and zero otherwise. Since, due to safety issues, the system has to be in condition to halt the user, we have  $0 \leq c_{(\cdot)} \leq 1$  and  $b_{\max}|\omega_{(\cdot)}| \geq |\tau_{(\cdot)}^h - b_w\omega_{(\cdot)}|$ ,  $\forall \omega_{(\cdot)}$ . The control law we are designing has a limited set of control possibilities. More precisely, the admitted control values for each wheel are either  $c_{(\cdot)} = 0$  and  $b_{(\cdot)} = 0$  or  $c_{(\cdot)} = 1$  and  $b_{(\cdot)} = b_{\max}$ , i.e., no brake or full brake. If the brake is fully active on the right wheel, from (5.1) it follows that  $\omega_r(t) \rightarrow 0$ . Hence, the vehicle will end-up in following a circular path with fixed curvature radius  $R = d/2$ , where  $d$  is the rear wheel inter-axle length, travelled in clockwise direction if  $v > 0$ . The circular path with the same radius  $R$  will be instead followed in counter-clockwise direction if the left brake is fully active and  $v > 0$ . As a consequence, in all the cases of active braking system,  $\omega = v/R$ . If no braking action is applied at all, the user will drive the *c-Walker* uncontrolled. On the other hand, if both brakes are fully active, the vehicle will reach the full stop.

As a consequence, the hybrid feedback controller designed in [4] will be considered as the vehicle angular desired velocity generator for the braking system. In practice, we will have the following braking system control law

$$\left\{ \begin{array}{ll} b_r = b_l = c_r = c_l = 0 & \text{if } Go \text{ Straight,} \\ b_r = b_{\max} \wedge c_r = 1 \wedge b_l = c_l = 0 & \text{if } Turn \text{ Right,} \\ b_l = b_{\max} \wedge c_l = 1 \wedge b_r = c_r = 0 & \text{if } Turn \text{ Left,} \\ ib_r = b_l = b_{\max} \wedge c_r = c_l = 1 & \text{if } Stop. \end{array} \right. \quad (5.9)$$

By direct experimental measurements made on the system at hand, we have observed that the forces applied by the user to the mechanical system are negligible with respect to the maximum braking action. Moreover, the inertia of the system as well as the maximum forward velocity  $v$  are limited. As a consequence, the time needed to stop the wheel rotation is negligible and, hence, it can be assumed  $\omega_{(\cdot)} = 0$  whenever the braking system is fully active.

Hence, to account for the loose path tracking solution, it is sufficient to define a controlled invariant [5] for the subspace  $(\tilde{y}_d, \tilde{\theta})$  of (5.5). In particular, by considering the

phase portrait of [4], the subspace of  $\mathcal{S} = \{(\tilde{y}_d, \tilde{\theta}) \mid -2 \leq \tilde{y}_d \leq 2, -\pi/2 \leq \tilde{\theta} \leq \pi/2\}$  is controlled invariant for the system (5.5). Therefore, it is possible to switch to the hybrid state *Go Straight* of (5.9) whenever the system enters the subspace. However, such an approach is prone to failure in a practical implementation, mainly due to the noisy localisation algorithm and to the presence of possible uncertainty in the fixed radius estimation. Therefore, to guarantee a minimum robust margin, the switch towards the state *Go Straight* is enforced whenever the following constraint is satisfied

$$\tilde{y}_d^2 + k_\theta \tilde{\theta}^2 \leq T_b, \quad (5.10)$$

where the level set curve  $\mathcal{L} = \{(\tilde{y}_d, \tilde{\theta}) \mid \tilde{y}_d^2 + k_\theta \tilde{\theta}^2 \leq T_b\} \subset \mathcal{S}$  and  $T_b > 0$  is a tuning constant. On the other hand, if (5.10) is not satisfied, the state is switched to the proper manoeuvre according to [4].

Nevertheless, the depicted approach may lead to chattering in the braking actuation, which is perceived as uncomfortable by the user. By adding an hysteresis to the switching curve, i.e., the state is switched from *Go Straight* to another manoeuvre if (5.10) is verified for  $T'_b > T_b$ , provided that  $\mathcal{L}' = \{(\tilde{y}_d, \tilde{\theta}) \mid \tilde{y}_d^2 + k_\theta \tilde{\theta}^2 \leq T'_b\} \subset \mathcal{S}$ . The proposed approach solves the loose path following as defined in (5.7).

Another important issue to discuss is the presence of the user in the loop, which can be regarded as the system actuator. Such an actuation could operate as the desired input (if it points forward), as a disturbance to be rejected or as a combination of desired effect and of disturbance. A possible case where the user operates as a pure disturbance is when the control law requires the *c-Walker* to turn right, while the user wants to turn left. Such a situation leads to a deadlock. Whenever this condition is detected (i.e., whenever one of the two brakes is powered and the overall velocity is zero), the system is disengaged since the user has evidently changed her mind on where to go.

Another point of departure from [4] is related to the possibility of having the user pulling the trolley rather than push it. In this case, the velocity will be negative. To account for this situation, we consider as controlled variable  $\hat{\theta} = \tilde{\theta} + \pi$  since the vehicle has to travel the path going backwards. Furthermore, due to the symmetric property of the unicycle-like vehicle, the manoeuvre to follow are reverted for  $v < 0$ . It has to be noted that the solution still works even if the user has a completely erratic behaviour, providing that the user maintains the same forward velocity direction for the amount of time needed to reach the controlled invariant.

Finally, we further notice that the control algorithm here designed uses only the localisation estimates as feedback data, namely  $(\tilde{y}_d, \tilde{\theta})$ , which are provided by an external module [59, 57]. This makes a remarkable difference with respect to existing solutions which needs an estimate of the user applied forces in order to be effective [27]. For ex-

ample, the *i-Walker* rollator [19] is equipped with triaxial force sensors on the handles to estimate the user applied forces.

### 5.3 Simulations and Experiments

To prove the validity of the proposed approach, we first report some of the simulation results obtained during the extensive tests, while in Sec. 5.3.2 field experiments on the *c-Walker* first prototype are offered and discussed.

#### 5.3.1 Simulation Results

In the simulation, the *c-Walker* starts from rest condition at distance of 1 meter from the path pointing in the opposite direction (see the *Initial Position* in Fig. 5.1 (a)). The human forces have been simulated using the dynamic controller in [78] plus additional disturbance torques, which resembles an user roughly following the direction suggestions coming from a visual interface. The path to follow and the trajectory followed by the user are reported in Fig. 5.1 (a) with a thick dashed and a solid line respectively. The phase portrait of the manoeuvres along the path is reported in Fig. 5.1 (b), where the invariant set as well as the outer and inner sets defined in (5.10) with  $T_b = 0.1$ ,  $T'_b = 0.4$  and  $k_\theta = 2$  are clearly visible. From the figure, we can clearly see that design requirements are met: a) the *c-Walker* is free to move inside the outer set (user comfort); b) the controller switches to turning manoeuvres whenever the cart exits from the outer set (braking control); c) the trajectories never exceed the controlled invariant set defined in (5.7).

For the same simulation scenario, Fig. 5.2 reports the braking action as defined in (5.8) and applied to the right (solid line) and left (thick dashed line) wheels, respectively. It is worth noting that the system never reaches the *Stop* state, i.e., there is not any  $t$  such that both brakes are active. After the first 10 seconds, the intervention of the braking system becomes sporadic, thus increasing the comfort of the user. On the contrary, in the first 5 seconds the system continuously releases and activates the brake on the right wheel. This is due by the fact that the user wants to follow a trajectory that is different from the one imposed by the proposed algorithm. As a consequence, the algorithm allows the user to move slowly along his desired direct and, hence, avoids deadlock. It is worthwhile that this behaviour can be shaped as desired based on the user needs.

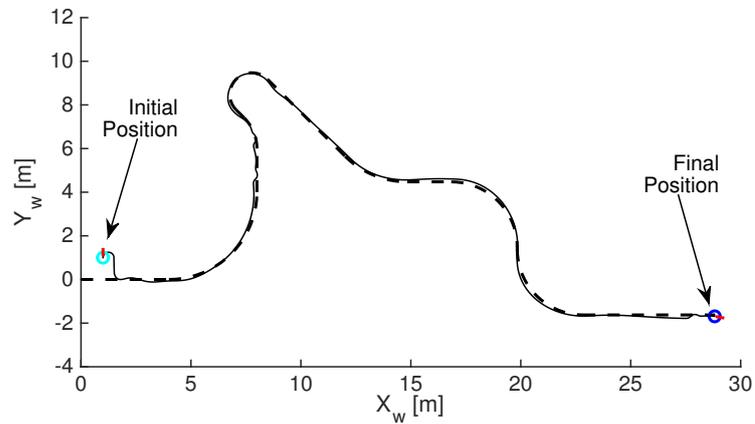
#### 5.3.2 Experimental Results

To prove the effectiveness of the system in practical situations, during the experimental trials the user did not receive any prior information on the path to follow. This way, he

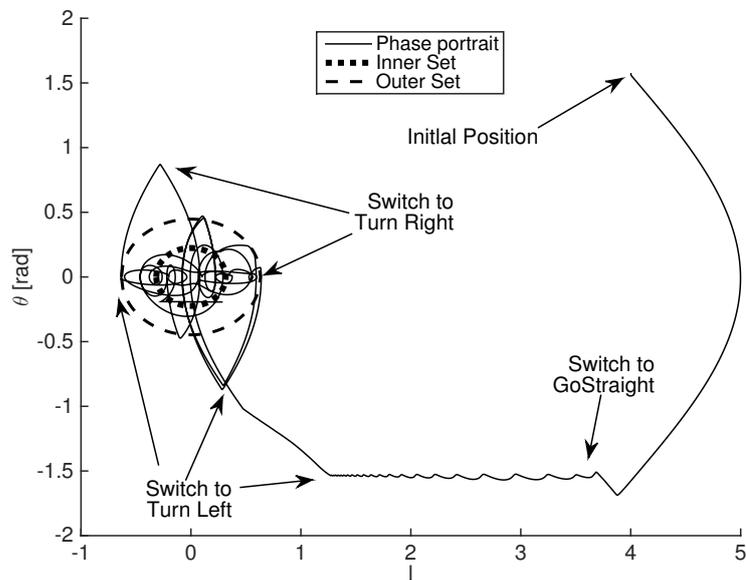
completely relies on the braking system guidance. The localisation algorithm of the cart has been implemented following [57] (as described in Cha. 3), while the design parameters for the inner and outer loop have been chosen equal to  $T_b = 0.4$ ,  $T'_b = 1.6$  and  $k_\theta = 2$ .

Fig. 5.3 (a) reports the trajectory followed by the user for an  $S$  shaped desired path of 6 meters length. The user starts at a distance of one meter and parallel to the path. It is evident how the  $c$ -Walker gently guides the user towards the end of the path. It is important to notice that in the  $Zero$  state the user may or may not follow a straight path, since it is completely uncontrolled. This behaviour is highlighted in Fig. 5.3 (b), where the phase portrait is reported. The switching right out of the outer loop is due to the change of the curvature of the  $S$  path, that changes the orientation of the Frenet-Serret frame as well. From Fig. 5.3 (b) remains evident how the system is steadily controlled inside the invariant set. Fig. 5.4 reports the braking action on the left and right wheels. Both, the commanded braking action (solid line) and the actual implemented action (thick dashed line) are reported. Again, the number of corrections due to the braking algorithm are very limited in time and do not generate any chattering or undesirable behaviour.

In the second experiment, Fig. 5.5, the user has to follow a  $U$  shaped path. In this experiment, the user starts from a position that is 1 meter far apart and oriented in the opposite direction with respect to the path. Again, the user is guided towards the end of the path. Notice that the followed path requires braking corrections that are spread out in time but still very relevant. Such a behaviour can be felt as uncomfortable by the user, but it is unavoidable if the curve to path has a greater turning radius with respect to the vehicle limited radius  $R$ .



(a)



(b)

Figure 5.1: Trajectory followed by the *c*-Walker (a) and phase portrait (b) in the case of a user roughly following the desired path.

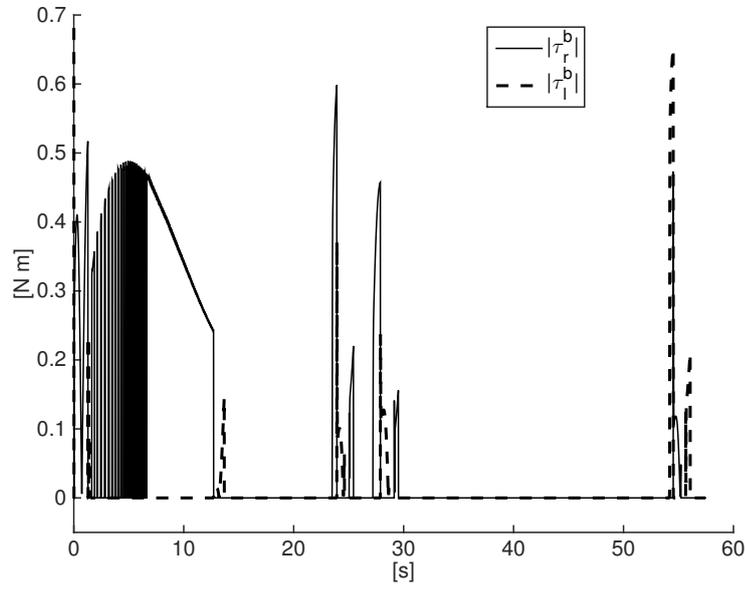


Figure 5.2: The braking action applied by the system in the case of the trajectory of Fig. 5.1.

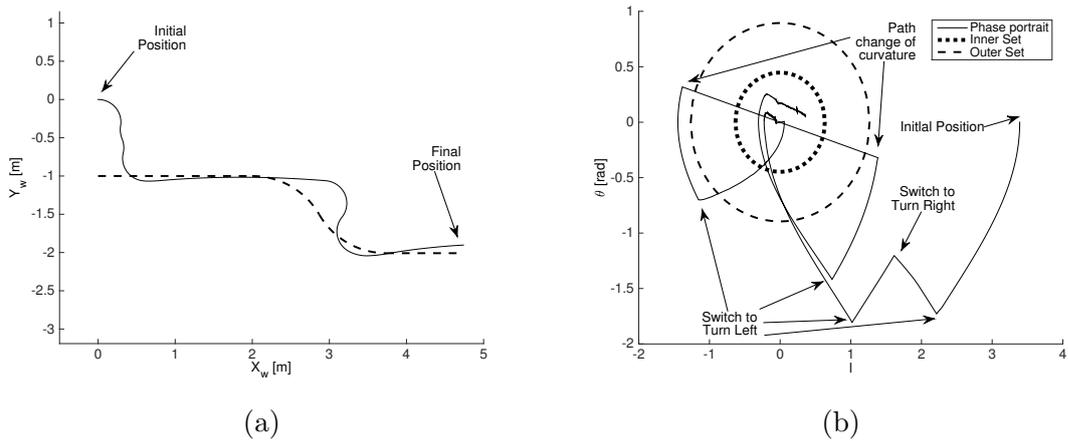


Figure 5.3: Experimental trajectory followed by the *c-Walker* (a) and phase portrait (b) in the case of a blind folded user following an *S* shaped path.

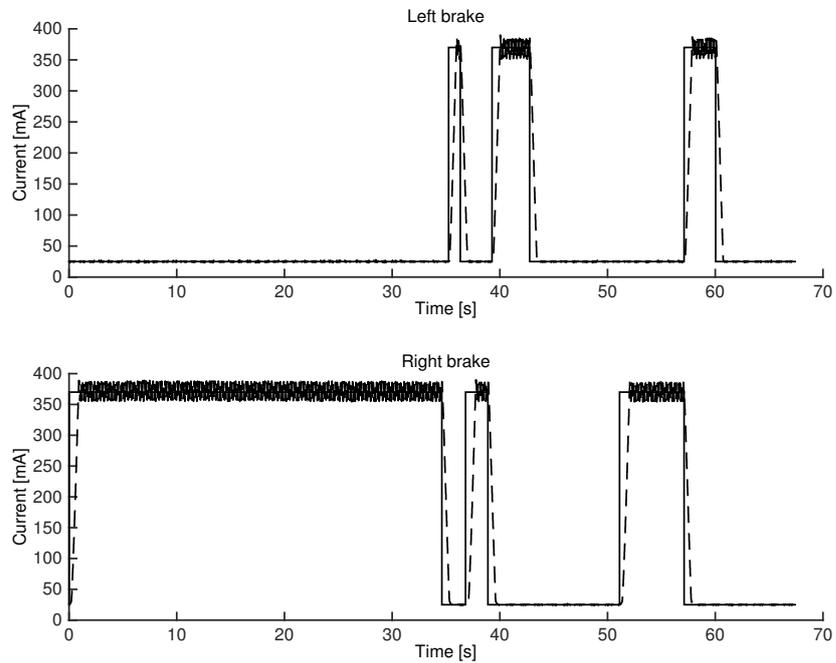


Figure 5.4: The braking action applied by the system in terms of commanded braking action (solid line) and the actual implemented action (thick dashed line) in the case of the  $S$  trajectory.

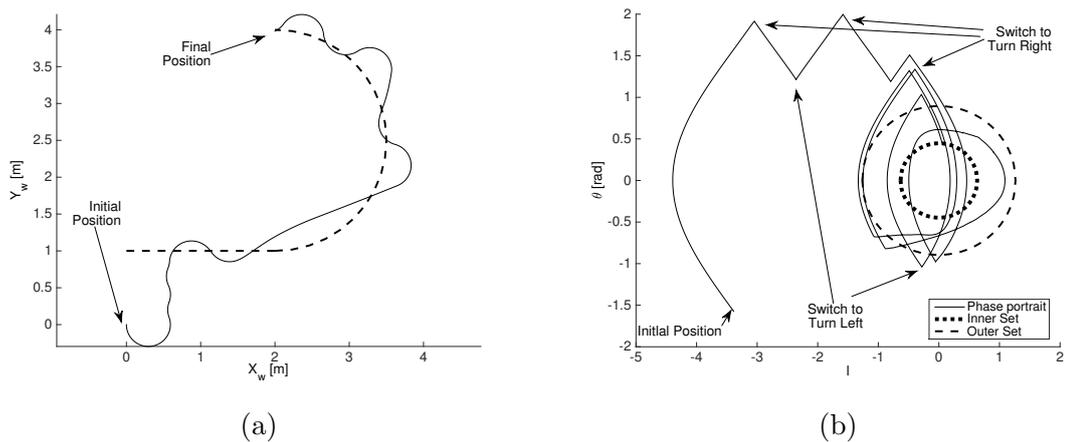


Figure 5.5: Experimental trajectory followed by the  $c$ -Walker (a) and phase portrait (b) in the case of a blind folded user following a  $U$  shaped path.



## Chapter 6

# Conclusion

We presented the development that led to the realization of a device for assisted living. The device, called *c-Walker*, is a smart device built around a common walker for elderly people.

We showed the difficulties that arise when building a robotic platform. The development focused on many aspects: hardware and software architecture for the basic functionalities and integration of high level software components. The hardware architecture is made of microcontrollers for control of low level components, and embedded platforms for the computation of more demanding algorithms. We use a Publish/Subscribe approach to handle communication and integration of different subsystems. The *c-Walker* is able to localise itself using on-board sensors or external services. We developed an EKF in such a way that we are able to select a configuration of sensors that meets our requirements of cost, accuracy, and robustness. We have presented four different solutions for guiding a user along a safe path. One of them is “active” meaning that the system is allowed to “force a turn” in a specified direction. The other ones are “passive” meaning that they merely produce directions that the user is supposed to follow on her own will. We have described the technological and scientific foundations for the four different guidance systems, and their implementation in the device.

The systems has been thoroughly evaluated with a group of volunteers. The study contributed a novel evaluation protocol for comparing the different guidance systems, and opens new challenges for interaction designers. The use of virtual corridors allowed us to test the precision of the guidance systems to maintain the correct trajectory in the absence of any visual indications of the route. However, in a real-life scenario, users would most likely walk along a wide corridor with walls on the left and right that might help maintaining a straight path in a particular part of the corridor (i.e., in the centre or towards the left/right). Moreover, corridors’ crossings are often orthogonal. In such scenarios, left and right instructions might be enough to allow the user to reach their

goal and the haptic solution could be the best trade off among precision, freedom and cognitive workload, leaving vision and audition free to perceive environmental stimuli. Future research, will repeat this study in more ecological contexts.

We have also presented a further control strategy based on a hybrid automaton that switches between a small number of possible behaviours (go straight, turn right, turn left). In this case, the mechanical actuation is based on brakes. This strategy confines the user within a “safe corridor” enclosing the path, guarantees a steady progress toward the destination (if the user does not oppose) and operates in a way that the user perceives as consistent with a few changes on the direction of motion.

The most important feature of the proposed solution is its possibility to drive the user without any measurement on the forces and on the torques she applies on the walker.

An important limitation of the proposed approach is that the virtual corridor is chosen of small size, the algorithm can determine oscillations around the path (especially when the path bends). If we narrow down the corridor, the system drives more smoothly, but the price to pay are frequent corrections, which could be annoying. This is due to an inevitable consequence of the lack of any measurement on the user’s applied force.

A promising strategy could be to combine the mechanical guidance with haptic or video signals communicating to the user the turn to take in advance. This way, the braking action would occur only as a “last resort” for distracted users and the followed path would be arguably smoother. Another potential problem on very smooth floors is the possibility that the wheel might slip reducing the precision of the manoeuvre and, potentially, the localisation accuracy. For this, we are currently studying a simple ABS strategy that should significantly alleviate the problem.

# List of Tables

3.1	Overview on Modalities. . . . .	20
3.2	Description of individual experiments conducted. . . . .	30
3.3	Relative Pose Errors (RPE) of the incremental modalities . . . . .	32
3.4	Self Localisation Errors (SLE) - Position in cm . . . . .	33
3.5	Self Localisation Errors (SLE) - Orientation in degree . . . . .	34
4.1	The structure of the periodic CAN messages used in the communication with the front wheels. . . . .	49
4.2	Average error (cm) for each experimental condition and significant post-hoc pairwise comparisons. . . . .	62
4.3	Average time (sec) for each experimental condition and significant post-hoc pairwise comparisons. . . . .	64
4.4	Average travelled length (m) for each experimental condition and significant post-hoc pairwise comparisons. . . . .	64
4.5	Average speed (m/sec) for each experimental condition and significant post-hoc pairwise comparisons. . . . .	65
4.6	Summary of the results. . . . .	69



# List of Figures

2.1	Views of the <i>c-Walker</i> with all the equipment (a) <i>c-Walker</i> seen from the front side, (b) <i>c-Walker</i> seen from the back side. . . . .	10
2.2	Front wheel: CAN bus node, motor and absolute encoder are visible. . . .	11
2.3	Back wheel: CAN bus node, brake and gear of the incremental encoder are visible. . . . .	11
2.4	Conceptual architecture of the <i>c-Walker</i> . . . . .	13
2.5	Functional diagram of the system where <i>c-Walker</i> 's internal modules and external modules are shown. . . . .	13
3.1	Model based Visual Localization Pipeline. Multiple overlapping Images are used to create a synthetic model of the environment. In the localization stage, 2D-3D correspondences are found to compute the camera pose within this model. . . . .	24
3.2	Room layout with placement of RFID tags (red) and placed waypoints (blue). . . . .	28
3.3	Impressions of the experimental setup. . . . .	29
3.4	Average and maximum error on position: localisation system with encoders and MVL. . . . .	36
3.5	Average and maximum absolute error on orientation: localisation system with encoders and MVL. . . . .	37
3.6	Average and maximum error on position: localisation system with Visual Odometry and MVL. . . . .	38
3.7	Average and maximum absolute error on orientation: localisation system with Visual odometry and MVL. . . . .	39
3.8	Average and maximum error on position: localisation system with all the componenets. . . . .	40
3.9	Average and maximum absolute error on orientation: localisation system with all the componenets. . . . .	41
3.10	Trajectory reconstruction using only incremental sensors and fusion of them. . . . .	42

3.11	Trajectory reconstruction using encoders or fusion of incremental sensors as prediction and RFID for updates. . . . .	42
3.12	Trajectory reconstruction using incremental sensors updated with MVL, and the fusion prediction updated with both MVL and RFID. . . . .	43
4.1	The vibrotactile bracelet equipped with two vibrating motors (A) attached to an elastic wristband (B). The Li-Ion battery and the Arduino board are in (C). . . . .	46
4.2	Graphic representation of the localisation of the <i>c-Walker</i> with respect to the path: (a) Frenet-Serret reference, (b) Virtual vehicle reference. In the Frenet-Serret reference frame, the vehicle always lies on the $y_d$ axis. . . . .	50
4.3	Graphic representation of the acoustic signal source: (a) <i>c-Walker</i> close to the path and sound projected ahead, (b) <i>c-Walker</i> too distant from the path and sound projected toward the Frenet-Serret point. . . . .	53
4.4	The block-scheme of the <i>c-Walker</i> architecture with its core components: mechatronic subsystem, localisation subsystem and planner. . . . .	57
4.5	The block-scheme of the three guidance systems. The three solutions interact with the <i>c-Walker</i> by means of the same software interface, but then are diversified in the way they perform the actuation. . . . .	57
4.6	Different metrics as function of Guidance and Path: (a) average error (cm), (b) average time (s), (c) average length (m), (d) average speed (m/s). . . . .	61
4.7	Some examples of qualitative results for different guidance algorithms: (a) mechanical, (b) haptic, (c) audio, (d) binaural. The trajectory of the user is the dash line. . . . .	63
4.8	Average error (cm) as a function of Guidance and Path. . . . .	68
5.1	Trajectory followed by the <i>c-Walker</i> (a) and phase portrait (b) in the case of an user roughly following the desired path. . . . .	79
5.2	The braking action applied by the system in the case of the trajectory of Fig. 5.1. . . . .	80
5.3	Experimental trajectory followed by the <i>c-Walker</i> (a) and phase portrait (b) in the case of a blind folded user following an <i>S</i> shaped path. . . . .	80
5.4	The braking action applied by the system in terms of commanded braking action (solid line) and the actual implemented action (thick dashed line) in the case of the <i>S</i> trajectory. . . . .	81
5.5	Experimental trajectory followed by the <i>c-Walker</i> (a) and phase portrait (b) in the case of a blind folded user following a <i>U</i> shaped path. . . . .	81

# Bibliography

- [1] A.P. Arias and U.D. Hanebeck. Wide-area haptic guidance: Taking the user by the hand. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 5824–5829, 2010.
- [2] Assistants for safe mobility (assam). <http://assam.nmshost.de/>.
- [3] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.
- [4] Andrea Balluchi, Antonio Bicchi, and Philippe Soueres. Path-following with a bounded-curvature vehicle: a hybrid control approach. *International Journal of Control*, 78(15):1228–1247, 2005.
- [5] Giuseppe Basile and Giovanni Marro. Controlled and conditioned invariant subspaces in linear system theory. *Journal of Optimization Theory and Applications*, 3(5):306–315, 1969.
- [6] DR Begault, EM Wenzel, AS Lee, and MR Anderson. Direct comparison of the impact of head tracking, reverberation, and individualized HRTFs on the spatial perception of a virtual speech source. *J. Audio Eng. Soc.*, 49(10):904–916, 2001.
- [7] Jens Blauert. *Spatial Hearing-Revised Edition: The Psychophysics of Human Sound Localization*. MIT press, 1996.
- [8] C.P. Brown and R.O. Duda. A structural model for binaural sound synthesis. *Speech and Audio Processing, IEEE Transactions on*, 6(5):476–488, 1998.
- [9] Herman Bruyninckx. Open robot control software: the orocos project. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2523–2528. IEEE, 2001.
- [10] Byoung-Suk Choi, Joon-Woo Lee, Ju-Jang Lee, and Kyoung-Taik Park. A hierarchical algorithm for indoor mobile robot localization using RFID sensor fusion. *IEEE Trans. on Industrial Electronics*, 58(6):2226–2235, Jun. 2011.

- [11] Byoung-Suk Choi and Ju-Jang Lee. Sensor network based localization algorithm using fusion sensor-agent for indoor service robot. *IEEE Trans. on Consumer Electronics*, 56(3):1457–1465, Aug. 2010.
- [12] M Yu Byron, Krishna V Shenoy, and Maneesh Sahani. Derivation of kalman filtering and smoothing equations. 2004.
- [13] John T. Cacioppo and Louise C. Hawkley. Perceived social isolation and cognition. *Trends in Cognitive Sciences*, 13(10):447 – 454, 2009.
- [14] Filippo Cavallo, Angelo M Sabatini, and Vincenzo Genovese. A step toward gps/ins personal navigation systems: real-time assessment of gait by foot inertial sensing. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1187–1191. IEEE, 2005.
- [15] Oscar Chuy, Yasuhisa Hirata, and Kazuhiro Kosuge. A new control approach for a robotic walking support system in adapting user characteristics. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 36(6):725–733, 2006.
- [16] Oscar Ylaya Chuy, Yasuhisa Hirata, Zhidong Wang, and Kazuhiro Kosuge. A control approach based on passive behavior to enhance user interaction. *Robotics, IEEE Transactions on*, 23(5):899–908, 2007.
- [17] Alessio Colombo, Daniele Fontanelli, Axel Legay, Luigi Palopoli, and Sean Sedwards. Motion planning in crowds using statistical model checking to enhance the social force model. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 3602–3608. IEEE, 2013.
- [18] Alessio Colombo, Daniele Fontanelli, David Macii, and Luigi Palopoli. Flexible indoor localization and tracking based on a wearable platform and sensor data fusion. *Instrumentation and Measurement, IEEE Transactions on*, 63(4):864–876, 2014.
- [19] Ulises Cortes, Cristian Barrue, Antonio B Martinez, Cristina Urdiales, Fabio Campana, Roberta Annicchiarico, and Carlo Caltagirone. Assistive technologies for the new generation of senior citizens: the share-it approach. *International Journal of Computers in Healthcare*, 1(1):35–65, 2010.
- [20] Davide Dardari, Andrea Conti, Ulric Ferner, Andrea Giorgetti, and Moe Z Win. Ranging with ultrawide bandwidth signals in multipath environments. *Proceedings of the IEEE*, 97(2):404–426, 2009.

- [21] M. De Cecco. Sensor fusion of inertial-odometric navigation as a function of the actual manoeuvres of autonomous guided vehicles. *IOP Measurement Science and Technology*, 14(5):643–653, May 2003.
- [22] Alessandro De Luca, Giuseppe Oriolo, and Claude Samson. Feedback control of a nonholonomic car-like robot. In *Robot motion planning and control*, pages 171–253. Springer, 1998.
- [23] The e-no-falls project. <http://www.e-nofalls.eu>.
- [24] J. B. F. Van Erp, H. A. H. C. Van Veen, Chris C. Jansen, and T. Dobbins. Waypoint navigation with a vibrotactile waist belt. *ACM Trans. Appl. Percept.*, 2(2):106–117, 2005.
- [25] Shih-Hau Fang, Tsung-Nan Lin, and Kun-Chou Lee. A novel algorithm for multi-path fingerprinting in indoor wlan environments. *Wireless Communications, IEEE Transactions on*, 7(9):3579–3588, 2008.
- [26] Zahid Farid, Rosdiadee Nordin, and Mahamod Ismail. Recent advances in wireless indoor localization techniques and system. *Journal of Computer Networks and Communications*, 2013, 2013.
- [27] Daniele Fontanelli, Antonio Giannitrapani, Luigi Palopoli, and Domenico Praticchizzo. Unicycle Steering by Brakes: a Passive Guidance Support for an Assistive Cart. In *Proceedings of the 52nd IEEE International Conference on Decision and Control*, pages 2275–2280, 10-13 Dec. 2013.
- [28] Daniele Fontanelli, Federico Moro, Tizar Rizano, and Luigi Palopoli. Vision-based robust path reconstruction for robot control. *Instrumentation and Measurement, IEEE Transactions on*, 63(4):826–837, 2014.
- [29] JB Gao and Chris J Harris. Some remarks on kalman filters for the multisensor fusion. *Information Fusion*, 3(3):191–201, 2002.
- [30] G. A. Gescheider, S. J. Bolanowski, K. L. Hall, K. E. Hoffman, and R. T. Verrillo. The effects of aging on information-processing channels in the sense of touch: I. absolute sensitivity. *Somatosens Mot Res.*, 11(4):345–357, 1994.
- [31] A. Ghosh, L. Alboul, J. Penders, P. Jones, and H. Reed. Following a robot using a haptic interface without visual feedback. In *Proc. Int. Conf. on Advances in Computer-Human Interactions*, pages 147–153, 2014.

- [32] Ismail Guvenc and Chia-Chin Chong. A survey on toa based wireless localization and nlos mitigation techniques. *Communications Surveys & Tutorials, IEEE*, 11(3):107–124, 2009.
- [33] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [34] Laura Hedley, Nicola Suckley, Lisa Robinson, and Pamela Dawson. Staying steady: A community-based exercise initiative for falls prevention. *Physiotherapy Theory and Practice*, 26(7):425–438, 2010.
- [35] Pieter Hintjens. *ZeroMQ: Messaging for Many Applications*. O’Reilly, 2013.
- [36] Y. Hirata, A. Hara, and K. Kosuge. Motion control of passive intelligent walker using servo brakes. *IEEE Transactions on Robotics*, 23(5):981–990, 2007.
- [37] Christof Hoppe, Manfred Klopschitz, Markus Rumpler, Andreas Wendel, Stefan Kluckner, Horst Bischof, and Gerhard Reitmayr. Online feedback for structure-from-motion image acquisition. In *Proceedings of the British Machine Vision Conference*, pages 70.1–70.12. BMVA Press, 2012.
- [38] The iwalkactive project. <http://www.iwalkactive.eu/>.
- [39] I. Karuei, K. E. MacLean, Z. Foley-Fisher, R. MacKenzie, S. Koch, and M. El-Zohairy. Detecting vibrations across the body in mobile contexts. In *Proc. Int. Conf. on Human Factors in Computing Systems*, pages 3267–3276, 2011.
- [40] Simeon Keates and John Clarkson. Countering design exclusion. In *Inclusive Design*, pages 438–453. Springer, 2003.
- [41] James Kennedy, James F Kennedy, and Russell C Eberhart. *Swarm intelligence*. Morgan Kaufmann, 2001.
- [42] Kay-Tee Khaw, Nicholas Wareham, Sheila Bingham, Ailsa Welch, Robert Luben, and Nicholas Day. Combined impact of health behaviours and mortality in men and women: The epic-norfolk prospective population study. *PLoS Med*, 5(1):e12, 01 2008.
- [43] Yungeun Kim, Yohan Chon, and Hojung Cha. Smartphone-based collaborative and autonomous radio fingerprinting. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(1):112–122, 2012.

- [44] Chun-Hsu Ko, Kuu-Young Young, Yi-Che Huang, and Sunil Kumar Agrawal. Walk-assist robot: A novel approach to gain selection of a braking controller using differential flatness. *Control Systems Technology, IEEE Transactions on*, 21(6):2299–2305, 2013.
- [45] Anja Kollmuss and Julian Agyeman. Mind the gap: why do people act environmentally and what are the barriers to pro-environmental behavior? *Environmental education research*, 8(3):239–260, 2002.
- [46] R.Hari Krishnan and S. Pugazhenth. Mobility assistive devices and self-transfer robotic systems for elderly, a review. *Intelligent Service Robotics*, 7(1):37–49, 2014.
- [47] V. Kulyukin, A. Kutiyawala, E. LoPresti, J. Matthews, and R. Simpson. iWalker: Toward a rollator-mounted wayfinding system for the elderly. In *Proc. IEEE Int. Conference on RFID*, pages 303–311, Las Vegas, NV, USA, Apr. 2008.
- [48] Lionel Lapierre and Didik Soetanto. Nonlinear path-following control of an AUV. *Ocean engineering*, 34(11):1734–1744, 2007.
- [49] Edward A Lee. Cyber-physical systems-are computing foundations adequate. In *Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*, volume 2. Citeseer, 2006.
- [50] Edward A Lee. Cyber physical systems: Design challenges. In *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, pages 363–369. IEEE, 2008.
- [51] Geunho Lee, Takanori Ohnuma, and Nak Young Chong. Design and control of jaist active robotic walker. *Intelligent Service Robotics*, 3(3):125–135, 2010.
- [52] Kai Lingemann, Andreas Nüchter, Joachim Hertzberg, and Hartmut Surmann. High-speed laser localization for mobile robots. *Robotics and Autonomous Systems*, 51(4):275–296, 2005.
- [53] David Macii, Alessio Colombo, Paolo Pivato, and Daniele Fontanelli. A data fusion technique for wireless ranging performance improvement. *Instrumentation and Measurement, IEEE Transactions on*, 62(1):27–37, 2013.
- [54] AKM Mahtab Hossain, Yunye Jin, Wee-Seng Soh, and Hien Nguyen Van. Ssd: A robust rf location fingerprint addressing mobile devices’ heterogeneity. *Mobile Computing, IEEE Transactions on*, 12(1):65–77, 2013.

- [55] Alain Micaelli and Claude Samson. Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. 1993.
- [56] AA Nazari Shirehjini, Abdulsalam Yassine, and Shervin Shirmohammadi. An rfid-based position and orientation measurement system for mobile objects in intelligent environments. *Instrumentation and Measurement, IEEE Transactions on*, 61(6):1664–1675, 2012.
- [57] P. Nazemzadeh, D. Fontanelli, D. Macii, and L. Palopoli. Indoor positioning of wheeled devices for ambient assisted living: A case study. In *Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014 IEEE International*, pages 1421–1426, May 2014.
- [58] Payam Nazemzadeh, Daniele Fontanelli, and David Macii. An Indoor Position Tracking Technique based on Data Fusion for Ambient Assisted Living. In *2013 IEEE Intl. Conf. on Computational Intelligence and Virtual Environments for Measurement Systems and Applications*, pages 7–12, Milan, Italy, 15-17 July 2013. IEEE.
- [59] Payam Nazemzadeh, Daniele Fontanelli, David Macii, Tizar Rizano, and Luigi Palopoli. Design and Performance Analysis of an Indoor Position Tracking Technique for Smart Rollators. In *Indoor Positioning and Indoor Navigation (IPIN)*, Montbeliard, France, 28-31 Oct. 2013. IEEE GRSS. To appear.
- [60] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2161–2168. IEEE, 2006.
- [61] <http://www.opendds.org>. Website.
- [62] OMG. Data distribution service for real-time systems – version 1.2. Technical report, The Object Management Group, 2007.
- [63] Stefania Pancanti, Lucia Pallottino, David Salvadorini, and Antonio Bicchi. Motion planning through symbols and lattices. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4, pages 3914–3919. IEEE, 2004.
- [64] Paschalis Panteleris and Antonis A Argyros. Vision-based slam and moving objects tracking for the perceptual support of a smart walker platform. In *Computer Vision-ECCV 2014 Workshops*, pages 407–423. Springer, 2014.

- [65] Sunhong Park and S. Hashimoto. Autonomous mobile robot navigation using passive RFID in indoor environment. *IEEE Trans. on Industrial Electronics*, 56(7):2366–2373, Jul. 2009.
- [66] T. Pilutti, G. Ulsoy, and D. Hrovat. Vehicle steering intervention through differential braking. In *Proceedings of the 1995 American Control Conference*, volume 3, pages 1667–1671, 1995.
- [67] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, 2009.
- [68] Luca Rizzon and Roberto Passerone. Embedded soundscape rendering for the visually impaired. In *Industrial Embedded Systems (SIES), 2013 8th IEEE International Symposium on*, pages 101–104. IEEE, 2013.
- [69] Luca Rizzon and Roberto Passerone. Spatial sound rendering for assisted living on an embedded platform. In *Applications in Electronics Pervading Industry, Environment and Society*, pages 61–73. Springer, 2014.
- [70] Peter J Rousseeuw and Christophe Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical association*, 88(424):1273–1283, 1993.
- [71] M. Saida, Y. Hirata, and K. Kosuge. Development of passive type double wheel caster unit based on analysis of feasible braking force and moment set. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 311–317, 2011.
- [72] G Santinelli, R Giglietti, and A Moschitta. Self-calibrating indoor positioning system based on zigbee® devices. In *Instrumentation and Measurement Technology Conference, 2009. I2MTC'09. IEEE*, pages 1205–1210. IEEE, 2009.
- [73] S. Scheggi, M. Aggravi, F. Morbidi, and D. Prattichizzo. Cooperative human-robot haptic navigation. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2693–2698, Hong Kong, China, 2014.
- [74] S. Scheggi, M. Aggravi, and D. Prattichizzo. A vibrotactile bracelet to improve the navigation of older adults in large and crowded environments. In *Proc. 20th IMEKO TC4 Int. Symp. and 18th Int. Workshop on ADC Modelling and Testing Research on Electric and Electronic Measurement for the Economic Upturn*, pages 798–801, 2014.
- [75] S. Scheggi, F. Morbidi, and D. Prattichizzo. Human-robot formation control via visual and vibrotactile haptic feedback. *IEEE Trans. on Haptics*, 2014.

- [76] Dmitry Sinyukov, Ross Desmond, Matthew Dickerman, James Fleming, Jerome Schaufeld, and Taskin Padir. Multi-modal control framework for a semi-autonomous wheelchair using modular sensor designs. *Intelligent Service Robotics*, 7(3):145–155, 2014.
- [77] Petr Smolik, Zdenek Sebek, and Zdenek Hanzalek. Orte—open source implementation of real-time publish-subscribe protocol. In *Proc. 2nd International Workshop on Real-Time LANs in the Internet Age*, pages 68–72, 2003.
- [78] D Soetanto, L Lapierre, and A Pascoal. Adaptive, non-singular path-following control of dynamic wheeled robots. In *IEEE Conf. on Decision and Control*, volume 2, pages 1765–1770. IEEE, 2003.
- [79] Jorge Torres-Solis, Tiago H Falk, and Tom Chau. *A review of indoor localization technologies: towards navigational assistance for topographical disorientation*. INTECH Open Access Publisher, 2010.
- [80] Cristina Urdiales, Jose Manuel Peula, M Fdez-Carmona, Cristian Barrué, Eduardo J Pérez, Isabel Sánchez-Tato, JC Del Toro, Francesco Galluppi, Ulises Cortés, Roberta Annichiarico, et al. A new multi-criteria optimization strategy for shared control in wheelchair assisted navigation. *Autonomous Robots*, 30(2):179–197, 2011.
- [81] Jelle Van Cauwenberg, Veerle Van Holle, Dorien Simons, Riet Deridder, Peter Clarys, Liesbet Goubert, Jack Nasar, Jo Salmon, Ilse De Bourdeaudhuij, Benedicte Deforche, et al. Environmental factors influencing older adults’ walking for transportation: a study using walk-along interviews. *Int J Behav Nutr Phys Act*, 9(1):85, 2012.
- [82] Darren ER Warburton, Crystal Whitney Nicol, and Shannon SD Bredin. Health benefits of physical activity: the evidence. *Canadian medical association journal*, 174(6):801–809, 2006.
- [83] Glenn Wasson, Pradip Sheth, Majd Alwan, Kevin Granata, Alexandre Ledoux, and Cunjun Huang. User intent in a shared control framework for pedestrian mobility aids. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 3, pages 2962–2967. IEEE, 2003.
- [84] S. Weinstein. Intensive and extensive aspects of tactile sensitivity as a function of body part, sex, and laterality. In *The skin senses*, pages 195–218. Erlbaum, 1968.
- [85] Kamin Whitehouse, Chris Karlof, and David Culler. A practical evaluation of radio signal strength for ranging-based localization. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(1):41–52, 2007.

- 
- [86] Christopher R Wilkinson and Antonella De Angeli. Applying user centred and participatory design approaches to commercial product development. *Design Studies*, 35(6):614–631, 2014.
- [87] Christopher R Wilkinson, Antonella De Angeli, et al. Demonstrating a methodology for observing and documenting human behaviour and interaction. In *DS 77: Proceedings of the DESIGN 2014 13th International Design Conference*, 2014.
- [88] Rui Zhang, F Hoflinger, and Leonhard Reindl. Tdoa-based localization using interacting multiple model estimator and ultrasonic transmitter/receiver. *Instrumentation and Measurement, IEEE Transactions on*, 62(8):2205–2214, 2013.
- [89] <http://zeromq.org>. Website.

