UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

RICARDO CARMINATI DE MELLO

UFES CLOUDWALKER: A CLOUD-ENABLED CYBER-PHYSICAL SYSTEM FOR MOBILITY ASSISTANCE

VITÓRIA, BRASIL
MARÇO 2018

RICARDO CARMINATI DE MELLO

UFES CLOUDWALKER: A CLOUD-ENABLED CYBER-PHYSICAL SYSTEM FOR MOBILITY ASSISTANCE

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Mestre em Engenharia Elétrica.
Orientador: Prof. Dr. Anselmo Frizera Neto
Co-orientador: Prof. Dr. Moises Renato Nunes Ribeiro

Vitória, Brasil
Março 2018

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Setorial Tecnológica,
Universidade Federal do Espírito Santo, ES, Brasil)
Sandra Mara Borges Campos – CRB-6 ES-000593/O

Mello, Ricardo Carminati de, 1990-

M527u UFES CloudWalker : a cloud-enabled cyber-physical system for mobility assistance / Ricardo Carminati de Mello. – 2018.

86f. : il.

Orientador: Anselmo Frizera Neto.

Coorientador: Moisés Renato Nunes Ribeiro.

Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Engenharia elétrica. 2. Engenharia biomédica. 3. Robótica – Reabilitação. 4. Robótica na medicina. 5. Equipamentos de autoajuda para deficientes. 6. Computação em nuvem. 7. Andador robótico. I. Frizera Neto, Anselmo. II. Ribeiro, Moisés Renato Nunes. III. Universidade Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 621.3

RICARDO CARMINATI DE MELLO

UFES CLOUDWALKER: A CLOUD-ENABLED CYBER-PHYSICAL SYSTEM FOR MOBILITY ASSISTANCE

Dissertação submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do Grau de Mestre em Engenharia Elétrica.

Aprovada em 01 de março de 2018.

COMISSÃO EXAMINADORA

Prof. Dr. Anselmo Frizera Neto
Universidade Federal do Espírito Santo - Brasil
Orientador

Prof. Dr. Moises Renato Nunes Ribeiro
Universidade Federal do Espírito Santo - Brasil
Co-orientador

Prof. Dr. Carlos Andrés Cifuentes Garcia
Escuela Colombiana de Ingeniaría Julio Garavito - Colômbia
Membro Externo

Prof. Dr. André Ferreira
Universidade Federal do Espírito Santo - Brasil
Membro Interno

Dedico este trabalho aos meus pais, Dayse e Claudio.
Tudo que sou, devo a vocês.

Acknowledgments

I wish to acknowledge the funding that made it possible for me to dedicate myself full time to the research that led to this work. Such help came from the European Commission H2020 program under the FUTEBOL grant agreement, as well from the Brazilian Ministry of Science, Technology, Innovation, and Communications through RNP and CTIC. This work was also supported by FAPES through the research project "Cloud Walker: Estratégias de Interação Humano-Ambiente e Computação em Nuvem para Marcha Assistida por Andador Robótico" (#80709036).

I would like to thank my advisors, Professors Anselmo Frizera Neto and Moises R. N. Ribeiro, for their guidance and teaching. This work would never be if it were not for their carefulness in sharing knowledge, experience, and thoughts with me. It has been a true privilege.

I want to extend my gratitude and thankfulness to all my friends and colleagues from NTA and NERDS. For all the work, knowledge, and laughter we shared, the time we have spent together encouraged me to become a better professional and human being. I'm also grateful to all the friends I made during my time at UFES, and grateful to the professors with the Electrical Engineering Department. In their own way, each of them has contributed to the underlying soul of this work.

Finally, I must thank my parents, Claudio and Dayse, my brother, Leonardo, and my partner, Katherine, for all their patience, support, and love. You inspire and strengthen me every day.

[Lawrence has just extinguished a match between his thumb and forefinger.
William Potter surreptitiously attempts the same]
William Potter: Ooh! It damn well 'urts!
T.E. Lawrence: Certainly it hurts.
Officer: What's the trick then?
T.E. Lawrence: The trick, William Potter,
is not minding that it hurts.
Lawrence of Arabia, Dir. David Lean. Columbia Pictures, 1962.

Resumo

Andadores inteligentes são dispositivos robóticos que fazem uso de uma série de sensores, atuadores e interfaces para prover funcionalidades de assistência à mobilidade para indivíduos com desordens de marcha. A crescente complexidade computacional demandada por algoritmos e técnicas recentes aplicados à robótica móvel, social e de saúde, comumente limitados pelo *hardware* embarcado do robô, aliados com avanços em tecnologias de rede e computação em nuvem deram origem ao paradigma da robótica em nuvem. Esta dissertação propõe, implementa e valida um sistema cyber-físico habilitado por computação em nuvem para assistência à mobilidade. O sistema, denominado UFES CloudWalker, tem por objetivo a integração entre andadores inteligentes e plataformas remotas de computação em nuvem, com foco na expansão das capacidades e funcionalidades oferecidas por estes dispositivos para usuários, pacientes, profissionais de saúde e familiares. O UFES CloudWalker explora conceitos de robótica em nuvem para expandir as capacidades dos andadores inteligentes apesar de possíveis limitações de *hardware*. Este trabalho apresenta um estudo sobre dispositivos de assistência à mobilidade, focado em andadores inteligentes, e um estudo em sistemas cyber-físicos e robótica em nuvem, com foco em suas aplicações na área de saúde. Ademais, também se discute desafios e potencialidades de combinar estes conceitos de sistemas cyber-físicos e robótica em nuvem aplicados a andadores inteligentes. Como parâmetros variáveis de rede e nuvem (e.g., latência fim-a-fim, perda de pacote, e disponibilidade) podem afetar tremendamente o controle desse tipo de sistema, dois grupos de experimentos são realizados para validar preliminarmente o sistema proposto e entender os impactos da qualidade de serviço prestado pela rede e pela nuvem sobre a avaliação do usuário final sobre sua qualidade de experiência. O sistema é implementado pela primeira vez através da integração entre um andador desenvolvido na UFES, o UFES Smart Walker, e uma plataforma de nuvem localizada em um *edge data center*. Para completar a validação do UFES CloudWalker, o sistema implementado é utilizado em um experimento realizado sobre um cenário de serviço de assistência à mobilidade. Os resultados obtidos mostram a viabilidade do sistema proposto mesmo sob condições desfavoráveis de qualidade de rede, abrindo as portas para o desenvolvimento de uma nova geração de andadores robóticos, integrados com a nuvem e capazes de lidar com um futuro sistema de saúde conectado.

Palavras-chave: CloudWalker. andadores inteligentes. robótica em nuvem. dispositivos assistivos. sistemas cyber-físicos.

Abstract

Smart walkers are robotic devices which rely on a series of sensors, actuators, and interfaces to provide mobility assistance functionalities to mobility impaired individuals. The increased computational complexity demanded by recent algorithms and techniques applied to mobile, healthcare, and social robotics, often limited by the robot's embedded hardware, together with advancements on networking and cloud computing enabled the so-called cloud robotics paradigm. This dissertation proposes, implements, and validates a cloud-enabled cyber-physical system for mobility assistance. The system, named UFES CloudWalker, envisions the integration of smart walkers and remote cloud computing platforms, aiming at expanding smart walkers' capabilities and the features those devices can offer to users, patients, medical staff, and family members. UFES CloudWalker explores cloud robotics concepts to unleash smart walker's capabilities despite possible hardware limitations. This work presents a study on mobility assistive devices, focused on smart walkers, and a study on cyber-physical systems and cloud robotics, focused on its applications in healthcare, and discuss the challenges and potentialities of combining cyber-physical systems and cloud robotics concepts concepts in smart walkers. Nevertheless, as network and cloud parameters (e.g.,end-to-end latency, packet loss, and availability) can largely affect control on such class of systems, two experiments sets are performed to preliminarily validate UFES CloudWalker's feasibility and to understand the impacts of the network and cloud quality of service over the end user's quality of experience. The system is implemented for the first time by integrating an in-house developed smart walker, the UFES Smart Walker, and a cloud platform located in an edge data center. To complete the validation of the UFES CloudWalker, the implemented system is used in an experiment set built over a mobility assistance service scenario. The results obtained argue for the feasibility of such system even under unfavorable network quality scenarios, opening a door for the development of a new generation of smart walkers, in which the devices are enabled by the cloud and able to cope with future connected healthcare systems.

Keywords: CloudWalker. smart walker. cloud robotics. assistive devices. cyber-physical systems. future healthcare.

Contents

List of Acronyms

CPS Cyber-Physical System

E2E End-to-End

HaaS Healthcare as a Service

HRI Human-Robot-Interaction

IaaS Infrastructure as a Service

IMU Inertial Measurement Unit

IP Internet Protocol

KTE Kinematic Tracking Error

LRF Laser Rangefinder

MOS Mean Opinion Score

NERDS Núcleo de Estudos em Redes Definidas por Software

NTA Núcleo de Tecnologia Assistiva

QoE Quality of Experience

QoS Quality of Service

ROS Robot Operating System

RSSI Received Signal Strength Indicator

SLAM Simultaneous Localization and Mapping

UFES Universidade Federal do Espírito Santo

uRLLC Ultra-Reliable Low-Latency Communication

VM Virtual Machine

VMR Variance to Mean Ratio

List of Figures

List of Tables

1 Introduction

1.1 Motivation

The world population is getting older and it is expected that in a few decades at least one in every five individuals will be more than 60 years old in most developing countries. In Brazil alone, the elderly accounted for 10% of the population in 2010 and are estimated to reach 20% *circa* 2030 [1]. Human aging is associated with problems that gradually affect mobility, which might lead to greater susceptibility to falls and accidents, loss of independence and also hamper performance on certain daily chores.

Mobility is an important human faculty and impacts one's capacity to freely move through multiple environments and to perform daily chores with ease [2]. Besides decreasing with age, neurological diseases also affect human mobility on different levels, and studies have found that mobility difficulties can be linked with increased mortality [3]. Scientific evidence also points to mobility restrictions also being associated with cognitive and psychosocial disorders, which further degrades independence and the quality of life of the individual [3].

Mobility assistive devices can be employed to mitigate mobility impairments. The use of each of those devices are recommended according to the type of condition affecting mobility and taking into account the level of impairment [2]. Technological developments led to the possibility of integrating robotics concepts, sensors, and circuitry on such devices, giving birth to a "smart" generation of mobility assistive devices. Smart wheelchairs [4], canes [5], and walkers [6] have been developed by multiple research groups over the last few decades to provide new or improved functionalities of physical support, and sensorial and cognitive assistance. Smart walkers, for instance, have been designed to improve pathological gait using support platforms for upper limbs and the exploitation of the user's own locomotion capabilities [7]. Recently, new features are being added to them ranging from health monitoring to physical and cognitive assistance, demanding ever-increasing embedded processing power.

The concept of cyber-physical systems is often applied to enhance the capacities of standalone systems. Cyber-physical systems are systems that combine sensing, communication, control, and computing to interact with a physical entity [8]. Such systems usually encompass physical elements, control and computation components, and communication elements, and its definition encloses multiple systems from various fields [9]. Despite the term not being directly linked to robotics, the concept is applied in networked and distributed robotics, in applications as teleoperation and telemedicine [8, 10].

The evolution of control algorithms within the robotics field, relying on great amounts of data and computational capabilities, combined with recent developments on communication and cloud computing technologies gave rise to the so-called cloud robotics paradigm [11]. On such paradigm, the individual robot does not rely solely on its own sensing and processing capabilities. It communicates with a remote computing platform to acquire and send data, instructions, and even control signals, while the cloud should be able to provide nearly unlimited storage space and processing power. This allows for the exploitation of techniques based on recent fields, such as big data and deep learning, on a larger range of service robots, also allowing for the use of computationally expensive control algorithms based on simultaneous localization and mapping, computer vision, trajectory planning, among others [12].

In the light of a paradigm shift, a new generation of robotic assistive devices shall arise. By leveraging the cloud capabilities, new control techniques will be applied on multi-robot environments, in which assistive devices should be able to learn in a faster pace and adapt to the ever-evolving impairment condition of different users. Over such scenario, mobility assistive devices should evolve to offer new features to users, patients, therapists, and caretakers. Nevertheless, the impact of relying on communication networks and remote platforms must first be evaluated. Quality of service constraints may render unfeasible such paradigm shift for devices with real-time requirements that physically interact with impaired and often frail individuals.

This work discuss the challenges and potentialities of combining cyber-physical systems and cloud robotics concepts in smart walkers. Moreover, a cloud-enabled cyber-physical system for mobility assistance, the UFES CloudWalker, is presented. This system envisions the integration of smart walkers and remote cloud computing platforms, aiming at expanding smart walkers' capabilities and the features those devices can offer to users, patients, medical staff, and family members. This work also presents UFES CloudWalker's validation experiments. Results obtained from experimentation argue for the feasibility of such system even under unfavorable network quality scenarios, opening a door for the development of a new generation of smart walkers, in which the devices are enabled by the cloud and able to cope with future connected healthcare systems.

1.2 Background

This work was carried out on the Assistive Technology and Robotics Group (NTA) and the Software Defined Networks Study Group (NERDS), both linked to the Graduate Program of Electrical Engineering at the Federal University of Espírito Santo (UFES).

Over the last decade, NTA has conducted research on smart walkers for assistance and rehabilitation purposes, mainly focused on human-robot interaction (HRI) and human-robot-environment interaction (HREI). An in-house developed smart walker, the UFES Smart Walker, is central to several published papers, in which multiple interaction techniques, interfaces, and control algorithms have been developed and demonstrated.

The NERDS group develops research on cloud computing and programmable data center networks. The group also participates on FUTEBOL, an EU-Brazil cooperation project under the European Commission H2020 program with research focus on wireless-fiber integration. Researchers at NERDS possess the expertise in managing cloud platforms and physical infrastructure resource, aiming at providing optimized levels of service for applications being executed on the cloud.

In this context, this work aims at discussing possible benefits and challenges of combining cloud robotics concepts in mobility assistive devices. This work presents a study on mobility assistive devices, focused on smart walkers, and a study on cyber-physical systems and cloud robotics, focused on its applications in healthcare, while discussing the challenges and potentialities of applying cyber-physical systems and cloud robotics concepts to smart walkers.

A cloud-enabled cyber-physical system for mobility assistance, the UFES CloudWalker, is proposed exploring resources and expertise from both NTA and NERDS. UFES CloudWalker's overall system and architecture are described and detailed. For not relying solely on local computation, the proposed system is susceptible to variable network and cloud conditions. Quality of service indexes involving communication and remote computation, such as end-to-end latency, information loss, and availability, can largely affect control and stability on cloud-enabled devices. Therefore, two experiments sets are performed to preliminarily validate UFES CloudWalker's feasibility and to understand the impacts of the network and cloud quality of service over the end user's quality of experience when making use of mobility assistance services. During this preliminary validation stage, the system is considered to be on its most vulnerable configuration, in which no processing is retained and all control algorithms are remotely executed.

CloudWalker is implemented for the first time by integrating the UFES Smart Walker and a cloud platform located the NERDS data center. The walker is adapted to be able to communicate to remote virtual machines running on the cloud, exploring different wireless and wired networks. To complete the validation of the UFES CloudWalker, the implemented system is tested against an experiment set built over a mobility assistance service scenario. Results and future directions are presented and discussed by the end of this work.

1.3 Objectives

The general objective of this work is to propose, develop, and implement a cloud-enabled cyber-physical system for mobility assistance. The system is composed by the integration of a smart walker and services executed in a remote cloud platform. The smart walker is intended to acquire data from sensors, communicate with the remote platform, and execute remotely executed control signals. The cloud platform receives the sensorial information, process the data, execute control algorithms and sends back to the smart walker the control signals to command its actuators.

The objectives of this work can be listed as:

- To perform a literature study regarding smart walkers;
- To perform a literature study regarding cloud robotics and its applications in healthcare;
- To design a system architecture integrating a smart walker and a cloud computing platform;
- To enable communication between a smart walker and a cloud computing platform;
- To develop control services to be remotely executed;
- To validate the real-time execution of the overall implemented system.

1.4 Dissertation Overview

After this introduction regarding the main concepts addressed in this work, the next chapter presents a study on mobility assistive devices focusing on smart walkers. Conventional devices used to provide support and assistance to mobility impaired individuals are presented. Conventional and smart walkers are further discussed and a literature review regarding smart walkers' functionalities is presented.

Chapter 3 introduces cyber-physical systems and cloud robotics concepts. A literature review on cloud robotics is presented, focusing on its applications in the healthcare field. The chapter ends with a discussion regarding potential benefits and challenges of exploring cloud robotics concepts in mobility assistive devices.

The UFES CloudWalker system is proposed in Chapter 4. The overall architecture is presented and the resources used for its implementation are shown. Preliminary validation studies are presented to discuss the impacts of variable network and cloud conditions over the proposed system focusing on the end-user point of view.

Chapter 5 details the first implementation of the UFES CloudWalker. The system is validated against a mobility assistance scenario, in which all control applications are remotely executed. Results are shown and discussed, pointing to the feasibility of employing the UFES CloudWalker to assist mobility impaired individuals.

The last chapter, Chapter 6, presents the conclusions and contributions from this work, together with a list of resulting publications. Future work is discussed to point and suggest developments and increments to the UFES CloudWalker, aiming at exploring the full possibilities of the system.

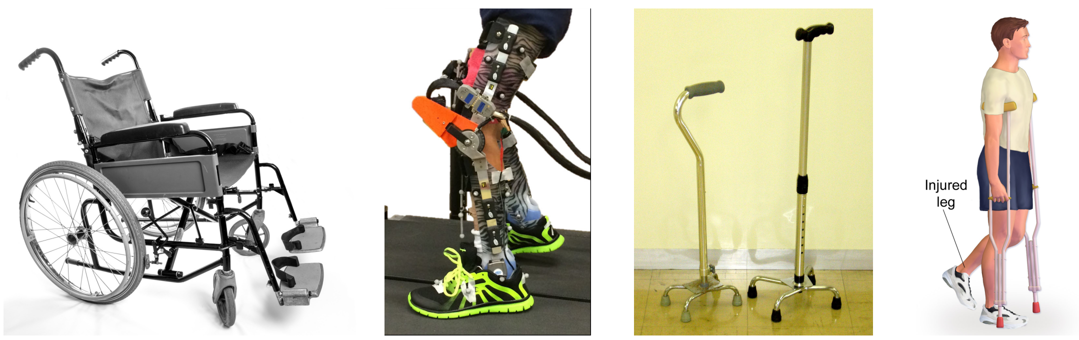
2 Technical Aids for Human Locomotion

2.1 Mobility Assistive Devices

Human mobility can be affected by multiple conditions and the causes for degraded mobility are seldom associated to only one disease [13]. Individuals of advanced age are more prone to suffer from mobility disorders due to problems often associated with aging. Therefore, gait disorders – which can be characterized by slower and esthetically abnormal gait – are particularly important for the elderly population [14]. Such disorders do not impact solely on the individual's independence and ability to perform daily chores but can also be a major risk factor for fall and injury. Reduced activity can lead to a decrease in muscle tone and strength, decreasing physical function, and can also reduce social, emotional, and cognitive function [13].

In this context, it is important to develop devices that can offer support and assistance to mobility impaired individuals. Indeed, multiple assistive devices have been studied and developed targeting different impairments conditions [15]. These solutions are selected by a therapist based on the condition and the degree of disability of the user [6].

Assistive devices can be classified as alternatives, in case of total incapacity of mobility, and augmentative, employed when the user has residual mobility capacity [16]. Among alternative devices, wheelchairs (e.g., Fig. 1(a)) and solutions based on autonomous special vehicles are the most common ones. Augmentative devices are employed whenever possible to stimulate residual mobility and improve physical and cognitive capabilities. Those devices can be categorized as mobility-training devices, such as parallel bars used in therapy, self-ported devices, such as orthoses (e.g., Fig. 1(b)), and external devices. Among the latter, canes, crutches (e.g., Fig. 1(c)-(d)), and walkers are most commonly employed [6].



(a) (b) (c) (d)

Figure 1 – Examples of mobility assistive devices: a) manual wheelchair; b) active orthoses; c) four-legged canes; d) auxiliary crutches.

2.2 Walkers

Walkers are external devices characterized by their structural simplicity, low cost and rehabilitation potential [17]. These devices provide support during bipedestation while leveraging the individual's locomotion capacity, stimulating lower-limbs musculature and avoiding deterioration of physical capacities. Therefore, walkers are used whenever possible to avoid early and deteriorative use of wheelchairs [6]. In addition, there is evidence pointing that walker-assisted gait is related to psychological benefits as increased confidence and safety perception during ambulation

There are many types of established walkers in the market and walkers developed by research groups. These various devices differ in constitutive materials, structural configuration, accessories, and offered functionalities. In general, walkers can be classified in two types: conventional walkers and smart walkers [6].

2.2.1 Conventional Walkers

Conventional walkers (e.g., Fig. 2) are classified in respect to their ground contact configuration. Considering the different kinds of these devices, they can be classified in three groups [15]: standard, front-wheeled, and rollators.

(a) (b) (c)

Figure 2 – Walker frames. a) standard; b) front-wheeled; c) rollator.

The standard walker, depicted in Fig. 2(a), is the most common configuration among conventional walkers. It is based on a four-legged rigid metal frame with rubber tips on the end of each leg. Due to such form, it is considered the most stable configuration, being employed when maximum balance assistance is required or when there are weight bearing restrictions [6]. The standard walker requires some degree of upper body strength, as the user must lift the walker on each stride, and also its use also results on an unnatural gait pattern. Such extra level of force required drastically raises energy expenditure, restricting the use of standard walkers for patients that also present severe levels of metabolic, cardiac, or respiratory dysfunctions [18]. As such walker configuration also demands cognitive ability to maneuver, there is still risk of falling – even backwards [19].

Front-wheeled walkers, Fig. 2(b)), are a variation of standard walkers, characterized by the presence of two wheels on the front legs, which affects ground contact [17]. These devices are indicated for individuals with weaker lower limbs or that tend to fall backwards while lifting standard walkers [6]. The front wheels allows for a more natural gait pattern, but also diminish dynamic stability during walking when compared to the standard model [15].

Rollators (also called four-wheeled walkers), Fig. 2(c), present wheels attached on the four legs of the walker and brakes on the handlers. This device can be used when the patient does not require higher levels of weight support, being the easiest to use and preferred by most patients [19]. The use of rollators allows for faster locomotion and more natural gait patterns. However, such devices are considered the most unstable version among the walkers by the increased risk of falling when full body-weight support is needed [17].

2.2.2 Smart Walkers

The use of conventional walkers provides enhanced support and weight bearing for the individual, but it does not come without disadvantages. Some degree of cognitive abilities is required to maneuver the walker in tight spaces and to command the brakes (in the case of rollators). The use of walkers also reduct normal arm swing and might lead to poor posture [2]. To solve such issues, research groups developed solutions encompassing the integration of robotic concepts on conventional walkers, giving birth to the so-called smart walkers.

Usually based – or inspired – on rollator frames, smart walkers integrate embedded computational systems, sensors, and actuators to mitigate the disadvantages of conventional walkers and to provide a whole new range of functionalities. Due to their design, smart walkers are capable of offering better support while maintaining safety during locomotion [20]. Some relevant examples of smart walkers from the literature are shown in Fig. 3, such as the PAMM [21], a classic smart walker able to monitor patient's vital signs, and the one presented in [22], built to guide visually impaired individuals. The functionalities offered by these smart walkers, among others, will by further discussed in the following section.

As individuals that require mobility assistance have multiple specific needs, smart walkers have evolved to provide different functionalities. These intelligent devices can present features such as obstacle avoidance assistance, guidance during navigation, gait monitoring, and fall prevention [6]. Usually, they can present five different functionalities [6]: physical support; sensorial assistance; cognitive assistance; health monitoring; and advanced human-robot interface. These functionalities are further discussed in the next section together with a literature review.

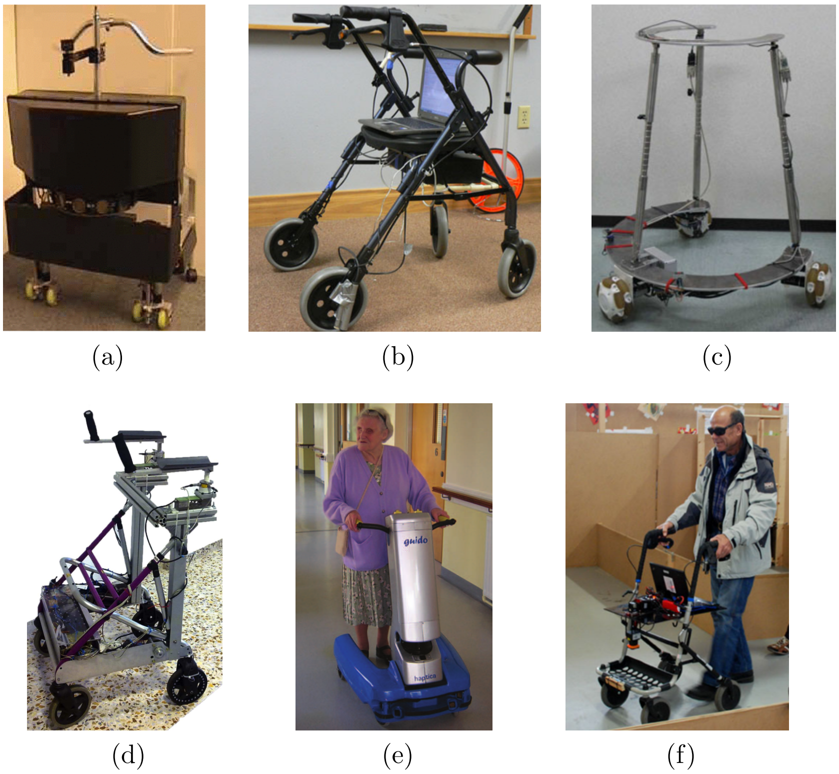


Figure 3 – Some relevant examples of smart walkers in the literature: a) PAMM [21]; b) iWalker [23]; c) JAROW [24]; d) SIMBIOSIS [25]; e) Guido [26]; f) smart walker presented in [22].

2.3 Smart Walkers: Literature Review

Given the nature of smart walkers and the conditions of those who use these devices, physical support functionalities are of great importance. Almost all smart walkers in the literature present some kind of physical support function among its characteristics [2]. There are mainly two types of physical assistance [6]: passive and active.

Passive physical support is usually accompanied by mechanical or structural enhancements to improve stability during gait, such as the enlargement of the base frame or the balanced placement of heavy elements at lower planes [2]. Other passive change is the replacement of the conventional handlebars of the walker by forearm support platforms, which offers a better coupling between user and walker [27]. Smart walkers such as the ASAS [28], SIMBIOSIS [27], ASBGo [29], and the UFES Smart Walker [17] are examples of devices which rely on forearm support platforms. Clinical tests showed that these supports eliminate the degree of freedom of the elbow articulation and also improve weight bearing, easing the handling of the device and reducing risks of glide [2].

Passive breaking strategies are also commonly applied to improve gait stability by controlling the device's motion. As breaking on conventional walkers requires some level of cognitive ability, motor coordination and dexterity, there is a risk of fall due to excessive acceleration. To mitigate such problem, some smart walkers – sometimes referred to as passive smart walkers – utilize automated breaking control strategies. These models do not present actuators to insert energy on the system, being literally pushed by the user. The ability to perform automated break control allows for adapting the device's apparent dynamics to each individual user [30], fall prevention [31], and even guidance [22].

Active physical support functionalities are explored in several smart walkers. Usually, motors are installed on the wheels of these devices to provide energy to the system, by means of controlling wheels velocities, or to simply control wheels direction [6]. These devices usually make use of advanced human-robot interfaces to detect, infer, or interpret user's commands and desires. Such interfaces can then translate those perceived signals into motor actuation [2].

Sensorial assistance functionalities are usually designed to assist navigation on environments with multiple obstacles or to help users with visual impairments [2]. Normally, such features employ ultrasonic [32–34], infrared [33, 35], vision [36, 37], laser rangefinder (LRF) [22, 33, 38], or even RFID [23] sensors to detect static and dynamic obstacles. The control system can assist the user in obstacle avoidance by sound or haptic signals, or even by directly acting upon the wheels, changing the path introduced by the user.

The PAM-AID [39] is a passive smart walker aimed to guide the elderly blind. Aware of the environment, such walker took as inputs user's desires (e.g., move forward or turn left) to guide through corridors, providing warnings of nearby obstacles. In later work [40], it was also able to steer the user around those obstacles. Point-to-point guidance was resolved on Guido [26], an evolution of the PAM-AID, which was able to guide the user to specific places.

On a more recent work, Wachaja et al. attempts to navigate people with visual and walking impairments using an off-the-shelf passive walker retrofitted with laser sensors [22]. It relied on two vibro-tactile interfaces to provide navigational information. The walker presented one operation mode in which the user was simply informed about surrounding obstacles, and another operation mode designed to guide the user to a specific location.

Obstacle avoidance features were also tested on passive smart walkers on experiments with blindfolded subjects. In Hellström et al., the walker's breaks were employed as means of influencing the direction of motion for obstacle avoidance [35], while in Hsieh et al., the breaks were used to assure that the user would remain over a pre-established path [41].

Besides passive strategies, active strategies were also employed on guidance. The Care-O-Bot II [42] is a walker capable of guiding the user throughout a established path while respecting user's inputs. Morris et al. presented the XR4000 [33], a walker able to weigh user's force inputs against a desired path. Other relevant examples of smart walkers that employ active strategies are the COOL Aide [43], and the PAMM [21]. Those strategies usually involve shared-control algorithms to still offer some degree of freedom to the user.

Cognitive assistance techniques usually encompass navigation and auto-localization.

Such features are especially important for people that present cognitive issues and problems related to memory loss and disorientation [2]. Patients with neurological impairments (e.g., Alzheimer, multiple sclerosis, and stroke victims) are also part of the target group for cognitive assistance [44]. Some smart walkers are able to perform simultaneous localization and mapping (SLAM) to place themselves in the environment, to plan trajectories, and to guide individuals from one place to another. When the destination, which can be set by the user or the caretaker, is known *a priori*, the walker may establish a path to be followed over a previously uploaded map, in the case of structured environments, or a dynamically generated map, when SLAM algorithms are used.

Guido [26] employed a SLAM algorithm based on laser sensors to build a map of the environment, which was used to plan a trajectory to be followed. The i-Walker [45] could present a map in a screen, allowing the user to chose a destination to be guided to. Devices such as the XR4000 [33] and the one presented by Wachaja et al. [22] are also able to use SLAM algorithms to guide users. In Panteleris and Argyros [36], a RGB-D camera is employed to perform SLAM and the tracking of moving objects. Such navigation systems usually incorporate sensorial assistance features, such as obstacle tracking and avoidance, to allow for a safe locomotion.

Some smart walkers can also be used to monitor health parameters of the user. This information can be used to keep a medical history or to detect emergency situations. For instance, the PAMM [46] presented continuous health monitoring sensors to detect changes in health trends through an ECG-based pulse monitor. The smart walker presented by Chan and Green [47] could perform physiological monitoring employing a cardiac monitor and a blood oxygenation monitor, and the one presented by Zhang et al. [48] collected pulse data to feed a web-based application.

The elements that enable interaction between human and walker are called interfaces. Such interaction usually relies on advanced human-robot interfaces to be perceived as user-friendly and natural [6]. Human-robot interaction techniques may depend on different interfaces, such as touch screens, joysticks, tactile sensors, inertial measurement units (IMU), and force/torque sensors [17].

Interfaces can be classified as either direct or indirect. Direct interfaces are the ones in which are passed directly to the device. This can be achieved via joysticks [49], voice communication [50], pressure or force sensors [17, 26, 33, 51], among others. Differently, indirect interfaces are used to infer user's movement and intention without direct human interference. Examples of such interfaces applied to smart walkers are computational vision [38, 52, 53], infrared sensors [24], ultrasonic sensors [2], and LRF [7, 54].

On the next chapter, cyber-physical systems and cloud robotics concepts will be introduced together with a discussion on how mobility assistive devices can benefit from the exploitation of such concepts. Those possibilities are further discussed for smart walkers in particular, and how can their features and capabilities be expanded by emerging communication and computation technologies.

3 Cyber-Physical Systems and Cloud Robotics

Robotics has been increasingly applied to common everyday devices in order to expand their capabilities and relieve people from tasks in which human intervention is no longer needed. From autonomous mail delivery vehicles to robotic museum guides, it is expected that several types of facilities outside industry will rely on such robotic employees working alongside – and even cooperating with – human staff. Healthcare is one of the areas in which robotics is being applied, and future medical facilities shall be equipped with devices such as robotic caretakers and autonomous stretchers, which will be also sharing space with patients and health professionals. The capabilities of the current generation of assistive and rehabilitation devices are enhanced by the integration with robotics: Exoskeletons are being used in physiotherapeutical treatment and traditional devices, such as walkers and wheelchairs, are now smart systems capable of interacting with patients and surrounding environment.

The last generation of these technologies can be classified into cyber-physical systems (CPS) as they combine sensing, communication, control, and computing to interact with a physical entity [8]. According to Bordel et al. [55], there are conflicting definitions of the CPS paradigm. Nevertheless, in this work the term CPS shall be understood as the integration of six components: physical world, transducers, control components, data analytics elements, computation elements and communication components.

Many research works have been conducted on CPS architectures, and Liu et al. [8] separate CPS's closed loop control in three layers. The first layer comprises the physical system (i.e., the hardware of the device, including sensors, actuators, among others). The second comprises the information system (i.e., real-time control and other data analytics elements). Lastly, the third layer is the user layer, comprising the way the user interacts with the CPS.

The CPS definition encompasses a variety of systems throughout different engineering fields, from cochlear implants to industrial plants, and the integration of the different CPS components intensified recently [9]. In other words, multiple systems that relied solely on embedded hardware have been migrated to distributed systems employing CPS's concepts [56]. In that way, computational tasks such as monitoring and control can be performed by remote platforms, which makes it possible to diminish the computational capacity available on the physical local device. Moreover, such remote computing platforms make it viable the utilization of high processing control algorithms, the access to large quantities of stored data, and the proper scaling when new applications or devices arise.

The exploitation of cloud computing concepts expands CPS capabilities and gives rise to the so-called cloud robotics systems [57]. According to Wan et al. [11], big data and other emerging computing and communication technologies allows for the integration of cloud technology and multi-robot systems, making it possible to implement such systems with improved energy efficiency, real-time performance, and low cost. This new paradigm allows for the construction of robots with simpler and smaller hardware, but with access to platforms with nearly unlimited capability of data processing and storage, making it viable to employ less expensive robots in compute-intensive unstructured tasks [58].

Kehoe et al. [59] cite four areas in which recent growth fosters the potential for cloud utilization:

- big data, providing access to remote libraries of images, maps, and general data;
- cloud computing, providing access to on demand parallel computing for statistical analysis, learning, and control;
- collective robot learning, with robots sharing trajectories, control policies, and outcomes;
- human computation, by using crowdsourcing to assess remote human expertise.

Wan et al. [11] cite open source applications as motivating factors for cloud robotics and highlight the use of the Robot Operating System (ROS) and RoboEarth initiative. ROS is a framework known to the scientific community and is presented as a meta-operating system [60]. It provides services as hardware abstraction, low-level device control, and package management in a peer-to-peer manner, making use of processes running in nodes that communicate in a publisher/subscriber architecture. ROS also provides libraries for obtaining, writing, and running multi-lingual code across multiple computers.

RoboEarth is a website dedicated to provide services to robotics systems [61]. It works as a big database, gathering data about grasping control, navigation, intelligent services, and other applications, from where robots can learn from other robots experience. RoboEarth works together with innitiatives such as Rapyuta, an open-source cloud robotics platform that helps robots to offload heavy computation to the cloud, also allowing for the easy access to the RoboEarth knowledge repository [62].

The exploitation of the so-called cloud robotics paradigm can expand the capabilities and features offered by assistive robots. Intelligent devices requires ever-increasing processing power to enable new functionalities and complex applications, impacting on computational costs involved. Algorithms responsible for tasks as fall prevention or guidance demands some degree of complexity to provide the safety and reliability needed. Migrating robot's processing capabilities to the cloud leads to lighter and less expensive robots [63]. Furthermore, through the access of big data sets, massively parallel computing, and the sharing of actions and outcomes, heavy adaptive control algorithms can be employed in order to command multiple connected robots in dynamic environments of healthcare facilities [11].

3.1 Cloud Robotics in Healthcare: Literature Review

Healthcare is an intricate field and can benefit from various technologies, robotics being on of them. It is foreseen that cloud computing can be exploited in healthcare initially in simple tasks, as exploiting multiple networked sensors for remote patient status monitoring, moving forward to more advanced tasks, that can be as demanding as remotely controlled robotic surgery.

In current literature, most cloud-enabled solutions explore networked sensors, wearable sensors and clothing, and wireless body area network concepts. Hossain [64] uses smartphones to perform patient localization in a cloud-supported CPS for patient monitoring. In such work, smartphones are the devices responsible for acquiring voice and electroencephalogram signals in real-time, feeding the cloud application with data. Zhang et al. [65] presented a CPS for patient-centric healthcare applications and services is proposed. This system collects data from wearable and household devices and leverages distributed storage and parallel computing to provide healthcare services. Chen et al. [66] present a similar approach of gathering patient data and connecting stakeholders through a cloud platform, though focused on the end-devices used for data collection. In such work, the design of the so-called smart clothing is discussed as a means to provide pervasive intelligence for healthcare systems.

Ontologies and databases are also being constructed on the cloud. Dogmus, Erdem and Patoglu [67] discuss the design and development of a ontology for rehabilitation robotics. Also with implications to rehabilitation robotics, Radu, Candea and Candea [68] present a cloud-enabled application to monitor and store data from an assistive exoskeleton. Such work is focused on human-robot interaction addressing exoskeletons for industrial needs, nevertheless most of the discussion also holds for rehabilitation purposes.

Li et al. [69] integrate a robot used for upper-limb rehabilitation to a cloud platform to allow the physician to remotely oversee therapy and adjust parameters. A cloud-based monitoring application can be accessed through the Internet, displaying both video and audio information, together with kinematic and physiological data for analysis. Moreover, libraries on the central server can be employed to give advice based on the training data.

There are also initiatives in social robotics to design cloud-enabled companion and caregiver robots for the elderly. Fiorini et al. [70] present a personal health management service, combining cloud computing services, a domestic robot, an Android app and a web portal. Services as localization and speech recognition are provided and the robot is able to reach and interact with the user.

Assistive mobility devices are beginning to benefit from cloud computing concepts. Fu et al. [71] use smartphone sensors to collect navigational information of a wheelchair and transmit them to the cloud for storage. Salhi et al. [72] integrate robotics concepts to the cloud to share multiple wheelchair positions and the map of the environment to feed the local controllers of the wheelchairs.

There are not many works exploring the integration of cloud computing and smart walkers. Benavidez et al. [73] propose and simulate a multi-robot system integrated with cloud data center. In such work, a conventional rollator was equipped with a Microsoft Kinect sensor and the cloud was responsible for data processing. Presented as a prototype of a smart walker, no functionalities other than the ones inherent to the rollator were presented.

A thorough search in the literature regarding smart walkers yielded ANG-MED robot [74] as the only smart walker currently making use of cloud-based services. It is a passive smart walker built over a commercially available 4-wheels rollator with independent brakes that can be used to stop the walker or to correct its trajectory during obstacle avoidance routines. The ANG-MED was developed within the RAPP project [75], which aims to provide cloud-based applications for robots in general. RAPP was a 3-year research project (2013-2016) funded by the European Commission that developed an open source software platform to support the creation and delivery of robotic applications. The goal was to increase the versatility and utility of service and assistive robots, providing generic robotics functionalities from a cloud platform. Until now, the only consumer robot supported by the RAPP platform is NAO, by SoftBank robotics. The ANG-MED walker is also supported by the platform, that also aims to respond to the intentions and needs of people at risk of exclusion. Tsardoulias et al. [75] enabled the caregiver to remotely monitor and act upon ANG-MED's brakes. This smart walker has been evaluated by care professionals with positive results, but there is no analysis of how overall aspects of the network might affect its use, specially when the caregiver must activate the brakes of the system. Moreover, there is no comprehensive discussion on how to assure the safety of such application.

3.2 A Paradigm Shift for Assistive Devices: Challenges and Potentials

Placing the Cloud at the center of a connected healthcare system can enable a systematic use of multiple heterogeneous devices at different facilities, allowing for the so-called Healthcare as a Service (HaaS) to arise. Figure 4 depicts the big picture of such system, where robotic devices exploits the use of the cloud to provide services for patients and to feed databases with information that can be used by remote libraries, other robots, adaptive control algorithms, and health reports generators to family, doctors, and caretakers. Local and edge data centers can be exploited for applications with low latency requirements

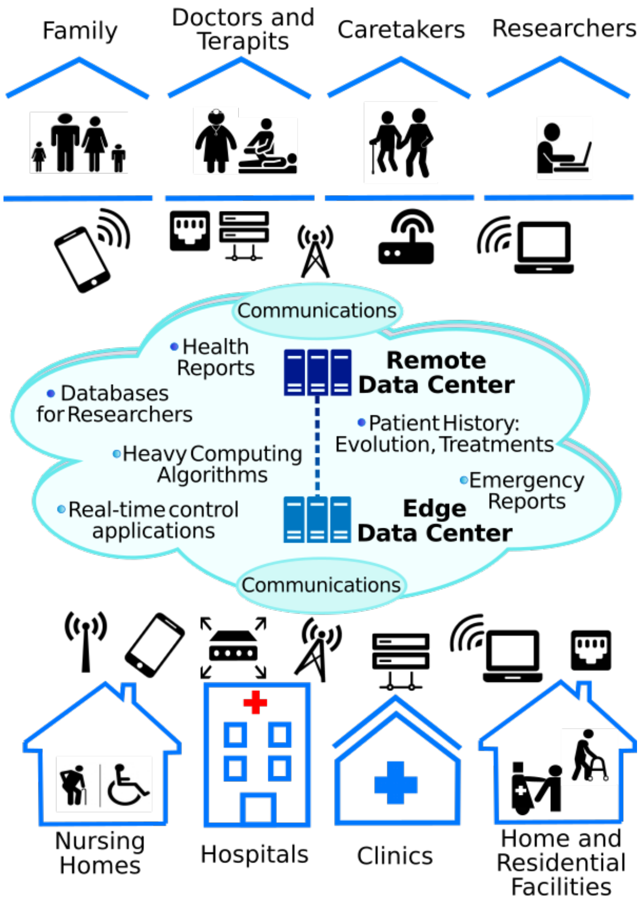


Figure 4 – Big picture overview of a healthcare system enhanced by networking and cloud computing technologies.

while delegating to remote clouds tasks such as databases construction and even update of global maps using multiple robots inputs.

While the use of robotic systems for rehabilitation is usually within ambulatory care, assistive devices may be employed without any direct supervision and the user must rely on the device alone. Such assistive robots, apart from most robotic systems, have to interact closely with patients - on both cognitive and physical levels - to provide functionalities where reliability and safety are essential [76]. On current paradigm, hardware and resources constraints are imposed by embedded computing power, so are data storing and learning capabilities. By leveraging cloud computing technologies, such constraints can be drastically diminished. In this context, cloud-enabled CPS for patient mobility assistance will face challenges related to end-to-end (E2E) quality of service (QoS) of a complex system combining communication networks and cloud computing. CPS used in healthcare are expected to require uninterrupted mobile and wireless connectivity, throughput guarantee, stringent latency and jitter features, and hitless cloud-related virtual machine migration and auto-scaling operations.

Another intricate problem arises when objective QoS metrics need be translated into indexes of quality of experience (QoE) to the end-user. When physically interacting with the device, the user is literally in-the-loop of the control algorithms, which affects stability [76]. Latency onto the control loop is another factor that greatly impacts the overall stability, affecting both physical and cognitive interaction between human and CPS.

The combination of recent and forthcoming mobile and wireless systems, cloud computing, and networking technologies (e.g., 5G wireless networks, mobile-edge computing, and software-defined networking), will help a great deal in addressing QoS issues. Nevertheless, a better understanding into QoE is needed not only in CPS for healthcare, but also in other systems heavily influenced by the human-in-the-loop effect such as car driver assistance [77], functional electrical stimulation treatment [78], and exoskeleton use [68]. QoS issues for remote healthcare CPS have been investigated by Shah et al. [79], which state that approaches to QoS in healthcare services are scarce in the literature as this is an emerging field. Hammer, Egger-lampl and Moller [80] consider that QoS needs to be carefully controlled in CPS and its impact on QoE need to be taken into account. Considering that the upcoming generation of assistive CPS for healthcare will deeply rely on network/cloud services, there is a need to understand how these new technologies will be affected by the quality of such services, specially on a user's perspective. As cloud services for healthcare CPS cannot be provided on a best-effort basis, QoE indexes should actually impose the QoS requirements for those future network/cloud infrastructures.

The powerful drive of the incoming 5G networks is another factor that helps to pave the way for current advances in the robotic field [11]. It is envisioned that 5G technology will provide network services from three categories: enhanced mobile broadband (eMBB), massive machine type communications (mMTC), and ultra-reliable, and low-latency communications (uRLLC) [81]. The latter will be a key enabler for providing mobility assistive services for patients dependent on heavy-processing critical procedures. Moreover, cloud architectures utilizing mobile edge computing can help providing stringent latency and jitter in case these also present real-time requirements. URLLC goals are to provide latency figures as strict as 1 ms alongside 10^{-5} to 10^{-9} unavailability ratios. Software-defined networking is expected to provide an uniform mechanism for E2E network programmability, this means not only in between patient's point of care and the cloud but also in inter and intra cloud networks [82] for dynamically provisioning E2E availability, throughput, and latency guarantees.

Assistive robots in general can benefit in multiple ways from this new ecosystem composed of wireless/mobile, programmable network and cloud computing. Figure 5 links some of the assistive devices' main features and communication and computation features as enablers to compose healthcare services. The black arrows in the figure exemplify a navigation system exploiting wireless connectivity, network's low latency, and cloud platform's high processing power, to provide a real-time mobility assistance healthcare service.

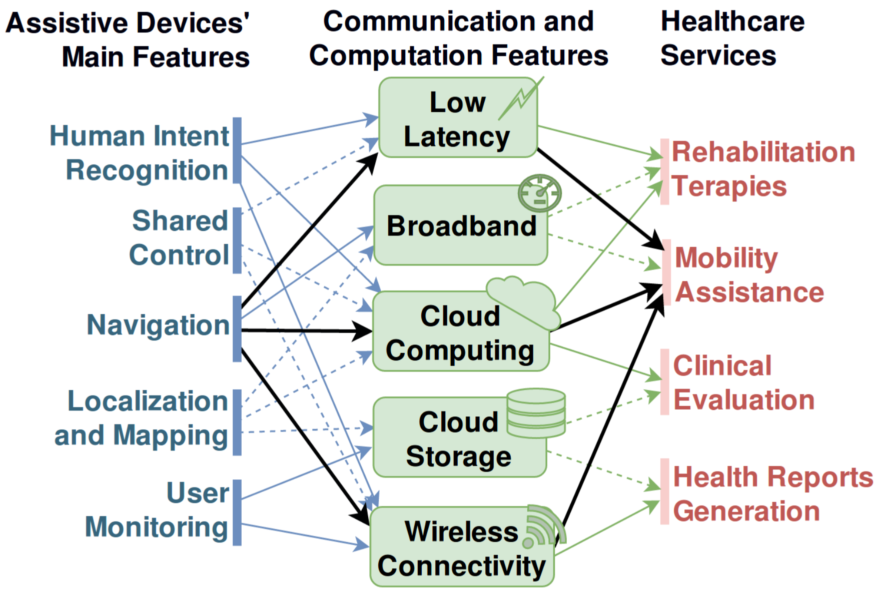


Figure 5 – Example of healthcare services enabled by communication and computation technologies.

Smart walkers, in particular, are also represented in Fig. 5, as each of the five functionalities previously listed are linked to at least one of the features in such figure. Wireless connectivity is needed due to the mobile nature of such devices, and it often must be combined with low latency networks to provide services in real-time, especially the ones directly involving human-robot interaction.

Features involving user and health monitoring demand storage capabilities to construct patient history. Such databases can feed adaptive algorithms that provide features such as navigation assistance and shared control to effectively respond to the patient's variable impairment condition. Moreover, in a connected healthcare system, patient monitoring can be used to directly contact ambulances in emergency situations [64]. This can be coupled with recent advancements in smart clothing [83] and low power wide area networks (LPWAN) technologies [84] for improved measurements and performance.

Sensorial and cognitive assistance functionalities offered by smart walkers can also benefit directly from such environment. Solutions that explore cloud points or images, such as the ones presented by Panteleris and Argyros [36] and Paulo, Peixoto and Nunes [53], are often computationally expensive and can be performed in the cloud [59, 85]. In that way, cloud platforms can host data processing services, including algorithms for person and object recognition, as the ones presented by Redmon et al. [86] and Cao et al. [87], for improved context-aware applications, in which the ability to interpret environmental information is of great importance. Moreover, cloud services can also be used to share information among devices and even help in path planning during navigation [12].

As safety is a main concern in assistive technologies, there is also debate on what processes should be kept local and what processes can be remotely deployed [63]. The constant exchange of information between walker and remote platform allows for different strategies regarding communication issues. For instance, less complex control algorithms can be locally processed, fed by remotely generated parameters, or even being assessed only in emergency situations, when communication issues are identified. Nevertheless, situations in which QoS is largely degraded can be a risk to the user and must be carefully administered not only over a network/cloud provider perspective, but also on the smart walker itself.

In the light of this new environment, a cloud-enabled CPS for mobility assistance is proposed in the next chapter. The proposed system envisions the integration of smart walkers and cloud computing platforms to expand the capabilities and features offered by such mobility assistive devices through the exploitation of the concepts previously presented. Thus, the proposal of this CPS aims at the development of a new generation of mobility assistive devices, enabled by the cloud and able to cope with future connected healthcare systems.

4 UFES CloudWalker: Proposal of a Cyber-Physical System for Mobility Assistance

A proposal of a cloud-enabled CPS for mobility assistance is presented in this chapter. The system, named UFES CloudWalker, arises from the integration of smart walkers and cloud computing, and its first implementation is built upon an in-house developed smart walker and services executed on a remote cloud platform. The overall implementation is described in this chapter, followed by preliminary validation studies. As there is a need to understand the feasibility of the proposed system and how network and cloud parameters might affect its use, two sets of experiments are performed to assess such impacts. This is done before implementation details are further discussed, which will be later resumed on the next chapter.

4.1 Overall System Design

4.1.1 Proposed Architecture

The UFES CloudWalker, from now on referred merely as CloudWalker, system architecture allows for the use of smart walkers as cloud-enabled CPS, encompassing a system that involves physical and cognitive human-robot-environment interaction to assist mobility impaired individuals. CloudWalker's overall system architecture is depicted in Fig. 6.

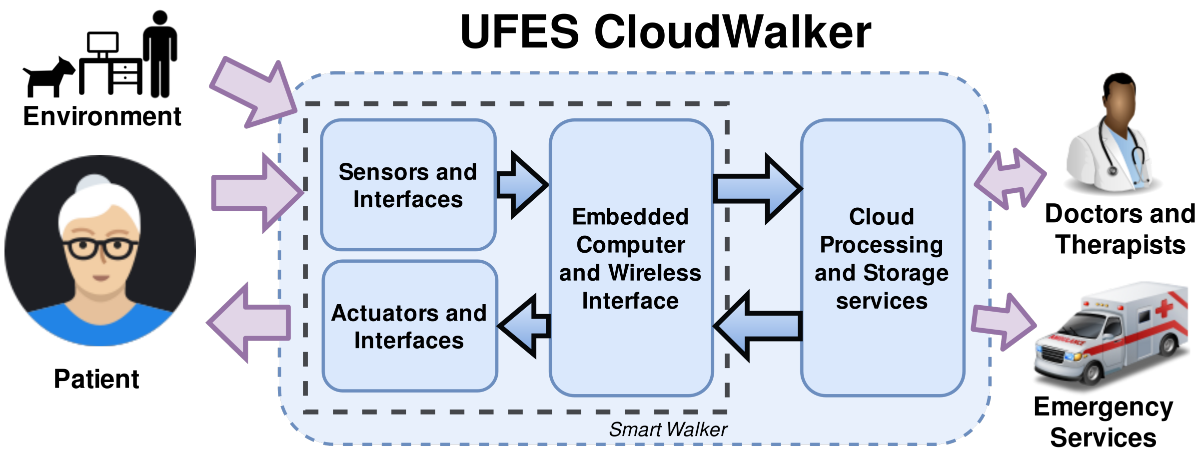


Figure 6 – The UFES CloudWalker architecture.

In such system, user's motion intents are perceived by embedded sensors on a smart walker, which communicates with a cloud platform, exchanging information. The smart walker can send sensors and interface data to be processed on the cloud, which can provide services such as storage and data processing. The cloud platform can also generate control signals to command actuators, in case control algorithms need to be remotely processed. Moreover, CloudWalker envisions a connected healthcare, in which data stored on database services can be accessed by healthcare professionals and emergency services can be directly triggered from monitoring services.

CloudWalker architecture decouples the physical assistive device (i.e., the smart walker and its embedded hardware) from the features it can offer to the user. The centralization of services running on the cloud allows for the use of multiple heterogeneous cloud-enabled smart walkers working under the same CloudWalker architecture while making use of subsets of the services provided by the cloud platform. The cloud is also able to offer elasticity and scalability to guarantee the proper execution of real-time services or to allow the insertion of new smart walkers into the system.

The cloud's processing and storing capabilities can be leveraged to provide services that can not be locally performed at the smart walker due to hardware limitations. Moreover, databases can be constructed from patients' history of use and the centralization of such services in the cloud decouples patients from local devices, allowing for multiple patients constantly interchanging smart walkers among themselves in shared facilities (e.g. hospital or nursing homes). The gathering of large amounts of data allows for the implementation of algorithms able to automatically identify the user and adapt to its specific needs.

4.1.2 Materials and Methods

In this work, CloudWalker is firstly implemented by the integration of an in-house developed smart walker, a cloud platform, and a series of controllers processed on virtual machines on the cloud. The materials and methods used on this implementation are introduced in this section and further details will be discussed on the next chapter, during the system's experimental validation.

An important building block in CloudWalker's first implementation is the UFES Smart Walker (Fig. 7), a smart walker developed by the NTA, at UFES. This smart walker has been used in several research works over the last years, such as [7, 16, 44, 88], mainly focused on human-robot interaction strategies.

The UFES Smart Walker possesses the kinematics of a unicycle robot, with a pair of differential rear wheels driven by DC motors and a front caster wheel. A couple of H1 (US Digital, USA) encoders and a inertial sensor BNO055 (Adafruit, USA) are used to capture the device's velocities and orientation, information used by odometry algorithms to infer its position. Two triaxial force sensors, MTA400 (Futek, USA), with their respective amplifiers CSG110 (Futek, USA), are positioned under each forearm supporting platform to capture the physical interaction forces between the user and the walker. The UFES

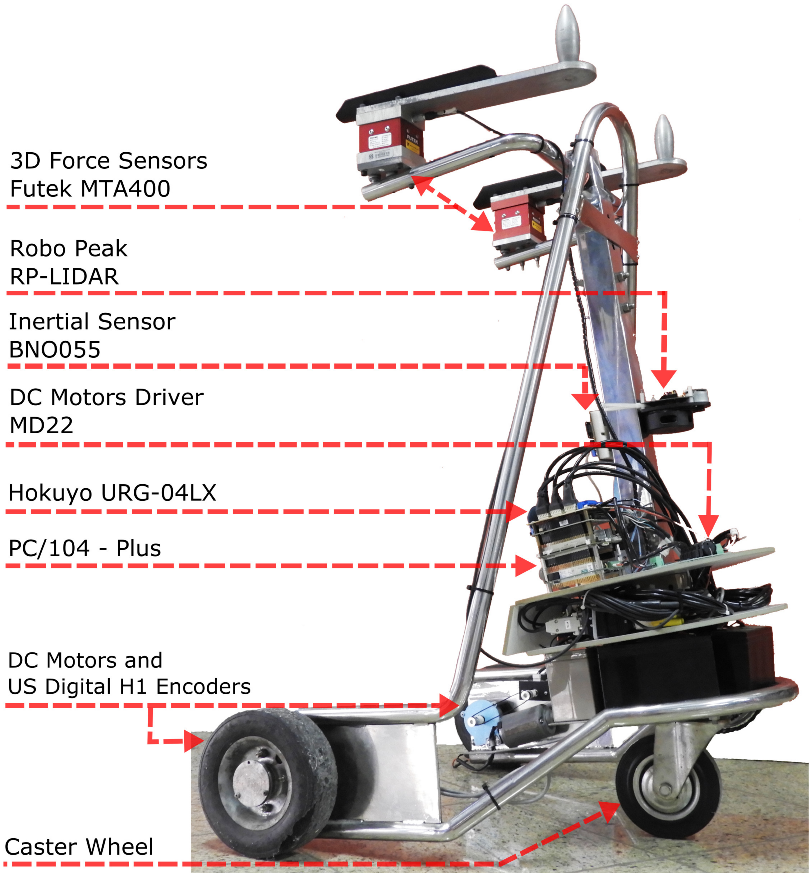


Figure 7 – The UFES Smart Walker.

Smart Walker can also make use of two LRF sensors: a URG-04LX (Hokuyo, Japan) is placed on the back of the walker to capture user's legs position, and a RP-Lidar (Robo Peak, China) is installed in the front to obtain environmental information. The embedded computational system is based on the PC/104-Plus standard, possessing a 1.67 GHz Atom N450, 2 GB of flash memory, and 2 GB of RAM. This embedded computer is integrated into the Simulink Real-Time (MathWorks, USA) environment.

The Simulink Real-Time environment provides a fast prototyping platform for the UFES Smart Walker. Its toolboxes allow the real-time acquisition of sensors data within a Simulink code that can be generated in an external computer and uploaded to the embedded PC/104-Plus. A sample time of 10 ms is used by the control loop due to time constants for low level actuation controllers. A custom toolbox for the UFES Smart Walker was generated within the duration of this work to easy the prototyping process. With such toolbox, each sensor became a block that can be added directly to the Simulink environment, providing as output the appropriate data. CloudWalker's implementation also makes use of such tools, as illustrated in Fig. 8, with the main difference being that some - or all - sensor data is sent to an external platform through datagram sockets, which are also used to retrieve the remotely generated information and control signals.

Another core building block of the CloudWalker is the cloud platform. In this work, the

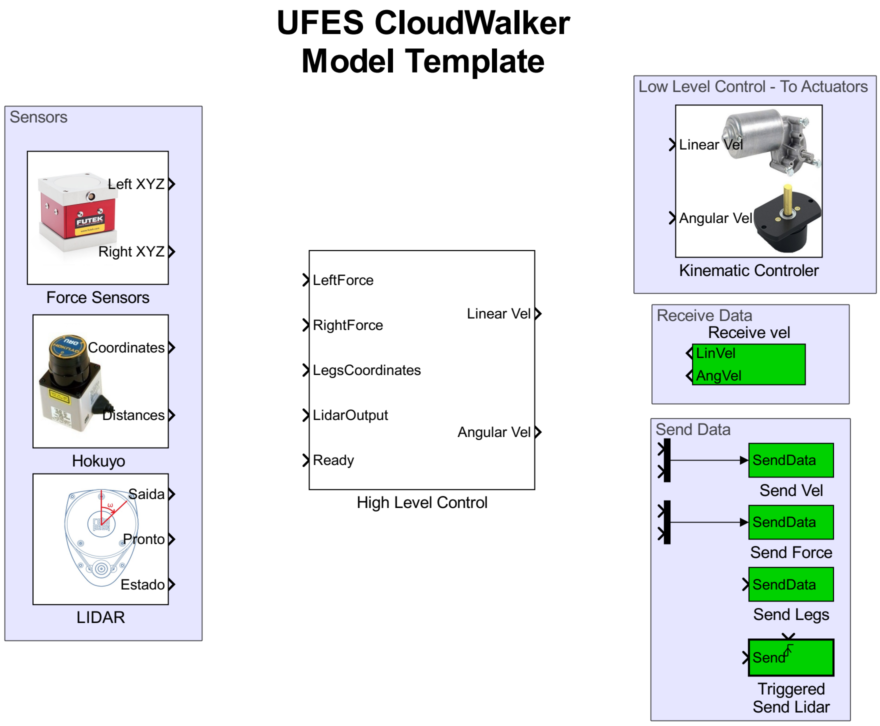


Figure 8 – Simulink template designed for data acquisition, actuators control, and the communication between the embedded PC/104-Plus on the UFES Smart Walker and the remote cloud computing platform.

cloud platform used is based on OpenStack version Ocata. OpenStack is an open-source cloud management software, actively developed by the community, and capable to provide and manage network, processing, and storage resources in a data center.[1] It is designed to allow the deployment of massively scalable public and private clouds. By the joint work of several services, OpenStack is able to provide the so-called Infrastructure as a Service (IaaS) cloud level.

Virtual machines (VM) are instantiated in the cloud platform to provide services for CloudWalker. Different computing resources can be set before VM instantiation, and OpenStack can be programmed to automatically manage life cycle and resources to provide elasticity and scalability if necessary. The platform also handles VM IP assignment for exterior communication, necessary to allow direct communication between the service and the smart walker.

In this work, ROS is installed in the cloud's VMs and is responsible for data processing, storage, and logging.[2] ROS is an open-source middleware that provides an abstraction layer to robot's hardware resources, treating physical data as streams on a publisher-subscriber architecture [89]. ROS also integrates a large set of libraries and open-source software, allowing code reutilization, which softens development time. Moreover, it has been used in

[1] More information available in the OpenStack Wiki: https://wiki.openstack.org/. Accessed February 1, 2018.

[2] More information available in the ROS Wiki: http://wiki.ros.org/. Accessed February 1, 2018.

many service, industrial, and research robots, including smart walkers, as in [22].

As the UFES Smart Walker does not possess a wireless network interface card, a Raspberry Pi 3 model B (Quad Core 1.2GHz Broadcom BCM2837 64bit CPU, 1GB RAM) was used as a wireless gateway in this implementation of CloudWalker.[3] The Raspberry Pi provides 2.4GHz 802.11n wireless LAN, which is suitable for a first version of CloudWalker, and runs the Raspbian Stretch Lite v4.9 operating system. ROS is also installed in the Raspberry for logging purposes, as the data stored in it can be used for offline measurements of network metrics, such as latency and packet loss. Communication between the Raspberry and the smart walkers's PC/104-Plus uses datagram sockets in a strategy based on the one presented in [90]. In this first implementation, communication between the Raspberry and the VM on the cloud is also based on datagram sockets, as each one is aware of the other one's local IP address.

Before the system's complete implementation and testing, there is a need to understand its own feasibility by assessing the impacts of network and cloud parameters on CloudWalker and how those parameters affect its use.

4.2 Preliminary Validation

CloudWalker architecture is built upon the assumption that the network and the cloud services are able to provide proper performance requirements, such as adequate availability, E2E latency, and throughput. For instance, E2E latency directly influences physical system's response time, what might degrade both control, leading to instability issues, and QoE. Latency variability also influences performance by inserting feed-backed information disorder. In a short-term scale, congestion and errors over wireless connection are causes of packet loss, resulting in throughput reduction. There are also long-term, network and cloud availability issues that also lead to packet loss events. VM availability, and instantiation and migration time in the cloud platform must be observed, and also wireless connectivity gaps, downtime and handover time should be taken into account when using such system.

To validate the proposed system, the feasibility of employing CloudWalker in the real world must be investigated. There is a need to access how network and cloud QoS parameters affects CloudWalker usage and safety. To do this, a set of experiments is designed to investigate user's perceived QoE under variable QoS by testing the system under different network and cloud conditions. By assessing user feedback over their usage experience, it is possible to subjectively evaluate CloudWalker's performance through the end-user point of view.

[3] More information available in the official Raspberry Pi documentation: https://www.raspberrypi.org/documentation/. Accessed February 1, 2018.

There is a need to understand beforehand the system's minutiae and how it is impacted by external factors. As some processing and control tasks might be performed within the walker's embedded system and different configurations and levels of services can be provided by the cloud, it is likely that CloudWalker will be more commonly implemented on systems running some tasks locally. Nevertheless, to better understand the system, during this preliminary validation stage, it will be considered that all data processing and control execution is performed remotely. During the preliminary validation experiments, CloudWalker system was not entirely implemented. Instead, QoS parameters were locally emulated on the walker's embedded computer to maintain full control over parameter values and variation. In this way, the system's feasibility can be verified, validating its use before the actual implementation of the cloud services and communication. Figure 9 exemplifies the architecture used during the preliminary validation stage.

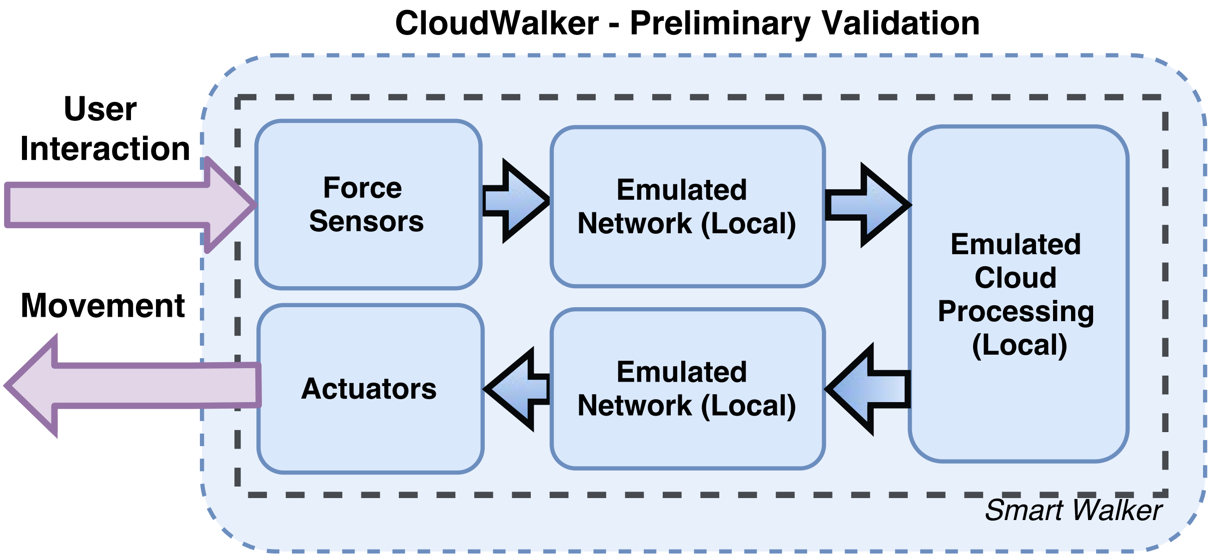


Figure 9 – UFES CloudWalker architecture during the preliminary validation experiments.

The network and the cloud are emulated in the Simulink environment installed on the smart walker's embedded computer. The developed algorithm used to emulate the network can insert deterministic and variable latency onto the system, allowing for packet superscription and disorder. It can also emulate packet loss and network availability according to probabilities set as part of the algorithm's configuration parameters. For instance, the algorithm can allow the experimenter to set a 50 ms deterministic latency and a 10 % probability of packet loss, while guaranteeing full network availability.

4.2.1 Proposed Scenarios

Two sets of experiments are performed over two different scenarios to validate Cloud-Walker's proposal. Both scenarios are built over mobility assistance services, different among themselves. On the first experiment set, the use of the walker is evaluated under variable latency conditions by the assessment of objective and subjective metrics. On the second set, more variable QoS parameters are tested, such as packet loss and availability, and subjective metrics are observed to evaluate the impacts of such parameters on user experience.

Experiment Set A

On experiment set A, the effects of E2E latency and its variability on CloudWalker are evaluated. The proposed scenario is based on a path following task, in which the user can freely control the walker's direction and velocities, while asked to follow a specific path. This illustrates a scenario in which patients in a hospital or nursing home must follow marks on the floor when going to the cafeteria or an appointment. Users are requested to follow a path marked on the floor multiple times under different latency conditions. Users are also asked to evaluate their experience in the usage of the walker after each run.

On total, ten test conditions are established. A benchmark condition is performed to provide baseline measurements of the local use of the controller. Three test conditions with deterministic latency values are also performed to evaluate the impact of latency alone. Such latency values are set to 100, 200, and 300 ms. The same latency values are used as mean latencies to test latency variability. This is emulated based on a normal distribution, with values centered on 100, 200, and 300 ms, and variances of 4 and 9 μs^2, forming the other six test conditions. This values are empirically chosen to avoid overlap in latency values among test conditions with different mean values, and the distribution variances are intentionally set to small values on this first validation experiment set.

Five volunteers participated on this first set of experiments. All subjects can be considered healthy and presented no gait dysfunctions. Their ages ranged from 25 to 35 years and their weights from 56 to 80 kg. Only two of the volunteers had some previous usage experience with a smart walker while the other three had none. Each of these volunteers were submitted to all the ten test conditions in randomized order to avoid distortions in their evaluations.

The same path is used on all tests regarding this experiment set. It consists of three straight lines connected by 90° curves, the first to the left, the second, to the right. All participants are advised to walk at comfortable and yet slow pace while following the path.

Two metrics are used to quantify the usage of CloudWalker. The first one is a subjective metric regarding user experience, the mean opinion score (MOS). After each run, users are asked to grade the quality of their experience in a discrete numerical scale ranging from one to five. They are required to answer a one-question questionnaire: how do you rate your experience in a scale of one to five, one meaning a negative and five a positive experience? The MOS is calculated as the average grading each test condition obtained [91]. This is done to capture user's perceived QoE under different QoS conditions.

The second metric employed is an objective measurement named kinematic tracking error (KTE) [51], in which the desired path is compared to the performed one. KTE is obtained using the following equation:

$$KTE = \sqrt{|\bar{\varepsilon}|^2 + \sigma^2} \tag{4.1}$$

in which $|\bar{\varepsilon}|$ is the absolute mean error between the discrete points that compose the performed and desired path, and σ^2 is the variance of these error values. The KTE gives a number that raises with the increase of the mean error and its variance. Therefore, it takes into account not only stationary errors on the path following, but also oscillations in this error value, both factors that directly reflect the struggle in maintaining the walker on track.

To calculate the KTE, the path marked on the floor is translated into a desired path composed by a discrete series of points distanced 20 cm one another. The walker's odometry is compared to this desired path and the error values are obtained from the distance of each odometry point to the nearest point in the desired path.

Experiment Set B

On experiment set B, not only latency and its variability are evaluated, but also network availability and packet loss. This experiment is built upon a scenario in which hospitalized patients and nursing homes inhabitants may benefit from the use of cloud-enabled assistive devices. Particularly in nursing home environments, assistive devices may be required to guide patient's displacement based on patient-lead human-robot-environment interaction. CloudWalker does not pull nor push the user, but haptically provides a preferred displacement by easing movements in the direction of a pre-established path, while hardening for attempts to move it elsewhere.

Ten test conditions are established and listed in Table 1. For each test, only one parameter (i.e., latency, availability, or packet loss) is changed whereas the other ones are considered ideal. On test conditions which involve variable latency, latency variability is modeled again using a normal distribution. On this set of experiments, a parameter called mean variance ratio (VMR) was used to set a fixed ratio between the mean latency and the variance of the distribution. This normalizes the measure of the dispersion of the E2E latency distribution so that results can be compared among test conditions with different mean latencies, providing an alternative way to look into those results.

Over this universe of ten test conditions, five are randomly chosen for each participant. Tests ordering is also randomized to avoid parameter-induced bias. The pre-established path is unknown to the users *a priori*, and they must find out which way to go while interacting with CloudWalker.

Twelve participants were recruited to take part in the experiment. All subjects can be considered healthy, with no disabilities or gait disorders, and only two of them had prior

Table 1 – Experiment set B: emulated network and cloud QoS test conditions.

Parameter	Condition
Benchmark	Local control
Deterministic E2E latency	10ms
Mean E2E latency (VMR 0.01 ms)	100 ms 300 ms 500 ms
Availability	75% 50%
Packet loss	10% 20% 30%

experience using the smart walker. Participants' ages ranged from 22 to 33 years old, their weights from 57 to 92 kg, and their heights from 1.61 to 1.85 m.

In this experiment set, the same path is used on all tests, consisting of three straight lines interspersed by smooth curves to both sides. All subjects are advised to walk at comfortable and yet slow pace. Each subject performed five tests under different test conditions, aware that some conditions could perform better than others.

MOS is used to subjectively evaluate user perceived QoE. At the end of each test, participants are asked to describe their experience using the walker, based on the same numerical scale as in experiment set A. This time, comments and answers were also registered to evaluate user's QoE.

4.2.2 Control Architecture

Experiment Set A

In experiment set A, the user is asked to follow a path marked on the floor while using the CloudWalker. The control algorithms that command the smart walker's motion allows the user to have full control over the system. The 3D force sensors installed under the forearms supports are used to measure the physical interaction forces between user and CloudWalker, which sets reference values for linear and angular velocities. The control strategy is based on [20] and the control law is described by equations 4.2 and 4.3:

$$v_c = K_1(F_r^f + F_l^f) \tag{4.2}$$

$$\omega_c = K_2(F_r^f - F_l^f) \tag{4.3}$$

where v_c and ω_c are the control signals for linear and angular velocities, respectively. K_1, K_2 are proportionality constants empirically set in values to allow a comfortable pace. F_r^f, F_l^f are the forward right and left force sensors signals, respectively, filtered by a first-order low-pass filter to remove noise and undesired signal components.

Despite its simplicity, this control strategy is based only in physical interaction forces and poses itself as a natural way for the user to command the walker's motion, resembling the familiar feeling of handling a supermarket trolley, while bearing weight on the forearms supports. The walker repeats the last received control signal if no new information arrives within the 10 ms time sample window.

Experiment Set B

In experiment set B, the user must follow a predetermined path without possessing full control over CloudWalker's motion. Instead, a shared control strategy is implemented to guide the user throughout a pre-established path while avoiding deviations from it. The control strategy is based on the one proposed in [92], which encompasses a path follower control module, leveraged to provide the desired orientation of the smart walker in relation to the path. An admittance controller, based on [93], is used to detect user's motion intention. As there is a pre-established path to be followed, the path follower controller [94] weights robot's current position and desired orientation to stay on - or to get on - the path. The admittance controller is also responsible for generating control signals for the device's linear and angular velocities.

CloudWalker's linear velocity is directly linked to user interaction forces, while the steering velocities also depend on the desired orientation given by the path follower algorithm. In such strategy, the user has no direct input in consciously changing the device's orientation, which is mainly dictated by the device's relative position to the path.

The control law implemented is described by equations 4.4 and 4.5:

$$v_c = \frac{(F_r^f + F_l^f) - m_v \dot{v}}{d_v} \tag{4.4}$$

$$\omega_c = \frac{v_c}{d_w} \tanh(\tilde{\theta}) \tag{4.5}$$

in which v_c and ω_c are the control signals for linear and angular velocities, respectively. As in experiment set A, the walker repeats the last received control signal if no new information arrives within the 10 ms time sample window. F_r^f, F_l^f are the right and left force sensors signals, respectively, filtered by a first-order low-pass filter to remove noise and undesired signal components. m_v is a virtual mass parameter, and d_v, d_w are damping parameters used for properly setting velocities and to define the dynamics of response and interaction between user and smart walker. $\dot{v}$ is the walker's velocity first order derivative. $\tilde{\theta}$ is the error between CloudWalker's desired and actual orientations. The desired orientation, θ_d, is defined by equation 4.6:

$$\theta_d = atan(\frac{x_d}{y_d}) \tag{4.6}$$

And the point (x_d, y_d) is given by the path follower algorithm, described by equation 4.7:

$$\begin{bmatrix} \dot{x_d} \\ \dot{y_d} \end{bmatrix} = \begin{bmatrix} v_r \cos(\theta_p) + l_x \tanh(\frac{k_x}{l_x}\tilde{x}) \\ v_r \sin(\theta_p) + l_y \tanh(\frac{k_y}{l_y}\tilde{y}) \end{bmatrix} \tag{4.7}$$

in which $\dot{x_d}, \dot{y_d}$ are the desired velocities, v_r is the reference velocity, or ideal path following velocity, θ_p is the path reference orientation, defined by the tangent of the nearest point in the path. $\tilde{x}, \tilde{y}$ is the position error, l_x, l_y are saturation limits, and k_x, k_y are gain parameters.

By implementing such control strategy, CloudWalker interacts with the user and, while respecting the user's own pace, leads the way by hinting the desired orientation of movement.

4.2.3 Preliminary Results and Discussion

Experiment Set A

Figure 10 shows the desired path, as marked on the floor, against some examples of paths performed by multiple users under different test conditions, each on described in terms of mean latency and variance. Figure 10 exemplifies what was expected and observed in practice: the increase of latency seems to harden the use of the walker, inducing oscillatory courses.

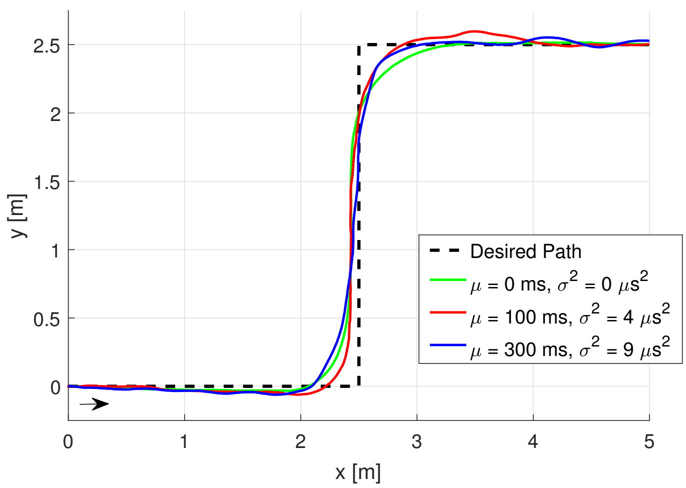


Figure 10 – Desired path (dashed line) and examples of performed paths under different test conditions, expressed in terms of mean latency and latency variance

The MOS obtained from subjects' evaluations and the calculated KTE can be observed in Fig. 11, in which each bar represents a specific test condition, grouped and colored according to mean latency and variance values. Figure 11(a) displays the MOS, and it is possible to note a clear correlation between the increase in E2E latency and user experience degradation, even though latency variability (i.e., jitter) does not seem - at first glance - to negatively impact the subjective evaluation.

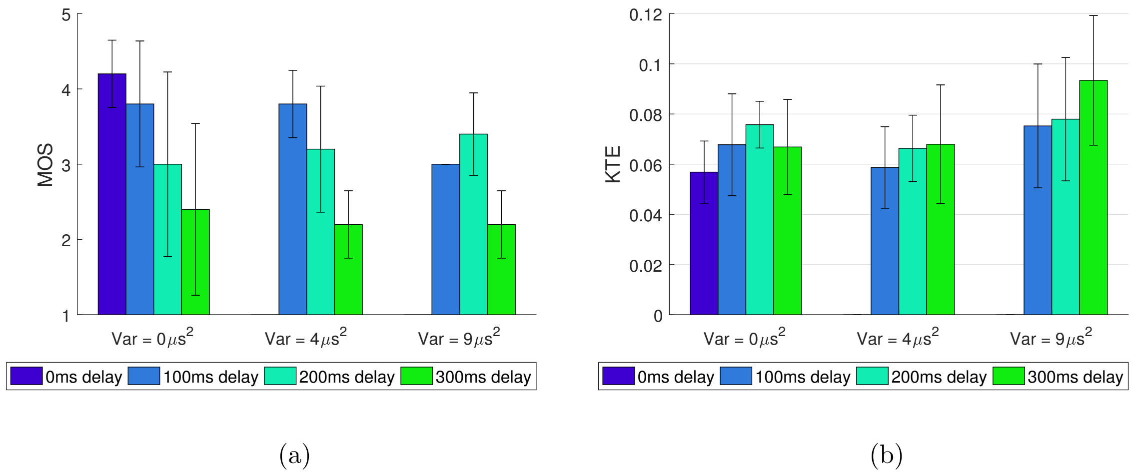
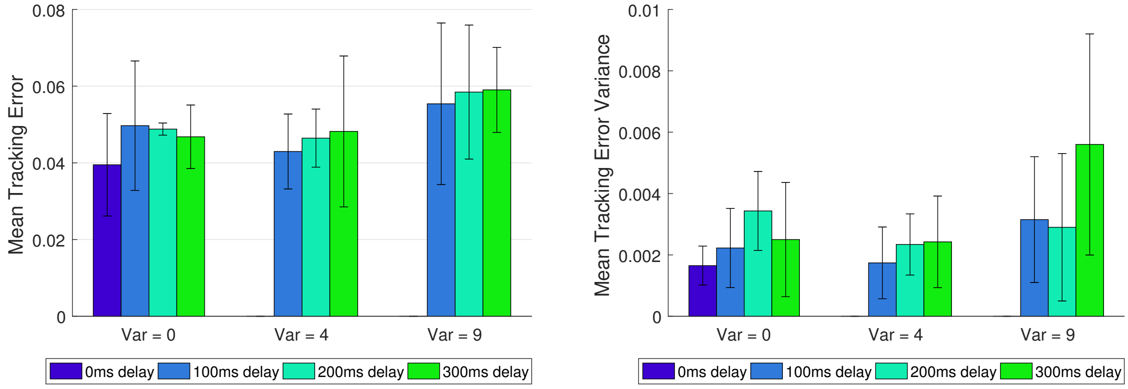


Figure 11 – Mean and standard deviation values of objective and subjective metrics for each test condition, grouped by latency variance and colored by mean latency. a) MOS Scale; b) Kinematic Tracking Error (KTE).

Figure 11(b) displays the values obtained from the objective analysis via the KTE metric. The least value was observed when there was no E2E latency, on the benchmark test condition. The increase in the average latency does seem to induce hardship upon the task, while latency variability seems to actually impact only when latency variance was larger.

The KTE should not be evaluated just *per se*. As it is composed by the mean tracking error and its variance, both factors must be taken into account when building an understanding on CloudWalker's system. Figure 12 shows, respectively, the mean tracking error and the mean error variance. While the mean error displays a slight tendency to increase with latency and its variation, error variance seems to better reflect user quality of experience.

In order to discuss the obtained results, some observations are necessary to be made. The control signal delay and the disorder inserted in this signal through latency variability generates unexpected and - eventually - intense interactions between user and walker. Frequently, this kind of interaction results in the user pulling the walker towards self, inducing wheel slips and degrading odometry. As the user rarely deviates too much from the path, such interaction is not clearly perceived on KTE measurements, but rather impacts the variability of the tracking error. Therefore, this metric may be more suited to



(a) (b)

Figure 12 – Mean and standard deviation of the factors that compose the KTE for each test condition. a) Mean tracking error; b) Tracking error variance.

reflect user experience.

Another necessary observation is related to test conditions in which latency variability is higher, which seem to - in a certain way - randomize the control signal sequence that commands the actuators, leading to the larger standard deviations observed in the third group in Figures 11(b) and 12. Lastly, it was noted that walking at slower paces while significantly depositing weight on the forearms supports mitigates the previously mentioned factors.

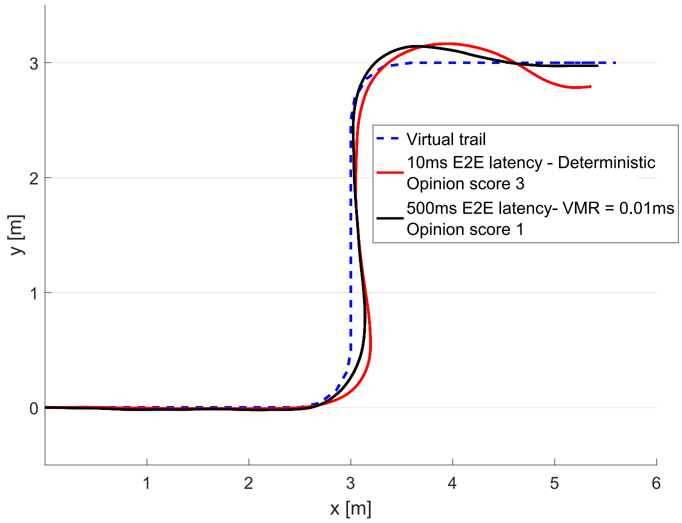
Although the number of volunteers on this experiment may be rendered small, it was enough to point strong tendencies of control degradation with the increase of latency and its variation. Latency mean value seems to be preponderant to user quality of experience, while the increase in its variation does not seem to largely affect the subjective evaluation. Nevertheless, as the variance values set are relatively small, the true impacts of real-world latency variability may not have been fully emulated in this experiment set. Regarding the objective metrics, error variance between desired and performed path indicates how oscillatory was the course.

Finally, it is important to state that E2E latencies of 100, 200, and 300 ms represent 10, 20, and 30 control samples, respectively. When evaluating test conditions with smaller latency variability, it can be noted that even 100 ms delay does not significantly degrade user experience. In other words, even though reliable high speed networks are desirable for this class of applications, this experiment indicates that latency values of few samples do not render CloudWalker unfeasible.

Experiment Set B

Although the participants of experiment set B did not know beforehand the path to be performed, they figured out how to interact with the walker and managed to successfully follow the predetermined path. Two participants had to be fully removed from the gathered data set as they intentionally pushed CloudWalker beyond the limits of wheels adherence or velocity controllability.

Figures 13 and 14 presents the outcomes for predetermined and actually performed paths, interaction forces, and linear velocities from two realizations from the same subject using CloudWalker under different E2E latency conditions: 10 ms (deterministic) and 500 ms (VRM 0.01 ms). The score for each realization is also shown so that the complex relations between physical interaction effects and subjective evaluation could be clearly illustrated. It can be observed in Fig. 13 that there are small deviations from the virtual trail even under 500 ms latency. In fact, the largest deviations occurred under the smallest latency condition, and data from all tests indicates that this is mainly due to user's gait speed, which tends to increase according to user's positive perception of the experience.

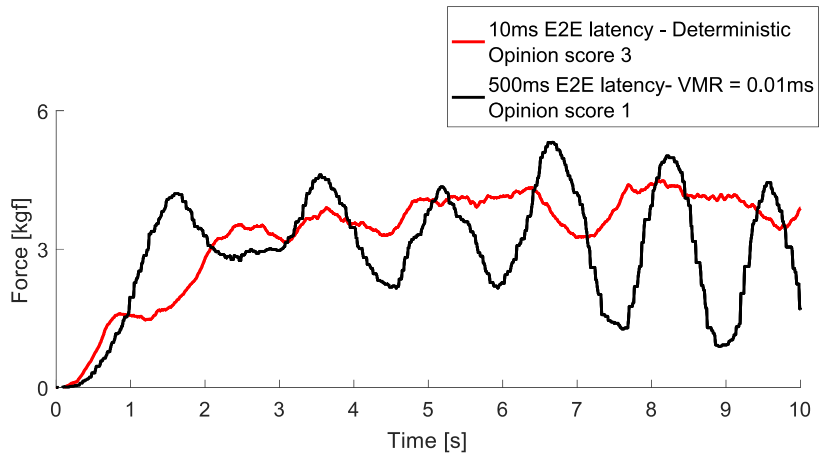


(a)

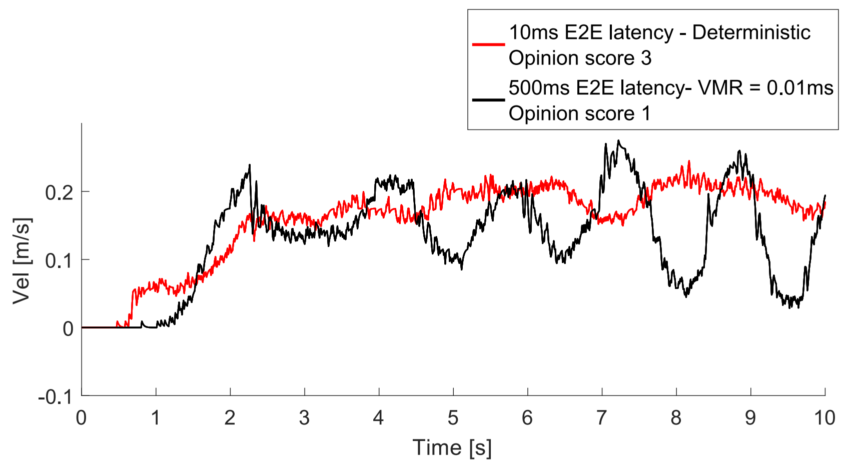
Figure 13 – Observations from experiment set B: path traveled by same subject under different test conditions.

As it might be expected, the subject better graded the test under smaller E2E latency, and Fig. 14 may provide evidences to understand the reason: 500 ms latency degraded the participant's ability to control the CPS. Cognitive and physical interactions on such human-in-the-loop system is impaired because of temporal mismatches between perception and action. This leads to intense oscillations in the interaction forces in Fig. 14(a) and subsequent speed variation in Fig. 14(b). Notice also that the largest deviations seen in Fig. 13 happened for 10 ms latency condition. This is so due to user's gait speed, which tends to increase according to user's positive perception of the experience, as the higher average velocity for 10 ms latency in Fig. 14(b) indicates.

Figure 15 brings two realizations from different participants to illustrate the complexity of assessing QoS effects on user and system overall performance. Although under the same 500 ms (VMR 0.01) latency test condition as in Fig. 13, in Fig. 15 it caused CloudWalker



(a)

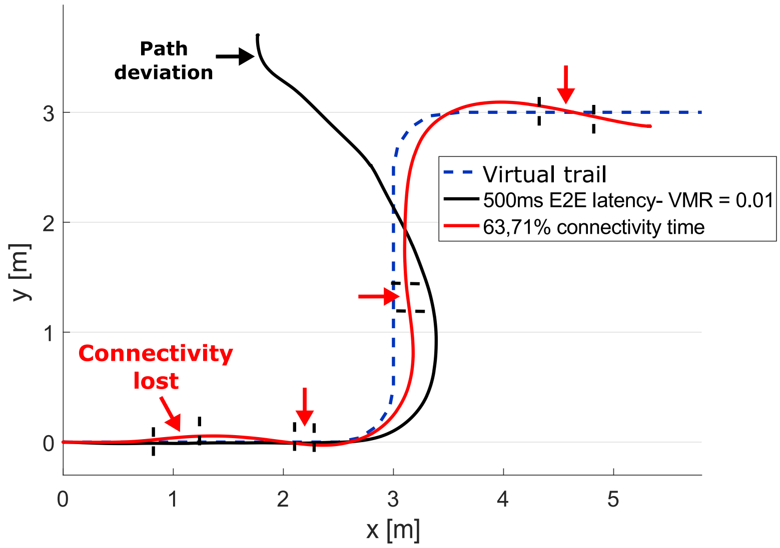


(b)

Figure 14 – Observations from experiment set B, same tests as the ones portraited in Fig. 13: a) forward interaction forces observed, first ten seconds; b) CloudWalker linear velocities, first ten seconds.

to completely miss the pre-established path. This leads to safety concerns and may draw limits for QoS degradation. In contrast, Figure 15 also presents a realization under poor QoS (only 63.71 percent availability) and yet successfully performed. However, it is worth noting that no period longer than two seconds of connection loss was observed along the curves of the path.

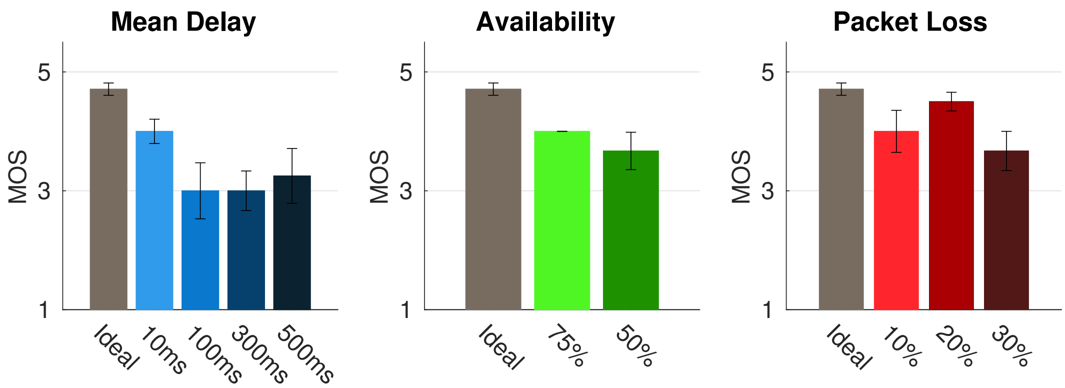
MOS was obtained and results are displayed in Fig. 16. It is noticeable that unfavorable QoS conditions clearly deteriorate user experience. Among parameters evaluated, E2E latency appears to be of most significance. Network availability also seems to degrade QoE, though the implemented control application showed itself quite resilient to loss of connectivity. Packet loss ratio over the network also seems to affect experience, but given the slow system dynamics and the relatively high sampling rate, no significant effects could be observed.



(a)

Figure 15 – Observations from experiment set B: a special case of path deviation, and impacts of connectivity losses larger than two seconds.

Statistical limitations from this reduced pool of realizations may be the reason for QoE outliers in Fig. 16. The human factor directly impacts the experiment itself. For example, four participants experienced 500 ms E2E latency, but only one (depicted in Fig. 14(a)) evaluated its QoE in the lowest score, whereas remainder participants had a more pleasant experience without large oscillations and all scored 4.



(a)

Figure 16 – Results of subjective MOS and coefficient of variation (i.e., standard deviation normalized by the mean) for each network parameter evaluated on the pilot experiment.

Most users followed the assigned virtual trail correctly with no prior information and under poor QoS conditions. Despite the limited number of participants in this pilot experiment, our outcomes unveil trends for relating QoS indexes and human-robot physical interactions with control loops affecting QoE degradation.

E2E latency is the major cause of lower QoE and also raised safety concerns as controllability might degrade to unstable conditions. There is need for deeper studies with larger VMR for more realistic latency variability assessment. It was also noticed that MOS scores analysis alone did not reflect entirely participants' experiences. For instance, one subject said "this test is better than the previous one" and then gave the same score in both tests. In another session, a test in which the participant said "the walker seems to be failing" received a 3 and it was not the lowest score given by that particular user. Comments like "this was the best one", "smooth", and "comfortable" were usually linked to highest scores, but negative comments were not necessarily followed by the lowest score.

4.2.4 Preliminary Remarks

As CloudWalker relies on the integration of physical smart walkers and cloud computing platforms, the remote execution of control algorithms can be largely impacted by network and cloud QoS parameters. The objective of performing a preliminary validation stage before fully implementing the system was to verify the proposed system's feasibility beforehand and to better understand the impacts of variable QoS onto the system from the end-user point of view.

Two different experiment sets were performed to assess both quantitative and qualitatively the impacts of variable QoS over the system. In experiment set A, the effects of E2E latency were evaluated and it was verified that, although the increase in latency affects control and user experience, QoE does not seem to be significantly degraded under lower latency parameters. Despite user difficulty to handle the system, users were able to follow the suggested path even under higher E2E latency conditions. In experiment set B, besides latency and its variability, network availability and packet loss were also evaluated as network and cloud QoS parameters. Such parameters are also pertinent to real world operation and CloudWalker use was not compromised even under poor QoS conditions.

It must be noted that the entire control was remotely executed during this preliminary validation stage. This takes CloudWalker to its most QoS vulnerable condition by directly inserting latency and information loss and disorder to the control loop. Ideally, any implementation of CloudWalker should retain at least a basic control algorithm running locally on the smart walker for safety assurance. This local control could be fed by remotely generated parameters or should, at least, be executed in times of connectivity loss or poor QoS, taking control of the system. Even without such mechanism, CloudWalker's preliminary validation experiments point to the system's feasibility on real-world usage. For instance, testing the exchange of 70 B packets from UFES to a major cloud provider on US' coasts, an average latency of about 700 ms was observed, decreasing to 50 ms with their edge-like service. This reinforces the possibility of repeating preliminary validation scenarios in commercial cloud providers.

In the next chapter, CloudWalker's first implementation will be discussed in details. The implemented system is validated in a scenario similar to the ones presented in this chapter, during the preliminary validation stage. The control algorithms are remotely executed in a cloud platform and communication is performed over a network linking a laboratory at UFES and an edge data center.

5 UFES CloudWalker: Experimental Validation

After the preliminary validation stage (described in Section 4.2), the proposed CloudWalker system was implemented for its final validation studies. In this chapter, CloudWalker's implementation is further detailed and the validation scenario is described. The validation experiments are exposed and the results obtained are discussed by the end of the chapter.

5.1 Validation Scenario and UFES CloudWalker Implementation

The proposed scenario for CloudWalker validation experiments is based on a mobility assistive service in which all the control algorithms are remotely executed. A set of experiments is performed to evaluate how network parameters actually impact CloudWalker use when assisting on user locomotion. Therefore, CloudWalker's implementation is validated in a non-simulated scenario built over a non-structured environment.

As the user interacts with CloudWalker, the walker's embedded computers are responsible only for reading data from sensors, communicating with the cloud, and executing the remotely generated control signals. The communication is performed over different network setups, which links the walker, a wireless access point (AP), NERDS laboratory local network, and the cloud setup running on the NERDS data center.

An admittance-based controller is implemented on the cloud to command CloudWalker's motion. The controller uses the interaction forces captured by the force sensors to provide full control over the walker to the user. As opposed to the performed preliminary experiments, the user is not asked to follow a line or to stay over a predetermined path, being completely able to walk and turn on his/her own pace and liking.

The user conducts CloudWalker on part of a pedestrian pathway (approximately 1.8 m wide) composed by straight lines and 90° curves to both left and right. An wireless AP working on the 802.11n 2.4 GHz band is positioned at approximately 10 meters away from the pathway midpoint to provide connectivity to CloudWalker. Figure 17 illustrates this scenario. The dashed double-sided arrow in the figure represents the wireless communication between the smart walker, specifically the wireless interface of the Raspberry Pi, and the AP, which is directly connected to a local NetGear 16 Port switch installed inside the NERDS laboratory. This switch is connected to a Mikrotik RouterBoard installed at the NERDS data center, in an adjacent room. The RouterBoard is responsible for connecting the laboratory's local network and the cloud platform network.

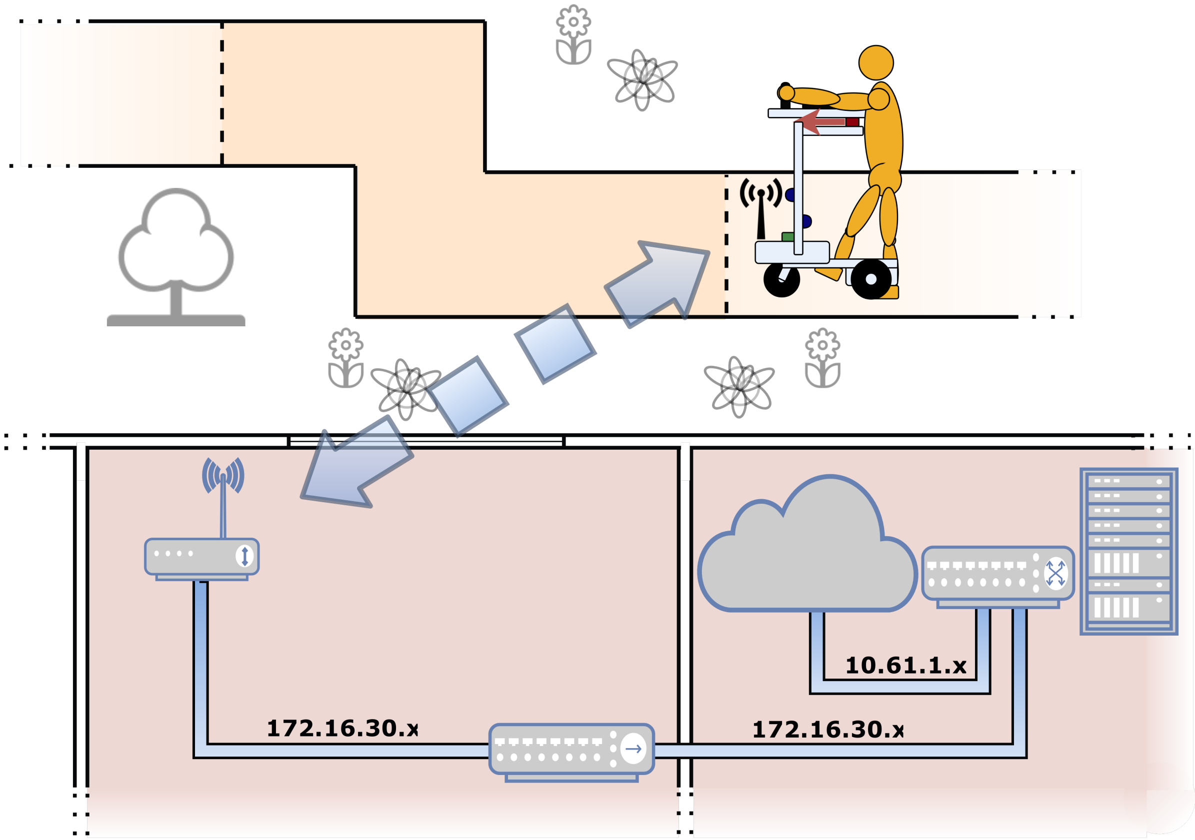


Figure 17 – UFES CloudWalker system validation: scenario and network architecture.

The CloudWalker system was implemented according to the description in Section 4.1. The overall implementation of the system is depicted in Fig. 18. The PC/104-Plus is responsible for gathering incoming data from the force sensors, wheel encoders, and IMU. It is also responsible for the walker's local odometry system and for generating sequence numbers used to uniquely identify each transmitted packet. This embedded computer sends force signals, velocity and odometry information to the local Raspberry Pi via UDP socket. Both devices have network interfaces sharing a local network and know each other's static IPs (ranging from 192.168.1.1 to .3, non-related to the networks depicted in Fig. 17). The Raspberry Pi, which is also located at the smart walker, is connected to the PC/104-Plus via a copper cable and uses another UDP socket to receive the information regarding the walker's sensors.

The Raspberry Pi makes use of the ROS middleware. It reads the received data, publish it to separate topics (one for the force signals, one for the velocities, and another one for odometry) and then uses its wireless network interface to send information to the VM running on the cloud. The information published to each of the ROS topics is encapsulated together with its sequence number, as generated by the PC/104-Plus, and with a time stamp. This information is stored in separate files, one file per topic, via

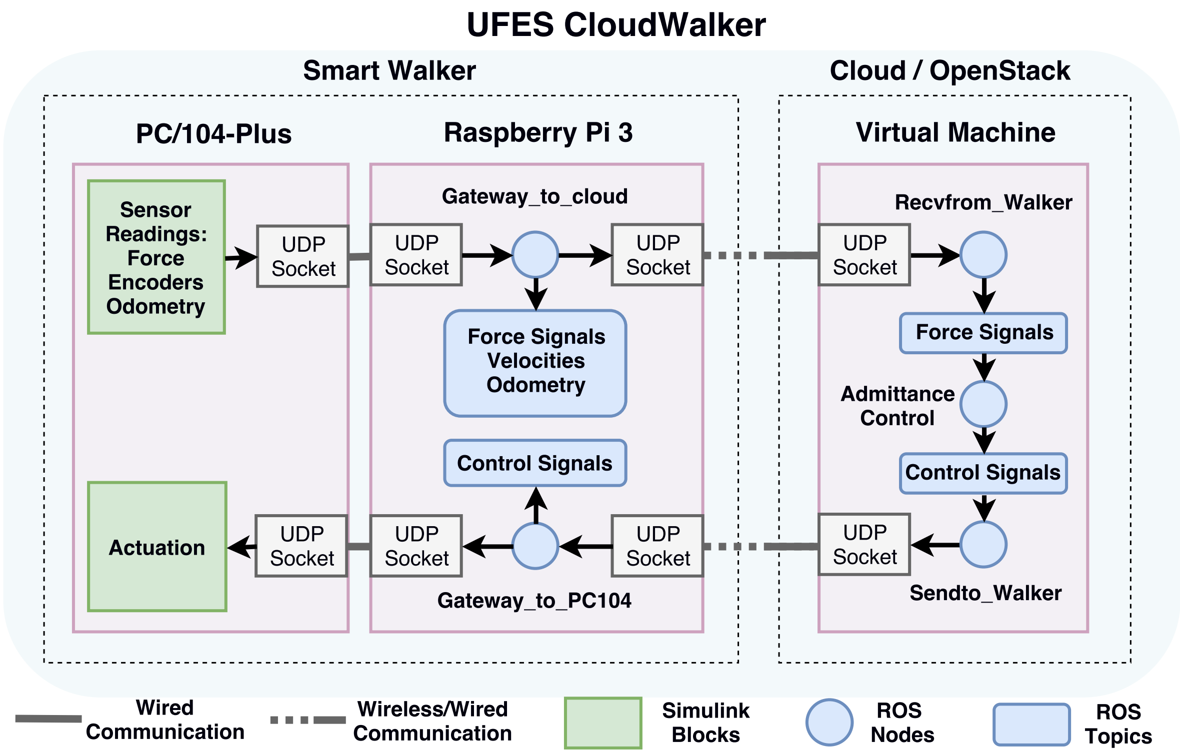


Figure 18 – UFES CloudWalker system validation: implemented architecture.

the rosbag package.[1] These files are used to store the raw information regarding each topic and for offline evaluation of packet loss and network latency. The force signals and its corresponding sequence numbers are encapsulated in a UPD packet and sent to the VM to be processed by CloudWalker's controller. UDP is chosen as networking protocol as lower latency values should be prioritized over higher reliability on the exchange of information. The dashed-and-continuous line in Fig. 18, representing the wireless and wired communication network depicted in Fig. 17.

Figure 19 illustrates the VM being executed inside a private network on OpenStack. The cloud platform addresses a public IP, or "floating IP" as called by OpenStack, to the VM, thus making it accessible to the Raspberry Pi. The VM receives the incoming data on a UDP socket and makes use of the ROS middleware to process the information. One ROS node is responsible for reading the incoming information and publishing the force signals and its sequence numbers to a specific ROS topic. Another ROS node is responsible for executing the control algorithm that commands CloudWalker's motion. It subscribes to the topic in which the force signals are published, process the data, executes the control algorithm, and publish the generated control signals to another ROS topic. A last node is responsible for subscribing this topic containing the control signals and for sending this information to the Raspberry Pi, located on the smart walker. Throughout all this process flow, the packet sequence number is always encapsulated together with the processed information (i.e., this number is gathered from the packet, published on both

[1] More information available in the ROS Wiki: http://wiki.ros.org/rosbag. Accessed February 1, 2018.

topics, and send back to the Raspberry Pi together with the control signal).

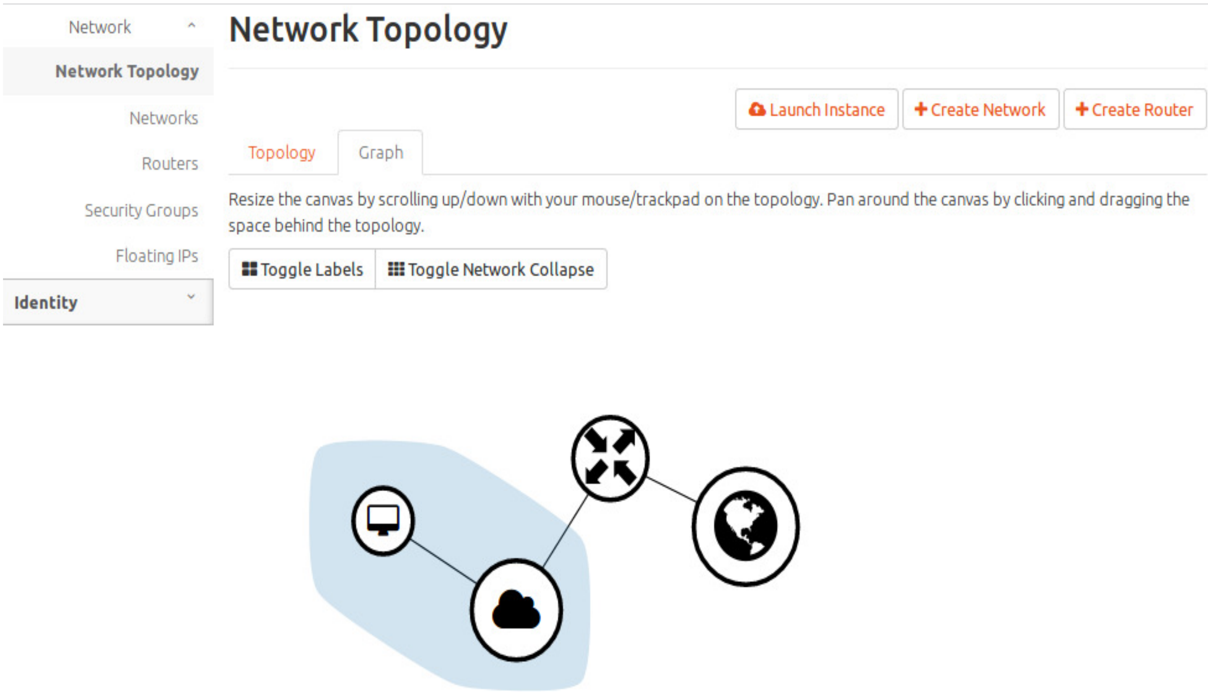


Figure 19 – Screen capture of Horizon, the OpenStack's dashboard, displaying the VM deployed to execute UFES CloudWalker control algorithms. There is a virtual local network inside the cloud in which other VMs can be instantiated.

As the Raspberry Pi receives the remotely generated control signals and its corresponding sequence numbers, a ROS node is responsible for time-stamping this data and publishing it on a topic. Again, the information published in this topic is stored on a separate file for offline evaluation of network metrics. This ROS node is also responsible for sending the control signals to the PC/104-Plus, which receives it and uses it to command CloudWalker's actuators. If no packet arrives at the PC/104-Plus within its sample time, which is set to 10 ms due to low level actuation control constants, the device repeats the execution of last control signal received.

5.2 Control Architecture

An admittance-based controller is used to provide the user liberty to command CloudWalker's motion. The controller is implemented within a ROS node running on the cloud and the interaction forces between user and walker, forward force, F, and torque, τ (check equations 5.1 and 5.2), are used to command the walker's velocities.

$$F = F_l + F_r \tag{5.1}$$

$$\tau = \frac{d_h(F_r - F_l)}{2} \tag{5.2}$$

in which F_l, F_r are, respectively, the forward forces measured by the left and right force sensors and d_h is the distance from one handle to another. The admittance-based controller uses those values as input to calculate the control signals.

The controller sets zones of constant and varying friction, depending on minimum values for force and torque, F_{min} and τ_{min}. It implicitly solves the differential equations given by equations 5.3 and 5.4 to generate the control signals for linear and angular velocities.

$$m\dot{v} = \begin{cases} 0 & \text{if } v = 0 \text{ and } |F| \leq F_{min} \\ F - F_{min} sign(F) & \text{if } v = 0 \text{ and } |F| > F_{min} \\ F - F_{min} sign(F) & \text{if } |v| > 0 \text{ and } B_v v \leq F_{min} \\ F - B_v v & \text{if } |v| > 0 \text{ and } B_v v > F_{min} \end{cases} \tag{5.3}$$

$$J\dot{w} = \begin{cases} 0 & \text{if } w = 0 \text{ and } |\tau| \leq \tau_{min} \\ \tau - \tau_{min} sign(\tau) & \text{if } w = 0 \text{ and } |\tau| > \tau_{min} \\ \tau - \tau_{min} sign(\tau) & \text{if } |w| > 0 \text{ and } B_w w \leq \tau_{min} \\ \tau - B_w w & \text{if } |w| > 0 \text{ and } B_w w > \tau_{min} \end{cases} \tag{5.4}$$

in which F, τ are the forward force and the torque calculated from the force sensors inputs. Apparent mass, m, moment of inertia, J, drag, B_v, and angular drag, B_w, are the parameters that regulate the system's apparent dynamics. The control signals are then generated according to equations 5.5 and 5.6

$$v_c = \begin{cases} 0 & \text{if } v_l = 0 \text{ and } |F| \leq F_{min} \\ \frac{(F - F_{min} sign(F))}{m} d_t & \text{if } v_l = 0 \text{ and } |F| > F_{min} \\ v_l + \frac{(F - F_{min} sign(F))}{m} d_t & \text{if } |v_l| > 0 \text{ and } B_v v_l \leq F_{min} \\ exp(\frac{-B_v d_t}{m}) v_l + (1 - exp(\frac{-B_v d_t}{m})) \frac{F}{B_v} & \text{if } |v_l| > 0 \text{ and } B_v v_l > F_{min} \end{cases} \tag{5.5}$$

$$w_c = \begin{cases} 0 & \text{if } w_l = 0 \text{ and } |\tau| \leq \tau_{min} \\ \frac{(\tau - \tau_{min} sign(\tau))}{J} d_t & \text{if } w_l = 0 \text{ and } |\tau| > \tau_{min} \\ w_l + \frac{(\tau - \tau_{min} sign(\tau))}{J} d_t & \text{if } |w_l| > 0 \text{ and } B_w w_l \leq \tau_{min} \\ exp(\frac{-B_w d_t}{J}) w_l + (1 - exp(\frac{-B_w d_t}{J})) \frac{\tau}{B_w} & \text{if } |w_l| > 0 \text{ and } B_w w_l > \tau_{min} \end{cases} \tag{5.6}$$

in which v_c, w_c are the control signals for linear and angular velocities, respectively, d_t is the time difference between last and current step, and v_l, w_l are the last step control signals for linear and angular velocities. Due to safety concerns, the control signals for linear and angular velocities are limited: $0 \geq v_c \geq 0.3$ m/s and $-0.5 \geq w_c \geq 0.5$ rad/s. Those values are considered to be fit due to prior empirical observations.

5.3 Experimental Protocol

The experiments are conducted on a pedestrian pathway in an outdoor environment, next to the building in which the NERDS laboratory and data center are located. The pathway is approximately 1.8 m wide and a track approximately nine meters long, consisting of straight lines and sharp curves to the right and to the left, is chosen for the experimentation. Starting and finishing lines are draw to delimit the track. The walker starts over the starting line, the user is able to fully control CloudWalker's motion and there is no specif path to be followed. The user interacts with the walker heading towards the finishing line, stopping only when reaching the track's limit.

The walker communicates with a wireless AP, which works in the 802.11n 2.4 GHz band and is positioned at approximately 10 meters away from the pathway midpoint. Received signal strength indicator (RSSI) levels along the pathway were measured[2] using the Raspberry Pi and average values ranging from -65 to -60 dBm where observed.

Six experiment realizations are conducted over the same track. Each experiment realization starts after communication between walker and cloud is established. Sensor data and control signals are timestamped and stored in the walker for offline processing. The walker is responsible for gathering data from its sensors, for performing an odometry algorithm for recording purposes only, and for commanding its actuators, delegating all other processing tasks to the cloud, which executes the control algorithm. Communication and network parameters can affect overall system performance.

Another experiment realization, the seventh one, is performed until complete loss of connectivity is observed. The walker is positioned in the middle of the track used in prior realizations and the user commands its displacement along the pathway, moving away from the wireless AP, until there is no communication between the smart walker and the cloud platform.

5.4 Results and Discussions

All seven realizations were performed along the same pathway and over the same test conditions, as previously described. During the first six realizations, no communication loss was observed and network and cloud parameters' impact over the system could not be observed by the user during most of the time. On two occasions, the user described a feeling of momentarily disobedience or discomfort. Figure 20 brings the odometry recorded from realization number four to illustrated the path walked along the pathway.

As there were no pre-established path nor guidelines marked on the floor, the user was free to walk and turn on its own pace, leading to the oscillatory S-shaped path. The interaction forces measured on CloudWalker's force sensors were also recorded, as the

[2] This can be done using the command *iwconfig* on a Raspbian terminal. More information available in Linux man page: https://linux.die.net/man/8/iwconfig. Accessed February 1, 2018.

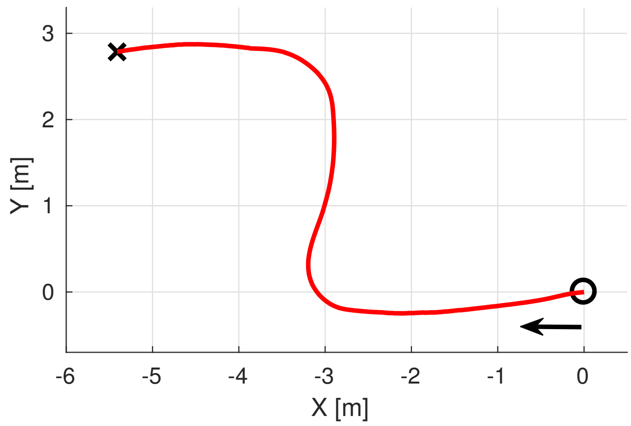


Figure 20 – Odometry recorded from realization number four. The walker starts on the lower right part of the image, over the starting line of the track, and is guided by the user towards the mark on the left upper part, representing the finishing line.

measured velocities and remotely generated control signals. Figure 21 brings the measured values during realization number four.

In Fig. 21, the readings from both force sensors are repeated in the top of the image so the impacts of the interaction forces over forward force and torque, control signals, and velocities can be better assessed. The oscillations on the measured force values result mainly from the user gait, which can be pointed the fact that those oscillations are usually on counter-phase. In other words, during each stride, there is a tendency to push or pull the walker, differing in each arm according to the stage of the gait. Forward force and torque are calculated according to Equations 5.1 and 5.2 and the graphs in the middle of Figure 21 illustrates the beginning of the movement, the moment of each turn, and the moment in which the user attempts to stop the walker as it approaches the finishing line.

The control signal values for linear and angular velocities displayed on the bottom of Fig. 21 result from the control law described in Section 5.2. The control signal for linear velocity is maximum during most of the experiment, decreasing mostly when the user is turning the walker. This can be observed on two periods of time: one ranging from 15 to 20 s, and the other one ranging from 25 to 32s, approximately. The turning periods are indicated by the opposition of signal in the forces measured by the forces sensors, which results in the largest torque values.

Measurements from realization number four, as displayed in Figs. 20 and 21, are representative for the first six experiment realizations. The values for odometry, forces, and velocities are similar to the ones measured on the other five realizations. The effects of network and cloud QoS could not be clearly perceived in any of them. As for the measurements of QoS parameters, latency and packet loss measurements from realizations

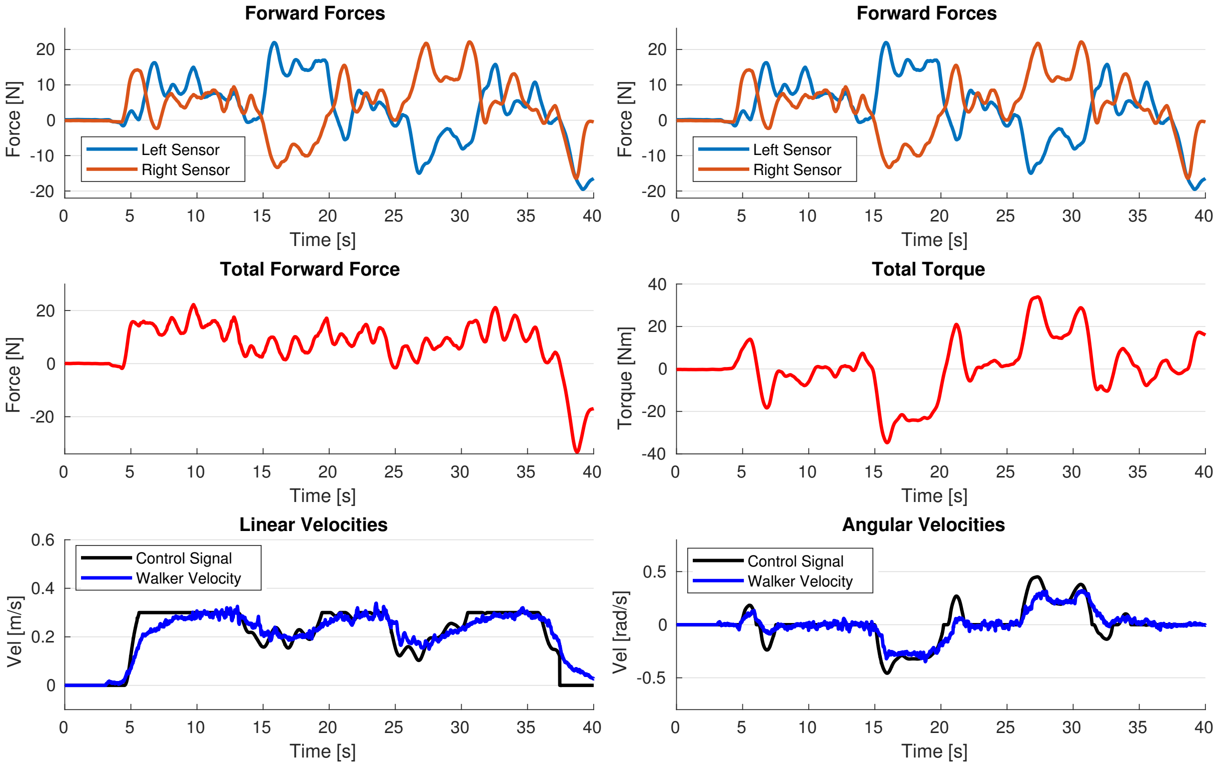


Figure 21 – Measured interaction forces and resulting velocities on realization number four. From right to left, top to bottom: forward forces measured in each 3D force sensor, graph repeated on the left, resulting total interaction forward force, resulting interaction torque, control signal and measured values for UFES CloudWalker linear velocities, and control signal and measured values for angular velocities.

one to six are shown in Table 2.

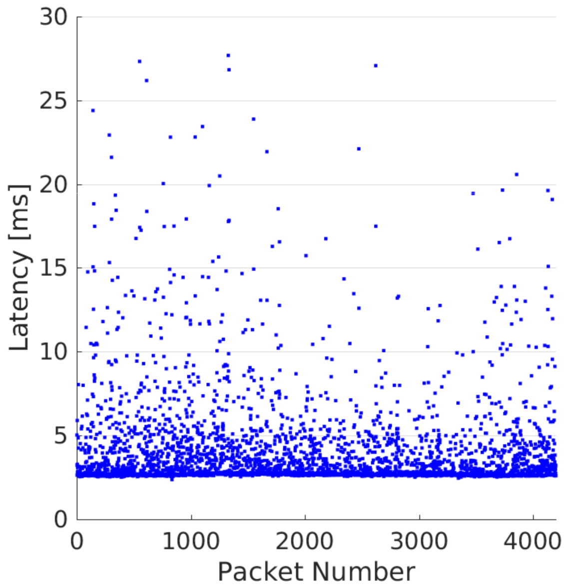
Table 2 – Validation experiments: latency and packet loss measurements. Realizations one to six were performed along the same track.

Realization Number	**E2E Latency**				**Packet Loss %**
	Mean (ms)	VMR (ms)	Min (ms)	Max (ms)	
1	9.7	39.60	2.4	235.2	1.08
2	4.1	3.34	2.7	74.1	0.00
3	4.0	3.06	2.3	57.1	0.01
4	5.8	41.96	2.4	212.0	0.44
5	11.6	72.00	2.6	297.1	1.31
6	5.8	7.97	2.5	100.0	0.00

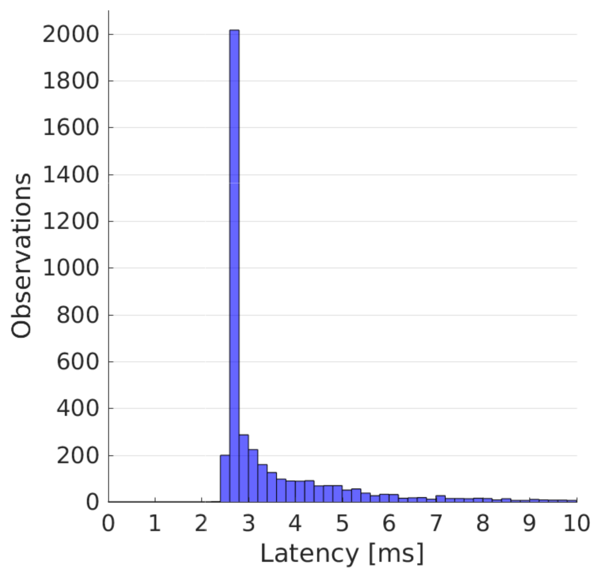
As it can be observed in Table 2, the measured mean values for E2E latency were far lower than most of the ones emulated during the preliminary validation experiments, which ranged from 10 to 500 ms. This is mainly due to the fact that the cloud platform was running on an local data center. Minimum latency values are also quite low, consuming only a fraction of the 10 ms sample window, though the maximum values observed often surpassed some of the conditions established during preliminary validation (i.e., 10, 100, and 200 ms). This impacts latency variability, and the resulting VMR values are far greater than the ones previously emulated, which were all smaller than 0.09 ms. Despite sporadic observations, the impact of such variability could not be perceived by the user.

Packet loss rates are also lower than the ones emulated during the preliminary validation experiments and apparently had no effect on the handling of CloudWalker. The deploy of cloud services to remote and edges platforms and the increased in services offered and their complexity can impact the loss of information along the network and cloud. Nevertheless, as the system showed itself quite resilient to the loss of information, rates of packet loss below 10 % should not impact CloudWalker's usage.

Figure 22 displays E2E latency measurements from realization number four. In Fig. 22(a), the latency observed in each packet received from the cloud is shown. It can be observed that spikes in latency are evenly distributed throughout the realization and that most observations lie below the 5 ms range. This is confirmed in Fig. 22(b), in which the histogram shows that most packets are received from the cloud less than 3 ms after the corresponding packet is sent from the walker.



(a)



(b)

Figure 22 – Latency measurements from realization number four: a) E2E latency observed in each received packet; b) Histogram for observed E2E latency values.

The last experiment realization, the seventh one, is related to connectivity loss. During this realization, as the walker moved away from the wireless AP, the effects of latency could be felt on the handling of the system. There was one short connectivity loss on the beginning of the realization. Connectivity was quickly reestablished and the realization continued. The walker was guided in a sort of sine-wave shaped path, so that increased latency or connectivity loss would be apparent. This was done until the walker lost connectivity once again, this time too far away from the AP to re-connect, in a zone in which RSSI was later measured to be around -85 dBm. Latency and packet loss measurements from this realization is shown in Table 3. As realization number seven was performed until connectivity loss, it was conducted over a larger period of time and under gradually worse RSSI conditions, thus explaining the differences observed in value ranges on Table 2 and Table 3 .

Table 3 – Validation experiments: latency and packet loss measurements. Realization number seven is related to connectivity loss.

Realization Number	**E2E Latency** Mean (ms)	VMR (ms)	Min (ms)	Max (ms)	**Packet Loss %**
7	100.4	761.48	2.6	3786.6	5.50

Table 3 shows a mean E2E latency value more compatible with the ones emulated during preliminary validation. As realization number seven starts with the walker positioned inside the track used by the previous experiments, the minimum latency observed is similar to the ones displayed in Table 2. As the smart walker gets away from the AP, RSSI decreases, resulting in increased latency and packet loss. The large VMR observed points to information disorder, which also impacts the handling of CloudWalker. Larger latency values were observed just before connectivity loss, period of time in which more packets were lost. Despite the poor quality of wireless signal during most of the experimentation, the packet loss rates were low again.

Figure 23 displays E2E latency measurements from realization number seven. The two red arrows in the figure point to the moments in which communication between smart walker and cloud was lost. The leftmost arrow points toward the first connectivity loss, which was quickly reestablished, while the rightmost arrow point to the moment of final loss of connection.

As opposed to Fig. 22(a), latency observations in Fig. 23 are not quite uniform. Despite the first latency peak that accompanied the first moment of connectivity loss, latency was kept low and slowly increased as the smart walker went away from the wireless AP. These higher latency values largely impacted on the handling of CloudWalker. The histogram for latency observations is not shown due to the large concentration of samples bellow the 100 ms range.

5.5 Final Remarks

Overall results from the preliminary validation stage pointed to the feasibility of deploying CloudWalker even under poor network and cloud QoS conditions. After CloudWalker's first implementation was completed, validation experiments were performed over an actual network linking the smart walker and a cloud platform. Despite network's

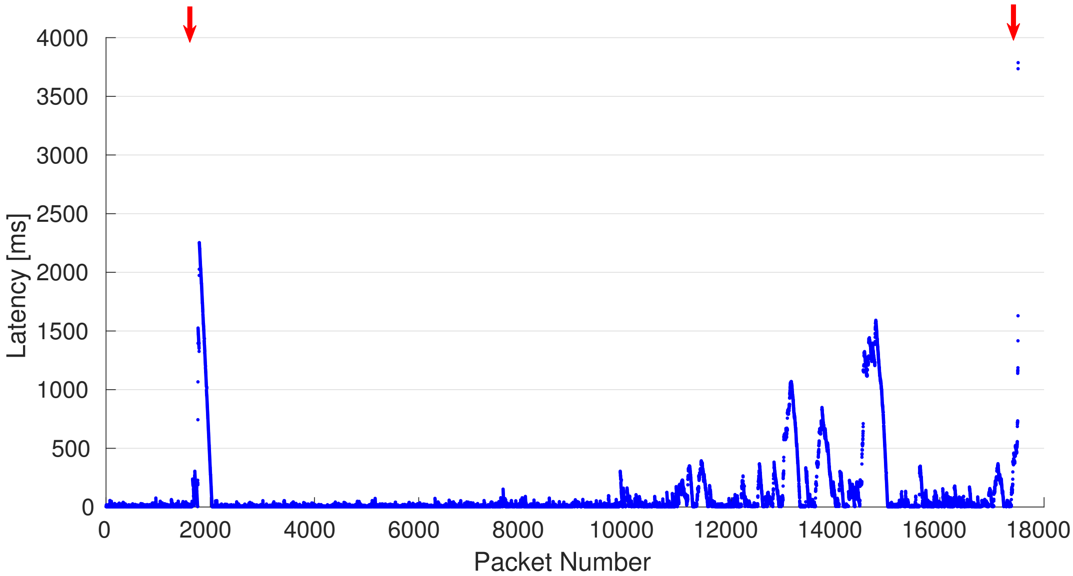


Figure 23 – Latency measurements from realization number seven: E2E latency observed in each received packet. The red arrows point to moments of connectivity loss

simplicity and the physical proximity of the cloud to the walker, results from the validation experiments indicates that CloudWalker can indeed be employed for mobility assistance.

The proposed system was validated under its most vulnerable configuration, counting on the full deployment of control algorithms to be remotely executed. Under such configuration, the system becomes latency-sensitive, and the time between the sending of sensorial information and the receiving of control signals is perceived directly as a delay onto the control loop. Nevertheless, the device's inherently low velocities and the walker's own inertia are mitigating factors that enhance the system's resilience to latency. The analysis of the obtained results indicates CloudWalker's feasibility when making use of in-site clouds, also pointing that the system is likely to work by exploring MEC concepts. This is probably enhanced by retaining local processing on the device, which should be able to perform its own control algorithms whenever poor QoS is detected.

Latency variability was not thoroughly assessed during the preliminary validation stage. The magnitude of such parameter was much greater on the actual network then it was previously emulated. Despite the high latency variability observed during CloudWalker validation, its effects were not perceived during the first six experiment realization. The evaluation of latency variability and its effects over realization number seven can not be decoupled from the high E2E latency values observed in such realization. Information disorder is likely to be mitigated by prioritizing newer information while discarding older packets at arriving. As latency, and not its variability, is indicated to be the main impacting factor over control, this strategy might be a solution to latency variability when latency values are low.

Information loss over the network was assessed during the preliminary validation stage and did not appear to heavily impact system's usability or end-user experience. The measured values for packet loss during CloudWalker's validation experiments were far lower than the ones previously emulated. Packet loss rates smaller than 5 % should not degrade user experience significantly and might even not be perceived affecting the control, as consecutive values for force and control signals tend to be similar.

Network and cloud availability are concerns due to mobile nature of the system. Wireless connectivity and seamless AP migration are also necessary to guarantee sufficient RSSI without downtimes. Again, during validation experiments, as the walker moved away from the wireless AP, latency times increased and connection was eventually lost. Such issues must be dealt with, and CloudWalker should be employed taking into consideration the wireless communication technology. The use of commercial WiFi demands an adequate distribution of AP and handover strategies to avoid downtimes during AP migration.

The proposed system was validated and CloudWalker showed itself capable of providing mobility assistance services even without no control algorithm being executed locally at the smart walker. The final conclusions for this work and future possibilities are discussed in the next Chapter.

6 Conclusions and Future Work

6.1 Conclusions and Contributions

This work proposed and validated a cloud-enable CPS for mobility assistance. A study was performed to evaluate the current state of art of research regarding mobility assistive devices and cloud robotics, looking for opportunities to integrate CPS and cloud robotics concepts into smart walker. It is shown that multiple features and functionalities commonly offered by smart walkers can be enhanced by leveraging the extended processing and storing capabilities of cloud computing platforms. By delegating computational tasks to the cloud, smart walkers can be designed to be more cost-efficient while extending battery autonomy and easing maintenance, mitigating deterrent factors to their acquisition and wide-spread utilization.

The cloud is able to provide different services to users of mobility assistive devices, tailoring the range of features offered according to individual needs. The construction of databases and the utilization of large amounts of data can assist healthcare professionals in monitoring the evolution of gait parameters in patients and even in alerting emergency staff when identifying potential health problems. Besides leveraging the cloud to perform only algorithms smart walkers are currently making use of, sensorial inputs from multiple robots can be used for global map construction, improved path planning with up-to-date information, and task collaboration.

A cloud-enable CPS for mobility assistance, named UFES CloudWalker, was proposed and had its overall architecture described. The system was firstly implemented by integrating a smart walker, the UFES Smart Walker, and a cloud platform running on a data center inside the university campus. To investigate how network and cloud quality of services parameters could affect the use of such system, especially from an end-user point of view, a preliminary validation stage was performed in which the communication network and the cloud were locally emulated at the smart walker. Results pointed to the feasibility of implementing and using the system even under unfavorable QoS conditions.

The UFES CloudWalker was implemented and validated against a scenario involving a mobility assistance service. The implemented system showed itself to be capable of providing assistive services even when all control algorithms are remotely executed. These results point to the emergence of a new generation of smart walkers, designed to leverage cloud computing concepts to provide an extended range of services to users, relatives, and healthcare professionals.

6.2 Publications

During the realization of this work, the following paper was published and is directly linked to the work here presented:

- MELLO, R.; JIMENEZ, M.; SOUZA, F.; RIBEIRO, M. R. N.; FRIZERA-NETO, A. Impacto da Latência em Sistemas Cyber-Físicos em Nuvem: Métricas Subjetiva e Objetiva para Andador Robótico. In: Anais do XIII Simpósio Brasileiro de Automação Inteligente, pp 1120-1125, 2017.

Moreover, this work also led to the submission of the following paper, currently under review:

- MELLO R.; JIMENEZ, M.; RIBEIRO, M. R. N. Ribeiro; GUIMARÃES, R. L.; FRIZERA-NETO, A. On Human-in-the-Loop Cyber-Physical Systems in Future Healthcare: A Cloud-Enabled Mobility Assistance Service. IEEE Systems Journal, 2018 *(Under review)*.

Lastly, the author participated on other research activities in the duration of this work and contributed in the papers listed bellow:

- SOUZA, F.; FRIZERA-NETO, A.; JIMENEZ, M.; MELLO, R. Desvio de obstáculos móveis em andador inteligente. In: Actas del IX Congreso Iberoamericano de Tecnología de Apoyo a la Discapacidad, v.1., p.625-632, 2017 *(Published)*.
- JIMENEZ, M.; MELLO, R.; FRIZERA-NETO, A.; BASTOS, T. Assistive Device for Guiding Visually Impaired People With Mobility Disorders. In: 13th Annual ACM/IEEE International Conference on Human Robot Interaction, 2018. *(Accepted)*.
- MARTÍNEZ, V. M. G.; MELLO, R.; HASSE, P.; RIBEIRO, M. R. N.; MARTINELLO, M.; GUIMARÃES, R.; FRASCOLLA, V. Ultra Reliable Communication for Robot Mobility enabled by SDN Splitting of WiFi Functions. In: IEEE Symposium on Computers and Communications 2018, 2018. *(Under Review)*.

6.3 Future Work

After the realization of the presented work and the validation of the UFES CloudWalker, multiple lines of future work remain possible. The implemented system, as making use of the UFES Smart Walker as robotic platform, must be extended to contemplate the full range of sensors and interfaces that are already integrated into the smart walker. Moreover, to leverage the research group expertise, HRI strategies previously developed at NTA should be replicated using the cloud to process the necessary algorithms.

The development of novel control and HRI strategies involving UFES CloudWalker should encompass the assessment of different configurations regarding which tasks must be locally processed and which ones can be remotely deployed. Safety mechanisms must be developed and implemented to assure the safe use of the system. After such enhancements over the already implemented system, the clinical validation of the device shall be possible. This experimentation should involve mobility impaired individuals, validating UFES CloudWalker's capacity to provide mobility assistance services.

Computationally intensive algorithms can now be incorporated into the UFES Smart Walker system. Information collected by the LRF sensors can be used by probabilistic techniques, such as SLAM, to perform self-localization and create environmental maps, feeding map libraries stored by cloud services. Path planning algorithms can be implemented to assist in guidance tasks, interactively creating routes to be followed by the user, based on voice acquired destinations and previously constructed maps. This is specially eased by making use of the ROS middleware, a design choice of this work, as several of the available ROS packages aim at techniques such as SLAM, path planning, and voice recognition, allowing the researchers to focus more on the interaction and in novel strategies *per se*.

The current platform can also be integrated with new sensors and interfaces. For instance, a touch-screen can be used to receive inputs of the patients desires and intentions. A camera system can also be installed to explore computational vision in context-aware applications, exploring interaction strategies in which facial recognition or the ability to discern humans from other moving objects are important. Wearable sensors can also be coupled with the use of UFES CloudWalker for patient health monitoring.

Despite possible enhancements over the implemented platform, the full capabilities of the UFES CloudWalker also involve multi-robot cooperation for mapping and learning, and direct communication between the smart walkers and healthcare professionals. To do so, the cloud computing platform must be able to provide services connecting information from multiple robots scattered among different facilities and communicating with doctors and therapists via web applications, always assuring for the privacy of the patient and the safety of its personal data.

This work envisions a smarter, new generation of mobility assistive devices. Future work shall encompass the design of cost-effective robotics devices fully integrated with cloud computing platforms, able to provide a wider range of services for mobility impaired individuals, aiming at mitigating independence restrictions and improving quality of life.

Bibliography

[1] WHO. *World report on ageing and health.* [S.l.], 2015.

[2] FRIZERA-NETO, A. et al. Empowering and assisting natural human mobility: The simbiosis walker. *International Journal of Advanced Robotic Systems*, v. 8, n. 3, p. 34–50, 2011. ISSN 17298806.

[3] HIRVENSALO, M.; RANTANEN, T.; HEIKKINEN, E. Mobility difficulties and physical activity as predictors of mortality and loss of independence in the community-living older population. *Journal of the American Geriatrics Society*, v. 48, n. 5, p. 493–498, 2000. ISSN 00028614.

[4] BASTOS-FILHO, T. F. et al. Towards a new modality-independent interface for a robotic wheelchair. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 22, n. 3, p. 567–584, May 2014. ISSN 1534-4320.

[5] WAKITA, K. et al. Human-walking-intention-based motion control of an omnidirectional-type cane robot. *IEEE/ASME Transactions on Mechatronics*, v. 18, n. 1, p. 285–296, 2013. ISSN 10834435.

[6] MARTINS, M. M. et al. Assistive mobility devices focusing on Smart Walkers: Classification and review. *Robotics and Autonomous Systems*, v. 60, n. 4, p. 548–562, 2012. ISSN 09218890.

[7] CIFUENTES, C. A. et al. Multimodal Human–Robot Interaction for Walker-Assisted Gait. *IEEE Systems Journal*, v. 10, n. 3, p. 933–943, sep 2016. ISSN 1932-8184.

[8] LIU, Y. et al. Review on cyber-physical systems. *IEEE/CAA Journal of Automatica Sinica*, v. 4, n. 1, p. 27–40, jan 2017. ISSN 2329-9266.

[9] SUH, S. C. et al. *Applied cyber-physical systems.* [S.l.: s.n.], 2014. v. 9781461473. 1–253 p. ISSN 00189162. ISBN 978-1-4614-7335-0.

[10] LEE, I. et al. Challenges and Research Directions in Medical Cyber-Physical Systems. *Proceedings of the IEEE*, v. 100, n. 1, p. 75–90, jan 2012. ISSN 0018-9219.

[11] WAN, J. et al. Cloud robotics: Current status and open issues. *IEEE Access*, v. 4, p. 2797–2807, 2016. ISSN 21693536.

[12] QURESHI, B.; KOUBA, A. Five traits of performance enhancement using cloud robotics: A survey. *Procedia Computer Science*, v. 37, p. 220–227, 2014. ISSN 18770509.

[13] DUXBURY, a. S. Gait disorders and fall risk: detection and prevention. *Comprehensive therapy*, v. 26, n. 4, p. 238–245, 2000. ISSN 0098-8243.

[14] WOLFSON, L. et al. Gait assessment in the elderly: a gait abnormality rating scale and its relation to falls. *Journal of gerontology*, v. 45, n. 1, p. M12–M19, 1990. ISSN 0022-1422.

[15] VAN-HOOK, F. W.; DEMONBREUN, D.; WEISS, B. D. Ambulatory devices for chronic gait disorders in the elderly. *American Family Physician*, v. 67, n. 8, p. 1717–1724, 2003. ISSN 0002838X.

[16] CIFUENTES, C. et al. Sensor fusion to control a robotic walker based on upper-limbs reaction forces and gait kinematics. In: *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*. [S.l.: s.n.], 2014. p. 1098–1103.

[17] CIFUENTES, C. A. *Human-Robot Interaction Strategies for Walker-Assisted Locomotion*. 105 p. Tese (Doutorado), 2015.

[18] PRIEBE, J. R.; KRAM, R. Why is walker-assisted gait metabolically expensive? *Gait and Posture*, v. 34, n. 2, p. 265–269, 2011. ISSN 09666362.

[19] CONSTANTINESCU, R. et al. Assistive devices for gait in Parkinson's disease. *Parkinsonism & Related Disorders*, v. 13, n. 3, p. 133–138, 2007. ISSN 13538020.

[20] RODRIGUEZ, C. et al. Metodologia para Obtenção de Comandos de Navegação de um Andador Robótico Através de Sensores de Força e Laser. In: *XI Simpósio Brasileiro de Automação Inteligente (SBAI), 2013*. [S.l.: s.n.], 2013. p. 1–6.

[21] YU, H.; SPENKO, M.; DUBOWSKY, S. An Adaptive Shared Control System for an Intelligent Mobility Aid for the Elderly . An Adaptive Shared Control System for an Intelligent Mobility Aid for the Elderly. *Autonomous Robots*, v. 15, p. 53–66, 2003.

[22] WACHAJA, A. et al. Navigating blind people with walking impairments using a smart walker. *Autonomous Robots*, Springer US, v. 2015-Decem, p. 1–19, aug 2016. ISSN 0929-5593.

[23] KULYUKIN, V. et al. IWalker: Toward a rollator-mounted wayfinding system for the elderly. *2008 IEEE International Conference on RFID (Frequency Identification), IEEE RFID 2008*, v. 60, p. 303–311, 2008. ISSN 09218890.

[24] LEE, G.; OHNUMA, T.; CHONG, N. Y. Design and control of JAIST active robotic walker. *Intelligent Service Robotics*, v. 3, n. 3, p. 125–135, 2010. ISSN 18612776.

[25] FRIZERA-NETO, A. et al. Extraction of user's navigation commands from upper body force interaction in walker assisted gait. *Biomedical engineering online*, v. 9, p. 37, 2010. ISSN 1475-925X.

[26] LACEY, G. J.; RODRIGUEZ-LOSADA, D. The evolution of guido. *IEEE Robotics and Automation Magazine*, v. 15, n. 4, p. 75–83, 2008. ISSN 10709932.

[27] FRIZERA, A. et al. The Smart Walkers as Geriatric Assistive Device. The SIMBIOSIS Purpose. *Proceedings of the 6th International Conference of the International Society for Gerontechnology*, v. 7, p. 1–6, 2008. ISSN 1741203X.

[28] CERES, R. et al. Andador activo para la rehabilitación y el mantenimiento de la movilidad natural. *Imserso*, n. 2003, p. 3–8, 2004.

[29] MARTINS, M. et al. Real time control of the ASBGo walker through a physical human-robot interface. *Measurement: Journal of the International Measurement Confederation*, v. 48, n. 1, p. 77–86, 2014. ISSN 02632241.

[30] HIRATA, Y.; HARA, A.; KOSUGE, K. Passive-type intelligent walking support system RT Walker. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, v. 4, p. 3871–3876, 2004.

[31] HIRATA, Y.; KOMATSUDA, S.; KOSUGE, K. Fall prevention control of passive intelligent walker based on human model. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. [S.l.: s.n.], 2008. p. 1222–1228. ISBN 9781424420582.

[32] LACEY, G.; DAWSON-HOWE, K. M. The application of robotics to a mobility aid for the elderly blind. *Robotics and Autonomous Systems*, v. 23, n. 4, p. 245–252, 1998. ISSN 09218890.

[33] MORRIS, A. et al. A robotic walker that provides guidance. *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, v. 1, p. 25–30, 2003. ISSN 1050-4729.

[34] MARTINS, M. et al. Design, implementation and testing of a new user interface for a smart walker. In: *2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. [S.l.: s.n.], 2014. p. 217–222. ISBN 978-1-4799-4254-1.

[35] HELLSTRÖM, T. et al. An intelligent rollator for mobility impaired persons, especially stroke patients. *Journal of medical engineering & technology*, p. 1–10, 2016. ISSN 1464-522X.

[36] PANTELERIS, P.; ARGYROS, A. A. Vision-based SLAM and moving objects tracking for the perceptual support of a smart walker platform. In: *Lecture Notes in Computer*

Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). [S.l.: s.n.], 2015. v. 8927, p. 407–423. ISBN 9783319161983. ISSN 16113349.

[37] PAULO, J. et al. An innovative robotic walker for mobility assistance and lower limbs rehabilitation. *ENBENG 2017 - 5th Portuguese Meeting on Bioengineering, Proceedings*, 2017.

[38] CHALVATZAKI, G. G. et al. Towards an intelligent robotic walker for assisted living using multimodal sensorial data. In: *Proceedings of the 2014 4th International Conference on Wireless Mobile Communication and Healthcare - "Transforming Healthcare Through Innovations in Mobile and Wireless Technologies", MOBIHEALTH 2014*. [S.l.: s.n.], 2015. p. 156–159. ISBN 9781631900143.

[39] LACEY, G. Context-Aware Shared Control of a Robot Mobility Aid for the Elderly Blind. *The International Journal of Robotics Research*, v. 19, n. 11, p. 1054–1065, 2000. ISSN 0278-3649.

[40] MACNAMARA, S.; LACEY, G. A smart walker for the frail visually impaired. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, v. 2, p. 1354–1359, 2000. ISSN 1050-4729.

[41] HSIEH, Y. H. et al. Motion Guidance for a Passive Robot Walking Helper via User's Applied Hand Forces. *IEEE Transactions on Human-Machine Systems*, v. 46, n. 6, p. 869–881, 2016. ISSN 21682291.

[42] GRAF, B. An Adaptive Guidance System for Robotic Walking Aids. *Journal of Computing and Information Technology*, p. 109–120, 2008. ISSN 1330-1136.

[43] HUANG, C. et al. Shared navigational control and user intent detection in an intelligent walker. In: *AAAI Fall 2005*. [S.l.: s.n.], 2005. ISBN 1577352483.

[44] FRIZERA-NETO, A. et al. Smart Walkers: Advanced Robotic Human Walking-Aid Systems. In: VITIELLO, N. et al. (Ed.). *Intelligent Assistive Robots*. [S.l.: s.n.], 2015. v. 106, p. 1–40. ISBN 978-3-319-12921-1.

[45] CORTÉS, U. et al. Towards an intelligent service to elders mobility using the i-Walker. In: *AAAI Fall Symposium - Technical Report*. [S.l.: s.n.], 2008. FS-08-02, p. 32–38. ISBN 9781577353942.

[46] SPENKO, M.; YU, H.; DUBOWSKY, S. Robotic personal aids for mobility and monitoring for the elderly. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 14, n. 3, p. 344–351, 2006. ISSN 15344320.

[47] CHAN, A. D. C.; GREEN, J. R. Smart rollator prototype. *MeMeA 2008 - IEEE International Workshop on Medical Measurements and Applications Proceedings*, p. 97–100, 2008.

[48] ZHANG, L. et al. The Development of Walking Assistant Robot for The Elderly. *Materials, Mechatronics and Automation, Pts 1-3*, v. 467-469, p. 1893–1898, 2011. ISSN 16800737.

[49] HASHIMOTO, H. et al. Walker with hand haptic interface for spatial recognition. *International Workshop on Advanced Motion Control, AMC*, v. 2006, p. 311–316, 2006.

[50] GHARIEB, W. Intelligent Robotic Walker Design. *ICGST Conference on Artificial Intelligence and Machine Learning*, n. 1, p. 6, 2006.

[51] FRIZERA-NETO, A. et al. Extraction of user's navigation commands from upper body force interaction in walker assisted gait. *Biomedical engineering online*, v. 9, p. 37, 2010. ISSN 1475-925X.

[52] PAGE, S. et al. Fast embedded feet pose estimation based on a depth camera for smart walker. *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, p. 4224–4229, 2015.

[53] PAULO, J.; PEIXOTO, P.; NUNES, U. J. ISR-AIWALKER: Robotic Walker for Intuitive and Safe Mobility Assistance and Gait Analysis. *IEEE Transactions on Human-Machine Systems*, v. 47, n. 6, p. 1110–1122, 2017. ISSN 21682291.

[54] GLAS, D. F. et al. Laser tracking of human body motion using adaptive shape modeling. *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, p. 602–608, 2007.

[55] BORDEL, B. et al. Cyber–physical systems: Extending pervasive sensing from control theory to the Internet of Things. *Pervasive and Mobile Computing*, v. 40, p. 156–184, 2017. ISSN 15741192.

[56] RAJKUMAR, R. et al. Cyber-physical systems: The next computing revolution. In: *Design Automation Conference (DAC), 2010 47th ACM/IEEE*. [S.l.: s.n.], 2010. p. 0–5.

[57] GOLDBERG, K.; KEHOE, B. Cloud Robotics and Automation:A Survey of Related Work. *... Tech. Rep. UCB/EECS-2013-5*, p. 1–12, 2013.

[58] YOSHIKANE, N. et al. First Demonstration of Geographically Unconstrained Control of an Industrial Robot by Jointly Employing SDN-based Optical Transport Networks and Edge Compute. *Optoelectronics and Communications Conference (OECC)*, v. 1, n. c, p. 1–3, 2016.

[59] KEHOE, B. et al. A survey of research on cloud robotics and automation. *IEEE Transactions on Automation Science and Engineering*, v. 12, n. 2, p. 398–409, 2015.

[60] QUIGLEY, M. et al. ROS: an open-source Robot Operating System. *Icra*, v. 3, n. Figure 1, p. 5, 2009. ISSN 0165-022X.

[61] WAIBEL, M. et al. RoboEarth: A World Wide Web for Robots. *Robotics and Automation Magazine*, v. 18, n. 2, p. 69–82, 2011. ISSN 1070-9932.

[62] MOHANARAJAH, G. et al. Rapyuta: A Cloud Robotics Platform. *IEEE Transactions on Automation Science and Engineering*, v. 12, n. 2, p. 481–493, 2015. ISSN 15455955.

[63] HU, G.; TAY, W. P.; WEN, Y. Cloud robotics: Architecture, challenges and applications. *IEEE Network*, v. 26, n. 3, p. 21–28, 2012. ISSN 08908044.

[64] HOSSAIN, M. S. Cloud-supported cyber-physical localization framework for patients monitoring. *IEEE Systems Journal*, v. 11, n. 1, p. 118–127, mar 2017. ISSN 19379234.

[65] ZHANG, Y. et al. Health-CPS : Healthcare Cyber-Physical System Assisted by Cloud and Big Data. *IEEE Systems Journal*, v. 11, n. 1, p. 88–95, mar 2017. ISSN 1932-8184.

[66] CHEN, M. et al. Smart Clothing: Connecting Human with Clouds and Big Data for Sustainable Health Monitoring. *Mobile Networks and Applications*, Mobile Networks and Applications, v. 21, n. 5, p. 825–845, 2016. ISSN 15728153.

[67] DOGMUS, Z.; ERDEM, E.; PATOGLU, V. RehabRobo-Onto: Design, development and maintenance of a rehabilitation robotics ontology on the cloud. *Robotics and Computer-Integrated Manufacturing*, v. 33, p. 100–109, jun 2015. ISSN 07365845.

[68] RADU, C.; CANDEA, C.; CANDEA, G. Towards an Assistive System for Human. In: *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '16*. [S.l.: s.n.], 2016. p. 1–4. ISBN 9781450343374.

[69] LI, H.-J. J.; SONG, A.-G. G. Architectural Design of a Cloud Robotic System for Upper-Limb Rehabilitation with Multimodal Interaction. *Journal of Computer Science and Technology*, v. 32, n. 2, p. 258–268, mar 2017. ISSN 10009000.

[70] FIORINI, L. et al. Enabling personalised medical support for chronic disease management through a hybrid robot-cloud approach. *Autonomous Robots*, Springer US, v. 41, n. 5, p. 1263–1276, jun 2017. ISSN 15737527.

[71] FU, J. et al. A novel mobile-cloud system for capturing and analyzing wheelchair maneuvering data: A pilot study. *Assistive Technology*, v. 28, n. 2, p. 105–114, 2016. ISSN 19493614.

[72] SALHI, K. et al. Navigation assistance to disabled persons with powered wheelchairs using tracking system and cloud computing technologies. *Proceedings - International Conference on Research Challenges in Information Science*, v. 2016-Augus, 2016. ISSN 21511357.

[73] BENAVIDEZ, P. et al. Design of a home multi-robot system for the elderly and disabled. *2015 10th System of Systems Engineering Conference, SoSE 2015*, p. 392–397, 2015.

[74] ITURBURU, M. et al. User Needs and Requirements for the Mobility Assistance and Activity Monitoring Scenario within the RAPP Project. v. 351, p. 105–117, 2015. ISSN 21945357.

[75] TSARDOULIAS, E. G. et al. Towards an integrated robotics architecture for social inclusion – The RAPP paradigm. *Cognitive Systems Research*, Elsevier B.V., v. 43, p. 157–173, jun 2017. ISSN 13890417.

[76] PONS, J. L. *Wearable Robots: Biomechatronic Exoskeletons*. Chichester, UK: John Wiley & Sons, Ltd, 2008. ISSN 1751-8113. ISBN 9780470987667.

[77] FLEMISCH, F. O. et al. Towards cooperative guidance and control of highly automated vehicles: H-Mode and Conduct-by-Wire. *Ergonomics*, v. 57, n. 3, p. 343–360, mar 2014. ISSN 1366-5847.

[78] MUNIR, S. et al. Cyber Physical System Challenges for Human-in-the-Loop Control. *The 8th International Workshop on Feedback Computing*, 2013.

[79] SHAH, T. et al. Remote health care cyber-physical system: quality of service (QoS) challenges and opportunities. *IET Cyber-Physical Systems: Theory & Applications*, v. 1, n. 1, p. 40–48, 2016. ISSN 2398-3396.

[80] HAMMER, F.; EGGER-LAMPL, S.; MOLLER, S. Position Paper: Quality-of-Experience of Cyber-Physical System Applications. In: *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*. [S.l.: s.n.], 2017. p. 1–3. ISBN 9781538640241.

[81] ITU-R M.2083-0. *IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond*. [S.l.], 2015. 21 p.

[82] DOMINICINI, C. K. et al. VirtPhy: A fully programmable infrastructure for efficient NFV in small data centers. In: *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2016*. [S.l.: s.n.], 2017. p. 81–86. ISBN 9781509009336.

[83] CHEN, M. et al. Wearable 2.0: Enabling Human-Cloud Integration in Next Generation Healthcare Systems. *IEEE Communications Magazine*, v. 55, n. 1, p. 54–61, 2017. ISSN 01636804.

[84] PETÄJÄJÄRVI, J. et al. Evaluation of LoRa LPWAN Technology for Indoor Remote Health and Wellbeing Monitoring. *International Journal of Wireless Information Networks*, v. 24, n. 2, p. 153–165, 2017. ISSN 15728129.

[85] BENAVIDEZ, P. et al. Cloud-based realtime robotic Visual SLAM. In: *9th Annual IEEE International Systems Conference, SysCon 2015 - Proceedings*. [S.l.: s.n.], 2015. p. 773–777. ISBN 9781479959273.

[86] REDMON, J. et al. You Only Look Once: Unified, Real-Time Object Detection. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. ISBN 978-1-4673-8851-1. ISSN 01689002.

[87] CAO, Z. et al. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In: *CoRR*. [S.l.: s.n.], 2016. ISBN 978-1-5386-0457-1.

[88] VALADÃO, C. et al. A new controller for a smart walker based on human-robot formation. *Sensors (Switzerland)*, v. 16, n. 7, p. 1–26, 2016. ISSN 14248220.

[89] KOUBAA, A. ROS As a Service : Web Services for Robot Operating System. *Journal of Software Engineering for Robotics*, v. 1, n. 1, p. 123–126, 2015.

[90] WAHBA, M. et al. A ROS-Simulink Real-Time Communication Bridge Using UDP With a Driver-in-the-Loop Application. In: *ASME 2016 Dynamic Systems and Control Conference*. [S.l.: s.n.], 2016. p. 1–8. ISBN 978-0-7918-5070-1.

[91] ITU-T. *P.800: Methods for subjective determination of transmission quality*. [S.l.], 1996. v. 800.

[92] MONLLOR, M. et al. Control de Seguimiento de Caminos Para Robots de Asistencia. In: *XVII Reunión de Trabajo en Procesamiento de la Información y Control - RPIC 2017*. [S.l.: s.n.], 2017. p. 1–6.

[93] GERAVAND, M. et al. An Integrated Decision Making Approach for Adaptive Shared Control of Mobility Assistance Robots. *International Journal of Social Robotics*, Springer Netherlands, v. 8, n. 5, p. 631–648, 2016. ISSN 18754805.

[94] ANDALUZ, V. H. et al. Adaptive Dynamic Path Following Control of an Unicycle-Like Mobile Robot. In: _____. *Intelligent Robotics and Applications: 4th International Conference, ICIRA 2011, Aachen, Germany, December 6-8, 2011, Proceedings, Part I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 563–574. ISBN 978-3-642-25486-4.