Designing a Low-Cost Web-Controlled Mobile Robot for Ambient Assisted Living

Yvon Autret, David Espes, Jean Vareille and Philippe Le Parc

Université Européenne de Bretagne, France
Université de Brest
Laboratoire en Sciences et Techniques de l'Information (LabSTICC UMR CNRS 6285)
20 av. Victor Le Gorgeu, BP 809, F-29285 Brest, France
Email: {yvon.autret, david.espes, jean.vareille, philippe.le-parc}@univ-brest.fr

Abstract—**In this paper, we focus on a Web-controlled mobile robot for home monitoring, in the context of Ambient Assisted Living. The key point is low-cost as dependent person have often tiny budget. The robot is built from standard components to reduce the cost of the hardware. A large part of the system is deported to the Internet to minimize the required software on the robot. Two low-cost positioning systems are also provided to make the robot more usable. The first one uses Received Signal Strength Indicators (RSSI), and the second one uses Infrared (IR) LEDs and an IR webcam. The result is a small robot that can be used inside or outside the house.**

***Keywords*–Mobile robot; Home monitoring; Ambient Assisted Living; Web control; WebSocket; RSSI positioning; IR positioning.**

I. Introduction

This paper is an extension of [1] published in UBICOMM2014. Web-controlled mobile devices are more and more used in ubiquitous environments [2][3][4][5]. Small monitoring robots such as the WowWee Rovio can be used [6]. Web control is not really new, but recent improvements of network performance has led to the emergence of Service Robotics [7]. Services Oriented Architectures (SOA) [8] start to be used to control physical devices [9].

Our aim is to use these approaches to control mobile home robots designed for Assisted Ambient Living (AAL) environments. For us, a typical application is helping elderly people who live in their houses and have sometimes some difficulties to move. A mobile home robot carrying a camera could help them monitoring their house either indoors or outdoors. The mobile home robot could also be used by care helpers or relatives, as a moving phone to communicate with the inhabitants of a house, or as a monitoring device.

In such an AAL environment, the total cost of the mobile home robot is the first key point. It must be kept as low as possible especially if it is an AAL environment for elderly people who often have tight budgets. This means that the mobile home robot must be built by using low-cost commercial components. Moreover, we always keep in mind that mechanical failures are unavoidable and reliability is a major key point. The basic mobile home robot design must be as simple as possible, while remaining flexible enough to carry sophisticated sensors.

The second main key point is software and network configurations. The mobile home robot should be plug an play. This means that software and network configurations should be reduced as much as possible. Deporting a part of the system to the Internet can be a solution if it helps to get a reliable plug and play system.

The third key point is security and access control. A Web-controlled mobile home robot can be used from anywhere in the world, but the interior of a house must not be seen by unauthorized users. It is necessary to avoid any intrusive access. More, in case of network failure, the mobile home robot should also be able to properly stop its current action and wait for a new order.

The fourth key point is the autonomy of the battery. The robot should have an autonomy close to one hour when moving, and automatically come back to a charging dock when the battery is low.

Another important aspect is the positioning of the mobile platform. Even if it is considered in this paper that the mobile home robot is remotely controlled all the time, i.e., not autonomous, having a rough idea of his position in the house could be helpful in several situations and could be used to improve user experience.

In this paper, the second section presents a mobile home robot solution based on a commercial low-cost robot and we discuss the advantages and the disadvantages. This lead us to the design of a mobile home robot built from commercial components such as a low-cost robotic platform and a smartphone to control it. In the third part, we present the distant control system and its performance. In the fourth part, we present two solutions for a low-cost positioning system, one using the Received Signal Strength Indicator, another using IR LEDs and cameras. The paper finishes by a conclusion and some perspectives.

II. Designing a home robot for an AAL environment

The first question to tackle with, is the AAL environment. Are people we plan to help totally dependent, or do they only suffer of little dependency? Scales are available to estimate the degree of dependency of old people. One well known is the "Bristol Activities of Daily Living Scale" [10]. It is a 20-item survey, proposed to measure the ability to perform daily activities as preparing a meal. The score goes from 0 (independent) to 60 (totally dependent). Our objective is to provide robots for people having a rather small score, around 15 points. Plugging or disconnecting a device like a mobile phone, preparing alone a meal, feeding a pet, should be feasible tasks. We target people for whom one device more at home is not a huge problem.

The second question to tackle with, is the usages of the mobile home robot (use-cases):

1) The old or dependent person may use the robot to project themselves in their home. It means that with a simple terminal (tablet or mobile), they can control the robot to see what is happening in another room (or floor). For example, they can check whether a noise is normal or what their pet is doing.
2) The family or the carers could take the control of the robot to discuss with the old or dependent person
3) The family or the carers could aso use the robot to verify some properties such as doors or windows are closed, gas is off and so on... These checks could be crucial in case of person suffering for example of light Alzheimer disease.
4) The family or the carers may also take the control in case of emergency to understand the situation very quickly and to be able to react in a proper manner.

In the next section, we will first describe the context of use, that we will briefly analyze, then we will list the requirements, and finally we will suggest several useful extended abilities.

A. Context of use and requirements

Basically the context of use is composed by a dependent person, the home, a telephone and an Internet connection, and the availability of electricity of course. Our purpose is to give services that contribute to the well being and can help, minimizing the modifications of the home.

Generally, dependent people living at home need a service of tele-assistance paid by subscription. This service is provided using a telephone basis, a wristband or a neck pendant. In case of emergency, for example, a fall of the dependent person, it is expected that he or she presses on the button of the worn accessory to call for help. The alarm is automatically relayed by a transmitter connected to the phone line, that calls the managing organization where someone reacts, questioning the dependent person in the aim to diagnose the problem. The usage of wireless transmission between the wristband or the pendant and the transmitter induces constraints like a maximal distance that could be reduced by the presence of obstacles or a difference of altitude. A one floor home without thick walls is preferred. This system is only used in case of emergency, although the phone line could be used for a wide range of new services, especially through Internet.

The classical services of tele-assistance are not usable to maintain the social links and to help the dependent person in a run-of-river manner. There are remaining problems like the localization of the person in case of problem, and the visual evaluation of its state.

Obviously, the presence of the telephone line permits a high rate access to Internet, using ADSL for example, at little extra cost. A second dedicated box for the ambient assistance could be connected to the first one, that would be able to offer extended wireless connectivity, like Bluetooth, ZigBee, IR, 3G/4G, etc. a storage space and the software for the data transmission and data processing. The issues are how to exploit this connectivity to solve the problems previously described, and what for devices and components are needed to insure the expected services. The social interactions are mainly performed in presence. The nowadays technologies allow the extension of the sense of presence, it means that for a lot of situations, a telepresence could be enough. The aim of such telepresence is to introduce a mobile telepresence in the home "where you are and when you need it", rather than making actions in place of the people, because they are not totally dependent, and they need more conversations, advices and stimulations, than acting help. The mobility could be achieved at least through three ways: by a smartphone worn by the person, by a device carried by a mobile platform, or by devices installed in each room of the home. The first option is not suitable here because the smartphones are designed for valid people able to use one arm while they are walking. The third option avoids the problem of the discontinuous power supply of the devices, is reliable because the failure of one of the sensors implies only a partial disability of the services, but needs some work in the home, and is heavy intrusive. The second option needs no work in the home, is cheaper and less intrusive, but needs to recharge the mobile device, and is less reliable. So, we have chosen to develop of the second option. Nevertheless, the three options are not antinomic.

A low cost robot could be a device added to the emergency-calling service, less intrusive than fixed cameras. The audio and video streams should be secured, using https or a VPN.

The required properties are:

- the telepresence services should be available where the person is, and when requested,
- the telepresence services could be activated remotely,
- the sensors are carried by a mobile platform,
- the dependent person could decide to activate it or not,
- the recharging is not automatic,
- when empty, the mobile platform asks to be charged, that stimulates the person to perform this action,
- the autonomy is sufficient to need only one charge per day in normal use, two charges maximum,
- the device is highly maintainable,
- the hardware maintenance could be done by people with a moderate knowledge in robotics, like caregivers or daily visitors, e.g., postmen.
- a modular design that permits replacements of subsystems instead the standard exchange of the whole,
- the software maintenance and the reconfiguration could be made remotely.
- the life span of the mobile platform should reach the life span of a car (or a pet),
- the price should be small in comparison to the annual cost of the subscriptions (phone, emergency-calling service), and within the mean price of a smartphone (€300),

When activated:

- the system is able to listen and transmit the sound,
- the system is able to watch (day and night),
- the mobile platform is able to move over all the kind of floor,
- it is able to cross flat obstacles like carpets,
- it is able to follow the person,
- the maximum mass should be smaller than that of a walker ($\leq$ 3 kg).

The features of our proposal are: a size of two shoes, a mass of 1.15 kg, an autonomy depending on the traveling distance, at least 30 minutes but could be a day for a quiet usage. A low cost robot cannot carry heavy batteries, many computers and a lot of sensors. We prefer a modular approach with standard and cheap components easy to associate. Each component does not bring radical novelty, but the robot could offer new services, that will be the novelty. Currently, the robot is more a platform that permits development of services using the resources of the embedded systems and those of the servers and the clients. Maintaining the permanence of the services is a crucial issue that we propose to solve using a modular approach both for the hardware as well for the software. We would like to test the flexibility and the agility of the "applications store" model of use.

The question of the acceptability cannot be avoided. We do not target to give a great autonomous behavior to the robot, because if the robot would decide itself to move, it could disturb the people present and induce a thought as "someone is watching me without my authorization". The behaviors like those of a pet are also not targeted.

Beyond the possibilities mentioned above, several extended functionalities could be added. The computational power embedded could allow to implement learning algorithms, based for instance on reduced simulated neural networks. Many recent smartphones include a voice recognition, that uses neural networks. We could add a sound analyzer in real time that would listen to the breathing of the person for the detection of apneas, or would detect an intrusion. One of the most important improvement would be an accurate localization and positioning system. Such an ability would open the door to useful characterization of the environment inside the home, like tribo-analysis of the floor, liquid detection, obstacle detection and 3D modelization. Another usage would be the localization and observation of a pet.

In Section II-B, commercial home robots are described and evaluated, while in Section II-C an alternative proposal is presented.

B. Commercial home robots

Several commercial robots are available. We present here a short description of four of them:

- The Miabot [11] robot is rather small (about 10 cm long) and fast (3.5 m/s). It has a built-in Bluetooth connection and must be connected to a local central computer to be web-controlled. Even if it was not really designed for that, it can carry a small camera or other sensors.
- Another interesting robot is the WowWee Rovio [6]. It includes a mobile base, a mobile camera and a Wi-Fi connection. Its size is 30 x 35 x 33 cm. It can be remotely controlled from anywhere in the world. When the battery is low, it is supposed to come back automatically to its charging dock. Its total cost is about €300, which is acceptable for our purpose. The WowWee Rovio is an interesting robot for an AAL environment, but it is not an open robot and it is difficult to add new features. In case of failure, the WowWee Rovio is also difficult to repair. For example, such a common operation as replacing batteries, requires soldering.
- The Jibo social robot [12] should be available soon, may be by the end of 2015, and should cost about €500. It is about 28cm tall and 15cm wide. Its face consists of a touchscreen and interaction is possible by poking it. It is designed to recognize the faces of the family members. It cannot move but can be motor-driven through 360 degrees. Thus, it could be a very interesting robot to monitor, not a whole house, but a room.
- The Romo [13] robot uses a smartphone to control the motors. It can be remotely controlled from anywhere by using the smartphone connectivity. When used as a toy, it can perform autonomous missions. When controlled remotely, it provides an interesting telepresence functionality. The physical separation of the smartphone and the mechanical base makes it easy to repair. The mobile base costs about €130, but the price of a smartphone (minimum Iphone 4S - €250) should be added, which lead to a minimal total cost of €380.

We can now examine the previous robots to see if they are suitable. The first key point is the camera, which is essential. The second essential key point is web-control.

Another main key point is the cost. It is difficult to evaluate the maximal possible cost of a home robot. However, it can be seen that commercial robots about €300 can be sold. €500 robots can also be sold if they are very well designed and powerful. We think that we must remain cautious for more expensive robots. Thus, we will consider that a €300 robots remains acceptable in an AAL environment where people often have narrow budgets.

The selling price is not the only thing that must be taken into account. The robots may have breakdowns and repairing must remain cheap. This is difficult to achieve if the robot is not modular. More, the technology evolves and a robot may rapidly become out of age. For example, a given robot may become useless if a new positioning technology is discovered. We think that replacing the whole robot because one component has become obsolete is not a good solution. The robot should be easily upgraded at minimal cost. For example, if the mechanical base is still suitable, there is no reason to replace it. From our point of view, the lifetime should reach at least five or ten years.

Automatic battery charging is an interesting, but not essential key point. In an AAL environment we can guess that the robot will not be used continuously. It should be used a limited number of times every day, and each use should last minutes rather than hours. Above, we supposed that the person using the robot is able to plug the battery to recharge it. Charging batteries once a day should be sufficient, and performed by the inhabitants.

Positioning capabilities are also mentioned in Table I. Rovio gets a "yes/no" as it is designed to get back automatically to its base station, but it does not know where it is.

In Table I, we can see that some common commercial robots already take into account most of those key points except positioning. However, modularity is very weak. In the

TABLE I. Comparison of commercial low-cost robots

	Miabot	Rovio	Jiho	Romo
Camera	no	yes	yes	yes
Web control	no	yes	yes	yes
Price <= €300	N/A	yes	no	no
Ease to repair, modularity	no	no	no	yes
Automatic battery charging	no	yes	yes	no
Positioning	no	yes/no	no	no

next section we will focus on that point to propose a fully modular robot, based on a smartphone, a control module and a mechanical base.

C. Using a smartphone, a control module, and a mechanical base

Using a smartphone as in [13], may help simplifying the building of a modular home robot. The smartphone is usually reliable and includes a webcam, Wi-Fi, Bluetooth and a touch screen. Wi-Fi will provide Web-control capabilities. Bluetooth will make it possible to control the mechanical base. The webcam is not perfect for our purpose. It consumes much power and highly reduces the autonomy. Thus, we prefer an additional, independent and external infrared webcam (IR). The whole system will be more modular without really increasing the total cost because a very cheap smartphone can be used to handle only Wi-Fi and Bluetooth. More, we will see later that an external IR camera is very useful to implement a positioning system. It also provides night vision.

The touch screen provided by the smartphone is an important part of the robot because it can provide information about the state of the robot. It is useless when everything is working. When something goes wrong, it is useful to show failures in a user interface. The touch screen will also be used to set the initial configuration (Wi-Fi, Bluetooth, and the distant server address).

An open mechanical robotic platform, which includes two tracks has been used. It is a 4WD Rover 5 from RobotBase. Its size is close to that of the WowWee Rovio. When powered, it can move forward or backward and turn. The maximum speed is 0.3 m/s. That speed is optimal in an AAL environment. It does not frighten inhabitants, and a standard room can be crossed in about 10 seconds. The Rover 5 is strong enough to carry up to two kilograms.

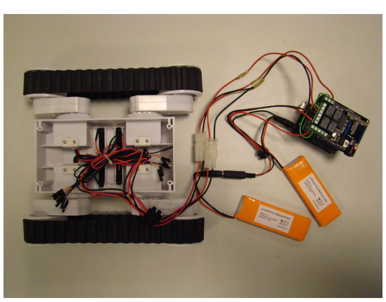

Figure 1. Components of the Web-controlled home robot.

Our control module is based on an Arduino microcontroller [14] that controls the mechanical base and communicates with the smartphone. Several Arduino shields are available to monitor the working speed and direction of the motors. We can use either a relay shield including four relays, or a motor shield based on a voltage regulator such as 78M05. An additional Arduino shield is required to allow Bluetooth communication between the Arduino and the smartphone.

The main advantage of our solution is its modularity. The home robot only includes commercial components (see Fig. 1, Fig. 2 and Fig. 3):

- A mobile Rover 5 robot used as a mechanical base (€60)
- An Android smartphone (less than €80)
- An IR Wi-Fi webcam (about €50)
- A control module including an Arduino UNO (€20), a Bluetooth shield (€10), a XBee shield (€15), and a motor command shield (€20)
- Batteries (one for the Arduino, one for the webcam, and one for the motors, 3x€15)

The control module is not yet a commercial component. Here, it is built by using commercial components that could be easily integrated on a single board. As soon as it would be done, the control module would be much cheaper, and the robot could easily be built and repaired at home. Building the robot would correspond to connecting the control module to the mechanical base, and to putting over a smartphone and a Wi-Fi webcam. This can be easily achieved if the mechanical base includes a plastic shell, plugs for the control module, and places to put the smartphone and the webcam. When a component fails, it can be easily replaced at home without replacing the whole robot. In an AAL environment, that task could be performed by caregivers.

Prices given are public prices that includes prices of the product itself but also commercial margin from the resellers. They are given here to fix ideas, but in the case of a mass production, the total cost will be lower. More we can also use an old smartphone, which has became useless or a recycled one.

The current total cost (€300), smartphone included, is comparable to that of a WowWee Rovio although our robot is a prototype. The reliability of our mobile home robot is significantly higher than that of a Rovio. In case of failure, we only need to replace one component. Moreover, the diagnosis is very easy because each component can be individually tested.

When using 2000 mAh lithium batteries, we have a 30 min autonomy when the robot is continuously moving. We have several hours of battery life when the robot is waiting for commands. Automatic battery charging is not available on our prototype, because the lack of localization procedures makes it difficult to achieve.

Apart this last point, we consider that our proposal respects constraints described above. The next part of this paper will concentrate more on software aspects.

III. A DISTANT CONTROL SYSTEM

We propose to use a Web server to reduce home configurations and installations. The Web server will be used to control

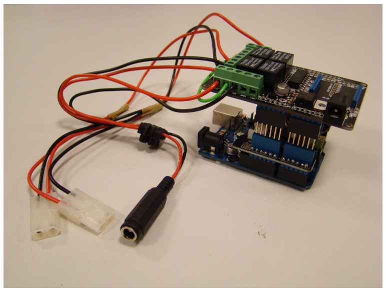

Figure 2. The Arduino command module.

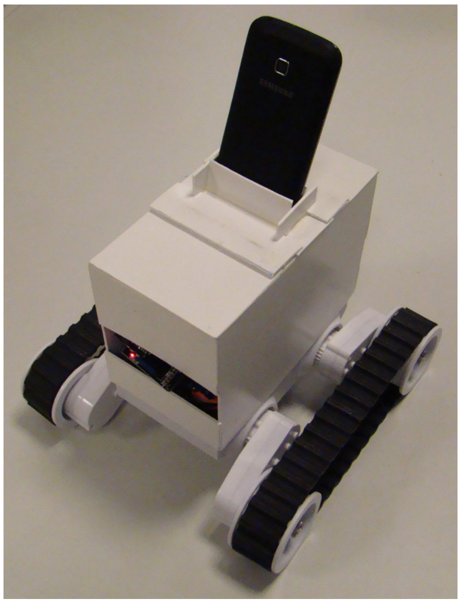

Figure 3. The Web-controlled home robot.

the robot. A user interface running on a standard Web Browser should make the robot usable without any special installation.

Using Hypertext Transfer Protocol (HTTP) is a solution to communicate with a distant server. Efficient HTTP Web servers such as Apache or Apache Tomcat are available. If the standard HTTP protocol easily handles problems such as client identification, it has severe limits when used for near real-time monitoring.

A. The HTTP limitations

The HTTP protocol is a stateless protocol, which was originally designed to get access to static HTML pages. Later, some web applications have implemented server-side sessions by using HTTP cookies. A Web server implementing sessions receives an HTTP request, establishes a connection with the server, executes the request, sends an HTTP response back, may keep a track of the HTTP request, and finally, releases the connection.

If a Web server is running on the robot, an identification sequence, which gives the right to monitor the robot through the Web server can be easily implemented. The communication can be secured by using the HTTPS protocol. The main problem is the execution time of a command sent to the robot. Let us take the example of an HTTP request, which should make the robot move for several seconds. As soon as the HTTP request is received on the server, the robot starts moving. If the robot moves for more than a few seconds, the HTTP response must be sent back before the robot has finished moving. In this case, the robot can get out of control.

This is a major problem because we must monitor a robot by using commands whose execution lasts about one second. A one meter trip would require sending at least three commands to a Rover 5 moving at 1km/h. Touring a house would require hundreds of commands. When a command is sent to a distant robot, a permanent connection is required. A moving robot left unsupervised just a few seconds can be dangerous. Presence and obstacle detectors working on the robot are never 100% reliable. This means that anyone who is monitoring from the outside or inside the house must have a permanent full control of the robot through the network. Moreover, the robot should be able to detect the smallest network failure, and to automatically adapt its behavior, for example, by reducing its speed.

This means that sending HTTP requests to a Web server running on the robot is not a good solution. We must continuously send HTTP requests to the robot to be able to detect network failures. That is a misuse of HTTP. Second, establishing a new connection from outside can be time consuming and sometimes takes several seconds. That is a risk we can not take. That is why we have chosen the WebSocket solution.

B. Using a WebSocket server

The WebSocket protocol was standardized in 2011 [15]. The communications are established by HTTP servers, and the communications may use TCP port 80 (or 443 when using secured communications). The client is responsible for making the connection by using an URL, consisting of a protocol, host, port, path, and optionally one or more additional parameters.

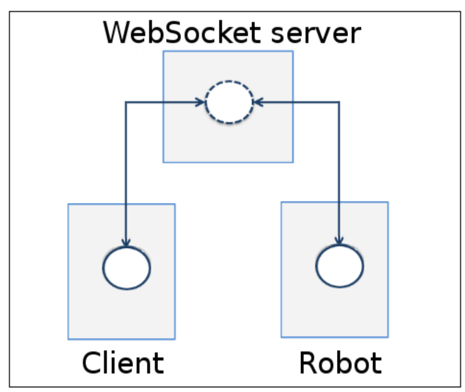

Figure 4. The WebSocket servlet.

The main advantage of WebSockets for our purpose is the fast responses coming from the server. That is due to the single connection that is established at the beginning of the communication. As soon as a connection is set, a bi-directional communication remains available. Full duplex communication over a single socket allows near real-time communication.

A standard Web browser can be used to monitor a robot through WebSockets. Most Web browsers now support WebSockets. Both the client and the robot send and receive information to and from the Web server through WebSockets. When a command is sent from the client to the Web server by using WebSockets, as soon as it is received on the server, it can be forwarded to the robot and executed. During the execution of the command on the robot, WebSockets are still used to send periodic acknowledges from the robot to the client, and from the client to the robot.

Thus, if the robot does not receive any acknowledgment, or receive them too late, it can modify its state. For example, it can reduce its speed if the network is too slow. If the network is no more working, the robot can stop properly, and remain waiting until the network is working again.

C. A WebSocket server

A WebSocket server greatly simplifies the installation of a Web-controlled home robot. The home robot just have to connect to the WebSocket server (Fig. 4). This does not require any special home configuration. An ordinary Wi-Fi connection can be used.

The well known Apache Tomcat Webserver now implements WebSockets. This means that we can use both the advantages of a standard Web server and those of WebSockets. A standard Tomcat application manages client and robot identification. The client uses an HTML form to ask for a robot. As soon as identification is successful on the server, a WebSocket communication becomes available between the client and the robot.

On the Tomcat server, we have a servlet to manage identification and robot allocation. We have also a WebSocketServlet to manage communication between the client and the robot.

The "manager" object is instantiated by the WebSocket server. From the robot point of view, it contains information about the client that is using the robot. From the client point of view, it contains information about the robot to control. The manager is stored as a Tomcat session object. It is a persistent object whose life duration is that of a session. A "manager" object is instantiated during the identification phase, when the client asks for a robot. Another "manager" object is instantiated when the robot connects to the WebSocket server When the WebSocket communications are set, the "manager" objects can be retrieved and modified to help clients and robots communicate. One client is allowed to send messages to one robot, and one robot is allowed to send messages to one client.

Both the client and the robot exchange messages by sending lines of text. For example, the client sends a line containing "forward" to make the robot move forward. Parameters can also be added in the line, for example to make the robot move forward for n seconds.

D. WebSockets on the robot

As seen above, the robot is controlled by the Arduino and the Arduino is controlled by an Android smartphone using a Bluetooth communication. We use the Tyrus API to connect the smartphone to the WebSocket server.

We use the Tyrus "ClientManager" class to set a connection between the robot and the WebSocket server. When messages come from the client, the "onMessage" method is triggered. The message is decoded and forwarded to the Arduino. During the execution of the command by the Arduino, the client and the smartphone periodically exchange messages to stop or slow down the robot in case of network failure. This program has been tested on Android 2.3 and Android 4.

E. WebSockets on the client

A WebSocket connection from the client to the server is only possible if the identification phase and robot selection has been successful. This is taken into account by the standard Apache Tomcat Webserver. As soon as a client is successfully registered on the distant Web server, a WebSocket connection is established. The client uses a Web page as user interface. The only thing required to use the user interface is a WebSocket compatible Browser. The user interface is managed by the distant Web server.

The Javascript "onMessage" function is triggered when a message comes from the WebSocket server. A widget such as a button in the user interface can trigger the "sendMessage" function and send commands to the robot.

F. Performance

In this section, we present an experimentation that illustrate the usability of our system and we justify our technological choices in term of communication medium.

For the experimentation, the server is connected to the local network of the laboratory, i.e., a gigabit Ethernet network. It is hosted to a public address so any user are able to access it from anywhere using just a web browser. Beside the server, one robot is available. The robot is equipped with an Arduino board, a Bluetooth shield and a smartphone. The Bluetooth shield is fully qualified to respect the Bluetooth version 2.0. Hence, the data rate is up to 2 Mbps. The smartphone is connected to the local network through a Wi-Fi connection. The Wi-Fi card on the smartphone is compliant to the IEEE 802.11g standard. Hence, the data rate is up to 54 Mbps.

In order to show the performance of the system, we define the following performance metrics:

- the *Round-trip time between components* is the time to receive a response after sending a request without counting the delay due to other components. For example, if the Arduino board sends a request to the smartphone, the round-trip time between these both components is the delay to receive a response without counting the delays imposed by smartphone-server connection and server-user connection.
- the *End-to-end round-trip time* corresponds to the time needed to receive a response after sending a request, i.e., it is the sum of the round-trip time between the whole components of the system. The increase of the end-to-end round-trip time degrades significantly the Quality of Service of applications and the Quality of Experiment of users.

Tests have been conducted from two different locations: our laboratory (i.e., LAN access) and the Military Technical Academy of Bucharest in Romania, i.e., Internet access, located about 2500 km from the laboratory. In both cases, the server is inside our laboratory. However, due to the flexibility of our architecture, the server could be hosted in the cloud. Each 30 minutes during one week, the round-trip time between components and the end-to-end round-trip time are measured. All the results represent the average of the measured

TABLE II. END-TO-END ROUND-TRIP TIME

Protocol	End-to-end round-trip time	
	Local (inside laboratory)	Distant (Romania)
HTTP	600 ms (±120 ms)	730 ms (±100 ms)
Web Sockets	175 ms (±60 ms)	190 ms (±50 ms)

times ± standard deviation. All times are expressed once the WebSocket connection is established.

In Table II, the end-to-end round-trip time is analyzed under two protocols (HTTP and WebSocket). The end-to-end round-trip time is an important parameter because it is the main criteria to determine if near real-time control is possible. To control a distant robot with an acceptable quality of experience, it is commonly accepted that the delay never exceeds 400 milliseconds [18]. We can see the HTTP protocol cannot guarantee the delay bound. Indeed, the time to establish the connection, to send a request and receive a response significantly exceeds the delay bound. In case a system requires the establishment of a TCP connection for each transaction, the near real-time control of the mobile robot is not possible. The WebSocket protocol is more suitable for near real-time control. Being designed to work well in the Web infrastructure, the protocol specifies that the WebSocket connection starts its life as a HTTP connection, offering backwards compatibility with no-WebSocket systems. The handshake of the WebSocket protocol has slightly the same time than the HTTP protocol. Once the connection is established, control frames are periodically sent to maintain the connection. Hence, the time is significantly reduced as compared with the HTTP protocol. For all scenarios, the end-to-end round-trip time does not exceed 300 milliseconds, which is quite acceptable to transmit QoS traffic.

TABLE III. ROUND-TRIP TIME RELATED TO ENTITY CONNECTIONS

Entity connection	Round-trip time	
	Local (inside laboratory)	Distant (Romania)
User - Server (Internet)	15 ms (±5 ms)	40 ms (±5 ms)
Server - Phone (Wi-Fi)	35 ms (±10 ms)	35 ms (±10 ms)
Phone - Robot (Bluetooth)	125 ms (±40 ms)	125 ms (±40 ms)

Table III presents the round-trip time related to robot's component inter-connections. It is interesting to see that the Internet delay, i.e., when the user is located in Romania, is almost negligible as compared with local access. Moreover, half of the Internet delay is due to the propagation time (if we assume a propagation speed of 200,000 km/s that is the common phase velocity of light in optical fibers). Nowadays, first-tier operators have 100 Gbps networks. In backbone networks, the bandwidth is so high that the transmission time of a packet is negligible.

To ensure near real-time communications, engineers have to be aware of the impact local access networks can have on the end-to-end delay. The round-trip time between both components change a lot according to the local access technology. The data rate of the Bluetooth shield is quite low (2 Mbps) in comparison to the data rate of the Wi-Fi card (54 Mbps). The time to transmit the data from the robot to the phone, or inversely, is proportional to the data rate. This is the principal factor to these delays. Moreover, Bluetooth and Wi-Fi systems are contention based systems. Bluetooth systems are based on a combination of frequency-hopping and CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) [16] methods to access to the medium. Wi-Fi systems are based on the CSMA/CA method to access to the medium. Whatever the system used, i.e., Wi-Fi or Bluetooth, the medium is shared between all the nodes belonging to the same system and other systems. The delay to access to a free medium or the retransmissions due to collisions increase the round-trip time significantly.

Connection between the phone and the server use Wi-Fi technology, as the range of Bluetooth devices is too short to allow connectivity inside a house. Unlike the connection phone-server, we have the choice between two wireless technologies (Wi-Fi and Bluetooth) to connect the phone to the robot. The choice between these technologies depends on energy consumption and response time. In order to reduce energy consumption, the use of a Bluetooth connection between the smartphone and the robot is interesting due to its low consumption. The mobile robot's operational time is limited before exhausting its battery power. Indeed, Bluetooth is much more power efficient than Wi-Fi. As mentioned by Pering et al. [17], the power consumption of Bluetooth is 10 times lower than Wi-Fi. In order to reduce response time, the use of a Wi-Fi connection between the smartphone and the robot is a more suitable solution for soft real-time systems that have relative short delay constraints. In general, the system must balance the conflicting goals of maximizing the response time and minimizing the energy consumption. To achieve these goals, the system could use the Wi-Fi card to send short delay-constrained commands to the robot and Bluetooth connection to send no delay-constrained commands.

Last versions of Bluetooth systems could also be used, as they include two specific modes of communication: Bluetooth High-Speed (BHS) and Bluetooth Low Energy (BLE). BHS is based on Wi-Fi protocol. BLE has a very low power consumption.They seem to be very good candidates for mobile near real-time systems. Due to the proposed modes of communication, near real-time applications can easily use a fast communication for short delay-constrained commands and a very low power consumption mode for no delay-constrained commands. Unfortunately, Bluetooth shields proposed by Arduino are not yet compliant with last versions of the Bluetooth protocol and do not include these features.

As we mentioned before and to provide a good user experience, the end-to-end delay is bounded at 300 ms. In our context, the use of a Bluetooth connection between the phone and the robot is the more suitable solution. Indeed, the end-to-end delay never exceeds the threshold and the energy consumption is reduced to its minimum.

IV. The low-cost positioning system

After an introduction to positioning, we will describe the two selected positioning systems.

A. Introduction

The purpose of our robot, as stated in Section I, is to be used either by the dependent person or by a remote carer. Displacements inside the house may be conducted while using the video signal coming from the camera of the smartphone. Nevertheless, providing a Web interface containing a map of the house and offering the ability to choose a destination, only by a click, will be an important improvement to the users.

A localization system has then to be added on the robot, or to be developed using already embedded equipment on the robot. It should be efficient and low cost, to keep the philosophy of our approach. As the GPS system can not be used indoor, as few off the shelves solutions exist, several alternative technologies have been studied: ultrasound, cameras, infrared, Radio Frequency Identification (RFID), ZigBee, or Ultra-wideband (UWB). Some of theses solutions and more are described in [19]. We do not investigate SLAM solutions [20] as our aim is not to build the environment of the robot and these solutions need lots of computations and are often based on multi-sensoring.

Ultra-wideband seems to be promising, but it is still too expensive for a low-cost system. The accuracy can be better than 10cm [21].

RFID also gives an accuracy of several centimeters. As it works with an antenna checking for either active transceivers or passive tags, it is difficult to use it. Too many passive tags should be installed inside the house. In our case, a RFID positioning system could not be used as a main positioning system. It could be used to improve another positioning system.

Positioning with Ultrasound is not recommended in houses because animals can be sensitive to it.

Thus, only three main technologies remain usable for our purpose: ZigBee, cameras or infrared.

- ZigBee: The IEEE 802.15.4 standard can be used to locate a mobile device in a house. The most used method consists of using the Received Signal Strength Indicators (RSSI). That information is provided by the network and is easy to extract.
- Cameras: An optical camera can be used as a unique sensor for positioning. If the camera is on the robot, only one is required and we can expect reduce the total cost. More, there is no need for additional infrastructure.
  The main problem is to analyse an image taken by the camera. Objects such as doors and windows can be detected and matched with a database containing a 3D description of a room [22]. The main difficulty will be the building of the 3D model of the house. A second approach consists of taking photos in the house, and let the system compare them to the image taken by the camera on the robot. Those approaches only rely on features of a room at one time. The recognition may easily produce errors as soon as something changes in the room, for example, the light, or the position of a chair. Specific markers such as patterns or barcodes can be used in a room to improve recognition [23]. A compromise must be found between a large pattern easy to recognize, and small patterns almost invisible in the house.
  Cheap cameras are available but image recognition seems to be impossible in a low-cost system.
- Infrared: Infrared light is invisible to the human eye. In a house, an infrared beacon is less intrusive than patterns such as barcodes seen above. A single IR beacon in a room may achieve room localization because IR signals do not cross walls. For meter-precision in a room, several techniques are under development. For example, cameras can be used.

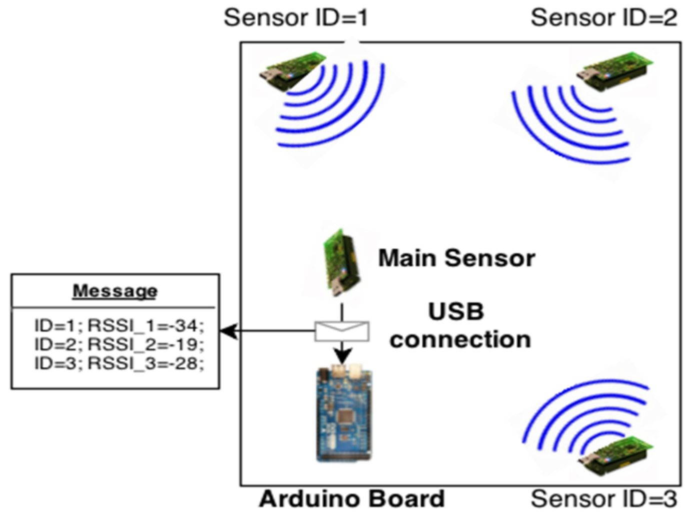

Figure 5. The positioning system.

In the following, we will focus on two positioning solutions. The first one will use ZigBee. The second one will use cheap IR cameras and IR emitters adapted to our problem. Our aim is to experiment well-known solutions in order to make the best proposal, fitting our requirements.

B. A positioning system using the Received Signal Strength Indicator (RSSI)

We designed a low-cost localization platform for 2D-positioning.

Let us assume the robot only has to monitor flat floor, i.e., the relative z-coordinate is always constant. In cases where different floors have to be monitored, a robot may be on each floor. They can communicate between them in order to extend the control in the whole habitation.

The positioning system involves 4 TelosB wireless devices. The 3 auxiliary sensors have a fixed position, being installed in strategic places of the room, in the corners for example. The places must be chosen in such way that the robot, which will have the Main Sensor attached to be in permanent Line of Sight with this sensors. This way, the communication would be done with very little interference.

Fig. 5 shows the whole system and the interaction between the components. Auxiliary sensors send messages periodically.

The main sensor do not know their position. After receiving a message from an auxiliary sensor, it gather information, such as receiver's Received Signal Strength Indicator (RSSI) and the identity of the sender. In order to optimize the energy consumption, the processing of the RSSI values is skipped in this moment, being the duty of the server application to make the necessary computations from which will result the distance approximation. Once the Main Sensor acquires a message from each of the 3 fixed sensors, it will create a data packet, which contains the 3 pairs of ID - RSSI value for each sender, and will send it through the USB interface to the Arduino board. The Arduino board forwards this message to the server that converts the raw values into physical distance, measured in meters. At this point, the server knows the distance between the main sensor and each auxiliary sensor.

In two-dimensional geometry, the trilateration technique uses three reference nodes to calculate the position of the target node. To be localized the target node should locate at the intersection of three spheres centered at each reference position. When the signal received from the reference nodes is noisy, the system is non-linear and cannot be solved. An estimation method has to be used. To get a satisfying approximation position of the mobile robot, we use the Newton-Raphson method [24]. This method attempts to find a solution in the non-linear least squares sense. The Newton-Raphson' main idea is to use multiple iterations to find a final position based on an initial guess (for example, the center of the room), that would fit into a specific margin of error.

The first results of our experiments show that RSSI values are not constant due to multipath components. Hence, the precision of our system is about 2 meters. Such a precision is sufficient to know the room where the robot is, but is insufficient to have a precise position. Our results fit with the results presented in [25]

This lack of precision leads us to propose another solution using both IR light and infrared camera.

C. The Infrared positioning system

In the previous section, we described a well known RSSI positioning system that can tell a distant user, which room the robot is closed to, but not where is the robot in a specific room. It is an interesting information, but far from perfect. More, it significantly increases the cost. A commercial sensor costs about €80. To achieve 2D-positioning, three sensors are required. Due to the range of a sensor, i.e., 10-20 meters, the cost of the positioning system is related to the size of the house. That positioning system must be considered as optional if the cost is critical.

By only using the video sent by the robot, controlling the distant robot is a difficult task. For example, if the video shows a wall, it is often impossible to say which wall it is. If the video shows a door, it is easy to recognize a door, but often impossible to say which door it is. We need a reliable system to help the distant user.

We have been working on an Ultra-wideband system (UWB) [21]. The precision can reach two centimeters. Unfortunately, UWB transceivers are very expensive (about €400 each).

Instead, we propose to hook IR LEDs at known positions on walls of a house, and to use an infrared camera laid on the robot to detect them, thus providing positioning. In the next sub-sections, we will show how to get a reliable detection of IR LEDs.

1) The IR LED: The first problem is to find an IR LED, which can be detected by using a low-cost IR camera, which produces 320x240 pixels images. A standard IR LED such as that shown in Fig. 6 is difficult to detect: when illuminating the camera from a distance of two meters, it appears on the image as a single pixel, which is almost impossible to find. To ensure that the IR bean will be visible on the image, one solution consists of using a more powerful IR LED, and concentrating the IR bean by adding a lens in front of the IR LED.

Figure 6. The standard IR LED.

We use a Mentor IR LED [26]. A 10mm lens is inserted in front of the LED, and encapsulated in a metal housing (Fig. 7). It costs about €8. The IR LED is visible in a 90 degree angle, up to several meters. If the IR LED is 2m from the camera in an angle of 30 degrees from the IR beam, we get a 5x5 pixels white rounded square on the image (Fig. 8). If the angle is close to zero, the camera is strongly illuminated and we get a 12x12 pixels white rounded square (Fig. 11).

Figure 7. The IR LED with lens.

2) Detecting an IR LED: An IR LED can be detected on the image up to 4m and within a 45 degree angle from the IR beam. The main problem is the image analysis. We must detect a white point in the image. The proposed solution uses XBee modules.

The IR LED is controlled by an Arduino connected to an XBee module. The whole IR LED module (Arduino, XBee and IR LED) is hooked somewhere.

As seen above, the control module on the robot includes and Arduino and an XBee module. Thus, communication is possible from the robot to the IR LED module. The robot is able to switch the IR LED ON and OFF. If the robot position must be obtained, the robot executes the following actions:

- Switch the IR LED ON, by using the XBee module to send the command
- Take a first photo
- Switch the IR LED OFF, by using the XBee module to send the command
- Take a second photo
- Compare the two images to detect the IR LED

The comparison of the two images gives the position of the LED on the image. We only have to find in the first image a group of white pixels that is absent in the second image. From the position of that group of white pixels (x, y), we can obtain an estimation of the direction of the robot (given by the x coordinate), and the distance from the wall (given by the y-coordinate).

This method is reliable under two conditions: the robot has been stopped, and nothing is moving in front of the webcam. If something is moving, the two images will be much more difficult to compare and the result not reliable.

From our point, that limitation is not a problem. The only working way to control a distant robot in a house is to make it move by steps of about one meter. A robot moving continuously is much more frightening for the inhabitants. After each step, the robot must be stopped for a short period of time, and the situation must be evaluated before the next step. As soon as the robot has been stopped, the control system on the robot can estimate the position, and send it to the distant user who can see the position and the direction of the robot. In fact, we do not obtain a precise position, but an area in which the robot is. That information is shown on a map on the user interface. By using the additional video sent by the robot, the distant user can easily guess an exact position of the robot. A simple click on the displayed map allows the distant user to give to the control system, both the exact position of the robot and the next target position.

If something or somebody is moving in front of the webcam, it is also not a problem. The robot is not an autonomous one, and when the distant user sees that something or somebody is moving in the house, the distant user should stop the robot, analyze the situation, and make sure a collision is not closed.

If there are several IR LEDS in the house, they must be numbered in such a way that the robot control system can switch them ON and OFF individually. Only one IR LED can be ON at a given time.

3) A positioning solution by using individual IR LEDs: Each IR LED module, including an Arduino and an XBee module, costs about 40 €when it is built by unit. Even if mass production of those modules could reduce the cost to €10 or €20, installing such devices in a whole house could be considered as too expensive.

One IR LED hooked near the door in each room, and another IR LED on the opposite wall seems to be a good compromise. At a given time, the robot control system would only have to check for only two IR LEDs. That takes about one second.

In the next sub-section, we will show a more precise positioning system, but requiring twice more LEDs.

4) A positioning solution by using sets of two IR LEDs: If two LEDs are hooked on a wall, the camera carried by the robot will detect two white spots on the image. The two white points will form a horizontal line if the camera is deflected towards the center of top of the door (Figs. 8 and 9).

If the camera is not deflected towards the door, but towards a point left or right to the door, the line passing through the white points on the image will be inclined (Fig. 10).

The robot position can be obtained from two white points on the image. The distance between the points gives the distance from the LEDs. The inclination of the line passing by the two points gives the angle of vision. In fact, we measure both the horizontal and the vertical distance between the points, We directly obtained a position from those two measures. We get the distance from the wall, and the distance from a plane orthogonal to the wall and passing through a point located between the diodes.

Figure 8. Two IR LED on top of a door.

Figure 9. The line joining the IR LED.

Figure 10. The IR LED seen from aside and the line joining the IR LEDs.

The direction of the robot is obtained by comparing the center of the image to the segment center joining the two white points. the x-axis. The (0,0,0) point is on the floor, between the two IR LEDs. Doors are often 210 cm high and about 100cm wide. The camera is at a height of 40cm. The IR LED

#1 coordinates are (50,0,210), and those of IR LED #2 are (-50,0,210).

Figure 11. The IR LED, maximum illuminating.

Both on the x-axis and y-axis, one pixel image is about 2cm in the coordinate system. Thus, we can easily achieve a 10cm precision and there is no need to have a greater precision for our purpose. However, some objects may hide the IR LEDs. In this case, several sets of two IR LEDs are required on several walls.

The use of IR camera makes it possible to locate rather precisely the mobile robot, but the cost remains a question.

V. Conclusion and Future Work

The aim of this paper was to present a mobile home robot that could be helpful for old and/or dependent person. Proposing a low cost solution, using high tech components, promoting simplicity were some of the key ideas that conducted this project. We started by comparing different "off the shelves" solutions, but in our opinion, they are difficult to maintain. We chose to have a components based approach in order to improve the maintainability of our solution. We also explain the software architecture of our distant control system. As positioning could also be helpful in some circumstances, we presented 2 different low cost solutions.

In this paper, we proposed no new technologies, neither for mobile robot platform, the remote control, nor for positioning. But, we tried to show that if the right technologies are chosen, a mobile robot platform can be built and used with the respect of good properties for AAL. Our aim was clearly not to build an autonomous robot assistant, but a robotic platform that can be helpful in several cases for the old and dependent person,the relatives and the carers.

The next step will be to propose this mobile robot platform to selected users in order to get feedback and to improve all aspects of the prototype. On the hardware part, the integration of all the component of the remote control part will be a great improvement. On the software part, the development of specific applications will make it possible to propose more usages. We also want to take advantage of the good properties of cloud computing in order to provide more reliability and more flexibility. Another important objective will be to work on the interaction between the robot and all the connected objects that may be installed at home.

References

[1] D. Espes, Y. Autret, J. Vareille, and P. Le Parc, “Designing a Low-Cost Web-Controlled Mobile Robot for Home Monitoring,” IARIA UBICOMM 2014, The Eighth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, 2014, Rome, Italy, pp. 178–183.

[2] A. Chibani, Y. Amirat, S. Mohammed, E. Matson, N. Hagita, and M. Barreto, “Ubiquitous robotics: Recent challenges and future trends,” Robotics and Autonomous Systems, Volume 61, Issue 11, November 2013, pp. 1162–1172, ISSN: 0921-8890.

[3] S. Nurmaini, “Robotics Current Issues and Trends,” Computer Engineering and Applications, Vol. 2-1, March 2013, pp. 119–122, ISSN: 2252-4274.

[4] A. Touil, J. Vareille, F. L’Herminier, and P. Le Parc, “Modeling and Analysing Ubiquitous Systems Using MDE Approach,” The Fourth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, Florence, Italy, October 2010.

[5] P. Le Parc, J. Vareille, and L. Marce, “Web remote control of machine-tools the whole world within less than one half-second,” International Symposium on Robotics, ISR 2004, Paris, France, March 2004.

[6] “WowWee Rovio, a Wi-Fi enabled mobile webcam,” 2015, URL: https://en.wikipedia.org/wiki/Rovio_(robot) [accessed: 2015-11-20].

[7] “Robots With Their Heads in the Clouds,” IEEE Spectrum, March 2011.

[8] “Service-Oriented Architecture (SOA) and Cloud Computing,” 2015, URL: http://www.service-architecture.com/articles/cloud-computing/ [accessed: 2015-11-20].

[9] Y. Chen, Z. Du, and M. Garcia-Acosta, “Fifth IEEE International Symposium on Service Oriented System Engineering,” Journal of Something, Nanjing, China, June 2010, pp. 151–158.

[10] RS. . Bucks, D.L. Ashworth, G.K. Wilcock, and K. Siegfried, “Assessment of activities of daily living in dementia: development of the Bristol Activities of Daily Living Scale,” Age and Ageing, 1996, pp. 113–120.

[11] “Introduction to the Miabots & Robot Soccer,” 2015, URL: http://eprints2.utem.edu.my/5831/1/Merlin_Miabot_Pro_Robot_Soccer_(2_Wheels)_24_Pages.pdf [accessed: 2015-11-20].

[12] “JIBO,” 2015, URL: http://www.jibo.com/ [accessed: 2015-11-20].

[13] “Romo, the programmable, telepresence robot toy for kids and adult,” 2015, URL: http://www.romotive.com/ [accessed: 2015-11-20].

[14] “The Arduino micro-controller,” 2015, URL: http://arduino.cc/ [accessed: 2015-11-20].

[15] “The WebSocket Protocol. Internet Engineering Task Force (IETF),” 2015, URL: http://tools.ietf.org/html/rfc6455 [accessed: 2015-11-20].

[16] M. Oliver and A. Escudero, “Study of different CSMA/CA IEEE 802.11-based implementations,” EUNICE, 1999, pp. 1–3.

[17] T. Pering, Y. Agarwal, R. Gupta, and C. Power, “Coolspots: Reducing the power consumption of wireless mobile devices with multiple radio interfaces,” Proc. ACM MobiSys, 2006, pp. 220–232.

[18] M. C. Yip, M. Tavakoli, and R. D. Howe, “Performance Analysis of a Manipulation Task in Time -Delayed Teleoperation,” Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, October 2010.

[19] D. Dardari, P. Closas, and P.M. Djuric, “Indoor tracking; theory, methods ans technologies,” IEEE Transaction on vehicular technology, vol. 64-4, April 2015, pp. 1263–1278.

[20] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping (SLAM): Part I The essential algorithms,” IEEE Robotics and Automation Magazine, vol. 13-2, June 2006, pp. 99–110.

[21] D. Espes, A. Daher, Y. Autret, E. Radoi, and P. Le Parc, “Ultra-wideband positioning for assistance robots for elderly,” 10th IASTED International Conference on Signal Processing, Pattern Recognition and Applications, SPPRA 2013, Feb. 2013, Austria.

[22] T.K. Kohoutek, R. Mautz, and A. Donaubauer, “Real-time Indoor Positioning Using Range Imaging Sensors,” Proceedings of SPIE Photonics Europe, Real-Time Image and Video Processing, 2010, vol. 7.

[23] A. Mulloni, D. Wgner, D.Schmalstieg, and I. Barakonyi, “Indoor Positioning and Navigation with Camera Phones,” Pervasive Computing, IEEE 2009, vol. 8, pp. 22–31.

[24] "The Newton-Raphson Method," 2015, URL: http://www.math.ubc.ca/ anstee/ math104/ newtonmethod.pdf [accessed: 2015-11-20].

[25] T. Alhmiedat, G. Samara, and A. Salem, "An Indoor Fingerprinting Localization Approach for ZigBee Wireless Sensor Networks," European Journal of Scientific Research, vol. 105-2, July 2013, pp.190-202, ISSN: 1450-216X / 1450-202X.

[26] "The Mentor M5070 IR LED," 2015, URL: http://www.mentor-bauelemente.de/en/products/product_search/?s=M%205070 [accessed: 2015-11-2].