ORIGINAL ARTICLE

Journal Section

Programmable Intelligent Spaces for Industry 4.0 : Indoor visual localization driving attocell networks

Alexandre P. do Carmo[1*] | Raquel F. Vassallo[2*] | Felippe M. de Queiroz[2*] | Rodolfo Picoreti[2*] | Mariana R. Fernandes[3] | Roberta L. Gomes[4] | Magnos Martinello[4] | Cristina K. Dominicini[4] | Rafael Guimarães[4] | Anilton S. Garcia[2] | Moises R. N. Ribeiro[2] | Dimitra Simeonidou[5]

[1]Electrical Engineering Dept, Federal Institute of Espírito Santo, Guarapari, Brazil

[2]Electric Engineering Dept, Federal University of Espírito Santo, Vitória, Brazil

[3]Electrical Engineering Dept, Federal Institute of Espírito Santo, Vitória, Brazil

[4]Informatics Dept, Federal University of Espírito Santo, Vitória, Brazil

[5]High Performance Networks Group, University of Bristol, Bristol, UK

Correspondence
Electrical Engineering Dept, Federal Institute of Espírito Santo, Guarapari, Brazil
Email: alexandre.carmo@ifes.edu.br

Present address
*High Performance Networks Group, University of Bristol, Bristol, UK

Funding information
European Commission H2020, Grant Number: 688941 (FUTEBOL); Brazilian Ministry of Science, Technology, Innovation, and Communication (MCTIC) through RNP and CTIC; CAPES; FAPES and CNPq.

Real-time and mission-critical applications for Industry 4.0 demand fast and reliable communication. Therefore, knowing devices' location is essential, but GPS is of little use indoors, while electromagnetic impairments and interferences demand new approaches to ensure reliability. The challenges include real-time feedback with E2E low latency; high data density due to large number of IoT devices per area; and smaller communication cells which increases the handover frequency and complexity. To tackle these issues we introduce a Programmable Intelligent Space (PIS) to deploy attocells, enable E2E programmability, provide a precise computer vision localization system and networking programmability based on SDN. To validate our approach, experiments were conducted controlling a mobile robot through a trajectory. We demonstrate the need for higher camera frame rate to achieve tighter precision, evaluating different trade-offs on localization, bandwidth and latency. Results have shown that PIS wireless attocell handover achieves

* Equally contributing authors.

seamlessly mobile communication delivering packets within the deadline window, with similar performance to a no handover baseline.

KEYWORDS
Attocells, Computer Vision, Location-awareness, Intelligent Spaces, E2E low latency

1 | INTRODUCTION

We live now at the dawn of a new industrial revolution where mobile Internet, wearable sensors, Internet of Things (IoT), artificial intelligence (AI), cloud-computing and autonomous robots will be combined in cyber-physical systems (CPS). The effects of this revolution on factories and other manufacturing facilities are usually referred to as Industry 4.0.

Mobile robotics is possibly one of the main driving forces behind this revolution [1]. It is envisioned that industrial processes in Industry 4.0 will massively adopt large and small mobile (and even autonomous) robots to revolutionize manufacturing and logistics by taking advantage of their higher degrees of freedom compared with their predecessors. Batteries have evolved immensely in the last decades and power cords are no longer needed. Thus, latency bounded, reliable, and ubiquitous wireless connectivity will, literally, unleash robots to move around more freely.

Cloud robotics paradigm will also bring important latency requirements [2] as it pushes processing power and feedback loops out from the physical robots into remote computing facilities elsewhere. Even with the advent of edge-computing (i.e., closer to premises than the main cloud), End To End (E2E) latency (e.g., sub-millisecond [3]) remains a fundamental constraint for ensuring stability of feedback loops where controlling algorithms are performed remotely.

Figure 1 provides a representation of the above described scenario of Industry 4.0 in a production line with an automated warehouse, mobile robots and users. There is also the call for scalable and simplified solutions for the upcoming Internet of Things (IoT) evolution. As a consequence of IoT widespread use in industry, the number of devices per square meter (and possibly the bandwidth demands) will soon skyrocket [4]. Regarding wireless communication, this fact may push current plans for deploying pico and femto cells toward attocell (i.e., sub-square-meter cells) in industrial facilities.

Therefore, in order to meet the aforementioned challenges, denser and more agile networks are expected. And context information, especially location information, can be a game-changer strategy to unleash dependable wireless real-time (and mission-critical) applications in the context of Industry 4.0. As mentioned in [5], location information has become a key factor in today's communication systems allowing specific services to be offered or executed according to the device or user location.

Innovative approaches exploiting cooperation and coordination with localization mechanisms for wireless/mobile technologies is certainly a trend to be followed by current network and robotic investigations. Approaches such the ones presented in [6] and [7] show how the usage of range error models and cooperation among devices can improve localization in wireless communication networks. Also in [8] the authors demonstrate that multisensor fusion and spatial cooperation can significantly increase localization in a large-scale scenario with hundreds of mobile agents.

For instance, precise location-awareness enables the communication system to perform beam-steering and dynamic power control, which can result on shorter symbols for lower latency communications. Moreover, predictive and preventive handover in extremely dense attocell-based coverage may be fed by robot tracking mechanisms. Last but not least, location-aware systems can help upper layer protocols as well. For example, reconfigurations, such as dynamic

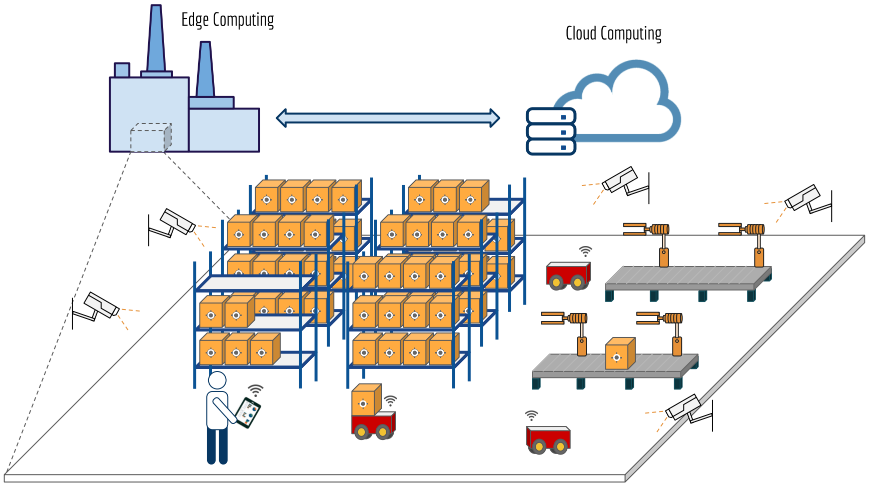


FIGURE 1 Mobile robots in Industry 4.0.

routing in the backhaul network [9], need to be implemented pro-actively in order to provide a hitless service experience to fifth generation (5G) users [10].

However, industrial environments usually hinder the use of traditional radio-frequency-based (RF-based) localization methods, as these environments are mostly full of obstacles of different materials, shapes, and sizes that may obstruct signals between emitters and receivers. Many RF-based localization methods require a non-obstructed path between a transmitter and a receiver, and the signal strength, signal delay and angle of arrival of electromagnetic wave propagation will be believably handicapped by severe levels of impairments and interferences in industrial indoor facilities [11].

A promising approach to obtain localization precision around few centimetres or less is to use computer vision methods in industrial environments [12, 11]. Commonly, these environments present many cameras which are used for monitoring and surveillance purposes. However extracting precise localization information, from images captured by a network of cameras, usually requires large bandwidth and complex algorithms to be executed in real-time. Besides mobile robots and other machineries, Industry 4.0 environments will also be crowded by mobile phones, tablets, IoT devices and other important services, requiring fast feedback, for example, to avoid accidents and delays on production.

Considering all these issues, this paper proposes the use of a Programmable Intelligent Space (PIS) to enable the cooperation between a visual localization system and the communication infrastructure of an industrial environment. More precisely, we have investigated the implementation of an indoor attocell coverage integrated to a tens of centimeters computer vision localization method as a way to meet real-time requirements of applications in the context of Industry 4.0. Edge-cloud services and Software Defined Network (SDN) have been used to reduce E2E latency. The performed experiments show that service processing and wireless communication are the most significant components of the E2E latency. We have thus employed computing resources orchestration to reduce service processing time and a programmable wireless architecture based on SDN to reduce handover delay while a robot moves in the environment.

The main contributions of the paper are summarized as follows:

- We demonstrate that the concept of Programmable Intelligent Space (PIS) works as an enabler to meet the strict requirements of robotic applications in Industry 4.0 environment.

- Then we show that attocells can be provided using computer vision localization services as a way to reduce data density per area and, as a consequence, to meet the response time and other requirements of real time robotic applications.
- Finally, we believe that the integration of the application and infrastructure layers through PIS can generate mutual benefits. The application can have its requirements met even in a high-density data environment per area, while the infrastructure assumes a new functionality that can be offered to other applications.

The remainder of the paper presents, in Section 2, the limitations on using just RF-based (wireless) indoor localization. Then, in Section 3, we describe how visual localization and PIS can be applied to implement indoor attocells based on location-awareness, on a programmable platform. Following, Section 4 shows the developed PIS, its infrastructure and architecture. Then, in Section 5, we describe how location-aware precision and attocell representation are experimentally assessed in our testbed. Finally, in Section 6, we present some conclusions and future work.

2 | WIRELESS COVERAGE AND INDOOR LOCALIZATION

Today's mobile network opeators deploy base stations with very heterogeneous coverage areas. They range from a few macrocells with several square meters (for sparsely populated rural areas) to a large number of small antennas (usually for indoor and densely populated spaces like underground stations) covering just a few square meters. The latter is known as femtocell. There is even the initiative of using visible light systems [13] and antennas in the floor [14] for creating femtocells covering just a few dm^2. However electromagnetic propagation problems in industrial facilities may require innovative solutions, given that current solutions for providing high data rates are still bound to fewer devices sharing a particular Access Point (AP).

Regarding the E2E latency chain for mobile devices, the wireless physical layer is the first link. Physical layer latency has three basic elements: encoding of symbols/transmission processing; over-the-air transmission; and receiving processing/decoding of symbols. The former and the latter are heavily (software/hardware) technology-oriented. Over-the-air transmission, however, will be defined by the chosen frame structure, from which symbol duration is an important part. For a start, current orthogonal frequency-division multiplexing (OFDM) symbol duration alone is excessively long (around 1 ms) to ensure low latency targets. One strategy is to decrease sub-carrier spacing. This will reduce OFDM symbol duration and, as a result, transmission time interval [3].

In Industry 4.0, there will be multiple devices simultaneously requiring communication services with high reliability and low latency [15]. But, unfortunately, link-control mechanisms that provide reliability in Long Term Evolution (LTE) might not be feasible at such low over-the-air latency using traditional Hybrid Automatic Repeat Request (HARQ) and Automatic Repeat Request (ARQ) [3].

Therefore reactive resource allocation methods, which integrate localization methods and wireless connectivity, have been employed by the communication system to tackle the mobility problem and low latency requirement in indoor wireless networks [16].

In the case of RF-based localization, there are methods designed from early generations of cellular mobile radio systems. Cell-identifier positioning was present in second generation (2G). In third-generation (3G), time-based positioning system using communication-relevant synchronization signals was the employed mechanism. Dedicated positioning reference signals were also employed in fourth generation (4G). This way, the achievable accuracy in 4G is around tens to a few meters, or even a little bit less [17, 18].

Additionally, as already mentioned in the previous section, there are methods that use a cooperation strategy between wireless devices to reduce the error and obtain better localization accuracy [7]. But, even though this type of approach can achieve an accuracy of less than 15 cm [8], it can still face latency problems in high-density environments. Delays and faults in communication may occur due to the high number of devices and it is likely that problems as connectivity gaps, causing handover latency, may appear in such situations.

Currently the RF-based localization and tracking methods rely on the quality of the collected measurements and the applied methodology. Most of them use the so called multilateration, that basically uses geometry to combine the range estimates from different devices. These estimates may come from different measurements such as RSS (Received Signal Strength), ToF (Time of Flight), TDoA (Time Difference of Arrival), and AoA (Angle of Arrival) [11]. Usually sub-meter ranging requires nanosecond-grade precision oscillator and accurate measurements.

Moreover, RF-based localization presents some intrinsic limitations such as low clock accuracy due to low-cost oscillators, synchronization, attenuation and stability. Time-based ranging methods, such as ToF and TDoA, rely on precise time of flight measurements, which are subject to drifts due to the presence of many metallic structures that cause direct path loss, delay and multipath signals [5]. Received signal strength (RSS) and ToF correlation with distance is severely weakened due to destructive and to constructive interference of components which generates fading effects at small-scale level [19]. And even when the method does not depend on synchronization, like AoA, it requires a direct line of sight to perform measurements [11]. As a result, the claims of 1-m precision RF-based localization system may not apply to most industrial environments.

Therefore, some location-aware operations and applications, like robotics and machinery visual tracking, find such range insufficiently precise. Precision must be then pushed to the order of less than 1-m level, while communication must happens with low E2E latency, moving towards the 5G direction [10].

3 | ATTOCELLS, VISUAL LOCALIZATION AND PROGRAMMABLE INFRASTRUCTURES

Regarding the physical layer latency, when it comes to the air interface, attocells can be applied in order to reduce multipath propagation conditions enabling the use of shorter symbols [3]. Nevertheless, in this case, more reconfigurations are needed in the backhaul and, therefore, location-awareness could be exploited as a method to proactively allocate network resources [20]. As network configuration might depend on table updates in a large and variable number of equipments [9], a better network agility can be achieved if mobility prediction is used to anticipate topology changes and perform re-routing prior to route disruptions.

In fact, location-awareness can aid in reducing overheads and latency both at physical and network layers, moving in the direction of addressing 5G challenges. From this perspective, localization methods based on computer vision and designed for mobile devices and robotics [21, 22, 23, 24] can provide a powerful location-awareness with enough precision to support attocells.

The authors of [21] discuss some of the issues when using vision-based methods for indoor localization, but also propose means of addressing these issues and implement a visual inertial odometry system for a mobile device. Their approach achieve sub-meter positioning accuracy and, from such result, the authors claim that visual inertial odometry can provide the levels of positioning accuracy needed for widespread adoption.

To help with the task of monitoring surveillance cameras, a vision-based indoor positioning for intelligent buildings is proposed in [22]. The authors apply background subtraction to separate the foreground object. After that, they track objects and use the direct linear transform to integrate camera information to a bird's-eye map and localize the moving objects.

In [23] the visual localization system uses a matrix of infrared leds on a black board to estimate the robot's pose. Such information is then used to calibrate the four cameras that belong to the intelligent space deployed in the lab. The precision achieved is sufficient to calibrate the cameras, correct the robot's odometry and control the robot during real-time applications in the environment.

Later, using the same intelligent space but with a programable infrastruture, the authors in [24] used a similar visual localization method to localize and control a robot. This time a black and white pattern was fixed on top of a mobile robot and also a person detection approach was applied to enable the robot to follow a user or avoid hitting people in the workspace. Once again the achieved precision was in the order of a few centimeters and all the experiments were executed on real-time.

Therefore, if we consider a PIS framework running in an Industry 4.0 environment, the concept of "localization as a service" can be offered by such system and used not only to provide navigation features to robots, but also help defining attocells to meet communication requirements. Thus, we argue that it can also be very important to create location-aware programmable communication infrastructures.

3.1 | Opportunities in Using Multi-Camera Localization Systems

Currently, many industrial environments have a network of cameras installed to monitor the surroundings usually due to security reasons. But this multi-camera system may be also used to localize mobile devices. Cameras are sensors with a wide field of view, capable of providing a large amount of information about objects and users in the workplace without directly interfering on the environment itself. Besides mobile robots localization, cameras in industrial environments can also be used for object and gesture recognition, as much as risk situations.

However, there are some challenges regarding the use of computer vision and image processing to localize mobile devices. One of the major challenges is dealing with adverse lighting conditions, especially in outdoors. Although, in indoor environments such as the one we are addressing in this paper, the lighting conditions are constant and do not significantly influence the data extraction algorithms.

The main drawback of using cameras as sensors is processing complex information such as video, which contains 3 dimensions (in grayscale images) or 6 dimensions (in color images). In these situations, algorithms for data extraction become also more complex and typically require robust hardware to process the amount of information fast enough. Despite this, computer vision algorithms have experienced great advances [25]. In addition, nowadays there are more powerful and cheaper hardwares and several solutions have been developed specifically for image processing [26], with the use of Graphics Processing Unit (GPU) [27] and dedicated microcontrollers [28]. Such technologies have allowed the development of new and varied applications for computer vision, which can even be embedded in robots. These solutions make the use of computer vision increasingly popular in domestic and industrial applications.

If the cameras are correctly calibrated and their relative position is known, it is possible to achieve a few centimetre-level precision when localizing any type of objects or individuals. When one of the object dimensions is given, estimating its location may be simplified, but it is still strongly dependent on the calibration quality. Nevertheless, once the calibration is done, there is no need for recalibration, unless the cameras have their pose (position and orientation) changed. Furthermore, there is no accumulative error in computer vision localization process, i.e., even though the robot moves by the workplace, the localization error is constant and does not change with displacement.

In addition, normally in multi-cameras environments, the more cameras there are with more overlapping areas, the better is the localization accuracy. It is also important to ensure that all frames are captured at the same time, i.e., the cameras must be synchronized, so the current position of moving objects can be determined with the least possible error.

Considering the large amount of data, use of computer vision also requires high bandwidth and low latency between processing nodes. But if the infrastructure allows heavy processing to be parallelized and performed in the Cloud, computer vision becomes a strategic solution for localization and control. In our view, PIS with a network of cameras and wireless antennas providing proper programmability at service and infrastructure level is a strategic enabler to implement this solution.

3.2 | Programmable Intelligent Spaces

An Intelligent Space can be described as an interactive environment equipped with a network of sensors (e.g., cameras, microphones, ultrasound), able to gather information about the surroundings, and a network of actuators, (e.g., robots, mobile devices, information screens) [29][30]. Sensors and actuators can be directly controlled by different computing services to sense and modify the environment. Besides controlling sensors/actuators, computing services can also analyze gathered information in order to support decision-making and a number of task executions.

However to get a Programmable Intelligent Space, sensors, actuators and computing services must be underpinned by a software infrastructure in charge of providing communication facilities and necessary abstractions. Services and resources from a specific device (sensor or actuator) can then be accessed and used by different entities like services, applications or even other devices.

The software infrastructure of a PIS should be conceived as a development platform. Application developers would be able to use different computing services, and many of these services could be used by different applications. At the same time, these services should be flexible enough to meet applications' specific requirements, while providing a high-level programming abstraction for the developers.

But, in some cases, programmability only on the service level might not be sufficient to properly support certain types of applications. For example, developers of bandwidth-intensive applications should be able to request the hardware infrastructure to provide the necessary bandwidth through the whole network path. Therefore, programmability on the hardware infrastructure level is also important, and Software-Defined Networking (SDN) [31] as well as Cloud orchestration [32] appear as important enablers for an effective Programmable Intelligent Space.

4 | PROTOTYPING A PROGRAMMABLE INTELLIGENT SPACE TESTBED

In order to investigate the requirements of indoor location-aware attocell networks, we designed a prototype of a PIS in the context of EU-Brazil experimental testbeds[1]. This PIS is based on computer vision, and has cameras as its main sensors. These cameras transmit images to the cloud that is responsible for processing data, determining the robot localization based on camera images, and, in turn, generating control commands back to the robot. The mobile robotic platform contains only the necessary components for wireless communication and execution of control commands.

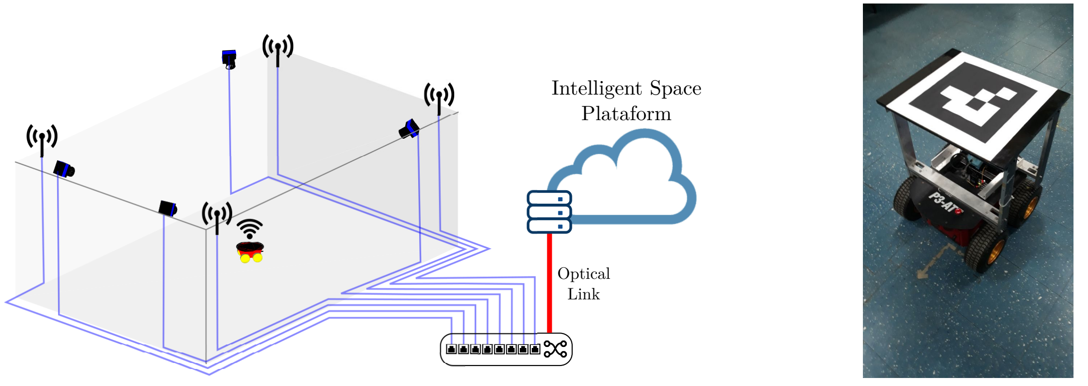
4.1 | PIS as Experimental Testbed

Our physical infrastructure, as shown in Figure 2a [24], is located at Federal University of Espírito Santo (UFES) and is composed by:

- four GigE IP cameras, FLIR BFLY-PGE-09S2C-CS, 1288x728

[1]http://www.ict-futebol.org.br/

- two Dell PowerEdge R230 (Intel Xeon E3 1240, 3.7GHz, 32GB RAM),
- one Dell PowerEdge R730 (Intel Xeon E5 2650, 2.2GHz, 128GB RAM),
- one Pica8 SDN Switch,
- four wireless Access Points,
- one mobile Robot Pioneer P3-AT.



(a) (b)

FIGURE 2 Our Programmable Intelligent Space (PIS): a) Infrastructure b) Mobile robot.

This physical infrastructure together with the software platform developed for the PIS, forms a testbed that offers programmability in different levels. Physical parameters such as the cameras' framerate FPS (Frames Per Second), image resolution or robot velocities can be modified at the device level. On the other hand, programability at the communication level is obtained through wireless and wired SDN enabled switches, as well as the message bus. The PIS framework also has a level of services. These services are autonomous and have low coupling, which facilitates their reuse by different applications and improves resource allocation. Finally, at the application level, different technologies to access the plataform, such as protobuf or Representational State Transfer (REST), are offered to the developers. This level of programmability allows the strict requirements needed in domais like computer vision and robotics to be met during application development.

As an example of such type of application in Industry 4.0 environment, we decided to command an indoor mobile robot in real-time through a visual feedback control strategy – visual information is used to retrieve robot's pose in order to control its velocity and trajectory. To facilitate its detection, the robot has a visual pattern attached to its platform (Figure 2b).

In this work, we adopted the pattern known as ArUco [33] as the marker to be attached to the robot. This type of maker can be detected very quickly and allows retrieving its pose without ambiguities.

ArUco is a binary pattern and presents a detection algorithm with self-correcting erros whenever necessary. Also each ArUco marker corresponds to an unic IDentification (ID) code. Because of the mentioned features, many markers can be detected at the same time in the same image with a fixed computational cost. The markers used in our tesbed had their dimensions registered and this information was used to recover the Three Dimensional (3D) pose of the robot, which has an ArUco marker attached to its plataform. Since the dimensions of such marker are known, the robot can be localized even if it is seen by just one of the cameras installed in the PIS.

4.2 | PIS Software Architecture

Figure 3 shows the simplified software architecture used in our PIS testbed. It is based on a service-oriented architecture (SOA) model [34], in order to provide the necessary programmability and reusabillity for our PIS infrastructure. All the communication between its components goes through a message broker (RabbitMQ[2]). Basically, from the developer's point of view, all entities are services. These services can be classified into infrastructure services and domain services. Examples of infrastructure services are Gateways, Message Bus, and Handover Orchestrator. On the other hand, Marker Localization and Robot Controller can be characterized as domain services.

All services are running on the Cloud, except the robot gateway that is running inside the robot. Gateways expose physical resources of the devices as services. For this, they must translate the form of interaction of each of these devices (cameras, robots, etc.) into a standardized model in the PIS architecture.

Services are implemented using containers as a virtualization model generating the Virtual Network Functions (VNFs). A container manager (Kubernetes[3]) is used to orchestrate the VNFs on the cloud. Kubernetes is an open source system for automating the deployment, scaling, and management of applications composed of containers. It was based on two container management systems used internally at Google: Borg and Omega, and years of experience running containers at scale at Google [35]. Kubernetes has native support for automatic orchestration based on infrastructure metrics like Central Processing Unit (CPU) and Memory utilization.

To implement the real-time robot control application, the following components of the architecture were used:

- **Camera Gateway:** there is a gateway associed with each camera. The gateway is the one in charge of establishing a connection to the camera using the camera's protocol. After that it should start receiving RAW images and converting them to a standard format (in this case JPEG), besides making them available to all other services. An example of service that receives images from the Camera Gateway is the Marker Localization. Also the Camera Gateway can receive commands to adjust parameters as colorspace, FPS and image resolution.
- **Marker Localization:** this service receives the images captured by the cameras and runs the ArUco detection algorithm. If the ID of a detected ArUco has an associated dimension, the service computes its 3D position and orientation in the workspace reference frame.
- **Robot Controller:** this service receives the robot pose from the Marker Localization and updates the information needed to the robot controller. The robot's pose is accumulated over time and then used by the controller, which applies a temporal constraint to define the linear and angular velocities that will be sent to the robot at each instant. Besides generating the control signals for the robot, this service also returns its current pose in the workspace reference frame.
- **Robot Gateway:** similar to the Camera Gateway, this service creates the connection with the robot and can be used to configure some of the robot's parameters. Besides that, this service is responsible for receiving the commands from the Robot Controller and transmitting them to the robot, but not before converting the commands to the robot's proprietary protocol.
- **Handover Orchestrator:** uses the robot's pose, returned by the Robot Controller, to orchestrate the handover among the wireless communication cells. The Handover Orchestrator receives information about the robot's localization in intelligent space as a part of its implementation. However, the Handover Orchestrator is also controlling other two fundamental blocks: i) a radio access network based on 802.11n standard using a novel software defined wireless solution and ii) a backhaul network composed of SDN switches to provide communication

[2] http://www.rabbitmq.com/

[3] https://kubernetes.io/

between intelligent space and the cloud for accessing the required applications and services [36]. Instead of using SNR information to make an handover operation, the Handover Orchestration has used the information provided by the intelligent space [37].

- **Message Bus:** enables separate applications to work together, but in a decoupled mode such that applications can be easily added or removed without affecting the others. The Message Bus routes all data between services through messages. In the PIS architecture it uses the publish/subscribe communication model with AMQP like a message protocol. Messages are published in the bus, and any service that has subscribed to that kind of message will receive it. It promotes agility and flexibility with regard to high-level protocol communication between applications in a standard way.

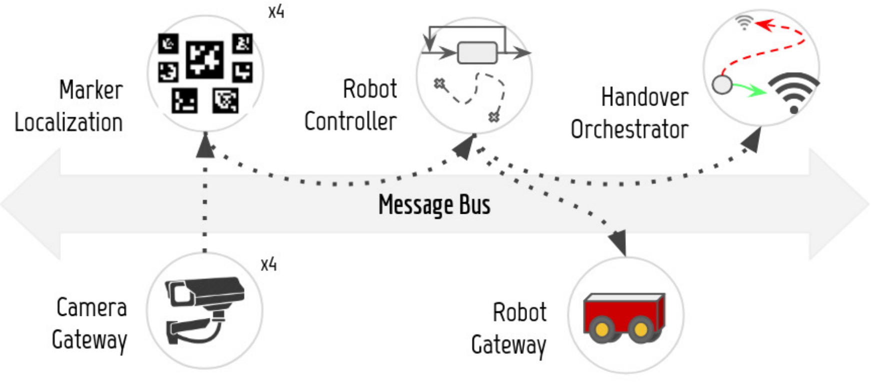


FIGURE 3 PIS Software Architecture.

Just to illustrate how these elements interact with each other, we consider a control loop for moving the robot with visual feedback. The control loop rate is associated to the FPS configured through the Camera Gateway of each camera. At every control cycle, the Marker Localization service consumes the images provided by the Camera Gateways and tries to detect the robot's marker. If the whole marker is found at least in one of the images, its 3D pose is estimated.

The robot's position and orientation is assumed to be the same as the center of the pattern. After that, this information is used by the Robot Controller service to produce a command, which is transmitted to the robot by the Robot Gateway, using wired and wireless networks in the PIS. Besides, the robot's location is also used by the Handover Orchestrator service to reprogram the wireless network whenever needed.

| PIS Wireless Architecture towards an efficient attocell handover

The objective of our attocell approach is to achieve low latency handover in wireless networks to support robot mobility. To that purpose, we have extended our architecture [36] where multiple IEEE 802.11 wireless (WiFi) clients are installed in the intelligent space and connected to the SDN-enabled infrastructure. As part of the WiFi communication module extended for the robot, a daemon hostapd is installed on it, so that a mobile AP is created.

In our PIS testbed, each zone has 1.2 m^2 of area and is covered by one access point (cell) as illustrated in Figure 4. This size is defined as an attocell, considering the goals set by Third Generation Partnership Program (3GPP) [38]. Our target is a traffic density higher than 1 Mbps/m^2 (or 1 Tbps/km^2) to provide a client data rate higher than 10 Mbps to one user per 10 m^2. Obviously adding more cells per m^2 provides more bandwidth per area. On the other hand, it requires faster handover to take advantage of the full data rate capacity 10 Mbps per zone, rather than 10 Mbps with no handover for all zones.

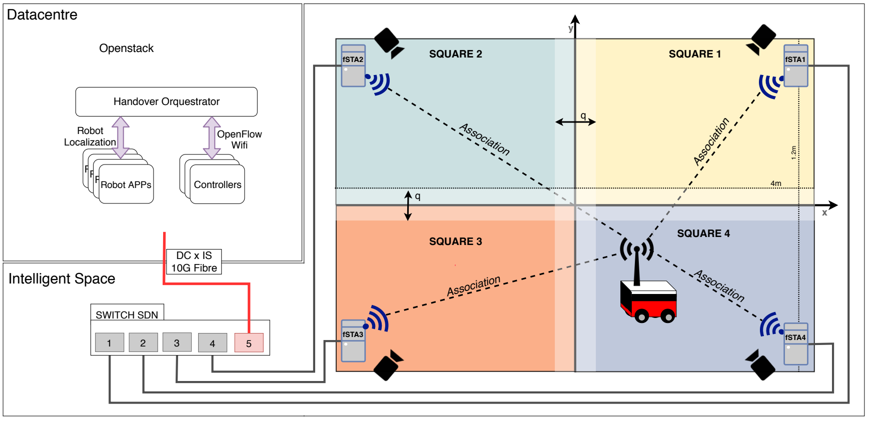


FIGURE 4 PIS WiFi Architecture with Handover.

In Figure 4, there are 4 attocells working as physical servers with Linux Operating System (OS) and Open vSwitch installed, which enables us to use Linux servers as bridges and requires being equipped with one WiFi interface and one wired interface. The WiFi Interface is used for association in the Basic Service Set Identifier (BSSID) created by the AP. In this way, all the clients in the intelligent space are associated with the mobile AP, creating a multi-connectivity scheme. Wired interfaces in the clients (optical and copper) are used for the backhaul architecture. Our solution exploits the advantage of a mobile AP with several clients associated with it. The operation of handover is performed in the backhaul by selecting the client by which to route the traffic using the OpenFlow protocol. With this single operation, the handover and the update of the routes in the backhaul are performed at the same time, without requiring additional synchronism mechanisms between them.

Initially, all attocells are associated with the robot, and based on the position of the robot, the SDN controller adds an OpenFlow rule in the switch of the backhaul to route the traffic by the corresponding attocell. This allows us to test different handover techniques, including based on the localization service from the intelligent space to set the suitable AP depending on the physical location of the robot. The rules in our Openflow switches will define the path to achieve the switch port to which the robot is connected. The scenario is shown in Figure 4, with all attocells associated and the traffic passing through $attocell_1$ after the writing of the OpenFlow rule to send traffic through SDN switch at port 5.

5 | EXPERIMENTAL EVALUATION

In order to evaluate the feasibility of our PIS approach, we have conducted two groups of experiments, structured as follows:

- In the first group, the first aspect to be investigated is the localization precision based on computer vision. We demonstrate the need for higher FPS values to achieve tighter precision. These FPS values are configurable parameters from a multi-camera network perspective at PIS testbed. However, the tighter precision the greater amount of resources to meet stringent real-time requirements so that there are different trade-offs on localization/bandwidth/latency to be evaluated.

- The second aspect that is investigated is the proposed attocell-based location-awareness. A network comunication latency assessment is performed when multiple devices (robots or others) share the same wireless infrastructure. We benchmark the latency as the number of devices increases per m^2, comparing to our SDN attocell wireless with efficient handover.

5.1 | Location-Awareness Enabled by PIS : Towards Attocell Representation

In order to evaluate the localization error in our PIS based on computer vision, we marked 18 equally spaced points on the floor to recover their position from the cameras images. Figure 5 shows an image frame captured by one of the cameras in the PIS, where 15 out of 18 points can be seen.

FIGURE 5 Image captured by one of the cameras in the PIS: with marked points on the floor.

As the points are visualized by more than one camera, all the estimated localizations are compared to the ground truth and used to calculate the mean error related to that position. The ground truth positions were directly measured considering the axis indication on the floor, which can also be seen in Figure 5. The results are presented in Figure 6a, where the circles' radius represent the localization mean errors obtained for each position and the circles' colors indicate how many cameras are able to see each particular position.

Considering the entire workspace, Figure 6b shows a cumulative distribution function (CDF) of the localization error (along with gaussian assumption curve) in our PIS prototype. Note that the localization error stays bellow 9 cm for 90% of the whole mapped space.

Normally, the minimum number of required cameras for robot's pose reconstruction is two, since each camera that observes the robot provides two equations for a three variables system (the robot's 3D coordinate). However, if the robot's workspace is flat, as it is in many industrial environments, a single camera could be enough if one of the robot's pose coordinate is known, for example, its height. Despite the possibility of localization with one camera, the mobile robot's workspace may contain occlusions and, therefore, more cameras are needed to cover the entire environment.

Analyzing this experiment, we can observe that the reconstruction error of the robot's pose depends on many

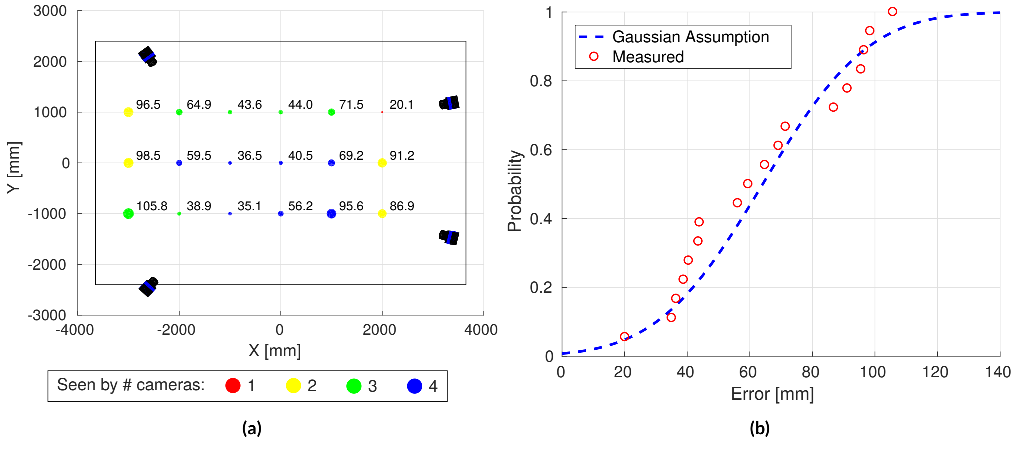


FIGURE 6 Measurements: a) Localization error for the 18 selected points on the floor b) CDF for the localization error in PIS.

variables. The first one is the quality of the cameras' calibration: the better the calibration quality, the less the reconstruction error. Nevertheless, the calibration process is complex, since each camera has 12 parameters that relate a 3D point in the environment to a Two Dimensional (2D) pixel in the image. Moreover, usually the accuracy of the localization becomes better as the number of cameras, that view the point to be reconstructed, increases. But this should be a point of attention, as cameras that are farther from the reconstructed point can also add errors and impair the position estimation. As we can observe in Figure 6a, although most points seen by more than one camera present better localization, the point viewed by only one camera was the one that obtained the least reconstruction error. The reason for that is because elements such image resolution, detection algorithm, quality and lens characteristics also directly influence the quality of the reconstruction. This dependency on many variables makes a mathematical modeling of the localization error unfeasible.

Although those first results are static measurements from the cameras, they demonstrate that our PIS prototype can easily support attocells packing representation with bonder uncertainty below 10 cm. In addition, the robot's location can be pinpointed with similar accuracy. Therefore, we can build a location-aware strategy for detecting migration across attocells using the visual localization service offered by the PIS, rather than using conventional (and imprecise) signal strength or other RF-based methods.

| Achieving trajectory precision by increasing FPS

The application used for the experiment consists in controlling a mobile robot through a predefined trajectory. The robot velocities are defined according to the desired time per lap to perform the whole path. In this case, at each control loop, the robot must be at a certain position with certain linear and angular velocities to guarantee that the path is executed in the desired time. The shape chosen for the trajectory path is the Lemunscata of Bernoulli. Both the path and the expected linear velocity for a 25 seconds/lap performance are illustrated in Figure 7.

This trajectory was chosen because of its variation on speed and path complexity. We can consider the trajectory divided into two parts – simple trajectory (straight line with constant velocity, or "easy part") and complex trajectory (curved path with positive and negative accelerations, or "hard part"). The difference between the desired trajectory

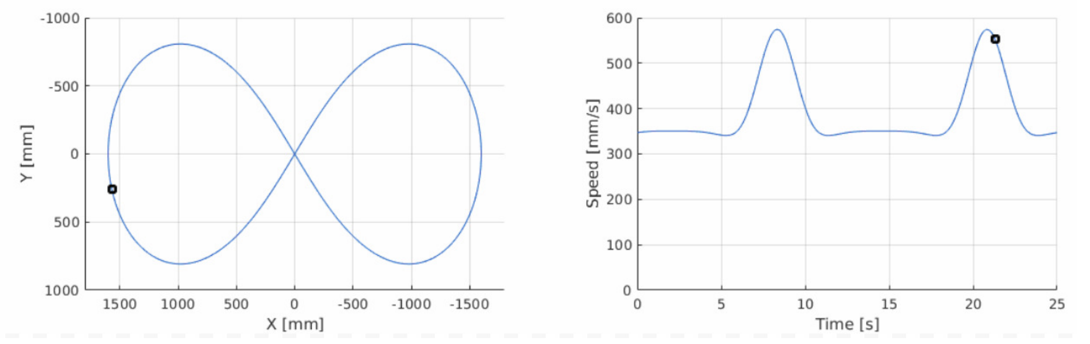


FIGURE 7 Robot's trajectory and velocity.

and the one performed by the robot is defined as the trajectory error. This error is expected to be directly associated to the path complexity and to the robot velocity.

To understand the impact of the FPS in the trajectory error, considering a fixed time per lap of 25 seconds, we performed two experiments. For the first one, the camera frame rate was set to 2.5 FPS, which means that the deadline window for commands to arrive at the robot, i.e., the E2E latency requirement, is 400 ms = $(1/2.5)$. Then, for the second case, the camera frame rate was increased to 10 FPS, narrowing the deadline window to 100 ms = $(1/10)$.

As it can be seen in Figure 8a, for 2.5 FPS, the robot struggles to follow the desired trajectory, not performing correctly the expected velocities to cover the whole path in 25 seconds. The mean error for that trajectory is around 100 mm, reaching 330 mm for the hardest part (Figure 8b).

However, when the frame rate is changed to 10 FPS, the robot executes the trajectory smoothly. The mean error is lower than 80 mm, even for the hardest part of the trajectory. Both trajectory and error are shown in Figures 8c and 8d.

Clearly the higher the FPS, the smaller the trajectory error [4]. Increasing the FPS allowed an improvement on precision regarding both the hard and the easy part of the trajectory, reducing the mean errors from 330 to 80mm, and from 100 mm to 15 mm, respectively. It is worth mentioning that the improvement range is associated to the adopted time per lap, 25 seconds/lap in this case. Also, with a higher FPS, a finer sampling is applied and thus a better visual feedback is achieved.

Thus, to meet a maximum trajectory error requirement, the application would need to change the FPS. Nevertheless, the challenge herein is that increasing the FPS (2.5 to 10 FPS) implies in a proportional decrease for the required E2E latency (400 ms to 100 ms), as the inter-frame period dictates deadlines for the visual feedback control. This is the topic addressed in the following section.

| How to meet the control loop deadline?

One of the challenges to be faced by our PIS is that there are many elements that may impose bottlenecks in terms of throughput and latency to the task of controlling a robot. Thus, network classic performance indicators are also important metrics to be assessed alongside with the localization error discussed previously.

For this application, as shown in Figure 9, the camera inter frame period defines a deadline window for the whole control loop. Basically, all the steps should happen within this interval, e.g. $t_{k+1} - t_k = 100$ ms, defined by the frame rate of 10 FPS. In other words, after an image frame is captured, it must be sent to the cloud platform, localization and control

[4] A video of one of the experiments can be found at the following URL: https://youtu.be/5r6p0ucESTw.

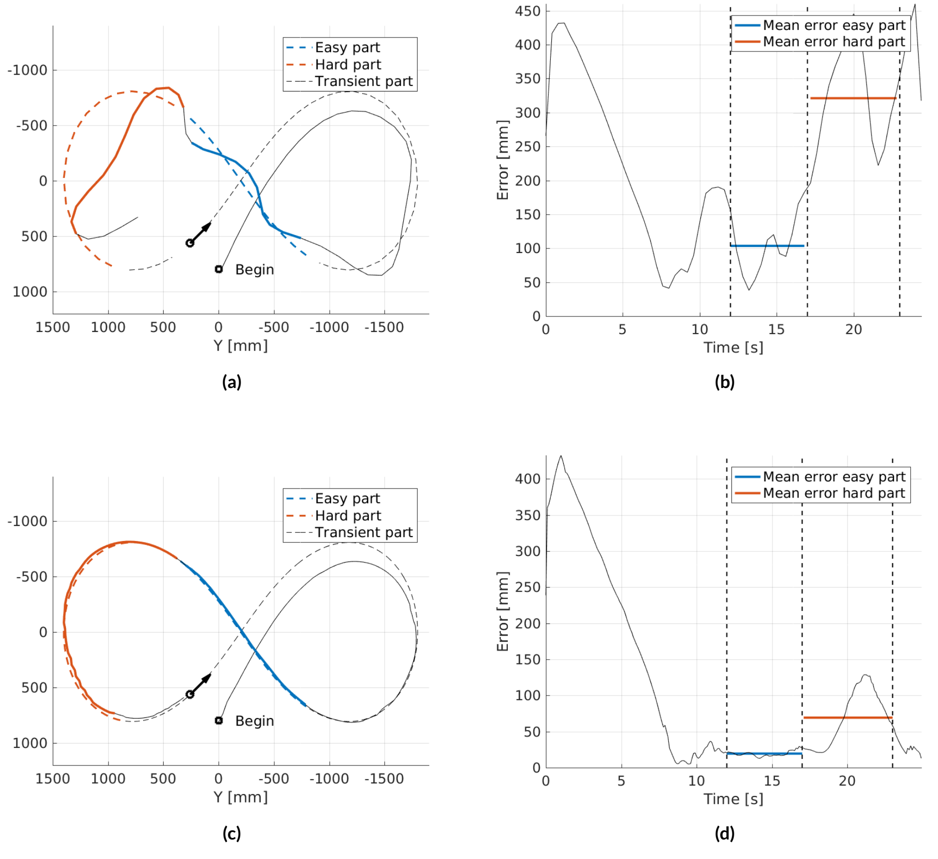


FIGURE 8 Trajectory performed and its error in experiments with 2.5 FPS (a,b) and 10 FPS (c,d).

services must be executed, and the control signal should arrive at the robot before the next image frame is captured. Otherwise, if the control command does not reach the robot on time, the outdated information is no longer accurate, causing the robot to deviate from the planned trajectory, which might lead to overall robot instability or collision in extreme cases.

Figure 9 illustrates the deadline window wherein the robot's control loop must happen. The commands resulting from the control loop reaches the robot depending on: i) T_p, the processing time; plus ii) T_c, the communication time. The values of these time intervals correspond to the mean of the measurements obtained during the experiments presented at the next section. The interval T_p includes the time needed to perform image acquisition, image processing to get the robot pose estimation, and the generation of the control commands. In turn, T_c represents the latency to transmit the control signal to the robot. Note that T_c includes the time to cross the multiple networks (DC network, wired and wireless network), but wireless is the most dominant component of the latency.

As a way to reach the lowest possible value for T_p in our testbed, we performed the orchestration of the services that compose the application in a similar way to [39]. In this case, the orchestrator observes the CPU utilization of the service with higher time consuming in the control loop and performs a horizontal scaling of such service. The

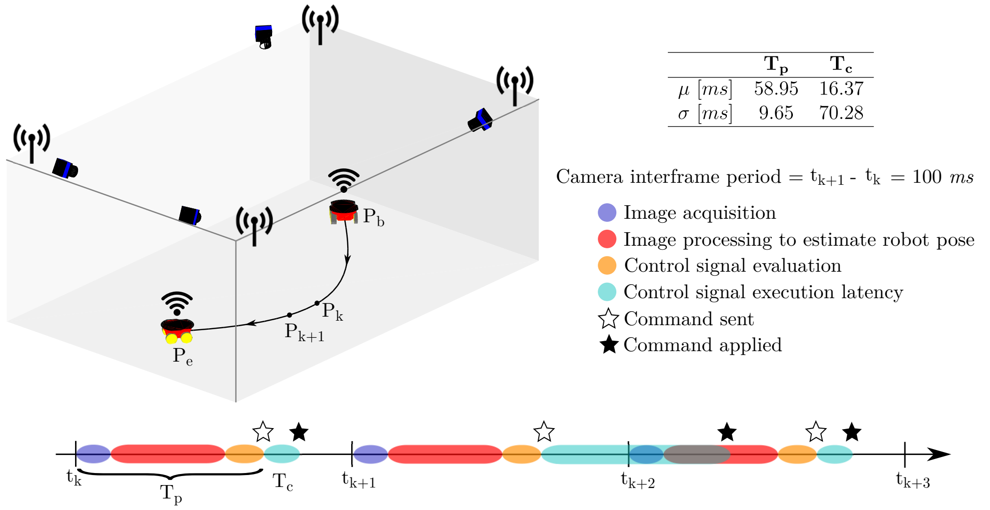


FIGURE 9 Illustration of the latency measurements during the robot's control loop.

horizontal scaling process consists of monitoring the CPU utilization as a metric for the image processing and robot's pose estimation service. When this metric exceeds a threshold, the orchestrator starts new instances of this service until the maximum allowed by the application. In this way, the processing load per instance decreases and consequently T_p decreases.

Another action performed by the orchestrator is to pin virtual CPUs that run the image processing service to guarantee CPU processing time so that T_p doesn't change very much (σ_{T_p} = 9.65ms). Regarding the impact of T_p and T_c on meeting the deadline window requirement, despite the average communication time T_c = 16.37ms being substantially shorter compared to the processing time T_p = 58.95ms, the communication latency T_c may vary significantly (σ_{T_c} = 70.28ms), mainly depending on how many devices share the same wireless network.

In an ideal case, both T_p and T_c will fit within the deadline window, as illustrated in the first time interval $[t_k, t_{k+1}]$. But if T_c increases too much, as in the time interval $[t_{k+1}, t_{k+2}]$, the sent commands will not be applied on time, i.e., before a new frame is received. In this case, the robot will keep the last control command much longer than one time interval, making the robot deviate from the desired trajectory and, thus, increase the overall error. If these missed deadlines happen rarely there will be no impact in the trajectory execution. But if that happens frequently, they will potentially impair the ability to properly control the robot.

Thus, it is clear that low latency network specifications must be pushed forward, since there is a need for higher camera frame rates to deal with faster mobile robots. More importantly, as the number of robots in the PIS raises, the wireless network will be shared by the robots and any other wireless devices. Given that the current wireless network is composed by only one WiFi access point, working as a big cell covering the entire PIS area, such T_c variability would be crucially affected. This is the topic covered by our wireless attocell approach in the next section.

5.2 | Packet delivery by efficient attocell handover

In the context of industrial indoor environments, different mobile devices, such as industrial robots, will be in motion throughout the production space. Therefore, they cannot always be served with quality by the same base station or access point due to coverage area restrictions or competition depending on the number of clients per area. In our testbed, we deploy four small cells (named attocell) to cover the PIS, an useful area of approximately 4.8 m^2 in which the robot can perform its navigational tasks.

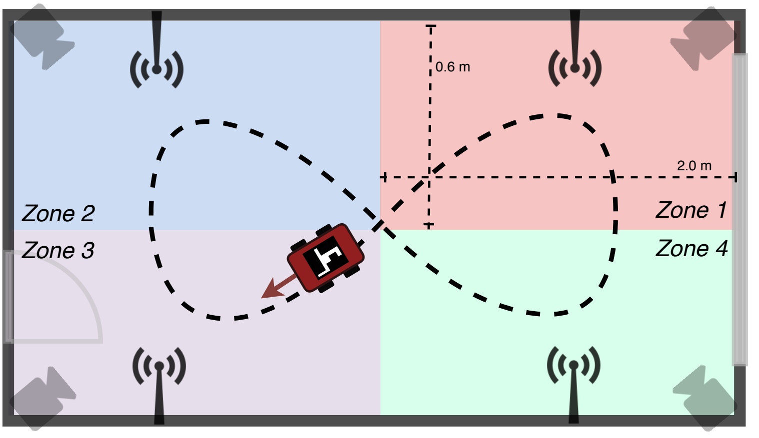


FIGURE 10 Cells coverage area in Programmable Intelligent Space handover experiment.

The purpose of this experiment is to show the effect of sharing a wifi cell in the robot control. In addition, an analysis of system performance during the handover process will be shown, observing the latency in the communication time T_c, i.e., between the cloud and the robot. As shown in Figure 10, the robot moves through different zones during its trajectory, each of them served by a respective $attocell_i$ serving $zone_i$. During the transit of the robot from one area to another, the orchestrator uses the location service as a trigger of the handover process every time the robot enters a new zone.

The experiments are conducted in the WiFi operating at 5 GHz band. We aim to evaluate the communication throughput (packet delivery) of our Attocell Handover underpinned by the localization information provided by the computer vision service. For all experimentation scenarios, the FPS is set to 10 (deadline window = 100 ms) and the robot's velocity is set to 25 sec/lap. To test our handover, we run four different scenarios. The first scenario **a** is a baseline with no handover and only one WiFi cell covering all the PIS area. Then for scenario **b** and **c**, we vary the number of clients competing for the WiFi cell with no handover, measuring the communication time T_c. Finally scenario **d** deals with our attocell handover approach.

Figure 11 shows the communication time histogram for all scenarios. Scenario **a** (Figure11a) is a baseline since the robot is the only client connected to just one WiFi cell covering the entire area and there is no sharing of the WiFi network. Indeed this is a improbable case because the wireless is usually shared by many devices, mainly in the context of Industry 4.0. Eventhough, note that around 90% of the packets are delivered within a few milliseconds.

The latency measurements shown in Figures 11b and 11c represent more realistic scenarios in which there is a workload of 5 clients generating concurrent traffic at 10 Mbps each in Figure11b, and 10 clients in Figure11c. As can be seen, for both cases the latency has risen, but it has drastically increased in the case of 10 clients, being closer to 100 ms for 40% of the samples. This result has a direct impact on how often the system misses its deadline (15.9% according to Table 1) and crucially on the trajectory error (see Figure 12) that may be 5 times worse than other cases.

The handover result can be observed in Figure11d, that shows the latency histogram for the handover case.

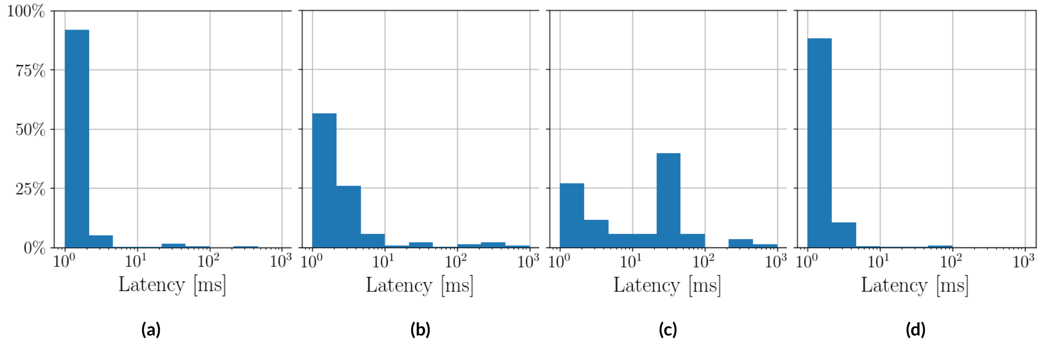


FIGURE 11 Communication time (T_c): (a) 0 clients, (b) 5 clients, (c) 10 clients and (d) PIS handover

TABLE 1 Percentage of missed deadline window for different number of clients sharing the wireless network.

Scenario	0 client (**a**)	5 clients (**c**)	10 clients (**c**)	PIS handover (**d**)
Exceeded count (%)	0.81	4.43	15.97	1.38

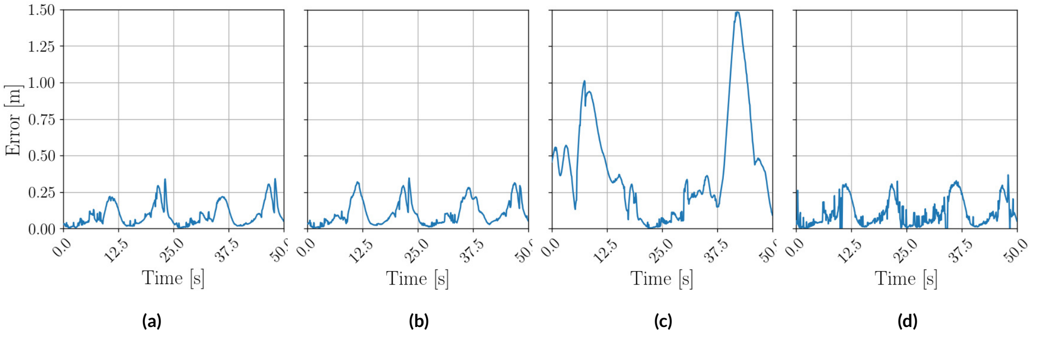


FIGURE 12 Trajectory error: (a) 0 clients, (b) 5 clients, (c) 10 clients and (d) PIS handover

Comparing to the previous cases, we can see that the obtained latency is very close to the baseline even though there are 10 concurrent clients generating traffic. This is explained by the integration between computer vision and SDN controller that drives our PIS handover orchestrator [37]. When the robot moves from one zone to another, the robot's pose is computed by the localization service so the orchestrator is able to change the attocell that serves the robot, in order to ensure the communication requirements at that zone. Despite all the concurrent traffic being transmitted at the same wireless network, they are served by the other attocells out of the current zone. Therefore, the PIS handover approach has achieved seamlessly mobile communication delivering packets within the deadline window, with missed deadlines and trajectory error similar to the baseline (see Figure 12d).

6 | CONCLUSIONS AND FUTURE DIRECTIONS

The upcoming applications for Industry 4.0 will bring the need for fast communication with mobile devices to be used in the industrial environment. Such requirement imposes E2E low latency and the usage of small wireless cells, as a way to reduce the data density per area and achieve real-time execution. Although smaller cells may cause an increase of the handover frequency, and thus an increase of the communication latency, if a precise localization method is applied for tracking the mobile devices, latency constrains can be met and real-time applications can be successfully deployed.

In this context, this paper proposes the use of a PIS as a way of providing attocells based on computer vision that allow the deployment of applications with real-time feedback. This type of applications has strict requirements for its full operation and a precise localization method can help meeting such requirements. The PIS used as our testbed provides a computer vision localization service, based on a multi-camera network, that allows the use of attocells and also a programmable wireless architecture.

To validate our approach, we deployed a mobile robot application which aim is to control the robot trajectory considering a fixed time of 25 seconds per lap. To perform this task a real-time control loop based on a visual feedback is applied. This application allowed us address many aspects involved such as localization, latency and bandwidth.

Usually, a vision-based system requires a large amount of infrastructure resources and directly impacts E2E latency. In the addressed application, to control the robot within an acceptable trajectory error, we have shown through preliminary experiments that the control command must arrive within a deadline window, defined by the cameras FPS.

From the first experiments, one can observe that it is possible to control the robot using WiFi without sharing the channel with other devices. However, this is an improbable scenario, especially in the context of Industry 4.0, where the increase in the use of wireless devices is precisely sought. In the following tests, by increasing the number of simultaneous connections using the same channel of the robot, the communication time T_c also increases until the control of the robot becomes unfeasible.

Thus, an alternative is to decrease the cell size, creating the so called attocells. Their areas are reduced until usually only one device is covered by each cell, and that would ultimately prevent channel sharing with other devices or at least not many of them. The problem with this approach is that the use of attocells would increase the need for handover between cells. The default handover time is extremely high (4s approximately) [40] which would also prevent the control of the robot.

Through the last experiments, we showed that, if a PIS with a visual localization system is used, it is possible to obtain a viable scenario where T_c can be similar to the improbable case of using a single cell without sharing. Using such PIS allows the creation of attocells and also a programmable wireless architecture to provide faster handover.

Therefore, with this paper, we believe we demonstrated the viability of using PIS based o computer vision to meet location-awareness, one of the requirements needed for 5G low latency indoor applications, in the context of Industry 4.0. The wireless latency can be addressed within the PIS by: changing physical layer parameters to reduce transmission time and reducing the handover time based on location-awareness. Handover typically refers to the task of connecting the mobile object to a neighbor attocell to maintain seamless communication. By knowing the robot localization, a SDN controller can set the wireless channels pro-actively to keep the connectivity whenever the robot leaves its attocell. This avoids dependence on reactive triggers based on signal strength.

As a future work, we plan to use attocells with dynamic areas that can assume variable formats and sizes orchestrated by FUTEBOL Control Framework [41]. We also intent to evolve our PIS prototype to integrate programmable network and cloud infrastructures tailored with a platform for computer vision [42]. Programmability includes parameters of physical devices such as cameras and robots (i.e. cameras fps, robot velocities, etc.), wireless and wired SDN enabled switches and Virtual Network Functions (VNFs) at the edge-cloud [43] supporting automatic scaling and service function chaining [44].

ACKNOWLEDGEMENTS

The research leading to these results received funding from the European Commission H2020 program under grant agreement no. 688941 (FUTEBOL), as well from the Brazilian Ministry of Science, Technology, Innovation, and Communication (MCTIC) through RNP and CTIC. This work was also partially supported by CAPES, FAPES, and CNPq.

REFERENCES

[1] Lu Y. Industry 4.0: A survey on technologies, applications and open research issues. Journal of Industrial Information Integration 2017;6:1–10.

[2] Kehoe B, Patil S, Abbeel P, Goldberg K. A survey of research on cloud robotics and automation. IEEE Transactions on automation science and engineering 2015;12(2):398–409.

[3] et al IP. A Survey on Low Latency Towards 5G: RAN, Core Network and Caching Solutions. Submitted in IEEE Communications Surveys and Tutorials;.

[4] Wollschlaeger M, Sauter T, Jasperneite J. The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0. IEEE Industrial Electronics Magazine 2017;11(1):17–27.

[5] Yassin A, Nasser Y, Awad M, Al-Dubai A, Liu R, Yuen C, et al. Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications. IEEE Communications Surveys & Tutorials 2017;19(2):1327–1346. http://ieeexplore.ieee.org/document/7762095/.

[6] Conti A, Member S, Guerra M, Dardari D, Member S, Member S, et al. Network Experimentation for Cooperative Localization 2012;30(2):467–475.

[7] Win MZ, Gifford WM, Thomas IBM. Network Localization and Navigation via Cooperation 2011;(May):56–62.

[8] Win MZ, Meyer F, Liu Z, Dai W, Bartoletti S, Conti A. Efficient Multisensor Localization for the Internet of Things: Exploring a New Class of Scalable Localization Algorithms. IEEE Signal Processing Magazine 2018;35(5):153–167.

[9] Liberato A, Ribeiro MRN, Martinello M, Marquez-Barja JM, Kaminski N, DaSilva LA. Dynamic Backhauling Within Converged Networks. In: Proceedings of the 2016 Workshop on Fostering Latin-American Research in Data Communication Networks LANCOMM '16, New York, NY, USA: ACM; 2016. p. 31–33. http://doi.acm.org/10.1145/2940116.2940122.

[10] Taranto RD, Muppirisetty S, Raulefs R, Slock D, Svensson T, Wymeersch H. Location-Aware Communications for 5G Networks: How location information can improve scalability, latency, and robustness of 5G. IEEE Signal Processing Magazine 2014 Nov;31(6):102–112.

[11] Brena RF, García-Vázquez JP, Galván-Tejada CE, Muñoz-Rodriguez D, Vargas-Rosales C, Fangmeyer J. Evolution of Indoor Positioning Technologies: A Survey. Journal of Sensors 2017;2017.

[12] Posada J, Toro C, Barandiaran I, Oyarzun D, Stricker D, de Amicis R, et al. Visual computing as a key enabling technology for industrie 4.0 and industrial internet. IEEE computer graphics and applications 2015;35(2):26–40.

[13] Dobroslav Tsonev HH Stefan Videv, Light fidelity (Li-Fi): towards all-optical networking; 2014. http://dx.doi.org/10.1117/12.2044649.

[14] Thielens A, Vermeeren G, Caytan O, Torfs G, Demeester P, Bauwelinck J, et al. Radiofrequency exposure near an attocell as part of an ultra-high density access network. Bioelectromagnetics 2017;38(4):295–306. http://dx.doi.org/10.1002/bem.22045.

[15] Tarneberg W, Karaca M, Robertsson A, Tufvesson F, Kihl M. Utilizing Massive MIMO for the Tactile Internet: Advantages and Trade-Offs. In: 2017 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops); 2017. p. 1–6.

[16] Muppirisetty LS, Yiu S, Wymeersch H. LAPRA: Location-Aware Proactive Resource Allocation. In: 2016 IEEE Global Communications Conference (GLOBECOM); 2016. p. 1–6.

[17] Xu S, Chou W. An Improved Indoor Localization Method for Mobile Robot Based on WiFi Fingerprint and AMCL. 2017 10th International Symposium on Computational Intelligence and Design (ISCID) 2017;p. 324–329. http://ieeexplore.ieee.org/document/8275781/.

[18] Zhou M, Tang Y, Tian Z, Xie L, Nie W. Robust Neighborhood Graphing for Semi-supervised Indoor Localization with Light-loaded Location Fingerprinting. IEEE Internet of Things Journal 2017;4662(c):1–1. http://ieeexplore.ieee.org/document/8115100/.

[19] Dardari D, Closas P, Djurić PM. Indoor Tracking: Theory, Methods, and Technologies. IEEE Transactions on Vehicular Technology 2015 April;64(4):1263–1278.

[20] Lee J, Yoo Y. Handover cell selection using user mobility information in a 5G SDN-based network. In: 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN); 2017. p. 697–702.

[21] Clark R, Trigoni N, Markham A. Robust Vision-based Indoor Localization. In: Proceedings of the 14th International Conference on Information Processing in Sensor Networks IPSN '15, New York, NY, USA: ACM; 2015. p. 378–379. http://doi.acm.org/10.1145/2737095.2742929.

[22] Tsai TH, Chang CH, Chen SW. Vision based indoor positioning for intelligent buildings. In: 2016 2nd International Conference on Intelligent Green Building and Smart Grid (IGBSG); 2016. p. 1–4.

[23] Rampinelli M, Covre VB, de Queiroz FM, Vassallo RF, Bastos-Filho TF, Mazo M. An Intelligent Space for Mobile Robot Localization Using a Multi-Camera System. Sensors 2014;14(8):15039–15064. http://www.mdpi.com/1424-8220/14/8/15039.

[24] Almonfrey D, do Carmo AP, de Queiroz FM, Picoreti R, Vassallo RF, Salles EOT. A flexible human detection service suitable for Intelligent Spaces based on a multi-camera network. International Journal of Distributed Sensor Networks 2018 mar;14(3):155014771876355.

[25] Schlessman J, Wolf M. Tailoring design for embedded computer vision applications. Computer 2015 May;48(5):58–62.

[26] Jaimes A. Computer Vision Startups Tackle AI. IEEE MultiMedia 2016 Oct;23(4):94–96.

[27] Nieto A, Vilariño DL, Brea VM. PRECISION: A Reconfigurable SIMD/MIMD Coprocessor for Computer Vision Systems-on-Chip. IEEE Transactions on Computers 2016 Aug;65(8):2548–2561.

[28] Fung J, Mann S. Using graphics devices in reverse: GPU-based Image Processing and Computer Vision. In: 2008 IEEE International Conference on Multimedia and Expo; 2008. p. 9–12.

[29] Wright S, Steventon A. Intelligent Spaces — The Vision, the Opportunities and the Barriers. BT Technology Journal 2004 Jul;22(3):15–26. https://doi.org/10.1023/B:BTTJ.0000047116.13540.e0.

[30] Lee JH, Hashimoto H. Intelligent space—concept and contents. Advanced Robotics 2002;16(3):265–280.

[31] Trois C, Del Fabro MD, de Bona LC, Martinello M. A survey on SDN programming languages: toward a taxonomy. IEEE Communications Surveys & Tutorials;18(4):2687–2712.

[32] Dua R, Raja AR, Kakadia D. Virtualization vs containerization to support paas. In: Cloud Engineering (IC2E), 2014 IEEE International Conference on IEEE; 2014. p. 610–614.

[33] Garrido-Jurado S, noz Salinas RM, Madrid-Cuevas FJ, Marín-Jiménez MJ. Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognition 2014;47(6):2280 – 2292. http://www.sciencedirect.com/science/article/pii/S0031320314000235.

[34] Laskey KB, Laskey K. Service oriented architecture. Wiley Interdisciplinary Reviews: Computational Statistics 2009;1(1):101–105.

[35] Burns B, Oppenheimer D. Design Patterns for Container-based Distributed Systems. In: HotCloud; 2016. .

[36] Martinez VMG, Guimaraes R, Mello R, Hasse P, Ribeiro MRN, Martinello M, et al. Ultra Reliable Communication for Robot Mobility enabled by SDN Splitting of WiFi Functions. In: IEEE ISCC; 2018. .

[37] Guimaraes R, et al. Make Before Degrade in Context-Aware Networks: URLLC on SDN-WiFi for Real-Time Industry 4.0. In: IEEE Journal on Selected Areas in Communications; 2018 (submitted). .

[38] Service requirements for next generation new services and markets. 3GPP; 2018, rev. 16.3.0.

[39] Picoreti R, Carmo AP, Queiroz FMD, Garcia AS. Multilevel Observability in Cloud Orchestration. In: 16th Int. Conf. on Pervasive Intelligence & Comp.; 2018. p. 776–784.

[40] Zehl S, Zubow A, Wolisz A. BIGAP – A Seamless Handover Scheme for High Performance Enterprise IEEE 802.11 Networks. In: 15th IEEE/IFIP Network Operations and Management Symposium No. Noms; 2016. p. 1015–1016.

[41] de A Ceravolo I, Cardoso DG, Dominicini CK, Hasse P, da S Villaca R, Ribeiro MRN, et al. O2CMF: Experiment-as-a-Service for Agile Fed4Fire Deployment of Programmable NFV. In: 2018 Optical Fiber Communications Conference and Exposition (OFC); 2018. p. 1–3. http://www.ict-futebol.org.br/wp-content/uploads/O2CMF-Experiment-as-a-Service-for-Agile-Fed4Fire-Deployment-of-Programmable-NFV.pdf.

[42] Gomes RL, Martinello M, Dominicini CK, Hasse P, Villaca R, Vassallo RF, et al. How can emerging applications benefit from EaaS in open programmable infrastructures? In: 2017 IEEE First Summer School on Smart Cities (S3C); 2017. p. 73–78. http://www.ict-futebol.org.br/wp-content/uploads/2017/08/How-can-emerging-applications-benefit-from-EaaS-in-open-programmable-infrastructures.pdf.

[43] Dominicini C, Vassoler G, F Meneses L, S Villaca R, Ribeiro MRN, Martinello M. VirtPhy: Fully Programmable NFV Orchestration Architecture for Edge Data Centers over a Fully Programmable SDN Infrastructure for Small Data Centers 2017 09;PP:1–1.

[44] Castanho MS, Dominicini CK, Villacça RS, Martinello M, Ribeiro RNM. PhantomSFC: A Fully Virtualized and Agnostic Service Function Chaining Architecture. In: 2018 IEEE Symposium on Computers and Communications (ISCC); 2018. p. 354–359.