

## Article

# An Introduction to Patterns for the Internet of Robotic Things in the Ambient Assisted Living Scenario

Bruno Andò <sup>1</sup>, Luciano Cantelli <sup>1</sup>, Vincenzo Catania <sup>1</sup>, Ruben Crispino <sup>1</sup>, Dario Calogero Guastella <sup>1,\*</sup>,  
Salvatore Monteleone <sup>2</sup> and Giovanni Muscato <sup>1</sup>

<sup>1</sup> Dipartimento di Ingegneria Elettrica, Elettronica e Informatica, Università degli Studi di Catania, 95125 Catania, Italy; bruno.ando@unict.it (B.A.); luciano.cantelli@dieei.unict.it (L.C.); vincenzo.catania@unict.it (V.C.); ruben.crispino@unict.it (R.C.); gmuscato@dieei.unict.it (G.M.)

<sup>2</sup> Faculty of Engineering and Architecture, University of Enna “Kore”, 94100 Enna, Italy; salvatore.monteleone@unikore.it

\* Correspondence: dario.guastella@dieei.unict.it

**Abstract:** The Internet of Things paradigm envisions the interoperation among objects, people, and their surrounding environment. In the last decade, the spread of IoT-based solutions has been supported in various domains and scenarios by academia, industry, and standards-setting organizations. The wide variety of applications and the need for a higher level of autonomy and interaction with the environment have recently led to the rise of the Internet of Robotic Things (IoRT), where smart objects become autonomous robotic systems. As mentioned in the recent literature, many of the proposed solutions in the IoT field have to tackle similar challenges regarding the management of resources, interoperation among objects, and interaction with users and the environment. Given that, the concept of the IoRT pattern has recently been introduced. In software engineering, a pattern is defined as a general solution that can be applied to a class of common problems. It is a template suggesting a solution for the same problem occurring in different contexts. Similarly, an IoT pattern provides a guide to design an IoT solution with the difference that the software is not the only element involved. Starting from this idea, we propose the novel concept of the IoRT pattern. To the authors’ knowledge, this is the first attempt at pattern authoring in the Internet of Robotic Things context. We focus on pattern identification by abstracting examples also in the Ambient Assisted Living (AAL) scenario. A case study providing an implementation of the proposed patterns in the AAL context is also presented and discussed.

**Keywords:** IoRT patterns; obstacle avoidance; indoor localization; inertial monitoring; smart walkers; assistive robotics



**Citation:** Andò, B.; Cantelli, L.; Catania, V.; Crispino, R.; Guastella, D.C.; Monteleone, S.; Muscato, G. An Introduction to Patterns for the Internet of Robotic Things in the Ambient Assisted Living Scenario. *Robotics* **2021**, *10*, 56. <https://doi.org/10.3390/robotics10020056>

Academic Editor: Aman Behal

Received: 16 February 2021

Accepted: 1 April 2021

Published: 4 April 2021

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Over the last decade, advances in technology and manufacturing processes have enabled the miniaturization of components to the point that, as noted by Biderman in an interview in [1], “Computers are becoming so small that they are vanishing into things”. This miniaturization, having an impact on the availability of communication-enabled devices, has allowed the spread of the Internet of Things (IoT) paradigm. The term Internet of Things was introduced in 1999 by Kevin Ashton [2] while proposing the use of RFID technology in supply chain management at Procter & Gamble. From that moment on, an increasingly rapid spread of IoT solutions has occurred. At first, the effort was focused on tracking objects, then on making them interconnected, and, further, on guaranteeing their interoperability. In fact, one of the main problems encountered in the application of the IoT paradigm concerns interoperability. If in an initial phase, the lack of standards designed for the IoT had caused the proliferation of a myriad of applications based on proprietary technologies and therefore not compatible with each other, now, the presence

of so many different standards creates the opposite problem of which standard should be supported in which context.

The development of a great variety of strategies in the plethora of IoT scenarios and domains has demonstrated the existence of common/recurring problems that can be managed using very similar solutions. Sometimes, a solution is so reliable and well performing that it is widely adopted. A solution that has proven to be a best practice to tackle a specific problem is usually defined as a pattern. The adoption of patterns is widespread in several fields ranging from architecture and engineering, for the design and construction of the means of transport, buildings, infrastructures, and urban nuclei [3], to narration and music composition [4,5], to behavioral and social skills [6], to information exchange [7]. A relatively common field of application regards software engineering, where design patterns are widely adopted to develop more reusable and reliable software solutions [8].

Depending on the area of application, a pattern may be composed of different parts. For example, in the case of IoT patterns [9], both the software and the hardware involved in the logical and physical interaction among objects or between objects and the surrounding environment should be described. It is important to note that a pattern does not represent a final implementation, but is a template that suggests a solution for the same problem occurring in different contexts. Similarly, an IoT pattern provides a guide to design an IoT system: it helps the designer to manage constraints regarding logical aspects such as the protocol to exchange information or the role and relations between objects, but also the position of objects, the number of features per object, etc.

When moving from the Internet of Things to the Internet of Robotic Things (IoRT) domain [10–12], the interaction and intervention of the “things” with the environment represent a relevant part of their activity.

The IoRT is built on robotic IoT-based systems, merging the digital and the physical worlds. The concept of robots seen as components (“things”) of the Internet of Things was introduced for the first time in [13]. The cloud-based framework behind these systems allows them to share and collect information among both humans and machines [14]. Although being introduced in the context of Industry 4.0, recently, the IoRT has also been employed in many other scenarios, such as museums, entertainment, and Ambient Assisted Living (AAL) [11,15], where it is crucial to consider both the presence and the intervention of humans.

According to the definitions given in [16–18], AAL refers to the ecosystem of assistive products, solutions, and services, aimed at realizing intelligent environments in order to support elderly and/or disabled people in their daily activities. The ultimate goal is to enhance both the health and the quality of life of the elderly by providing a safe and protected living environment.

Especially when dealing with the assisted living scenario, a set of further constraints and considerations must be taken into account while designing IoRT solutions. In fact, although IoT solutions have already been proposed in the literature for the AAL scenario [16,18,19], the presence of humans and the interaction with them become much more relevant in the IoRT case. This leads to the need for identifying and defining patterns that can support designers. Starting from the definition of IoT patterns, we propose the introduction of IoRT patterns, which is the main contribution of this paper. In fact, the concept of patterns in the Internet of Robotic Things context has never been adopted in the related literature, although we believe they can considerably help the designers in the development and implementation of IoRT solutions. A case study from the AAL scenario will also be presented, where the authors have been involved in the framework of the SUMMITproject for the development of multi-platform IoT applications.

The remainder of this paper is structured as follows. Section 2 reports a summary of the concept of patterns in the IoT context, concluding with the pattern description format adopted in this paper. Sections 3–5 present the three patterns proposed, respectively obstacle avoidance, indoor localization and inertial monitoring. These sections include a



related work subsection, reporting the works from which the patterns were identified and presented in the subsequent pattern definition subsection. A case study taken from the AAL scenario is presented in Section 6, including a real-world implementation of the three proposed patterns. In Section 7, a critical discussion on the limits of the design pattern approach and the challenges related to their use is reported, whereas final conclusions are drawn in the last section.

## 2. Related Work

As mentioned in the Introduction, patterns have been adopted for a long time in a wide range of fields. Recently, the fast spread of IoT-based applications has led to the definition of IoT patterns [9,20–22], with a special emphasis on the industrial scenario as well [23,24]. In all these works, the authors identified existing products and technologies in the IoT field to abstract patterns and, in some cases, divided template solutions and best practices into two categories, namely patterns and pattern candidates (not fully formulated templates that can be promoted to patterns in the future). Furthermore, in robotics, solutions to common problems have often been addressed through modularity [25], which can be somehow seen as the hardware counterpart of the software pattern. However, in the context of system-level hardware design, the problem of identifying hardware design patterns was explicitly addressed in [26], where the authors aimed at reinterpreting the original idea of software design patterns for the hardware case. A similar approach was also adopted in [27] for cloud computing and in [28] for so-called smart contracts in financial services.

In their extensive work [9,29,30], the authors went through major works regarding the identification, definition, and application of patterns [31–34] to select a set of elements, some of them required and other optional, useful to describe IoT patterns. In particular, there are the name and aliases to identify the pattern, the icon to associate a graphical element to the pattern in a diagram, a problem section to state the issue for which the pattern is designed, the context to describe the situation in which the problem occurs and any constraints, and the forces represented by possibly contradictory considerations to bear in mind while choosing a solution. Then, there is the solution section that describes steps to solve the problem and a result section that describes the context after applying the pattern. Any similar solution that does not deserve a separate pattern description is listed in the variants section. There is also a list of related patterns, patterns that can be applied together or exclude each other. Finally, an example section completes the pattern description. To keep pattern descriptions general, in accordance with [33], we describe three patterns, namely obstacle avoidance, indoor localization, and inertial monitoring, through the following elements: name, context, problem, forces, solution, and consequences (these ones generically adopted instead of results, but with a very similar meaning). We further extend pattern descriptions by including variants and providing a graphical representation of the pattern components and their interactions. In particular, light grey and dark grey boxes are respectively software and hardware components. In contrast, dashed boxes are those components that can be both local (i.e., within the device) or remote (i.e., in a companion device or following the principles of edge, fog, and cloud computing) in the IoRT network.

## 3. A Pattern for Obstacle Avoidance

In the following, the first pattern related to the obstacle avoidance function for a mobile robotic platform within an IoRT context is introduced. After a brief review of the well-established approaches to this problem, we proceed with the description of the pattern. It is worth underlining that most of the works presented in the following section are not designed for an IoRT network. However, the seed of the IoT interoperability and data exchange can often be found in the distributed systems proposed. Furthermore, some parts of the presented solutions do not strictly require the presence of a network, while some other functions can be generalized to network-based or even cloud-based solutions, as better explained in the following.

### 3.1. Related Work

The obstacle avoidance problem is part of the wider navigation problem for a robotic vehicle. It consists of computing a motion control that avoids collisions with the obstacles detected by the sensors while the robot is driven to reach the target location by a higher level planner. The result of applying this control action at each time is a sequence of motions that drive the vehicle free of collisions towards the target [35].

A classification of obstacle avoidance algorithms can be done, starting from the so-called one-step methods that directly transform the sensor information to motion control. These methods assimilate the obstacle avoidance to a known physical problem like, for example, the Potential Field Method (PFM) [36,37]. The PFM adopts an analogy in which the robot is a particle that is repelled by the obstacles due to repulsive forces  $F_{rep}$ . Moreover, this algorithm is also capable of handling a target location, which exerts an attractive force  $F_{att}$  on the particle. In this case, the most promising motion direction is computed to follow the direction of the artificial force induced by the sum of both potentials  $F_{tot}(q) = F_{att}(q) + F_{rep}(q)$ .

On the other hand, methods with more than one step compute some intermediate information, which is then further processed to obtain the motion. In particular, Reference [38] proposed the Vector Field Histogram (VFH), which solves the problem in two steps, while [39], with the Obstacle Restriction Method (ORM), solved the problem in three steps, where the result of the two first steps is a set of candidate motion directions. Finally, there are methods that compute a subset of velocity control commands, as for example the Dynamic Window Approach (DWA) described in [40], which provides a controller for driving a mobile base in the plane. Some of the algorithms listed above can be more suited to a specific navigation context compared to others, depending on environment uncertainty, speed of motion, etc.

Besides the algorithms, sensing and actuation play a crucial role in obstacle avoidance as well. In the last few years, the spread of the Robot Operating System (ROS) framework [41] has allowed the development of a common approach in tackling the general problem of robot navigation by bridging the gap between the hardware interface and software development. Such an approach is included in the so-called navigation stack (<http://wiki.ros.org/navigation> (accessed on 2 April 2021)) which exploits range-sensing devices and obstacle maps. In particular, the obstacle avoidance part is performed by a local path planner module, implementing several types of control algorithms [42]. A large variety of robots (<https://robots.ros.org/> (accessed on 2 April 2021)) adopt the ROS navigation stack to perform both local and global path planning, thus representing a noticeable example of a modular and generally applicable solution to the recurring problem of navigation.

In the AAL scenario as well, several examples of mobile robotic platforms can be found, which are designed for the mobility support of the elderly and people with impairments. Most of the times, they are typically referred to as smart walkers [43] since they provide high-level functions such as monitoring of the user's status, localization, and obstacle avoidance. In relation to their propulsion, smart walkers can be classified into two broad categories: passive and active walkers. In the former, the user is responsible for the motion, while in the latter, the wheels are actuated by electric motors. We will now focus on the obstacle avoidance function of the smart walkers. For both walker categories, the related literature presents a considerable number of works, which in any case show similar features and share the same grounding idea.

In [44], a smart walker prototype was developed on top of an omnidirectional drive robot platform. To detect obstacles at various heights, the robot is equipped with arrays of ultrasonic transducers and infrared near-range sensors, while a laser range finder is used for mapping and localization. Force sensors are embedded in the handlebars for the acquisition of the force exerted by the user. Furthermore, a map of the environment is provided to consider those obstacles that the sensors could not detect, e.g., staircase. The collision avoidance module directly controls the velocity and the heading of the robot

to avoid obstacles while the platform is driving to the target location. Actual robot motion commands through waypoints are given by the local motion planner [45].

The walker presented by [46] consists of a custom frame, two caster wheels, two servo-braked wheels for steering, a laser range finder, and a controller. In this work, the obstacle avoidance was performed by an adaptive motion control algorithm incorporating environmental information, which was based on the artificial potential field method [36]. In [47], a passive smart walker was proposed, with obstacle avoidance and navigation assistance functions. It is equipped with a laser range finder (used for obstacle detection), frontal and side sonar sensors (to detect transparent objects and objects outside the laser scanning plane), two encoders for odometric localization, and a potentiometer on the steering wheel for heading acquisition. A geometry-based obstacle avoidance algorithm [48] is used, which searches for clear paths in the area in front of the walker while still reacting to user inputs. Similarly, References [46,49] presented a passive walker with a 2D laser scanner and wheel brakes actuated by servos, thus ensuring passive guidance. In this case as well, force sensing on the two handles, for the recognition of user intention, was included.

In [50], a passive smart walker was developed, by enhancing a commercial rollator with three low-cost Time-of-Flight (ToF) laser-ranging sensors (one frontal and two on the sides), a magnetometer for the heading acquisition, force sensors on the handlebars, brake actuators, and a companion computer, used for the wireless connection to a custom IoT platform. The obstacle avoidance solution proposed can handle both static and dynamic obstacles. Static obstacles are included in a map. The 2D localization inside the map is given by an indoor localization system, whose data can be accessed through the IoT platform. The braking system is operated in such a way to guide the elderly user's walking to avoid a collision every time a close obstacle is localized from the map or through the ranging sensors for dynamic obstacle detection.

### 3.2. Pattern Definition

**Context:** The real world where a mobile robotic platform has to move is characterized by the presence of "obstacles", which may hinder or even damage the vehicle during its motion. We assume to have a map of the environment (including static obstacles) and a localization system.

**Problem:** A mobile vehicle must be capable of avoiding obstacles in the considered environment (while trying to reach the target position given by the high-level planning). If no target is given, a mobile vehicle must be capable of wandering in its operating environment without hitting obstacles.

**Forces:**

- The possibility to consider any type of obstacle (both static and dynamic obstacles).
- The definition of a proper control strategy.
- The adoption of suitable sensors.

**Solution:** Provide the mobile robot with range sensing, a steering system, and a processing unit. Define a control algorithm to process the range and the localization data, and act on the navigation system to divert the path of the mobile robot from the obstacles. The map of the environment is used for static obstacles and those obstacles undetectable by the range sensors. A representation of this IoRT pattern is given in Figure 1.

**Consequences:**

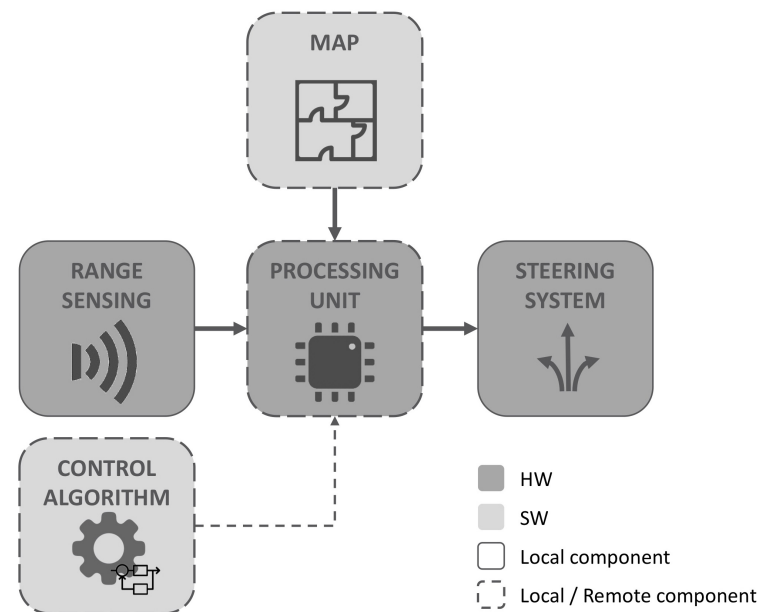
**Benefits:**

- The guidance of the mobile robot in such a way to keep it safely away from known or sudden obstacles.
- The avoidance of potentially destructive impacts.
- Extended time of exploration of the environment.

**Liabilities:**

- Sensors' and actuators' integration on the platform.
- Increased weight and, possibly, size of the platform.

- Potential presence of blind spots, depending on the adopted solution.  
Variants: local re-planning, motion control, and traction control.  
Related patterns: patterns for environment mapping, device localization solutions, and high-level path planning.  
Aliases: collision avoidance.



**Figure 1.** Representation of the obstacle avoidance pattern detailing hardware (HW), software (SW), local, and local/remote components.

#### 4. A Pattern for Indoor Localization

Similar to what was done in the previous section, the pattern related to indoor localization is introduced below. After a brief overview of the state-of-the-art regarding the main approaches to this problem, we proceed with the description of the pattern.

##### 4.1. Related Work

Indoor localization is an across-the-board research topic, and in recent years, some of the most prominent solutions introduced at the academic level have reached the market. The variety of technological solutions proposed in the literature is usually strongly dependent on the specific application domain. It is common to apply data fusion algorithms to obtain better results. Applications typically fall within the scope of indoor localization, indoor navigation, and Location-Based Services (LBS). Among the systems described in the literature, the most common are:

**Dead reckoning systems:** They use sensors such as accelerometers, magnetometers, and gyroscopes to provide a quick estimate of the user's current position starting from his/her (known) initial position. Although dead reckoning-based systems can theoretically be accurate, in practice, they are subject to strong drift errors introduced, for example, by the quality of the sensors adopted. It is also important to manage paths that are "non-continuous" due to the presence of elevators or escalators. For these reasons, a periodic re-calibration is performed to restore the cumulative drift error, such as in [51,52]. In general, dead reckoning systems are used in combination with other indoor location technologies to improve the accuracy of the entire system.

**Systems based on interaction with wireless communication technologies:** Considering the widespread use of mobile devices with wireless connectivity, the localization systems based on WLAN and Bluetooth (in both standard and Low Energy versions) are among the most common [53]. Although these systems may also be used outdoors, the existence of other more common and accurate solutions for open-air environments (such as GPS)

makes WiFi-based solutions more advisable for indoors. Among the main measurement paradigms for these solutions, we have:

- **AoA (Angle of Arrival):** This technique, through the measurement of the angle of arrival of a signal sent from the user to a BS (Base Station), is able to define a portion of the area in which the user is located. The error in the position estimation is strongly affected by errors in the estimation of the arrival angle;
- **RSS (Received Signal Strength):** This approach mainly uses two classes of methods. Trilateration methods work by evaluating at least three distances between the user and base station; then, the user's position is calculated through classical trigonometric techniques. Fingerprinting methods, on the other hand, are based on the definition of a database of "fingerprints" of the signal as the position of the receiving nodes changes in the space. Distance estimation is carried out by checking the intensity of the received signal (on-line phase) with the previously defined database (off-line phase);
- **ToA (Time of Arrival):** This approach measures the time taken by the signal to propagate from the user to the receiving node. In this case as well, with the measurements coming from at least three different receiving nodes, it is possible to localize the user through trilateration algorithms.

**Magnetic field-based systems:** These systems exploit the magnetic fields that characterize an environment due to the presence of elevators, escalators, doors, pillars, and other ferromagnetic structures [54,55]. In these systems, magnetic signatures are collected on a map (off-line phase) and can be subsequently exploited by the user, through his/her smartphone, to know his/her position (on-line phase). The interest around this approach has increased in recent years, and there are some commercial solutions such as Indoor Atlas [56].

**Audio-based systems:** Audio-based systems exploit uncontrolled environmental sounds (e.g., noise from the background) or controlled sounds (such as those generated from loudspeakers in shopping malls, shops, and museums) [57,58]. Depending on the particular approach adopted, they can be used both to locate the user precisely and as an add-on for increasing the accuracy of other technologies.

**Localization systems based on ultrasonic transducers:** These systems are generally composed of two elements: the transmitting node (the node from which the ultrasonic signal starts, in general, fixed to the roof) and the receiving node, which in the context of the location, represents the node that we want to determine the position (target) [59–61].

**RFID-based systems:** A typical system based on RFID is composed of tags, readers, and back-end electronics for data processing. The performance of these systems is highly dependent on the type of tag (active or passive) and on the specific frequency band [62,63].

**Systems based on Visible Light Communication (VLC):** These exploit the susceptibility of LEDs to amplitude modulation at high frequencies to transmit information in the environment [64,65]. If the frequency exceeds a "flicker fusion" threshold, the lighting functionality is kept intact since the modulation is not perceived by the human eye, while it is possible to exploit the information generated through modulation to locate the user. This information, for example, can be associated with a point on a map. Thanks to the fact that the sensors inside smartphone cameras are composed of photodiode arrays, it is possible to use a smartphone as the device for indoor location based on VLC. The use of light sources such as those used in the VLC framework allows exploiting algorithms valid for electromagnetic waves (such as the AoA and ToA) briefly introduced for the systems based on the interaction with wireless communication technologies.

**Systems based on computer vision algorithms:** These systems exploit the recognition of particular tags (marker-based systems) or of characteristic objects/points (marker-less systems) associated with specific positions within an environment [66–69]. Marker-based approaches require that a set of 2D markers be placed in the environment in which the localization mechanism has to be enabled. Each ID and therefore each marker is associated during the off-line setup phase with a known position on a map. On the other hand,



marker-less approaches are based on the analysis of the characteristics of perceived images: the user's position is obtained from what the smartphone sees in the environment.

Hybrid systems: As noted above, usually to increase accuracy while reducing the cost/performance ratio of an indoor localization system, it is preferable to use multiple solutions at the same time, as presented in [70–72]. This choice is particularly interesting for the systems associated with the use of smartphones, thanks to the fact that this category of devices features different types of sensors (inertial, magnetic, and microphone), at least one camera, and antennas to manage wireless connectivity (WiFi, BT/BLE).

#### 4.2. Pattern Definition

Context: It is necessary to access localization information in environments where outdoor positioning systems (such as GPS, GLONASS, and GALILEO) do not directly work. Localization can be performed with different accuracy levels depending on the application.

Problem: A user needs to know his/her own position in an indoor environment or there is a need to know the position of a user/object.

Forces:

- The possibility to intervene in the environment, e.g., place tags, lights, beacons, and base stations (in case they are not already present).
- Setup time and costs: localization could be necessary for temporary events or to monitor a single person, as may happen in AAL scenarios.
- Required accuracy level.

Solution: As stated in the problem paragraph, there are two indoor localization cases: (a) a user tracks his/her position; (b) someone else tracks the position of a user/object. In the first case (a), using a handheld device to know the current position is a common habit indoors. Given that, solutions require a map of the environment that can contain additional information such as the position of tags (e.g., visual, textual, and graphic), the distribution of magnetic or radio fingerprints, or even the position of landmarks (used in dead reckoning systems to reduce drift). When approaches requiring trilateration are applied, users receive their position from external parties (e.g., base stations or servers that compute the position). In the second case (b), external entities monitor the user/object position, given that the device held by the user or attached to the object (e.g., a smart walker) can be straightforward. A graphic representation of this IoRT pattern is depicted in Figure 2, where, as already said above, dashed boxes represent those components that can be either on-board the device (or worn by the user) or remote, i.e., other nodes within the IoRT network.

Consequences:

Benefits:

- Monitor the position of objects and people.
- Enable services based on position (LBS).
- Find people in cases of emergency.
- Help people move inside buildings, airports, and malls.

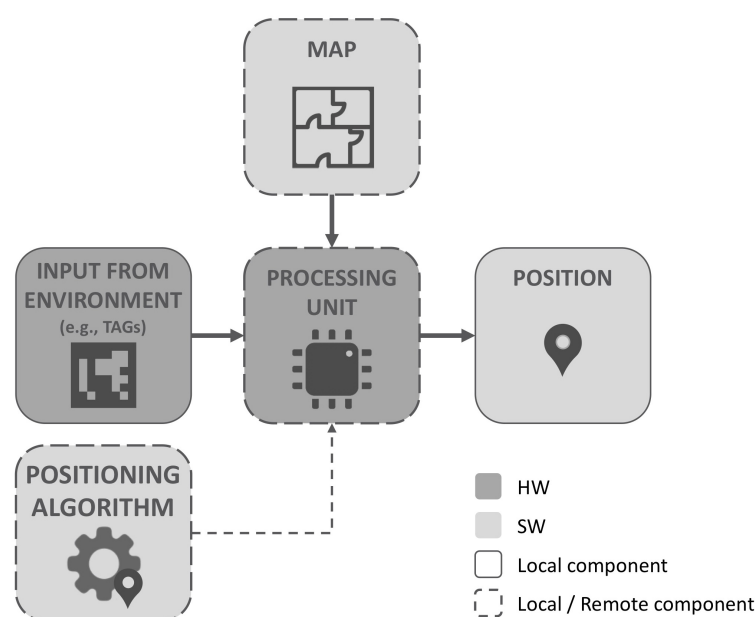
Liabilities:

- Need to calibrate the system over time.
- Effort to map/set up an environment (e.g., to build a database of “fingerprints” or to place beacons and tags).

Variants: indoor navigation.

Related patterns: patterns for environment mapping.

Aliases: indoor positioning and indoor tracking.



**Figure 2.** Representation of the indoor localization pattern detailing hardware (HW), software (SW), local, and local/remote components.

## 5. A Pattern for Inertial Monitoring

Following the structure already defined in the previous sections, a pattern for inertial monitoring is now introduced and described. As will be clarified in Section 5.1, this type of pattern is crucial in AAL, industrial, and building maintenance scenarios.

### 5.1. Related Work

Inertial sensors have been widely adopted in a huge number of applications in recent years: production of smartphones, consoles, avionics, and robotic systems. Nevertheless, inertial systems can also be found in applications concerning seismology and structural vibration control, terrain roughness classification, fall detectors, and so on. Although the applications can be different, the way the data are processed and the hardware architectures often share common properties. In the following, a brief overview of the state-of-the-art in the area of inertial-based systems is given, and common properties, which will be used to define the pattern, are highlighted.

An area that has attracted a considerable amount of applications is related to health care and AAL. As an example, in [73], a human activity recognition system with multiple combined features to monitor human physical movements from continuous sequences via tri-axial inertial sensors was proposed. The author used a decision tree classifier optimized by the binary grey wolf optimization algorithm, and the experimental results showed that the proposed system achieved accuracy rates of 96.83% over the proposed human-machine dataset.

In the same context, Andò et al. presented in [74] the development of a fall and Activity of Daily Living (ADL) detector using a fully embedded inertial system. The classification strategy was mainly based on the correlation computed between run-time acquired accelerations and signatures (a model of the dynamic evolution of an inertial event). In particular, the maximum among the correlation coefficients was used as a feature to be passed to the classifier.

In the case of robotic applications, and especially in the context of autonomous ground vehicles, the assessment of the traversability level of the terrain is a fundamental task for safe and efficient navigation [75]. Oliveira et al. proposed in [76] the use of inertial sensors to estimate the roughness properties of the terrain and to autonomously regulate the speed according to the terrain identified by the robot in order to avoid abrupt movements over

terrain changes. The classification was performed by employing a classifier capable of clustering different terrains based only on acceleration data provided by the inertial system.

In the area of personal service robots, a novel idea was proposed by [77]. Due to the growing interest in the use of robots in everyday life situations, and especially their interaction with the users, the article proposed the use of wearable inertial systems to improve the interaction between humans and robots while walking together. The inertial system was used to compute real-time features from the user's gait, which were consequently used to act on the control system of the robot.

In the context of gesture recognition for robot control, an interesting study was presented by [78]. With the development of smart sensing and the trend of aging in society, telerobotics is playing an increasingly important role in the field of home care service. In this domain, the authors presented a human-robot interface in order to help control the telerobot remotely by means of an instrumented wearable sensor glove based on Inertial Measurement Units (IMUs) with nine axes. The developed motion capture system architecture, along with a simple but effective fusion algorithm, was proven to meet the requirements of speed and accuracy. Furthermore, the architecture can also be extended to up to 128 nodes to capture the motion of the whole body.

A similar study was also presented in [79]. In addition to the previous one, in this study, the authors addressed the definition of a proper open-source framework for gesture-based robot control using wearable devices. A set of gestures was introduced, tested both with respect to performance aspects and considering qualitative human evaluation (with 27 untrained volunteers). Each gesture was described in terms of the three-dimensional acceleration pattern generated by the movement, and the architecture was developed in such a way to have a dedicated module to analyze on-line the acceleration data provided by the inertial sensor and a modeling module to create a representation of the gesture that maximized the accuracy of the recognition. The gesture classification was based on the Mahalanobis distance for the recognition step and on Gaussian mixture models and Gaussian mixture regression for the modeling step.

Another common research topic in the area of robotics is fault detection and diagnosis. As an example of an application, Zhang et al. presented in [80] a method for the fault detection of damaged UAV rotor blades was presented. Fault detection is a relevant topic, especially when dealing with UAVs since any defect in the propulsion system of the aerial vehicle will lead to the loss of thrust and, as a result, to the disturbance of the thrust balance, higher power consumption, and the potential crashing of the vehicle. The presented fault detection methodology was based on a signal processing of the vehicle acceleration and a Long Short-Term Memory (LSTM) network models. The vehicle accelerations, caused by unbalanced rotating parts, were retrieved from the onboard inertial measurement unit. Features based on wavelet packet decomposition were then extracted from the acceleration signal, and the standard deviations of the wavelet packet coefficients were employed to form the feature vector. The LSTM classifier was finally used to determine the occurrence of rotor fault.

## 5.2. Pattern Definition

**Context:** The need for detection and classification of inertial phenomena in order to act on the system/device under control.

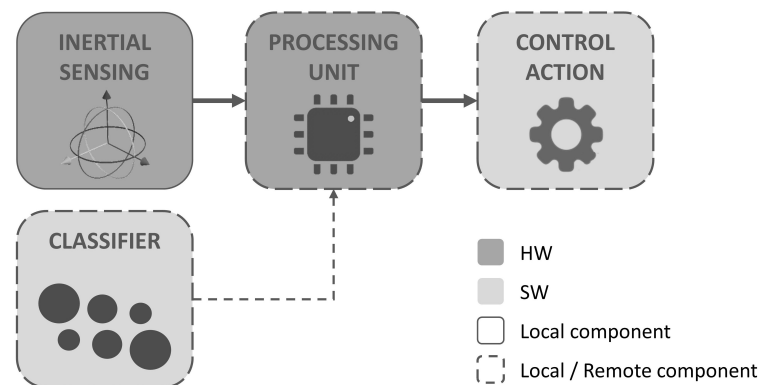
**Problem:** Detect and classify either environmental conditions or potentially dangerous inertial events for the activation of suitable control actions.

**Forces:**

- The definition of a suitable classification strategy.
- The definition of the required inertial sensor type and quantity for the task accomplishment.
- The available computing power.

**Solution:** Provide the entity to be monitored with a system including inertial sensors (such as accelerometers and gyroscopes) and a processing unit (microcontroller or micro-

processor). Define a classification methodology that, by processing the acquired inertial quantities, can associate a given unknown event with a given class of events. Define the classification-dependent control actions. A representation of this IoRT pattern is given in Figure 3.



**Figure 3.** Representation of the inertial monitoring pattern detailing hardware (HW), software (SW), local, and local/remote components.

Consequences:

Benefits:

- Readily activate proper control actions according to the inertial event.
- Enhanced comprehension of your system/device.

Liabilities:

- The development of the classification methodologies.
- For custom applications, the need to develop custom devices.
- Feature selection is usually application dependent.

Variants: gesture recognition and terrain classifier.

Related patterns: patterns for inertial signature extraction.

Aliases: inertial event classification.

## 6. Case Study

In this section, a real-world case study including the implementation of the three patterns proposed, namely obstacle avoidance, indoor localization, and inertial monitoring, is reported and described. The solution was developed in the framework of the SUMMIT project, which aimed at realizing an Internet of Things platform for three application scenarios: smart health, smart cities, and smart industry.

In the context of smart health and smart assisted living, a smart walker [50] was developed as a smart “robotic thing” of the platform. As partially described in the previous sections, the walker is able to:

- localize itself through a multi-sensor system, combining trilateration data acquired by a network of ultrasound sensors and odometry data acquired by an inertial measurement unit [81];
- avoid obstacles by means of time of flight laser sensors, a map of known obstacles, and actuators in the brakes of the rear wheels [50]

Moreover, inertial monitoring of the walker user was performed by classifying accelerometer, gyroscope, and compass data acquired by a wearable sensing system, through an event correlated approach, in order to detect falls and, more generally, the activity of daily living. Furthermore, statistics on walker usage by the elderly person were assessed and how the user’s walking changed. All these tasks were carried out within a domestic environment, and all the data (e.g., ADL monitoring data, walker position, etc.) were shared through the IoT cloud both among the modules of the solution and with other external

components of the IoT network. A picture of the final setup of the smart walker is shown in Figure 4, whereas a block diagram showing the interconnection of the three patterns is depicted in Figure 5. Clearly, the testing environment was equipped with the necessary infrastructure for wireless communication and the indoor localization system [81].

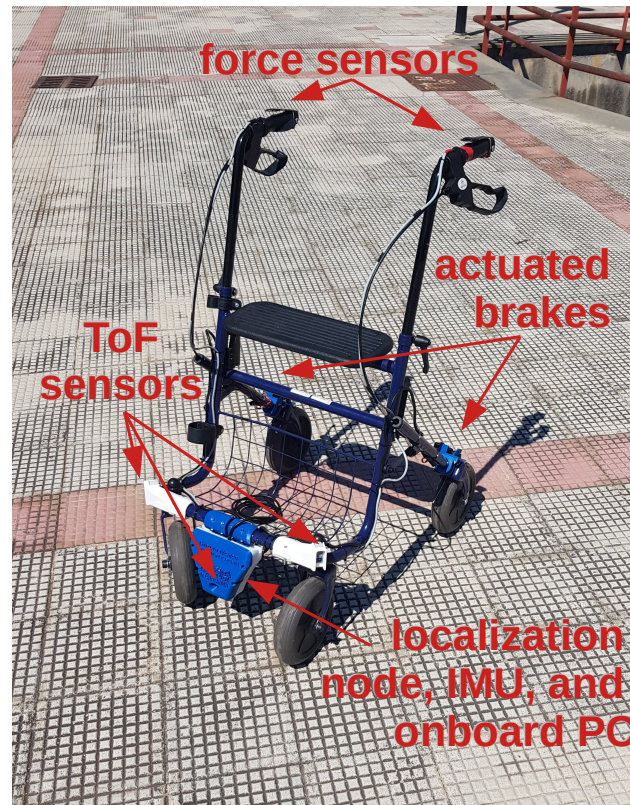


Figure 4. The smart walker developed for the SUMMIT project.

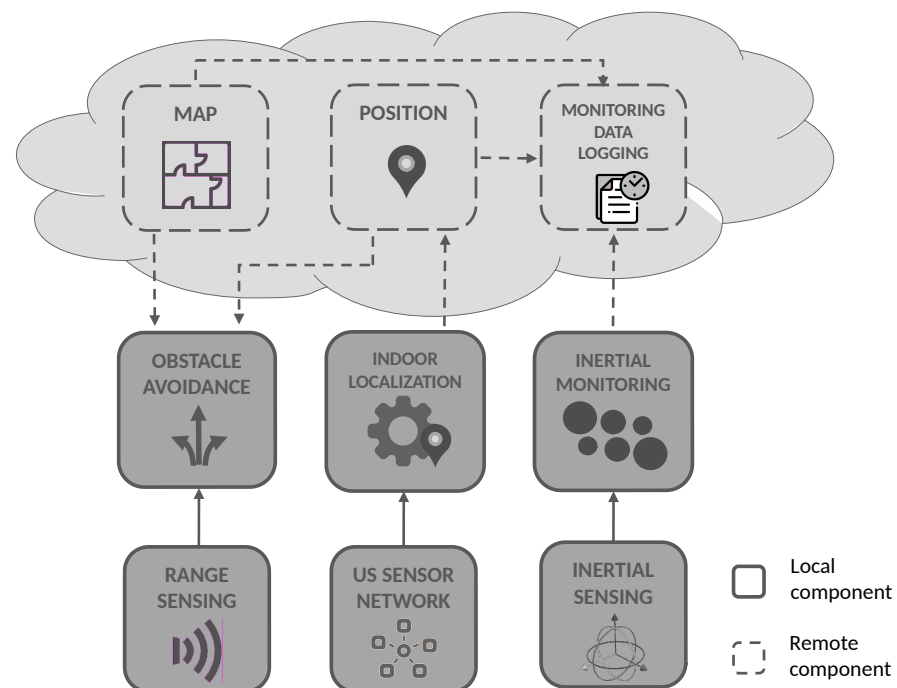
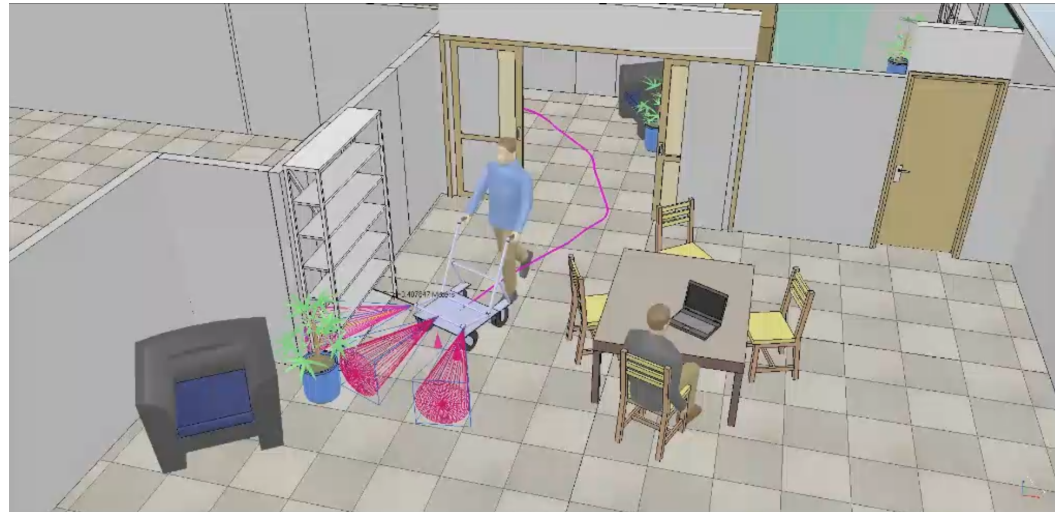


Figure 5. A block diagram showing the interconnection between the implemented patterns.



A preliminary testing phase was performed in the simulation in order to properly tune the adopted obstacle avoidance approach, as shown in Figure 6. Once the resulting behavior of the walker was acceptable, tests on the real setup were performed in order to assess the proper interconnection and communication among the modules of the solution through the network. A set of different operating conditions of the walker, depending on the presence and position of the obstacle, is shown in Figure 7.



**Figure 6.** A simulation session during the testing of the obstacle avoidance algorithm.



(a)

**Figure 7.** *Cont.*



(b)



(c)

**Figure 7.** Different operating conditions of the smart walker: (a) no obstacle is present: brakes are not actuated; (b) an obstacle on the left is detected: the left brake is actuated; (c) a frontal obstacle is detected: both brakes are actuated.

## 7. Discussion

The adoption of patterns during the design and development of the IoRT solution described in the previous section allowed us to clearly identify the boundaries of each part of the whole robotic system and to treat it as a separate element. The modularity of this approach is particularly suitable for the continually changing nature of a network of “things”, in terms of both continuous integration and types of interacting devices.

However, this comes at the expense of a considerable overhead in the data flow. For instance, thinking of the proposed case study, the walker position is first shared to the cloud by the walker and then retrieved again by the walker itself for obstacle avoidance (taking into account the obstacle map). Although this approach might look convoluted, as localization data produced on board the smart walker might be directly “consumed” by the walker itself, such an approach allows the smooth replacement of the walker with any other device that may need localization.

Clearly, this implies the need to have a stable and reliable connectivity among all the network components, especially in those cases where real-time demanding applications are required. Such a requirement introduces another criticality of the IoRT approach; however, this issue is becoming minor thanks to the possibility to have well-performing networks currently, even in domestic environments (as in the AAL scenario). This issue becomes more relevant both in the smart industry and in the smart city scenarios, as in the former, electromagnetic disturbances may compromise the quality of the connection, whereas, in the latter, typically, very large spaces have to be covered.

## 8. Conclusions

After the establishment of the IoT, the novel paradigm of the Internet of Robotic Things is now growing in the research community, as the crossover of the IoT and robotics. The integration of robotic systems into the wider context of the IoT provides such platforms with a higher level of situational awareness [82], which can be fruitfully exploited. The possibility to access a map, to localize within an indoor environment, or even to exploit remote computational resources opens up a wider variety of application scenarios and opportunities that could not be fully explored before.

As very similar approaches to basic functions, such as obstacle avoidance, indoor localization, and inertial monitoring, have been identified in the literature, in this paper, we proposed to abstract such approaches and to rearrange them by leveraging the concept of patterns. In this manner, three design patterns were presented, with the aim of simplifying the work of the IoRT designer, by allowing him/her to quickly determine whether the solution fits the problem at hand and to identify the core elements for the subsequent implementation.

We hope this first attempt of pattern definition in the IoRT context may encourage the identification and definition of many more patterns, with the final goal of coming to a large and thorough pattern language for the Internet of Robotic Things. As a future development, we plan to define and extract patterns for the IoRT from other application contexts, besides the AAL, such as agricultural robotics, navigation in unstructured environments, and robot cooperation.

**Author Contributions:** Conceptualization, B.A., L.C., V.C., R.C., D.C.G., S.M., and G.M.; methodology, S.M. and V.C.; investigation, L.C., R.C., D.C.G., and S.M.; writing—original draft preparation, L.C., R.C., D.C.G., and S.M.; writing—review and editing, R.C., D.C.G., and S.M.; supervision, B.A., V.C., and G.M.; funding acquisition, B.A., V.C., and G.M. All authors read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Italian Ministry of Economic Development (MISE) within the project “SUMMIT”-PON 2014-2020 FESR-F/050270/01-03/X32.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

## References

1. Kellmereit, D.; Obodovski, D. *The Silent Intelligence: The Internet of Things*; DnD Ventures: Grove City, MN, USA, 2013.
2. Ashton, K. That ‘Internet of Things’ thing. *RFID J.* **2009**, *22*, 97–114.
3. Alexander, C. *A Pattern Language: Towns, Buildings, Construction*; Oxford University Press: Oxford, UK, 1977.
4. Snowden, D. Narrative patterns: Uses of story in the third age of knowledge management. *J. Inf. Knowl. Manag.* **2002**, *1*, 1–6. [[CrossRef](#)]
5. Rolland, P.Y.; Ganascia, J.G. Musical Pattern Extraction and Similarity Assessment. In *Readings in Music and Artificial Intelligence*; Routledge: London, UK, 2013; pp. 125–154.
6. Kinatader, M.; Warren, W.H. Social Influence on Evacuation Behavior in Real and Virtual Environments. *Front. Robot. AI* **2016**, *3*, 43. [[CrossRef](#)]



7. Pitonakova, L.; Crowder, R.; Bullock, S. Information Exchange Design Patterns for Robot Swarm Foraging and Their Application in Robot Control Algorithms. *Front. Robot. AI* **2018**, *5*, 47. [[CrossRef](#)] [[PubMed](#)]
8. Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 1995.
9. Reinfurt, L.; Breitenbücher, U.; Falkenthal, M.; Leymann, F.; Riegg, A. Internet of Things Patterns. In Proceedings of the 21st European Conference on Pattern Languages of Programs (EuroPLOP), Kaufbeuren, Germany, 6–10 July 2016; ACM: New York, NY, USA, 2016; p. 5. [[CrossRef](#)]
10. Kara, D.; Carlaw, S. *The Internet of Robotic Things*; Technical Report; ABI Research: New York, NY, USA, 2014.
11. Ray, P.P. Internet of Robotic Things: Concept, technologies, and challenges. *IEEE Access* **2016**, *4*, 9489–9500. [[CrossRef](#)]
12. Vermesan, O.; Bahr, R.; Ottella, M.; Serrano, M.; Karlsen, T.; Wahlström, T.; Sand, H.E.; Ashwathnarayan, M.; Gamba, M.T. Internet of Robotic Things Intelligent Connectivity and Platforms. *Front. Robot. AI* **2020**, *7*, 104. [[CrossRef](#)] [[PubMed](#)]
13. Plauska, I.; Damaševičius, R. Educational Robots for Internet-of-Things Supported Collaborative Learning. In *Information and Software Technologies*; Dregvaite, G., Damasevicius, R., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 346–358.
14. Romeo, L.; Petitti, A.; Marani, R.; Milella, A. Internet of Robotic Things in Smart Domains: Applications and Challenges. *Sensors* **2020**, *20*, 3355. [[CrossRef](#)]
15. Gomez, M.A.; Chibani, A.; Amirat, Y.; Matson, E.T. IoRT cloud survivability framework for robotic AALs using HARMS. *Robot. Auton. Syst.* **2018**, *106*, 192–206. [[CrossRef](#)]
16. Maskeliūnas, R.; Damaševičius, R.; Segal, S. A review of internet of things technologies for ambient assisted living environments. *Future Internet* **2019**, *11*, 259. [[CrossRef](#)]
17. Abtoy, A.; Touhafi, A.; Tahiri, A. Ambient Assisted living system's models and architectures: A survey of the state of the art. *J. King Saud-Univ.-Comput. Inf. Sci.* **2020**, *32*, 1–10.
18. Marques, G. Ambient Assisted Living and Internet of Things. In *Harnessing the Internet of Everything (IoE) for Accelerated Innovation Opportunities*; IGI Global: Hershey, PA, USA, 2019; pp. 100–115. [[CrossRef](#)]
19. Dohr, A.; Modre-Oprian, R.; Drobics, M.; Hayn, D.; Schreier, G. The Internet of Things for Ambient Assisted Living. In Proceedings of the 2010 Seventh International Conference on Information Technology: New Generations, Las Vegas, NV, USA, 12–14 April 2010; pp. 804–809. [[CrossRef](#)]
20. Qanbari, S.; Pezeshki, S.; Raisi, R.; Mahdizadeh, S.; Rahimzadeh, R.; Behinaein, N.; Mahmoudi, F.; Ayoubzadeh, S.; Fazlali, P.; Roshani, K.; et al. IoT Design Patterns: Computational Constructs to Design, Build and Engineer Edge Applications. In Proceedings of the 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI), Berlin, Germany, 4–8 April 2016; pp. 277–282. [[CrossRef](#)]
21. Tkaczyk, R.; Wasielewska, K.; Ganzha, M.; Paprzycki, M.; Pawlowski, W.; Szmeja, P.; Fortino, G. Cataloging design patterns for Internet of Things artifact integration. In Proceedings of the 2018 IEEE International Conference on Communications Workshops (ICC Workshops), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
22. Washizaki, H.; Ogata, S.; Hazeyama, A.; Okubo, T.; Fernandez, E.B.; Yoshioka, N. Landscape of Architecture and Design Patterns for IoT Systems. *IEEE Internet Things J.* **2020**, *7*, 10091–10101. [[CrossRef](#)]
23. Bloom, G.; Alsulami, B.; Nwafor, E.; Bertolotti, I.C. Design patterns for the Industrial Internet of Things. In Proceedings of the 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), Imperia, Italy, 13–15 June 2018; pp. 1–10.
24. Ghosh, A.; Edwards, D.J.; Hosseini, M.R. Patterns and trends in Internet of Things (IoT) research: Future applications in the construction industry. *Eng. Constr. Archit. Manag.* **2020**, *28*, 457–481. [[CrossRef](#)]
25. Virk, G.S.; Park, H.S.; Yang, S.; Wang, J. ISO modularity for service robots. In *Advances in Cooperative Robotics*; World Scientific: Singapore, 2017; pp. 663–671.
26. Damasevicius, R.; Majauskas, G.; Stukys, V. Application of design patterns for hardware design. In Proceedings of the Design Automation Conference (IEEE Cat. No.03CH37451), Anaheim, CA, USA, 2–6 June 2003; pp. 48–53. [[CrossRef](#)]
27. Malcher, V. Design Patterns in Cloud Computing. In Proceedings of the 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), Krakow, Poland, 4–6 November 2015; pp. 32–35. [[CrossRef](#)]
28. Bartoletti, M.; Pompianu, L. An Empirical Analysis of Smart Contracts: Platforms, Applications, and Design Patterns. In *Financial Cryptography and Data Security*; Springer International Publishing: Cham, Switzerland, 2017; pp. 494–509.
29. Reinfurt, L.; Breitenbücher, U.; Falkenthal, M.; Leymann, F.; Riegg, A. Internet of Things Patterns for Devices: Powering, Operating, and Sensing. *Int. J. Adv. Internet Technol. IARIA* **2017**, *10*, 106–123.
30. Reinfurt, L.; Breitenbücher, U.; Falkenthal, M.; Leymann, F.; Riegg, A. Internet of Things patterns for communication and management. In *Transactions on Pattern Languages of Programming IV*; Lecture Notes in Computer Science (LNCS); Springer: Berlin/Heidelberg, Germany, 2019; Volume 10600, pp. 139–182. [[CrossRef](#)]
31. Meszaros, G.; Doble, J. Metapatterns: A pattern language for pattern writing. In Proceedings of the 3rd Pattern Languages of Programming Conference, Monticello, Monticello, IL, USA, 6–8 September 1996; pp. 1–39.
32. Harrison, N. Advanced Pattern Writing. *Pattern Lang. Program Des.* **2006**, *5*, 433.
33. Wellhausen, T.; Fiesser, A. How to write a pattern? A rough guide for first-time pattern authors. In Proceedings of the 16th European Conference on Pattern Languages of Programs, Irsee, Germany, 13–17 July 2012; p. 5.

34. Fehling, C.; Barzen, J.; Breitenbücher, U.; Leymann, F. A process for pattern identification, authoring, and application. In Proceedings of the 19th European Conference on Pattern Languages of Programs, Irsee, Germany, 9–13 July 2014; p. 4.
35. Siciliano, B.; Khatib, O. *Springer Handbook of Robotics*; Springer: Berlin/Heidelberg, Germany, 2016.
36. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous Robot Vehicles*; Springer: Berlin/Heidelberg, Germany, 1986; pp. 396–404.
37. Krogh, B.; Thorpe, C. Integrated path planning and dynamic steering control for autonomous vehicles. In Proceedings of the 1986 IEEE International Conference on Robotics and Automation, Francisco, CA, USA, 7–10 April 1986; Volume 3, pp. 1664–1669.
38. Borenstein, J.; Koren, Y. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Trans. Robot. Autom.* **1991**, *7*, 278–288. [\[CrossRef\]](#)
39. Minguez, J. The obstacle-restriction method for robot obstacle avoidance in difficult environments. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005; pp. 2284–2290. [\[CrossRef\]](#)
40. Fox, D.; Burgard, W.; Thrun, S. The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. [\[CrossRef\]](#)
41. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In *ICRA Workshop on Open Source Software*; ICRA: Kobe, Japan, 2009; Volume 3, p. 5.
42. Filotheou, A.; Tsardoulas, E.; Dimitriou, A.; Symeonidis, A.; Petrou, L. Quantitative and qualitative evaluation of ROS-enabled local and global planners in 2D static environments. *J. Intell. Robot. Syst.* **2019**, *98*, 567–601.
43. Martins, M.; Santos, C.; Frizera, A.; Ceres, R. A review of the functionalities of smart walkers. *Med. Eng. Phys.* **2015**, *37*, 917–928. [\[CrossRef\]](#)
44. Morris, A.; Donamukkala, R.; Kapuria, A.; Steinfeld, A.; Matthews, J.T.; Dunbar-Jacob, J.; Thrun, S. A robotic walker that provides guidance. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), Taipei, Taiwan, 14–19 September 2003; Volume 1, pp. 25–30.
45. Roy, N.; Thrun, S. Motion planning through policy search. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 30 September–4 October 2002; Volume 3, pp. 2419–2424.
46. Hirata, Y.; Hara, A.; Kosuge, K. Motion control of passive intelligent walker using servo brakes. *IEEE Trans. Robot.* **2007**, *23*, 981–990. [\[CrossRef\]](#)
47. Rentschler, A.J.; Simpson, R.; Cooper, R.A.; Boninger, M.L. Clinical evaluation of Guido robotic walker. *J. Rehabil. Res. Dev.* **2008**, *45*, 1281–1293. [\[CrossRef\]](#)
48. Rodriguez-Losada, D.; Matia, F.; Jimenez, A.; Galan, R.; Lacey, G. Implementing map based navigation in Guido, the robotic smartwalker. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 3390–3395.
49. Huang, Y.C.; Yang, H.P.; Ko, C.H.; Young, K.Y. Human intention recognition for robot walking helper using ANFIS. In Proceedings of the 8th Asian Control Conference (ASCC), Kaohsiung, Taiwan, 15–18 May 2011; pp. 311–316.
50. Cantelli, L.; Guastella, D.; Mangiameli, L.; Melita, C.D.; Muscato, G.; Longo, D. A walking assistant using brakes and low cost sensors. In Proceedings of the 21th International Conference on Climbing and Walking Robots (CLAWAR) 2018, Panama City, Panama, 10–12 September 2018; pp. 53–60.
51. Zhao, H.; Zhang, L.; Qiu, S.; Wang, Z.; Yang, N.; Xu, J. Pedestrian dead reckoning using pocket-worn smartphone. *IEEE Access* **2019**, *7*, 91063–91073. [\[CrossRef\]](#)
52. Martinelli, A.; Gao, H.; Groves, P.D.; Morosi, S. Probabilistic context-aware step length estimation for pedestrian dead reckoning. *IEEE Sens. J.* **2017**, *18*, 1600–1611. [\[CrossRef\]](#)
53. Yassin, A.; Nasser, Y.; Awad, M.; Al-Dubai, A.; Liu, R.; Yuen, C.; Raulefs, R.; Aboutanios, E. Recent advances in indoor localization: A survey on theoretical approaches and applications. *IEEE Commun. Surv. Tutor.* **2016**, *19*, 1327–1346. [\[CrossRef\]](#)
54. Ashraf, I.; Kang, M.; Hur, S.; Park, Y. MINLOC: Magnetic field patterns-based indoor localization using convolutional neural networks. *IEEE Access* **2020**, *8*, 66213–66227. [\[CrossRef\]](#)
55. Galván-Tejada, C.E.; Zanella-Calzada, L.A.; García-Domínguez, A.; Magallanes-Quintanar, R.; Luna-García, H.; Celaya-Padilla, J.M.; Galván-Tejada, J.I.; Vélez-Rodríguez, A.; Gamboa-Rosales, H. Estimation of Indoor Location Through Magnetic Field Data: An Approach Based On Convolutional Neural Networks. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 226. [\[CrossRef\]](#)
56. Haverinen, J. Utilizing Magnetic Field Based Navigation. U.S. Patent 8,798,924, 5 August 2014.
57. Chen, H.; Li, F.; Wang, Y. SoundMark: Accurate indoor localization via peer-assisted dead reckoning. *IEEE Internet Things J.* **2018**, *5*, 4803–4815. [\[CrossRef\]](#)
58. Leonardo, R.; Barandas, M.; Gamboa, H. A framework for infrastructure-free indoor localization based on pervasive sound analysis. *IEEE Sens. J.* **2018**, *18*, 4136–4144. [\[CrossRef\]](#)
59. Carter, S.A.; Avrahami, D.; Tokunaga, N. Using Inaudible Audio to Improve Indoor-Localization-and Proximity-Aware Intelligent Applications. *arXiv* **2020**, arXiv:2002.00091.
60. Lindo, A.; Garcia, E.; Urena, J.; del Carmen Perez, M.; Hernandez, A. Multiband waveform design for an ultrasonic indoor positioning system. *IEEE Sens. J.* **2015**, *15*, 7190–7199. [\[CrossRef\]](#)



61. Murata, S.; Yara, C.; Kaneta, K.; Ioroi, S.; Tanaka, H. Accurate indoor positioning system using near-ultrasonic sound from a smartphone. In Proceedings of the 2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies, Oxford, UK, 10–12 September 2014; pp. 13–18.
62. Zeng, Y.; Chen, X.; Li, R.; Tan, H.Z. UHF RFID indoor positioning system with phase interference model based on double tag array. *IEEE Access* **2019**, *7*, 76768–76778. [\[CrossRef\]](#)
63. Xu, H.; Wu, M.; Li, P.; Zhu, F.; Wang, R. An RFID indoor positioning algorithm based on support vector regression. *Sensors* **2018**, *18*, 1504. [\[CrossRef\]](#) [\[PubMed\]](#)
64. Jovicic, A.; Li, J.; Richardson, T. Visible light communication: Opportunities, challenges and the path to market. *IEEE Commun. Mag.* **2013**, *51*, 26–32. [\[CrossRef\]](#)
65. Li, Y.; Ghassemlooy, Z.; Tang, X.; Lin, B.; Zhang, Y. A VLC smartphone camera based indoor positioning system. *IEEE Photonics Technol. Lett.* **2018**, *30*, 1171–1174. [\[CrossRef\]](#)
66. Yang, S.; Ma, L.; Jia, S.; Qin, D. An Improved Vision-Based Indoor Positioning Method. *IEEE Access* **2020**, *8*, 26941–26949. [\[CrossRef\]](#)
67. Li, J.; Wang, C.; Kang, X.; Zhao, Q. Camera localization for augmented reality and indoor positioning: A vision-based 3D feature database approach. *Int. J. Digit. Earth* **2019**, 727–741. [\[CrossRef\]](#)
68. La Delfa, G.C.; Monteleone, S.; Catania, V.; De Paz, J.F.; Bajo, J. Performance analysis of visual markers for indoor navigation systems. *Front. Inf. Technol. Electron. Eng.* **2016**, *17*, 730–740. [\[CrossRef\]](#)
69. Dong, J.; Noreikis, M.; Xiao, Y.; Ylä-Jääski, A. Vinav: A vision-based indoor navigation system for smartphones. *IEEE Trans. Mob. Comput.* **2018**, *18*, 1461–1475. [\[CrossRef\]](#)
70. Yu, Y.; Chen, R.; Chen, L.; Guo, G.; Ye, F.; Liu, Z. A robust dead reckoning algorithm based on WiFi FTM and multiple sensors. *Remote Sens.* **2019**, *11*, 504. [\[CrossRef\]](#)
71. Qiu, S.; Wang, Z.; Zhao, H.; Qin, K.; Li, Z.; Hu, H. Inertial/magnetic sensors based pedestrian dead reckoning by means of multi-sensor fusion. *Inf. Fusion* **2018**, *39*, 108–119. [\[CrossRef\]](#)
72. Wang, X.; Yu, Z.; Mao, S. DeepML: Deep LSTM for indoor localization with smartphone magnetic and light sensors. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
73. Jalal, A.; Batool, M.; Kim, K. Stochastic Recognition of Physical Activity and Healthcare Using Tri-Axial Inertial Wearable Sensors. *Appl. Sci.* **2020**, *10*, 7122. [\[CrossRef\]](#)
74. Andò, B.; Baglio, S.; Crispino, R.; L'Episcopo, L.; Marletta, V.; Branciforte, M.; Virzi, M.C. A Smart Inertial Pattern for the SUMMIT IoT Multi-platform. In *Ambient Assisted Living*; Leone, A., Caroppo, A., Rescio, G., Diraco, G., Siciliano, P., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 311–319.
75. Guastella, D.C.; Muscato, G. Learning-Based Methods of Perception and Navigation for Ground Vehicles in Unstructured Environments: A Review. *Sensors* **2021**, *21*, 73. [\[CrossRef\]](#) [\[PubMed\]](#)
76. Oliveira, F.G.; Santos, E.R.S.; Neto, A.A.; Campos, M.F.M.; Macharet, D.G. Speed-invariant terrain roughness classification and control based on inertial sensors. In Proceedings of the 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), Curitiba, Brazil, 8–11 November 2017; pp. 1–6.
77. Moschetti, A.; Cavallo, F.; Esposito, D.; Penders, J.; Di Nuovo, A. Wearable Sensors for Human–Robot Walking Together. *Robotics* **2019**, *8*, 38. [\[CrossRef\]](#)
78. Wang, Y.; Lv, H.; Zhou, H.; Cao, Q.; Li, Z.; Yang, G. A Sensor Glove Based on Inertial Measurement Unit for Robot Teleoperation. In Proceedings of the IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 18–21 October 2020; pp. 3397–3402. [\[CrossRef\]](#)
79. Coronado, E.; Villalobos, J.; Bruno, B.; Mastrogiovanni, F. Gesture-based robot control: Design challenges and evaluation with humans. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 2761–2767. [\[CrossRef\]](#)
80. Zhang, X.; Zhao, Z.; Wang, Z.; Wang, X. Fault Detection and Identification Method for Quadcopter Based on Airframe Vibration Signals. *Sensors* **2021**, *21*, 581. [\[CrossRef\]](#) [\[PubMed\]](#)
81. Andò, B.; Baglio, S.; Lombardo, C.O. RESIMA: An Assistive Paradigm to Support Weak People in Indoor Environments. *IEEE Trans. Instrum. Meas.* **2014**, *63*, 2522–2528. [\[CrossRef\]](#)
82. Simoens, P.; Dragone, M.; Saffiotti, A. The Internet of Robotic Things: A review of concept, added value and applications. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1729881418759424. [\[CrossRef\]](#)