

## Article

# Exploiting the Internet Resources for Autonomous Robots in Agriculture

Luis Emmi <sup>1,\*</sup>, Roemi Fernández <sup>1</sup>, Pablo Gonzalez-de-Santos <sup>1</sup>, Matteo Francia <sup>2</sup>, Matteo Golfarelli <sup>2</sup>,  
Giuliano Vitali <sup>3</sup>, Hendrik Sandmann <sup>4</sup>, Michael Hustedt <sup>4</sup> and Merve Wollweber <sup>4</sup>

<sup>1</sup> Centre for Automation and Robotics (UPM-CSIC), 28500 Arganda del Rey, Madrid, Spain

<sup>2</sup> Department of Computer Science and Engineering (DISI), Alma Mater Studiorum-University of Bologna, 40127 Bologna, Italy

<sup>3</sup> Department of Agricultural and Food Sciences (DISTAL), Alma Mater Studiorum-University of Bologna, 40127 Bologna, Italy

<sup>4</sup> Laser Zentrum Hannover e.V., Hollerithallee 8, 30419 Hannover, Germany

\* Correspondence: luis.emmi@car.upm-csic.es

**Abstract:** Autonomous robots in the agri-food sector are increasing yearly, promoting the application of precision agriculture techniques. The same applies to online services and techniques implemented over the Internet, such as the Internet of Things (IoT) and cloud computing, which make big data, edge computing, and digital twins technologies possible. Developers of autonomous vehicles understand that autonomous robots for agriculture must take advantage of these techniques on the Internet to strengthen their usability. This integration can be achieved using different strategies, but existing tools can facilitate integration by providing benefits for developers and users. This study presents an architecture to integrate the different components of an autonomous robot that provides access to the cloud, taking advantage of the services provided regarding data storage, scalability, accessibility, data sharing, and data analytics. In addition, the study reveals the advantages of integrating new technologies into autonomous robots that can bring significant benefits to farmers. The architecture is based on the Robot Operating System (ROS), a collection of software applications for communication among subsystems, and FIWARE (Future Internet WARE), a framework of open-source components that accelerates the development of intelligent solutions. To validate and assess the proposed architecture, this study focuses on a specific example of an innovative weeding application with laser technology in agriculture. The robot controller is distributed into the robot hardware, which provides real-time functions, and the cloud, which provides access to online resources. Analyzing the resulting characteristics, such as transfer speed, latency, response and processing time, and response status based on requests, enabled positive assessment of the use of ROS and FIWARE for integrating autonomous robots and the Internet.

**Keywords:** precision agriculture; autonomous robots; artificial intelligence; IoT; cloud computing



**Citation:** Emmi, L.; Fernández, R.; Gonzalez-de-Santos, P.; Francia, M.; Golfarelli, M.; Vitali, G.; Sandmann, H.; Hustedt, M.; Wollweber, M. Exploiting the Internet Resources for Autonomous Robots in Agriculture. *Agriculture* **2023**, *13*, 1005. <https://doi.org/10.3390/agriculture13051005>

Academic Editors: Jin Yuan, Wei Ji, Qingchun Feng and Massimo Cecchini

Received: 16 March 2023

Revised: 17 April 2023

Accepted: 29 April 2023

Published: 2 May 2023



**Copyright:** © 2023 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

The year 2022 ended with more than 8 billion inhabitants of the world. Most governments understand that feeding this vast and growing population is one of the significant challenges they must face in the coming years. Some associations have predicted that food production will need to increase by 70% to feed the entire population in 2050 [1]. In developed countries, cultivated land is close to its maximum output; therefore, the solution is oriented toward optimizing the available resources. Many different cultural and technological methods for increasing crop yield are being used. Some improve crop yields, but at the extra cost of increasing environmental pollution and the carbon footprint. These side effects are unacceptable in many industrialized nations, such as those in the European Union, which is committed to using sustainable methods.

Precision agriculture leverages technologies to achieve those objectives and avoids undesired effects. PA is a concept for farm management founded on observation, measurement, and response to crop variability [2]. It assembles different methods to manage variations in a farm to enhance crop yield, improve commercial profit, and guarantee environmental sustainability. PA uses current information and communication technologies (ICT), automation, and robotics to monitor crop growth, predict the weather accurately, perform optimal irrigation, apply fertilizers smartly, manage weeds and pests accurately, test soil quality precisely, etc.

Since the late 1980s, precision agriculture techniques have been introduced step by step in the agricultural production sector, integrating the following:

- Sensors to acquire geolocated biodata of crops and soil, e.g., nitrogen sensors, vision cameras, global navigation satellite systems (GNSS), etc.
- Computers for analyzing those data and running simple algorithms to help farmers make simple decisions (applying or not applying a given process, modifying a process application map, etc.).
- Actuators in charge of executing the decisions (opening/closing valves, altering a trajectory, etc.) for modifying crops. As an actuator, we consider the agricultural tool, also called the agricultural implement, and the vehicle, manually or automatically driven, to move the tool throughout the working field and apply the farming process.

The integration of subsystems onboard robotic vehicles started in the late 1990s. Some illustrative examples, based on retrofitting conventional vehicles, are the autonomous agricultural sprayer [3], which focuses on achieving a pesticide spraying system that is cheap, safe, and friendly to the environment, and the autonomous orchard vehicles for mowing, tree pruning, and training, spraying, blossoming, and fruit thinning, fruit harvesting, and sensing [4], both deployed in the USA. In Europe, we can find the RHEA fleet (see Figure 1a), consisting of a fleet of three tractors that cooperate and collaborate in the application of pesticides [5]. Regarding robotic systems based on specific structures designed for agriculture (see Figure 1b), we can remark on LadyBird in Australia, intended for the valuation of crops using thermal and infrared detecting systems, hyperspectral cameras, stereovision cameras, LIDAR, and GPS [6], and Vibro Crop Robotti in Europe, built for accurate seeding and mechanical row crop cleaning [7]. These robots were integrated around computing systems based on centralized or elementary distributed architectures to handle a few sensors and control unsophisticated agricultural tools.

In addition to those developments, related technologies have evolved drastically in recent years, and now sensors can be spread throughout the field and communicate with each other. This is possible because of the Internet of Things (IoT). This computing concept describes how to cluster and interconnect objects and devices through the Internet, where all are visible and can interact with each other. IoT defines physical objects with devices (mainly sensors) and includes processing power, software applications, and other technologies to exchange data with other objects through the Internet.

Moreover, computers can run artificial intelligence (AI) algorithms, considering AI as the ability of a machine (computer) to emulate intelligent human actions. The application of AI to agriculture has been focused on three primary AI techniques: expert systems, artificial neural networks, and fuzzy systems, with significant results in the management of crops, pests, diseases, and weeds, as well as the monitoring of agricultural production, store control, and yield prediction, for example [8].

AI techniques are also applied to provide vehicles with autonomy; therefore, autonomous agricultural robots leverage this technology. AI-based vision systems can fulfill the following roles:

- Detecting static or dynamic objects in their surroundings.
- Detecting row crops for steering purposes.
- Identifying plants and locating their positions for weeding are clear examples of the current use of AI techniques in agricultural robotics [9].



(a)



(b)

**Figure 1.** (a) Agricultural robots based on retrofitted conventional vehicles (RHEA fleet); (b) agricultural robots designed on purpose (Courtesy of AgreenCulture SaS).

Another technology that has evolved in the last decade is cloud computing, defined as the on-demand delivery of computing services, mainly data storage and computing power, including servers, storage, databases, networking, software applications, artificial intelligence methods, and analytics algorithms over the Internet. The main objective of cloud computing systems is to provide flexible resources at adapted prices. A cloud computing system allows the integration of data of different types, loaded from many sources in batch and real-time. In particular, the integration can be based on georeferenced data in the precision farming area. Data can range from trajectory data to images and videos related to fields and missions and any sensors installed on the autonomous robot.

Cloud computing allows the use of services available in the cloud (computing, storing, etc.), with increasing advantages provided by big data techniques. Many agricultural applications of big data technologies have already been introduced in agriculture [10] and should be present in future robotic systems.

This article presents an architecture to integrate new technologies and Internet trends in agricultural autonomous robotic systems and has two main objectives. The first objective is to provide an example of designing control architectures to connect autonomous robots to the cloud. It is oriented toward robot designers and gives significant technical details. The second objective is to disclose to farmers the advantages of integrating the new technologies in autonomous robots that can provide farmers with significant advantages regarding (i) data storage, which is a secure and efficient way to store, but also access and share, data, eliminating the need of physical storage and, thus, reducing the risk of data loss; (ii) scalability, which allow the farmers to expand or reduce their storage needs, efficiently optimizing their resources, and (iii) analytics services, which allow a farmer to analyze their own data to make informed decisions taking advantage of the AI tools available on the cloud. These are general advantages of using the cloud, but autonomous robots have great potential for collecting data and must facilitate communicating those data to the cloud.

To base the architecture on a specific example, the integration of a laser-based system for weed management is considered. Thus, Section 2 presents the material, defining the robot's components, and the methodology, detailing the system's architecture. Section 3 then introduces the experiments to be assessed and discussed in Section 4. Finally, Section 5 summarizes the conclusions.

## 2. Materials and Methods

This section first describes the components and equipment integrated for building the autonomous robot used to validate and assess the proposed integration methodology. Second, the methods for the integration of components are detailed.

### 2.1. System Components

#### 2.1.1. Main Process Loop in PA Autonomous Robots

The autonomous systems used for precision agriculture generally follow the structure of an automatic control loop that consists of the following (see Figure 2):

- Selecting the references for the magnitudes to be controlled, i.e., defining the desired plan.
- Measuring the magnitudes of interest.
- Making decisions based on the measured and desired values of the magnitudes (control strategy).
- Executing the decided actions

In our application, the selecting references are made with the smart navigation manager (mission planner), the measures of the magnitudes of interest are performed with the perception system and the IoT sensor network, the decisions are made with the smart navigation manager (smart operation manager), and the actions are executed with the agricultural tool and the autonomous robot that move the implement throughout the mission field. In addition, our system also takes care of the interaction with the cloud and the operator. In our proposed integration method, these components are grouped into modules, as illustrated in Figures 2 and 3. These modules are as follows.

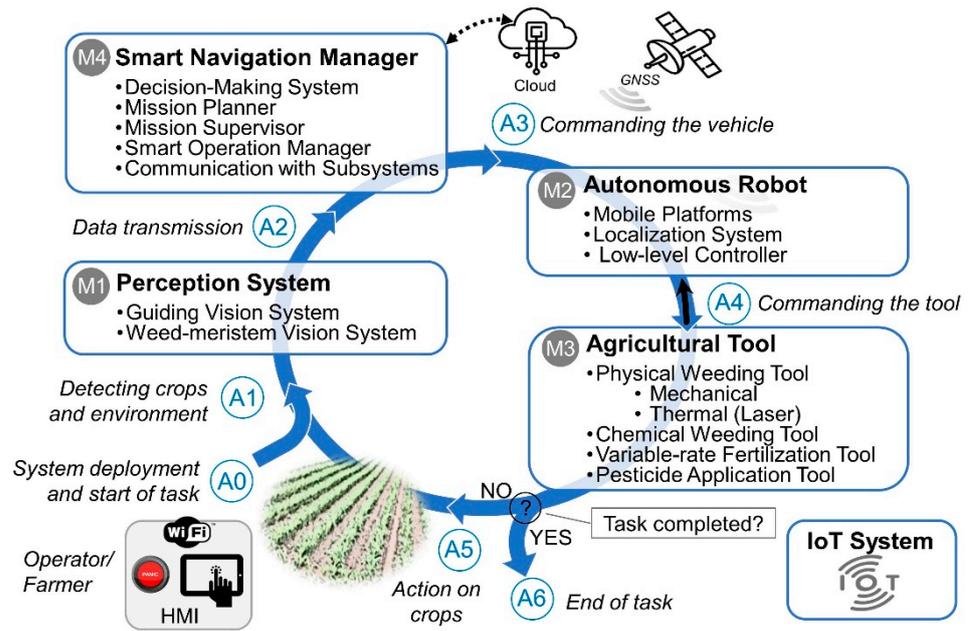


Figure 2. Components of a precision agriculture robotic system and main information flow.

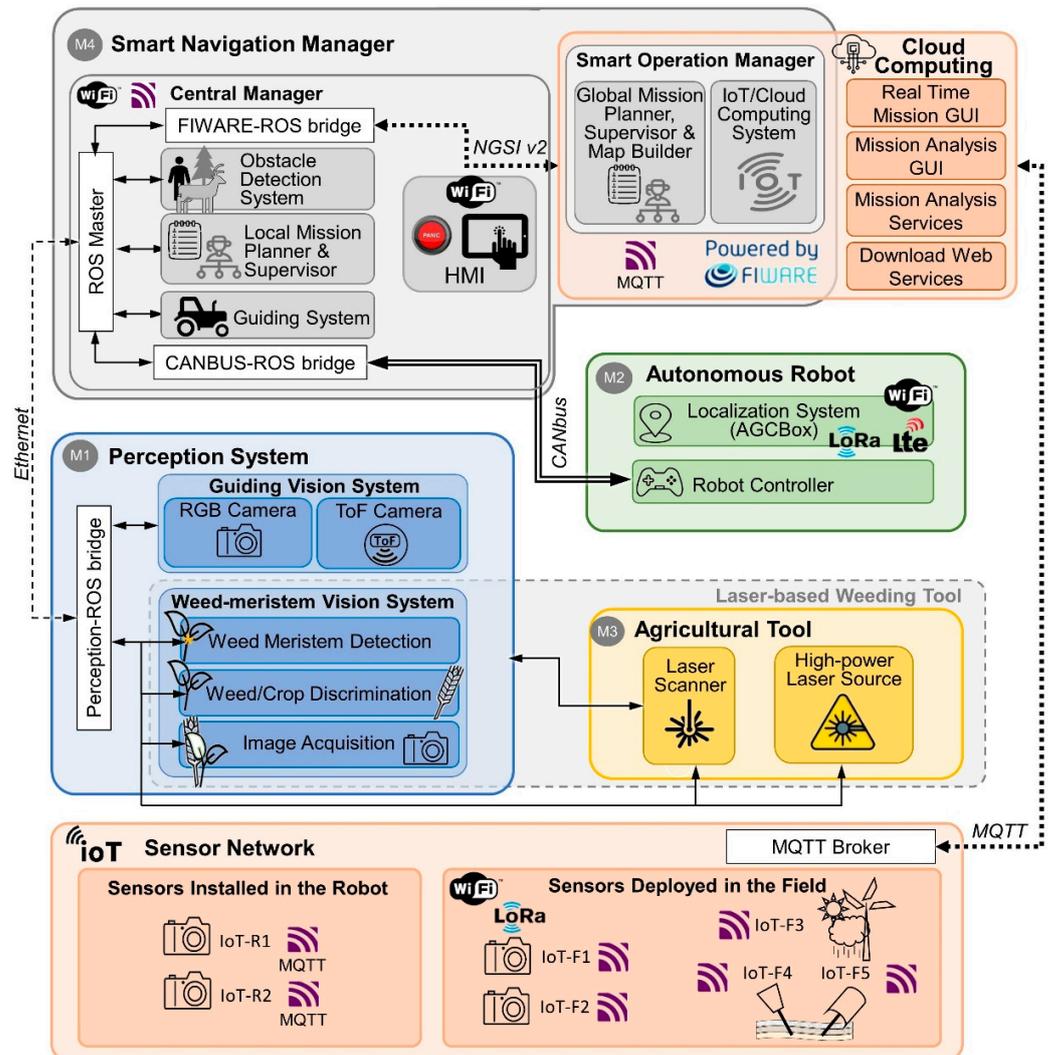


Figure 3. Computing architecture.

### 2.1.2. Agricultural Robot

A manually driven or autonomous vehicle is essential in agricultural tasks to perform the necessary actions throughout the working field. In this case, we use a compact mobile platform based on a commercial vehicle manufactured by AgreenCulture SaS, France. This is a tracked platform, and, thus, it operates as a skid-steer mechanism. The track distance can be adapted to the crop row space. Equipped with an engine or batteries, the platform can follow predefined trajectories at 6 km/h with a position accuracy of  $\pm 0.015$  m using a global positioning system (GPS) based on the real-time kinematic (RTK) technique. This mobile platform is illustrated in Figure 4a.



(a)



(b)

**Figure 4.** (a) Mobile platform (AgreenCulture SaS) and (b) autonomous laser weeding system.

### 2.1.3. Perception System

A perception system is based on computer vision algorithms that obtain, process, analyze, and understand images and data from the environment. With these inputs, the system produces numerical and symbolic information for making decisions. The perception system for this study consists of the following systems:

- Guiding vision system: This system aims to detect static and dynamic obstacles in the robot's path to prevent the robot tracks from stepping on the crops during the robot's motion. Furthermore, it is also used to detect crop rows in their early growth stage to guide the robot in GNSS-denied areas [8]. The selected perception system consisted of a red–green–blue (RGB) wavelength vision camera and a time-of-flight (ToF) camera attached to the front of the mobile platform using a pan-tilt device, which allows control of the camera angle with respect to the longitudinal axis of the mobile platform,  $x$ . Figure 4 illustrates both cameras and their locations onboard the robot.
- Weed–meristem vision system: The system is based on 3D vision cameras to provide the controller with data on crops and weeds. These data are used to carry out the main activity of the tool for which it has been designed: weed management, in this case. For example, the perception system used in this study consists of an AI vision system capable of photographing the ground and discriminating crops from weeds in a first step using deep learning algorithms. In the second step, the meristems of the detected weeds are identified. Figure 3 sketches this procedure.

#### 2.1.4. Agricultural Tools

Agricultural tools focus on direct action on the crop and soil and rely on physical (mechanical, thermal, etc.) or chemical (pesticides, fertilizers, etc.) foundations. This study used a thermal weeding tool based on a high-power laser source that provided lethal laser doses to be deployed on the weed meristems using scanners.

An AI video system provided the positions of the weed meristems. Indeed, this specific solution physically integrated the AI vision system, the laser scanner, and the high-power laser source into the laser-based weeding tool component. The video frames acquired with this system were sent to the central controller at a rate of 4 frames/s. After the mission, all stored images were sent to the cloud.

#### 2.1.5. The Smart Navigation Manager (SNM)

This manager is a distributed software application responsible for driving the autonomous robot and coordinating all other modules and systems. The SNM is split into (i) the smart operation manager and (ii) the central manager, which also includes the human–machine interface (HMI).

##### Smart Operation Manager (SoM)

The smart operation manager is a human–computer interaction module that can acquire, process, and deliver information based on computer algorithms and is devoted to assisting farmers in making accurate, evidence-based decisions. The SoM is specialized for laser weeding technology, the tool selected for this study.

Data management is performed through the Internet using FIWARE. Data access control is provided via a virtual private network (VPN) to secure data transfer to/from the cloud. The visual dashboard will also be available on the HMI for field operations. Through the dashboard, the operator will also interact with the robot.

The smart operation manager is allocated in the cloud. It contains the global mission planner and supervisor, the map builder, and the module for managing the IoT and cloud computing system (see Figures 3 and 5). The hardware of the SoM relies on a cluster of 10 servers.

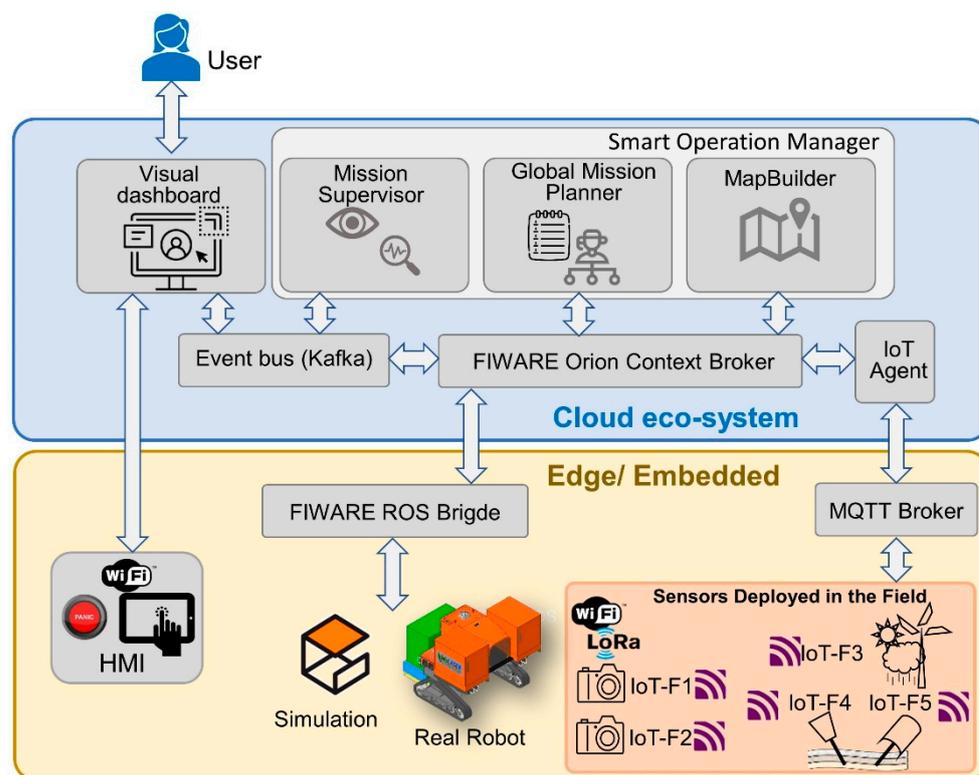


Figure 5. Cloud computing modules/containers.

(a) Global Mission Planner

A planner is a software tool responsible for computing the trajectories of the vehicle and an a priori known treatment map. The planner obtains some types of information from the Internet, including the following:

- Map information according to the data models on the Internet;
- Other information provided by third parties, such as weather forecasts;
- Data models to create maps for accessing already known treatment maps (sets of points in the field) which commonly originate from third-party map descriptions (Google Earth; Geographic Information System (GIS); GeoJSON.io, an open standard format to represent geographical features with nonspatial qualities).

Regarding robot location, two types of systems are envisaged, as follows:

- Absolute location based on GNSS: GNSS integrates several controllers for line tracking and is based on Dubins paths [11];
  - Relative location based on RGB and ToF cameras, LIDAR, and IoT sensors: These methods are based on different techniques for navigation in the field and navigation on the farm, such as hybrid topological maps, semantic localization and mapping, and identification/detection of natural and artificial elements (crops, trees, people, vehicles, etc.) through machine learning techniques.

(b) Global Mission Supervisor

A supervisor is a computational tool responsible for overseeing and monitoring the execution of the mission plan while helping the farmer (operator) manage potential failures. Most supervisor systems are designed around two actions: fault detection and fault diagnosis. The supervisor executes the following actions:

- Receiving alarms from the system components (vehicle, sensors, weeding tool, etc.).
- Detecting faults in real-time.
- Executing diagnosis protocols.

- Collecting all available geo-referred data generated by every module onboard the robot. The data are stored in both the robot and the cloud.

(c) Map Builder

A map builder is an application used to convert maps based on GeoJSON into FIWARE entities. Its main function is to support farmers in using the robotic system in a simple, reliable, and robust way by giving the robot enough information a priori (e.g., farm schema and boundaries, field locations and shapes, crop types, and status). This module takes advantage of the data models created by the FIWARE community to represent the farm and other environments digitally, where they have been conditioned to be adapted to robotic systems and especially oriented to navigation [12]. The design of the Map Builder allows the user to accomplish the following:

- Select the field in GeoJSON.IO, an open-source geographic mapping tool that allows maps and geospatial data to be created, visualized, and shared in a simple and multiformat way.
- Assign essential attributes to comply with FIWARE. These attributes are those based on the farmer's knowledge. They can include static (i.e., location, type, category) and dynamic (i.e., crop type and status, seeding date, etc.) attributes.
- Export in \* GeoJSON format. The map obtained will be imported for extracting the information required to fill in the FIWARE templates, which include the farms and parcel data models, and other elements in a farm, such as buildings and roads.

This conversion makes it easier to connect the robot to the cloud by standardizing data. These data, after processing, constitute a source for the design of processes with the robot, and its storage and subsequent analysis can provide forecasts of future events in the field or behavior of the robot.

(d) IoT System

This study integrates an IoT sensor network to collect data from the following:

- The autonomous vehicle: The data and images acquired with IoT sensors onboard the vehicle are used to monitor and evaluate performances and efficiency and to identify the effects of treatments and traffic on surfaces.
- The environment: Data acquired with IoT sensors deployed on the cropland are used to (i) monitor crop development and (ii) collect weather and soil information.

Two IoT sets of devices are used in our study, as follows:

- Robot-IoT set: It consists of two WiFi high-definition cameras installed onboard the autonomous robot (IoT-R1 and IoT-R2 in Figure 3). The cameras are triggered from the cloud or the central controller to obtain a low frame rate (approximately 1/5 sec). The pictures are stored in the cloud and are used to monitor the effects of the passage of the autonomous vehicle; therefore, they should include the robot's tracks.
- Field-IoT set: It consists of the following (see Figure 3):
  - Two multispectral cameras (IoT-F1 and IoT-F2) placed at the boundary of cropped areas to obtain hourly pictures of crops.
  - A weather station (IoT-F3) to measure precipitation, air temperature (Ta), relative humidity (RH), radiation, and wind.
  - Three soil multi-depth probes (IoT-F4) for acquiring moisture (Ts) data and three respiration probes (IoT-F5) to measure CO<sub>2</sub> and H<sub>2</sub>O.

Every one of these components or nodes exchanges messages with the Message Queuing Telemetry Transport (MQTT) protocol, carrying JavaScript Object Notation (JSON) serialized information from node sensors/cameras interpreted as the entity. While metering nodes (weather, soil probe, and respirometer) communicate by MQTT messages, camera nodes have to transmit images (maximum of 100 pictures/day for periodic snapshots of the area or alarms), and the use of FTP made a wide-band networking solution, such as WiFi, mandatory instead of narrowband solutions.

### (e) Cloud Computing System

This study sets up a cloud-based data platform, which is an ecosystem that incorporates data acquired in the field. The data platform supports end-to-end data needs, such as ingestion, processing, and storage, to provide the following:

- A data lake repository for storing mission data to be downloaded in batches for post-mission analysis.
- A web interface for post-mission data analysis based on graphical dashboards, georeferenced visualizations, key performance indicators, and indices.
- A container framework for implementing “Decision Support System” functionalities that define missions to be sent to the robot. These functionalities (e.g., the mission planner) can be implemented and launched from the cloud platform.
- A soft real-time web interface for missions. The interface visualizes real-time robot activities and performances or sends high-level commands to the robot (e.g., start, stop, change mission).

These functionalities are ordered based on the strictness of real-time constraints.

The cloud-computing platform is based on the Hadoop stack and is powered by FIWARE. We adopted an open-source solution with well-known components that can be imported into different cloud service providers if no on-premises hardware is available. The core component of the platform is the (FIWARE) Orion Context Broker (OCB) from Telefonica [13], a publish/subscribe context broker that also provides an interface to query contextual information (e.g., obtain all images from the cameras in a specific farm), update context information (e.g., update the images), and be notified when the context is updated (e.g., when a new image is added into the platform). The images and raw data are stored in the HDFS (Hadoop distributed file system), while the NoSQL (not only structured query language) MongoDB database is used to collect the contextual data from FIWARE and further metadata necessary to manage the platform [14]. Additionally, we use Apache KAFKA, an open-source distributed event bus, to distribute context updates from FIWARE to all the modules/containers hosted on the cloud platform. The different cloud computing modules/containers used in this study are illustrated in Figure 5.

### Central Manager

This central manager is an application that is divided into the following:

- Obstacle detection system. This module acquires visual information from the front of the robot (robot vision system) to detect obstacles based on machine vision techniques.
- Local mission planner and supervisor. The planner plans the motion of the robot near its surroundings. The local mission supervisor oversees the execution of the mission and reports malfunctions to the operator (see Section 2.1.5).
- Guidance system. This system is responsible for steering the mobile platform to follow the trajectory calculated by the planner. It is based on the GNSS if its signal is available. Otherwise, the system uses the information from the robot vision system to extract the crop row positions and follow them without harming the crop.
- Human–machine interface

A human–machine interface (HMI) is a device or program enabling a user to communicate with another device, system, or machine. In this study, a HMI using portable devices (android tablets) is addressed to allow farmers to perform the following:

- Supervise the mission.
- Monitor and control the progress of agricultural tasks.
- Identify and solve operational problems.
- Obtain real-time in-field access in an ergonomic, easy-to-use, and robust way.
- Maintain the real-time safety of the entire system.

To achieve these characteristics, a graphic device was integrated with the portable/remote controller of the mobile platform. This controller provides manual and remote vehicle control and integrates an emergency button.

### 2.1.6. Sequence of Actions

The relationships among these components and modules and the information flow are illustrated in Figures 2 and 3. The process is a repeated sequence of actions (A0 to A6), defined as follows:

- A0 The system is installed in the field, The operator/farmer defines or selects a previously described mission using the HMI and starts the mission.
- A1 The sensors of the perception module (M1) installed onboard the autonomous robot (M2) extract features from the crops, soil, and environment in the area of interest in front of the robot.
- A2 The data acquired in action A1 are sent to the smart operation manager, determining the consequent instructions for the robots and the agricultural tool.
- A3 The required robot motions and agricultural tool actions are sent to the robot controller, which generates the signal to move the robot to the desired positions.
- A4 The robot controller forwards the commands sent by the smart navigation manager or generates the pertinent signals for the agricultural tool to carry out the treatment.
- A5 The treatment is applied, and the procedure is repeated from action A1 to action A5 until field completion (A6).
- A6 End of mission.

## 2.2. Integration Methods

Integrating all of the components defined in the previous section to configure an autonomous robot depends on the nature of the applications the robot is devoted to and the connections and communication among the different components that must be precisely defined. Thus, this section first describes the computing architecture of the controller, which integrates the different subsystems and modules. Second, the interfaces between subsystems are precisely defined. Finally, the operation procedure is defined.

### 2.2.1. Computing Architecture

A distributed architecture based on an open-source Robot Operating System (ROS) is proposed to integrate the system's main components onboard the mobile platform in this study. ROS is the operating system most widely accepted by software developers to create robotics applications. It consists of a set of software libraries and tools that include drivers and advanced algorithms to help developers build robot applications [15].

In this study, ROS, installed in the central controller, is used as a meta-operating system for the testing prototype. The necessary interfaces (bridges) are developed to establish communication with the autonomous vehicle, the perception system, and the laser-based weeding tool. Because of ROS versatility and its publisher/subscriber communication model, it is possible to adapt the messages to protocols commonly used in IoT, such as Message Queuing Telemetry Transport (MQTT).

ROS supports software developers in creating robotics functionalities to monitor and control robot components connected to a local network. However, this solution is not extendible to a wider network, such as the Internet. Fortunately, there exist some ROS modules that solve the problem. One is ROSLink, a protocol for extensions defining an asynchronous communication procedure between the users and the robots through the cloud [16]. ROSLink performance has been shown to be efficient and reliable, and it is widely accepted by the robotics software community [17]. Although ROSLink has been widely used to connect robotic systems with the cloud, it is oriented toward transmitting low-level messages. There is no convention to define standard data models that allow intelligent robotics systems to be scalable.

One alternative to a more internet-oriented communication framework is FIWARE, which offers interaction with the cloud using cloud services that provide well-known benefits, such as (a) cost and flexibility, (b) scalability, (c) mobility, and (d) disaster recovery [18]. FIWARE is an open software curated platform fostered by the European Commission and the European Information and Communication Technology (ICT) industry for the

development and worldwide deployment of Future Internet applications. It attempts to provide a completely open, public, and free architecture and a collection of specifications that allows organizations (designers, service providers, businesses, etc.) to develop open and innovative applications and services on the Internet that fulfill their needs [19].

In this study, a cloud-based communication architecture has been implemented using FIWARE as the core, which allows messages between the edge and the cloud to be transferred and stored. The selection was made because this is an open-source platform that provides free development modules and has many enablers already developing and integrating solutions for smart agriculture.

In addition to FIWARE, we use KAFKA, a robust distributed framework for streaming data (see Section 2.1.5) that allows producers to send data and for consumers to subscribe to and process such updates. KAFKA enables the processing of streams of events/messages in a scalable and fault-tolerant manner, and decouples producers and consumers (i.e., a consumer can process data even after a producer has gone offline). For historic data, HDFS allows the download of batches of data at any time and replicates each data in three copies to prevent data loss.

The visual dashboard will also be available on the HMI for the field operations. Through the dashboard, the operator will also interact with the robot. FIWARE smart data models do not suffice to represent our application domain or to integrate the agricultural and robotic domains; therefore, we have extended the existing models and updated some existing entities. Since smart data models from FIWARE are overlapping and sometimes inconsistent, we had to envision a unified model to integrate and reconcile the data. To connect the robotic system with the cloud, specific data models were developed to represent the different robotic elements, following the guidelines of FIWARE and its intelligent data models [12].

The IoT devices deployed in the field must be able to establish connections through WiFi and LoRa technologies. WiFi is a family of wireless network protocols. These protocols are generally used for Internet access and communication in local area networks, allowing nearby electronic devices to exchange data using radio waves. LoRa technology is a wireless protocol designed for long-range connectivity and low-power communications and is primarily targeted for the Internet of Things (IoT) and M2M networks. LoRa tolerates noise, multipath signals, and the Doppler effect. The cost of achieving this is a very low bandwidth compared to other wireless technologies. This study uses a 4G LTE-M modem to connect to the Internet.

At a lower level of communication, CANbus or ISOBUS is generally used to control and monitor the autonomous vehicle. This study uses CANbus and its communication protocol CANopen. Autonomous vehicles and agricultural tools typically contain their own safety controllers. The first behaves as a master and, in the case of a risky situation, it commands the tool to stop.

The human-machine interface (HMI) will include a synchronous remote procedure call-style communication over the services protocol and asynchronous communications to ensure the robot's safety. In addition to these ROS-based protocols, the HMI has a safety control connected to the low-level safety system (by radiofrequency) for emergency stops and manual control.

Figure 6 illustrates the overall architecture, indicating the following:

- The modules (Mi), presented in the previous sections.
- The interconnection between modules, presented in the next section.
- The communication technologies and protocols to configure agricultural robotic systems that integrate IoT and cloud computing technologies.

The main characteristics of this architecture are summarized in Table 1.



(a) Field areas



(b) Field distribution

**Figure 6.** Experimental fields.

**Table 1.** Architecture components.

Architecture Component	Solutions/Comments
Operating system	ROS (Robot Operating System)
IoT-controller bridge	Hypertext Transfer Protocol (HTTP) to FIWARE Note: FIWARE is used as a communication protocol in the cloud; therefore, it is not necessary to use ROSLink.
ROS-based system for FIWARE tools	HTTP protocol to FIWARE Note: FIROS has several disadvantages when developing new data models to represent the robot, so a particular enabler will not be used to establish communication between the robot and the cloud.
Communication with IoT devices	WiFi, serial communication Note: Since a certain amount of data needs to be transmitted, WiFi would suffice.
The Internet	4G LTE-M modem
Devices onboard the mobile platform	CANopen, serial
Human-machine interface (HMI).	Synchronous remote procedure call-style communication over services protocol. Asynchronous communications to ensure the safety of the robot. Note: The HMI is used to provide access to SoM services through a web interface.

### 2.2.2. Interfaces between System Components

This architecture considers four main interfaces between systems and modules, as follows:

#### Smart Navigation Manager (M4)/Perception System (M1) interface

To receive the raw information from the perception system (sensors, cameras, etc.), the central manager uses direct connections via the transmission control protocol/Internet protocol (TCP/IP) for sensors and the universal serial bus (USB) for RGB and ToF cameras. All IoT devices use the available wireless communication technologies (WiFi and LoRa) to access the Internet and the cloud.

To guide the robot, the obstacle detection system obtains data from the guiding vision system (RGB and ToF cameras) through the Ethernet that communicates the central manager with the perception system. This communication is stated using the ROS manager and the perception-ROS bridge (see Figure 3).

#### Smart Navigation Manager (M4)/Agricultural Tool (M3) interface

These systems can communicate through ROS messaging protocols, where the publisher/subscriber pattern is preferred. This interface exchanges simple test messages to verify the communication interface.

It is worth mentioning that the perception system and the agricultural tool are connected directly in some specific applications. This solution decreases the latency of data communication but demands moving a portion of the decision algorithms from the smart navigation manager to the tool controller; therefore, the tool must exhibit computational features. This scheme is used in the weeding system to test the proposed architecture.

#### Smart Navigation Manager (M4)/Autonomous Robot (M2) interface

Initially, these systems communicate via CANbus with the CANopen protocol. The central manager uses this protocol to receive information on the status of the autonomous vehicle and basic information from the onboard sensors (GNSS, IMU, safety system, etc.). A CANbus-ROS bridge is used to adapt the communication protocols.

#### Autonomous Robot (M2)/Agricultural Tool (M3) interface

Usually, it is not necessary for the vehicle to directly communicate with the tool because the smart navigation manager coordinates them. However, as autonomous vehicles and agricultural tools usually have safety controllers, there is wired communication between the two safety controllers. In such a case, the autonomous vehicle safety controller works as a master and commands the tool safety controller to stop the tool if a dangerous situation appears.

#### Perception System (M1)/Agricultural Tool (M3)

This communication is required to inform the agricultural tools about the crop status. In weeding applications, the information is related to the positions of the weeds. In this specific application, the perception system (weed meristem detection module) sends the weed meristem positions to the laser scanner module of the agricultural tool. This communication is carried out using a conventional Ethernet connection. The metadata generated via the detection system are made available in the existing ROS network and sent to the smart navigation manager.

#### Smart Navigation Manager internal/cloud communications

The smart navigation manager is a distributed system that consists of three main modules:

- The central manager running on the central controller.
- The smart operation manager running on the cloud.
- The HMI running in a portable device.

The central manager and the smart operation manager communicate via NGSi v2, a FIWARE application programming interface, using a FIWARE–ROS bridge to adapt ROS protocols to NGSi v2 messages. In contrast, the HMI communicates with the central manager via WiFi and Internet, directly accessing the web services hosted in the cloud. The HMI exhibits a panic button connected via radiofrequency to the safety systems of the autonomous robot and the agricultural tool.

#### IoT system/Cloud

There is a direct link from the IoT system to the cloud using MQTT.

#### 2.2.2.8. Operation Procedure

To use the proposed architecture and method, the user must follow the method below.

- Creating the map: The user creates the field map following the procedure described in the MapBuilder module (see Section 2.1.5).
- Creating the mission: The user creates the mission by selecting the mission's initial point (home garage) and destination field (study site).
- Sending the mission: The user selects the mission to be executed with the HMI (all defined missions are stored in the system) and sends it to the robot using the cloud services (see Section Smart Operation Manager (SoM)).
- Executing the mission: The mission is executed autonomously following the sequence of actions described in Section 2.1.6. The user does not need to act except for when alarms or collision situations are detected and warned of by the robot.
- Applying the treatment: When the robot reaches the crop field during the mission, it sends a command to activate the weeding tool, which works autonomously. The tool is deactivated when the robot performs the turns at the headland of the field and is started again when it re-enters. The implement was designed to work with its own sensory and control systems, only requiring the mobile platform for mobility and information when it must be activated/deactivated.
- Supervising the mission: When the robotic system reaches the crop field, it also sends a command to the IoT sensors, warning that the treatment is in progress. Throughout the operation, the mission supervisor module analyzes all the information collected by the cloud computing system, generated by both the robotic system and the IoT sensors. It evaluates if there is a possible deviation from the trajectory or risk of failure.
- Ending the mission: The mission ends when the robot reaches the last point in the field map computed by the MapBuilder. Optionally, the robot can stay in the field or return to the home garage. During the mission execution, the user can stop, resume, and abort the mission through the HMI.

### 3. Experimental Assessment

This section states the characteristics of the described autonomous robot with IoT and cloud computing connectivity. To achieve this purpose, the experimental field for this study is first described. Then, a test mission is defined to acquire data from the different subsystems. Finally, the system characteristics are analyzed and assessed.

The characteristics obtained are not compared with similar robotic systems due to the lack of such information in the literature. There are no published results in weeding applications; therefore, it is difficult to compare, and the indicators have been geared towards general cloud computing and mobile robotics characteristics. Therefore, cross-validation has been carried out, comparing the features of the autonomous robot with the general performance of the robot and cloud communication. Productivity, cost, and other indicators of the presented architecture are those of the general use of cloud computing.

#### 3.1. Study Site

The system developed for this study was tested in an experimental field located in Madrid, Spain ( $40^{\circ}18'45.166''$ ,  $-3^{\circ}28'51.096''$ ). The climate of the study site is classified as

a hot summer Mediterranean climate with an average annual temperature of 14.3 °C and precipitation of 473 mm.

The experimental field consisted of two areas of 60 × 20 m<sup>2</sup> that grew wheat (*Triticum aestivum* L.), with crop rows at a distance of 0.10 m, and maize (*Zea mays* L.), with crop rows at a distance of 0.50 m, respectively. Each area was divided into three sections of 20 × 20 m<sup>2</sup>. The sections in one area were seeded in consecutive weeks, allowing us to conduct experiments in three-week windows. Figure 6 shows the experimental field and the distribution of the areas and sections.

### 3.2. Description of the Test Mission

Tests were conducted to assess the performance and quality of integrating new technologies in autonomous robots for agriculture. First, the testing prototype was integrated with the components introduced in Section 2; then, several IoT devices were disseminated throughout the field (RGB and multispectral cameras, weather stations, soil probes, etc.); finally, a mission was defined to acquire data in the study site to perform quantitative analyses. The mission consisted of covering sections of 20 × 20 m<sup>2</sup> with wheat and maize crops while the following occurred:

- Acquiring data from the IoT sensor network.
- Taking pictures of the crop.
- Acquiring data from the guidance system.
- Sending all the acquired information to the cloud.

The mission proposed by the planner is illustrated in Figure 7. The robot tracked the path autonomously, and the following procedures were carried out.



**Figure 7.** Robot's path from the home garage to the study site. The planner provides the mission for covering the study site.

#### Perception system procedure

- Guiding vision system: This experiment was conducted in the treatment stage, where the crop was detected to adjust the errors derived from planning and the lack of precision of the maps. YOLOv4 [20], a real-time object detector based on a one-stage

object detection network, was the base model for detecting early-stage growth in maize [8], a wide-row crop. The model was trained using a dataset acquired in an agricultural season before these tests using the same camera system [21]. Moreover, in the case of wheat, which is a narrow-row crop, a different methodology was applied through the use of segmentation models, such as MobileNet, a convolutional neural network for mobile vision applications [22], trained using a dataset acquired in an agricultural season before these tests [23], with the same camera system. The detection of both crops was evaluated with regard to the GNSS positions collected manually for the different crop lines.

The maize and wheat datasets were built with 450 and 125 labeled images, respectively. Data augmentation techniques (rotating, blurring, image cropping, and brightness changes) were used to increase the size of the datasets. For both crops, 80% of the data was destined for training, 10% for validation, and 10% for testing.

- The AI vision system: This system uses data from the installed RGB cameras to enable robust automated plant detection and discrimination. For this purpose, the state-of-the-art object detection algorithm YOLOv7 is used in combination with the Nvidia framework DeepStream. Tracking the detected plants is performed in parallel by a pretrained DeepSort algorithm [24]. The reliability of the object detection algorithm is evaluated using test datasets with the commonly used metrics “intersection over union” (IoU) and “mean average precision” (mAP). This system works cooperatively with laser scanners as a stand-alone system. The information is not stored in the cloud.

The dataset used for training weed/crop discrimination was generated in fields in several European countries. It contains 4000 images, 1000 of which are fully labeled. Distinctions are made according to the processing steps to be applied: weeds, grasses, and crops. In addition, the dataset was expanded to three times its original size through augmentation measures. As well as generating new training data, this enables robustness against changing environmental influences, such as changing color representation, motion blur, and camera distortion. The YOLOv7 network achieved a mean average precision (mAP) of 0.891 after 300 epochs of training. The dataset was divided into 80%, 10%, and 10% for training, validation, and testing subsets, respectively.

#### Autonomous robot procedure

The navigation controller: Given a set of trajectories based on RTK-GNSS, the performance of the guidance controller was evaluated by measuring lateral and angular error through the incorporation of colored tapes on the ground and using the onboard RGB camera and ToF to extract the tape positions to compute the errors concerning the robot’s pace.

#### Smart Navigation Manager procedure:

- Smart operation manager: The processing time, latency, success rate, response time, and response status based on requests of the mission planner, IoT sensors, and cloud computing services were evaluated using ROS functionalities that provide statistics related to the following:
  - The period of messages by all publishers.
  - The age of messages.
  - The number of dropped messages.
  - Traffic volume to be measured in real-time.
- Central manager: The evaluation is similar to that used for the navigation controller.
- Obstacle detection system: YOLOv4 and a model already developed based on the COCO database were introduced to detect common obstacles in agricultural environments and were also used for evaluation. YOLOv4 is a one-stage object detection model, and COCO (common objects in context) is a large-scale object detection, segmentation, and captioning dataset.

#### 4. System Assessment and Discussion

The mission described in the previous section produced crop images, sensor data, and traffic information with the following characteristics:

- Crop images: During the robot's motion, images are acquired at a rate of 4 frames/s to guide the robot. The RGB images are  $2048 \times 1536$  pixels with a weight of 2.2 MB (see Figures 8 and 9), and the ToF images feature  $352 \times 264$  points (range of 300–5000 mm) (see Figure 10). The images are sent to the guiding and obstacle detection system through the Ethernet using ROS (perception-ROS bridge in the perception system and ROS manager in the central manager). A subset of these images is stored in the cloud for further analysis. Using a FIWARE-ROS bridge with the NGSI application programming interface, the system sends up to 4 frames/s.
- Sensor data: IoT devices send the acquired data using 2.4 GHz WiFi with the MQTT protocol and JSON format.
- Traffic information: The ROS functionalities mentioned above revealed that during a field experiment (10 min duration), the total number of delivered messages was 2,395,692, with a rate of only 0.63% dropped messages (messages that were dropped due to not having been processed before their respective timeout), with average traffic of 10 MB/s and maximum traffic of 160 MB at any instant of time. No critical messages (command messages) were lost, demonstrating robustness within the smart navigation manager. Regarding cloud traffic, during a period of time of approximately 3 h, the messages sent to the cloud were monitored, where the number of messages received by the cloud was measured; the delay time of the transmission of the messages between the robot (edge) and the OCB, and between the robot and the KAFKA bus (see Figure 3), were also measured. During this interval of time, around 4 missions were executed, and a total of 14,368 messages were sent to the cloud, mainly the robot status and the perception system data. An average delay of about 250 ms was calculated between the moment the message is sent from the robot and the moment it is received in the OCB (see Figure 11a). Moreover, the KAFKA overhead, i.e., the time it takes for a message received by the OCB to be forwarded to the KAFKA bus and eventually processed by a KAFKA consumer, was approximately 1.24 ms, demonstrating that the internal communications within the server and hosted cloud services are robust (see Figure 11b).



**Figure 8.** Example of a wheat image acquired with the guiding vision system and uploaded to the cloud.

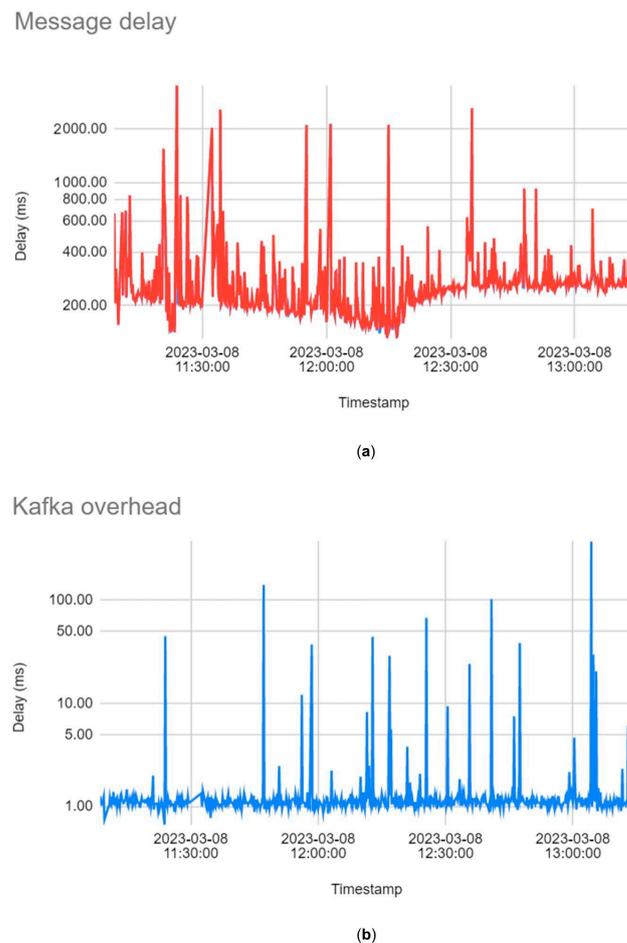


**Figure 9.** Example of a maize image acquired with the guiding vision system and uploaded to the cloud.

The system has been tested in a field with two different crops. Data related to cloud communication and robot guidance algorithms have been collected. The communication performance is similar to that obtained using conventional mechanisms, so we benefit from using ROS and FIWARE without compromising performance.



**Figure 10.** Example of a ToF intensity image acquired with the guidance system and uploaded to the cloud.



**Figure 11.** Example of a ToF intensity image acquired with the guidance system and uploaded to the cloud. (a) Message delay and (b) Kafka overhead.

## 5. Conclusions

An architecture is presented to configure autonomous robots for agriculture with access to cloud technologies. This structure takes advantage of new concepts and technologies, such as IoT and cloud computing, allowing big data, edge computing, and digital twins to be incorporated into modern agricultural robots.

The architecture is based on ROS, the most universally accepted collection of software libraries and tools for building robotic applications, and FIWARE, an open architecture that enables the creation of new applications and services on the Internet. ROS and FIWARE provide attractive advantages for developers and farmers. ROS and FIWARE offer powerful tools for developers to build control architectures for complex robots with cloud computing/IoT features, making development easier and leveraging open-source frameworks. ROS and FIWARE, as in the proposed integration, provide reusability, scalability, and maintenance using the appropriate hardware resources. In addition, integrating the robot controller into the Internet allows the exploitation of autonomous robot services for agriculture through the Internet.

On the other hand, the use of this type of architecture reveals to farmers the advantages of communicating autonomous robots with the cloud, providing them with leading benefits to storing data safely and efficiently, eliminating physical storage, and, thus, reducing the risk of data loss. Data stored in the cloud makes it easy to access data from anywhere and share it with other farmers or platforms. In addition, the services offered in the cloud are very flexible to contract the actual storage needed at all times, optimizing the farmer's resources. Finally, farmers can use the analysis tools available in the cloud to make their

own decisions. In any case, working in the cloud requires an initial investment, which is usually recovered quickly.

The different components of the robot, particularized for a laser-based weeding robot, are described, and the general architecture is presented, indicating the specific interfaces. Based on these components, the article presents the action sequence of the robot and the operating procedure to illustrate how farmers can use the system and what benefits they can obtain.

Several experiments with two crops were conducted to evaluate the proposed integration based on the data communication characteristics, demonstrating the system's capabilities. The crop row detection system works correctly for both crops, tracking the rows with an accuracy of  $\pm 0.02$  m. The evaluation concluded that the system could send image frames to the cloud at 4 frames/s; messages between subsystems and modules can be passed with a 0.63% rejection rate. Regarding the traffic of the information exchanged, an average delay of 250 ms was detected in the messages between the robot and the OCB. In contrast, the OCB and the KAFKA bus measured an average message of 1.24 ms. This indicates the robustness of internal communications within the server and hosted cloud services. This performance is in the range obtained when a system communicates with the cloud using conventional methods, so ROS and FIWARE facilitate communication with the cloud without compromising performance.

Future work will focus on extending cloud computing architecture to integrate digital twins, orchestrate big data ensembles, and facilitate the work of robots with edge computing performance.

**Author Contributions:** Conceptualization, L.E., R.F., P.G.-d.-S., M.F., M.G., G.V., H.S., M.H. and M.W.; methodology, L.E. and R.F.; software, L.E., M.F., H.S. and M.W.; validation, L.E., M.F., G.V. and H.S.; investigation, L.E., R.F., P.G.-d.-S., M.F., M.G., G.V., H.S., M.H. and M.W.; writing—original draft preparation, P.G.-d.-S.; writing—review and editing, L.E., P.G.-d.-S. and R.F.; supervision, L.E. and P.G.-d.-S.; funding acquisition, P.G.-d.-S., G.V., M.G. and M.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This article is part of a project that has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 101000256.

**Institutional Review Board Statement:** The study was conducted in accordance with the Declaration of Helsinki, and approved by the Ethics Committee of CSIC.

**Data Availability Statement:** Herrera-Diaz, J.; Emmi, L.A.; Gonzalez de Santos, P. Maize Dataset. 2022. Available online: <http://doi.org/10.20350/digitalCSIC/14566> (accessed on 1 April 2023). Herrera-Diaz, J.; Emmi, L.; Gonzalez de Santos, P. Wheat Dataset. 2022. Available online: <http://doi.org/10.20350/digitalCSIC/14567> (accessed on 1 April 2023).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Ghose, B. Food security and food self-sufficiency in China: From past to 2050. *Food Energy Secur.* **2014**, *3*, 86–95. [[CrossRef](#)]
2. Zhang, N.; Wang, M.; Wang, N. Precision agriculture—A worldwide overview. *Comput. Electron. Agric.* **2002**, *36*, 113–132. [[CrossRef](#)]
3. Stentz, A.; Dima, C.; Wellington, C.; Herman, H.; Stager, D. A System for Semi-Autonomous Tractor Operations. *Auton. Robot.* **2002**, *13*, 87–104. [[CrossRef](#)]
4. Bergerman, M.; Maeta, S.M.; Zhang, J.; Freitas, G.M.; Hamner, B.; Singh, S.; Kantor, G. Robot Farmers: Autonomous Orchard Vehicles Help Tree Fruit Production. *IEEE Robot. Autom. Mag.* **2015**, *22*, 54–63. [[CrossRef](#)]
5. Gonzalez-De-Santos, P.; Ribeiro, A.; Fernandez-Quintanilla, C.; Lopez-Granados, F.; Brandstoeffer, M.; Tomic, S.; Pedrazzi, S.; Peruzzi, A.; Pajares, G.; Kaplanis, G.; et al. Fleets of robots for environmentally-safe pest control in agriculture. *Precis. Agric.* **2017**, *18*, 574–614. [[CrossRef](#)]
6. Underwood, J.P.; Calleja, M.; Taylor, Z.; Hung, C.; Nieto JFitch, R.; Sukkarieh, S. Real-time target detection and steerable spray for vegetable crops. In Proceedings of the International Conference on Robotics and Automation: Robotics in Agriculture Workshop, Seattle, WA, USA, 9–11 May 2015.
7. Kongskilde. New Automated Agricultural Platform—Kongskilde Vibro Crop Robotti. 2017. Available online: <http://conpleks.com/robotech/new-automated> (accessed on 14 December 2022).

8. Emmi, L.; Herrera-Diaz, J.; Gonzalez-De-Santos, P. Toward Autonomous Mobile Robot Navigation in Early-Stage Crop Growth. In Proceedings of the 19th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2022), Lisbon, Portugal, 14–16 July 2022; pp. 411–418. [CrossRef]
9. Bannerjee, G.; Sarkar, U.; Das, S.; Ghosh, I. Artificial Intelligence in Agriculture: A Literature Survey. *Int. J. Sci. Res. Comput. Sci. Appl. Manag. Stud.* **2018**, *7*, 3.
10. Osinga, S.A.; Paudel, D.; Mouzakitis, S.A.; Athanasiadis, I.N. Big data in agriculture: Between opportunity and solution. *Agric. Syst.* **2021**, *195*, 103298. [CrossRef]
11. Yang, D.; Li, D.; Sun, H. 2D Dubins Path in Environments with Obstacle. *Math. Probl. Eng.* **2013**, *2013*, 291372. [CrossRef]
12. Emmi, L.; Parra, R.; González-de-Santos, P. Digital representation of smart agricultural environments for robot navigation. In Proceedings of the 10th International Conference on ICT in Agriculture, Food & Environment (HAICTA 2022), Athens, Greece, 22–25 September 2022; pp. 1–6.
13. Orion Context Broker. Telefonica. Available online: <https://github.com/telefonicaid/fiware-orion> (accessed on 22 February 2023).
14. Francia, M.; Gallinucci, E.; Golfarelli, M.; Leoni, A.G.; Rizzi, S.; Santolini, N. Making data platforms smarter with MOSES. *Futur. Gener. Comput. Syst.* **2021**, *125*, 299–313. [CrossRef]
15. ROS—The Robot Operating System. 2023. Available online: <https://www.ros.org/> (accessed on 24 April 2020).
16. ROSLink. 2023. Available online: <https://github.com/aniskoubaa/roslink> (accessed on 5 January 2023).
17. Koubaa, A.; Alajlan, M.; Qureshi, B. ROSLink: Bridging ROS with the Internet-of-Things for Cloud Robotics. In *Robot Operating System (ROS)*; Koubaa, A., Ed.; Studies in Computational Intelligence; Springer: Cham, Switzerland, 2017; Volume 707. [CrossRef]
18. Fiware Community Fiware: The Open Source Platform for Our Smart Digital Future. Available online: <https://www.fiware.org/> (accessed on 5 January 2023).
19. López-Riquelme, J.; Pavón-Pulido, N.; Navarro-Hellín, H.; Soto-Valles, F.; Torres-Sánchez, R. A software architecture based on FIWARE cloud for Precision Agriculture. *Agric. Water Manag.* **2017**, *183*, 123–135. [CrossRef]
20. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
21. Herrera-Diaz, J.; Emmi, L.A.; Gonzalez de Santos, P. Maize Dataset. 2022. Available online: <https://digital.csic.es/handle/10261/264581> (accessed on 1 April 2023).
22. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
23. Herrera-Diaz, J.; Emmi, L.; Gonzalez de Santos, P. Wheat Dataset. 2022. Available online: <https://digital.csic.es/handle/10261/264622> (accessed on 1 April 2023).
24. Wojke, N.; Bewley, A.; Paulus, D. Simple Online and Realtime Tracking with a Deep Association Metric. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.