Smart Agricultural Technology 7 (2024) 100426

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

Cable-driven agribot prototype: Enabling precision agriculture through innovative design

Stefano Leonori *, Stefano Mattei, Luigi Anniballi, Fabio Massimo Frattale Mascioli

DIET, Department Information Electronics Telecommunications, Via Eudossiana 18, Rome, 00184, Italy

A R T I C L E I N F O

Editor: Spyros Fountas

Keywords:
Agribot
Multi-robot
Agriculture 4.0
Sustainability
ROS

A B S T R A C T

The ongoing depopulation of rural areas and the increasing demand for high-quality food, along with the need to improve the quality of work in terms of stress, injuries, and social disputes, needs significant attention towards rethinking and organizing agricultural practices to meet these needs and improve the quality of life. These changes call for the implementation of new reforms, incentive systems, and the exploration of innovative systems for field operations. In this context, the research, design, and commercialization of electrified autonomous and collaborative systems capable of precision agriculture applications, known as agribots, are gaining paramount importance. This article presents an innovative multi-robot for smart farming. The agribot is composed by a lightweight cart designed to traverse crop field rows through a cable-way system installed on two heavy carts equipped with a power train system and PV rooftop panels. The agribot is conceived to achieve a high level of autonomy, reduce energy consumption, and minimize soil compaction while working on flat fields with low-height crops. The article is structured in two parts: the first part describes the agribot's ICT network, the heavy carts electric grid and the power systems, providing an energy consumption analysis related to the traversing of a crop field row.

The second part focuses on the control, actions coordination, and supervision system (supervisor), implemented using Robot Operating System (ROS). The supervisor system, driven by a decision-making system based on a finite state machine, oversees the execution of path planning for the agribot, providing high modularity and flexibility to adapt to various system actions.

1. Introduction

In the last years international organizations are alarming the society about the rising of the population that is not enough supported by an increase in food productivity and sustainability. Tripathi et al. [42] analyze how much the world population has risen and will rise until 2050 and the corresponding need of food production to ensure and enhance food security, namely, "the physical and economic access to sufficient, safe and nutritious food that meets the population dietary needs and food preferences for an active and healthy life" [15]. After examining the data reported by *The World Bank* [47] concerning historical trends of agricultural land worldwide, as well as the historical data on global crop field area per capita reported by FAO Press Release [14], discouraging figures become evident. Indeed, the data indicate that the dedicated land per capita has continuously decreased from approximately 0.45 hectares in 1961 to 0.21 hectares in 2016, while the global agricultural land has decreased by over one million hectares since 2000, currently covering 47 million km^2.

Numerous factors have contributed and continue to contribute to these negative trends, extending beyond the increasing recurrence of droughts, floods, forest fires, and new pests, which serve as constant reminders that our food system is under threat.

Firstly, the swift migration of local and small farmers to urban areas not only reduces the agricultural workforce but also leads to urban expansion at the expense of land dedicated to food production. It is projected that by 2030, approximately 60% of the world's population will reside in cities, a figure expected to rise to 68% by 2050 according to FAO and Organization [13], Ayaz et al. [2]. This phenomenon is largely driven by imbalanced competition with major food corporations and inadequate labor conditions, particularly in developing countries.

Following, another concern involves biofuels production which increased more than threefold between 2000 and 2008. Although biofuels contribute to energy sustainability, especially in the transportation sec-

* Corresponding author.
E-mail address: stefano.leonori@uniroma1.it (S. Leonori).

https://doi.org/10.1016/j.atech.2024.100426
Received 7 November 2023; Received in revised form 27 February 2024; Accepted 29 February 2024
Available online 8 March 2024

tor, they also diminish the available land for food production, involving significant implications for food security. In 2007 – 2008, the total utilization of coarse grains for the production of ethanol reached 110 million tons, about 10% of global production [42].

In a similar vein, the uncontrolled spread of distributed generation, such as wind turbines and PV generators, is currently reducing crop field areas. The conversion of productive farmlands into energy generation infrastructures renders land portions inaccessible. This phenomenon also contributes to a misguided perception among urban communities that power generation from renewable sources holds greater importance than agricultural production.

To address this issue, it becomes imperative to introduce incentive programs for the development of *agrivoltaic* systems. This term denotes “the strategic co-development of land for both solar PV energy production and agriculture which can meet growing demands for energy and food simultaneously while reducing fossil fuel consumption” as asserted by Pascaris et al. [31], Guerin [16]. On this purpose, Miskin et al. [25] investigated innovative and suitable solar generation technologies for electricity production in agricultural lands. Their effectiveness is evaluated through a shadow simulation model that calculates the power output of the chosen generation system and evaluates its interaction with the local environment where it is intended to be installed.

The challenging conditions of working the land constitute another critical point. As emphasized by Benos et al. [4,5,6], safety and healthy conditions in agricultural environments have garnered significant attention in the literature. Indeed, among all work environments, agriculture is considered one of the most hazardous, with alarming rates of epidemiological evidence revealing numerous health issues, including hearing loss, respiratory diseases, and various types of cancer. For instance, the extensive use of toxic herbicides and fertilizer sprays poses risks not only to the health of farm workers and people, but also to soil contamination and the overall costs of food products, as highlighted by Swan et al. [41], Berenstein and Edan [7]. Similar to many sectors, agriculture faces a high prevalence of non-fatal diseases, particularly Musculo-Skeletal Disorders (MSD).[1]

In the United States, agriculture ranks as the second sector, after mining, with the highest number of occupational injuries surpassing other sectors such as construction or manufacturing. A similar scenario is observed in Europe, where agriculture and forestry exhibit a greater frequency of accidents compared to any other industrial sector, mostly caused by slips, falls, and loss of machine control as reported by Robert et al. [34]. In this document, authors also present a study on the advantages and disadvantages of tractor usage, asserting that while tractors have historically provided valuable solutions in various functionalities, operating them involves interacting with engaging buttons, levers, clutch, and brake pedals. Additionally, during maneuvers, tractor operators need to periodically turn to look behind, which can impact the load on body parts that come into contact with mechanized components. Consequently, the combination of prolonged sitting and these repetitive movements may contribute to the development of MSD. Moreover, the extensive use of toxic herbicides and fertilizer sprays poses risks not only to the health of farm workers and people, but also to soil contamination and the overall costs of food products, as highlighted by Swan et al. [41], Berenstein and Edan [7].

In light of these critical concerns, as supported by the FAO and the European Community, there is a clear need to initiate a significant system of incentives and support for ethically sustainable food production, particularly aimed at assisting small producers. To this end, the European Green Deal launched a program called the *Farm to Fork Strategy* [44], outlining a new approach to ensure that agriculture, fisheries, aquaculture, and the food value chain contribute appropriately to this process. The Farm to Fork Strategy aims to adopt a comprehensive approach to increase productivity while focusing on energy sustainability, improving work quality, reducing dependency on pesticides and antimicrobials, minimizing excess fertilization, promoting organic farming, enhancing animal welfare, and reversing biodiversity loss.

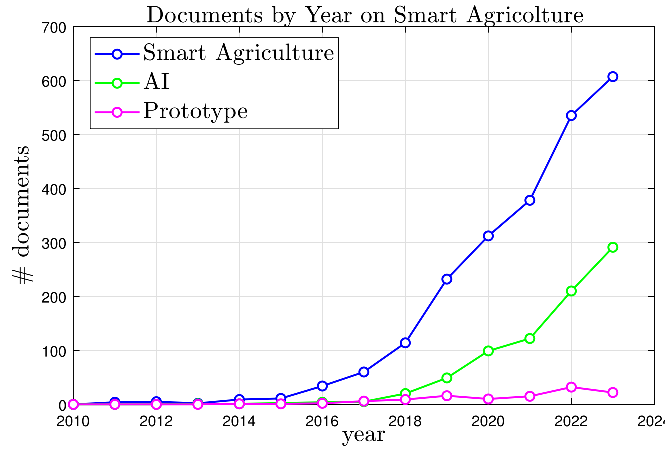

Fig. 1. Number of publications registered on Scopus related to: “agribot”, “smart agriculture”, “agriculture 4.0”, “PA”, “smart farming”. In green are considered only those which include in *title* or in *keywords* one of the following words: “deep learning”, “neural networks”, “machine learning”, “artificial intelligence”, “data driven”. In magenta only the documents which include the word “agribot”.

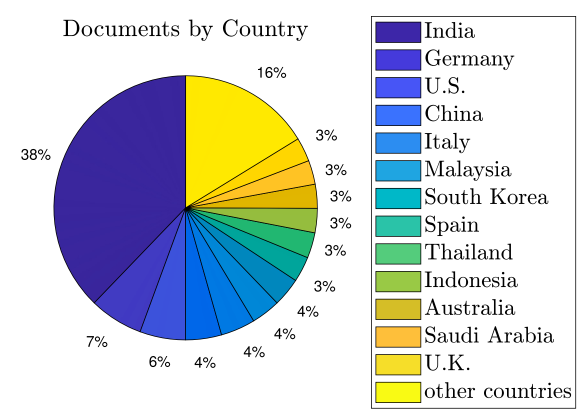

Fig. 2. Publications of Fig. 1 distinguished by country.

In recent years, significant advances have been made in developing a more efficient, sustainable, and ethically sound agricultural system. This progress is also the result of an extensive research, as evidenced by the graph in Fig. 1, which clearly indicate the exponential growth of published documents (articles and book chapters) registered on Scopus from 2010 to 2023. Further insights are given by the pie charts in Fig. 2, 3 and 4, illustrating the most prolific countries in terms of research publications, the research areas most involved, and the type of published documents, respectively.

In the literature, engineering and scientific journals predominantly focus on Precision Agriculture (PA) technologies. According to Lowenberg-DeBoer and Erickson [23], the term *PA* refers to the use of electronic information and other technologies to collect, process, and analyze spatial and temporal data, aiming to guide targeted actions that enhance the efficiency, productivity, and sustainability of agricultural operations. It has gained prominence with the rise of *Industry 4.0*, facilitated by the increasing adoption of Internet of Things (IoT), advanced sensors, robotics, and Artificial Intelligence (AI) technologies, allowing for the development of smart, autonomous, and collaborative robots in the farming context, commonly known as *agribots*.

[1] MSD is a generic term used to describe medical conditions associated with awkward postures, repetitive movements, heavy lifting, vibrations, and work in adverse weather conditions.

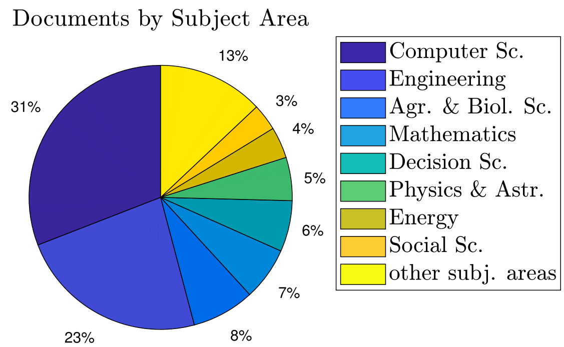

Fig. 3. Publications of Fig. 1 distinguished per subject area.

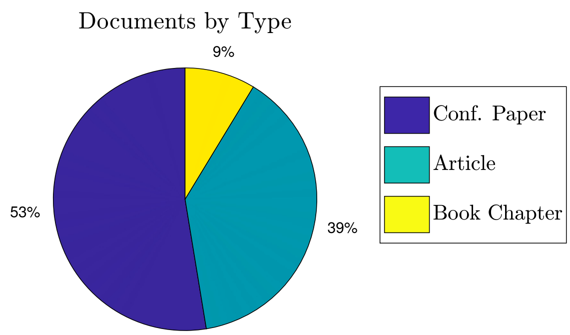

Fig. 4. Publications of Fig. 1 distinguished by type of document.

The development of agribots aims to achieve several objectives, including optimizing and reducing human labor, mitigating global and local emissions, reducing soil compaction, improving product quality, and supporting the use of energy produced by distributed generation and renewable energy sources (*e.g.*, agrivoltaic systems).

Wakchaure et al. [46] categorize the main functions of PA for agribots according to a chronological framework, which includes:

- Cultivation: planning of crops to be planted, planning of land, land preparation, planning of water irrigation, seed sowing.
- Monitoring: continuous monitoring, data collection, disease detection, weed control, use of fertilizers, pesticide spraying.
- Harvesting: segmentation, cutting, picking of crops and fruits, storing.

Tasks related to monitoring, prediction, control, and planning are often delegated to remote systems that manage large volumes of data and employ computationally intensive algorithms. Botta et al. [9] provides a survey of the most adopted sensors and smart devices for accomplishing these functions.

As stated by Sachithra and Subhashini [38] and Wakchaure et al. [46], monitoring activities are the most extensively studied. These activities are primarily conducted with the use of infrared and stereo cameras to establish image recognition systems. Conversely, harvesting activities are deemed to require the most attention and study due to their greater complexity. They rely on special cameras and lighting devices to support segmentation functions for crop cutting, along with high-sensitive mechanical arms (*e.g.* air gripper, grasping by soft mechanical fingers) able to avoid damaging the crop.

The execution of PA functions must be assisted by a series of supportive tasks aimed at ensuring accurate and safe maneuvering and positioning of the agribot. These tasks include localization, obstacle avoidance navigation, object detection, and decision-making for effective operation. Additionally, adaptability in route planning is essential for agribots to interact with humans, other robots, untrained devices, or to handle unforeseen scenarios [9]. To meet these requirements, agribots need to be equipped with sensors capable of detecting distances, indoor or outdoor positioning, speeds, and accelerations while considering uncertainties. As a result, they are typically outfitted with Real Time Kinematik (RTK) Global Navigation Satellite System (GNSS), accelerometers, gyroscopes, as well as sonars, radars, lasers, lidars, and infrared sensors.

With regards to the types of agribots discussed in the literature, they can be categorized as follows: large-scale ones, such as tractors and forklifts retrofitted into autonomous systems; medium to small-scale UGV including rovers, rail carriages, and sliding systems; UAV, namely drones and fixed-wing aircraft. Rail carriages and sliding systems are primarily utilized for indoor applications, such as greenhouses and vertical farming, making them more suitable for harvesting activities [32]. Conversely, rovers can cover both indoor and outdoor environments due to their navigational adaptability. Moreover, they are particularly suitable for experimental purposes and initial prototyping due to their lower costs and simpler design, control, and powertrain machinery. UAV systems are extensively employed for remote sensing activities such as monitoring, mapping, crop inspection, soil assessment, and vegetation health assessment. They can also be utilized for weed control, sowing, and pesticide spraying, although these applications are less prevalent due to challenges in control, costs, energy consumption, and autonomy [24,38].

Smart agribots commonly integrate AI models to perform the agribot activities and supportive tasks. [38] reported a systematic review of commonly adopted AI models, such as fuzzy logic, k-nearest neighbor, random forest, support vector machine, neural networks, and deep learning. The review includes multiple case studies detailing the algorithms used and their applications. It concludes that the majority of projects focus on developing prediction models, monitoring systems, and data management, with deep learning being the most commonly adopted model. Similarly, Wakchaure et al. [46] present a review focused on 150 agribot AI models implementation case studies, leading to similar conclusions. The widespread use of deep learning is further emphasized by Chen et al. [10], Montoya-Cavero et al. [26], Li et al. [21,22], who highlight its effectiveness in image recognition for precision agriculture activities, providing valuable insights into current practices.

However, as criticized by Yerebakan and Hu [48] and Wakchaure et al. [46], further research is required for the implementation of AI in PA. Indeed, the proposed solutions discussed in the literature are often tested in simulators and virtual environments rather than real-world settings. Additionally, as depicted in Fig. 4, the literature on PA applications predominantly consists of conference papers. Another critical issue, as pointed out by Rose et al. [37], is the lack of transparency in algorithms, machine learning, and AI, commonly referred to as the ‘black box problem’. This can lead to bias and discrimination issues within machine learning, perpetuating inequalities. Therefore, regulatory oversight of equality by design is necessary to ensure that programmers address any bias and discrimination that may arise in algorithms, ultimately ensuring fairness in the use of AI technology.

Concerning the development of fully autonomous agribots, researchers such as Benos et al. [4] and Esram et al. [11] express a critical point of view on the replacement of human capabilities in thinking, perceiving, decision-making, and action-taking. In relation to Industry 4.0, the agricultural sector faces higher volatility due to the handling of crops that are highly sensitive to temperature, humidity, and climate, as well as rapid deterioration. Moreover, soil irregularity, weather unpredictability, and randomness of crop growth further complicate device maneuvers. Additionally, as noted by Yerebakan and Hu [48] and Wakchaure et al. [46], research on prototype development is still far from achieving commercial viability, highlighting the prevalence of simple small-scale prototypes in the literature, as well as a significant lag behind AI applications (see Fig. 1). Due to the challenges involved in developing fully autonomous systems, several studies in the literature advocate technological solutions based on Human Robot Interaction

(HRI) and collaboration, as extensively discussed by Benos et al. [4] and Hentout et al. [17]. HRI is an emerging research field, which originates in the manufacturing domain (*i.e.*, Industry 4.0). It focuses on studying the physical, cognitive, and social interaction between humans and robots to extend and enhance human capabilities and skills. It also concentrates on the design, understanding, and evaluation of interaction between people and robots that can communicate and/or share physical space [45]. HRI is defined by Fang et al. [12] as: "the process that conveys and channels human operators' intention and interprets task descriptions into a sequence of robot movements that conform to the robot's capabilities and task requirements". Whereas, in Schmidtler et al. [40], it is simply defined as a general term for all forms of interaction between humans and robots. The developing of HRI systems in agriculture robotics would ease the labor pressure, for example replacing those repetitive efforts like transporting products to the storage, helping the operators through AI based detection systems which indicates where to intervene, supporting in decision-making predicting and thus deciding when it is necessary to start specific activities optimizing the human effort [2,8].

Collaborative robots, often referred to as *cobots*, represent a natural evolution that could address a range of challenges in agricultural activities such as picking and placing, and packaging. In such context, HRI combines human capabilities for judgment and response with the strength and repeatability of robots. Their design enables versatile usage across numerous applications within an ever-evolving workflow.

The advent of cobots in smart farming can provide the following advantages: rapid capital depreciation, leading to reduced production costs; flexibility in system reconfiguration; optimization of productivity and service quality, thereby ensuring value-added products; minimization of required workspace [4]. Lytridis et al. [24] presented a review on collaborative agribots designed for monitoring, spraying, harvesting, and transport, stating that the research literature focuses on two main areas: improving the sensory limitations of current vision-based systems, where the human operator complements the robot's automatic detection capabilities finalizing the work, and supporting manual labor, where the robot acts as a human assistant.

It is important to note that both full autonomous and collaborative agribots must ensure human safety and protection. According to Basu et al. [3] and Saenz et al. [39], the study and establishment of standards related to HRI technology development have become well-established within the context of Industry 4.0, characterized by organized and rational work environments. However, extending this technology to smart farming or Agriculture 4.0 demands substantial efforts in terms of regulations, costs, and technological solutions due to the more challenging, uncertain, and unpredictable work settings. For this reason, the authors underline the importance of formulating and adapting safety standards for the advancement of both collaborative and fully automated robots in agricultural contexts. This adaptation should involve aligning legal regulations and standards with those established in the Industry 4.0 framework, as discussed by Lytridis et al. [24] and Mukherjee et al. [27], Robla-Gómez et al. [35], aiming to find a reasonable balance between the obligation to protect operators and the necessity to foster innovation in a rapidly evolving technological environment.

This paper presents a novel full electric agribot prototype equipped with rooftop PV, specifically designed for PA applications. It consists of two multi-motor robotic vehicles that navigating along the crop field edges carry out their mission towing an operating machine, a lightweight cart, on the crop field rows through a rope-way based transport system. The benefits of this proposed technology include:

- reduction of soil compaction for the presence of the light-weight cart alone on the cultivated soil;
- reduction of overall emissions since the system is electrified, is equipped with rooftop panels, and is low power demanding due to the adoption of a cable way transportation system for the lightweight cart and the low operating speed of the other robots;
- high level of safety guaranteed by the low operating speed of the robots and by the presence of the only light-weight cart on the cultivated area.

Detailed information on the consequences of soil compaction on cultivated fields is extensively documented by Nawaz et al. [29] and Oliveira et al. [30].

The developed prototype is unique in its kind, and although it is limited to low-height crops, the use of the cable-way system allows for potentially much greater autonomy than that of electrified tractors, drones, and rovers, as the carts are mostly stationary, and the lightweight cart is characterized by low consumption. Moreover, the low speeds make the system intrinsically safe and easier to configure for collaborative use.

During our search for similar systems, we discovered a resemblance to historical agricultural machinery produced by the British company John Fowler & Co, based on a patent filed in 1856. This machinery employed a plowing method using two self-propelled engines, comprising two steam locomotives, each weighing 19 tons, and a plow weighing 45 quintals, equipped with a single furrow balance beam measuring 8 meters in length[2, 3]. Additionally, our agribot shares some similarities with the FarmBot agribot [28]. The FarmBot is a stationary agribot designed for crop cultivation, operating similarly to a 3D printer on a designated cultivation area, where it is mounted above. In this setup, the operating machine moves using two linear guides, similar to the heavy vehicles that drive the lightweight cart.

In this manuscript, the agribot system architecture is presented, focusing on both power and communication aspects. Special attention is given to the system's activities, energy consumption, and the supervision, actions coordination, and control systems developed to accomplish the path planning of the operating machine, coordinating all three vehicles through the implementation of a Finite State Machine (FSM). For simplicity, the supervision and actions coordination systems will be referred to as the supervisor throughout the manuscript.

The paper is organized as follows. In Sec. 2, the agribot system is described, explaining its conception and functioning. The Information and Communications Technology (ICT), power network and power systems are detailed in Sec. 3. Sec. 4 focuses on the programming of the agribot actions, while their supervisor and control are discussed in Sec. 5. Sec. 6 introduces the Robot Operating System (ROS) application, which allows showing and discussing the functioning and architecture of the supervisor in the next section, with a special concern on the FSM decision-making system (Sec. 7). Finally, conclusions and future works are written in Sec. 8.

2. Introduction to the agribot system

The agribot system here discussed comprises two heavy robotic vehicles, named H-Carts, equipped with an autonomous navigation system, and one lightweight operating device, named L-Cart. The H-Carts are designed to travel along the edges of the crop field and guide the L-Cart along the crop rows through a rope-way-based transport system while it operates on the soil. Once the L-Cart completes its operation on a row, traversing the farm land entirely, the nearby H-Cart lifts it up using a four-arm forklift. Afterward, both H-Carts move to the ends of a new row in accordance with the system's path planning, which is defined by a set of waypoints referring to the endpoints of each crop field row. The application of the rope-way-based transport system is to avoid burdening the L-Cart with a power system, limiting its hardware to devices dedicated to PA applications.

It is important to clarify that the use of the rope-way system implies that the system's application is limited to flat and level agricultural

[2] https://www.tractorhouse.it/blog/attualita/2015/10/fowler-una-macchina-da-favola.

[3] https://merl.reading.ac.uk/explore/online-exhibitions/john-fowler/.

Fig. 5. Photos of the project inside the POlo per la MObilità Sostenibile (Po.Mo.S.) Lab. and outdoor.

Fig. 6. Photos of the project from another prospective and the L-Cart.

fields with slight inclinations. Moreover, the crops should have heights lower than the bottom part of the L-Cart, approximately one meter.

Fig. 5 shows a photo of the project where the H-Carts can be observed driving the L-Cart positioned in between while Fig. 6 displays the L-Cart equipped with a spreader seeder and a tank capable of containing water, pesticides, and fertilizers.

For the accomplishment of the L-Cart path planning, a series of sequential actions are necessary, which are summarized in five *tasks* as follows:

- Operate the rope-way based transport system for controlling and driving the L-Cart (**L-Cart driving** task).
- Utilize H-Cart lifters to place the L-Cart on the crop field rows (**lift up** and **lift down** tasks).
- Navigate the H-Carts by means the navigation system to an assigned waypoint (**H-Cart navigation** task).
- Execute the PA tasks (this task won't be discussed in the manuscript).

Before detailing each task, the H-Carts nanogrid system, along with its components, sensors, and devices, will be described in the next section.

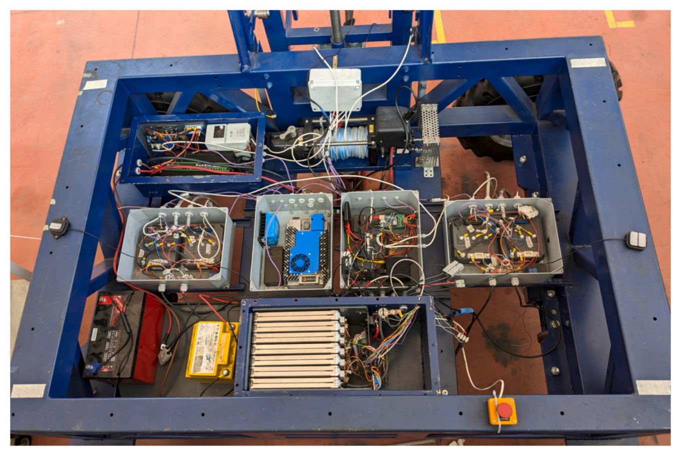

Fig. 7. Photo of the H-Cart power grid and the ICT network.

3. H-cart power & ICT networks

In Fig. 8, a map of the H-Cart nanogrid is presented, encompassing the power systems, power lines, the ICT network, boards, sensors, and devices, along with details on the voltage and type of power and communication lines. Additionally, a photo of the H-Cart nanogrid is displayed in Fig. 7.

The following subsections will provide detailed descriptions of the power systems and ICT components.

3.1. Power network

As shown in Fig. 5, the H-Cart powertrain system consists of four drive wheels mounted on their respective gear motors, which have a gear ratio of 1 : 10 that ensure a turning radius below 13 m on asphalt surfaces. The gear motors consist of 2 kW permanent magnet synchronous motors with a nominal voltage of 48 V. They are powered by a second life lithium polymer battery pack assembled by 14 3.7 V lithium-polymer cells (with a 3.6 V nominal voltage) sourced from a *Nissan Env200* battery. Each motor is connected to the battery through a DC/AC converter equipped with an embedded loop current control driver programmed for torque control.

The 48 V power line also connects two 335 W PV panels equipped with a MPPT controller, which increase system autonomy.

The H-Cart lifting mechanism utilizes a worm gear motor fitted with a trapezoidal screw for vertical translation of a frame composed of 4 lifting forks. The gear reducer, or screw jack, powered by a 1.1 kW three-phase asynchronous motor, is regulated by a single-phase DC-AC inverter and a motion controller generating three-phase power supply. The converter is powered by a 12 V lead-acid battery. The lifting mechanism control system operates in three discrete configurations: lift up, lift down, and stop.

The rope-way system for driving the L-Cart comprises two winches mounted on their respective H-Carts. They are powered by a DC motor supplied by the same battery pack as the lifting mechanism. A chopper has been installed upstream of the winch motor to enable Pulse Width Modulation (pwm) control for torque modulation, facilitating the application of a Proportional Integral Derivative (PID) control loop on the L-Cart's position to limit maximum speed.

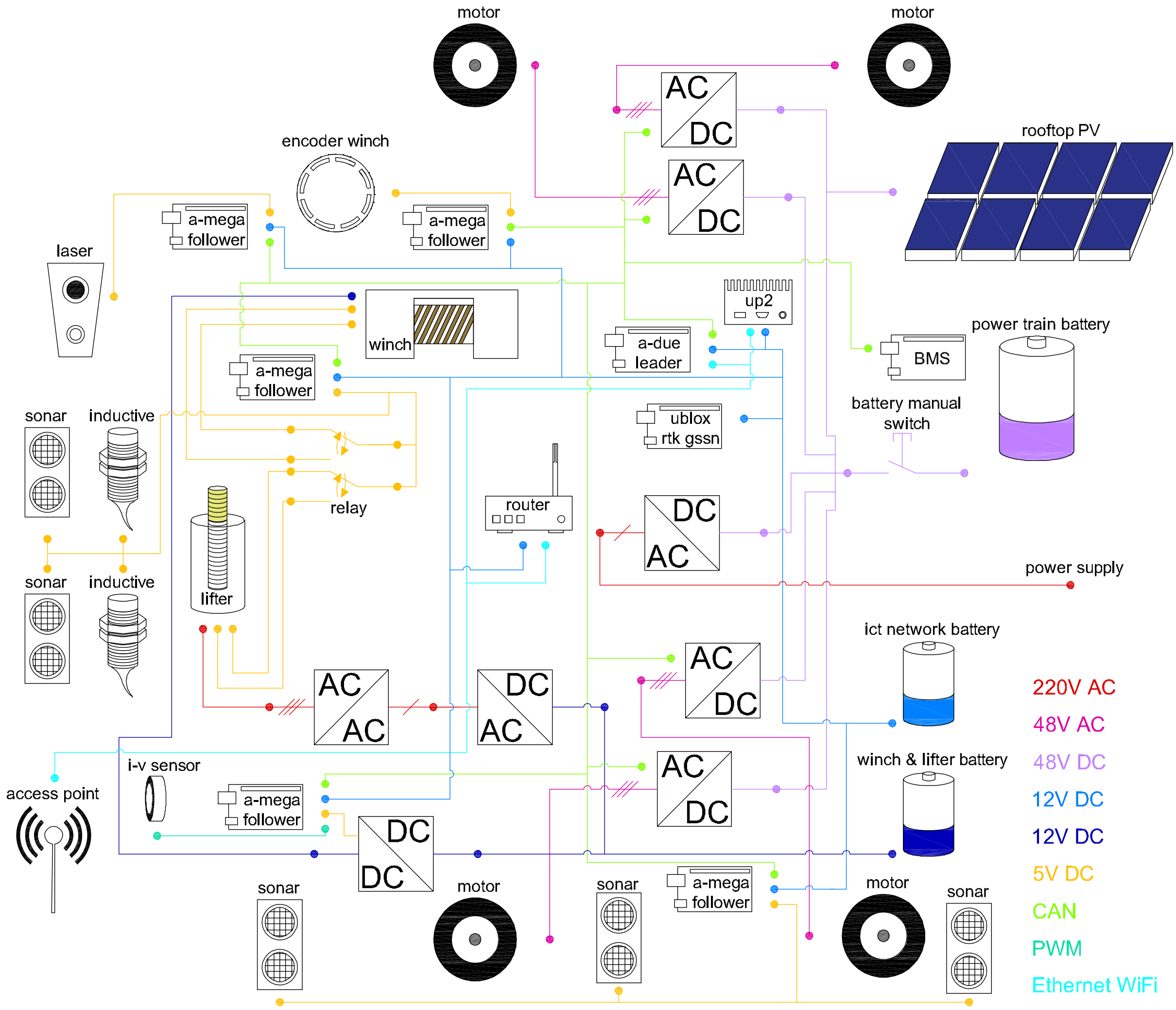

Fig. 8. H-Cart power grid and ICT network map.

3.2. ICT network

The ICT network is powered by a 12 V lead-acid battery, which supplies all controller boards, the supervisor board, the router for Ethernet transmission, as well as the Wi-Fi access point. Using a dedicated battery enables isolation from the power systems, allowing the separation of the ICT network's mass from that of the electric motors. This helps preventing disturbances and interferences caused by high current demands. As illustrated in Fig. 8, the supervisor system is implemented on a UP2 board, and communicates via Ethernet with an Arduino Due board named *leader* and via Wi-Fi with the other leader boards of the H-Cart and L-Cart. Ethernet communication enables remote communication across wide geographical areas with high bandwidth. Additionally, the router facilitates remote communication between the operator and the cart devices using the UMTS protocol. Communication between leader and follower boards occurs via the Controller Area Network (CAN) bus, while sensors and small devices communicate with follower boards and the supervisor via serial I2C, TTL, and USB protocols. Lastly, the integration of these boards, communication systems, and protocols for the ICT network is in line with the communication technologies utilized in agribot systems, as documented by Tzounis et al. [43], which extensively surveys communication systems, IoT technologies, software, and hardware predominantly adopted in smart agriculture.

4. Tasks description

In this section are described in details the tasks summarized in Sec. 2 for accomplishing the agribot mission. Table 1 presents a summary of the sensors grouped by task, along with their respective functions. They are configured to fulfill two roles: first, providing all the information necessary to execute the planned tasks, and second, determining when the system should stop for safety reasons.

4.1. H-cart navigation

The navigation system is responsible for guiding the H-Carts to predefined waypoints, which correspond to the ends of each crop land row. For this purpose, the H-Carts are equipped with dual-band RTK GNSS boards [20] that enables the determination of the cart's position with an accuracy of less than 20 cm and the heading evaluation by installing another simpler GNSS board. Specifically, the RTK GNSS system achieves highly precise positioning by utilizing correction data received from a nearby permanent GNSS station via the Internet. This station, known as the caster, transmits correction data using the Networked Transport of RTCM via Internet Protocol (NTRIP) protocol to the leader board installed on the L-Cart. Upon receiving this data, the leader board converts it into a Radio Technical Commission for Maritime Services3 (RTCM3) message, formatted according to the requirements of the supervisor software, that is based on ROS. Subsequently, the supervisor

Table 1
List of sensors sorted for each task.

H-Cart Navigation	
Sensors	**Function**
motors encoders	odometry calc.
sonar	obstacle detection
differential GNSS	geog. position, heading
H-Cart Lifter	
Sensors	**Function**
proximity sensors	up, down, btw. status
motor curr.-volt.	motor state meas.
Rope-Way System	
Sensors	**Function**
laser	L-Cart-H-Cart dist.
winch encoders	rope released meas.
sonar	L-Cart-lifter proximity
cell weight	rope tension
motor curr.-volt.	winch current control

software reads the message and sends the corrected data to the GNSS card chip via USB, facilitating necessary adjustments. Additionally, the RTCM3 data transmitted by the L-Cart leader board is utilized by the L-Cart's own GNSS receiver, as well as the GNSS receivers on the participant H-Carts. For the H-Carts heading measurements, a differential GNSS receiver is connected to the primary receiver with a second antenna positioned along the vehicle's axis at a distance greater than 1 m from the primary GNSS. This configuration enables the computation of differential measurements, providing the orientation angle between the two antennas with respect to the east. Since both receivers feature similar positioning errors, the heading measurement achieves high precision, making the error negligible in this case study.

Because of the GNSS's low characteristic frequency of approximately 1 Hz and its susceptibility to instability caused by atmospheric and environmental conditions, the navigation algorithm also utilizes motor encoders with a frequency of 20 Hz. The higher frequency of the encoders allows for the calculation of odometry, used to update the geographical position measurement and heading.

Regarding the safety system, the navigation system is equipped with two sonar strings for detecting obstacles and objects within a maximum distance of 4 m. Each string consists of four sonars positioned at the front and rear of the carts. If a sonar string detects an obstacle, the navigation task stops automatically. Moreover, according to [33,18], the maximum speed of the H-Carts is set to 1 km/h to ensure safety conditions.

In Algorithm 1 a pseudocode of the navigation algorithm is provided. For simplicity, the safety aspect and the specification of the callback functions have been omitted.

The navigation algorithm begins by defining the input goal **B** as the desired geographical position, or waypoint, and recording the initial geographical position **A**. Then, the path planning computes the trajectory **AB** to follow. During navigation, the current position **C** and the heading error are updated. Since the H-Cart lacks steering wheels and relies on torque vectoring for maneuvering, it possesses a minimum turning radius. Thus, if the H-Cart encounters a heading error, it may struggle to reach the goal. In such situations, the H-Cart alternates between backward and forward driving every 5 sec. to reduce the error until it falls below a specified threshold. Nevertheless, it's important to consider that H-Carts are designed to navigate along the edges of crop fields, which are typically straight or may have slight deviations. Therefore, H-Cart navigation should proceed without the need for starting and stopping maneuvers.

In relation to the navigation task, the power and energy consumption of each power train motor are depicted in Fig. 9 considering a traveled distance of 10 m, with the H-Cart speed maintained at 1 km/h

Algorithm 1: Navigation Task.

```
Initialize the Action Function
The Goal B := (λ_B, φ_B) has been Received
Register the current position A by the GNSS
Define the AB path and tilt angle
while the goal is not reached do
    if the GNSS signal is accurate then
        Refresh the current position C through the GNSS
    Refresh C through the odometry
    Refresh AC distance and angle
    Calculate the Heading Error and the Distance Error
    if Heading Error > Heading Threshold then
        Stop the H-Cart and set the moving direction to backward
        while Heading Error > Heading Threshold do
            move backward(forward) and curve for 5 sec.
            Stop
            Refresh AC distance and angle
            Calculate the Heading Error and the Distance Error
            change the moving direction
    else if Distance Error > Distance Threshold then
        move the H-Cart to the goal
    else if The H-Cart is moving then
        Stop the H-Cart
    else
        The goal is reached!
```

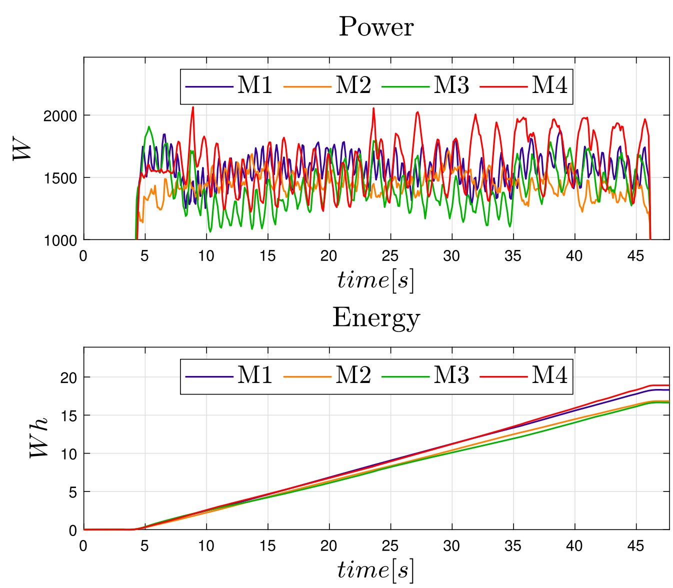

Fig. 9. Power and energy consumption of the H-Cart power train motors for a distance of 10 m.

through a Proportional Integral (PI) controller. It can be observed that each motor consumes approximately 17.5 Wh of energy. Therefore, considering a more realistic scenario where the distance to be traveled is reduced to 3 m, the motor consumption decreases to approximately 5.25 Wh, and the travel time is shortened to 12 sec.

4.2. H-cart lifter

On each H-Cart, two inductive proximity sensors are installed to identify the status of the L-Cart: lifted up, lifted down, or in between. The addition of information about the motor's state (lifting up, lifting

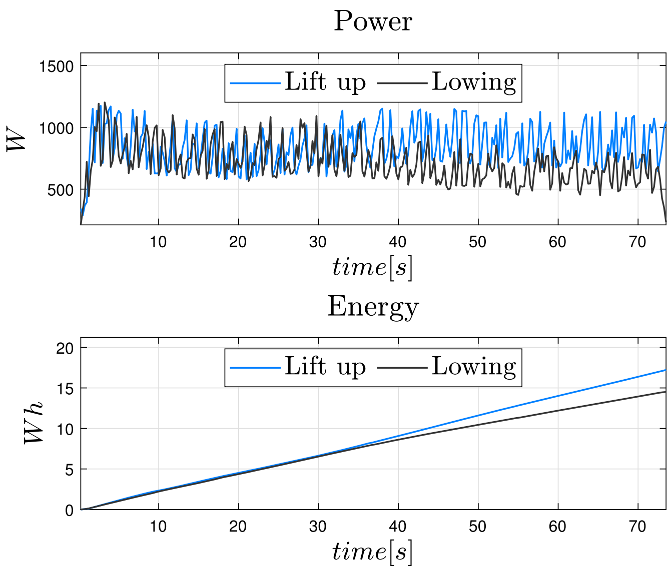

Fig. 10. Power and energy required by the lifter to raise and lower the L-Cart.

down) helps alert potential malfunctions or faults in the sensors and motors, thereby preventing possible accidents.

In Fig. 10, the graphs display the power and energy consumption of the lifter's motor for raising and lowering the L-Cart, considering a lifting distance of 68 cm. The energy required to lift the carriage (17.5 Wh) is approximately 14% higher than that needed to lower it (15 Wh). As shown in the figure, in both cases, the motor starting exhibits a peak power of about 1200 W.

4.3. L-cart driving

Once the H-Cart places the L-Cart on the row it must follow, the H-Cart positioned on the opposite side is configured to drag the L-Cart, while the nearest one is set to release the rope. L-Cart driving is executed through a PID controller, which takes as input the distances between the L-Cart and H-Carts measured by the laser sensors installed near the H-Cart lifters.

A load cell is mounted on each side of the L-Cart, in series with a spring, for monitoring the rope tension. The springs allow for a tolerance of 50 cm, preventing excessive strain on both the L-Cart and the winches. An encoder is installed on the winch motor to calculate the amount of rope released by each winch, ensuring it is neither too loose nor too tight by adjusting the PID output.

Lastly, the pair of sonars mounted on the lifter are configured to immediately deactivate the winch's drag mode through the switch relay by-passing the PID controller, if they measure a distance less than 60 cm.

A safety system is implemented to halt L-Cart Driving in case of the following events or anomalies:

- Presence of an obstacle, detected when the sum of the two laser output distances is lower than the distance between the H-Carts calculated through the H-Carts' GNSS.
- Misalignment of the L-Cart with respect to the row it is supposed to traverse, detected when the sum of the two laser output distances is greater than the distance between the H-Carts calculated through the H-Carts' GNSS.
- Malfunction of a sensor, detected when one (or both) of the rope length measures is lower than its respective laser output distance or if any sensor stops transmitting.
- Excessive release of a rope that could compromise or damage the crop, detected when one (or both) of the rope length measures is greater than its respective laser output distance multiplied by a given tolerance factor.
- Dangerously high tension in the rope, detected when one (or both) load cells register a force exceeding a given safety threshold.

When the safety system is activated, the winches are automatically set to release mode for 2 sec. to prevent any potential damage or risk.

In Algorithm 2, a pseudocode of the L-Cart driving algorithm is provided. For simplicity, safety measures and specifications of callback functions have been omitted.

The function begins by reading the input goal, which can be either *H-Cart 0* (Coordinator) or *H-Cart 1* (Participant). Then, it computes the relative distances between the H-Carts and the L-Cart, and sends this information to the respective H-Carts. Once the ropes are properly tensioned, navigation can commence. At this point, the PID parameters, previously set to zero for safety precautions, are transmitted to the H-Carts, enabling the winches to initiate movement.

Depending on the PA activity to be implemented, this task can be easily enhanced by defining an array of geographical positions the L-Cart must reach along each row, along with corresponding pauses. In this case, the task can be modified to take advantage of the L-Cart's GNSS information.

Algorithm 2: L-Cart driving Task.

```
Initialize the Action Function
The L-CartGoal := (H − Cart 0 || H − Cart 1) has been Received
Compute and send to the H-Carts their distance to the goal
if at least one of the ropes is slack then
    if L-CartGoal = H − Cart 0 then
        Let H-Cart 0 wind his rope until is reached the minimum
          tension required
    else
        Let H-Cart 1 wind his rope until is reached the minimum
          tension required
if the ropes have the correct tension then
    if L-CartGoal = H − Cart 0 then
        Assign drag PID controller parameters to H-Cart 0
        Assign release PID controller parameters to H-Cart 1
    else
        Assign drag PID controller parameters to H-Cart 1
        Assign release PID controller parameters to H-Cart 0
while L-Cart navigation is in safety state do
    check safety state
    refresh distance error
    if distance error < distance threshold then
        Assign PID controller parameters to zero for both H-Carts
        if L-CartGoal = H − Cart 0 then
            Release H-Cart 0 rope for one second
        else
            Release H-Cart 1 rope for one second
        The goal is reached!
```

Fig. 11 illustrates the power and energy consumption of the L-Cart driving task over a 3 m path. The total energy consumption is approximately 1 Wh, with the L-Cart maintaining an average speed of around 1.2 km/h. Considering an extended distance of 100 m, which is the maximum distance that can be set between the two H-Carts, the consumption would increase by approximately 34 Wh, with the time interval extending to 300 sec.

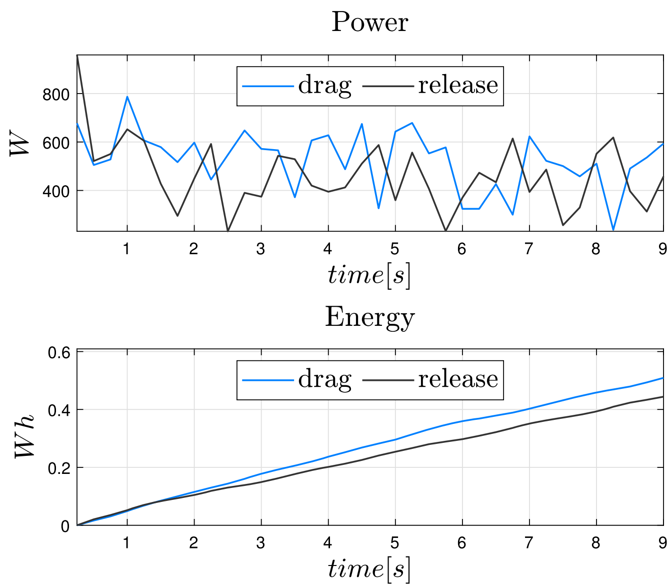

Fig. 11. L-Cart driving task power and energy consumption of both winches, considering a distance of 3 m.

4.4. Overall energy balance evaluation

Considering the energy consumption of each task outlined in Sec. 4.1, 4.2, and 4.3, and assuming a working cycle where the L-Cart driving task covers a distance of 100 m while the H-Carts' path is set to 3 m, the total energy consumption for one working cycle is calculated as follows:

$$5.25 \times 4 + 17.5 + 15 + 34 = 87.5 \text{ Wh} \quad (1)$$

The time required to complete one cycle is approximately:

$$12 + 75 \times 2 + 300 = 462 \text{ sec.} \quad (2)$$

Furthermore, the rooftop PV modules, with a power capacity of 335 W, actively contribute to charging the power train battery pack. Therefore, considering that the system operates for 12 hours during daylight hours, with a utilization factor of 0.2 (which increases to 0.4 after excluding the nocturnal hours), the average energy production in a working cycle amounts to approximately:

$$(2 \times 335) \cdot 0.4 \cdot 462/3600 = 34.4 \text{ Wh} \quad (3)$$

Consequently, the energy consumption for the task cycle reduces by 41% to 53.1 Wh.

5. Supervisor and control system

The ICT architecture, namely the supervisor and control system of the agribot, has four layers: the supervisor system, the leader control system, the follower control system and finally actuator devices and sensors. The system architecture is illustrated in Fig. 12. Specifically, the figure is referred to the H-Cart architecture identified as the *Coordinator* as it is equipped with the supervisor. The other H-Cart named *Participant* presents an identical architecture, with the exception of the supervisor. The coordinator supervisor is responsible for managing both H-Carts and L-Cart tasks (see Sec. 2). It communicates with the leader controller boards of each sub component which are configured to receive and elaborate the supervisor commands as well as communicate the component's current state (see Sec. 3.2). Moreover, the leader controller boards forward the commands to each assigned follower controller, receive their feedback, and the sensor outputs. In the scheme are specified the communication protocols adopted for interfacing each layer, whose components and lines were previously introduced in Sec. 3.2 in accordance with Fig. 8.

Below, layers are examined in a bottom-up approach, explaining the rationale behind the adopted configuration.

5.1. Control systems

The agribot is equipped with multiple follower controller boards to independently transmit information from sensors and/or to actuators, as well as exchange information with the leader board. This involves selecting the most appropriate transmission frequency for each set of sensors and control algorithms mounted and embedded in the same board. Moreover, the continuous frequency of message publication enables the observation of controller halts, triggering an outage alert through the implementation of a dedicated watchdog function. The alert is then communicated to both leader and supervisor boards.

Following are described the follower controller boards functionalities illustrated in Fig. 12 below the leader controller. Starting from the left one in figure:

1) The controller board referred to the inverters driver which communicates via CAN to the leader the measures of the motor encoder, the current, the voltage, and the inverter state that notifies if it is operative or not. Its frequency is set to 20 Hz;
2) The controller board dedicated to the laser sensor. Its is set to 10 Hz because of the frequency of the control loop used for monitoring the L-Cart position is of 4 Hz;
3) The controller board which actuates the lifter command given by the leader, stops the lifter in case of a proximity sensor, warns the arrival to the goal, and automatically opens the winch circuit in case the sonar mounted on the lifter alerts the proximity of the L-Cart. Its frequency is set to 7 Hz;
4) The controller board dedicated to the lecture of the winch encoder measure. Its frequency is set to 4 Hz;
5) The *winch motor* board has an embedded PID control algorithm for regulating the winch motor current. It receives in input the current desired from the leader with 20 Hz frequency and sends a pwm command to the chopper with 100 Hz frequency. Moreover, the board reads and sends to the leader board the winch motor current and voltage measures at a frequency of 100 Hz;
6) the *H-Cart obstacle detection* board has an embedded algorithm which implements geometric triangulation model to detect an obstacle presence in proximity of the H-Cart front and rear sides, and calculates its position. Its frequency can't be higher than 5 Hz because of the 50 ms time delay imposed by each HC-SR04 sonar [1].

Regarding the leader controller, it is programmed through an Object Oriented Programming (OOP) in order to distinguish every H-Cart power system and device. In case of safety compromising, the leader controller must immediately communicate a safety message informing the supervisor why the action has been arrested (*e.g.*, for the presence of an obstacle, sensor malfunction). However, in more complex cases where more information is involved, the controller board simply informs the supervisor of the action interruption, transferring the evaluation and interpretation of such an event to the supervisor. Lastly, the messages transmitted by the supervisor to the controller are communicated asynchronously and are read by a dedicated callback function.

5.2. Supervisor

The supervisor is designed to query and exchange information with all subsystems. It interfaces with the user through the Human Machine Interface (HMI) and the leader controller board, performing the following functions:

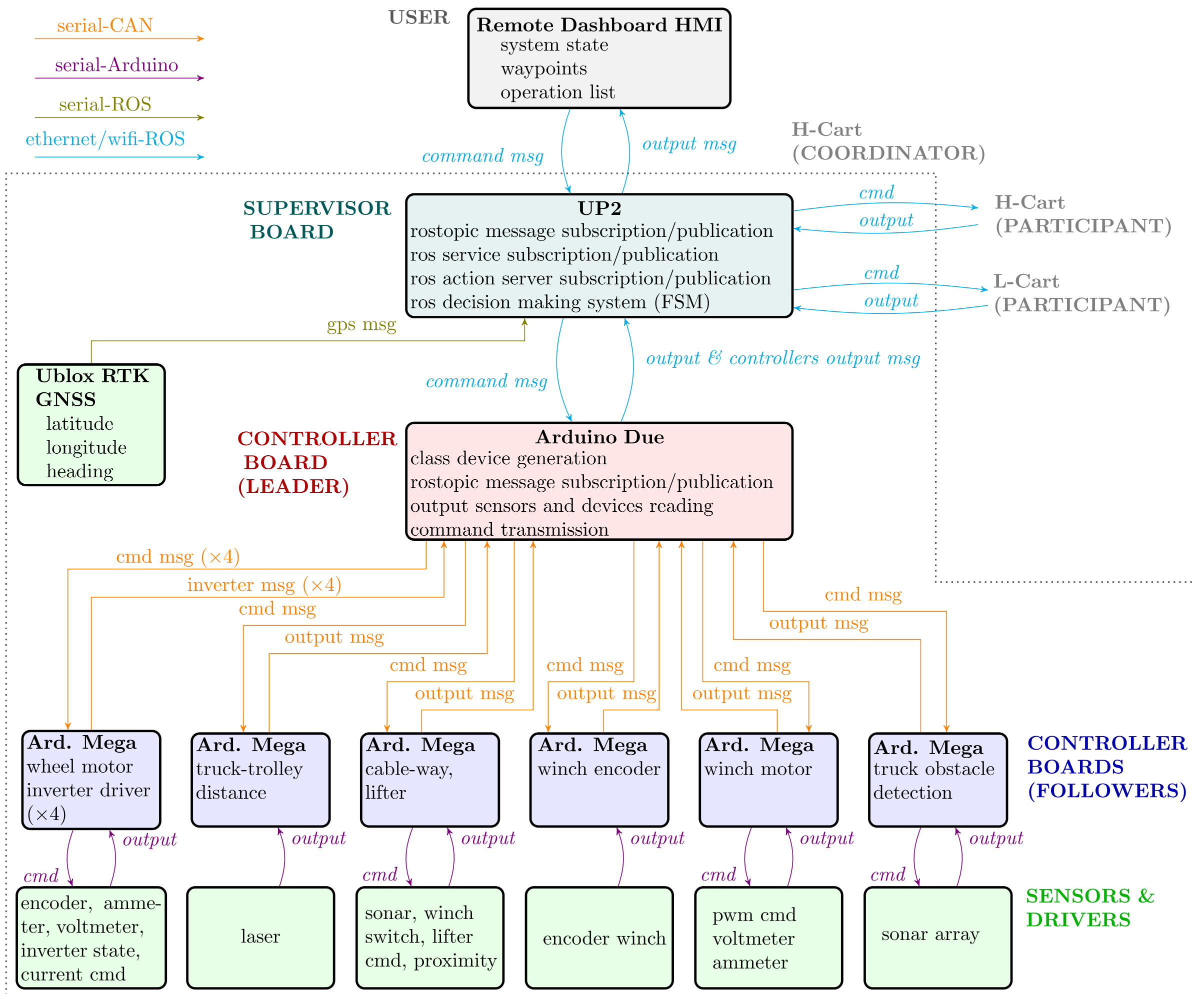

Fig. 12. Supervisor and control system architecture. The Figure illustrates the system hierarchy, functionalities and communications protocols adopted.

- read the information transmitted by the user and communicated through the HMI. The user needs to inform the supervisor about the rows that the L-Cart needs to cover by specifying the geographical waypoints where the H-Carts must stop and place the L-Cart down;
- run a decision-making system in order to start and coordinate each task introduced in Sec. 2 and detailed in Sec. 4, according to the following sequences: place the L-Cart (**lift down**), drive the L-Cart to the H-Cart on the opposite side (**L-Cart driving**), lift up the L-Cart (**lift up**), read the next way-point and let the H-Carts navigate there (**H-Cart navigation**). This routine must be repeated until all waypoints are reached by the H-Carts and thus all rows are run across by the L-Cart.
- ensure the system is working safely and without any system malfunction. In case one of them is compromised, stop every action in execution.
- communicate the current status of the system to the user, namely the decision-making system state, output set and input set.

5.3. Decision-making system

The supervisor manages and monitors the agribot tasks by utilizing FSM. A generalized framework of the FSM, illustrated in Fig. 13, demonstrates its conceptualization and organization, facilitating the straightforward modification, addition, or removal of tasks.

As can be observed in figure, each task is defined by two states: one that indicates its starting point and another its execution. Therefore, once a task successfully terminates, the FSM transients to the starting of the next task. On the contrary, if a task is in progress but the safety conditions are no longer met or a system malfunction is detected (denoted by the *safety nv* input), the FSM must interrupt the ongoing action and transition to the *stop* state. Subsequently, if the safety conditions are restored (denoted by *safety v* input), the FSM returns to the starting state of the interrupted task. In case a software error occurs or the operator decides to terminate the FSM prematurely (*error v*), the FSM stops and concludes without completing the mission. Every time the predefined sequences of tasks are successfully completed, whose event is identified in figure by the term *success and continue*, the system is ready to start a new routine in case a new couple of waypoints, specifically a new crop field row, needs to be fulfilled. In case all waypoints are fulfilled, which means the path planning is successfully completed, then the FSM reaches the *end* state.

The FSM input set is defined as follows. A discrete number refers to the advancement of task states, which can assume 4 values: *not started*, *in progress*, *accomplished*, *default*. The first value indicates that the task has not started yet, the second indicates that it is currently being executed, the third indicates if it is the last task that has been successfully completed, and finally, the default value appears in case none of the other states is verified. This approach allows to easily track the most recent task accomplished and thus identify the task to be restarted once the safety conditions are restored. Moreover, two additional inputs indicate whether safety conditions are fulfilled (or if a system malfunction event occurs), and whether an error has arisen (or an external quit command is received). These inputs consist of binary values that denote whether the conditions are *verified* or *not verified*. Finally, the last in-

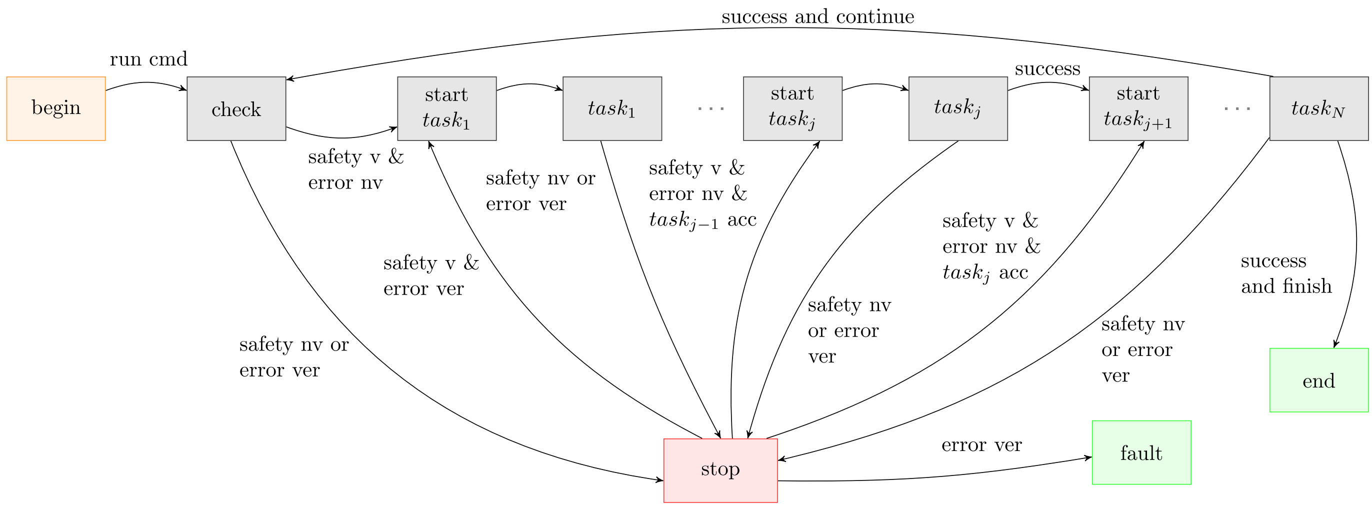

Fig. 13. Diagram illustrating the FSM logic for the agribot tasks management and coordination. The orange block represents the initial starting state, while the green blocks indicate the termination points of the process. The execution of tasks is depicted by the grey blocks. Lastly, the red block signifies the idle and waiting state.

put corresponds to an external command for initiating the FSM, it is transmitted alongside the set of waypoints to be covered.

6. Introducing ROS for supervisor implementation

The supervisor has been written in cpp and relies on ROS, an open-source robotic middleware for the large scale development of complex robotic systems. ROS development consists in a collection of software libraries and tools for creating robotic applications, from drivers to cutting edge algorithms and powerful development tools. ROS is not just a simulation environment but a complex open source platform, a middleware based on an anonymous publish/subscribe mechanism (server/client) that enables message passing between different ROS processes and also incorporates real time operational capabilities.

When programs are generated, they are identified by ROS as nodes. Upon starting, each program communicates with a master program called ROS *Master*. Each node communicates its information, including the types of data it will send or receive, to the ROS Master. Nodes that transmit data are called Publisher Nodes, while those that receive data are called Subscriber Nodes. The ROS Master maintains knowledge of all the publisher and subscriber information running on the computer. For example, if a node sends specific data of a certain type and the same data is requested by another node, the ROS Master can establish a connection between the two nodes to initiate a dialogue. The exchange of data between nodes, namely the nodes interactions or connections, is specified with the term interfaces. ROS nodes can exchange various types of data, including primitive types such as integers, floats, and strings. The different types of data that can be sent are called ROS Messages. With ROS messages, it is possible to send a single type of data or sets of data composed of different types. Messages containing measures and generic data are usually transmitted through paths called ROS topics.

Although the publish/subscribe model, over which the ROS topics are based, is a very flexible communication paradigm, its many-to-many one-way transport is not appropriate for Remote Procedure Call (RPC)[4] request/reply interactions, which are often required in a distributed system. For this reason, in ROS exist three types of interfaces: topics, services, and Action Servers (ASs) [36,19].

- Topic Message: it is an unidirectional, asynchronous, strongly typed, communication channel used in the publish-subscribe mechanism. Topic messages are ideal for continuous data streams (sensor data, robot state, ...), since data might be published and subscribed at any time independently of any senders/receivers. Concerning the topic reading, subscribers use callback functions to read the data from their assigned topic whenever it is published, allowing them to update internal variables within their nodes. On the other hand, the publisher decides when data are sent. The topic message is used to describe robot state variables (*e.g.*, velocity, position, orientation), transmit sensor values, and generate command messages sent to control boards, which must subscribe to that topic to read the value and respond accordingly. The format of topics can vary, including standard types such as arrays of integers, booleans, or RTCM type (see Sec. 4.1), as well as twist and odometry messages. The latter two are used to convey speed commands to a generic robot or to communicate its velocity in a 3D space, respectively. Additionally, it is possible to customize the message type within the topic to meet specific user requirements.
- Service: unlike a topic, the service publisher after sending the message checks if there is a client listening and if it received the message correctly. It is a request/reply interface between nodes, which is defined by a pair of messages: one for the request and one for the reply. It works as RPC that terminates quickly, *e.g.* for querying the state of a node or doing a quick calculation. They should not be used for longer-running processes, especially those that might be preempted due to exceptional situations.
- AS: if the service takes a long time to execute, the user might want the ability to cancel the request during execution or get periodic feedback about how the request is progressing. It is a higher-level mechanism built on top of topics and services for long-lasting or preemptable tasks with intermediate feedback to the caller. AS provides tools to create servers that execute long-running goals, which can be preempted. It also provides a client interface in order to send requests to the server. An AS is not a message itself, but a program written with OOP that defines an action to be performed. The AS uses a specific structured message defined by multiple topics to initiate the action and subsequently monitor its status. Specifically, the message of the AS is defined by three predefined topics:
  – goal, which defines the objective the action (*i.e.*, the AS) should pursue;
  – feedback, which updates the client on the state of the AS;

[4] Here RPC is meant as a form of client–server interaction (caller is client, executor is server), typically implemented via a request–response message-passing system.

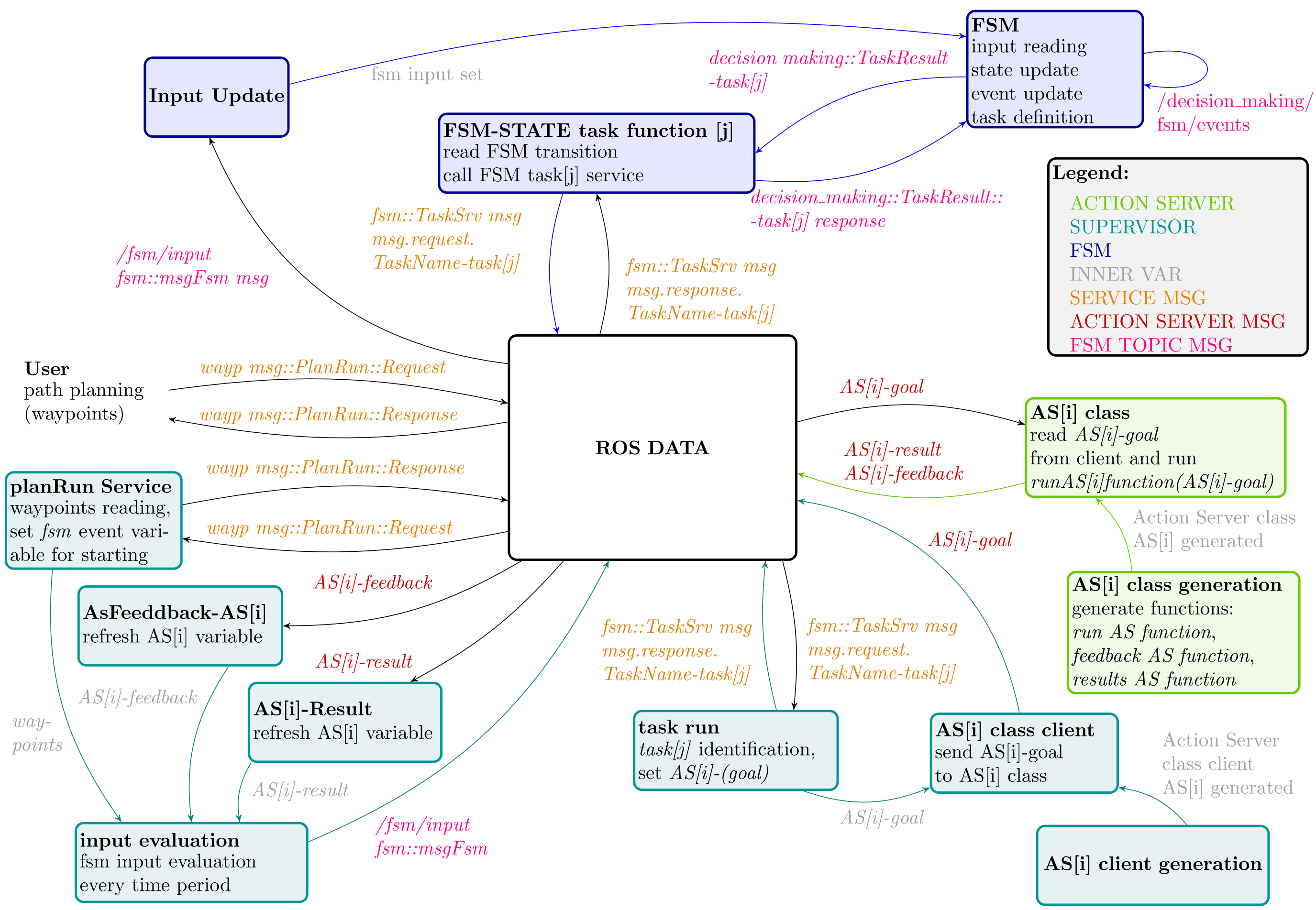

Fig. 14. Software architecture of the supervisor system, depicted in blocks. The legend in the upper right corner specifies the affiliation of each block to the corresponding software part and the type of exchanged messages.

– result, which communicates whether the action has been successfully completed, defining the final state of the AS once finished.

While the feedback and result are read by the client through dedicated callback functions, the goal is published by the client itself, thus initiating the AS, which remains in a listening state. Goal, feedback, and result need to be specifically coded to best represent the action associated with the AS. ASs should be utilized for any discrete behavior that involves moving a robot or runs for an extended period, providing feedback during execution. Multiple action goals can be executed concurrently on the same server. Each client can maintain a separate state instance for each goal, as each goal is uniquely identified by its ID.

In this project ROS topic messages are mainly used to communicate commands and measurements. ASs are utilized for task execution and monitoring, while services are employed to verify the correct initiation of a task and to transmit coordinate waypoints for path planning execution.

7. Supervisor software architecture

The software of the system can be summarized in three main parts:

- the AS classes: they receive the goal from the supervisor via the corresponding AS client and inform it about the current state of the corresponding task;
- the FSM: it determines the task to be executed by reading the input data transmitted by the supervisor;
- the supervisor: it interfaces with the user needs, updates the FSM input array, runs the tasks assigned by the FSM through their corresponding AS clients, monitors if the AS is interrupted or successfully ends and evaluates whether the safety conditions have been restored after its compromise was signaled by an AS interruption.

Fig. 14 displays a representation of the software scheme, highlighting each part. These are defined by a few scripts, distinguished in blocks of the same color. The figure also details the block interfaces, specified by the legend, which highlights the type of messages exchanged among the blocks. For the sake of simplicity, the communication with the leader controller boards, as well as the subscription and publication of the topic messages, are not illustrated, except for the topic messages concerning the FSM input set. The FSM is generated through the ROS library *decision_making*, which takes advantage of special service messages for task activation, internal communications, and the communication with the supervisor. The other blocks utilize local variables for internal communication, while communication among external blocks occurs via services and ASs.

The blocks in Fig. 14 are described below.

The **planRunService** reads the waypoints provided by the user. The supervisor function **input evaluation** updates the FSM input set every 3 sec. by monitoring the state of the ASs and the presence of the waypoints to be reached. Additionally, the function raises alerts in case of unsafe system operation, system malfunction, software er-

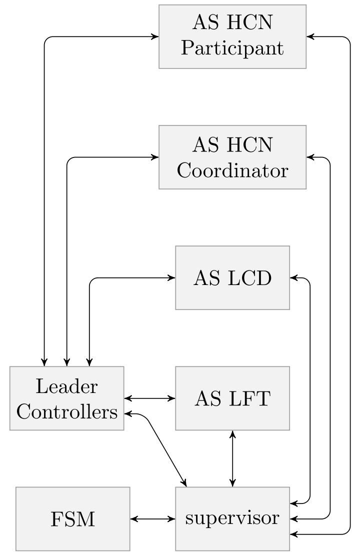

Fig. 15. Generated *ros node* graph specifying the node connection. In figure, HCN stands for the H-Carts Navigation, LCD for L-Cart Driving and LFT refers to the lifters tasks.

ror or stop command, monitoring the carts states by subscribing to the topic messages published by their respective leader controller boards. As shown in the figure, the output set of **input evaluation** is then transmitted as input to the **FSM** through the **Input Update** function. Subsequently, the FSM updates its state and, in case of a state transition, sends a service request to the **task run** function specifying the task to be performed. For example, if the FSM transients to the task related to AS^i, the **task run** function activates communicating the AS goal to the AS^i client function block **AS [i] -class client**, which then forwards the message to its corresponding AS class, denoted as **AS [i] -class**. The AS^i client function is generated by its corresponding **AS [i] client generation** function, while the AS^i class by **AS [i] class generation**. After AS^i starts running, the supervisor functions **AS [i] -Result** and **AS [i] -Feedback** monitor and update the state of AS^i, relaying information to **input evaluation** about the task's progress (feedback), and whether it has successfully completed or been aborted (result).

In Fig. 15 are detailed the nodes generated in ROS which distinguishes the ASs, the supervisor, the FSM, while the leader controllers of the three carts are all grouped in one single node. As shown, the supervisor is the only node that communicates to all the other nodes.

Fig. 16 details all topic message exchanges between nodes of the previous figure, including the ASs messages (goal, feedback e results). As depicted, only the supervisor node collects (almost all) the topics published by the other nodes, directly communicating with them. The ASs nodes are limited to publish the command related to their corresponding electrical machines (*.../cmd_winch*, *.../cmd_vel*, *.../cmd_lift*), the electric machines control system setup (*.../pid_param*), and the AS messages. Instead, the leader controller node reads the control systems setup (*.../pid_param*), the command published by the AS nodes, and some control input variables, namely the H-Carts GNSS measurements, the load-cells values and the cable-way system state (*.../controller_lcart*).

It is important to specify that the load cell measurements are transmitted by the L-Cart leader board to both H-Cart boards, and the supervisor node. For this reason, the *loadcell* topic originates from and is received by the Leader Controllers node since it represent all three leader controllers.

Finally, the figure highlights that the FSM is featured by one input message from the supervisor, *fsm/input* and one internal *decision_making* topic referred to the occurred event. It is used by the operator to monitor the process. Indeed, the FSM communicates to the supervisor via ROS service due to the need to verify the receiving of the message and ensure a certain level of robustness in sending data (see Fig. 14).

In Fig. 13 a diagram of the FSM and the task execution through the ASs activation for the L-Cart path planning is shown. The figure details all state transitions, focusing on when the process ends and in which cases the agribot stops respecting the generalized scheme and the operating principles introduced in Sec. 5.2. Concerning the ASs blocks, depicted in blue, in brackets are defined their respective goals. Specifically, in the figure, the number 0 refers to the coordinator H-Cart while 1 to the participant. Therefore, **LCD 01** refers to the L-Cart driving from the H-Cart coordinator to the participant. **LFT UP0** to the lift up of the only coordinator H-Cart. Similarly, the input *lcd01* informs that the L-Cart must be driven from the H-Cart coordinator to the participant. For the sake of completeness, in Appendix A is illustrated the state transition table referred to the FSM in Fig. 17.

The supervisor system is available in https://gitlab.com/StefanoLeonori/SPOS-Agribot.git. A ROS topics simulating program is available to test and run the software.

8. Conclusions

This paper presents a novel agribot consisting of three carts: two heavy ones equipped with a powertrain system enabling autonomous navigation, and an operating lightweight cart designed to perform precision agriculture activities on the land. The heavy carts, positioned along the edges of the crop field, are responsible for the lightweight cart's path planning, placing, and driving it along the designated crop rows through a cable-way system. The use of heavy carts as support platforms allows for an extended autonomy through the hosting of a large battery pack and the installation of rooftop PV panels. The adoption of the cable-way system would enable a significant reduction in energy consumption for path planning, ground compaction, and emissions in general. However, the cable-way system also implies that the agribot is restricted to operate on flat terrains with low slopes and low-height crops. The system discussed presents a high level of safety, with the lightweight cart's low speeds, as well as the heavy carts designed to drive slowly for short distances and periods at the edges of the crop field, spending most of their time in an idle mode. This feature makes the agribot suitable for col-

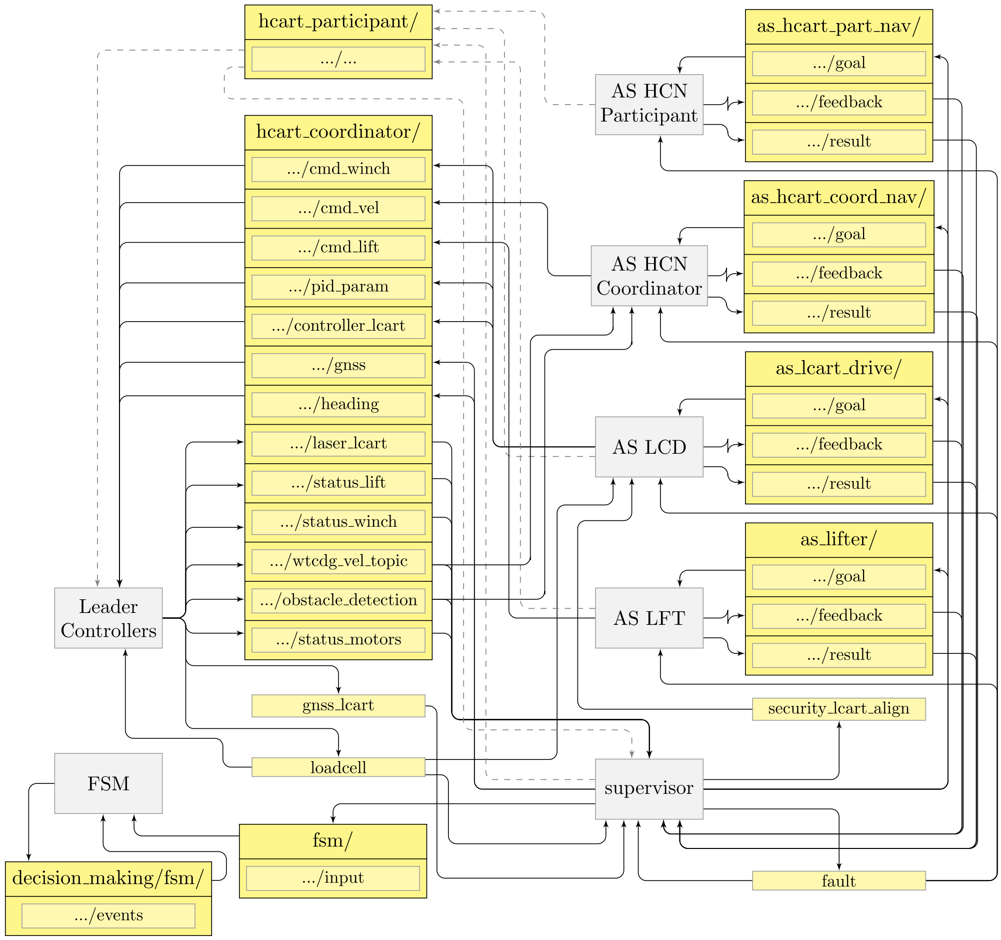

Fig. 16. Graph nodes, topic and AS interfaces.

laborative work with human operators. The paper provides a comprehensive illustration of the system's composition, focusing on the power systems, electric grid of the heavy carts, and the ICT network, including the control and supervisor systems. Moreover, a study on the energy consumption demonstrates the effectiveness of the designed system. The second part of the manuscript focuses on the detailed explanation of the control system architecture and supervisor software, both of which rely on ROS. The modularity of the decision-making system, based on a FSM, offers adaptability to various operational activities and customization depending on the user needs.

Future works will focus on upgrading the lightweight machine by incorporating a robotic arm assisted by a camera, which would enable the performance of monitoring and crop production operations. An adaptation to a collaborative mode must also be investigated, as the design and low speeds of the carts seem to fit well, especially for harvesting activities. Finally, efforts will also focus on reducing weight and dimensions to facilitate the carts' maneuvering.

CRediT authorship contribution statement

Stefano Leonori: Conceptualization, Data curation, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Stefano Mattei:** Formal analysis, Software, Writing – original draft, Writing – review & editing. **Luigi Anniballi:** Conceptualization, Methodology, Resources, Software, Writing – review & editing. **Fabio Massimo Frattale Mascioli:** Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

The project discussed in this manuscript has been financed by the Italian Ministry of Economic Development and by the European Union—NextGenerationEU (National Sustainable Mobility Center CN00000023, Italian Ministry of University and Research Decree n. 1033—17/06/2022, Spoke 9). The tests were conducted at the Po.Mo.S. laboratory of Sapienza University. A special thanks to Eng. Massimo Dore, who helped our group in the supervisor development and ROS programming.

Appendix A. FSM table

In Table A.2 the FSM is illustrated reporting each state transition.

Table A.2
FSM state transition table.

	BGN	CHK	st. LFT DWN	LFT DWN	st. LCD 01	st. LCD 10	st. HCN	HCN	LCD 01	LCD 10	st. LFT UP0	st. LFT UP1	LFT UP0	LFT UP1	STOP	HCN ALIGN	ERROR	END
BEGIN	-	run cmd	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CHECK	-	-	safety v & error nv	-	-	-	-	-	-	-	-	-	-	-	safety nv | error v	-	-	-
st. LIFT GO DOWN	-	-	-	Ø	-	-	-	-	-	-	-	-	-	-	-	-	-	-
LIFT GO DOWN	-	-	-	-	lcd01 & lftdw acc & safety v & error nv	lcd10 & lftdw acc & safety v & error nv	-	-	-	-	-	-	-	-	safety nv | error v	-	-	-
st. L-CART DRIVING 01	-	-	-	-	-	-	-	-	Ø	-	-	-	-	-	-	-	-	-
st. L-CART DRIVING 10	-	-	-	-	-	-	-	-	-	Ø	-	-	-	-	-	-	-	-
st. H-CART NAVIGATION	-	-	-	-	-	-	-	Ø	-	-	-	-	-	-	-	-	-	-
H-CART NAVIGATION	-	hcn acc & safety v & error nv & new wayp	-	-	-	-	-	-	-	-	-	-	-	-	safety nv | error v	-	-	hcn acc & safety v & error nv & no new wayp
L-CART DRIVING 01	-	-	-	-	-	-	-	-	-	-	-	lcd01 acc & safety v & error nv	-	-	safety nv | error v	-	-	-
L-CART DRIVING 10	-	-	-	-	-	-	-	-	-	-	lcd10 acc & safety v & error nv	-	-	-	safety nv | error v	-	-	-
st. LIFT 0 GO UP	-	-	-	-	-	-	-	-	-	-	-	-	Ø	-	-	-	-	-
st. LIFT 1 GO UP	-	-	-	-	-	-	-	-	-	-	-	-	-	Ø	-	-	-	-
LIFT 0 GO UP	-	-	-	-	-	-	lftup0 acc & safety v & error nv	-	-	-	-	-	-	-	safety nv | error v	-	-	-
LIFT 1 GO UP	-	-	-	-	-	-	lftup1 acc & safety v & error nv	-	-	-	-	-	-	-	safety nv | error v	-	-	-
STOP	-	-	-	-	-	-	safety v & error nv & (lftup0 acc | lftup1 acc)	-	-	-	safety v & error nv & lcd10 acc	safety v & error nv & lcd01 acc	-	-	-	safety v & error nv & align nv & lftdw acc	error v	-
HCN ALIGN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	safety nv | error v | align acc	-	-	-
ERROR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
END PLAN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

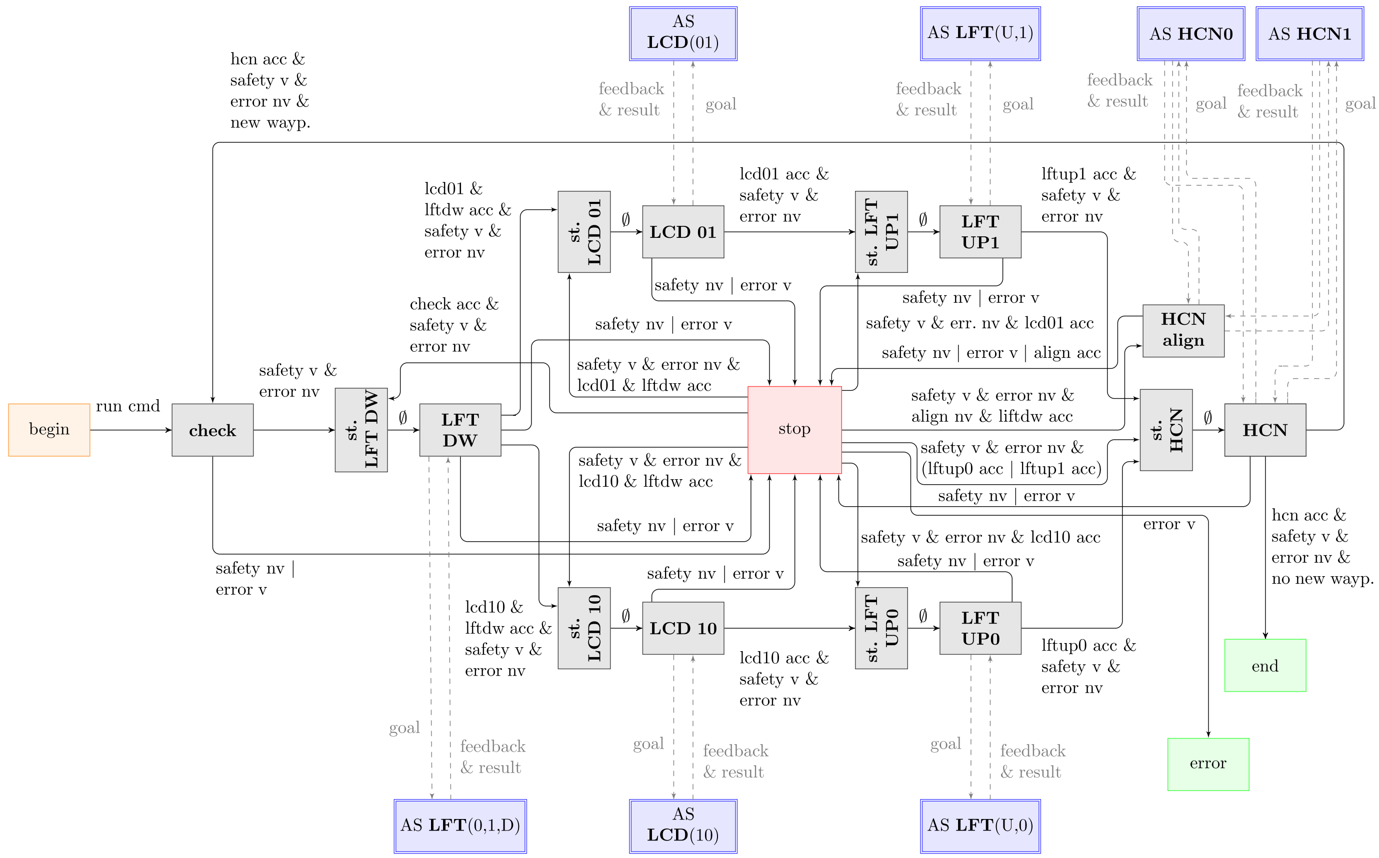

Fig. 17. The FSM and its relation with the AS. In grey the FSM state referred to the system tasks to execute. In orange, green and red the states referred to starting, the ending and the stop and wait actions. In blue are specified the ROS ASs associated to each agribot task. The number 0 refers to the coordinator H-Cart while 1 to the participant.

References

[1] A.A. Arabi, P. Sarkar, F. Ahmed, W.R. Rafie, M. Hannan, M.A. Amin, 2D mapping and vertex finding method for path planning in autonomous obstacle avoidance robotic system, in: 2017 2nd International Conference on Control and Robotics Engineering (ICCRE), 2017, pp. 39–42.

[2] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour, E.H.M. Aggoune, Internet-of-things (IoT)-based smart agriculture: toward making the fields talk, IEEE Access 7 (2019) 129551–129583, https://doi.org/10.1109/ACCESS.2019.2932609.

[3] S. Basu, A. Omotubora, M. Beeson, C. Fox, Legal framework for small autonomous agricultural robots, AI Soc. 35 (2020) 113–134.

[4] L. Benos, A. Bechar, D. Bochtis, Safety and ergonomics in human-robot interactive agricultural operations, Biosyst. Eng. 200 (2020) 55–72, https://doi.org/10.1016/j.biosystemseng.2020.09.009, https://www.sciencedirect.com/science/article/pii/S1537511020302506.

[5] L. Benos, D. Tsaopoulos, D. Bochtis, A review on ergonomics in agriculture. Part I: manual operations, Appl. Sci. 10 (2020), https://doi.org/10.3390/app10061905, https://www.mdpi.com/2076-3417/10/6/1905.

[6] L. Benos, D. Tsaopoulos, D. Bochtis, A review on ergonomics in agriculture. Part II: mechanized operations, Appl. Sci. 10 (2020), https://doi.org/10.3390/app10103484, https://www.mdpi.com/2076-3417/10/10/3484.

[7] R. Berenstein, Y. Edan, Human-robot collaborative site-specific sprayer, J. Field Robot. 34 (2017) 1519–1530, https://doi.org/10.1002/rob.21730.

[8] R. Berenstein, Y. Edan, Human-robot collaborative site-specific sprayer, J. Field Robot. 34 (2017) 1519–1530, https://doi.org/10.1002/rob.21730.

[9] A. Botta, P. Cavallone, L. Baglieri, G. Colucci, L. Tagliavini, G. Quaglia, A review of robots, perception, and tasks in precision agriculture, Appl. Mech. 3 (2022) 830–854, https://doi.org/10.3390/applmech3030049, https://www.mdpi.com/2673-3161/3/3/49.

[10] Y. Chen, Y. Huang, Z. Zhang, Z. Wang, B. Liu, C. Liu, C. Huang, S. Dong, X. Pu, F. Wan, X. Qiao, W. Qian, Plant image recognition with deep learning: a review, Comput. Electron. Agric. 212 (2023) 108072, https://doi.org/10.1016/j.compag.2023.108072, https://www.sciencedirect.com/science/article/pii/S016816992300460X.

[11] R. Esram, B.B.V.L. Deepak, U.R. Mogili, P. Syam Sundar, Agribots concepts and operations—a review, in: B.B.V.L. Deepak, D. Parhi, B. Biswal, P.C. Jena (Eds.), Applications of Computational Methods in Manufacturing and Product Design, Springer Nature Singapore, Singapore, 2022, pp. 31–40.

[12] H. Fang, S. Ong, A. Nee, A novel augmented reality-based interface for robot path planning, Int. J. Interact. Des. Manuf. 8 (2014) 33–42.

[13] F. FAO, A. Organization, Migration agriculture and rural development: addressing the root causes of migration and harnessing its potential for development, in: Migration, Agriculture and Rural Development, 2016, www.fao.org/3/a-i6064e.pdf.

[14] FAO Press Release, Food and agriculture statistics, https://www.fao.org/sustainability/news/detail/en/c/1274219/. (Accessed 30 June 2023).

[15] H.C.J. Godfray, J.R. Beddington, I.R. Crute, L. Haddad, D. Lawrence, J.F. Muir, J. Pretty, S. Robinson, S.M. Thomas, C. Toulmin, Food security: the challenge of feeding 9 billion people, Science 327 (2010) 812–818, https://doi.org/10.1126/science.1185383.

[16] T.F. Guerin, Impacts and opportunities from large-scale solar photovoltaic (PV) electricity generation on agricultural production, Environ. Qual. Manag. 28 (2019) 7–14, https://doi.org/10.1002/tqem.21629.

[17] A. Hentout, M. Aouache, A. Maoudj, I. Akli, Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017, Adv. Robot. 33 (2019) 764–799, https://doi.org/10.1080/01691864.2019.1636714.

[18] International Organization for Standardization, Robots and robotic devices - safety requirements for industrial robots - Part I: Robots, iso 10218-1:2011, 2012.

[19] A. Koubâa, et al., Robot Operating System (ROS), vol. 1, Springer, Cham, 2017.

[20] E. Lenz, et al., Networked transport of RTCM via internet protocol (NTRIP)–application and benefit in modern surveying systems, in: FIG Working Week, 2004, pp. 22–27.

[21] Y. Li, M. Iida, T. Suyama, M. Suguri, R. Masuda, Implementation of deep-learning algorithm for obstacle detection and collision avoidance for robotic harvester, Comput. Electron. Agric. 174 (2020) 105499, https://doi.org/10.1016/j.compag.2020.105499, https://www.sciencedirect.com/science/article/pii/S016816992030137X.

[22] Y. Li, M. Li, J. Qi, D. Zhou, Z. Zou, K. Liu, Detection of typical obstacles in orchards based on deep convolutional neural network, Comput. Electron. Agric. 181 (2021) 105932, https://doi.org/10.1016/j.compag.2020.105932, https://www.sciencedirect.com/science/article/pii/S0168169920331379.

[23] J. Lowenberg-DeBoer, B. Erickson, Setting the record straight on precision agriculture adoption, Agron. J. 111 (2019) 1552–1569, https://doi.org/10.2134/agronj2018.12.0779, https://acsess.onlinelibrary.wiley.com/doi/abs/10.2134/agronj2018.12.0779.

[24] C. Lytridis, V.G. Kaburlasos, T. Pachidis, M. Manios, E. Vrochidou, T. Kalampokas, S. Chatzistamatis, An overview of cooperative robotics in agriculture, Agronomy

11 (2021), https://doi.org/10.3390/agronomy11091818, https://www.mdpi.com/2073-4395/11/9/1818.
[25] C. Miskin, Y. Li, A. Perna, R. Ellis, E. Grubbs, P. Bermel, R. Agrawal, Sustainable co-production of food and solar power to relax land-use constraints, Nat. Sustain. 2 (2019) 972–980, https://doi.org/10.1038/s41893-019-0388-x.
[26] L.E. Montoya-Cavero, R. Díaz de León Torres, A. Gómez-Espinosa, J.A. Escobedo Cabello, Vision systems for harvesting robots: produce detection and localization, Comput. Electron. Agric. 192 (2022) 106562, https://doi.org/10.1016/j.compag.2021.106562, https://www.sciencedirect.com/science/article/pii/S0168169921005792.
[27] D. Mukherjee, K. Gupta, L.H. Chang, H. Najjaran, A survey of robot learning strategies for human-robot collaboration in industrial settings, Robot. Comput.-Integr. Manuf. 73 (2022) 102231, https://doi.org/10.1016/j.rcim.2021.102231, https://www.sciencedirect.com/science/article/pii/S0736584521001137.
[28] B. Murdyantoro, D. Atmaja, H. Rachmat, Application design of farmbot based on internet of things (IoT), Int. J. Adv. Sci. Eng. Inf. Technol. 9 (2019) 1163, https://doi.org/10.18517/ijaseit.9.4.9483.
[29] M. Nawaz, G. Bourrié, F. Trolard, Soil compaction impact and modelling. A review, Agron. Sustain. Dev. 33 (2013), https://doi.org/10.1007/s13593-011-0071-8.
[30] L.F.P. Oliveira, A.P. Moreira, M.F. Silva, Advances in agriculture robotics: a state-of-the-art review and challenges ahead, Robotics 10 (2021), https://doi.org/10.3390/robotics10020052, https://www.mdpi.com/2218-6581/10/2/52.
[31] A.S. Pascaris, C. Schelly, J.M. Pearce, A first investigation of agriculture sector perspectives on the opportunities and barriers for agrivoltaics, Agronomy 10 (2020) 1885, https://doi.org/10.3390/agronomy10121885.
[32] V. Rajendran, B. Debnath, S. Mghames, W. Mandil, S. Parsa, S. Parsons, A. Ghalamzan-E., Towards autonomous selective harvesting: a review of robot perception, robot design, motion planning and control, J. Field Robot., https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.22230, https://doi.org/10.1002/rob.22230.
[33] P. Ratsamee, Y. Mae, K. Kamiyama, M. Horade, M. Kojima, T. Arai, Social interactive robot navigation based on human intention analysis from face orientation and human path prediction, Robomech J. 2 (2015) 1–18.
[34] K. Robert, Q. Elisabeth, B. Josef, Analysis of occupational accidents with agricultural machinery in the period 2008–2010 in Austria, Saf. Sci. 72 (2015) 319–328, https://doi.org/10.1016/j.ssci.2014.10.004, https://www.sciencedirect.com/science/article/pii/S0925753514002367.
[35] S. Robla-Gómez, V.M. Becerra, J.R. Llata, E. González-Sarabia, C. Torre-Ferrero, J. Pérez-Oria, Working together: a review on safe human-robot collaboration in industrial environments, IEEE Access 5 (2017) 26754–26773, https://doi.org/10.1109/ACCESS.2017.2773127.
[36] Open Robotics, Topics vs services vs actions, https://docs.ros.org/en/iron/How-To-Guides/Topics-Services-Actions.html. (Accessed 30 July 2023).
[37] D.C. Rose, J. Lyon, A. de Boon, M. Hanheide, S. Pearson, Responsible development of autonomous robotics in agriculture, Nat. Food 2 (2021) 306–309.
[38] V. Sachithra, L. Subhashini, How artificial intelligence uses to achieve the agriculture sustainability: systematic review, Artif. Intell. Agric. 8 (2023) 46–59, https://doi.org/10.1016/j.aiia.2023.04.002, https://www.sciencedirect.com/science/article/pii/S2589721723000144.
[39] J. Saenz, J. Bessler-Etten, M. Valori, G.B. Prange-Lasonder, I. Fassi, C. Bidard, A.B. Lassen, I. Paniti, A. Toth, T. Stuke, S. Wrede, K. Nielsen, An online toolkit for applications featuring collaborative robots across different domains, IEEE Trans. Human-Mach. Syst. (2022) 1–11, https://doi.org/10.1109/THMS.2022.3213416.
[40] J. Schmidtler, V. Knott, C. Hoelzel, K. Bengler, Human centered assistance applications for the working environment of the future, Occup. Ergon. 12 (2015) 83–95, https://doi.org/10.3233/OER-150226.
[41] S.H. Swan, R.L. Kruse, F. Liu, D.B. Barr, E.Z. Drobnis, J.B. Redmon, C. Wang, C. Brazil, J.W. Overstreet, Semen quality in relation to biomarkers of pesticide exposure, Environ. Health Perspect. 111 (2003) 1478–1484, https://doi.org/10.1289/ehp.6417, https://ehp.niehs.nih.gov/doi/abs/10.1289/ehp.6417.
[42] A.D. Tripathi, R. Mishra, K.K. Maurya, R.B. Singh, D.W. Wilson, Chapter 1 - estimates for world population and global food availability for global health, in: R.B. Singh, R.R. Watson, T. Takahashi (Eds.), The Role of Functional Food Security in Global Health, Academic Press, 2019, pp. 3–24, https://www.sciencedirect.com/science/article/pii/B9780128131480000013.
[43] A. Tzounis, N. Katsoulas, T. Bartzanas, C. Kittas, Internet of things in agriculture, recent advances and future challenges, Biosyst. Eng. 164 (2017) 31–48, https://doi.org/10.1016/j.biosystemseng.2017.09.007, https://www.sciencedirect.com/science/article/pii/S1537511017302544.
[44] European Union, Farm to fork strategy for a fair, healthy and environmentally-friendly food system, https://ec.europa.eu/food/sites/food/files/safety/docs/f2f_action-plan_2020_strategy-info_en.pdf, 2022.
[45] J.P. Vasconez, G.A. Kantor, F.A. Auat Cheein, Human–robot interaction in agriculture: a survey and current challenges, Biosyst. Eng. 179 (2019) 35–48, https://doi.org/10.1016/j.biosystemseng.2018.12.005, https://www.sciencedirect.com/science/article/pii/S1537511017309625.
[46] M. Wakchaure, B. Patle, A. Mahindrakar, Application of AI techniques and robotics in agriculture: a review, Artif. Intell. Life Sci. 3 (2023) 100057, https://doi.org/10.1016/j.ailsci.2023.100057, https://www.sciencedirect.com/science/article/pii/S2667318523000016.
[47] World Bank, Food and agriculture statistics, agricultural land (% of land area), https://data.worldbank.org/indicator/AG.LND.AGRI.ZS. (Accessed 30 June 2023).
[48] M.O. Yerebakan, B. Hu, Human–robot collaboration in modern agriculture: a review of the current research landscape, Adv. Intell. Syst. (2024) 2300823, https://doi.org/10.1002/aisy.202300823, https://onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202300823.