electronics

MDPI

Article

A Semi-Autonomous Multi-Vehicle Architecture for Agricultural Applications

El Houssein Chouaib Harik [1,*], Frédéric Guinand [2,3] and Jakob Geipel [1]

[1] Center for Precision Agriculture, The Norwegian Institute of Bioeconomy Research (NIBIO), Nylinna 226, 2849 Kapp, Norway; jakob.geipel@nibio.no
[2] LITIS Laboratory, Normandie University UNIHAVRE, 25 Rue Philippe Lebon, 76600 Le Havre, France; frederic.guinand@univ-lehavre.fr
[3] Cardinal Stefan Wyszynski University (UKSW), Dewajtis 5, 01-815 Warszawa, Poland
* Correspondence: elhousseinchouaib.harik@nibio.no

Abstract: The ageing population, climate change, and labour shortages in the agricultural sector are driving the need to reevaluate current farming practices. To address these challenges, the deployment of robot systems can help reduce environmental footprints and increase productivity. However, convincing farmers to adopt new technologies poses difficulties, considering economic viability and ease of use. In this paper, we introduce a management system based on the Robot Operating System (ROS) that integrates heterogeneous vehicles (conventional tractors and mobile robots). The goal of the proposed work is to ease the adoption of mobile robots in an agricultural context by providing to the farmer the initial tools needed to include them alongside the conventional machinery. We provide a comprehensive overview of the system's architecture, the control laws implemented for fleet navigation within the field, the development of a user-friendly Graphical User Interface, and the charging infrastructure for the deployed vehicles. Additionally, field tests are conducted to demonstrate the effectiveness of the proposed framework.

Keywords: precision agriculture; agricultural robots; leader–follower navigation; multi-vehicle operations; robot operating system

check for updates

Citation: Harik, E.H.C.; Guinand, F.; Geipel, J. A Semi-Autonomous Multi-Vehicle Architecture for Agricultural Applications. *Electronics* **2023**, *12*, 3552. https://doi.org/10.3390/electronics12173552

Academic Editor: Felipe Jiménez

Received: 22 July 2023
Revised: 15 August 2023
Accepted: 18 August 2023
Published: 22 August 2023

1. Introduction

The mechanisation of agriculture at the beginning of the twentieth century was a real game-changer when it came to sustaining food production and increasing the yield potential of arable land. However, decades of monoculture and uniform practices contributed greatly to an increased negative impact on the environment [1,2]. This led researchers from different fields to focus their efforts on trying to find efficient solutions that make agricultural practices more sustainable and environmentally friendly [3]. A recent study [4] investigated the potential of replacing fossil-fuel-based tractors with a fleet of lighter, electrically or hydrogen-based tractors to reduce the negative impact of conventional fossil-fuel-based tractors. The study concluded that due to the current battery technology (charging time, and energy density vs weight), only 10% of current fossil-fuel-based tractors can be replaced with electric ones. However, the progress witnessed in the electric vehicle industry will definitely bring a positive impact on its agricultural counterpart, making a shift in farm machinery possible. Another study [5] outlined an alternative in the use of a fleet of smaller robots that can work autonomously. To achieve the same efficiency level as conventional tractors, there will be a need to include a swarm of robots [6] that, in a cooperative manner and around the clock, can perform the same tasks at the same efficiency level as conventional tractors.

Although the environmental impact is a key driver of the political efforts towards the adoption of different agricultural practices, the economic pressure that farmers are facing makes them prioritise financial viability over environmental concerns. Nevertheless, one of the main challenges farmers are facing today is a lack of labour [7]. The reason for this is twofold. Firstly, agriculture is not an appealing career for the younger generation [8], which makes the supply of a qualified workforce for taking over increasingly difficult [9]. Secondly, there is a significant dependency on imported labour, which has been greatly impacted by the recent pandemic and the geo-political situation [10]. This summed up, exposes the weaknesses of the established food supply system. On the other hand, the use of robots in agriculture is contributing greatly to mitigating these problems by providing alternatives to farmers to sustain their production levels [11]. Robots have been used for decades in dairy production [12], the food processing chain [13], plant [14] and animal [15] monitoring, and for field input applications [16] and livestock feeding [17].

Nevertheless, alongside the burgeoning array of solutions offered to farmers, an accompanying escalation in complexity and challenges emerges concerning the unified management of these systems. Concurrently, the task of maintaining efficient organization of daily operations becomes increasingly intricate. Farm Management Systems (FMS), as highlighted by [18], currently represent the principal organizational tool available to farmers. FMS empower farmers to oversee day-to-day tasks, optimize labor distribution, and uphold comprehensive records, facilitating the tracking of seasonal progress. However, the domain of agricultural robotics introduces a distinctive challenge; individual manufacturers supply their proprietary control and supervision software for their respective robotic systems. This scenario often results in farmers owning a multitude of robots, each designated for specific tasks, thereby necessitating the navigation of a plethora of distinct software interfaces. Nonetheless, a common thread among contemporary agricultural robot manufacturers lies in their adoption of the Robot Operating System (ROS) [19]. This underlying consistency provides a foundation for the development of cross-platform user interfaces, accommodating robots from diverse origins. However, the available literature remains nascent in addressing this particular concern. While certain web-based frameworks for robot control harness ROS [20,21], these solutions predominantly adhere to task-specific functionalities, often catering exclusively to those well-versed in the ROS environment.

The innovations of the work presented in this paper are twofold. Firstly, to the best of our knowledge, we are the only ones to propose a web-based framework to control and supervise ROS-based robots in an agricultural context. And secondly, we are the first to include conventional agricultural machinery in a heterogeneous framework to make the overall scheme more appealing, and closing the gap between research-based solutions and real-world applications.

The rest of the paper is organised as follows: Section 2 introduces the different elements of the system. The control laws that allow autonomous navigation inside the field are presented in Section 3. In Section 4, the experimental results followed by a discussion are presented. The current study is concluded in Section 5, where some guidelines are given for future possibilities to enhance the overall management system.

2. Multi-Vehicle Management System

The need to simplify the supervision and control of a heterogeneous group of vehicles/robots is the main idea behind the development of the multi-vehicle management system (MMS) presented in this paper. The tool performs the background processes, such as communication, management, and supervision, through a central server. It also provides the farmer with the ability to allocate day-to-day tasks, as well as a real-time overview of the vehicle's state through a simple-to-use Graphical User Interface (GUI). Access to the tool is available to the farmer using wireless connectivity [22] and a web browser. To this end, the initial hypothesis is that the farmer has access to at least 4G [23] in order to use the GUI.

We present in this paper the management system currently deployed and being tested at the NIBIO-Apelsvoll (Kapp, Norway) research station. It is composed of a human-driven electric tractor and an unmanned electric robot tractor for field operations, an autonomous mobile robot equipped with a robotic arm that acts as the charging infrastructure, and a central server for communication and the user experience (Figure 1).

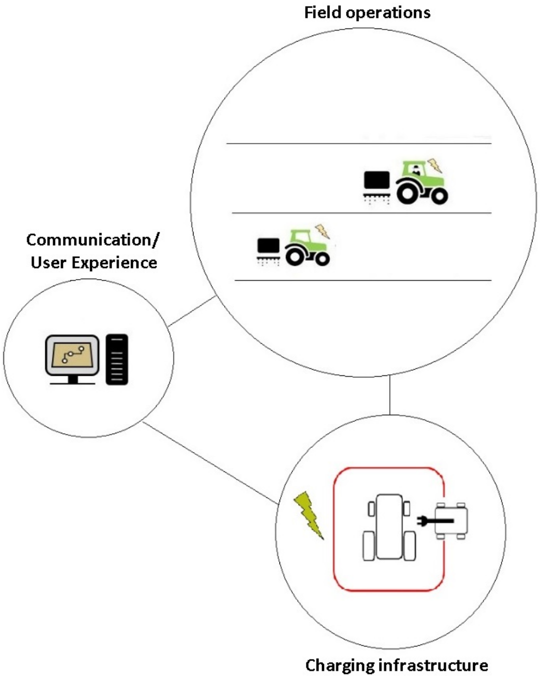


Figure 1. The overall architecture of the case study. The system covers the field operations, communication and user experience, and charging infrastructure for the electrical vehicles.

2.1. Central Server

The MMS revolves around a Central Server (CS) physically located at one of our research facility buildings. The CS is tasked with logging, supervision, and re-routing of information between different entities in the system. Nevertheless, one of the goals of the MMS is to provide a flexible framework that can include non-ROS enabled vehicles (e.g., tractors). To this end, we use Rosbridge [24] that provides this functionality and allows communication between the CS and all the entities in the system, namely the robots, tractors, and the GUI. The CS is running an HTTP Apache server [25] to host the web-based GUI which contains the different elements that allow the supervision and task allocation of the different vehicles, representing the farmer's portal to the park of machinery/robots.

2.2. Vehicle Descriptions

The vehicles used in this case study consist of an electric tractor, a robot-tractor, and a mobile charging station. The focus is on replacing heavy machinery with a fleet of lighter, electric ones. The idea behind including a manned tractor with an unmanned robot-tractor is to increase the working width by a single driver. For instance, if the tractor has a 12 m spraying boom, adding a robot-tractor with the same spraying boom width doubles the working width to 24 m, while still having only one driver. This can be achieved when the vehicles are driving in a leader (conventional tractor)-follower (robot-tractor) mode [26].

The manned vehicle (Figure 2a) is a custom-made electric tractor [27] built on a Ford 7810 tractor chassis. We equipped the tractor with an embedded computer, localisation sensors (Global Navigation Satellite System (GNSS) with Real-Time-Kinematics (RTK) enabled), an Inertial Measurement Unit (IMU), and a wireless router. The pose estimation runs on the embedded computer and is broadcast continuously via the wireless router. An illustration of the software architecture of the tractor can be seen in Figure 2b.

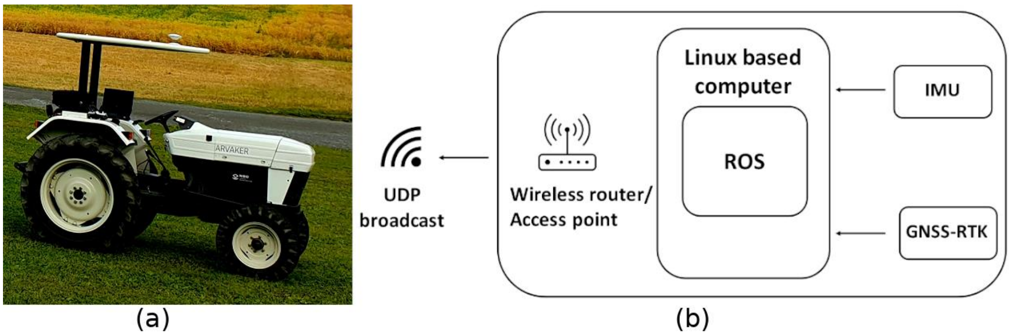


Figure 2. (**a**) The electrical tractor (manned vehicle). (**b**) The software architecture of the tractor, including the Global Navigation Satellite System (GNSS)-Real Time Kinematics (RTK) receiver (Piksi Multi-Evaluation kit, SwiftNav, CA, USA [28]), Inertial Measurement Unit (IMU) (VN-100, Vectornav, TX, USA [29]), Jetson Xavier board (Nvidia, CA, USA [30]) running a Linux distribution (Ubuntu 20.04 LTS), and wireless router (Asus RT-AC68U, Taipei, Taiwan [31]). The embedded computer is running the Robot Operating System (ROS) (Noetic) middleware to manage the data acquisition from the different sensors, as well as the pose estimation of the tractor, which is broadcasted using User Datagram Protocol (UDP) packages.

The unmanned vehicle (Figure 3a) is a differential-drive mobile robot equipped with a three-point mount system, similar to the one used in conventional tractors, which opens for multipurpose utilisation using appropriate conventional implements. The robot tractor uses the data from its onboard sensors, as well as the received pose from the tractor, as inputs to its motion controller. The results are then used to define the control signal sent to the Electronic Speed Control (ESC) of the motors for the navigation part. The robot tractor uses User Datagram Protocol (UDP) to communicate with both the tractor (receive its pose) and the charging station (send and receive charging status). An illustration of the software architecture of the robot tractor can be seen in Figure 3b.

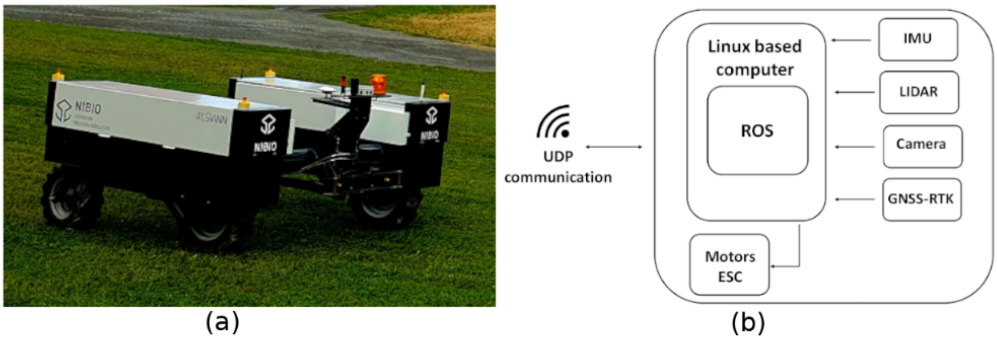


Figure 3. (**a**) The electrical robot tractor (unmanned vehicle). (**b**) The software architecture of the robot tractor. The robot tractor runs on Linux Ubuntu and ROS Noetic. It is equipped with an RTK-GNSS receiver (Piksi Multi-Evaluation kit, SwiftNav, CA, USA [28]), RGB camera (Logitech C925E, Lausanne, Switzerland [32]), and a Light Detection And Ranging (Lidar) sensor (SICK TiM571, Breisgau, Germany, [33]). The robot tractor runs the same Linux distribution and ROS version as the tractor. It is also equipped with a long-range wireless access point (PicoStation2, Ubiquiti, NY, USA [34]).

When driving to the field, the robot tractor follows the tractor in "follow me" mode (Figure 4). This means that the robot tractor drives behind the tractor on narrow roads until both vehicles reach the field. Once inside the field, the robot tractor changes the following mode to a parallel driving mode to double the working width of the human driver, as previously stated.

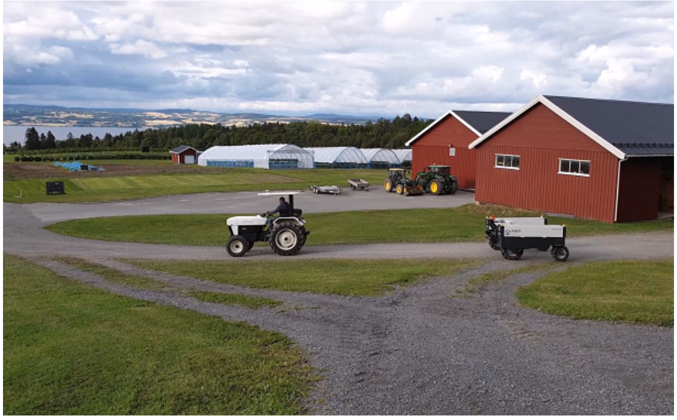


Figure 4. Leader–follower "follow-me" operation mode.

Managing the logistics of the vehicles when it comes to charging their batteries is a necessity. It is impractical for the farmer to halt a task in order to drive back to the charging station each time a robot tractor's batteries are depleted. To this end, we developed [35] an autonomous charging station (ACS) to address this issue. The ACS is mobile and can move freely within the parking area of the vehicles, allowing it to charge multiple robots. Whenever a robot tractor enters the charging zone, it communicates directly with the ACS. Upon receiving a charging request, the ACS unplugs the power cable from the wall and plugs it into the robot tractor. The robot tractor monitors its internal battery state and requests the ACS to unplug the cable when it reaches the desired power level to resume its duties in the field. Upon receiving the unplug request, the ACS performs the necessary actions to unplug the cable from the robot tractor and plug it back into the wall. An illustration of the software architecture of the autonomous charging station can be seen in Figure 5.

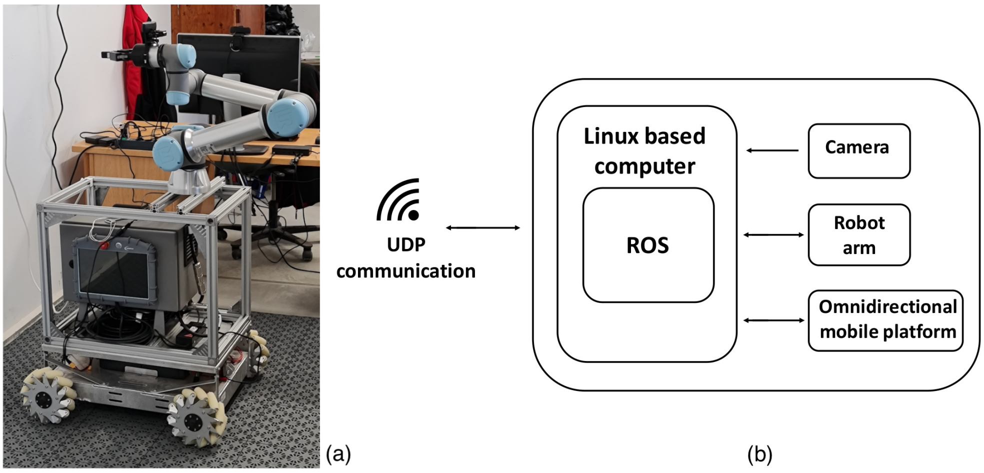


Figure 5. (**a**) The autonomous charging station [35]. (**b**) The software architecture of the autonomous charging station. The ACS is a combination of a UR5e robotic arm (Universal Robot, Odense, Denmark [36]), an omnidirectional mobile platform (Programmable Mecanum Wheel Vectoring Robot-IG52 DB, Superdroid Robots, Fuquay-Varina, NC, USA [37]), and an RGB camera (Logitech C925E, Lausanne, Switzerland [32]). The embedded computer is a Jetson TX2 (Nvidia, CA, USA [30]), running Ubuntu and the ROS.

2.3. Communication Infrastructure

The communication scheme can be divided into two levels: global and local. On one hand, the global communication scheme covers information exchanged between the different entities of the architecture (tractor, robot-tractor, charging station, and GUI) and the CS. We refer to this in the remainder of the text as m2s (machine to server) communication.

On the other hand, the local communication scheme, which we refer to in the text as m2m (machine to machine) communication, is limited to inter-vehicle communication (limited to the ground vehicles). Figure 6 illustrates the network architecture of the system.

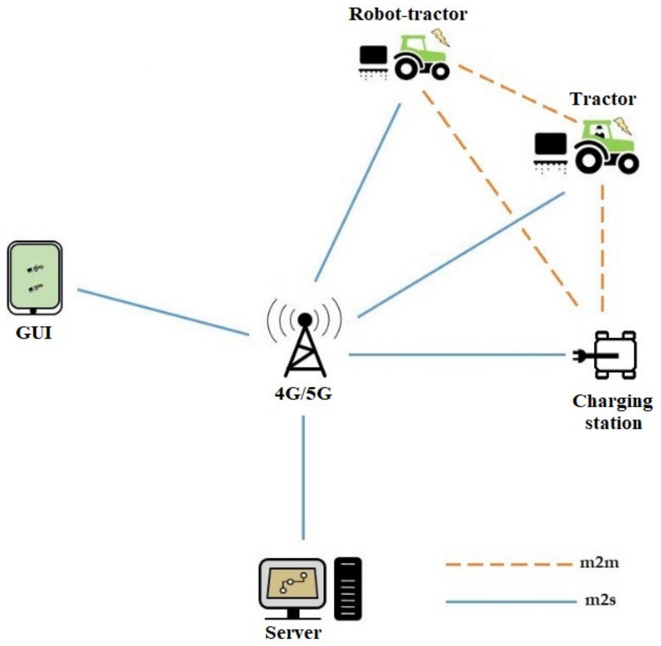


Figure 6. Communication scheme.

All vehicles in the system transmit information about their status periodically to the CS and receive task allocations in return. As a preliminary requirement, all vehicles in the system are supposed to have Internet access. For this reason, the vehicles are equipped with 4G/5G modems to transmit and receive data to and from the CS. The GUI serves as the farmer's portal to the system, allowing status checks for the vehicles and system control.

When connected to a cellular network, the device receives a private IP address from the Internet Service Provider (ISP), making it inaccessible from outside the network. Since we have different entities connected to different networks, we needed to ensure that all entities could communicate with each other. To achieve this, all devices use a Virtual Private Network (VPN), virtually placing them on the same subnet network.

2.4. Graphical User Interface

The GUI was developed keeping in mind that it can be used independently of the Operating System (OS). Since all today's smart devices and personal computers have access to a browser, a web-based solution is a suitable candidate that can fulfil this requirement. This means that the farmer can use the GUI through a portable smart device (phone or tablet), or from an office using a Personal Computer (PC) device. Additionally, the web-based solution removes the hassle of making a dedicated application for each operating system, makes maintenance easier, and enables more frequent updates.

The GUI is based on HTML5 [38], coupled with JavaScript [39] and ROSLIBjs [40] for executable nodes. The first functionality of the GUI is to show in real time the GNSS locations of the vehicles in the field. To achieve this, we used the Leaflet [41] plugin to visualise a map and overlay the vehicles and icons of other areas of interest on the map (Figure 7).

Figure 7 represents an aerial image of the fields located at the NIBIO-Apelsvoll research station. The vehicle names are *Arvaker* for the tractor and *Alsvinn* for the robot tractor. Clicking on any icon on the map opens a contextual menu showing different action possibilities. For instance, the robot tractor has different options that differ from the other vehicles (tractor or charging station) (Figure 8).

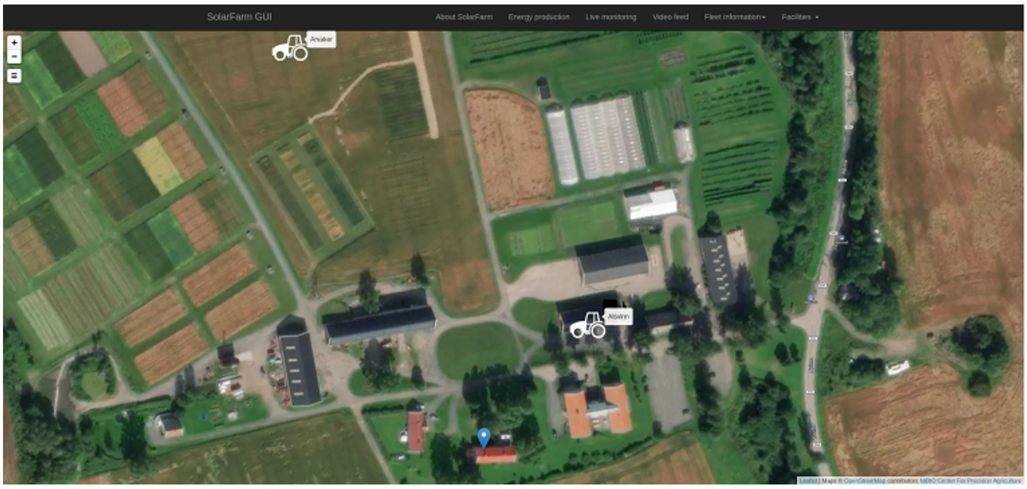


Figure 7. The graphical user interface, showing the positions of the tractor (labelled as Arvaker) and robot tractor (labelled as Alsvinn) inside the field.

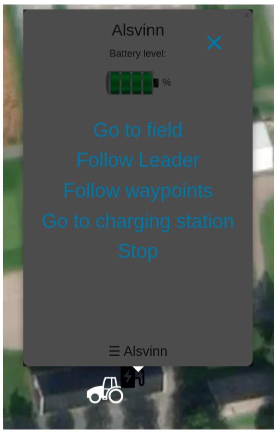


Figure 8. Contextual menu of the robot tractor for simple task allocation.

Since the robot tractor can operate autonomously, more options have been added to its contextual menu compared to the tractor or the charging station. For instance, the option "Follow waypoints" allows the robot tractor to follow a preset list of waypoints that the farmer has uploaded in advance. An illustrative example of the result of waypoint following can be seen in Figure 9a. Additionally, the farmer can request the robot tractor to drive back to the charging station, which also serves as a location where the vehicles can be parked when charging or not in use. When "go to charging station" is selected, an exit point from the field is considered to be the first waypoint for the robot tractor, followed by a sequence of waypoints that lead to the charging station location (Figure 9b).

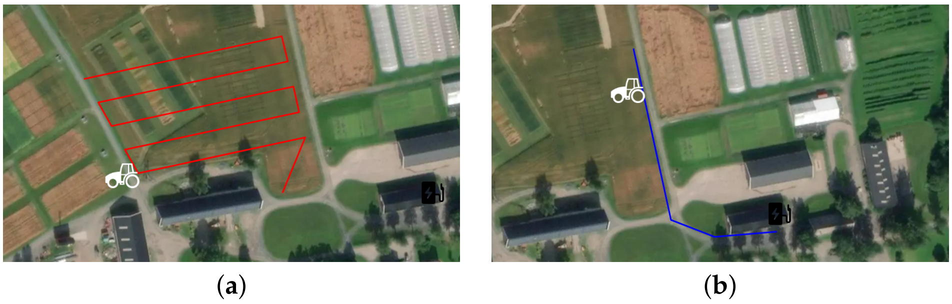

(**a**) (**b**)

Figure 9. Two examples of the GUI control of the mobile robot. (**a**) Follow waypoint example. (**b**) Go to charging station example.

The robot tractor receives requests from the GUI and processes them in a state-machine fashion. Each request triggers a subset of actions that fulfil the desired action. Table 1 displays the different elements of the contextual menu for the robot tractor and the charging station (the tractor's contextual menu shows the battery status and video feedback from the roof-mounted camera):

Table 1. Contextual menu elements.

Vehicle	Contextual Menu Option
Robot	Go to field Follow leader Follow waypoints Go to charging station Stop
Charging station	Plug cable Unplug cable Stop

Descriptions of the contextual menu elements:

Go to field: Sends a list of waypoints that leads the robot-tractor to the field.
Follow leader: Switch the driving mode to a leader–follower approach. The pose of the leader is taken as a reference point.
Follow waypoints: Read the waypoints file and execute them consecutively.
Go to charging station: Sends a list of waypoints that leads the robot-tractor to the charging station.
Plug cable: This mode is mainly used for manually controlling the charging process, as by default, the ACS and the robot tractor communicate, and the charging process is performed in an autonomous fashion.
Unplug cable: Similar to "Plug cable", this mode is used to manually ask the ACS to unplug the cable from the robot tractor.
Stop: Halt any ongoing tasks.

3. Leader–Follower Navigation Scheme

The primary objective of the leader–follower navigation scheme is to facilitate the autonomous driving of the robot tractor while using the manned tractor as a reference point [42]. To achieve this, the established communication between the vehicles enables continuous broadcasting of the reference point T_{ref} from the tractor to the robot tractor.

The inertial frame in which the tractor and the robot tractor are located is defined by the global reference $O : (X_i, Y_i)$. The reference point on the tractor, denoted as T, is situated within the tractor's inertial frame (X_T, Y_T). The heading angle of the tractor is represented by θ_t, which signifies the deviation of the tractor's attached frame from the inertial frame (Figure 10a).

The tractor's pose (T) is defined using Equation (1) as follows :

$$T = \begin{pmatrix} X_t \\ Y_t \\ \theta_t \end{pmatrix} \tag{1}$$

As for the representation of the robot tractor's kinematics, we denote R as the centre of the robot tractor's inertial frame (X_R, Y_R). The heading angle of the robot tractor θ_r is the deviation of the tractor's attached frame to the inertial frame (Figure 10b).

The robot tractor's pose (R) is defined using Equation (2) as follows :

$$R = \begin{pmatrix} X_r \\ Y_r \\ \theta_r \end{pmatrix} \tag{2}$$

Please note that X_t, Y_t, X_r, and Y_r correspond to the X and Y Universal Transverse Mercator (UTM) coordinates.

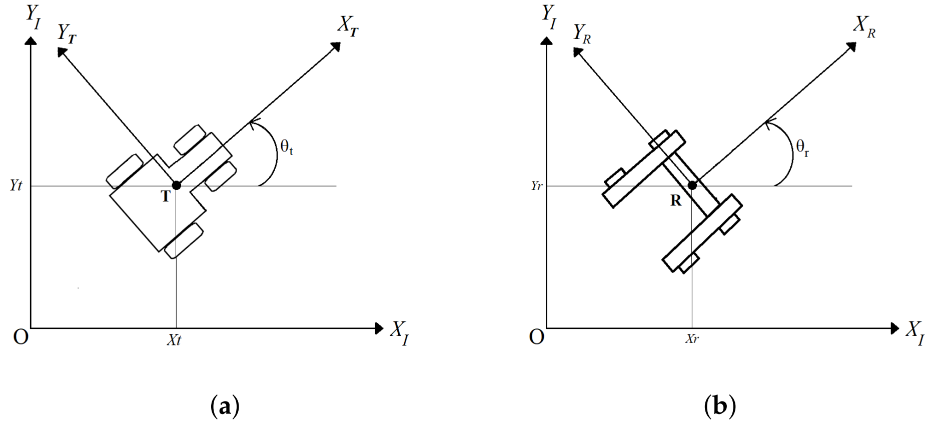


Figure 10. Pose representation of the tractor and the robot tractor. (**a**) Tractor pose representation within a 2D planar surface. (**b**) Robot tractor pose representation within a 2D planar surface.

The Control Law

To establish the leader–follower navigation scheme, the robot tractor must be capable of navigating to a reference point T_{ref} that is relative to the tractor's pose P. T_{ref} depends on the following mode f_{mode}: *back*, *left*, or *right*, as depicted in Figure 11. Depending on the following mode (indicating where the robot should position itself in relation to the tractor's pose), the angle β_{mode} and distance λ_{mode} are defined as follows for the corresponding modes: left (l), right (r), or back (b).

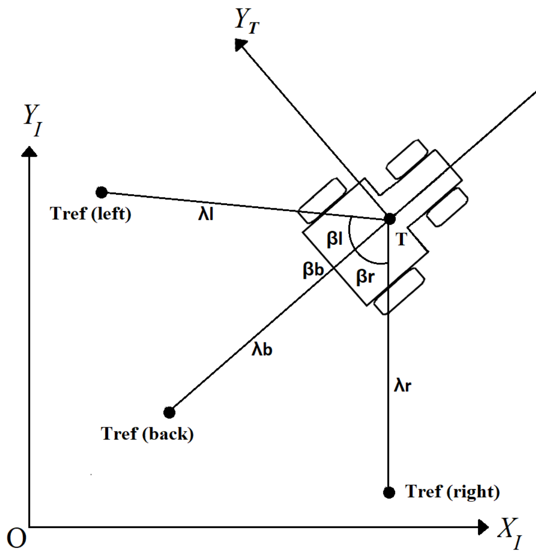


Figure 11. Illustration of the T_{ref} pose in reference to the tractor's pose T.

Using Equation (1), we define the coordinates of T_{ref} as follows:

$$\begin{aligned} T_{ref}x &= X_t + \lambda_{mode}.cos(\theta_t - \beta_{mode}) \\ T_{ref}y &= Y_t + \lambda_{mode}.sin(\theta_t - \beta_{mode}) \end{aligned} \tag{3}$$

Equation (3) describes the relationship between the reference target's positions and the tractor's pose:

- $T_{\text{ref}}x$ and $T_{\text{ref}}y$: Represent the reference target's positions in the x and y coordinates, respectively. These are the desired positions that the robot has to reach.

- X_t and Y_t: Represent the current positions of the tractor in the inertial frame (X_I, Y_I).
- θ_t: Represents the heading of the tractor.
- λ_{mode}: Represents the distance that the robot should keep from the tractor.
- β_{mode}: Represents the angle that the robot should keep towards the tractor.

The developed control law generates the necessary linear u and angular r velocities that control the motion of the robot tractor (Figure 12).

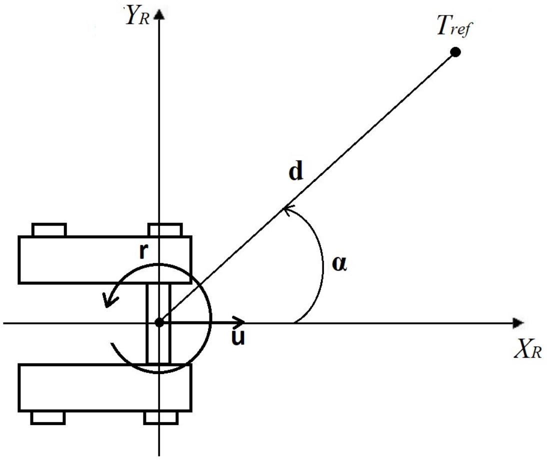


Figure 12. The robot tractor's kinematics.

u and r, respectively, represent the linear and angular velocities of the robot tractor. d represents the Euclidean distance between the robot tractor and the reference point T_{ref}, and α represents the heading of the robot tractor towards the reference point T_{ref}.

The control objective is to regulate the distance d and the steering angle α as follows:

$$\lim_{t\to\infty} d(t) = 0 \lim_{t\to\infty} \alpha(t) = 0 \tag{4}$$

In order to allow the robot tractor to navigate to the reference point T_{ref}, for the linear velocity u, we used a simple Proportional controller, whereas for the angular velocity r, we used a Proportional–Integral Derivative (PID) controller:

$$\begin{aligned} u &= k_{p_u} \cdot d \\ r &= k_{p_r} \cdot \alpha + k_{i_r} \cdot \int_{t_i}^{t_{i+1}} \alpha \cdot dt + k_{d_r} \cdot \frac{\alpha}{dt} \end{aligned} \tag{5}$$

The choice for the control law, as represented in Equation (4), is justified by a trial-and-error method to choose the best model-based independent method that fulfils the control objective while making the physical robot tractor navigate in a stable manner.

4. Fleet Navigation Experimental Results

The developed management system was tested in real conditions, and the experiments were carried out at the NIBIO Apelsvoll research station (60.8707° N, 10.6108° E). We can divide the experimental results into two parts. The first part is related to the user interaction with the system through the developed GUI, whereas the second part is dedicated to the field navigation of the leader-follower scheme.

Fleet Navigation

The main focus on the experiments was to assess the connectivity issues that may arise since the leader broadcasts its position continuously to the follower, and how well the follower navigates taking the reference pose T_{ref} as a goal.

The tractor and the robot-tractor are both connected to the same wireless network via the router installed on the tractor. A connectivity loss results in the robot-tractor halting the navigation and waiting until the connection is established. During the experiments, we have not faced any connectivity loses or lags, which means that the hardware setup satisfies the requirements of robustness when it comes to m2m communication inside the field.

The parameters used during the fleet navigation experimental setup can be found in Table 2.

Table 2. Experimental parameters.

Parameter	Significance	Values
f_{mode}	following mode	right
β_r	absolute angle between the tractor and the robot tractor	$2{*}\pi$ rad
λ_r	distance between the tractor and the robot tractor	7 m
k_{p_u}	positive proportional gain for the linear controller	0.3
k_{p_r}	positive proportional gain for the angular controller	0.6
k_{i_r}	positive integral gain for the angular controller	1.0
k_{d_r}	positive derivative gain for the angular controller	2.2

The other goals of the experimental setup is to assess the feasibility of the proposed architecture, evaluate the resilience of the developed control laws, and verify the conditions of the control objective (Equation (4)). For this purpose, we conducted test-drive experiments within one of our fields after the harvesting season, manually driving the tractor along randomly selected paths. These paths encompassed both the traditional driving method (transitioning from one row to another) and driving across the field in various directions to assess the robot-tractor's response to uphill and downhill conditions during leader-follower navigation.

While driving in the field, along with visually evaluating the driving scheme, we recorded the GNSS locations of both the robot tractor and the tractor. This was done to evaluate the precision of the developed control scheme and the resilience of the fleet management. Figure 13 displays the paths of the robot tractor and the tractor in UTM coordinates, illustrating a sequence example of driving in a straight line and subsequently executing a "U-turn" to reach the next row.

Figure 14 portrays the diverse sequences from the field navigation, utilising both the manned electric tractor and the robot tractor.

The navigation paths depicted in Figure 13 demonstrate that the robot tractor effectively followed the reference point T_{ref} with an acceptable degree of accuracy. It converged to replicate the paths with minor deviations, primarily attributed to the uneven terrain of the field. Despite employing data filtering, these deviations led to unintended oscillations that affected the pose estimation of both vehicles.

However, when transitioning to a different row, as seen in the case of executing a U-turn, the robot tractor deviated from the reference trajectory. It lagged behind until it reached the second row, where it then started converging once again to the desired path, following the tractor as intended.

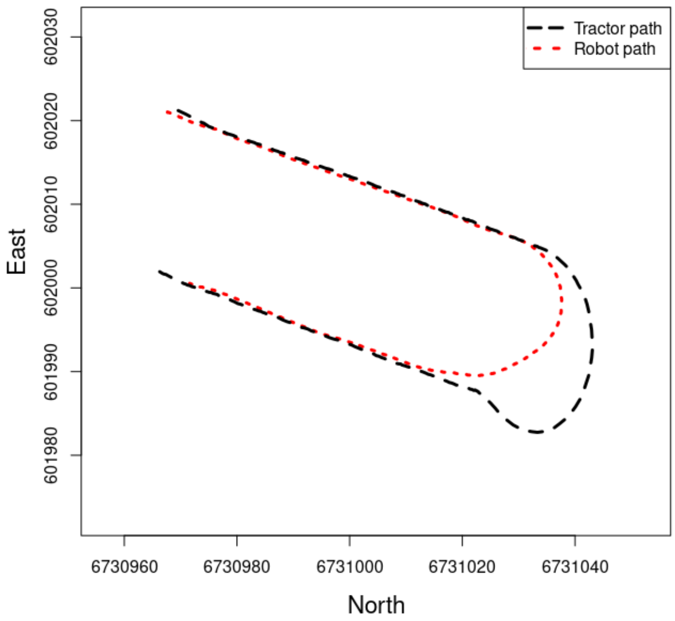
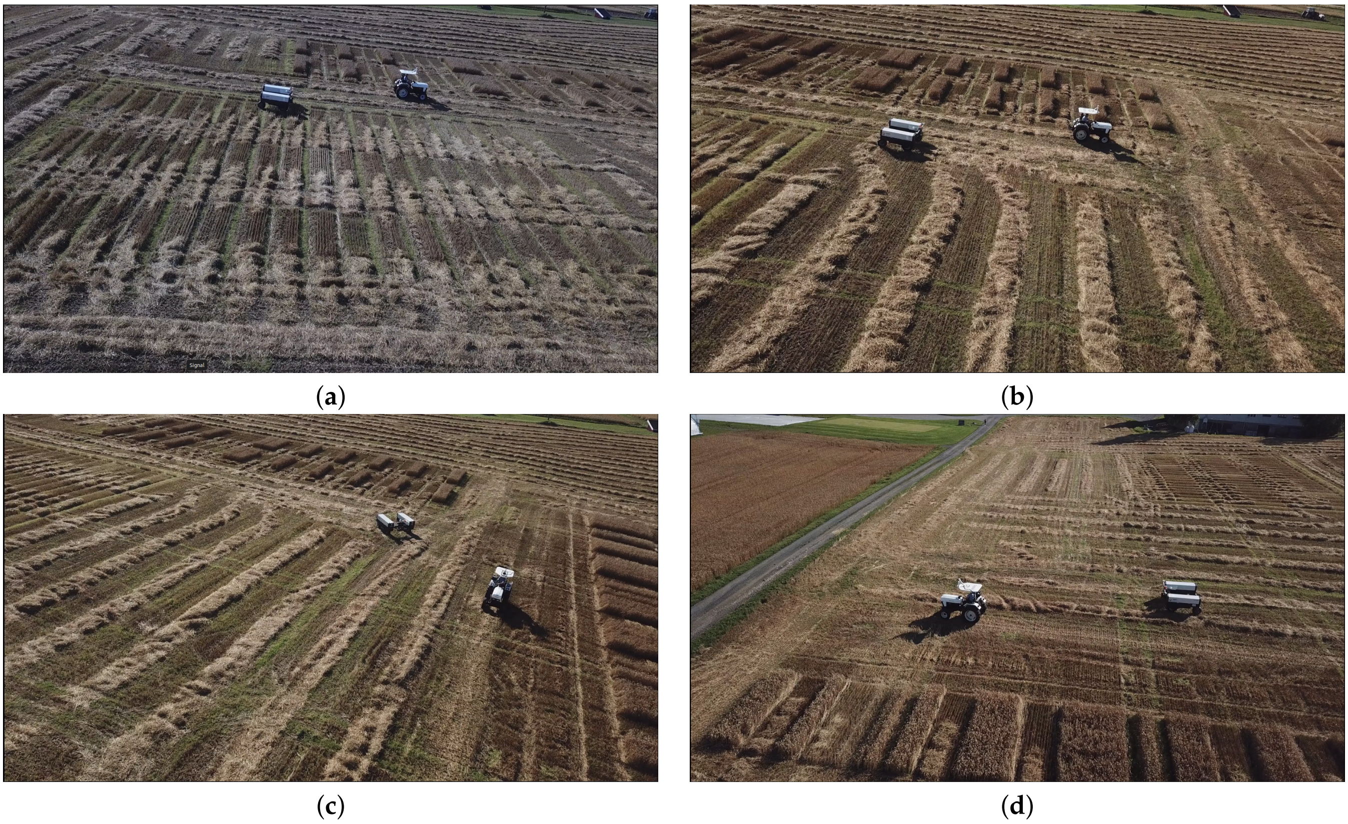
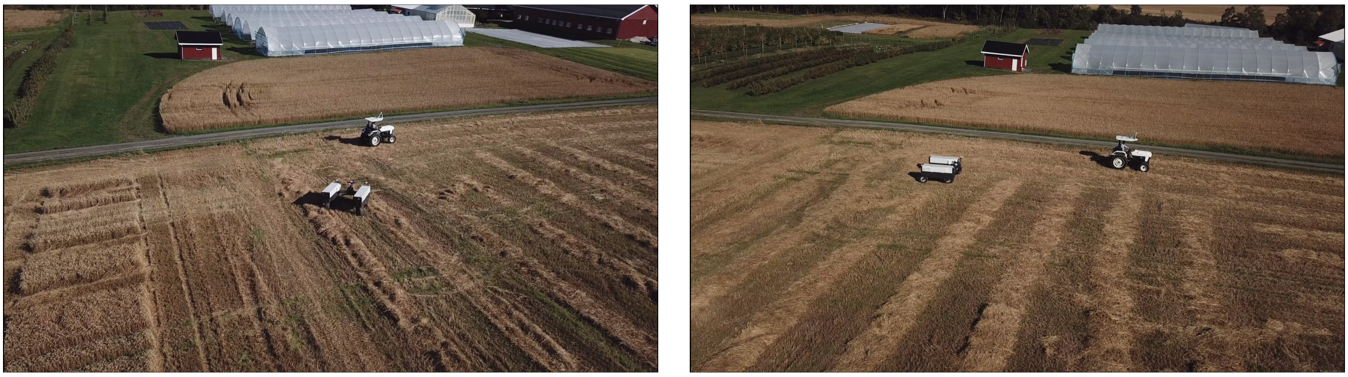


Figure 13. The recorded paths of the tractor and the robot tractor in UTM coordinates while performing field navigation. Units are in meters (m).


(**a**) (**b**)

(**c**) (**d**)

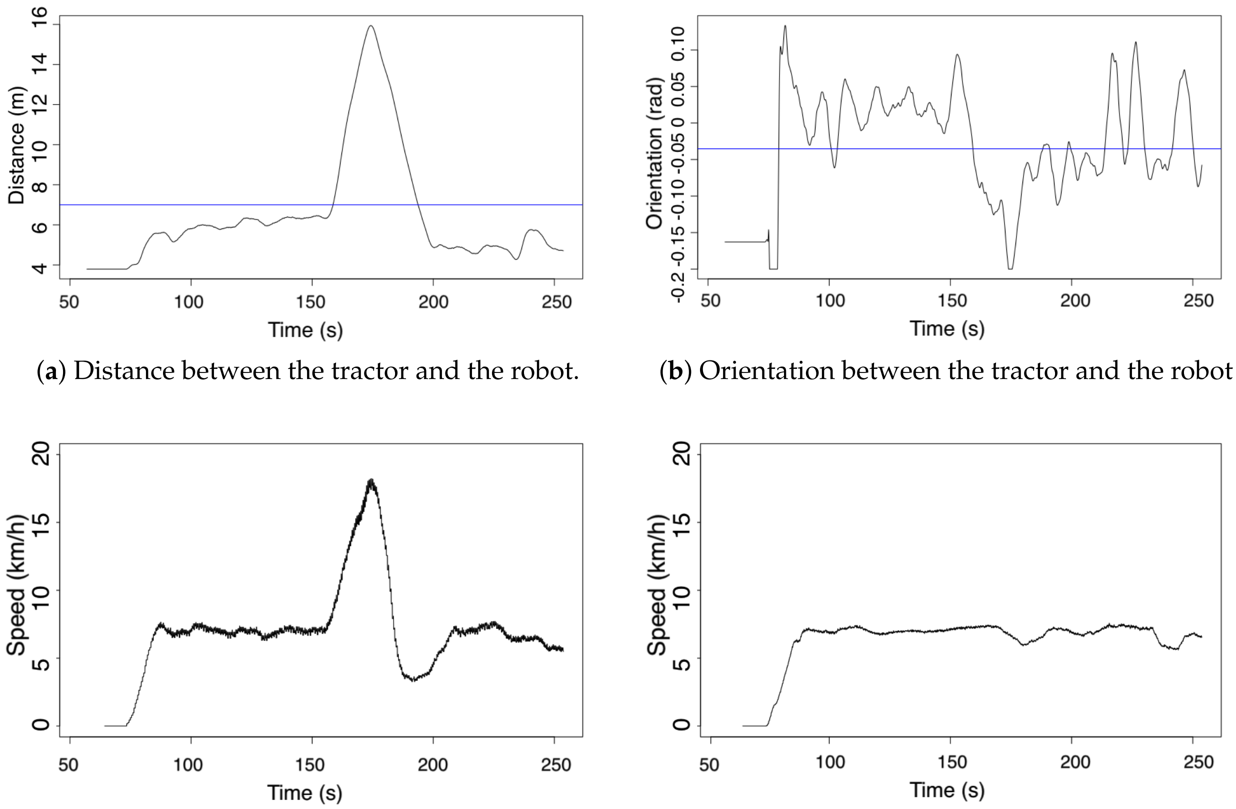
Figure 14. *Cont.*



(**e**) (**f**)

Figure 14. Experimental results of fleet navigation using a manned electric tractor and an unmanned robot tractor. (**a**) Field navigation sequence 1: driving in a straight line. (**b**) Field navigation sequence 2: approaching the headland. (**c**) Field navigation sequence 3: initiating a right turn. (**d**) Field navigation sequence 4: driving in a straight line. (**e**) Field navigation sequence 5: performing another right turn. (**f**) Field navigation sequence 6: driving in a straight line.

Further investigations were conducted to comprehend the underlying cause of the observed lag of the robot tractor during U-turns. This analysis involved examining data related to the distance (Figure 15a) and orientation (Figure 15b) between the robot tractor and the reference point, as well as the velocities of both the tractor (Figure 15c) and the robot tractor (Figure 15d).



(**a**) Distance between the tractor and the robot.

(**b**) Orientation between the tractor and the robot

(**c**) Tractor speed.

(**d**) Robot speed

Figure 15. The collected data from the field navigation: (**a**) distance and (**b**) orientation that separates the tractor and the robot. (**c**) Tractor speed. (**d**) Robot speed.

In Figure 15a,b, it can be seen that the average distance and orientation remain close to the predefined goals of the leader–follower navigation scheme (0 degrees for orientation and 7 m for distance). However, starting from Time = 158 s, we can see a divergence from the reference goals in terms of distance and orientation. This can be attributed to the fact that the tractor driver initially maintained a consistent speed while driving in a straight line, but during the execution of U-turns, the driver increased the tractor's speed. This speed increase was primarily a safety precaution, given that the robot tractor lacked obstacle-detection capabilities at the time of the experiments. Consequently, the robot tractor was unable to closely follow the tractor, leading to an increased distance.

The distance between the vehicles continued to increase until they returned to the second row (Time = 190 s). At this point, the tractor driver slowed down to resume the normal speed of around 6 km/h, as evidenced in Figure 15c.

The root cause was the manual operation of the tractor. While driving straight, the tractor was maintained at a constant speed. However, during U-turns, the driver accelerated. Due to safety limitations, the robot tractor has an upper-speed limit. As a result, it lagged behind when the tractor sped up and only managed to catch up when the driver slowed down in the second row. To mitigate this issue in future scenarios, one potential approach is to utilise cruise control on the tractor to maintain a consistent speed. This would prevent the robot tractor from reaching its upper speed limit and subsequently lagging behind the tractor. Another drawback of the proposed system pertains to its adaptability when integrating multiple followers. The existing state of the proposed approach enforces a predetermined position for followers to follow the leader, which cannot be altered while navigating. Consequently, if a follower encounters a failure or requires a battery recharge, the designated area it was intended to cover remains vacant. To address this issue, a potential solution involves introducing reconfigurability to enable the fleet to adjust its configuration dynamically, thereby compensating for any missing components during the navigation procedure.

5. Conclusions

The agricultural industry is facing increasing pressure to adopt more sustainable practices without compromising food production levels, essential for both farmers' incomes and national food security objectives. This study explores a potential avenue to reduce emissions from agricultural machinery while upholding workloads and production levels akin to conventional methods. The focus of this paper is on the design of a management system for a diverse fleet of vehicles, utilising the Robot Operating System (ROS). A pivotal aspect is the development of an intuitive Graphical User Interface (GUI), enabling real-time monitoring of vehicle statuses and positions within the field. Moreover, the GUI facilitates the assignment of straightforward tasks to autonomous robots. This paper also extensively outlines a control strategy for fleet navigation based on the leader–follower model. Experimental outcomes showcase the successful achievement of the control goals. Looking ahead, the integration of additional robots into the system, reconfiguration management, and the incorporation of user feedback, particularly from farmers, will allow for a comprehensive evaluation of the usability and effectiveness of the proposed user interface.

Author Contributions: E.H.C.H. and J.G. conceptualised the communication infrastructure and multi-vehicle management system. E.H.C.H. conceived, designed, and performed the experiments, analysed the data, and wrote the paper with F.G. and J.G. All authors have read and agreed to the published version of the manuscript.

Funding: The work presented in this paper was part of the project SOLARFARM, funded by the Norwegian Research Council (program "Lavutslipp 2030", project no. 280390).

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

References

1. Parkhomenko, G.; Voinash, S.; Sokolova, V.; Krivonogova, A.; Rzhavtsev, A. *Reducing the Negative Impact of Undercarriage Systems and Agricultural Machinery Parts on Soils*; IOP Conference Series: Earth and Environmental Science; IOP Publishing: St. Petersburg, Russia, 2019; Volume 316, p. 012049.
2. Lovarelli, D.; Fiala, M.; Larsson, G. Fuel consumption and exhaust emissions during on-field tractor activity: A possible improving strategy for the environmental load of agricultural mechanisation. *Comput. Electron. Agric.* **2018**, *151*, 238–248. [CrossRef]
3. El Ghobashy, H.; Shaban, Y.; Okasha, M.; Abd El-Reheem, S.; Abdelgawad, M.; Ibrahim, R.; Ibrahim, H.; Abdelmohsen, K.; Awad, M.; Cottb, M.; et al. Development and evaluation of a dual-purpose machine for chopping and crushing forage crops. *Heliyon* **2023**, *9*, e15460. [CrossRef] [PubMed]
4. Hjelkrem, A.G.R.; Fagerström, J.; Kvalbein, L.; Bakken, A.K. *Potential for Replacing Fossil Energy by Local PV Energy for Field and Transport Work in Norwegian Farming*; NIBIO Rapport; NIBIO: Oslo, Norway, 2020.
5. Spykman, O.; Gabriel, A.; Ptacek, M.; Gandorfer, M. Farmers' perspectives on field crop robots–Evidence from Bavaria, Germany. *Comput. Electron. Agric.* **2021**, *186*, 106176. [CrossRef]
6. Albiero, D.; Garcia, A.P.; Umezu, C.K.; de Paulo, R.L. Swarm Robots in Mechanized Agricultural Operations: Roadmap for Research. In Proceedings of the Congresso Brasileiro de Automática-CBA, Online, 23–26 November 2020; Volume 2.
7. Oliveira, L.F.; Moreira, A.P.; Silva, M.F. Advances in agriculture robotics: A state-of-the-art review and challenges ahead. *Robotics* **2021**, *10*, 52. [CrossRef]
8. Mpetile, Z.; Chinyamurindi, W. Motivational factors into agriculture as a career path: Narratives of Black emerging farmers as entrepreneurs in South Africa. *J. Enterp. Communities People Places Glob. Econ.* **2021**, *15*, 739–754. [CrossRef]
9. Loukatos, D.; Kondoyanni, M.; Kyrtopoulos, I.V.; Arvanitis, K.G. Enhanced Robots as Tools for Assisting Agricultural Engineering Students' Development. *Electronics* **2022**, *11*, 755. [CrossRef]
10. Xiuhtecutli, N.; Shattuck, A. Crisis politics and US farm labor: Health justice and Florida farmworkers amid a pandemic. *J. Peasant. Stud.* **2021**, *48*, 73–98. [CrossRef]
11. Sparrow, R.; Howard, M. Robots in agriculture: Prospects, impacts, ethics, and policy. *Precis. Agric.* **2021**, *22*, 818–833. [CrossRef]
12. Simões Filho, L.M.; Lopes, M.A.; Brito, S.C.; Rossi, G.; Conti, L.; Barbari, M. Robotic milking of dairy cows: A review. *Semin. Ciênc. Agrár.* **2020**, *41*, 2833–2850. [CrossRef]
13. Khan, Z.H.; Khalid, A.; Iqbal, J. Towards realizing robotic potential in future intelligent food manufacturing systems. *Innov. Food Sci. Emerg. Technol.* **2018**, *48*, 11–24. [CrossRef]
14. Geipel, J.; Bakken, A.K.; Jørgensen, M.; Korsaeth, A. Forage yield and quality estimation by means of UAV and hyperspectral imaging. *Precis. Agric.* **2021**, *22*, 1437–1463. [CrossRef]
15. Germani, L.; Mecarelli, V.; Baruffa, G.; Rugini, L.; Frescura, F. An IoT architecture for continuous livestock monitoring using LoRa LPWAN. *Electronics* **2019**, *8*, 1435. [CrossRef]
16. Danton, A.; Roux, J.C.; Dance, B.; Cariou, C.; Lenain, R. Development of a spraying robot for precision agriculture: An edge following approach. In Proceedings of the 2020 IEEE Conference on Control Technology and Applications (CCTA), Montreal, QC, Canada, 24–26 August 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 267–272.
17. Yiguang, Z.; Liang, Y.; Shanshan, Z.; Benhai, X. Advances in the development and applications of intelligent equipment and feeding technology for livestock production. *Smart Agric.* **2019**, *1*, 20.
18. Rose, D.C.; Sutherland, W.J.; Barnes, A.P.; Borthwick, F.; Ffoulkes, C.; Hall, C.; Moorby, J.M.; Nicholas-Davies, P.; Twining, S.; Dicks, L.V. Integrated farm management for sustainable agriculture: Lessons for knowledge exchange and policy. *Land Use Policy* **2019**, *81*, 834–842. [CrossRef]
19. Koubâa, A. *Robot Operating System (ROS)*; Springer: Cham, Switzerland, 2017; Volume 1.
20. Koubaa, A. ROS as a service: Web services for robot operating system. *J. Softw. Eng. Robot.* **2015**, *6*, 1–14.
21. Crick, C.; Jay, G.; Osentoski, S.; Jenkins, O.C. ROS and rosbridge: Roboticists out of the loop. In Proceedings of the 2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Boston MA, USA, 5–8 March 2012; IEEE: Piscataway, NJ, USA 2012; pp. 493–494.
22. Khanh, Q.V.; Nguyen, V.H.; Minh, Q.N.; Van, A.D.; Le Anh, N.; Chehri, A. An efficient edge computing management mechanism for sustainable smart cities. *Sustain. Comput. Inform. Syst.* **2023**, *38*, 100867. [CrossRef]
23. Khanh, Q.V.; Chehri, A.; Quy, N.M.; Han, N.D.; Ban, N.T. Innovative Trends in the 6G Era: A Comprehensive Survey of Architecture, Applications, Technologies, and Challenges. *IEEE Access* **2023**, *11*, 39824–39844.
24. Crick, C.; Jay, G.; Osentoski, S.; Pitzer, B.; Jenkins, O.C. Rosbridge: Ros for non-ros users. In *Robotics Research*; Springer: Cham, Switzerland, 2017; pp. 493–504.
25. Fielding, R.T.; Kaiser, G. The Apache HTTP server project. *IEEE Internet Comput.* **1997**, *1*, 88–90. [CrossRef]
26. Harik, E.H.C.; Guérin, F.; Guinand, F.; Brethé, J.F.; Pelvillain, H. UAV-UGV cooperation for objects transportation in an industrial area. In Proceedings of the 2015 IEEE International Conference on Industrial Technology (ICIT), Seville, Spain, 17–19 March 2015; pp. 547–552. [CrossRef]
27. Tractor, S.E. Solectrac. 2023. Available online: https://solectrac.com (accessed on 27 April 2023).
28. Swift Piksi Multi Evaluationkit. Available online: https://store.swiftnav.com/products/piksi-multi-evaluation-kit ((accessed on 1 July 2023).
29. Vectornav VN-100. Available online: https://www.vectornav.com/products/detail/vn-100 (accessed on 1 July 2023).

30. NVIDIA Jetson AGX Xavier. Available online: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-agx-xavier/ (accessed on 1 July 2023).
31. Asus 4G-AC68U Wireless Router. Available online: https://www.asus.com/no/networking-iot-servers/modem-routers/all-series/4g-ac68u/ (accessed on 1 July 2023).
32. Logitech C925e Webcamera. Available online: https://www.logitech.com/no-no/products/webcams/c925e-business-webcam.960-001076.html (accessed on 1 July 2023).
33. SICK LIDAR. Available online: https://www.sick.com/be/en/lidar-sensors/2d-lidar-sensors/tim/tim571-2050101/p/p412444 (accessed on 1 July 2023).
34. Ubiquiti PicoStation2. Available online: https://www.netwifiworks.com/PicoStation-M.asp (accessed on 1 July 2023).
35. Harik, E.H.C. Design and implementation of an autonomous charging station for agricultural electrical vehicles. *Appl. Sci.* **2021**, *11*, 6168. [CrossRef]
36. Universal Robots UR5e. Available online: https://www.universal-robots.com/products/ur5-robot/ (accessed on 1 July 2023).
37. Superdroid Robots Superdroid. Available online: https://www.superdroidrobots.com/ (accessed on 1 July 2023).
38. Lawson, B.; Sharp, R. *Introducing HTML 5*; New Riders: Indianapolis, IN, USA, 2011.
39. Guha, A.; Saftoiu, C.; Krishnamurthi, S. The essence of JavaScript. In Proceedings of the European conference on Object-Oriented Programming, Maribor, Slovenia, 21–25 June 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 126–150.
40. Alexander, B.; Hsiao, K.; Jenkins, C.; Suay, B.; Toris, R. Robot web tools [ros topics]. *IEEE Robot. Autom. Mag.* **2012**, *19*, 20–23. [CrossRef]
41. Crickard, P., III. *Leaflet.js Essentials*; Packt Publishing Ltd.: Birmingham, UK, 2014.
42. Harik, E.H.C. Tractor-Robot Cooperation: A Heterogeneous Leader-Follower Approach. *Robotics* **2023**, *12*, 57. [CrossRef]

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.