Autonomous Robotic Platform for Precision Orchard Management: Architecture and Software Perspective

Published Version:
Mengoli, D., Tazzari, R., Marconi, L. (2020). Autonomous Robotic Platform for Precision Orchard Management: Architecture and Software Perspective. Piscatawey (NJ) : Institute of Electrical and Electronics Engineers Inc. [10.1109/MetroAgriFor50201.2020.9277555].

Availability:
This version is available at: https://hdl.handle.net/11585/798598 since: 2021-02-12

Published:

DOI: http://doi.org/10.1109/MetroAgriFor50201.2020.9277555

Autonomous Robotic Platform for Precision Orchard Management: Architecture and Software Perspective

Dario Mengoli
DEI department "G. Marconi"
University of Bologna
Bologna, Italy
dario.mengoli2@unibo.it

Roberto Tazzari
DEI department "G. Marconi"
University of Bologna
Bologna, Italy
roberto.tazzari2@unibo.it

Lorenzo Marconi
DEI department "G. Marconi"
University of Bologna
Bologna, Italy
lorenzo.marconi@unibo.it

***Abstract*—We present an autonomous ground robotic platform for agriculture application. The design is specifically targeted for small/medium farms with orchards and in this paper we propose a new vehicle concept as well as the sensor suite and software architecture to accomplish the implementation of the navigation algorithm designed to autonomously operate within the rows of an orchard. In this context, we also show how to exploit the structure of the orchard to optimize the Hough Transform algorithm to detect tree rows. Simulations and experimental results are presented.**

***Index Terms*—UGV, autonomous rover, precision agriculture, robotics, autonomous navigation, orchards.**

I. INTRODUCTION

Robotics and automation are constantly increasing their influence in agricultural context to improve productivity and quality.

The first attempts were made in the early 2000s [1], where extensive automation tasks were performed in large estates, usually planted with corn or wheat, exploiting also the capabilities of Real-Time Kinematics (RTK) GPS localization [2]. Recently, vineyard growers have increasingly adopted technology to automate crop yield estimation [3], [4] and multi-spectral monitoring and mapping [5]. As the final product (wine) is characterized by an high added value, this has enabled an early adoption of technology and a greater capability to invest in automation, exploiting a shorter return-to-investment time.

Nowadays, technology is mature enough to provide a cost-effective automated robotic platform suitable also to other agriculture field, such as orchards, where profit margins are not as high as in vineyards, but still automation can be of paramount importance to increase competitiveness and sustainability of companies.

In this context the concept of *Orchard Precision Agriculture* (OPA) arises, in which robots perform tasks according to environment conditions, distributing fertilizers or water only where/when necessary, thereby optimizing treatments and energy resources [6], [7]. To maximize exploitation of these procedures, it becomes crucial to develop versatile robotic platforms able to perform a wide range of tasks and overtaking principal drawbacks of commercial tractors like excessive weight and size, pollution and soil compaction [8].

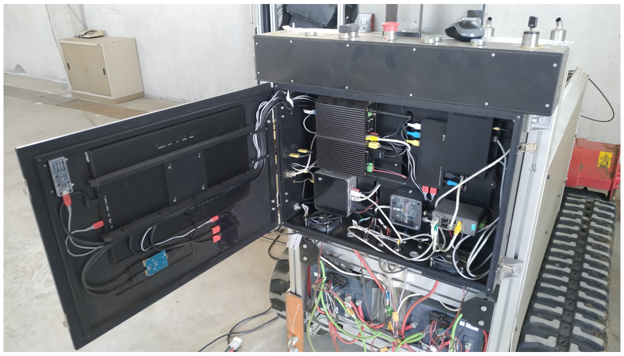


Fig. 1: Prototype electronics and processing units.

OPA also offers challenges from a pose estimation point of view: apart from open-field navigation, it is not advisable to completely rely on GPS positioning, since it cannot provide info about eventual obstacles and it would require top class GPS receivers to achieve the required accuracy. [9] and [10] show that relative localization with respect to trees is a reliable and GPS-free method to achieve *in-row* navigation, even though Bergerman's group relied on special markers at the end of each tree rows to properly detect the end of the lane. To avoid this, we propose to make use of track motor encoders in order to robustify end-of-row detection.

This paper presents the concept behind the development of our Unmanned Ground Vehicle (UGV) prototype (Fig. 1) for OPA, which serve as a versatile robotic platform able to perform several autonomous tasks. Then it is presented the hardware and software architecture implemented on the prototype, and the localization and navigation algorithms used to autonomously operate both in open field and within orchard rows. Finally, it is also presented the simulation environment we created in order to develop and test the software subsystem, as well as practical experiments aiming to validate in-row navigation on real orchards.

II. DESIGN OBJECTIVES AND BACKGROUND IDEA

Farming environments are quite peculiar, especially for what concerns working terrain. The soil is extremely not regular, both in terms of ground flatness and in terms of obstacles. Moreover, the sizing of the platform have to comply with existing orchards and activities inside them. In terms of power and mechanical design, the platform is designed to target small/medium farms that represent a relevant economic and social entry market.

The consequent small and lightweight structure fulfills the specification of a platform easy-to-maintain, affordable, efficient, easily transportable, and "low-weight" to allow field operations also in presence of adverse terrain conditions. The rover adopts the "weather independent" concept delineated in [8], augmented to an expandability concept based on replication of a basic "small" platform if higher performances were needed rather than making a single unit bigger.

A. Adaptability and flexibility

The current prototype relies on a "core" locomotion system given by two rubber tracks that can be flexibly composed by means of a mechanical structure and attached implements according to the targeted operation and working environment. For instance, the inter-axis distance between the caterpillars can be potentially adapted to different row dimensions. The adaptability and flexibility that characterize the platform are used to accomplish demanding operations in orchard management with "plug & play" implements. The rover adopts the concept of machine with "Task Oriented Automatic Subsystems" [8]. The platform is conceived to be a "socket in the field", a source of power easily accessible, able to dispense energy to all the interchangeable implements.

B. Scalability and Expandability

Although the mechatronic design is scalable to be suited to larger/smaller scenarios, it naturally fits the concept of expandability by replication of identical "small" modules: if higher work rates are needed, the vision is to add further "small" modules rather than designing a larger one. This concept allows one to keep the basic platform sustainable (in terms of initial investment, maintenance, maneuverability, robustness, incremental costs while expanding, etc.) and replicating it to increase work rates. Redundancy of platforms to increase precise execution of time-sensitive operations (like spraying) or face emergencies like faults and ordinary maintenance of single units is an additional key benefit.

C. Versatility: Autonomous platform and "Facilitator"

A further feature of the rover is the flexibility in being used in totally autonomous scenarios as well as "facilitator" in contexts where human workers are directly involved. For instance, in the shredding or spraying operation, the platform operates autonomously properly equipped with the dedicated implement and with the navigation algorithm. On the other hand, the same platform, properly transformed with the appropriate implement, could support labor during harvesting.

D. Implement design

The over-sized and over-powered traction system able to pull implements, which underlies actual working platforms, is replaced by a "motorized implement" concept. The final goal is to optimize the whole platform in terms of weight, power efficiency, and maneuverability through an integrated design between the different components. Currently, two customized implements that are specially tailored to the rover prototype have been developed. Both the implements share a common power-train that provides adequate torque. The common mechanical interface can be adapted to both tools in order to provide a quick change operation. The actual implements are:

a) branch shredder and lawn mower: A commercial shredder/mower was adapted to fit the compactness of the rover in terms of power takeoff. This allows shredding and lawn mowing tasks inside orchards, also with wet terrain or mud.

b) orchard sprayer: A commercial multi-nozzle sprayer with turbine has been customized to be mounted directly on the implement support interface of the rover, without using a separate tow trolley. Power distribution is achieved using a belt with pulleys that provide also a suitable reduction factor to achieve nominal speed of the turbine.

The nozzles are equipped with solenoid valves for selective spaying according to the precision treatment map to follow.

III. HARDWARE AND SOFTWARE ARCHITECTURE

A. Logical architecture

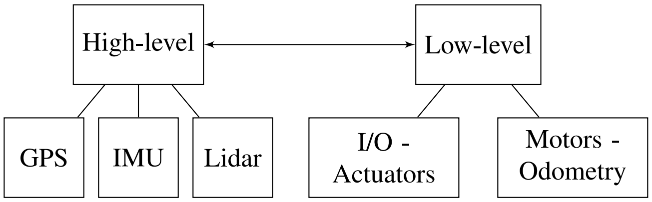


Fig. 2: HW/SW architecture scheme.

The whole control and navigation stack is an evolution of our former work [11]. The logic architecture of the robot is organised into a high-level (HL) component and a low-level (LL) one (Fig. 2). The LL software and components are in charge of communicating directly with the motor drivers and all I/O interfaces to provide basic safety features and simple/manual driving of the rover itself. The HL computational unit runs ROS (Robot Operative System) framework on Linux OS and is devoted to run all the HL navigation algorithms to accomplish self-driving tasks on open field and inside orchards' rows.

1) Hardware and Sensors: The main components installed onboard are:

- GNSS
- LiDAR
- IMU
- Tracks odometry
- Computational Unit

a) GNSS: We used the *Trimble R8s* GNSS system, a RTK GPS/GLONASS receiver that provides position reference to the whole autonomous platform with sub-meter accuracy. The RTK correction acquired from existing topographic stations is applied to enhance precision up to centimeter accuracy. The standard NMEA interface is used to allow transparent substitution of sensor type.
The whole system has been configured also to work with non-RTK GNSS sensors, with a reduction of position accuracy that can be improved by using an appropriate sensor fusion algorithm.

b) LIDAR: The 3D *Velodyne LiDAR PUCK VLP-16* laser scanner is used to provide environment perception, thus making possible to detect rows of trees, obstacles, branches or other orchard features, as well as ditches, tranches and other non-transitable areas.
This sensor is of paramount importance to provide reliable navigation inside orchards rows.

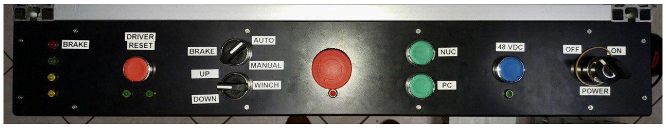
c) IMU: The *HMC5883L* Inertial Measurement Unit (IMU) is composed of a three-axis gyroscope, a three-axis accelerometer and a three-axis magnetometer. This provides attitude estimation for rover's rigid-body frame and contribute to sensor fusion algorithms to improve the accuracy of the position estimation.

d) Tracks odometry: The electric brushless motors, that provide locomotion by powering the tracks, are equipped with sin-cos encoders providing axle position and revolutions estimation. The Electronic Speed Controllers (ESCs) can use this data to give RPM speed feedbacks. This information is then used to compute robot odometry by integrating linear and angular velocity over time.
The motor model is Metalrota MR250 equipped with sin-cos encoder and the ESCs are two *SEVCON Gen4*^{TM} . The skid-steering kinematics model is used to compute the odometry.

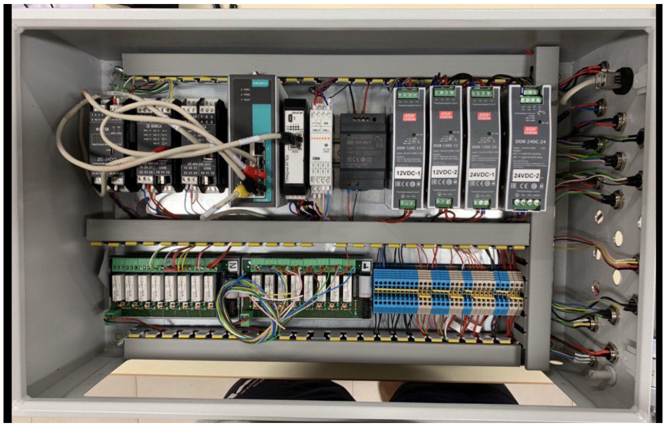
e) CPU: Two different computers are used to separate HL computations from LL logic that controls critical systems such as the motor controllers. This separation was also enforced by the fact that all the HL localization, navigation and control algorithms are implemented using ROS (on Linux OS), rather the most reliable and stable drivers for CANBus (used for communication between I/O devices and to control the motors) are available for Windows OS. Therefore, the two computational units are interconnected using an Ethernet network and exchange real-time data during normal operation. The choice to not rely on virtual machines was taken in order to maximize efficiency on both sides, being them equally critical for the task accomplishment.
Several micro-controller boards are also been used to simplify sensor readings such as the IMU sensor.
The Computer Units used are Cincoze DL-1000 and Intel NUC D54250WYKH. Micro-controllers belong to the Arduino family.

2) Communication: All the sensors and hardware described need to interact via a reliable communication backbone, able to support the whole data throughput coming from the sensors and allowing low-latency commands to the actuators. This is accomplished by using an Ethernet infrastructure to allow connection between LL and HL processing units and acquiring sensor data (especially the massive point cloud information coming from the LiDAR. Conversely, I/O and motor drivers are connected using a Controller Area Network (CAN bus). This choice is driven on the Ethernet side by bandwidth and speed, while reliability and robustness against electro-magnetic interference is of primarily importance for the CAN bus side. This separation can be crucial also for future certification purposes as critical components may rely on a certifiable environment for safety purposes.

a) I/O Interface: Nominal operations require different interfaces for actuators (e.g. water pumps), buttons or switches (e.g. safety switch and some physical interface), and notification (e.g. activity LEDs), which are shown in Fig. 3a.



(a) Switches, buttons and LEDs panel



(b) Internal view of I/O box components.

Fig. 3: I/O interfaces

The I/O interfaces includes some low current relays, digital I/O pins, PWM drivers, CANBus interfaces, as in Fig. 3b.

b) Graphical User Interface: The rover prototype is also equipped with an HD monitor which provides information to user about telemetry, current rover position and operational state (Fig. 4). In addition, mission commands can be given using the same Graphical Interface. This GUI can also be remoted to a central control station using commercial or open-source desktop sharing applications.

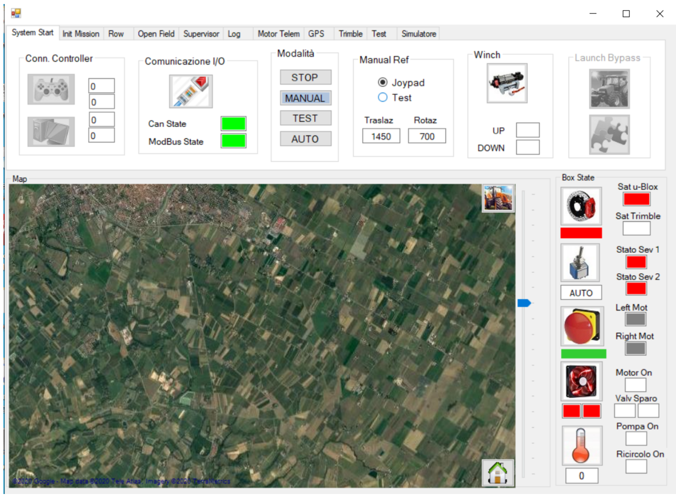


Fig. 4: Graphical User Interface (GUI).

IV. LOCALIZATION, NAVIGATION AND SENSOR FUSION

A multi-rate Extended Kalman Filter (EKF) is used to provide accurate pose estimation by merging the available sensors: Encoders, GPS, IMU readings and LiDAR pointclouds. The EKF "changes" the set of available sensors accordingly to the navigation scenario considered:

- **Open-field navigation**, used to exit from parking place and reach the orchard in a reliable way. This scenario requires the vehicle to strongly rely on GPS positioning, using the LiDAR for obstacle detection and avoidance.
- **Orchard navigation**, used to perform working activities inside the orchard. For this navigation context, the GPS is not used and the LiDAR measurement are processed to provide relative localization with respect to the tree rows, using the Hough Transform algorithm.

A. Open-field navigation

In this context, the rover has been setup to exploit the state of the art ROS Navigation Stack [12]. The idea behind this feature is to have a structured environment with a known map either available a-priori or obtained via a recognition activity across the farm. Using the static farm map, the robot leverages standard graph optimization algorithms, such as A* [13], [14] to compute a set of waypoints leading to the final goal. The localization filter [15] is used to fuse motor encoders, GPS and IMU to achieve a reliable pose estimate, then LiDAR sensor is used to enable obstacle avoidance capabilities, updating the set of waypoints toward the goal, as well as real-time local map refinement to improve navigation. As soon as the target goal waypoint (orchard) has been reached, the controller can switch to the orchard navigation algorithm.

B. Orchard navigation

As the orchard is typically a semi-structured environment organized in several rows of trees aligned one to each other and with known dimensions, there is no specific advantage in using global localization systems, especially when the accuracy is not high enough (e.g. low-cost sensors) or the environment is constantly changing (e.g. vegetation is growing). On the contrary, an accurate local position is of paramount importance to keep the robot on-track. It then follows from the orchard structure that the in-row environment can be modeled as a pair of parallel lines, [9], which can be estimated using the line detection algorithm named Hough Transform (HT) [16].

LiDAR pointclouds are processed and filtered to reduce algorithm complexity and then processed by the ROS HT node we implemented. The latter consists in an iterative procedure that scans the spatial quantized matrix of the LiDAR points image by applying a "vote" [17] for each cell that represents the probability to have a correspondence to the line described by the cell coordinates. Then the algorithm looks for two different lines, which must be parallel and distant from each other about the lane width. This redundant recognition, enforced by parallelism and distance constraints, increases the level of robustness of the algorithm.

One of the contribution of this work is to show that it is possible to optimize the computational complexity of the HT algorithm for line detection, and at the same time further improve its reliability and robustness.

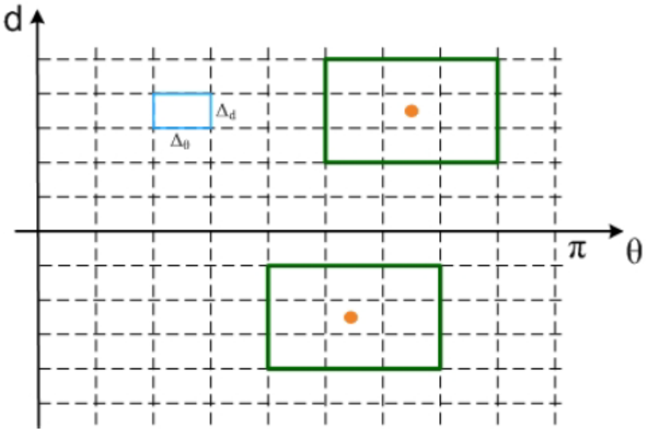


Fig. 5: Search area (green) around last solution found (orange points) for the improved algorithm.

Since the lane structure is preserved along the whole row length, after the first HT iterations on the whole space, the search grid can be reduced, asking the algorithm to find lines that are "close" to the previous solutions (e.g. the lines do not change so much when the rover is inside the row), as represented in Fig. 5. Therefore, the solutions for the new pointcloud frame is not obtained by scanning the entire matrix, but only the cells that are adjacent to those from the previous solutions.

This improvement is also meant to increase the robustness to misdetections due to missing trees along the row, since in this case one of the two lines would be strongly different from the previous ones, but the reduced search span automatically filters out those votes.

According to this navigation scenario, it is therefore possible to see data coming from the HT as additional heading and lateral displacement information with respect to some *orchard-fixed* reference frame, i.e. an additional sensor to be used as input for the EKF. It follows that, in this second scenario, the EKF fuses together IMU, econders and HT estimates to produce a refined position estimate.

As mentioned above, reliable detection of the end of the orchard lane is a key aspect to achieve timely planning of lane-switching trajectories: LiDAR data can be used to detect it. When the end of the lane is approaching, it would not be possible to detect points belonging to the lines in front of the robot, but just behind it: this triggers the trajectory planning for lane switching. But it can happen that, due to missing trees on both sides of the robot, the latter detection provides false positive, to avoid this, Bergerman's group [10] placed optical markers at the end of each row. Since visual markers are inherently prone to deterioration in outdoor applications and the farmers themselves spray products along the whole lane, possibly reducing their visibility, we propose a solution not relying on them.

Since GPS waypoints defining the shape of the different orchard fields are available for open field navigation, it is possible to use them to compute the length of each lane, then use this data, along with the global position provided by the EKF, to trigger the end-of-row recognition reliably, only when actually approaching it.

V. EXPERIMENTAL VALIDATION

As claimed above, simulated results are reported for what concerns open-field navigation, to illustrate both path planning and obstacle avoidance capabilities and the simulated farming environment developed, while for in-row navigation, real data coming from test orchard is presented.

A. Open-field navigation

This benchmark aims to show waypoint navigation capabilities of the rover by selecting a target goal point on the map and then enabling autonomous control by following computed reference trajectory. A goal point is transmitted to the ROS navigation stack and the 2D path planner computes a feasible trajectory that avoids known obstacles. The global map is also updated locally to detect unknown obstacles. If this occurs, a re-planning is automatically initiated to avoid the new obstruction.

Fig. 6 shows the virtual environment used to test path planning and navigation algorithms, which features a simulated orchard field and rest depot for the robot.

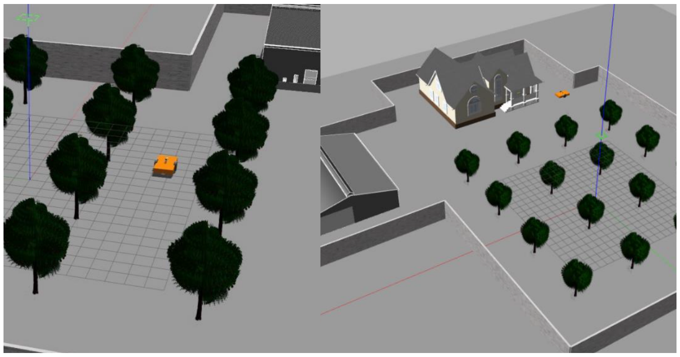


Fig. 6: Gazebo simulation showing the robotic platform inside a mockup virtual orchard

B. Row navigation

In these navigation tests, the robot starts at the beginning of a row inside the orchard. Then the distance between the UGV and the left and right tree rows is measured to benchmark the algorithm performances. Measurement of the smoothness of the trajectory are recorded in order to meet the requirements imposed by the mission typology. Fig. 7 shows the output of the HT as two lines (green and blue) extracted from the LiDAR features as an estimation of the tree rows.

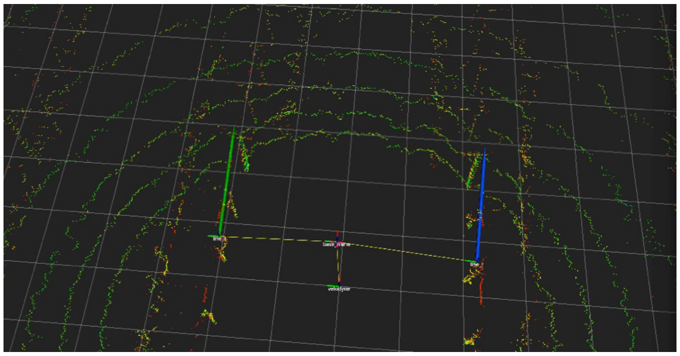


Fig. 7: Hough transform output.

During operation, the distance from the rows reflects some noise, mainly due to the vegetation of the trees (Fig. 8). Furthermore, the correct distance is maintained as the rover proceeded in the middle of the lane: rows width in this case is about 4 meters and an average of 2 meters is measured from both sides.

Detecting foliage and possibly spurious invasive branches of vegetation while performing autonomous navigation is generally a good feature, as the operation may be aware of vegetation and branches that may protrude significantly from the main trunks. Similarly to what a human operator would do, the control algorithm will steer the vehicle to compensate this and adjust the distance from the canopy detected.

Possibly missing or very thin trees have been proved to

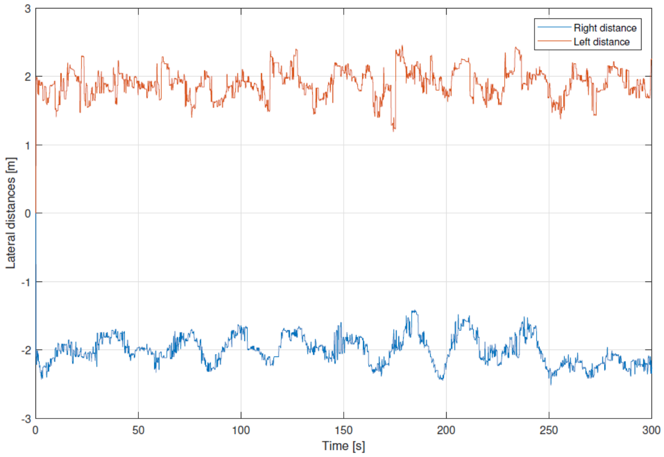


Fig. 8: Hough transform output.

not affect the HT line detection, since the algorithm relies on multiple points "vote" and several trees are interpolated to find correct orientation, besides the reduced search span filters out spurious vote peaks. Furthermore, the intrinsic robustness of the closed-loop controller used to track the planned trajectory, makes it less sensitive to glitches or temporary misplacement of the row lines estimation.

C. Navigation Supervisor

A ROS *supervisor* node has been implemented in order to enable full mission execution and therefore exploiting the two autonomous navigation algorithms to reach at first the working orchard, starting from the parking shelter, and then performing operations inside the orchard itself.

The process consists in defining proper waypoints to travel along existing paths in the farm, using *open-field navigation*, then reaching the border of the orchard and entering a workable row defined with its GPS endpoints coordinates. Once the goal waypoint is reached, the supervisor switches to *in-row navigation*, and once the task is completed, the robot travels home by switching back to *open-field navigation*.

VI. CONCLUSIONS

In this work, we present some aspects of the design and implementation of a multi-purpose robotic platform for orchard precision agriculture.

The structure of the sensor fusion algorithm is reported, highlighting its variable sensors set. We also presented the navigation stack, endowing the UGV with the capability of autonomously navigate from a resting position to the orchard and then to perform its mission inside it. Then we remarked how the structured property of orchard can improve in-row navigation, both to robustify the Hough Transform line recognition and to trigger at the proper time end-of-the-lane recognition. In the end, simulation and experimental results were presented in order to validate the design and navigation algorithms.

Future works will be mainly focused on improving the navigation algorithms, such as obstacle recognition, in order to distinguish human workers the robot has to follow, from other objects occluding the path.

ACKNOWLEDGMENT

The authors would like to warm-heartedly thank Flavio Callegati and Alessandro Samorì for their time and the fruitful discussions, as well as give credits to ASLATECH for the practical realization of the robot.

REFERENCES

[1] S. Sukkarieh, E. M. Nebot, and H. F. Durrant-Whyte, "A high integrity imu/gps navigation loop for autonomous land vehicle applications," *IEEE transactions on robotics and automation*, vol. 15, no. 3, pp. 572–578, 1999.

[2] R. Lenain, B. Thuilot, C. Cariou, and P. Martiner, "A new nonlinear control for vehicle in sliding conditions: Application to automatic guidance of farm vehicles using rtk gps," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 5. IEEE, 2004, pp. 4381–4386.

[3] S. Liu, M. Whitty, and S. Cossell, "Automatic grape bunch detection in vineyards for precise yield estimation," in *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*. IEEE, 2015, pp. 238–241.

[4] G. Riggio, C. Fantuzzi, and C. Secchi, "A low-cost navigation strategy for yield estimation in vineyards," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2200–2205.

[5] J. A. Berni, P. J. Zarco-Tejada, L. Suárez, and E. Fereres, "Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle," *IEEE Transactions on geoscience and Remote Sensing*, vol. 47, no. 3, pp. 722–738, 2009.

[6] C. Lehnert, A. English, C. McCool, A. W. Tow, and T. Perez, "Autonomous sweet pepper harvesting for protected cropping systems," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 872–879, 2017.

[7] F. A. A. Cheein and R. Carelli, "Agricultural robotics: Unmanned robotic service units in agricultural tasks," *IEEE industrial electronics magazine*, vol. 7, no. 3, pp. 48–58, 2013.

[8] S. Blackmore *et al.*, "New concepts in agricultural automation," in *HGCA conference*, 2009.

[9] M. Bergerman, S. Singh, and B. Hamner, "Results with autonomous vehicles operating in specialty crops," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1829–1835.

[10] M. Bergerman, S. M. Maeta, J. Zhang, G. M. Freitas, B. Hamner, S. Singh, and G. Kantor, "Robot farmers: Autonomous orchard vehicles help tree fruit production," *IEEE Robotics & Automation Magazine*, vol. 22, no. 1, pp. 54–63, 2015.

[11] F. Callegati, A. Samorì, R. Tazzari, N. Mimmo, and L. Marconi, "Autonomous tracked agricultural ugv configuration and navigation experimental results," in *Workshop on Small UAVs for Precision Agriculture*, 2018.

[12] ROS.org. Ros navigation stack. [Online]. Available: http://wiki.ros.org/navigation

[13] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 300–307.

[14] ROS.org. Ros global planner. [Online]. Available: http://wiki.ros.org/global/_planner

[15] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," in *Intelligent autonomous systems 13*. Springer, 2016, pp. 335–348.

[16] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[17] L. G. Shapiro and G. C. Stockman, *Computer Vision*, 1st ed. USA: Prentice Hall PTR, 2001.