

## Field Report

# Multi-Sensor Fusion based Robust Row Following for Compact Agricultural Robots

Andres Eduardo Baquero Velasquez<sup>1</sup> , Vitor Akihiro Hisano Higuti<sup>2</sup> ,  
Mateus Valverde Gasparino<sup>3</sup> , Arun Narenthiran V. Sivakumar<sup>3</sup>, Marcelo Becker<sup>4</sup>  and  
Girish Chowdhary<sup>5</sup> 

<sup>1</sup>Department of Agricultural and Biological Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801

<sup>2</sup>EarthSense Co., Champaign, IL 61820

<sup>3</sup>University of Illinois at Urbana-Champaign, Urbana, IL 61801

<sup>4</sup>São Carlos School of Engineering, University of São Paulo, São Carlos, SP, Brazil

<sup>5</sup>Coordinated Science Lab, University of Illinois at Urbana-Champaign, Urbana, IL 61801

**Abstract:** This paper presents a state-of-the-art light detection and ranging (LiDAR) based autonomous navigation system for under-canopy agricultural robots. Under-canopy agricultural navigation has been a challenging problem because global navigation satellite system (GNSS) and other positioning sensors are prone to loss of accuracy due to attenuation and multi-path errors caused by crop leaves and stems. Reactive navigation by detecting crop rows using LiDAR measurements has proved to be an efficient alternative to GNSS. Nevertheless, it presents challenges, due to occlusion from leaves under the canopy. Our system addresses these issues by fusing inertial measurement unit (IMU) and LiDAR measurements in a Bayesian framework on low-cost hardware. In addition, a local goal generator (LGG) is introduced to provide a local reference trajectory to the onboard controller. Our system is validated extensively in real-world field environments over a distance of 50.88 km, on multiple robots, in different field conditions, across different locations. We report leading distance between intervention results for LiDAR+IMU-based under-canopy navigation, showing that our system is able to safely navigate without interventions for 386.9 m on average, in fields without significant gaps in the crop rows.

**Keywords:** sensor fusion, row following, agricultural navigation, agricultural robotics, Kalman filter

## 1. Introduction

Robotics and digital agriculture technologies hold great potential in improving the sustainability, productivity, and access to agriculture (Sparrow and Howard, 2020). Labor remains one of the key issues facing sustainable intensification. As a result, many research groups around the world have

---

Received: 27 May 2021; revised: 31 January 2022; accepted: 6 February 2022; published: 17 June 2022.

**Correspondence:** Andres Eduardo Baquero Velasquez, Department of Agricultural and Biological Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, Email: [andru89@illinois.edu](mailto:andru89@illinois.edu)

This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © 2022 Velasquez, Higuti, Gasparino, Sivakumar, Becker and Chowdhary

DOI: <https://doi.org/10.55417/fr.2022043>

focused their efforts on automating various agricultural tasks, such as planting and harvesting, plant scouting, detection, and treatment of pests and diseases (Neves, 2017; Oliveira et al., 2021; Shamshiri et al., 2018). The expectation is that intelligent automation may provide ways to mitigate the impact of major global problems such as climate change, soil depletion, loss of biodiversity, water scarcity, and population growth (Sparrow and Howard, 2020).

Small agricultural robots, or agbots, could provide new options to farmers who are not enabled by existing large agricultural machines. While large machines certainly are unmatched in tasks requiring a lot of power, such as tilling or harvesting, large machines are not ideal for late-season tasks where they are known to damage the crop. Small autonomous under-canopy robots that may travel in between the rows of crops could provide new tools to growers of commodity crops such as corn (*Zea mays*), soybean (*Glycine max*), cotton (*Gossypium*), and vegetables. For example, small under-canopy robots may enable mechanical weeding (Bakker et al., 2006; McAllister et al., 2019, 2020; Reiser et al., 2019), cover-crop planting (Cavender-Bares, 2021), and under-canopy high-throughput phenotyping (Kayacan et al., 2018; Mueller-Sim et al., 2017; Shafiekhani et al., 2017). This has led to significant interest in autonomous small robots for agriculture which have become the focus of study for various research groups around the world. There are also several companies focused on different form factors, including Naio, Ecorobotix, Row-Bot, and EarthSense, which developed the TerraSentia robot that is used in this study. Shamshiri et al. (2018) and Fountas et al. (2020) provide a systematic review of various applications of agricultural robots and list different research and commercial agricultural robotic platforms developed and used in crop field operations. Such agbots may help farmers to improve yield and productivity while reducing levels of fertilizer and pesticide as well as water wastage (Sparrow and Howard, 2020). Indeed, topsoil compaction has been associated with reduced agricultural productivity. In this context, lighter (weight less than 33 tons) teleoperated or autonomous robots potentially reduce the compaction issue (Fue et al., 2020; Sparrow and Howard, 2020). Furthermore, due to their lower cost, these smaller robots could also make agriculture profitable at *small* as well as large scales, helping improve agricultural diversity, and productivity in developing nations. This would provide an interesting departure from growing farm sizes (and machine sizes) (Chen, 2018; King, 2017) towards appropriate scale mechanization and automation around the world.

However, reliable autonomy remains the key challenge preventing widespread adoption of small robots. Although there are a significant number of publications about the navigation systems for agbots using different sensors such as global navigation satellite system (GNSS), light detection and ranging (LiDAR), or cameras, they are predominantly focused on over-the-canopy robots and larger machines. Among the limited work that exists in under-canopy navigation, extensive field validation in diverse real-world conditions has not been performed and no other work has shown an extensive autonomous navigation as we show in this paper for this type of environment. This paper presents significant field results demonstrating that sensor fusion with LiDAR and an inertial measurement unit (IMU) enables reliable under-canopy autonomy. GNSS is often unreliable under plant canopy and near to the ground due to multi-path and signal attenuation effects. Furthermore, real-time kinematic (RTK) correction is often observed to be not always available, especially in adverse weather and in remote locations. Although cameras are commonly used in robot navigation in indoor environments, they are more sensitive to illumination conditions and also present a shorter range of view compared to LiDAR (Hiremath et al., 2014b; Reina et al., 2016). Both types of sensor (LiDAR and cameras) are affected by weather phenomena (e.g., fog and rain) or other environmental factors (e.g., dust, smoke, and occlusion); the LiDAR is able to provide off-the-shelf distance measurements with accurate ranging over medium range (up to ~30–40 m) and fast operation (Reina et al., 2016).

The key contribution of this paper is the under-canopy navigation system for improving the LiDAR-based row-following task of autonomous ground robots in corn (*Zea mays*). To achieve this, the paper describes the implementation of an extended Kalman filter (EKF) and a local goal generator (LGG). The first improves the estimation of the lateral distances and the robot heading from the perception system outputs using the IMU and encoders information, while the second one creates

feasible paths to smooth the control action behavior. Another contribution is the extensive field testing as a way to validate the proposed system in multiple real-world scenarios. We report 50.88 km of an autonomous solution for under-canopy navigation in corn crops. Our findings present an average distance between interventions of 386.9 m for fields without gaps [Illinois Autonomous Farm (IAF) and Earthsense (ES) #2], 56.1 m for production fields, and 47.5 m for fields (ES #1) with gaps (space of 1 m without plants in both sides of the row). Additionally, we show that the use of EKF improves the distances, and the robot heading estimated by the perception system helps to increase the distances between intervention. It went from 51.6 m without EKF [perception LiDAR (PL)] to 400 m with EKF (PL+EKF) in fields without gaps, and from 16.3 m without EKF to 56.1 m with EKF, in the production fields. Our results also show that the proposed methods lead to significantly improved autonomy when compared with the earlier single-sensor (only-LiDAR-based) attempts. Indeed, the combination of the EKF with LGG significantly increases the distance between intervention without increasing the cost of the overall system. Our previous work (Higuti et al., 2019) demonstrated that low-cost 2D LiDAR is a viable option for navigation in under-canopy environments. It also utilized distance between interventions as a key metric to evaluate the reliability of autonomous navigation in a variety of crop environments. However, the reported results in Higuti et al. (2019) were limited to 41.4 m. On the other hand, our results are highly promising, as they indicate that low-cost LiDAR-based navigation is a reliable and effective means of under-canopy navigation.

## 2. Related Works

Autonomous navigation of agricultural robots has been studied for a long time, especially with tractors. Shamshiri et al. (2018) and Fountas et al. (2020) provide a systematic review of various applications of agricultural robots. They also enlist different research and commercial agricultural robotic platforms developed and used in crop field operations.

Earlier work on autonomous navigation in agriculture was focused on autoguidance of large agricultural machinery. GNSS-based navigation systems have been widely used to allow automatic guidance of traditional crop machinery such as harvesters (Pilarski et al., 2002; Stoll and Kutzbach, 2001) and tractors (Abidine et al., 2004; Bell, 2000; Blackmore et al., 2004; Lenain et al., 2003; Nørremark et al., 2008; Reid et al., 2000; Thuilot et al., 2002; Wang and Noguchi, 2016; Zhang et al., 2016). In addition, researchers have also investigated the potential of using GNSS-based navigation systems for small autonomous robots that navigate over the canopy on horticultural crops such as sugar beet (Bak and Jakobsen, 2004; Bakker et al., 2011). These methods work reasonably well when there is a direct line of sight to GNSS satellites, but suffer from inaccuracies due to multi-path errors when the robot is under the canopy. The multi-path error happens because of thick leaf canopies in crops like corn, soybean, cotton, sugarcane, sorghum, and taller vegetable crops such as okra. Robotanist, an under-canopy robotic platform developed for sorghum phenotyping, is an example (Mueller-Sim et al., 2017). It uses GNSS for navigation, and hence faces challenges when the sorghum plants grow taller than the level of the GNSS antenna. The authors note that GNSS navigation is a capability that is lost as the season progresses and sorghum plants become taller than the GNSS antenna. Also, GNSS is not reliable in some countries in the Southern Hemisphere, because of the South Atlantic Magnetic Anomaly (SAMA) that affects the GNSS signal (Abdu et al., 2005; Spogli et al., 2013). Additionally, since GNSS is an extrinsic sensor that only provides the position of the robot, it cannot overcome challenges such as the presence of static and dynamic obstacles such as rocks and animals in the planned GNSS waypoint path (Reina et al., 2016; Rovira-Más et al., 2015). This shows that GNSS-based navigation cannot be a stand-alone solution for autonomous agricultural robots. In particular, GNSS is not reliable for autonomous navigation in orchards and under-canopy environments (Bergerman et al., 2015; Mueller-Sim et al., 2017; Santos et al., 2015). Other commercial systems such as Naio, Row-Bot, etc., seem to also navigate primarily using GNSS. Therefore, work on GNSS free navigation, such as pursued in this paper, is of broad importance to the agricultural robotics community.

Vision-based navigation systems have been extensively studied for the row-following problem in agricultural fields. The geometric structure of the crop rows as parallel lines in agricultural fields enables the use of computer vision. As in the case of GNSS systems, earlier works focused on autoguidance for tractors and heavy machinery such as harvesters (Reid and Searcy, 1987). Subsequent works also focused on vision-based navigation of over-the-canopy agricultural robots (Ball et al., 2016; Jiang et al., 2015; Montalvo et al., 2012; Zhai et al., 2016; Zhao and Zhang, 2016). The camera looks down at the crop rows from a top-down view in these cases and a clear view of multiple crop rows as vanishing lines is visible in the images. Therefore, the common approach is the segmentation of vegetation from the soil background, using color indices and line fitting. This is possible with different approaches such as the Hough transform. From the fitted lines, the relative orientation and offset of the robot with respect to the crop row is obtained and used to steer the robot to the middle of the lane between the crop rows (English et al., 2014; Hiremath et al., 2014a; Radcliffe et al., 2018; Subramanian et al., 2006; Zhang et al., 2018). Also, sensor fusion with GNSS was used in a lot of these cases to improve the robustness of the system. However, this approach faces challenges in under-canopy environments such as row crops, because of the presence of significantly more clutter and occlusion. Xue et al. (2012) developed and tested a vision-based navigation system for under-canopy navigation in corn crops. Yet, its validation was limited to cases when the occlusion and clutter are little or insignificant, and the vegetation and soil are distinguishable. In general, this classical vision-based navigation approach is affected by many factors. Some of them are continuously dynamic lighting conditions and appearance of crops and soil in the field throughout the season. VINS-Fusion (Qin et al., 2018) is a popular simultaneous localization and mapping (SLAM) algorithm for robot navigation that uses visual-inertial sensors to provide high-accuracy odometry. According to Qin et al. (2018), it is primarily designed for state estimation and feedback control of autonomous drones. Notwithstanding, it is also capable of providing accurate localization for augmented reality (AR) applications. While VINS-Fusion is mostly used in indoor and well-structured outdoor (urban) environments, it is rarely used in agricultural ones.

LiDAR sensors are active sensors that sense geometric information of the environment. They are commonly used for mobile robot navigation (localization and obstacle detection) in different applications (Pouliot et al., 2012; Reina et al., 2016; Siegwart et al., 2011). In the context of agricultural navigation, LiDAR has been widely studied, mostly in orchard environments (Barawid et al., 2007; Bell et al., 2016; Bergerman et al., 2015; Lemos et al., 2018; Zhang et al., 2013). They use 2D LiDAR sensors and similar line fitting on images to detect crop rows. Lines are fit on the 2D laser scans to detect the orientation and offset of the robot with respect to the crop rows. Some work also studied the use of 3D LiDAR sensors to overcome the problem of vegetation occluding tree trunks. Thereby, it results in noisy 2D LiDAR scans where fitting lines is challenging (Bell et al., 2016; Zhang et al., 2013). Since commercially available 3D LiDAR sensors are expensive, Zhang et al. (2013) used a planar LiDAR, and added a rotational degree of freedom to capture 3D point clouds but this approach adds complexity to the system. A push-broom-based modification to 2D LiDAR was another approach that was tested to capture 3D point clouds (Baldwin and Newman, 2012). Row crop environments such as corn and sorghum are very different from orchards. Unlike in orchards, where individual trees are separated by a large distance and also subsequent rows have a large gap between them, the canopy is dense in row crops and also the row spacing is limited to around 0.8 m in corn/sorghum. This results in significant occlusion and clutter present in the LiDAR scans making it challenging to find the crop rows in the scans. Hiremath et al. (2014b) developed a 2D LiDAR-based navigation system for corn, using a particle filter to handle noise present in the environment. However, tests were limited to only a very early growth stage of the crops. Troyer et al. (2016) used a 2D LiDAR to develop a navigation system for corn but they did not validate their approach in a real field or plants. In late growth stage corn and sorghum, as a robot navigates under the canopy, the above mentioned challenges of clutter and noise appear in LiDAR scans. This requires careful analysis of the scans to detect the crop rows as shown by Higuti et al. (2019). While using sensor fusion with encoders and IMU the robustness of under-canopy navigation is improved, as shown in our results.

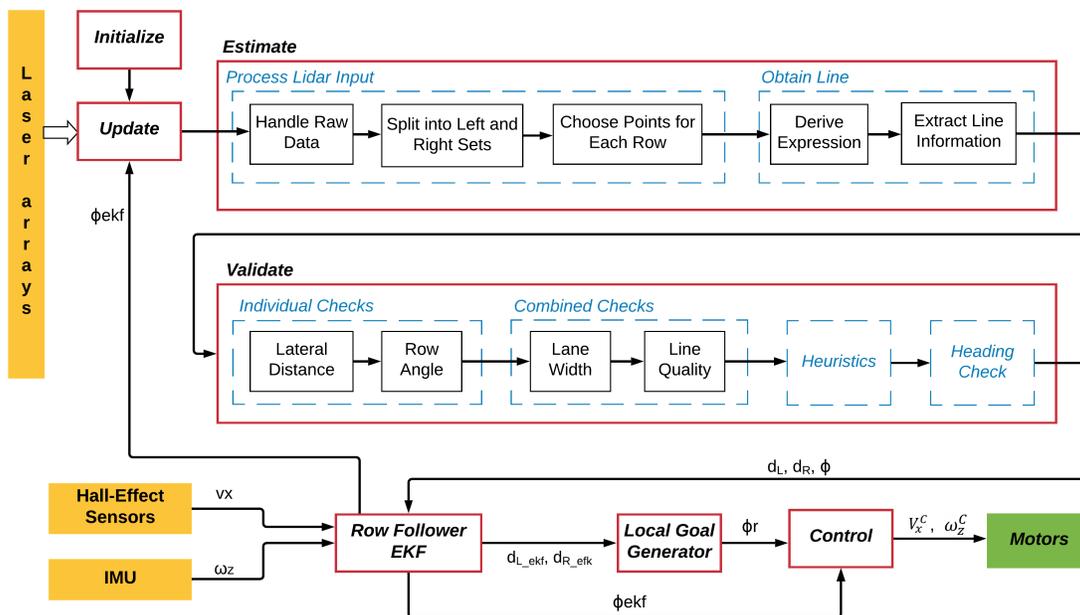


Figure 1. Navigation system diagram.

### 3. System Design

The proposed navigation system (Figure 1) is composed by seven steps: **Initialize**, **Update**, **Estimate**, **Validate**, **Row Follower EKF**, **Local Goal Generator**, and **Control**. The first four steps make up the perception subsystem that uses the raw LiDAR measurements from the surroundings to build two virtual representation of the rows. These virtual walls are used to estimate the distances ( $d_R$  and  $d_L$ ) and the robot's heading ( $\phi$ ). Then, **Row Follower EKF** gets the linear ( $v_x$ ) and angular ( $\omega_z$ ) velocity estimations from the Hall-effect sensors, as well as the IMU to correct  $d_R$ ,  $d_L$ , and  $\phi$ . Afterwards, the corrected distances are used by the **Local Goal Generator** to generate a reference heading ( $\phi_r$ ). Finally, the **Control** calculates the control action ( $\omega_{c_z}$ ) using  $\phi_{ekf}$  and  $\phi_r$ .

#### 3.1. Robotic Platform

The TerraSentia robot system is a compact and autonomous robot designed particularly for high-throughput phenotyping. It was conceived by Distributed Autonomous Systems Laboratory (DASLAB) at the University of Illinois at Urbana-Champaign. Since 2017, it has been manufactured and commercialized by EarthSense. TerraSentia has been designed to be rugged for all-season field deployment, yet remaining lightweight (16.55 kg with battery) and low cost. The robot dimensions are 0.54 m × 0.32 m × 0.35 m (length × width × height) and because of this it is able to operate under the canopy in crops with row spacing as little as 0.4 m. Its body, which was designed to ensure the system does not harm young plants, is made of hard plastic and reinforced by a lightweight metal frame.

The robot has four Maytech Brushless Outrunner Hub Motors with Hall-effect sensor to drive each of its four wheels, which are built through additive manufacturing with polylactic acid. Each motor is positioned in the middle of the wheel and driven by a custom version of the VESC4 motor controllers. For navigation, TerraSentia has embedded a Bosch BNO055 IMU and a Zed-F9P GNSS manufactured by Ublox. These sensors are integrated in a main board manufactured by EarthSense.

Two 2D Hokuyo LiDAR (UST-10LX) scan the environment. The first one is positioned in the front part and it provides a surrounding's horizontal scan, which serves as input of the proposed



**Figure 2.** (a) TerraSentia robot in a corn crop. (b) CAD drawing of the robot.

navigation system. The second one is positioned in the rear part to provide a vertical scan that is used to make an offline 3D crop reconstruction. Additionally, four wide-angle full HD USB 1080p camera modules record videos to extract meaningful phenotype features using the automatically time- and geotagged data from them. This includes plant count, stem width, leaf area index under the canopy, plant height, and disease detection (Choudhuri and Chowdhary, 2018; Kayacan et al., 2018).

All sensors are connected to a Raspberry Pi 3, which serves as the main autonomous computer of the robot for the purpose of this paper. It acquires the readings from the front LiDAR, IMU, and Hall-effect sensors, runs the proposed navigation system, and generates the desired control signals (desired robot angular and linear velocities). Then, it transforms the control signals to the desired velocity for each motor, and sends these values to the VESC4 controllers as pulse width modulation (PWM) signals to drive the motors. Additionally, the robot has an Intel NUC i7 computer with 500GB SSD and 16GB RAM to store all collected data.

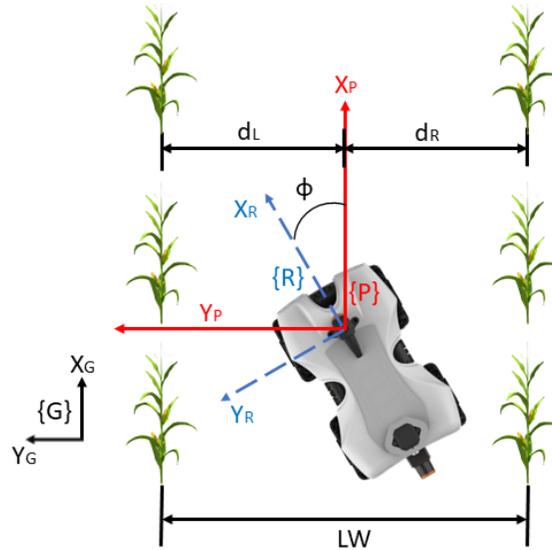
### 3.2. LiDAR-Based Perception Subsystem (PL)

The core perception subsystem is composed of four stages: **Initialize**, **Update**, **Estimate**, and **Validate**. They are thoroughly described by Higuity et al. (2019) with step-by-step and numeric examples from situations where rows are easily visible, and also when leaf occlusion is predominant. This section brings an overview to show how LiDAR readings and previous estimates, if available, are used to estimate current lateral distances and the robot's heading angle.

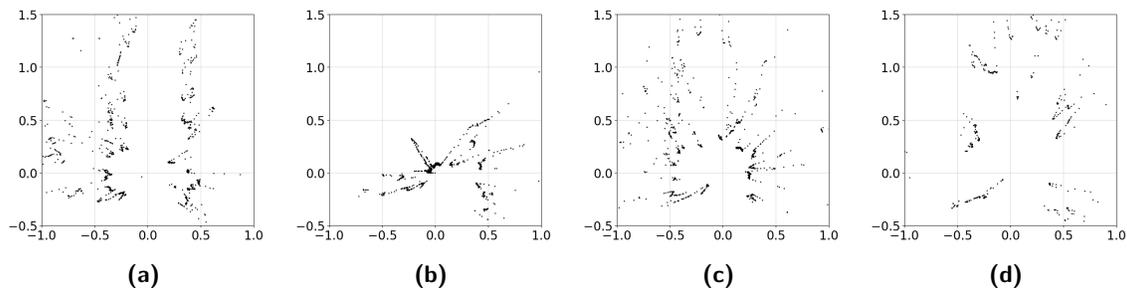
In the first stage, the configuration files are parsed and stored as variables in the appropriate classes, and the robot's embedded devices are initialized. These configuration files contain the information about the physical characteristics of the robot, threshold values, function limits, main crop characteristics, and other initial conditions. During the **Update** stage, the LiDAR readings, the robot's current orientation angle  $\phi$ , and the previous estimates, if available, are obtained and sent to the **Estimate** stage. Also, the update stage receives the constants set from the **Initialize** stage during the first loop. All these variables are used in the **Estimate** stage to filter out extraneous information and create a data set to determine the new values for  $d_R$ ,  $d_L$ , and  $\phi$ .

In the first step (*ProcessLiDARInput*) of the **Estimate** stage, the raw LiDAR readings are projected from polar to Cartesian coordinates, in the robot local frame ( $\{R\}$  in Figure 3). Then, they are transformed from the local to the path frame ( $\{P\}$  in Figure 3) using the robot's heading ( $\phi$ ) to compensate the robot's orientation deviations.  $\{P\}$  is a translation of the global coordinate frame  $\{G\}$ , which is located at a corner of the corn crop, to the origin of  $\{R\}$ .  $\{P\}$  and  $\{R\}$  have their origins in the geometric center of LiDAR. The first has its  $x$  axis ( $X_P$ ) always parallel to one of the rows, and the latter rotates with the robot. Also  $\phi$  is considered as the angular difference from  $X_P$  to  $X_R$ .

Figure 4 brings examples of the LiDAR scan's projections. Figure 4a shows a LiDAR scan, when the robot is around an ideal scenario, where the rows are parallel, and there is absence of hanging



**Figure 3.** Global (G), path (P), and robot (R) reference frames. The origins of P and R are fixed at LiDAR geometric center. Robot y axis ( $X_R$ ) is aligned with robot's longitudinal axis; path y axis ( $X_P$ ) is parallel to rows.  $\phi$  is the angle between these two x axes.  $d_R$  and  $d_L$  are the orthogonal distances between the center of the sensor and each adjacent row.



**Figure 4.** LiDAR scans from experiments in corn fields. (a) Well-behaved case with both rows clearly distinguishable and parallel. (b) Sensor occlusion caused by hanging leaves. (c) Portion of right row hidden by leaves. (d) Sparseness combined with fallen leaves causing a difficult environment to understand.

leaves, fallen stalks, or gaps. These situations may generate total occlusion of the sensor (Figure 4b) or hide portions of the rows (see top part of right row in Figure 4c). Finally, Figure 4d shows a scenario where the combination of sparse stalks with fallen leaves makes it challenging to determine the crop rows in the LiDAR scan.

As a single LiDAR scan has 1081 distances on a plane with  $270^\circ$  angular range, there are some readings without any relation with the rows. For that reason, the readings inside of a fixed rectangle ( $y \in (-D_y, D_y), x \in (D_{x0}, D_{x1})$ ) around the origin of the frames {R} and {P} are used in the next steps.  $L_x$  is defined to consider the points related to neighboring rows while the points of the farther rows are discarded.  $D_{x0}$  was defined as 0.1 m in order to discard the points behind the  $Y_P$  and  $D_{x1}$  was 1.9 m. Opposed to previous work, the distance readings behind LiDAR are not considered anymore because the leaves located in the rear part of the robot usually form a dense curved shape which makes multiple starting points for straight lines generate wrong estimations of the distances and the robot's heading. We observed that this problem is more common when the robot is used to navigate in denser crops. Once the LiDAR readings are filtered, they are split into two sets: right and left sides. The use of  $\phi$  to rotate the raw readings helps to guarantee that one of the captured

rows is almost parallel to the  $X_P$  axis. This condition allows the use of a histogram to detect where the relevant information about the position of each row is. The histogram generates two peaks (one for the right side and the other one for the left side) and all the readings around these peaks are considered to obtain the virtual walls in the *ObtainLine* step.

In the *ObtainLine* step, the least minimum squares linear regression is used to obtain the virtual lines. Also the orthogonal distance to such segments provides the values of  $d_R$  and  $d_L$ . Some properties (such as length, standard deviation, difference between current orthogonal distance/slope with previous ones) of the virtual lines are used to determine the quality of the estimation, and they are used by a heuristic to define if the distances are valid or not. If they are valid then their values are sent to the **Row Follower EKF**. If they are not valid, then their last valid value is used in the EKF. The validation process happens in the **Validate** stage.

### 3.3. Extended Kalman Filter (EKF) for Perception

To improve performance of lateral distances and heading estimations, this study takes advantage of the complementary characteristics of the perception algorithm that estimates the absolute position of the robot in relation to the crop row. The robot's embedded sensors estimate the robot's relative displacement at every time instant. Thus, an EKF was chosen to provide a better estimation of the true values, given the fusion between available sensors.

To provide a reference frame for the mechanization equations that represent the movement of the robot in relation to the rows, a new coordinate frame is defined. This new coordinate frame  $\{N\}$  assumes two straight rows in the form of an aisle with constant distance between the rows, as shown in Figure 3. The inertial coordinate frame  $\{N\}$  is defined with origin localized in the entrance of the corridor. The  $x$  axis of coordinate frame  $\{N\}$  shows the longitudinal distance along the path. The  $y$  axis indicates the perpendicular distance across this same path and indicates the robot's cross-track error along the path. For reference purposes, the  $z$  axis points upwards.

A differential system of equations may be written to represent the movement of the robot along this path, as shown in Equation (1).  $p$  is the three-dimensional position of the robot in relation to the origin of  $\{N\}$ ,  $C_{\{R\}}^{\{N\}}$  is the rotation that aligns the frame  $\{R\}$  to the frame  $\{P\}$ , and  $v$  is the three-dimensional robot's velocity written in terms of the local frame  $\{R\}$ :

$$\dot{p} = C_{\{R\}}^{\{N\}} v. \quad (1)$$

We assume displacements along the inertial  $z$  axis, and rotations around  $x$  and  $y$  axes are negligible. Thus, a planar version with only the  $x$  and  $y$  axes is used. Equation (2) shows the rotation matrix that transforms from the local robot's body frame to the  $\{N\}$  inertial frame using this planar assumption:

$$C_{\{R\}}^{\{N\}} = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix}. \quad (2)$$

And then

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \end{bmatrix} = \begin{bmatrix} v_x \cos \phi - v_y \sin \phi \\ v_x \sin \phi + v_y \cos \phi \end{bmatrix}. \quad (3)$$

$p_y$  is a metric of cross-track error, and for this specific problem, we define cross-track error as  $0.5(d_R - d_L)$ . Also, the sum  $d_R + d_L$  is assumed constant and it is equal to the crop lane width  $LW$ . Since the lateral distances  $d_R$  and  $d_L$  are the values of interest in this project, the differential equation may be rearranged to highlight these metrics, as shown in Equation (4):

$$\begin{bmatrix} \dot{p}_x \\ \dot{d}_L \\ \dot{d}_R \end{bmatrix} = \begin{bmatrix} v_x \cos \phi - v_y \sin \phi \\ -v_x \sin \phi - v_y \cos \phi \\ v_x \sin \phi + v_y \cos \phi \end{bmatrix}. \quad (4)$$

Further extension of state estimation is given by adding the estimation of the angle  $\phi$ , where  $\dot{\phi} = \omega$ . Since displacement on the robot's longitudinal axis, i.e., the  $x$  axis, is not a state that will contribute to the task of row following, it is no longer used in the formulation, to reduce calculation complexity. In addition, because a nonholonomic model formulation is adopted to represent the robot's kinematics, the lateral velocity  $v_y$  is also disregarded. The control variables  $v_x$  and  $\omega$  are chosen as inputs of our differential system, such that  $u = [v_x, \omega]^T$ :

$$\begin{bmatrix} \dot{d}_L \\ \dot{d}_R \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -v_x \sin \phi \\ v_x \sin \phi \\ \omega \end{bmatrix}. \quad (5)$$

Using the differential system stated in Equation (5) and solving it numerically using the forward Euler method, a solution model is presented, where  $\hat{x}_k = f(x_{k-1}, u_{k-1})$ . The input  $u_k$  is provided by embedded sensor measurements, which we define as  $u_k = (v_x, \omega)$ . Because the model is updated using noisy sensor measurements, and these measurements are susceptible to uncertainties, then we assume a process uncertainty term  $\Delta_k$  added to the prediction model. The resultant equation is  $\hat{x}_k = f(x_{k-1}, u_{k-1}) + \Delta_k$ , and the uncertainty is assumed as a zero mean Gaussian noise. This solution model is used to predict future robot positions according to the specified coordinate system  $\{N\}$ :

$$\begin{aligned} \hat{x}_k &= f(x_{k-1}, u_k) + \omega_k, \\ z_k &= h(x_k) + \nu_k. \end{aligned} \quad (6)$$

To correct state prediction from the obtained mechanization model, an extended Kalman filter framework was chosen. It accounts for the uncertainties in distance and heading estimation (Thrun, 2002). We model the process with Equation (6) where estimated state  $\hat{x}_k$  is a function  $f(\cdot)$  of past state  $x_{k-1}$  and control inputs  $u_k$ , as shown in Equation (7).  $dt$  is the time step used in the forward Euler method and represents the time taken to run one EKF iteration:

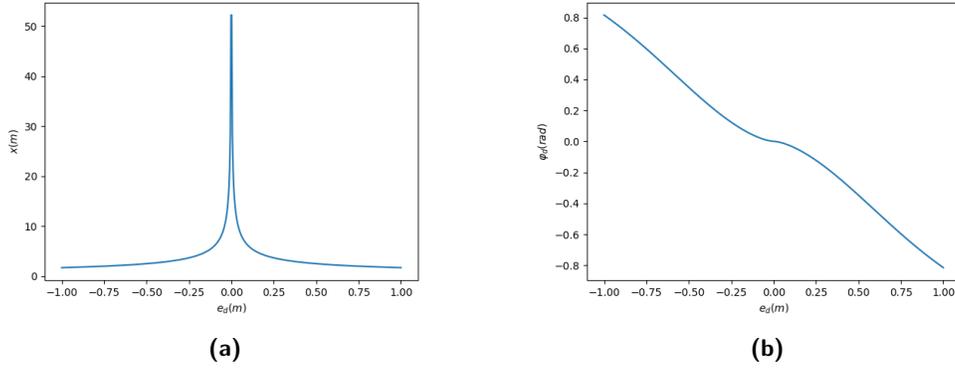
$$f(x_{k-1}, u_k) = \begin{bmatrix} \hat{d}_{L_k} \\ \hat{d}_{R_k} \\ \hat{\phi}_k \end{bmatrix} = \begin{bmatrix} d_{L_{k-1}} - v_x \sin \phi_{k-1} dt \\ d_{R_{k-1}} + v_x \sin \phi_{k-1} dt \\ \phi_{k-1} + \omega dt \end{bmatrix}. \quad (7)$$

To calculate the predicted covariance estimate, the Jacobian of function  $f(\cdot)$  evaluated at each time step is used. The Jacobian matrix is shown in Equation (8). The function  $h(\cdot)$  relates the state  $x_k$  to the measurement  $z_k$ , and they are directly obtained from the LiDAR-based perception model as shown in Equation (9). The Jacobian function  $h(x_k)$  to calculate the updated covariance estimate is the identity matrix. In our implementation, we assume the process and measurement covariance matrices are constant. Therefore,  $\mathbf{Q}_k = \mathbf{Q}$  and  $\mathbf{R}_k = \mathbf{R}$ .

$$F_k = \begin{bmatrix} 1 & 0 & -v_x \cos \phi_k dt \\ 0 & 1 & v_x \cos \phi_k dt \\ 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

$$h(x_k) = \begin{bmatrix} d_{L_k} \\ d_{R_k} \\ \phi_k \end{bmatrix}. \quad (9)$$

In the prediction step,  $f(\cdot)$  (Equation (7)) estimates the state using the linear velocity  $v_x$  and angular velocity  $\omega_z$ , respectively, from wheel encoder measurements, and an embedded gyroscope sensor. Subsequently, in the **Update** step, the innovation occurs by taking into account the calculated values of lateral distances and the robot's heading that are obtained from the LiDAR point cloud (either raw 2D or processed 3D). The IMU sensor used in our experiments contains an autocalibration algorithm that runs in the background and removes the accelerometer and gyroscope measurement bias (Sensortec, 2020). Therefore, an additional bias estimation is not necessary and is not present in our EKF formulation.



**Figure 5.** (a) Desired trajectory given by Equation (10). (b) Desired behavior of  $\phi_r$ .

### 3.4. Local Goal Generator (LGG) for Row Follower and Controller

The local goal generator is a function that simulates a vector field where the value only varies in  $x$ . The behavior of the field is designed to point the robot towards the middle of the lane, and gradually turns the robot to the normal orientation as it approaches the middle. This reduces the oscillatory behavior that arises when the control action is only based on the position, which commonly leads to oversteering. The vector field (Equation (11)) was defined as the arctangent of the derivative of Equation (10). It is used to establish the trajectory that the robot should follow, when it is in a row:

$$f(x) = \left( \frac{b}{|e_d|} \right)^c, \quad (10)$$

$$\phi_r = \arctan \left( \frac{df(x)}{dx} \right) = -\arctan \left( c \left( \frac{e_d}{e^2} \right) b^c \left( \frac{1}{|e_d|} \right)^{c+2} \right), \quad (11)$$

where  $e$  is the curve width constant,  $a$  is the curve smoothness constant,  $c$  is the curve deadband constant,  $d_{ref}$  is the reference distance, and  $e_d$  is the error distance which is defined as  $e_d = d_{ref} - 0.5(d_{R-ekf} - d_{L-ekf})$ .  $d_{ref}$  is zero because it is expected that the robot travels following the lane in order to minimize possible collisions with lateral rows. Figure 5a shows the trajectory given by Equation (10) and Figure 5b shows the desired behavior of  $\phi_r$ .

The output of the local goal generator is the target heading ( $\phi_r$ ). The next step of the navigation system is a proportional, integral and derivative (PID) controller used to keep the robot in the path center. Its outputs are  $\phi_r$  and  $\phi_{ekf}$  and its output is an angular velocity ( $\omega_z^C$ ) command which is sent, together with the linear speed ( $V_x^C$ ) command, to the motors.

## 4. Success Metric

A successful autonomous run of mobile robots is often subjected to assessment of how well it followed the desired path. Higuti et al. (2019) discusses two derived values from perception subsystem lateral distance outputs: cross-track error (CTE) and lane width. The first is the distance between the robot and the reference path, i.e., the middle of the lanes. It is given by half of the difference between right and left lateral distances,  $CTE = 0.5(d_R - d_L)$ . The second is the measured navigable space and it may be compared to nominal row spacing, when seeds were planted. It is given by the sum of lateral distances  $Lw = d_R + d_L$ .

Such values highlight the need for ground truth which will either give the robot's exact positioning over time to compare with cross-track error or provide what the lane width should be. Nevertheless, determination of ground truth in crop environments faces a variety of challenges. Commonly used for outdoor applications, RTK-GNSS suffers from signal degradation due to loss of satellite view and

multi-path errors under canopy. A stick to raise the GNSS antenna above coverage is not feasible since plants are more than 2 m tall, and such a stick would get entangled and heavily oscillate due to ground unevenness. Markers placed on the ground would require constant maintenance because the robot would go over it, which also raises a ground condition change. Since LiDAR itself provides raw distance measurements, a manual ground truth may be obtained from them and the results may be directly compared. For such a task, 1 of every 10 scans is analyzed, and the visually best fitting line is annotated for each side, if available.

Since RTK-GNSS is often used for ground truth, to show how manual ground truth compares to that, the upper part of five rows on a late stage corn crop was cut down from nominal 2.4 m to 1.4 m (Figure 7e) to reduce the environmental influence on GNSS signal. To be easily comparable to perception's cross-track error, the GNSS readings are converted to Cartesian coordinates using the Haversine formula, where changes in  $x$  axis refer to longitude displacements with fixed latitude, whereas  $y$ -axis values refer to latitude displacements with fixed longitude. A subsequent reference transformation brings position on the lane start to the origin, and flattens the following positions inside a straight lane to the  $x$  axis. Finally, the CTE is matched on a timestamp basis and plotted as  $y$  displacements against the flattened  $x$  values.

Given that the sensor operates on a 40 Hz scan rate, manual ground truth is hardly scalable as field experiments increase in quantity and duration. Since the robot is still in a phase where it is followed by an operator, the current autonomous mode is expected to relieve people from the tedious task of driving the robot while going through crop lanes. Therefore, the success metric adopted in this work is the *distance per intervention*, that is, how much crop was covered between interventions, which is defined as situations where failures in perception or control subsystems required the operator to recover manual control of the robot. It also counts situations where system fails soon after starting (bad start) and failures due to presence of gap, an important feature in research fields.

## 5. Experimental Results

Field experiments were carried out aiming to evaluate the perception and control subsystems on corn crops. They may be divided into two categories, as follows: (1) controlled tests and (2) uncontrolled tests.

The controlled tests were designed to show the improvement obtained, with the use of EKF (for perception) and the local goal generator, in relation to our previous navigation system (Higuti et al., 2019). Additionally, these tests were used to pinpoint the failure sources when the TerraSentia is used in different field situations (rows with a curve, border rows, and uneven terrains). This is opposed to previous study, where corn experiments were restricted to well behaved lanes, and sorghum experiments varied from well behaved to intensely cluttered, but they were limited to 3 m of lane length.

The uncontrolled tests used only the proposed system (referred to as PL+EKF+LGG). They have been performed by third parties with no specific knowledge about the perception and control subsystems. For such cases, the autonomous navigation is an auxiliary tool for another research (mainly a collection of visual data along rows) in states in the USA (Illinois and Missouri). These tests exposed the robot to diverse field conditions, and also different user cases as going through the lane from beginning to end could not be the main goal, but rather running a few meters to capture a specific part of the crop for offline analysis.

### 5.1. Experimental Setup

For the validation tests (controlled and uncontrolled tests) presented in this paper, the core perception subsystem ran with the same configurations as the one presented in Higuti et al. (2019) except for nominal lane width, which was set to 0.75 m since all fields share this characteristic within a 0.10 m interval, and desired forward speed, which was raised from 0.22 to 0.7 m/s.

The EKF model and measurement error covariance matrices,  $\mathbf{Q}_k$  and  $\mathbf{R}_k$ , respectively, were set as constant diagonal matrices whose values were determined from the expected error of the



**Figure 6.** (a) UIUC Illinois Autonomous Farm (IAF) research field. (b) Production field farm with curve. The red lines indicate the rows used for the controlled tests.

sensors (LiDAR, encoders and IMU).  $\mathbf{Q}_k$  was set to  $diag(0.001\ 0.001\ 0.01\ 0.01)$  while  $\mathbf{R}_k$  was set to  $diag(0.05\ 0.05\ 0.5\ 0.5)$ . The parameters of the local goal generator were empirically determined using Equation (11). We chose initial values for  $b$ ,  $c$ , and  $e$  to generate the plot of different trajectories, then we chose the one with the softest shape to test it in the field. Finally, we fine-tuned the parameters during field tests and their final values are  $b = 3.8$ ,  $c = 0.55$ , and  $e = 0.7$ .

We tuned the proportional and derivative (PD) coefficients by hand in the field. We initialized  $K_p$  and  $K_d$  to zero. We started with  $K_p$ , reducing overshoot. We then adjusted  $K_d$  until the system was no longer underdamped. The integral gain  $K_i$  is not used because there is no observable steady error in the system behavior. Our final coefficients were  $K_p = 0.7$  and  $K_d = 1.5$ .

To allow usage of the monocular camera version of the off-the-shelf ROS package<sup>1</sup> VINS-Fusion (Qin et al., 2018), further referred to as VINS-mono, the robot’s front camera and IMU readings were suitably collected in rosbag format for a manual run in lane G. Additionally, the extrinsic and intrinsic parameters of the robot’s front camera were found and updated in the configuration files. Finally, VINS-mono offline uses such data to produce odometry estimations, which may be compared to CTE inside a corn crop.

## 5.2. Controlled Tests

The controlled tests were performed in two fields (Figure 6): (1) the UIUC Illinois Autonomous Farm research field (further referred to as IAF) and (2) a production field farm with a curve. Both fields are located in Urbana-Champaign, Illinois, USA, and their main characteristics are (1) nominal row spacing equal to 0.75 m, (2) plant spacing around 0.1 m, except for lanes with random sparseness or artificially induced gaps, and (3) growth stage R4–R5.

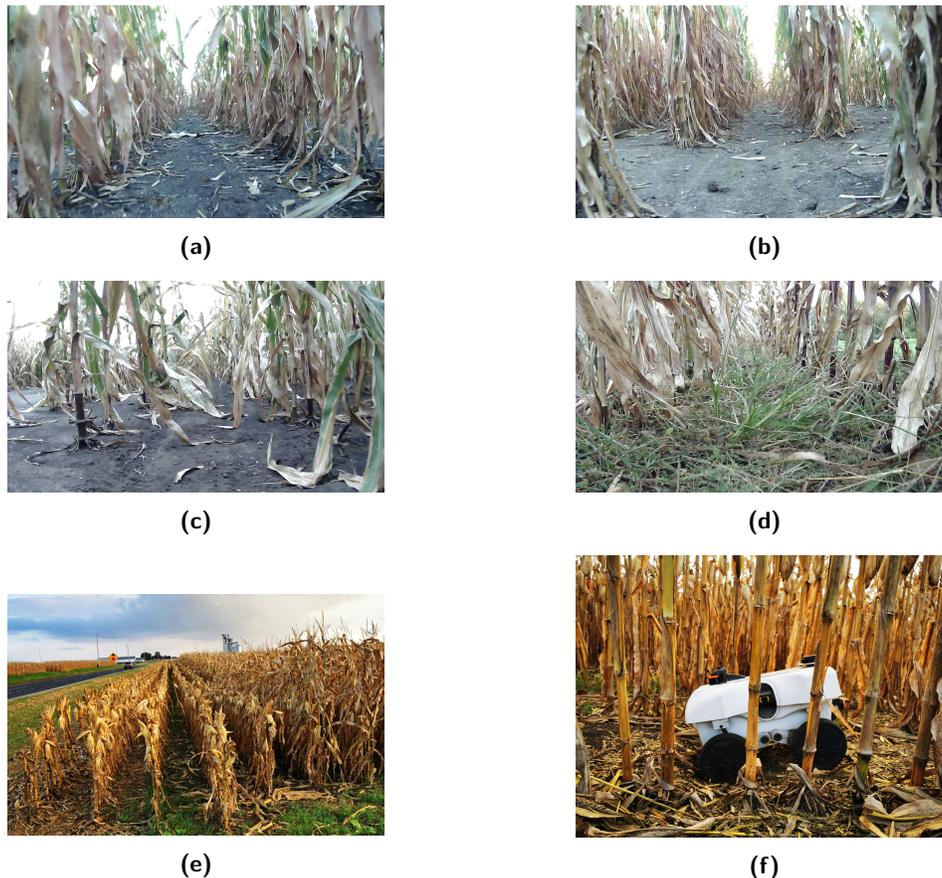
### 5.2.1. Controlled Tests in IAF

Six situations in IAF were used to evaluate the introduction of EKF and LGG modules into the system. The first scenario (lane A) is the best case where both rows exist from the beginning to the end of the lane, and they are straight and mostly parallel (Figure 7a). The second (lane B) is similar to the first one. However, it has 12 gaps from 0.5 m to 1 m and 1 gap of 2 m (Figure 7b). The lengths of both rows were 220 m. The third scenario used the same lanes and the LiDAR readings were acquired with different update rates of 10 and 5 Hz. The fourth and fifth were 125-m-length rows characterized by sparseness and fallen stalks (Figure 7c) or the presence of high grass and weeds (Figure 7d). Finally, the sixth scenario (lanes G and H) had 90-m-length rows, where the plant tops were cut (lowering height from 2.4 to 1.4 m) to get better RTK corrections along their length (Figure 7e). Additionally, lane G had artificial bumps (Figure 7f) in its first 30 m to check

<sup>1</sup> <https://github.com/HKUST-Aerial-Robotics/VINS-Fusion>

**Table 1.** Number of interventions on lanes A (straight with gaps) and B (straight and continuous) in IAF.

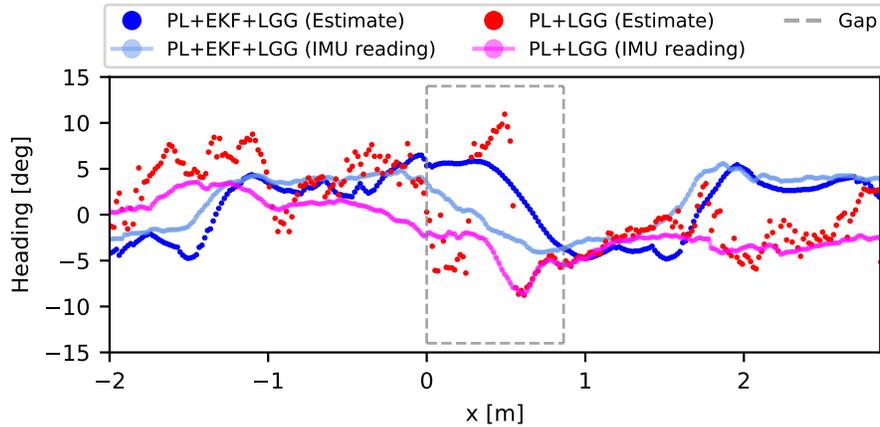
	Run (Lane)			
	1(A)	2(B)	3(A)	4(B)
PL+EKF+LGG	1	1	1	2
PL+LGG	1	5	2	4



**Figure 7.** Scenarios used for the controlled tests in the Autonomous Farm. (a) Lane with continuous rows (without gaps) on both sides. (b) Lane with rows where there are small gaps (less than 1 m) and one big gap (around 2 m) located close to the end. (c) Internal border lane with missing plants (only on one side) and fallen stalks inside it. (d) External border lane (without gaps) with grass, weeds, and clutter. (e) Upper part of five rows was cut down to improve RTK GNSS accuracy. (f) TerraSentia with front wheels (left side) in one of the artificial holes in the uneven lane.

the system behavior when the robot is running in this kind of terrain. While all lanes A–H provided field data to compare the estimation performance for the full system (PL+EKF+LGG) against the system without EKF (PL+LGG), tests with respect to LGG were carried out on lanes G and H.

Lane A provides a baseline as it is the less externally influenced. Table 1 shows that five interventions happened in runs 1 and 3. Except for the second one for perception LiDAR without EKF (PL+LGG), all of them were end-of-lane interventions, i.e., the robot got to the end of the lane, and required manual operation to go to the next lane. That intervention followed a collision after the heading estimation diverged due to most of both sides being occluded by leaves in the middle of the path.



**Figure 8.** A major intervention cause is the existence of gaps, i.e., no crop row in any of the sides for more than 0.5 m. Plot depicts how heading changed 2 m before and after gap, which is represented by the gray dashed box. For lane 2(B) with PL+EKF+LGG, blue circles show the estimate from perception system (dark) and measured with IMU (light). Similarly, red markers depict the estimate and the magenta ones are the IMU readings for lane 2(B) with PL+LGG. The inclusion of EKF to the system has made a clear impact on heading estimation with more continuous and less erroneous (compared to IMU reading) estimates as we contrast the dark blue with red circles.

All not end-of-lane interventions (8 of 12 for B) happened while crossing or soon after the robot crossed the gap. This happened because of heading estimation error. In this case, as the robot approached a gap, there were fewer readings from the rows and they probably were not from the stems. These two factors combined destabilize the heading estimation near the gap. To illustrate this, Figure 8 shows heading estimates from the perception subsystem (blue and red dots) and yaw measurements from the IMU (light blue and magenta lines) for the same gap in both 1(B) experiments.

The gray dashed box indicates the gap position as the robot moved along the lane. It may be noted that estimates worsen as the robot approaches the gaps. Because of such a bad heading, even though rows may be visible after the gap, they are split into wrong sets and the robot may be following badly fitted lines. Indeed, the estimates greatly differed from expected inside the gap window. Notwithstanding, they recovered as the robot approached the rows (right side of the dashed box in Figure 8). Depending on how the robot is misaligned when it is close to leaving the rows before gap, it may run diagonally, and may hit one of the rows, enter the neighbor lane, or recover to the correct lane. This last option, when the robot is already going away, is usually accompanied by a strong control action that causes an oscillatory driving and the robot may crash soon after it enters the lane. Finally, Figure 8 also exposes the difference between the perception subsystem with (blue) and without EKF (red). The estimates with EKF do not disperse and resemble the IMU readings. Even though differences are bigger around gaps, they are not as drastic as without EKF. Moreover, they allowed the robot to cross all minor gaps. The only collision occurred in the 2-m gap, when the robot lost track of the lane it was going to and went away.

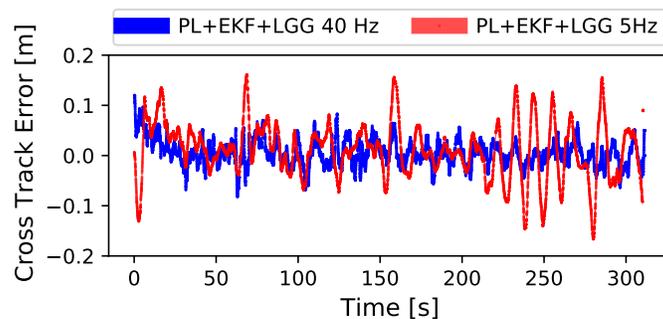
Table 2 shows the number of intervention runs with variable LiDAR update rate. Since the nominal update rate is 40 Hz, the results obtained in Table 2 for this frequency are similar to the ones shown in Table 1. The PL+EKF+LGG system had two interventions only in lane B when the robot tried to pass through the big gap (bigger than 2 m) while PL+LGG had problems in both lanes (two interventions for lane A and five for lane B). Also, it may be noted that the number of interventions does not change when EKF is used. In lane B, both 5 Hz and 10 Hz presented three more interventions, all collisions, than 40 Hz. Besides the gap problem, less updated measurements compromised stabilization after occlusions. This happens because the robot blindly drives forward while a scan without occlusion is yet to be processed by the perception subsystem. Indeed, since

**Table 2.** Number of interventions in runs that simulate different LiDAR frequencies. The tests occurred on the same A and B lanes.

	Run (Lane)					
	40 Hz		10 Hz		5 Hz	
	1(A)	2(B)	1(A)	2(B)	1(A)	2(B)
PL+EKF+LGG	1	2	1	2	1	2
PL+LGG	2	5	2	8	1	8

**Table 3.** Number of interventions in the sparse rows with frequent hanging leaves or fallen stalks (C and D) and cluttered with high grass and weeds (E and F) lanes in IAF.

	Run (Lane)							
	1(C)	2(D)	3(C)	4(D)	5(E)	6(F)	7(E)	8(F)
PL+EKF+LGG	2	1	1	2	1	3	1	2
PL+LGG	6	9	4	3	4	5	4	6

**Figure 9.** The cross-track error comparison between PL+EKF+LGG with 5-Hz (blue) and 40-Hz (red) LiDAR scan rate. Note that with 5 Hz, the system is more prone to larger swings (around 60 and 160 s), and also to oscillatory behavior (around 220 s). Nevertheless, when also using EKF coupled to PL, the number of interventions did not rise.

calculation of the control action requires updated estimates, while these are not available, the control remains the same. For such reason coupled to the fact that the robot is traveling at 0.7 m/s, the robot may overshoot the reference—the middle of the lane. This oscillating behavior is clear on Figure 9 between 200 and 300 s.

Twenty-eight interventions happened in the sparse rows (C and D in Table 3) with 8 end-of-lane, 2 operator stops, 1 bad start, and 1 hardware issue. From the remaining 16, all collisions because of a perception failure, the most significant error sources are heading estimation (8) and converging aspect of the perceived rows (3) because of hanging leaves and fallen stalks occluding the actual rows. Similar to the issue in the gaps (see Figure 8), the sparseness in the rows randomly diminished the available readings from the actual row, which raises the possibility of readings from fallen stalks and hanging leaves to be considered as an extension of the rows. While PL+LGG tests concentrated 14 of these perception-related collisions, PL+EKF+LGG had only 2 (both heading estimation failures). The cluttered lanes had 26 interventions (E and F in Table 3). They had 8 end-of-lane, 4 bad starts, and 1 hardware issue. The other 13 ended in collision due to a perception error, and again, the most significant source is heading estimation (8), now due to row occlusion because of the presence of high grass and weeds within the lane. Similarly, most of the perception-related collisions happened on the PL+LGG experiments (10 against 3 in PL+EKF+LGG).

As the final part of the controlled tests, three sets of experiments were carried out to show the effect of the EKF and LGG on the system when the robot is running on uneven (first 30 m of lane

**Table 4.** Number of interventions for runs in lanes G and H in IAF. Lane G has initial 30 m with artificial bumps. Both lanes had a significant part of their coverage cut to improve GNSS measurements. That also reduced the presence of fallen leaves, leading to mostly clear lanes as it may be seen in Figure 7f.

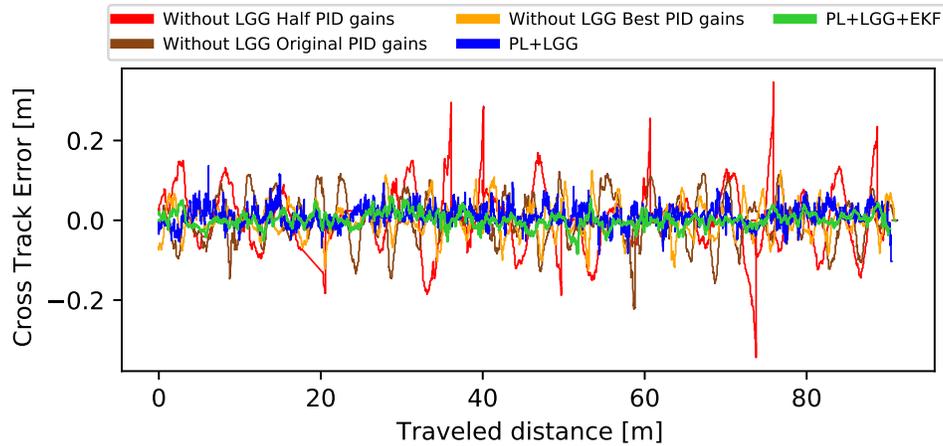
	Run (Lane)									
	1(G)	2(H)	3(G)	4(H)	5(G)	6(H)	7(G)	8(H)	9(G)	10(H)
PL+EKF+LGG	1	1	1	1	1	1	1	1	1	1
PL+LGG	1	1	1	2	2	1	1	1	1	1
PL+EKF ( $K_p = 0.7, K_d = 1.5$ )	1	3	2	2	4	3	2	5	2	2
PL+EKF ( $K_p = 0.35, K_d = 0.7$ )	9	8	8	2	31	-	-	-	-	-
PL+EKF ( $K_p = 1.7, K_d = 1.7$ )	2	2	1	1	5	1	1	1	1	1

G) and flat terrains (lane H). The first set of tests consisted of 10 runs using the proposed system PL+EKF+LGG. For the second one, we made another 10 runs using the system without EKF (PL+LGG). Finally, 22 runs were performed with the system without LGG (PL+EKF). Table 4 summarizes the results from those tests.

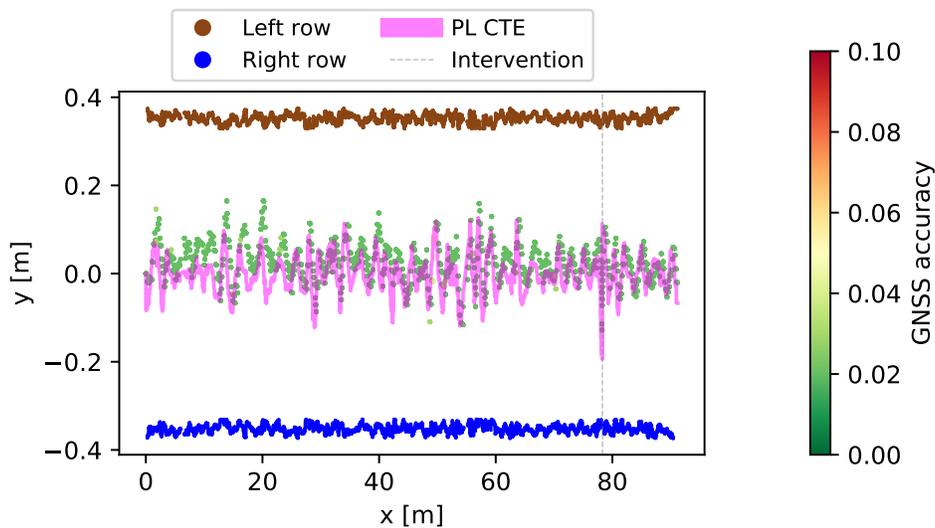
On lanes G and H, only one intervention per run occurred with the proposed system (PL+EKF+LGG). This intervention was related to the end row, where the user needs to regain control and manually drive the robot to the next lane. Apart from the end row interventions, the system without EKF (PL+LGG) got two additional interventions. Both interventions were related to sensor (LiDAR) occlusion. Based on those initial results, we may say that the system with and without EKF did not have any problem with the uneven terrains. Nevertheless, the system without EKF was more sensitive to the occlusion.

Unlike EKF, a module that produces the same outputs as PL, LGG suitably converts its input CTE into a target heading ( $\phi_r$ ), which is further used as reference of the controller's input. Therefore, when LGG is absent, there are two options without changing the subsequent controller block: (a) set the target heading to zero since this value is no longer calculated, and the controller turns into a simple null heading controller, or (b) use CTE, the other PL or PL+EKF output, as input for the PID controller. Option (a) is not suitable for the under-canopy navigation because only keeping the orientation does not guarantee that the robot will not collide into lateral rows. For such reason, option (b) is chosen to keep the controller when LGG is absent. As the magnitude of CTE is similar to the heading error (from  $-0.4$  to  $0.4$ ), the gains of the original PID ( $K_p = 0.7$  and  $K_d = 1.5$ ) were used as initial values. The obtained results of this configuration were summarized in the third row of Table 4. Subsequently, five runs (fourth row of Table 4) were executed with PID gains reduced to half of the original values ( $K_p = 0.35$  and  $K_d = 0.75$ ). It was not possible to test more times with these values because the frequent robot collisions caused some damage to the plants, which is highly undesirable. Finally, the best behavior of the system without LGG was obtained with  $K_p = 1.7$  and  $K_d = 1.7$ . Ten runs were made with such values and the results are shown in the fifth row of Table 4. The best run of each system (PL+EKF+LGG, PL+LGG, PL+EKF ( $K_p = 0.7, K_d = 1.5$ ), PL+EKF ( $K_p = 0.35, K_d = 0.75$ ), and PL+EKF ( $K_p = 1.7, K_d = 1.7$ )) was selected to plot the CTE behavior for each selected test in Figure 10.

According to the results shown in Table 4, the worst behavior of the system was obtained when LGG (PL+EKF) is not used and the PID gains were  $K_p = 0.35$  and  $K_d = 0.75$  (58 interventions in 5 runs). With this configuration, the robot heavily oscillated. Sometimes it could not return to the path center (red line in Figure 10). Additionally, it always crashed in the uneven part of lane G. Twenty-six interventions (10 of them were end row interventions and the other ones were related to sensor occlusion, uneven terrain, and the oscillatory behavior) happened with the system without LGG (PL+EKF) and the same values of PID gains as the original ones ( $K_p = 0.7, K_d = 1.5$ ). With this configuration, the robot kept the path center with a small oscillation (brown line in Figure 10). Still, an overshoot appeared when the robot ran close to one row, generating a constant oscillatory behavior or a crash. The best behavior of the tests made without LGG was obtained with the PID gains  $K_p = 1.7$  and  $K_d = 1.7$ . With this configuration the robot was able to keep the path

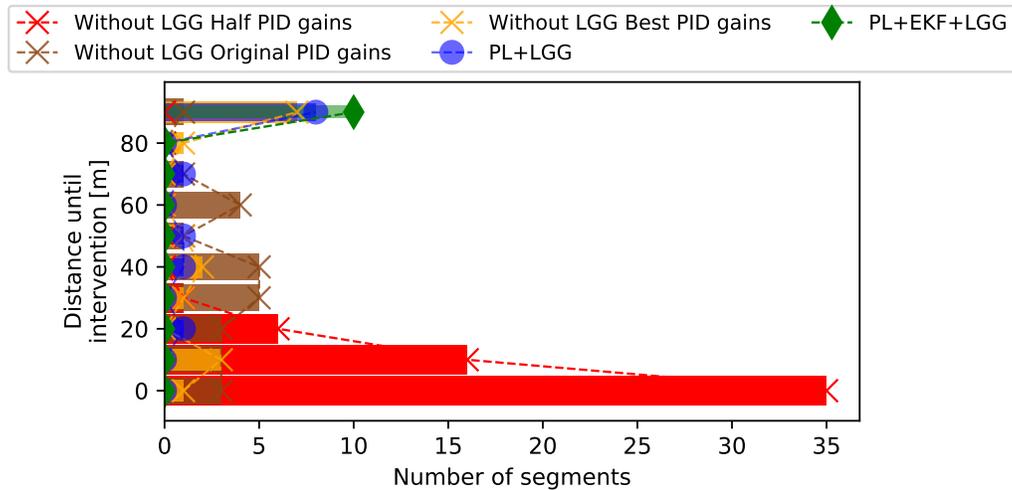


**Figure 10.** CTE for the best test of PL+EKF ( $K_p = 0.35$ ,  $K_d = 0.75$ ) (red line), PL+EKF ( $K_p = 0.7$ ,  $K_d = 1.5$ ) (brown line), PL+EKF ( $K_p = 1.7$ ,  $K_d = 1.7$ ) (orange line), PL+LGG (blue line), and PL+EKF+LGG (green line). An oscillatory behavior is seen on all experiments without LGG. The oscillations are minimized for  $K_p = 1.7$ ,  $K_d = 1.7$  gains. When using LGG in continuous straight lanes, such as G or H, performance is similar. However, it may be noted that estimates without EKF (blue) are noisy when compared to PL+EKF+LGG (green).



**Figure 11.** Representation of lane H when system ran PL+EKF with the best PID gains ( $K_p = 1.7$ ,  $K_d = 1.7$ ). Although controller led to an oscillatory behavior, PL correctly estimated the robot's position (pink line), which shows great agreement with RTK-GNSS positioning (green dots). The blue and brown lines represent the position of each row.

center with a small oscillation and the overshoot was deleted (orange line of Figure 10). Fifteen interventions were obtained with this configuration where 10 of them were end row interventions and the other 5 were related to the sensor occlusion. Although this configuration was the best one when LGG was not used, the proposed system (PL+EKF+LGG) had a better behavior (green line in Figure 10) with less number of interventions. Also, the system without EKF (PL+LGG) was better (see blue line in Figure 10) than PL+EKF ( $K_p = 1.7$ ,  $K_d = 1.7$ ). Yet, it had more noise than PL+EKF+LGG. Figure 11 shows a comparison between CTE (pink line) and GNSS+RTK (green line) for a run made with PL+EKF ( $K_p = 1.7$ ,  $K_d = 1.7$ ) where the information given by GNSS+RTK shows the same oscillatory behavior of CTE.



**Figure 12.** Since lanes G and H had significantly better GNSS measurements (98% below 0.05-m accuracy), they were used to compare the distance until intervention for different configurations of the navigation system. Without LGG and with poorly tuned PID (red and brown colors), most segments do not exceed 20 m before intervention. The system without LGG and best PID gains (yellow) has similar performance to PL+LGG, i.e., the system without EKF (blue). The full system PL+LGG+EKF (green) traversed all 10 runs without intervention.



**Figure 13.** Average condition in the production field corn crop (a) and an example of fallen stalks and hanging dry leaves (b) and its associated LiDAR scan (c). It may be seen that after 0.5 m, the right row is completely not visible. For the left side, the dry leaves form an additional cluster of readings from 0.5 to 2 m, which also covers the stalks.

Finally, Figure 12 shows the distances that the robot was able to run before each intervention presented in Table 4. The longest distances traveled by the robot happened when the proposed system (PL+EKF+LGG) was used (around 90 m) because it only has end row interventions. The shortest distances come from the tests using the system without LGG (PL+EKF) with the lowest PID gains ( $K_p = 0.35$  and  $K_d = 0.75$ ) because it had a big number of interventions. In this case the maximal distance run by the robot was 20 m. When the EKF (PL+LGG) was not used, the maximal distance was 90 m and the minimal one was 20 m.

### 5.2.2. Controlled Tests in the Production Field Farm with Curve

An additional six runs were performed in a production field corn crop containing a curved section with minimum curvature radius of 8 m (Figure 13a). The length of the lanes was around 300 m. Besides the curve, which assesses how well perception performs when rows are not parallel, this field has been left untouched for the season. Therefore, it possesses all challenges at once: angled stalks (lodging), a high number of dry leaves, fallen plants, and ground clutter. The angled stalks

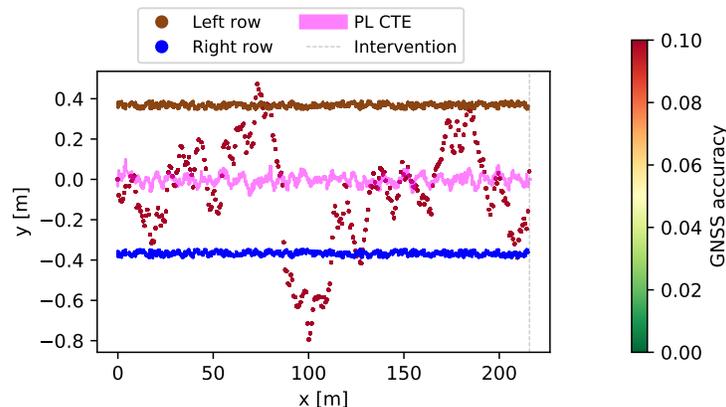
increased the possible lane width because the stalks grew outward. The dry leaves cause a variety of changes to the LiDAR scan. They may simply block the LiDAR field of view, form an inner “row”, form a block on one side that does not allow seeing through, or form a converging pattern, among other unseen cases. The fallen plants may be insurmountable obstacles, cover the row, or provide extra clutter. The ground clutter makes the robot unstable, bouncing in all directions.

Seventy-two interventions occurred in the six tests, from which 10 will not be considered in this analysis because 2 are hardware issues, 2 are operator manual stop, and 6 are end-of-lane interventions. For the remaining 62 interventions, 5 were bad starts and 46 happened with PL+LGG and 11 with PL+EKF+LGG. For PL+LGG, the most significant error sources were convergence of rows (23), classification of valid side (7), and heading estimation (7). The 11 collisions for PL+EKF+LGG occurred because of convergence of rows (4), classification of valid side (3), control not responding to distance error (2), challenging row identification in the LiDAR data (1), and wrong distance estimation (1).

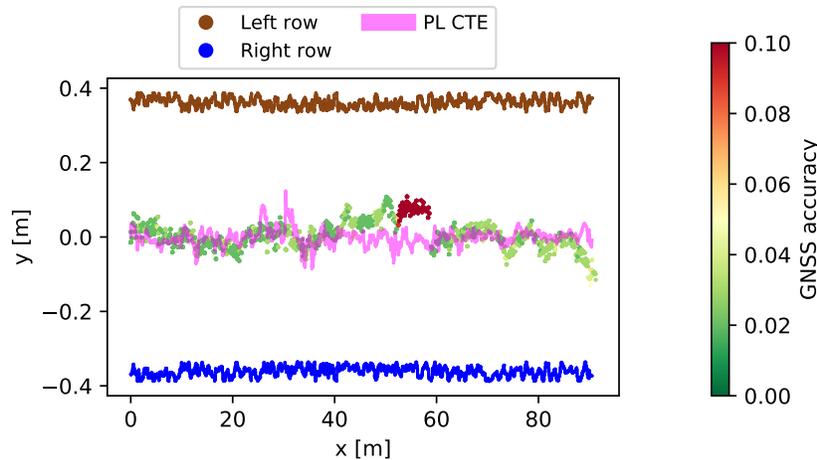
Figures 13b and 13c highlight the issue of having a single scanning plane. There is a horizontal leaf that blocks all readings on the right side behind it. Moreover, a converging pattern of the rows for  $y \in (0, 0.5)$  may be noted. Because our algorithm assumes that the angle of the best defined line is the heading error, when the scan was rotated to make the right row parallel to the  $y$  axis, the left row became highly tilted to the right. This compromised the point-choosing step, which relies on a histogram applied to the  $x$  axis to find the 0.05-m bin with the most readings. Since the right row is already parallel to the  $y$  axis, most of its readings were counted in the same bin. Still, the same does not happen with a diagonal row, such as the left one, whose readings were scattered on several bins. Subsequently, only the readings within the highest count bin and its neighbors are used. Therefore, the derived expression of the left row contains only a small section of it.

### 5.2.3. Remarks about Ground Truth

One of the most challenging tasks of the data analysis is to check the accuracy of the estimated distances due to the lack of a reliable source of ground truth for under-canopy navigation. GNSS measurements are the most popular data used as ground truth in robotics. Notwithstanding, they present inaccuracies in under-canopy scenarios. One example of the low accuracy of the GNSS in under-canopy scenarios is shown in Figure 14, where the brown and blue dots show, respectively, the left and right lateral rows while the magenta line depicts the CTE. The scattered points with the colormap represent the GNSS measurements. Although cross-track error and experimental notes



**Figure 14.** Representation of lane A, where straight and mostly parallel rows exist from beginning to end. CTE (pink) shows the robot’s estimated position when autonomously driving with PL+EKF+LGG. Although the robot kept itself in the middle of the lane for the whole trajectory, note that GNSS positioning (scattered dots) does not exhibit such a path. Indeed, GNSS accuracy had an average of 2.09 m with a minimum of 1.72 m. This is expected when canopy and multiple rows surround the robot. The vertical gray line represents the end-of-lane intervention.



**Figure 15.** CTE (pink line) and RTK-GNSS (scattered dots with colormap) of a run made with PL+EKF+LGG in lane G. The blue and brown lines represent the position of each row. Although GNSS accuracy is below 0.05 m for most of the time, accuracy went up to 0.62 m, between 52 and 59 m, before dropping back to the average 0.02 m. For such a period, GNSS positioning was compromised.

indicate that the robot stayed within rows and around the center of the lane, the same cannot be observed from the positioning given by GNSS. In this experiment, GNSS accuracy had an average of 2.09 m and a minimum of 1.72 m.

Such GNSS issues prevented a comparison between GNSS measurements and CTE for all tests. Nevertheless, this analysis was still possible for some runs in lane D (border rows), and lanes G and H, where their top part was cut to mitigate the canopy effect. Figures 11 and 16 show a comparison between CTE and GNSS measurements. They presented accuracy less than 0.05 m along the trajectory. It is possible to see that the difference between CTE and GNSS is lower than 0.1 m and their shapes are similar. Figure 15 shows the comparison between CTE and GNSS during a run made on lane G, where the accuracy was greater than 0.1 m in some part of the trajectory. It is possible to see as the GNSS information started to diverge from CTE when accuracy was higher than 0.1 m misleading information for the ground truth analysis.

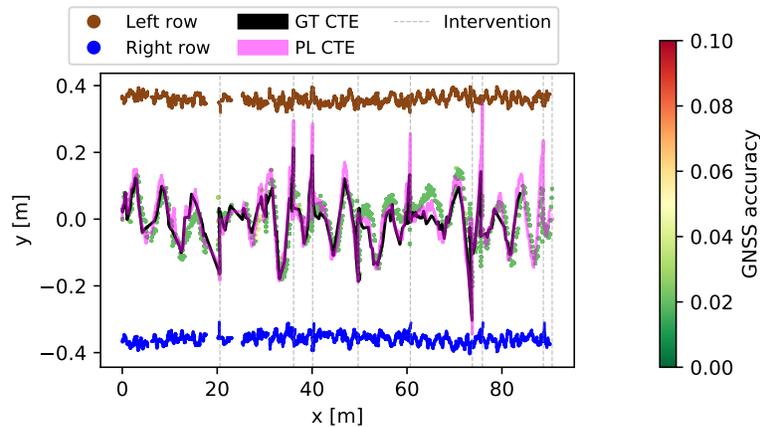
All the above shows that GNSS is limited for our field experiments. Hence, we also produced manual ground truth (GT CTE) from raw LiDAR readings. It was obtained for at least 200 scans for each scenario. Although using raw LiDAR readings is prone to the same challenges faced by perception, such as sensor occlusion, leaves, weeds, among others, a human labeler may take advantage of all four onboard cameras to clarify any situation. One example is to confirm the robot's orientation and leaf coverage, and from this information, choose which side would have best line fitting. Additionally, unlike PL, which tries to fit rows on all scans and then checks the validity, the human labeler may classify a scan as invalid and skip its associated row fitting process.

Figure 16 shows that ground truth with GNSS whose RTK accuracy is lower than 0.05 m (scattered points), manual ground truth with LiDAR (black line), and our perception output (pink line) have a great agreement. Indeed, the difference between GNSS and GT CTE has an average of 0.094 m with standard deviation of 0.082 m, the difference between GNSS and PL CTE has an average of 0.016 m with standard deviation of 0.062 m, and the difference between GT CTE and PL CTE has an average of 0.078 m with standard deviation of 0.097 m. Figure 17 shows the dispersion for the manual lane width. Although nominal row spacing is 0.75 m, actual lane width varies greatly, and we may expect values from 0.548 to 1.072 m. The most extreme outliers are represented by the black circles in the picture.

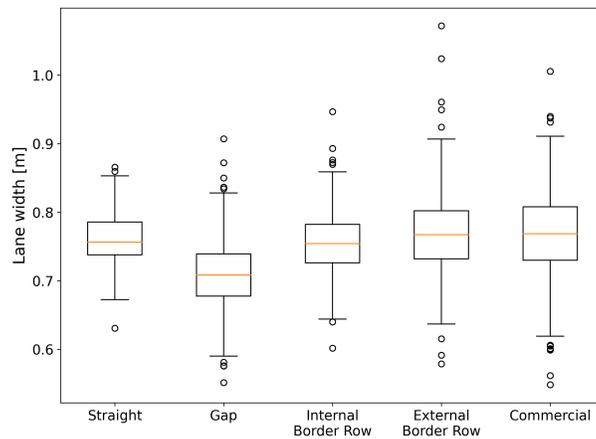
Table 5 shows more details. Both sides may not be always detectable on a scan (see percentage of valid sides) due to occlusion by leaves, gap, or sparseness. With respect to the difference between annotated lane width and expected value, the difference is lower in the continuous straight lane as

**Table 5.** Manual ground truth for the controlled scenarios.

	Valid left side [%]	Valid right side [%]	Avg. LW [m]	LW 0.05 m [%]	LW 0.1 m [%]
Straight	96	<b>99.500</b>	<b>0.762</b>	<b>83.246</b>	<b>97.906</b>
Gap	<b>96.011</b>	93.162	0.708	53.355	87.700
Internal Border Row	78.200	91.600	0.754	76.218	95.129
External Border Row	84.053	97.010	0.771	65.164	89.344
Production Field	94.340	81.761	0.766	54.959	83.884



**Figure 16.** Representation of lane G, where the top part of the rows was cut to improve RTK-GNSS. The robot used PL+EKF and  $K_p = 0.35$  and  $K_d = 0.7$  for PID gains. Note the agreement between GNSS readings (greenish scattered dots due to exceptional accuracy below 0.02 m), CTE from manual ground truth (black line), and CTE from system (pink line). This shows that PL+EKF correctly tracked robot position with respect to lateral rows in this oscillatory path intentionally induced by removing LGG and using different PID gains for control. Moreover, opposed to the experiment in the well behaved lane A in Figure 14, there were several interventions (vertical gray lines).



**Figure 17.** Dispersion of manual ground truth for the controlled scenarios. Although average value is close to nominal 0.75-m row spacing, note that annotated lane width may vary within a large range. The straight scenario has the smallest interquartile range (0.67–0.85 m) while commercial has the largest (0.62–0.91 m). This shows that perceived lane width may be significantly different from nominal value.

**Table 6.** Interventions in uncontrolled tests. Percentage with respect to total number of relevant interventions for that location.

	Bad start	Control	Gap	Perception	Total
ES #1	10 (7.2%)	7 (5.1%)	93 (67.4%)	28 (20.3%)	138
ES #2	10 (12.1%)	13 (15.7%)	29 (34.9%)	31 (37.3%)	83

expected (83.246% within 0.05 m). Such difference is higher in the other scenarios as percentages within 0.05 or 0.1 m get lower. This is caused due to the environment characteristics. Gaps are particularly problematic because they introduce several beginnings and endings of a lane. The beginning may be blocked by leaves, weeds, or fallen plants, and LiDAR may not clearly see ahead. There is a progressive drop in readings as the robot approaches the end. These two interfere in the process of finding a line representation for the respective side. Similarly, sparseness of plants (mostly in the internal border rows) makes the system more prone to consider an eventual cluster of leaves or weeds as part of the row since there are few plants that actually form the row. In the cluttered lanes, there was a great influence of ground unevenness combined with the large amount of clutter along the lane. Finally, in the production field crop, besides the mentioned issues, lane width consistency is compromised because of corn that grew angled, the curved part of the lane, and higher foliage.

### 5.3. Uncontrolled Tests

The system has also been tested by third parties, EarthSense Co. partners, in another two locations across the United States. The datasets have been provided by EarthSense Co. and they will be labeled as ES. The first is a corn research field whose 125–150-m lanes were further divided into 5-m sections with 0.5-m gaps between them. The system was used from 10 August 2020 to 21 September 2020. The operator’s goal was the collection of visual data, but in this case, for the whole lane. There were 373 trials, where 365 lanes were completed with a combination of manual and autonomous mode. The latter was used for 17235.844 m (35% of total distance). From this set, a subset of 51 experiments (6551 m) was chosen as they met two criteria: (1) total distance over 100 m and (2) more than 90% of the traversed distance in autonomous mode.

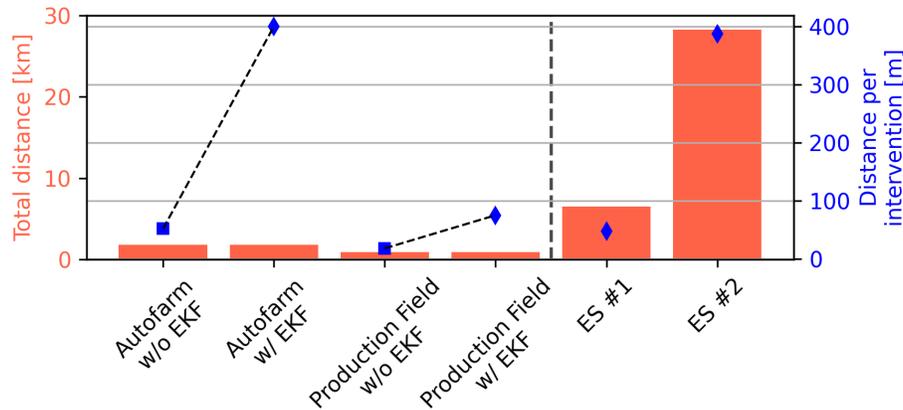
The second is another corn research field whose lanes comprised seventy-five 4.8-m sections separated by 0.5-m gaps, which results in 397-m lanes. The row spacing is the standard 0.8 m. The system was used between 31 July 2020 and 27 August 2020. Since the operator’s task was the visual data collection for either the whole lane or specific sections, expected operation distance varied from as low as 4.8 m to 397 m. The 159 runs autonomously covered 28245.670 m.

Due to different goals for the system’s usage, a significant number of the interventions are either end-of-lane (42 and 13, respectively) when it was expected to drive the whole lane or an operator stopped the robot (14 and 227, respectively) when a partial data collection was aimed. Table 6 summarizes the relevant interventions for this study: bad start, control, gap, and perception.

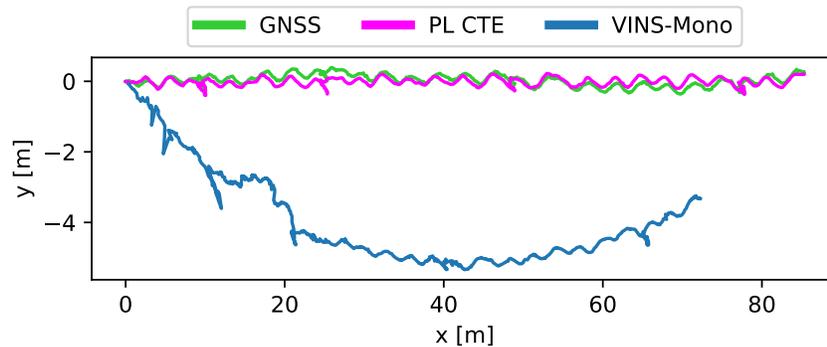
Finally, Figure 18 summarizes all experiments with a *distance per intervention* metric, which is defined as the total distance divided by the total number of interventions from the aforementioned four relevant sources. It clearly shows that the EKF use on IAF and production field improved the system’s performance. The uncontrolled tests used PL+EKF+LGG from the start with ES #2 displaying a similar performance (386.9 m/interv) to IAF with EKF (400 m/interv) over an extended distance (28 km). Unfortunately, ES #1 had a major influence of gaps (67.4% of considered interventions) which greatly reduced the distance per intervention to 47.5 m/interv (similar to production field with EKF, 56.1 m/interv).

### 5.4. Remarks about Calculation Time

Our algorithms ran on a Raspberry Pi 3, a low-cost (\$35 USD) embedded computer that enables mass deployment. Although LiDAR retrieves environment readings with 40 Hz update rate, each



**Figure 18.** Distance per intervention. Note how IAF went from 51.6 m/interv without EKF (PL) to 400 m/interv with EKF (PL+EKF). Production field also increased from 16.3 m/interv without EKF to 56.1 m/interv with EKF. Both ES tests were performed with EKF. ES #1 was heavily affected by gaps and achieved 47.5 m/interv while ES #2 had similar performance to IAF with 386.9 m/interv.



**Figure 19.** Performance of VINS-Mono (blue line), CTE of PL+EKF+LGG (pink line), and GPS, in under-canopy environments. It may be noted that VINS-Mono failed to track the robot's position within lane.

scan has only 1081 elements, from which is also expected a considerable reduction (around 60%) in number of points that would be used to extract the rows (Higuti et al., 2019). Hence, there are much fewer data to process than a camera frame, which in low resolution would be in the order of  $320 \times 240$  pixels (177:1 ratio when compared to reduced LiDAR set). Coupled to assumptions and constraints that simplify the problem, the core perception subsystem (PL) runs under 5 ms (Higuti et al., 2019). In this study's experiments, PL keeps its running time at 4.83 ms. Therefore, it is similar to previous work. When used, EKF adds at most 2.23 ms while LGG adds another 0.77 ms. The total time for all three methods is still considerably far from the 25 ms threshold. Since it is less than the time between LiDAR scans, further optimization was not needed.

### 5.5. PL+EKF+LGG vs VINS-Fusion with Monocular Camera

Using collected data from a manual run made in lane G, we ran the VINS-Mono algorithm in order to check if it is able to work in under-canopy environments. The result is shown in Figure 19, where it is possible to see that the trajectory given by the GNSS (red line) and the proposed (PL+EKF+LGG) system (pink line) are similar. However, the trajectory given by VINS-Mono (blue line) is far from them. Thus, it is possible to say that VINS-Mono has a bad performance to be used for under-canopy navigation. Maybe the VINS-Fusion for stereo cameras could get better results. Nevertheless, we could not test it because our robot does not have stereo cameras.

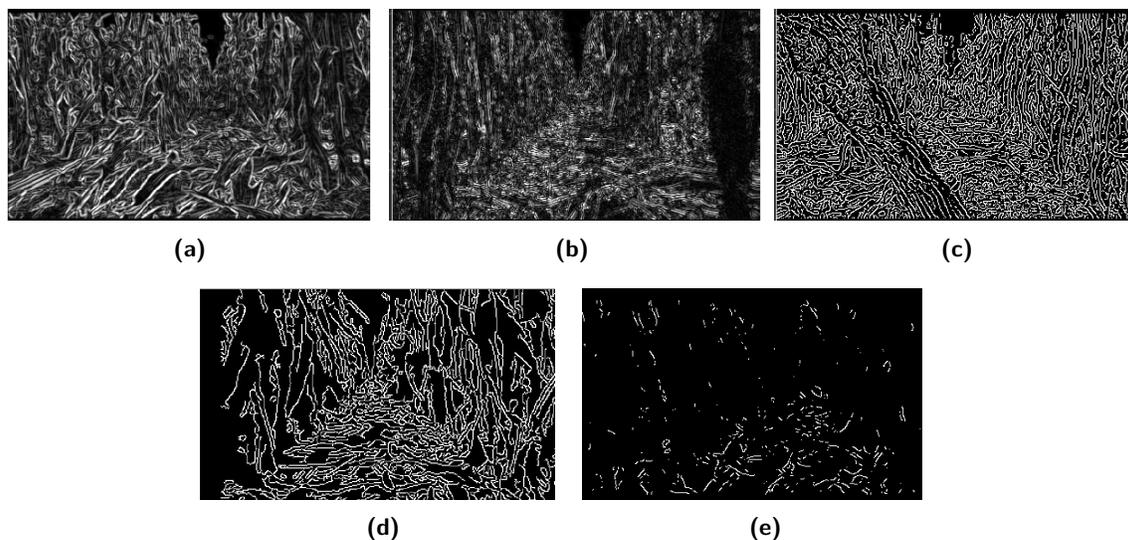
## 6. Discussion

Through controlled tests, it was possible to verify the behavior of the proposed system (PL+EKF+LGG) in a variety of scenarios not tested in previous studies. Although the system exhibited reliable performance in scenarios with long continuous rows (lane A and H) and uneven terrains (lane G), interventions are still expected in edge cases such as the presence of gaps (lane B), fallen leaves/stalks or sparseness (lane C and D), and grass and weeds (lane E and F). While gaps present a situation with discontinuity of rows in both sides, the others cause total/partial occlusion (see Figures 4b, 4c, and 4d).

A total/partial occlusion of LiDAR leads to a bad heading estimation, which generates a wrong distance estimation. To find the lateral rows, the perception subsystem assumes that the  $Y_P$  axis (see Figure 3) divides readings into two regions: one to the left and another to the right. Subsequently, the lateral distance estimate is computed from the respective side set. When the heading is wrong, e.g., a  $\phi = 0$  in Figure 3 and  $Y_P$  collapses to  $Y_R$ , we may expect that the top right readings become part of the left set. With this, the left side estimate will point towards the right, and the robot will drive away from the left side instead of going towards it, to return the robot to the lane center. We tried to use other methods to fit the lines such as RANSAC (Fischler and Bolles, 1981) in our previous work (Higuti et al., 2019). However, there was a considerable increment in the computational cost and loop time. The obtained result was similar to the least-squares approach. For that reason, we did not consider it in the proposed system. Since the root cause of this issue is the LiDAR's inability to see ahead using only a single plane of view, the robot requires a second source of information to determine where it is safe to go. In such context, visual information would enable perception to get more semantic information from the environment and distinguish situations, for example, between blocking leaves on the sensor field of view to a full-grown crop row, which should be treated as untraversable. However, vision algorithms require a higher computational cost, which would be unfeasible to run on a Raspberry Pi 3 in real time. A change in the main robot's processing unit would be necessary to run a camera + artificial intelligence (AI) algorithm, while it might offer better results in terms of autonomous navigation.

Indeed, the use of cameras in under-canopy navigation has been shown as complex and computationally costly. Researchers have used the Hough transform (Winterhalter et al., 2018) to fit the lines from camera input or VINS-Fusion to estimate the robot pose from vision and IMU. The Hough transform is a good algorithm to be used for above-canopy navigation where the camera gets the top view of the rows. To use this algorithm, a corner or edge detection algorithm to create a binary image of features should be used because it is not able to work with raw images. This conversion has some problems in under-canopy views, because some undesired corners or edges coming from the leaves or fallen plants are detected. Those undesired corners or edges will generate a wrong estimation of the lines for the Hough algorithm (analogous to the occlusion problem with the LiDAR). Figure 20 shows some results obtained from three different edge detector methods: the Sobel derivative (Figure 20a), Laplace operator (Figure 20b), and Canny (Figures 20c, 20d, and 20e) where rows are indistinguishable. Then, it is not possible to use the Hough transform to fit the lines using images from under-canopy environments because the current edge detector methods are not able to generate good binary images with relevant information to extract the lines.

During the gaps, field conditions vastly vary, and most times the system does not have any reference to follow. Besides the lack of defined rows on the gap itself, the presence of weeds or hanging leaves on the lane entrance blocks the LiDAR scan and it falls to the same issue of sensor occlusion. Currently, it uses the last valid values of distances and heading to continue its navigation until the next part of the lane, going through what was blocking the sensor view and, subsequently, starts using our proposed perception subsystem. The success of this approach depends on the heading that the robot had at the beginning of the gap. If the heading is close to zero, then the robot is able to pass through the gap without any problem. Otherwise, the probability of missing the trajectory will be greater if the heading is big. One possible solution to fix this problem should be the combination of the proposed system with a navigation based on RTK-GNSS. When the robot is in the gap, the



**Figure 20.** Edge detector methods: (a) Sobel derivative; (b) Laplacian operator; (c) Canny edge detector with Canny threshold1 = 0 and Canny threshold2 = 0; (d) Canny edge detector with Canny threshold1 = 0 and Canny threshold2 = 250; and (e) Canny edge detector with Canny threshold1 = 250 and Canny threshold2 = 250.

GNSS accuracy could be enough to keep the trajectory until the other part of the row is reached. Additionally, the GNSS-based system could drive the robot at the end of the row (when the robot is out of row and the proposed system does not have any reference to follow) in order to find the next row to continue the navigation. This approach could address the end-of-row interventions.

The production field experiments showed that there is still a huge gap to overcome and enable our perception subsystem in real-world crops, and not only research ones. In particular, a convergence of rows heavily affects the lateral distance estimation. Such convergence, either originated from the curve or perceived due to leaves, fallen stalks, or weeds occluding the actual rows, breaks the PL assumption that rows are mostly parallel for the next meters (Higuti et al., 2019). Since PL relies on both pre-rotation of LiDAR data to attempt having at least one of the rows parallel to the robot's longitudinal axis and subsequent application of a histogram to detect the rows, the lack of parallelism significantly induces errors in estimation. This may be addressed by adding a classification step to remove unwanted readings from objects other than the stalks and also reformulating PL to consider curved/converging rows. Such reformulation would require generalizing the line fitting to a curve fitting, and therefore all subsequent validation metrics and estimate calculations would need to change accordingly. Instead of using a single value of cross-track error for control, another approach would be to change the controller to a waypoint-based one, where waypoints may be generated from the extracted curves and they would mark the path that the robot should follow for the next few meters.

## 6.1. Future Works

Although the performance of the proposed system (PL+EKF+LGG) was significantly more efficient than our previous work (Higuti et al., 2019) with a lower number of interventions, some challenges must be addressed in a future version. Some of them are (1) gaps, (2) lane change, (3) partial or total occlusion of the LiDAR view, and (4) presence of untraversable objects as fallen stalks. For that reason our list of future works to improve the results of the proposed system is as follows:

- Reduction of the end-of-row interventions by a hybrid system between GNSS and the proposed system. This new system might help to improve the problem with the gaps.
- Addition of a classification step to check whether a scan is usable and also remove unwanted readings from objects other than the stalks.

- Generalization of line to curve fitting. This would address curves and row convergence.
- Change of controller to accommodate a set of waypoints for the next few meters instead of using only the current cross-track error. The waypoints would represent the center of the lane.
- Creation of a system to detect when the robot collides, which enables it to stop its locomotion and, subsequently, reduces the damage to the plants. Additionally, a recovery maneuver could be used to return the robot to the trajectory after a collision.
- Application and possibly fusion with vision-based perception systems for under-canopy navigation to reduce the interventions due to the sensor occlusion.
- Evaluation of maps and AI techniques to improve the performance of the robot navigation in under-canopy environments.

## 7. Conclusions

This work reports 50.88 km field testing of an autonomous solution for under-canopy navigation in corn crops. On its controlled tests, it showed that distance per intervention is significantly improved when the core perception subsystem (PL) is enhanced by the proposed EKF. Such performance is corroborated by the field results from uncontrolled tests. The intervention analysis has shown that three major problems remain. Heading estimation and convergence of rows are present for lanes with frequent objects such as hanging leaves, fallen stalks, and high grass and weeds, all capable of occluding the actual rows to be followed. The convergence problem also indicates that curves requires further attention. Finally, the most influential source of intervention was the presence of gaps, which greatly affected ES #1 performance. All of them break the assumptions once devised to constrain the challenge of navigating in a corn crop using a LiDAR sensor to make it feasible in the real world. While such questions remain open for a robust autonomous navigation, the presented PL+EKF already relieves the robot operator of the tedious task of manually driving the robot to collect field data.

## Acknowledgments

The work was partially supported by Sao Paulo Research Foundation (FAPESP) grant number 2018/10894-2. The authors thank EarthSense support with TerraSentia robots and field data. The authors also thank Jacob Washburn, plant genetics researcher at the University of Missouri, for performing part of the uncontrolled tests. We also would like to thank DASLAB members (Sri Vuppala Theja, Nolan Replogle, Billy Doherty, and Rajan Aggarwal) for the technical support in the deployment of the proposed system.

## ORCID

Andres Eduardo Baquero Velasquez  <https://orcid.org/0000-0002-9762-2214>

Vitor Akihiro Hisano Higuti  <https://orcid.org/0000-0002-7008-4081>

Mateus Valverde Gasparino  <https://orcid.org/0000-0001-7900-8061>

Marcelo Becker  <https://orcid.org/0000-0002-7508-5817>

Girish Chowdhary  <https://orcid.org/0000-0002-4657-307X>

## References

- Abdu, M. A., Batista, I. S., Carrasco, A. J., and Brum, C. G. (2005). South Atlantic magnetic anomaly ionization: A review and a new focus on electrodynamic effects in the equatorial ionosphere. *Journal of Atmospheric and Solar-Terrestrial Physics*, 67(17–18 Special Issue):1643–1657.
- Abidine, A. Z., Heidman, B. C., Upadhyaya, S. K., and Hills, D. J. (2004). Autoguidance system operated at high speed causes almost no tomato damage. *California Agriculture*, 58(1):44–47.
- Bak, T., and Jakobsen, H. (2004). Agricultural robotic platform with four wheel steering for weed detection. *Biosystems Engineering*, 87(2):125–136.

- Bakker, T., van Asselt, K., Bontsema, J., Müller, J., and van Straten, G. (2006). An autonomous weeding robot for organic farming. In Corke, P. and Sukkariah, S., editors, *Field and Service Robotics*, pages 579–590, Berlin, Heidelberg. Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-540-33453-8\\_48](https://doi.org/10.1007/978-3-540-33453-8_48).
- Bakker, T., van Asselt, K., Bontsema, J., Müller, J., and van Straten, G. (2011). Autonomous navigation using a robot platform in a sugar beet field. *Biosystems Engineering*, 109(4):357–368.
- Baldwin, I., and Newman, P. (2012). Road vehicle localization with 2D push-broom LIDAR and 3D priors. In *2012 IEEE International Conference on Robotics and Automation*, pages 2611–2617. IEEE.
- Ball, D., Upcroft, B., Wyeth, G., Corke, P., English, A., Ross, P., Patten, T., Fitch, R., Sukkariah, S., and Bate, A. (2016). Vision-based obstacle detection and navigation for an agricultural robot. *Journal of Field Robotics*, 33(8):1107–1130.
- Barawid, O. C., Mizushima, A., Ishii, K., and Noguchi, N. (2007). Development of an autonomous navigation system using a two-dimensional laser scanner in an orchard application. *Biosystems Engineering*, 96(2):139–149.
- Bell, J., MacDonald, B. A., and Ahn, H. S. (2016). Row following in pergola structured orchards. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 640–645, Daejeon, South Korea. IEEE.
- Bell, T. (2000). Automatic tractor guidance using carrier-phase differential GPS. *Computers and Electronics in Agriculture*, 25(1–2):53–66.
- Bergerman, M., Maeta, S. M., Zhang, J., Freitas, G. M., Hamner, B., Singh, S., and Kantor, G. (2015). Robot farmers: Autonomous orchard vehicles help tree fruit production. *Robotics & Automation Magazine*, 22(March):54–63.
- Blackmore, S., Griepentrog, H. W., Nielsen, H., Norremark, M., and Resting-Jeppesen, J. (2004). Development of a deterministic autonomous tractor. *CIGR International Conference*, (May 2014):6.
- Cavender-Bares, K. (2021). Robotic platform and method for performing multiple functions in agricultural systems. <https://www.google.com/patents/US20210132608>.
- Chen, G. (2018). *Advances in Agricultural Machinery and Technologies*, chapter 1. CRC Press, Boca Raton, FL.
- Choudhuri, A., and Chowdhary, G. (2018). Crop stem width estimation in highly cluttered field environment. *Proceedings of the Computer Vision Problems in Plant Phenotyping (CVPPP 2018)*, Newcastle, UK, pages 6–13.
- English, A., Ross, P., Ball, D., and Corke, P. (2014). Vision based guidance for robot navigation in agriculture. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1693–1698, Hong Kong. IEEE.
- Fischler, M. A., and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.
- Fountas, S., Mylonas, N., Malounas, I., Rodias, E., Hellmann Santos, C., and Pekkeriet, E. (2020). Agricultural robotics for field operations. *Sensors*, 20(9):2672.
- Due, K. G., Porter, W. M., Barnes, E. M., and Rains, G. C. (2020). An extensive review of mobile agricultural robotics for field operations: Focus on cotton harvesting. *AgriEngineering*, 2(1):150–174.
- Higuti, V. A. H., Velasquez, A. E. B., Magalhaes, D. V., Becker, M., and Chowdhary, G. (2019). Under canopy light detection and ranging-based autonomous navigation. *Journal of Field Robotics*, 36(3):547–567.
- Hiremath, S., van Evert, F. K., ter Braak, C., Stein, A., and van der Heijden, G. (2014a). Image-based particle filtering for navigation in a semi-structured agricultural environment. *Biosystems Engineering*, 121:85–95.
- Hiremath, S. A., van der Heijden, G. W. A. M., van Evert, F. K., Stein, A., and ter Braak, C. J. F. (2014b). Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter. *Computers and Electronics in Agriculture*, 100:41–50.
- Jiang, G., Wang, Z., and Liu, H. (2015). Automatic detection of crop rows based on multi-ROIs. *Expert Systems with Applications*, 42(5):2429–2441.
- Kayacan, E., Zhang, Z., and Chowdhary, G. (2018). Embedded high precision control and corn stand counting algorithms for an ultra-compact 3D printed field robot. In *Proceedings of Robotics: Science and Systems*, pages 1–9, Pittsburgh, PA.
- King, A. (2017). The future of agriculture. *Nature*, 544(7651):S21–S23.

- Lemos, R. A. d., Nogueira, L. A. C. d. O., Ribeiro, A. M., Mirisola, L. G. B., Mauro F. Koyama, Paiva, E. C. d., and Bueno, S. S. (2018). Unisensory intra-row navigation strategy for orchards environments based on sensor laser. In *Congresso Brasileiro de Automática*, volume 1.
- Lenain, R., Thuilot, B., Cariou, C., and Martinet, P. (2003). Rejection of sliding effects in car like robot control: Application to farm vehicle guidance using a single RTK GPS sensor. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, volume 3, pages 3811–3816. IEEE.
- McAllister, W., Osipych, D., Davis, A., and Chowdhary, G. (2019). Agbots: Weeding a field with a team of autonomous robots. *Computers and Electronics in Agriculture*, 163:1–14.
- McAllister, W., Whitman, J., Varghese, J., Axelrod, A., Davis, A., and Chowdhary, G. (2020). Agbots2.0: Weeding denser fields with fewer robots. In *Robotics: Science and Systems*, volume 5, pages 1–9, Corvallis, OR. Robotics: Science and Systems Foundation.
- Montalvo, M., Pajares, G., Guerrero, J., Romeo, J., Guijarro, M., Ribeiro, A., Ruz, J., and Cruz, J. (2012). Automatic detection of crop rows in maize fields with high weeds pressure. *Expert Systems with Applications*, 39(15):11889–11897.
- Mueller-Sim, T., Jenkins, M., Abel, J., and Kantor, G. (2017). The Robotanist: A ground-based agricultural robot for high-throughput crop phenotyping. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3634–3639. IEEE.
- Neves, A. (2017). *Service Robots*, chapter 4. InTech, Rijeka.
- Nørremark, M., Griepentrog, H. W., Nielsen, J., and Sogaard, H. T. (2008). The development and assessment of the accuracy of an autonomous GPS-based system for intra-row mechanical weed control in row crops. *Biosystems Engineering*, 101(4):396–410.
- Oliveira, L. F. P., Moreira, A. P., and Silva, M. F. (2021). Advances in forest robotics: A state-of-the-art survey. *Robotics*, 10(2):1–20.
- Pilarski, T., Happold, M., Pangels, H., Ollis, M., Fitzpatrick, K., and Stentz, A. (2002). The Demeter system for automated harvesting. *Autonomous Robots*, 13(1):9–20.
- Pouliot, N., Richard, P.-L., and Montambault, S. (2012). LineScout power line robot: Characterization of a UTM-30LX LIDAR system for obstacle detection. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4327–4334. IEEE.
- Qin, T., Li, P., and Shen, S. (2018). VINS-Mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020.
- Radcliffe, J., Cox, J., and Bulanon, D. M. (2018). Machine vision for orchard navigation. *Computers in Industry*, 98:165–171.
- Reid, J., and Searcy, S. (1987). Vision-based guidance of an agriculture tractor. *IEEE Control Systems Magazine*, 7(2):39–43.
- Reid, J. F., Zhang, Q., Noguchi, N., and Dickson, M. (2000). Agricultural automatic guidance research in North America. *Computers and Electronics in Agriculture*, 25(1–2):155–167.
- Reina, G., Milella, A., Rouveure, R., Nielsen, M., Worst, R., and Blas, M. R. (2016). Ambient awareness for agricultural robotic vehicles. *Biosystems Engineering*, 146:114–132.
- Reiser, D., Sehsah, E., Bumann, O., Morhard, J., and Griepentrog, H. W. (2019). Development of an autonomous electric robot implement for intra-row weeding in vineyards. *Agriculture*, 9(18):1–12.
- Rovira-Más, F., Chatterjee, I., and Sáiz-Rubio, V. (2015). The role of GNSS in the navigation strategies of cost-effective agricultural robots. *Computers and Electronics in Agriculture*, 112:172–183.
- Santos, F. B. N. d., Sobreira, H. M. P., Campos, D. F. B., Santos, R. M. P. M. d., Moreira, A. P. G. M., and Contente, O. M. S. (2015). Towards a reliable monitoring robot for mountain vineyards. In *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*, pages 37–43, Vila Real. IEEE.
- Sensortec, B. (2020). BNO055 intelligent 9-axis absolute orientation sensor. Bosch Sensortec, Baden-Württemberg, Germany, pages 1–106.
- Shafiekhani, A., Kadam, S., Fritschi, F. B., and DeSouza, G. N. (2017). Vinobot and Vinoculer: Two robotic platforms for high-throughput field phenotyping. *Sensors*, 17(1).
- Shamshiri, R. R., Weltzien, C., Hameed, I. A., Yule, I. J., Grift, T. E., Balasundram, S. K., Pitonakova, L., Ahmad, D., and Chowdhary, G. (2018). Research and development in agricultural robotics: A perspective of digital farming. *International Journal of Agricultural and Biological Engineering*, 11(4):1–14.
- Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D. (2011). *Introduction to Autonomous Mobile Robots*. MIT Press.

- Sparrow, R., and Howard, M. (2020). Robots in agriculture: Prospects, impacts, ethics, and policy. *Precision Agriculture*, pages 1–16.
- Spogli, L., Alfonsi, L., Romano, V., De Franceschi, G., Joao Francisco, G. M., Hirokazu Shimabukuro, M., Bougard, B., and Aquino, M. (2013). Assessing the GNSS scintillation climate over Brazil under increasing solar activity. *Journal of Atmospheric and Solar-Terrestrial Physics*, 105–106:199–206.
- Stoll, A., and Kutzbach, H. D. (2001). Guidance of a forage harvester with GPS. *Precision Agriculture*, 2(3):281–291.
- Subramanian, V., Burks, T. F., and Arroyo, A. A. (2006). Development of machine vision and laser radar based autonomous vehicle guidance systems for citrus grove navigation. *Computers and Electronics in Agriculture*, 53(2):130–143.
- Thrun, S. (2002). Probabilistic robotics. *Communications of the ACM*, 45(3):52–57.
- Thuilot, B., Cariou, C., Martinet, P., and Berducat, M. (2002). Automatic guidance of a farm tractor relying on a single CP-DGPS. *Autonomous Robots*, 13(1):53–71.
- Troyer, T. A., Pitla, S., and Nutter, E. (2016). Inter-row robot navigation using 1D ranging sensors. *IFAC-PapersOnLine*, 49(16):463–468.
- Wang, H., and Noguchi, N. (2016). Autonomous maneuvers of a robotic tractor for farming. In *2016 IEEE/SICE International Symposium on System Integration (SII)*, pages 592–597, Sapporo. IEEE.
- Winterhalter, W., Fleckenstein, F. V., Dornhege, C., and Burgard, W. (2018). Crop row detection on tiny plants with the pattern hough transform. *IEEE Robotics and Automation Letters*, 3(4):3394–3401.
- Xue, J., Zhang, L., and Grift, T. E. (2012). Variable field-of-view machine vision based row guidance of an agricultural robot. *Computers and Electronics in Agriculture*, 84:85–91.
- Zhai, Z., Zhu, Z., Du, Y., Song, Z., and Mao, E. (2016). Multi-crop-row detection algorithm based on binocular vision. *Biosystems Engineering*, 150:89–103.
- Zhang, C., Noguchi, N., and Yang, L. (2016). Leader–follower system using two robot tractors to improve work efficiency. *Computers and Electronics in Agriculture*, 121:269–281.
- Zhang, J., Chambers, A., Maeta, S., Bergerman, M., and Singh, S. (2013). 3D perception for accurate row following: Methodology and results. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5306–5313, Tokyo. IEEE.
- Zhang, S., Wang, Y., Zhu, Z., Li, Z., Du, Y., and Mao, E. (2018). Tractor path tracking control based on binocular vision. *Information Processing in Agriculture*, 5(4):422–432.
- Zhao, S., and Zhang, Z. (2016). A new recognition of crop row based on its structural parameter model. *IFAC-PapersOnLine*, 49(16):431–438.

**How to cite this article:** Velasquez, A. E. B., Higuti, V. A. H., Gasparino, M. V., Sivakumar, A. N. V., Becker, M., & Chowdhary, G. (2022). Multi-sensor fusion based robust row following for compact agricultural robots. *Field Robotics*, 2, 1291–1319.

**Publisher’s Note:** Field Robotics does not accept any legal responsibility for errors, omissions or claims and does not provide any warranty, express or implied, with respect to information published in this article.