Adaptive Ultrasound-Based Tractor Localization for Semi-Autonomous Vineyard Operations

Matteo Corno [1,*,†] , **Sara Furioli** [1,†] , **Paolo Cesana** [2] **and Sergio M. Savaresi** [1]

1   Dipartimento di Elettronica Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy; sara.furioli@polimi.it (S.F.); sergio.savaresi@polimi.it (S.M.S.)
2   SDF Group, 24047 Treviglio, Italy; paolo.cesana@sdfgroup.com
*   Correspondence: matteo.corno@polimi.it
†   These authors contributed equally to this work.

Abstract: Autonomous driving is greatly impacting intensive and precise agriculture. Matter-of-factly, the first commercial applications of autonomous driving were in autonomous navigation of agricultural tractors in open fields. As the technology improves, the possibility of using autonomous or semi-autonomous tractors in orchards and vineyards is becoming commercially profitable. These scenarios offer more challenges as the vehicle needs to position itself with respect to a more cluttered environment. This paper presents an adaptive localization system for (semi-) autonomous navigation of agricultural tractors in vineyards that is based on ultrasonic automotive sensors. The system estimates the distance from the left vineyard row and the incidence angle. The paper shows that a single tuning of the localization algorithm does not provide robust performance in all vegetation scenarios. We solve this issue by implementing an Extended Kalman Filter (EKF) and by introducing an adaptive data selection stage that automatically adapts to the vegetation conditions and discards invalid measurements. An extensive experimental campaign validates the main features of the localization algorithm. In particular, we show that the Root Mean Square Error (RMSE) of the distance is 16 cm, while the angular RMSE is 2.6 degrees.

Keywords: crop row guidance; localization; situation awareness

1. Introduction

Farmers have been using some forms of automatic driving technologies for years. Vehicle automatization has an appealing potential in agriculture, mainly due to the relative simple regulations, seasonally long working hours, and the repetitive nature of many agricultural tasks. Matter-of-factly, semi-autonomous tractors have been common for decades [1]. Agricultural vehicle automation can reduce costs, by operating longer hours without the need of employing personnel, and, most importantly, is an enabling technology for precision farming, a farm management approach that uses real time information on the state of the crops, and responds—as automatically as possible—to varying crops conditions [2,3].

So far, the most successful autonomous driving systems in agriculture have been those used in open fields. These tasks require the vehicle to track a pre-computed path and are, thus, easy to automate. Open fields are easily mapped using standard land surveying techniques. The scenario becomes more challenging if one considers, for example, orchards and vineyards. These high value cultivars are often planted very densely, in non-flat terrains and grow irregularly. A map based navigation of orchards and vineyards is not the preferred option as the maps would need to be constantly updated.

Autonomous driving systems execute three different tasks: localization, planning and tracking. While tracking (i.e., actuating the steering to minimize the error between the planned path and the current position) is relatively easy in low speed applications, planning and localization can be challenging in some environments.

This work focuses on high value vineyards and proposes a new localization system. In developing the system, one has to consider cost-effectiveness and robustness to the different conditions the vineyard may be in. We guarantee cost-effectiveness by using standard, automotive ultrasonic proximity sensors, whereas robustness is built in the design of the algorithm.

Localization is the most sensor intensive task of an autonomous driving system. There exist three main approaches to navigation:

- Global Navigation Satellite System (GNSS) Localization [4–8]. Information from positioning satellites can yield accurate positioning (sub centimeter accuracy) in open fields. However, this solution alone is not adequate for vineyard navigation. It suffers from two drawbacks: (1) surrounding vegetation may cause loss of accuracy in the reception of the satellite signal [9,10], and (2) it can be costly to keep up-to-date maps of vineyards.
- Vision-based Localization uses cameras to position the vehicle in the field [11,12]. These methods are very cost-effective, but suffer from dependence on lighting conditions and may not be robust in all scenarios.
- Distance-based Localization. In these methods, distance sensors return the distance of the tractor from various obstacles, and the tractor is localized with respect to these features.

In the third category, the actual characteristics of the available sensors determine the properties of the algorithm. For example, Reference [13] employs a 2D laser scanner to navigate an orchard. The system identifies the trees and, through the Hough transform [14], fits a line. The main disadvantage of this approach is that it is static in nature, and the localization fails whenever the scanner does not detect enough trees for the Hough transform to provide a fit. LiDARs are, in general, very accurate. They can (especially 3D LiDARs) not only detect plants but can also monitor their vegetative status, as in Reference [15]. Three-dimensional LiDARs generate very dense point-cloud; the best approach to employ all this information is through particle filters, as in Reference [16]. The latter work develops an autonomous navigation system for a robot in a maize field. The authors of [17] propose a vineyard autonomous navigation system based on fusing the information of a 3D LiDAR, an Inertial Measurement Unit (IMU) and GPS. The algorithm creates a map of the environment which is then used also by the path planner. The paper concludes that vineyards are a challenging environment because of variability, namely of the vegetation, weather, and soil.

Most solutions available in the literature are based on LiDARs. LiDARs, while being unsurpassed from the accuracy standpoint, have not yet reached the technological maturity where they can be cost-effectively sourced. For this reason, we explore the possibility of using acoustic proximity sensors as the main information source for localization. Ultrasonic proximity sensors are standard in automotive applications and their price is in the single digit range, whereas LiDARs cost around hundreds of dollars. The lower cost, of course, entails lower accuracy and resolution.

We focus on the localization algorithm for an Advance Driver Assistance System (ADAS) of level 3. In particular, the ADAS is to be activated only when the tractor is in between two crop rows. Like in lane keeping systems for cars, drivers need to take control when they need to change direction. Thanks to this type of ADAS, the driver can focus on the farming work rather than on steering the vehicle.

Figure 1 shows the functional structure of the algorithm. The core of the localization is an Extended Kalman Filter (EKF) based on a control-oriented model of the tractor. This module estimates the distance from the vine row and the incident angle using the ultrasonic sensors signals, the steering angle, and the wheel velocity. Before being employed by the EKF, the ultrasonic sensor signals pass through an adaptive data selection step. The data selection module removes outliers considering the current vegetative state of the vineyard as estimated by the EKF. Practical and industrial considerations guide the

design of the system starting from the sensor selection to the algorithm, designed with the objective to be implemented on standard on-board electronic control units.
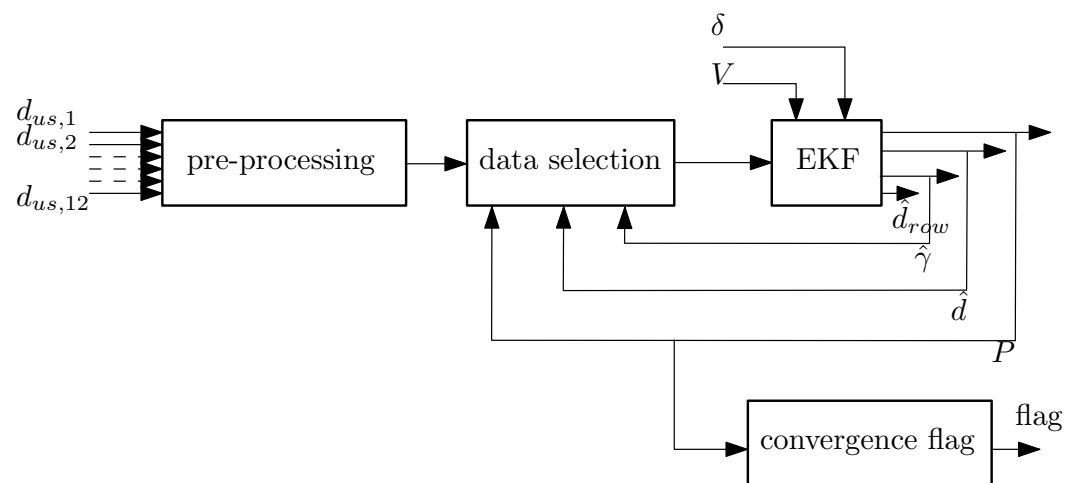
Figure 1. Localization algorithm functional architecture.

This work expands the preliminary results shown in Reference [18] from several standpoints:

- We introduce an adaptive data selection logic. We show that the adaptive data selection improves robustness against varying vegetative conditions.
- We introduce a converge detection logic that is to be employed as a master switch for the ADAS system.
- We considerably expand the validation considering data taken from the entire production cycle of a vineyard.

The structure of the paper is as follows: in Section 2, we design the localization algorithm detailing all its components and modules. Section 3 presents the experimental results under different conditions. In this section, we show-case the impact and importance of each module. Finally, Section 4 draws some final remarks.

2. Materials and Methods

We strive to implement a cost-effective and industrialisable solution. This requirement translates into selecting a simple and robust sensor suite. In particular, we consider a standard SDF SAME Frutteto agricultural tractor equipped with the following sensors:

- Wheel velocity V sensor. This is the standard wheel velocity sensor installed on the tractor. It is an encoder.
- A front wheel steering sensor. It provides the steering angle δ at the wheel. In addition, in this case, this is a standard sensor of the tractor.
- A set of 12 ultrasonic (us) sensors. This is the main sensor employed by the localization algorithm. It is produced by Laserline. The minimum measurable distance is 0.2 m and the maximum is 3.88 m. The manufacturer claims an accuracy of 0.01 m for obstacles closer than 1.2 m and 0.02 m for farther obstacles.
- A 3D LiDAR sensor installed on the vehicle hood. This sensor calculates the ground-truth and is used only for validation purposes. The LiDAR is a Quanergy M8 device. It has a 8 vertical layers spread over a vertical Field of View of 20 degrees. It performs 10 full rotations per second

In terms of actuators, the tractor is equipped with a steering actuator and a cruise control.

Figure 2 shows the installation of the sensors on the vehicle. The ultrasonic sensors are active time-of-flight sensors. The source emits a conic sound wave: any obstacle in the field of view of the sensor will return an echo and be detected as an obstacle. The sensor only

provides the distance of the obstacle not its direction. This is the main disadvantage with respect to LiDARs. A dedicated control unit polls the ultrasonic sensors sequentially with a sampling time of 0.05 s; thus, it takes 0.6 s to perform a complete acquisition of all sensors.
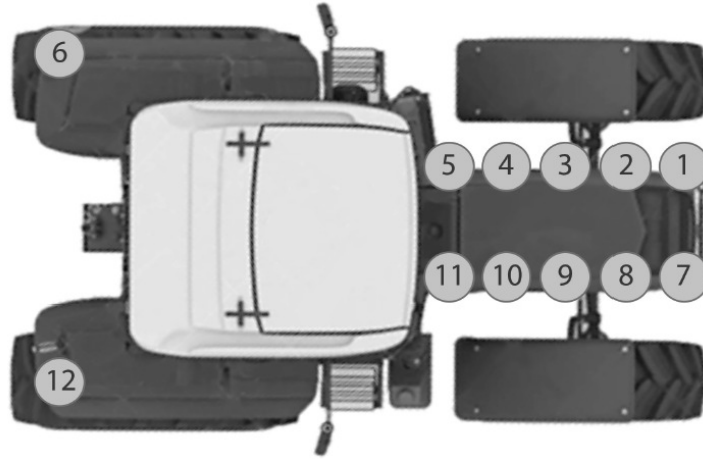
Figure 2. Mounting of the 12 ultrasonic sensors.

2.1. Problem Formulation

The row centering ADAS described in the Introduction operates locally with respect to the plant row. In particular, it controls the position of the tractor with respect to the left row, conventionally, the one on which the farmer operates. Given this context, the localization problem can be parametrized as shown in Figure 3.
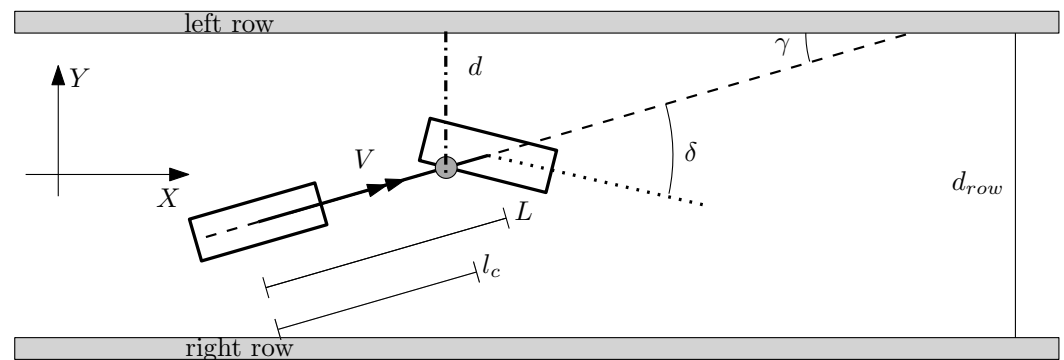
Figure 3. Single track model and main variables.

We define:

- d as the distance from the left row.
- d_{row} as the distance between the left and right rows.
- L is the wheelbase of the vehicle.
- l_c as the distance from the rear axle to the point where the 3D LiDAR is mounted.
- γ as the angle of incidence, i.e., the angle between the longitudinal direction of the tractor and the direction of the left row.
- δ as the steering angle.
- V as the magnitude of the vehicle velocity at the rear axle.
- r as the vehicle yaw rate.

By assuming straight rows and a constant longitudinal velocity, the kinematic model of the vehicle is:

$$\begin{cases} \dot{X} = V\cos(\gamma) - l_c r \sin(\gamma) \\ \dot{Y} = V\sin(\gamma) + l_c r \cos(\gamma) \quad , \\ \dot{\gamma} = r \end{cases} \tag{1}$$

where

$$r = -\frac{V}{L}\delta. \tag{2}$$

Figure 4 plots the comparison of the simulated distance and angle of incidence against the ones measured by the 3D LiDAR with the tractor traveling at 2 m/s.

The model correctly captures the dynamics of interest.
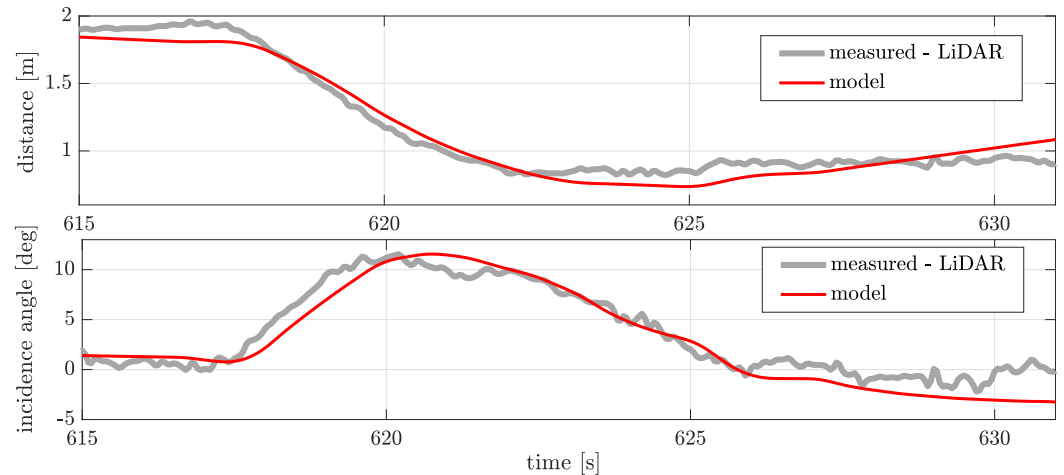
Figure 4. Comparison between simulated and measured position and angle of incidence.

2.2. Localization Algorithm

As also discussed in the Introduction and shown in Figure 1, the localization algorithm is composed of four main modules: data pre-processing, measurement selection, the Kalman Filter itself, and the convergence analysis. In what follows, we detail each component.

2.2.1. Ground-Truth Computation

Before describing the components of the localization algorithm itself, it is important to explain how the properties and performance of the algorithm are assessed. In this application, building the ground-truth is not trivial, and this will have some consequences on the interpretation of the results.

We build the ground-truth using a 3D LiDAR. The 3D LiDAR returns a point-cloud, where each point has its coordinates in the reference frame centered onto the LiDAR itself. The LiDAR will, thus, measure all obstacles comprising the ground. In order to measure the actual position of the tractor with respect to the left row, we first remove the points on the ground based on a volumetric filter and subsequently, we separate the points belonging to the left and right rows. Once the two rows are clustered, the RANSAC algorithm [19] fits a straight line describing the two rows. From the equation of the line, one can trivially derive d and γ.

As the attentive reader will notice, this approach does not rely on a unique definition of row. The ground-truth definition depends on the result of the RANSAC algorithm. Based on the vegetative state, the RANSAC algorithm may fit a line that goes through the center of the row, in case of a growth state that allows the LiDAR to pierce through the leaves and detect both sides of the row, or it could fit a line that goes through the leaves closer to the tractor in case of dense growth. This uncertainty is inevitable, and its effect will be discussed when needed.

2.2.2. Data Pre-Processing

The pre-processing module is responsible for two tasks (1) out-of-range removal and (2) sampling.

The electronic unit that manages the sensors outputs a conventional value when a sensor does not detect any obstacle. The pre-processing stage simply removes all measurements that are set to that value.

The 12 sensors are polled sequentially with a refresh rate of 50 ms; however, the electronic unit publishes, at each sampling, the entire measurement vector. This vector contains one current value and 11 *old* measurements. To avoid using a piece of information that can be up to 0.6 s old, the pre-processing stage filters the measurement vector by selecting only the most recently updated value.

2.2.3. Extended Kalman Filter

The core of the estimation is an Extended Kalman Filter [20], a model based prediction-correction estimation algorithm.

Our EKF implementation is based on a control-oriented model obtained by redefining the state variables of (1). By discretizing the model with step T_s, we obtain

$$\begin{cases} d(k+1) = d(k) - T_s V(k)\left(\gamma(k) + \frac{l_c}{L}\cos(\gamma)\delta\right) \\ \gamma(k+1) = \gamma(k) - T_s \frac{V(k)}{L}\delta \\ d_{row}(k+1) = d_{row}(k). \end{cases} \tag{3}$$

In the prediction model, the state d_{row}, i.e., the width of the row, allows the algorithm to effectively employ the sensors on the right to increase robustness in case of holes or protruding branches in the left row. The prediction model is completed by a measurement equation that describes how the measurements depend on the states of the system. Note that, despite having 12 sensors, the fact that they are sequentially polled means that only one measurement at a time is used. This yields a single, but time-varying, measurement equation. Depending on the position of the active sensor, the measurement equation takes the form of:

$$\begin{cases} d_{us_i} = \frac{d}{\cos(\gamma)} + l_i\tan(\gamma) - d_{T_i} \text{ for } i \in [1,6] \\ d_{us_i} = \frac{d_{row}-d}{\cos(\gamma)} - l_i\tan(\gamma) + d_{T_i} \text{ for } i \in [7,12] \end{cases}, \tag{4}$$

where

- i is the number of the sensor (see Figure 2);
- l_i is the longitudinal mounting offset with respect to the measurement point shown in Figure 3 of the *ith* sensor (taken positive toward the rear axle); and
- d_{T_i} is the lateral offset with respect to the center of the vehicle and taken positive.

From model (3), one can design an EKF using well known techniques. The tuning of an EKF consists in determining two matrices Q and R, where Q represents the process noise (i.e., the model uncertainties), and R the measurement noise. In particular, we adopted the following matrices parametrization:

$$Q = \begin{bmatrix} \frac{1}{\alpha} & 0 & 0 \\ 0 & \frac{1}{\beta} & 0 \\ 0 & 0 & \frac{1}{\lambda} \end{bmatrix} \qquad R = \begin{bmatrix} 1 & 0 & \cdots \\ \vdots & \ddots & \\ 0 & \cdots & 1 \end{bmatrix}. \tag{5}$$

The above expressions depend on λ, α, and β. In tuning the algorithm, one has to pay attention to make sure that the row distance is slowly updated with respect to the other two states. In building the rest of the algorithm, we will consider a first EKF tuning obtained via trial-and-error. Once we build the complete algorithm, we will then propose a quantitative tuning.

As a concluding remark on the EKF, it is important to consider that the filter assumes a Gaussian density of both the process and measurement noises. If these hypotheses do not hold, one could employ a number of other filters, most notably the particle filter [21].

However, particle filters tend to be very computationally demanding for the standard Electronic Control Units employed on tractors.

Matter-of-factly, in the following subsection, we will show how the measurement noise Gaussian hypothesis does not hold for the problem at hand, and we will propose a solution.

2.2.4. Data Selection

To better understand the type of noise affecting the ultrasonic measurements, refer to Figure 5.

Figure 5. The vineyard during the winter (left), spring (center), and summer (right).

It shows the same vineyard in three different periods of the production cycle: winter, spring, and summer. The characteristics of the raw ultrasonic signals are completely different in the three seasons. For example, in winter, it is very likely for a sensor to miss the closest tree and either return an *out of scale* or the distance to another object. Conversely, in summer, the sensors will very likely pick up protruding branches, which should not be considered in the distance computation, as the objective of the ADAS is to keep a desired distance from the entire row and not to follow the plants growth. Spring represents an intermediate condition.

Figures 6–8 more rigorously quantify the noise signature in the seasons. In all the experiments, the tractor manually drove at a constant distance from the row.
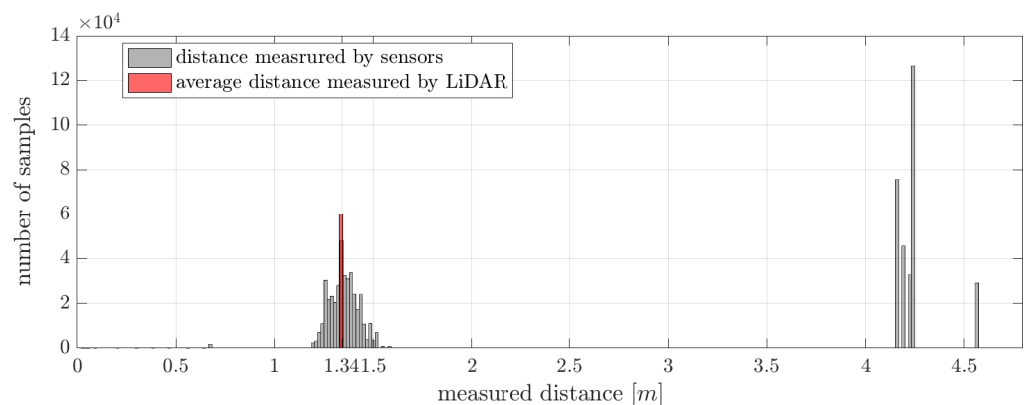
Figure 6. Measurement histogram for a straight trajectory during the winter.

These figures show the raw ultrasonic measurements obtained by applying the inverse measurement function (4) so that all measurements refer to the same point. Figure 6, referring to the winter scenario, shows an asymmetric and long-tailed distribution. As explained before, the long-tail of the distribution is due to the ultrasonic beams missing the first row. Clearly, the distribution is far from being Gaussian.
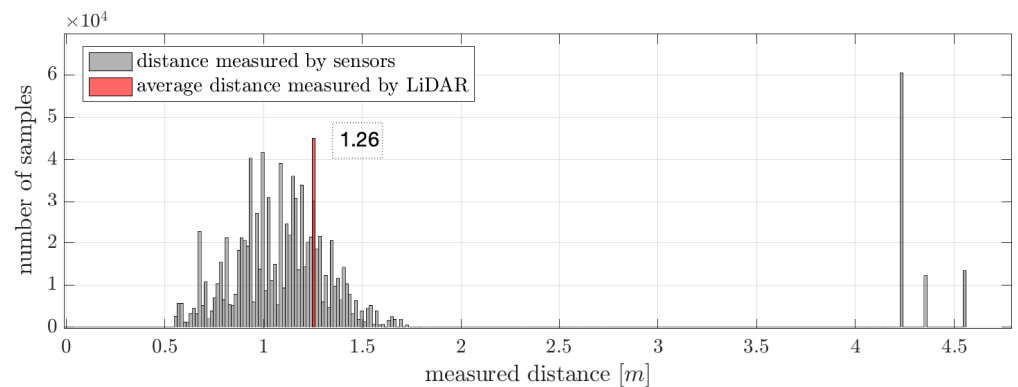
Figure 7. Measurement histogram for a straight trajectory at the end of the summer, before harvest.

Figure 7 plots the measurement histogram of a summer test. In addition, in this case, the measurement is not Gaussian, but it is asymmetric and skewed toward shorter distances with respect to the ground-truth. This is due to the protruding branches. Given the wide field of view of the acoustic sensors, it is more likely to pick up a protruding branch. The LiDAR ground-truth is less affected by this phenomenon, thanks to its higher resolution.

Finally, Figure 8 shows the situation during the spring. Here, the noise distribution approaches the ideal Gaussian case.
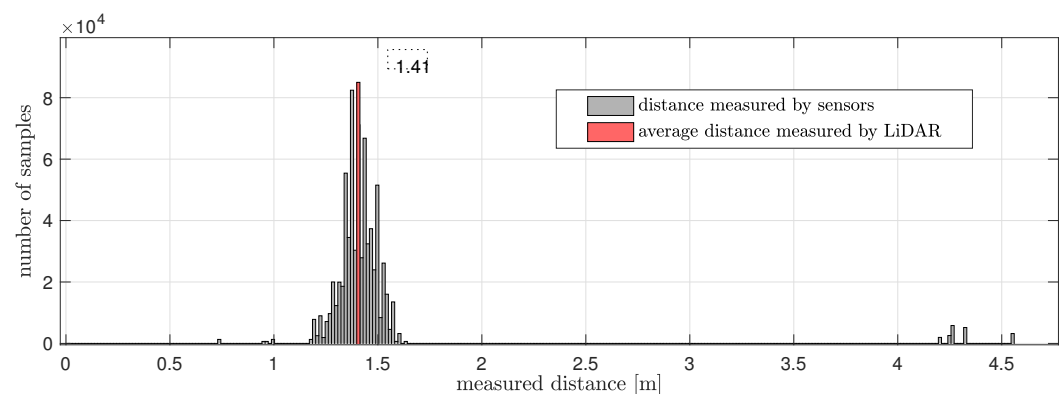
Figure 8. Measurement histogram for a straight trajectory during the spring.

Our approach deals with the non-Gaussian density of the noise with a data selection stage. This step uses the last available a posteriori estimante of the state to discard measurements that would otherwise bias the next algorithm iteration. Two moving windows (left and right) centered around the last available estimation implement this data selection. Figure 9 graphically represents the moving window method.

The most important tuning parameter of this module is the size of the window. If the size is too large, then its benefits are lost; if it is too small, the algorithm will neglect valuable measurements. Its optimal size depends on the condition of the vineyard; Figure 10 plots the Root Mean Square Error (RMSE) of the estimated distance as a function of the widow size in the three seasons for straight driving tests.
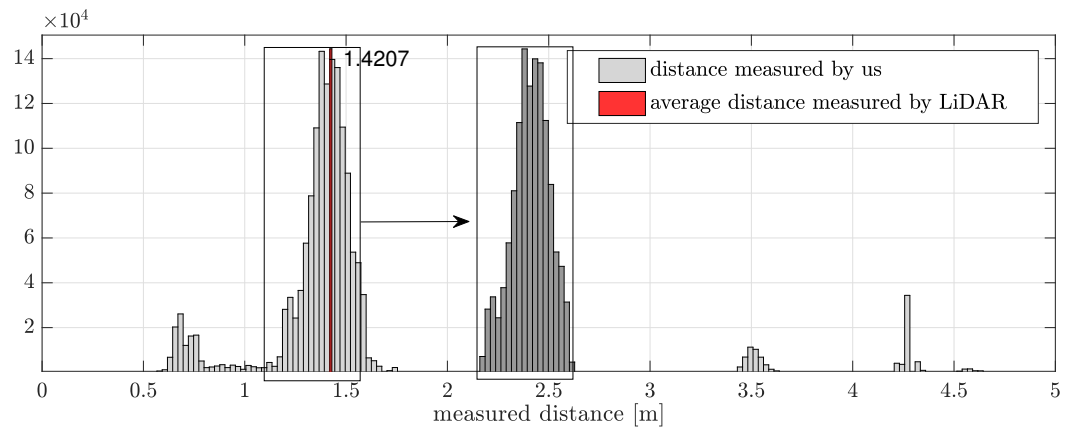
Figure 9. Graphical representation of the data selection stage.
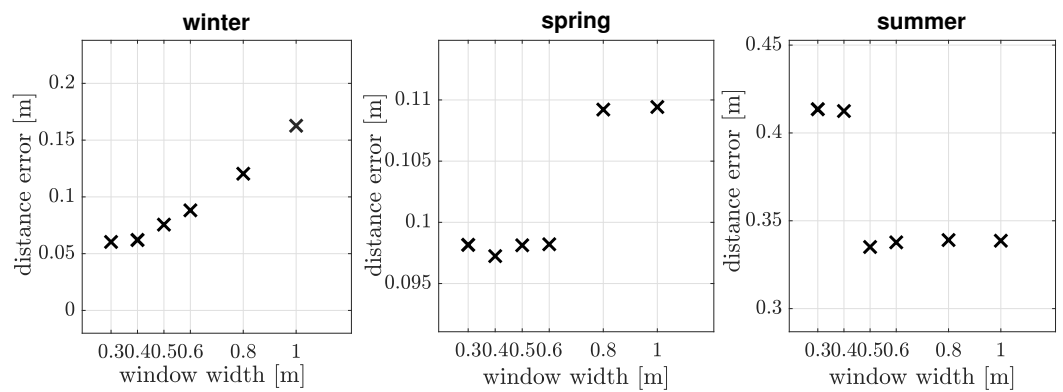
Figure 10. Row distance Root Mean Square Error (RMSE) as a function of the window size and seasonal conditions.

Note that different seasons show different trends. Winter shows a fairly smooth monotonically increasing trend. Spring shows a discontinuous *approximately* monotonically increasing trend, whereas summer exhibits a discontinuous *approximately* monotonically decreasing trend. The trend inversion in the summer makes the choice of a constant window size problematic. An all-season window would lay around the discontinuity of the characteristics and this could lead to robustness issues. To overcome this problem, we introduce a scheduling approach to automatically adapt the window size to the current vegetative conditions. The method exploits the fact that the Kalman Filter provides the estimation error covariance matrix P.

Figure 11 plots the relationship between the second element on the main diagonal of the covariance matrix and the variance of the estimation error of the incidence angle.
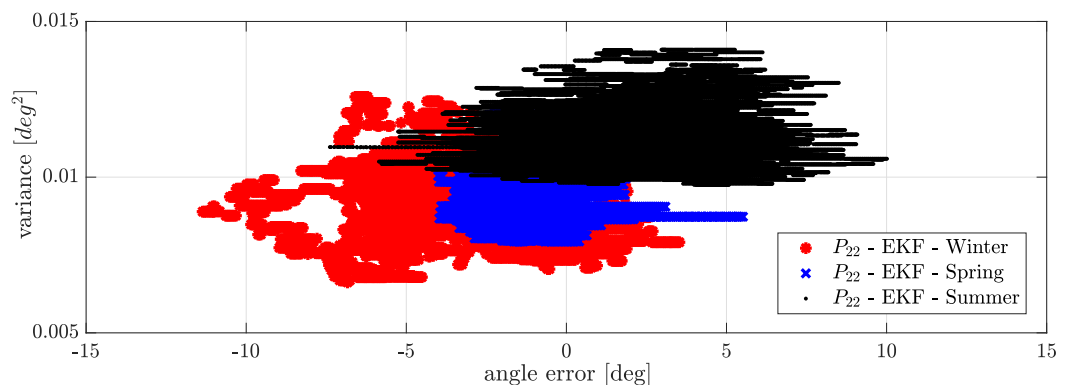
Figure 11. $P_{22}(t)$ for experiments for in different conditions: spring, summer, and winter.

The figure shows that the covariance matrix values tend to cluster around different values based on the condition of the vegetation. In particular:

- In spring, the covariance matrix values are narrowly clustered. This is due to the fact that the spring vines tend to be orderly and uniform and sufficiently dense to provide enough acoustic echoes.
- In summer, the variance is more spread and higher than in spring. The summer growth tends to create a discontinuous row with many protruding branches.
- In winter, we have the highest variability of the covariance matrix but, at same time, the lowest minimum values. In this season, the algorithm has fewer features to use for localization (explaining the high variability), but, at the same time, the few features are better aligned and representative of the straight row.

From the above considerations (we found that these considerations hold also for P_{11}, but P_{22} provides a crispier clustering), we introduce a scheduling of the pre-selection window as a function of the instantaneous value of the covariance matrix. Figure 12 plots the proposed scheduling. The scheduling does not simply connect the minima of the characteristics in Figure 10 but exploits the fact that, before and after the discontinuities, the characteristics are rather flat and imposes a robustness margin.
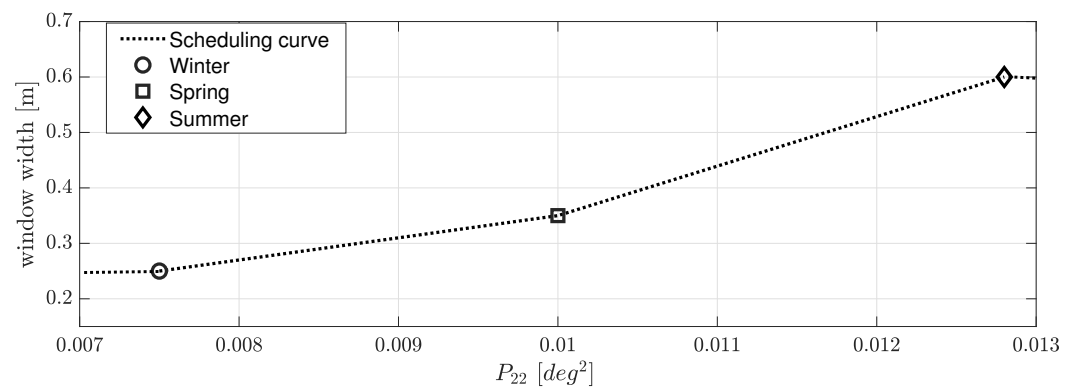
Figure 12. Scheduling of the data selection window as a function of $P_{22}(t)$.

The pre-selection window may deprive the estimation of all measurements. As it will be shown later, the estimation can remain accurate for a few seconds without measurements, but it will inevitably diverge for a sufficiently long period of time without valid measurements. In this case, if the estimated position diverges from the real one, the window (that is centered on the the estimated position) may prevent the algorithm from receiving accurate measurements. To avoid this phenomenon, the data selection is deactivated, if no measurement passes the test for a sufficiently long period of time.

2.2.5. EKF Tuning

Now that we described the entire algorithm, all the elements are in place to propose the final tuning of the EFK, namely to determine the values appearing in (5). The tuning is done via solving the following optimization problem:

$$\min_{\alpha, \beta, \lambda} \left\{ \sqrt{\frac{\sum\left(\frac{d_{real} - \hat{d}}{d_{real}}\right)^2}{N}} + \sqrt{\frac{\sum\left(\frac{\gamma_{real} - \hat{\gamma}}{\gamma_{real}}\right)^2}{N}} \right\}. \tag{6}$$

In order to avoid overfitting, we include data taken from three different seasons.

2.2.6. Convergence Flag

The EKF is a recursive algorithm, as such, it needs time to converge to an accurate estimation. During this period of time, the accuracy may not be enough to guarantee an

effective tracking of the row distance. It is, thus, important to provide a flag to the ADAS that indicates whether the estimation is accurate.

In the Kalman Filtering framework, the covariance matrix evolution provides indications on whether the estimation algorithm has converged. However, the absolute values of the covariance matrix elements are not indicative of the converge because, as seen in the above subsection, the covariance matrix depends on the seasonal conditions. For this reason, the activation logic computes the magnitude of the time derivative of P_{22} and uses a threshold on that value to detect convergence.

In order to avoid chattering, we implement a hysteresis mechanism. In particular, at start-up the flag is false and it is turned true when the time derivative of P_{22} crosses the threshold from above.

3. Discussion

This section discusses, showing data collected on the field, the main features of the localization algorithm and validates its performance in different scenarios. We will discuss: the role of the row width d_{row} state in the estimation, the adaptive data selection benefits, and the convergence flag. We also show the complete validation of the estimation. The data discussed in the section refers to a different set of experiments than the one employed in the tuning.

3.1. State Augmentation

We augmented the prediction model with an additional fictitious state: the distance between the right and left rows. Strictly speaking this is not a dynamic state, but its inclusion increases the robustness of the algorithm.

Figure 13 shows a straight driving experiment. It compares the state-augmented solution against a solution that does not estimate the distance between the left and right rows.
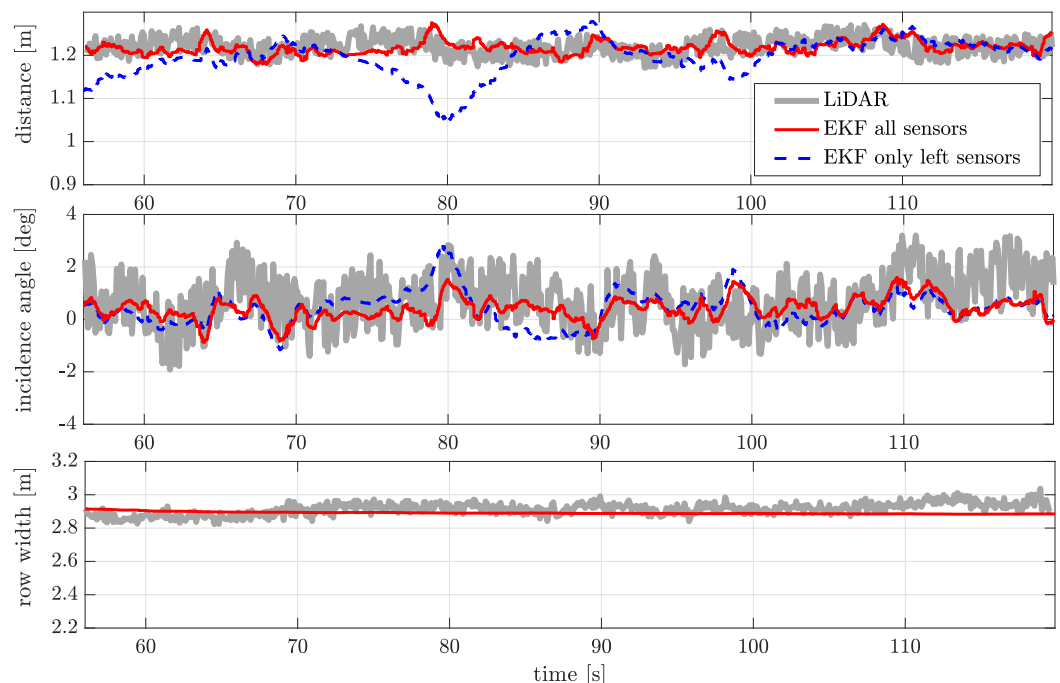
Figure 13. Estimation results during a straight driving experiment, comparison between using only the left sensors and state augmentation.

From figure, one can note that all three variables are correctly estimated when using the augmented state version. At around t = 80 s, the tractor drives in front of a protruding branch. If the algorithm does not implement the state augmentation, and thus uses only

the left sensors, the estimation is not accurate. On the other hand, the augmented version of the algorithm is accurate also when a protruding branch is encountered because the sensors on the right, together with $\hat{d}_{row}$, still provide information on the distance from the left row.

3.2. Data Selection

Figures 14–16 plot the comparison between the estimated distance and incidence angle in the three seasons for different versions of the data selection module. In detail, the figures show the results with an EKF that does not use any data preselection; an EKF that uses a fixed window of 0.5 m and the EKF with the adaptive data selection.
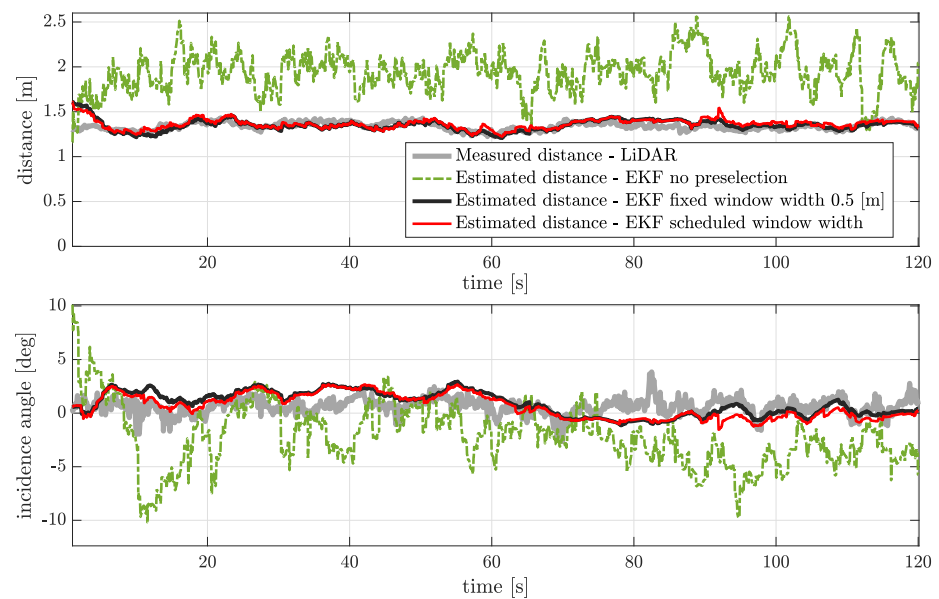
Figure 14. Estimation results during the winter—data selection effect assessment.
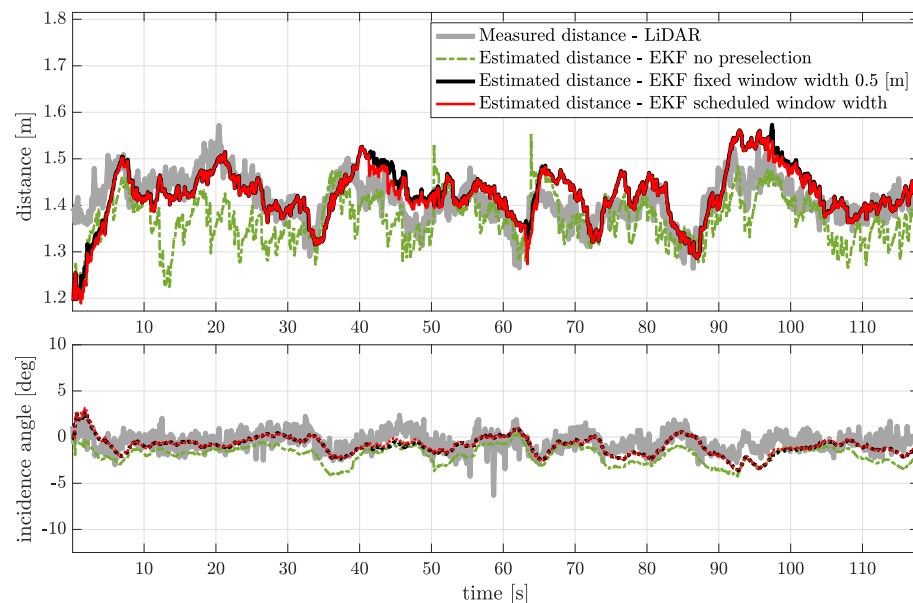
Figure 15. Estimation results during the spring—data selection effect assessment.
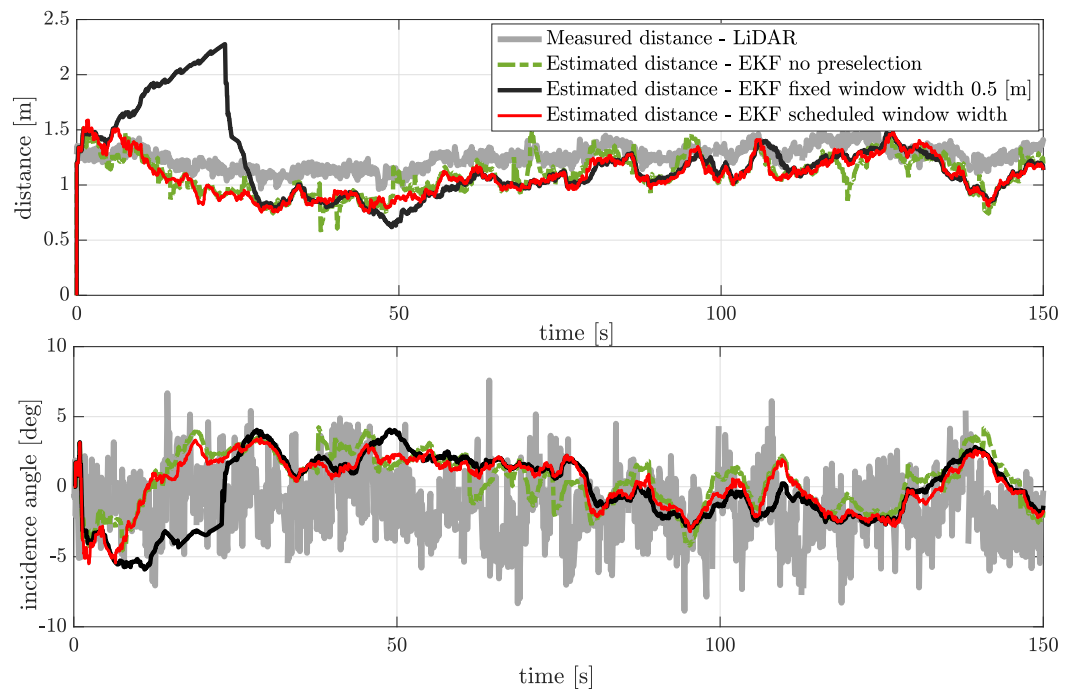
Figure 16. Estimation results during the summer—data selection effect assessment.

From the figures, one concludes that:

- In winter, without data selection, the estimated distance exhibits both a large bias and noise. There are too many measurements that do not fall on the adjacent row.
- In winter, the data selection step considerably improves the accuracy of both estimated variables.
- In winter, the benefit of the adaptive window is marginal. Recall that the error dependence on the window size is fairly smooth in winter.
- In spring, both solutions of data filtering provide a benefit against the EKF without data selection.
- The summer experiment is the one that better underlines the advantages on the adaptive window. The experiment shows that the solution with the fixed window can be subject to local divergence. The adaptive windowing helps preventing these issues because it selects a window size far from the performance discontinuity shown in Figure 10.

3.3. Validation for Different Driving Scenarios

We tested the complete algorithm in more challenging maneuvers: a sudden steering action (see Figure 17) and a sinusoidal maneuver (see Figure 18); both performed while driving at 2 m/s. These maneuvers are useful to assess the dynamic performance of the estimation.
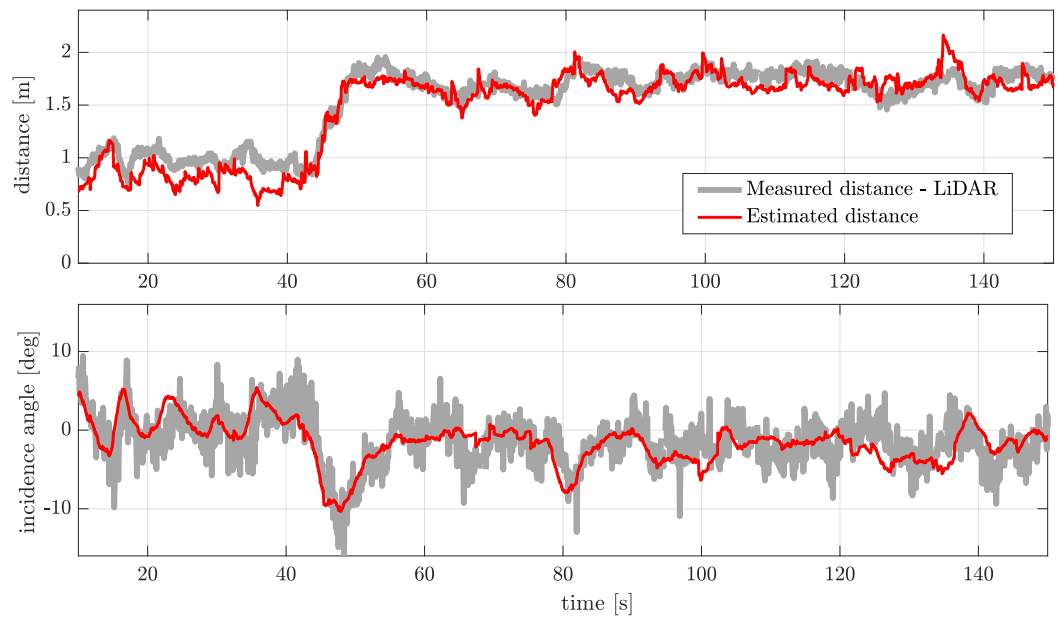
Figure 17. Estimation results for a step-like change of distance.
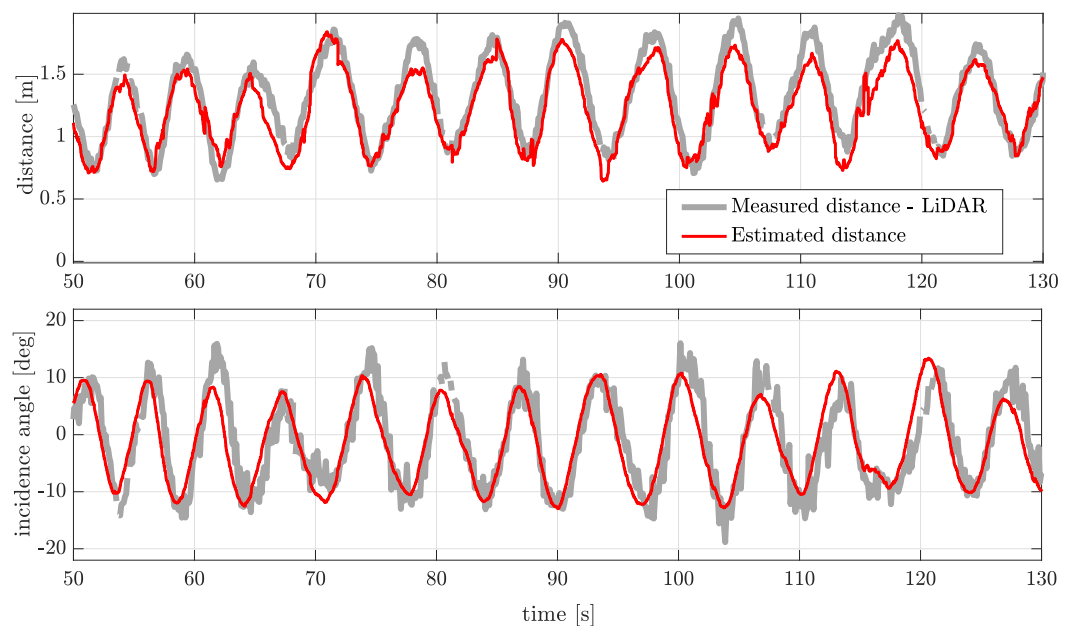
Figure 18. Estimation results for a sinusoidal maneuver.

The algorithm accurately tracks the distance from the row and the angle of incidence also during dynamic maneuvers. Overall, considering all the available experiments (winter, spring, and summer, straight and dynamic maneuvers) the proposed estimation algorithm yields a Root Mean Squared Error (RMSE) of the distance of about 16 cm, and on the incidence angle of 2.6 degrees.

3.4. Validation of the Convergence Flag

The converge flag algorithm is based on a threshold s_i for the time derivative of the element P_{22} of the covariance matrix. The tuning of the threshold is based on empirical considerations and determines a trade-off between the time needed to confirm convergence and the risk of wrongly signaling a convergence.

In order to provide a quantitative tuning method, we define the following cost indexes:

- Convergence advance (Δ_{t_i}): the time between when the threshold triggers and when the estimation error reaches 0 for the first time. The larger this time, the quicker the flag response.
- Convergence error (e_i): the distance error at the time the threshold is crossed. The smaller this index, the better.

One expects that as s_i increases, the algorithm triggers earlier, but, at the same time, it will have a greater convergence error. Figure 19 plots the average of the two cost indexes over the available data as s_i varies.
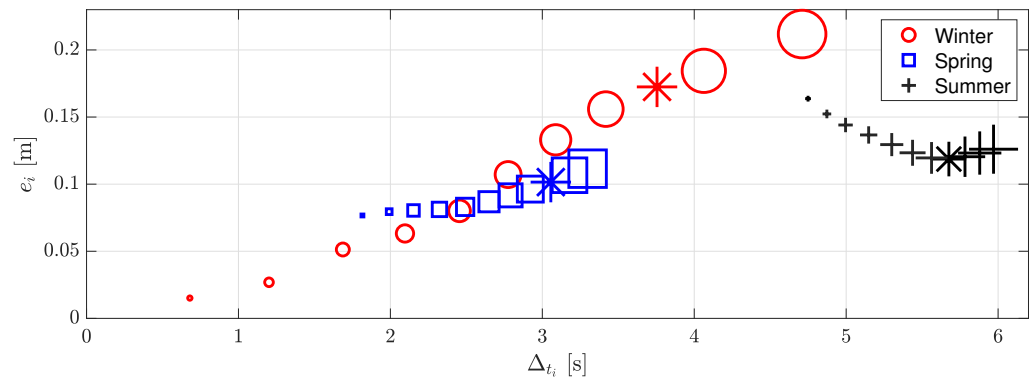
Figure 19. Converge flag cost function as s_i varies for three different seasonal conditions. The larger the symbol, the higher s_i.

Note that, for winter and spring conditions, one finds the expected results. The summer follows a different trade-off. Furthermore, one should consider that low values of s_i determine unwarranted deactivations. Based on the above considerations, we propose a threshold of 0.002 deg^2, which corresponds to the stars in the figure.

Figure 20 plots the time domain analysis of the convergence flag.
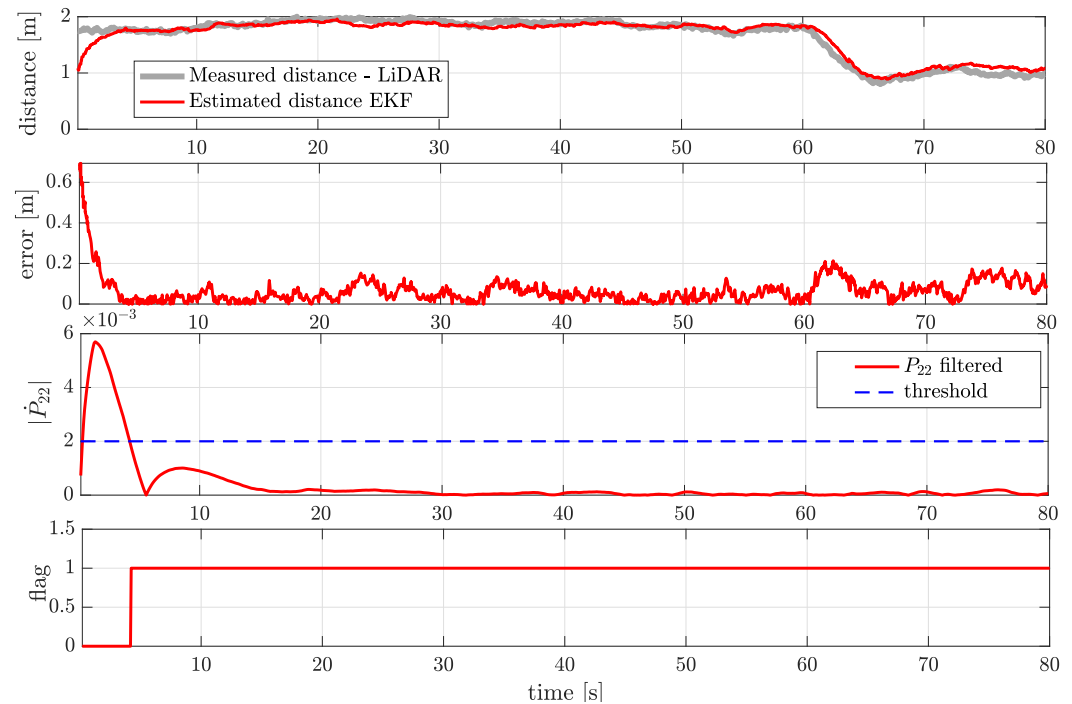
Figure 20. Converge flag time domain analysis.

The figure shows that as soon as the system is switched on, the localization error is larger than 0.5 m, and it converges to its steady state value in around 3.5 s. As expected,

the absolute value of the time derivative of P_{22} decreases as the accuracy improves, and the flag activates at around 4.5 s. This convergence time is adequate to the activation dynamics of an ADAS. Further consider that, at around $t = 60$ s, the tractors gets closer to the row, and this change of trajectory does not negatively affect the convergence flag.

4. Conclusions

The paper discusses a cost-effective localization system for (semi-) autonomous navigation of agricultural tractors in vineyards. The algorithm estimates the distance from the left vine row and the incidence angle.

The estimation algorithm is based on an Extended Kalman Filter and a data selection step. The EKF is based on a model that also considers, as a state, the distance between the left and right rows. This makes the estimation more robust to protruding branches and holes in the rows.

Our research indicates that, because of varying seasonal conditions, a single tuning of the localization algorithm does not provide robust performance. We, thus, introduce an adaptive term that further robustifies the estimation by discarding wrong measurements. The adaptive term uses the information from the algorithm to automatically determine the optimal size of the filtering window. This adaptation allows the algorithm to discard measurements that pass through the rows or hit branches far from the main row.

We use a complete experimental campaign to exemplify the main features of the estimation. In particular, we find a distance RMSE of 16 cm and an angular RMSE of 2.6 degrees.

The estimation of both the distance and the incidence angle will enable the implementation of full-state trajectory feedback control.

Author Contributions: Conceptualization, S.F. and M.C.; methodology, S.F. and M.C.; software, S.F.; validation, S.F. and P.C.; formal analysis, M.C.; investigation, S.F.; resources, P.C.; data curation, S.F.; writing—original draft preparation, M.C.; writing—review and editing, M.C.; visualization, M.C.; supervision, M.C. and S.M.S.; project administration, P.C.; funding acquisition, S.M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the SDF Group.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors wish to thank Nunzio Ghiraldi Farm for granting access to their vineyards.

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

References

1. Surbrook, T.; Gerrish, J.B. Mobile Robots in Agricolture. Proceedings of the First International Conference on Robotics and Intelligent Machines in Agriculture. *Am. Soc. Agric.* **1984**, *84*, 30.
2. Prasad, D.; Singla, K.; Baggan, V. System Model for Smart Precision Farming for High Crop Yielding. *J. Comput. Theor. Nanosci.* **2019**, *16*, 4406–4411. [CrossRef]
3. Romeo, L.; Petitti, A.; Marani, R.; Milella, A. Internet of Robotic Things in Smart Domains: Applications and Challenges. *Sensors* **2020**, *20*, 3355. [CrossRef] [PubMed]
4. Dong, F.; Heinemann, W.; Kasper, R. Development of a row guidance system for an autonomous robot for white asparagus harvesting. *Comput. Electron. Agric.* **2011**, *79*, 216–225. [CrossRef]
5. Reid, J.F.; Zhang, Q.; Noguchi, N.; Dickson, M. Agricultural automatic guidance research in North America. *Comput. Electron. Agric.* **2000**, *25*, 155–167. [CrossRef]
6. Wang, X.; Fu, M.; Ma, H.; Yang, Y. Lateral control of autonomous vehicles based on fuzzy logic. *Control Eng. Prac.* **2015**, *34*, 1–17. [CrossRef]
7. Lenain, R.; Thuilot, B.; Cariou, C.; Martiner, P. A new nonlinear control for vehicle in sliding conditions: Application to automatic guidance of farm vehicles using RTK GPS. In Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA'04), New Orleans, LA, USA, 26 April–1 May 2004; Volume 5, pp. 4381–4386.
8. Guo, J.; Li, X.; Li, Z.; Hu, L.; Yang, G.; Zhao, C.; Fairbairn, D.; Watson, D.; Ge, M. Multi-GNSS precise point positioning for precision agriculture. *Prec. Agric.* **2018**, *19*, 895–911. [CrossRef]

9. Ekambaram, V.N.; Ramchandran, K. Distributed high accuracy peer-to-peer localization in mobile multipath environments. In Proceedings of the 2010 IEEE Global Telecommunications Conference GLOBECOM 2010, Miami, FL, USA, 6–10 December 2010; pp. 1–5.

10. Liu, K.; Lim, H.B.; Frazzoli, E.; Ji, H.; Lee, V.C. Improving positioning accuracy using GPS pseudorange measurements for cooperative vehicular localization. *IEEE Trans. Veh. Technol.* **2013**, *63*, 2544–2556. [CrossRef]

11. Subramanian, V.; Burks, T.F.; Arroyo, A. Development of machine vision and laser radar based autonomous vehicle guidance systems for citrus grove navigation. *Comput. Electron. Agric.* **2006**, *53*, 130–143. [CrossRef]

12. Guerrero, J.M.; Ruz, J.J.; Pajares, G. Crop rows and weeds detection in maize fields applying a computer vision system based on geometry. *Comput. Electron. Agric.* **2017**, *142*, 461–472. [CrossRef]

13. Barawid, O.C., Jr.; Mizushima, A.; Ishii, K.; Noguchi, N. Development of an autonomous navigation system using a two-dimensional laser scanner in an orchard application. *Biosyst. Eng.* **2007**, *96*, 139–149. [CrossRef]

14. Illingworth, J.; Kittler, J. A survey of the Hough transform. *Comput. Vision Graph. Image Process.* **1988**, *44*, 87–116. [CrossRef]

15. Weiss, U.; Biber, P. Plant detection and mapping for agricultural robots using a 3D LIDAR sensor. *Robot. Autonom. Syst.* **2011**, *59*, 265–273. [CrossRef]

16. Hiremath, S.A.; Van Der Heijden, G.W.; Van Evert, F.K.; Stein, A.; Ter Braak, C.J. Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter. *Comput. Electron. Agric.* **2014**, *100*, 41–50. [CrossRef]

17. Astolfi, P.; Gabrielli, A.; Bascetta, L.; Matteucci, M. Vineyard Autonomous Navigation in the Echord++ GRAPE Experiment. *IFAC-PapersOnLine* **2018**, *51*, 704–709. [CrossRef]

18. Furioli, S.; Onesto, L.; Corno, M.; Cesana, P.; Savaresi, S. Cost-effective Tractor Localization for Semi-Autonomous Vineyard Operations. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–6.

19. Schnabel, R.; Wahl, R.; Klein, R. Efficient RANSAC for point-cloud shape detection. Wiley Online Library. *Comput. Graph. Forum* **2007**, *26*, 214–226. [CrossRef]

20. Chui, C.K.; Chen, G. *Kalman Filtering*; Springer: Berlin/Heidelberg, Germany, 2017.

21. Gustafsson, F.; Gunnarsson, F.; Bergman, N.; Forssell, U.; Jansson, J.; Karlsson, R.; Nordlund, P.J. Particle filters for positioning, navigation, and tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 425–437. [CrossRef]