POLITECNICO DI TORINO
Master degree in Computer Engineering

Master Degree Thesis

Position determination of a mobile robot in a precision agriculture scenario

Improving agricultural UGV localization using georeferenced low-complexity maps

Supervisors
Dr. Fabrizio Dabbene
Dr. Martina Mammarella
Prof. Davide Ricauda Aimonino

Candidate
Cesare Donati

October 2021

Abstract

The adoption of autonomous vehicles in agricultural scenarios can aspire to become a reality if the validation of their effectiveness is sustained by the contemporary and shared improvement of all those technological gaps identified by current research projects. In particular, a crucial point is still linked to the autonomy of robots, related to the concept of guidance, navigation and control. Using a combination of conceptually different sensing techniques and integrating the subsequent data, more accurate property estimates can be provided, leading to more robust management and increased adoptability of sensor-based crop management. For example, to properly locate and operate autonomous vehicles for in-field tasks, the knowledge of their instantaneous position needs to be combined with an accurate spatial description of their environment. In agricultural fields, especially when operating within crops, GPS data are not reliable nor always available, therefore high-precision maps are difficult to be obtained and exploited for in-field operations. Recently, low-complexity, georeferenced 3D maps have been proposed to reduce computationally demand without losing relevant crop shape information.
This thesis focuses on the localization of an unmanned ground vehicle (UGV) which is moving between the rows of a vineyard performing farming operations. Due to the limited space where the autonomous vehicle is moving, an high precision filter needs to be designed to avoid collision with the crops, providing additional information on the vehicle's position and compensating the lack of accuracy of the GPS in the field. In particular, we propose an innovative approach that allows us to *fuse* data collected by *distance sensors* and the information obtained with the use of an a priori provided *simplified map* to improve the estimation of the UGV location within crops. Indeed, starting with the distance measurements between the UGV and a crop row, measured by the ultrasound sensors mounted on the vehicle, it is possible to obtain a precise information on the UGV position by fusing it with the data coming from the georeferenced map, exploiting a trigonometric approach. This improved estimation of the UGV location can be integrated with additional data, merging it with those provided by other sensors as GPS and IMU, using classical filtering schemes. This result leads to a very precise estimation of the pose of the vehicle within the rows and allows to perform an efficient control action to make the UGV follows the designed trajectory avoiding collisions.

Contents

Introduction

In precision agriculture, autonomous ground and aerial vehicles can lead to favourable improvements to in-field operations, extending crop scouting to large areas and performing in-field tasks in a timely and effective way [1]. However, automated navigation and operations within complex scenarios require specific and robust path planning, navigation and control [2]. Thus, in addition to a proper knowledge of their instantaneous position, robotic vehicles and machines require an accurate spatial description of the environment. Hence, reliable and high-precision maps shall be realized and used to properly locate the vehicle within its operative scenario and to properly operate it. In the autonomous driving framework, several solutions have been proposed, where digital maps data are merged with those obtained by other sensors mounted on-board the vehicle, e.g. Global Positioning System (GPS), vision-based sensors and Inertial Measurement Unit (IMU).
For example, in [3], the authors propose a method that supports current positioning provided by a LiDAR sensor with the processed point data referenced to high-precision maps. The vehicle position estimation is improved by filtering the map data and extracting landmarks in the form of building edges and infrastructure objects. A map-based vehicle self-localization approach is described in [4], where information from the vehicular environment perception is associated with data of a high-precision digital map in order to deduce the vehicle's position. In [5], a probabilistic graph-based navigation algorithm resilient to GPS errors is presented, where a robust trajectory estimate and maps of the surrounding environment are generated by fusing GPS pseudo-range and LiDAR odometry measurements with 3D building maps. In [6], a research on an application that localizes the ego-vehicle in its lane and allows a precise positioning and a lateral control using an accurate digital map of the lane marking as a powerful additional sensor is presented. Finally, a novel two-step approach to vehicle positioning founded on the appropriate combination of the in-car sensors, GPS signals, and a digital map is described in [7] where a further improvement of the estimated position

is obtained by a comparison of the continuous vehicle trajectory with the candidate trajectories on a digital map.

On the other hand, in the agricultural field, when operating within crops, GPS data are neither reliable nor always available. In addition, high-precision maps are difficult to be obtained and exploited for in-field operations. Recent studies focused on the development of 3D models [8], using point clouds or triangulated meshes, generated by 3D sensors (e.g., LiDAR [9] and depth cameras [10]) or photogrammetry from structure from motion (SfM) algorithm, processing appropriate sets of 2D images (see e.g., [11, 12]). In agriculture, 3D modelling representations would facilitate comprehension of the environment, but proper algorithms for detecting and mapping crops and identifying soil and obstacles are needed [13, 14]. This task is not trivial since large 3D models of crops, including remotely sensed imagery and measurements made using in-field or on-vehicle sensors, require new processing algorithms to process big data and to extract appropriate information depending on the required final goal [15].
In [16], the authors propose a localization approach to estimate the vehicle pose relative to a global navigation satellite system (GNSS)-referenced map of crop rows fusing crop row detections with GNSS signals to obtain a pose estimate with the accuracy comparable to a row-following approach in the heading and lateral offset, while at the same time maintaining at least GNSS accuracy along the row.

However, reducing the amount of data is crucial to minimize computational times for large original datasets, thus enabling the exploitation of high-precision maps or 3D point cloud information in real-time during field operations. When considering scenarios involving cooperating machines and robots, data reduction will allow rapid communication and data exchange between in field actors. Hence, an innovative modelling framework can be exploited to semantically interpret 3D point clouds of vineyards and to generate low complexity 3D mesh models of vine rows, as the one presented in [17]. The information gathered from these simplified maps can be combined with those provided by the sensors equipped on board the vehicle to obtain a more accurate estimate of the vehicle location and orientation, that leads to a more robust localization.
In [18] the concept of sensor fusion relevant to precision agriculture is analyzed and a framework for future research in this area is provided by exploring some of the latest practices and research works about sensor fusion.

Case study

In the case considered in this thesis, an Unmanned Ground Vehicle (UGV) is moving within the rows of a vineyard, performing some farming tasks along its route. In order to define the correct control input to make the UGV follow the desired trajectory, the position, the orientation and the speed of the vehicle needs to be estimated with an high level of precision to avoid collisions while it is moving in the tight space between the crop rows. To accomplish those needs, the use of sensors like GPS or IMU is not sufficient, thus additional source of information have to be introduced.
As stated before, some possible solutions where sensors data are merged with high-precision map information are proposed in the literature. For example, in [19] a map-based localization approach for a robot is presented: a segment of a map is built from distance sensor data and the exploration of the environment, then, a map matching process is carried out by matching this segment with the global map. Another solution is presented in [20], where a map-aided adaptive fusion scheme uses map constraints to detect and mitigate GPS errors in urban environments: after an initialization phase, the method estimates the currently active map segment using dead-reckoning and a robust map-matching algorithm. A different approach is reported in [21], where a precise localization method for autonomous driving systems is obtained by correcting GPS bias error with camera vision systems and a road map database. Similarly, in [22], GPS and inertial navigation solutions are improved with vision-based measurements of nearby lanes and stop lines referenced against a known map of environmental features.

It is clear that using a map to provide additional information to compensate other sensors lack in the localization process is a fundamental requirement for this thesis. In the literature, most of the presented cases exploit sensors data to build online representations of the environment that are matched with a digital map, performing *map matching* techniques. *Vision system* like cameras or LiDAR are used to find points of interest in the surroundings that can be referenced in a known map.
On the other hand, in this work, an innovative approach is presented to gain further position information by *directly fusing* data from additional distance sensors and a georeferenced *low-complexity* map, avoiding time-consuming map-matching processes and the use of expensive and elaborated vision systems. In particular, this procedure allows to *merge* the distance measurements from the rows collected by ultrasonic sensors mounted on the vehicle and geometric data obtained by the low-complexity map to improve the final

estimation of the vehicle location within crops. Then, this more accurate estimation of the UGV location can be further improved and integrated with orientation, speed, and additional position data, merging them with data provided by the previous mentioned sensors mounted on board the UGV, i.e. GPS and IMU, using classical filtering schemes, e.g., Kalman filter [23]. The proposed scheme is summarized in Figure 1.

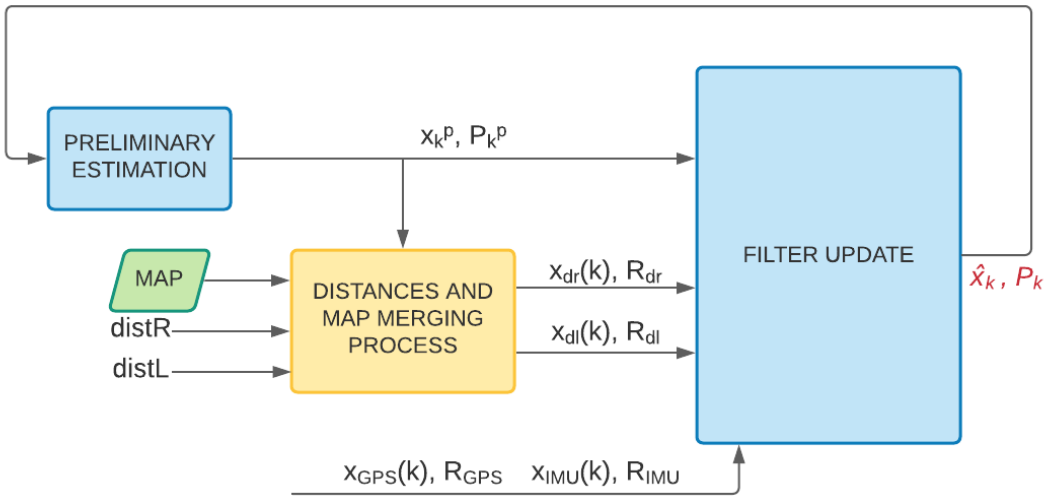


Figure 1: Given the left and right measured distances $distR, distL$ and the preliminary estimation x_k^p (affected by its uncertainty, defined by P_k^p) at time k, those data can be merged with the georeferenced information provided by the *map* to obtain additional position data $x_{dr}(k)$, $x_{dl}(k)$ and their errors R_{dr}, R_{dl} and update the state estimation, later fused with the data coming from GPS and IMU, i.e. x_{GPS} and x_{IMU} respectively, exploiting a filtering scheme to propagate the state estimation at time $k+1$.

Hence, starting with the estimated position at the previous time step $x(k)$, the distances measured by the ultrasound sensors between the vehicle and the rows are merged with georeferenced map data through a trigonometric approach. This provides an additional precise measure of the position of the UGV. Moreover, since the uncertainty of the distance sensors needs to be also considered, at every time step the position of the UGV is bounded by a Gaussian ellipsoid, which represents the probability of the points whose distances from the rows are coherent with the measured ones and their possible errors. To obtain this information directly from the measured values, the sensed distance and its nominal error (i.e. minimum and maximum possible distances) are considered as a parallel pair of constraints (half-planes) on

the starting estimate and the ellipsoid method is used to update the initial prediction and its uncertainty [24] (see Figure 2).

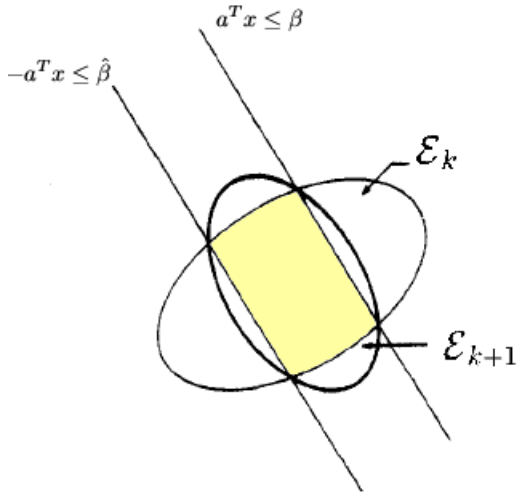


Figure 2: Two half-planes (constraints) are considered simultaneously and the minimum volume ellipsoid $\mathcal{E}_{k+1}$ (updated estimate) containing the area constrained among the ellipsoid $\mathcal{E}_k$ (starting estimate) and the two half-planes (yellow area) is obtained.

Since late 1960s, different approaches based on the ellipsoid method have been proposed in the literature, e.g. [25, 26]. Originally, the ellipsoid method was used in optimization to minimize convex functions. Indeed, it allows to generate a sequence of ellipsoids whose volume uniformly decreases at every step, thus enclosing a minimizer of a convex function. The classic formulation is used to obtain the ellipsoid of minimum volume $\mathcal{E}_{k+1}$ which contains all the points in $\mathcal{E}_k$ that also belong to the half-plane passing through its center x_k. In the *deep cut* variant [27], the half-plane is not constrained to pass through the center of the ellipsoid. More recently, [28] provides a different method to find the ellipsoid that tightly bounds the intersection between two given ellipsoids. Another method is described in [29], which allows to use simultaneously the constraints imposed by a pair of *parallel cuts* and to generate the new ellipsoid $\mathcal{E}_{k+1}$ having minimum volume and containing all the points between the two half-spaces (see Figure 2).
As previously mentioned, differently from the literature, in this case the ellipsoid method is used for a different, innovative purpose to get the updated measure of the position and its related uncertainty, starting from the information about the distance between a row and the UGV, provided by ultrasound

sensors.

The thesis is structured as follows. In Chapter 1 a detailed description of the system is presented, reporting the characteristics of the UGV, details on the equipped sensors and a brief description of the georeferenced map. Chapter 2 is about the localization of a mobile robot: after a brief introduction, the Kalman filter is described into details and the navigation setup of the robot with the available information and the data provided by the algorithm is presented. Chapter 3 explains how the ellipsoid method operates, the preliminary checks on the constraints, the parameters that should be defined to perform the parallel cuts variant, and its results. In Chapter 4 a detailed step-by-step procedure of the filter design is reported, starting from the model initially considered, up to how the map and the sensors are used to obtain information on the position. Chapter 5 shows the main results obtained with this approach and Chapter 6 focuses on the main conclusions and the possible future extensions.

Chapter 1

System description

The system considered in this thesis is a 4 wheel-steering electric UGV which is moving in a vineyard, performing some in-field operations during its route. In order to make the UGV following the desired trajectory autonomously between vine rows, the vehicle is equipped with a set of sensors which provide essential information for the autonomous navigation of the mobile robot, like GPS, IMU and distance sensors. Moreover, a georeferenced low-complexity map is provided to have also a spatial description of the environment, useful to reach a high level of precision for the localization of the UGV. In this chapter, each element of the considered system is analyzed, with particular focus on the features of the equipped sensors and the map.

1.1 UGVs

An UGV is a vehicle which moves in contact with the ground in an autonomous way, with no interaction with any (human) operator (Figure 1.1). Typically, it is adopted in inconvenient or dangerous situations, or scenarios where the presence of a human operator is impossible or not necessary. The autonomy of the vehicle is reached thanks to a set of sensors equipped on the robot that provides essential information on the environment where it is moving. Moreover, UGVs are able to autonomously take decisions on their behaviour to track given trajectories with the lower possible errors.
In precision agriculture, the use of autonomous robots for in-field tasks helps the farmer to increase crop productivity, ecosystem sustainability and business profitability. In the case of UGVs, their unique ability to travel through fields while transporting sizable payloads makes them ideal tools to be exploited in an agricultural environment. Indeed, the development of UGVs for

Figure 1.1: Photo taken from different angles of an UGV.

particular agricultural applications represents an expanding research field, also in the industrial field, to improve the efficiency of agricultural production [30].
For example, activities like fertilization-spreading tasks can be applied in an isolated manner and performed in an autonomous way if the appropriate tank is filled with fertilizer and attached to a fueled autonomous UGV; the same concept is applicable to planting and spraying [31]. In the case of soil sampling, an UGV can be adopted to collect consistent samples at uniform distances that can reveal the nutrient composition, and can indicate to farmers which fertilizers should be provided to reduce associated costs and to improve overall crop health. Then, other common applications of agricultural UGVs can be found in literature such as soil mapping, irrigation management, mechanical weeding, and harvesting [32]. As a consequence, the combination of all these automated activities can lead to a fully automated farm in which humans are relegated to mere supervisors.

There are a wide variety of UGVs in use today, and their features (dimensions, weight, speed) mainly depend on the type of application in which the vehicle is used.
The UGV considered in this thesis is shown in Figure 1.1 and it was developed at the Department of Agricultural, Forest and Food Sciences (DISAFA) of Università degli Studi di Torino. It is a four-wheel steering (4WS) electric UGV, i.e. it uses wheels to move in the environment and both front and rear ones can turn to control its direction. The movement is provided by the spin of electric motors mounted on the wheels and on the steering axes. The main features of the UGV, wheels and electric motors are reported in Table 1.1.
As previously stated, sensors are fundamental for the autonomy of each kind of vehicle; in the next section, more details about the sensors considered in this work and their main features are presented.

Table 1.1: List of UGV parameters.

Parameter	Description	Value
L	Distance between the center of front and rear wheels (length of the UGV)	1.5 $[m]$
T	Distance between the center line of each of the two wheels on the same axis (Width of the UGV)	1 $[m]$
a	Distance between the CoM the center of front wheels	0.75 $[m]$
b	Distance between the CoM the center of rear wheels	0.75 $[m]$
R	Wheel radius	254 $[mm]$
W	Wheel width	203 $[mm]$
w_{max}	Maximum angular velocity of the wheels motors	8.38 $[rad/s]$
$\dot{w}_{max}$	Maximum angular acc. of the wheels motors	1.40 $[rad/s^2]$
v_{max}	Maximum speed of the CoM of the vehicle	7.67 $[Km/h]$

1.2 Sensors

The use of sensors is fundamental in all robotics applications. Indeed, the collected data are necessary to properly close the feedback of the control actions and guarantee that the movements of the vehicle correctly correspond to the planned actions. Moreover, the capacity of a vehicle to move with a high degree of autonomy strongly depends on the elaboration of the information on its state and on the environment where it is placed. There exists two types of sensors:

- *Proprioceptive sensors* e.g. encoders (dead reckoning), inertial navigation system (INS), global positioning system (GPS), compass, gyroscopes. They measure values internal to the system, such as velocity,

accelerations and wheel rotational speed. Starting from this measurement, for example, the position of the UGV can be determined with the integral of the wheels speed (odometry).

- *Exteroceptive sensors* e.g. ultrasonic distance sensors, infrared sensors, laser sensors, force sensors. They measure values relative to the environment where the vehicle is moving.

In our case study, the 4WS-UGV is equipped with:

- 1 *Novatel OEM7600* receiver GPS
- 1 *XSens MTI-10 series* IMU composed by:
  - 1 Accelerometer;
  - 1 Gyroscope;
  - 1 Magnetometer;
- 4 *HC-SR04* ultrasonic sensors, two for each lateral side

The main features of the equipped sensors are reported in Table 1.2. The fusion of the information coming from these sensors allow us to obtain a more precise estimation of the UGV location and orientation. In particular, the use of the ultrasound sensors in combination with a detailed map of the environment, give a further measure on the position of the vehicle, fundamental to obtain an high precision localization.

Table 1.2: List of sensors features.

Sensor	Parameter	Value
Ultrasonic	Distance range	0.02–4.00 [m]
	Measure Angle	15 [°]
	Accuracy	3 [mm]
GPS	RTK accuracy	1 [cm] + 1 [ppm]
	SBAS accuracy	0.6 [m]
IMU	Position accuracy	1 [m]
	Velocity accuracy	0.05 [m/s]
	Accelerometer accuracy	200 [m/s^2]
	Gyroscope accuracy	450 [deg/s]

1.2.1 GPS

The Global Positioning System is a satellite-based radionavigation system. It is one of the global navigation satellite systems (GNSS) that provides *geolocation* and *time information* to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. GPS is composed by three segments:

1. *The space segment*: now consisting of 28 satellites, each in its own orbit about 20,000 km above the Earth.

2. *The user segment*: consisting of receivers of the signal coming from the satellites.

3. *The control segment*: consisting of ground stations (five of them, located around the world) that make sure the satellites are working properly.

In the field of autonomous vehicles navigation systems based on Global Positioning System, real time geographical data received from several GPS satellites are used to calculate longitude, latitude, speed and course to help navigate a vehicle. The advantage of using GPS is that the obtained data does not depend on previous readings and as a consequence errors in localization do not grow over the course of time. On the other hand, problems could appear in terms of accuracy and precision, which are dependent on the environment where the vehicle is placed and on the number of satellites it reads [33].
In precision agriculture, GPS-derived products help farmers to become more productive and efficient in their precision in-field activities and enhance farming operations. For example, location information is collected by GPS receivers for mapping field boundaries, roads, irrigation systems, and problem areas in crops such as weeds or disease. Moreover, GPS allows farmers to create farm maps, road locations and distances between points of interest and to accurately navigate to specific locations in the field to collect soil samples or monitor crop conditions [34]. Then, farming machines with GPS receivers on-board can be able to localize themselves within farm fields and adjust operation to maximize productivity and efficiency.
As stated before, the main disadvantage of this system is that obstacles such as mountains, buildings, trees, crop rows block the relatively weak GPS signals. Moreover, there are several additional sources of other error that can affect the accuracy of a GPS-derived position, like orbital errors, errors caused by atmospheric delays, multipath errors, errors caused by receiver

noise, and many others. Techniques and systems have been designed to improve the accuracy of the GPS that range from the integration of GPS user equipment with other navigation/positioning systems, such as inertial navigation units, to the enhancement of GPS through differential and carrier tracking techniques. In this case study, the UGV is equipped with a Novatel OEM7600 receiver GPS which provides a first measure of the positioning of the vehicle, with an accuracy of about 0.6m. Since the inter-rows space is, on average, only 2.5m and the goal is to obtain a centimeter accuracy for the localization of the UGV, the measure of the GPS needs to be improved with other sensors, also to avoid collision with the crops.

1.2.2 IMU

The IMU has been widely exploited to determine movements of a vehicle in terms of acceleration, angular velocity, and rotation [35]. In particular, there exists two IMU technologies:

- *The earlier technology*: it consists of two types of sensors, i.e. accelerometers and gyroscopes. The accelerometer is used to measure the inertial acceleration, while gyroscope measures the angular rotation.

- *The new technology*: it includes also a magnetometer to measure the bearing magnetic direction. Thus, it can improve the reading of gyroscope (see Figure 1.2).

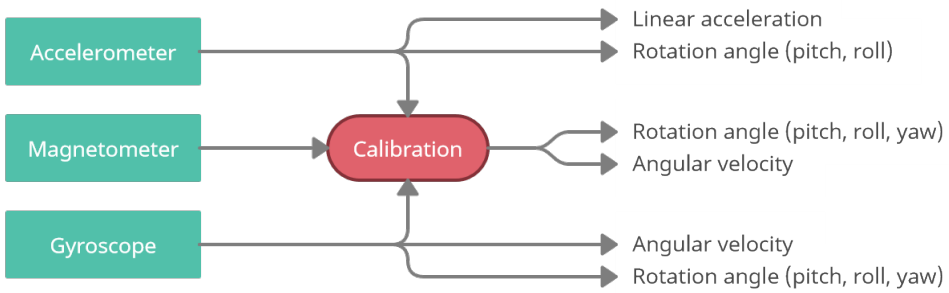


Figure 1.2: This type of IMU consists of accelerometer, gyroscope and magnetometer. The magnetometer is used to measure yaw angle rotation, thus it can be calibrated to the gyroscope data to improve the resulting data.

The IMU technology has expanded into various areas such as land and air vehicle navigation, robotics, medical devices. Nowadays, a lot of researches are about to upgrade to IMUs as an alternative (or a better choice) to the GNNS. Indeed, it is a cheaper and lighter device, which information can be combined with GPS data for better accuracy and it can be still used in locations with no GPS signals.
In precision agriculture, and in particular in the considered case study, IMU is used to improve the positioning provided by the GPS, compensating its common lack of signal within the crops. The selected model, *XSens MTI-10 series*, provides also important data on the orientation of the vehicle and about its speed and angular velocity.

1.2.3 Distance sensors

Active ranging sensors continue to be the most popular sensors in mobile robotics [36]. They are primarily used for object detection and collision avoidance, but they are also used for *localization*. While a *distance sensor* directly gives information on the distance between the vehicle and an object along a given direction, a *proximity sensors* can be classified as a simplified version of it, which identifies only the presence of an object nearby the sensor. The most used distance/proximity sensors are based on light or ultrasounds:

- *Light sensors*: They use a ray of light emitted by a source (infrared, laser), which is captured by a receiver (fototransistor) after that it has been reflected by an object. The intensity of the reflected light is proportional to the distance of the object. Due to the low accuracy of the measure of the distance and the limited working range, this kind of sensors are often used as proximity sensors.

- *Ultrasound sensors* (SONAR - SOund, NAvigation and Ranging): they use acoustic pulses and their echoes to measure the distance between objects. More precisely, given the speed of sound propagation c_s in a given medium (e.g. air, water), the distance d from an object is proportional to the time interval t_v needed by the acoustic wave to travel from the sensor to the object and coming back to the sensor. This time interval is called Time Of Flight (see Figure 1.3):

$$d = \frac{c_s t_v}{2} \tag{1.1}$$

  Ultrasonic pulses are emitted and received thanks to a transducer which relay back information about an object proximity [37].

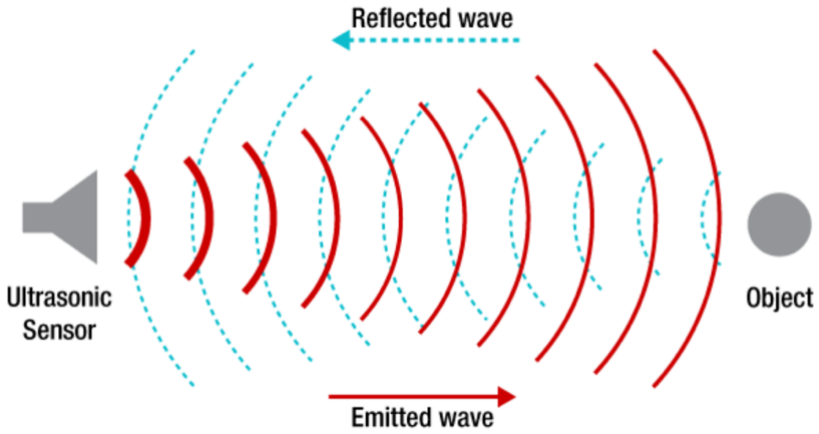


Figure 1.3: The sound wave is emitted by the sensor and reflected by the object on its path.

In the specific case of UAV, in most application they use ultrasonic sensors on the bottom of the drone for detecting ground and also for use in terrain-following. More recently, they have been also used for obstacle avoidance tasks as in [38, 39]. On the other hand, for UGVs, ultrasonic sensors have been introduced in some studies in combination with GPS, vision, and laser range scanner sensors to navigate the mobile robot in different environments, including crops as in [40].
This philosophy can be extended to the specific case study considered in this work, where the UGV can be equipped with ultrasonic sensors on the four angles of its body to detect lateral distance from the vehicle with respect to an obstacle, when constrained to operate in narrow environments. Each ultrasonic sensor provides the distance to the closest obstacle *perpendicularly* to the direction of forward motion of the vehicle itself.

1.3 Georeferenced low-complexity map

During in-field tasks, like planting, seeding or fertilization-spreading, an high level of accuracy is requested to aerial and ground vehicles. Hence, mapping, modelling and spatial describing the crops play a crucial role to fulfill this requirement. Thus, particular attention is given to provide an accurate knowledge of the the environment in which the mobile robots are going to be operated so that fully autonomous navigation and operations within complex, irregular and unstructured scenarios can be implemented. In this context, enhanced performance can be achieved by 3D path planning, which

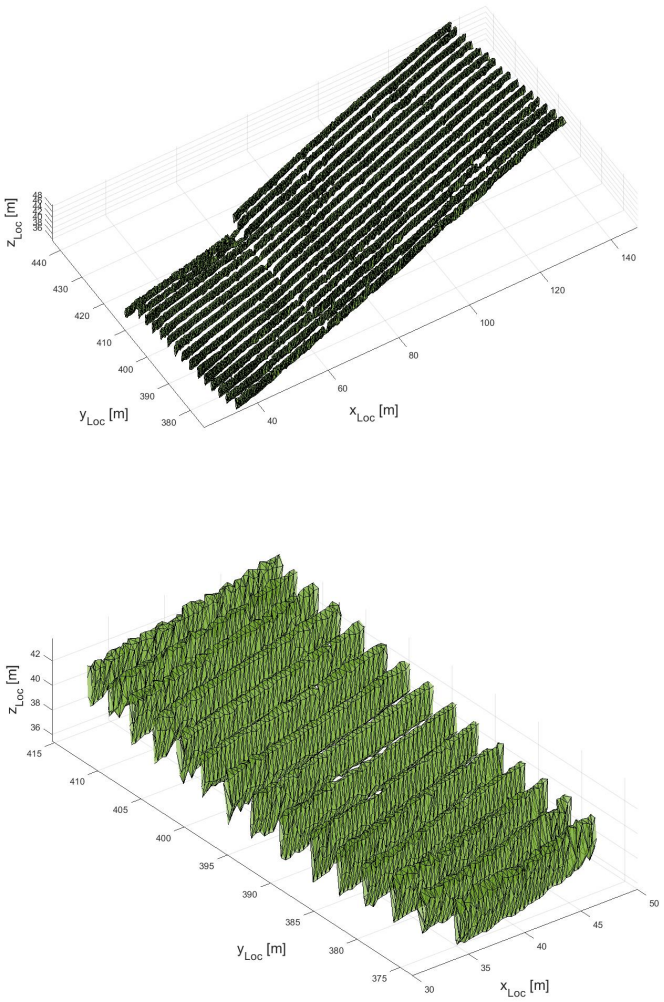
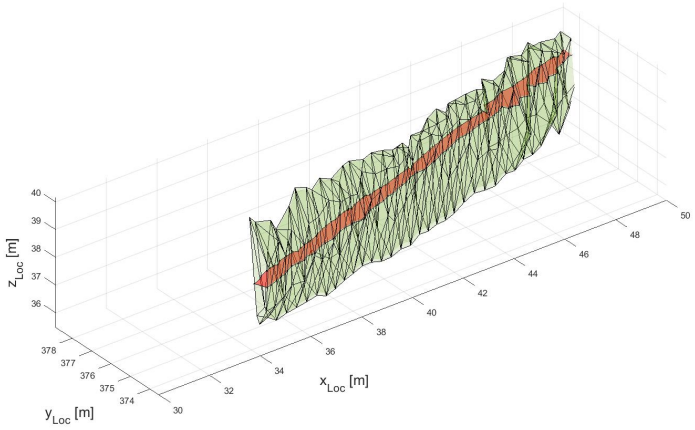


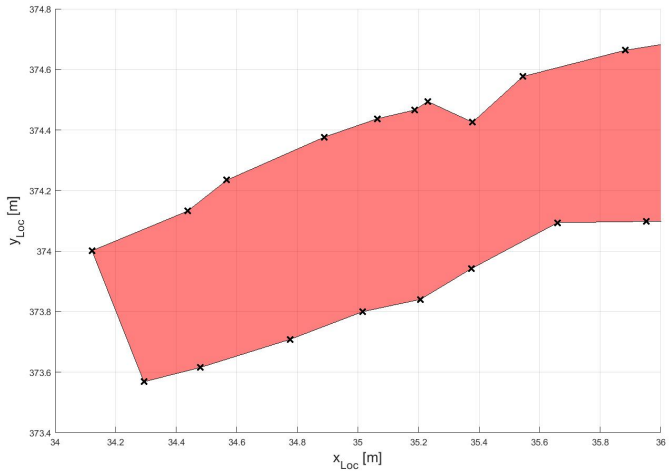
Figure 1.4: 3D overviews of the low-complexity mesh of a vineyard located in Barolo, Northwest of Italy.

exploit 3D models of the environment. These representations, which can be in the form of point clouds or triangulated mesh, can be generated exploiting different tools. All those methods share one main drawback: the generated datasets are typically vast (and heavy from a memory demand viewpoint) and require (multiple) post-processing steps for extracting valuable information. Moreover, when these datasets are intended to use in field, the information that can be gathered from classic on-board sensors, e.g. GPS, result not enough reliable to be fused with. For this reason, when considering agricultural scenarios involving drones or ground robots, unsupervised methods

to semantically interpret the models and to perform data reduction are key elements in the Agriculture 4.0 framework.



(a)



(b)

Figure 1.5: (a) 3D overview of one of the row mesh with the sliced plane (red area) identified at a fixed relative altitude from the terrain. (b) 2D projected slice of the 3D mesh.

To this aim, an innovative point cloud processing pipeline to automatically detect parcels and vine rows location was proposed in [13] and later exploited in [17] to generate low complexity 3D mesh models of vine rows. The output of the processing flow is made by a reduced set of elements still properly

describing the spatial layout and shape of vine, allowing a drastic reduction of the amount of data required without losing relevant crop shape information, as shown in Figure 1.4.

To exploit this map for in-field navigation, a reference relative altitude h_{ref} is selected from the terrain with respect to which extrapolate the profile of the row mesh of interest. First, 3D plane approximating the terrain is defined and then translated by h_{ref} to intersect the mesh and to obtain a 2D slice as shown in Figure 1.5a. The obtained 2D mesh (red area) is defined by a series of vertices (black crosses in Figure 1.5b), representing the intersecting points among plane and 3D mesh, and segments connecting them, each one characterized by a given orientation with respect to the global frame. This angle will be later used to derive the correlation among the possible UGV location within the ellipsoid $\mathcal{E}_k$, its orientation θ, and $d_\perp$, i.e. the distance from the vehicle to the map defined as

$$d_m = d_\perp + e_d, \tag{1.2}$$

where e_d is probabilistic measurement error, defined with its standard deviation σ.

Chapter 2

Navigation and sensor fusion

Any feedback controller, which aim is to track a given trajectory, requires the knowledge of the vehicle configuration in terms of position and attitude at each time instant to be effective. GPS provides information about the position of the UGV, but it is usually affected by large errors or, especially between the rows of a vineyard, it is not available. On the other hand, the IMU gives information about linear and angular velocity of the vehicle, but does not provide directly the position and the orientation of the UGV. For this reason, it is necessary to exploit specific tools for localization in order to estimate the position and the orientation of the UGV in real time.
The simplest solution, called *odometric localization*, uses only the information coming from the proprioceptive sensors and it is based on the integration of the kinematic model, given the velocity values provided by sensors. However, the quality of the results obtained by the iterated integration depends on the goodness of the initial configuration and it is subject to drift effects, i.e. increasing errors in time. Hence, it is not possible to only use the odometry to correctly locate the UGV within vine rows.
A more robust and reliable solution is represented by the so-called *active localization*, based on:

- the use of both odometric measures (IMU) and data coming from exteroceptive sensors (ultrasound sensors);

- the comparison with a map of the environment where the robot is moving, generated either a prior, or during the motion.

In this case, the estimation provided by the odometric localization is corrected using the information coming from the other sensors via *probabilistic filters*, where the objective is not only the estimation of the robot's configuration, but also the quantification of the uncertainty related to such estimation, modeled with a "belief" function (Figure 2.1).

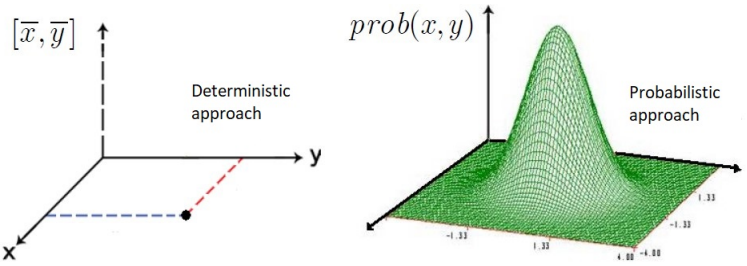


Figure 2.1: On the left, in the deterministic approach each estimation represents a possible pose of the vehicle. On the right, the estimation is associated with a probability function which represents the uncertainty.

The most common approaches for estimation of ground robot pose are based on:

- *Kalman filter (KF)*: based on Guassian probability density functions, it is a powerful filter which supports estimations of past, present, and future states [41];

- *Particle filter*: the probability densities are approximated with a set of weighted particles: higher is the weight of a particle, higher is the probability that it represents the effective pose of the robot [42].

While Kalman filter can be used for linear or linearized processes and measurement system with uncertainty modeled as Gaussian distribution, the particle filter can be used for nonlinear systems and can deal with non-Gaussian noise, so it can be more widely applicable. On the other hand, the particle filter performance are, in general, smoother and more robust, at the expanse of longer computation times than Kalman filters [43]. For this reason, since several additional operations are involved in the filter proposed in this work, Kalman filter is employed to preserve rapid communication and data exchange between sensors during the sensors fusion process. More details are given in the next section.

2.1 Kalman filter

Kalman filter was proposed by R. E. Kalman in 1960 [44] and it started to attract much attention with the increasing demands of target tracking applications. Indeed, several Kalman filter schemes with different features have been proposed in the last 30 years and, today, it represents the standard approach for optimal estimation. KF is an algorithm which aim is to estimate unknown variables with more accuracy, using a series of data collected over the time, tacking into account noise and other inaccuracies. More precisely, it optimally estimates the variables of interest when they cannot be measured directly but an indirect measurement is available, or it finds the best estimate of state variables by combining measurements from various sensors in the presence of noise. KF is widely applied in the fields of orbit calculation, target tracking and navigation [45, 46, 47]. Furthermore, it also plays an important role in dynamic positioning and sensor data fusion [48].
In precision agriculture, Kalman filter is largely used to collect and fuse measurements coming from different sensors available on field. Agricultural parameters, like soil moisture and temperature, can be measured and predicted so that the noise associated with noisy measurements is filtered [49]. Otherwise, in the case of aerial or ground robots performing in-field operations, KF is useful to improve the positioning accuracy of the current GPS receivers, that in agriculture scenarios are characterized by common loss of signal. This can be done by integrating the GPS measurement with additional information coming from other sensors in order to provide better position estimates. For example, in [50] the Kalman filter is implemented to reduce the quantization errors in the positioning of tractors equipped with some low-cost GPS receivers. In [51], extended Kalman filter is used to accomplish sensor fusion for an agricultural vehicle that travels down crop rows. In this case, it uses a sensor package that includes machine vision, odometers, accelerometers and a compass.

Regarding the implementation of the Kalman filter, let us consider a discrete-time linear system[1]:

$$\begin{aligned} x_{k+1} &= \mathbf{F_k} x_k + \mathbf{G_k} u_k + d_k \\ y(x_k) &= \mathbf{H_k} x_k + d_k^y \end{aligned} \tag{2.1}$$

[1]Usually, the system is discretized in order to get a more suitable representation for an online implementation.

where x_k is the state variable at time k, u_k the control input at time k, $\mathbf{F_k}$ is the state transition matrix, $\mathbf{G_k}$ is the control-input matrix, d_k is the process noise at time k, y_k is the sensor measurement (observation) at time k, $\mathbf{H_k}$ is the observation matrix and d_k^y is the measurement noise at time k.
Then, suppose that measurements of x_k, d_k and d_k^y are not available, while y_k and u_k are measured and known. The goal of the filter is to retrieve accurate estimation $\hat{x}_k$ of x_k, from current and past measurements of y_k and u_k.
Methods based on Kalman Filter works in two different phases:

1. *Prediction*: it provides an a-priori estimation given the command input and the model of the system (odometry), i.e.

$$x_k^p = \mathbf{F_k}\hat{x}_{k-1} + \mathbf{G_k}u_{k-1} \tag{2.2}$$

   where x_k^p is a prediction of x_k, computed at step k and $\hat{x}_{k-1}$ is the estimation of x_{k-1}.

2. *Update* or *Innovation*: it includes a-posteriori information coming from sensors (output measurements) to improve the state estimate as:

$$\hat{x_k} = x_k^p + \mathbf{K_k}\Delta y_k \tag{2.3}$$

   where $\Delta y_k = y_k - \mathbf{H_k}x_k^p$ and $\mathbf{K_k}$, the gain matrix, is chosen to minimize the variance of the estimation error norm, given by $E\left[||x_k - \hat{x_k}||_2^2\right]$

In order to present the KF algorithm, the following quantities are defined:

- $\hat{x_k}$: estimation of x_k, computed at time step k;
- $\mathbf{P_k} = E\left[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T\right]$: covariance matrix of the estimation error $(x_k - \hat{x_k})$;
- $\mathbf{Q^d} = E\left[d_k d_k^T\right]$: covariance matrix of d_k;
- $\mathbf{R^d} = E\left[d_k^y (d_k^y)^T\right]$: covariance matrix of d_k^y.

Preliminarly, matrices $\mathbf{Q^d}$ and $\mathbf{R^d}$ are defined from the available information on d_k and d_k^y. In alternative, in the case of poor available information on the disturbances, a trial-and-error procedure can be exploited. These matrices are typically chosen as diagonal, with the variances of d_k and d_k^y on the main diagonal. Once defined the covariance matrices, it is necessary to define also the following initial conditions, typically initialized as

$$\hat{x}_0 = \mathbf{0} \tag{2.4a}$$

$$\mathbf{P_0} = \mathbb{I} \tag{2.4b}$$

where $\hat{x}_0$ is the initial state estimation, $\mathbf{P_0}$ the estimated initial covariance matrix and $\mathbb{I}$ is the identity matrix.
When all these quantities have been designed, the iterative algorithm can start.
In the *prediction step*, as explained above, the filter uses the model to obtain a preliminary estimate of the state as follows:

$$x_k^p = \mathbf{F_k}\hat{x}_{k-1} + \mathbf{G_k}u_{k-1} \tag{2.5a}$$
$$\mathbf{P_k^p} = \mathbf{F_{k-1}}\mathbf{P_{k-1}}\mathbf{F_{k-1}^T} + \mathbf{Q^d} \tag{2.5b}$$

Then, in the *update step*, the filter corrects the preliminary estimate according to the current output measurement as:

$$\mathbf{S_k} = \mathbf{H_k}\mathbf{P_k^p}\mathbf{H_k^T} + \mathbf{R_d} \tag{2.6a}$$
$$\mathbf{K_k} = \mathbf{P_k^p}\mathbf{H_k^T}\mathbf{S_k^{-1}} \tag{2.6b}$$

$$\hat{x}_k = x_k^p + \mathbf{K_k}\Delta y_k \tag{2.7a}$$
$$\mathbf{P_k} = (\mathbf{I} - \mathbf{K_k}\mathbf{H_k})\,\mathbf{P_k^p} \tag{2.7b}$$

This procedure allows to increase the accuracy of the estimation, to enhance the filter stability properties, and to filter the effects of disturbances and noises.
In conclusion, the KF and its variants are very good estimator for linear, nonlinear and linearized systems: their main advantages are their simplicity and the vast kind of applications where they can be applied. On the other hand, the main drawback of Kalman filters is that their performance strongly depends on the knowledge of the system model and the presence of uncertainties can seriously affect the results.

In the considered case, the Kalman filter is implemented for the navigation of the UGV: the vehicle states, position, attitude and velocities are estimated by fusing sensor outputs from an inertial measurement unit, a GPS receiver, and the elaboration of distances from the crop rows provided by four ultrasound sensors mounted on the vehicle.

2.2 The navigation framework

The filter proposed in this thesis, based on the Kalman theory explained in the previous section, makes use of several additional elements and procedures

involved in the *navigation* process of the UGV. The term "*navigation*" refers to a part of the more general concept of *guidance, navigation, and control* (GNC). It consists in the determination, at a given time, of the vehicle's location, velocity and attitude (instantaneous state of the vehicle). Using this information, the guidance system generates the optimum trajectory to track the target and the desired vehicle steering command to realize the optimum trajectory in real time. Finally, the vehicle control system receives reference values from the guidance system to generate the control commands to follow the desired trajectory.
In this section, all the available information elaborated by the navigation algorithm are presented and analyzed. Then, the process that allows to retrieve further location data from the distance measure is described and later used to build the overall filtering schema, as explained in Chapter 4.

Current prediction

At each time instant k, the Kalman filter provides an initial prediction of the states in terms of estimation (x_k^p) and covariance matrix of its error ($\mathbf{P_k^p}$), according to the defined model of the system (2.5).
The covariance matrix defines both the spread (variance) and the orientation (covariance) of our data. Therefore, it can be represented by a vector and its magnitude, where the vector points into the direction of the largest spread of the data, and its magnitude equals the spread (variance) in this direction. In particular, it can be seen that the largest eigenvector of the covariance matrix always points towards the direction of the largest variance of the data, and the magnitude of this vector equals the corresponding eigenvalue. Then, the second largest eigenvector is always orthogonal to the largest one, and points into the direction of the second largest spread of the data (and so on, with same considerations for other dimensions) [52]. It follows that the uncertainty of the vehicle's states can be modeled with a confidence ellipsoid (see Figure 2.2) written in the following form

$$\mathcal{E}_k = \{x \in \mathbb{R}^n | (x - x_k^p)^T \mathbf{P_k^p}^{-1} (x - x_k^p) \leq 1\} \tag{2.8}$$

where x_k^p is the center of $\mathcal{E}_k$ and the vector of the current estimation of the state variables at time k, while $\mathbf{P_k^p}$ is a positive semi-definite matrix which defines the shape of the ellipsoid $\mathcal{E}_k$ and represents the covariance matrix relative to the uncertainty on the prediction. More precisely, the eigenvectors of $\mathbf{P_k^p}$ define the principal axes of the ellipsoid (directions of the largest variances of the data) and its eigenvalues are the reciprocals of the squares

of the semi-axes length (variances of the data).

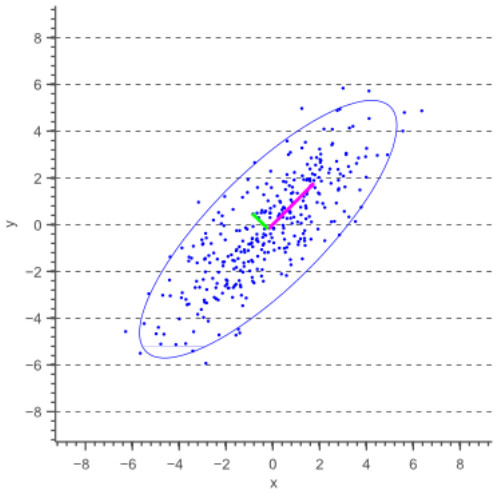


Figure 2.2: 2D ellipse representation for normally distributed data.

This preliminary estimation and its associated ellipsoid are two required inputs for the application of the ellipsoid method. Indeed, new information about the UGV position are retrieved, merging them with distance measurements and crop rows data, provided by the digital map.

Crop rows in the map

The map of the environment provides essential information about the rows and how they can be modeled in the space (see Figure 2.3). Each row_i is composed by N segments and it can be modeled as

$$row_i = \begin{bmatrix} a_1x + b_1y + c_1 = 0 \\ a_2x + b_2y + c_2 = 0 \\ \vdots \\ a_Nx + b_Ny + c_N = 0 \end{bmatrix} \tag{2.9}$$

The i-th segment is defined by the equation of a straight line as

$$a_ix + b_iy + c_i = 0 \tag{2.10a}$$
$$y = m_ix + q_i \tag{2.10b}$$

where (2.10b) is obtained from (2.10a) with $m_i = -\frac{a_i}{b_i}$ and $q_i = -\frac{c_i}{b_i}$.
Starting from (2.10b), the angle between a segment and the x-axis of the

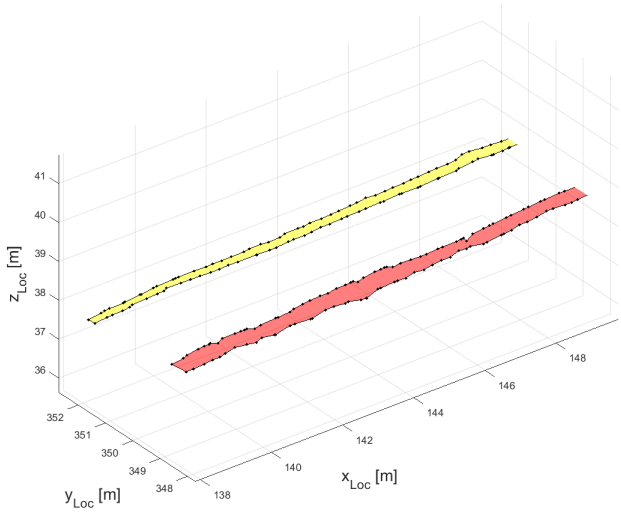


Figure 2.3: An example of two rows of the considered field modeled with a set of segments.

global reference system (Figure 2.4) is obtained as:

$$\beta = \arctan(m) \tag{2.11}$$

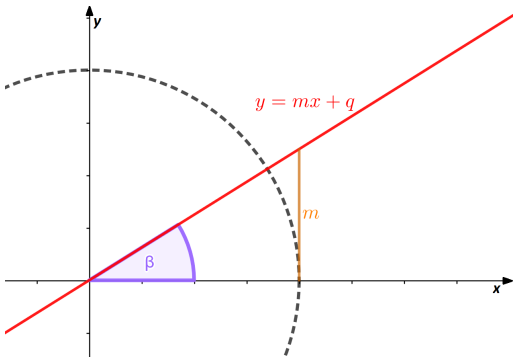


Figure 2.4: Angle between a segment and the x-axis obtained with the arctan function.

Particular attention is given to this angle, fundamental in the design process to derive some important relations between different kind of distances from the vehicle to the row.

Distance measurements from sensors

In addition to GPS, which provides position, and IMU, which provides orientation, velocities and position (indirectly), each ultrasound sensors mounted on the UGV measures the perpendicular distance between its position and the row. Hence, if the center of mass (CoM) of the UGV has a given orientation θ_0 with respect to local frame at time k, the distance $d_\perp$ measured from the UGV by the sensor, which is supposedly mounted on its side, with respect to an obstacle has an orientation of $\theta' = \theta_0 - \frac{\pi}{2}$ (Figure 2.5).

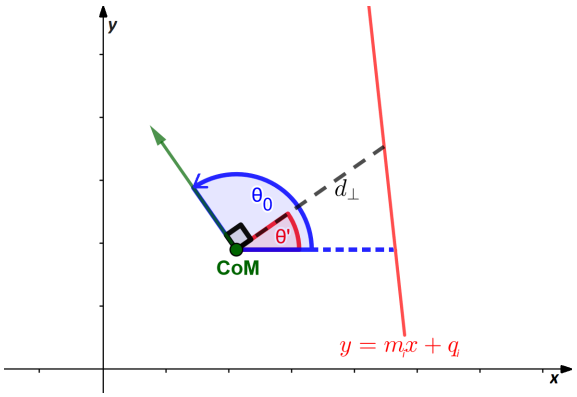


Figure 2.5: Center of mass orientation θ_0 (blue angle) and measured perpendicular distance orientation θ' (red angle) with respect to the x-axis.

The measured value can be defined as the sum of the perpendicular distance $d_\perp$ and a measurement error e_d, i.e.

$$d_m = d_\perp + e_d \tag{2.12}$$

where e_d is a probabilistic measurement error defined by its standard deviation σ and it is given by the dispersion of the measurement error of the selected sensor around its mean.
Starting from this data and considering the measurement error, the initial estimated position of the UGV is improved by modeling a new probability ellipsoid, which represents the set of feasible points $\mathcal{F}$, i.e. the points which distance from the row is consistent with the measured one. For each point, its probability on the base of the error (standard deviation σ) of the distance sensor is defined, obtaining an additional measure of the position of the UGV.

2.3 Position values from distance measures

During the in-field navigation, at each time instant and starting from the initial ellipsoid $\mathcal{E}_k$ (2.8), the uncertainty on the prediction is reduced by providing some additional information on the position of the UGV, obtained by an elaboration of the distance measures received by the ultrasound sensors. Hence, this supplementary indirect measures of the location can be considered as further output in the update phase of the Kalman filter algorithm. Let's define first the above-mentioned feasible points subset $\mathcal{F} \subseteq \mathcal{E}_k$ as

$$\mathcal{F} = \left\{ x \in \mathcal{E}_k | d_m^{min} \leq d_\perp(x, \theta) \leq d_m^{max} \right\} \tag{2.13}$$

where $d_\perp(x, \theta)$ is the perpendicular distance computed on the map from the generic point $x_i = (xi, yi) \in \mathcal{E}_k$ with orientation θ with respect to the x-axis, and d_m^{min}, d_m^{max} are the minimum and maximum 1-σ values of the measurement d_m, respectively, taking into account the standard deviation of its error e_d.
The subset $\mathcal{F}$ depends primarily on the standard deviation σ of the error e_d of the measured distance: higher is the uncertainty of the sensor measurement, higher is the number of points whose distance from the row is consistent with the measured one, accounting its possible errors. Since all the points which have the same distance with respect to the i-th segment (2.10) lay on the same parallel line, $\mathcal{F}$ will be a portion of $\mathcal{E}_k$, bounded within two straight lines parallel to the considered row (see Figure 2.6). These two lines contain the points at distances d_m^{min} and d_m^{max} from the considered segment.
The goal is to find a new ellipsoid $\mathcal{E}_{k+1}$ that models the feasible points set $\mathcal{F}$ and that represents an additional measurement of the position of the UGV with its relative uncertainty (defined by x_{k+1} and P_{k+1}). In particular, it can be obtained exploiting the initial ellipsoid $\mathcal{E}_k$ and the two lines that bound the set $\mathcal{F}$, by applying the *ellipsoid method*.
More details on the definition of the function $d_\perp(x, \theta)$ (2.13) are given in the next section. This function allows to have a representation of the sensed distance on the map of the environment and it is the first step to compute the boundaries of $\mathcal{F}$.

2.3.1 Perpendicular and minimum distance

Exploiting the relation between minimum distance and perpendicular distance, the values measured by distance sensors and the information provided

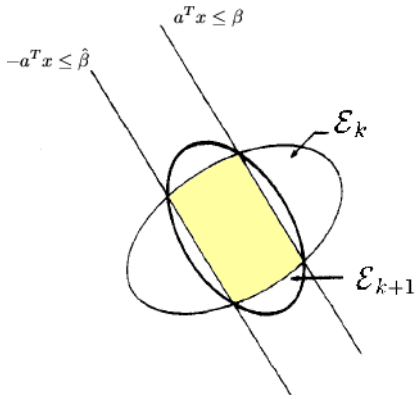


Figure 2.6: A representation of the feasible points set (yellow area) $\mathcal{F}$.

by the digital map can be used together to obtain the bounds of the set $\mathcal{F}$. While the minimum distance is the segment at minimum length which connects the CoM of the UGV to the line with an angle of incidence of 90°, the perpendicular distance, as defined before, has a variable angle that depends on the orientation of the UGV (Figure 2.5). Minimum distance could be easily obtained knowing the point $x_k = (x, y)$ and (2.10a) as

$$d_{min}(x_k) = \frac{|ax + by + c|}{\sqrt{a^2 + b^2}} \tag{2.14}$$

Then, as shown in Figure 2.7, it is possible to evaluate $d_\perp$ from $d_{min}(x_k)$ applying trigonometric relations as follows

$$d_\perp(x_k) = \frac{d_{min}(x_k)}{cos(\gamma)} \tag{2.15}$$

where γ is the angle between $d_{min}(x_k)$ and $d_\perp(x_k)$. By referring to Figure 2.7, this angle can be obtained as

$$\gamma = \epsilon + \theta' = \epsilon + \theta_0 - \frac{\pi}{2} \tag{2.16}$$

where ϵ is derived by considering the triangle composed by d_{min}, the segment parallel to the x-axis and the row (Figure 2.7). Indeed, the sum of its internal angles is π and two of them are known: $\frac{\pi}{2}$ is by definition the incidence of d_{min} on the line and β, that is obtained with (2.11) (or $\pi - \beta$, depending on the case). However, depending on the slope of the row and on the configuration of the angles, six different cases can occur, but it can be demonstrated that

for all of them the result is the same. For simplicity, only the two main cases are reported in the following definiton of ϵ

$$\begin{aligned} \epsilon &= \pi - \frac{\pi}{2} - \beta \\ &= \frac{\pi}{2} - \beta \end{aligned} \qquad \begin{aligned} \epsilon &= \pi - \frac{\pi}{2} - (\pi - \beta) \\ &= \beta - \frac{\pi}{2} \end{aligned} \tag{2.17}$$

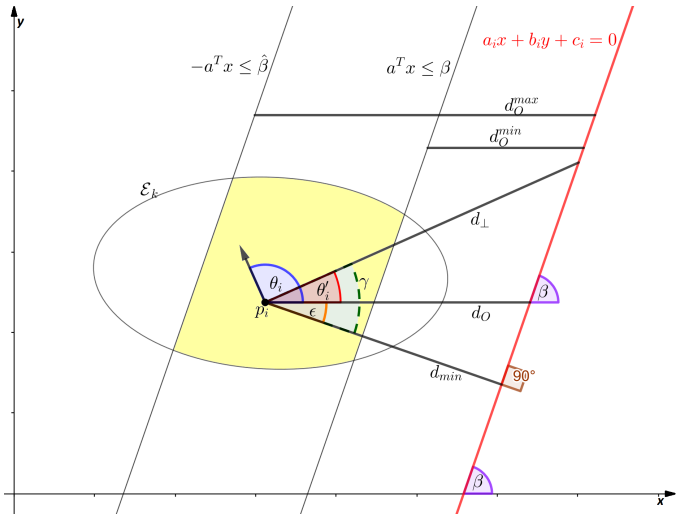


Figure 2.7: The considered triangle used to compute all the angles of interest. In the alternative configuration, the slope of the row is such that the d_{min} segment shift above the d_m segment making the considered angle of the triangle equals to $\pi - \beta$. Other configurations of the segments are possible, for all of them (2.19) holds.

Hence, γ (2.16) can be defined in two different ways: as the sum of ϵ and θ' in the first case of (2.17) (represented in Figure 2.7), and as their difference in the second one, i.e.

$$\gamma = \frac{\pi}{2} - \beta + \theta_0 - \frac{\pi}{2} = \theta_0 - \beta \tag{2.18a}$$

$$\gamma = \beta - \frac{\pi}{2} - (\theta_0 - \frac{\pi}{2}) = -\theta_0 + \beta \tag{2.18b}$$

Since in (2.15) only the cosine of the angle γ is considered and the two cases in (2.18) differs only for a minus sign, we could consider only one of them to represent both cases ($\cos(\alpha) = \cos(-\alpha)$). Notice that the same

considerations can be done for all the other possible configurations. Finally, considering (2.14), (2.15) and (2.18), $d_\perp(x, \theta)$ can be defined as

$$d_\perp(x_k, \theta) = \frac{|ax + by + c|}{\sqrt{a^2 + b^2}\cos(\theta_0 - \beta)} \tag{2.19}$$

Bounds on the set $\mathcal{F}$

As mentioned before, to correctly approximate $\mathcal{F}$ with the minimum volume ellipsoid $\mathcal{E}_{k+1}$, the information on the two boundaries of the set is necessary. Since the two bounds can be modeled as two lines, parallel to the considered row, they can be written as follows

$$l1 \equiv m_i(x + d_O^{min}) + q_i; \tag{2.20a}$$

$$l2 \equiv m_i(x + d_O^{max}) + q_i; \tag{2.20b}$$

where d_O^{min} and d_O^{max} are the two offsets to apply to the equation of the selected segment of the row (2.10) along the x axis. Hence, the problem is now reduced to find this two offsets. Let's consider a generic offset O of a generic point x_O from the line, the right-angled triangle (see Figure 2.7) obtained from the offset (segment parallel to the x-axis), the minimum distance and the row. The angle between the offset and the minimum distance is ϵ (2.17) and the relation between the two segments is defined by

$$O = \frac{d_{min}}{cos(\epsilon)} \tag{2.21}$$

In this case, the minimum distance cannot be obtained with (2.14), since the purpose of this process is to obtain an analytic relation between the offsets and the measured distance, without knowing the points of the set $\mathcal{E}_k$. Considering d_{min} (2.15) and (2.18), an alternative way to compute the minimum distance is easily obtained as:

$$d_{min} = d_m \cos(\gamma) \tag{2.22}$$

where, in this case, d_m is the perpendicular distance measured by the sensor, defined in (2.12). Knowing the measured distance and its error (2.12), the row's incidence angle with the x-axis (2.11), and the UGV orientation (provided by the IMU), d_O^{min} and d_O^{max} are calculated as

$$d_O^{min} = \frac{d_m^{min}\cos(\theta_0 - \beta)}{\cos(\beta - \frac{\pi}{2})} \tag{2.23a}$$

$$d_O^{max} = \frac{d_m^{max}\cos(\theta_0 - \beta)}{\cos(\beta - 90°)} \tag{2.23b}$$

and the two bounds (2.20) representing the feasible points set $\mathcal{F}$ are obtained. These two bounds of the set $\mathcal{F}$ could be seen as two half-planes defining the boundaries of the feasible points area in the starting ellipsoid $\mathcal{E}_k$. Then, this information is used in the *ellipsoid method* to perform the minimum volume ellipsoid which approximates $\mathcal{F}$ and represents the probabilistic uncertainty for the new measurement of the position, as described in the next chapter.

Chapter 3

Ellipsoid method

In the literature, the ellipsoid method is an optimization procedure used to minimize convex functions. In particular, it allows to generate a sequence of ellipsoids whose volume uniformly decreases at every step, thus enclosing a minimizer of a convex function.
In this chapter, its main formulations will be described: the *classic* one, the *deep cut* variant and the *parallel cuts* variant. This last one will be later used to process the measured distance that, as seen in the previous chapter, gives rise to a parallel pair of constraints (half-spaces) on the initial prediction. Differently from the literature, in this case the ellipsoid method is used with a different purpose: the resulting ellipsoid, $\mathcal{E}_{k+1}$, will be the ellipsoid at minimum volume, which provides an additional measurement of the position of the UGV (its center) and an estimation of its uncertainty, to be used in the update phase of the Kalman filter.

3.1 Classic Method

The classic formulation is used to estimate the new ellipsoid of minimum volume $\mathcal{E}_{k+1}$, which contains all the points in $\mathcal{E}_k$ that also belong to the half-space passing through its center. The ellipsoid $\mathcal{E}_k$ is defined as in (2.8), while the half-space is represented by

$$a^T(x - x_k) \leq 0 \tag{3.1}$$

The new ellipsoid $\mathcal{E}_{k+1}$, is obtained by computing $\mathbf{P_{k+1}}$ and x_{k+1} as follows:

$$x_{k+1} = x_k - \tau \frac{\mathbf{P_k} a}{\sqrt{a^T \mathbf{P_k} a}} \tag{3.2}$$

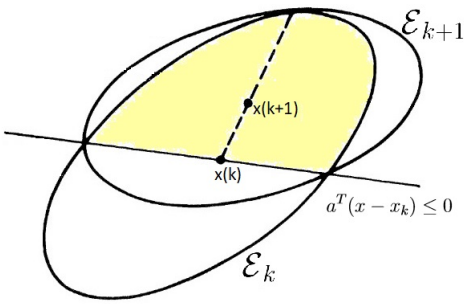


Figure 3.1: The ellipse $\mathcal{E}_k$ with center in $x(k)$, cut by the half-plane passing through its center. $\mathcal{E}_{k+1}$ is the new ellipse and $x(k+1)$ the new center.

$$\mathbf{P_{k+1}} = \delta(\mathbf{P_k} - \sigma\frac{\mathbf{P_k}a(\mathbf{P_k}a)^T}{\sqrt{a^T\mathbf{P_k}a}}) \tag{3.3}$$

where

$$\tau = \frac{1}{n+1}, \sigma = \frac{2}{n+1}, \delta = \frac{n^2}{n^2-1} \tag{3.4}$$

are respectively the step, dilation and expansion parameters, and n is the space dimension.
The obtained $\mathcal{E}_{k+1}$ is the ellipsoid of smallest volume that contains the half-ellipsoid $\mathcal{E}_k$ divided by the half-plane (3.1). The drawback of this formulation is the constraint on the half-plane: it is required that it passes through the center of the ellipsoid $\mathcal{E}_k$, as in Figure 3.1. A first generalization of this approach, presented in the next section, is made by the use of a free "deep" cut instead of the previous centered cut.

3.2 Deep Cut Method

In the deep cut variant, the constraint which forces the half-plane to pass through the center of the ellipsoid is relaxed and (3.1) becomes

$$a^T x \leq \beta \tag{3.5}$$

Notice the difference between (3.1) and (3.5), where the constant term of the half-plane, first imposed to be $a^T x_k$, becomes the generic constant β.
Also in this case, $\mathbf{P_{k+1}}$ and x_{k+1} of the new ellipse $\mathcal{E}_{k+1}$ can be computed

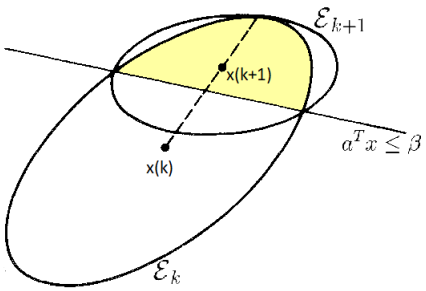


Figure 3.2: In the deep cut variant the half-plane does not pass through the center of the ellipse $\mathcal{E}_k$, which is divided in two different parts.

using (3.2) and (3.3), with the parameters τ, σ and δ chosen as:

$$\tau = \frac{1 + n\alpha}{n + 1} \tag{3.6a}$$

$$\sigma = \frac{2(1 + n\alpha)}{(n + 1)(1 + \alpha)} \tag{3.6b}$$

$$\delta = \frac{n^2(1 - \alpha^2)}{n^2 - 1} \tag{3.6c}$$

The new quantity α represents the algebraic distance of x_k from the half-plane in the metric corresponding to the matrix $\mathbf{P_k}$ and can be computed with:

$$\alpha = \frac{a^T x_k - \beta}{\sqrt{(a^T \mathbf{P_k} a)}} \tag{3.7}$$

3.3 Parallel Cuts Method

Finally, the parallel cut variant allows to use simultaneously the constraints imposed by a pair of of parallel cuts and to generate the new ellipsoid $\mathcal{E}_{k+1}$ having minimum volume and containing all the points between the two half-spaces (Figure 3.3).
More precisely, the two bounds obtained with the measured distance and its uncertainty ($l1 = m(x + d_O^{min}) + q$ and $l2 = m(x + d_O^{max}) + q$) (2.20) can be redefined as two half-spaces by

$$a^T x \leq \beta \wedge -a^T x \leq \hat{\beta} \tag{3.8}$$

where $a^T = \begin{bmatrix} -m & 1 \end{bmatrix}$, $\beta = m d_O^{min} + q$, and $\hat{\beta} = m d_O^{max} + q$.
Then, the two parameters α and $\hat{\alpha}$, representing the algebraic distance of

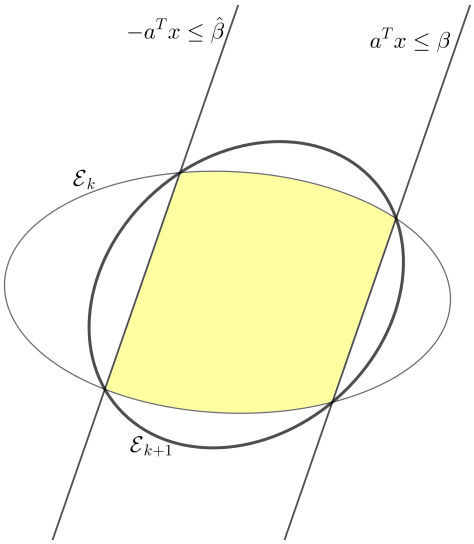


Figure 3.3: Two half-plane are considered simultaneously and the minimum volume ellipsoid $\mathcal{E}_{k+1}$ containing the area constrained among the ellipsoid $\mathcal{E}_k$ and the two half-planes (yellow area) is obtained.

each half-spaces from the center of the ellipse, can be computed as

$$\alpha = \frac{a^T x_k - \beta}{\sqrt{(a^T \mathbf{P_k} a)}} \tag{3.9a}$$

$$\hat{\alpha} = \frac{\hat{\beta} - a^T x_k}{\sqrt{(a^T \mathbf{P_k} a)}} \tag{3.9b}$$

3.3.1 Preliminary checks

It is important to highlight that α and $\hat{\alpha}$ need to be checked because some special cases may occur:

1. The quantity α ($\hat{\alpha}$) represents the algebraic distance of the center of $\mathcal{E}_k$ from the corresponding half-space (3.8) in the metric corresponding to the matrix $\mathbf{P}$.
   This means that if $|\alpha|$ ($|\hat{\alpha}|$) is greater than 1, the half-space is outside of the starting ellipsoid and the generated $\mathcal{E}_{k+1}$ will contain points which are already excluded. For this reason, this case can be simply handled by imposing $\alpha = sign(\alpha)$ ($\hat{\alpha} = sign(\hat{\alpha})$).

$$|\alpha| > 1 \implies \alpha = sign(\alpha)$$

This operation is equivalent to ignore the constraints outside of the area, and to consider only the other one, performing the deep cut variant previously described.

2. Then, if the product between α and $\hat{\alpha}$ is greater or equal then $\frac{1}{n}$ (after the previous check), the smallest ellipsoid containing the portion of $\mathcal{E}_k$ is $\mathcal{E}_k$ itself, i.e.

$$\alpha\hat{\alpha} \geq \frac{1}{n} \implies \mathcal{E}_{k+1} = \mathcal{E}_k$$

3.3.2 Parameters

After the above-mentioned checks on α and $\hat{\alpha}$, the following parameters can be computed:

$$\sigma = \frac{1}{n+1}\Big[n + \frac{2}{(\alpha - \hat{\alpha})^2}(1 - \alpha\hat{\alpha} - \frac{\rho}{2})\Big] \tag{3.10a}$$

$$\tau = \frac{\alpha - \hat{\alpha}}{2}\sigma \tag{3.10b}$$

$$\delta = \frac{n^2}{n^2 - 1}\Big(1 - \frac{\alpha^2 + \hat{\alpha}^2 - \frac{\rho}{n}}{2}\Big) \tag{3.10c}$$

$$\rho = \sqrt{4(1 - \alpha^2)(1 - \hat{\alpha}^2) + n^2(\hat{\alpha}^2 - \alpha^2)^2} \tag{3.10d}$$

and x_{k+1} and $\mathbf{P_{k+1}}$ can be obtained as

$$x_{k+1} = x_k - \tau\frac{\mathbf{P_k}a}{\sqrt{a^T\mathbf{P_k}a}} \tag{3.11}$$

$$\mathbf{P_{k+1}} = \delta(\mathbf{P_k} - \sigma\frac{\mathbf{P_k}a(\mathbf{P_k}a)^T}{\sqrt{a^T\mathbf{P_k}a}}) \tag{3.12}$$

where x_{k+1}, the center of the new ellipsoid, is the updated measurement based on the sensed distance from the row and $\mathbf{P_{k+1}}$ is the positive semi-definite matrix which defines the shape of the new ellipsoid, and represents its probabilistic uncertainty.
This last approach has been selected to propagate the uncertainty ellipsoid when the information gathered from the ultrasound sensor and the map are used to generate the parallel cuts. Moreover, using the new quantity x_{k+1} and the positive semi-definite matrix $\mathbf{P_{k+1}}$, this procedure can be iterated in the navigation algorithm at each time instant to improve the estimation of the UGV position by fusing these results with data coming from other sensors (GPS, IMU) exploiting a classic Kalman filter approach.

Chapter 4

Filter

In this chapter, the methodologies previously introduced are collected together to build and simulate a filter that, for each time instant k, provides a precise estimation on the state variables of the system and allows the UGV to move following the desired trajectory between the vine rows. Indeed, the ellipsoidal approach, described in the previous chapter, allows to exploit the information coming from the 2D map to construct an ellipsodal estimation of the UGV position. As described in this chapter, this is then exploited in an ad-hoc designed Kalman-like filter to fuse this information with IMU and GPS signals. In particular, details on the model used to describe the UGV dynamics are provided, together with the method selected for propagating the position estimation ellipsoid according to the merged information gathered from the ultrasonic sensors and the simplified map.

4.1 System model

In this work, a simplified model for the UGV is considered, described by three double integrators as:

$$\ddot{x} = \frac{F_x}{m}, \quad \ddot{y} = \frac{F_y}{m}, \quad \ddot{\theta} = \frac{M}{I_z}, \tag{4.1}$$

where (x, y, θ) represents the position coordinates and attitude of the UGV in the local frame, respectively, m and I_z are the UGV mass and moment of inertia. (F_x, F_y) are the control force components and M the control torque with respect to the vehicle z axis. Values associated to these parameters are reported in Table (4.1).

Table 4.1: List of system features.

Parameter	Description	Value
m	Mass of the vehicle	10 [kg]
I_z	Moment of inertia	0.2527 $[kg \cdot m^2]$
F_{max}	Maximum applicable force	± 0.15 [N]

Hence, defining the state and input vectors as

$$\mathrm{x} = \begin{bmatrix} x \\ y \\ \theta \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}, \quad \mathrm{u} = \begin{bmatrix} F_x \\ F_y \\ M \end{bmatrix} \tag{4.2}$$

the system equation can be written as:

$$\dot{\mathrm{x}} = \mathbf{A}\mathrm{x} + \mathbf{B}\mathrm{u} = \begin{bmatrix} \mathbf{0_{3\times 3}} & \mathbb{I}_\mathbf{3} \\ \mathbf{0_{3\times 3}} & \mathbf{0_{3\times 3}} \end{bmatrix} \mathrm{x} + \begin{bmatrix} & \mathbf{0_{3\times 3}} & \\ \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix} \mathrm{u}. \tag{4.3}$$

Finally, the corresponding discretized form of the matrices in (4.3) is computed with a *zero-order hold discretization*, that holds each sample value for one sample interval. The resulting discretized system is given by

$$\mathrm{x}_{k+1} = \mathbf{A_d}\mathrm{x}_k + \mathbf{B_d}\mathrm{u}_k = \begin{bmatrix} \mathbb{I}_\mathbf{3} & \mathbb{I}_\mathbf{3} \\ \mathbf{0_{3\times 3}} & \mathbb{I}_\mathbf{3} \end{bmatrix} \mathrm{x}_k + \begin{bmatrix} \frac{1}{2m} & 0 & 0 \\ 0 & \frac{1}{2m} & 0 \\ 0 & 0 & \frac{1}{2I_z} \\ \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix} \mathrm{u}_k. \tag{4.4}$$

This model will be used in the prediction phase of the Kalman filter to provide a first estimation of the state variable, and then to propagate the uncertainty ellipsoid $\mathcal{E}_k$ with the ellipsoid method.

4.2 Filtering procedure

In this section, the procedure applied by the filter at every time step of the simulation is presented in detail. The proposed filter was developed in Simulink, based on the Kalman filter theory (see Chapter 2), and it was adapted to process the required data and to apply the ellipsoid method. Its general work flow is represented in Figure 4.1.

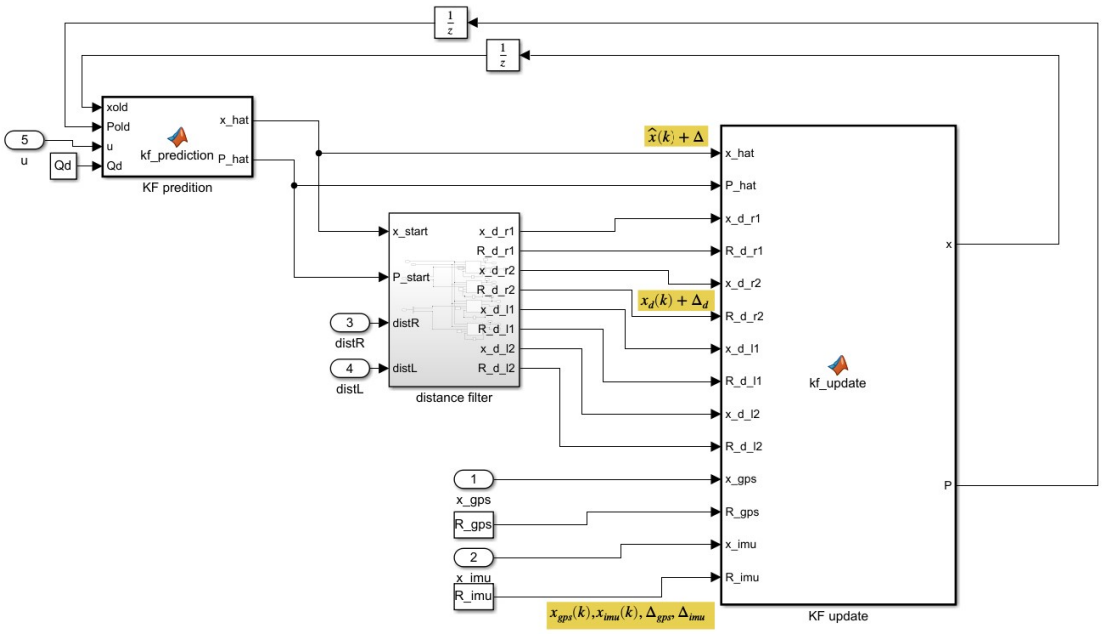


Figure 4.1: Simulink schema of the filter: a first prediction is made with the block on the left, in the middle block information coming from the ultrasound sensors are elaborated to produce measures of the UGV position, while in the last block the update phase of the Kalman filter produces a final estimation of the variables using the observations coming from all the sensors. Finally, with a feedback loop this estimation will be used as input for the prediction phase of the next time instant.

Moreover, particular focus is given on how the ellipsoid method is applied, starting with the 6×6 matrix $\mathbf{P_k}$, i.e. the covariance of the estimation, which refers to all the six states of the system, and the selection of the segment of the row to correctly perform the algorithm.

4.2.1 Prediction and uncertainty propagation

In the first block of the Simulink schema, the prediction phase of the Kalman filter is executed. In particular, the discrete-time state space description of the simplified model, composed by three double integrators and used to describe the behaviour of the UGV dynimics (Section 4.1), is exploited where the control input $\mathbf{u_{k-1}}$ and the previous step state prediction ($\hat{x}_{k-1}$ and $\mathbf{P_{k-1}}$) are given as input to obtain a preliminary estimation of the state variables at the current step k as

$$x_k^p = \mathbf{F_k}\hat{x}_{k-1} + \mathbf{G_k}u_{k-1} \tag{4.5a}$$

$$\mathbf{P_k^p} = \mathbf{F_{k-1}}\mathbf{P_{k-1}}\mathbf{F_{k-1}^T} + \mathbf{Q^d} \tag{4.5b}$$

where x_k^p is the preliminary estimation, simply defined as the state vector at time k+1, while $\mathbf{F_k}$ and $\mathbf{G_k}$ are respectively equal to matrices $\mathbf{A_d}$ and $\mathbf{B_d}$, defined in (4.4). The positive semi-definite matrix $\mathbf{P_k^p}$ represents the propagation of the uncertainty on the estimation, that will be later reduced in the *update phase*. Then, the behaviour of the noise is represented by the covariance matrix $\mathbf{Q^d}$ (already known or defined with a trial-and-error procedure) that defines the propagation of the uncertainty on the prediction $\mathbf{P_k^p}$. This matrix $\mathbf{Q^d}$, together with the input u_{k-1}, the estimations at the previous time step $\hat{x}_{k-1}$ and $\mathbf{P_{k-1}}$, are received as input (see Figure 4.2). In the case of $k = 1$ (simulation start), the initial conditions, $\hat{x}_0$ and $\mathbf{P_0}$ (2.4) are considered.
Notice that this is a theoretical response of the system to the given input $\mathbf{u_{k-1}}$ that, due to the system noise $\mathbf{d_k}$ and to the approximations of the system description, can be different from the real response.

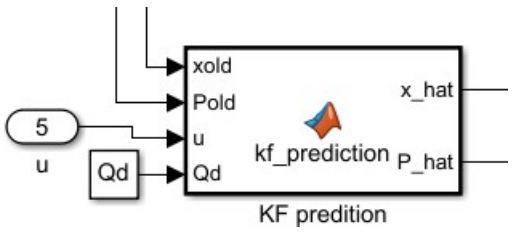


Figure 4.2: Prediction block of the Simulink schema. It receives as input the previous estimation, the input value and the covariance matrix of the system noise and provides as output a preliminary estimation of the state vector of the system.

4.2.2 Distance measurement and row selection

The second block of the schema in Figure 4.1 receives as inputs the prediction variables x_k^p and $\mathbf{P_k^P}$ obtained with (4.5) and the distance measurement coming from the sensors. Hence, for each of them, it provides as output a new measurement of the position with its covariance matrix, exploiting the ellipsoid method. The general representation of the block in Simulink is shown in Figure 4.3, while in Figure 4.4 a more precise view of its inner composition is provided.
Since each row of the field in the digital map is represented by a number N of segments (see Figure 4.5), defined by different equations, and the equation is part of the process to get the position from the distance, a method to determine the one to which the distance measurement is referring becomes necessary. More precisely, two different approaches are presented:

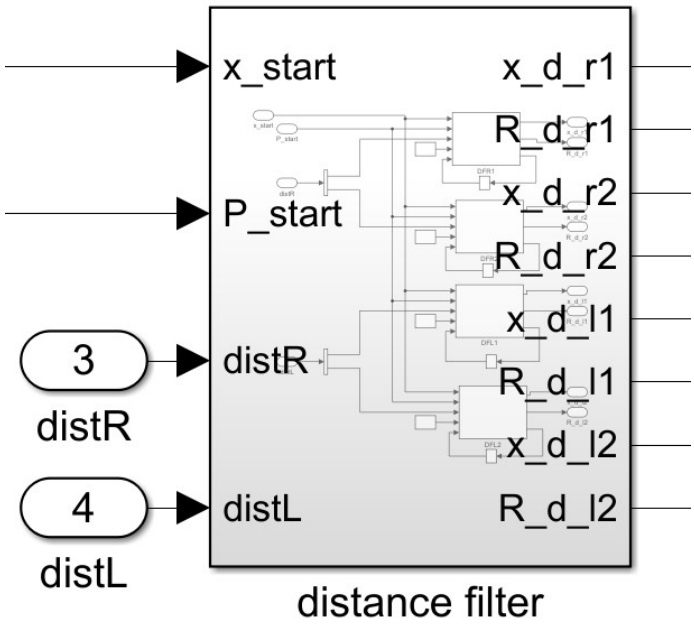


Figure 4.3: Simulink block of the distance filter: it process the right and left distance measurements and provides the new quantities (x_d_l/x_d_r) with their uncertainties (R_d_l/R_d_r) of the UGV position for each distance sensor.

1. In the first proposed method, starting and ending coordinates of each segment needs to be known.
   Beginning from the distance measurement and the estimation provided

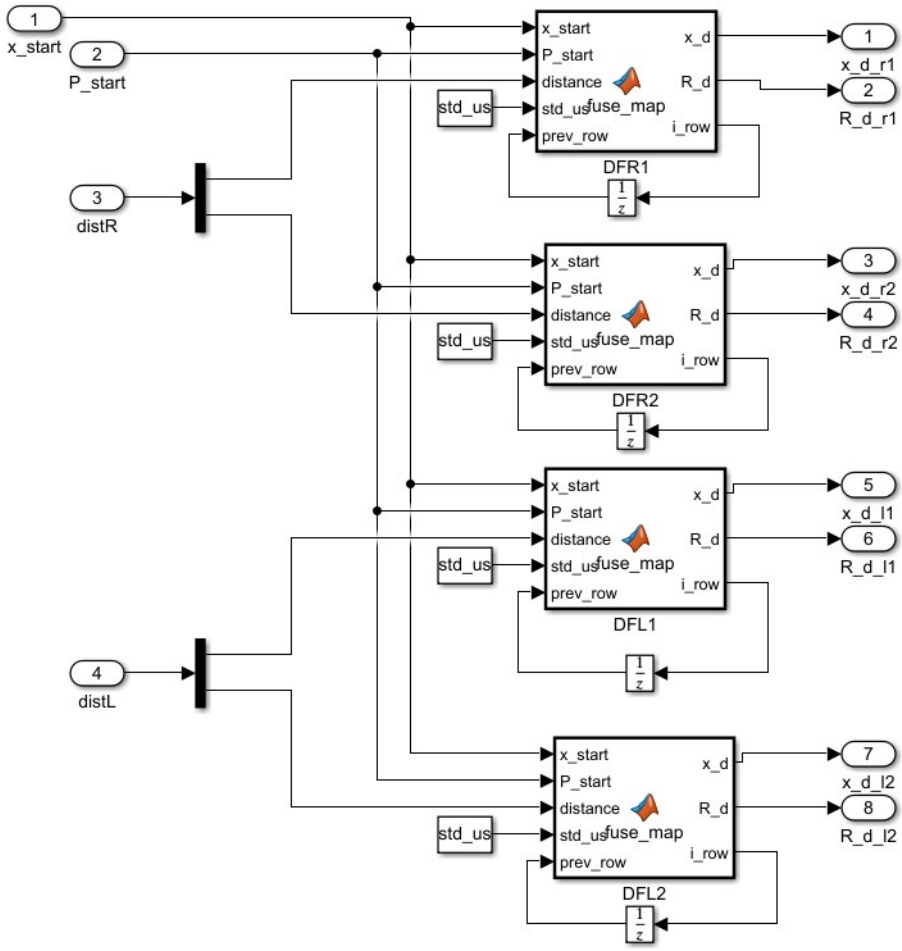


Figure 4.4: Inner composition of the distance filter: for each measurement, the initial prediction and the information of the error of the ultrasound sensor are used to perform an algorithm which returns information on the UGV position. To speed up the procedure, a feedback of the index of the selected row is used in the next time instant to check only a defined set of segments instead of all of them.

by the prediction block, the estimated measured point x_p is computed as:

$$x_p = \begin{bmatrix} x_k^{p(1)} + d_m \cos(x_k^{(3)} - \frac{\pi}{2}) \\ x_k^{p(2)} + d_m \sin(x_k^{(3)} - \frac{\pi}{2}) \end{bmatrix} \tag{4.6}$$

where $x_k^{p(i)}$ is the i-th component of the state vector prediction x_k^p. Therefore, x_p is an estimation on the digital map of the point on the row from

which the sensor is measuring the distance, according to the current prediction of the state vector.
Then, the distance between each segment of the rows and x_p is computed and the equation of the one containing it with the minimum error is selected.

2. The second method does not require the knowledge of the boundaries of each segment and exploits approach explained in Section 2.3 to calculate the minimum distance. This procedure leads to a lighter digital map but, on the other hand, the accuracy of the obtained results is lower. First, the minimum distance of the point x_k from each segment is computed in the standard way, as

$$d_{min}(x(k)) = \frac{|ax_k^{p(1)} + bx_k^{p(2)} + c|}{\sqrt{a^2 + b^2}} \tag{4.7}$$

Then, for each segment, another set of minimum distances could be obtained, starting from the sensor data, remembering the previously defined relation between the measured perpendicular distance and the minimum distance of a point and a row, i.e.

$$d_{min}(x_k) = d_m \cos(x_k^{p(3)} - \beta) \tag{4.8}$$

where β is the angle of incidence between the row and the x-axis, $x_k^{p(3)}$ is the orientation of the vehicle, and d_m is the distance data coming from each sensor.
The correct row equation could be selected, finding the one having the related pair of minimum distances, obtained with (4.7) and (4.8), with the lower difference.

To reduce the possibility of errors, this two methods could be used simultaneously, to check if the returned results are equals. In case of discrepancies, different strategies could be exploited. For example, the result that gives the minimum error between the two methods can be selected.

However, there is an important drawback in this step that can't be neglected: for each time instant k, all the segments of all the rows need to be processed in order to find the correct one. Since each row is composed by hundreds segments and a field has dozens of crop-rows, the duration of the selection process could become too long to be applicable in real time.
A possible solution for this problem is the definition of a "*sliding windows*": at each time step, a subset of the whole set of segments is selected to reduce

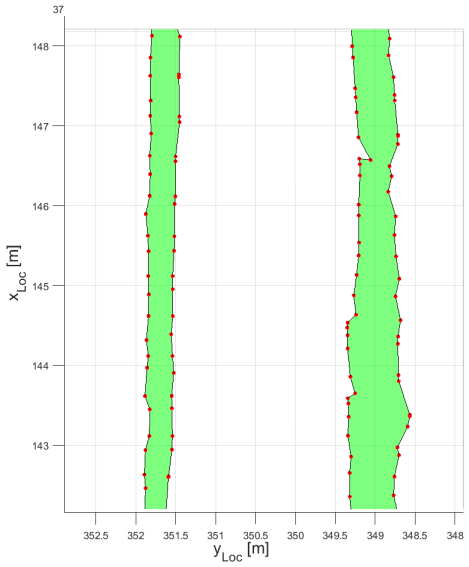


Figure 4.5: An example of two crop-rows in the digital map: between each pair of red dots there is a segment, defined by its equation. The purpose of the selection procedure described in this section is to find in the map the correct one from which the distance is measured by the sensor.

the number of data to process and speed up the overall operation. More precisely, by identifying each segment with an increasing index, the one of the previously selected segment can be employed to define a windows of $2w + 1$ "near possible segments" to be checked in the next selection phase. For example, with $w = 10$ and $idx(k) = 42$, the segments checked at the time $k + 1$ will be the ones in the range from $idx = idx(k) - w = 32$ to $idx = idx(k) + w = 52$. Indeed, reasonably, it can be assumed that in the next time step the sensors of the UGV will measure the distance from a near segment (or from the same).
A possible drawback of these methods is that, near to map line-string corners, the definition of the distance measurement can be affected. Indeed, according to the vehicle orientation, near the corners the ultrasonic sensor can capture one of the two segments and, correspondingly, the resulting position estimate can be more affected by the measurement error, since both the segments should be considered.
This limitation is due to the ellipsoid method, that could be applied only with a parallel pair of straight lines. Despite this, the introduced error is small and shows up only in few special cases.

4.2.3 Ellipsoid projection

Once collected the data of the considered segment, the ellipsoid method can be applied once the uncertainty relative to the first two members of the state vector x_k (position data) is extracted from the global uncertainty of the prediction $\mathbf{P_k^p}$, obtained in the prediction block (Figure 4.1). Indeed, the fusion of the distance measurement and the map information leads to an improvement only on the estimation of the UGV position, since no data about velocities or accelerations are obtained from the ultrasound sensor. For this reason, the $\mathbf{P_k^p}$ matrix needs to be projected in the 2D space to obtain a 2×2 positive semi-definite matrix $\mathbf{P_k^{p(2)}}$ representing the uncertainty of the position estimation $\mathcal{E}_k$, to be used in the ellipsoid method.

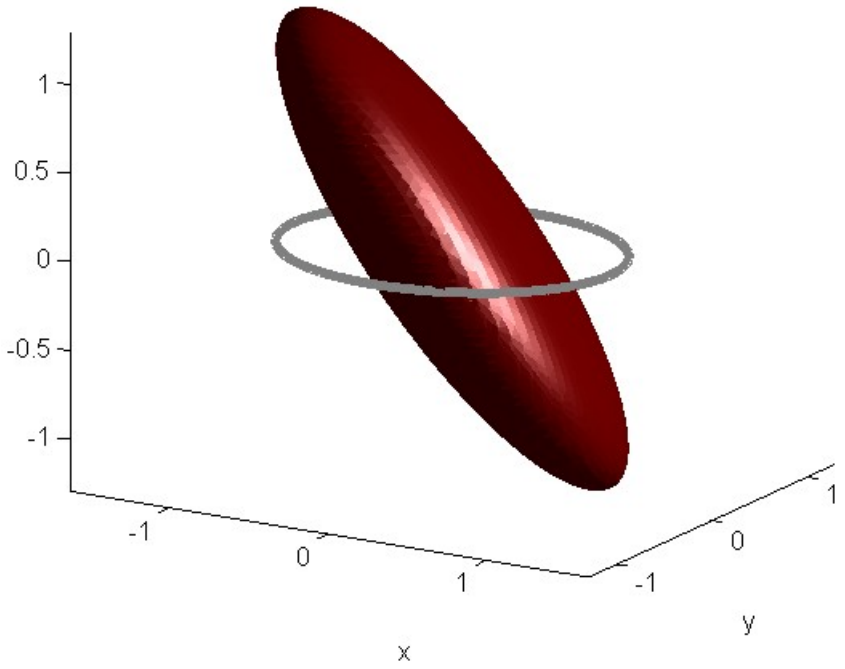


Figure 4.6: In this example a 3D ellipsoid (red ellipsoid) is projected into the 2D space (gray ellipse). In the considered case a 2D ellipse is obtained from a 6D ellipsoid.

The searched projection into the (x_1, x_2) space is the set of points where the gradient ∇f has no $(x_3, \ldots, x_n)$ components.
For the general n-dimensional case, the matrix $\mathbf{P_k^p}$ could be partitioned into

sub-matrices in the following way:

$$\mathbf{P_k^p} = \left[\begin{array}{c|c} \mathbf{J} & \mathbf{L^T} \\ \hline \mathbf{L} & \mathbf{K} \end{array}\right] \tag{4.9}$$

where $\mathbf{J}$ is a 2×2 sub-matrix, $\mathbf{L}$ is $(n-2) \times 2$, and $\mathbf{K}$ is $(n-2) \times (n-2)$. Let y be the vector (x_1, x_2) and let z be the vector $(x_3, \ldots, x_n)$ so that $x = \begin{pmatrix} y \\ \hline z \end{pmatrix}$, considering $\mathbf{P_k^p} x$ the condition on the gradient would be:

$$\mathbf{L}y + \mathbf{K}z = 0$$

that solved with the equation of the original ellipsoid becomes:

$$\left(\begin{array}{c|c} y^T & z^T \end{array} \right) \begin{bmatrix} \mathbf{J} & \mathbf{L^T} \\ \mathbf{L} & \mathbf{K} \end{bmatrix} \begin{pmatrix} y \\ \hline z \end{pmatrix} = 1.$$

This provides us the equation of the final curve, given by

$$y^T \mathbf{P_k^{p(2)}} y = 1 \tag{4.10a}$$

$$\mathbf{P_k^{p(2)}} = \left(\mathbf{J} - \mathbf{L'K^{-1}L}\right) \tag{4.10b}$$

where $\mathbf{P_k^{p(2)}}$, the Schur complement of $\mathbf{P_k^p}$, is a 2×2 positive semi-definite matrix.
Notice that, to obtain the general q-projection represented by a $q \times q$ matrix $\mathbf{P_k^{p(q)}}$, is sufficient to apply the above described procedure, defining the $\mathbf{J}$ matrix as a $q \times q$ matrix and, accordingly, $\mathbf{L}$ as an $(n-q) \times q$ and $\mathbf{K}$ as an $(n-q) \times (n-q)$.

Finally, exploiting the collected data, the ellipsoid method is used to generate the relative measure of the position elaborated from the distance measure given by the sensor, with a new ellipsoid representing its uncertainty. This results are the outputs of the second block in the simulink schema (see Figure 4.3) and will be later used in the next block to update and improve the state estimation.

4.2.4 Relative position of the sensors

Until now, the resulting position obtained from the elaboration of the distance measurement of each sensors is assumed to be the position of the CoM of the UGV. Actually, this is true only in the case where each sensor is

mounted exactly on the CoM. In all the other cases, the obtained result is the position of the considered sensor in the global frame, since the measured value refers to the distance between the sensor and an obstacle. This problem can be simply solved by knowing the position of the sensors with respect to the CoM in the reference system jointed to the vehicle (see Figure 4.7). Indeed, knowing the orientation of the UGV, it is possible to obtain the position of the CoM with a change of coordinates.

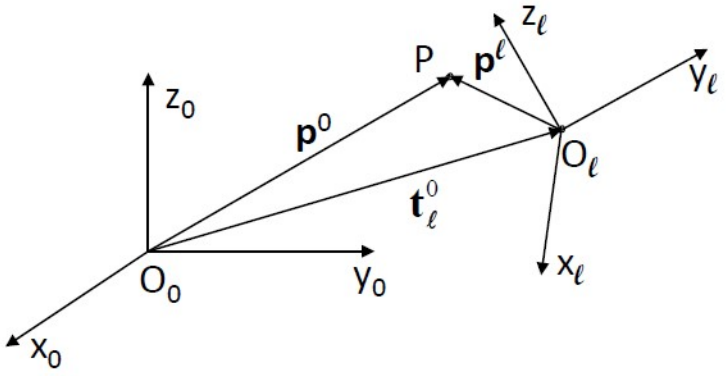


Figure 4.7: The global frame, with center in $\mathbf{O_0}$ and the local frame, with center $\mathbf{O_l}$ fixed to the CoM and jointed to the UGV.

In particular, let consider the system in Figure 4.7, where the reference system with center in $\mathbf{O_0}$ is the global frame, while the other is the one fixed to the UGV, with the center $\mathbf{O_l}$ fixed to the CoM (local frame).
Considering this schema, the following values are defined:

- $\mathbf{p^l}$ is the position of the sensor with respect to the CoM: for example, if it is placed in the right-upper angle of an UGV with length l and width w, its relative position will be $(w/2, l/2)$, if we assume that the CoM is in the center of the UGV and it is the center of the local frame (see Figure 4.8). Notice that this value is known, since it is a design choice and does not change during the route of the vehicle.

- $\mathbf{p^0}$ is the position of the sensor with respect to the global reference. This value is obtained as output of the ellipsoid method, starting from the distance reading of the considered sensor.

- $\mathbf{t_l^0}$ is the translation vector representing the origin of the local frame $\mathbf{O^l}$ with respect to the global one that is, by definition, the position of the CoM of the UGV.

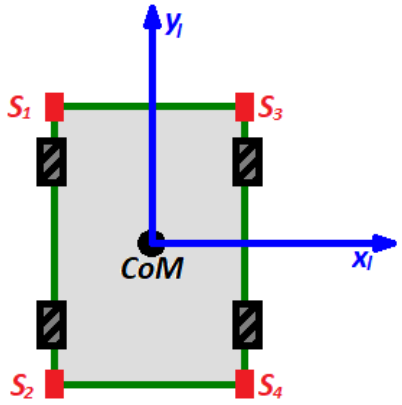


Figure 4.8: Sensors position in the considered case (red rectangles S_i).

- $\mathbf{R_l^0}$ is the rotation matrix of the local frame with respect to the global frame that, exploiting the orientation of the UGV obtained by the first prediction x_k^p, can be written as:

$$\mathbf{R_l^0} = \begin{bmatrix} \cos(x_k^{p(3)} - \pi/2) & -\sin(x_k^{p(3)} - \pi/2) \\ \sin(x_k^{p(3)} - \pi/2) & \cos(x_k^{p(3)} - \pi/2) \end{bmatrix} \tag{4.11}$$

  The quantity $x_k^{p(3)}$ is the orientation of the UGV considering where the vehicle is pointing and, by definition of the local frame, this is the direction of the y_l axis. As a consequence, $x_k^{p(3)}$ will be the angle between y_l and x_0 and, since in a rotation matrix the considered angle is the one between the same axes of two different reference systems, in (4.11) appears the quantity $x_k^{p(3)} - \pi/2$

Hence, since the position of the sensor in the global frame can be expressed as:

$$\mathbf{p^0} = \mathbf{R_l^0 p^l} + \mathbf{t_l^0}, \tag{4.12}$$

the position of the CoM ($\mathbf{t_l^0}$) starting from the one of the sensor (received as output of the ellipsoid method) and its relative position in the local frame (already known) can be obtained as

$$\mathbf{t_l^0} = \mathbf{p^0} - \mathbf{R_l^0 p^l} \tag{4.13}$$

4.2.5 Estimation update and final prediction

The position information, elaborated in the previous phase, enters as input together with GPS and IMU measurements and their relative uncertainty in

the last block of the schema (Figure 4.1), where the filter corrects the preliminary estimate (also received as input), updating it with each observation received as:

$$\mathbf{S_k} = \mathbf{H_k P_k^p H_k^T} + \mathbf{R_d} \tag{4.14a}$$
$$\mathbf{K_k} = \mathbf{P_k^p H_k^T S_k^{-1}} \tag{4.14b}$$

$$\Delta y_k = y_k - \mathbf{H_k} x_k^p \tag{4.15a}$$
$$\hat{x}_k = x_k^p + \mathbf{K_k} \Delta y_k \tag{4.15b}$$
$$\mathbf{P_k} = (\mathbf{I} - \mathbf{K_k H_k})\, \mathbf{P_k^p} \tag{4.15c}$$

Quantities in equations (4.14) and (4.15) are defined and initialized as follows:

- $\mathbf{H_k}$ is the observation matrix for the considered sensor. It relates the state vector x_k with the values measured by the sensor. For example, in the case of GPS or ultrasound sensors, which provide information only on the UGV position (first two elements of x_k), $\mathbf{H_k}$ will be:

  $$\mathbf{H_{GPS/U.S.}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

  so that the result of $\mathbf{H_k} x_k$ is a 2-element vector composed by the values of the position contained in the state vector x_k.
  In the IMU case, thanks to the measurements of the accelerometer (accelerations) and the gyroscope (angular velocities), information on all the values of the state vector are indirectly provided using the classical dead-reckoning estimate approach, i.e. integrating the inertial sensor measurements. Hence, for simulation purpose, the corresponding observation matrix is defined as the identity matrix $\mathbb{I}_6$:

  $$\mathbf{H_{IMU}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- x_k^p and $\mathbf{P_k^p}$ represent the preliminary estimation obtained in the prediction phase with (4.5).

- $\mathbf{R_d}$ is the covariance matrix of the sensor measurement, representing its uncertainty and received as input in this block. In the case of GPS and IMU it is defined following the design parameters of the sensors (see Table 1.2) while in the case of distance sensors it is obtained with the ellipsoid method as explained before, exploiting the uncertainty parameters of the sensors.

- $\mathbf{y_k}$ is the observation provided by the considered sensor, received as input.

Finally, the obtained output of this block is the final prediction of the state vector at the time k, $\hat{x}_k$, with its covariance matrix $\mathbf{P_k}$, representing the uncertainty. These values will be then used in the control part, to make the UGV follow the desired trajectory with the greater possible precision. Moreover, with a feedback loop, they will be send back to the prediction block for the next time instant $k + 1$.

Chapter 5

Results

This chapter focuses on the main results collected during the development of this thesis. In particular, improvements and upgrades made to increase the effectiveness of the proposed filter are analyzed and described, comparing then final results with the performance obtained using standard filters, without the merging with low complexity digital maps.
The considered case study involves a vineyard located in Barolo (Piedmont, Italy), which extends over a surface of about 0.7 hectares and its elevation ranges from 460 to 490 m above sea level. The spaces between vine plants and the inter-row space are about 0.9 m and 2.5 m, respectively.
As reported in Chapter 1, a 4-wheel steering electric UGV is operated, equipped with two HC-SR04 ultrasonic sensors on both its lateral sides, a Novatel OEM7600 receiver GPS, and a XSens MTI-10 series IMU (see Table 1.2 for their main features).
Moreover, as described in Section 4.1, the vehicle dynamics is modeled with three double integrators, considering the vehicle and actuators features and the system constraints reported in Table 1.1 and Table 4.1, respectively.

In a very first phase of this work, the objective was to understand how to represent and approximate in the space a subset of points whose distances from a given line are coherent with the one measured by a sensor.
In this experiment, an initial, cylindrical, random set of points is defined in a graph where x, y, and z axes define the position (x and y coordinates) and the orientation (z) of the vehicle, respectively. This cylinder is the combination of all the possible position and orientation of the UGV at a given time. First,

the real position and orientation (blue star in Figure 5.1a) are defined as

$$p_0 = \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \end{bmatrix} = \begin{bmatrix} 1 \\ -3 \\ \frac{2\pi}{3} \end{bmatrix}. \tag{5.1}$$

Then, a simplified field composed by a single, straight line $l \equiv ax+by+c = 0$ (red line in Figure 5.1a) is defined initializing $a = -1$, $b = -1$ and $c = 10$, respectively, and the true distance d_0 of the UGV CoM from the simplified vine row is computed with (2.14), (2.16) and (2.15).
In the 2D space (Figure 5.1a), a preliminary set of possible estimated positions is simulated defining a circle (black, dashed circle) with radius $\rho_{max} = 3$, which center is the preliminary estimation $\hat{p}$, obtained adding to the real position a random error in a defined range of values. Indeed, at the beginning of this work, an unknown-but-bounded (UBB) deterministic measurement error was considered as uncertainty of each sensor measurement. In particular, in this approach, the uncertainty is described by a noise which is known only to have given bounds, i.e. minimum and maximum values that it can assume:

$$\delta_{UBB} \in [\Delta_m, \Delta_M]. \tag{5.2}$$

In particular, $\hat{p}$ was defined as:

$$\hat{p} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{\theta} \end{bmatrix} = \begin{bmatrix} x_0 + \delta_x \\ y_0 + \delta_y \\ \theta_0 \end{bmatrix}, \tag{5.3}$$

where, $\delta_x \in [-3, 3]$, $\delta_y \in [-3, 3]$ and, for simplicity, zero uncertainty on θ is assumed.
First, all the deterministic set of points contained in the circle can be obtained exploiting

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} + \rho \begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix}, \tag{5.4}$$

for each possible value of $\rho \in [0, \rho_{max}]$ and $\alpha \in [0, 2\pi]$. Then, all the points, whose distances from the above-mentioned line are equal to the distance measured by the distance sensor (or in a range of possible values), are selected. The range of possible distance values can be obtained considering the uncertainty related to the sensor measurement. In the considered example, $\delta_d \in [\Delta_m^d, \Delta_M^d] = [-0.3, 0.3]$ is selected as the ultrasound sensors uncertainty, so that the measured distance can be defined as:

$$d_m = d_0 + \delta_d. \tag{5.5}$$

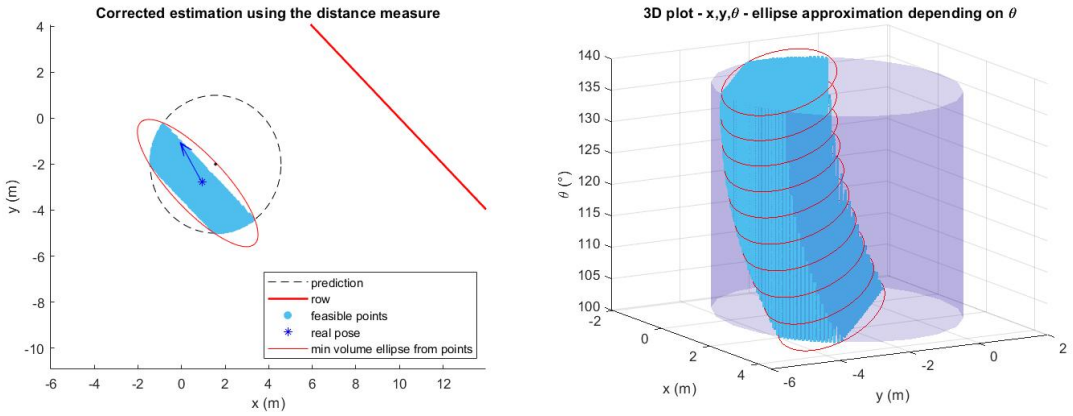
Hence, the points contained in the light blue area ($\mathcal{A}$) in the figure, i.e. the set of all the feasible points in the preliminary estimation (black, dashed circle), satisfy the following condition:

$$d_m + \Delta_m^d \leq d_m^{(i)} \leq d_m + \Delta_M^d, \forall i \in \mathcal{A} \tag{5.6}$$

where $d_m^{(i)}$ is defined as the distance from the line of the i-th point contained in the previously defined circle, and it is obtained as before, exploiting (2.14), (2.16) and (2.15).
As a result, after this analysis, a shape that models the subset of the above-mentioned feasible points was obtained exploiting the ellipsoid method as described in Section 3.3. More precisely, this set represents all the possible positions of the UGV whose distances from the line are consistent with the measured one, and it can be modeled by a minimum-volume ellipse, i.e. the ellipse with the lower possible volume, which contains all the points that satisfy this condition. Indeed, this was the first step that leads to the adoption of the ellipsoid method, exploited to reduce the uncertainty, merging initial data with distance measurements and map information (rows equations).
Same consideration can be done in a more complex scenario, considering that also the orientation is affected by an uncertainty and defining, in this case, $\hat{p}$ as

$$\hat{p} = \begin{bmatrix} x_0 + \delta_x \\ y_0 + \delta_y \\ \theta_0 + \delta_\theta \end{bmatrix}, \tag{5.7}$$

where, $\delta_x \in [-3, 3]$, $\delta_y \in [-3, 3]$ and, $\delta_\theta \in [-\pi/9, \pi/9]$. In this case, the set of the feasible points depends also on the orientation and it is obtained exploiting the same procedure described before, obtaining a shape in the 3D space, showed in Figure 5.1b (light blue volume).
Figure 5.1 highlights the results collected using the ellipsoid method: starting from an initial set of points (black, dashed circle and purple cylinder), the extracted subset (light blue area and light blue volume) can be approximated with the minimum-volume ellipsoid (red ellipses) provided by the algorithm. This experiment was useful to observe that modeling the light blue area using a simplified shape (ellipse) is necessary to obtain a representation of the updated prediction uncertainty and allows the iteration of this procedure when, in a scenario where the UGV is moving, the uncertainty is first propagated and increased and then, exploiting the ellipsoid method, reduced. Moreover, for simplicity, in the next experiments the ellipsoid method will be applied in



(a) (b)

Figure 5.1: (a) Starting from an initial set of possible positions (dashed line), the points at the provided distance are selected and then approximated with the minimum-volume ellipse. In (b), considering also the orientation, results about the shape of the feasible points set are shown. For each orientation, an ellipsoidal approximation can be done.

the 2D space to only improve position estimates, considering a good starting orientation prediction. However, potentially, it can also be applied in the 3D space, using ellipsoids instead of ellipses, to reduce the uncertainty that affects the orientation. In this case, however, the complexity of the algorithm significantly increase.

In the following test, a simulation in a dynamic scenario with a simple deterministic filter was executed. In this setup, the vehicle moves in a simplified field but, in this case, the vine row is composed by a small set of segments. About the UGV, the features contained in Table 5.1 are considered in all the following simulations.

Table 5.1: UGV features.

Parameter	Description	Value
L	Length of the UGV	1.5 $[m]$
T	Width of the UGV	1 $[m]$
m	Mass of the vehicle	10 [kg]
I_z	Moment of inertia	0.2527 $[kg \cdot m^2]$
F_{max}	Maximum applicable force	$\pm$0.15 [N]

First, at each time step $\Delta t = 1s$, a random sequence of input forces was randomly generated to make the UGV move in the environment, considering a range of possible applicable values on the base of the constraints on the maximum applicable force in Table 5.1, i.e. $F_u \in [-0.15N, 0.15N]$.
Then, after that initial conditions of the UGV and the filter have been initialized with

$$x_0 = [x_0, y_0, \theta_0, v_{x0}, v_{y0}, w_0]^T = [0.2, 9, 2\pi/3, 0, 0, 0]^T, \tag{5.8a}$$

$$\hat{x}_0 = [0, 0, 0, 0, 0, 0]^T, \tag{5.8b}$$

$$\mathbf{P_0} = \mathbb{I}_\mathbf{6}, \tag{5.8c}$$

respectively, the UGV position, orientation and velocity were estimated with (4.5a), considering the defined model (4.4) and an increasing uncertainty, represented by an ellipsoid and obtained with (4.5b). The behaviour of the noise, defined by its covariance matrix $\mathbf{Q^d}$, was initialized in this phase with the following diagonal matrix:

$$\mathbf{Q^d} = \begin{bmatrix} 0.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.3 \end{bmatrix}. \tag{5.9}$$

Then, exploiting the Schur complement on the obtained matrix $\mathbf{P_k^p}$ as explained in Section 4.2.3, the 2D projected ellipse representing the uncertainty that affect the position estimate is computed. Hence, starting from this preliminary estimation and the distance measurement provided by the sensor, obtained exploiting the row selection procedure described in Section 4.2.2 on the digital map, and then using (2.14), (2.16) and (2.15), an improved prediction with a reduced uncertainty was obtained, using the ellipsoid method and considering the two bounds (offsets) calculated with (2.23). Notice that, as in the previous phase, an unknown-but-bounded (UBB) deterministic measurement error was considered for the ultrasound sensors, defining $\Delta_m^d = -0.1m$ and $\Delta_M^d = 0.1m$.
Finally, in the next time step, the reduced ellipse that models the uncertainty of the updated prediction (red ellipse in Figure 5.2) was propagated using the system model and (4.5) with the current, updated prediction, iterating for 7 steps the above explained procedure.
During this simulation, ellipsoid method performance in a dynamic scenario

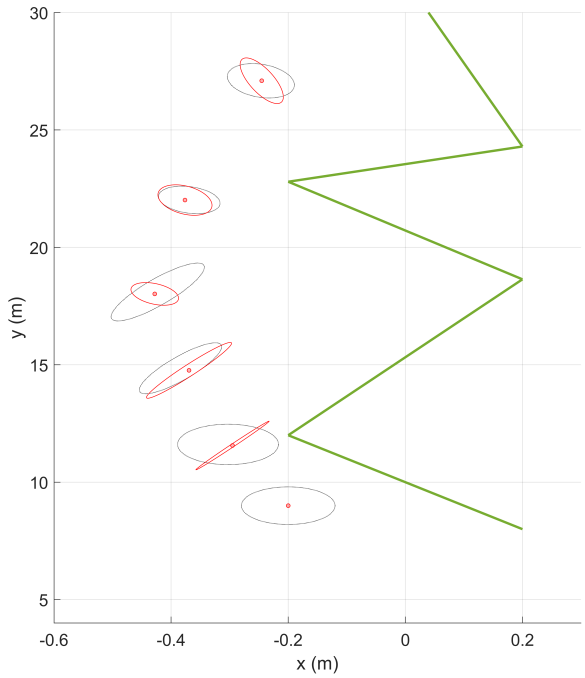


Figure 5.2: Evolution of the estimation uncertainty (black dotted line) and minimum volume ellipsoid (red line) when data from sensors are fused with the georeferenced information provided by the simplified map (green linestring).

were analyzed, collecting outcomes in terms of uncertainty propagation and reduction. In particular, these results can be seen in Figure 5.2, where the evolution and the improvement of the estimation in an environment composed by a single row (green line) are highlighted. Indeed, starting from the initial states estimation and its uncertainty (black dotted ellipses), provided by the system model, a new, updated prediction (red dots) is gained exploiting the ellipsoid method. It can be seen that the uncertainty of the updated estimation, represented by the red ellipses, is notably reduced at each time step.

After that, the simulation setup was improved providing data from other sensors to the model estimation. In this phase, the objective was to see if the introduction of the map and the distance sensors was really effective and functional for the improvement of localization results, provided by the filter. Maintaining the same setup and procedure of the previous simulation, GPS measurements were added, defining their UBB uncertainty with $\Delta_m^{GPS} = -0.6m$ and $\Delta_M^{GPS} = 0.6m$ and merging them with the preliminary

estimation obtained as explained before. Then, the distance measure was fused with map data to obtain an additional position information and further improve the current prediction.

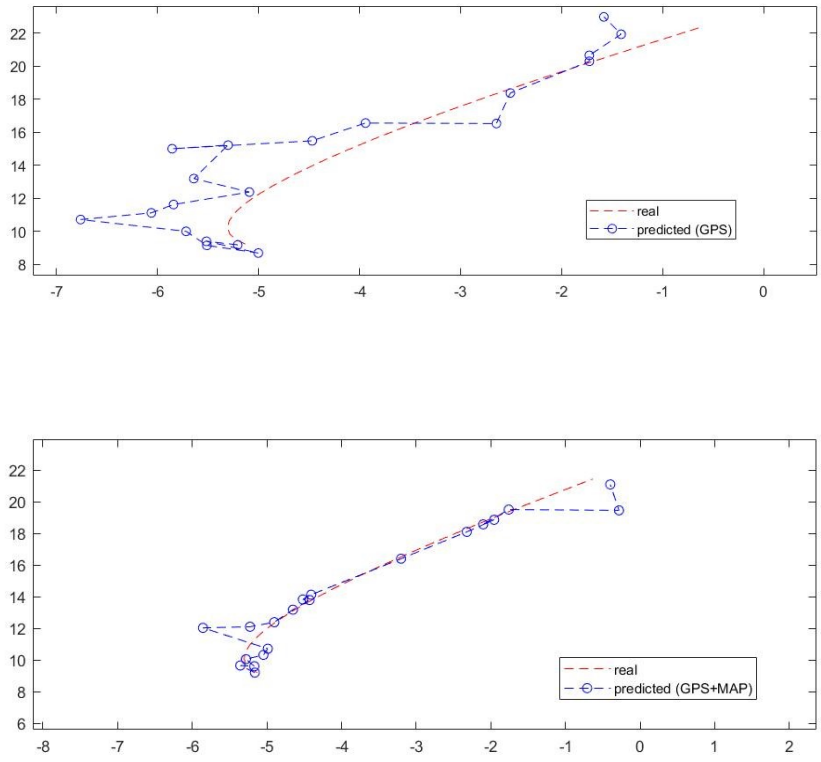


Figure 5.3: Upper figure: estimated trajectory (dashed, blue line) and real trajectory (dashed, red line) when only GPS data are exploited to estimate the position of the UGV. Lower figure: results collected adding distance measurement merged with map information are shown.

Simulation results collected in terms of trajectory estimation are shown in Figure 5.3, where the estimated UGV trajectory (blue, dashed line) is compared with the real one (red, dashed line) when two different filters are exploited. In the upper figure, a filter which uses only GPS data to estimate the UGV position is considered. On the other hand, in the lower figure, the filter merges GPS data with distance and map fused information, to obtain a more accurate estimation. Indeed, as it can be seen, in this figure the estimated trajectory is more precise and closer to the real one, proving that distance sensors and map merged information improve the accuracy of the

filter, providing better position estimates.
Collected data in terms of average error ($e_{avg} = \frac{1}{N}\sum_i |x_i - \hat{x}_i|$) and maximum error ($e_{max} = \max_i |x_i - \hat{x}_i|$) during the trajectory estimation also highlights the obtained improvement, as it can be seen in Table 5.2.

Table 5.2: GPS and GPS + distance and MAP comparison

	e_{avg}	e_{max}
Only GPS	0.9 [m]	1.7 [m]
GPS+ distance and MAP	0.4 [m]	0.9 [m]

Until this point, the filter was considered to be deterministic, modeling the noise of the sensors measurements as an unknown-but-bounded error. However, sensors uncertainties were usually given in terms of standard deviation, providing a stochastic interpretation of the noise behaviour. Moreover, standard filtering approaches like Kalman filters operate with probabilistic functions, assuming that errors have a normal (Gaussian) distribution.
These reasons lead to the conversion of the above-mentioned filter into a probabilistic one, applying the Kalman filter theory explained in Chapter 2. Both GPS and IMU measurements are included in this simulation, comparing standard filter localization with positioning data obtained including map and distance information. In this case, the uncertainties of the selected sensors are modeled as errors with Gaussian distribution, using the parameters resumed in Table 5.3.

Table 5.3: List of sensors features.

Sensor	Parameter	Standard deviation (σ)
Ultrasonic	Distance accuracy	3 [mm]
GPS	Horizontal accuracy	0.6 [m]
	Vertical accuracy	0.6 [m]
IMU	Position accuracy	1 [m]
	Velocity accuracy	0.05 [m/s]
	Accelerometer accuracy	200 [m/s^2]
	Gyroscope accuracy	450 [deg/s]

The behaviour of the noise, defined by the covariance matrix $\mathbf{Q^d}$, was initialized after a trial and error procedure with the following diagonal matrix:

$$\mathbf{Q^d} = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}. \tag{5.10}$$

As before, at each time step $\Delta t = 1s$, a random sequence of input forces was randomly generated to make the UGV move in the environment, considering again the range of possible applicable values in Table 5.1 ($F_u \in [-0.15N, 0.15N]$).
First, the initial condition of the UGV is defined as

$$x_0 = [x_0, y_0, \theta_0, v_{x0}, v_{y0}, w_0]^T = [350.5, 139, \pi/2, 0, 0, 0]^T, \tag{5.11}$$

while the filter is initialized with

$$\hat{x}_0 = [0, 0, 0, 0, 0, 0]^T, \tag{5.12a}$$
$$P_0 = \mathbb{I}_{\mathbf{6}}. \tag{5.12b}$$

Then, exploiting the previously defined model of the UGV (4.4), defined by the matrices

$$\mathbf{A_d} = \begin{bmatrix} \mathbb{I}_{\mathbf{3}} & \mathbb{I}_{\mathbf{3}} \\ \mathbf{0_{3\times 3}} & \mathbb{I}_{\mathbf{3}} \end{bmatrix} \text{ and } \mathbf{B_d} = \begin{bmatrix} \frac{1}{2m} & 0 & 0 \\ 0 & \frac{1}{2m} & 0 \\ 0 & 0 & \frac{1}{2I_z} \\ \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix}, \tag{5.13}$$

and the sensors observation matrices $\mathbf{H_{GPS}}, \mathbf{H_{U.S.}}, \mathbf{H_{IMU}}$ (defined in Section 4.2.5) applied to the real state, the real behaviour of the robot in the environment, x, and the quantities measured by the sensors (x_{GPS}, x_{IMU}, x_d) were simulated. Hence, using Simulink and the filter defined in Chapter 4, a preliminary estimation was obtained using the system equations (4.5) with matrices $\mathbf{A_d}, \mathbf{B_d}$ (5.13) (prediction phase) and then improved exploiting distance data and map fused information in the distance filter, thanks to the row selection process (Section 4.2.2) and the ellipsoid method application (Section 3.3). Finally, these information can be merged with GPS and IMU measures, to obtain a final, improved estimation at each time step in the

update phase, using (4.14) and (4.15).
Results are collected in terms of position estimation errors along the direction perpendicular to the rows. Indeed, a low error in this direction is the most important requirement to avoid collision and large inaccuracies in the UGV trajectory. However, a low complexity in the simulation setup is still maintained: only one distance measurement is merged with the map, composed by one simple row on the right side of the vehicle. The use of only one distance sensor is obtained activating only one element in the Simulink distance block (Figure 4.4), and disabling the others.

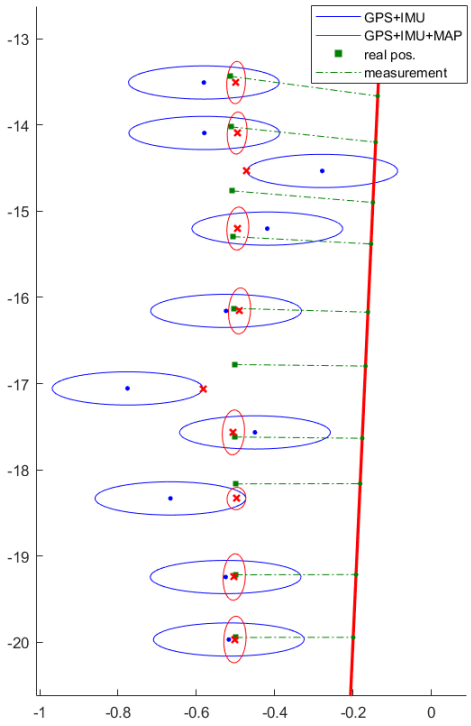


Figure 5.4: Comparison between prediction and uncertainties provided by a classic filter provided by GPS and IMU sensors (blue ellipses and blue circles) and the ones obtained adding map information and distance sensors on board the UGV (red ellipses and red crosses). The green square is the real position of the UGV CoM and the green, dashed lines represent the distance sensors measurements.

Simulation results can be seen in Figure 5.4, where estimations obtained

when only GPS and IMU data are exploited (blue ellipses) are compared with improved predictions (red ellipses), obtained adding information coming from the fusion of the map with distance sensor measurements. As it can be seen, not only the final uncertainty (represented by the ellipse dimension) is notably reduced, but also the updated predictions, i.e. the centers of the red ellipses, are more precise and closer to the real positions (green squares). In Table 5.4, results obtained in terms of average error, maximum error and standard deviation (σ) in the two cases are shown.

Table 5.4: GPS+IMU and GPS+IMU+distance and MAP comparison

	e_{avg}	e_{max}	σ
GPS+IMU	0.28 [m]	1.05 [m]	0.34 [m]
GPS+IMU+ distance & MAP	0.08 [m]	0.78 [m]	0.07 [m]

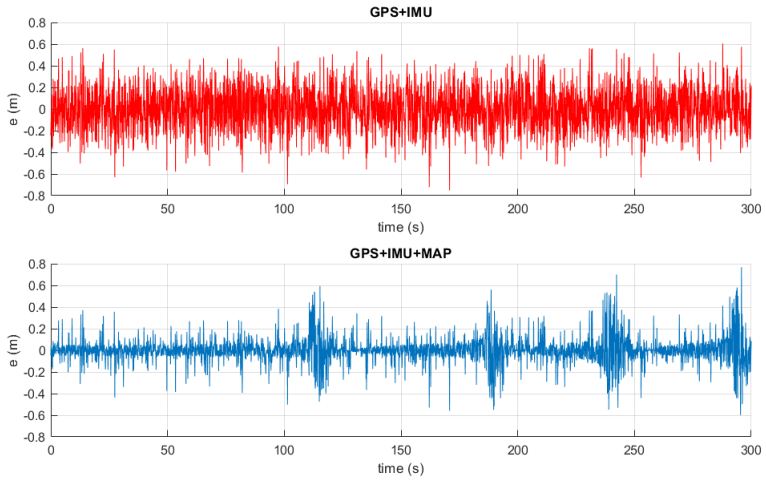


Figure 5.5: In the upper figure the error when only on board sensors are exploited, in the lower one the error when they are combined with low-complexity maps and distance measures.

Also in Figure 5.5 an important decrease of the error when the map is introduced can be seen. Indeed, while in the upper figure (GPS+IMU) the error ranges from 0.2 to 0.8m, in the lower figure (GPS+IMU+MAP) the error is particularly reduced, going from 0.05m to 0.2m. However, this procedure presents a not negligible drawback: some anomalies are still present in the

improved filter error. Indeed, in some cases, it reaches high peaks of 0.4m, 0.6m and 0.8m that, especially in a space of 2m, can seriously affect the localization performance, leading to failures and collisions during the UGV operations. More precisely, these anomalies can be attributed to the effect of map line-string corners or in large errors in the preliminary estimation, due to poor GPS signal and model approximations, which can affect the result of the row selection process and the resulting position estimate.

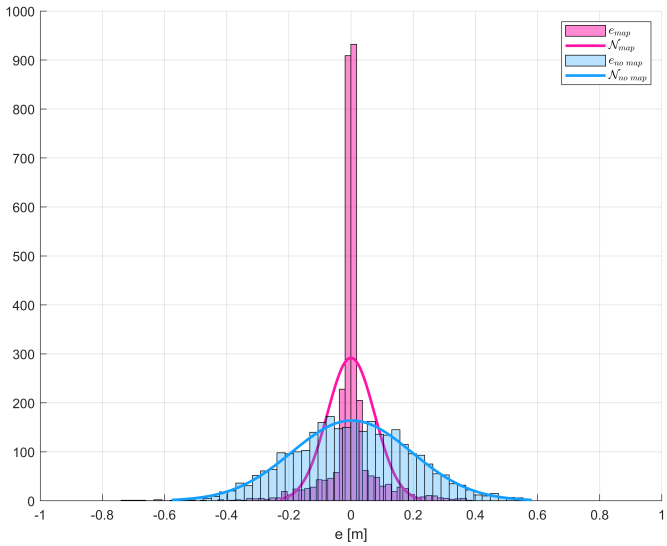


Figure 5.6: Histograms for error estimation when maps are combined with on-board sensors (magenta) or they are not exploited (light blue).

On the other hand, the effectiveness of the proposed approach is confirmed by the histograms reported in Figure 5.6, that highlight the significant reduction of the standard deviation of the error when the maps are exploited.

These results lead to the final phase of the filter design, where the objective was to further improve the position estimations and reduce the effects of the above mentioned anomalies. More precisely, the same procedure is executed, but the map is substituted with a real representation of the considered field and four ultrasound sensors are properly mounted on-board the UGV, two for each side, activating all the four elements in the Simulink block described in Section 4.2.2 (Figure 4.4). In this configuration, since more than one distance sensor is exploited, it is important to remark that the roto-translation explained in Section 4.2.4 needs to be applied to the ellipsoid method output (that is, in this case, the position of the sensor), to convert it into the

position of the UGV CoM, knowing the relative position of each sensor in the local reference system, reported in Table 5.5. In the considered case, the four ultrasound sensors are mounted at the four angles of the UGV, and the local frame is centered into the UGV CoM, in the middle point of the vehicle (see Figure 4.8).

Table 5.5: Sensors position in the local frame (L=1.5m, T=1m)

	x	y
Sensor 1	$-T/2$ [m]	$L/2$ [m]
Sensor 2	$-T/2$ [m]	$-L/2$ [m]
Sensor 3	$T/2$ [m]	$L/2$ [m]
Sensor 4	$T/2$ [m]	$-L/2$ [m]

It follows that four additional position data are exploited in the update phase of the filter, providing a further increase in the final localization accuracy of the UGV. Moreover, considering both the vine rows at the sides of the vehicle also provides some benefits. Indeed, it allows to compensate the effects of the map line-string corners, removing the anomalies that appears in the previous chart.
Obtained results are shown in Figure 5.7, where position estimation error obtained fusing data measured by GPS and IMU (red line) and the one got merging the same information with the map (blue line) are represented. It is possible to observe that, in the second case, the estimation error values are significantly lower. Moreover, the improvement obtained by adding additional distance sensors and considering also the second row at the other side of the UGV is clear: while in Figure 5.5 errors goes from 5cm up to 20cm, reaching peaks of 80cm in the worst cases, in Figure 5.7 the resulting estimation error is notably lower (2cm or 3cm in the worst cases).
Same results are highlighted in Figure 5.8, where the two errors are represented with two different plots with the same scale, and in Table 5.6, where data in terms of average error, maximum error and standard deviation in the two cases are shown.
A comparison between errors obtained with one, two and four sensors equipped on board the UGV and the digital map of the real environment is shown in Figure 5.9 and Table 5.7. It can be seen that better performance of the designed filter are obtained increasing the number of equipped sensors, reaching best results when four sensors are employed. The small improvement obtained with a total of eight sensors (four for each side) does not justify the

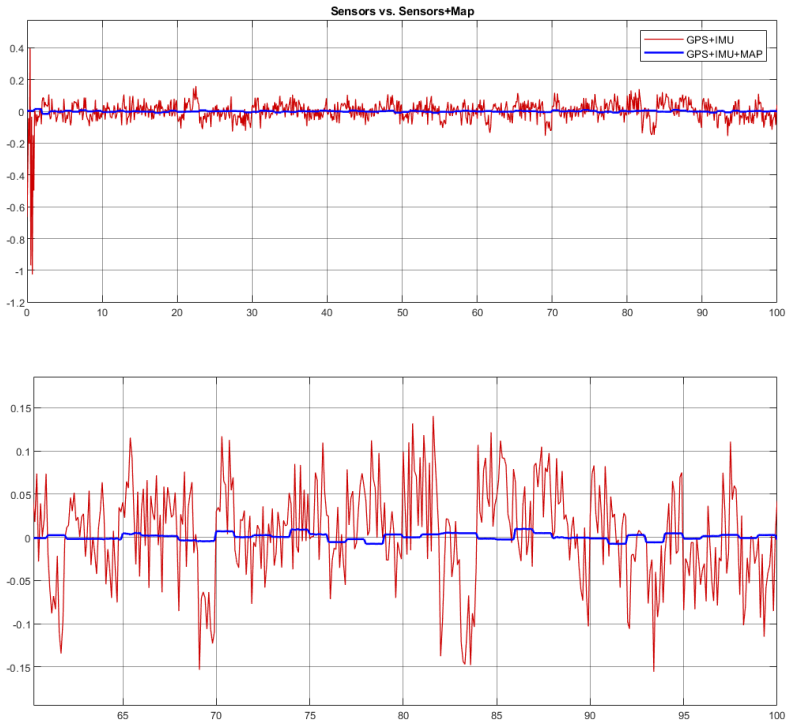


Figure 5.7: Comparison among estimation error when only on board sensors are exploited (red line) or when they are combined with low-complexity maps (blue line). In the second figure the same plot with more evidence on the behaviour of the errors in an advanced state of the simulation.

Table 5.6: GPS+IMU and GPS+IMU+distance and MAP comparison in the real environment

	e_{avg}	e_{max}	σ
GPS+IMU	0.13 [m]	0.68 [m]	0.21 [m]
GPS+IMU+ distance & MAP	0.01 [m]	0.02 [m]	0.01 [m]

adoption of four additional sensors. Indeed, in this case, the behaviour of the error is similar to the one obtained with four sensors, that still the optimal number of ultrasound sensors to equip on the UGV.
A further comparison is highlighted in Figure 5.10: the upper plot shows the error obtained when one distance from only one row is measured, while, in the lower plot, both the rows at the sides of the UGV are considered in the measurement process, exploiting data provided by two sensors for each side

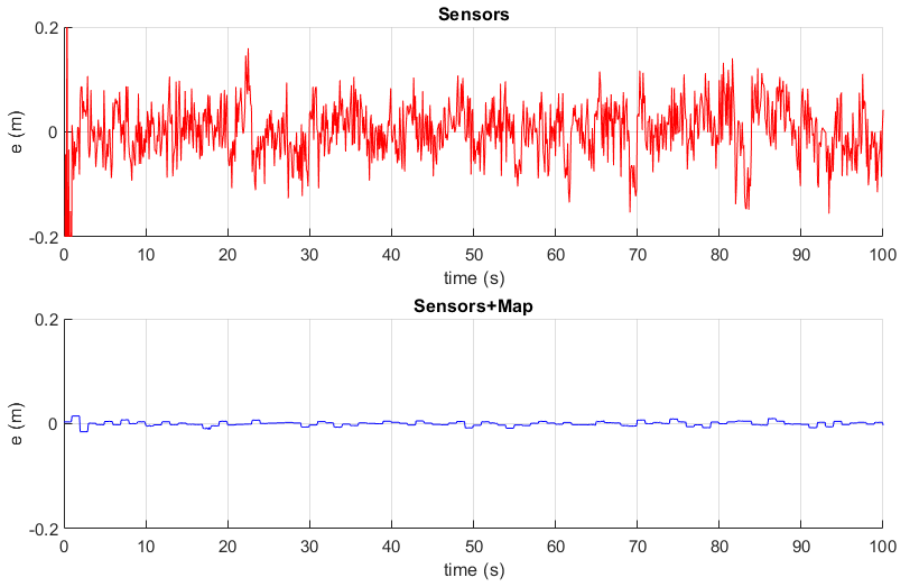


Figure 5.8: In the upper figure the error when only on board sensors are exploited, in the lower one the error when they are combined with low-complexity maps.

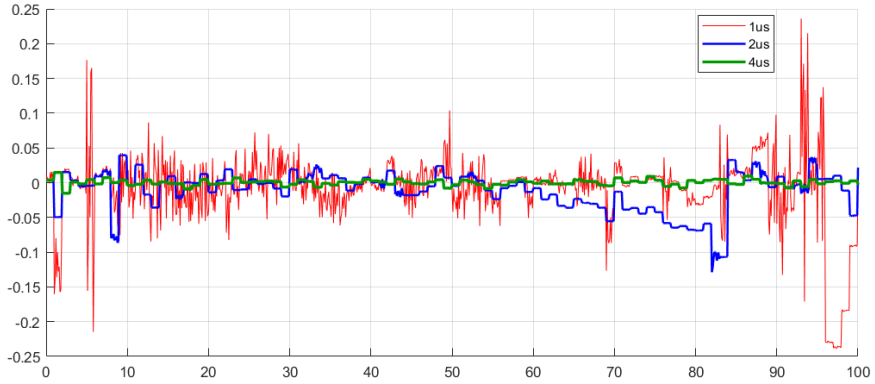


Figure 5.9: Error obtained with one ultrasound sensor on the right side of the UGV (red line), two ultrasound sensors, one for each side (blue line) and four ultrasound sensors, two for each side (green line).

of the UGV. It can be easily seen the obtained improvement: considering both the rows the anomalies caused by the effect of map line-string corners are notably reduced, gaining a lower, linear error.
Finally, in Figure 5.11, the benefits of the proposed filtering approach applied

Table 5.7: Results obtained with different numbers of sensors mounted onboard the UGV.

	e_{avg}	e_{max}	σ
1 sensor	0.03 [m]	0.23 [m]	0.05 [m]
2 sensors	0.02 [m]	0.12 [m]	0.02 [m]
4 sensors	0.01 [m]	0.02 [m]	0.01 [m]
8 sensors	0.01 [m]	0.02 [m]	0.01 [m]

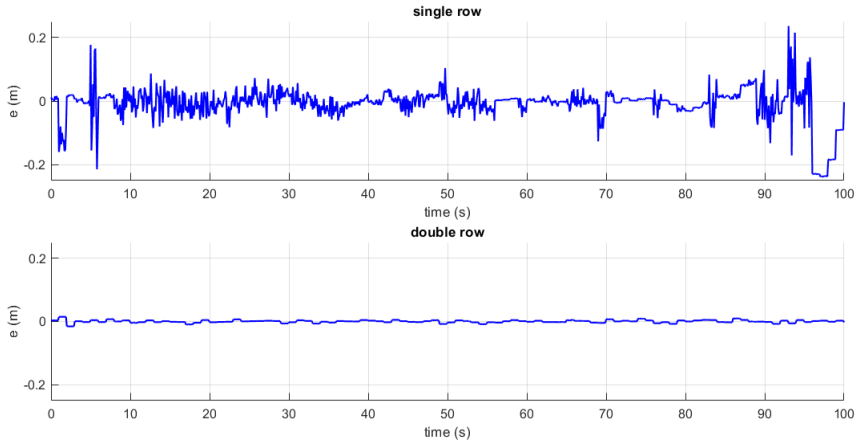


Figure 5.10: In the upper figure the error when only the distance from one row is measured, in the lower one the error when distances from both the rows at the sides of the UGV are measured.

to the selected vineyard are shown. In this example, a comparison between predictions and uncertainties propagation is proposed. In particular, a first estimation is obtained exploiting the model of the system (black dotted line), then, the uncertainty is reduced and approximated with the minimum volume ellipsoid (red line), when the data from the map (green linestring) are fused with those measured by the four ultrasonic sensors.

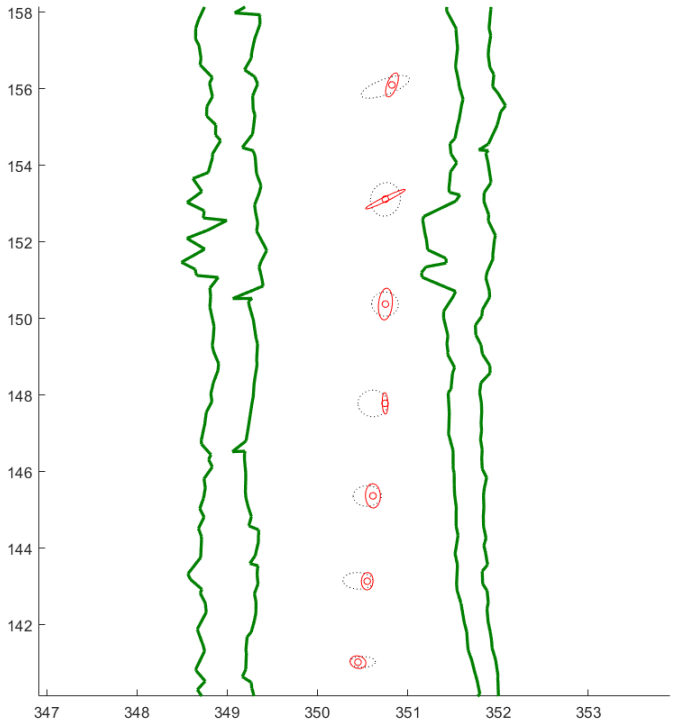


Figure 5.11: Evolution of the estimation ellipsoid (black dotted line) and the minimum volume ellipsoid (red line) when data from sensors are fused with the georeferenced information provided by the simplified map of the vine rows (green linestrings).

Chapter 6

Conclusions and future works

In this thesis, an innovative approach is proposed, based on the Kalman filter and the so-called ellipsoid method, to fuse data provided by the on-board sensors with low-complexity, georeferenced maps, thus improving the estimation of the UGV location when operated inside the crops.
The main motivation for a filter of this kind is the need for precise estimation of the position of autonomous vehicles in environments like crop-rows, due to not reliable or not available GPS signal and the tight space where the robot is moving. Employing a map of the environment where the UGV is moving in cooperation with exteroceptive sensors is a suitable way to prevent this problem and successfully exploit available data from sensors, to obtain an accurate estimation. However, the weight of the introduced map needs to be limited, in order to avoid algorithms with high computational times that make the filter to be inapplicable in an online localization process. Then, also the procedures involved in the filter need to be optimized to avoid delays in the estimation at each time step.
The effectiveness of this scheme has been validated into a simulation setup, considering a 4 wheel steering electric UGV operated within a Nebbiolo vineyard and equipped with ultrasonic sensors, GPS, and IMU board. The obtained results highlight the efficiency of the low-complexity georeferenced map, exploited in cooperation with ultrasound sensors, that increase the accuracy of the position estimation. In particular, the performed tests highlight the effectiveness of this filter for position estimation in the direction perpendicular to the rows, gaining errors lower than 1 cm. In the longitudinal direction (parallel with respect to the rows) the estimation precision can be

lower, with an average error of 3/4 cm. Hence, this method requires a precise initial estimation on some variables that are not involved in the sensors-map data elaboration: for example, a precise orientation is fundamental to provide the correct row selection in the process and obtain a correct position estimation at the end of the algorithm.
About its implementation, the ellipsoid method is one of the possible procedures to exploit the distance measure to get a localization of the UGV, but other possible solutions can be studied, tested and employed to this aim. Moreover, distance sensors are only one of the many types of exteroceptive sensors which measures can be fused with the map information. Indeed, the use of additional measurements can further improve the estimation of the position and provides also information on other state variables.
Then, further methods can be studied to improve the position in the longitudinal direction that actually presents some inaccuracies. This can be done, for example, by exploiting other types of sensors, like lidars or simple vision systems, that identify points of interest in this direction and use them with the map in a similar way to the one described in this work. Otherwise, further ultrasound sensors can be added to the UGV configuration, having different orientations with respect to the one of the CoM, providing as a result more information in the longitudinal direction. However, all these possible solutions increase the complexity of the final system configuration.
Until now, the proposed filter has been developed to be used for a ground vehicle. Indeed, the 3D map is elaborated into the 2D space to get the slice of the rows in the map at height h_{ref}, involved in the measurements of the distance sensors. In possible future works, the filter can be improved to work also into the 3D space, allowing it to be employed in more complex vehicles and scenarios. For example, in the case of an UAV that operates in a field, where the ultrasound sensors measurements depend on the vertical position and roll, pitch, and yaw values.
In conclusion, this thesis highlights the benefits of exploiting low-complexity maps and exteroceptive sensors, allowing the vehicle to reach full autonomy and a high level of accuracy in its operations, also in complex scenarios. Moreover, it promotes a new approach for filters and sensors fusion that take advantage of external resources (map, distance sensors) and different kind of internal sensors, providing more robust property estimates and leading to an increased adoptability of sensor-based crop management.

Bibliography

[1] D. Rose and J. Chilvers, "Agriculture 4.0: responsible innovation in an era of smart farming," *Frontiers in Sustainable Food Systems*, vol. 2, p. 87, 2018.

[2] M. Mammarella, L. Comba, A. Biglia, F. Dabbene, and P. Gay, "Cooperative agricultural operations of aerial and ground unmanned vehicles," in *2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*. IEEE, 2020, pp. 224–229.

[3] S. Eckelmann, M. Frommel, C. Haut, and T. Trautmann, "Position determination based on sensor data fusion with digital maps and 3D point cloud data," *2018 15th Workshop on Positioning, Navigation and Communications, WPNC 2018*, 2018.

[4] A. Schindler, "Vehicle self-localization with high-precision digital maps," *IEEE Intelligent Vehicles Symposium, Proceedings*, no. Iv, pp. 141–146, 2013.

[5] D. Chen and G. X. Gao, "Probabilistic graphical fusion of LiDAR, GPS, and 3D building maps for urban UAV navigation," *Navigation, Journal of the Institute of Navigation*, vol. 66, no. 1, pp. 151–168, 2019.

[6] D. Gruyer, R. Belaroussi, and M. Revilloud, "Map-aided localization with lateral perception," *IEEE Intelligent Vehicles Symposium, Proceedings*, no. Iv, pp. 674–680, 2014.

[7] D. Obradovic, H. Lenz, and M. Schupfner, "Fusion of map and sensor data in a modern car navigation system," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 45, no. 1-2, pp. 111–122, 2006.

[8] A. Miranda-Fuentes, J. Llorens, J. L. Gamarra-Diezma, J. A. Gil-Ribes, and E. Gil, "Towards an optimized method of olive tree crown volume measurement," *Sensors*, vol. 15, no. 2, pp. 3671–3687, 2015.

[9] J. Mack, C. Lenz, J. Teutrine, and V. Steinhage, "High-precision 3d detection and reconstruction of grapes from laser range data for efficient

phenotyping based on supervised learning," *Computers and Electronics in Agriculture*, vol. 135, pp. 300–311, 2017.

[10] I. C. Condotta, T. M. Brown-Brandl, S. K. Pitla, J. P. Stinn, and K. O. Silva-Miranda, "Evaluation of low-cost depth cameras for agricultural applications," *Computers and Electronics in Agriculture*, vol. 173, p. 105394, 2020.

[11] A. Feng, J. Zhou, E. D. Vories, K. A. Sudduth, and M. Zhang, "Yield estimation in cotton using uav-based multi-sensor imagery," *Biosystems Engineering*, vol. 193, pp. 101–114, 2020.

[12] J. Wijesingha, T. Moeckel, F. Hensgen, and M. Wachendorf, "Evaluation of 3d point cloud-based models for the prediction of grassland biomass," *International Journal of Applied Earth Observation and Geoinformation*, vol. 78, pp. 352–359, 2019.

[13] L. Comba, A. Biglia, D. R. Aimonino, and P. Gay, "Unsupervised detection of vineyards by 3d point-cloud uav photogrammetry for precision agriculture," *Computers and Electronics in Agriculture*, vol. 155, pp. 84–95, 2018.

[14] A. K. Mortensen, A. Bender, B. Whelan, M. M. Barbour, S. Sukkarieh, H. Karstoft, and R. Gislum, "Segmentation of lettuce in coloured 3d point clouds for fresh weight estimation," *Computers and Electronics in Agriculture*, vol. 154, pp. 373–381, 2018.

[15] L. Serazetdinova, J. Garratt, A. Baylis, S. Stergiadis, M. Collison, and S. Davis, "How should we turn data into decisions in agrifood?" *Journal of the Science of Food and Agriculture*, vol. 99, no. 7, pp. 3213–3219, 2019.

[16] W. Winterhalter, F. Fleckenstein, C. Dornhege, and W. Burgard, "Localization for precision navigation in agricultural fields—Beyond crop row following," *Journal of Field Robotics*, vol. 38, no. 3, pp. 429–451, 2021.

[17] L. Comba, S. Zaman, A. Biglia, D. R. Aimonino, F. Dabbene, and P. Gay, "Semantic interpretation and complexity reduction of 3D point clouds of vineyards (In press)," *Biosystems Engineering*, 2020.

[18] V. I., R. A. Viscarra Rossel, K. A., and P. Schulze, "Sensor Fusion for Precision Agriculture," *Sensor Fusion - Foundation and Applications*, no. May 2014, 2011.

[19] E. Aguirre, A. González, and R. Muñoz-Salinas, "Mobile robot map-based localization using approximate locations and the extended kalman filter," *10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2004)*,

pp. 191–198, 2004.

[20] M. M. Atia and S. L. Waslander, "Map-aided adaptive GNSS/IMU sensor fusion scheme for robust urban navigation," *Measurement: Journal of the International Measurement Confederation*, vol. 131, pp. 615–627, 2019.

[21] K. Jo, K. Chu, and M. Sunwoo, "GPS-bias correction for precise localization of autonomous vehicles," *IEEE Intelligent Vehicles Symposium, Proceedings*, no. Iv, pp. 636–641, 2013.

[22] I. Miller and M. Campbell, "Particle filtering for map-aided localization in sparse GPS environments," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1834–1841, 2008.

[23] T. Oksanen, M. Linja, and A. Visala, "Low-cost positioning system for agricultural vehicles," in *2005 international symposium on computational intelligence in robotics and automation.* IEEE, 2005, pp. 297–302.

[24] R. G. Bland, D. Goldfarb, and M. J. Todd, "Feature Article - The Ellipsoid Method : A Survey," *Operations Research 29(6):1039-1091*, vol. 29, no. 6, pp. 1039–1091, 1981.

[25] F. C. Schweppe, "Recursive state estimation: Unknown but bounded errors and system inputs," *6th Symposium on Adaptive Processes, SAP 1967*, pp. 102–107, 1967.

[26] E. Fogle and Y. F. Huang, "On the Value of Information in System Identification-Bounded Noise Case," *Automatica, Vol. 18, No. 2, pp. 229–238. 1982*, vol. 18, no. 2, pp. 229—-238, 1982.

[27] J. B. G. Frenk, J. Gromicho, and S. Zhang, "A deep cut ellipsoid algorithm for convex programming: Theory and applications," *Mathematical Programming*, vol. 63, no. 1, pp. 83–108, 1994.

[28] L. Ros, A. Sabater, and F. Thomas, "An Ellipsoidal Calculus based on propagation and fusion," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 32, no. 4, pp. 430–442, 2002.

[29] A. Liao and M. J. Todd, "The ellipsoid algorithm using parallel cuts," *Computational Optimization and Applications*, vol. 2, no. 4, pp. 299–316, 1993.

[30] S. Bonadies and S. A. Gadsden, "An overview of autonomous crop row navigation strategies for unmanned ground vehicles," *Engineering in Agriculture, Environment and Food*, vol. 12, no. 1, pp. 24–31, 2019.

[31] P. Gonzalez-De-Santos, R. Fernández, D. Sepúlveda, E. Navas, and M. Armada, "Unmanned Ground Vehicles for Smart Farms," *Agronomy - Climate Change and Food Security*, pp. 1–23, 2020.

[32] S. Bonadies, A. Lefcourt, and S. A. Gadsden, "A survey of unmanned ground vehicles with applications to agricultural and environmental sensing," *Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping*, vol. 9866, no. May 2016, p. 98660Q, 2016.

[33] W. Rahiman and Z. Zainal, "An overview of development GPS navigation for autonomous car," *Proceedings of the 2013 IEEE 8th Conference on Industrial Electronics and Applications, ICIEA 2013*, pp. 1112–1118, 2013.

[34] https://www.gps.gov/applications/agriculture/.

[35] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, "Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications," *International Journal of Signal Processing Systems*, vol. 1, no. 2, pp. 256–262, 2013.

[36] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[37] G. A. Demetriou, "A survey of sensors for localization of unmanned ground vehicles (ugvs)." in *IC-AI*. Citeseer, 2006, pp. 659–668.

[38] J. N. Yasin, S. A. Mohamed, M.-H. Haghbayan, J. Heikkonen, H. Tenhunen, and J. Plosila, "Low-cost ultrasonic based object detection and collision avoidance method for autonomous robots," *International Journal of Information Technology*, vol. 13, no. 1, pp. 97–107, 2021.

[39] M. Guanglei and P. Haibing, "The application of ultrasonic sensor in the obstacle avoidance of quad-rotor uav," in *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*. IEEE, 2016, pp. 976–981.

[40] N. Shalal, T. Low, C. McCarthy, and N. Hancock, "A review of autonomous navigation systems in agricultural environments," *Innovative Agricultural Technologies for a Sustainable Future*, pp. 1–16, 2013.

[41] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," *In Practice*, vol. 7, no. 1, pp. 1–16, 2006.

[42] S. Thrun, "Particle filters in robotics." in *UAI*, vol. 2. Citeseer, 2002, pp. 511–518.

[43] N. Y. Ko and T. G. Kim, "Comparison of Kalman filter and particle filter used for localization of an underwater vehicle," *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence, URAI 2012*, no. Urai, pp. 350–352, 2012.

[44] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering, Transactions of the ASME*, vol. 82, no. 1, pp. 35–45, 1960.

[45] G. J. Bierman and C. L. Thornton, "Numerical comparison of kalman filter algorithms: Orbit determination case study," *Automatica*, vol. 13, no. 1, pp. 23–35, 1977.

[46] R. Zhan and J. Wan, "Iterated unscented kalman filter for passive target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 3, pp. 1155–1163, 2007.

[47] A. Fakharian, T. Gustafsson, and M. Mehrfam, "Adaptive Kalman filtering based navigation: An IMU/GPS integration approach," *2011 International Conference on Networking, Sensing and Control, ICNSC 2011*, no. April, pp. 181–185, 2011.

[48] Q. Li, R. Li, K. Ji, and W. Dai, "Kalman filter and its application," *Proceedings - 8th International Conference on Intelligent Networks and Intelligent Systems, ICINIS 2015*, pp. 74–77, 2016.

[49] Y. E. Hamouda and M. M. Msallam, "Smart heterogeneous precision agriculture using wireless sensor network based on extended Kalman filter," *Neural Computing and Applications*, vol. 31, no. 9, pp. 5653–5669, 2019.

[50] J. Gomez-Gil, R. Ruiz-Gonzalez, S. Alonso-Garcia, and F. J. Gomez-Gil, "A Kalman filter implementation for precision improvement in Low-Cost GPS positioning of tractors," *Sensors (Switzerland)*, vol. 13, no. 11, pp. 15 307–15 323, 2013.

[51] T. Hague, J. A. Marchant, and N. D. Tillett, "Ground based sensing systems for autonomous agricultural vehicles," *Computers and Electronics in Agriculture*, vol. 25, no. 1-2, pp. 11–28, 2000.

[52] V. Spruyt, "A geometric interpretation of the covariance matrix," pp. 1–21, 2014. [Online]. Available: http://www.visiondummy.com/2014/04/geometric-interpretation-covariance-matrix/