A PRACTICAL LOCALIZATION SYSTEM FOR ORCHARD VEHICLES

Gustavo Freitas*, Ji Zhang†, Jacqueline Libby†, Bradley Hamner†, Marcel Bergerman†, George Kantor†

*Universidade Federal do Rio de Janeiro/COPPE, Brasil

†Robotics Institute, Carnegie Mellon University, EUA

Emails: gfreitas@coep.ufrj.br, zhangji@andrew.cmu.edu, jlibby@cmu.edu, bradley.hamner@gmail.com, marcel@cmu.edu, kantor@ri.cmu.edu

Abstract— This paper addresses the design, development and field-testing of a localization system for agricultural vehicles. The Autonomous Prime Movers (APMs) are electrical vehicles designed to operate in specialty crops, more specifically in orchards. In order to accomplish geo-referenced tasks inside the crops, the vehicles must know their pose with sub-metric accuracy. One of our requirements is that the localization system shouldn't add to the vehicle's hardware cost, so as to keep the acquisition cost to growers as low as possible. Therefore, we confine ourselves to solutions that use only the sensor suite already installed on the robot for navigation - in our case, laser scanners and steering and wheel encoders. The developed localization methodology employs an Extended Kalman Filter. The APM pose is predicted using encoder odometry and updated with point and line features detected with the laser. Tests conducted at our experimental orchard-like environment in Pittsburgh and actual apple orchards in Pennsylvania and Washington states indicate that the localization system is able to estimate the vehicle pose with sub-metric accuracy.

Keywords— Autonomous Agricultural Vehicles, GPS-Free Localization, Extended Kalman Filter.

Resumo— Este artigo apresenta o projeto, implementação e testes de campo de um sistema de localização para veículos agrícolas. Os *Autonomous Prime Movers* (APMs) são veículos elétricos projetados para operações em pomares e viveiros. A fim de realizar tarefas geo-referenciadas dentro da plantação, os APMs devem conhecer sua posição com precisão sub-métrica. O sistema de localização não deve adicionar custos ao veículo, de forma a facilitar sua aquisição por produtores agrícolas. Desse modo, o sistema desenvolvido utiliza apenas os sensores já disponíveis no veículo para navegação - no caso, *laser scanners* e *encoders*, integrados utilizando um filtro de Kalman estendido (EKF). Testes realizados em pomares comerciais e experimentais demonstram que o sistema de localização é capaz de estimar a posição do veículo de maneira eficiente, e indicam condições de operação onde o sistema apresenta precisão sub-métrica.

Palavras-chave— Veículos Agrícolas Autônomos, Localização sem GPS, Extended Kalman Filter.

1 Introduction

Specialty crops are defined in the US as fruits and vegetables, tree nuts, dried fruits and nursery crops including floriculture. Their market value in 2007 neared U$ 50 billion, or almost 17% of the entire US agricultural market value. In that year, the five largest fruit and tree nut crops (grapes, apples, almonds, strawberries and oranges) brought US$ 11 billion in cash receipts to farmers. Fruit and tree nut production alone generate about 13% of all farm cash receipts in the country (Singh et al., 2010).

There is a great opportunity to introduce automation solutions into tree fruit production to lower labor costs, smooth out labor requirements, and increase production efficiency. This opportunity is compounded by the introduction of high-density planting architectures in the past twenty years, where fruit grows along "walls" formed by the branches of trees just four to six feet apart. Autonomous vehicles driving down along these fruit walls can mow and spray, as well as carry workers pruning, thinning, performing tree maintenance, and harvesting.

The ability to accurately determine the position is a fundamental component to increase productivity in many agriculture applications. Geo-referenced data about the crop can reduce costs by limiting the use of valuable resources. Most of the localization systems used in agriculture are based on GPS (O'Connor et al., 1996). A standard GPS provides accuracy between $2 - 20$ m. This precision may be enough for program crops such as corn, soy, and wheat. However, considering specialty crops adopting high density architecture, a sub-metric localization precision is required. Existing GPS based solutions for high accuracy in field are considerably expensive, and may be prohibitive for small and medium farmers. Other problem related to GPS is the loss of connection, caused when the line-of-sight to the GPS satellites is occluded by trees and other structures.

For the past four years we have been developing a family of vehicles for agricultural automation, which we call Autonomous Prime Movers, or APMs (Figure 1). The current APMs are capable of autonomously driving between a row of trees, turning at the end of the aisle and entering the next one. Row following is conducted at the center of the aisle (e.g., for sensing or mowing) or at a predefined distance from the trunk line (e.g., for pruning, thinning, or spraying). To be affordable to growers, the APMs do not carry a high-accuracy GPS system. Rather, they navigate using only laser rangefinders and steering and wheel encoders.

The APMs must accomplish geo-referenced tasks inside the orchards. Associated to localization, all data acquired by the vehicles sensors can be incorporated into maps composing a geographic information system (GIS) framework. By knowing their pose, the APMs can execute precision missions like spraying and applying inputs only in the required trees.

The role of positioning becomes even more critical in the case of automated or partially-automated vehi-

Figura 1: Two versions of vehicles in the Autonomous Prime Mover family. (Left) This GPS-free vehicle is equipped with a platform from where workers prune, thin, and train trees. Timed trails conducted by Pennsylvania State University indicate an efficiency increase of up to 58% when workers use this APM versus a ladder. (Right) This vehicle is equipped with a variety of laser scanners and cameras for, among other tasks, automatic disease and insect detection and crop yield estimation. It is also capable of mowing and spraying entire orchard blocks autonomously.

cles. While performance when driving between rows is satisfactory, the current navigation system employs a pure-pursuit approach (Coulter, 1992), and does not rely on the vehicle's pose inside the orchard. This information could improve the navigation reliability and safety.

Clearly, this is an important requirement that must be addressed before APMs can become part of the tree fruit grower toolbox. Additionally, any localization system must not add to the hardware cost of the vehicle, lest cost issues increase the adoption barrier. Finally, the system must reliably estimate the vehicle pose inside the orchard with sub-metric accuracy.

In this paper we address localization for agricultural vehicles driving inside orchards, using only the sensing suite already in place for row following - namely, laser scanners and steering and wheel encoders. Our localization methodology is based on an Extended Kalman Filter (EKF), that predicts the APM pose with encoder odometry and updates the estimative using laser data. The laser detects point and line features from the orchard architecture, that are compared to a map obtained *a priori* in order to correct the vehicle pose. Tests conducted at our experimental orchard-like environment in Pittsburgh and actual apple orchards in Pennsylvania and Washington states indicate that the localization system presents satisfactory performance, obtaining consistent and accurate results during the experiments. We present online results illustrating the system functionality on over 5 km operations. Stretch tests indicate orchard driving characteristics where the system is able to provide the required precision.

The paper is structured as follows. Section II presents a review of related work, with special focus on pose estimation for ground vehicles operating in orchard-like agricultural environments. Section III describes the autonomous orchard vehicle used as a platform in which the localization system was implemented and tested. In Section IV we present the method per se, and in Section V the localization system implementation. The results of extensive field experiments conducted in orchards are presented in Section VI. We conclude the paper in Section VII indicating avenues for future work.

2 Literature Review

In past years several results about ground vehicles outdoor localization have been presented (Borenstein et al., 1996). Considering the adopted sensor suite, localization systems use encoder, GPS (Buehler et al., 2007), inertial unit (Dudek and Jenkin, 2008), cameras (Se et al., 2002; Cauchois et al., 2003), sonar (Xiang et al., 2003) and laser range finder (Guivant, Nebot and Baiker, 2000; Hahnel et al., 2003). The information provided can be integrated using multi-sensor fusion data techniques.

Odometry provides relative information between two consecutive locations. Data can be obtained from the vehicle's encoders, characterizing the motion during a period. It can be used to recover the vehicle past locations with respect to the initial position. However, odometry measurements are noisy and corrupted by wheel slippage, and dead reckoning inevitably diverges from the truth (Thrun and Leonard, 2008).

Range and bearing lasers have become some of the most attractive sensors for localization purposes due to their accuracy and low cost relative to the price of the vehicle (Guivant, Nebot and Baiker, 2000). To obtain the vehicle pose, lasers can be employed to identify and localize landmarks. For environments with locally distinguishable features, landmark-based maps have been extensively used (Burgard and Hebert, 2008).

Lasers can also be employed for feature detection. Several authors use use line feature for pose estimation and navigation (Sack and Burgard, 2004; Lu and Milios, 1997; Leonard et al., 2003). Most of them correspond to indoor application and do not deal with natural environments. One example of outdoor employment of line features is presented in (Barawid et al., 2007), where agricultural autonomous guidance leveraged the crop structure. The difference between that work and ours is that the problem here consists on extracting linear features from vegetation.

For data fusion, it is necessary to integrate information provided by different sensors to estimate the vehicle's position. Considering the errors associated with the measurements, probabilistic approaches have been employed to deal with uncertainty in perception systems. A common option for pose estimation is the Extended Kalman Filter (Thrun et al., 2005).

Regarding localization methods, the EKF has been already employed for outdoor localization in semi-structured environments. For example, Thrapp et al. fuse Odometry and GPS data with an EKF in a robust localization method developed for an outdoor robot that gives tours of the Rice University campus (Thrapp et al., 2001). Guivant et al. use an EKF to integrate wheel encoder and laser data for Simultaneous Localization and Mapping (SLAM) in a park setting, using tree trunks as land marks (Guivant, Nebot and Durrant-Whyte, 2000).

Our localization system for orchard vehicles has been developed since 2010, and the initial implementation and results are described in (Libby and Kan-

tor, 2011). The original system presented acceptable accuracy and was useful to validate the proposed localization methodology. The results presented in this paper were obtained using the localization system new implementation, described in Section 5. The obtained error curves illustrate the localization accuracy and also point to system limitations.

3 Autonomous Orchard Vehicle

The base vehicle used in this work is the APM "Laurel", on the right in Figure 1. It is based on the Toro eWorkman MDE electric utility vehicle, retrofitted to operate either in manual or drive-by-wire modes. The base retrofitting process consisted on installing a steering motor, brake motor, motor drivers and steering and wheel encoders. Laurel is a research vehicle, where we implement and test orchard navigation technologies before they are ported to other vehicles in the APM family. It is important to note that, while Laurel is equipped with a high-accuracy Applanix POS 220 LV INS/GPS system, we do not use it for the pose estimation proposed here, since the other APMs do not have such a system onboard. The Applanix data is used during the Extended Kalman Filter setup, including the mapping process. It also provides ground truth in order to obtain the localization system error. This does not constitute an operation limitation, and in Section VII we discuss the localization system deployment without the Applanix.

The relevant sensors for this work are: steering and wheel encoders with angular resolution of $0.38°$/tick and linear resolution of 2.33×10^{-5} m/tick; one SICK LMS 291 laser scanner with $180°$ field-of-view and $1°$ resolution and maximum scanning range of 80 m; and a SICK LMS 111 with $270°$ field of view and $0.5°$ resolution, providing scanning range up to 30 m. The encoder odometry is computed at 45 Hz; the SICK LMS 111 is used for autonomous navigation, detecting apple trees and extracting line feature from the tree rows at 15 Hz (Hamner et al., 2011); the lower SICK LMS 291 laser looks for reflective tape located at the end of tree rows at 75 Hz.

The main onboard processor is a rugged, waterproof, industrial computer with an Intel Core 2 Duo 1.6 GHz CPU with 4GB DDR2 DRAM from Small PC. The localization software runs on Ubuntu Linux, with the message passing provided by Willow Garage's Robot Operating System (ROS). ROS contains drives for an increasing number of sensors, and offers a mechanism for changing information between the different nodes - processes that perform computation - composing the system. More information is available at the ROS web page (`http://www.ros.org/`).

4 Localization Methodology

The APM pose is estimated using an Extended Kalman Filter. The orchard terrain is assumed locally flat, therefore localization is simplified to a $2D$ problem. The APM pose is defined as $x = [x_r, y_r, \theta_r]^T$, where x_r, y_r represent the vehicle's planar position and θ_r its orientation with respect to an initial predefined configuration.

The Kalman Filter uses encoder measurements to predict the vehicle's position, and corrects the estimative using laser data. Laser range finders detect point and line features to update the APM pose. The features are compared to the terrain map obtained before operation. The system's simplified functional architecture is illustrated in Figure 2.

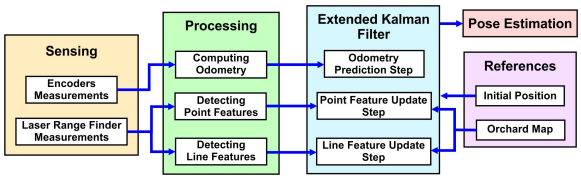

Figura 2: Functional architecture of the localization system. The Extended Kalman Filter uses data from encoders and laser rangefinder to estimate the vehicle's pose. The software uses as reference the initial position and an orchard map obtained off-line.

The proposed solution assumes the vehicle starts in a known initial position and orientation with respect to a previously-built reference map that contains the tree rows' ending positions. The current procedure used to create the map consists on driving the vehicle around the orchard and combining the measurements from the Applanix and lower laser to obtain the position of landmark reflective tape placed on the row ends (Figure 3).

Figura 3: (Left) Apple orchard with reflective tape installed on the row ends for mapping purposes. (Right) An example of the resulting orchard map, used for vehicle localization.

The localization methodology proposed here can be extended to other types of autonomous agricultural vehicles traversing crops arranged in rows in locally flat terrain. The requirements are availability of a local map, laser rangefinder, steering and wheel encoders, and sufficient computing infrastructure.

4.1 Extended Kalman Filter

Kalman filtering is a recursive method to estimate the state of a dynamic system in the presence of noise (Choset et al., 2005). The filter estimates the state vector as a Gaussian probability density function with mean $\hat{x}$ and covariance P.

The objective of the Kalman filter is to estimate x at time instant t considering the previous estimates and the system input $u(t)$ and output $y(t)$. It can be employed in systems represented by:

$$x(t+1) = f(x(t), u(t), t) + v(t) \quad (1)$$

$$y(t) = h(x(t), t) + w(t) \quad (2)$$

where $x \in \mathbb{R}^n$ denotes the full system state, $y \in \mathbb{R}^p$ is the system output containing sensor data, $v \in \mathbb{R}^n$ is the process noise represented by a white Gaussian noise with zero mean and covariance matrix $V(t)$, and w is the measurement noise with zero mean white noise and covariance matrix $W(t)$.

It is assumed that

$$f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{Z}^+ \rightarrow \mathbb{R}^n$$

and

$$h : \mathbb{R}^n \times \mathbb{Z}^+ \rightarrow \mathbb{R}^p$$

are continuously differentiable in $x(t)$. To deal with nonlinear equations, the EKF uses linearized equations about the current estimated state of the system.

The state estimation follows a two-step process. First it generates the state prediction $\hat{x}(t+1|t)$ employing the current estimate $\hat{x}(t|t)$ and system input $u(t)$ in Equation (1). Then the prediction is corrected using the system output $y(t+1)$ defined by Equation (2) to generate the next estimate $\hat{x}(t+1|t+1)$. The two steps are commonly defined as prediction and update, respectively.

The APM localization system uses odometry data from steering and wheel encoders to calculate the APM pose during the prediction step. The system has two update steps, correcting the vehicle's pose using point and line feature detections.

In what follows, the EKF operation is illustrated using playback data acquired during experiments at our half-acre ornamental tree nursery planting composed of eight tree rows. The APM was driven inside the rows, executing a trajectory with seven lines and six turns.

4.1.1 Prediction Step

The vehicle state is given by $x(t) = [x_r(t), y_r(t), \theta_r(t)]^T$ at time t. Its input is $u(t) = [u_1(t); u_2(t)]^T$, where $u_1(t)$ and $u_2(t)$ correspond to the linear and angular velocities respectively, measured using the steering and wheel encoders. The vehicle pose is predicted using the nonlinear motion system:

$$\hat{x}(t+1|t) = f(\hat{x}(t|t), u(t)) = \begin{bmatrix} u_1(t)cos(\theta_r)\Delta t + x_r(t) \\ u_1(t)sin(\theta_r)\Delta t + y_r(t) \\ u_2(t)\Delta t + \theta_r \end{bmatrix} \tag{3}$$

where Δt represents the time interval between t and $t+1$.

The state covariance P is given by:

$$P(t+1|t) = F(t)P(t|t)F(t)^T + V(t) \tag{4}$$

where

$$F(t) = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}(t|t)} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix}_{x=\hat{x}(t|t)} \tag{5}$$

The process noise covariance $V(t)$ is obtained differentiating the encoders readings and the Applanix ground truth localization. Then we compute the error between the input linear and angular velocities obtained with both sensors to estimate the covariances. In fact, two covariance matrices are considered, one for straight and one for turning motions. The angular velocity variance is higher when the vehicle is turning at the end of the row in comparison with driving straight down a row.

Figure 4 presents the pose estimated by the localization system using just odometry data. The encoder odometry errors are mainly caused by wheel slippage, that happens specially during turns. After each turning the orientation error increases, and consequently the estimate drifts from the real vehicle pose.

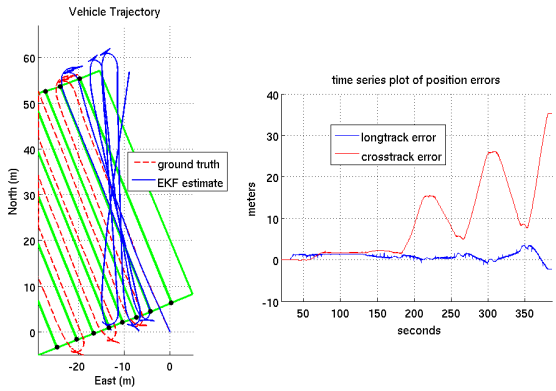

Figura 4: Trajectory and position error obtained by the localization system using odometry data only.

4.1.2 Update Step

After predicting $\hat{x}(t+1|t)$, it is possible to correct the vehicle's pose based on the system output $y(t+1)$ to generate the next estimative of the state $\hat{x}(t+1|t+1)$. The state space update equation is given by:

$$\hat{x}(t+1|t+1) = \hat{x}(t+1|t) + R\nu \tag{6}$$

and the updated error covariance is obtained with

$$P(t+1|t+1) = P(t+1|t) - RH(t+1)P(t+1|t) \tag{7}$$

where

$$\nu = y(t+1) - h(x(t+1), t+1) \tag{8}$$

$$S = H(t+1)P(t+1|t)H(t+1)^T + W(t+1) \tag{9}$$

$$R = P(t+1|t)H(t+1)^T S^{-1} \tag{10}$$

and

$$H(t+1) = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}(t+1|t)} \tag{11}$$

Matrix ν corresponds to the innovation error, and matrix R works as a weighting factor between the predicted estimate accuracy and measurement noise.

- Point Feature Update

The localization system uses the beginning and ending of tree rows as landmarks, represented in the map by planar positions $p_i = [x_{pi}, y_{pi}]^T, i = 1, 2, ..., n_p$. During operations in orchards, the APM laser range finder is able to detect point features when approaching the rows endings. The laser measures the range and bearing to the landmarks, and each measure corresponds to $y_i = [r, \theta]^T$. The output equation for this system is given by Equation 2, calculating $y(t+1)$ with the following function:

$$y(t+1) = \begin{bmatrix} \sqrt{(x_r(t+1) - x_{pi})^2 + (y_r(t+1) - y_{pi})^2} \\ atan2(y_r(t+1) - y_{pi}, x_r(t+1) - x_{pi}) - \theta_r(t+1) \end{bmatrix} \quad (12)$$

We associate a measured landmark to a point in the map according to the closest Mahalanobis distance, using standard chi-square gating.

After obtaining the system model, it is possible to calculate the Jacobian Matrix $H(t+1) = \frac{\partial h}{\partial x}$ using the chain rule; the result is presented in (Choset et al., 2005).

The covariance matrix W related to the laser measurements is obtained using the Applanix as ground truth. For that, Equation (12) is calculated considering the vehicle's actual poses x obtained by the Applanix and also the estimated pose $\hat{x}$ calculated with the localization system. Thus it is possible to compute the innovation covariance, and then obtain W.

Figure 5 presents results obtained using odometry and point feature detection. Whenever the laser identifies a landmark, the system updates the pose. With that, the estimated position remains inside the map during the entire operation, and the crosstrack error is limited to 10 m. The problem here is data association. The estimate drifts so much that the vehicle appears to start in one aisle and end in another one. The position is corrected, however using the wrong landmark.

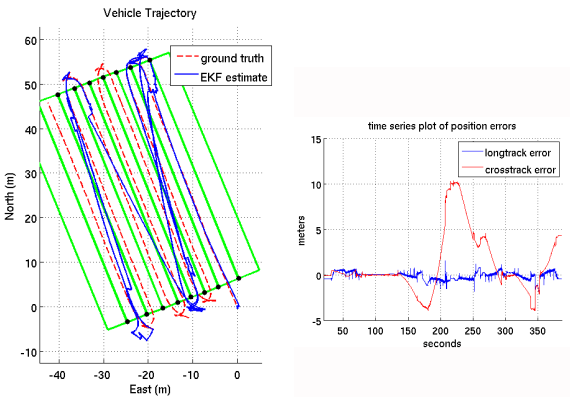

Figura 5: Trajectory and position error obtained by the localization system using odometry data and point feature detection.

- Line Feature Update

The APM accomplishes most of its tasks inside tree rows. The crop is organized in lines, whose features are employed for localization purposes. The laser obtains point clouds that represents apparent section of the canopy from the left and right rows surrounding the vehicle.

In order to fit the clouds points into line features, a particle filter was implemented (Hamner et al., 2011). The filter makes multiple guesses of where the tree rows could be, and scores each guess by how much it agrees with the laser data. High-scoring row lines are kept from one iteration to the next, therefore the detection get better over time.

The output equation for this system corresponds to lines represented in polar coordinates $y_i = [d_i, \alpha_i]^T$ with respect to the reference coordinate frame, where d_i is the perpendicular distance between line and reference frame and α_i is the angle between line and reference coordinate x axis.

Similarly to the point association correction, the idea here is to determine which line measurement is expected considering the APM current position $x(t+1)$ and the orchard map. Here data association does not represent a problem, since the pose estimate is accurate enough to identify the correct aisle the vehicle is traversing.

The matrix covariance W related to the line measurements is obtained using the Applanix as ground truth. For that $h_i(x(t+1), i)$ is calculated considering the vehicle's actual poses x obtained by the Applanix and also the estimated pose $\hat{x}$ calculated by the localization system. Again we compute the innovation covariance to obtain W.

Figure 6 shows the pose estimate obtained using odometry and the point and line features. The crosstrack and longtrack errors are smaller than 1 m during most of the operation. Part of the crosstrack error comes from the line feature detection, since the parameters from tree rows obtained using particle filter deviates due to the canopy shape. The longtrack error is mainly caused by wheel slippage. The longtrack accumulates faster than the crosstrack error, since the lateral error is constantly corrected by the line features, but the longitudinal error is corrected only at the end of the rows, when the vehicle identifies a landmark.

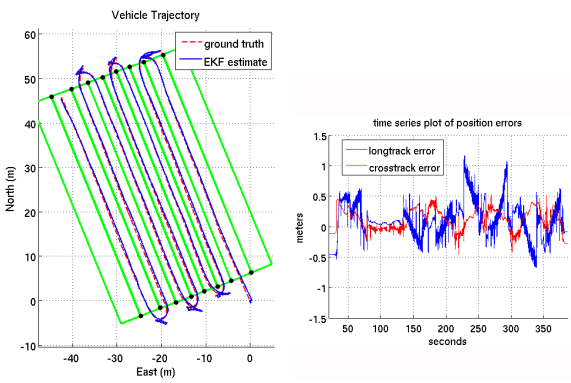

Figura 6: Trajectory and position error obtained by the localization system using odometry data and point and line feature detection.

Figure 7 presents the error distribution. Considering the crosstack error histogram, it is possible to see that the maximum error is about 0.5 m, and the localization estimate is within 30 cm of ground truth for more than 75% of the time. The 3σ (99.7% of final value) interval is 0.45 m. The maximum longtrack error is about 1.2 m, and the position estimate is within 30 cm of ground truth for more than 80% of the time. The 3σ (99.7% of the steady state) interval is smaller than 1 m.

5 Pose Estimation System

The pose estimation software implements the Extended Kalman Filter through object oriented code writ-

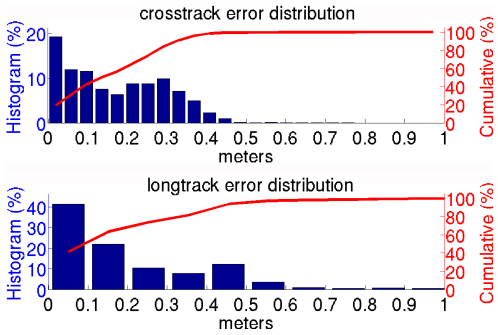

Figura 7: Crosstrack and longtrack distribution errors of the localization system compared with ground truth from Applanix. For both cases the position estimative is within 30 cm of ground truth for more than 75% of the time.

ten in C++. The entire APM control software including the localization program runs on ROS - Robot Operating System, an open-source software developed by Willow Garage. ROS is not an operating system in the traditional sense of process management and scheduling; it is a framework to manage complexity and help software developers create robot applications. ROS provides a structured communication layer above the host operating systems of a heterogenous compute cluster (Quigley et al., 2009).

Each process running in the APM control software corresponds to a ROS node. The nodes exchange information subscribing and publishing messages to the ROS topics. The APM sensors correspond to ROS nodes, transmitting measurements for the entire system through messages published into topics. The localization system is also implemented by a ROS node. The respective software subscribes to the sensor topics, receiving inputs to run the EKF. After processing the information, the node publishes the pose estimation into another specific ROS topic. With that, processes such as the autonomous driving node are able to use the pose information.

A positive aspect of the Robot Operating System is the capability of recording messages in ROS bag files (*.bag). With that, it is possible to playback messages to recreate operation conditions. This was specially useful during the localization software implementation.

The pose estimation corresponds to an object running on ROS. It starts the operation loading the orchard map obtained *a priori*. After that, the node subscribes to different sensor topics and acquires the vehicle position to initialize the Kalman Filter.

After initialization, the software enters in a polling mode running at 1 KHz. For every polling cycle, the localization node checks for new measurement messages stored in the sensor topics. If they are available, the program calls the respective EKF steps - prediction for encoder measurements and update for point and line feature detections – passing as argument the latest measurement provided by each sensor.

The localization node publishes a new pose estimation message at each polling cycle that invokes the Kalman filter. The software also creates log files, providing information to evaluate the localization software performance.

The polling mode is implemented using the command *ros::spinOnce()* inside a while loop. The command is employed for checking the ROS messages topics sporadically, according to a specified frequency. The software main algorithm is briefly described in the pseudo-code:

```
global applanixDataMsg, encoderDataMsg, pointFeatDataMsg, lineFeatDataMsg;

void treatRosApplanix(applanixDataMsg);
void treatRosEncoder(encoderDataMsg);
void treatRosPointFeat(pointFeatDataMsg);
void treatRosLineFeat(lineFeatDataMsg);

int main () {

 poseEstimation = [x, y, theta];

 loadMap();
 getInitialPosition(applanixDataMsg);

 ros::Subscriber applanixDataSubscriber => treatRosApplanix(applanixDataMsg);
 ros::Subscriber encoderDataSubscriber => treatRosEncoder(encoderDataMsg);
 ros::Subscriber pointFeatDataSubscriber => treatRosPointFeat(pointFeatDataMsg);
 ros::Subscriber lineFeatDataSubscriber => treatRosLineFeat(lineFeatDataMsg);
 ros::Publisher poseEstimationDataPublisher => poseEstimationDataMsg;

 while(true){
  ros::spinOnce();
  loop_rate.sleep(1KHz);

  if encoderDataMsg
   odometryPredictionStep(encoderDataMsg, poseEstimation);

  if pointFeatDataMsg
   pointFeatUpdateStep(pointFeatDataMsg, poseEstimation);

  if lineFeatDataMsg
   lineFeatUpdateStep(lineFeatDataMsg, poseEstimation);

  poseEstimationDataPublisher(poseEstimation);
 }
}
```

To ensure the localization reliability, some effort was done in debugging and improving the communication procedure between sensors and main process. It was noticed that the system, under specific configuration, does not process all the sensor messages. The localization system runs at 1 KHz, receiving odometry messages at 45 Hz, laser measurements for landmark detection at 75 Hz and line feature detections at 15 Hz. Considering the operation frequencies, it was expected that the pose node would be able to process all sensor measurements with out the necessity of buffering more than one message from each sensor.

During debugging procedures, however, we identified that the node loses 5% of the odometry and line feature messages. The problem is aggravated for the landmark detection, with 60 % of the messages not being processed. It was also noticed that the localization software receives the different sensor messages out of order. The ROS messaging inefficiencies influence the localization performance, causing non deterministic functionality.

The observed problems were sent by email to Willow Garage, available at the ROS mailing list (`http://bit.ly/LggVu7`). As feedback, we were informed that the time when messages buffers are serviced can vary greatly, independently from the frequency of the main node. The same answering email also confirmed that when receiving messages from different sources, ROS does not guarantee in each order you will receive/process them.

To solve the reported problems, we use vectors to store messages coming from different sensors. An algorithm looks into the vectors and selects which message should be send to the localization node. The code does not lose measurements, and also selects the messages in right time order for calling the EKF steps.

In order to process the messages in right order, the script includes a delay in the process. This is not

good for real time operations in field, because the estimated position is always delayed in relation to real one. The implemented solution considers the time delay and vehicle's last velocity to estimate the APM's current position. This is possible since accelerations during operation are small.

The APM localization system is written in C++ and runs on ROS according to the rest of the APM control software. Even so, the proposed localization methodology could be implemented in other programming languages, running on other operating systems.

6 Experimental Validation

To assess the localization system feasibility, we have accomplished several field tests in experimental and commercial apple orchards. The former is Washington State University's Sunrise Orchard in Rock Island, WA. The latter is commercial orchard Ridgetop in Fishertown, PA. At both locations we manually drove the vehicle inside the tree rows. The objective is to verify the system robustness and functionality for long time operations in real crops.

Figure 8 presents one representative results at Sunrise; Table 1 summarizes three of them. The vehicle traveled a total of $5,924$ m in blocks 9A, 9B and 9C, and its position was estimated with sub-metric precision. The mean errors range from 0.17 to 0.23 m, and all the 3σ distances are less than 1 m. The good results are in part due to the flat, dry terrain, which does not generate much wheel slippage, and short tree rows, which brings the landmarks quickly into view, reducing odometry-associated errors. All were obtained on-line as the vehicle drove in between rows.

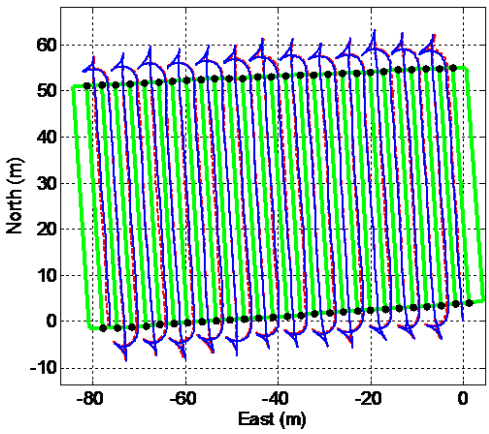

Figura 8: Trajectory obtained by the localization system at Sunrise Orchard's block 9C. The estimated position is plotted in blue, and the Applanix ground truth data is plotted in red.

Figure 9 presents the results obtained at Ridgetop Orchard. This is a much more challenging environment, with rows that are 300 m long and very steep (up to 12.5° of inclination). At Ridgetop we traversed four rows. When the vehicle is driving inside the rows, there is no correction for the longtrack error and the encoder odometry drifting accumulates. The maximum long track error is about 2 m when the vehicle is going downhill and up to 6 m when going uphill, again compatible with wheel odometry.

Tabela 1: Summary of experiments conducted at Sunrise.

Test site	Block 9A	Block 9B	Block 9C
Total distance [m]	1897.7	2135.5	1892.8
Mean longtrack error [m]	0.22	0.19	0.21
Mean crosstrack error [m]	0.23	0.17	0.22
3σ longtrack error [m]	0.91	0.65	0.98
3σ crosstrack error [m]	0.76	0.51	0.72

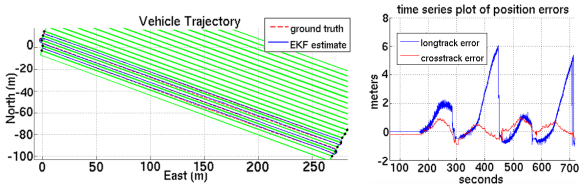

Figura 9: Trajectory and position error obtained by the localization system at Ridgetop Orchards.

7 Conclusions and Future Work

The localization system described in this paper is part of the larger goal of demonstrating an APM's feasibility to operate year-round in a commercial production environment.

Using playback data, it was possible to illustrate the localization methodology, highlighting the effects of each EKF step during the pose estimation process. Several field experiments have been accomplished with Laurel using the localization system. The system has presented satisfactory results during most part of the tests, obtaining sub-metric precision in several orchards with different characteristics.

However, due to the current sensors employed, the localization system may lose accuracy in special circumstances. This fact is illustrated with test results obtained at Ridgetop orchard, where the terrain is inclined and the tree rows are very long. When driving inside rows, the longtrack error is not corrected until the laser detects the end of row. Because of wheel slippage, the encoder odometry error can achieve values about 2% of the traveled linear distance.

One possible solution to improve accuracy consists on computing odometry using cameras, since visual odometry is not affected by slippage. We are already accomplishing tests using visual odometry, and the expected result is to enhance the localization system accuracy in one order of magnitude.

The localization methodology was developed and tested with data from two lasers, one detecting point and the other detecting line features. In an actual field deployment, only one laser scanner would have to provide data to both the navigation and localization. Also, Laurel is the only vehicle using the Applanix high accuracy localization system. The others rely only on the system presented here to obtain position estimation. The vehicles do not have GPS, requiring alternative solutions to obtain the initial position and also the orchard map.

For initialization purposes, the vehicle can start the operation in a known position. Regarding the mapping requirement, one procedure already in use consists on placing a low cost Ublox GPS on each landmark for about one hour. Then, the acquired position is corrected with RTK post-processing data from the USGS, obtaining the reflective tape's positions with 30 cm accuracy.

In the future, the mapping procedure may even be eliminated entirely and replaced with a SLAM–type method, with the vehicle simultaneously mapping and computing its localization while crossing an unknown environment. In the future we intend to compare the performance of the proposed localization methodology with SLAM-type approaches that do not require an *a priori* mapping step.

8 Acknowledgments

This work is supported by the US Department of Agriculture under the Specialty Crop Research Initiative, award number 2008-51180-04876.

Gustavo Freitas is funded by a grant from the Brazilian National Council for Research and Development (CNPq). He thanks his advisors Prof. Liu Hsu and Prof. Fernando Lizarralde from the Federal University of Rio de Janeiro, Brazil, for all the support during his PhD internship at Carnegie Mellon University.

Referências

Barawid, O. C., Jr., Mizushima, A., Ishii, K. and Noguchi, N. (2007). Development of an autonomous navigation system using a two-dimensional laser scanner in an orchard application, *Biosystems Engineering* **96**(2): 139 –149.

Borenstein, J., Everett, H. and Feng, L. (1996). *Navigating mobile robots: systems and techniques*, AK Peters, Ltd. Natick, MA, USA.

Buehler, M., Iagnemma, K. and Singh, S. (eds) (2007). *The 2005 DARPA Grand Challenge*, Springer, Berlin.

Burgard, W. and Hebert, M. (2008). World modeling, *in* B. Siciliano and O. Khatib (eds), *Springer Handbook of Robotics*, 1st edn, Springer-Verlag Ltd., pp. 853–867.

Cauchois, C., Brassart, E., Delahoche, L. and Drocourt, C. (2003). Spatial localization method with omnidirectional vision, *IEEE International Conference on Advanced Robotics*, Coimbra, pp. 287–292.

Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L. and Thrun, S. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementations*, The MIT Press.

Coulter, R. (1992). Implementation of the pure pursuit path tracking algorithm, *Technical report*, DTIC Document.

Dudek, G. and Jenkin, M. (2008). Inertial sensors, GPS, and odometry, *in* B. Siciliano and O. Khatib (eds), *Springer Handbook of Robotics*, 1st edn, Springer-Verlag Ltd., pp. 477–488.

Guivant, J., Nebot, E. and Baiker, S. (2000). Autonomous navigation and map building using laser range sensors in outdoor applications, *Journal of Robotic Systems* **17**(10): 565–583.

Guivant, J., Nebot, E. and Durrant-Whyte, H. (2000). Simultaneous localization and map building using natural features in outdoor environments, *Intelligent Autonomous Systems*, Vol. 6, pp. 581–586.

Hahnel, D., Fox, D., Burgard, W. and Thrun, S. (2003). A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, pp. 206–211.

Hamner, B., Bergerman, M. and Singh, S. (2011). Autonomous orchard vehicles for specialty crops production, *ASABE Annual International Meeting*, Louisville, Kentucky.

Leonard, J., Newman, P., Rikoski, R., Neira, J. and Tardós, J. (2003). Towards robust data association and feature modeling for concurrent mapping and localization, *Robotics Research* pp. 7–20.

Libby, J. and Kantor, G. (2011). Deployment of a point and line feature localization system for an outdoor agriculture vehicle, *IEEE International Conference on Robotics and Automation*, Shangai, pp. 1565–1570.

Lu, F. and Milios, E. (1997). Robot pose estimation in unknown environments by matching 2d range scans, *Journal of Intelligent and Robotic Systems* **18**(3): 249–275.

O'Connor, M., Bell, T., Elkaim, G. and Parkinson, B. (1996). Automatic steering of farm vehicles using GPS, *International Conference on Precision Agriculture*, Minneapolis, pp. 767–777.

Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R. and Ng, A. (2009). ROS: an open-source robot operating system, *ICRA Workshop on Open Source Software*.

Sack, D. and Burgard, W. (2004). A comparison of methods for line extraction from range data, *IFAC symposium on intelligent autonomous vehicles (IAV)*, Citeseer.

Se, S., Lowe, D. and Little, J. (2002). Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks, *The International Journal of Robotics Research* **21**(8): 735.

Singh, S., Bergerman, M., Cannons, J., Grocholsky, B. P., Hamner, B., Holguin, G., Hull, L., Jones, V., Kantor, G. A., Koselka, H., Li, G., Owen, J., Park, J., Shi, W. and Teza, J. (2010). Comprehensive Automaton for Specialty Crops: Year 1 results and lessons learned, *Journal of Intelligent Service Robotics*, pp. 245–262.

Thrapp, R., Westbrook, C. and Subramanian, D. (2001). Robust localization algorithms for an autonomous campus tour guide, *IEEE International Conference on Robotics and Automation*, Seoul, pp. 2065–2071.

Thrun, S., Burgard, W. and Fox, D. (2005). *Probabilistic robotics*, The MIT Press, London, England.

Thrun, S. and Leonard, J. (2008). Simultaneous localization and mapping, *in* B. Siciliano and O. Khatib (eds), *Springer Handbook of Robotics*, 1st edn, Springer-Verlag Ltd., pp. 871–886.

Xiang, Z., Santos, V. M. and Liu, J. (2003). Robust mobile robot localization by fusing laser and sonar, *IEEE International Conference on Advanced Robotics*, Coimbra, pp. 276–280.