2006

Navigational Context Recognition for an Autonomous Robot in a Simulated Tree Plantation

Lav R. Khot
Iowa State University

Lie Tang
Iowa State University, lietang@iastate.edu

Simon Blackmore
Unibots, Ltd.

Michael Nørremark
The Royal Veterinary and Agricultural University

NAVIGATIONAL CONTEXT RECOGNITION FOR AN AUTONOMOUS ROBOT IN A SIMULATED TREE PLANTATION

L. R. Khot, L. Tang, S. B. Blackmore, M. Nørremark

ABSTRACT. *A sensor fusion technique was developed for estimating the navigational posture of a skid-steered mobile robot in a simulated tree plantation nursery. Real-time kinematic GPS (RTK-GPS) and dynamic measurement unit (DMU) sensors were used to determine the position and orientation of the robot, while a laser range finder was used to locate the tree positions within a selected range. The RTK-GPS error was modeled by a second-order autoregressive model, and error states were incorporated into extended Kalman filter (EKF) design. Through EKF filtering, the mean and standard deviation of error in the easting direction decreased from 4.05 to 2.21 cm and from 8.27 to 1.89 cm, respectively, while in the northing direction, they decreased from 4.64 to 1.81 cm and from 11 to 2.16 cm, respectively. The geo-referenced tree positions along the navigational paths were also recovered by using a K-means clustering algorithm, achieving an average error of tree position estimates of 4.4 cm. The developed sensor fusion algorithm was proven to be capable of recognizing and reconstructing the navigational environment of a simulated tree plantation, which offers a great potential in improving the applicability of an autonomous robot to operate in nursery tree plantations for operations such as intra-row mechanical weeding.*

Keywords. *Autoregressive model, Extended Kalman filter, Robotic weeding, Robot navigation, Sensor fusion, Tree plantation nursery.*

In Europe, particularly in countries like Denmark and The Netherlands, nursery tree production has significant economic value, with hundreds of millions of dollars of annual turnover. Weeding is a critical operation in tree production nurseries, and chemical weed control methods are currently very common. However, with increasingly strict regulations on herbicide usage, tree growers are demanding mechanical weeding solutions (Melander et al., 2005). Existing mechanical weeding devices suffer from impractically low work rates in intra-row weeding and are not yet applicable for nursery trees due to the high potential for damage to the trees. Under such circumstances, an autonomous weeding robot capable of navigating through nursery tree plantations for intra-row weeding is a promising alternative (Have et al., 2005). To enable a successful weeding operation by an autonomous robot in tree plantations, such a robot not only needs to detect trees but also needs to understand its navigational context. Although tree positions can be surveyed manually, it is a time-consuming task that may hinder the practical use of such a robotic approach.

Submitted for review in September 2005 as manuscript number IET 6060; approved for publication by the Information & Electrical Technologies Division of ASABE in July 2006. Presented at the 2005 ASAE Annual Meeting as Paper No. 053092.

The authors are **Lav Ramchandra Khot, ASABE Member Engineer,** Graduate Student, and **Lie Tang, ASABE Member Engineer,** Assistant Professor, Department of Agricultural and Biosystems Engineering, Iowa State University, Ames, Iowa; **Simon Blackmore, ASABE Member Engineer,** Managing Director, Unibots, Ltd., Bedford, U.K.; and **Michael Nørremark, ASABE Member,** Graduate Student, Department of Agricultural Sciences, The Royal Veterinary and Agricultural University, Taastrup, Denmark. **Corresponding author:** Lie Tang, Department of Agricultural and Biosystems Engineering, 203 Davidson Hall, Iowa State University, Ames, IA 50011; phone: 515-294-9778; fax: 515-294-3366; e-mail: lietang@iastate.edu.

Two commonly used sensors for robot navigation are Global Positioning System (GPS) sensors and inertial measurement units (IMU). Although GPS is commonly used to determine the global position, it is subject to ionospheric, system dynamics, and signal multipath errors. Inertial sensors are widely used to measure system dynamics, including acceleration and rotational rates. Errors that affect IMU measurements are due to static bias drift, scale factor non-linearity, and temperature changes. In addition, errors are also caused by angular random walk (white noise), bias instability, and magnetic sensitivity (Lauro et al., 2000). Therefore, it is important to remove these errors to provide more accurate navigational signals to an autonomous mobile robot.

Errors that appear in GPS (absolute positioning) and in the outputs of dead-reckoning sensors (relative positioning) are complementary in nature. The errors of these sensors can be rectified by fusing measurements from them. An extended Kalman filter (EKF) is an effective tool that is applicable to non-linear systems. EKF allows integration of measurements from the complementary sensors for an overall improvement in signal accuracy. Bergeijk et al. (1998) developed an EKF sensor fusion algorithm to integrate GPS, radar, wheel odometer, and electronic compass signals. Their position filter model consisting of velocity and acceleration states was the most accurate, as the acceleration states were useful for maintaining the accuracy of the positioning system. Kiriy and Buehler (2002) applied an EKF to localize a golf course lawnmower using a fiber optic gyroscope, odometry, and two digital cameras. They suggested that the performance of EKF can be further improved by online adjustment of the system and measurement noise covariance matrices, and by increasing the measurement data frequency. Guo et al. (2002) developed a Kalman filter algorithm to estimate tractor position by fusing signals from a six-axis IMU and a GPS

  ISSN 0001–2351

sensor to remove the noise from the GPS signal and the drift error of the IMU. The low-cost system improved position accuracy by smoothing GPS noise and improved the heading angle measurements in terms of noise level and drifting.

In this article, we address the problem of a navigational context recognition algorithm that would allow the use of onboard sensors to generate signals for both intra-row weeding and inter-row navigation. There are basically two steps involved in accomplishing this task. The first step is to estimate the robot posture (position and heading) using the signals from positioning and dynamic sensors; the second step is to combine posture measurements with onboard dead-reckoning sensor signals for navigational environment recognition and reconstruction. In this way, a map of tree positions can be developed that enables the robot to be acquainted with its global and local navigational context, thus facilitating its navigation between the rows for performing intra-row weeding operation. Therefore, this article describes an EKF implementation for fusing RTK-GPS and dynamic measurement unit (DMU) sensors to estimate the posture of a weeding robot in a simulated tree plantation nursery and estimating tree positions with laser range scanner. This research had the following specific objectives:

- To estimate the posture of an autonomous robot in a simulated tree plantation nursery by fusing the RTK-GPS and DMU sensors through an EKF algorithm.
- To recognize and reconstruct the navigational environment for an autonomous robot through the development of a map of tree positions by combining the EKF-generated posture estimates with the signals from an onboard dead-reckoning sensor.

Materials and Methodology

Experimental Setup

The research robot platform (model Atrv-Jr, iRobot, Burlington, Mass.) was equipped with RTK-GPS, laser range scanner, odometry, and DMU.

Tree rows were simulated by placing 25 surveying poles (1.2 × 1.2 m spacing) in an open flat terrain as shown in figure 1a. The robot was teleoperated through the poles, and synchronized sensor readings were collected at a sampling frequency of 10 Hz. The average travel speed of the robot vehicle was about 1.5 m/s. Prior to the inter-row navigation, onboard sensors were calibrated using a central-pivot double-rail frame calibration platform (fig. 1b). The rails had a U-shaped cross-section, with the inner dimension closely matching the tire width of the robot. The maximal space between tires and the sidewall of the rails was less than 5 mm. The length of the rail tracks was 6 m.

The robot was driven along the rail frame four times to evaluate the errors in GPS and DMU sensors for the case when the robot moved in a straight line. After the first two runs (dataset: "Line1.dat" and "Line2.dat"), the calibration platform was rotated 90° and calibration runs were conducted again (dataset: "Line3.dat" and "Line4.dat"). After that, the robot was immobilized and positioned at one end of the calibration platform. The calibration platform was then rotated 360° clockwise and counterclockwise (dataset: "Circle1.dat" and "Circle2.dat") to study the GPS sensor errors in circular motion while excluding the dynamic noise generated by vehicle locomotion.

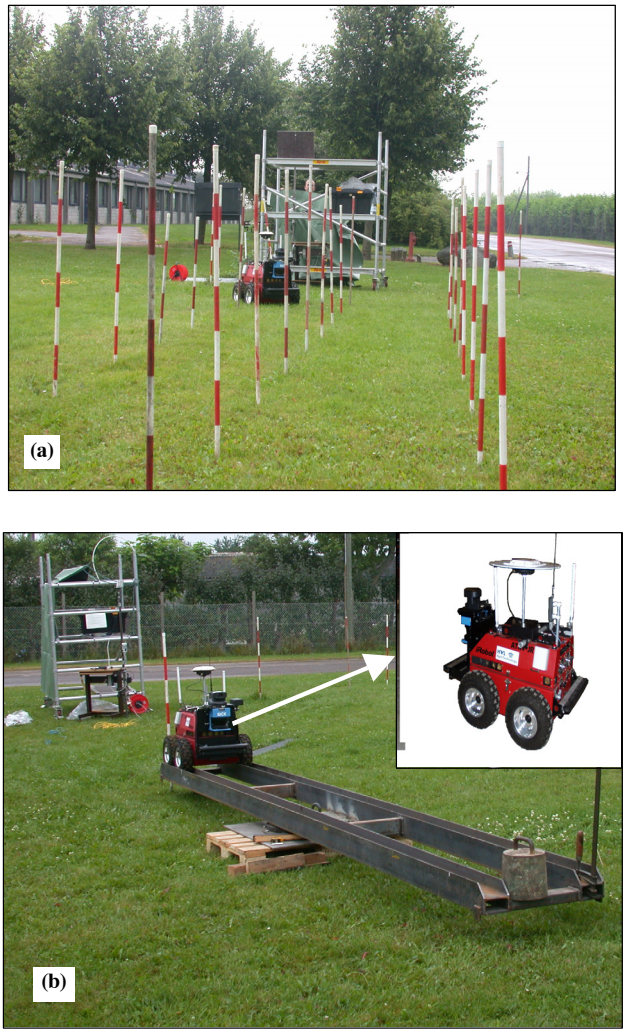

Figure 1. Experimental setup: (a) simulated nursery tree plantation, and (b) calibration platform.

The datasets obtained from the straight-line and circular trajectories were used to develop a GPS error model. In the case of straight-line trajectories, the robot was made stationary at the beginning and the end of the calibration platform to collect more data points for generating ground truth reference positions. These reference points were averaged, and a straight-line reference trajectory was obtained. The RTK-GPS measurement offsets in the x and y directions were then obtained by using the straight line as a reference trajectory. The calibration platform prevented the vehicle from changing its heading. Therefore, the error estimate of yaw rate measurements from the DMU was obtained. Circular calibration tests were conducted to observe the error in GPS measurements without the interference of vehicle dynamics. In these tests, the robot was immobilized at one end of the calibration platform and the platform was rotated 360° at an approximately constant speed to generate circular reference trajectories. These circular and straight-line reference trajectories were then used for the GPS measurements error modeling.

Sensor Calibration and Error Modeling

The three main sensor signals used for determining navigational context of the robot were the dynamic measure-

ment unit (DMU), real-time kinematic GPS (RTK-GPS), and laser range scanner.

Dynamic Measurement Unit

An IMU300CC series DMU (Crossbow, San Jose, Cal.) was used to measure acceleration and rotational rates along the course of robot navigation. As mentioned earlier, many factors affect the DMU signal. Theoretically, a perfect linear scale factor (SF) would provide an accurate conversion that is exactly proportional to the rate of rotation (Borenstein, 1998). However, in practice, that requires a delicate calibration device to generate different rotational speeds for calibration. This research only focused on the removal of static bias drift from the DMU. Static bias is the DMU's output when the DMU is not rotating, i.e., it is at zero point. Static bias drift, after integration, results in unbounded growth of orientation measurements. If the drift is considered to be consistent under known conditions, it can be compensated externally; otherwise, compensation can be difficult. Hence, a common compensation procedure is used to remove the static bias drift. To achieve this, a number of samples were collected from a stationary robot platform prior to every test run (10 to 20 s), and the static bias drift calculated using equation 1 was later subtracted from subsequent readings:

$$\omega_s = \frac{1}{n}\sum_{i=1}^{n}\omega_{g,i} \tag{1}$$

where ω_s is static bias drift, $\omega_{g,i}$ is yaw rate of a sample, and n is the total number of samples.

Laser Range Scanner

The laser range scanner (LMS200, SICK, Reute, Germany) had 180° coverage and 1° angular resolution and was used to locate surveying poles when the robot was driven through the rows. Although the maximum range of the scanner is 150 m, the effective range is reduced to 30 m with supplemental reflection and to 4 m with minimum reflectivity (1.8%). The expected quantization error in pole positioning is 1.74 cm per 1 m range based on 1° angular resolution. In this research, laser signals within 1 m range were used for locating inter-row poles along the course of navigation.

Real-Time Kinematic GPS

The RTK-GPS receiver (MS750 Trimble, Sunnyvale, Cal.) had an accuracy of about ±2 cm and was used to measure the position of the robot. The GPS data obtained from the straight-line and circular calibration paths were used to formulate an autoregression (AR) error model. This error modeling was motivated by an attempt to regulate the signal noise so that it could be better described as white Gaussian noise, which is an inherent assumption of Kalman filter theory. In addition, instead of either assuming or fine-tuning measurement noise covariance, as found in many reported algorithms (Bergeijk et al., 1998; Guo et al., 2002; Kiriy and Buehler, 2002), error modeling provides a mathematical means to calculate the covariance values. The AR model was also helpful in estimating the process noise covariance values. Babu and Wang (2004) found that the accuracy of Doppler estimates obtained from GPS/INS integration improved by 45% to 50% with a second-order AR model when compared to the first-order Gauss-Markov model. Similarly, Will (2001) used a first-order AR model to model the differential GPS error, which reduced position estimation error from several meters down to 25 cm.

In AR analysis, the root mean squared (RMS) error between the series estimated by the AR coefficients and the actual series decreases rapidly and ultimately stabilizes as the order increases. In this research, a second-order AR model, AR(2), was used based on the following criterion:

$$N = 2 + (f_s / 1\ \text{kHz}) \tag{2}$$

where N is the order of the AR model, and f_s is the measurement frequency (Hz).

The first step in AR modeling is to determine the coefficients by means of making the average residual zero and making the standard deviation as small as possible. Several techniques exist for computing AR coefficients. The two major methods are the least squares method and Burg's method (Bourke, 1998). The least squares method is more commonly used and is based on the Yule-Walker equations. In this research, the Yule-Walker autocorrelation method was used to determine the AR coefficients using time averaging, as shown in equations 3 to 7. Given a time series $x(n)$, where n = 1, 2, ..., M, the autocorrelation function at lag k is defined as:

$$R_{xx}(k) = \frac{1}{M-k}\sum_{n=1}^{M-k}\left(x[n]\times x[n+k]\right) \tag{3}$$

Having values of $R_{xx}(0)$, $R_{xx}(1)$, and $R_{xx}(2)$ from equation 3, the AR coefficients (A_1 and A_2) and white noise covariance (v) are determined by the following equations (Babu and Wang, 2004):

$$A_1 = \frac{R_{xx}(0)R_{xx}(1) - R_{xx}(1)R_{xx}(2)}{R_{xx}^2(0)R_{xx}^2(1)} \tag{4}$$

$$A_2 = \frac{R_{xx}(0)R_{xx}(2) - R_{xx}(1)R_{xx}(1)}{R_{xx}^2(0)R_{xx}^2(1)} \tag{5}$$

$$\nu = R_{xx}(0) - A_1R_{xx}(1) - A_2R_{xx}(2) \tag{6}$$

The final AR model can be written as:

$$x_n = A_1x_{n-1} + A_2x_{n-2} + \nu \tag{7}$$

Navigational Algorithm

Sensor Fusion

The RTK-GPS and DMU data were fused to obtain the robot posture. Once the posture of the robot was determined by the EKF, the laser range scanner data could be used jointly to locate the pole positions.

An extended Kalman filter is mainly used to estimate system states that can only be observed indirectly or inaccurately by the system itself. In an EKF, a set of mathematical equations provides an efficient computational (recursive) means to estimate the past, present, as well as future states, even when the precise nature of the modeled system is unknown (Welch and Bishop, 2004). In general, the discrete EKF consists of the following equations:

Non-linear dynamic model:

$$x_k = f(x_{k-1}, k-1) + w_k$$

$$w_k \approx N(0, Q_k) \tag{8}$$

Non-linear measurement model:

$$z_k = h(x_k, k) + \nu_k$$

$$\nu_k \approx N(0, R_k) \quad (9)$$

In equation 8, x_k and x_{k-1} are the present and preceding states of the system, w_k is the system noise signal, and Q_k is the system noise covariance. Equation 9 relates the system state x_k with the measurement vector z_k, which has v_k as the measurement noise signal and R_k as the measurement noise covariance. The measurement function (h) in equation 9 is used in updating estimates with measurements.

Developing a state space model is an important step in sensor fusion when using a Kalman filter. The state space model can be designed based on the relationship between the state change, which corresponds to the change in position and heading of a vehicle. Figure 2 represents a change in state space and the corresponding change in position and heading of the robot in the vehicle coordinate system as well as in the global coordinate system. GPS, a satellite-based positioning system, provides the position of the vehicle globally in easting (X) and northing (Y) coordinates. In the vehicle coordinate system, the x-axis was defined as the axis in the direction of vehicle travel, pointing forward, and was in the longitudinal plane of symmetry. The y-axis was defined as the axis in the lateral plane of symmetry, perpendicular to the direction of travel. At time step k, (X_k, Y_k) symbolizes the vehicle state in the global coordinate system, (x_k, y_k) symbolizes the vehicle state in the vehicle coordinate system, and Φ_k is the vehicle heading in the global coordinate system.

Kinematic equations representing the position and orientation of the robot were derived from figure 2 and are given in equations 10 to 16:

$$f_x = X_{k+1} = X_k + V_k T \sin(\phi_k) \quad (10)$$

$$f_Y = Y_{k+1} = Y_k + V_k T \cos(\phi_k) \quad (11)$$

$$f_\phi = \phi_{k+1} = \phi_k + r_k T \quad (12)$$

$$f_{X_{err1}} = X_{err1,k+1} = A_{x1} X_{err,k} \quad (13)$$

$$f_{X_{err2}} = X_{err2,k+1} = A_{x2} X_{err,k-1} \quad (14)$$

$$f_{Y_{err1}} = Y_{err1,k+1} = A_{y1} Y_{err,k} \quad (15)$$

$$f_{Y_{err2}} = Y_{err2,k+1} = A_{y2} Y_{err,k-1} \quad (16)$$

where k, $k-1$, and $k+1$ are the present, preceding, and next time steps; V is the vehicle velocity in the direction of travel (m/s); T is the sampling interval (s); r is the yaw rate (°/s); and A_{x1}, A_{x2}, A_{y1}, and A_{y2} are the AR coefficients.

In above equations, f_X, f_Y, and f_ϕ are system functions for (X, Y) positions, and ø is heading angle change; $f_{X_{err1}}$, $f_{X_{err2}}$, $f_{Y_{err1}}$, and $f_{Y_{err2}}$ are components of the error in GPS measurements, which are defined by the AR model devel–

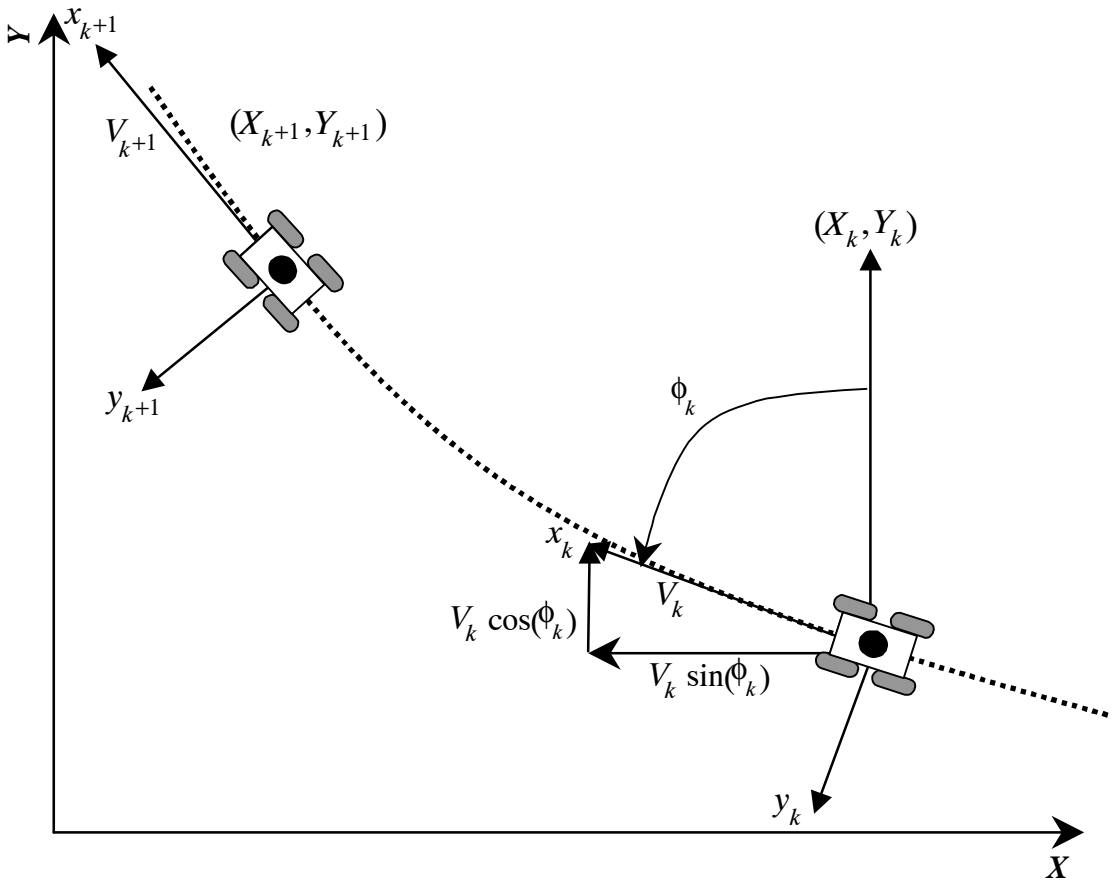

Figure 2. Vehicle states in local and global coordinate systems.

oped for the GPS sensor. The overall system function, $f(x)$, represented by equations 10 to 16 can be written as:

$$f(x) = [f_X\, f_Y\, f_\phi\, f_{X_{err1}}\, f_{X_{err2}}\, f_{Y_{err1}}\, f_{Y_{err2}}]^T \quad (17)$$

The system state vector (x_k) is defined as:

$$x_k = [X_k, Y_k, \phi_k, X_{err1,k}, X_{err2,k}, Y_{err1,k}, Y_{err2,k}]^T \quad (18)$$

The following equation represents the system Jacobian matrix (A_k) for the above system:

$$A_k = \begin{pmatrix} 1 & 0 & VT\cos(\phi_k) & 0 & 0 & 0 & 0 \\ 0 & 1 & -VT\sin(\phi_k) & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{x1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{x2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_{y1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_{y2} \end{pmatrix}_{x_k} \quad (19)$$

The input vector may be written as $u_k = [V_k\, r_k]^T$, where V_k is the velocity of the vehicle and r_k is the yaw rate from the DMU, as control inputs. The input Jacobian matrix (B_k) relates the system function $f(x)$ with these inputs. The input Jacobian matrix (B_k) is:

$$B_k = \begin{pmatrix} T\sin(\phi_k) & 0 \\ T\cos(\phi_k) & 0 \\ 0 & T \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}_{x_k} \quad (20)$$

The measurement vector consists of the GPS measurements, i.e., X_{gps}, Y_{gps}, ϕ_{gps} for northing, easting, and heading (in UTM coordinates), as given in equation 21. The yaw rate from the DMU was used in the prediction stage of the filter to estimate the heading of the robot, as shown in equation 12. The error acquired from error states during the prediction step of the EKF was subtracted from subsequent measurements of GPS X and Y before correction to improve the performance of the filter:

$$Z_k = [X_{gps} - X_{err1} - X_{err2}, Y_{gps} - Y_{err1} - Y_{err2}, \phi_{gps}]^T \quad (21)$$

The measurement Jacobian matrix (H_k), which relates the measurements to the state variables, is given in equation 22:

$$H_k = \begin{pmatrix} 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (22)$$

The EKF was applied offline to the experimental data. Before commencing each run, the robot was held stationary for a few seconds and ground truth position was determined from averaging these data. The initial true heading corresponds to the angle between the ground truth position and the changed position of the robot. The initial state (x_k) of the EKF was taken from ground truth values. An *a priori* estimate of the system state was made at the prediction stage of the EKF. The covariance matrix of the prior estimate was calculated by equation 23. Using the initial error covariance matrix (P_k^-), measurement Jacobian matrix (H_k), and measurement noise covariance matrix (R), the Kalman gain (K_k) was computed at the correction stage:

$$P_k^- = A_k P_{k-1} A_k^T + B_k \Gamma B_k^T + Q \quad (23)$$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1} \quad (24)$$

where Γ is the input noise covariance.

The Kalman gain (K_k) was then used to update the *a priori* state estimates with the measurements and state error covariance matrix, as shown in equation 25:

$$P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R K_k^T \quad (25)$$

This recursive method was thus repeated between the prediction and correction steps to estimate all the states of the vehicle.

Pole Recognition and Reconstruction

The state estimates were then integrated with the scanner data to locate the pole position in global coordinates. The distance from the laser range scanner and the scanner angle for the visible poles in polar coordinates s_i (eq. 26) was transformed to vehicle coordinates (x_{ij}, y_{ij}) using equation 27:

$$s_i = [d_{ij}, \theta_{ij}];\ j = 0, 1, ..., 180 \quad (26)$$

$$\begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} = \begin{bmatrix} d_{ij} \times \sin(\theta_{ij}) \\ d_{ij} \times \cos(\theta_{ij}) \end{bmatrix} + \begin{bmatrix} l \\ 0 \end{bmatrix} \quad (27)$$

where θ is the angle ($^\circ$) and d is the distance (m) between laser range scanner and the object scanned, and l is the distance (m) between the laser range scanner and the robot center.

The pole position in vehicle coordinates was then transformed to global coordinates (X_{ij}, Y_{ij}) (eq. 28):

$$\begin{bmatrix} X_{ij} \\ Y_{ij} \end{bmatrix} = \begin{bmatrix} X_{f,i} \\ Y_{f,i} \end{bmatrix} + \begin{bmatrix} \cos\phi_i & \sin\phi_i \\ \sin\phi_i & -\cos\phi_i \end{bmatrix} + \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} \quad (28)$$

where ϕ is the heading angle, and ($X_{f,I}$, $Y_{f,i}$) is the robot position; both were estimated by the developed EKF.

After estimating the pole positions in 1 m scanning range, a K-means clustering algorithm was implemented to obtain the centroid of the pole position clusters. The observations were partitioned into K mutually exclusive clusters, and the algorithm provided a vector of indices indicating to which of the K clusters it assigned each observation. A clustering algorithm defines each cluster by its member measurements and by its centroid, or center. The centroid for each cluster is the point at which the sum of distances from all measurements in that cluster is minimized (Johnson and Wichern, 2002). A clustering algorithm moves measurements between clusters until the sum cannot be decreased further. The result is a set of clusters that is as compact and well-separated as possible. The K-means algorithm can produce different results depending on the starting condition of clustering. Therefore, for a stable cluster solution, initial cluster seeds

were selected from the intersections of a grid (1.0 × 0.5 m) that covers the entire experimental area.

Filter Realization

In this research, two types of EKFs were implemented in order to study the effectiveness of the AR model on posture estimation of the robot. The first EKF included position and heading states in a state vector, i.e., $x_k = [X_k\ Y_k\ \phi_k]^T$, while the second EKF also included error states $X_{err\ 1}$, $X_{err\ 2}$, $Y_{err\ 1}$, and $Y_{err\ 2}$ (see eq. 18). It is important to have system (Q), input (Γ), and measurement (R) noise as white Gaussian and uncorrelated. Fine-tuning of covariance matrices manually in order to remove high-frequency errors is delicate work (Sukkarieh et al., 1999). The determination of the process noise covariance is generally complicated, as it is difficult to directly observe the process that is being estimated. In the case of the first EKF implementation, for system position states, noise covariance was taken to be $\sigma^2_x = \sigma^2_y = 4 \times 10^{-4}$ m^2, derived from the nominal accuracy of the RTK-GPS sensor used in this research. The DMU has an angular bias rate accuracy of less than 2°, so system noise covariance of $\sigma^2_{ø} = (0.15°) = 7.62 \times 10^{-6}$ radians^2 was used accordingly for the heading state. The noise estimated above formed the system noise covariance (Q). As for R, there were no direct indicators for determining the covariance values of every measurement state; thus, fine-tuning of R by trial and error was decisive in nature and could have greatly affected the performance of the filter.

In the case of the second EKF with the AR error model, the Gaussian white noise statistics were estimated using AR model coefficients. This was made possible by using the measurements conducted over reference paths. The AR model was derived and implemented based on error statistics through analyzing data generated from the straight-line and circular reference calibration paths. The coefficient values obtained are summarized in table 1. The residual error was calculated by subtracting the output of the AR model from the measurement error (true error) and was used as measurement noise covariance. In order to compensate signal noise as well as external vibration noise on the GPS antenna, the average values of $A_{x,k}$ and $A_{y,k}$ from measurements of two circular paths and three straight-line paths were used in actual error modeling. Having estimated the measurement noise covariance from the AR(2) model on reference trajectories, fine-tuning of system noise covariance became trivial when the EKF was implemented on inter-row navigational data. Since the error estimated by error states was subtracted from measurements before incorporating them in the EKF, X, Y position state noise covariance values were about four times smaller than that of the EKF without the AR model, reducing the sensitivity of manual fine-tuning.

The system and measurement error covariance matrices used in the EKF with the AR model are shown in equations 29 and 30:

System noise covariance matrix (Q):

$$Q = \begin{pmatrix} 1\times10^{-4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1\times10^{-4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7.62\times10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1\times10^{-4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1\times10^{-4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1\times10^{-4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1\times10^{-4} \end{pmatrix} \quad (29)$$

Measurement noise covariance matrix (R):

$$R = \begin{pmatrix} 1.74\times10^{-3} & 0 & 0 \\ 0 & 1.69\times10^{-3} & 0 \\ 0 & 0 & 5\times10^{-4} \end{pmatrix} \quad (30)$$

Velocity noise covariance of $\sigma^2_v = (4.47 \times 10^{-3})^2 = 2 \times 10^{-5}$ (m/sec)^2 and yaw rate noise covariance $\sigma^2_r = (0.5°/\text{sec})^2 = 7.62 \times 10^{-5}$ (radian/sec)^2 was used as the input noise covariance (Γ) in EKF implementation:

$$\Gamma = \begin{pmatrix} 2\times10^{-5} & 0 \\ 0 & 7.62\times10^{-5} \end{pmatrix} \quad (31)$$

Results and Discussion

To evaluate the effectiveness of the model, the EKF was applied to the data obtained from reference trajectories. Usually, the efficacy of an EKF algorithm depends on fine-tuning of the Q and R values to a great extent, as discussed in the previous section. The estimated error from the AR(2) model was compared with actual GPS measurement noise in the X and Y directions (fig. 3). When the robot was not driven, the actual noise of GPS X and Y was very small, which was likely due to the absence of external vibrating forces acting on the GPS antenna. In contrast, actual noise of GPS in the X and Y directions was greater when the robot was driven along the straight-line trajectories (fig. 4).

The Line 1 data had a mean error of about 3.47 cm and 0.41 cm and standard deviations of 7.76 cm and 0.93 cm in the X and Y directions, respectively. The mean error after incorporating the AR model with the EKF was reduced considerably, to 3.13 cm and 0.037 cm in the X and Y directions, with standard deviations of 2.64 cm and 0.62 cm. A similar trend was observed for Line 2, with the larger mean error in the X direction (fig. 4a), which was reduced from 4.63 cm to 2.68 cm with the standard deviation reduced from 8.78 cm to 0.99 cm. For Line 3 and Line 4, the mean measurement errors in the Y direction (fig. 4b) were 9.17 cm and 0.10 cm with standard deviations of 11.41 cm and

Table 1. Autoregressive coefficients of the second-order model.

	Coefficient for X		Coefficient for Y		Residual (m)	
Robot Path	Ax_1	Ax_2	Ay_1	Ay_2	*Erx*	*Ery*
Circle 1	0.4344	−0.0913	0.4966	−0.0085	1.47E−05	1.62E−05
Circle 2	0.4147	−0.1920	0.4948	−0.1001	1.07E−05	1.16E−05
Line 1	0.5769	−0.1692	0.5769	−0.1692	0.0044	6.34E−05
Line 2	0.7073	−0.0522	0.7073	−0.0522	0.0042	5.95E−05
Line 4	0.4702	0.0525	0.4702	0.0525	8.50E−05	0.0083
Average	0.5207	−0.0904	0.5492	−0.0555	1.7421 E−03	1.6901E−03

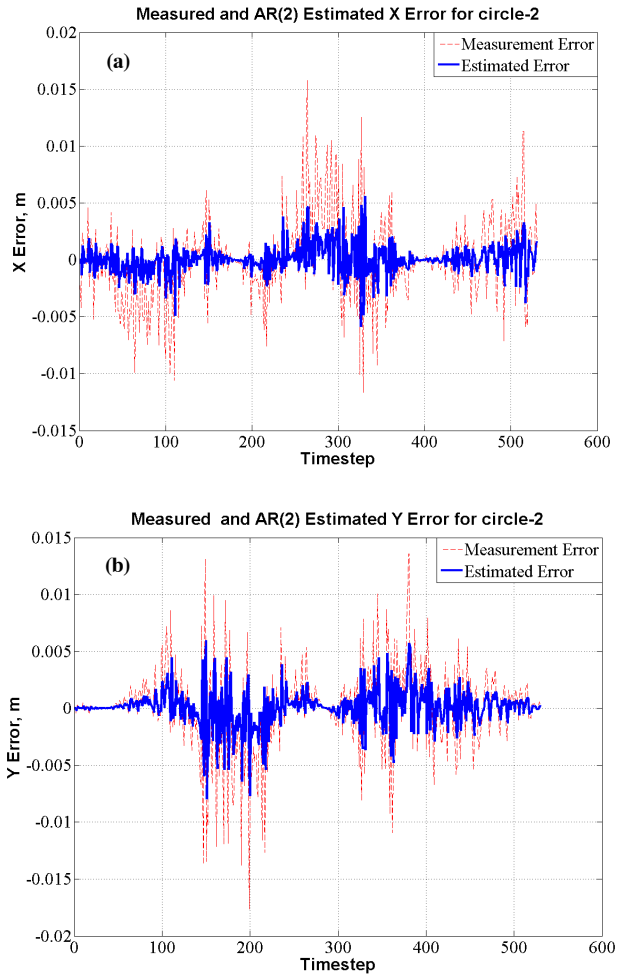

Figure 3. Measured and AR(2) estimated error for circle-2: (a) *X* error, and (b) *Y* error.

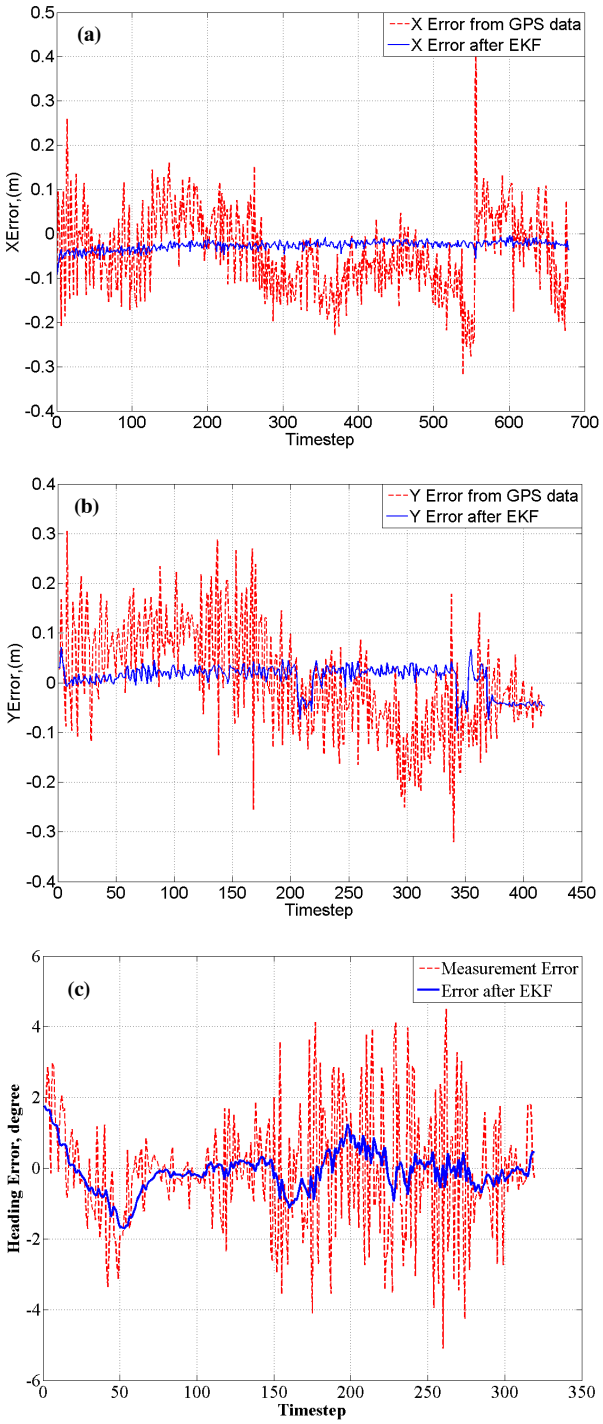

Figure 4. Error with and without EKF during straight-line calibration: (a) *X* error, (b) *Y* error, and (c) heading error.

10.51 cm, respectively. The mean errors and standard deviations of Line 3 and Line 4 after EKF with AR were reduced by about 64% and 82%, respectively. It could be inferred from these four test results that the noise is always greater in the direction of travel of the robot than in the perpendicular direction. The mean and standard deviation of heading error (fig. 4c) for all the test runs were also reduced after applying the EKF with the AR model. Specifically, in Line 3 and Line 4, the standard deviation of heading error was reduced from 1.70° to 0.58° and from 1.16° to 0.77°, respectively.

During the calibration experiments, the robot was teleoperated along a straight line in the centrally pivoted rail frame, as shown in figure 2b. The clearance between the robot tires and the walls of the U-shaped frame rails was about 5 mm. It was difficult to drive the robot exactly straight, causing friction between the tires and frame walls, which in turn resulted in an unsmooth motion and sometimes a sudden stop of the robot. In addition, the GPS receiver antenna and a heavy metal disk that was used to prevent the near-ground multipath error were mounted on top of a single supporting aluminum rod (fig. 1b). This mounting mechanism made the antenna more subject to vibratory noise.

During the robot navigation experiments in the simulated tree plantation, the robot was intentionally driven back and forth and around the poles to determine the effect of sharp maneuvers on the GPS measurements. In the case of vehicle localization, the GPS signal is always coupled with the vehicle body dynamics, which can raise the noise to a much high level than the rated level of the selected GPS sensor. The EKF algorithm was developed to overcome the overall noise effects in GPS measurements on a mobile robotic vehicle. As a result, the overall mean and standard deviation of error for the GPS signals in the *X* direction decreased from 4.05 cm to 2.21 cm and from 8.27 cm to 1.89 cm; in the *Y* direction, they decreased from 4.64 cm to 1.81 cm and from 11 cm to 2.16 cm, respectively, after filtering. The EKF algorithm was thus shown to be applicable for reducing noise from sensor

signals, and in turn improving robot posture estimation. Therefore, after estimating the robot posture, the poles within the range of the laser range scanner were located using the measurement data from the scanner.

Figure 5 illustrates the clustering results when cluster seeds were generated randomly in the *K*-means clustering algorithm. In this case, the clustering algorithm was unable to locate 25 poles properly, although 25 clusters were formed. As shown in figure 5, many clusters were falsely formed, especially around the areas where the robot experienced rapid turning maneuvers during teleoperation. In order to drive the *K*-means clustering algorithm to converge to all true pole clusters, cluster seeds were selected from the intersections of a grid of 1.0 × 0.5 m cells that covers the entire experimental area. Figures 6 and 7 illustrate the clustering results after this algorithm modification, where all 25 pole clusters were correctly formed. The effectiveness of the developed EKF with the AR(2) model on path recovery is further depicted by closeup views of segments where the raw GPS signal was substantially noisier.

The mean error between the true and the estimated pole positions for path 3 decreased from 5.12 cm (EKF without error model) to 3.77 cm when EKF with AR(2) was applied; the standard deviation also decreased from 3.65 cm to 2.79 cm. Similarly, the mean and standard deviation of the pole positioning error for path 4 decreased from 6.25 cm

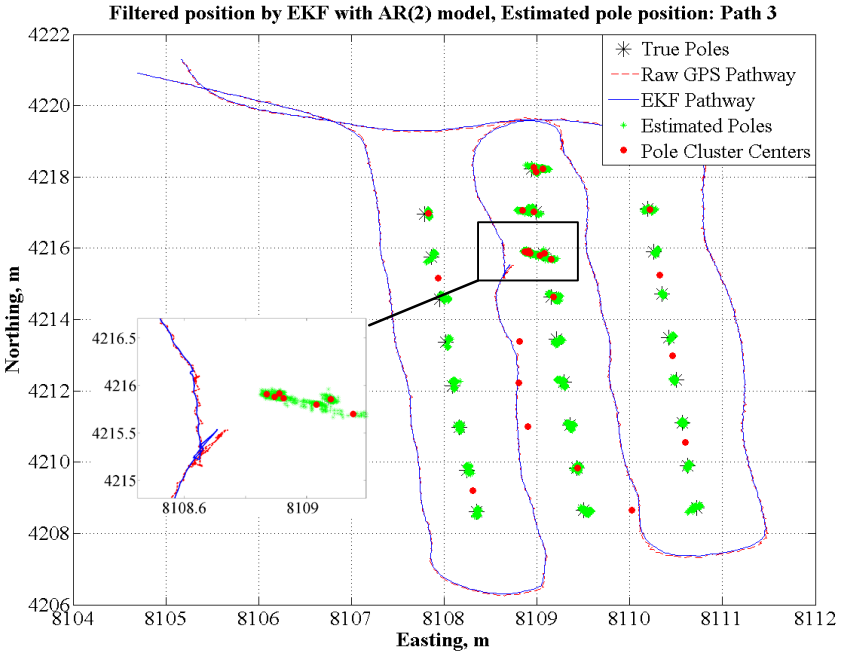

Figure 5. Path of the robot with EKF and the estimated pole position with random initial seeds (path 3).

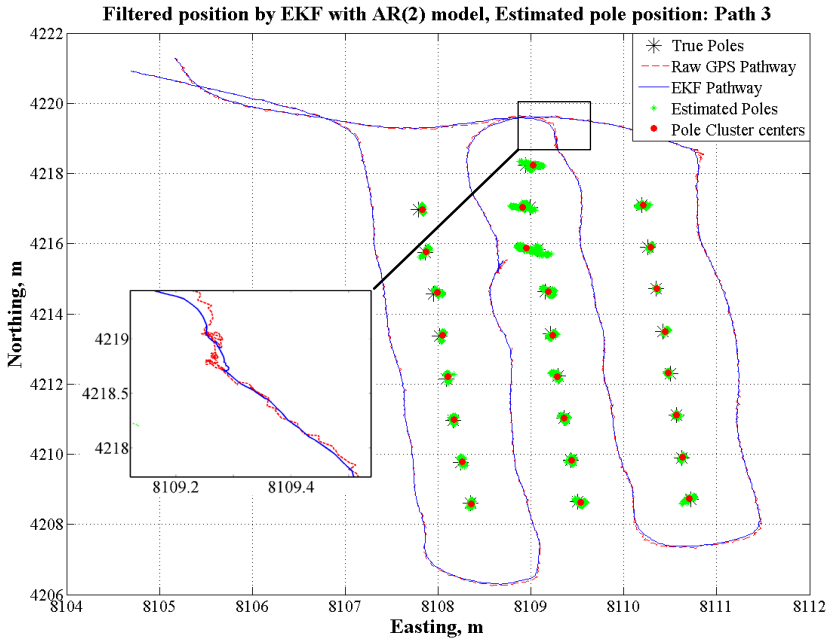

Figure 6. Path of the robot with EKF and the estimated pole position using grid intersections as initial seeds (path 3).

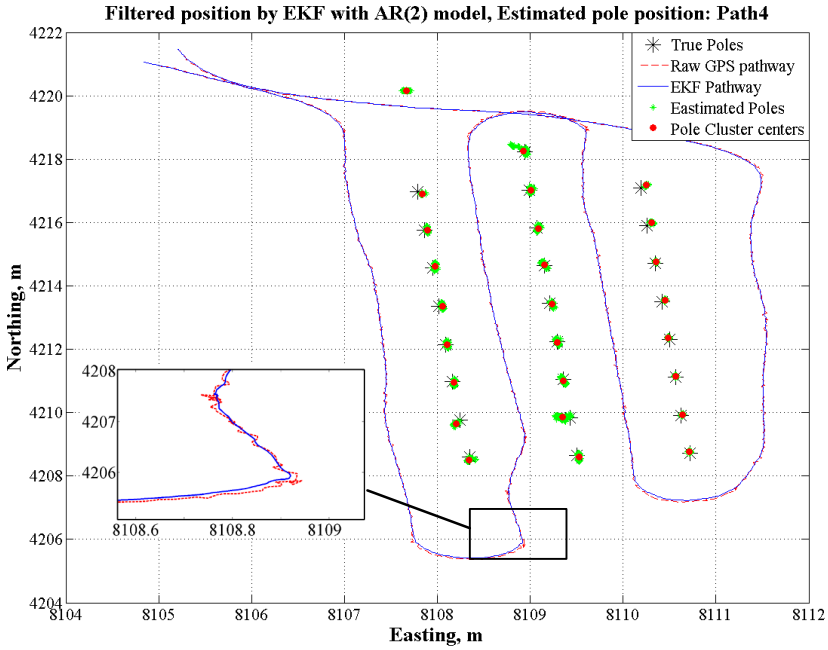

Figure 7. Path of the robot with EKF and the estimated pole position using grid intersections as initial seeds (path 4).

to 5.17 cm and from 3.27 cm to 2.86 cm, respectively. When evaluating all four trials of inter-row navigation, the pole positions could be located globally with an overall accuracy of 4.4 cm. Therefore, the developed EKF algorithm with error modeling improved the robot positioning and heading estimation, and in turn the accuracy of pole position estimation, with *K*-means clustering.

Conclusions

An extended Kalman filtering algorithm was developed and implemented to improve the accuracy of posture estimation of a skid-steered autonomous robot. A kinematic system model consisting of seven states was developed for implementing the EKF algorithm. The GPS error, along with external vibration noise and DMU drift, was found to be the main source of error in determining the position and heading of the robot vehicle. In addition to the EKF, a second-order autoregression error model was developed to model the RTK-GPS errors.

The developed EKF algorithm was effective in improving the accuracy of the robot posture estimation. The EKF with the AR model enhanced robot's navigational ability over the EKF without incorporating an error model. The EKF algorithm reduced the sensor signal noise to a great extent. In the case of the EKF without the AR model, the manual fine-tuning of the system and measurement noise covariance matrices affected the performance of the filter considerably, while the AR model reduced the need for fine-tuning through a trial and error process. Furthermore, the developed filtering and clustering algorithms were successful in recognizing and reconstructing the navigational context of an autonomous weeding robot in a simulated tree plantation nursery.

In this research, pitching and rolling effects were not taken into account because the experimental field was almost flat. To represent a more general scenario for an all-terrain robot, 3D attitude estimation needs to be incorporated into the EKF design. The field experiments were conducted in an open field, and the GPS signal was therefore continuous and without blockage. However, in a nursery tree plantation, the GPS signal may not be continuous and may become unavailable for a certain period of time. In that case, the developed Kalman filtering algorithm can still use DMU measurements to estimate the posture of the autonomous robot vehicle.

References

Babu, R., and J. Wang. 2004. Improving the quality of IMU-derived Doppler estimates for ultra-tight GPS/INS integration. GNSS2004 Paper 144. Sydney, Australia: University of New South Wales. Available at: www.gmat.unsw.edu.au/snap/publications/. Accessed 10 May 2005.

Bergeijk, V. J., D. Goense, K. J. Keesman, and L. Speelman. 1998. Digital filters to integrate Global Positioning System and dead reckoning. *J. Agric. Eng. Res.* 70(2): 135-143.

Borenstein, J. 1998. Experimental evaluation of a fiber optics gyroscope for improving dead-reckoning accuracy in mobile robots. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, 3456-3461. Piscataway, N.J.: IEEE.

Bourke, P. 1998. Autoregressive analysis (AR). Available at: http://astronomy.swin.edu.au. Hawthorn, Australia: Swinburne University of Technology, Centre for Astrophysics and Supercomputing. Accessed 05 May 2005.

Guo, L., H. Yong, Z. Qin, and H. Shufeng. 2002. Real-time tractor position estimation system using a Kalman filter. *Trans. CSAE* 18 (5): 96-101.

Have, H., J. Nielsen, S. Blackmore, and F. Theilby. 2005. Development and test of an autonomous Christmas tree weeder. In *Proc. 5th European Conference on Precision Agriculture*, 629-635. Uppsala, Sweden: Uppsala University and the Swedish University for Agricultural Sciences.

Johnson, R. A., and D. W. Wichern. 2002. *Applied Multivariate Statistical Analysis.* New Jersey, N.J.: Pearson Education.

Kiriy, E., and M. Buehler. 2002. Three-state extended Kalman filter for mobile robot localization. Technical Report. Montreal,

Quebec, Canada: McGill University, Centre for Intelligent Machines (CIM). Available at: www.cim.mcgill.ca/. Accessed 25 March 2005.

Lauro, O., H. Chung, and J. Borenstein. 2000. Precision-calibration of fiber-optics gyroscopes for mobile robot navigation. In *Proc. 2000 IEEE Intl. Conf. on Robotics and Automation*, 2064-2069. Piscataway, N.J.: IEEE.

Melander, B., I. A. Rasmussen, and, P. Bàrberi. 2005. Integrating physical and cultural methods of weed control: Examples from European research. *Weed Sci.* 53(3): 369-381.

Sukkarieh. S., E. M. Nebot, and H. F. Durrant Whyte. 1999. A high-integrity IMU/GPS navigation loop for autonomous land vehicle applications. *IEEE Trans. Robotics and Automation* 15(3): 572-578.

Welch, G., and G. Bishop. 2004. An introduction to the Kalman filter. Technical Report TR95 041. Chapel Hill, N.C.: University of North Carolina-Chapel Hill. Available at: www.cs.unc.edu/~welch/media. Accessed 11 March 2005.

Will, J. 2001. Sensor fusion for field robot localization. PhD diss. Urbana, Ill.: University of Illinois at Urbana-Champaign, Department of Electrical and Computer Engineering.