Three-state Extended Kalman Filter for Mobile Robot Localization

Evgeni Kiriy
kiriy@cim.mcgill.ca

Martin Buehler
buehler@cim.mcgill.ca

April 12, 2002

Summary

This report describes the application of an extended Kalman filter to localization of a golf course lawn mower using fiber-optic gyroscope (FOG), odometry, and machine vision sensors. The two machine vision cameras were used to extract angular measurements from the previously surveyed artificial ground-level markers on the course. The filter showed an average 0.352 m Cartesian error with a standard deviation of 0.296 m in position estimation and a mean error of 0.244^o and standard deviation of 3.488^o in heading estimation. The errors were computed with respect to the existing high-performance localization system.

1 Introduction

For an autonomous lawn mower to be used on a golf course its navigation system must provide accuracy and robustness to assure precision of the resulting mowing pattern. The mower state - its position and orientation in real time - is obtained by combining the available internal and external sensory information using an Extended Kalman Filter (EKF).

The internal sensor set consists of front wheel odometers[3] and a fiber optic gyroscope (FOG)[4], the external, absolute sensing is currently achieved by 2 cm accuracy NovAtel ProPack DGPS[5]. The internal and the absolute positioning information is fused in real time by an extended Kalman filter for localization and control of the mower. The existing DGPS high-performance localization system maintained 97% tracking error below a target value of 10 cm [7], and is used as a ground truth. At the same time the high cost of DGPS and its dependence on good satellite visibility calls for an alternative absolute localization system.

Absolute positioning based on passive or active landmarks could replace the high-cost DGPS. In addition such a system could also provide reliable absolute positioning in places where the GPS signal is degraded or unavailable. A vision-based passive marker detection was implemented as an alternative to DGPS absolute localization. The local marker approach is general, so the results of the experiment will estimate the localization accuracy obtainable by a marker-based system irrespective of the actual hardware implementation.

Two DFW-VL500[8] digital camera modules were mounted on the mower, one in the front and the back, as shown in Figure 1 on page 2. The markers were scattered so that there were three to four markers per 10 m^2 in the testing area, allowing the GPS signal to provide 2 cm accuracy positioning and the existing localization system could be used as the ground truth. so that GPS signal provided 2 cm accuracy positioning and the existing localization system could be used as the ground truth)

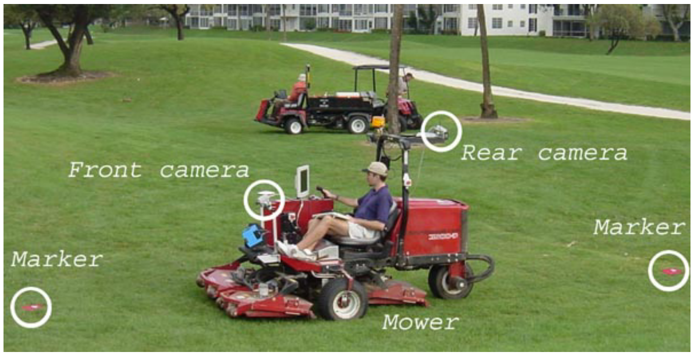


Figure 1: The experimental setup: the mower (foreground) is equipped with the cameras in the front and in the back; the red markers are in front and in the back of the mower. The mower is manually driven during the experiment.

The mower was manually driven in a spiral-like trajectory over 50 m by 75 m open flat area.

It is assumed that each marker is identifiable and its position is known with at least centimeter level accuracy (actual marker locations were surveyed by the NovAtel DGPS system).

2 Image Processing at CMU

The image processing was done at Carnegie Mellon University (CMU), by Parag Batavia and Jeff Mishler. Figure 2 on page 3 shows a typical camera image and is referred to in the overview of the process, according to Jeff Mishler:

The mower was driven on a spiral pattern on a field with 28 markers placed at roughly six meter intervals. During the motion 181 image points were extracted and 181 sets of angles are produced using the estimated pose, extracted image points, camera intrinsic and extrinsic parameters, and the marker map.

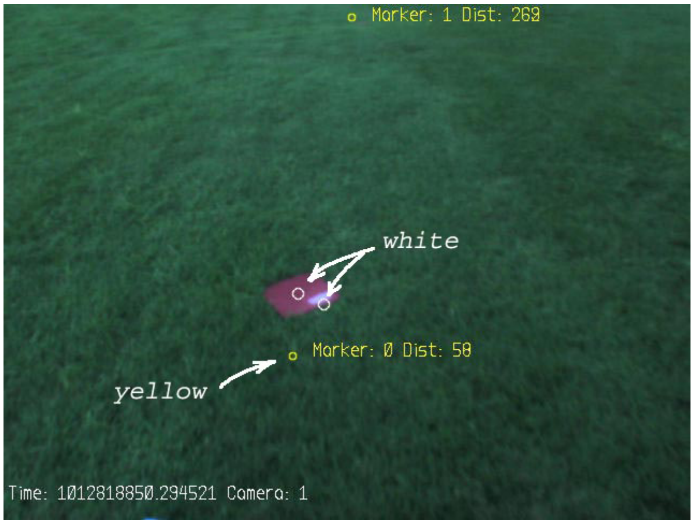


Figure 2: A typical frame taken by the on-board camera. The colors of the circles are indicated in the figure.

- The image name, camera ID, and time stamp are written to the "out-image.log" file as images are obtained and time stamped.

- The position information is logged with a time stamp as well, but the image and position time stamps are not identical. To associate the position information with the image, the first position time stamp which is greater or equal to the image time stamp is taken. This gives a reasonable approximation because the position data ("marker-001-poselog.txt" file) is obtained more frequently (about 100 Hz versus 0.5 Hz) than the image data.

- The images are run through a segmentation process to obtain the observed marker positions, which correspond to the white circles in the images. They should always line up with the marker in the image, but depend on the quality of the segmentation and may include false positives or missed targets. (This output is the "marker image locations" file, "marker-001-003-segmented.txt".)

- The identity of the markers observed in the image is established. To identify the markers and determine where the target should appear in the given image the information about the vehicle's location, the marker locations, and the camera locations is used. This involves projecting the 3D marker position onto the camera's image plane, and corresponds to the yellow circles in the images. If all the data were perfect (segmentation, vehicle position, marker location, camera extrinsic and intrinsic parameters) the yellow circle would be inside the white circle. Our data is not perfect, so the yellow circles do not fall exactly on the white circles. The marker ID of the closest yellow circle (that is less than 64 pixels away from the white circle) is associated with this image observation. (This step can be eliminated if we have barcodes or other means of performing this data association.)

- If there is a successful match, the angles to this marker are computed and stored along with the time stamp in the “Marker angles” file, “marker-001-003-angles.txt”.

The data from “Marker angles” file is used as measurement input in the Extended Kalman Filter algorithm.

3 Discrete Extended Kalman Filter

The discrete Extended Kalman Filter [2] was used to fuse the internal position estimation and external measurements to the markers. The following are the general discrete Extended Kalman Filter [6] equations particular realizations of which for our system are given in the Subsection 3.1.

The general nonlinear system (Equation 1) and measurement (Equation 2) where $\mathbf{x}_k$ and $\mathbf{z}_k$ represent the state and measurement vectors at time instant k, $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ are the nonlinear system and measurement functions, $\mathbf{u}_k$ is the input to the system, $\mathbf{w}_{k-1}$, $\boldsymbol{\gamma}_{k-1}$ and $\mathbf{v}_{k-1}$ are the system, input and measurement noises:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k; \mathbf{w}_{k-1}, \boldsymbol{\gamma}_{k-1}) \tag{1}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k; \mathbf{v}_{k-1}) \tag{2}$$

removing the explicit noise descriptions from the above equations and representing them in terms of their probability distributions, the state and measurement estimates are obtained:

$$\hat{\mathbf{x}}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{0}, \mathbf{0}) \tag{3}$$

$$\hat{\mathbf{z}}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{0}) \tag{4}$$

The system, input and measurement noises are assumed to be Gaussian with zero mean and are represented by their covariance matrices $\mathbf{Q}$, $\boldsymbol{\Gamma}$, and $\mathbf{R}$:

$$p(\mathbf{w}) = N(0, \mathbf{Q}) \tag{5}$$

$$p(\boldsymbol{\gamma}) = N(0, \boldsymbol{\Gamma}) \tag{6}$$

$$p(\mathbf{v}) = N(0, \mathbf{R}) \tag{7}$$

The Extended Kalman Filter predicts the future state of the system $\hat{\mathbf{x}}_k^-$ based on the available system model $\mathbf{f}(\cdot)$ and projects ahead the state error covariance matrix $\mathbf{P}_k^-$ using the **time update equations**:

$$\hat{\mathbf{x}}_k^- = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{0}, \mathbf{0}) \tag{8}$$

$$\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{B}_k \boldsymbol{\Gamma}_{k-1} \mathbf{B}_k^T + \mathbf{Q}_{k-1} \tag{9}$$

Once measurements $\mathbf{z}_k$ become available the Kalman gain matrix $\mathbf{K}_k$ is computed and used to incorporate the measurement into the state estimate $\hat{\mathbf{x}}_k$. The state error covariance for the updated state estimate $\mathbf{P}_k$ is also computed using the following **measurement update equations**:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{10}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-, \mathbf{0})) \tag{11}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^- \tag{12}$$

Where $\mathbf{I}$ is an identity matrix and system ($\mathbf{A}$), input ($\mathbf{B}$), and measurement ($\mathbf{H}$) matrices are calculated as the following Jacobians of the system ($\mathbf{f}(\cdot)$) and measurement ($\mathbf{h}(\cdot)$) functions:

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{\mathbf{x}}_k^-, \mathbf{u}_k, \mathbf{0}, \mathbf{0}) \tag{13}$$

$$B_{[i,j]} = \frac{\partial f_{[i]}}{\partial u_{[j]}}(\hat{\mathbf{x}}_k^-, \mathbf{u}_k, \mathbf{0}, \mathbf{0}) \tag{14}$$

$$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}(\hat{\mathbf{x}}_k^-, \mathbf{0}) \tag{15}$$

3.1 Mower System Modeling

The mower's planar Cartesian coordinates (x, y) and heading (Φ) describe pose of the mower and are used as the state variables of the Kalman filter. The wheels encoder and the FOG measurements are treated as inputs to the system. The angles to the markers are treated as the measurements. The mower is modeled by the following kinematic equations representing the position of the mid-axis (x, y) and the orientation in the global frame (Φ) [9].

$$f_x = x_{k+1} = x_k + \Delta D_k \cdot cos(\Phi_k + \frac{\Delta\Phi_k}{2}) \tag{16}$$

$$f_y = y_{k+1} = y_k + \Delta D_k \cdot sin(\Phi_k + \frac{\Delta\Phi_k}{2}) \tag{17}$$

$$f_\Phi = \Phi_{k+1} = \Phi_k + \Delta\Phi_k \tag{18}$$

Where ΔD_k is the distance travelled by the mid-axis point given the values that the right and the left wheels have travelled, ΔD_{kR} and ΔD_{kL} respectively:

$$\Delta D_k = \frac{\Delta D_{kR} + \Delta D_{kL}}{2} \tag{19}$$

The incremental change in the orientation $\Delta\Phi_k$,can be obtained from odometry given the effective width of the mower b:

$$\Delta\Phi_k = \frac{\Delta D_{kR} - \Delta D_{kL}}{b} \tag{20}$$

Thus the system state vector may be written as $\mathbf{x}_k = [x_k \ y_k \ \Phi_k]^T$, the input vector as $\mathbf{u}_k = [\Delta D_{kL} \ \Delta D_{kR}]^T$ and the system function $\mathbf{f}(\mathbf{x}) = [f_x \ f_y \ f_\Phi]^T$ where the function components are represented by the Equations (16 to 18). The system ($\mathbf{A}_k$) and input ($\mathbf{B}_k$) Jacobians for our system are given below:

$$\mathbf{A}_k = \begin{bmatrix} \frac{\partial f_x}{\partial x_k} & \frac{\partial f_x}{\partial y_k} & \frac{\partial f_x}{\partial \Phi_k} \\ \frac{\partial f_y}{\partial x_k} & \frac{\partial f_y}{\partial y_k} & \frac{\partial f_y}{\partial \Phi_k} \\ \frac{\partial f_\Phi}{\partial x_k} & \frac{\partial f_\Phi}{\partial y_k} & \frac{\partial f_\Phi}{\partial \Phi_k} \end{bmatrix}_{\mathbf{x}_k} = \begin{bmatrix} 1 & 0 & -\Delta D_k \cdot \sin(\Phi_k + \frac{\Delta\Phi_k}{2}) \\ 0 & 1 & \Delta D_k \cdot \cos(\Phi_k + \frac{\Delta\Phi_k}{2}) \\ 0 & 0 & 1 \end{bmatrix}_{\mathbf{x}_k} \tag{21}$$

$$\mathbf{B}_k = \begin{bmatrix} \frac{\partial f_x}{\partial \Delta D_{kL}} & \frac{\partial f_x}{\partial \Delta D_{kR}} \\ \frac{\partial f_y}{\partial \Delta D_{kL}} & \frac{\partial f_y}{\partial \Delta D_{kR}} \\ \frac{\partial f_\Phi}{\partial \Delta D_{kL}} & \frac{\partial f_\Phi}{\partial \Delta D_{kR}} \end{bmatrix}_{\mathbf{x}_k} \tag{22}$$

$$= \begin{bmatrix} \frac{1}{2}\cos(\Phi_k + \frac{\Delta\Phi_k}{2}) + \frac{\Delta D_k}{2b}\sin(\Phi_k + \frac{\Delta\Phi_k}{2}) & \frac{1}{2}\cos(\Phi_k + \frac{\Delta\Phi_k}{2}) - \frac{\Delta D_k}{2b}\sin(\Phi_k + \frac{\Delta\Phi_k}{2}) \\ \frac{1}{2}\sin(\Phi_k + \frac{\Delta\Phi_k}{2}) - \frac{\Delta D_k}{2b}\cos(\Phi_k + \frac{\Delta\Phi_k}{2}) & \frac{1}{2}\sin(\Phi_k + \frac{\Delta\Phi_k}{2}) + \frac{\Delta D_k}{2b}\cos(\Phi_k + \frac{\Delta\Phi_k}{2}) \\ -1/b & 1/b \end{bmatrix}_{\mathbf{x}_k}$$

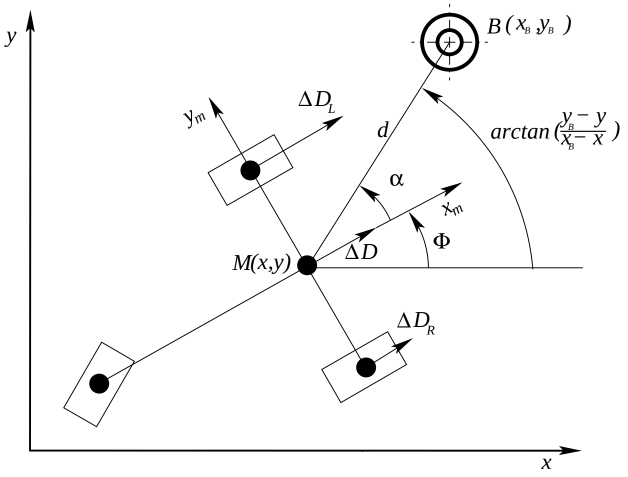


Figure 3: The system schematic (the bird's eye view.) The blocks represent the mower wheels and the concentric circles represent a marker.

The azimuth α_i with respect to the mower x-axis and elevation η_i with respect to the mower x–y plane (observed from the mid-axis point at the distance μ form the ground-plane) angles to the i-th marker obtained from the vision at a time instant k can be related to the current system state variables x_k, y_k, and Φ_k as follows:

$$\alpha_i = h_{\alpha i}(\mathbf{x}_k) = \arctan(\frac{y_{Bi} - y_k}{x_{Bi} - x_k}) - \Phi_k \tag{23}$$

$$\eta_i = h_{\eta i}(\mathbf{x}_k) = -\arctan(\mu / d_i(\mathbf{x}_k)) \tag{24}$$

$$d_i(\mathbf{x}_k) = \sqrt{(x_{Bi} - x_k)^2 + (y_{Bi} - y_k)^2}$$

and the measurement matrix H_k is obtained:

$$\mathbf{H}_{ki} = \begin{bmatrix} \frac{\partial h_{\alpha i}}{\partial x_k} & \frac{\partial h_{\alpha i}}{\partial y_k} & \frac{\partial h_{\alpha i}}{\partial \Phi_k} \\ \frac{\partial h_{\eta i}}{\partial x_k} & \frac{\partial h_{\eta i}}{\partial y_k} & \frac{\partial h_{\eta i}}{\partial \Phi_k} \end{bmatrix}_{\mathbf{x}_k} \tag{25}$$

$$\begin{aligned}
\frac{\partial h_{\alpha i}}{\partial x_k} &= \frac{y_{Bi} - y_k}{(x_{Bi} - x_k)^2 + (y_{Bi} - y_k)^2} \\
\frac{\partial h_{\alpha i}}{\partial y_k} &= \frac{-x_{Bi} + x_k}{(x_{Bi} - x_k)^2 + (y_{Bi} - y_k)^2} \\
\frac{\partial h_{\alpha i}}{\partial \Phi_k} &= -1 \\
\frac{\partial h_{\eta i}}{\partial x_k} &= \frac{\mu(x_{Bi} - x_k)}{\sqrt{(x_{Bi} - x_k)^2 + (y_{Bi} - y_k)^2}(\mu^2 + (x_{Bi} - x_k)^2 + (y_{Bi} - y_k)^2)} \\
\frac{\partial h_{\eta i}}{\partial y_k} &= \frac{\mu(y_{Bi} - y_k)}{\sqrt{(x_{Bi} - x_k)^2 + (y_{Bi} - y_k)^2}(\mu^2 + (x_{Bi} - x_k)^2 + (y_{Bi} - y_k)^2)} \\
\frac{\partial h_{\eta i}}{\partial \Phi_k} &= 0
\end{aligned}$$

Now, once all the components of the extended Kalman filter are defined, the particular filter realization is described.

3.2 Filter Realization

The discrete Extended Kalman Filter (EKF) presented here works off-line on the experimental data. The mower's sensory information (apart from the marker measurements) is stored at about 100 Hz in the "marker-001-poselog.txt" [1] file, the angles to the markers obtained from vision are stored at about $0.5Hz$ in the "marker-001-003-angles.txt" file, and the marker map is stored in the "marker-001-map.txt" file. The position and orientation output of the existing high-performance positioning is taken to be the ground truth. The EKF initial state $\mathbf{x}_0$ is taken to be equal to that of the ground truth. The initial state error covariance matrix is initialized to the value of the expected system error noise covariance: $\mathbf{P}_0^- = \mathbf{Q}$.

The time update stage of the EKF estimates (Equations 8 and 9) the system state $\hat{\mathbf{x}}_k = [\hat{x}_k \ \hat{y}_k \ \hat{\Phi}_k]^T$ using dead reckoning Equations 16, 17, and Equation 18 on page 5 that uses the fiber-optic gyroscope angular rate as

[1] due to non-real-time operation of the on-board computer, the time stamp of the *poselog.txt* file is non-uniform and contains identical time stamps. On the other hand, replacing this time stamp by a uniform one results in almost 5 seconds time lag over about 9 minute (549 seconds) interval, which will introduce large error in the data association required to incorporate the angle to the markers measurements. Thus, assuming that at the high ($100Hz$) data rate the non-uniformness could be neglected in favor of more precise data association, the original time stamp is used.

input independent of odometry. The covariance matrix of the prior estimate is calculated by the formula:

$$\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \sigma_\gamma^2 \mathbf{B}_k \mathbf{B}_k^T + \mathbf{Q} \tag{26}$$

where σ_γ^2 is the the input noise variance (the encoders are assumed to have the same noise variance), and $\mathbf{Q}$ — the 3×3 covariance matrix of the system noise is taken to be time-invariant.

The measurement update stage of the EKF operates only when the measurement is available. To associate the position information with the measurement, the first measurement with the image time stamp which is less or equal to the position time stamp is taken.[2] The measurement matrix $\mathbf{H}_k$ (Equation 25) and the Kalman $\mathbf{K}_k$ gain are computed as follows:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R})^{-1} \tag{27}$$

Where the measurement noise covariance matrix $\mathbf{R}$ is taken to be time-invariant (a simple assumption.) The measurement is incorporated by the estimate update in Equation 11 on page 5, and the state error covariance matrix for the updated estimate is calculated by a more computationally-stable version of Equation 12 on page 5 — the *Joseph form* [1]:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \tag{28}$$

The dead reckoning data of the “poselog.txt” file is given at much higher frequency (about 200 times more frequent), than the information in the angles.txt file, thus the time update and the measurement update parts of the EKF operate at different rates: the measurement can only be incorporated when it is available, that is approximately once every two seconds. When the measurement is not available the *a priori* state estimate and state error covariance matrix of the time update stage are used as virtual *a posteriori* state estimate and state error covariance matrix for the next iteration of the filter:

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}^-$$
$$\hat{\mathbf{P}} = \hat{\mathbf{P}}^-$$

Since the measurement errors to different markers are assumed to be uncorrelated, the measurements to markers visible at a given time instant may be computed consecutively, as opposed to simultaneously which allows to reduce the computational complexity and to facilitate tracking of the intermediate results. For sequential processing of the measurements available at a given time the *measurement update* equations are used iteratively replacing the *a priori*

[2] This is in accordance with the data association convention used in image processing — the first position timestamp which is greater or equal to the image time stamp is taken there.

values of the state estimate and its error covariance matrix by the updated values after each iteration:

$$\hat{\mathbf{x}}^- = \hat{\mathbf{x}}$$
$$\hat{\mathbf{P}}^- = \hat{\mathbf{P}}$$

4 Localization Simulations

The filter was used to estimate the mower state $\hat{\mathbf{x}}$ (position and orientation in planar motion) by fusing the odometry, gyroscope, and the angle information from the absolute marker detection. The algorithm was run on the experimental as well as on the simulated ideal data that would have been obtained if the measurements had been perfect.

The system noises are assumed to be uncorrelated and time-invariant, therefore the system noise covariance matrix is chosen to be diagonal and time-invariant. The system position noise standard deviation for the x and y coordinates was taken to be $\sigma_x = \sigma_y = 0.01\,\mathrm{m}$ (variances $\sigma_x^2 = \sigma_y^2 = 10^{-4}\,\mathrm{m}^2$), and the orientation noise standard deviation $\sigma_\Phi = 0.5^o$ (variance $\sigma_\Phi^2 = 7.62 \cdot 10^{-5}\,\mathrm{rad}^2$), therefore the system noise covariance matrix:

$$\mathbf{Q} = \begin{bmatrix} 10^{-4} & 0 & 0 \\ 0 & 10^{-4} & 0 \\ 0 & 0 & 7.62 \cdot 10^{-5} \end{bmatrix} \tag{29}$$

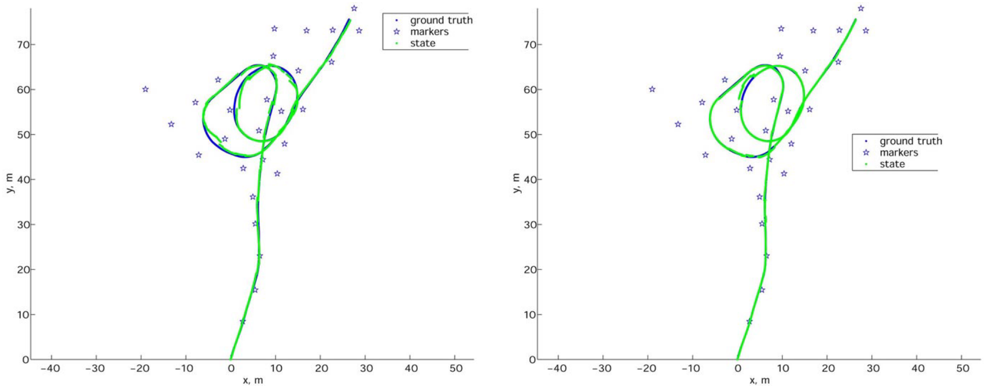
The experimental marker azimuth (α) and elevation (η) measurements were compared to the simulated set of measurements that would have been obtained at the time instants of the real measurements if the sensors used had been perfect. The variances of the differences between the experimental and simulated marker angle measurements were taken as characteristics of the measurement data quality: $\sigma_\alpha^2 = (0.048\,\mathrm{rad})^2 = 2.30 \cdot 10^{-3}\,\mathrm{rad}^2$ for the azimuth (α) angle measurement and $\sigma_\eta^2 = (0.006\,\mathrm{rad})^2 = 3.60 \cdot 10^{-5}\,\mathrm{rad}^2$ for the elevation (η) angle measurement; thus the measurement noise covariance matrix:

$$\mathbf{R} = \begin{bmatrix} 2.30 \cdot 10^{-3} & 0 \\ 0 & 3.60 \cdot 10^{-5} \end{bmatrix} \tag{30}$$

For the localization using the ideal angle measurements (flat ground was assumed) the confidence in the measurements was expressed by small values of the noise covariance entries:

$$\mathbf{R} = \begin{bmatrix} 1 \cdot 10^{-12} & 0 \\ 0 & 1 \cdot 10^{-12} \end{bmatrix} \tag{31}$$

The reference (ground truth) and the estimated position are shown in Figure 4 on page 10. The trace[3] of the state error covariance matrix $\mathbf{P}$ decreases as



(a) Real angle to marker measurements used

(b) Ideal angle to marker measurements used

Figure 4: Ideal and estimated paths. The markers are represented by the stars; the ground truth (reference) path is darker, the estimated path is lighter.

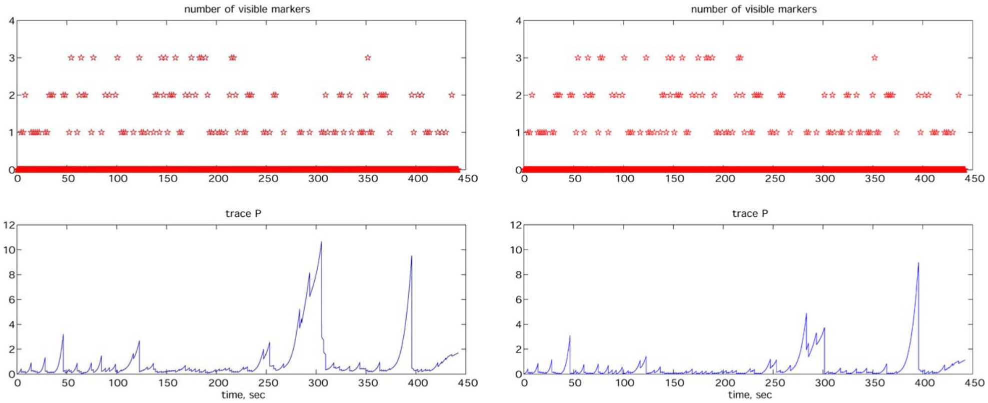
the filter converges. The trace decreases with each measurement reading as shown in Figure 5 on page 11.

The position and orientation differences between the estimated and reference states are presented in Figure 6 on page 11. The error statistics are gathered in Table 1 on page 11. The ideal marker measurements do not make the position estimation perfect, although they reduce the mean position error and standard deviation of position and heading by half. This can be explained by the inherently inexact time data association (the "ideal" measurements are taken at the time instances of the real measurements that do not exactly correspond to the time stamp of the rest of the data) and the system model state error accumulation between the estimates. The largest position and orientation error (in the $285 - 295$ second interval) corresponds to the trajectory interval with no marker or single marker visibility (see Figure 7, page 12). It is seen that the measurement accuracy and proper time association of the measurement is critical in the curved sections.

5 Conclusion

The developed Extended Kalman Filter for fusion of the odometry, fiber-optic gyroscope, and the angular measurements to the markers (obtained from the video frames taken during motion) showed an average 0.352 m Cartesian error with standard deviation 0.296 m and a heading mean error of 0.244^o and a standard deviation 3.488^o. The error measurements were taken with respect

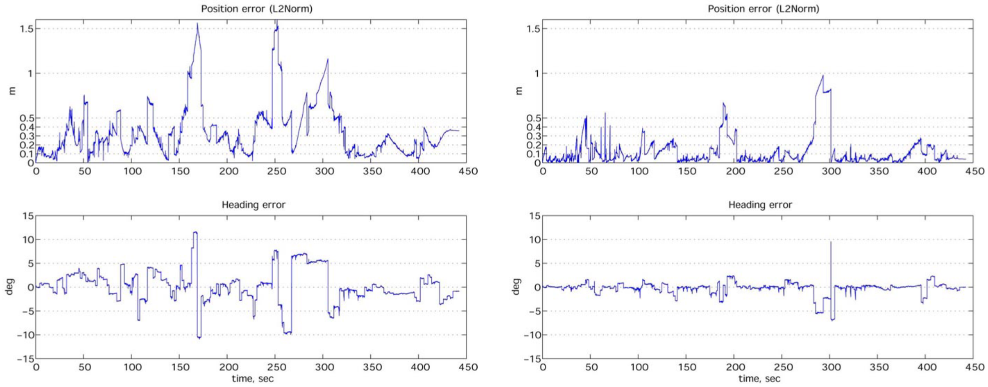
[3] The sum of the elements in the main diagonal; here these elements correspond to the covariances of the estimated states and thus the sum is indicative of the uncertainty of the state.



(a) Real angle to marker measurements used

(b) Ideal angle to marker measurements used

Figure 5: The number of visible markers and the trace of the state error covariance matrix. The filter converges once the marker measurements are taken. Long interruptions in marker data flow result in filter divergence.



(a) Real angle to marker measurements used

(b) Ideal angle to marker measurements used

Figure 6: The position and heading errors.

	Position and Heading Error Statistics			
	Real marker measurements		*Ideal marker measurements*	
	Position Err. *meters*	*Heading Err.* *degrees*	*Position Err.* *meters*	*Heading Err.* *degrees*
Mean	0.352	0.244	0.123	-0.233
Standard deviation	0.296	3.488	0.171	1.301
Variance (unit^2)	0.088	2.168	0.029	1.695

Table 1: Position and Heading Error statistics.

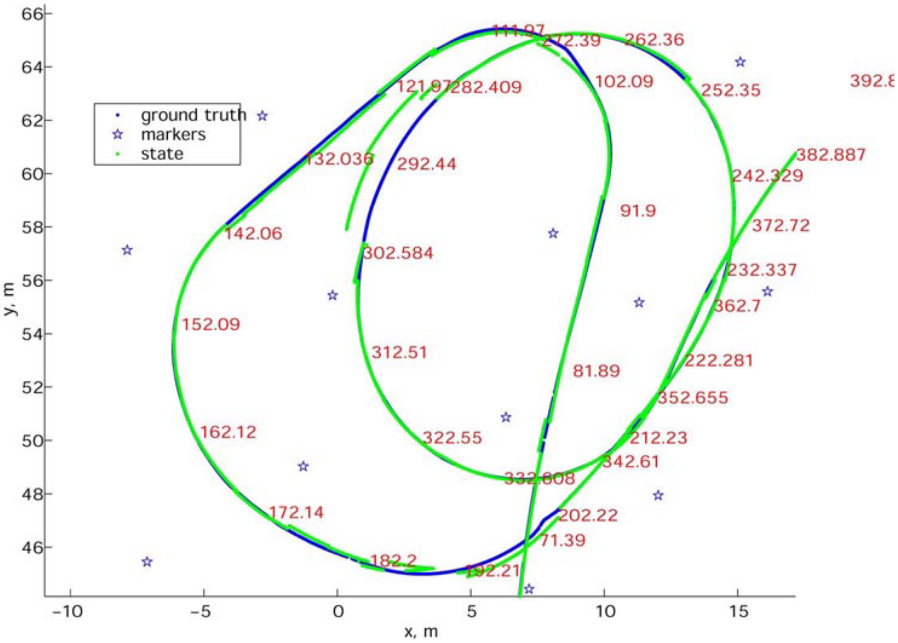


Figure 7: The close-up to the spiral region of the trajectory. Position estimation is done using the "ideal" marker measurements. The numbers along the trajectory correspond to the time from the beginning of the experiment. Note that the large jump at 285-300 seconds corresponds to single or no marker visibility.

to the ground truth trajectory. The filter performance can be improved by:

- On line adjustment of the system ($\mathbf{Q_k}$) and measurement ($\mathbf{R_k}$) covariance matrices based on the statistical properties of the incoming data.
- Extending the state of the filter to include the translational and rotational velocities.
- Improved real time data correlation.
- Increased external measurement data frequency.
- Improved external data precision.

6 Acknowledgements

The authors would like to thank the CMU team (Sanjiv Singh, Jeff Mishler, Parag Batavia, and Stephan Roth) for taking the data, processing, and providing us with the data that went into this report. We also thank Dana Lonn for sponsoring this work and making the trip to Palm Aire possible.

References

[1] *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions*, chapter 6.6, page 261. John Wiley Sons, Inc., third edition, 1997.

[2] Ph. Bonnifait and G. Garcia. "A Multicensor Localization Algorithm for Mobile Robots and its Real-Time Experimental Validation". In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 1395–1400, Minneapolis, Minnesota, April 1996. IEEE.

[3] Dynaher Corporation, http://dynapar-encoders.com. *Dynapar encoder series H20 data sheet.*

[4] KVH, http://www.kvh.com/Products/Product.asp?id=39. *KVH2060 data sheet*, 2000.

[5] NovAtel Inc., http://www.novatel.com/Documents/Papers/Rt-2.pdf. *NovAtel ProPack data sheet*, 2000.

[6] P. Y. C. Hwang R. G. Brown. *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions.* John Wiley Sons, Inc., third edition, 1997.

[7] Stephan Roth, Parag Batavia, and Sanjiv Singh. Palm Aire Data Analysis. report, February 2002.

[8] Sony, http://www.sony.co.jp/en/Products/ISP/pdf/catalog/DFWV500E.pdf. *Sony DFW-VL500 Digital Camera Module data sheet.*

[9] C. Ming Wang. “localization estimation and uncertainty analysis for mobile robots”. In *IEEE Int. Conf. on Robotics and Automation*, Philadelphia, April 1988.

A MATLAB Code Listing

```
% this is an extended Kalman filter driver script
% with sequential measurements incorporation.
switch 1
case 1
clear all
% Load the input files:
%========================================================================
positionLog_o  =load(’marker-001-poselog.txt’);
% [timeStamp, MarkerID, azimuth, inclination, vehSpeed, headingRate; ...]
finalMarkerLog  =load(’marker-001-003-angles.txt’);
%finalMarkerLog =load(’FinalMarkerLog’);
%MarkerMap file: [markerID, markerX, markerY; ...]
markerMap      =load(’marker-001-map.txt’);
%========================================================================
case 2
clear state
end %switch

bgn=10464; % first element of the positionLog to be considered;
endd=bgn+10000;
positionLog=positionLog_o(bgn:end,:);

% get phi within +/- 2pi range (wrap)
positionLog(:,21)=mod(positionLog(:,21),sign(positionLog(:,21))*2*pi);

%hight of the reference point of the mower.
h=0.226;
% clock (generate times at 0.01 sec for the length of the experiment)
%========================================================================
switch 2 % 1) 0.01 sec time stamp. 2) original time stamp. 3) equalized time step.
case 1
dT=0.01;
time=0:dT:dT*(size(positionLog,1)-1);
timePosLog=time’+positionLog(1,3);
case 2
```

```
dT=0.01;
timePosLog=positionLog(:,3);
case 3
dT=(positionLog(end,3)-positionLog(1,3))/size(positionLog,1);
timePosLog=0:dT:dT*(size(positionLog,1)-1);
timePosLog=timePosLog+positionLog(1,3);
end %switch
timeMarker=finalMarkerLog(:,1);
%========================================================================
b=1.41; %b=1.65-0.24; % mower effective width (Hal)

% Initialize the Kalman filter:
%========================================================================
%enter loop with prior state estimate and its error covariance
%x(0)projected or prior; supplied form the driver script.
%P(0)its error covariance; supplied form the driver script.
% Initial state guess:[xM; yM; Phi];
initHead=mean(positionLog(1+1:100,21));
initHead=mod(initHead,sign(initHead)*2*pi); % get phi within +/- 2pi range (wrap)
x=[positionLog(1,19);
positionLog(1,20);
   initHead];
PhiOdom=initHead;
dPhiOdom=0;

% System noise covariance:
Q11=1e-4*1.00;
Q22=1e-4*1.00;
Q33=7.62e-5*1.00;
Q=diag([Q11,Q22,Q33]);

% Error covariance matrix:
P_initial=Q;
%P_initial=diag([1e-4,1e-4,1e-4]);
traceP_initial=trace(P_initial);
traceResetValue=traceP_initial*0.5;
P=P_initial;

% Build the measurement error covariance martix:
azimuthCov=0.048^2;
inclinCov=0.006^2;
singleSensorCov=[azimuthCov inclinCov];
R=[];
R=diag(singleSensorCov);
```

```
varIn=1e-4; % input noise covariance
%==========================================================================
numOfVisibleMarkers=zeros(size(positionLog,1),1);
msrmtCnt=0;

for i=1:size(positionLog,1) %main
% store state
state(:,1,i)=x;
%==========================================================================
% input:
dDl=positionLog(i,4);     % [m]
dDr=positionLog(i,5);     % [m]
dPhiFOG=positionLog(i,9); % [rad/sec]
%==========================================================================
%===========================================================================
%the  Kalman filter
% The state vector:
xM=x(1);
yM=x(2);
Phi=x(3);
dD=(dDl+dDr)/2;
%===========================================================================
% TIME UPDATE EQUATIONS (predict)
% EKF time update equations:
%x(k+1)project ahead=f(x(k)estim updated w meas);
%P(k+1)project ahead=A(k)*P(k)*A(k)'+Q(k);
argmnt=Phi+dPhiFOG/2*dT;
x(1)=xM+dD*cos(argmnt);
x(2)=yM+dD*sin(argmnt);
x(3)=Phi+dPhiFOG*dT;
x(3)=mod(x(3),sign(x(3))*2*pi); % get phi within +/- 2pi range (wrap)

% Build the system matrix A:
A=eye(3,3);
%A=zeros(3,3);
A(1,3)=-dD*sin(argmnt); A(2,3)=dD*cos(argmnt);

% Build the input matrix B:
cPhi=0.5*cos(argmnt); dDs=dD/2/b*sin(argmnt);
sPhi=0.5*sin(argmnt); dDc=dD/2/b*cos(argmnt);
B(1,1)=cPhi+dDs; B(1,2)=cPhi-dDs;
B(2,1)=sPhi-dDc; B(2,2)=sPhi+dDc;
B(3,1)=-1/b; B(3,2)=1/b;

P=A*P*A'+varIn*B*eye(2,2)*B'+Q;
```

```
%P=A*P+P*A'+Q;
%============================================================================
%============================================================================
% Now, if a measurement is available - incorporate it:
if ~isempty(finalMarkerLog) % check if markers exist
% find the measurements corresponding to the current time instance:
% according to CMU association: t_posLog>=t_marker
measurementEntries=find(timeMarker>=timePosLog(i)-0.01 & timeMarker<=timePosLog(i));
% Search the final marker log for the time stamps equal to the curent time
%and retrieve the corresponding data entry numbers:
if ~isempty(measurementEntries) % if there exist contemporary measurements:
numberOfMeas=length(measurementEntries);
numOfVisibleMarkers(i)=numberOfMeas;

markerID=finalMarkerLog(measurementEntries,2)+1;

for n=1:numberOfMeas %loop for multiple measurements:

% Build the measurement martix:
xB=markerMap(markerID(n),2);
yB=markerMap(markerID(n),3);
H=[]; distSq=(xB-xM)^2+(yB-yM)^2; SqrtDistSq=sqrt(distSq);
H(1,1)=-(-yB+yM)/( distSq );
H(1,2)=-( xB-xM)/( distSq );
H(1,3)=-1;
H(2,1)=-h*(xB-xM)/(SqrtDistSq*(h^2+distSq));
H(2,2)=-h*(yB-yM)/(SqrtDistSq*(h^2+distSq));
H(2,3)=0;

% measurement function:
hx=[];
hx(1,1)=atan2(yB-yM,xB-xM)-Phi;    %azimuth; non-linear version
hx(2,1)=-atan(h/SqrtDistSq);       %inclination; non-linear version
% measurement:
z=[finalMarkerLog(measurementEntries(n),3);
finalMarkerLog(measurementEntries(n),4)];

% Compute Kalman Gain:
%K(k)=P(k)projected*H(k)'*inv(H(k)*P(k)projected*H(k)'+R(k));
%R - variance of the measurement noise
K=P*H'*inv(H*P*H'+R);

condK=cond(K);
if condK>100,
disp(['condition number for K is > 100; it is ' num2str(condK) ' at step ' num2str(i)]);
```

```
end

% Update estimate with measurement y(k):
%x(k)=x(k)projected + K(k)*(y-H(k)*x(k)projected);%x=x+K*(z-hx); % extended KF
estAzimuth=hx(1);
%bring to +/- 180
if abs(estAzimuth)>pi, estAzimuth=-sign(estAzimuth)*2*pi+estAzimuth; end;
hx(1)=estAzimuth; %substitute the corrected alpha (azimuth) values.

innov=z-hx;
%bring alpha to +/- 180
if abs(innov(1))>pi, innov(1)=-sign(innov(1))*2*pi+innov(1); end;
corr=K*innov;

x=x+corr;

% Compute error covariance for updated estimate:
%P(k)=(I-K(k)*H(k))P(k)projected;
%P=(eye(size(K,1))-K*H)*P;
%"Joseph form" - better numerical behavior (Brown p.261)
P=(eye(size(K,1))-K*H)*P*(eye(size(K,1))-K*H)'+K*R*K';
end % loop for multiple measurements (sequential meas. incorporation)

end % if there exist contemporary measurements;
end % if markers exist
%=======================================================================
% end of the Kalman filter

%check for filter divergence (+ve definiteness of the state error covariance matrix):
if eig(P)<0,
disp(['covariance matrix P is not positive-definite in step ' num2str(i)]);
end
traceP(i)=trace(P); %(becomes smaller as filter converges, see Greg's p.25)
xVar(i)=P(1,1);
yVar(i)=P(2,2);
PhiVar(i)=P(3,3);
end %main
state = squeeze(state);
```