Pose Estimation of Ackerman Steering Vehicles for Outdoors Autonomous Navigation

Alejandro J. Weinstein, Kevin L. Moore
Colorado School of Mines
Division of Engineering
1500 Illinois St.
Golden, CO 80401, USA
Email: kmoore@mines.edu

***Abstract*—This paper presents a localization scheme for Ackerman steering vehicles, to be used in outdoors autonomous navigation, using a low cost GPS and inclinometer. A complementary filter fuses the bearing from the inclinometer with the bearing of the GPS. We then use an Extended Kalman Filter to estimate the pose of the vehicle and the sensor biases. We validate our system with experimental results.**

I. Introduction

As stated by Leonard and Durrant-Whyte in 1991 [1], answering the question “Where I am?” is the first step to successfully solve an autonomous navigation problem. In outdoor localization problems, due to its inherent advantages, the Global Positioning System (GPS) plays a central role in the majority of the solutions. To overcome the GPS weak points (temporary loss of the GPS signal and multipath errors), its measurement is usually fused with an Inertial Measurement Unit (IMU) [2].

The problem of using a GPS with other sensors for robot localization has been addressed by many researchers. Bonnifait *et al.* fused the four ABS sensors of a car with GPS measurement [3]. Georgiev and Allen used a differential GPS (DGPS), a camera, a compass, and a tilt sensor to estimate the pose of a skid-steered robot [4]. Bevly and Parkinson used a DGPS, a radar and a gyroscope to estimate the pose of a tractor, using an Extended Kalman Filter [5]. In general, most current localization implementations use expensive DGPS units, which provide submeter accuracy, and high-grade IMUs. For instance, the winner of the 2007 Darpa Urban Challenge, team Tartan Racing [6], used a Applanix POS-LV unit [7], which integrates a Trimble DGPS, a Honeywell IMU, and the vehicle speed to produce an estimate of the position and orientation. However, this kind of solution is cost prohibitive in many cases. In this paper, we explore the use of low-cost, off-the-shelf sensors to produce a pose estimation suitable for autonomous navigation.

We use an Extended Kalman Filter (EKF) to estimate the pose of the vehicle, and a GPS and an IMU as the sensors of the system. We also use a string potentiometer and an encoder to determine the vehicle steering angle and the speed, respectively. At the heart of the model used in the EKF is the kinematic model of the Ackerman vehicle. We augment this model with the bias of the sensors. We also exploit the key observation that the GPS provides a good but slow measurement of the heading, while the IMU provides a good dynamic measurement, by fusing these two measurements through a complementary filter.

This paper is organized as follows: Section II presents the model used for the vehicle and the sensors. Section III shows the proposed pose estimation scheme. Section IV describes the implementation details and the experimental results. Section V concludes the paper.

II. Vehicle and sensor model

The model of the vehicle is based on the Ackerman steering geometry. While purely a kinematic model, it is a good representation for slow speeds [8]. Figure 1 shows the vehicle scheme and the main variables. The pose is defined by its position (x, y) and the bearing θ. The actuations are the car speed v and the steering angle δ. The only parameter is the wheelbase L. The model is

$$\dot{x} = v\cos(\psi) = v\cos\left(\frac{\pi}{2} - \theta\right) \tag{1a}$$

$$\dot{y} = v\sin(\psi) = v\sin\left(\frac{\pi}{2} - \theta\right) \tag{1b}$$

$$\dot{\theta} = \frac{v}{L}\tan(\delta). \tag{1c}$$

A low cost inclinometer, based on a IMU and a magnetometer, provides a measurement of the bearing; an off-the-shelf

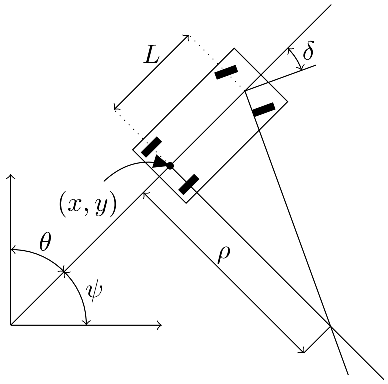

Fig. 1. Ackerman model.

978-1-4244-5697-0/10/$25.00 ©2010 IEEE

GPS provides a measurement of the position. We are also using the bearing computed by the GPS, which is available in one of the standard NMEA messages [9]. In general, we are modeling the output of a sensor as the actual variable plus a bias, which in turn is modeled as a Wiener process [10]. Figure 2 shows the general scheme for the sensor model.

It is well known that bearing is a critical measurement for localization [11], since small errors integrate over time. Our inclinometer has an accuracy of 0.5° and 2° under static and dynamic conditions, respectively. Moreover, the inclinometer measures the magnetic bearing, which, due to the magnetic declination [12], has an offset with respect to true north. Magnetic declination varies along the earth and changes slowly over time. For instance, at our location the declination is 9.23° and changes 0.13° per year. On the other hand, the bearing output of the GPS is true north. So we have two measurements of the same variable, where one provides a fast signal but with a significant bias, and the other one provides a slow but accurate output. It is natural to fuse these two measurements through a complementary filter [10].

Figure 3 shows the complementary filter. The inclinometer signal θ^{IMU} passes through a high-pass filter, while the GPS signal θ^{GPS} passes through a low-pass signal. The output of the two signals are added to form the fused signal $\theta^{IMU+GPS}$. This fused signal combines the best of two worlds: the accuracy of the GPS and the fast response of the inclinometer. Figure 4 shows the result of the complementary filter while driving the car in an approximate straight line, with a known bearing of 324° . As expected, the low frequency component of the fused signal follows the bearing computed by the GPS, which matches the true bearing; at the same time, the fused signal incorporates the high frequency component of the IMU.

III. Pose Estimation

We define our pose estimation problem as estimating the vehicle state, defined as $x_t = [x\ y\ \theta]^T$, using wheel encoders, steering angle, COTS GPS and an IMU-based inclinometer. The states x and y are the Universal Transverse Mercator (UTM) [13] coordinates of the vehicle, and θ is the true bearing. Based on the taxonomy proposed by Thrun *et al.* [11], we classify our problem as:

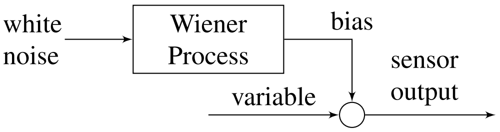

Fig. 2. Sensor model.

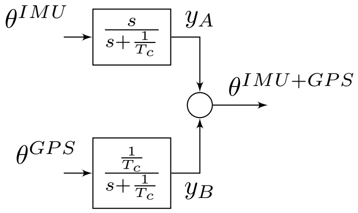
θ^{IMU}
$\frac{s}{s+\frac{1}{T_c}}$
y_A
$\theta^{IMU+GPS}$
θ^{GPS}
$\frac{\frac{1}{T_c}}{s+\frac{1}{T_c}}$
y_B

Fig. 3. Complementary filter.

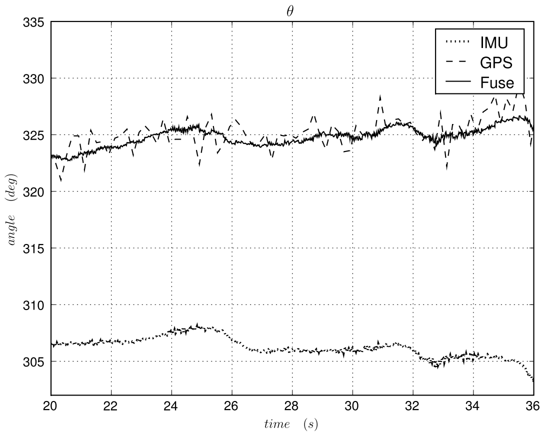

Fig. 4. Complementary filter results.

- Position tracking problem: The initial pose of the vehicle is known, and its uncertainty is approximated as unimodal (Gaussian).
- Static environment.
- Passive localization: The estimator can only observe. It can't control the vehicle movement.
- Single vehicle localization.
- "Loosely Coupled System:" Only the position output from the GPS is used (no access to the internal GPS raw data).

As an estimator, we use an EKF [10]. The first step to use an EKF is to determine a model for the vehicle. The state equations for x_t are given by (1). These states must be augmented to incorporate both the sensor bias and the complementary filter. The inclinometer bias θ_B and GPS biases x_B and y_B correspond to Wiener processes:

$$\begin{aligned}\dot{\theta}_B &= G_1 w_1\\ \dot{x}_B &= G_2 w_2\\ \dot{y}_B &= G_3 w_3,\end{aligned}\tag{2}$$

where the w_i are white noise random processes with unitary spectral amplitude, and the G_i are the noise amplitude of each bias.

Let x_{hp} and x_{lp} be the states of the high pass and low pass filter of the complementary filter respectively. Referring to Fig. 3, we can write

$$\begin{aligned}\dot{x}_{hp} &= -\frac{1}{T_c}x_{hp} + \theta^{IMU}\\ y_{hp} &= -\frac{1}{T_c}x_{hp} + \theta^{IMU}\\ \dot{x}_{lp} &= -\frac{1}{T_c}x_{hp} + \frac{1}{T_c}\theta^{IMU}\\ y_{lp} &= x_{lp}.\end{aligned}\tag{3}$$

Since

$$\theta^{IMU+GPS} = y_{hp} + y_{lp}, \tag{4}$$

we can write a state space representation for the complementary filter as

$$\begin{aligned}
\begin{bmatrix} \dot{x}_{hp} \\ \dot{x}_{lp} \end{bmatrix} &= \begin{bmatrix} -\frac{1}{T_c} & 0 \\ 0 & -\frac{1}{T_c} \end{bmatrix} \begin{bmatrix} x_{hp} \\ x_{lp} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{T_c} \end{bmatrix} \begin{bmatrix} \theta^{IMU} \\ \theta^{GPS} \end{bmatrix} \\
\theta^{IMU+GPS} &= \begin{bmatrix} -\frac{1}{T_c} & 1 \end{bmatrix} \begin{bmatrix} x_{hp} \\ x_{lp} \end{bmatrix} + \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta^{IMU} \\ \theta^{GPS} \end{bmatrix}.
\end{aligned} \tag{5}$$

Figure 5 shows a diagram of the augmented model used for the EKF, described by (1), (2) and (5). Table I summarizes the inputs, outputs, and states.

An EKF requires the model to be expressed in the form:

$$\begin{aligned}
x_{k+1} &= \Phi_k x_k + \Gamma u_k \\
y_k &= H_k x_k + D u_k.
\end{aligned} \tag{6}$$

To obtain the Φ matrix, we start by finding a linearized continuous time state space representation of the form

$$\begin{aligned}
\dot{x} &= Ax + Bu \\
y &= Cx.
\end{aligned} \tag{7}$$

Notice, however, that only (1) is nonlinear. Moreover, since (1c) is only nonlinear with respect to the inputs, we only need to linearize (1a) and (1b):

$$\begin{aligned}
\dot{\Delta x} &= v \sin\left(\frac{\pi}{2} - \theta\right) \\
\dot{\Delta y} &= -v \cos\left(\frac{\pi}{2} - \theta\right).
\end{aligned} \tag{8}$$

TABLE I
DESCRIPTION OF THE MODEL VARIABLES.

Inputs	δ	Steering angle
	v	Vehicle speed
Outputs	θ^{IMU}	Bearing from the IMU
	$\theta^{IMU+GPS}$	Bearing from the complementary filter
	θ^{GPS}	Bearing from the GPS
	x^{GPS}	X from the GPS
	y^{GPS}	Y from the GPS
States	θ	Heading
	x	X component of the position
	y	Y component of the position
	θ_B	IMU bias
	x_B	X component of the GPS bias
	y_B	Y component of the GPS bias
	x_{hp}	High pass filter internal state
	x_{lp}	Low pass filter internal state

This leads to the following A matrix:

$$A = \begin{bmatrix}
0 & 0 & v_o \sin\left(\frac{\pi}{2} - \theta_o\right) & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -v_o \cos\left(\frac{\pi}{2} - \theta_o\right) & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{T_c} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{T_c}
\end{bmatrix}, \tag{9}$$

where v_o and θ_o are the current speed and bearing. Then the Φ matrix can be computed at each iteration as

$$\Phi = e^{AT_s}, \tag{10}$$

where T_s is the sampling time. The measurement matrix is given by

$$H_k = \begin{bmatrix}
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & \frac{-1}{T_c} & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix}. \tag{11}$$

However, the output rates of the sensors we are using are different: the inclinometer runs at 75[Hz], while the GPS run at 5[Hz]. Moreover, since the bearing output of the GPS is based on the relative movement of the unit between two sampling times, we only consider this measurement as valid if the vehicle is moving fast enough. We address this issue by using different measurement matrices depending on the available measurements. If the vehicle is going faster than 0.5[m/s] and there is available both GPS and inclinometer data, we use the H_k given by (11). If only an inclinometer measurement is available, we use

$$H_k = \begin{bmatrix}
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}. \tag{12}$$

If we have both an inclinometer and a GPS measurement, but the vehicle is not moving fast enough, we use

$$H_k = \begin{bmatrix}
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix}. \tag{13}$$

IV. EXPERIMENTAL RESULTS

We tested our algorithms in an electric drive mini-Baja [14] vehicle, equipped with a GPS and inclinometer unit, as shown in Fig. 6. In addition, a motor encoder and a string potentiometer allow us to measure the speed and steering angle, respectively. The motor drive and two motors attached to the steering column and the brake, are controlled by a

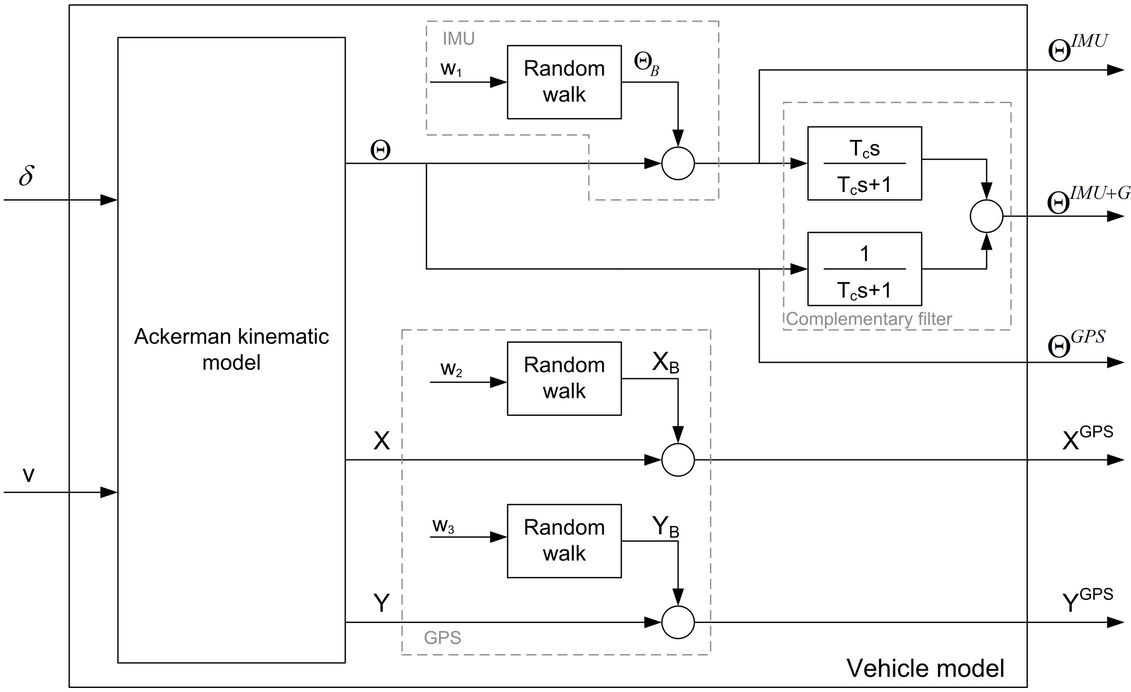

Fig. 5. Model used in the EKF.

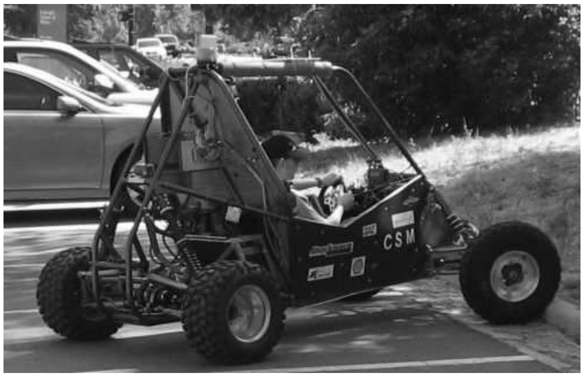

Fig. 6. Electric mini-Baja vehicle.

National Instrument CompactRIO controller, providing drive-by-wire capabilities.

The ultimate objective of our project is to be able to perform autonomous navigation, based on the pose estimated by our localization scheme. The localization and autonomous navigation algorithms are implemented in a single board computer with a Pentium-M processor. The code is written in Python, using a multi threaded architecture; the SciPy and NumPy libraries are used for all the matrix computations. A detailed description of the implementation is available in [15].

One of the threads of our program is dedicated to execute the EKF. This thread waits until there is a measurement available from the inclinometer (the fastest of the sensors); it also check if there is a GPS measurement available, and depending on that and on the current speed, it computes the appropriate measurement matrix H_k using equation (11), (12) or (13), according to the logic described in section III. It then computes matrices A and Φ, as described by equations (9) and (10), respectively. The estimate for the current iteration is then computed using the standard EKF equations:

$$\begin{aligned} K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^-) \\ P_k &= (I - K_k H_k) P_k^- \\ \hat{x}_{k+1}^- &= \phi_k \hat{x}_k + B u_k \\ P_{k+1}^- &= \phi_k P_k \phi_k^T + Q_k . \end{aligned} \tag{14}$$

Since we don't known the process covariance Q_k, we tuned this parameter experimentally, by trial and error.

To show the usefulness of our localization scheme for autonomous navigation, we implemented a simple autonomous controller. This controller read waypoints from a user-defined waypoints queue, where each element of the queue is a tuple with the UTM coordinates that must be visited. The controller reads coordinates from the queue, and use these values as setpoints. When the waypoint is reached a new setpoint is read from the queue. The process repeats until the queue is empty.

Figure 7 shows the variables used to define the control law. The idea is to control the throttle in proportion to the distance to the goal, and the steering in proportion to the relative bearing of the setpoint and the actual bearing. Thus, the control law is:

$$\text{Steering} = K_s(\psi^{sp} - \psi) = K_s\left(\arctan\left(\frac{y^{sp} - y}{x^{sp} - x}\right) - \psi\right)$$
$$\text{Throttle} = K_t\sqrt{(x^{sp} - x)^2 + (y^{sp} - y)^2}. \tag{15}$$

Due to the nonholonomic constrains of Ackermann steering, the control law defined by (15) will not work when the car is close to the goal. We overcome this by going to the next waypoint when the vehicle is close to the setpoint, or stopping if there are no more waypoints.

Figure 8 shows some results of the pose estimation algorithm for an experimental run consisting of following a set of waypoints inside a parking lot. For ground truth, we use a Trimble GPS Pathfinder Pro XR Receiver that provides an accuracy better than 15cm after applying a correction based in the Continuously Operating Reference Stations (CORS) data [16].

As can be seen, the EKF estimate is able to increase the accuracy of the GPS to about 1.5 meters part of the time, while converging to the GPS measurement the remainder of the time. To quantify the improvement of the estimate over the GPS measurement, we compute the Root Mean Square Error (RMSE) for both the GPS and the estimates:

$$\begin{aligned} GPS_{RMSE} &= \sqrt{\frac{1}{n}\left(x^{GPS} - x^{GT}\right)^2 + \left(y^{GPS} - y^{GT}\right)^2} \\ &= 1.8(m), \end{aligned} \tag{16}$$

$$\begin{aligned} Est_{RMSE} &= \sqrt{\frac{1}{n}\left(\hat{x} - x^{GT}\right)^2 + \left(\hat{y} - y^{GT}\right)^2} \\ &= 1.2(m). \end{aligned} \tag{17}$$

where x^{GT} and y^{GT} are the ground truth measurements, and n is the number of samples. Thus the scheme gives better results than using the GPS alone.

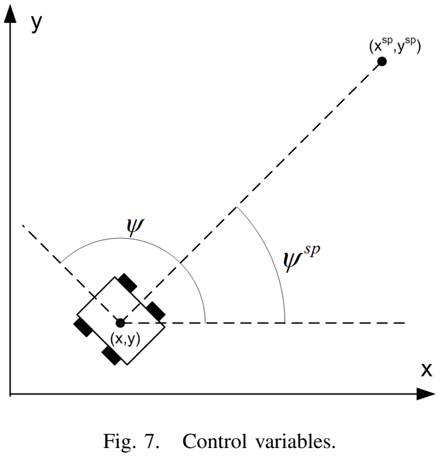

Fig. 7. Control variables.

V. Conclusion

This paper presented a localization scheme for Ackerman steering vehicles, based on an EKF, using a low-cost, off-the-shelf GPS, an IMU-based inclinometer, a motor encoder, and a string potetiometer. The model used for the EKF was augmented with the biases of the GPS position and the IMU. A complementary filter fused the bearing computed by the GPS with the inclinometer measurement, allowing estimation of the inclinometer bias. The proposed algorithm was tested experimentally in an electric vehicle, using the estimated pose to autonomously follow a set of waypoints. The estimated pose was more accurate than the GPS measurements, producing smaller RMS and peak errors.

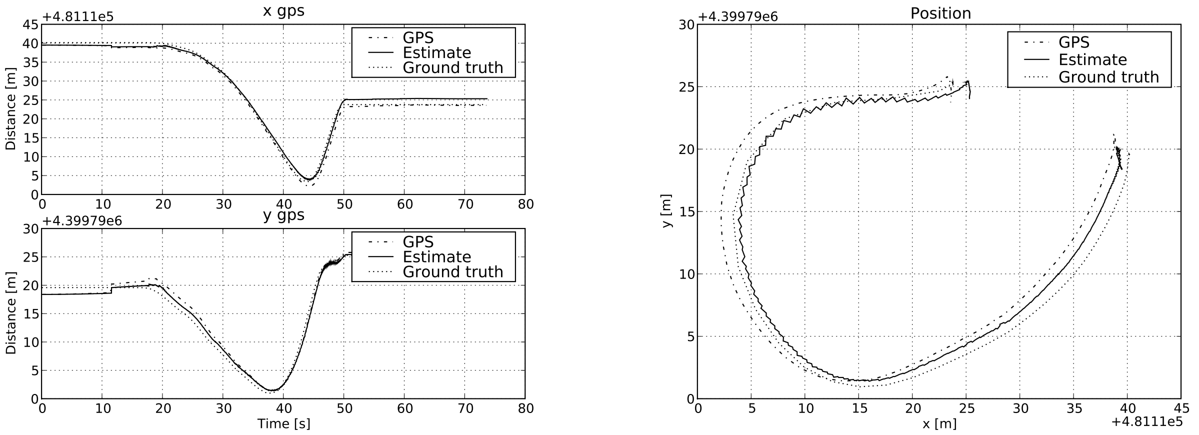

Fig. 8. EKF results.

Acknowledgment

The authors gratefully acknowledge the financial support of Kuchera Defense Systems under CSM grant #4-42873, as well as useful discussions with engineers at both Kuchera Defense Systems and Visionary Products Inc.

References

[1] J. Leonard and H. Durrant-Whyte, “Mobile robot localization by tracking geometric beacons,” *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, 1991.

[2] S. Ge and F. Lewis, *Autonomous mobile robots: sensing, control, decision-making, and applications*. CRC press, 2006.

[3] P. Bonnifait, P. Bouron, P. Crubille, and D. Meizel, “Data fusion of four ABS sensors and GPS for an enhanced localization of car-like vehicles,” in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 2, 2001.

[4] A. Georgiev and P. Allen, “Localization methods for a mobile robot in urban environments,” *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 851–864, 2004.

[5] D. Bevly and B. Parkinson, “Cascaded Kalman filters for accurate estimation of multiple biases, dead-reckoning navigation, and full state feedback control of ground vehicles,” *IEEE Transactions on Control Systems Technology*, vol. 15, no. 2, pp. 199–208, 2007.

[6] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, J. Dolan, D. Duggins, D. Ferguson, T. Galatali, C. Geyer *et al.*, “Tartan racing: A multi-modal approach to the Darpa Urban Challenge,” DARPA Technical Report, April 2007.(www. darpa. mil/grandchallenge/resources. asp), Tech. Rep., 2007.

[7] Applanix, “Applanix - pos lv,” 2009, [Online; accessed 7-August-2009]. [Online]. Available: http://www.applanix.com/products/land/pos-lv.html

[8] S. Lavalle, *Planning Algorithms*. Cambridge: Cambridge University Press, 2006.

[9] Wikipedia, “NMEA 0183 — Wikipedia, the free encyclopedia,” 2009, [Online; accessed 8-August-2009]. [Online]. Available: http://en.wikipedia.org/wiki/NMEA_0183

[10] R. Brown, *Introduction to Random Signals and Applied Kalman Filtering*. New York: Wiley, 1997.

[11] S. Thrun, *Probabilistic Robotics*. Cambridge: MIT Press, 2005.

[12] Wikipedia, “Magnetic declination — Wikipedia, the free encyclopedia,” 2009, [Online; accessed 5-August-2009]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Magnetic_declination&oldid=306199291

[13] ——, “Universal Transverse Mercator Coordinate System — Wikipedia, the free encyclopedia,” 2009, [Online; accessed 27-July-2009]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Universal_Transverse_Mercator_coordinate_system&oldid=304383647

[14] SAE, “SAE Collegiate Design Series: Baja SAE: About Baja SAE,” 2009, [Online; accessed 7-August-2009]. [Online]. Available: http://students.sae.org/competitions/bajasae/about.htm

[15] K. Moore, M. Whitehorn, A. Weinstein, and J. Xia, “Toward a generic UGV autopilot,” in *Proceedings of SPIE*, vol. 7332, 2009, p. 73321B.

[16] N. G. Survey, “CORS - continuously operating reference stations,” 2009, [Online; accessed 14-September-2009]. [Online]. Available: http://www.ngs.noaa.gov/CORS/