2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

GPS-INS State Estimation for Multi-Robot Systems with Computational Resource Constraints

Luke M. Wachter, and Laura E. Ray, *Member, ASME and IEEE*

Abstract—A decoupled Kalman Filter for GPS-INS sensor fusion is developed for a high-speed multi-robot system with computational resource constraints. An eighth-order filter describing system and bias dynamics is decoupled into four second-order filters. Process and measurement noise statistics and first-order bias dynamics are derived from experimental data. The decoupled filter reduces computation time by a factor of seven over the coupled filter, enabling real-time implementation on an inexpensive processor at the required control update rate of 20 Hz. The decoupled filter is evaluated through simulation and experiments and provides sub-meter position error for over a minute, an order of magnitude improvement over GPS alone.

I. Introduction

Due to the advantages they provide over single robots, multi-robot teams have been studied for a number of applications that require precise relative positioning, including mapping, search and rescue, and object manipulation [1-3]. Before a robot team can address such high-level goals, agents must identify collision-free paths between targets in real-time and attain locations with a given accuracy. This requires each agent to access its state information as well as the state of other agents. Minimizing computation in state estimation is critical for high-speed coordination, since a high control update rate is needed and processing bandwidth is limited. To be successful, algorithms explicitly consider of hardware constraints, which can be severe for high-speed maneuvers in close quarters.

In this paper, a real-time state estimation architecture is developed for a multi-robot system with limited processing bandwidth and estimation requirements for motion control in close quarters. The methodology is validated on the Dynabots, a fleet of nonholonomic, differential-steered, four-wheel drive robots that operate at speeds up to 10 m/s [4]. Each Dynabot uses a Garmin-18 5Hz GPS receiver and a Memsense nIMU to track its location. Additionally, wheel speeds and motor currents are measured. Each robot communicates state information to neighboring robots.

While GPS measures position, heading and speed, GPS alone does not provide sub-meter position accuracy. Position measurements are corrupted by time-varying process noise that depends on the quality of the receiver and GPS signal characteristics [5]. Moreover, the 5 Hz GPS data rate is slow compared to the dynamics; a Dynabot moving at 10m/s travels 2m between samples and could crash without estimation and control between samples. Simulation of Dynabots whose motion is coordinated based on shared position shows that GPS position uncertainty and latency can result in collision [6]. Although higher quality receivers provide better accuracy at 20 Hz, such systems are costly.

The IMU provides complimentary information at up to 100 Hz. The most widely used method for fusing IMU and GPS data is the Extended Kalman Filter (EKF). Numerous EKF implementations for vehicle state estimation have been proposed [7-13]. [7] distinguishes between loose and tight coupling of IMU and GPS data. Loose coupling uses GPS position to update the state, while tight coupling uses raw GPS pseudorange measurements, bypassing the receiver's post-processing. A tightly coupled filter extracts information from pseudorange even with single satellite visibility. Under such low visibility, the receiver cannot triangulate position and thus a loosely coupled filter has no new information. A tightly coupled filter, however, requires state variables for each visible satellite. Therefore, under normal visibility, a tightly coupled filter would be computationally cumbersome.

[8] demonstrates that incorporating the vehicle dynamics into state estimation improves performance over standard kinematic filters. However this requires knowledge of dynamic parameters, which depend on the terrain. Unscented Kalman Filters improve accuracy by propagating the state covariance through the nonlinear system dynamics [9]. Unscented Kalman Filters, however, are more computationally intensive than their EKF counterparts.

It was shown in [6] that a system like the Dynabots, where processing bandwidth is saturated, achieves better performance by sacrificing state estimation accuracy for faster control update. With this tradeoff in mind, a real-time, computationally efficient EKF with adequate accuracy for maneuvering in close-quarters is developed for the Dynabot. Specifically, we require one meter accuracy for at least a minute and a control period of at most 50 ms. The remaining sections introduce the structure of the filter and provide the rationale for its design, characterize the sensors, and report simulation and experimental results.

II. Filter Topology

The filter design is based on the state estimator in [12], wherein a set of two-state linear Kalman filters combines inertial and GPS measurements to estimate velocity and bearing. We augment the filter to provide position estimates as well. Although in doing so the linear dynamics become nonlinear and coupled, the filter still lends itself well to decoupling as a means of reducing computation.

The state consists of four primary elements plus four associated biases. The primary state variables are global N-S position X, global W-E position Y, longitudinal velocity in

This research is supported by NIST under Grant No. 60NANB4D1144 and by the Army Research Office under contract No. W911NF- 06-1-0153.

Luke M. Wachter was with the Thayer School of Engineering, Dartmouth College.

Laura E. Ray. is with the Thayer School of Engineering, Dartmouth College 603-646-1243 (email: lray@dartmouth,edu)

the body-fixed reference frame v_x and bearing ϕ measured from the *X*-axis of the inertial system. The bias elements are colored noise in each measurement: the N-S and E-W GPS positions, b_x and b_y, the accelerometer b_a, and the yaw rate gyroscope b_ϕ. The IMU bias state elements track drifting offsets, e.g., due to temperature compensation. The GPS bias states are white-noise driven Gauss-Markov processes due to underlying physical processes associated with transmission of GPS signals and are almost universally included in filters described in the literature [7-13].

We distinguish between bearing ϕ and direction of motion θ measured by GPS. In general, $\phi \neq \theta$ due to side-slip β, which can be large during dynamic maneuvers on low adhesion surfaces. Due to β, bearing ϕ is unobservable using an IMU and a single GPS receiver [12,13]. Thus, we assume that Dynabots drive on high adhesion surfaces. A steering controller is explicitly designed to avoid control effort that leads to lateral sliding [14]. Thus, $v_y \approx 0$ and $\phi \approx \theta$. Although the Dynabot's IMU has magnetic flux sensors to measure bearing, time-varying electro-magnetic fields from the motors render the sensor unusable; there is little consistency in measured bearing over trials with comparable motor torque sequence [14]. We also assume that all motion is in a horizontal plane. Future implementations could relax this assumption to allow robots to track their state while moving on rugged terrain.

It is instructive to consider the filter dynamics in terms of the natural pairing between primary and bias states, as in Table 1. The "IMU" subscript denotes an IMU measurement, and "GPS" denotes a GPS measurement. ν_v and ν_ϕ are zero-mean, white noise acting on GPS velocity and bearing measurements, respectively. ω variables are white zero-mean, Gaussian random variables corresponding to process noise driving bias states. w_a and $w_{\dot\phi}$ are white process noise acting on velocity and yaw rate dynamic equations, respectively. IMU data are inputs to the filter dynamics, while GPS data are treated as measurements. The only coupling is that $\dot X$ and $\dot Y$ depend on v_x and ϕ. Biases are modeled as first-order Gauss-Markov processes and are discussed in section III. The logical grouping between primary state variable and associated bias will reappear in section IV when decoupling is discussed.

By its standard definition, process noise describes unmodeled inputs, e.g., bumps and vibrations affecting the robot's acceleration in addition to known control inputs. Since the inputs to the filter are IMU measurements a_{IMU} and $\dot\phi_{IMU}$, which, in theory, record all such bumps and vibrations, it is as if there are no random inputs. That is, these sensors explicitly record instantaneous disturbances and additionally include measurement noise. However, an important exception exists for the accelerometer: bumps and vibrations that shift the IMU's body-fixed coordinate system out of the plane via pitching or rolling cause the longitudinal accelerometer to record a component of vertical acceleration g = 9.8 m/s^2. Since the filter dynamics assume that the vehicle is moving in a plane, components of g in a_{IMU} can cause the state estimate to deviate from the actual state. The magnitude of this effect depends strongly on the terrain and velocity. In principle, it is possible to track the component of g that enters a_{IMU}, but doing so would require measurement of instantaneous pitch and roll angles, introducing additional sensors and computation. Instead, this disturbance is modeled as a rapidly drifting bias with a time constant dependent on the frequency of the vehicle's vibrations. If the frequency is on the order of the sampling frequency, the bias will appear as white noise. The bias may also contain slowly drifting components associated with gradual changes in topography. In section III, noise on the accelerometer is characterized for both a stationary and constant-velocity vehicle to bound the process noise statistics.

The full-state dynamics in state space form are

$$\dot{\mathbf{x}} = A_c\mathbf{x} + B_c\mathbf{u}_{IMU} + L_c\mathbf{w} \quad (1)$$

$$\mathbf{y}_{GPS} = C\mathbf{x} + \mathbf{v} \quad (2)$$

where $\mathbf{x} = [X, b_x, Y, b_y, v_x, b_a, \phi, b_\phi]^T$, $\mathbf{w} = \begin{bmatrix} \omega_{bx} & \omega_{by} & w_a & \omega_{ba} & w_\phi & \omega_{b\phi} \end{bmatrix}^T$, $\mathbf{u}_{IMU} = \begin{bmatrix} a_{IMU} & \dot\phi_{IMU} \end{bmatrix}^T$, $\mathbf{y}_{GPS} = \begin{bmatrix} X_{GPS} & Y_{GPS} & v_{GPS} & \phi_{GPS} \end{bmatrix}^T$, $\mathbf{v} = \begin{bmatrix} 0 & 0 & \nu_v & \nu_\phi \end{bmatrix}^T$, and $\{A_c, B_c, L_c, C\}$ are derived from Table 1.

A process noise vector $\mathbf{w}$ is described by its spectral density matrix Q_c, where $E[\mathbf{w}(t)\mathbf{w}(\tau)^T] = Q_c\delta(t-\tau)$. It is assumed that process noise is independent such that Q_c is a diagonal matrix. The elements of Q_c, denoted Φ, are approximated from sampled statistics in section III. Typically, Φ is approximated by assuming noise is band-limited above frequency f, then [15,16]

$$\Phi = E[w^2]/f \quad (3)$$

where $E[w^2]$ is the variance of the white random variable in question. [17] chooses f as $1/\Delta t$, where Δt is the filter update interval. We adopt this convention here such that

$$Q_c = \Delta t E[\mathbf{w}\mathbf{w}^T] \quad (4)$$

The filter dynamics in eq. 1 are discretized with sampling interval Δt using a zero-order hold approximation:

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}_{IMU}(k) + L\mathbf{w}(k) \quad (5)$$

$$\mathbf{y}_{GPS}(k) = C\mathbf{x}(k) + \mathbf{v}(k) \quad (6)$$

The update rate is chosen to be fast enough such that error in assuming constant acceleration and yaw rate during the sample period is minimized [12,15]. L is computed in the same manner as B.

The variance of a function of random variables with known covariance can be approximated as

$$E[f(X)^2] \approx \nabla f(X)^T E[XX^T] \nabla f(X) \quad (7)$$

Using the approximation $L \approx \Delta t L_c$ and $w_d = L_c \Delta t w$, where L_c is the continuous-time process noise input matrix, the discrete process noise covariance matrix Q_d is

$$Q_d = E[w_d w_d^T] = L_c \Delta t E[ww^T] L_c^T \Delta t = L_c Q_c L_c^T \Delta t \quad (8)$$

Eq. 8 is the standard approximation of the discrete-time covariance matrix [17].

With the discretized system, the standard recursive Kalman filter can be used. At each sampling interval, the state estimate $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$ are propagated in time as

Table 1 Dynamics of the estimator state.

X -position dynamics	Y -position dynamics
$\dot X = v_x \cos(\phi)$ $\dot b_x = (-b_x + \omega_{bx})/\tau_r$ $X_{GPS} = X + b_x$	$\dot Y = v_x \sin(\phi)$ $\dot b_y = (-b_y + \omega_{by})/\tau_r$ $Y_{GPS} = Y + b_y$
Velocity dynamics	Yaw dynamics
$\dot v_x = a_{IMU} - b_a + w_a$ $\dot b_a = (-b_a + \omega_{ba})/\tau_a$ $v_{GPS} = v_x + \nu_v$	$\dot\phi = \dot\phi_{IMU} - b_\phi + w_\phi$ $\dot b_\phi = (-b_\phi + \omega_{b\phi})/\tau_g$ $\phi_{GPS} = \phi + \nu_\phi$

$$\hat{\mathbf{x}}(k+1|l) = A\hat{\mathbf{x}}(k|l) + B\mathbf{u}_{IMU}(k)$$
$$\mathbf{P}(k+1|l) = A\mathbf{P}(k|l)A + Q_d \quad (9)$$

where the notation $(i|j)$ denotes the quantity at the i th time step given j GPS measurements. When a new GPS measurement is available, the Kalman filter update of $\hat{\mathbf{x}}$ and $\mathbf{P}$ is performed immediately following the time update:

$$\hat{\mathbf{x}}(k|l+1) = \hat{\mathbf{x}}(k|l) + K\left[\mathbf{y}_{GPS}(l) - C\hat{\mathbf{x}}(k-n|l)\right] \quad (10)$$
$$\mathbf{P}(k|l+1) = \left[I - KC\right]\mathbf{P}(k|l) \quad (11)$$

where the Kalman gain K is given by

$$K = \mathbf{P}(k|l)C^T\left[C\mathbf{P}(k|l)C^T + R\right] \quad (12)$$

R is the covariance of the four measurements derived from GPS. The offset n in eq. 10 accounts for delay between the GPS signal time and the time the measurement is incorporated into the state estimate [10,12]. The GPS receiver provides a precision pulse signal (PPS), the rising edge of which indicates the time of a GPS fix. Figure 1 shows an oscilloscope trace of this signal compared to the output at the receiver's serial data transmission port. Serial communication is indicated by the large signal fluctuations, which begin ~50ms after the rising edge of the pulse and complete ~100ms after the pulse. The timing of serial transmission varies significantly. When GPS data are finally incorporated into the state estimate, the filter uses the measure of delay to determine the value of n to use in eq. 10. Since GPS and IMU data are synchronized, GPS data are always aged by an integer number of sample times $\Delta t = 50$ ms. [14] discusses synchronization between the GPS and the sampling loop in more detail.

III. Sensor Characterization

In order to design an effective Kalman filter, the noise statistics of all sensors must be known. We characterize the noise and bias dynamics of each measurement in this section.

The IMU's longitudinal accelerometer and yaw rate gyro are corrupted by non-stationary biases in addition to zero-mean white noise. The bias is due primarily to fluctuations in temperature. When integrated, even small bias causes divergence; therefore, bias must be estimated. As discussed in section II, the acceleration measurement also incurs error as the robot pitches; therefore, the accelerometer is characterized for both a stationary robot and for a robot driving at constant velocity, as noise levels when driving are several orders of magnitude greater than for a stationary robot [14]. The true level of the disturbance on acceleration likely lies somewhere between the two.

The bias on inertial sensors is typically modeled as a Gauss-Markov process [12,13,16]. In continuous time,

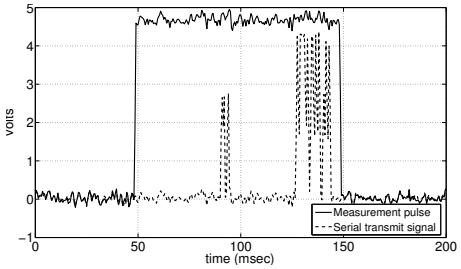

Figure 1. Oscilloscope trace showing the precision pulse signal and the serial transmit port of the Garmin-18 GPS receiver.

$$\dot{b}(t) = -\frac{1}{\tau}b(t) + \frac{1}{\tau}\omega(t) \quad (13)$$

where τ is the time constant, and in discrete time,

$$b(k+1) = Ab(k) + B\omega(k). \quad (14)$$

where $A \approx 1 - \frac{\Delta t}{\tau}$ and $B \approx \frac{\Delta t}{\tau}$ for small sampling interval Δt. The sequence of sensor outputs m are expressed as

$$m(k) = b(k) + w(k) \quad (15)$$

where w is the white noise on the output. The challenge is to identify τ and the variance of the white noise sequences w and ω for both the accelerometer and the rate gyro.

Several methods for extracting noise statistics from inertial data are discussed in the literature. Allan Variance identifies frequencies corresponding to white noise and bias processes [18]. Another frequency domain technique is used in [19] in which a second order lowpass filter is best-fit to the Bode plot of the power spectral density of a noisy measurement. Observer-Kalman filter Identification (OKID) with residual whitening [20] can identify the dynamics and noise statistics simultaneously in a least-squares manner. However, this approach requires identification of a higher order Gauss-Markov process, which would increase the complexity of the Kalman filter. Instead, we use a simpler approach that assumes a first-order Gauss-Markov process. First, the bias sequence is extracted from a measurement history. The Gauss-Markov time constant is identified from the extracted bias sequence. Finally, the variance of the white noise driving the Gauss-Markov model is estimated.

The bias is extracted from measurement data by taking advantage of the fact that the sampling interval Δt is much smaller than the bias time constant τ to be estimated. Since $\Delta t \ll \tau$, the white noise w acting on the output is much higher frequency than the bias dynamics. Therefore, the bias sequence is found by filtering measurements m through a low-pass filter with time constant τ_{LP}. Let the output of this filter be y and the true bias sequence be b. Then $y \to b$ as the residual $r = m - y$ becomes white. In practice, the residual will never be perfectly white, so we find τ_{LP} that maximizes the whiteness of r using the auto-covariance of the discrete sequence r to measure whiteness

$$\psi_k(r) = \frac{1}{N}\sum_{n=1}^{N-k} r(n)r(n+k) \quad (16)$$

where N is the length of the sequence. For white noise, $\psi_k = 0 \forall k$. To measure whiteness of r, the root-mean-square of $\psi_k(r)$, $k = 1...5$ is computed and normalized with respect to the variance of r. To extract the bias from the measurement sequence m, we iterate over τ_{LP} until a value is found that minimizes the normalized RMS auto-covariance of r. Then b is taken to be the output of the low pass filter y.

Figure 2 shows representative results of the first step of the identification process for a yaw rate measurement sequence; $\tau_{LP} = 186.7$ sec is found to give the whitest residual. As τ_{LP} increases, the normalized RMS auto-covariance of r approaches that of the raw measurement sequence. This procedure is performed on accelerometer measurement sequences for the stationary robot, with $\tau_{LP} = 3.5$ sec and for the robot moving at constant velocity, with $\tau_{LP} = 36.2$ sec. For all measurement sets, τ_{LP} that gives

the whitest residual is at least two orders of magnitude greater than Δt .

The variance of the residual provides an estimate of the variance of w, the white noise acting on the output of the bias dynamics in eq. 15. Figure 3 show the variance of r as a function of τ_{LP} for the yaw rate measurement. Above τ_{LP} =1 sec, the variance is relatively constant. For small values of τ_{LP} , the variance plunges; as $\tau_{LP} \to \Delta t$, the low-pass filter is fast enough to track not only the bias but also the white noise on the output. The variance of the output noise w_ϕ is found to be $E(w_\phi^2) = 1.47 \times 10^{-4} (rad/s)^2$. Following this procedure for the accelerometer, the variance of the output noise w_a is found for the stationary robot and for the robot in motion, as reported in Table 2.

Next, we identify the Gauss-Markov time constant τ from the estimated bias. We do so by recognizing that τ is a function of the auto-covariance of b . Multiplying eq. 14 by $b(k)$ and taking the expectation gives

$$E[b(k+1)b(k)] = E[Ab(k)^2] + E[B\omega(k)b(k)] \tag{17}$$

Since b and ω are uncorrelated, $E[B\omega(k)b(k)] = 0$. $E[b(k+1)b(k)] = \psi_1(b(k))$ is the single-shift auto-covariance of b ; in stochastic steady state, $E[b(k)^2]$ and $\psi_1(b(k))$ are simply the variance and auto-covariance of b , respectively, or $E[b^2]$ and $\psi_1(b)$. Then eq. 17 becomes $\psi_1(b) = AE[b^2]$. Using the approximation $A \approx 1 - \frac{\Delta t}{\tau}$, $\tau \approx \Delta t / \left(1 - \psi_1(b)/E[b^2]\right)$. This process can be generalized to larger shifts by multiplying eq. 14 by $b(k-l+1)$ and taking the expectation, resulting in [15]

$$\psi_l(b) = A^l E[b(k-l+1)^2]. \tag{18}$$

In stochastic steady state, $E[b(k-l+1)^2] = E[b^2]$. Substituting in the expression for A , τ becomes

$$\tau \approx \Delta t / \left(1 - \left(\frac{\psi_l(b)}{E[b^2]}\right)^{1/l}\right). \tag{19}$$

Eq. 19 is used to identify τ from the bias estimate for each sensor for shift values ranging from $l = 1$ to $l = 50$. τ is plotted as a function of l in Fig. 4 for the yaw rate measurement. τ is taken as the mean over the fifty time constant estimates and is reported in Table 2 for each sensor.

The final step in the noise model identification process is to determine the variance of the process noise ω driving each bias. Consider the variance of the bias at an arbitrary time step $k+1$:

$$E[b(k+1)^2] = A^2 E[b(k)^2] + B^2 E[\omega(k)^2] \tag{20}$$

In the stochastic steady state $E[b(k+1)] = E[b(k)] = E[b]$ and $E[\omega(k)] = E[\omega]$ so eq. 20 becomes $E[b^2](1 - A^2) = B^2 E[\omega^2]$. Inserting the approximations for A and B gives

$$E[b^2]\left(1 - 1 + 2\frac{\Delta t}{\tau} - \frac{\Delta t^2}{\tau^2}\right) = \frac{\Delta t^2}{\tau^2} E[\omega^2] \tag{21}$$

Since $\Delta t \ll \tau$, eq. 21 is approximated as $2\frac{\Delta t}{\tau} E[b^2] = \frac{\Delta t^2}{\tau^2} E[\omega^2]$. Solving for $E[\omega^2]$ yields

Table 2. Summary of noise statistics for the longitudinal accelerometer and yaw rate sensor. All data used for characterization was collected at 20Hz.

Sensor	τ	$E[\omega^2]$	$E[w^2]$
yaw rate gyro	292s	$5\times10^{-3}(rad/s)^2$	$1.47\times10^{-4}(rad/s)^2$
accelerometer (stationary)	49s	$7\times10^{-3}(m/s^2)^2$	$2.65\times10^{-5}(m/s^2)^2$
accelerometer (moving)	4.05s	$0.01(m/s^2)^2$	$0.11(m/s^2)^2$

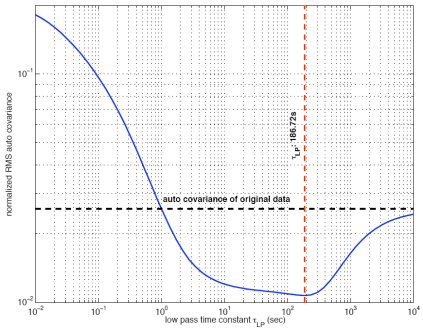

Figure 2. Normalized RMS auto-covariance of the residual as a function of low-pass filter time constant for yaw rate measurement.

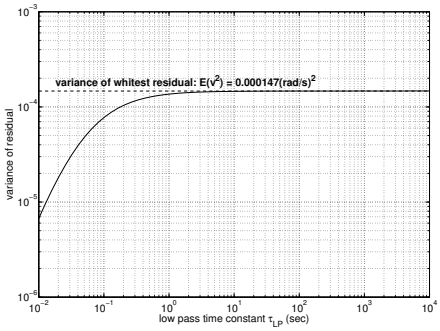

Figure 3. Variance of the residual noise as a function of low-pass filter time constant τ_{LP} for the yaw rate measurement.

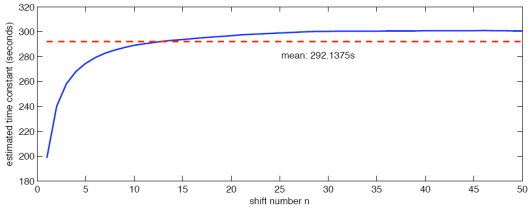

Figure 4. Representative results of estimating the Gauss-Markov time constant for bias shift values from $l = 1$ to 50 for yaw rate sensor.

$$E[\omega^2] = 2\frac{\tau}{\Delta t} E[b^2] \tag{22}$$

Eq. 22 is used to find the variance of the process noise from the value of τ derived above. The resulting variance of ω is listed in Table 2 along with the other noise statistics for both the accelerometer and the yaw rate sensor.

The state estimation algorithm uses four measurements from the GPS receiver: X_{GPS} and Y_{GPS} , v_{GPS} and θ_{GPS} . The velocity and heading measurements are derived from the phase shift of the GPS carrier wave and are not obtained by differencing position measurements. Therefore, GPS position and velocity can be treated as independent measurements. In this section, we characterize each GPS measurement.

GPS velocity can be treated as an unbiased signal corrupted by zero-mean white noise [12]. A 600-second V_{GPS} measurement history from a stationary Dynabot was used to estimate variance $E[v_v^2] = 0.0014(m/s)^2$, which is consistent with the accuracy quoted by the manufacturer. Heading measurement θ_{GPS} is derived from GPS velocity measurements in the North-South and East-West directions, denoted V_{EW} and V_{NS} , respectively. Specifically,

$$\theta_{GPS} = \tan^{-1}\left(V_{EW}/V_{NS}\right) \tag{23}$$

Therefore, the variance of θ_{GPS} depends on V_{GPS} and its variance. The variance of a function of random variables

with known covariance can be approximated as in eq. 7. Assuming V_{EW} and V_{NS} are uncorrelated, the variance of the GPS bearing measurement is approximated as

$$var[\theta_{GPS}] \approx \begin{bmatrix} \frac{\partial \theta_{GPS}}{\partial V_{EW}} & \frac{\partial \theta_{GPS}}{\partial V_{NS}} \end{bmatrix} \begin{bmatrix} var[V_{EW}] & 0 \\ 0 & var[V_{NS}] \end{bmatrix} \begin{bmatrix} \frac{\partial \theta_{GPS}}{\partial V_{EW}} & \frac{\partial \theta_{GPS}}{\partial V_{NS}} \end{bmatrix}^T$$
$$\approx \begin{bmatrix} \frac{V_{NS}}{V_{GPS}^2} & \frac{-V_{EW}}{V_{GPS}^2} \end{bmatrix} \begin{bmatrix} var[V_{EW}] & 0 \\ 0 & var[V_{NS}] \end{bmatrix} \begin{bmatrix} \frac{V_{NS}}{V_{GPS}^2} & \frac{-V_{EW}}{V_{GPS}^2} \end{bmatrix}^T \quad (24)$$

where $V_{GPS}^2 = V_{NS}^2 + V_{EW}^2$. Assuming $var[V_{EW}] \approx var[V_{NS}] \approx var[V_{GPS}]$, this reduces to

$$var[\theta_{GPS}] \approx var[V_{GPS}] \left(\frac{V_{NS}^2}{V_{GPS}^4} + \frac{V_{EW}^2}{V_{GPS}^4} \right) \approx \frac{var[V_{GPS}]}{V_{GPS}^2} \quad (25)$$

which is similar to the result obtained in [12].

Errors in GPS position measurements arise from pseudorange errors. Pseudorange is corrupted by clock offsets between receiver and satellites, atmospheric attenuation, and multipath error. GPS position error is typically modeled as a colored noise bias using a Gauss-Markov process [5] uncorrupted by white measurement noise. The GPS position bias changes slowly. [5] estimates the time constants associated with error mechanisms that affect GPS position and cites values ranging from 600 sec. for multipath error to 3600 sec. for tropospheric effects.

The time constant and noise statistics of the error model for GPS position are computed by the same technique used to characterize the bias dynamics of IMU measurements. Figure 5 shows the estimated bias time constants. Position data used to characterize the bias were collected from a stationary robot for 600 seconds. All deviation from the initial position is attributed to bias since, when initialized, the filter assumes its initial position relative to other robots or a target is known; absolute position is less important for multi-robot coordination than relative distances between robots. The bias time constant estimate is similar for both x and y and $\tau = 125$ sec. is chosen for each.

The variance of the white noise driving the Gauss-Markov process is related to the variance of the bias by the time constant, as discussed above. The estimated variance for x and y measurements, from the same 600-sec sample are found to be 5150 m^2 and 10074 m^2, respectively. The variance of the white noise driving the position bias states is taken as the average of the variance measured for the X and Y positions or $E[\omega_{bx}{}^2] = E[\omega_{by}{}^2] = 7611$ m^2. The noise statistics for GPS measurements are summarized in Table 3.

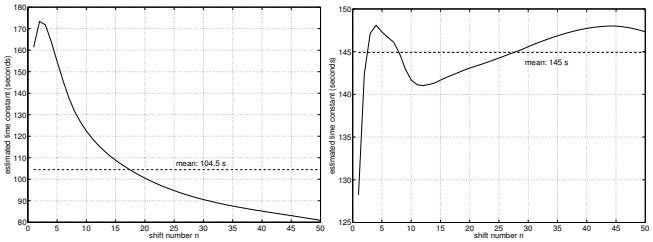

Figure 5. Estimates of the Gauss-Markov process time constants for (left) x and (right) y position measurements from the GPS.

Table 3. Summary of the GPS measurement statistics, sample rate 5Hz.

Measurement	τ	$E[w^2]$	$E[v^2]$
Position	125s	7611 (m/s)^2	0
Velocity	N/A	N/A	1.4 x 10^{-3} (m/s)^2
Bearing	N/A	N/A	1.4×10^{-3}(m/s)^2/$v_x{}^2$

IV. Filter Performance

State estimation using the proposed EKF is computationally intensive. The timing for the coupled filter running on the Dynabot's microcontroller, a Z-World 29.4 MHz RCM3100 is shown in Table 4. In the standard form, each filter iteration takes more than twice the target sample interval $\Delta t = 50$ ms, and thus the target update rate cannot be achieved using the coupled filter. A decoupled filter is developed to reduce processing time by breaking the eighth-order filter into four second-order systems tracked by individual Kalman filters. Grouping of state variables is guided by the magnitude of the off-diagonal elements in the steady-state covariance matrix of the coupled filter: if an element is large relative to other values in its row or column, the pair of state variables associated with it are grouped together [21,22]. In the decoupled filter, the state propagation step is identical to that of the coupled filter.

Degree of coupling between state variables is measured by examining the correlation between each pair. The correlation $\rho_{i,j}$ between the i th and j th state variables is found directly from the state covariance matrix of the coupled filter since

$$\rho_{i,j} = \frac{E[(x_i - \bar{x}_i)(x_j - \bar{x}_j)]}{\sqrt{E[(x_i - \bar{x}_i)^2]E[(x_j - \bar{x}_j)^2]}} = \frac{P_{i,j}}{\sqrt{P_{i,i}P_{j,j}}} \quad (26)$$

Figure 6 shows the steady state correlation between states of the coupled filter using measured data from a Dynabot to compute the correlation. X and b_x depend significantly on one another since the largest correlation for X is associated with b_x and the largest correlation for b_x is associated with X. Thus, X and b_x should be coupled in the same sub-filter. The same relationship holds for Y and b_y, v_x and b_a, and ϕ and b_ϕ. Groupings are identical to state pairs identified in Table 1. The only coupling between state pairs is due to the position's dependence on velocity and bearing. By decoupling, these dynamics are excluded from the dynamics Jacobian, so uncertainty in $\hat{v}_x$ and $\hat{\phi}$ no longer affects the variance of $\hat{x}$ and $\hat{y}$ directly. Instead, we treat these uncertainties as continuous-time process noise w_x and w_y with $\dot{\hat{x}} = v_x \cos(\hat{\phi}) + w_x$ and $\dot{\hat{y}} = v_x \sin(\hat{\phi}) + w_y$. By eq. 7,

$$E[w_x^2] = \cos(\hat{\phi})^2 var(\hat{v}_x) + \hat{v}_x{}^2 \sin(\hat{\phi})^2 var(\hat{\phi}) \quad (27)$$
$$E[w_y^2] = \sin(\hat{\phi})^2 var(\hat{v}_x) + \hat{v}_x{}^2 \cos(\hat{\phi})^2 var(\hat{\phi}) \quad (28)$$

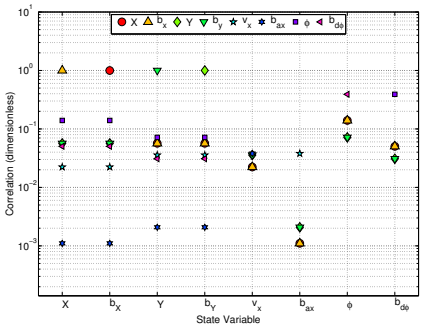

Figure 6. Graphical depiction of the correlation between state variables in the coupled filter, computed from the steady state covariance matrix. Each symbol represents a state variable. Values smaller than 10^{-6} are not shown.

Table 4. Processing time of the coupled and decoupled filters by task.

	Propagate dynamics	Propagate covariance	Compute gain	State update	Covariance update
Coupled	2ms	35ms	32 ms	2 ms	30 ms
Decoupled	2ms	4ms	3 ms	1 ms	4 ms

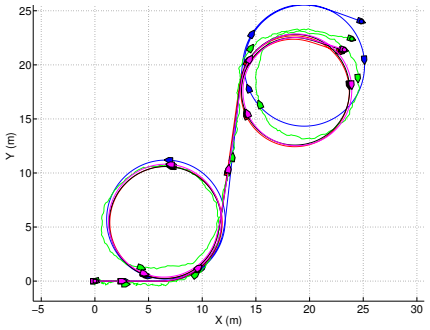

Figure 7. Comparison of true (red) and estimated vehicle trajectory using simulated data: coupled filter (black), decoupled filter (magenta), raw GPS position (green). An idealized, undisturbed trajectory is shown in blue.

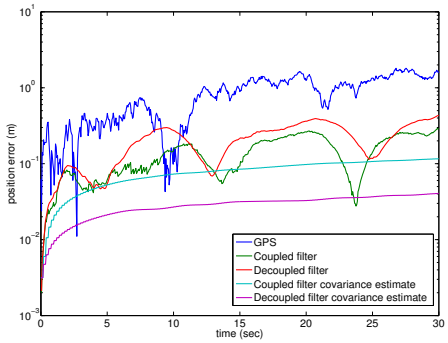

Figure 8. Comparison of error in estimated position for the coupled filter, decoupled filter, and raw GPS, for simulated data. Both the actual position error and the standard deviation based on filter covariance are shown.

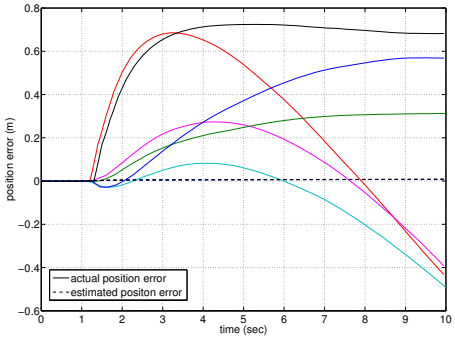

Figure 9. Error in estimated position measured by driving a Dynabot along a repeatable straight line trajectory due North several times. The filter's estimate of position accuracy is also shown.

The time-varying variances of eq. 27 and 28 are obtained from the state covariance matrix P. These continuous-time noise processes are then discretized as in eq. 9. The discrete-time process noise variance for X and Y is found to be $Q_{dx} = \Delta t^2 E[w_x^2]$ and $Q_{dy} = \Delta t^2 E[w_y^2]$.

The computational complexity of a Kalman filter is $O(2n^2m) + O(2nm^2) + O(n^3) + O(m^3)$ where n is the size of the state and m is the number of measurements [23]. For the coupled filter, $n = 8$ and $m = 4$. The decoupled filter consists of four $n = 2$ filters each with $m = 1$; therefore, the coupled and decoupled filters require $O(1344)$ and $O(84)$ floating point operations, respectively, an order of magnitude difference. The timing of the decoupled filter running on the RCM3100 processor given in Table 4 shows that it is indeed nearly an order of magnitude faster than the coupled filter.

To compare accuracy of the two filters, both filters are run using simulation data corrupted by noise processes identified in section III, with accelerometer noise corresponding to noise levels of the robot in motion. Figure 7 shows the simulated trajectory with process noise (red), without process noise (blue), GPS position (green), and the estimated trajectory for the coupled (black) and decoupled (magenta) filters. At $t = 0$, the state covariance is set to 0. Comparing the ideal, disturbance free trajectory with the true, process-noise corrupted trajectory shows the impact of process noise is non-negligible. Both filters reject a drifting bias on GPS position. The loss in accuracy between the filters is small compared to the significant improvement over GPS. An explicit comparison of the position accuracy is shown in Fig. 8. Actual position error for the two filters is shown along with the standard deviation in position derived from state covariance matrices. At 30-sec, position estimates are nearly an order of magnitude more accurate than GPS position. The coupled filter estimates (at $t = 30$ sec) are ~0.1 m more accurate than the decoupled filter for the sample interval $\Delta t = 50$ ms. Keep in mind, however, that as Table 4 shows, it is not possible to run the coupled filter in real-time at $\Delta t = 50$ ms. The coupled filter could be run no faster than $\Delta t = 125$ ms due to overhead from other processes. At this Δt, performance of the coupled filter degrades in comparison with the faster decoupled filter. Even without such degradation, it is beneficial to sacrifice accuracy in favor of higher sampling rate as results from [6] suggest.

The decoupled filter was implemented on the Dynabot processor to verify its functionality. First, a robot was driven at a constant velocity along a straight line due North. The trajectory was repeated several times. The robot's East-West position estimate is taken to be representative of the magnitude of the estimation error. Figure 9 shows the results. The position estimate deviates from the true course by at most ~0.7m. Some error is likely the result of the robot not driving along a perfectly straight trajectory, i.e., the trajectory is not repeatable to infinite precision. However, overall, the errors measured are on the same order of magnitude as those predicted by simulation. Figure 9 also shows the filter's estimated accuracy based on the filter covariance matrix from experimental data. Clearly, for the decoupled filter, the estimated accuracy is not representative of the filter's true accuracy, as discussed above.

Figure 8 indicates that while the accuracy of position estimates from coupled filter are in line with the accuracy indicated by the state covariance estimate, the same is not true for the decoupled filter. The actual accuracy of the estimate from the decoupled filter is significantly worse than its estimated standard deviation. This is a direct result of decoupling the filter states. Since the state transition matrix no longer describes the true dynamics of the system, the state covariance is not accurately propagated in time.

In a second test, the robot was driven under manual control through four waypoints arranged in a square 6m on a side on bumpy grass, which represents a worst case condition for state estimation. A video camera recorded the run and was synchronized with the robot's motion after the run. Figure 10 shows superimposed snapshots of the robot's trajectory at various points in time during the run, indicated by the white text. The red dots indicate waypoint s. The robot was driven along a non-trivial path with several multi-point turns, starting and stopping at the same position. Figure 10 also shows a bird's eye view of the robot's position estimate and heading throughout the run.

Depictions of the robot are shown at the same points in time as the snapshots. Also shown is the GPS position measurement. The estimated trajectory matches well with video snapshots. Moreover, the filter rejects large deviations in GPS position. In particular, at 100 sec, the GPS position swings several meters away from the true location of the robot. Yet the robot's estimated position remains reasonably close to its location in the video snapshot at 100 sec. This experiment confirms that the decoupled filter is schedulable at 20Hz on the Dynabot's processor and it improves upon the accuracy of raw GPS significantly.

In experiments with multi-robot formation control, the decoupled filter shows a marked improvement over GPS alone. Figure 11 shows the initial and final frames of a video for formation control of a group of four robots using a potential function control algorithm described in [14]. At t=0, the robots face a target and must arrange themselves in a circle around the target. The controller requires each robot to know the position of each other robot. The control update frequency is 20 Hz, and position from each robot's state estimator is shared through an ad-hoc 802.11b wireless network as described in [6]. Robots achieve speeds exceeding 5 m/s. The final frame shows good performance in reaching a formation with approximately 1-m position accuracy. No collisions due to position error were observed in nine runs involving four robots. In contrast, earlier runs at comparable speeds, with raw GPS position communicated to locate each robot show one collision and numerous near collisions with only three robots.

V. Conclusion

An efficient state estimation algorithm has been developed for use on a multi-robot platform with limited computational resources. The algorithm fuses GPS and IMU measurements using a decoupled Kalman filter. In addition to estimating the robot's state, it tracks biases on the sensor measurements. Statistics and dynamic characteristics of the noise and bias processes acting on sensors are extracted from experimental data. By grouping pairs of related state variables into two-state sub-filters, the computational complexity of the filter is reduced by an order of magnitude over the coupled filter without a significant loss of accuracy. The decoupled filter runs at a significantly faster rate on the Dynabot's processor than would the coupled filter. The increase in update rate afforded by decoupling is expected to more than compensate for any loss in accuracy.

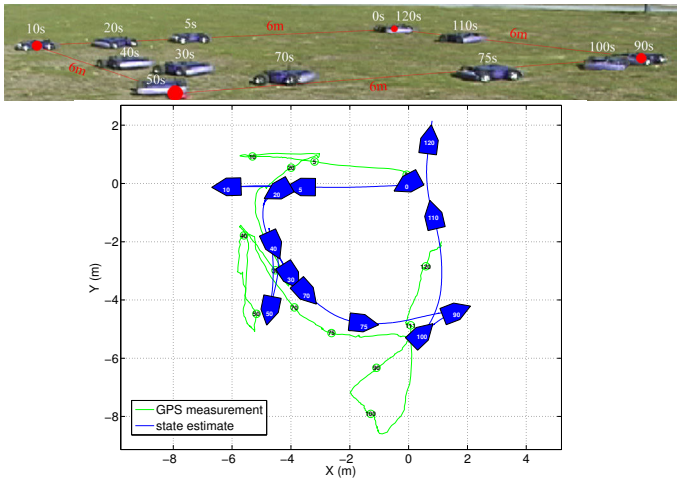

Figure 10. Comparison of estimated state computed in real-time on board a Dynabot (blue) to video recording and to reported GPS position (green). Numbers in indicate the time in seconds for each frame shown.

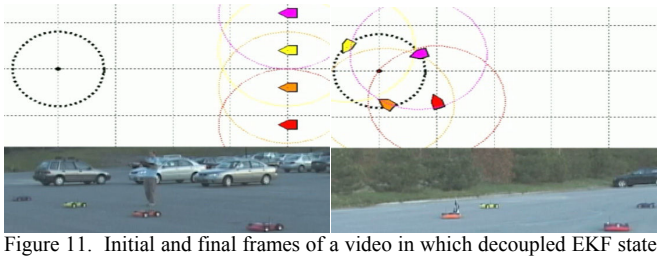

Figure 11. Initial and final frames of a video in which decoupled EKF state estimation is used for potential function formation control of four Dynabots.

References

[1] J.L. Baxter, E.K. Burke, J.M. Garibaldi, and M. Normal. Multi-Robot Search and Rescue: A Potential Field Approach. *Autonomous Robots and Agents*, vol. 76. Springer, Berlin, 2007.

[2] B.S. Pimental and M. F. M. Campos. Cooperative Communication in ad hoc networked mobile robots. *IEEE Int. Conf. on Intelligent Robots and Systems,* 2876-2881, 2003.

[3] J. Fink, N. Michael, V. Kumar. Composition of vector fields for multi-robot manipulation via caging. *Proc. of Robotics: Science and Systems*, Atlanta GA, June 2007.

[4] L. Ray, D. Brande, J. Murphy, J. Joslin Cooperative Control of Autonomous mobile Robots in Unknown Terrain, *ASME Int. Mech. Eng. Conf. and Exhibition*, Chicago IL, Nov. 2006.

[5] James Rankin. GPS and differential GPS: An error model for sensor simulation. *IEEE Position Location and Nav. Symp.*, 260-266, 1994.

[6] J.P. Murphy, L.M. Wachter, and L.R. Ray, Computational Resource Allocation in Cooperative Control of Mobile Robots, *First Int. Conf. on Robot Communication and Coordination*, Oct 2007.

[7] M. George and S. Sukkarieh. Tightly coupled ins/gps with bias estimation for uav applications. *Proc. Australiasian Conf. on Robotics and Automation (ACRA)*, 2005.

[8] I. Ashokaraj, P. Silson, and A. Tsourdos. Application of an extended Kalman flter to multiple low cost navigation sensors in wheeled mobile robots. *Sensors*, 2:1660-1664, 2002.

[9] F. Azizi and N. Houshangi. Sensor integration for mobile robot position determination. *IEEE Int. Conf. on Systems, Man and Cybernetics*, 2:1136-1140, 2003.

[10] T. Aono, K. Fujii, S. Hatsumoto, and T. Kamiya. Positioning of vehicle on undulating ground using GPS and dead reckoning. *IEEE Int. Conf. on Rob. and Automation*, Belgium, 3443-3448, 1998.

[11] J. Wang, H. K. Lee, and C. Rizos. GPS/INS integration: A performance sensitivity analysis. *Wuhan University Journal of Nature Sciences*, 8(2B):508-516, 2003.

[12] D. M. Bevly. Global positioning system (gps): A low-cost velocity sensor for correcting inertial sensor errors on ground vehicles. *J. Dynamic Systems, Measurement, and Control*, 126:255–264, 2004.

[13] Y. Yang and J. A. Farrell. Magnetometer and differential carrier phase GPS-aided INS for advanced vehicle control. *IEEE Trans. on Robotics and Automation*, 19(2):269-282, 2003.

[14] L.M. Wachter. Design of a Motion Coordination Architecture for a Multi-Robot System with Hardware Constraints, M.S. Thesis, Dartmouth College, 2008.

[15] R. F. Stengel. *Stochastic Optimal Control*. John Wiley & Sons, 1986.

[16] P. Zarchan and H. Musoff. *Fundamentals of Kalman Filtering*, vol. 190 of *Progress in Astronautics and Aeronautics*. AIAA, Inc., 2000.

[17] J. Farrell and M. Livstone. Exact calculation of discrete-time process noise statistics for hybrid continuous/discrete time applications. In *Proc. Conf. on Decision and Control*, pages 857–858, 1993.

[18] H. Kim, J. Lee, and C. Park. Performance improvement of GPS/INS integrated system using Allan Variance analysis. *Proc. of the International Symposium on GNSS/GPS*, 2004.

[19] K. Mohammadi and Mohammad Reza Zamani. Using GPS velocity information in enhancement of GPS position accuracy. *Proc. Asian GPS Conference*, 2002.

[20] M. Phan, L. G. Horta, J.-N. Juang, and R. W. Longman. Improvement of observer/kalman filter identification (OKID) by residual whitening. *J. Vibrations and Acoustics*, 117:232–239, 1995.

[21] F. E. Daum and R. J. Fitzgerald. Decoupled Kalman Filters for phase array radar tracking. *IEEE Trans. on Aut. Cont.*, 28(3):269-283, 1983.

[22] Steven R. Rogers. Steady-state performance of the decoupled Kalman Filter. *Proc. Aerospace and Electronics Conference*, 334-339, 1988.

[23] M.J. Goris, D.A. Gray, and J.M.Y Mareels. Reducing the computational load of a Kalman Filter. *Electronics Letters*, 33(18):1540-1541, 1997.