

# **KALMAN FILTER STUDY OF A MICRO-ROBOT WITH TRACK SLIPPAGE**

*TECHNICAL REPORT*

Hanspeter Schaub<sup>1</sup> and John L. Junkins<sup>2</sup>

April 10, 2003

<sup>1</sup>Post-Doctoral Research Associate, Aerospace Engineering Department, Texas A&M University, College Station TX 77843.

<sup>2</sup>George Eppright Chair Professor of Aerospace Engineering, Aerospace Engineering Department, Texas A&M University, College Station TX 77843.

# Contents

1	Introduction . . . . .	1
2	Kalman Filter Outline . . . . .	2
	2.1 Linearized Equations of Motion . . . . .	2
	2.2 Covariance Matrices . . . . .	4
	2.3 Sensor Updates . . . . .	5
3	Observation Vectors . . . . .	7
	3.1 GPS Sensor . . . . .	7
	3.2 Compass Heading Sensor . . . . .	7
	3.3 Range Sensor . . . . .	8
	3.4 Heading Sensor . . . . .	8
4	Track Slippage Modeling . . . . .	9
5	Numerical Study of Slippage affect . . . . .	10
	5.1 GPS Sensor Information Only . . . . .	12
	5.2 GPS and Compass Sensor Information . . . . .	15
6	Alternate Motion Sensor . . . . .	17

### **Abstract**

A Kalman filter is presented for a micro-robotic vehicle which is propelled by two tracks. The vehicle may be equipped with either GPS, compass heading sensor, heading sensor to a non-directional beacon, or any combination thereof. Of particular interest is what effect track slippage will have on the internal position and heading estimates.

# 1 Introduction

A Kalman filter is developed for a planar Mobile Robotic Vehicle (MRV). This vehicle is assumed to be propelled by two tracks on either side of the chassis. The tracks speed are measured by odometers and fed back to the state estimator to forward propagate the internal state estimates. Since the vehicle is to operate over unprepared ground (sand, rocks, gravel, etc.), it must be assumed that the tracks will slip from time to time. This track slippage will corrupt the internal state estimates.

The MRV's are further assumed to have either GPS, compass heading or Non-Directional Beacon (NDB) sensors. The question is whether it is beneficial to implement a Kalman filter in the presence of track slippage, or if it's performance is no better than using the raw GPS and compass heading sensor data.

A sample MRV with its two tracks is illustrated in Figure 1. The origin of the coordinate system is assumed to be the MRV target. The MRV heading is given by the angle  $\theta$ . The range to target is denoted by  $r$  and the relative heading is  $\phi$ .

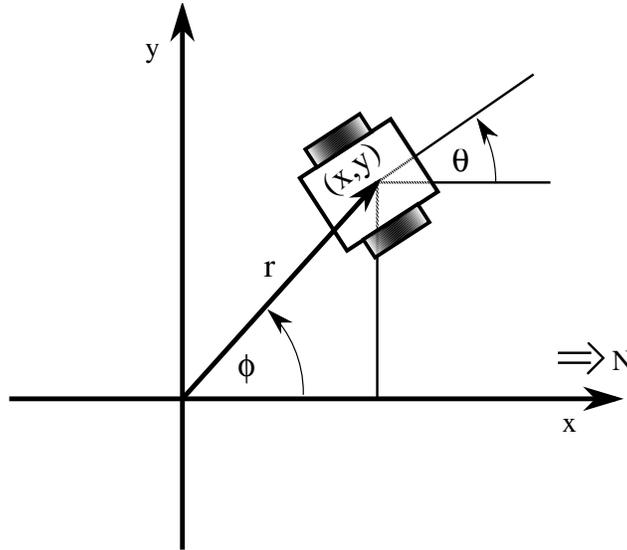


Figure 1: Illustration of the Miniature Robot Crawler

Let  $p = (x, y, \theta)^T$  be the state vector of the MRV, then the equations of

motion are given by<sup>1</sup>

$$\dot{p} = B(p)\omega = \frac{1}{2} \begin{bmatrix} R_r \cos \theta & R_l \cos \theta \\ R_r \sin \theta & R_l \sin \theta \\ 2\frac{R_r}{R_w} & -2\frac{R_l}{R_w} \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} = f(p, \omega) \quad (1)$$

where  $\omega = (\omega_r, \omega_l)^T$ ,  $R_r$  and  $R_l$  are the right and left track radii and  $R_w$  is the distance between the two tracks. The MRV rotates by having each track operate at different speeds. Note that a MRV cannot slip sideways. For it to move purely to the right, it must change its heading first and then propel itself forward.

## 2 Kalman Filter Outline

### 2.1 Linearized Equations of Motion

A Kalman filter for a MRV is outlined in this section. Since the position and heading sensor updates typically occur at a low frequency, it is necessary to estimate the states between these sensor updates to calculate the current control input. Between these sensor updates it is assumed that the track angular velocities are sampled at a high frequency. The measured track angular velocities  $\omega_m$  will contain some white noise  $w$  which is assumed to have a standard deviation of  $\sigma_w$ . The true track angular velocity is given as

$$\omega(t) = \omega_m(t) - w(t) \quad (2)$$

The following extended Kalman estimator provides current state estimates and is able to filter out some measurement noise by optimal weighting of the sensor information based upon forward propagating the state covariance matrix.<sup>2</sup> With the following formulas, a subscript indicates the number of data sets that were used to find its estimate. A superscript indicates the time step of the state estimate. Let  $\hat{p}_k^k$  be the current state estimate at time step  $k$ . Until the next sensor update is available at time  $k + 1$ , the state estimates are then forward integrated using the nonlinear equations of motion in Eq. (1), this process is indicated formally as

$$\hat{p}_k^{k+} = \hat{p}_k^k + \int_k^{k+} \dot{p} dt \quad (3)$$

The notation  $p^{k+}$  indicates the state  $p$  at time  $k+$ , where  $k < k+ < k + 1$ . To be able to forward propagate the state covariance estimate derived

from measurement error covariances, the linearized dynamics about some reference states  $p_{ref}$  and  $\omega_{ref}$  are required.

$$\dot{p}_{ref} = f(p_{ref}, \omega_{ref}) \quad (4)$$

Taking the first Taylor expansion of Eq. (1) about the reference motion we obtain

$$\dot{p} \approx f(p_{ref}, \omega_{ref}) + \left. \frac{\partial f(p, \omega)}{\partial p} \right|_{ref} (p - p_{ref}) + \left. \frac{\partial f(p, \omega)}{\partial \omega} \right|_{ref} (\omega - \omega_{ref}) + \dots \quad (5)$$

After subtracting Eq. (4) from (5) we obtain

$$\delta \dot{p} = \left. \frac{\partial f(p, \omega)}{\partial p} \right|_{ref} \delta p + \left. \frac{\partial f(p, \omega)}{\partial \omega} \right|_{ref} \delta \omega \quad (6)$$

where  $\delta p = p - p_{ref}$  and  $\delta \omega = \omega - \omega_{ref}$ . For linearization of the dynamics, the reference states are set equal to the best present estimates. Therefore  $p_{ref} = p_k^{k+}$  and  $\omega_{ref} = \omega_m$ . Using Eq. (2) the vector  $\delta \omega$  is clearly  $-w$  and is the driving process noise of the linearized system. The linearized system can now be written as

$$\delta \dot{p} = F(p_k^{k+}, \omega_m) \delta p + G(p_k^{k+}) \delta \omega \quad (7)$$

where the matrices  $F$  and  $G$  are defined as

$$F = F(p_k^{k+}, \omega_m) = \left. \frac{\partial f(p, \omega)}{\partial p} \right|_{(p_k^{k+}, \omega_m)} = \frac{1}{2} \begin{bmatrix} 0 & 0 & -(R_r \omega_r + R_l \omega_l) \sin \theta \\ 0 & 0 & (R_r \omega_r + R_l \omega_l) \cos \theta \\ 0 & 0 & 0 \end{bmatrix} \quad (8)$$

$$G = G(p_k^{k+}, \omega_m) = \left. \frac{\partial f(p, \omega)}{\partial \omega} \right|_{(p_k^{k+}, \omega_m)} = \frac{1}{2} \begin{bmatrix} R_r \cos(\theta) & R_l \cos(\theta) \\ R_r \sin(\theta) & R_l \sin(\theta) \\ 2R_r/R_w & -2R_l/R_w \end{bmatrix} \quad (9)$$

From here on we will only use the short hand notation  $F$  and  $G$  where their dependence on  $p$  and  $\omega$  is understood implicitly.

## 2.2 Covariance Matrices

Let the  $3 \times 3$  matrix  $P_k^k$  be the state covariance matrix at time step  $k$ . This covariance matrix provides a measure of how uncertain the current state estimates are. A high entry in  $P$  indicates a high uncertainty of the current state estimate. Let  $Q$  be the covariance matrix associated with the driving process noise  $w$  and the  $2 \times 1$  vector  $\sigma_w$  be the standard deviation of the process noise. The  $2 \times 2$  matrix  $Q$  is then defined as

$$Q = \begin{bmatrix} \sigma_{w_1}^2 & 0 \\ 0 & \sigma_{w_2}^2 \end{bmatrix} \quad (10)$$

Without including the effect of process noise, the covariance matrix  $P$  would eventually tend to zero. This means that only the previous measurements will be trusted and future updates are ignored. If this occurs, then any remaining errors in the state estimates are retained and forward propagated using the  $\omega$  measurements. Including the covariance matrix  $Q$  allows the estimator to be tuned such that  $P$  never will go to zero. Past measurements are never perfectly trusted and it is now assumed that the  $\omega$  measurements also contain measurement noise. In between sensor updates, the covariance matrix  $P$  is updated using

$$P_k^{k+} = P_k^k + \int_k^{k+} \dot{P} dt \quad (11)$$

where the covariance matrix derivative is given as the inhomogeneous Lyapunov equation<sup>3</sup>

$$\dot{P} = FP + PF^T + GQG^T \quad (12)$$

Standard literature on continuous covariance propagation includes an extra “learning” term in the above equation, resulting in the Riccati equation.

$$\dot{P} = FP + PF^T - PH^T \Lambda_{vv}^{-1} HP + GQG^T$$

This term would “decrease” the covariance matrix if the sensor output were sampled continuously. However, the sensor output is only sampled at discrete times and not “continuously” like the track angular velocities. Therefore this term is dropped here since no sensor based learning occurs between the times  $k$  and  $k + 1$ . Once a new sensor measurement is available, the covariance matrix  $P$  is also updated along with the state vector.

Since the  $F$  and  $G$  matrices for the MRVs contain large blocks of zeros, the calculation of  $\dot{P}$  can be simplified. Let the  $P$  matrix be partitioned as

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \quad (13)$$

where  $P_{11}$  is a  $2 \times 2$  matrix,  $P_{12}$  is a  $2 \times 1$  matrix and  $P_{22}$  is a scalar. The matrix  $F$  is partitioned as

$$F = F(p, \omega_m) = \begin{bmatrix} 0 & F_{12} \\ 0 & 0 \end{bmatrix} \quad (14)$$

where the  $2 \times 1$  matrix  $F_{12}$  is defined as

$$F_{12} = \frac{1}{2} \begin{bmatrix} -(R_r \omega_{m_r} + R_l \omega_{m_l}) \sin \theta \\ (R_r \omega_{m_r} + R_l \omega_{m_l}) \cos \theta \end{bmatrix} \quad (15)$$

The matrix  $G$  is written as

$$G = G(p) = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \quad (16)$$

with the  $2 \times 2$  matrix  $G_1$  being

$$G_1 = \frac{1}{2} \begin{bmatrix} R_r \cos(\theta) & R_l \cos(\theta) \\ R_r \sin(\theta) & R_l \sin(\theta) \end{bmatrix} \quad (17)$$

and the  $2 \times 1$  matrix  $G_2$  being

$$G_2 = \begin{bmatrix} \frac{R_r}{R_w} & -\frac{R_l}{R_w} \end{bmatrix} \quad (18)$$

Using these definitions, the time derivatives of the  $P$  matrix partitions are expressed as

$$\dot{P}_{11} = P_{12} F_{12}^T + F_{12} P_{12}^T + G_1 Q G_1^T \quad (19a)$$

$$\dot{P}_{12} = F_{12} P_{22} + G_1 Q G_2^T \quad (19b)$$

$$\dot{P}_{22} = G_2 Q G_2^T \quad (19c)$$

### 2.3 Sensor Updates

Let  $\hat{Y}^{k+1}$  be the estimated output vector of the sensors at time  $k+1$ . It is defined as

$$\hat{Y}^{k+1} = h(\hat{p}_k^{k+1}) \quad (20)$$

The generally nonlinear function  $h(p_k^{k+1})$  maps the current state estimate into a best prediction of the observation vector. The measured sensor output vector  $\tilde{Y}^{k+1}$  at time  $k + 1$  given in terms of the true state vector  $p^{k+1}$  is

$$\tilde{Y}^{k+1} = h(p^{k+1}) + v \quad (21)$$

where the vector  $v$  is the gaussian measurement noise with standard deviation  $\sigma_v$ . The covariance matrix associated with the measurement noise, assuming no correlation of measurement errors, is

$$\Lambda_{vv} = \begin{bmatrix} \sigma_{v_1}^2 & 0 & \cdots \\ 0 & \sigma_{v_2}^2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \quad (22)$$

Assume that at time  $k+1$  a new sensor update  $\tilde{Y}^{k+1}$  is available. The current state estimate  $\hat{p}_k^{k+1}$  is updated to incorporate the new sensor measurement through the extended Kalman filter recursion<sup>2</sup>

$$\hat{p}_{k+1}^{k+1} = \hat{p}_k^{k+1} + K^{k+1} (\tilde{Y}^{k+1} - \hat{Y}^{k+1}) \quad (23)$$

The matrix  $K^{k+1}$  is the optimal Kalman gain matrix which is found through

$$K^{k+1} = P_k^{k+1} H^T (\Lambda_{vv} + H P_k^{k+1} H^T)^{-1} \quad (24)$$

where  $H$  is defined as

$$H = \frac{\partial h}{\partial p} (p_k^{k+1}) \quad (25)$$

The state covariance matrix  $P$  is updated to reflect the presence of a new sensor measurement through

$$P_{k+1}^{k+1} = (I - K^{k+1} H) P_k^{k+1} \quad (26)$$

If  $H$  is a  $n \times m$  matrix, where  $n$  is the number of sensor observations and  $m$  is the number of states (3 for the MRVs), then the matrix inverse required in Eq. (24) is of order  $n$ . Because the sensor updates typically arrive at different frequencies, this number  $n$  can vary from update to update. Implementing the sensor update algorithm directly in this manner would require performing matrix inverses for various matrix sizes, depending on the combination of sensors providing an update the the current time step.

Also, the order  $n$  of the matrix inverse can quickly grow, slowing down the numerical arithmetic.

Due to the linear nature of the Kalman filter, we are able to invoke the superposition principle and incorporate the various sensor updates in a *sequential manner*, rather than in a bulk approach. Flags are added to the program structure that signal that a particular sensor has a new observation vector for the current time step. At a given time step, these various observation vectors are then processed individually one at a time using the algorithms in Eq. (23) through Eq. (26). This sequential approach allows for a more flexible program structure that can incorporate different update frequencies very easily. Also, the order of the matrix inverses involved is kept very small. Since most sensor observation return scalar quantities, the inverse in Eq. (24) typically simply requires a scalar inverse.

### 3 Observation Vectors

This section provides the mapping between the state and the observation vectors for various sensor types. Due to the sequential nature of the Kalman filter used, the sensor updates are processed individually and at different update frequencies if needed. The linearized mapping matrix  $H$  is also provided for each sensor type.

#### 3.1 GPS Sensor

The GPS sensor is assumed to provide planar  $(x, y)$  coordinates of the MRV, where the GPS coordinate frame is assumed to be the same as the one used by the MRV. Therefore the corresponding  $2 \times 1$  observation vector  $\hat{Y}_{GPS}^{k+1}$  is simply given by the linear relationship

$$\hat{Y}_{GPS}^{k+1} = h_{GPS}(p^{k+1}) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{H_{GPS}} p_k^{k+1} \quad (27)$$

The  $2 \times 3$  matrix  $H_{GPS}$  for the GPS sensor is defined in the above expression.

#### 3.2 Compass Heading Sensor

The compass heading sensor returns the orientation angle  $\theta$  of the MRV relative to the north pole axis. For the given coordinate system, this is the

$x$ -axis. The  $1 \times 1$  observation vector  $\hat{Y}_{Compass}^{k+1}$  is given by

$$\hat{Y}_{Compass}^{k+1} = h_{Compass}(p^{k+1}) = \underbrace{[0 \ 0 \ 1]}_{H_{Compass}} p_k^{k+1} \quad (28)$$

The  $1 \times 3$  matrix  $H_{Compass}$  for the GPS sensor is defined in the above expression.

### 3.3 Range Sensor

The Range sensor is assumed to provide a scalar distance measurement of the MRV relative to a stationary object with a *known* location  $(X, Y)$ . This object could be the target itself or a beacon along the path to the target. Given the planar MRV location  $(x, y)$ , the distance  $d$  between the MRV and this object is

$$d(x, y) = \sqrt{(x - X)^2 + (y - Y)^2} \quad (29)$$

The  $1 \times 1$  observation vector  $\hat{Y}_{Range}^{k+1}$  is then given by

$$\hat{Y}_{Range}^{k+1} = h_{Range}(p^{k+1}) = d(x, y) = \sqrt{(x - X)^2 + (y - Y)^2} \quad (30)$$

The linear mapping  $H_{Range}$  is then found to be

$$H_{Range} = \frac{\partial h_{Range}}{\partial p}(p_k^{k+1}) = \left[ \frac{x-X}{d} \quad \frac{y-Y}{d} \quad 0 \right] \quad (31)$$

### 3.4 Heading Sensor

Assume an object is located at a known location with coordinates  $(X, Y)$  and is transmitting a radio frequency (RF) signal (such as a radio station). The heading sensor will sense the relative orientation of the MRV relative to the line of sight vector to this RF sender. In essence, this RF source acts like an “artificial north pole” and provides an alternative heading information to triangulate the MRV location. In aviation, such RF sources are referred to *Non-Directional Beacons* (NDBs). The geometry of a MRV-NDB layout is shown in Figure 2.

The heading of the line of sight vector of the MRV to the NDB relative to the north pole direction ( $x$ -axis in this discussion) is expressed through the angle  $\gamma$ . The relative orientation of the MRV relative to this line of sight vector is labelled as  $\beta$ .

$$\beta(x, y, \theta) = \theta - \tan^{-1} \left( \frac{Y - y}{X - x} \right) \quad (32)$$

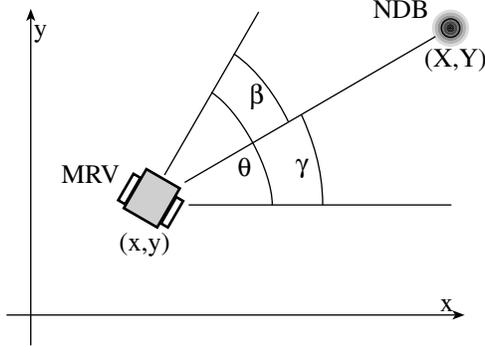


Figure 2: MRV-NDB Layout Geometry

Note that the  $\beta$  angle contains both information about the MRV heading angle  $\theta$  and the MRV location coordinates  $(x, y)$ . The scalar observation vector  $\hat{Y}_{Heading}^{k+1}$  is given by

$$\hat{Y}_{Heading}^{k+1} = h_{Heading}(p^{k+1}) = \beta = \theta - \tan^{-1} \left( \frac{Y - y}{X - x} \right) \quad (33)$$

The linear mapping  $H_{Heading}$  is then found to be

$$H_{Heading} = \left[ -\frac{Y-y}{d^2} \quad \frac{X-x}{d^2} \quad 1 \right] \quad (34)$$

where  $d(x, y)$  is defined in Eq. (29).

## 4 Track Slippage Modeling

The odometers on the two MRV tracks provides an indirect velocity feedback of the vehicle motion. However, because the tracks may slip from time to time, this feedback can be very unreliable at times and will corrupt the internal state forward propagation.

To model track slippage in the numerical simulations of the MRV motion the following scheme was used. A steering law is commanding a desired track speed  $\omega_d(t)$ . The true effective track speed  $\omega_e$  is the track speed corresponding to the actual MRV motion. Thus  $\omega_e$  might be less than  $\omega_d$  at times when the tracks are slipping on the surface. At any instant there is a certain probability that one of the two tracks may start to slip. If it does, than a random amount of slip is computed with a given standard deviation. This track slipping magnitude is held constant over a period of

time. The slip duration is again chosen at random with a given standard deviation. Once the slip duration is over, the program logic again checks periodically to see if this track is slipping again. If yes, then it computes new slip magnitudes and duration. Note that the slip logic is applied to each track individually.

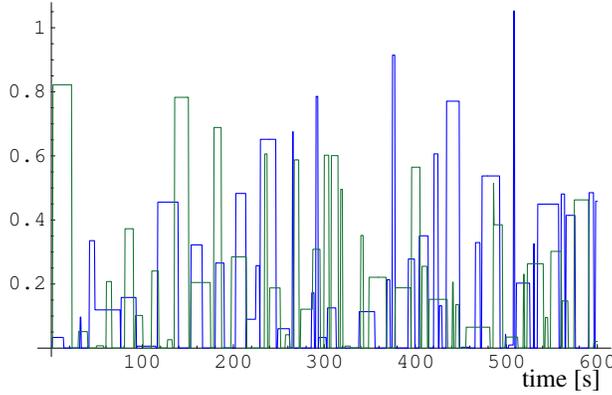


Figure 3: Loss in Effective Track Performance Due to Slippage

The track slippage applied during the following numerical simulations is shown in Figure 3. Note that this is a rather severe case where one or both tracks are slipping at any given time.

## 5 Numerical Study of Slippage affect

The Kalman filter estimator performance is studied for various sensor packages being active and track slippage being either on or off. The MRV geometry is given by  $R_l = R_r = 0.127m$  and  $R_w = 0.508m$ . The track speeds are commanded through the open-loop control

$$\begin{aligned}\omega_l &= 1.05\text{rad/sec} \\ \omega_r &= 1.00\text{rad/sec}\end{aligned}$$

which result in the MRV nominally performing a large circle of roughly 20 meter diameter. Figure 4 shows both the nominal MRV track path without track slippage present and when the track slippage shown in Figure 3 is applied. Clearly the track slippage applied in this study is very severe and represents a worst case type scenario.

The effect of track slippage on the Kalman filter is as follows. The filter does not know that the angular velocity feedback it is receiving of the track

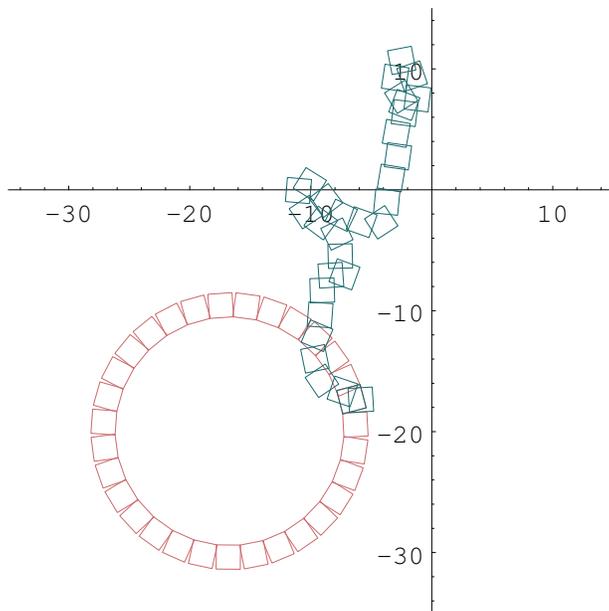


Figure 4: MRV Track Paths With and Without Track Slippage

speeds is erroneous at times. The covariance matrix still becomes very small, indicating that the filter will mainly rely on its internal state estimates and more or less ignore future sensor measurements. Doing so will cause the states estimates to grow very large. Much larger in fact than the typical sensor noise itself. To avoid the filter starting to rely to heavily on the internal state estimates, the procedure used in this study is to increase the process noise covariance matrix  $[Q]$  to a large value. The size of this matrix was determined through numerical simulations. Having a large  $[Q]$  prevents the state covariance matrix from becoming too small and the Kalman filter continuous to rely heavily on the sensor input. The goal of the  $[Q]$  matrix tuning is to achieve better position tracking than what is possible with pure GPS measurement. If the matrix is set too large, then very little filtering is done and we essentially obtain pure GPS data. If the matrix is set too small, then very large drift in the position estimates are possible when track slippage occurs.

All of the following numerical studies have a simulation time of 600 seconds. A simple Euler integration method is used with a time step of 0.1 seconds. GPS sensor information is updated at 5 Hz, compass heading updates arrive at 10 Hz and relative heading sensor information is also at 10

Hz. The GPS measurement noise has a standard deviation of 1 meter, while both the compass and heading measurement noise has a standard deviation of 1 degree. The track speed measurements are assumed to have a standard deviation of 0.1 degrees. These noise levels are used to construct the various sensor covariances.

## 5.1 GPS Sensor Information Only

### Standard Process Noise Matrix

Here the Kalman filter is only updated using GPS position measurements at 5 Hz. The process noise matrix  $[Q]$  is set equal to

$$[Q] = \sigma_\omega^2 [I_{3 \times 3}] = 0.01 I_{3 \times 3} \quad (\text{rad/s})^2$$

For the case where no track slippage is present, the  $(x, y)$  position estimate errors are marked as dark green and blue lines in Figure 5(a). The corresponding GPS sensor noise in the  $x$  and  $y$  direction are shown in the background as light colored lines. With the relatively small process noise  $\sigma_\omega$ , the filtered position errors typically lie within  $\pm 0.3$  meters.

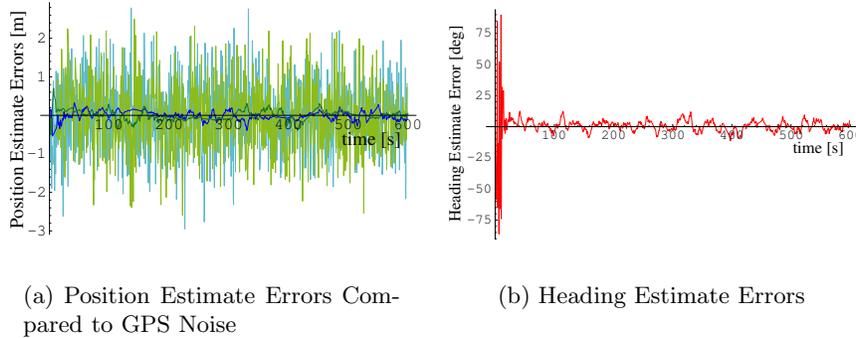
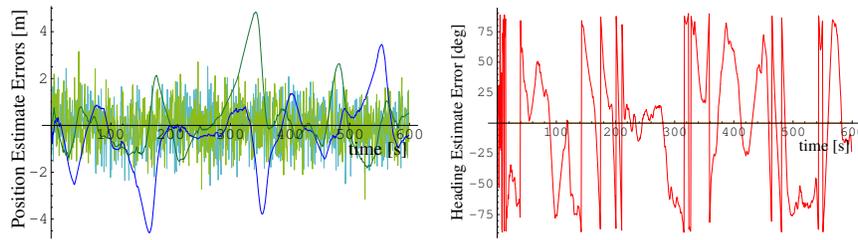


Figure 5: GPS-Only Study without Track Slippage

The compass heading estimate errors are shown in Figure 5(b) as a red line. Without any compass sensor updates, this information is relatively poor. The reason for is as follows. For the MRV to move from the left to right, it must first turn to face the right (i.e. change its heading) and then move forward. The MRV are incapable of “sliding” to the right. If the GPS noise tells the filter that it senses the MRV to be at a slightly different location, then it must assume that the vehicle actually rotated to move in

that direction. Due to the relatively accurate  $(x, y)$  estimates, the filtered compass heading errors remain within  $\pm 10$  degrees.

However, if we add track slippage, these filtered state estimates become erroneous very quickly. The position estimate errors are shown in Figure 6(a). Since the Kalman filter does not know about the tracks slipping, it trusts the  $\omega$  measurements as much as it did in the previous simulation and reduces the covariances very quickly. The result is that the internal state estimates are trusted too much and propagated with poor  $\omega$  values. As is seen in Figure 6(a), the estimate error can become worse than the GPS noise itself.



(a) Position Estimate Errors Compared to GPS Noise

(b) Heading Estimate Errors

Figure 6: GPS-Only Study with Track Slippage

The compass heading estimate errors are shown Figure 6(b). Due to the track slippage and without any direct heading measurements, the heading estimates are essentially useless for this case.

### Exaggerated Process Noise Matrix

To make the Kalman filter more robust to track slippage, we increase the process noise matrix  $[Q]$  to

$$[Q] = \sigma_{\omega}^2 [I_{3 \times 3}] = 2.00 I_{3 \times 3} \quad (\text{rad/s})^2$$

Figure 7 shows the corresponding position and heading estimate errors. As is expected, with the increased process noise matrix the filter capability is reduced. The position estimate errors are roughly within  $\pm 0.5$  meters and the heading estimates are essentially useless, even for this no-track-slip situation.

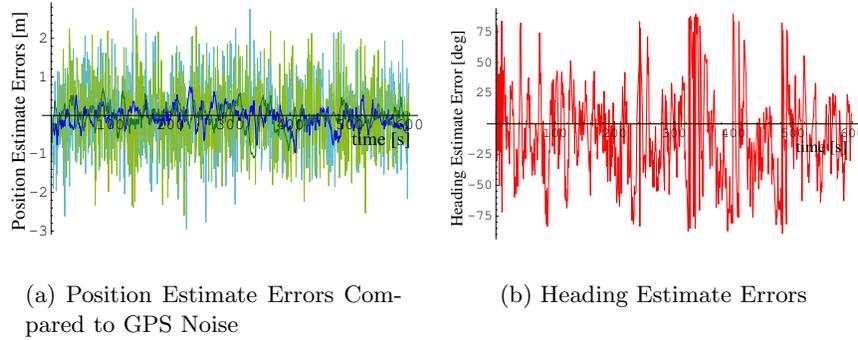


Figure 7: GPS-Only Study without Track Slippage and Exaggerated Process Noise Matrix

The effect of severe track slippage on the state estimator is shown in Figure 8. Despite the erroneous  $\omega$  measurements, the position estimate error are only slightly larger than when no track slippage is occurring. In particular, the position estimate errors are much smaller than the ones shown in Figure 6(a). The heading estimate errors at this point are starting to look like “pure noise.”

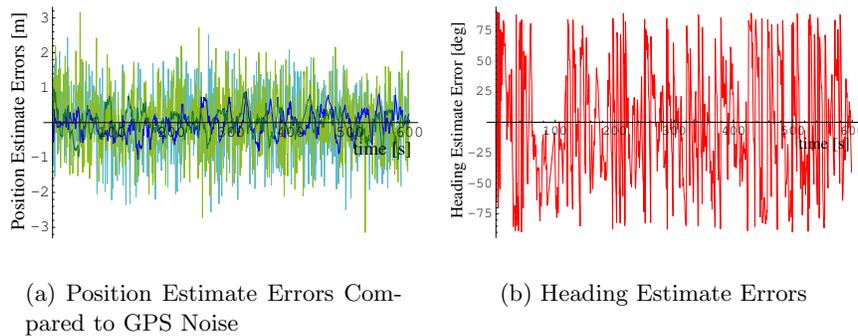


Figure 8: GPS-Only Study with Track Slippage and Exaggerated Process Noise Matrix

A pure GPS based estimator can therefore only be expected to provide reasonably accurate position estimates, not heading information. Especially when the track slippage effects are included.

## 5.2 GPS and Compass Sensor Information

### Standard Process Noise Matrix

In this study both GPS sensor (5 Hz update frequency) and compass heading information (10 Hz update frequency) are available to the MRV. The first simulation is run without slippage and the process noise matrix  $[Q]$  set to  $0.01 I_{3 \times 3}$ . The position and heading estimate errors are shown in Figure 9 relative to the respective sensor noises.

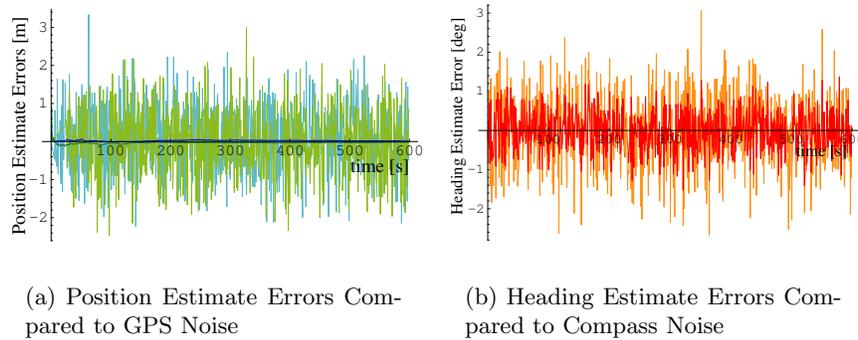
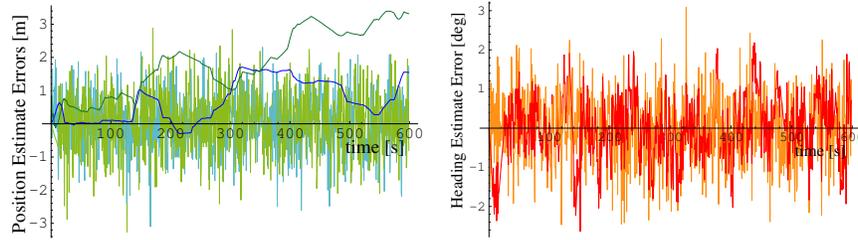


Figure 9: GPS and Compass Study with Standard Process Noise Matrix and without Track Slippage and

With the direct compass heading feedback available, the position estimate errors are very small for this ideal case without track slippage. The errors are reduced to within  $\pm 0.1$  meters. Having both GPS position updates and heading measurements also allows the heading estimate errors to be reduced to a level below the compass noise. Though the improvement here isn't very substantial. Again, this is due to noisy GPS measurements being poor at estimating the MRV attitude.

If we add the same severe track slippage as was done in the previous studies, the state estimate errors are greatly corrupted. The position and heading estimate errors are shown in Figure 10 compared to the respective sensor noise levels.

In particular, the position estimate errors begin to drift to levels larger than the GPS noise levels. The internal position estimates are being trusted to much due to the small  $[Q]$  matrix. The erroneous  $\omega$  feedback then drives the state estimates off. Because every time a GPS update arrive, there is also a compass update, the heading estimate errors are still slightly less than the compass noise level. If a GPS update occurs without simultaneously



(a) Position Estimate Errors Compared to GPS Noise

(b) Heading Estimate Errors Compared to Compass Noise

Figure 10: GPS and Compass Study with Track Slippage and Standard Process Noise Matrix

updating the heading, then the heading errors may actually grow larger than the compass noise levels.

### Exaggerated Process Noise Matrix

To make the Kalman filter more robust to track slippage, the process noise matrix is set to  $[Q] = 5000 I_{3 \times 3}$ . Note that this value is much larger than the GPS only case. Finding a suitable  $[Q]$  matrix for a given sensor package involves running some test cases and finding a medium between filter robustness and performance. The filter performance with this exaggerated process noise matrix is shown in Figure 11 for the ideal case without track slippage.

The position estimate errors are now increased to within  $\pm 0.25$  meters. While the heading estimate errors are now essentially the same as the compass noise level, adding the compass updates does provide better  $(x, y)$  estimates than the GPS-only case.

The filter performance when track slippage is added is shown in Figure 12. The position estimate errors are still well below the GPS noise levels.

In conclusion, adding a Kalman filter to the MRVs improves the GPS and compass heading estimates. In particular, by tuning the process noise matrix  $[Q]$ , the filter can be made more robust to track slippage effects. The  $(x, y)$  estimates are typically 2-3 times smaller than the GPS sensor noise during periods of severe track slippage. Without track slippage present, the errors are reduced further to relative small levels or roughly 0.3-0.5 meters.

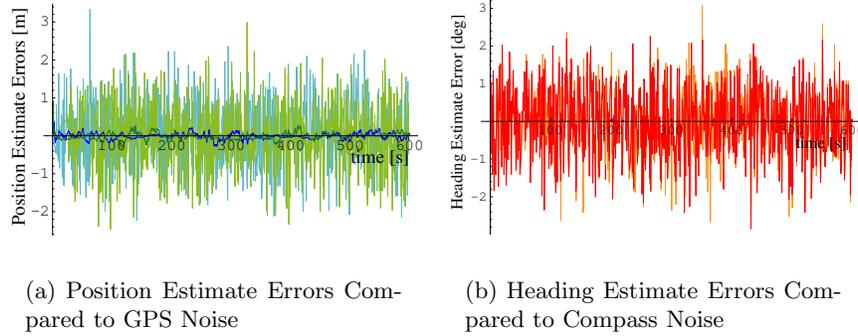


Figure 11: GPS and Compass Study with Exaggerated Process Noise Matrix and without Track Slippage

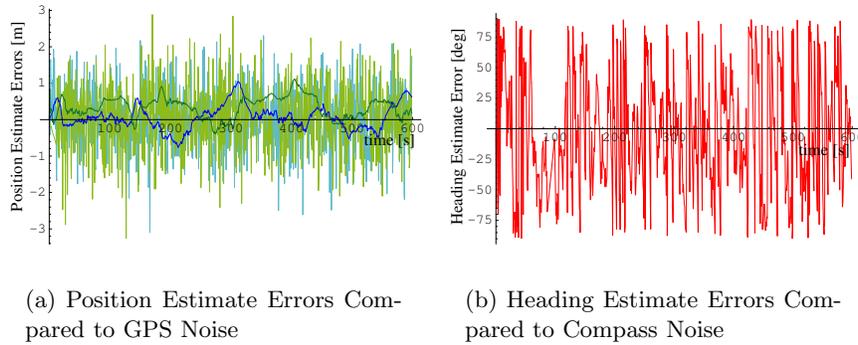


Figure 12: GPS and Compass Study with Track Slippage and Exaggerated Process Noise Matrix

## 6 Alternate Motion Sensor

The previous discussion assumed that the actual MRV motion is obtained by measuring the instantaneous left and right vehicle track speeds. The main drawback here is that this is only an indirect method of measuring motion. Under ideal circumstances, the track speeds can be directly related to the MRV motion. However, because of track slippage, this measurement can often be very erroneous.

Another method to directly measure the MRV motion is suggested here. An imaging system is proposed which would be mounted on the MRV such that it looks straight down as illustrated in Figure 13. A frame grabber cap-

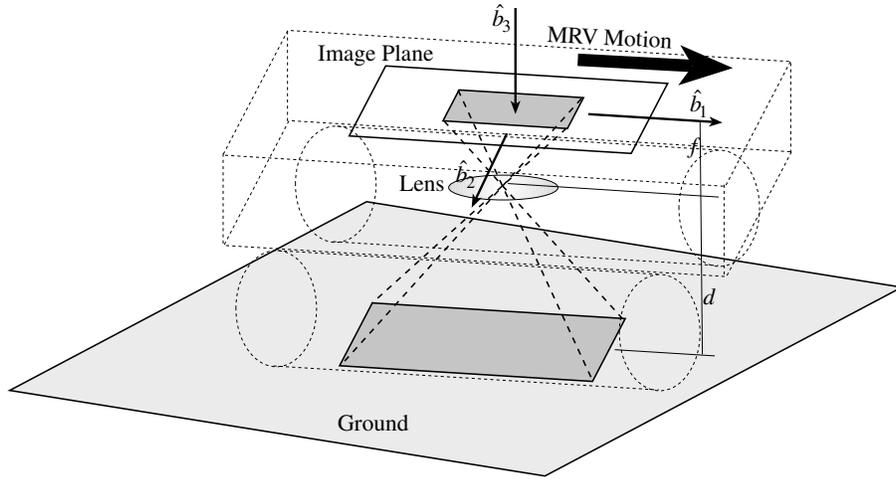


Figure 13: Motion Sensor Illustration

tures the ground picture below the MRV between two discrete time steps and compares the images. Using software similar to that used in laser doppler velocimetry in experimental windtunnel testing, it is possible to track the ground motion below the MRV.

In Figure 13 the perpendicular distance from the MRV to the ground is denoted by  $d$ . The focal length distance between the lens and the image plane is  $f$ . The body axes are assigned as is standard with aircraft type vehicles. The first (roll) body axis points out to the front of the MRV, the second (pitch) out to the right and the third (yaw) points down into the vehicle.

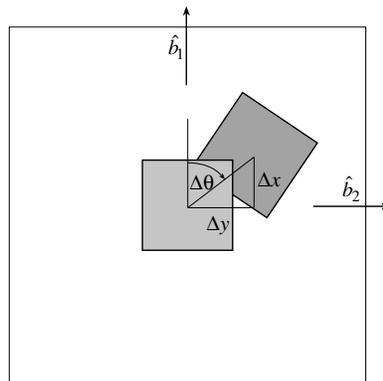


Figure 14: Frame Grabber Image Plain Illustration

The position and orientation of the identical ground patch pattern at two different time steps are shown in Figure 14. The current area below the sensor is shown in light grey, the position and orientation of the same ground pattern at the previous time step is shown in dark grey. The figure indicates that the MRV has moved forward and rotated to the right. Note that the image plane displays a mirror image of what is seen below the MRV. If the MRV is moving forward, then the *relative* ground motion should appear to be moving backward. However, due to the mirror image effect of the lens, this motion will appear in the image plane as the ground moving forward. The same holds true for rotations.

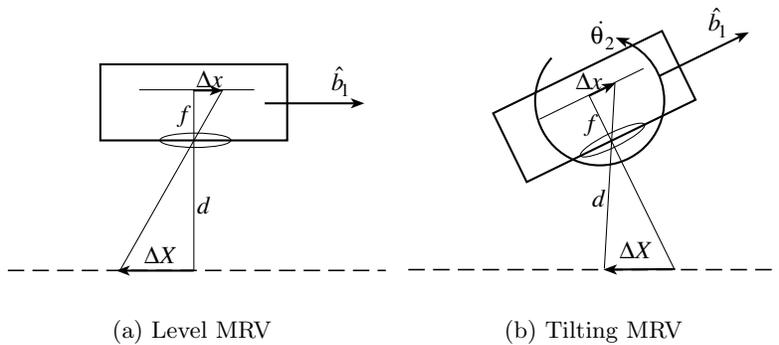


Figure 15: Comparison of Level and Tilting MRV

Using Figure 15(a), the actual ground covered  $\Delta X$  and  $\Delta Y$  is related to the corresponding image plane distances  $\Delta x$  and  $\Delta y$  through through

$$-\frac{\Delta X}{d} = \frac{\Delta x}{f} \quad (35)$$

$$-\frac{\Delta Y}{d} = \frac{\Delta y}{f} \quad (36)$$

The negative sign is due the ground moving backward as seen by the MRV if the MRV is moving forward. The translational MRV velocity  $\mathbf{v}$  is then given by

$$\mathbf{v} = -\frac{1}{\Delta t} \begin{pmatrix} \Delta X \\ \Delta Y \end{pmatrix} = \frac{d}{f \Delta t} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \quad (37)$$

The heading rate of change can be compute directly from  $-\Delta\theta/\Delta t$ . One limitation of this method is immediately apparent studying Eq. (37). Any

errors in either  $d$  or  $f$  directly relate to errors in the measured velocity  $\mathbf{v}$ . The distance  $f$  can be determined through proper calibration. However, the ground clearance  $d$  can vary with time as the MRV travels over an unknown terrain. It is therefore necessary to have some distance measuring device installed which can measure the line of sight distance from the sensor lens to the ground. For example, such a device could be a small laser ranging device. Without this capability the proposed sensor would return very erroneous velocity measurements as the MRV travels over rough, uneven ground.

Another issue that would need to be addressed is the tilting of the MRV. This may occur when the vehicle tracks roll over a rock or stump. As the vehicle continues to roll forward, it is also being rotating about the  $\hat{\mathbf{b}}_1$  axis (rolling rate  $\dot{\theta}_1$ ) or rotated about the  $\hat{\mathbf{b}}_2$  axis (pitching rate  $\dot{\theta}_2$ ). A positive pitch or roll angle is established using the right hand rule. What these rotations cause is an erroneous apparent motion by the sensor. Consider a MRV which is not translating, but which is pitching upwards at a rate  $\dot{\theta}_2$  as illustrated in Figure 15(b). The ground would also appear to move, indicating a forward motion. On unprepared ground, the MRV is expected to tilt often as it traverses over small obstacles. One method to partially compensate for this is to have the MRV be equipped with rotation rate sensors about the  $\hat{\mathbf{b}}_1$  and  $\hat{\mathbf{b}}_2$  axes. Assuming that the MRV rotation occurs about the sensor lens, then the rotational component can be extracted from the velocity measurement through

$$\mathbf{v} = \frac{d}{f \Delta t} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} + \begin{pmatrix} -\dot{\theta}_2 \\ \dot{\theta}_1 \end{pmatrix} d \quad (38)$$

However, note that in practice the MRV will typically not tilt perfectly about the sensor lens. For example, if the right tracks are rolling over a rock and the left tracks remain on the ground, then the tilting is occurring about the left MRV edge. Using the formula in Eq. (38) it is possible to partially compensate for the tilting induced velocity measurement errors.

# Bibliography

- [1] Feddema, John T., Kwan S. Kwok, Brian J. Driessen, Spletzer, Barry L., and Weber, Thomas M., “Miniature Autonomous Robotic Vehicle (MARV),” *IEEE International Conference on Robotics and Automation*, (Albuquerque, NM), April 1997. submitted for conference.
- [2] Junkins, John L., *An Introduction to Optimal Estimation of Dynamical Systems*. Alphen aan den Rijn, Netherlands: Sijthoff & Noordhoff International Publishers, 1978.
- [3] Strikwerda, T. E. and Junkins, John L., “Star pattern recognition and Spacecraft Attitude Determination,” tech. rep., U. S. Army Corps of Engineers, Engineering Topographic Laboratories, 1981.