A 3D State Space Formulation of a Navigation Kalman Filter for Autonomous Vehicles

Alonzo Kelly

CMU-RI-TR-94-19 - REV 2.0

The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

May 24, 2006

Rev 2.0 is dated May 2006 and issued to correct many typos. Changes from Rev 1.0 are identifed by change bars in the margin

This research was sponsored by ARPA under contracts "Perception for Outdoor Navigation" (contract number DACA76-89-C-0014, monitored by the US Army Topographic Engineering Center) and "Unmanned Ground Vehicle System" (contract number DAAE07-90-C-RO59, monitored by TACOM).

The views and conclusions expressed in this document are those of the author and should not be interpreted as representing the official policies, either express or implied, of the US government.

Abstract

The Kalman Filter has many applications in mobile robotics ranging from perception, to position estimation, to control. This report formulates a navigation Kalman Filter. That is, one which estimates the position of autonomous vehicles. The filter is developed according to the state space formulation of Kalman's original papers. The state space formulation is particularly appropriate for the problem of vehicle position estimation.

This filter formulation is fairly general. This generality is possible because the problem has been addressed

- in 3D
- in state space, with an augmented state vector
- asynchronously
- with tensor calculus measurement models

The formulation has wide ranging uses. Some of the applications include:

- as the basis of a vehicle position estimation system, whether any or all of dead reckoning, triangulation, or terrain aids or other landmarks are used
- as the dead reckoning element and overall integration element when INS or GPS is used
- as the mechanism for map matching in mapping applications
- as the identification element in adaptive control applications

It can perform these functions individually or all at once. The filter is formulated for a general redundant asynchronous sensor suite. It provides a single place for the integration of every sensor on a autonomous vehicle, and the measuremnt models for most of them are included. All sensors provide indirect measurements of state and any number can be accommodated.

The filter subsumes many applications of the Kalman filter to mobile robot navigation problems as special cases. It complements the RANGER vehicle controller which is the subject of another technical report that appears later in this series.

Table of Contents

1. Introduction

There seems at first glance to be two kinds of Kalman filter out there. The mobile robotics community uses the techniques of Smith and Cheeseman and kinematic analysis to “compound” and “merge” measurements together when they have “uncertainty”. The navigation industry uses dynamic analysis and talks “states”, “system model”, “measurement model”, “random walk” and “bias stability”. Faced with the problem of integrating a large number of dead reckoning and triangulation sensors together, it is not immediately clear which form of filter should be used, if indeed, there is any difference.

In investigating the difference between the two fomulations, the linear systems model of Kalman’s original papers - which happens to be the one used in the navigation industry - emerges as the most general formulation of the problem of integration of a navigation sensor suite. The kinematic model of Smith and Cheeseman is a special case which says little about dynamical systems or the use of redundant asynchronous sensors used inside the dead reckoning process itself. The state space formulation includes the concept of a system, and a model for it that provides a Kalman filter with additional information which amounts to

- an ability to predict the system state independent of measurements
- the ability to treat measurements of velocity or other derivatives of the system state, and incorporate their uncertainty into the model
- the ability to explicitly account for modelling assumptions, disturbances, etc.
- the ability to identify a system in real time.

1.1 Commentary

R. E. Kalman solved the optimal estimation problem for a certain class of linear systems about 35 years ago. There is really only ***one*** Kalman filter with a few trivial variations and most research on the Kalman filter amounts to filling in the elements in the matrices which appear in the original formulation. This is not to say that this **modelling problem** is trivial. In fact it is a very difficult, hand waving business that is hard to justify with true rigor without extensive experiments that no one has the time for.

The most important decisions in the filter design problem are the **modelling** decisions which outline the states, the measurements, the noise models, coordinate systems, linear approximations, etc. This is the real issue in practice, because the equations are 30 years old. Thus, the purpose of this document is to outline a set of plausible modelling decisions for the navigation problem for autonomous vehicles which have led to a working Kalman filter, and to record in one place the considerable tedious mathematics behind it.

When the model is finally constructed, and all of the assumptions are listed, it becomes clear that the result is a practical tool, but it should not be confused with the mathematical idealization of a Kalman filter.

1.2 Acknowledgments

This work was partially inspired by a task given to me by Martial Hebert in the introduction to mobile robots course at CMU when I served as the class teaching assistant. Ben Motazed conceived the basic idea of a general Kalman filter that could be used for position estimation and to damp relative crosstrack error for autonomous vehicle convoys. My interest in navigation theory has its roots in a collaborative effort with R. Coulter where we attempted to uncover such a theory. My own interest in vehicle calibration forced the investigation of state vector augmentation. All of these elements come together in this filter.

1.3 Notational Conventions

In the discussion, the 3 X 3 matrix R_a^b denotes a rotation matrix which transforms a displacement from its expression in coordinate system ‘a’ to its expression in coordinate system ‘b’. The 4 X 4 matrix T_a^b denotes the homogeneous transform which transforms a vector from its expression in coordinate system ‘a’ to its expression in coordinate system ‘b’. The 4 X 4 matrix P_a^b denotes a nonlinear projection operator ***represented*** as a homogeneous transform. In such notation, the vector normalization step is implicit in the transform. The matrix H_a^b denotes the Jacobian of the transformation from system ‘a’ to system ‘b’.

The entire report will be necessarily loose about the specification of derivatives. If x and y are scalars, $\bar{x}$ and $\bar{y}$ are vectors, and X and Y are matrices, then all of the following derivatives can be defined.

$\frac{\partial y}{\partial x}$	a partial derivative	$\frac{\partial y}{\partial \bar{x}}$	a gradient vector
$\frac{\partial}{\partial x}\bar{y}$	a vector partial derivative	$\frac{\partial}{\partial \bar{x}}\bar{y}$	a Jacobian matrix
$\frac{\partial Y}{\partial x}$	a matrix partial derivative	$\frac{\partial Y}{\partial \bar{x}}$	a Jacobian tensor

Generalization to higher order tensors are obvious.

Bolded italic text is used for emphasis, whereas bolded nonitalic text highlights key words that appear in the index.

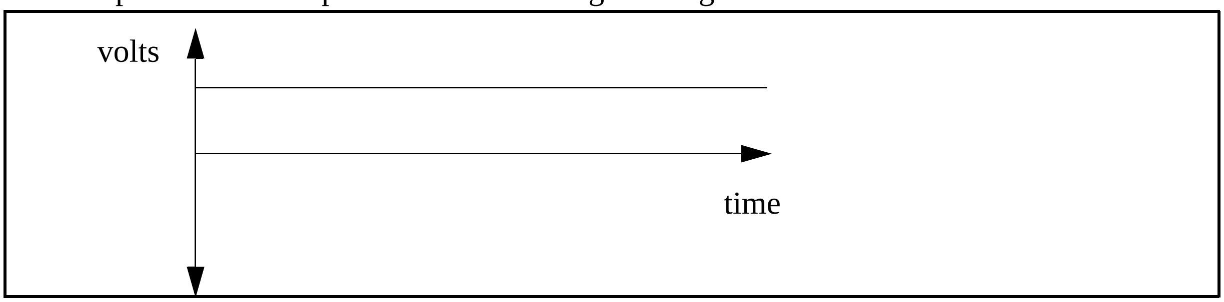
2. Random Processes

Before looking into the Kalman filter itself, it is necessary to cover the error models upon which it is based. This section presents a very abbreviated discussion of the aspects of the theory of random signals which are applicable to the report.

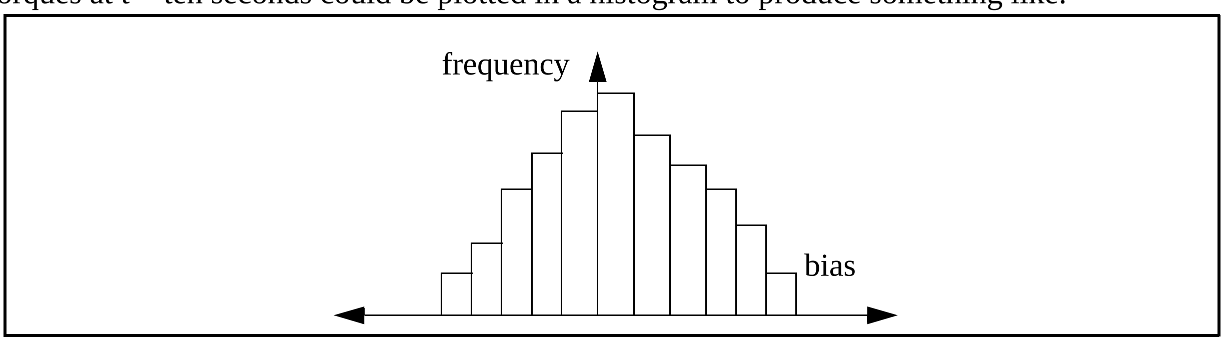
Noise sources in the Kalman filter are modelled as random processes. The **random process** can be considered to be a collection of functions of time called an **ensemble**, any one of which may be observed in a particular experiment. Exactly which one occurs is a random variable, and therefore the value of the chosen function at any time ***across all experiments*** is a random variable. Usually, the statistical variation of the ensemble of functions at any time is known.

2.1 Random Constant

Consider for example a random constant process. How can a “process” be constant and random at the same time? Well, consider a bin which contains alot of gyroscopes, and suppose that the manufacturer has said that the bias torques of all gyroscopes manufactured follow some sort of distribution. To be sure, the bias torque for any particular gyroscope could be plotted on, say, an oscilloscope and it would produce the following time signal:



Nothing random about that. Suppose, however, that alot of people reach into the bin at the same time and pick out a gyroscope, wait ten seconds, and record the bias value. The values of the bias torques at t = ten seconds could be plotted in a histogram to produce something like.



This distribution is the probability distribution of the random constant process at time equals ten seconds. Thus, while any member of the ensemble of functions is ***deterministic in time***, the ***choice*** of the function is random. In general, there may be a family of related functions of time where some important parameter varies randomly - a family of sinusoids of random amplitude, for instance. Even though there is no way to predict which function will be chosen, it may be known that all functions are related by a single random parameter and from this knowledge, it is possible to compute the distribution for the process as a function of time.

2.2 Bias, Stationarity and Ergodicity and Whiteness

A random process is **unbiased** if its expected (i.e. average) value is zero for all time. A random process is said to be **stationary** if the distribution of values of the functions in the ensemble is not varying with time. Conceptually, a movie of the histograms above for the gyroscopes for each second of time would look like a still picture. An **ergodic** random process is one where time averaging is equivalent to ensemble averaging - which is to say that everything about the process can be discovered by watching a single function for all time, or by watching all signals at a single instant. A **white** signal is one which contains all frequencies.

It is clear that fluency with these concepts requires the ability to think about a random process in three different ways:

- in terms of its probability distribution
- in terms of it evolution over time
- in terms of is frequency content.

These different views of the same process will be discussed in the next sections, as well as methods for converting back and forth between them.

2.3 Correlation Functions

Correlation is a way of thinking about both the probability distributions of a random process and its time evolution. The **autocorrelation function** for a random process x(t) is defined as:

$$R_{xx}(t_1, t_2) = E[x(t_1)x(t_2)]$$

so its just the expected value of the product of two random numbers - each of which can be considered to be functions of time. The result is a function of both times. Let:

$$x_1 = x(t_1) \qquad\qquad x_2 = x(t_2)$$

then the autocorrelation function is, by definition of expectation:

$$R_{xx}(t_1, t_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1 x_2 f(x_1, x_2) dx_1 dx_2$$

where $f(x_1, x_2)$ is the joint probability distribution. The autocorrelation function gives the "tendency" of a function to have the same sign and magnitude (i.e. to be **correlated**) at two different times.

For smooth functions it is expected that the autocorrelation function would be highest when the two times are close because the smooth function has little time to change. This idea can be expressed formally in terms of the frequency content of the signal, and conversely, the autocorrelation function says alot about how smooth a function is. Equivalently, the autocorrelation function specifies how fast a function can change, which is equivalent to saying something about the magnitude of the coefficients in its Taylor series, or its Fourier series, or its Fourier transform. All of these things are linked.

The crosscorrelation function relates two different random processes in an analogous way:

$$R_{xy}(t_1, t_2) = E[x(t_1)y(t_2)]$$

For a **stationary process** the correlation function is dependent only on the difference $\tau = t_1 - t_2$. When the processes involved are unbiased, the correlation functions give the variance and covariance of the indicated random variables. This is easy to see by considering the general formula for variance and setting the mean to zero.

Thus, for stationary unbiased processes, the correlation functions ***are*** the variances and covariances expressed as a function of the time difference:

$$R_{xx}(\tau) = \sigma^2_{xx}(\tau) \qquad\qquad R_{xy}(\tau) = \sigma^2_{xy}(\tau)$$

Also, for stationary unbiased processes:

$$\sigma^2_{xx} = R_{xx}(0) \qquad\qquad \sigma^2_{xy} = R_{xy}(0)$$

2.4 Power Spectral Density

The **power spectral density** is just the Fourier transform of the autocorrelation function, thus:

$$S_{xx}(j\omega) = \Im[R_{xx}(\tau)] = \int_{-\infty}^{\infty} R_{xx}(\tau)e^{-j\omega\tau}d\tau$$

The power spectral density is a direct measure of the frequency content of a signal, and hence, of its power content. Of course, the inverse Fourier transform yields the autocorrelation back again.

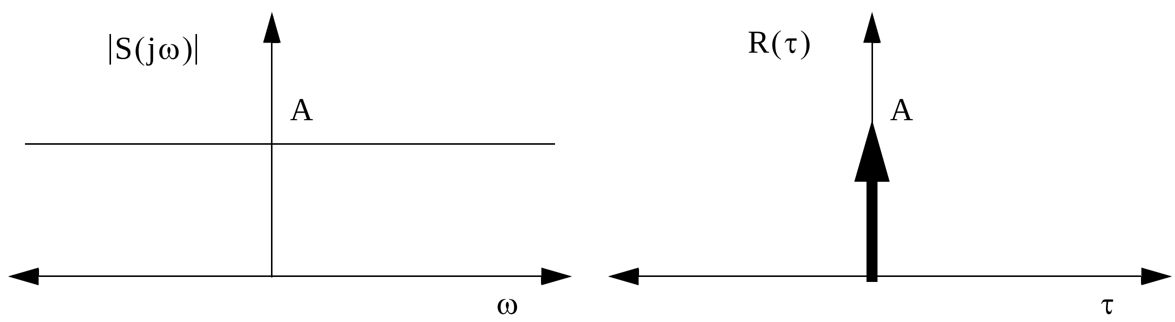
$$R_{xx}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_{xx}(j\omega)e^{j\omega\tau}d\omega$$

Similarly, the **cross power spectral density** function is:

$$S_{xy}(j\omega) = \Im[R_{xy}(\tau)] = \int_{-\infty}^{\infty} R_{xy}(\tau)e^{-j\omega\tau}d\tau$$

2.5 White Noise

White noise is defined as a stationary random process whose power spectral density function is constant. That is, it contains all frequencies of equal amplitude. If the constant **spectral amplitude** is A, then the corresponding autocorrelation function is given by the inverse Fourier transform of a constant, which is the Dirac delta.



$$R(\tau) = A\delta(\tau)$$

Thus, knowing the value of a white noise signal at some instant of time says absolutely nothing about its value at any other time, and this is because it is possible for it to jump around at infinite frequency.

2.6 Kalman Filter Noise Model

With these preliminaries, it is possible to define and understand the noise model of the Kalman filter. The noises modelled in a Kalman filter must be:

- unbiased (have zero mean for all time)
- Gaussian (have a Gaussian distribution for all time)
- white (contain all frequencies)

This model is a mathematical idealization since white noise cannot occur in nature because it requires infinite energy. The formulas of the previous section permit mapping these white noise processes onto variances and covariances which are easier to think about.

2.6.1 White Noise Process Covariance

Specifically, let the white unbiased gaussian random process $x(t)$ have power spectral density S_p. Then the variance is:

$$\sigma_{xx}^2 = S_p$$

thus ***the variance of a white noise process is its spectral amplitude***. This one is easy.

2.6.2 Random Walk Process Covariance

Now suppose that the white noise process represents the time derivative of the real variable of interest. Thus:

$$\sigma^2_{\dot{x}\dot{x}} = S_p$$

The question to be answered is what is $\sigma^2_{\dot{x}x}$, $\sigma^2_{x\dot{x}}$ and σ^2_{xx}. These can be derived from first principles. Consider that the value of x at any time must be, by definition, the integral of $\dot{x}$ as follows:

$$\sigma^2_{xx} = E(x^2) = E\left[\int_0^t \dot{x}(u)du \int_0^t \dot{x}(v)dv\right]$$

this can be written easily as a double integral since the variables of integration are independent thus:

$$\sigma^2_{xx} = E\left[\int_0^t\int_0^t \dot{x}(u)\dot{x}(v)dudv\right]$$

interchanging the order of expectation and integration:

$$\sigma^2_{xx} = \int_0^t\int_0^t E[\dot{x}(u)\dot{x}(v)]dudv$$

but now the integrand is just the autocorrelation function, which, for white noise is the Dirac delta, so:

$$\sigma^2_{xx} = \int_0^t\int_0^t S_p\delta(u-v)dudv = \int_0^t S_p dv = S_p t$$

Thus the variance of the integral of white noise grows linearly with time. Also, the standard deviation grows with the square root of time. The process $x(t)$ is called a **random walk**.

2.6.3 Integrated Random Walk Process Covariance

If the second derivative of $x(t)$ with respect to time is a white process, then the variance in $x(t)$ is:

$$\sigma^2_{xx} = E(x^2) = E\left[\int_0^t \dot{x}(u)du \int_0^t \dot{x}(v)dv\right]$$

The velocities here come from accelerations..

$$\sigma_{xx}^2 = E(x^2) = E\left[\int_0^t\left(\int_0^t \ddot{x}(u)du\right)du\int_0^t\left(\int_0^t \ddot{x}(v)dv\right)dv\right]$$

This can be written as a quadruple integral since the variables of integration are independent thus:

$$\sigma_{xx}^2 = E\left[\int_0^t\int_0^t\left(\int_0^t \ddot{x}(u)du\right)\left(\int_0^t \ddot{x}(v)dv\right)dudv\right] = E\left[\int_0^t\int_0^t\left(\int_0^t\int_0^t \ddot{x}(u)\ddot{x}(v)dudv\right)dudv\right]$$

Interchanging the order of expectation and integration:

$$\sigma_{xx}^2 = \int_0^t\int_0^t E\left(\int_0^t\int_0^t \ddot{x}(u)\ddot{x}(v)dudv\right)dudv = \int_0^t\int_0^t\left(\int_0^t\int_0^t E[\ddot{x}(u)\ddot{x}(v)]dudv\right)dudv$$

Now the inner double integral is just the autocorrelation function, which, for white noise is the Dirac delta, so:

$$\int_0^t\int_0^t E[\ddot{x}(u)\ddot{x}(v)]dudv = \int_0^t\int_0^t S_p\delta(u-v)dudv = \int_0^t S_p dv = S_p t$$

Now, the variance is:

$$\sigma_{xx}^2 = \int_0^t\int_0^t S_p t dudv = \int_0^t \frac{S_p t^2}{2}dv = \frac{S_p t^3}{6}$$

Thus the variance grows with the cube of time. This is called an **integrated random walk**. This technique can be used in general to compute the elements of a covariance matrix given the spectral amplitudes of the noises.

2.7 Scalar Uncertainty Propagation

It is useful to know the following rough rule for the propagation of error in a sum of scalar random variables with identical statistics:

$$y = \sum_{i=1}^{n} x_i \qquad \sigma_y^2 = \sum_{i=1}^{n} \sigma_{x_i}^2 = n\sigma_x^2$$

Thus, the variance in the sum is the sum of the variances of the individual terms of the sum. When they are equal, the variance grows linearly with the number of elements in the sum. In practical use, the expression gives the development of the standard deviation versus time because:

$$n = \frac{t}{\Delta t}$$

in a discrete time system. In this simple model, uncertainty expressed as a standard deviation ***grows with the square root of time***. The net result of this is that uncertainty apparently grows rapidly and then levels off as time evolves. This simply arises from the fact that truly random errors tend to cancel each other if enough of them are added.

2.8 Combined Observations of a Random Constant

Suppose several redundant measurements of a constant are obtained and that they all have identical statistics. Then the true value can be approximated as the mean of the observations. Under these circumstances, the uncertainty in this mean is:

$$y = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \left(\frac{1}{\sigma_y}\right)^2 = \sum_{i=1}^{n}\left(\frac{1}{\sigma_{x_i}}\right)^2 = \frac{n}{\sigma_{x_i}^2} \Rightarrow \sigma_y^2 = \frac{1}{n}\sigma_x^2$$

Thus, the variance in the mean decreases with increasing numbers of measurements. Under the assumptions, the standard deviation ***decreases with the square root of time***. Experimenters use this every time they take multiple observations and average. This idea that taking and merging multiple observations ***reduces the uncertainty of the combined result*** is the basic idea of the Kalman filter.

3. Fundamentals of the Discrete Kalman Filter

The Kalman filter was invented by R.E. Kalman and first published in about 1960 [6]. It is a method of estimating the state of a system based on ***recursive*** measurement of ***noisy*** data. The filter comes in many forms, including continuous and discrete time variants and linear and nonlinear variants. The Kalman filter was recognized immediately by engineers in the navigation industry as a solution to many formerly intractable problems, and it continues to be used throughout the navigation industry today.

The practical utility of the filter stems from its ability to estimate, say vehicle position, based on a number of measurements which are [2]:

- incomplete: related to some but not all of the variables of interest
- indirect: related indirectly to the quantities of interest
- intermittent: available at irregularly spaced instants of time
- inexact: corrupted by many forms of error

A central idea in the Kalman filter is to model the system of interest as a linear dynamic system which is excited by noise and whose sensors are also excited by noise. By knowing something about the nature of the noise (its first order statistics), it is possible to construct an optimal estimate of the system state even though the sensors are inexact. This is the fundamental idea of estimation theory. Without knowing the errors themselves, knowledge of their statistics allows construction of useful estimators based solely on that information.

3.1 The State Model of a Random Process

3.1.1 Continuous Model

In the linear systems model, or **state model** of a random process, if time is considered to be continuous, the process is described in the following form:

$\dot{\bar{x}} = F\bar{x} + G\bar{w}$	"STATE" OR "PROCESS" MODEL
$\bar{z} = H\bar{x} + \bar{v}$	"MEASUREMENT" OR "OBSERVATION" MODEL

3.1.2 Discrete Model

If time is considered to be discrete, the process is described in the following form:

$\bar{x}_{k+1} = \Phi_k\bar{x}_k + \Gamma_k\bar{w}_k$	"STATE" OR "PROCESS" MODEL
$\bar{z}_k = H_k\bar{x}_k + \bar{v}_k$	"MEASUREMENT" OR "OBSERVATION" MODEL

There is, of course, a way to transfer from one form to the other, which will be given shortly. In the discrete model, the names and sizes of the vectors and matrices are:

$\bar{x}_k$ is the (n X 1) system **state vector** at time t_k

Φ_k is the (n X n) **transition matrix** which relates $\bar{x}_k$ to $\bar{x}_{k+1}$ in the absence of a forcing function

Γ_k is the (n X n) **process noise distribution matrix** which transforms the $\bar{w}_k$ vector into the coordinates of $\bar{x}_k$

$\bar{w}_k$ is a (n X 1) white **disturbance sequence** or **process noise sequence** with known covariance structure.

$\bar{z}_k$ is a (m X 1) **measurement** at time t_k

H_k is a (m X n) **measurement matrix** or **observation matrix** relating $\bar{x}_k$ to $\bar{z}_k$ in the absence of measurement noise

v_k

The **state vector** for a dynamic system is any set of quantities sufficient to completely describe the unforced motion of the system. Given the state at any point in time, the state at any future time can be determined from the control inputs and the system model. Intuitively, a state vector contains values for all variables in the system up to one order less than the highest order derivative represented in the model. This is, of course, the exact number of initial conditions required to solve a differential equation.

The system model is basically a matrix linear differential equation. Such a model considers the process to be the result of passing white noise through a system with linear dynamics.The covariance matrices for the white sequences are:

$$E(\bar{w}_k\bar{w}_i^T) = \delta_{ik}Q_k \qquad E(\bar{v}_k\bar{v}_i^T) = \delta_{ik}R_k \qquad E(\bar{w}_k\bar{v}_i^T) = 0, \forall(i, k)$$

where δ_{ik} is the Kronecker delta.

3.2 A Word on the Transition Matrix

Often, the differential equations of a system in their time continuous form are known. However, a Kalman filter is generally implemented in discrete time in a computer cycling at a finite rate. In the theory of linear systems, this matter is discussed in some detail, and the transition matrix Φ figures prominently. In order to implement the Kalman filter, much of what is known about the transition matrix is unnecessary. It suffices to know that the time continuous matrix differential equation:

$$\dot{\bar{x}} = F\bar{x}$$

can always be transformed into:

$$\bar{x}_{k+1} = \Phi_k \bar{x}_k$$

The only question is how hard it is to do. When the F matrix is constant in time and the equation is linear (no elements of x occur inside F), then the transition matrix is a function only of the time step Δt and it is given by the matrix exponential:

$$\Phi_k = e^{F\Delta t} = I + F\Delta t + \frac{(F\Delta t)^2}{2!} + \ldots$$

In practice, the transition matrix can often be written by inspection. When this is not possible, writing a few terms of the above series often generates recognizable series in each element of the matrix partial sum, and the general form for each term can be generated by inspection. Other times, higher powers of F conveniently vanish anyway. When Δt is much smaller than the dominant time constants in the system, just the two term approximation:

$$\Phi_k = e^{F\Delta t} = I + F\Delta t$$

is sufficient.

3.3 Low Dynamics Assumption

When F depends on time, so does Φ and it satisfies the matrix version of the same differential equation as the state vector thus:

$$\frac{d\Phi}{dt} = F(t)\Phi$$

if F is assumed to be slowly varying relative to Δt, then the matrix exponential can still be used.This will be called the **low dynamics assumption**. It is a huge assumption as the time step gets larger[1].

1. The matrix exponential arises simply because the system model is a differential equation. When F is constant, it is the exact solution and it is possible to ***approximate the known solution*** by a Taylor series. When F is time dependent, the matrix exponential is no longer the right answer. Then the approach is to ***approximate the differential equation*** itself by assuming that F is constant to get an ***exact solution to an approximate differential equation***.

3.4 Discrete Filter Equations

The Kalman filter propagates both the state and its covariance forward in time, given an initial estimate of the state. At any point in time, the state estimate prior to incorporation of any new measurements will be denoted by $\hat{x}^-$, where the hat denotes an estimate and the super minus denotes the estimate prior to incorporation of the measurements (running one iteration of the filter equations).

The Kalman filter equations for the system model are as follows:

LINEAR KALMAN FILTER (DISCRETE TIME)

$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1}$	compute Kalman gain
$\hat{x}_k = \hat{x}_k^- + K_k[z_k - H_k \hat{x}_k^-]$	update state estimate
$P_k = [I - K_k H_k] P_k^-$	update its covariance
$\left.\begin{array}{l} \hat{x}_{k+1}^- = \Phi_k \hat{x}_k \\ P_{k+1}^- = \Phi_k P_k \phi_k^T + \Gamma_k Q_k \Gamma_k^T \end{array}\right\}$	project ahead

The proof that these equations constitute an optimal filter is remarkably straightforward for an algorithm that is only 30 years old. It is provided in [3]. ***The equations are not run all at once***. The last two run at high frequency and the first three are run when measurements are available. For the first three, the process is started by entering the prior estimate $\hat{x}_k^-$ and its covariance P_k^-. For each cycle of the system model, the state transition matrix Φ_k, and the disturbance covariance Q_k must be known. For each cycle of the Kalman filter equations, the measurement matrix H_k, and the sequence covariance R_k must be known a priori or computed based on the measurements and partial prior knowledge.

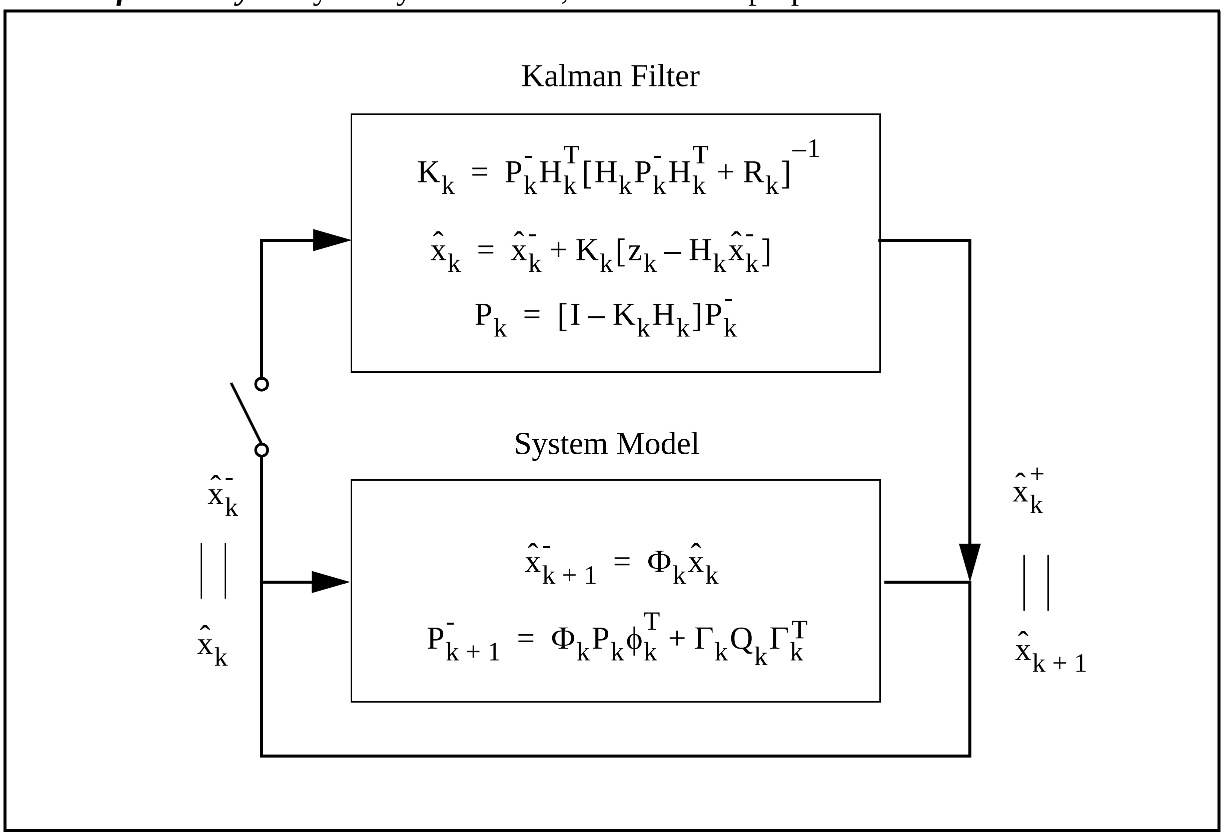
While there are several roughly equivalent forms of the Kalman filter equations and several different forms of system model, the one adopted here is in common use in the navigation industry since it avoids certain numerical problems with the matrix R_k. Another advantage of this formulation is that it requires inversions of matrices of order m (number of measurements) which is usually less than n (the number of states). Indeed, it is possible to, under assumptions necessary for other reasons, set m to 1 and ***avoid matrix inversion completely***.

3.5 Time and Updates

In thinking about the operation of the filter, it is important to distinguish the ***time element*** from the ***arrival of measurements***. Conceptually, the projection of the system state forward in time proceeds based ***solely on a measurement of time***. That is, by the definition of state, it is possible to propagate the state indefinitely forward in time based only on the initial conditions, and the forcing function, if there is one. The formulation used here assumes no forcing function.

Measurements are conceptualized as indirect measurements of state, and they arrive intermittently. When they arrive, they are incorporated into the state estimate through the Kalman gain but they are not strictly necessary. The number of measurements m, may be greater or less than n, and they may be redundant measurements of the same quantity.

This conceptualization of the filter is represented in the figure below. The state equations cycle forever, and whenever a measurement is available, the switch closes ***after the state has been predicted for that cycle*** by the system model, and the filter proper is executed.



This situation has a perfect analog in control theory. If the system model is perfect in a feedback loop, then there is no need of feedback, and the control law can run “open loop”. Sensors are incorporated to provide feedback as a pragmatic answer to the problem that the model is never perfect. Similarly, the Kalman filter is the feedback element here, which is necessary in practice, but it is important to recognize that it is completely independent of the system dynamics.

3.6 Measurement Model

Notice that the model presents how the measurements are derived from the state, that is, the operation of the sensor itself. Often, in other applications, the process which converts a measurement into a state estimate is considered. That is, the problem of ***perception***. However, this model is the simpler reverse process of ***sensing*** itself.

This is important to keep in mind because the filter is able to use underdetermined measurements of state for this reason. For example, if a single range measurement is available, the filter can use it to attempt to estimate two position coordinates or even more. This situation cannot persist for too long a period of time but single underdetermined measurements of multidimensional state vectors are quite legal.

3.7 Observability

In general, situations may arise, however, where there are not enough measurements in the entire sensor suite to predict the system state over the long term. These are called **observability** problems and they can be detected when diagonal elements of P_k are diverging with time.

Observability problems can be fixed by reducing the number of state variables (i.e by incorporating the assumption that some are not too relevant) or by adding additional sensors. Observability is a property of the entire model including both the system and the measurement model, so it changes every time the sensors change.

Formally, ***a system is observable if the initial state can be determined by the observing the output for some finite period of time***. Generalizing from Gelb [7], consider the discrete, nth order, constant coefficient linear system:

$$x_{k+1} = \Phi x_k$$

for which there are m noise free measurements:

$$z_k = Hx_k \qquad k = 0, m-1$$

where each H is an $m \times n$ matrix. The sequence of the first n measurements can be written as:

$$\begin{aligned} z_0 &= Hx_0 \\ z_1 &= Hx_1 = H\Phi x_0 \\ z_2 &= Hx_2 = H(\Phi)^2 x_0 \\ &\dots \\ z_{n-1} &= Hx_{n-1} = H(\Phi)^{n-1} x_0 \end{aligned}$$

If the initial state x_0 is to be determined from this sequence of measurements then, the matrix:

$$\Xi = \left[H^T \,\middle|\, \Phi^T H^T \,\middle|\, \dots \,\middle|\, (\Phi^T)^{n-1} H^T \right]$$

must have rank[1] n.

3.8 Uncertainty Transformation

If $\bar{z}$ is an arbitrary measurement of the vector $\bar{x}$ through some nonlinear relationship:

$$\bar{z} = f(\bar{x})$$

Then it can be easily shown using a Taylor series approximation and the definition of the covariance matrix that:

$$Cov(\Delta z) = Exp[\Delta z \Delta z^T] = HExp[\Delta x \Delta x^T]H^T = HCov(\Delta x)H^T$$

where H is the Jacobian of f. This relationship is responsible for the terms involving the measurement matrix H, the transition matrix Φ, and the Γ matrix in the Kalman filter equations.

3.9 Sequential Measurement Processing

In situations where the errors in individual measurements are uncorrelated, it can be shown that processing them one at a time gives the same result as processing them as one large block[1]. That is, the measurement matrix can be reduced to individual rows or any logical group of submatrices and presented to the filter as such.

This has extreme advantages in real time systems with intermittent asynchronous sensor suites. It allows fairly modular software implementations which adapt in real time to the presence or absence of measurements at any particular time step.

Thus, the software complication involved in restructuring the matrices to accommodate presence or absence of measurements can be completely avoided. The technique has computational advantages as well since inverting two matrices of order n/2 is much cheaper than inverting one of order n.

3.10 The Uncertainty Matrices

It is important to distinguish the different roles of the three covariance matrices in the equations. The Q_k matrix models the uncertainty which corrupts the system model. The R_k matrix models the uncertainty associated with the measurement, and its elements are expressed in the coordinate systems and units of the measurements, not the states they measure. For example, the element to be entered into R_k for a potentiometer is related to the number of counts of noise on the pot output. Finally, the P_k matrix is largely managed by the filter itself, and it gives the total uncertainty of the state estimate as a function of time.

1. Recall that the rank of a matrix is the size of the largest nonzero determinant that can be formed from it. The rank of an m X n matrix can be no larger than the smaller of m and n. A square nXn matrix of rank n is called ***nonsingular***. The rank of the product of matrices is never larger than the smallest rank of the matrices forming the product.
1. See [3] pp 264-265.

3.11 Connection to Navigation Theory

Notice that the fourth and fifth filter equations can be used to estimate the state between measurements by integrating over themselves only until a new measurement is obtained. This is the basic mechanism of augmenting **dead reckoning** by a position **fix**. The system model, whose solution is the second last equation of the filter, can be identified with the process of dead reckoning in navigation theory.

The measurement model can be identified with both the process of measuring velocity and attitude for dead reckoning purposes, or with any mechanism for generating a position fix. The distinction rests on whether the measurements project directly onto the position through the H matrix (**triangulation**) or indirectly through the Φ matrix (**dead reckoning**).

In particular, when the measurements are the positions of landmarks, the H matrix can be identified with the process which transforms landmark position measurements into vehicle position measurements. Indeed, the "measurement matrix" H_k is also the **Jacobian** in nonlinear problems and its norm is the well known **GDOP** or **geometric dilution of precision** from triangulation theory.

3.12 Connection to Smith and Cheeseman

The Kalman filter equations are generalized versions of the "merging" operation discussed in the robotics literature [19] and the state propagation equations are the "compounding" operations of dead reckoning.

4. Linearization of Nonlinear Problems

The filter formulation presented earlier is based on a linear systems model and it is therefore not applicable in situations when either the system model or the measurement relationships are nonlinear. Consider an exact nonlinear model of a system as follows:

$$\dot{\bar{x}} = f(\bar{x}, t) + g(t)\bar{w}(t)$$

$$\bar{z} = h(\bar{x}, t) + \bar{v}(t)$$

Where f and h are vector nonlinear functions and $\bar{w}$ and $\bar{v}$ are white noise processes with zero crosscorrelation. Let the actual trajectory of the system be written in terms of an approximate trajectory $\bar{x}^*(t)$ and an error trajectory $\Delta\bar{x}(t)$ as follows:

$$\bar{x}(t) = \bar{x}^*(t) + \Delta\bar{x}(t)$$

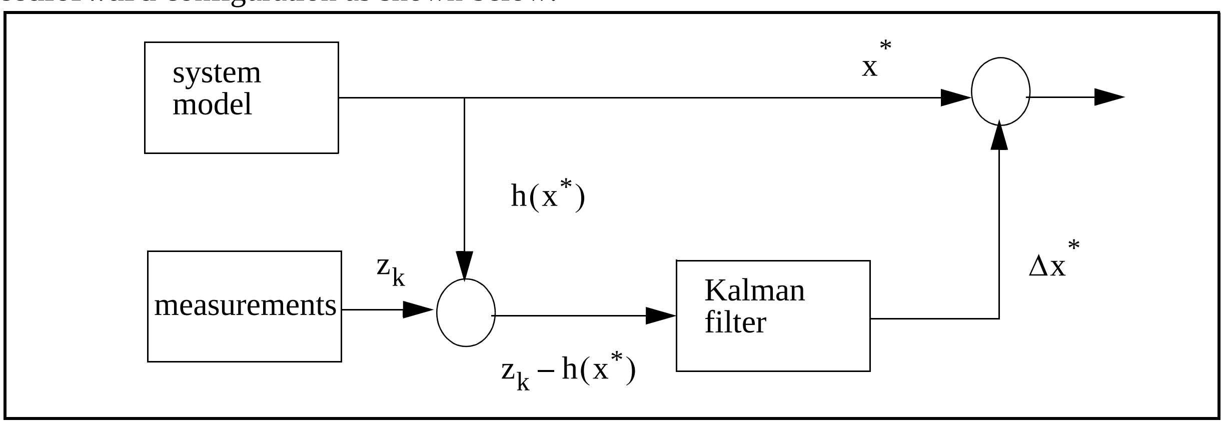
By substituting this back into the model and approximating f and h by their Jacobians evaluated at the reference trajectory:

$$\Delta\dot{\bar{x}} = \frac{\partial f}{\partial \bar{x}}(\bar{x}^*, t)\Delta\bar{x} + g(t)\bar{w}(t)$$

$$\bar{z} - h(\bar{x}^*, t) = \frac{\partial h}{\partial \bar{x}}(\bar{x}^*, t)\Delta\bar{x} + \bar{v}(t)$$

4.1 Linearized Kalman Filter

It is clear that this **linearized system model** can be used to implement a **linearized Kalman filter** because the error dynamics and error measurement relationships are linear. The ***deviation from the reference trajectory is the state vector*** and the measurements are the true measurements less that predicted by the nominal trajectory in the absence of noise. The linearized filter is used in a **feedforward** configuration as shown below:



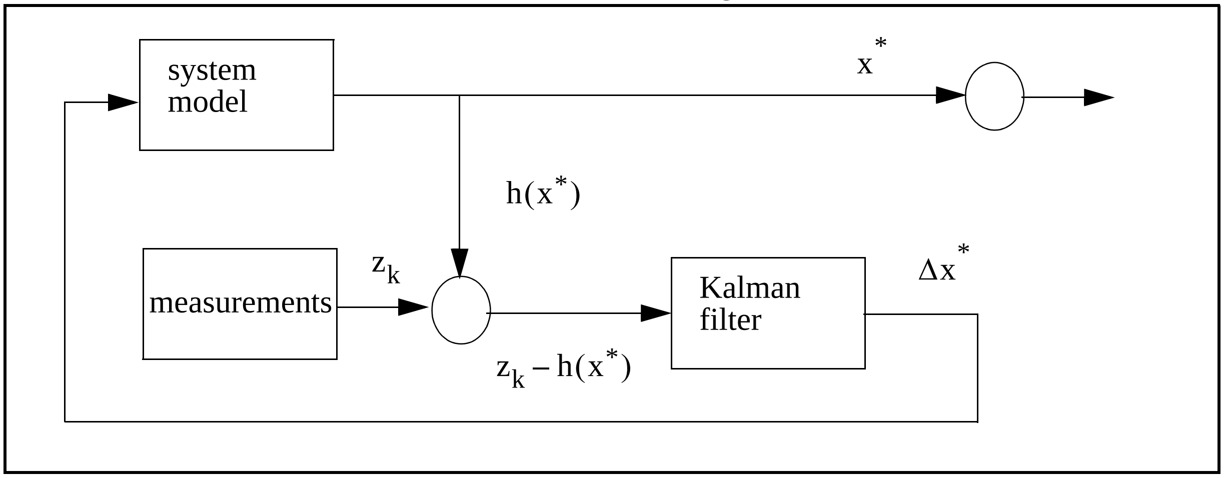
In this form, the nominal trajectory is ***not*** updated to reflect the error estimates computed by the filter. One of the primary advantages of the linearized filter is that, because it operates exclusively on errors, the unfiltered system model output provides high fidelity response in the presence of high dynamics. Such a filter is difficult to use for extended missions because, after a time, the reference trajectory may diverge to the point where the linear assumption is no longer valid across the variation in the state vector.

The feedforward model can be used to integrate an INS, or **inertial navigation system**, with a number of navigation aids. The INS is considered to be the system model and its outputs are regarded as the reference trajectory. Measurement aids are used to compute errors and they are applied to the reference to generate the combined output.

4.2 Extended Kalman Filter

In the **extended Kalman filter**, the trajectory error estimates are used to update the reference trajectory as time evolves. This has the advantage that it is more applicable to extended missions. The precise distinction between the two forms of filter is based on the measurement function $h(x^*)$, that is, whether it is updated based on the ***corrected*** (extended filter) or the ***nominal*** (linearized filter) trajectory.

The extended filter can be visualized in a **feedback** configuration as shown below:



In the case of the extended Kalman filter, it is possible to formulate the filter in terms of the state variables themselves rather than the error states. This can be seen as follows. The linearized measurement relationships are:

$$\bar{z} - h(\bar{x}^*, t) = \frac{\partial h}{\partial \bar{x}}(\bar{x}^*, t)\Delta\bar{x}$$

so that the left hand side is the “measurement” presented to the filter. The discrete time state update equation for this measurement is:

$$\Delta\hat{x}_k = \Delta\hat{x}_k^- + K_k[\bar{z}_k - h(\bar{x}_k^{-*}) - H_k\Delta\hat{x}_k^-]$$

Now, associating the last two terms in the brackets rather than the first two, define the **predictive measurement** to be the sum of the ideal measurement of the reference state and the deviation given by the Jacobian of $h(\bar{x}, t)$.

$$\hat{z}_k^- = h(\bar{x}_k^{-*}) + H_k \Delta \hat{x}_k^- = h(\hat{x}_k^-)$$

Then the **measurement residual** is the difference between the predictive measurement and the actual measurement:

$$\Delta z_k = z_k - \hat{z}_k^-$$

Under this substitution, the state update equation can be written as:

$$\hat{x}_k = \hat{x}_k^- + K_k[z_k - \hat{z}_k^-]$$

The discrete time extended Kalman filter equations for the system model are now as follows:

EXTENDED KALMAN FILTER (DISCRETE TIME)

$\dot{\bar{x}} = f(\bar{x}, t) + g(t)\bar{w}(t)$	system model	$E(\bar{w}_k \bar{w}_k^T) = Q_k$
$\bar{z} = h(\bar{x}, t) + \bar{v}(t)$	measurement model	$E(\bar{v}_k \bar{v}_k^T) = R_k$

$H_k = \dfrac{\partial h}{\partial \bar{x}}(\hat{x}_k^-)$	compute measurement Jacobian
$F_k = \dfrac{\partial f}{\partial \bar{x}}(\hat{x}_k^-)$	compute system Jacobian
$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1}$	compute Kalman gain
$\hat{x}_k = \hat{x}_k^- + K_k[z_k - h(\hat{x}_k^-)]$	update state estimate
$P_k = [I - K_k H_k] P_k^-$	update its covariance

$$\left.\begin{aligned} &\hat{x}_{k+1}^- = \phi_k \hat{x}_k \quad \text{see section 4.4.3} \\ &P_{k+1}^- = \Phi_k P_k \Phi_k^T + \Gamma_k Q_k \Gamma_k^T \end{aligned}\right\} \begin{aligned} &\text{project ahead} \\ &\text{see section 4.5} \end{aligned}$$

where the usual conversion to the discrete time model has been performed.

4.3 Taylor Remainder Theorem

One of the most useful approximation tools in all of calculus is the **Taylor remainder theorem**. It quantifies the cost of truncating a power series in terms of lost accuracy of the approximation. For reference, the theorem is:

$$f(x + \Delta x) = f(x) + f'(x)\Delta x + \frac{f''(x)\Delta x^2}{2!} + \sum_{i=3}^{n} \frac{f^{(n)}(x)\Delta x^n}{n!} + R_{n+1}(x)$$

$$R_{n+1}(x) = \frac{f^{(n+1)}(x)(x - x^*)^n}{n!}$$

Notice that, although the series is not infinite, the equality is strict if the remainder is evaluated at some point x^* between x and $x + \Delta x$. The practical significance of this extraordinarily useful result is that $(x - x^*)$ is bounded, and the derivative is also bounded, so even though x^* is not known precisely, it is possible to compute the maximum value of the truncation remainder. This is the formal basis of all linearization.

This means that it is perfectly legitimate to invoke the low dynamics assumption, and an estimate of the highest value of the first neglected derivative estimates the modelling error incurred by the assumption. The low dynamics assumption will be used to estimate the system model of a nonlinear plant and the remainder theorem will be used to estimate the error involved in doing this.

4.4 Forms of Linearization

It is very important to distinguish linearization across time and linearization across the states in any form of linearized Kalman filter. One leads to the transition matrix and the other to the system Jacobian. Four totally different cases of the linear assumption have been encountered thus far.

4.4.1 First Order Statistics

The Kalman filter itself is derived based on the assumption that the noises are Gaussian white sequences. The Gaussian assumption amounts to assuming that higher moments of the probability distributions are all zero. For this reason, the algorithm uses a covariance matrix model of uncertainty. In reality this amounts to ***approximating the probability distributions***.

4.4.2 Low Dynamics Assumption

The low dynamics assumption amounts to assuming that the dynamics matrix $F(t)$ is constant. This is a trivial kind of linearization across time where a zeroth order Taylor series approximation of the true answer is used. The true answer is:

$$F(t + \Delta t) = F(t) + \dot{F}(t)\Delta t + \frac{(\dot{F}(t)\Delta t)^2}{2!} + \dots$$

Assuming the **dynamics matrix** F to be constant is ***linearizing in time***. It is often the case in practice that the mere conversion from time continuous to time discrete form involves this approximation, although there are ways to avoid it.

4.4.3 Quasilinearization Of a Differential Equation

Notice that the transition matrix is, strictly speaking, not defined at all when the system model is nonlinear in the states. In this context, the transition matrix can be thought of as an expression of whatever technique is used to solve the original nonlinear system differential equation, which is a subject in itself, outside of estimation theory. See the linear systems literature, for example, [2] for what little is known about this matter. For a nonlinear plant, this amounts to ***approximating the system model***.

4.4.4 Linearized Filters

A separate matter dealt with in estimation theory is the propagation of uncertainty when the plant is nonlinear. Thus the linearized and extended Kalman filters incorporate an assumption of ***linearization across states*** and, ultimately, the transition matrix and the system Jacobian both appear in the formulation.

The linearized and extended Kalman filters are **first order filters**, thus:

$$\boxed{F_k = \frac{\partial f}{\partial \bar{x}}(\hat{x}_k^-)}$$

The **higher order filter** is a generalization of the EKF where a higher order Taylor series is used to approximate the nonlinearities. All of these filters are approximations, so it is pointless to argue their merits without getting quantitative about neglected terms.

When the whole list of approximations used in a particular real model are tallied, it is clear that the Kalman filter is a mathematical idealization, which happens to be useful in practice, but which can only be used in practice after some strong assumptions are made.

4.5 The Transition Matrix and System Jacobian for Nonlinear Problems

When the system differential equation is nonlinear:

$$\dot{\bar{x}} = f(\bar{x}(t), t)$$

the system Jacobian for the EKF comes from state space linearization:

$$F = \frac{\partial}{\partial \bar{x}} f(\bar{x}, t)$$

The uncertainty propagation matrix can often be approximated by:

$$\Phi = I + F\Delta t$$

and the “transition matrix” comes from time linearization because:

$$\bar{x}_{k+1} = \bar{x}_k + \dot{\bar{x}}_k \Delta t = \phi_k \bar{x}_k$$

$$\bar{x}_{k+1} = \bar{x}_k + f(\bar{x}_k, t_k)\Delta t$$

Therefore, the transition matrix can be written by inspection as an identity matrix with linear and angular velocity cross terms added (when they are present in the state vector) which are multiplied by dt.

4.6 Other Kalman Filter Algorithms

It is worth noting briefly that there are other forms of the Kalman filter. Two such forms may be applicable in special cases. The **correlated measurement and process noise** filter incorporates a model of the correlation between disturbances and measurement noise. It is also possible to readily model **deterministic inputs** in a Kalman filter. These often correspond to the command signals in automatic control applications.

4.7 The Measurement Conceptualization in the EKF

It is important to recognize that the ***measurement process itself*** is used in the ***extended*** Kalman filter, and the computation of the deviation from predicted to actual measurement is automatic in the formalism. More specifically, the state update equation is:

$$\hat{x}_k = \hat{x}_k^- + K_k[z_k - h(\hat{x}_k^-)] \qquad \text{update state estimate}$$

and this contains the computation of the predicted measurement $h(\hat{x}_k^-)$ already. The predicted measurement is computed inside the filter itself.

5. State Vector Augmentation

One of the secrets to high performance navigation systems is the mechanism by which systems utilize the Kalman filter to **identify**[1] themselves in real time. In the language of estimation theory, the mechanism is known as **state vector augmentation**. It is intimately related to the idea that, although the Kalman filter requires white noise processes, unknown system parameters and nonwhite noise sources can be modelled as the result of passing a white noise process through a system with linear dynamics.

The Kalman filter is very sensitive to the exact form of the model chosen. In particular, it requires that the errors and disturbances be precisely zero mean, gaussian, white sequences. More often than not, the driving functions are not white and they must be modelled by differential equations which relate them to fictitious white noise processes. This process of creating additional state variables allows the filter to account for measurement noise and disturbances that are not white.

While vendors often like to quote the number of states in their filter algorithms, it is often the ***choice*** of states that matters, and not their number. Indeed, filter performance often degrades in practice when too many states are used. For example, very much is known about the propagation of errors in inertial navigation systems, and intimate knowledge of the system dynamics, and the likely sources of error, and their dynamics is necessary for optimal filter performance.

In situations when disturbances and measurement errors are changing very slowly relative to the system state or measurements, the error sources themselves can be estimated and added to the system state vector. In this way, the filter estimates the system parameters as well as state. This is known as **state vector augmentation**. Additional states are assumed to follow a **correlation model** which is based upon knowledge of the underlying physics of the system.

5.1 Principle

Suppose the measurement noise $\bar{v}$ in the continuous time model:

$$\boxed{\begin{aligned} \dot{\bar{x}} &= F\bar{x} + G\bar{w} \\ \bar{z} &= H\bar{x} + \bar{v} \end{aligned}}$$

is correlated. Oftentimes it is possible to consider that the correlated measurement noise arises through passing uncorrelated white noise $\bar{w}_1$ through a system with linear dynamics (i.e by filtering it[2]) thus:

$$\boxed{\dot{\bar{v}} = E\bar{v} + \bar{w}_1}$$

Using this model, the correlated component of noise can be considered to constitute a random element in the state vector. This element is added to the existing states to form the augmented state

1. In linear system theory, "identification" is the term used for calibration of system parameters.
2. A first order differential equation is also the defining equation of a first order filter. There is a direct equivalence between the "memory" of a dynamic system which arises from its expression as a differential equation, and its ability to operate as a filter. They are the same animal under two different names.

vector:

$$\frac{d}{dt}\begin{bmatrix}\bar{x}\\ \bar{v}\end{bmatrix} = \begin{bmatrix}F & 0\\ 0 & E\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{v}\end{bmatrix} + \begin{bmatrix}G & 0\\ 0 & I\end{bmatrix}\begin{bmatrix}\bar{w}\\ \bar{w}_1\end{bmatrix}$$

Then the measurement equation becomes:

$$\bar{z} = \begin{bmatrix}H & I\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{v}\end{bmatrix}$$

With this reformulation, the measurement noise has disappeared completely because it has been absorbed into the system model. Under these circumstances, the Kalman filter equations used in this report are ***necessary*** to address the singularity of the R_k matrix.

It is also just as easy to model deterministic new state variables with the technique and these can be used to calibrate a system. For example, the random constant model described in the next section can be used to define an unknown constant, provided the measurement matrix H is updated to reflect the new state. The basic operation of the filter is to project the measurement residual onto the states, so it will determine a value for this constant and even allow it to vary slowly with time.

5.2 Correlation Models

Some important **correlation models** are as follows:

5.2.1 Random Walk

The **random walk** is also known as the **Weiner process** or **Brownian motion**. Essentially, this is the integral of a random gaussian noise signal. That is to say, the position of a drunk when each new step is of random size either forward or backward. The state differential equation is:

$$\dot{x}_i = w$$

where w is a random signal.

5.2.2 Random Constant

Also called **random bias**. It is a constant value which varies randomly from one experiment to the next, but which is both constant over time and unknown for any particular experiment. It is important for modelling system biases. The state differential equation is:

$$\dot{x}_i = 0$$

5.2.3 Random Harmonic

This is a sinusoid of random amplitude and phase. The state differential equation is:

$$\dot{x}_i = x_{i+1} + w$$

$$\dot{x}_{i+1} = -\omega^2 x_i - 2\beta x_{i+1} + (b - 2a\beta)w$$

5.2.4 Random Ramp

This is a process which grows linearly with time but has a random initial value and slope. It is important for modelling system scale errors. The state differential equation is:

$$\dot{x}_i = x_{i+1}$$

$$\dot{x}_{i+1} = 0$$

5.2.5 Gauss Markov Process

The **Gauss Markov process**, or first order Markov process, is one which has an exponentially decreasing autocorrelation function. In practical terms, this amounts to an assumption that the process exhibits little correlation between values which are sufficiently well separated in time. It is often used to represent errors about which little is known other than that they are bandlimited. The state differential equation is:

$$\dot{x}_i = -ax + w$$

When the estimation period is much smaller than the time constant of this process, it is well approximated by the random walk.

The random walk, constant, and ramp can be modelled together by only two state variables. The number of states for any correlation model depends on the order of the underlying differential equation and not on the number of parameters.

6. AHRS Dead Reckoning System Model

This section begins the business of writing out the math. The rest of the document depends heavily on the kinematic notation and relationships set out in a separate document [9]. Most of the measurement and system model is a 3D kinematic model. The reader will have to refer to this separate document in order to follow the kinematics here.

AHRS is an acronym for Attitude and Heading Reference System[1] which is used in the navigation industry to mean a suite of components used for measuring the attitude of a vehicle and sold as an integrated package. The primary distinction between the AHRS and the INS is the absence of accelerometer dead reckoning. The AHRS typically provides no position output at all and is best suited for odometric dead reckoning. This section proposes a filter for a mobile robot position estimation system which utilizes an AHRS as the attitude indicator.

6.1 Concept

The AHRS can be considered to be a sensor package consisting of any of the following components:

- integrating gyroscopes
- a magnetic compass
- inclinometers

The position estimation system will integrate these attitude indications with:

- wheel or transmission encoder
- Doppler ground speed radar
- steering wheel encoder

In many cases, either the system model or the measurement matrices or both are nonlinear so a linearized Kalman filter of some form is necessary. In order to support wide excursion missions, and because a nominal trajectory is not always available, an EKF will be formulated.

6.1.1 Deterministic Inputs

It is possible to incorporate deterministic control inputs into the Kalman filter, and this requires slight changes to the following equations. It is a simple matter to incorporate steering dynamics via the steering control signal[2]. However, in the case of autonomous commercial vehicles, little is known in land based autonomous vehicle circles about powertrain dynamics, so any such model might be grossly in error to the point of destroying the utility of the filter. For this reason, only the **unforced Kalman filter** is considered here.

A second source of deterministic inputs are the disturbances which result from terrain following in cross country vehicles. These amount to forces generated by the terrain which prevent the vehicle from sinking into the terrain. For example, a high speed vehicle pitches violently when it encounters a steep grade. These can be estimated and folded into the filter.

1. Two commercial examples of AHRS are the Teledyne VNAS and the Watson Industries AHRS. Systron Donner and Honeywell also have AHRS products.
2. Note control signals are distinguished from sensory feedback. In the case of steering, the drive amp current demand is the drive signal.

6.1.2 Low Dynamics[1] Assumption

For terrestrial vehicles operating at low speed on moderate terrain, it is very useful to adopt the assumption that the vehicle linear and angular velocities are substantially constant over a single step of the dead reckoning (or DR) calculations. This amounts to the assumption that the acceleration terms can be neglected in what is, basically, a Taylor series system model. To do this reduces the number of states in the filter by six, over a filter with acceleration states, which is well worth doing.

6.2 Nav Frame System Model

Under the low dynamics assumption, the discrete system model can be formulated as the six axis dead reckoning equations. The first question to resolve is the coordinate system in which to represent the linear velocity states. A practical choice depends on the observability question.

Let x, y, z, θ, ϕ, and ψ denote the vehicle position and attitude in the navigation frame. For notational convenience, translational and attitude variables will be grouped together as follows:

$$\bar{r} = \begin{bmatrix} x & y & z \end{bmatrix}^T \quad \bar{\rho} = \begin{bmatrix} \theta & \phi & \psi \end{bmatrix}^T \quad \bar{x} = \begin{bmatrix} \bar{r}^T & \bar{\rho}^T & \dot{\bar{r}}^T & \dot{\bar{\rho}}^T \end{bmatrix}^T$$

The state equations are then:

$$\frac{d}{dt}\begin{bmatrix} \bar{r} \\ \bar{\rho} \\ \dot{\bar{r}} \\ \dot{\bar{\rho}} \end{bmatrix} = \begin{bmatrix} [0] & [I] \\ [0] & [0] \end{bmatrix} \begin{bmatrix} \bar{r} \\ \bar{\rho} \\ \dot{\bar{r}} \\ \dot{\bar{\rho}} \end{bmatrix} \qquad \dot{\bar{x}} = F\bar{x}$$

This matrix is trivial because the assumed dynamics were trivial. It is recommended that the equations be partitioned for efficiency reasons, but they are cast in this form to identify the state variables. Given measurements of the rate variables and the time step, it is clear that the transition matrix is:

$$\bar{x}_{k+1} = \begin{bmatrix} [I] & dt[I] \\ [0] & [I] \end{bmatrix} \bar{x}_k$$

because the F matrix is constant and vanishes when squared, so that:

$$\Phi_k = I + Fdt$$

exactly.

This defines the Φ_k matrix in the model, and notice that it makes the constant velocity assumption quite explicit. This is also a literal expression of the usual equations of dead reckoning.

1. For low dynamics, read low acceleration.

6.2.1 Observability Problem

Unfortunately, the above model was attempted and found to have observability problems with the standard suite of dead reckoning sensors. The nav frame model is ideal when an INS or GPS is available to provide measurements in the nav frame, but without such sensors, the filter diverges unacceptably. Even with perfect DR sensors, this formulation does not work as a matter of mathematics.

To see this, consider a sensor suite that provides direct measurement of attitude and attitude rate, plus a measurement of the forward component of velocity. Let the vehicle drive along the x axis of the navigation frame - then, for this trajectory:

$$z = Hx \qquad \begin{bmatrix} \theta \\ \phi \\ \psi \\ \dot{x} \\ \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \theta \\ \phi \\ \psi \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix}$$

Recall that the observability question rests on the rank of the following matrix being that of the system model, which is 12 in our case:

$$\Xi = \left[H^T \,\middle|\, \Phi^T H^T \,\middle|\, \dots \,\middle|\, (\Phi^T)^{n-1} H^T \right]$$

Each vertical partition of this matrix is of the form:

$$(\Phi^T)^n H^T = \begin{bmatrix} 0&0&0&1&0&0&0&0&0&ndt&0&0 \\ 0&0&0&0&1&0&0&0&0&0&ndt&0 \\ 0&0&0&0&0&1&0&0&0&0&0&ndt \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}^T$$

so that each partition contains a column linearly independent on the corresponding columns of all other partitions. Therefore the rank is that of a single partition. Notice that for any partition, columns 1,2,3,8 and 9 are all zeros so that the rank of this matrix is 7. Hence the system is not observable.

To see this practically, notice that either a transmission encoder or Doppler groundspeed radar measures only the component of vehicle velocity which is directed along the body y axis. The measurement matrices for such sensors act so as to project the speed residual onto all three of the state variables $\dot{x}$, $\dot{y}$ and $\dot{z}$. However, once the velocity vector is of correct magnitude, all adjustment stops. So if the predicted ***speed*** is correct but the ***velocity vector*** is wrong, the filter will not rotate the velocity vector in order to correct it.

The measurement models of the nav frame model generate the body to nav transform in the measurement models of the DR sensors, so provided the measurement models of the forthcoming section are modified, the above nav frame model can be successfully used with INS or GPS, or any sensor suite which measures translation along the nonobservable axes.

Some AHRS provide accelerations in the body frame. These can be integrated in a straightforward manner with either the nav frame model or the body frame model discussed next.

6.3 Body Frame System Model

These problems can be overcome by hacking the system model, to force the velocity vector to be oriented along the body y axis. However, this would be like having one state variable and faking three others. In truth, only one is independent, so there **is** only one. There is a more elegant way. Specifically, the state variables are reduced in number and the system model is reformulated to ***explicitly*** assume that:

- the vehicle translates only along the body y axis
- the vehicle rotates only around the body z axis

These are assumptions, of course, which will be violated under certain circumstances. However, there is no choice without some mechanism to measure translation in the vertical and sideways. This will be called the **principal motion assumption**, and it amounts to replacing six legitimate states with two special combinations of themselves. The principal motion assumption solves the observability problem.

6.3.1 State Vector

In this model, the state variables are:

$$\boxed{\bar{x} = \begin{bmatrix} x & y & z & V & \theta & \phi & \psi & \dot{\beta} \end{bmatrix}^T}$$

where V is the projection of the vehicle velocity onto the body y axis, and $\dot{\beta}$ is the projection of the vehicle angular velocity onto the body z axis.

Starting from scratch, the observability problem is fixed by writing the continuous time system differential equations as follows:

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \\ V \\ \theta \\ \phi \\ \psi \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -Vs\psi c\theta \\ Vc\psi c\theta \\ Vs\theta \\ 0 \\ \dot{\beta}s\phi \\ -\dot{\beta}t\theta c\phi \\ \dot{\beta}c\phi / c\theta \\ 0 \end{bmatrix}$$

Where the result incorporates kinematic transforms from [9]. Technically, this system model is ***nonlinear***, therefore it must be linearized according to the rules for an EKF.

6.3.2 System Jacobian

The true linearized continuous time differential equation is:

$$\frac{d}{dt}\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta V \\ \Delta\theta \\ \Delta\phi \\ \Delta\psi \\ \Delta\dot{\beta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -s\psi c\theta & Vs\psi s\theta & 0 & -Vc\psi c\theta & 0 \\ 0 & 0 & 0 & c\psi c\theta & -Vc\psi s\theta & 0 & -Vs\psi c\theta & 0 \\ 0 & 0 & 0 & s\theta & 0 & 0 & Vc\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dot{\beta}c\phi & 0 & s\phi \\ 0 & 0 & 0 & 0 & -\dot{\beta}c\phi / c^2\theta & \dot{\beta}t\theta s\phi & 0 & -t\theta c\phi \\ 0 & 0 & 0 & 0 & \dot{\beta}s\theta c\phi / c^2\theta & -\dot{\beta}s\phi / c\theta & 0 & c\phi / c\theta \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta V \\ \Delta\theta \\ \Delta\phi \\ \Delta\psi \\ \Delta\dot{\beta} \end{bmatrix}$$

which gives the F matrix of the EKF.

6.3.3 Transition Matrix

Remembering earlier comments on linearization, the above system Jacobian matrix should be distinguished from the transition matrix. Admittedly, Φ does not exist for nonlinear plants, but the system differential equation can be approximated by simply reinvoking the low dynamics assumption. The above nonlinear plant is linearized in time as follows. Under the assumption that V and $\dot{\beta}$ represent all components of the linear and angular velocity of the vehicle, then the dead reckoning equations are:

$$\begin{bmatrix} x \\ y \\ z \\ V \\ \theta \\ \phi \\ \psi \\ \dot{\beta} \end{bmatrix}_{K+1} = \begin{bmatrix} 1 & 0 & 0 & -s\psi c\theta dt & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & c\psi c\theta dt & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & s\theta dt & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & s\phi dt \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -t\theta c\phi dt \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & c\phi dt/c\theta \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_K \begin{bmatrix} x \\ y \\ z \\ V \\ \theta \\ \phi \\ \psi \\ \dot{\beta} \end{bmatrix}_K$$

where the kinematic transforms of [9] were used to compute the projections of the velocity and angular velocity vectors. This transition matrix is just the equations of 3D dead reckoning. Formally, this matrix has been generated by reexpressing the nonlinear plant as a matrix function of the states. It is first order in time because the highest order time derivatives included are order one.

This matrix representation is useful for preserving the matrix form of implementations of system dynamics.

6.4 Advantages

It is worth noting the advantages of applying the Kalman filter to the problem in such a simple case. The advantage comes mainly from the redundancy of measurements. First, the encoder and doppler provide redundant measurements of velocity which can be used to improve an estimate. The 3 axis AHRS output makes 3D dead reckoning possible and the steering encoder may be able to improve attitude response of the AHRS around the most important axis for dead reckoning purposes since it does not suffer from the settling problems of the AHRS.

Accelerometers or acceleration states would permit modification of the measurement noises to account for this. Of course, if any fix information is available, the formulation provides a simple mechanism for improving a dead reckoned estimate considerably.

The advantage of the EKF over the linear KF is that the uncertainty propagation is more accurate. For example, if a poor heading reference is used, it is the direction transverse to the direction of travel which can incur the most error. The cross coupling terms in this system Jacobian can correctly account for this.

Another advantage, specifically of the state space formulation of the Kalman filter, is its true filtering behavior. Appropriate choice of the system disturbance model will cause the filter to automatically reject erroneous high frequency input. This arises as a side effect of considering the new estimate to be a weighted sum of the old estimate and the estimate generated by the measurements. In practice, however, a “spiky” sensor is best managed by a true low pass prefiltering operation.

7. AHRS Dead Reckoning Measurement Model

Certain of the measurement relationships for dead reckoning sensors are nonlinear. Therefore, a linearized Kalman filter is necessary in this application whether or not the system model is nonlinear. A slow moving vehicle assumption has already been adopted in the system model, so it is unnecessary to use a simple linearized filter in order to track high accelerations. Therefore, an Extended Kalman filter will be formulated. This section provides the 3D measurement models for all of the sensors for such a filter.

One of the advantages of the body frame system model is that almost all of the measurement models are trivial. The tradeoff is that now the system model has rotation transforms in it. These rotation matrices cannot be avoided - they can only be moved around.

7.1 Encoder

While, strictly speaking, a transmission encoder measures differential distance, it can be considered to be integral with the system clock and therefore it is a device which measures velocity. Relative computer clock accuracies usually exceed one part in 100,000, so they can be considered to be perfect. This avoids the considerable difficulty associated with the arc length derivative called **curvature** in a model where everything else is a time derivative.

The measurement model is trivial.

$$V_{enc} = V \qquad H_{enc} = \frac{\partial V_{enc}}{\partial \bar{x}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

7.2 Doppler Groundspeed Radar

The Doppler sensor has a model similiar to the encoder. As a **range rate** measurement device, its output must be calibrated to recover the constant cosine of the slant angle between the direction of the beam and the direction of the body forward axis. This amounts to a scale factor which can be assumed to have been calibrated out, so that V_{dop} is considered to represent the vehicle speed[1].

$$V_{dop} = V \qquad H_{dop} = \frac{\partial V_{dop}}{\partial \bar{x}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

7.3 Compass

Compasses differ in their ability to reduce heeling error, and may have nontrivial dynamics. The fluxgate can sometimes handle as much as 45 degrees of heel without significant error. Newer three coil devices are mostly immune to heeling error. With no information on compass dynamics or construction, the following model can be adopted. Under the assumption that **variation** and

1. As an aside, a second doppler radar mounted transverse near the front of an Ackerman steer vehicle should provide a good redundant measurement of yaw rate. Note however, that the measurement must be adjusted for pitch and roll.

deviation[1] are already accounted for in the measurement.

$$\psi_{com} = \psi \qquad H_{com} = \frac{\partial \psi_{com}}{\partial \bar{x}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

7.4 AHRS

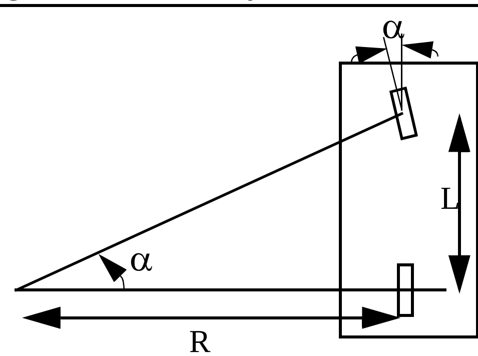
The clinometers, compass, and gyros in an AHRS can be assumed to measure the vehicle attitude directly so their measurement matrices are the identity matrix. For example:

$$\bar{\rho}_{ahrs} = [I]\bar{\rho} \qquad H_{\rho} = \frac{\partial \bar{\rho}_{ahrs}}{\partial \bar{x}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

7.5 Steering Wheel Encoder

A steering wheel encoder provides a low fidelity measurement of trajectory curvature, and hence, when multiplied by the speed, a measurement of the rate of rotation of the vehicle about the body z axis[2]. The 3D formulation of this relationship is surprisingly difficult. Let the angular velocity vector directed along the body z axis be called $\dot{\beta}$. Using the bicycle model approximation, the path curvature κ, radius of curvature R, and steer angle α are related by the wheelbase L.

$$\kappa = \frac{1}{R} = \frac{t\alpha}{L} = \frac{d\beta}{ds}$$



Where $t\alpha$ denotes the tangent of α. Rotation rate is obtained from the speed V as:

$$\dot{\beta} = \frac{d\beta}{ds}\frac{ds}{dt} = \kappa V = \frac{Vt\alpha}{L}$$

Finally, then, the steer angle α is the quantity indicated by the encoder. It is an indirect measurement of the ratio of $\dot{\beta}$ to velocity through the measurement function:

$$\alpha = atan\left(\frac{L\dot{\beta}}{V}\right) = atan(\kappa L)$$

1. These terms and others are defined later in the report.
2. Note that this is not yaw. Yaw is measured about the z axis of the navigation frame. Consider driving up a steep hill. The vehicle steers in the plane of the hill, not about the gravity vector.

The Jacobian is the measurement matrix. It is easier to formulate the measurement matrix by utilizing the chain rule and considering the product κL to constitute a single variable.

$$H_\alpha = \begin{bmatrix} 0 & 0 & 0 & \frac{\partial\alpha}{\partial V} & 0 & 0 & 0 & \frac{\partial\alpha}{\partial\dot{\beta}} \end{bmatrix}$$

$$\frac{\partial\alpha}{\partial V} = \left(\frac{1}{1 + (\kappa L)^2}\right)\left(-\frac{\kappa L}{V}\right) = \left(\frac{-L\dot{\beta}}{V^2 + (L\dot{\beta})^2}\right)$$

$$\frac{\partial\alpha}{\partial\dot{\beta}} = \left(\frac{1}{1 + (\kappa L)^2}\right)\left(\frac{\kappa L}{\dot{\beta}}\right) = \left(\frac{LV}{V^2 + (L\dot{\beta})^2}\right)$$

Note that V is derived from the states and not from the encoder or radar. In general, any measurement model uses only the current state and parameter estimates and never any other measurement. Both of the rightmost expressions can be computed when either the linear or angular velocity vanishes. When both vanish, the expressions are of the form $0/0$ and L'hopital's rule does not provide a solution (they retain their $0/0$ form).

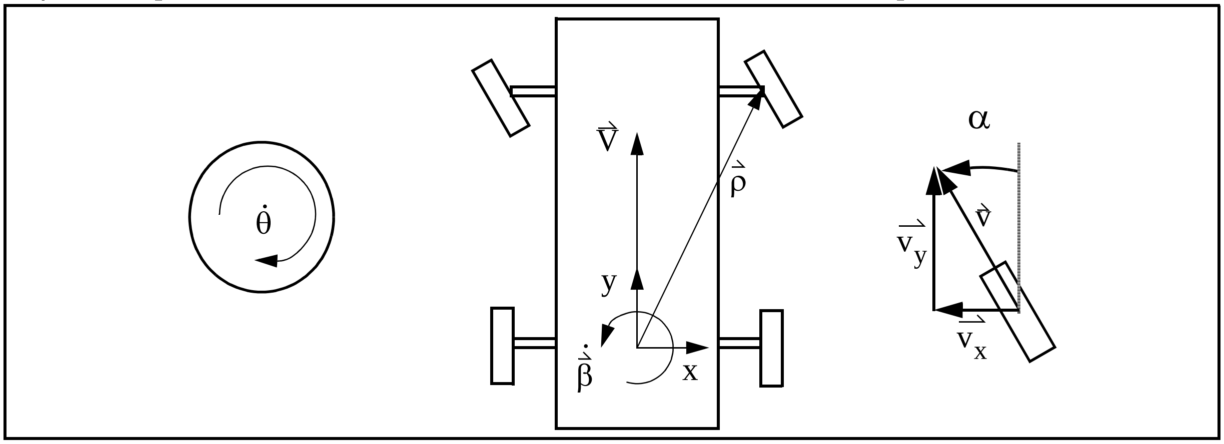
Physically, the steer angle is irrelevant when the vehicle is not moving, so the measurement must be discarded.

The steering measurement model is also unique in that it is the only DR measurement model for which the predictive measurement is not directly equal to a state. This results simply from the choice of the steer angle as the measurement instead of the angular velocity state.

7.6 Wheel Encoders

It is more convenient on many vehicles to measure wheel rotations rather than engine or transmission rotations. Indeed, if the wheels are actuated by electric motors, there is no engine or central transmission from it to the wheels at all. It is important to distinguish several kinds of measurements that may be available from instrumented wheels. A fixed wheel is permitted to rotate about a single axis - the one associated with forward motion. A free wheel may rotate about a vertical axis as well as the axis associated with forward motion. Either of these two degrees of freedom may be powered or not and either may be instrumented.

Consider a single wheel on a vehicle that has two degrees of rotational freedom as shown below. Let $\vec{\rho}$ be the position vector of the wheel relative to the vehicle control point. Let the wheel radius



be r. It is simplest to ***formulate the measurements in the body frame***. The velocity of the end of the wheel axle relative to the world is available from vector algebra as:

$$\vec{v} = \vec{V} + \dot{\vec{\beta}} \times \vec{\rho} = V\hat{j} + \dot{\beta}\hat{k} \times (\rho_x\hat{i} + \rho_y\hat{j})$$

$$\vec{v}_x = -\dot{\beta}\rho_y\hat{i} \qquad \vec{v}_y = (V + \dot{\beta}\rho_x)\hat{j}$$

7.6.1 Steer Angle

Now the available measurements actually invert these relationships. First the steer angle α of the wheel and its gradient are:

$$\alpha = atan(\sigma) = atan(v_y/v_x)$$

$$\frac{\partial\alpha}{\partial V} = \frac{\partial\alpha}{\partial\sigma}\frac{\partial\sigma}{\partial V} = \left(\frac{1}{1+\sigma^2}\right)\frac{1}{v_x} = -\left(\frac{1}{\dot{\beta}\rho_y}\right)\left(\frac{1}{1+\sigma^2}\right)$$

$$\frac{\partial\alpha}{\partial\dot{\beta}} = \frac{\partial\alpha}{\partial\sigma}\frac{\partial\sigma}{\partial\dot{\beta}} = \left(\frac{1}{1+\sigma^2}\right)\left(\frac{\rho_y v_y + \rho_x v_x}{v_x^2}\right) = (V\rho_y)\left(\frac{1}{1+\sigma^2}\right)$$

This is a measurement of the ratio of angular to linear velocity and hence is a measure of curvature just as is the Ackerman steer angle.

7.6.2 Free Wheel Velocity

A free wheel will rotate automatically about the body z axis by the necessary steer angle due to friction. Its measurement relationship in radians is:

$$\dot{\theta} = \frac{1}{r}\left(\sqrt{v_x^2 + v_y^2}\right) = \frac{1}{r}\left[\sqrt{(\dot{\beta}\rho_y)^2 + (V + \dot{\beta}\rho_x)^2}\right]$$

$$\frac{\partial}{\partial V}\dot{\theta} = \frac{1}{r}\left(\frac{2v_y}{2v}\right) = sin(\alpha) \qquad \frac{\partial}{\partial \dot{\beta}}\dot{\theta} = \frac{1}{r}\left(\frac{2v_x\rho_y + 2v_y\rho_x}{2v}\right) = \frac{1}{r}(v_x cos\alpha + v_y sin\alpha)$$

This is a measurement that responds to both the linear and angular velocity of the vehicle but they cannot be distinguished from a single measurement.The filter will automatically distinguish linear and angular velocity when two or more wheel velocities are measured.

7.6.3 Fixed Wheel Velocity

A fixed wheel will not rotate automatically about the body z axis. Its measurement relationship in radians is:

$$\dot{\theta} = \vec{v} \bullet \hat{j} = v_y/r = \frac{1}{r}(V + \dot{\beta}\rho_x)$$

$$\frac{\partial}{\partial V}\dot{\theta} = \frac{1}{r} \qquad \frac{\partial}{\partial \dot{\beta}}\dot{\theta} = \frac{\rho_x}{r}$$

Again, this is a measurement that responds to both the linear and angular velocity of the vehicle but they cannot be distinguished from a single measurement. The filter will automatically distinguish linear and angular velocity when two or more wheel velocities are measured.

7.7 Complete DR Measurement Matrix

The complete measurement matrix for the 3D AHRS dead reckoning system is then:

$$\bar{z} = \left[V_{enc}\ V_{dop}\ \psi_{com}\ \theta_{ahrs}\ \phi_{ahrs}\ \psi_{ahrs}\ \alpha\right]^T$$

$$\bar{x} = \left[x\ y\ z\ V\ \theta\ \phi\ \psi\ \dot{\beta}\right]^T$$

$$\frac{\partial\alpha}{\partial V} = \frac{-L\dot{\beta}}{1 + (L\dot{\beta})^2}$$

$$\frac{\partial\alpha}{\partial\dot{\beta}} = \frac{LV}{1 + (L\dot{\beta})^2}$$

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{\partial\alpha}{\partial V} & 0 & 0 & 0 & \frac{\partial\alpha}{\partial\dot{\beta}} \end{bmatrix}$$

If direct access to the attitude package is available, additional states can be added to the filter to model the dynamics of these sensors. As currently formulated, there is little coupling between the attitude reference and the odometry.

8. AHRS Dead Reckoning Uncertainty Model

In theory, one must estimate every element of a covariance matrix in order to provide the filter with the information it needs. In practice, there are often correlated and systematic error sources which are roughly known in magnitude to far exceed any random errors but which are nonetheless not known well enough to model.

Often, there is also no knowledge of correlation of error sources, and both the Q matrix and the R matrix are assumed to be diagonal.

$$Q = \mathrm{diag}\left[\sigma_x^2\ \sigma_y^2\ \sigma_z^2\ \sigma_V^2\ \sigma_\theta^2\ \sigma_\phi^2\ \sigma_\psi^2\ \sigma_{\dot{\beta}}^2\right]$$

$$R = \mathrm{diag}\left[\sigma_{enc}^2\ \sigma_{dop}^2\ \sigma_{com}^2\ \sigma_{pitch}^2\ \sigma_{roll}^2\ \sigma_{yaw}^2\ \sigma_{steer}^2\right]$$

The P matrix will evolve off diagonal terms naturally as the filter runs. Having taken the **uncorrelated measurement error assumption**, the remaining issue is the estimation of systematic error sources for:

- each state
- each measurement

8.1 Perspective

A Kalman filter is a mathematical idealization that happens to be useful in practice. However, it is important to note that ***there is a big difference between an optimal estimate and an accurate estimate***. In practical use, the uncertainty estimates take on the significance of ***relative weights*** of state estimates and measurements. So it is not so much important that uncertainty is absolutely correct as it is that it be relatively consistent across all models. For example, if a superb GPS fix becomes suddenly available, then if the filter treats it as far more trustworthy than the rest of the information available, then the correct things will happen.[1]

1. There is, however, the issue of discontinuous jumps in the state which may be undesirable even if the state estimate is improved.

8.2 State Uncertainty

The state uncertainty model models the disturbances which excite the linear system. Conceptually, it estimates how bad things can get when the system is run open loop for a given period of time. A useful effect of the state uncertainty model is that it can be used to filter high frequencies from the state estimate as a by-product of its operation. It is difficult to estimate the driving white sequences which drive the system model that are required by the filter. One method is based on the interpretation of the Q matrix as the weight of the dynamics prediction from the state equations ***relative*** to the measurements.

Given the assumption of low dynamics, and the neglecting of the entire control input, the errors associated with these assumptions far outweigh the effects of any truly random forcing function. In the absence of any other information, a plausible approach is to estimate error as the Taylor remainder[1] in the dead reckoning equations, because, after all, dead reckoning is a truncated Taylor series in time. The Q_k matrix can be assumed diagonal, and its elements[2] set to the predicted magnitude of the truncated terms in the constant velocity model. They can arise from:

- disturbances such as terrain following loads
- neglected control inputs such as sharp turns, braking or accelerating
- neglected derivatives in the dead reckoning model
- neglected states

8.2.1 The Gamma Matrix

The Γ matrix is convenient because the uncertainties of some of the variables can be represented in the body frame where they can be arrived at by intuition and they will be automatically converted as necessary. Let the Γ matrix be given by:

$$\Gamma = \begin{bmatrix} [R_b^n] & 0 & [0] & 0 \\ 0 & 1 & 0 & 0 \\ [0] & 0 & [\Omega] & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad Q = \mathrm{diag}\left[\sigma_x^2 \ \sigma_y^2 \ \sigma_z^2 \ \sigma_V^2 \ \sigma_\theta^2 \ \sigma_\phi^2 \ \sigma_\psi^2 \ \sigma_{\dot\beta}^2\right]$$

$$\Omega = \begin{bmatrix} c\phi & 0 & s\phi \\ t\theta s\phi & 1 & -t\theta c\phi \\ -\frac{s\phi}{c\theta} & 0 & \frac{c\phi}{c\theta} \end{bmatrix} \qquad R_b^n = \begin{bmatrix} (c\psi c\phi - s\psi s\theta s\phi) & -s\psi c\theta & (c\psi s\phi + s\psi s\theta c\phi) \\ (s\psi c\phi + c\psi s\theta s\phi) & c\psi c\theta & (s\psi s\phi - c\psi s\theta c\phi) \\ -c\theta s\phi & s\theta & c\theta c\phi \end{bmatrix}$$

The Ω matrix contains infinite entries as the vehicle approaches 90 degrees of pitch because the “yaw” and “roll” axes become coincident then. This is an essential singularity of the Euler angle definition of 3D attitude.

1. Clearly, this approach is equivalent to associating the magnitude of the truncated term with the spectral amplitude of a white sequence and integrating over the time step to get the variance.
2. Remember that the elements in the covariance matrix are the variances, not the standard deviations, so the following expressions must all be squared when used.

8.2.2 Linear Position States

For example, the translational uncertainty can be set to one half the maximum acceleration times the square of the time step. This is the error expected when constant velocity is assumed. This gives:

$$\sigma_x = \sigma_y = \sigma_z = \frac{a_{max}(\Delta t)^2}{2}$$

This error source alone is expected to grow roughly with the square root of the number of observations because:

$$\sigma_{total}^2(t) = \sum_{i=1}^{n} \sigma_i^2$$

$$\sigma_{total}(t) = \sqrt{n}\sigma_i = \frac{a_{max}(\Delta t)^2 \sqrt{t/(\Delta t)}}{2}$$

For a sampling rate of 10 Hz this gives about 10 meters of accumulated error per g of acceleration per hour of operation. For a rate of 1 Hz it is 300 meters per g per hour[1]. A very rough number for the maximum acceleration is about 1/10 g on roads and perhaps 1 g on rough terrain. The number is not truly acceleration - it represents the best guess for wheel slip, vehicle acceleration, and terrain disturbances

A better model would also account for the fact that the errors vary greatly with their direction with respect to the vehicle. Let the maximum acceleration be a vector quantity. Its components directed along the body x, y, and z axes are roughly weighted as follows:

$$\bar{a}_{max} = \begin{bmatrix} 0.1a_{max} & 1.0a_{max} & 0.3a_{max} \end{bmatrix}^T$$

A basic assumption of encoder dead reckoning is that vehicle motion is always aligned with the forward body axis. This assumption is violated in situations where the wheels slip laterally and it is completely violated when the terrain is not aligned with the body frame. One important special case of the latter is the deflections of the vehicle suspension. This is a very large error source which is appropriately modelled in the state equations and not in the sensors, since it amounts to a modelling assumption. A nav frame formulation would not suffer from this type of error.

Nevertheless, it is generally the case that a vehicle does not slide sideways or accelerate upward very quickly, so this model takes this into account. The uncertainties expressed in the nav frame are then automatically computed by the Γ matrix.

1. This is the formal reason why velocity dead reckoning must cycle quickly on an accelerating vehicle.

8.2.3 Angular Position States

For the angular position states, there are not even measurements of velocity in the DR equations, for all axes, so the truncation error is a velocity term. These uncertainties cannot be left at zero because the filter will then first of all not project forward in the state equations, and secondly, compute zero gains for the measurements. The states will not move at all. When the measurement residuals are computed for these, the state vector is one cycle old and has not moved. A very rough error estimate for these is:

$$\sigma_\theta = \sigma_\phi = \sigma_\psi = \Omega_{max}\Delta t$$

$$\sigma_{total}(t) = \sqrt{n}\sigma_i = \frac{\Omega_{max}\Delta t\sqrt{t/(\Delta t)}}{2} = \frac{2\pi f\Delta t\sqrt{t/(\Delta t)}}{2}$$

Where f is the estimated natural frequency of the vehicle. Using a cycle rate of 10 Hz and a vehicle natural frequency of 1/24 Hz or 15 degrees = 1/4 rad per second, this gives about 4 rads per hour of operation. This may sound unreasonable, but remember that the issue at hand is the error in predicting constant attitude for a vehicle which can rotate at 15 degrees per second over an hour with no measurement. This simply tells the filter to trust the readings of the sensors now instead of ten minutes ago and forces the angular states to track the sensors.

A more refined model could make explicit use of the assumption that the vehicle cannot pitch or roll more than about 15 degrees, but this is not really necessary unless the sensors have drift problems[1].

One addition to the model is to account for the fact that the errors vary greatly with their direction with respect to the vehicle. Let the maximum angular velocity be a vector quantity. Its components directed along the body x, y, and z axes are roughly weighted as follows:

$$\bar{\Omega}_{max} = \begin{bmatrix} 0.3\Omega_{max} & 0.3\Omega_{max} & 1.0\Omega_{max} \end{bmatrix}^T$$

where it is assumed that the primary direction of rotation is about the body z axis.

1. As an idea for dealing with drift, it may be possible to add fake measurements of zero attitude with high uncertainty. This may have the useful side effect of enforcing level indications over the long term while still allowing the attitude sensors to capture dynamics. This idea is used in inertial systems which slave the gyros to the accelerometers because they indicate the direction of gravity.

8.2.4 Linear Velocity States

In the case of linear velocity, there are no acceleration states which propagate it forward in time via the transition matrix, so it will not move if its uncertainty is set to zero. Again using the remainder theorem:

$$\sigma_V = a_{max}\Delta t$$

$$\sigma_{total}(t) = \sqrt{n}\sigma_i = \frac{a_{max}\Delta t\sqrt{t/(\Delta t)}}{2} = \frac{a_{max}\Delta t\sqrt{t/(\Delta t)}}{2}$$

This gives 10 meters of error for an hour of operation at 10 Hz cycling rate using 1 g.

8.2.5 Angular Velocity States

Following the technique, an estimate of the maximum angular acceleration of the vehicle is needed in order to estimate the truncation error. By Euler's equation, this is the ratio of the applied torque to the moment of inertia, assuming a diagonal inertia dyadic. Numbers for these quantities are hard to come by. One way to get a reasonable number is to assume that the angular velocity cannot change any faster than zero to the maximum in less than some number of seconds, which amounts to an assumption about the magnitude of the disturbance loads. This would give:

$$\alpha_{max} = \Omega_{max}/\tau$$

where τ is an adjustable "time constant". Then the angular velocity state is uncertain at:

$$\sigma_{\dot{\beta}} = \alpha_{max}\Delta t$$

8.3 Measurement Uncertainty

The measurement uncertainties are far more critical to the filter operation, because, after all, the whole system is considered to fail if sensors are lost for only a few seconds and filter optimality is a nonissue. For sensors models, some of the error sources to be estimated include:

- bias and scale instability
- neglected sensor dynamics
- noise, vibration, backlash, EMI, and compliance, etc.

This section will model all sources of error as if they were random. A better form of model would augment the state vector to ***include bias and scale factor states*** for some of these sensors to allow the filter to tune itself.

8.3.1 Encoder

The random error in the encoder measurement can be estimated by driving the vehicle at constant velocity and plotting the encoder output. Since the vehicle is a giant filter on speed inputs, any vibrations or electrical noise will manifest itself in such an experiment.

However, there are far more important systematic sources. It is expected that scale errors will predominate because the wheel radius actually varies with time and temperature, compliance has been an important historical issue, and there is of course, longitudinal wheel slip. Since the assumed error is a scale factor error, it is modelled as a fraction of the measurement itself. This has the property that the uncertainty is correctly increased when the time step increases.

A final issue with encoders is that, although the clock is considered perfect, the software using it may not be. Specifically, encoder readings must be time stamped or read synchronously because, otherwise the velocity computed will be in error by the degree to which the filter cycle time is asynchronous[1]. In a non real time testing scenario, this error can be several hundred percent, so it must be addressed. With these caveats, the uncertainty model is:

$$\boxed{\sigma_{enc} = SFE_{enc}V_{enc}}$$

A number of about 5% of total distance travelled is appropriate from experience for the position states, but the issue here is the error in the encoder estimate of the forward component of velocity, so something less than 5% is appropriate unless there are significant errors beyond scale errors.

8.3.2 Doppler

Without more information about the doppler device, little can be said about the nature of its systematic error sources after calibration. There is likely a velocity dependence, and hence a scale error, and it would be surprising if it performs significantly better than a well calibrated encoder on pavement. Hence, tentatively set:

$$\boxed{\sigma_{dop} = SFE_{dop}V_{dop}}$$

1. The encoder is the only sensor for which this is necessary. The dt used by it must be the real dt between when the sensor was physically read. For the state equations, it is appropriate to read the clock just before execution and subtract it from last time around to get the dt.

8.3.3 Attitude

All of the attitude sensors will be considered as a unit. It is difficult to quantify the attitude uncertainty without more knowledge of the components inside. It is expected that considerable systematic error will be present which will have complicated dynamics, so an output noise test may not be very meaningful. Barring any information, the manufacturer specifications can be interpreted conservatively and used to determine a constant uncertainty.

In the absence of information, the best that can be done is to assume some constant uncertainty. However, it is common to find that yaw uncertainty is worse than that of the other two angles, because the former is often based on the gravity vector, and the latter on the weak and unreliable local magnetic field so tentatively set:

$$\boxed{\sigma_\theta = \sigma_\phi = \sigma_{ATT} \qquad \sigma_\psi = 2\sigma_{ATT}}$$

Some well known issues for attitude sensors are listed below for consideration in the estimates.

8.3.3.1 Accelerometers and Inclinometers

Both of these devices basically function by measuring the deflection of a mass attached to a calibrated restraint where the indicated degree of freedom is either linear or rotary. As a matter of basic physics[1], no instrument can instantaneously distinguish acceleration from gravity.

Inclinometers will measure vehicle linear acceleration as well as the direction of gravity, so their dynamics may have to be considered depending on their transfer function. Oftentimes, inclinometers intended for use on vehicles are constructed as low pass filters for this reason. The filtering of inclinometer outputs has the side effect of filtering vehicle attitude dynamics from the inclinometer signal as well. Based upon the time constant of the inclinometer filter, a Taylor remainder analysis may be appropriate.

Accelerometer measurements of acceleration are corrupted by gravity and inertial forces due to the earth's rotation if they are low pass devices. The technology of accounting for this leads to the inertial navigation system before the accounting is complete.

It is possible to account for the corruption of attitude by centrifugal acceleration in inclinometers by changing the measurement relationship to reflect that the measurement is the sum of either pitch and forward acceleration or roll and lateral acceleration.

1. Specifically, Einstein's principle of relativity implies that a man in an elevator in free-fall in a gravitational field can never know he is falling based on any measurement conducted in the elevator frame. Thus, accelerometers confuse gravity for acceleration, and inclinometers confuse acceleration for gravity.

8.3.3.2 Gyroscopes

The rate gyro is a nice complement to the inclinometer because it responds well to high frequency inputs. Gyro drift is the major source of error and it arises from very small disturbance torques generated on the device rotary bearings, whatever their nature, whether the devices are strapdown or not. Frequency response and drift models may be available from the manufacturer. Gyro drift rate is well modelled by a combination of random bias, exponentially correlated error, and random walk. It has also been suggested that a random ramp is necessary.

8.3.3.3 Compasses

The magnetic compass comes in various forms, but regardless of the form, there are three issues to account for. First, the **variation** of the local geomagnetic field and geographic north is a constant for small excursions that can be easily calibrated out[1]. Second, the term **deviation** is reserved for the effects of the vehicle residual magnetic field. The geomagnetic field is fixed in the earth frame whereas the vehicle field is fixed in the vehicle frame. Hence, the device measures the time varying sum of the two. A short Fourier series in vehicle heading can be constructed to calibrate this out against a reference by spinning the vehicle one full turn in azimuth.

Heeling error is the term for the error induced when the vehicle approaches vertical because the local field no longer projects on to the sensor sensitive axis. For fluxgates, heeling error is not an issue. However, fluxgates require nontrivial settling times, so they have the same filtering property on heading as clinometers do on pitch and roll. Therefore, a heading gyro is often necessary to augment a compass in even moderate angular velocity vehicles.

8.3.4 Steering

Steering wheel position can be measured with any number of simple transducers. It is a very low fidelity measurement that does not benefit from overly precise error characterization. Let:

$$\boxed{\sigma_{steer} = \text{constant}}$$

where the constant is determined from a linearity test.

1. The variation for the eastern seaboard of the US is about 15 degrees east of north. However, this is an outdoor assumption for an isolated compass. Any magnetic sources close to the vehicle will distort the local field. My thanks to R Coulter for educating me on compasses.

9. Aided AHRS Dead Reckoning

The pure dead reckoning filter of the previous section is unlikely to achieve an accuracy which exceeds a few percent of the distance travelled. This is because of the essential integration of errors in the process of dead reckoning. In particular, notice that none of the measurement matrices of the previous section have a nonzero element in the first three columns. That is, ***none of them measures the vehicle position*** - even indirectly. They all measure attitude or derivatives of position.

Whatever the fidelity of the measurements used in practical dead reckoning, a fix is needed at regular intervals to damp the DR and the mechanism for doing this is the subject of this section. The Kalman filter is an ideal formalism for integration of dead reckoning and position fixes because fixes are simply additional measurements which can be folded into the equations in like manner to the DR measurements.

9.1 Fixes in the Navigation Frame

The simplest form of position fix is a direct measurement of the vehicle position in the navigation frame. In practice, the **survey point** is the only such fix available because position indicating devices cannot usually be mounted at the center of the body frame. Here, the vehicle is positioned at a point which has been presurveyed in the nav frame. Once the filter is told that this is the case, it can use its stored knowledge of the true coordinates of the survey point to generate the fix. Survey points are useful for development as well as operational purposes since they can be used to calibrate the system models.

When generating survey points using a positioning device it is important to place it in exactly the same place on the vehicle each time it is used, and to account for its position explicitly.

A distinctive turn in the trajectory may constitute a survey point. Such **path features** can be used provided:

- there is some mechanism to ensure that the vehicle is actually on the stored trajectory (say, a human driver)
- there is some mechanism to recognize when the feature is encountered

The measurement matrix for survey points is trivial:

$$\bar{z} = \begin{bmatrix} x_{sp} & y_{sp} & z_{sp} \end{bmatrix}^T \qquad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

9.2 Fixes in a Positioner Frame

Sensors which can provide fixes on their own position include the GPS receiver[1] and the inertial navigation system. The measurement matrix for such a sensor is:

$$\boxed{\bar{z} = \underline{r}_p^n = T_b^n(\underline{x})\underline{r}_p^b = T_b^n(\underline{x})\begin{bmatrix} x_{gps} \\ y_{gps} \\ z_{gps} \end{bmatrix}}$$

Here, $\begin{bmatrix} x_{gps} & y_{gps} & z_{gps} \end{bmatrix}^T$ means the position of the positioner frame with respect to the body frame.

The transformation $T_b^n(\underline{x})$ then computes the positioner pose in the nav frame - which is what a GPS reciever observes

The Jacobian is

$$H_L^s = \frac{\partial}{\partial \underline{x}}(\underline{r}_p^n) = \frac{\partial}{\partial \underline{x}}(T_b^n(\underline{x}))\underline{r}_p^b$$

Estimation of GPS uncertainty is a difficult matter because it depends on such matters as the presence or absence of Selective Availability (the dominant source of error), various technological error sources such as receiver noise, dynamics, multipath, etc., and the satellite geometry. Ideally, a receiver would provide the covariance matrix in the navigation frame directly. Without such information, the only option is to assume a constant covariance which is relatively conservative.

Estimation of INS uncertainty is even a harder problem because of the Schuler dynamics involved. Again, a large number is better than no number. Technically, the INS is a DR device, and it does not actually generate a “fix”. The implication of this is that INS uncertainty models must have a time growth associated with them.

A later section provides the equations necessary to convert INS and GPS outputs into a common coordinate system. In practice, it is often possible to slave the DR output to the global positioning device. In this case, the whole issue can be avoided by simply using the first fix to initialize the position states into the appropriate nav frame.

1. In truth, the GPS receiver has a complicated measurement model which is four dimensional range triangulation. It is possible to ignore this if an estimate of uncertainty in the GPS output is available. Even the GDOP output of a receiver can be used somewhat. A later section will reveal the internal operation of a GPS filter implemented inside a GPS receiver.

10. Roadfollower Aiding

A roadfollowing system provides an observable which can be described as **relative crosstrack error.** It is measured relative to the path in view (the road), is measured more or less transverse to the path, and is measured in a visual feedforward sense (i.e it is the crosstrack error in front of the vehicle).

An EKF can accept any indirect measurement of state and since the road deviation in an image is dependent on the vehicle position and attitude, it provides an indirect measurement of state. Integration of a road follower into the EKF amounts to a continuous landmark observable that should provide an excellent damping signal for dead reckoning. Note however that this will not improve the performance of the roadfollower - it has direct visual feedback already, but it should improve the position estimate of the vehicle.

10.1 Measurement Model

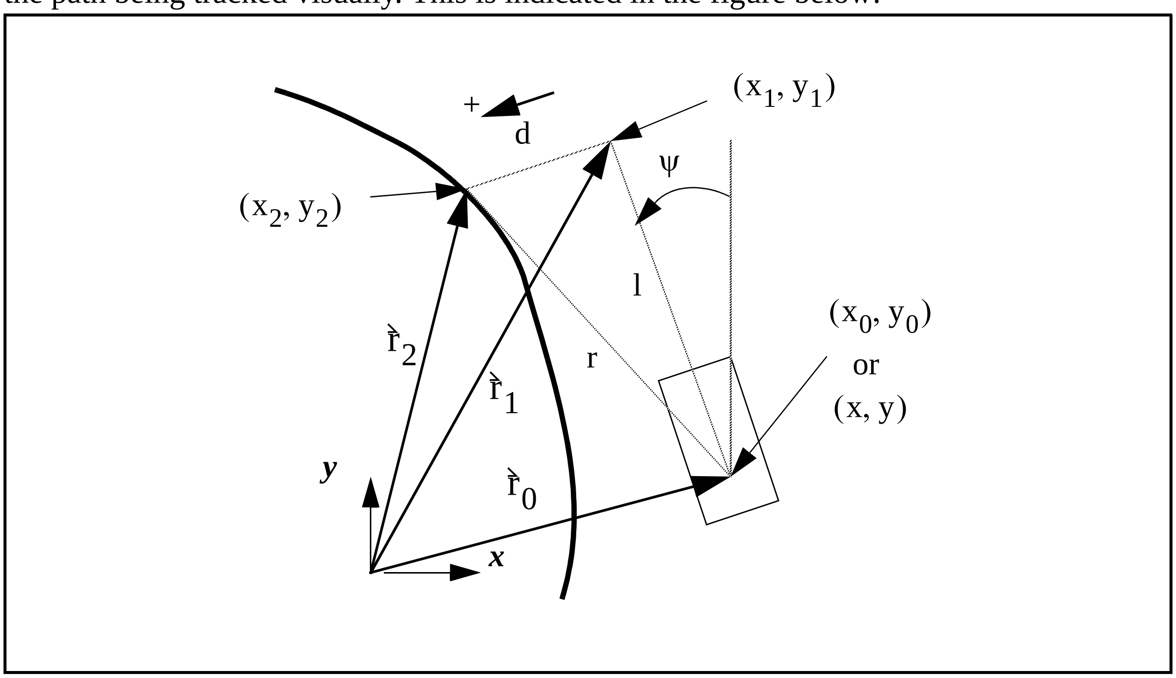
The measurement model must be formulated in an indirect sense - that is, the relationship between the path and the vehicle state which generates the observable is required.

10.1.1 Issues

In reality, the observable depends on the complete six axis pose of the vehicle, but because a typical roadfollower does not make such distinctions and typically does not store a 3D path, a 2D formulation will be used.

10.1.2 Observable

Let l be the **lookahead distance**, and let the roadfollower generate the crosstrack error d, at the lookahead distance that ***would be observed if the vehicle travelled in a straight line on its current heading***. Let point 1 be at the end of the lookahead vector and point 2 be the corresponding point on the path being tracked visually. This is indicated in the figure below:



Define the vectors:

$$\vec{r} = \vec{r}_{20} = \vec{r}_2 - \vec{r}_0 \qquad \vec{l} = \vec{r}_{10} = \vec{r}_1 - \vec{r}_0 \qquad \vec{d} = \vec{r}_{21} = \vec{r}_2 - \vec{r}_1$$

and the letters r, l and d will refer to the magnitudes of these vectors except that the latter will be a signed quantity. The vector $\vec{l}$ will be called the **lookahead vector**, $\vec{d}$ will be called the **crosstrack vector**, and $\vec{r}$ will be called **goal vector**. The observable is clearly:

$$h(\bar{x}) = d = \pm|\vec{r}_{21}| = \pm\sqrt{\vec{r}_{21} \bullet \vec{r}_{21}} = \pm\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

The crosstrack observable d will be defined to be positive ***when the path is to the left*** of the end of the lookahead vector as shown in the figure. This depends on the sign of the angle ***from*** the lookahead vector ***to*** the goal vector, it being positive when this angle is counterclockwise. This can be determined without using the arctangent by a vector cross product as follows:

$$\text{signof}(\vec{l} \times \vec{r}) = \text{signof}(l_x r_y - l_y r_x)$$

$$\text{signof}(\vec{l} \times \vec{r}) = \text{signof}[(x_1 - x)(y_2 - y) - (y_1 - y)(x_2 - x)]$$

The coordinates of point 1 depend on the vehicle state vector thus:

$$x_1 = x - l\,sin\psi$$

$$y_1 = y + l\,cos\psi$$

This gives the observable as an indirect measurement of vehicle position and orientation in 2D.

10.1.3 Measurement Jacobian

The measurement Jacobian is available through partial differentiation. The signs come out correctly with the chosen convention for the sign of the observable:

$$H_d = \begin{bmatrix} \frac{\partial d}{\partial x} & \frac{\partial d}{\partial y} & 0 & 0 & 0 & 0 & \frac{\partial d}{\partial \psi} & 0 \end{bmatrix}$$

$$\frac{\partial d}{\partial x} = \frac{x - l\,sin\psi - x_2}{d} = \frac{x_1 - x_2}{d} \qquad \frac{\partial d}{\partial y} = \frac{y + l\,cos\psi - y_2}{d} = \frac{y_1 - y_2}{d}$$

$$\frac{\partial d}{\partial \psi} = \frac{-l\,cos\psi[x - l\,sin\psi - x_2] - l\,sin\psi[y + l\,cos\psi - y_2]}{d}$$

$$\frac{\partial d}{\partial \psi} = \frac{-l\,cos\psi[x_1 - x_2] - l\,sin\psi[y_1 - y_2]}{d}$$

Notice that the denominator in all 3 elements of the gradient is just the crosstrack. When this is zero, the numerators are also zero. L'Hopital's rule could be used to resolve this case, but physically, the observable is irrelevant when the residual is zero anyway as shown in the next section, so a practical approach is to simply avoid processing the crosstrack measurement when it is very small.

10.2 Predictive Measurement

Recall the state update equation of the EKF:

$$\hat{x}_k = \hat{x}_k^- + K_k[z_k - h(\hat{x}_k^-)]$$

This equation runs in the filter when a measurement arrives. It amounts to the formation of a measurement residual - the difference between what is observed z_k and what was expected based on the current state estimate $h(\hat{x}_k^-)$.

In order to generate the prediction, the system must:

- know the coordinates of every point on the path in the navigation frame
- be able to search the path forward to generate the crosstrack prediction

Let $\hat{u}$ be a unit vector oriented along the lookahead vector:

$$u = \begin{bmatrix} -sin\psi & cos\psi \end{bmatrix}$$

and then the lookahead distance for any point (x_2, y_2) on the path is:

$$l = \vec{r} \bullet \hat{u} = (x_2 - x)(-sin\psi) + (y_2 - y)(cos\psi)$$

The predictive measurement can be obtained by searching forward from a point on the path near the vehicle until the above expression is equal to the required lookahead distance. Then the formula for d gives the predictive measurement.

10.3 Uncertainty Model

The uncertainty model must account for the random error in the crosstrack observable. Issues to consider are the video camera resolution, the error in mapping images to observables, any delays involved etc. The error is clearly very sensitive to the curvature of the path because a small change in the vehicle position generates a very large deviation in the crosstrack - this is particularly true of the heading state. The measurement jacobian accounts for the sensitivity of the observable to the ***state*** but not to the ***path***.

A simple way to account for path curvature is to assume that the random component of crosstrack increases as the crosstrack itself does. That is, large observables indicate high path curvature and therefore high measurement error. As a first estimate then, an uncertainty model is:

$$\boxed{\sigma_d = SFE \times d}$$

where a scale factor of perhaps 10% is reasonable. A power of the crosstrack error may be even more appropriate.

Another issue to be accounted for is the resolution of the stored path and the mechanism for interpolation used in the path tracker. It will be assumed that the path points are either very dense or that the tracker interpolates appropriately so that ***the predictive measurement coincides precisely with the lookahead distance***.

A better uncertainty estimate accounts for the roadfollower's own estimate of confidence. Let c be a number from 0 to 1 where 0 indicates complete lack of confidence, and 1 indicates full confidence. It is also reasonable to allow the uncertainty to increase with the lookahead distance. A model which accounts for this is:

$$\boxed{\sigma_d = \frac{k \times l}{c^n}}$$

where k is a constant, l is the lookahead, c is the confidence, and n is an exponent. This model could also be modified to include a term for the crosstrack.

10.4 Interface

Searching forward on a path is a function that is typically already implemented in a feedforward path tracker. Further, it is desirable to insulate the EKF from the need to manage a path data structure and the need to get path information into and out of the EKF. For this reason, it is desirable to provide two interface functions which connect the EKF to both the road follower and the path tracker in order to simplify and reduce duplication of effort in interfaces.

The first function provides the crosstrack observable. Its specification is:

```
int get_roadfollower_measurement(crosstrack, stdv)
double *crosstrack;
double *stdv
{
}
```

The function returns the crosstrack computed from the current image through its argument. An uncertainty model is used to return an estimate of the standard deviation of crosstrack as well. It returns a 1 or zero through its name to signify success or failure.

The second function returns the predictive measurement from the path tracker. The specification is:

```
int get_tracker_prediction(crosstrack,state,gradient)
double *crosstrack;
double state[8];
double gradient[8];
{
}
```

The lookahead distance will be hidden in the path tracker so that the EKF needs no knowledge of it. The function returns both the predictive measurement and the measurement gradient vector. The state is an input argument provided to ensure that the EKF and the predictive measurement are based on the exact same state estimate.

10.5 Advantages

The primary advantage of the roadfollower observable is that it constitutes the only landmark observable in the absence of GPS or visual landmarks. This element of the Kalman filter will force the state estimate to track the known or recorded position of the path

11. Terrain Aided AHRS Dead Reckoning

Vehicle perception sensors provide many mechanisms for generating a fix. Landmark recognition can be folded into the filter, but the measurement model is nontrivial because it must account for the imaging process. This section presents a general model of the perception measurement process which is applicable to many different scenarios.

11.1 Terrain Aids

Terrain aiding schemes can be distinguished and classified along several independent dimensions:

11.1.1 Image Dimension

The image dimension amounts to the size of the measurement vector. Some options include:

- 3D scanning laser rangefinder
- 2D scanning laser rangefinder
- 3D stereo range images
- 2D color images
- 2D intensity images

An important distinction of the last two is that their imaging transform is not invertible. In this case, it is not possible to use a single image to convert the image coordinates of a landmark into its coordinates in the nav frame.

11.1.2 Image Information Density

Schemes can also be classified based on the amount of information extracted from a single image.

- **correlation** schemes match pieces of images in order to generate a single vector measurement
- **feature based** schemes match a small number of “interesting” points to generate a small number of vector measurements
- **iconic** schemes match every pixel in the image in order to generate a large number of vector measurements

11.1.3 Absolute and Differential Observations

Schemes can be classified based upon whether landmarks are considered known in the nav frame or whether they are considered unknown in the nav frame and observed to change from image to image.

- **absolute observations** match observations against the known position of a “landmark” in the nav frame
- **differential observations** match observations in the current image against either their position in the last image or their predicted position in the current image

11.2 Measurement Uncertainty

Uncertainty models for perception sensors can be generated from knowledge of the physics of the operation of the sensor and the conditioning of the embedded computations, if any. Measurement uncertainty is modelled in the R_k matrix. Some specific cases are:

11.2.1 AM Laser Rangefinders

These devices have a well known variation in range accuracy which varies directly with the square of the range and inversely with the angle of incidence and the reflectance of the surface. Angular uncertainty can be generated as at least the width of a pixel but may also include an accounting for the fidelity of the mirror drive control laws.

11.2.2 TOF Laser Rangefinders

For time of flight devices, variation with range is likely a function of the range quantization and more or less constant. Variation with reflectance and angle of incidence can be accounted for by considering the operation of the thresholding electronics. Similiar comments about angular uncertainty apply.

11.2.3 Stereo Vision Rangefinders

Stereo uncertainty can be generated from a knowledge of the particular algorithm used, the baseline, and the sensory hardware. It can sometimes be produced as a by-product of the stereo algorithm.

11.2.4 CCD Images

For CCD images, a model of the transformation between intensity noise and the noise on the measured angle to a feature is required. The basic issue is the fidelity of localization of the landmark in the image. Accuracies larger or considerably smaller than a pixel are possible depending on the feature extraction algorithm used.

12. Absolute Landmark Recognition

This application is distinguished by the true knowledge of the landmark position in the nav frame.

12.1 Absolute Landmarks

Absolute landmarks can be distinguished by the interpretation of the predictive measurement $h(\hat{x}_k^-)$ in the state update equation. In this case, the predictive measurement arises from "simulating" the generation of the landmark position in the image according to the measurement model evaluated at the current position estimate and the known landmark position.

12.2 Feature Based Scheme

In a feature based scheme, an interest operator identifies interesting points in an image and a mechanism is provided which constructs candidate matches of landmarks against their predicted positions in the image.

Let the position of a landmark in the navigation frame be known to be:

$$\bar{r}_L^n = \begin{bmatrix} x_L & y_L & z_L \end{bmatrix}^T$$

Consider a generalized perception sensor which generates a 3D image of which most real sensors are special cases. Such a sensor can be modelled as generating the landmark position somewhere in the image through the image formation process expressed in terms of a concatenation of transformations through intermediate frames. The transformation from the navigation to the sensor frame is:

$$\bar{r}_L^s = \begin{bmatrix} x_L^s \\ y_L^s \\ z_L^s \end{bmatrix} = T_b^s T_n^b(\bar{x}) \begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix} = T_b^s T_n^b(\bar{x}) \bar{r}_L^n$$

Where $T_n^b(\bar{x})$ is the nav frame to body frame homogeneous transform, T_b^s is the body frame to sensor frame homogeneous transform. The measurement matrix for the transform is evaluated as the product of the constant body to sensor transform, the Jacobian tensor of the nav to body transform, and the landmark position vector.

$$H_L^s = \frac{\partial}{\partial \bar{x}}(\bar{r}_L^s) = T_b^s \frac{\partial}{\partial \bar{x}}(T_n^b(\bar{x}))\bar{r}_L^n$$

Now let a generalized nonlinear imaging function map a point in the sensor frame into image coordinates.

$$\bar{r}_L^i = f(\bar{r}_L^s) = f(T_b^s T_n^b(\bar{x})\bar{r}_L^n)$$

where $\bar{r}_L^i$ is the triple (range, azimuth, elevation) for a rangefinder or the pair (row, column) for a video camera. The complete measurement Jacobian is then:

$$H_L = \left(\frac{\partial \bar{r}_L^i}{\partial \bar{r}_L^s}\right)\left(\frac{\partial}{\partial \bar{x}}(\bar{r}_L^s)\right) = H_s^i H_L^s = H_s^i T_b^s \frac{\partial}{\partial \bar{x}}(T_n^b(\bar{x}))\bar{r}_L^n$$

This is the general case for any sensor. Recall that the measurement Jacobian provides the information necessary to project the residual onto the state vector. The measurement uncertainty itself arises in the R_k matrix. So the analysis so far has nothing to do with the sensor itself. Rather, it answers the question of how an error in vehicle position relates to an error in the position of a landmark in the image for a perfect sensor.

This will be called the **landmark Jacobian**. In order to use this formula for any particular sensor, the imaging Jacobian must be substituted for the particular sensor from the previous section on kinematics.

The matrix partial is a tensor. Let it be 4 X 4 X n. Thus, its second index is matched with the row index of the landmark position vector to generate the 4 X n matrix:

$$H_{i,k} = \left[\frac{\partial T}{\partial \bar{x}}\right]_{i,j,k} \bar{r}_j \qquad \begin{matrix} i=1,4 \\ j=1,4 \\ k=1,n \end{matrix}$$

Notice that the fact that the filter is using the difference between the projected position of the landmark and the actual position of the landmark in the image is not explicit. The operation of the filter is such that it automatically computes what the differential change in the state vector has to be in order for the observed measurement to be made.

12.3 Uncertain Absolute Landmarks

The above treatment of landmark recognition assumes that the landmark position is known precisely in the navigation frame. However, in some cases, the landmark has its own uncertainty. In particular, an observation of another moving vehicle will have to account for the uncertainty of the other vehicle's position estimate. Another case is when the surveying sensor is suspected of being sufficiently inaccurate to warrant modelling.

This can be managed in the general formulation because the R_k matrix exists to quantify measurement uncertainty. However, there is some difficulty involved because the R_k matrix is expressed in image coordinates. Recall that the transformation of the landmark position into the image plane is given by:

$$\bar{r}_L^i = f(\bar{r}_L^s) = f(T_b^s T_n^b(\bar{x})\bar{r}_L^n)$$

Thus, the uncertainty in the image plane arising from an uncertain landmark can be computed as

$$\Delta \bar{r}_L^i = \frac{\partial}{\partial \bar{r}_L^n}[f(T_b^s T_n^b(\bar{x})\bar{r}_L^n)]\Delta \bar{r}_L^n$$

$$\Delta \bar{r}_L^i = H_s^i T_b^s T_n^b(\bar{x})\Delta \bar{r}_L^n$$

where the chain rule has been used and it was noticed that the argument to the imaging function is linear in the landmark position in the nav frame. Now from the expectation operator:

$$E[(\Delta \bar{r}_L^i)(\Delta \bar{r}_L^i)^T] = E[(H_s^i T_b^s T_n^b(\bar{x})\Delta \bar{r}_L^n)(H_s^i T_b^s T_n^b(\bar{x})\Delta \bar{r}_L^n)^T]$$

$$Cov[\Delta \bar{r}_L^i] = H_s^i T_b^s T_n^b(\bar{x}) Cov[\Delta \bar{r}_L^n](H_s^i T_b^s T_n^b(\bar{x}))^T$$

So that if C_L is the covariance of the landmark expressed in the navigation frame, then the R_k matrix for the measurement is:

$$R_k = H C_L H^T$$

$$H = H_s^i T_b^s T_n^b(\bar{x})$$

This analysis produces the contribution of landmark uncertainty to the R_k matrix. The effect of sensor uncertainty must be (literally) added to this, and the effect of vehicle position uncertainty is already accounted for in the measurement Jacobian.

In this formulation of the uncertain landmark, the landmark position is not considered part of the state vector and, hence, is not updated. It is possible to augment the state vector to include the landmark position. This will be discussed in a later section.

12.4 Iconic Scheme

In an iconic absolute landmark scheme, the “landmarks” are really a continuous geometric description of all or part of the environment which is known a priori. In 2D a line segment world description or a collection of occupancy points suffices. In the latter case, and in general, it may be necessary to interpolate a sparse a priori model of world geometry in order to generate the predictive measurements. The predictive measurements are generated by “simulating” the image that would be expected if evaluated at the current position estimate and the known world geometry model.

Such a scheme implemented in 3D amounts to a least squares continuous match of the entire terrain map from a single image against the known continuous world model.

12.5 Examples of Absolute Aiding

Many special cases of absolute aiding can be distinguished.

12.5.1 3D Landmark Recognition With a Rangefinder

If a rangefinder is used, the rangefinder imaging Jacobian is used. In this case, the sensor to image transform is spherical polar and the measurement is 3D, so it is 3 X n.

12.5.2 2D Landmark Recognition with a Video Camera

If a camera is used, the camera imaging Jacobian applies. In this case, the sensor to image transform is a perspective projection and the measurement is 2D, so it is 2 X n.

12.5.3 Video Crosstrack Observations

A degenerate example of the above landmark recognition is the use of the crosstrack error observable generated by a road-follower. This case was presented earlier in detail.

12.5.4 Convoy Image Sequences

If sufficient radio bandwidth is available, following vehicles in a convoy can maintain a kind of **video lock** by computing the position of the correlation peak over a small window between the current image and the image generated by the front vehicle when it was close to the current position. Relative crosstrack and alongtrack error should be easy to compute in this case [1].

12.5.5 Vehicle Convoying Using the Leader as a Landmark

When two or more vehicles operate in a convoy, a perception system which can identify the vehicle in front of it can consider it to be a landmark provided the front vehicle position is known[2]. This is a special case of the 3D landmark. Even though it is moving, there is no mathematical difference. Note however that a single point on the front vehicle must be chosen as the landmark for optimal measurement. Also, if the leader position is uncertain its uncertainty must be estimated and accounted for or it can be taken as ground truth in a relative navigation mode.[3]

12.5.6 2D Point and Line Feature Matching

For indoor 2D applications, it is possible to easily fit into the formalism, say, matching a wall of known shape, against the current range image from a single line rangefinder. Based on the current position estimate and a known map of the environment, every pixel in a single line rangefinder constitutes a measurement whose residual can be used to update the position estimate. For 2D problems, the state vector is only 3 X 1 and no matrix inversion is required.

---

1. Both of the previous scenarios will benefit if only the lower image pixels are used because the parallax observable will be largest and more useful in this region of the image. Of course, the entire image will not correlate anyway because of the perspective geometry of the imaging process. If they did, stereo would never work.
2. This can be achieved with a radio link which passes each vehicle’s position backward down the link at regular frequency.
3. In any distributed implementation of a Kalman filter, the delays associated with communication will have to be taken into account.

13. Differential Landmark Recognition

Differential aids have a very close connection to **mapping**[1] applications. The distinction is based on whether the vehicle position or the landmark position (or both) is considered unknown. That is, whether one or the other or both appear in the state vector. It is possible to consider both unknown if the uncertainties are treated correctly and the state vector is augmented. This is considered in the next section.

In differential applications, if the vehicle position estimate is updated ***before*** the image is transformed into a map section, then ***mapping is performed simultaneously*** with position estimation. After all, the filter is attempting to minimize the residual in order to produce the optimal position estimate, but in this case, the residual is the map mismatch, so it is directly minimized. This can only be done, of course, when an invertible sensor transform is used, but this is necessary to form the terrain map anyway, so it is not an extra requirement.

13.1 Differential Landmarks

When landmarks are observed differentially, the predictive measurement is used to generate a residual between ***where it should appear*** based on the last time it was seen, and ***where it does appear*** in the current image. Under the measurement models for sensors used here, the residual is formed in image space. Nav frame schemes are possible too, but they are a mathematical equivalent where the transforms in the measurement models are just moved around from the H_k matrix to the R_k matrix.

Again, iconic and feature based schemes are possible and these can be distinguished by the interpretation of the predictive measurement $h(\hat{x}_k^-)$ in the state update equation:

$$\boxed{\hat{x}_k = \hat{x}_k^- + K_k[z_k - h(\hat{x}_k^-)] = \hat{x}_k^- + K_k[z_k - \hat{z}_k^-]}$$

13.2 Feature Based Scheme

Here, discrete features are used and matched from image to image. This is done by tracking the nav frame positions of the landmarks and transforming them into image space as needed.

After deduced reckoning has supplied a position estimate, the predictive measurement can be simply generated through the measurement model evaluated at the current position estimate:

$$\boxed{\hat{z}_k^- = \bar{r}_L^i\Big|_k = f\left(T_b^s T_n^b(\bar{x}_k)\bar{r}_L^n\Big|_{k-1}\right)}$$

13.3 Iconic Scheme

Iconic schemes ***require*** a map unless images are acquired fast enough that flow information is high enough in fidelity to justify the linear assumption of the filter. When a map is available, the predictive measurement for every pixel in the image is generated by "simulating" what the sensor should see in the current position given by the current position estimate. This simulated image is generated from the evolving map as it exists before the current image is incorporated.

Thus, considering the computed point of intersection of the image ray with the terrain map to

1. The ***map matching*** of local terrain map navigation is a trivial kind of mapping.

constitute the "landmark" $\hat{\bar{r}}_L^n$, the measurement model is:

$$\boxed{\hat{z}_k = \bar{r}_L^i = f(\bar{r}_L^s) = f(T_b^s T_n^b(\bar{x})\hat{\bar{r}}_L^n)}$$

where the notation suggests that the landmark position is predicted.

Complete iconic schemes are probably only feasible in 2D worlds with 2D sensors because the simulation is expensive computationally. However, it is also possible to undersample the image to reduce the load in the 3D case.

13.4 Examples of Differential Aids

13.4.1 3D Terrain Aided Navigation

In a manner very similiar to video lock, a rangefinder navigation system can use the mismatch difference between two sequential terrain maps in order to update the vehicle position estimate. If the range images are close together in time, the computation may be poorly conditioned, so it is probably best to compute the mismatch over the largest excursion for which two images still overlap. In this scenario, the position of a feature from the last image forms the landmark position and the position in the current image forms the measurement. It would be very computationally intensive to process every range pixel in this manner, so a feature extraction process will be a practical necessity.

13.4.2 2D Point and Line Feature Matching

For indoor 2D applications, the evolving map may be a set of geometric primitives such as lines or simply a set of occupancy points. When the map is interpolated, a mechanism for simulating the image expected is available. Based on the current position estimate and the evolving map of the environment, every pixel in a single line rangefinder constitutes a measurement whose residual can be used to update the position estimate.

14. Simultaneous Mapping and Position Estimation

With an augmented state vector, it is possible to "distort" an evolving map in order to produce a better estimate of both the vehicle position and the environmental map. This is achieved, of course, with state vector augmentation.

14.1 State Vector Augmentation

The augmented state vector includes the vehicle position estimate as well as the position of the landmark in the nav frame:

$$\bar{x}_n = \begin{bmatrix} x & y & z & V & \theta & \phi & \psi & \dot{\beta} \end{bmatrix}^T \qquad \bar{x}_L = \begin{bmatrix} r_x^n & r_y^n & r_z^n \end{bmatrix}^T$$

The system model then becomes:

$$\frac{d}{dt}\begin{bmatrix} \bar{x}_n \\ \bar{x}_L \end{bmatrix} = \begin{bmatrix} f(\bar{x}) \\ 0 \end{bmatrix}$$

14.2 Measurement Model

The measurement model now generates a better reflection of the uncertainties in both subvectors of the state. Now that the landmark position is part of the state vector, the measurement model becomes:

$$\bar{r}_L^i = f(\bar{r}_L^s) = f(T_b^s T_n^b(\bar{x}_n)\bar{x}_L)$$

14.3 Measurement Jacobian

The measurement Jacobian is available by **stacking** the Jacobians of the state subvectors:

$$\Delta\bar{r}_L^i = H_s^i \Delta\bar{r}_L^s$$

$$\Delta\bar{r}_L^s = T_b^s\left(\frac{\partial}{\partial\bar{x}_n}T_n^b(\bar{x}_n)\bar{x}_L\Delta\bar{x}_n + T_n^b(\bar{x}_n)\Delta\bar{x}_L\right)$$

$$H_L = H_s^i T_b^s \begin{bmatrix} \frac{\partial}{\partial\bar{x}_n}T_n^b(\bar{x}_n)\bar{x}_L & 0 \\ 0 & T_n^b(\bar{x}_n) \end{bmatrix}$$

where the usual dubious assumption of uncorrelated measurement error has been made. This formulation will both estimate vehicle position and produce an optimal map. The extension to differential observations is straightforward.

15. Real Time Identification

The filter formulation can be modified to recover important system calibration constants using state vector augmentation techniques. Consider, for example, that the most important calibration constants in a rangefinder perception system are:

- the height c_b^s of the sensor
- the tilt angle Θ_b^s of the sensor
- the range image bias R_{bias}
- the range image scale factor K_{range}

This section is the clearest example of why the tensor notation has been adopted. The measurement models are very involved and it would be hopeless to attempt to formulate them without the compactness afforded by the tensor notation.

15.1 State Vector Augmentation

The state vector can be augmented to include these variables. Formulating in general, let the additional state variables be denoted:

$$\bar{x}_n = \begin{bmatrix} x & y & z & V & \theta & \phi & \psi & \dot{\beta} \end{bmatrix}^T \quad \bar{x}_b = \begin{bmatrix} c_b^s & \Theta_b^s \end{bmatrix}^T \qquad \bar{x}_s = \begin{bmatrix} R_{bias} & K_{range} \end{bmatrix}^T$$

where the notation suggests parameters of a particular transformation, and introduce them into the system model through the additional differential equations:

$$\frac{d}{dt}\begin{bmatrix} \bar{x}_b \\ \bar{x}_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Then the system model has been appropriately modified. The only measurements which involve these new states are the landmarks, so they can only be observed with landmark aided navigation.

15.2 Measurement Model

In the simplest formulation of the problem, consider that landmarks are known in the navigation frame and they generate point features in a range image.

An interest operator is used to identify features in a range image, and for each feature, the image position forms the z_k observation vector. The predictive measurement $h(\bar{x})$, where the augmented state vector is used, computes the predicted position of the features in each image based on their known positions in the navigation frame.

$$\bar{r}_L^i = f(\bar{r}_L^s, \bar{x}_s) = f(T_b^s(\bar{x}_b)T_n^b(\bar{x}_n)\bar{r}_L^n, \bar{x}_s)$$

15.3 Measurement Jacobian

The measurement Jacobian is available from the matrix interpretation of the chain rule and the mechanism of **stacking**. This formulation will require modifications to the imaging Jacobian to include differentials with respect to the parameters $\bar{x}_s$. It also requires the formulation of the Jacobian tensor of the body to sensor transform which is straightforward.

Consider that the expression for the landmark position in the image is a nonlinear function of three vector variables. Then the total derivative is given by the chain rule

$$\Delta\bar{r}_L^i = H_s^i(\bar{x}_s)\Delta\bar{r}_L^s + \frac{\partial f}{\partial \bar{x}_s}\Delta\bar{x}_s$$

$$\Delta\bar{r}_L^s = \frac{\partial}{\partial \bar{x}_b}T_b^s(\bar{x}_b)T_n^b(\bar{x}_n)\bar{r}_L^n\Delta\bar{x}_b + T_b^s(\bar{x}_b)\frac{\partial}{\partial \bar{x}_n}T_n^b(\bar{x}_n)\bar{r}_L^n\Delta\bar{x}_n$$

$$H = \begin{bmatrix} \frac{\partial f}{\partial \bar{x}_s} & 0 \\ 0 & H_s^i(\bar{x}_s)H_1 \end{bmatrix}$$

$$H_1 = \begin{bmatrix} H_s^i(\bar{x}_s)\frac{\partial}{\partial \bar{x}_b}T_b^s(\bar{x}_b)T_n^b(\bar{x}_n)\bar{r}_L^n & 0 \\ 0 & H_s^i(\bar{x}_s)T_b^s(\bar{x}_b)\frac{\partial}{\partial \bar{x}_n}T_n^b(\bar{x}_n)\bar{r}_L^n \end{bmatrix}$$

This formulation will both estimate vehicle position and calibrate the system kinematic models. The extension to differential observations is straightforward.

16. Pragmatics

The Kalman filter is a is a mechanism that accounts for random errors in such a way as to generate an optimal estimate. If nonrandom errors exist in the model or the measurements, then the filter cannot do much about them unless it is told about them. Further, the estimate is only as good as the uncertainty estimates that are given to the filter. The mathematical ideal of the zero mean white sequence noise source must be stretched significantly in practice.

16.1 Calibration and Tuning

Once the filter equations are implemented, the remaining practical issues include:

- the calibration of systematic errors
- the characterization of remaining error sources, ***as if they were random***, in terms of their first order statistics

A real filter ***must be tuned*** very precisely for "optimal performance", and it may not work at all if this is not done properly. This following section outlines some techniques used to achieve these goals.

Just as Kalman filtering is an estimation process, calibration of a sensor is an estimation process in its own right. The rules for calibration have not changed since Gauss first wrote them when he invented least squares in 1795 [6]. Calibration of a sensor requires:

- a **reference** which is more accurate than the goal accuracy of the calibrated sensor
- a **measurement model** for the sensor
- multiple **observations** of the reference and the raw sensor output over the domain of operation of the sensor in an operational setting

In practice, the reference is nonexistent, is not accurate enough, or is not regarded as trustworthy; the measurement model is a crude approximation, omits important variables such as temperature, compliance, vibration, electromagnetic interference, etc., or is not understood at all; and the observations are made unscientifically in an artificial setting. Conceptually, the process is one of fitting a curve to data where the raw sensor indication is the abscissa and the reference output is the ordinate. When the best fit model is then applied to the raw sensor output, it provides the best estimate of the true value of the variable of interest based solely on the raw output after the reference is removed.

The significance of calibration to Kalman filtering is that zero mean noise sources are assumed, and any bias in the distribution will result in suboptimality. In practical terms, most sensors in use are nominally linear, and the measurement model is a straight line. For such a model, the calibration problem becomes one of estimating **bias** and **scale factor**, which are just the y intercept and the slope of the measurement model. Any remaining misfit in the model can be regarded, in practice, as uncertainty to be given to the Kalman filter.

On occasion, the model parameters are known to have stability problems (i.e. to vary with unmodelled parameters such as time), and adaptive techniques can be used to estimate them in real time. Constant bias occurs in several sensors in the suite. The compass for instance may not be calibrated for the local region. Gyros may exhibit slight biases which may or may not be constant. When **bias stability** is known or suspected to be a problem, the following techniques are used.

16.1.1 Relative Navigation

Relative navigation avoids the bias issue completely because it cancels out in the mathematics of the application. For example, bias in the compass heading can be ignored if no measurements of landmarks are referenced to a geocentric frame of reference. This is possible since high pass gyros will not require knowledge of the transform to the geocentric earth frame, and thus, vehicle azimuth can be treated as a relative quantity. The east and north directions are arbitrary. In inertial navigation systems, this idea is called **wander azimuth** but it is done for different reasons. However, if any device such as GPS is used which reports position in a geographic frame, the details of the alignment to that frame will have to be addressed. Differential GPS is another form of relative navigation, as is traditional encoder dead reckoning when the start point is taken at zero.

16.1.2 Zups

Another type of bias removal is the zero velocity update, or **zup**. For terrestrial vehicles, sustained speeds below a few inches per second are uncommon enough to be considered to result from sensor biases and not vehicle motion. The zup can be engaged after sufficiently long periods of very slow velocities are observed coming from the sensors. Attitude zups are possible too, for if the velocity is zero, it is likely that the attitude is not changing either.

Zups are useful and sometimes necessary for other reasons. Specifically, when the Q matrix is not zero, the P matrix grows in magnitude with each iteration of the system model, and this may be unrealistic and even harmful to the operation of the filter. Once the filter enters zup mode, this can be checked until motion begins again. Zups must be triggered by the measurements and not the state vector. Otherwise, there is no simple way to exit zup mode.

16.1.3 State Vector Augmentation

Another mechanism for bias removal, state vector augmentation, was discussed in an earlier section. Analogous techniques are available for adaptive measurement of scale factor.

16.2 Initialization

The filter equations make no distinction between the first cycle and any other, so any choice of the initial state vector and its uncertainty can be made ***provided it is consistent***. Two general cases are possible and they can be mixed across the elements of the state vector and covariance matrix.

16.2.1 Relative Navigation

Set the initial position to zero and its uncertainty to zero. Provided the appropriate elements of Q_k are nonzero, the state estimate will evolve normally in time as if the initial position was known perfectly. This technique applies mostly to linear position states, and since it requires that landmarks be known with respect to the initial position, it may be difficult to use with any landmark observations unless they are observed differentially.

16.2.2 Absolute Navigation

Set the initial state value to zero and its uncertainty to a large number. The first measurement of that state will naturally replace the initial value when it is incorporated into the estimate. Note however, that the filter will run open loop with respect to that state until a measurement is received. Depending on the time interval involved, this may or may not be a problem.

16.3 Attitude Measurement

It can be shown that even very small systematic error in the attitude indications can ultimately cause enormous error in the filter output position. This is effectively because these errors act forever. It is very important to remove systematic attitude error through calibration techniques.

16.4 Reasonableness Checks

Notice that the Kalman state update equation, and the system model both constitute places where angles are added. After each iteration of these equations, and during the formation of the measurement residual, the angular states ***must*** be forced into some canonical sector of the unit circle such as $-\pi$ to π. If this is not done, the measurement residuals will begin to, apparently randomly, overestimate the mismatch by a full revolution and the state vector will rotate one revolution over a few cycles. This is usually quite a disaster if the filter output forms the sensor for a control loop.

At other times, a sensor that is not wired correctly, not turned on, poorly calibrated or in any other way not functioning correctly can cause a computational tug of war in the filter that may lead to divergence. One good generic reasonableness check is to compare the Kalman residual for each measurement with its own nominal uncertainty. If the residual is much larger than the uncertainty, then something is probably wrong and the measurement can be discarded. This is a good way of eliminating spikes and outliers. If a sensor fails to meet this criterion consistently, then it can be disabled forever and an error message issued.

16.5 Reconfiguration

Certain issues arise during the development of a filtering algorithm for which a small number of simple techniques are useful.

16.5.1 Removing Sensors

When for some reason, a sensor becomes unavailable, it may be necessary to change the operation of the filter. The simplest case is when the filter is implemented for asynchronous measurements and the appropriate variables can be set to indicate that the measurement is no longer available. If this is not possible, the uncertainty of the measurement can be set to a large number in the R_k matrix. If the removed sensor is the last sensor which measured a particular state, then that state may need to be removed as well because cross coupling in the matrices will generate fictitious values for the state which will then begin to corrupt other states. In the event that observability is compromised, the whole filter is not viable without the lost sensor.

16.5.2 Removing States

In order to remove a state, the numerical operation of the filter must be as if the filter was originally formulated without it. It is not enough to set the state uncertainty to a large number, the value of the state must be set to zero every cycle in order to damp the errors introduced by every iteration.

16.6 Asynchronous Implementation

One matter to be addressed in implementation is the asynchronous nature of the measurements. For example, Doppler readings may not be available as frequently as encoder readings. GPS measurements cannot be generated faster than 2 Hz whereas inertial systems can be 100 times faster than this. In general, it is a good idea to adopt a time step which is equal to the period of the fastest sensor and to run the state equations at that speed. Then a loop is entered which establishes whether a particular measurement is available, and if so, runs the filter equations on it.

Thus the basic algorithm is as given below:

```
State_Update() /* entered every cycle */
  {
  update state estimate for a time step of dt
  via the transition matrix(dt);
  if( Doppler measurement available)
    run Kalman() on Doppler;
  if( Encoder measurement available)
    run Kalman() on encoder;
  if( AHRS measurement available)
    run Kalman() on AHRS;
  if( Compass measurement available)
    run Kalman() on compass;
  if( Steering measurement available)
    run Kalman() on steering;
  }

Kalman()
  {
  }
```

16.7 Support Software

A few software tools go a long way to making it possible to debug a filter. As a minimum, a data logging facility of some kind is required which does not significantly degrade real time performance, and a signal plotting facility makes off line analysis much more pleasant than perusing megabytes of floating point numbers. In any data logger, time must be recorded to ensure deterministic playback, because the system clock is a sensor like any other.

16.8 Efficiency

In implementing the filter equations, it is typical to find that most of the time is spent computing the uncertainty matrices, and the Kalman gain, and in inverting and multiplying matrices. Some simple steps can be taken to reduce this problem. The following two steps make it possible to run the filter at 100 Hz.

16.8.1 Uncertainty Propagation

The matrix uncertainty propagation equation for an EKF is:

$$P = [I - (KH)]P$$

This can be computed many times faster as:

$$P = P - K(HP)$$

where the product HP is to be evaluated first. This rewrite is valid regardless of the form of the measurement matrix H.

16.8.2 Measurements Related to a Single State

16.8.2.1 Kalman Gain

The matrix Kalman gain equation for an EKF is:

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1}$$

Let a single measurement arrive for integration with the state estimate. Then, the R matrix is a scalar:

$$R = [r]$$

Let the measurement project onto a single state whose index is s with a coefficient of unity. Then the H matrix is a row vector with a single unit element in the s'th position:

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The expression $P_k^- H_k^T$ is the s'th column of P_k^- and the expression $H_k P_k^- H_k^T$ is the (s,s) element of P_k^-. Define:

$$p = P_{ss}$$

Finally, the Kalman gain is a column vector equal to a constant times the s'th column of P_k^-:

$$K = \left(\frac{1}{p + r}\right) P_{is} \qquad \forall i$$

16.8.2.2 Uncertainty Propagation

Further simplification is possible in the uncertainty update equation in the case of a single scalar measurement. Let a single measurement arrive for integration with the state estimate. Again, let the measurement project onto a single state whose index is s with a coefficient of unity. Then the H matrix is a row vector with a single unit element in the s'th position:

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The expression HP is then the s'th row of P_k^-. Reusing the last result, the expression KHP is simply a constant times the outer product of the s'th column and the s'th row of P_k^-:

$$(KHP)_{ij} = \left(\frac{1}{p+r}\right) P_{is} P_{sj} \qquad \forall i \forall j$$

16.8.3 Measurements Related to Few States

In some cases, such as in SLAM applications, the covariance matrix may be 100 X 100 (for 100 landmarks) whereas the measurement Jacobian projects onto the vehicle pose and only one landmark.

16.8.3.1 Prediction Uncertainty

The matrix Kalman gain equation for an EKF is again:

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1}$$

Let a single measurement arrive for integration with the state estimate. Then, the R matrix is a scalar:

$$R = [r]$$

Let the measurement project onto a a few states with nonzero coefficients. The uncertainty of the predicted measurement is:

$$H_k P_k^- H_k^T$$

For example, suppose there are two nonzero elements in H. Rewrite it as the sum of two matrices which project onto one state:

$$H = H_1 + H_2$$

Then, the uncertainty is:

$$\boxed{(H_1 + H_2)P_k^-(H_1 + H_2)^T = H_1P_k^-H_1^T + H_2P_k^-H_1^T + H_1P_k^-H_2^T + H_2P_k^-H_2^T}$$

Due to the symmetry of P, the second and third terms are mutual transposes. This result implies that there are 4 component matrices to the covariance update which have very simple forms. Assuming scalar measurements, if the H_1 matrix is nonzero in the s element and the H_2 matrix is nonzero in the t element, then these four expressions are the scalars:

$$H_1P_k^-H_1^T = P_{ss}$$

$$H_2P_k^-H_1^T = H_1P_k^-H_2^T = P_{st}$$

$$H_1P_k^-H_1^T = P_{tt}$$

In a more general case of m nonzero coefficients there will be m diagonal elements of P and m(m-1) off diagonal elements (with coefficients of 2) added for a total cost of m*m.

This computation compares favorably with the straightforward application of matrix multiplication. If there are n elements in the state vector, then $P_k^-H_k^T$ generates a column vector in n*n operations and the product $H_kP_k^-H_k^T$ requires n more operations, for a total of n(n+1) operations. If m << n the above technique may be worth the effort.

16.8.3.2 Kalman Gain

Computing the Kalman gain from the above result requires the computation of:

$$P_k^-H_k^T = P_k^-(H_1 + H_2)^T$$

Based on the earlier analysis, this computation requires the addition of the s’th and t’th columns of P_k^- which is an order n operation. For m nonzero coefficients, the cost is mn. A straightforward matrix multiplication would require n*n operations.

16.8.3.3 Uncertainty Propagation

The matrix uncertainty propagation requires the computation of $K(HP)$. If there are m nonzero elements of H, the product HP is a row vector that can be evaluated in mn operations (by adding the relevant m rows of P) rather than n*n. Then, the product $K(HP)$ is an outer product that requires n*n operations on the assumption both vectors are fully populated.

The final step is to compute $P = P - K(HP)$ which requires n*n operations unless we have an explicit list of the nonzero elements in $K(HP)$. Such a list of elements (i,j) is those pairs for which K_i and $(HP)_j$ are both nonzero. Both vectors are derived from sums of rows or columns of P so there will be no nonzero elements unless there are uncorrelated states.

16.8.4 Partitioning

In some cases (as in SLAM), there are opprotunities to partition the matrices involved in order to generate extreme efficiencies (1000 times faster for 100 landmarks).

Let the state vector be partitioned into a vehicle component and a landmark component thus:

$$\underline{x} = \begin{bmatrix} \underline{x}_v^T & \underline{x}_L^T \end{bmatrix}^T \qquad (1)$$

The transition matrix is:

$$\Phi = \begin{bmatrix} \Phi_{vv} & 0 \\ 0 & I \end{bmatrix} \qquad (2)$$

Similiarly, the state covariance can be written as:

$$P = \begin{bmatrix} P_{vv} & P_{vL} \\ P_{Lv} & P_{LL} \end{bmatrix} \qquad (3)$$

Recall the state covariance update equation:

$$P^-_{k+1} = \Phi_k P_k \Phi_k^T + \Gamma_k Q_k \Gamma_k^T \qquad (4)$$

The first term in the state covariance update equation is therefore:

$$\Phi_k P_k \Phi_k^T = \begin{bmatrix} \Phi_{vv} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} P_{vv} & P_{vL} \\ P_{Lv} & P_{LL} \end{bmatrix} \begin{bmatrix} \Phi_{vv}^T & 0 \\ 0 & I \end{bmatrix}$$

$$\Phi_k P_k \Phi_k^T = \begin{bmatrix} \Phi_{vv} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} P_{vv}\Phi_{vv}^T & P_{vL}I \\ P_{Lv}\Phi_{vv}^T & P_{LL}I \end{bmatrix}$$

$$\Phi_k P_k \Phi_k^T = \begin{bmatrix} \Phi_{vv}P_{vv}\Phi_{vv}^T & \Phi_{vv}P_{vL}I \\ IP_{Lv}\Phi_{vv}^T & IP_{LL}I \end{bmatrix}$$

So that the individual elements are:

$$P_{vv} = \Phi_{vv} P_{vv} \Phi_{vv}^T$$

$$\underset{(nXm)}{P_{vL}} = \underset{(nXn)}{\Phi_{vv}} \underset{(nXm)}{P_{vL}}$$

$$P_{LL} = P_{LL}$$

If there are n (=5) vehicle states and m(=200) landmark states, the total processing is 2n^3+n^2m = 250+25*200 = 5250. This compares to 255^3 = 8 million operations if the unpartitioned equations are used. Hence, the partitioning is 1000 times faster for 100 landmarks.

The process noise distribution is partitioned as:

$$\Gamma = \begin{bmatrix} \Gamma_{vv} & 0 \\ 0 & I \end{bmatrix}$$

The process noise is:

$$Q = \begin{bmatrix} Q_{vv} & 0 \\ 0 & 0 \end{bmatrix}$$

The second term in state covariance update equation is:

$$\Gamma_k Q_k \Gamma_k^T = \begin{bmatrix} \Gamma_{vv} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} Q_{vv} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Gamma_{vv}^T & 0 \\ 0 & I \end{bmatrix}$$

$$\Gamma_k Q_k \Gamma_k^T = \begin{bmatrix} \Gamma_{vv} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} Q_{vv} \Gamma_{vv}^T & 0 \\ 0 & 0 \end{bmatrix}$$

$$\Gamma_k Q_k \Gamma_k^T = \begin{bmatrix} \Gamma_{vv} Q_{vv} \Gamma_{vv}^T & 0 \\ 0 & 0 \end{bmatrix}$$

So that the calculation is entirely independent of the number of landmarks.

The complete partitioned state covariance update equation is therefore:

$$P_{vv} = \Phi_{vv} P_{vv} \Phi_{vv}^T + \Gamma_{vv} Q_{vv} \Gamma_{vv}^T$$

$$P_{vL} = \Phi_{vv} P_{vL}$$

16.9 Ignoring Transforms

At times, aspects of the transformation matrices can be ignored. An AHRS, for instance can be positioned anywhere provided it is aligned with the body axes. The vehicle is basically a rigid body, so the attitude of the AHRS is the attitude of the vehicle, wherever the AHRS is positioned.

For positioners which provide their own position, this can only be done under certain circumstances (i.e. relative navigation).

16.10 Modular Code

Any implementation of these algorithms will benefit significantly if the code is organized to compute the fundamental transformation matrices in a single place and pass them to the consumers. Particularly in the case of the real time identification application, it is unwise to copy the same strings of cosines and sines into too many separate places.

16.11 State Discontinuities

When high quality measurements, like fixes, arrive infrequently, it is possible for the state estimate to undergo a rapid change in a short period of time. This can wreak havoc in any closed loop system since it appears as a large deviation between the reference input and the feedback signal.

One technique for dealing with this has been presented in the form of the simple linearized Kalman filter which computes the state error vector instead of the state. Another technique is to low pass filter the state output but this has the disadvantage of delaying control response to legitimate high dynamic feedback if the entire state vector is filtered.

Since discontinuities will only arise in practice in the position states, it is appropriate to slowly filter the x, y, and z components of the state discontinuity ***only*** into the state estimate used as position feedback by control algorithms. An appropriate choice of time constant in the filter will prevent unusual behavior. This approach becomes increasingly less viable as the frequency of fix information is reduced.

16.12 Outlier Rejection

Although a Kalman filter can handle random measurement noise, a single outlier can cause a large erroneous change in the state estimate unless it is rejected before it is used in the state update equation. This section describes some techniques used to achieve outlier rejection.

16.12.1 Angular Addition

Notice that the Kalman state update equation and the system model constitute two places where angles are added. After each iteration of these equations, and during the formation of the measurement residual, the angular states ***must*** be forced into some canonical sector of the unit circle such as $-\pi$ to π. If this is not done, the measurement residuals will begin, apparently randomly, to overestimate the mismatch by a full revolution and the state vector will rotate one revolution over a few cycles. This is usually quite a disaster if the filter output forms the sensor for a control loop.

16.12.2 Reasonableness Checks

At other times, a sensor that is not wired correctly, not turned on, poorly calibrated or in any other way not functioning correctly can cause a computational tug of war in the filter that may lead to divergence. One good generic reasonableness check is to compare the Kalman residual for each measurement with its own nominal uncertainty. If the residual is much larger than the uncertainty, then something is probably wrong and the measurement can be discarded. This is a good way of eliminating spikes and outliers. If a sensor fails to meet this criterion consistently, then it can be disabled forever and an error message issued.

16.12.3 Legitimate State Discontinuities

When high quality measurements, like fixes, arrive infrequently, it is possible for the state estimate to undergo a rapid change in a short period of time. This can wreak havoc in any closed loop system since it appears as a large deviation between the reference input and the feedback signal.

One technique for dealing with this is presented in the appendices in the form of the simple linearized Kalman filter which computes the state error vector instead of the state. Another technique is to low-pass filter the state output but this has the disadvantage of delaying control response to legitimate high dynamic feedback if the entire state vector is filtered.

Since discontinuities will only arise in practice in the position states, it is appropriate to slowly filter the x, y, and z components of the state discontinuity ***only*** into the state estimate used as position feedback by control algorithms. An appropriate choice of time constant in the filter will prevent unusual behavior. This approach becomes increasingly less viable as the frequency of fix information is reduced.

17. Test Results

The 3D dead reckoning AHRS filter has been implemented and tested in the field. As a practical necessity, graphical output including error ellipse generation and overhead 2D animated graphics are included. A debugging mode dumps all important matrices to a file for later analysis. A data logger permits recording and playback of results. A plotfile generation facility provides time, filter state, and measurement data for analysis purposes. Cycle numbers in the plotfiles permit reference of graphical events to the debug files.

Tuning of the state uncertainties was required to optimize the tradeoff between data smoothing and sensor tracking. A zup mode was included in order to check undesirable growth of the state uncertainty when the vehicle was stopped. The filter can switch from linear (LKF) to extended (EKF) mode. It was observed that heading uncertainty was treated much more accurately in EKF mode. Specifically, uncertainty grows primarily in the direction of travel in LKF mode, whereas when the coupling terms of the EKF are introduced, uncertainty grows transverse to it. The latter is more realistic in situations when the heading indication is poor. Indeed, the overwhelming component of position error was due to poor attitude indications.

The sensor used were a steering wheel potentiometer for “yawrate” measurement, transmission encoder and redundant doppler radar for measurement of velocity, and an AHRS for three axis attitude. Time stamps were used to compute encoder velocity precisely.

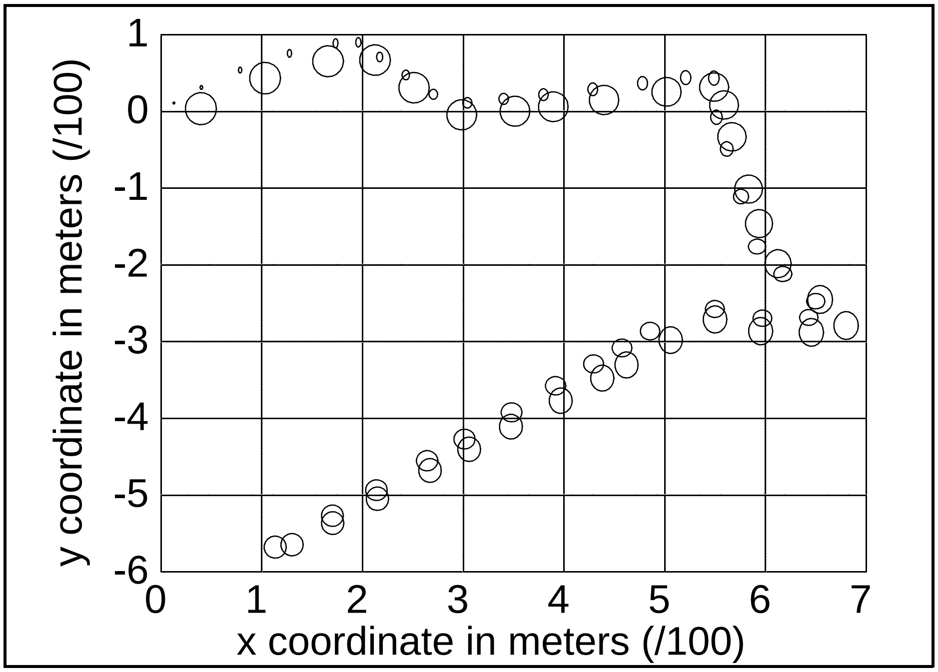
Significant systematic errors were easily identified from the signal plots which were not resolved before this writing.

17.1 Test Run

One of many test runs will be used to illustrate performance. In this run, winding mountainous city streets were driven. The total excursion was about 4 Km in the horizontal plane and 200 meters vertically. Aggressive braking and accelerating maneuvers were common. On many occasions, the vehicle stopped for traffic lights, and the filter idled in zup mode until motion resumed.

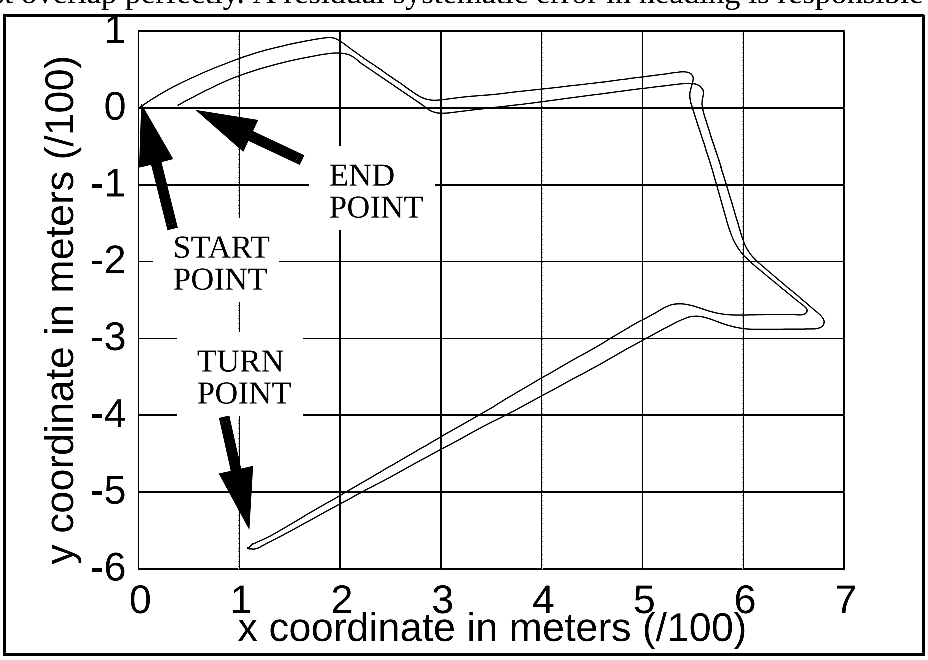
17.2 Uncertainty Growth

The qualitatively correct growth of uncertainty is illustrated in the following figure because the uncertainty ellipses touch the path when it was driven in the other direction. Point repeatability less than 1% of the travelled distance was normally achieved. Relative and absolute accuracy were not quantified.



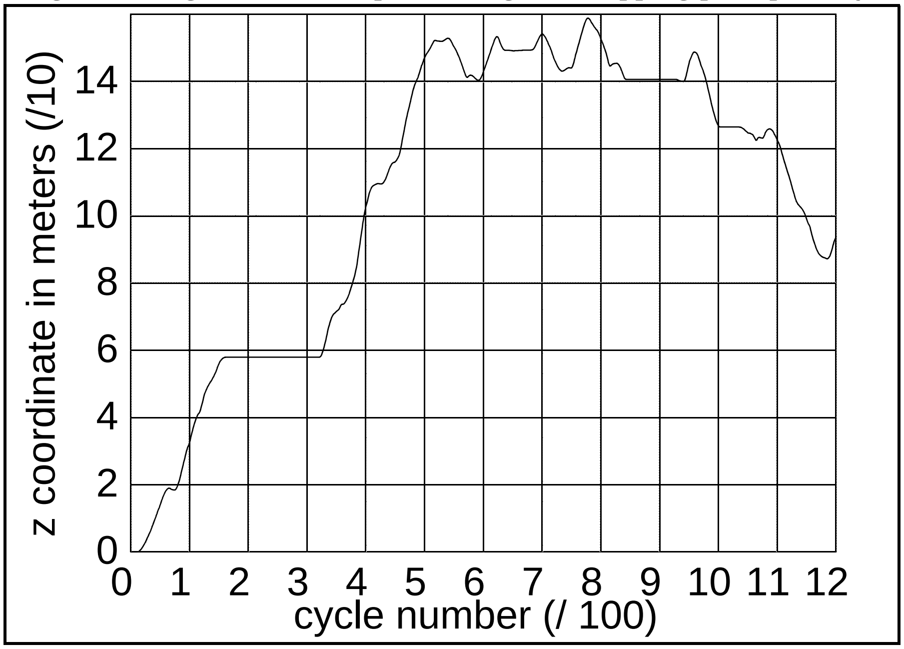
17.3 Plane Position Output

The following figure indicates the position output in the plane for the run. Notice that the return path from the top of the hill could be rotated through a small angle at the turn point and the graphs would almost overlap perfectly. A residual systematic error in heading is responsible for this.



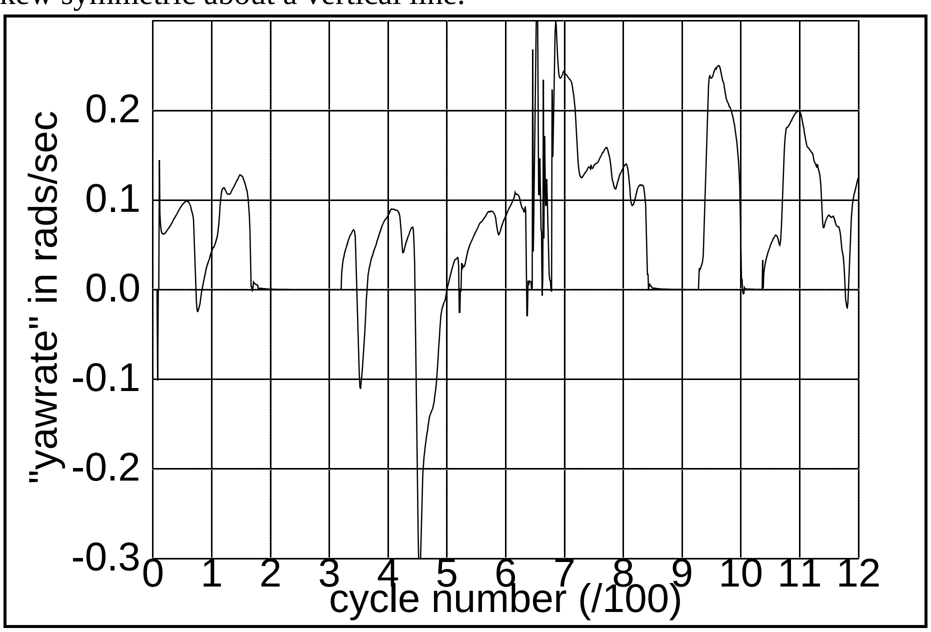
17.4 Vertical Position Output

One of the advantages of the 3D formulation is the availability of the z coordinate. In the following figure, zups appears as flat regions because the abscissa is time. Notice that the start and end do not agree. This is likely due to the high vehicle accelerations corrupting the inclinometers used. In fact, the plotted pitch signal shows significant periods when the output during the downhill leg was positive. However the relative accuracy for a few seconds is very good. This suggests that the z output should be good enough to avoid map matching in a mapping perception system.



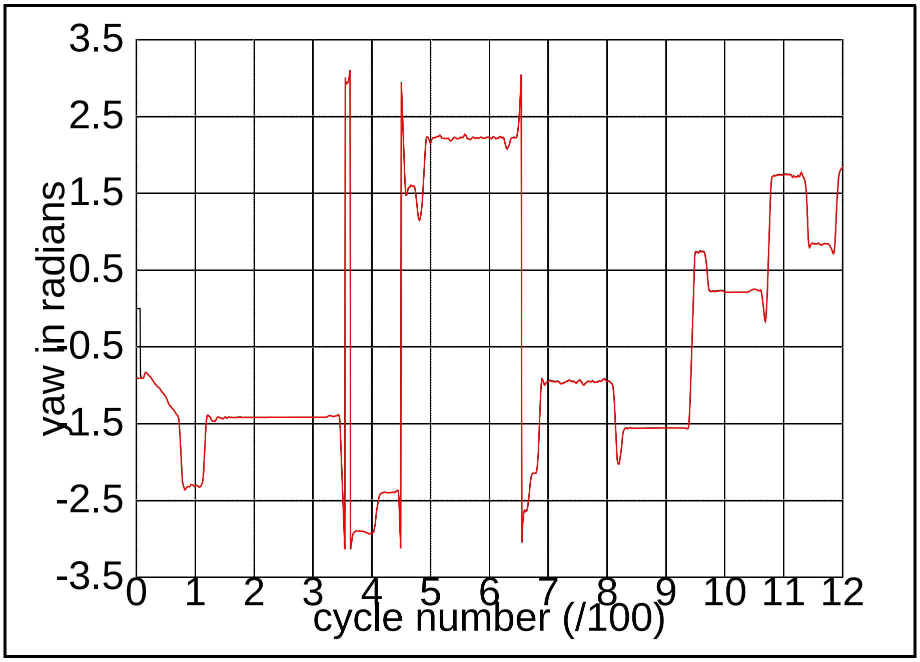
17.5 Yawrate Output

The yawrate state is available from the steering wheel encoder. A nonlinearity in the steering sensor causes a bias near zero which causes the graph to be predominantly positive, when, in fact, it should be roughly skew symmetric about a vertical line.



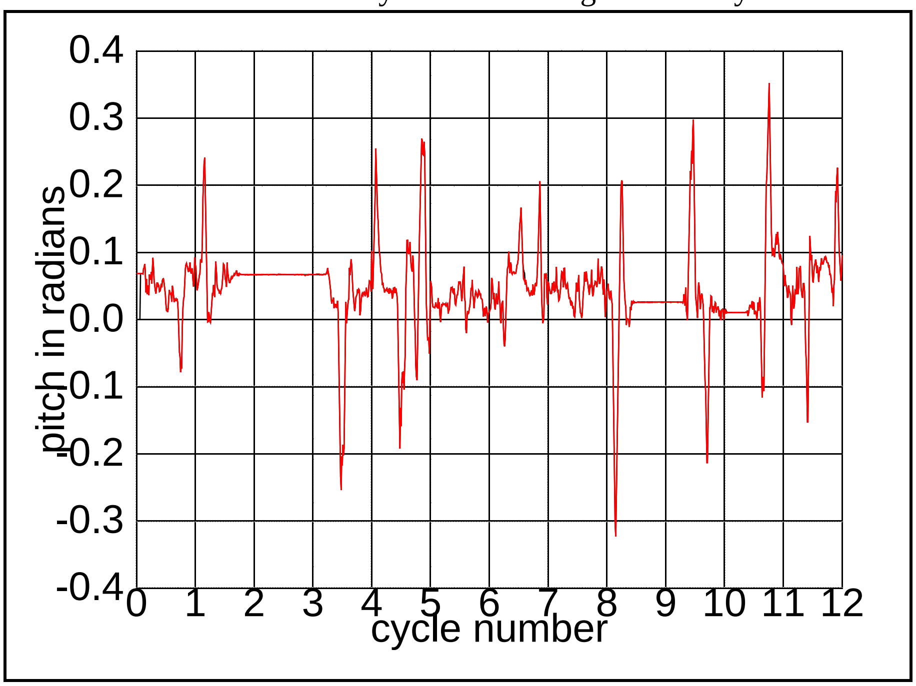
17.6 Yaw Output

The yaw output indicates the success of the reasonableness checks in angle addition in the filter. Both the yaw sensor and the filter state are plotted but they cannot be distinguished beyond the first few cycles.



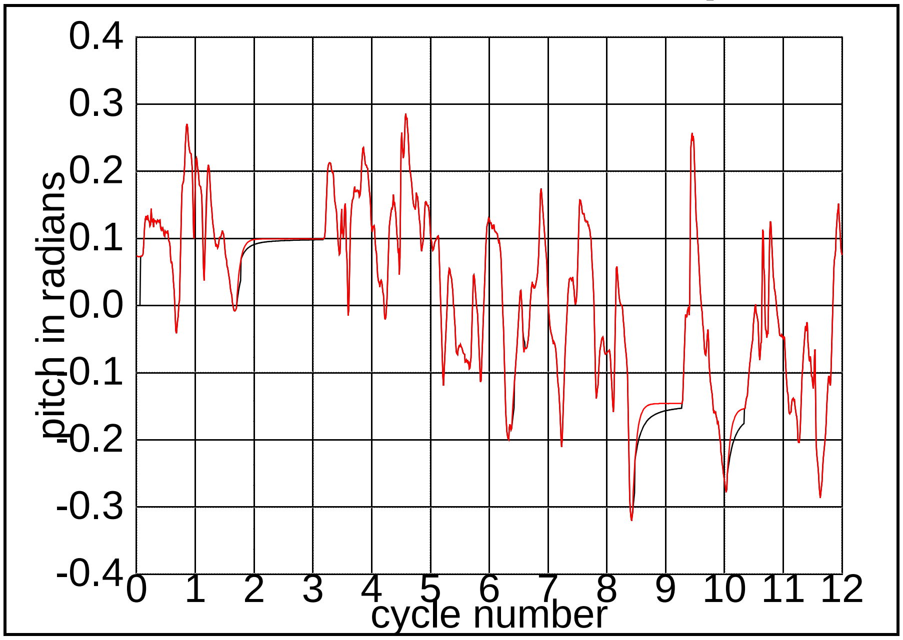
17.7 Roll Output

The roll state and sensor reading are plotted below. Again, the state simply tracks the single sensor. Corruption via vehicle acceleration is likely because the signal is not symmetric.



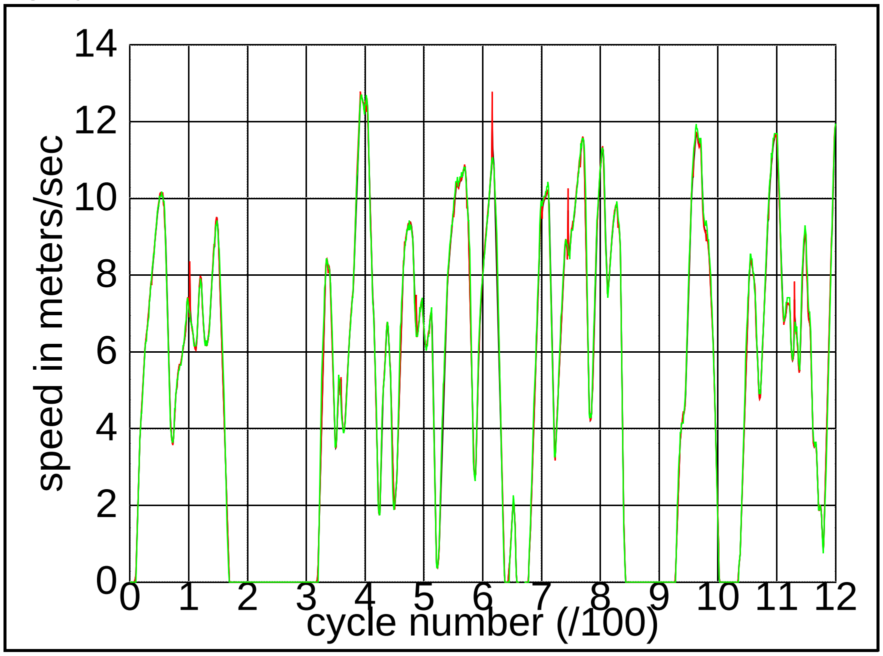
17.8 Pitch Output

The pitch signal shows how vehicle acceleration during braking maneuvers causes the state to lag the sensor output. Curiously, the output is again more positive than it should be. This is a calibration issue which cannot be accounted for in the filter itself, at present.



17.9 Speed Output

The speed output shows how redundant sensors are both used to determine the state. The doppler and encoder readings agree so well that the averaging of the filter is hard to see. On magnification, it is clear that the filter is using both and determining an estimate which is a weighted average when they disagree slightly.



18. Kalman Filter for Aided Inertial Systems

A description of this filter is given in [3]. Inertial navigation systems[1] which operate close to the earth and which close a Shuler Loop exhibit oscillatory position errors with the characteristic Shuler period of 84 minutes. The error dynamics of the free inertial navigator are the most significant result of inertial navigation theory.

18.1 System Model

Let the x, y, and z axes of the navigation coordinate system correspond to the north, west, and up directions respectively. The system dynamics of a slow moving free inertial navigator can be approximated[2] by the following continuous time model:

$$\boxed{\begin{aligned} \varepsilon_x &= \dot{\psi}_x - \Omega_z \psi_y \\ \varepsilon_y &= \dot{\psi}_y + \Omega_z \psi_x - \Omega_x \psi_z \\ \varepsilon_z &= \dot{\psi}_z + \Omega_x \psi_y \end{aligned}}$$

where ψ_i denote the vehicle position ***errors*** in the west, south, and azimuth directions[3], expressed in radians, ε_i is the gyro drift rate for gyro axis i, and Ω_i is the earth rate component about the appropriate IMU axis. The latter can be computed from the earth's sidereal rate and approximate knowledge of the latitude. In this model, the gyro drifts are considered to be the driving inputs.

At this point, the state vector is given by the 3 vehicle position coordinates. However, the gyro drifts cannot reasonably be modelled as white sequences. It is necessary to augment the state vector. [1] suggests a random constant plus a Markov component with a large time constant for all three drift rates. This gives six additional state variables and six new differential equations.

$$\boxed{\begin{array}{lll} \varepsilon_x = \varepsilon_{xm} + \varepsilon_{xc} & \dot{\varepsilon}_{xm} = -\beta_x \varepsilon_{xm} + u_x & \dot{\varepsilon}_{xc} = 0 \\ \varepsilon_y = \varepsilon_{ym} + \varepsilon_{yc} & \dot{\varepsilon}_{ym} = -\beta_y \varepsilon_{ym} + u_y & \dot{\varepsilon}_{yc} = 0 \\ \varepsilon_z = \varepsilon_{zm} + \varepsilon_{zc} & \dot{\varepsilon}_{zm} = -\beta_z \varepsilon_{zm} + u_z & \dot{\varepsilon}_{zc} = 0 \end{array}}$$

Where β_i is a Markov process time constant, u_i is a white noise process, and the subscripts m and

1. In this report, the poorer cousins of the INS, the attitude and heading reference systems (AHRS) are distinguished from the INS since they are not Schuler tuned.
2. This is an approximate model. It is not observable and it glosses over some coordinate system transformation issues.
3. The first two are related to latitude and longitude error and the last is related to heading error.

c indicate Markov and constant components respectively. The complete state equations become:

$$\dot{x} = Fx + Gu$$

$$\begin{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} \\ \begin{bmatrix} \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} \\ \begin{bmatrix} \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 & \Omega_z & 0 \\ -\Omega_z & 0 & \Omega_x \\ 0 & -\Omega_x & 0 \end{bmatrix} & [I] & [I] \\ [0] & \begin{bmatrix} -\beta_x & 0 & 0 \\ 0 & -\beta_y & 0 \\ 0 & 0 & -\beta_z \end{bmatrix} & [0] \\ [0] & [0] & [0] \end{bmatrix} \begin{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\ \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} \\ \begin{bmatrix} x_7 \\ x_8 \\ x_9 \end{bmatrix} \end{bmatrix} + \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix}$$

The F matrix is not constant unless the system is gimballed and the mission is limited in excursion. Constant or not, standard techniques can be used to convert F to its discrete time transition matrix Φ. The β_i are chosen to be large, and the Q matrix is constant and diagonal, and based on knowledge of the magnitude of the random component of gyro error[1]. Based on this much information, the nominal system trajectory and its covariance can be extrapolated forward in time. This covers the last two equations in the filter.

18.2 Aided Inertial Updates

The inertial system operates as a computational flywheel which supplies position updates at high rates until a fix is obtained which can be used to damp the inevitable growth of pure inertial position errors with time. Until a fix is obtained, the last two equations of the filter continue to cycle and basically solve the differential equations of the system model in order to determine the position errors.

In this simplified model, consider that a fix is obtained from some navaid which provides redundant measurements of only the first two state variables (the plane position errors). This gives

1. Kalman filters are so entrenched in the navigation industry that the vendors all talk the lingo. Sometimes, random walk and gyro bias can be lifted right off the vendor product brochure.

the following measurement model:

$$\bar{z}_k = H_k \bar{x}_k + \bar{v}_k$$

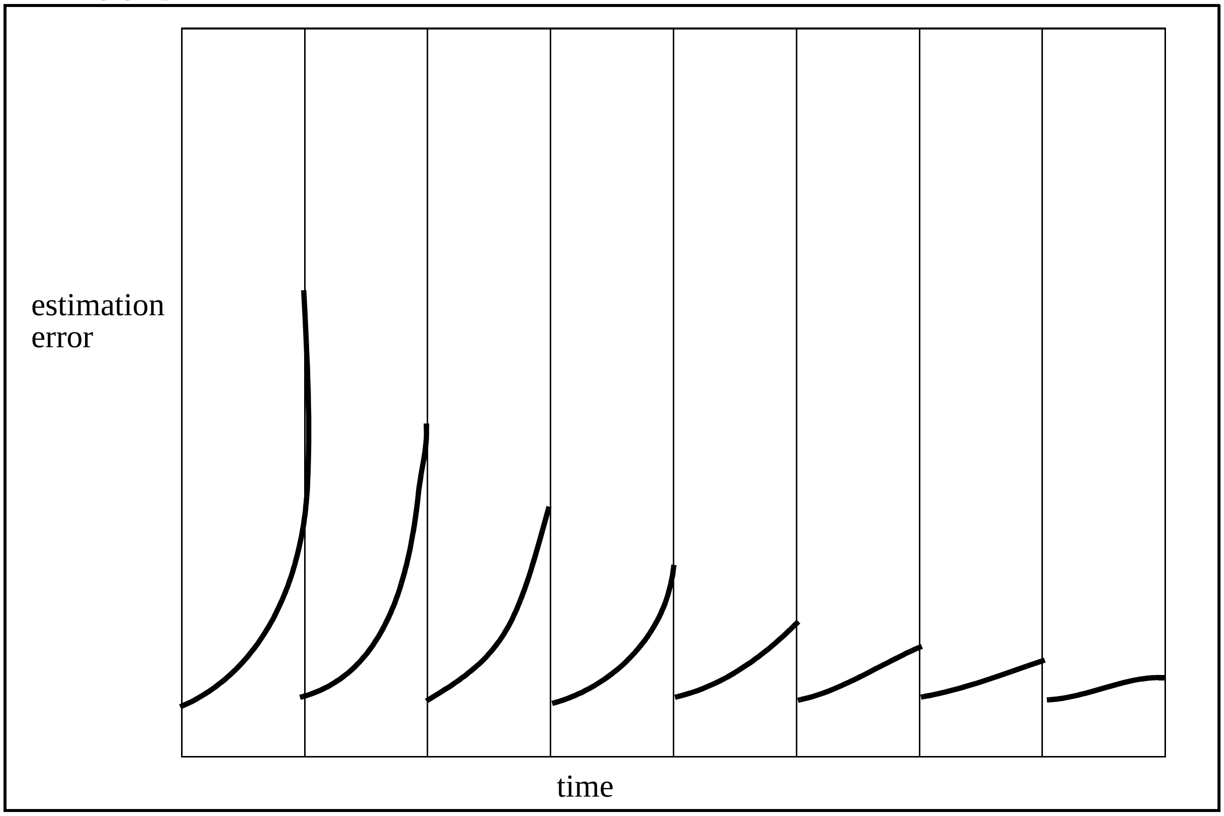
$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\ \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} \\ \begin{bmatrix} x_7 \\ x_8 \\ x_9 \end{bmatrix} \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

This gives the H_k matrix, and the R_k covariance matrix will be assumed to be a diagonal matrix whose diagonal components are the variances of the external fix. No more can be said about these at this point since they depend on the exact nature of the device providing the fix. Given these two matrices, the first three equations of the filter can be iterated to update the entire state vector.

18.3 Significance of the State Vector Augmentation

By augmenting the state vector, this system is capable of identifying itself in real time. That is, while the statistical variation of gyro performance across all systems is all that is known before the filter is run, both the constant gyro bias and the random component are estimated as the system runs. As a consequence of the nonzero entries coupling the vehicle position to these artificial states, the system will learn about its own nonidealities and compensate over time.

It is the measurement process which provides information, albeit very indirectly, on gyro biases. Indeed, the time evolution of system errors for a periodic fix would be expected to resemble the following graph:



Thus, in the presence of a regular fix, the system will approach a kind of steady state estimation error as position fixes allow it to generate increasingly better estimates of gyro nonidealities.

19. A GPS Filter

In order to illustrate the operation of a filter implemented inside a GPS receiver, the following filter is reproduced from [3]. This filter also provides insights into formulations for laser and radar triangulation systems.

19.1 The Measurement Model

The operation of a GPS receiver is based upon 3D range triangulation. The measurement model is easiest to formulate in a cartesian geocentric navigation frame such as WGS-84, and this is the frame in which inexpensive receivers typically report position. The pseudoranges to four satellites are given by:

$$\boxed{\begin{aligned}
\psi_1 &= \sqrt{(X_1 - x) + (Y_1 - y) + (Z_1 - z)} + c\Delta t \\
\psi_2 &= \sqrt{(X_2 - x) + (Y_2 - y) + (Z_2 - z)} + c\Delta t \\
\psi_3 &= \sqrt{(X_3 - x) + (Y_3 - y) + (Z_3 - z)} + c\Delta t \\
\psi_4 &= \sqrt{(X_4 - x) + (Y_4 - y) + (Z_4 - z)} + c\Delta t
\end{aligned}}$$

where ψ_i is the noiseless pseudorange to satellite i, $\begin{bmatrix} X_i & Y_i & Z_i \end{bmatrix}^T$ is the geocentric cartesian position of satellite i, Δt is the user clock bias, and c is the speed of light. The position of the receiver is $\begin{bmatrix} x & y & z \end{bmatrix}^T$ and it also given in the nav frame.

19.2 The Measurement Jacobian

It is a simple matter to differentiate the above expression with respect to the states. Without writing it out, using a linearized filter, the result is:

$$\boxed{\begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{bmatrix} - \begin{bmatrix} \psi_1(\bar{x}_0) \\ \psi_2(\bar{x}_0) \\ \psi_3(\bar{x}_0) \\ \psi_4(\bar{x}_0) \end{bmatrix} = \begin{bmatrix} \frac{\partial\psi_1}{\partial x} & \frac{\partial\psi_1}{\partial x} & \frac{\partial\psi_1}{\partial x} & 1 \\ \frac{\partial\psi_2}{\partial x} & \frac{\partial\psi_2}{\partial x} & \frac{\partial\psi_2}{\partial x} & 1 \\ \frac{\partial\psi_3}{\partial x} & \frac{\partial\psi_2}{\partial x} & \frac{\partial\psi_2}{\partial x} & 1 \\ \frac{\partial\psi_4}{\partial x} & \frac{\partial\psi_4}{\partial x} & \frac{\partial\psi_4}{\partial x} & 1 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ c\Delta t \end{bmatrix}}$$

where $\bar{x}_0$ is the nominal “trajectory” which can be a fixed point for short excursions, or the state vector itself in an Extended Kalman filter.

19.3 System Model

The random process describing the error in the clock is dependent on the clock construction. It is often useful to define two states to model this error, a **bias** which is the frequency error of the oscillator and a **drift** which quantifies the stability of the frequency.

The random process which describes the errors in the position state is dependent on the dynamics of the receiver. For a moving receiver, position and at least velocity states are necessary. This gives the following system model.

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ b \\ d \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ b \\ d \end{bmatrix}$$

from which the transition matrix is immediate. The measurement Jacobian of the previous section would require slight modification for this enlarged and augmented state vector.

19.4 System Disturbance Model

The disturbance model is derived by considering that the velocity states are corrupted by white noise of spectral amplitude S_p and the clock states are corrupted, respectively by noises of spectral amplitude S_b and S_d. Thus:

$$\bar{w} = \begin{bmatrix} 0 & w_2 & 0 & w_4 & 0 & w_6 & w_7 & w_8 \end{bmatrix}^T$$

Under this model, the noise covariances are given by the corresponding autocorrelation functions which generates a banded Q matrix.

20. Conversion Between Conventional Global Navigation Frames

Although it is almost never directly useful to an autonomous vehicle to know its exact position referenced to the earth itself, it is often the case that commercial navigation systems report results in a geocentric or locally level frame of reference and different products use different systems. When one attempts to use any global navigation device which reports position in a coordinate system referenced to any absolute point on the earth, the picture changes markedly. In general, the output of the filter and the global navigation device must in one way or another be referred to common coordinates. The construction of the transformation between them involves:

- registration - the position of one system wrt the other
- alignment - the orientation of one system wrt another

It is an unfortunate bother to have to convert coordinate systems between, say the output of a land inertial navigation system and a GPS receiver, but that is how it is. Faced with an off the shelf GPS product which generates WGS-84 and an INS which generates UTM, it is very useful to know the equations which convert one to the other. The standard reference for this information is [21]. This section quotes these equations without explanation for reference purposes.

20.1 The World Geodetic Frame 1984 (WGS-84)

Coordinates expressed in WGS-84 and its later revisions are commonly output by inexpensive GPS receivers. This situation arises because the GPS receiver is slaved to the coordinate system in which the satellite positions (the "landmarks") are known. This coordinate system is a right handed cartesian system centered at the mass center of the earth[1]. The z axis points through the north pole and the x axis points through the prime meridian at the equator. The coordinates of a point will be denoted simply x, y, and z here.

20.2 Geodetic Coordinates

The geodetic latitude ϕ, longitude λ, and height h are the familiar spherical polar coordinates for positioning a point on or above the earth's surface.

20.3 Universal Transverse Mercator (UTM) Coordinates

UTM is a map projection of the spherical earth onto the plane. In the northern hemisphere, the y coordinate, called **northing**, is zero at the equator and increases northward. In the southern hemisphere, the **southing** is 10×10^6 and decreases to the south. The x coordinate, called **easting**, is 500,000 at the central meridian and increases to the east. The unit system is meters.

Portions of the earth are projected onto an imaginary cylinder whose axis of symmetry is, unlike the Mercator projection, ***transverse*** to the north south axis of the earth. Then the cylinder is unrolled to produce a flat map. In polar latitudes, the Universal Polar Stereographic (UPS) projection is used instead.

The process produces unacceptable distortion over a large area, so the earth has been divided in the east-west direction into 60 slices of longitude, called zones, each 6° (360 nautical miles) wide,

1. There are many subtleties in the exact specification of all of these coordinate systems. The issue here is the equations which convert from one to the other and the relevant parameters. All other issues are ignored.

centered around the longitudes:

$$3 + 6n \quad \text{in degrees}$$

and the projection is intended to be used for any one longitudinal zone up to 40 km into the adjacent zones. Zone number 1 is centered at 183° E and zone numbers increase to the east.

In the north south direction, the system is defined only between 84° N and 80° S and it is likewise divided into 60 zones each of which is 6° wide in latitude. Each latitude zone has a letter designation, starting with C in the north and increasing alphabetically to the south to X, where the letters O and I are excluded.

20.4 Geodetic to WGS-84

$$x = (R_N + h)\cos\phi\cos\lambda$$

$$y = (R_N + h)\cos\phi\sin\lambda$$

$$z = \left(\frac{b^2}{a^2}R_N + h\right)\sin\phi$$

$$R_N = \frac{a^2}{\sqrt{a^2\cos^2\phi + b^2\sin^2\phi}}$$

Where for WGS-84 a=6378137m. and b = 6356752.3142 m.

20.5 WGS-84 to Geodetic

$$\lambda = atan\left(\frac{y}{x}\right)$$

to find ϕ iterate a few time these four equations

$$\beta = atan\left(\frac{az}{b\sqrt{x^2+y^2}}\right) \qquad tan\phi = \frac{z + \delta^2 sin^3\beta}{-a\varepsilon^2 cos^3\beta + \sqrt{x^2+y^2}}$$

$$\varepsilon^2 = \frac{a^2 - b^2}{a^2} \qquad \delta^2 = \frac{a^2 - b^2}{b^2}$$

in nonpolar areas

$$h = \frac{\sqrt{x^2+y^2}}{cos\phi} + R_N$$

20.6 Geodetic to UTM

First compute the zone ζ where the longitude λ is given in positive radians:

$$\zeta = \left\lfloor 31 + \frac{180\lambda}{6\pi} \right\rfloor \qquad 0 \le \lambda < \pi$$

$$\zeta = \left\lfloor \frac{180\lambda}{6\pi} - 29 \right\rfloor \qquad \pi \le \lambda \le 2\pi$$

The central meridian is given by:

$$\lambda_0 = (6\zeta - 183)\frac{\pi}{180}\text{rads} \qquad \zeta \ge 31$$

$$\lambda_0 = (6\zeta + 177)\frac{\pi}{180}\text{rads} \qquad \zeta \le 30$$

The transverse Mercator coordinates come from a three term power series:

$$x_{TM} = N\Lambda \mathit{cos}\phi + \frac{N\Lambda^3 \mathit{cos}^3\phi}{3!}(\tau_1) + \frac{N\Lambda^5 \mathit{cos}^5\phi}{6!}(\tau_2)$$

$$y_{TM} = S_\phi + \frac{N\Lambda^2 \mathit{sin}\phi\,\mathit{cos}\phi}{2!} + \frac{N\Lambda^4 \mathit{sin}\phi\,\mathit{cos}^3\phi}{4!}(\tau_3) + \frac{N\Lambda^6 \mathit{sin}\phi\,\mathit{cos}^5\phi}{6!}(\tau_4)$$

where:

$$\tau_1 = (1 - t^2 + \eta^2) \qquad \tau_3 = (5 - t^2 + 9\eta^2 - 4\eta^4)$$

$$\tau_2 = (5 - 18t^2 + t^4 + 14\eta^2 - 58t^2\eta^2) \qquad \tau_4 = (61 - 58t^2 + t^4 + 270\eta^2 - 330t^2\eta^2)$$

and:

$$N = \frac{a}{\sqrt{1 - \varepsilon^2 \mathit{sin}^2\phi}} \qquad t = \mathit{tan}\phi \qquad \eta = \delta\,\mathit{cos}\phi \qquad \varepsilon^2 = \frac{a^2 - b^2}{a^2} \qquad \delta^2 = \frac{\varepsilon^2}{1 - \varepsilon^2}$$

and:

$$S_\phi = a[A_0\phi - A_2 \mathit{sin}2\phi + A_4 \mathit{sin}4\phi - A_6 \mathit{sin}6\phi + A_8 \mathit{sin}8\phi]$$

and:

$$A_0 = 1 - \frac{1}{2^2}\varepsilon^2 - \frac{3}{2^6}\varepsilon^4 - \frac{5}{2^8}\varepsilon^6 - \frac{175}{2^{14}}\varepsilon^8$$

$$A_2 = \frac{3}{8}\left(\varepsilon^2 + \frac{1}{4}\varepsilon^4 + \frac{15}{128}\varepsilon^6 - \frac{455}{4096}\varepsilon^8\right)$$

$$A_4 = \frac{15}{256}\left(\varepsilon^4 + \frac{3}{4}\varepsilon^6 - \frac{77}{128}\varepsilon^8\right)$$

$$A_6 = \frac{35}{3072}\left(\varepsilon^6 - \frac{41}{32}\varepsilon^8\right)$$

$$A_8 = -\frac{315}{131072}\varepsilon^8$$

$$\Lambda = (\lambda - \lambda_0)\text{rads}$$

$$\lambda_0 = \text{central\_meridian\_in\_rads}$$

Finally, the UTM coordinates are given in the northern hemisphere by:

$$x_{UTM} = 0.9996x_{TM} + 500000 \qquad y_{UTM} = 0.9996y_{TM}$$

and in the southern hemisphere by:

$$x_{UTM} = 0.9996x_{TM} + 500000 \qquad y_{UTM} = 0.9996y_{TM} + 100000000$$

20.7 UTM to Geodetic

Given UTM coordinates, zone, and hemisphere. First solve the equations immediately above for the transverse Mercator coordinates. Then find the central meridian from the zone using the equation given earlier.

The geodetic coordinates are then given by a three term power series:

$$\phi = \phi_1 - \frac{t_1 N_1}{R_1}\left[\frac{1}{2}\left(\frac{x}{N_1}\right)^2 - \frac{B_4}{24}\left(\frac{x}{N_1}\right)^4 + \frac{B_6}{720}\left(\frac{x}{N_1}\right)^6\right]$$

$$\lambda = \lambda_0 + \sec\phi_1\left[\left(\frac{x}{N_1}\right) - \frac{B_3}{6}\left(\frac{x}{N_1}\right)^3 + \frac{B_5}{120}\left(\frac{x}{N_1}\right)^5\right]$$

where:

$$B_3 = (1 + 2t^2 + \eta^2) \qquad B_4 = (5 + 3t^2 + \eta^2 - 4\eta^4 - 9t^2\eta^2)$$

$$B_5 = (5 + 28t^2 + 24t^4 + 6\eta^2 + 8t^2\eta^2)$$

$$B_6 = (61 + 90t^2 + 46\eta^2 + 4t^4 - 252t^2\eta^2 - 3\eta^4 - 66t^2\eta^4 - 90t^4\eta^2 + 225t^4\eta^4)$$

and:

$$N_1 = \frac{a}{\sqrt{1 - \varepsilon^2 \sin^2\phi_1}} \qquad \begin{array}{l} t = \tan\phi_1 \\ \eta = \delta\cos\phi_1 \end{array} \qquad \varepsilon^2 = \frac{a^2 - b^2}{a^2} \qquad \delta^2 = \frac{\varepsilon^2}{1 - \varepsilon^2}$$

21. References

[1] B.E. Bona and R. J. Smay, “Optimum reset of SINS”, IEEE Trans on Aerospace and Electronic Systems, AES-2, No 4, 409-414 (July 1966).

[2] W. L. Brogan, “Modern Control Theory”, Quantum Publishers, 1974

[3] R. G. Brown and P. Y. C. Hwang, “Introduction to Random Signals and Applied Kalman Filtering”, Wiley, 1992

[4] J. A. Farrel, “Integrated Aircraft Navigation”, Academic Press

[5] M. Fernandez, and G. R. Macomber, “Inertial Guidance Engineering”, Prentice Hall, Englewood Cliffs, NJ, 1962

[6] K. G. Gauss, “Theory of Motion of the Heavenly Bodies”, New York, Dover, 1963.

[7] A. Gelb editor, “Applied Optimal Estimation, MIT Press.

[8] A. Gelb, “Synthesis of a Very Accurate Inertial Navigation System”, IEEE Trans on Aerospace and Navigational Electronics., Vol 12, June 1965, pp 119-128

[9] A. J. Kelly, “Essential Kinematics for Autonomous Vehicles”, CMU Robotics Institute Technical Report CMU-RI-TR-94-14.

[10] A. J. Kelly, “Modern Inertial and Satellite Navigation Systems”, CMU Robotics Institute Technical Report CMU-RI-TR-94-15.

[11] A. J. Kelly, “A Partial Analysis of the High Speed Autonomous Navigation Problem”, CMU Robotics Institute Technical Report CMU-RI-TR-94-16.

[12] A. J. Kelly, “A Feedforward Control Approach to the Local Navigation Problem for Autonomous Vehicles”, CMU Robotics Institute Technical Report CMU-RI-TR-94-17.

[13] A. J. Kelly, “Adaptive Perception for Autonomous Vehicles”, CMU Robotics Institute Technical Report CMU-RI-TR-94-18.

[14] A. J. Kelly, “An Intelligent Predictive Controller for Autonomous Vehicles”, CMU Robotics Institute Technical Report CMU-RI-TR-94-20.

[15] A. J. Kelly, “Concept Design of A Scanning Laser Rangefinder for Autonomous Vehicles”, CMU Robotics Institute Technical Report CMU-RI-TR-94-21.

[16] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems”, Trans. of the ASME - J of Basic Engr., 35-45, March 1960.

[17] R. E. Kalman and R. S. Bucy, “New Results in Linear Filtering and Prediction Theory”, Trans of the ASME - J or Basic Engr., 95-108, March 1961.

[18] P. S. Maybeck, “Stochastic Models, Estimation, and Control”, Academic Press, 1982.

[19] R. C. Smith and P. Cheeseman, “On the Representation and Estimation of Spatial Uncertainty”, Int’l J of Robotics Research, Vol 5, No 4, Winter 1986.

[20] A. A. Sullivan & J. A. Gelb, “Application of the Kalman Filter to Aided Inertial Systems”, Analytical Sciences Corporation.

[21] Military Handbook: transformation of Datums, Projections, Grids, and Common Corodinate Systems, MIL-HDBK-600008.

Index